iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 6 **Issue:** I **Month of publication:** January 2018

DOI: http://doi.org/10.22214/ijraset.2018.1019

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor :6.887
Volume 6 Issue I, January 2018- Available at www.ijraset.com

High Quality Real Time Image Inpainting

B. Sridhar[1], Bachu Srinivas[2], T. Kranthi Kumar[3], Velisetti Sadhana[4], T. Sridhar[5]
[1, 2]*Associate Professor, Department of ECE, GNITC, Hyderabad, Telangana, India.*
[3,4,5]*B.Tech IV[th] Year, Department of ECE, GNITC, Hyderabad, Telangana, India.*

Abstract*: *Image Inpainting, the method of adjusting a picture in an imperceptible shape, is as old as workmanship itself. The objectives and uses of inpainting are various, from the reclamation of harmed works of art and photos to the expulsion/substitution of chose objects. In this paper, we present a novel calculation for advanced inpainting of still pictures that endeavours to repeat the essential methods utilized by proficient rest orators. After the client chooses the districts to be re-established, the calculation consequently fills-in these locales with data encompassing them. The fill-in is done such that isophote lines touching base at the locales' limits are finished inside. Conversely with past methodologies, the strategy here presented does not require the client to determine where the novel data originates from. This is naturally done, accordingly permitting to all the while fill-in various districts containing totally unique structures and encompassing foundations. Also, no confinements are forced on the topology of the district to be inpainted. Utilizations of this procedure incorporate the reclamation of old photos and harmed film; expulsion of superimposed content like dates, subtitles, or attention; and the expulsion of whole protests from the picture like amplifiers or wires in enhancements.

Keywords: ***Image restoration, Inpainting, Isophotes, Anisotropic diffusion.***

I. INTRODUCTION

Image inpainting (aka context-aware fill, image synthesis, image completion, etc.) has recently become widely available as part of image manipulation tools. While image inpainting has been researched for quite some time it only recently has achieved a sufficient quality at an acceptable speed, allowing for integration in standard software. Nevertheless quality is still an issue and achieving sophisticated results for non-trivial image backgrounds still requires a significant amount of time [1, 2]. Even for rather low-resolution images (such as VGA) most approaches will not allow for proper inpainting at interactive frame rates. In painting, the strategy of altering a picture in an imperceptible frame is as antiquated as workmanship itself. The objectives and utilizations of in painting are various, from the rebuilding of harmed artistic creations and photos to the expulsion/substitution of chose objects. In this paper, we present a novel calculation for computerized in painting of still pictures that endeavors to recreate the essential strategies utilized by proficient rest speakers [1, 3, and 4]. After the client chooses the locales to be re-established, the calculation naturally fills-in these districts with data encompassing them. The fill-in is done such that isophote lines touching base at the locales' limits are finished inside. Conversely with past methodologies, the strategy here presented does not require the client to indicate where the novel data originates. This is consequently done (and fastly), along these lines permitting to at the same time fill-in various locales containing totally unique structures and encompassing foundations. What's more, no restrictions are forced on the topology of the locale to be in painted. Utilizations of this system incorporate the reclamation of old photos and harmed film; expulsion of superimposed content like dates, subtitles, or attention; and the expulsion of whole protests from the picture like mouthpieces or wires in enhancements [5-7]. The change of pictures in a way that is non-recognizable for an eyewitness who does not know the first picture is training as old as imaginative creation itself. Medieval work of art began to be re-established as ahead of schedule as the Renaissance, the thought processes being frequently as much to bring medieval pictures "a la mode" as to fill in any holes. This training is called modifying or in painting [7, 8]. The question of in painting is to reconstitute the absent or harmed parts of the work, keeping in mind the end goal to make it clearer and to re-establish its solidarity. Photos yet in addition to evacuate undesired articles and compositions on the picture, the districts to be in painted must be set apart by the client, since they rely upon his/her subjective determination. Here we are worried on the best way to "fill-in" the areas to be in painted, once they have been chosen. Checked districts are naturally loaded with the structure of their environment [9, 10].

II. PROPOSED METHODOLOGY

The block diagram of proposed method is shown in figure 1. It can be explained indetailly in below section.

A. Input Image

A sample image having a foreground & background can be used for inpainting by this algorithm. The given input image undergoes several changes during the process of inpainting.

B. Pre-processing

In imaging science, image processing is any form of signal processing for which the input is an image, such as a photograph or video frame; the output of image processing may be either an image or a set of characteristics or parameters related to the image. Most image-processing techniques involve treating the image as a two-dimensional signal and applying standard signal-processing techniques to it. Image processing usually refers to digital image processing, but optical and analog image processing also are possible. This article is about general techniques that apply to all of them. The acquisition of images (producing the input image in the first place) is referred to as imaging [5, 8].

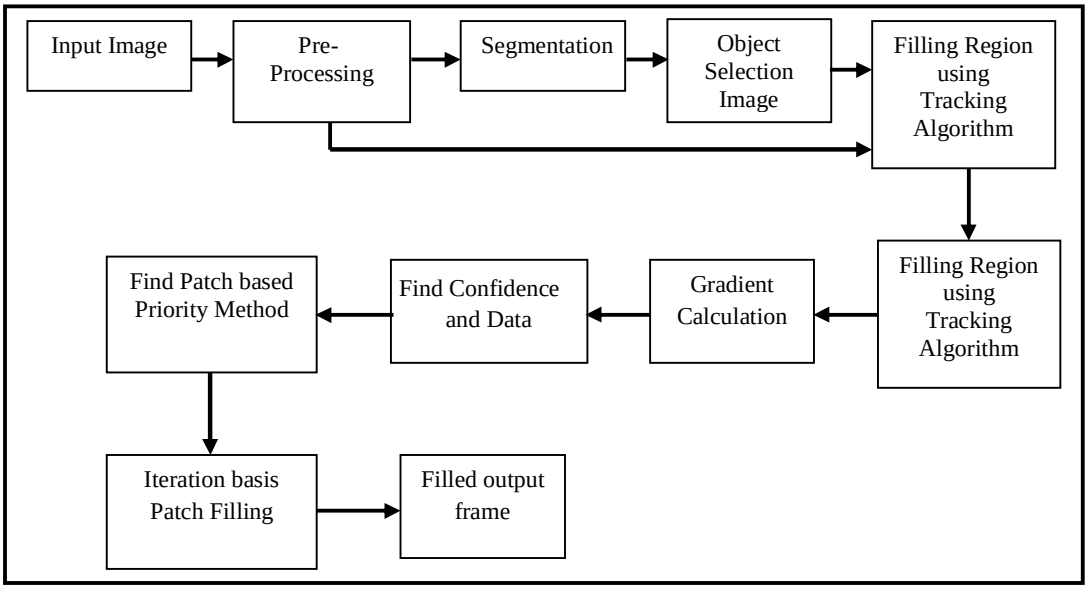


Figure 1: Block Diagram

Image processing refers to processing of a 2D picture by a computer. Basic definitions:

An image defined in the “real world” is considered to be a function of two real variables, for example, a(x,y) with a as the amplitude (e.g. brightness) of the image at the real coordinate position (x,y). Modern digital technology has made it possible to manipulate multi-dimensional signals with systems that range from simple digital circuits to advanced parallel computers. The goal of this manipulation can be divided into three categories:

Image processing (image in -> image out)

Image Analysis (image in -> measurements out)

Image Understanding (image in -> high-level description out)

An image may be considered to contain sub-images sometimes referred to as regions-of-interest, ROIs, or simply regions. This concept reflects the fact that images frequently contain collections of objects each of which can be the basis for a region. In a sophisticated image processing system it should be possible to apply specific image processing operations to selected regions. Thus one part of an image (region) might be processed to suppress motion blur while another part might be processed to improve colour rendition.

Most usually, image processing systems require that the images be available in digitized form, that is, arrays of finite length binary words. For digitization, the given Image is sampled on a discrete grid and each sample or pixel is quantized using a finite number of bits. The digitized image is processed by a computer. To display a digital image, it is first converted into analog signal, which is scanned onto a display. Closely related to image processing are computer graphics and computer vision. In computer graphics, images are manually made from physical models of objects, environments, and lighting, instead of being acquired (via imaging devices such as cameras) from natural scenes, as in most animated movies. Computer vision, on the other hand, is often considered high-level image processing out of which a machine/computer/software intends to decipher the physical contents of an image or a sequence of images (e.g., videos or 3D full-body magnetic resonance scans) [9, 10].

In modern sciences and technologies, images also gain much broader scopes due to the ever growing importance of scientific visualization (of often large-scale complex scientific/experimental data). Examples include microarray data in genetic research, or

real-time multi-asset portfolio trading in finance. Before going to processing an image, it is converted into a digital form. Digitization includes sampling of image and quantization of sampled values. After converting the image into bit information, processing is performed. This processing technique may be Image enhancement, Image restoration, and Image compression.

C. Segmentation

In computer vision, image segmentation is the process of partitioning a digital image into multiple segments (sets of pixels, also known as super pixels). The goal of segmentation is to simplify and/or change the representation of an image into something that is more meaningful and easier to analyse. Image segmentation is typically used to locate objects and boundaries (lines, curves, etc.) in images. More precisely, image segmentation is the process of assigning a label to every pixel in an image such that pixels with the same label share certain visual characteristics.

The result of image segmentation is a set of segments that collectively cover the entire image, or a set of contours extracted from the image (see edge detection). Each of the pixels in a region is similar with respect to some characteristic or computed property, such as colour, intensity, or texture. Adjacent regions are significantly different with respect to the same characteristic(s). When applied to a stack of images, typical in medical imaging, the resulting contours after image segmentation can be used to create 3D reconstructions with the help of interpolation algorithms like marching cubes.

D. Object Selection

Any part of the foreground in the image can be used for inpainting. The selected object part is removed from the original image.

E. Filling Region

The removed part of the original image in the foreground leaves an empty space in the background. This empty place in the background is recovered in this process. Once the object's contour is found, the determined object contour has to be tracked in successive video frames. Therefore, we apply a two phase contour tracking approach. In the first phase, a homography based contour tracking is used, while in the second phase the new contour is refined and adjusted regarding to the undesired object area. Typically, the contour points found will be rather planar. Thus, the relation of a set of tracked contour points between two consecutive frames may be described by a homography. For cases with non-planar backgrounds the motion between two frames can be expected to be very small. Thus, even for these situations, determining the homography may provide a good approximation of the contour movement. For homography determination, only the strongest contour points are tracked between two video frames by using a pyramid based motion detection. A Harris corner vote distinguishes between contour points with good and bad motion tracking properties. Finally, within several RANSAC iterations a reliable homography is determined. However, determination of the homographymay fails if only an insufficient number of contour points can be tracked reliably. This may happen if the point motion detection is inaccurate due to almost homogenous image content. The second phase for contour tracking is necessary to adjust the new contour positions (determined by the homography) more precisely to the real object contour. If no adjustment would be applied, a solely homography based tracking would accumulate small drift errors over time. Therefore, the fingerprint dissimilarity is used again. This time, the reference fingerprints are automatically defined outside the new contour (equally distributed) in the current frame. However, in order to ensure the performance required, the contour adjustment is performed for a randomly selected subset of the new contour points only. Foreach random point a virtual line perpendicular to the new contour starting outside and ending inside the new contour is defined. For each pixel, covered by this virtual line, the fingerprint dissimilarity is determined. Therefore, to avoid the influence of fingerprints at the opposite side of the contour, the new fingerprints in the direct neighbourhood are investigated only. The first pixel on the virtual line identified as undesired and followed by at least two successive undesired pixels defines a contour landmark for the virtual line. Finally, all contour points extracted in the first tracking phase are adjusted according to the contour landmarks in their neighborhood. The adjusted positions define the final and accurate contour. We wish to emphasize the aspect that the homography is calculated in the first tracking phase, while the final synthesis mask is determined after the second phase. Therefore, the homography is based on strong correspondences between contour points, while the mask is based on the actual object border. This separation is important as the homography determined will later be used to ensure a visual coherence of subsequent frames for non-linear camera movements. In contrast to the active snake approach of our previous approach the improved object tracking provides unique correspondences for contour points between successive video frames.

F. Constraints

More advanced constraints may be used to guide the inpainting algorithm providing improved visual results. Depending on the structure of the background or the desired and undesired visual elements, the final inpainting results may be optimized according to the expectations of the users.

However, in a real-time inpainting approach such as a Diminished Reality application, explicit user-defined constraints cannot easily be applied. The constraint definition itself requires a certain amount of time that might violate the real-time execution. Additionally, e.g., in a Diminished Reality application, users might not be willing to spend any significant amount of time on defining constraints. Thus, user-defined constraints as typically used for image inpainting, may not be directly applied to inpainting approaches. However, automated or semi-automated constraints may also be applied to video sequences. Simple structural constraints, e.g., lines or regular objects, which can be detected automatically in a real-time approach and therefore do not need to be specified by the user. Further, when inpainting is applied to video streams not requiring real-time performance or for slightly time shifted live broadcasts, even individual user-defined constraints may be used. Our pixel mapping approach allows seamless integration of constraints into the synthesis pipeline by simply extending the cost measure from (2) by an additional constraint cost:

$$\cos t_{\alpha}(p,\ f) = \alpha_{s}.\cos t_{spatial}(p,f) + \alpha_{a}.\cos t_{appear}(p,f) + \alpha_{c}.\cos t_{constr}(p,f),\ p \in T \tag{1}$$

G. Area Constraints

The most obvious form of inpainting constraints guides the algorithm to explicitly use or avoid image regions from the remaining image content. The algorithm is forced to use image content explicitly selected by the user, discarding content the user does not prefer.

A simple inverse importance map rates all undesired visual elements with an infinite high value while weighting the desired content with zero:

$$\overline{m}(q) = \begin{cases} 0, q \in S_{desired}, \forall_{q} \in S \\ 0, \text{else} \end{cases} \quad \text{and}$$

$$constr_{A}(p,\ f) = \overline{m}(f(p)) \tag{2}$$

The inverse importance map provides an infinite cost for conspicuous elements. The inpainting result using the area constraint doesnot contains repetitions of these elements. For video inpainting area constraints will be applied to the first frame only. However, subsequent frames will automatically avoid that content due to the frame to frame coherence based on an additional appearance cost term

H. Structural Constraints

Structural constraints may be used to explicitly preserve straight lines or strong borders during the inpainting process. Any number of individual structural constraints can be considered concurrently.

The constraint cost corresponds to the distance between the actual mapping position f(p) and the projected ideal position for a given point p.

The distance measure $d_{cs}(.)$ for infinite straight lines is defined by:

$$d_{cs}(p, cs) = (n_x, n_y, d).(p_x, p_y, 1)^T, \quad p \in I \tag{3}$$

with line normal $(n_x; n_y)$ for which $|(n_x,n_y)|=1$must hold, while the constant d defines the line distance to the origin. Similar to infinite lines, finite line constraints are represented by a slightly modified distance measure. For a finite line with endpoints P_0 and P_1, the distance is determined

for an arbitrary point p Є I by

$$d_{cs}(p, cs) = \begin{cases} (n_x, n_y, d).\begin{pmatrix} p_x \\ p_y \\ 1 \end{pmatrix}, 0 \le (p_1 - p_o)^T.(p - p_o) \le |p_1 - p_o|^2 \\ k, \text{otherwise} \end{cases} \tag{4}$$

The higher the penalty k, the lesser the application of mapping positions outside the finite constraint. Thus, if k is set to 1, mapping positions projecting outside the finite line are rejected. More complex structures such as splines and curves may be realized in the same manner as the distance function for straight lines.

Finally, the constraint filter function $g_{cs}(p,cs)$ needs to be defined:

$$g_{cs}(p,cs)=\begin{cases}1, \omega_{cs}(p,cs)\geq \omega_{cs}(p,cs') \ \forall cs' \in C_{cs}\setminus\{cs\}\\ 0, \text{elsewhere}\end{cases} \quad (5)$$

Without clustering, an application of the deviation would be useless for background images represented by more than one visual characteristic. Selection situation with individual background areas illustrating the necessity of data clustering. Finally, the cluster calculation has to be investigated. A wide variety of clustering algorithms exist providing individual clustering accuracies and run-time complexities as hierarchical, k-means, or distribution-based clustering approaches. However, concerning the real-time necessity we decided to apply a very simple but for our needs adequate clustering algorithm:

1) First, one fingerprint is selected randomly and defined as the center of the first cluster.
2) Afterwards, the remaining fingerprints are assigned to the new cluster if their distance is close enough to the cluster's center.
3) If still fingerprints are left, again one of them is selected randomly, defining the center of a new cluster. The algorithm then continues with step 2.

This algorithm is repeated several times and the clustering result with lowest maximal deviation within all clustersis finally accepted. Obviously the clustering method is very efficient and we found that the result is accurate enough forour needs. Another benefit is that the number of clusters do not have to be defined in advance as e.g., for k-means algorithms.

Thus, the more variations within the finger prints, the higher the number of final clusters. We defined ϒ=0:95 to detect undesired pixels if at least95 percent of the fingerprints reject the corresponding pixel. As data base we use the color image with three channels. Several individual fingerprint functions ϕ(.) have been evaluated and we found that a simple Gaussian filter with a large kernel provides sufficient results. However, due to performance reasons we apply a modified mean filter with one large and one small kernel size in combination with an integral image. This optimized mean filter needs eight integral image lookups only and performs much faster than a Gaussian filter while providing sufficient good results. Further, the entire fingerprint segmentation is combined with a multi resolution approach reducing the computational effort. The approach starts on the coarsest image resolution and forwards the result to the next finer layer. On this finer layer only the border pixels are investigated according to their fingerprint dissimilarity. Finally, a dilation filter is applied to removed possible tiny gaps. All pixels considered as undesired build the binary synthesis mask. A corresponding synthesis contour directly enclosing this mask is determined. According to the available processing time, the performance and accuracy of the approach can be tailored easily by adding or removing contour fingerprints or visual characteristics fi (e.g., color and texture channels). Obviously, the more information is used to determine the dissimilarities, the better the selection result. We found that the application of fingerprints with three elements, one for each image color channel, is a good compromise between segmentation accuracy and application speed. However, if accuracy is more important than computational time or if the computational hardware is powerful enough, we use a fourth fingerprint element by default. The fourth fingerprint element represents a simple textureness property determined by averaging the Scharr response for the gray scale image information.

I. Gradient Calculation

In the case of a given mapping function for the previous synthesis, the number of iterations to find a transformation for the new frame can be reduced significantly. A mapping initialization for each mask pixel is determined from the corresponding pixel in the previous image by application of the associated mapping. Therefore, the already determined homography is applied. An arbitrary mask pixel p in frame n is translated to the corresponding
pixel p' in frame n -1 by

$$p' = Hp$$

Thus, a sufficiently precise prediction of this mapping for the current frame can be calculated by applying theinverse homography:

$$\hat{m} = H^{-1}m'$$

Please note that the application of the inverse homography commonly will not result in a precise mapping for the current frame. This is due to the determination of the homography from the object contour, providing exact estimations for totally planar contours only while non-planar contours will introduce a certain error in the calculation of the homography. However, as the change in content between two successive video frames is low, this in accuracy is negligible as it will be adjusted in the subsequent cost minimization iterations. These subsequent cost minimization iterations also handle cases when the predicted initial guess ^ fn lies outside the camera frame. In this case our algorithm starts with a random choice and iteratively tries to reduce the matching cost.

J. Iteration Basis Patch Filling

Finding the optimal transformation function f is realized by starting with a rather rough guess of f, followed by a series of iterative refinements steps. At each iteration, the mapping for each target pixel is sought to be improved. Randomly, individual source positions are tested according to the local cost function and accepted whenever the local cost can be reduced. The improved matching is then propagated to neighboring positions in T. This approach is similar to that proposed by Barnes et al. and the approach we applied in our previous work. However, as these two approaches apply the dissimilarity measure of Wexler et al.or Simakov et al. respectively, each refinement needs a target information update of an entire image patch, requiring the application of the individual contribution from each patch followed by normalization. Our current approach directly updates only a single pixel and thus avoids expensive normalizations. We apply a multi resolution inpainting approach. Iterative refinement is applied on an image pyramid starting with a reduced resolution layer and increasing the image size until the original resolution has been reached. Depending on the mask size and frame dimension, typically between three and eight layers are used. The coarsest pyramid layer is found by the first layer in that no mask pixel exists that has a larger distance than three pixels to the inpainting border. The algorithm starts with an initial mappinguess fn-1 in the coarsest pyramid layer Ln-1and stops if an improved mapping fn-1 with minimal overall cost has been determined. This mapping is then forwarded to the next pyramid layer Ln-2 and is used as the new initialization ^ fn_2. Again, after a series of iterations within the current layer, the optimized transformationFn-2 is forwarded as the initialization of the next layer until the final layer L0 (providing the highest resolution) has been reached and processed. The applied image pyramid allows the covering of visual structures with individual frequency, speeds up the mapping convergence and significantly reduces the chance that the algorithm gets trapped by some local minima. Barnes et al. applied an information propagation improving the overall process significantly. However, originally the propagation idea in the context of image inpainting had been proposed by Ashikhmin and Demanet et al.. Our work applies a comparable propagation to benefit from the significant speedup opportunity. However, instead of propagating the position of entire image patches, our approach forwards single pixel mapping positions only. As in our previous approach, the iterative refinement approach as developed in this work benefits from multi-core CPUs. Thus each subset Ti can be processed by an individual thread in parallel. In our previous work, static frame subsets have been applied, restricting propagation of mapping information to be within individual subsets. This isolated refinement may produce undesired synthesis blocks in the final image as the mapping exchange between subsets is restricted to the subset borders. Further, damped propagation may reduce synthesis performance. Our current approach applies random subsets changing between forward and backward propagation while the subsets' size stays constant. Neighboring subsets propagate their mapping in opposite directions starting from a common start row. These start rows are optimized explicitly before any refinement iteration is processed, so that neighboring subsets have access to the same mapping information. Successive refinements toggle between forward and backward propagation, have changing target subsets, and start from common (already refined) rows. The random subsets have a significant impact on the final image quality and convergence performance as information propagation is applied to the entire synthesis mask, rather than limited to the sub-blocks. The comparison between our previous and current approach is presented.

K. Filled Output Frame

The object selected in the foreground is finally removed completely and the background area of the foreground image is restored.

In this section a detailed comparison between well-known benchmark images of recent state-of-the-art approaches, our previous approach [18] and our current image inpainting algorithm is provided. The Figs. 15 and 16 depict that our proposed approach is able to recover simple image structures with visual results comparable to related approaches while processing several magnitudes faster. Our previous approaches provided undesired blurring effects for heterogeneous images. Although the patch similarity approaches of Wexler et al. is known as tending to minor blurring artifacts, the blurring of our previous approach might be slightly larger due to the real-time performance constraints. In contrast, the pixel mapping approach introduced in this paper cannot result inblurring artefacts as the final inpainting result is not determined by superimposing image patches. The image inpainting result of our approach for an image with more than 2.6 mega pixels.

L. Performance Issues

We measured the performance of our approach with a laptop Intel i7-3840QM Core with 2.8 GHz running Windows 7. The implementation is realized in C++. We measured the entire performance for the video inpainting(including tracking, line detection, reference frame creation and video synthesis) applying a video resolution of640 * 480 pixels. Table 1 provides the detailed performance values showing the dependency between performance and the amount of pixels to be removed in each frame. The table shows a comparison between the Standard video inpainting and the video inpainting applying a line constraint. The line constraint is

determined by application of a Hough line detector. The measurements show that the video inpainting approach applying a line constraint needs approx. 30 percent more time compared to the standard video inpainting approach. The performance loss is due to the additional line detection and the additional computational time for the constraint cost as defined in (8). Therefore, our system reaches between 24 and 41 fps providing the visual quality results.

III.RESULTS AND DISCUSSIONS

The results of real time image inpainting are shown in figure 2 to figure 12.

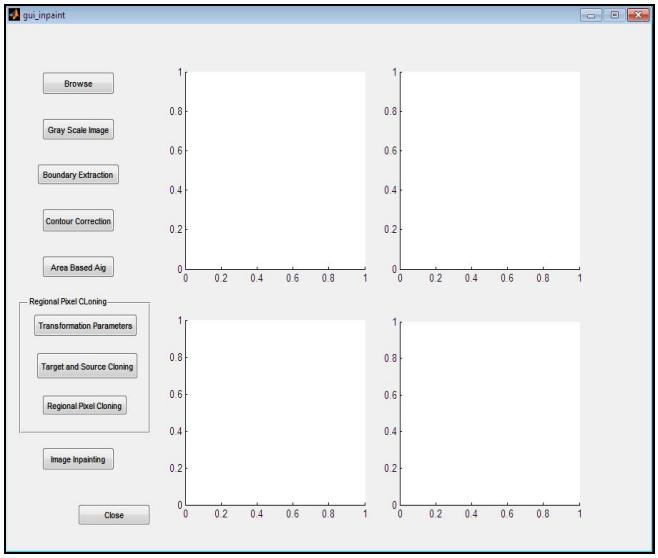


Figure 2: GUI of Inpainting

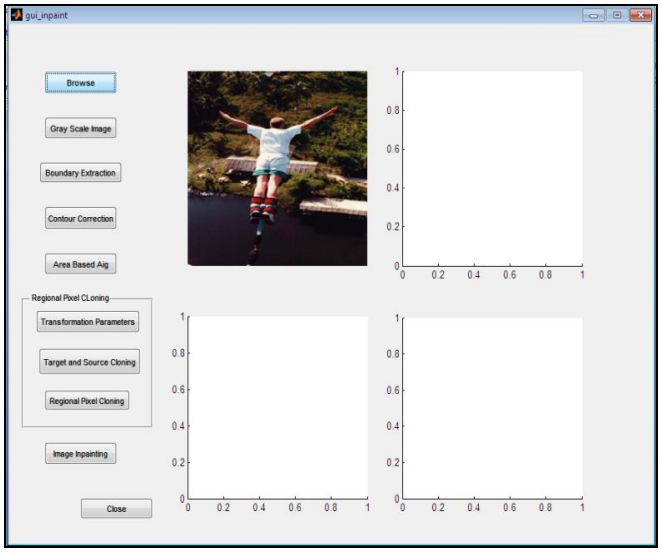


Figure 3: Input Image

GUI of inpainting contains various tools like browse, grayscale image, boundary extraction, countour correcton, area based aig, regional pixel cloning (transformation parameters, target and source cloning, regional pixel cloning), image inpainting. The desired image is selected from the browse option. It is placed at the top of the left side of inpaint window. The screen shown after the selection will be as shown above.

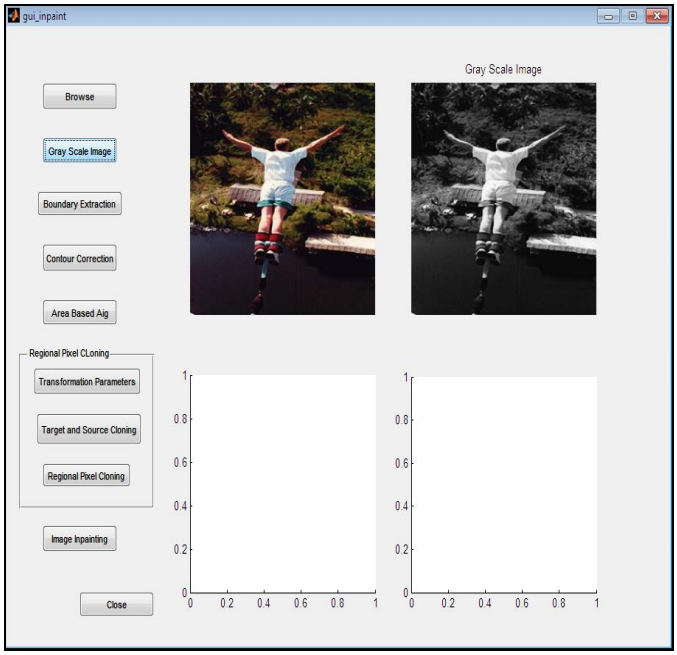


Figure 4: Gray Scale Image

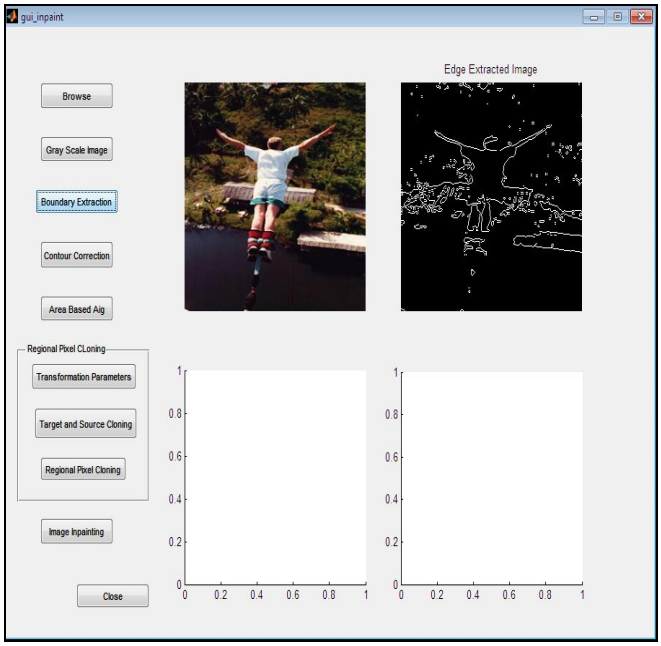


Figure 5: Boundary Extracted Image

After clicking on gray scale image option, the image is converted to gray scale as shown in the above window. For better identification of foreground object, we convert the image to gray scale.

L. Boundary Extraction

In the boundary extraction, the edge of the object is pictorized and is achieved in the inpaint window as shown. By this, the inpaint tool comes to know which part of the image is to be removed.

M. Counter Correction

In countour correction, the area of the image where the object is removed is extracted in the form of Region of Index.
The outputs are received as in above window.

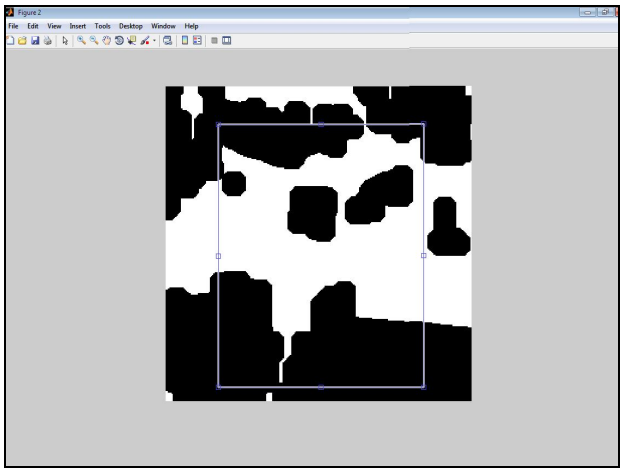


Figure 6: Edge Extracted Image

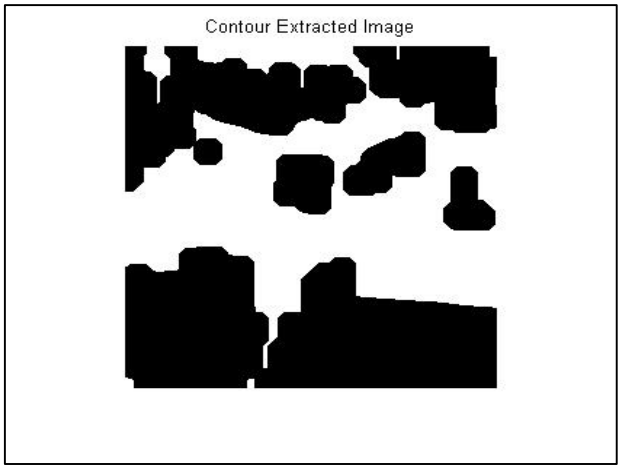


Figure 7: Contour Extracted Image

N. Area Based Aig

In area based aig, the sketch of the object is received or shown in the adjacent window as shown above. In the next step, removal of the object takes place.

O. Regional Pixel Cloning

1) Transformation Parameters: In regional pixel cloning, the LSMR transformation parameters of the image are obtained.

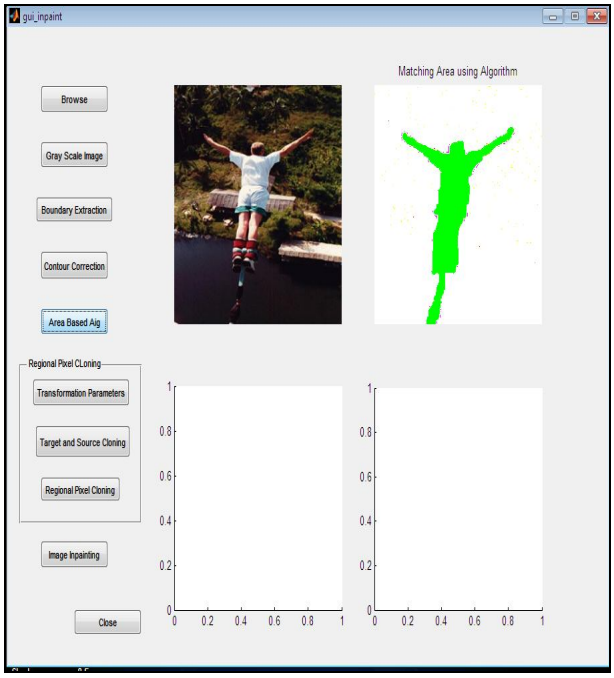


Figure 8: Area Based Aig

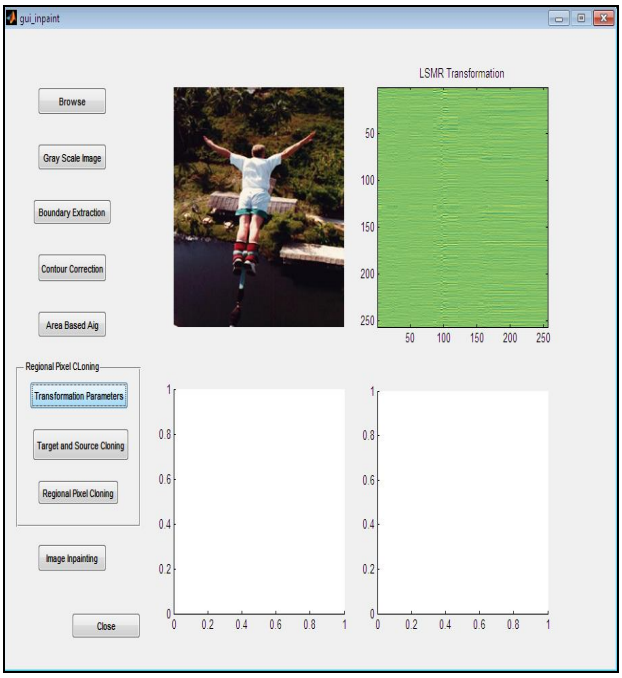


Figure 9: Transformation Parameters

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor :6.887
Volume 6 Issue I, January 2018- Available at www.ijraset.com

Figure 10: ROI Image

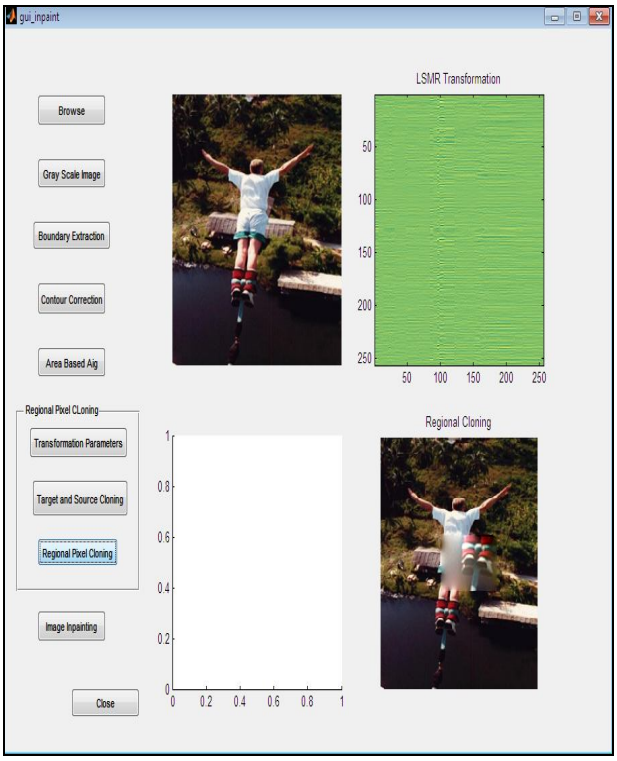


Figure 11: Regional Pixel Cloning

This is the ROI image. Cloning is done with the parameters and inpainting is done. Region of cloning is done finally and the appeared window is as shown above. In the fourth plot, you can see the cloned image.

P. Image Inpainting

Finally in this step, we get the inpainted image from the original input image. The object from the foreground is removed, by recreating the space in the background. Hence the received image is the inpainted image.

Q. Applications

1) Content-based image retrieval
2) Machine Vision
3) Medical imaging
4) Locate tumors and other pathologies
5) Measure tissue volumes
6) Diagnosis, study of anatomical structure
7) Surgery planning
8) Virtual surgery simulation
9) Intra surgery navigation
10) Pedestrian detection
11)
12) Brake light detection
13) Locate objects in satellite images (roads, forests, crops, etc.)
14) Recognition Tasks
15)
16) Fingerprint recognition

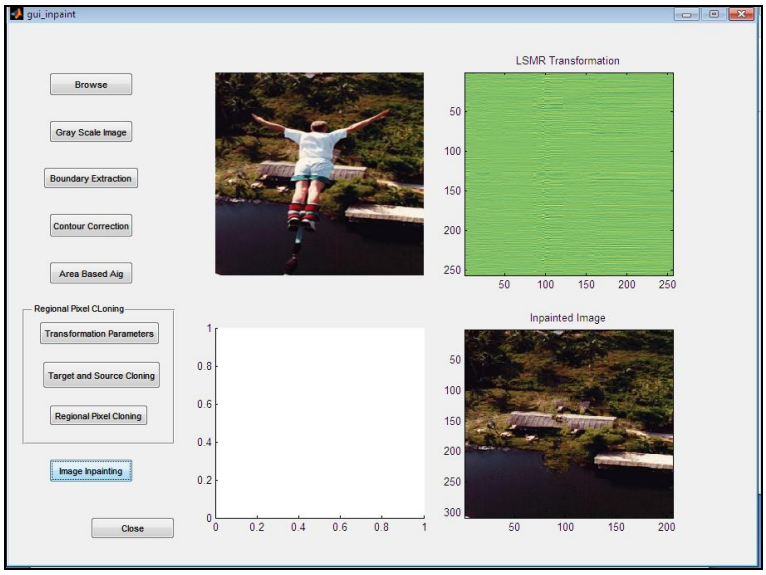


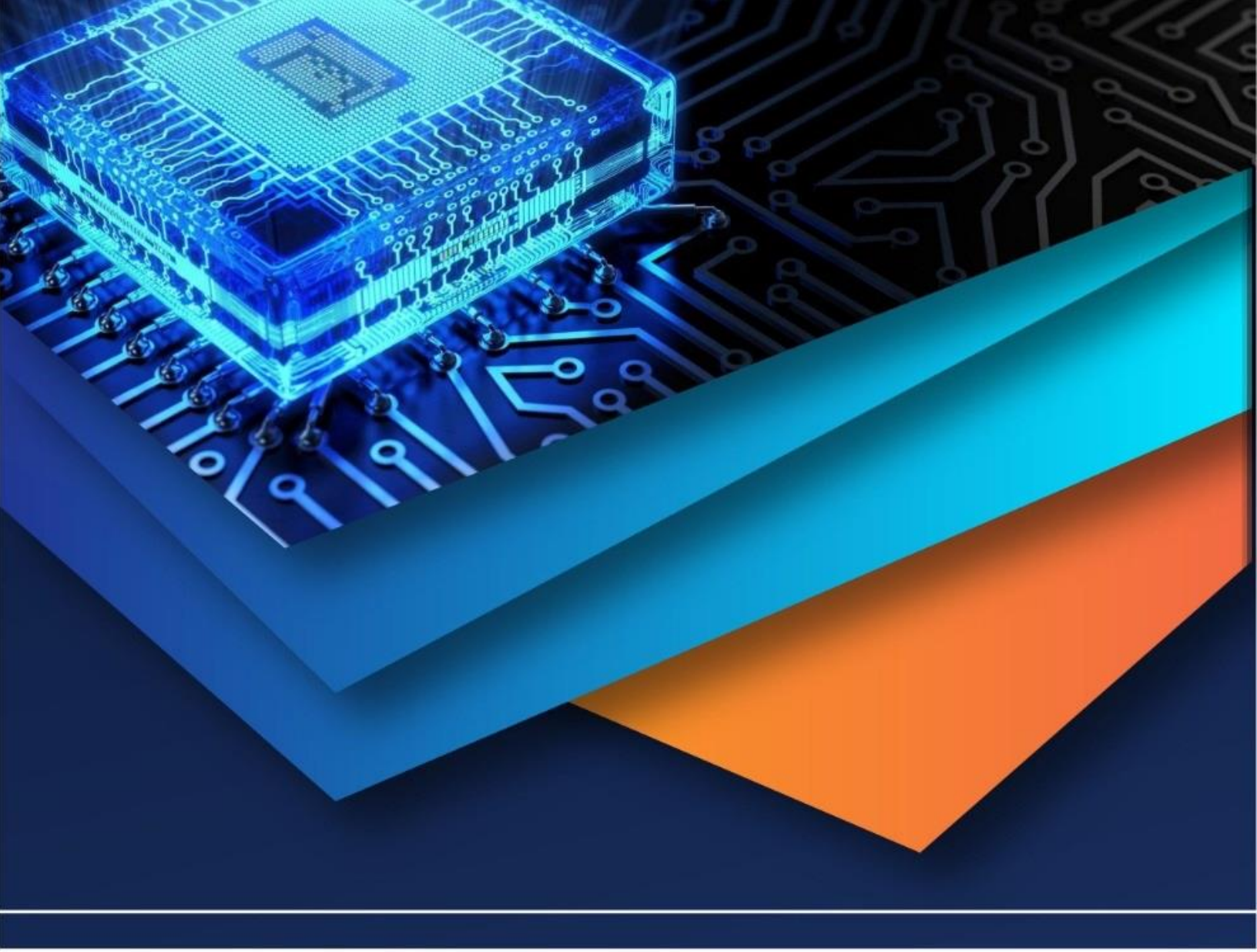
Figure 12: Inpainted Image

IV. CONCLUSIONS

In this paper we presented PixMix, our pixel based approach to real-time image inpainting. We further showed how this approach enables the realization of Diminished Reality. Additionally, a real-time capable object selection and tracking algorithm has been introduced. Our inpainting approach considers adjusting between the spatial and the appearance term of the cost work keeping in mind the end goal to give ideal inpainting comes about. The overall results showed fewer artifacts than other approaches, allowing for high-quality image inpainting. We achieved this by extending the overall cost function by a frame-to frame coherence term and by applying a homography as a first guess for the mapping in the next frame providing a significantly better initialization. In our future work, we are planning to extend the homography based approach to arbitrary 3D objects. While the actual user study has already been conducted, it requires further analysis of the data obtained.

REFERENCES

[1] Criminisi, P. Perez, and K. Toyama, “Region Filling and ObjectRemoval by Exemplar-Based Image Inpainting,” IEEE Trans.Image Processing, vol. 13, no. 9, pp. 1200-1212, Sept. 2004.
[2] I. Drori, D. Cohen-Or, and H. Yeshurun, “Fragment-Based ImageCompletion,” Proc. ACM SIGGRAPH, pp. 303-312, 2003.
[3] J. Sun, L. Yuan, J. Jia, and H.-Y. Shum, “Image Completion withStructure Propagation,” Proc. ACM SIGGRAPH, pp. 861-868,2005.
[4] D. Simakov, Y. Caspi, E. Shechtman, and M. Irani, “SummarizingVisual Data Using Bidirectional Similarity,” Proc. IEEE Conf. ComputerVision and Pattern Recognition (CVPR ’08), 2008
[5] A. Bugeau, M. Bertalm_ıo, V. Caselles, and G. Sapiro, “A ComprehensiveFramework for Image Inpainting,” IEEE Trans. ImageProcessing, vol. 19, no. 10, pp. 2634-2645, Oct. 2010.
[6] J. Jia, T.-P. Wu, Y.-W. Tai, and C.-K. Tang, “Video Repairing: Inferenceof Foreground and Background under Severe Occlusion,”Proc. IEEE Conf. Computer Vision and Pattern Recognition(CVPR ’04), vol. 1, pp. 364-371, June 2004.
[7] J. Jia, Y.-W. Tai, T.-P. Wu, and C.-K. Tang, “Video Repairing underVariable Illumination Using Cyclic Motions,” IEEE Trans. PatternAnalysis and Machine Intelligence, vol. 28, no. 5, pp. 832-839, May2006
[8] K.A. Patwardhan, G. Sapiro, and M. Bertalmio, “Video Inpaintingunder Constrained Camera Motion,” IEEE Trans. Image Processing,vol. 16, no. 2, pp. 545-553, Feb. 2007
[9] Y. Shen, F. Lu, X. Cao, and H. Foroosh, “Video Completion forPerspective Camera under Constrained Motion,” Proc. 18th Int’lConf. Pattern Recognition (ICPR ’06)), vol. 3, pp. 63-66, June 2006
[10] T. Shiratori, Y. Matsushita, X. Tang, and S.B. Kang, “Video Completionby Motion Field Transfer,” Proc. IEEE Conf. Computer Visionand Pattern Recognition (CVPR ’06), vol. 1, pp. 411-418, June 2006.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY