Scientific Journal of Impact Factor (SJIF): 3.134

e-ISSN (O): 2348-4470
p-ISSN (P): 2348-6406

International Journal of Advance Engineering and Research Development

Volume 3, Issue 1, January -2016

Reversible Texture Synthesis through Stego Technique

Patil Tushar [1], Sonawane Atul [2], Sonawane Pankaj [3] ,Bankar Atul [4] Muzaffar Shabad [5]

[1,2,3,4,5]Department of Computer Engineering, Sinhgad Academy of Engineering

Abstract — *We are designing a technique of steganography using a reversible texture synthesis. A texture synthesis means re-sampling of a smaller texture image which synthesizes a new texture image with a similar local appearance and arbitrary size. We are combining the texture synthesis process with steganography to hide secret messages. Instead of using an existing cover image to hide messages, our technique hides the source texture image and embeds secret messages with the help of texture synthesis. By using this approach we can extract secret messages and source texture from stego synthetic texture. Our technique has three distinct advantages. First, our scheme offers the embedding capacity that is proportional to the size of the stego texture image. Next, a steganalytic algorithm can not defeat our steganographic approach. Third, the reversible capability used in our scheme allows us to recover the source texture. Experimental results can verify that our proposed algorithm can give various numbers of embedding capacities, produce a visually appreciable texture images, and recover the source texture.*

Keywords: Data hiding, Data embedding and Extracting, Patch-based approach, reversible texture steganography, texture synthesis.

I. INTRODUCTION

In the most recent decade numerous advances have been made in the territory of computerized media, and much concern has emerged with respect to steganography for advanced media. Steganography a solitary strategy for data hiding procedures. It embedds messages into a host medium keeping in mind the end goal to disguise mystery messages so as not to stimulate suspicion by a meddler [2]. A commonplace steganographic application incorporates undercover correspondences between two gatherings whose presence is obscure to a conceivable assailant and whose achievement relies on upon distinguishing the presence of this correspondence. All in all, the host medium utilized as a part of steganography incorporates significant computerized media, for example, advanced picture, content, sound, feature, 3D model , and so forth. An expansive number of picture steganographic calculations have been researched with the expanding ubiquity and utilization of advanced pictures [7].

Most picture steganographic calculations embrace a current picture as a spread medium. The cost of inserting mystery messages into this spread picture is the picture mutilation experienced in the stego picture. This prompts two disadvantages. In the first place, subsequent to the measure of the spread picture is settled, the more mystery messages which are installed take into consideration more picture twisting[4]. Hence, a trade off must be come to between the implanting limit and the picture quality which brings about the constrained limit gave in any particular spread picture. Review that picture steganalysis is a methodology used to distinguish mystery messages covered up in the stego picture. A stego picture contains some bending, and paying little heed to how minute it is, this will meddle with the characteristic components of the spread picture [3]. This prompts the second downside on the grounds that it is still conceivable that a picture steganalytic calculation can overcome the picture steganography and hence uncover that a concealed message is being passed on in a stego picture [1].

In this paper, we propose a novel methodology for steganography utilizing reversible surface union. A surface blend process re-tests a little composition picture drawn by a craftsman or caught in a photo keeping in mind the end goal to combine another composition picture with a comparative nearby appearance and subjective size. We weave the composition amalgamation process into steganography covering mystery messages and additionally the source surface. Specifically, as opposed to utilizing a current spread picture to shroud messages, our calculation disguises the source composition picture and installs mystery messages through the procedure of surface union. This permits us to remove the mystery messages and the source composition from a stego manufactured surface[9]. To the best of our insight, steganography exploiting the reversibility has ever been exhibited inside of the writing of composition amalgamation.

Our methodology offers three favorable circumstances. To begin with, since the surface union can combine a self-assertive size of composition pictures, the implanting limit which our plan offers is corresponding to the measure of the stego composition picture[5]. Also, a steganalytic calculation is not prone to thrashing this steganographic methodology since the stego composition picture is made out of a source surface instead of by adjusting the current picture substance. Third, the reversible ability acquired from our plan gives usefulness to recuperate the source composition[11]. Since the recouped source composition is the very same as the first source surface, it can be utilized to continue onto the second round of mystery messages for steganography if necessary. Test results have confirmed that our

proposed calculation can give different quantities of implanting limits, deliver outwardly conceivable surface pictures, and recoup the source composition. Hypothetical investigation shows that there is an immaterial likelihood of separating our steganographic methodology, and the plan can oppose a RS steganalysis attack [12].

II. LITERATURE REVIEW

1) Exploring steganography: Seeing the unseen

AUTHORS: N. F. Johnson and S. Jajodia,

Steganography is the technique of hiding information in such a way that it could prevent the detection of hidden messages [1]. It includes a vast data of secret communications methods that hides the message's existence itself. These methods include hidden inks, microdots, character manipulation, digital signatures, covert channels, and spread spectrum communications. Steganography and cryptography are brothers in the spy craft family: cryptography encrypts a message so that it cannot be understood by the hacker while steganography hides the message so their existence cannot be predicted by anyone. In this article the writers discuss images and how to hide information within the images, and later they discuss about results obtained from extracting available steganographic software. They debate about steganography by itself does not provide privacy, but neither does simple encryption. If these methods are combined, more stronger encryption methods can be resulted and obtained. If an encrypted message is obstructed, the hacker knows that the text is an encrypted message. But with steganography, the hacker may not know that a hidden message even if is there or not. For a brief look at how steganography evolved, there is s article titled "Steganography: Some History."

2) Hide and seek: an introduction to steganography,

AUTHORS: N. Provos and P. Honeyman,

Although people have hidden secrets in plain sight-now called steganography throughout the past times, the recent improvement in computational technology and power has driven it to the forefront of today's security approaches [2]. Essentially, the information-hiding method in a steganographic system starts by identifying a cover medium's redundant bits (those that can be modified without destroying that medium's probity or honesty). The embedding method creates a stego medium by replacing these no longer useful bits with data from the hidden message. This article discusses existing steganographic systems and presents latest research in detecting them by statistical steganalysis. Here, we present latest research and discuss the real life application of detection methods and the mechanisms for getting around them.

3) Information hiding-a survey

AUTHORS: F. A. P. Petitcolas, R. J. Anderson, and M. G. Kuhn,

Images, video, and digital audio are increasingly developed with distinguishing but imperceptible marks, which may contain a concealed copyright trademark or serial number or even help to avoid unauthorized copying directly or indirectly. It is well known that military conversation systems make increasing use of traffic security techniques which, instead of merely hiding the content of a message using data encryption, try to conceal its sender [3]. Similar techniques are being used in some mobile phone devices and techniques derived for digital elections. Criminals try to use whatever traffic security characteristics are provided knowingly or otherwise in the present communications systems, and police try to minimize and restrict their usage. However, many of the methods proposed in this young and rapidly evolving field can trace their history, and many of them are surprisingly easy to implement. In this article, writers try to provide an overview of the area, of what we may know, what may work, what may not, and what are the interesting points for research.

4) A high-capacity steganographic approach for 3D polygonal meshes,

AUTHORS: Y.-M. Cheng and C.-M. Wang,

Authors are presenting a high-capacity steganographic approach for 3 dimensional (3D) meshes with polygonal shapes. Authors use the representation information of a 3D model to embed and paste secret messages. Their method successfully combines both the spatial domain and the representation domain for steganography [4]. In the spatial domain, each vertex point of a 3D mesh with polygonal shape can be represented by at least 3 bits using a modified multi-level embed procedure (MMLEP). In the representation domain, the representation order of polygons and vertices and even the topology of polygons can be represented with an average of 6 bits per vertex using the proposed representation rearrangement(modification) procedure (RRP), which is quite interesting. Experimental results show that the proposed technique is effective and secure, has high capability and limited distortion, and is resistant against transformations. Their technique is efficient and probable alternative to other steganographic approaches.

5) Line-based cubism-like image—A new type of art image and its application to lossless data hiding

AUTHORS: S.-C. Liu and W.-H. Tsai,

A new technique of mixing art image generation and data hiding for various information-hiding approaches are proposed [5]. It is also used to improve the camouage effect. Here writers propose a new type of computer art, called line-based Cubism-like image, which keeps a properties of the Cubism art-abstraction by prominent lines and regions from multiple viewpoints. In the development process with an input image, writers detect particular line segments in an image and

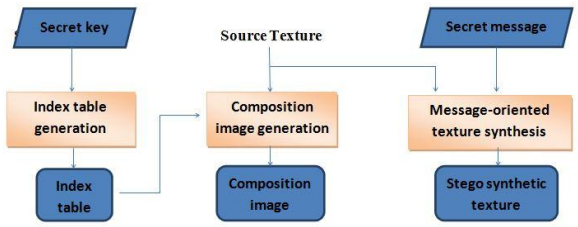
rearrange it to form an abstract area-type art image of the Cubism type. Data hiding with the minimal alteration(distortion) is done skillfully during the method of recoloring the areas in the developed art image by shifting the pixels' colors for the minimum amount of 1 and the average colors of the regions are not changed. Writers have used rounding of property in integer-valued color computation. Thus, the proposed data hiding method is proved by theorems to be reversible, and thus useful for lossless recovery of the cover art picture from the stego-image.

III. PROPOSED SYSTEM

First we illustrate our proposed method. The basic unit used for our steganographic texture synthesis is a "patch". A patch can be defined as an image block of a source texture and its size can be is specified by user. The size of a patch can be denoted by its width (Pw) and height (Ph.). A patch consists of the central part and an outer part, where the central part is the kernel region with size of Kw*Kh , and the part surrounding the kernel region is the boundary region with the depth (Pd). Next is the kernel block. A source texture has the size of Sw*Sh. We can subdivide the texture into a number of kernel blocks which are non overlapped, each of which has the size of Kw*Kh. Let KB is the collection of all kernel blocks, and "KB" represents the number of elements in this set. We have the indexing for each source patch kbi, i.e., KB= (kbi,i = 0 to KB -1). We will expand a kernel block with the depth Pd at each side to produce a source patch. This expanding process will overlap its neighbor block. If a kernel block is located around the boundary of a source texture, we operate the boundary mirroring using the kernel block's symmetric contents to produce the boundary region Similar to the kernel block, we can denote SP as the collection of all source patches and SPn=SP is the number of elements in the set SP. We can have the indexing for each source patch spi, i.e., SP= (spi,i = 0 to SP-1). We have the source texture with the size of Sw*Sh, we can calculate the number of source patches SPn by using the formula "SPn=Sw/Kw*Sh/Kh". Where Kw*Kh is the kernel block size. In our project, we assume the size of the source texture is a factor of the size of the kernel block to ease the complexity. Our steganographic texture synthesis algorithm needs to generate candidate patches when synthesizing synthetic texture. We employ a window PwPh and then travel the source texture (Sw*Sh) by shifting a pixel each time following the scan-line order. Let CP= (cpi, i=0,1...., CPn-1) represent the set of the candidate patches where CPn=CP denotes the number of elements in CP. We can derive CPn using the formula \CPn= CP= (Sw-Pw+1) (Sh-Ph+1)". This formula is used to calculate CPn. When creating a candidate patch, we need to check each candidate patch should be unique; otherwise, we will get an incorrect private message. In our implementation, we have used a flag mechanism. We first verify whether the original source texture has any duplicate candidate patches. For a duplicate candidate patch, we set the flag on for the first one. For the rest of the duplicate candidate patches we set the flag of to ensure the uniqueness of the candidate patch in the candidate list.

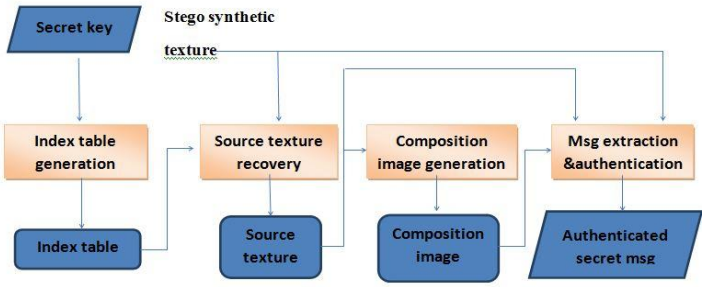
IV. PROPOSED SYSTEM MODULES

- **Message Embedding Procedure**



  - **Index Table Generation Process**
  - **Patch Composition Process**
  - **Message-oriented Texture Synthesis Process**

- **Capacity Determination**
- **Message Extracting Procedure**



MODULES DESCRIPTON:-

Message Embedding Procedure:

It has following three sub steps

Index Table Generation Process:

The primary procedure is the list table generation where we deliver a list table to record the source's area patch set SP in the synthetic texture. The file table permits us to get to the engineered surface and recover the source composition totally. Such a reversible implanting style uncovers one of the significant advantages our proposed calculation offers.

We first determine the dimensions of the index table (*Tpw*×*Tph*). Given the parameters *Tw* and *Th* , which are the width and the height of the synthetic texture we intend to synthesize, the number of entries in this index table can be determined using where *TPn* denotes the number of patches in the stego synthetic texture.

TPn= TPw*TPh= [Tw-Pw/Pw-Pd+1][Th-Ph/Ph-Pd+1]

Then we define the first-priority position L1 and the second-priority position L2, for two types of priority locations where L1 and L2. L1 and L2 can be calculated by using the formula L1=(Tpw 2/2)*(Tph-2/2)" and L2=(Tpw - 2/2)*(Tph-2/2). It represent the _rst and second priority locations respectively. The number of patches SPn subdivided from the source texture, the patch distribution is done by distributing patches perfectly on the first-priority positions before pasting patches to the second-priority positions. On the basis of the resolution of the synthetic texture, we will have two cases: the sparse distribution and dense distribution. Its explanation is give below. When the number of source patches is less than or equal to the number of the first-priority positions (SPn is less than or equal to L1) then the patch will be distributed sparsely. when the number of source patches is greater

than the first-priority position (SPn is greater than L1), the patch will be distributed densely. Initial value for the index table is the -1 for each entry; this shows that the table is blank. Now, we have to re-assign values when we distribute the source patch ID in the synthetic texture. In the implementation, we employ a random seed for patch ID distribution, which increases the security of our steganographic algorithm making it more difficult for malicious attackers to extract the source texture. As a result, the index table will be scattered. In this index table, the entries with non-negative values indicate the corresponding source patch ID subdivided in the source texture, while those entries with the value of -1 represent that the patch positions will be synthesized by referring to the secret message in the message-oriented texture synthesis. Taking the above condition into consideration, we can now use the random seed Rs to disarrange the ID of the source patches subdivided in the source texture.

Patch Composition Process:

The second process of our algorithm is to paste the source patches into a workbench to produce a composition image. First, we establish a blank image as our workbench which is equal to the size of the workbench is equal to the synthetic texture. By referring to the source patch IDs stored in the index table, we then paste the source patches into our workbench. During the pasting process, if no overlapping of the source patches is happened, we paste the source patches directly into the our workbench, however, if pasting locations cause the source patches to overlap each other, we employ the image quilting technique [17] to reduce the visual artifact on the overlapped area.

Message-oriented Texture Synthesis Process:

Now we have index table and a composition image, and we have pasted source patches directly into the workbench. We will embed our secret message via the message-oriented texture synthesis to produce the final stego synthetic texture. The three main differences between our proposed message-oriented texture synthesis and the conventional patch-based texture synthesis are as follows. The first difference is the shape of the overlapped area. In the conventional synthesis process, an L-shape overlapped area is used to determine the similarity of every candidate patch. But in our algorithm the shape of the overlapped area varies because we have pasted source patches into the workbench. The second difference is related to selection of candidates. In conventional texture synthesis, a threshold rank is usually given so that the patch can be randomly selected from candidate patches when their ranks are smaller than the given threshold. But our algorithm selects appropriate patches by considering secret messages. The source texture being converted into a number of source patches has been pasted as part of the contents in the large synthetic texture. In addition, the output large texture has been concealed with the secret message. While the conventional texture synthesis method has an L-shaped overlapped region,

our approach may have another four shapes of the overlapped area. Assume that the texture synthesis is carried on using the scan-line order. The texture area reveals a normal L-shape of an overlapped region. However, when a nearby applied source patch has occupied the right side of the working location, this leads to a \downward U-shape" of the overlapped area. In addition, if a nearby pasted source patch has occupied the bottom side, this leads to a \rightward U-shape" of the overlapped area. If a nearby pasted source patch has occupied the lower right corner of the working location, this leads to a disjointed overlapped area containing an \L shape and a small but isolated par. Finally, if two nearby pasted source patches have occupied the right and bottom side of the working location, this will contribute to an "O-shape" of the overlapped area. For each candidate patch within the candidate list, one of the five shapes of overlapped area described above will occur when referring to the synthesized area in the working location. Thus, we can compute the mean square error (MSE) of the overlapped region between the synthesized area and the candidate patch. After all MSEs of the patches in the candidate list are determined, we can further rank these candidate patches according to their MSEs. Once the ranks of all candidate patches are determined, we can select the candidate patch where its rank equals the decimal value of an n-bit secret message. In this way, a segment of then-bit secret message has been concealed into the selected patch to be pasted into the working location. In our implementation, we employ a simple but effective image quilting technique [17]to reduce the visual artifact encountered in the overlapped area.

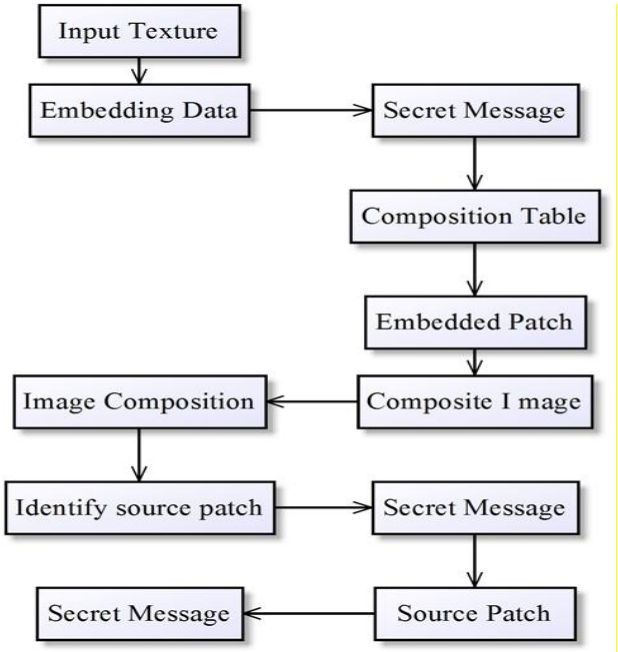
Capacity Determination:

The embedding capacity our algorithm can offer is related to the capacity in bits that can be concealed at each patch (BPP, bit per patch), and to the number of embeddable patches in the stego synthetic texture (EPn). Each patch can conceal at least one bit of the secret message; thus, the lower bound of BPP will be 1, and the maximal capacity in bits that can be concealed at each patch is the upper bound of BPP, as denoted by BPPmax. In contrast, if we can select any rank from the candidate list, the upper bound of BPP will be log2(CPn) . The total capacity (TC) Of our algorithm is the multiplication of BPP and EPn. The number of the embeddable patches is the difference between the number of patches in the synthetic texture (TPn) and the number of source patches subdivided in the source texture (SPn).

Message Extracting Procedure:

The message extraction process at the receiver side involves generating the index table, retrieving the source texture, performing the texture synthesis, and extracting and authenticating the secret message hided into the stego synthetic texture. The extracting procedure contains four steps. The secret key held in the receiver side, the same index table as the embedding procedure can be generated. The next step is the source texture recovery. Each kernel region with the size of Kw*Kh and its corresponding order with respect to the size of Sw*Sh source texture can be retrieved by referring to the index table with the dimensions Tpw*Tph. We can arrange kernel blocks based on their order, thus retrieving the recovered source texture which will be exactly the same as the source texture. In the third step, we apply the composition image generation to paste the source patches into a workbench to produce a composition image by referring to the index table. This generates a composition image that is identical to the one produced in the embedding procedure. The _nal step is the message extraction and authentication step, which contains three sub-steps. The _rst sub-step creating a candidate list based on the overlapped area by referring to the current working location. This substep is the same as the embedding procedure, producing the same number of candidate lists and their corresponding ranks. The second sub-step is the match-authentication step. Given the current working location Cur (WL) on the workbench, we refer to the corresponding stego synthetic texture at the same working location Stg (WL) to determine the stego kernel region SKw*SKh. Then, based on this stego kernel region, we search the candidate list to determine if there is a patch in the candidate list where its kernel region is the same as this stego kernel region. If this patch is available, we refer to it as the matched patch, and denote it as MKw*MKh. Clearly, we can locate the rank R of the matched patch, and the given rank represents the decimal value of the secret bits we conveyed in the stego patch when operating the texture synthesis in the message embedding procedure. However, if we cannot disclose any matched patch in the candidate list where the kernel region is the same as the stego kernel region, it means that the stego kernel region has been tampered with, leading to a failure of the message authentication. In this way, we can authenticate and extract all of the secret messages that are hidden in the stego synthetic texture patch by patch. Our method is resistant against malicious attacks as long as the contents of the stego image are not changed. With some side information, for example, our scheme can survive the attacks of the image mirroring or image rotation by 90, 180, or 270 degrees. Nevertheless, if malicious attacks lead to alteration of the contents of the stego texture image, the message authentication step will justify the authenticity of the secret messages.

V. SYSTEM ARCHITECTURE



VI. CONCLUSION

This paper proposes a steganographic algorithm using reversible texture synthesis. Given an original source texture, our scheme can produce a large stego synthetic texture concealing secret messages. To the best of our knowledge, we are the first that can exquisitely weave the steganography into a conventional patch-based texture synthesis. Our method is novel and provides reversibility to retrieve the original source texture from the stego synthetic textures, making possible a second round of texture synthesis if needed. With the two techniques we have introduced, our algorithm can produce visually plausible stego synthetic textures even if the secret messages consisting of bit 0 or 1 have an odd occurrence of probabilities. The presented algorithm is secure and robust against an RS steganalysis attack. We believe our proposed scheme offers substantial benefits and provides an opportunity to extend steganographic applications.

VII. REFERENCES

[1] N. F. Johnson and S. Jajodia, “Exploring steganography: Seeing the unseen," Computer, vol. 31, no. 2, pp. 26-34, 1998.
[2] N. Provos and P. Honeyman, “Hide and seek: an introduction to steganography," Security Privacy, IEEE, vol. 1, no. 3, pp. 32 44, 2003.
[3] F. A. P. Petitcolas, R. J. Anderson, and M. G. Kuhn, “Information hiding survey," Proceedings of the IEEE, vol. 87, no. 7, pp. 1062-1078, 1999.
[4] Y.-M. Cheng and C.-M.Wang, “A high-capacity steganographic approach for 3D polygonal meshes," The Visual Computer, vol. 22, no. 9, pp. 845-855, 2006.
[5] S.-C. Liu and W.-H. Tsai, “Line-based cubism-like image |A new type of art image and its application to lossless data hiding," IEEE Trans. Inf. Forensics Security, vol. 7, no. 5, pp. 1448-1458, 2012.
[6] I.-C. Dragoi and D. Coltuc, “Local-prediction-based difference expansion reversible watermarking," IEEE Trans. Image Process., vol. 23, no. 4, pp. 1779-1790, 2014.
[7] J. Fridrich, M. Goljan, and R. Du, “Detecting LSB steganography in color, and gray-scale images," MultiMedia, IEEE, vol. 8, no. 4, pp. 22-28, 2001.
[8] Y. Guo, G. Zhao, Z. Zhou, and M. Pietik□ainen, “Video texture synthesis with multi-frame LBP-TOP and di_eomorphic growth model," IEEE Trans. Image Process., vol. 22, no. 10, pp. 3879-3891, 2013.
[9] L.-Y. Wei and M. Levoy, “Fast texture synthesis using tree-structured vector quantization," in Proc. of the 27th Annual Conference on Computer Graphics and Interactive Techniques, 2000, pp. 479-488.
[10] A. A. Efros and T. K. Leung, “Texture synthesis by non-parametric sampling," in Proc. of the Seventh IEEE International Conference on Computer Vision, 1999, pp. 1033-1038.
[11] C. Han, E. Risser, R. Ramamoorthi, and E. Grinspun, “Multiscale texture synthesis," ACM Trans. Graph., vol. 27, no. 3, pp. 1-8, 2008.
[12] H. Otori and S. Kuriyama, “Data embeddable texture synthesis," in Proc. of the 8th International Symposium on Smart Graphics, Kyoto, Japan, 2007, pp. 146-157.
[13] H. Otori and S. Kuriyama, “Texture synthesis for mobile data communications," IEEE Comput. Graph. Appl., vol. 29, no. 6, pp. 74-81, 2009.

[14] M. F. Cohen, J. Shade, S. Hiller, and O. Deussen, \Wang Tiles for image and texture generation," ACM Trans. Graph., vol. 22, no. 3, pp. 287-294, 2003.
[15] K. Xu, D. Cohen-Or, T. Ju, L. Liu, H. Zhang, S. Zhou, and Y. Xiong, "Feature-aligned shape texturing," ACM Trans. Graph., vol. 28, no. 5, pp. 1-7, 2009.
[16] L. Liang, C. Liu, Y.-Q. Xu, B. Guo, and H.-Y. Shum, "Real-time texture synthesis by patch-based sampling," ACM Trans. Graph., vol. 20, no. 3, pp. 127-150, 2001.
[17] A. A. Efros and W. T. Freeman, "Image quilting for texture synthesis and transfer," in Proc. of the 28th Annual Conference on Computer Graphics and Interactive Techniques, 2001, pp. 341-346.
[18] Z. Ni, Y.-Q. Shi, N. Ansari, and W. Su, "Reversible data hiding," IEEE Trans. Circuits Syst. Video Technol., vol. 16, no. 3, pp. 354-362, 2006.
[19] X. Li, B. Li, B. Yang, and T. Zeng, "General framework to histogram shifting- based reversible data hiding," IEEE Trans. Image Process., vol. 22, no. 6, pp. 2181-2191, 2013.
[20] J. L. Rodgers and W. A. Nicewander, "Thirteen ways to look at the correlation coe_cient," The American Statistician, vol. 42, no. 1, pp. 59-66, 1988.
[21] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," IEEE Trans. Image Process., vol. 13, no. 4, pp. 600-612, 2004.

AUTHORS

Tushar Patil, Pursuing B.E. in computer engineering at Sinhgad Academy Of Engineering.

Atul Sonawane, Pursuing B.E. in computer engineering at Sinhgad Academy Of Engineering.

Pankaj Sonawane, Pursuing B.E. in computer engineering at Sinhgad Academy Of Engineering.

Atul Bankar, Pursuing B.E. in computer engineering at Sinhgad Academy Of Engineering.

Guide: Mr. Muzaffar A. Shabad
Prof. SAOE, Kondhwa, Pune

