Hiding Images as a Single Image Using Arithmetic Operations

B. Rahul, S.S. Dhenakaran

***Abstract**: This paper proposes a new mechanism for hiding images by blending two or more color images together. The resultant image formed from the blending process does not show the input images. The proposed mechanism makes it difficult to identify images taken for blending process. The work is attempted to mix two or more color images together for new hiding mechanism. The proposed mechanism first considered two color images and then following the processes such as resizing images, segmentation, rotation of pixel bits to make images more complex, and then hides the input images by combines these complex images by performing arithmetic operations on pixels. Math activities performed on pixels unmistakably combine the images to hide input images. The reverse procedure obviously concentrates input images with no bit loss or quality in the images. The work can be stretched out to blend more images to keep privacy in image information, diminishing spaces for dumping images.*

Keywords: color images, image hiding, arithmetic operations, segmentation, and bit rotation

I. INTRODUCTION

In the proposed work all the blending concepts are ignored and some new meaning is given to the blending process. That is, here we use blending mechanism for hiding images. There are many images we have that are to be kept secure and confidential. So now we have to think about how these images are to be kept securely. If there is only one image can handle or protect that image easily. But if there are hundreds of thousands of images, then how these are hiding together, these thoughts are reached to my research work.We can produce a single image as output from two or more images by combining them together. The important thing is that no pixel values of any image will be lost. All the images are hidden in one image. Here doing not only the blending process, but also doing different operations on each image. Before combining each image the appearances of the each image are changed using different operations. All the operations are performing on the pixels of images.

We can change the appearance of any image by performing some mathematical operations on that pixel value. For that we have to extract the pixel values from the image and put those values into three dimensional arrays. Each three dimensional arrays contain a full image. That is, we can create an image using these pixel values in the arrays. Any operations on the image are done using the arrays. The important thing is that when converting the array into an image it is possible only if the array contain the values of uint8 type. That is, all the pixel values are ranged from the values 0 to 255.

Revised Manuscript Received on September 03, 2019

Arthur G. Ibañez, Mphil Research Scholar , Department of Computer Science, Alagappa University, Karaikudi, Tamil Nadu, India.

Denis V. Policar, Professor, is working in the Dept. of Computer Science, Alagappa University, and Tamil Nadu State, India

Any negative values or the values above the 255 cannot be converted to an image. When extracting pixel values of an image and put it into an array it must be the values in between the range 0 to 255. There is a possibility of changing the pixel values beyond the rgb range when perform some operations on the pixel values. So in these cases we must keep the values, not going beyond the range.

II. RELATED WORK

R.Johnson Suthakar & J.Monica Esther M.E. "Study of Image Fusion- Techniques,Method and Applications"[1] has descussed the survey of image fusion. The paper is sorted out as follows: Evolution of image fusion research which contains the simple image fusion and pyramid decomposition based image fusion. Simple image fusion includes pixel by pixel operations like addition, subtraction, average and division. Image Fusion Techniques which ordinarily depend on simple pixel operations on the input image values. Fusion techniques are classified into signal level, pixel/data level, feature level and decision level. The image fusion methods explained in this paper include Intensity-Hue-Saturation (IHS) Image Fusion Method, The Brovey Transform image fusion, Principal Component Analysis. Multi resolution analysis is one of the most encouraging strategies in image processing. The MRA concepts were intimated by Meyer and Mallat, which provides a natural framework for the understanding for wavelet and pyramid, transform bases. Applications of image fusion include Ocean surveillance, Air-to-air and surface to air defense, Battlefield intelligence, surveillance and target acquisition, Strategic warning and Defense, Robotics, Medical Diagnoses, Environmental monitoring.

Lin-Yu Tseng , Yung-Kuan Chan, Yu-An Ho & Yen-Ping Chu "Image Hiding with an Improved Genetic Algorithm and an Optimal Pixel Adjustment Process", [8] implant a secret image into a spread image. A secret image can be moved securely utilizing this procedure. Disarrange every pixel in the secret image and alter every one of them to frame a reasonable series of bits that could be inserted. At that point these bits are inserted into the spread image in comparing spots, and this image would turn into a stego-image that conceals secret information. This paper proposes new image disarranging procedure. It utilizes an improved genetic algorithm and an Optimal Pixel Adjustment Process, OPAP, to upgrade the nature of a Stego-picture. Exploratory outcomes demonstrate that a stego-image is vague from the spread image. The stego-image can install 4 bits for every pixel, and the mean-square mistake of a stego-image is much lower than results for previous technique

III. PROPOSED APPROACH

In proposed work following operations are performing on the pixel values in the arrays to change the appearance of the image before combining them

- Make arrays equal size
- Perform segmentation
- Interchange segments of second input image
- Rotate bits of pixel values

We can say that each array is the storage space for an image. The operations mentioned above are performed on the pixel values stored on the arrays and then convert it into an image the appearance of the images will totally changed. Now too complex images are getting. they combining to form a single blended image. Output image getting after these operations provide security and confidentiality for input images.

A. Working principle

1) Take two color images.
2) Put pixel values of images into 3 dimensional arrays.
3) Make images equal size
4) Divide images into 4 segments
5) Interchange the segments of second image
6) Rotate pixel bits of both images.
7) Then perform binary addition operation between pixel values on the segments of two input images.
8) Replace all pixel values greater than the value 255 with zero (At reverse process these zeroes are replacing with original values)
9) Convert array of pixels into rgb value.
10) Display the output Image

There are numerous techniques for consolidating images. But, these techniques are insufficient to hide images. By joining images, the data in the input images is unmistakable in the blended image. But, here we change all mixing ideas. Here we utilize a mixing component for concealing significant images. That is, the input images are covered up in the blended image. Not get any piece of information of input images from the output image. Before consolidating images treat them independently. That is, changing the appearance of each image before they join. Every one of the assignments is performed utilizing the three-dimensional arrays. That is, the pixel values of each image are put into these arrays. Then all activities are performed utilizing these arrays. We can change the appearances of the images by rolling out certain improvements in pixel values in the arrays. If the input images are alternate size then, they should make equivalent size. Later need to perform addition operation on the pixel values in the arrays to join the images, so the array of pixel values of input images should be equivalent size.

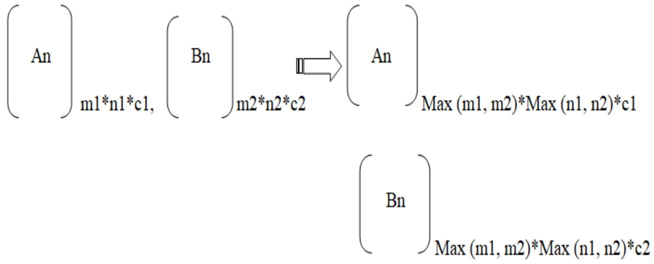

Fig 1: making of two images equal sized

'An' and 'Bn' are the two diverse size arrays to hold the pixels of input images. ‘An’ represent for first image and ‘Bn’ represent for second image. ‘An’ has the size m1*n1*c1 and ‘Bn’ has the size m2*n2*c2. The size of ‘An’ and ‘Bn’ may be equal or not. If the arrays are different size then they should make equivalent. For that take maximum of row size and maximum of column size of the two arrays. The channel size consistently 3 for all color images. So no compelling reason to take the most extreme channel size and can utilize a similar channel size. Utilizing these dimensions change the arrays by including zeros toward the end of the rows and columns of the two arrays. By utilizing these equivalent size arrays we can make equivalent size images.Then, divide these equal-sized images into four segments. A11, A12, A21 and A22 are the segments of first image. B11, B12, B21 and B22 are the segments of the second image. The important thing is that the size of (A11 and B22), (A12 and B21), (A21 and B12) and (A22 and B11) should be equal.

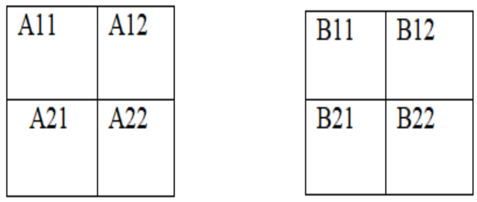

Fig 2: Segments of first image and second image

Then interchange the segments of the second image as follows:

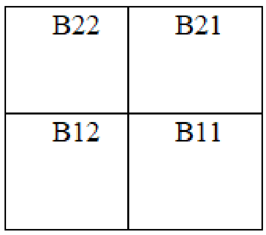

Fig3: Segments of second image after interchange

Then rotate the bits of pixel values on each segment of two images four times to the left. After bit rotation, the appearance of each image is changing.

Now too complex images are getting. Then combine these images by performing the addition operation between the array of pixel values of each segment of the first image and array of pixel values of corresponding segments placed in the second image

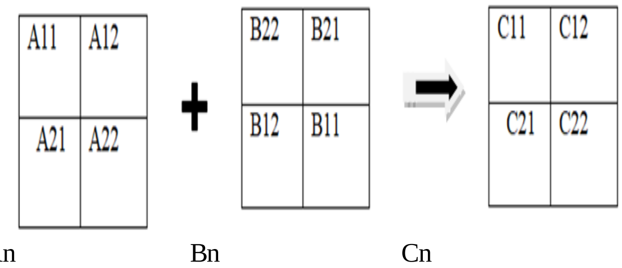

Fig 4: combining of two images

Addition of pixel values of segments in 'An' and 'Bn' as follows:

- convert pixel values type in each segment of the An and Bn to the type signed 16-bit integer
  An=int16 (An);
  Bn=int16 (Bn);
- perform addition operation between corresponding segments in An and Bn
  Cn=plus (An, Bn)
- after the addition operation, the segments in 'Cn' may contain the values greater than the values 255. The pixel values must be the range 0 to 255. So replace all the values which are beyond the limit with zeroes. Then convert the pixel values type into unsigned 8-bit integer. Then display the blended image using these pixel values. This is the final output. The input images are hiding in the new image.

For the reverse process, must perform the following actions after Cn=plus (An, Bn) operation. Replace all the values less than or equal to 255 from the Cn with zeroes. Then pixel values in the 'Bn' are subtracting from these pixel values. Then replace all negative values with zeroes. Keep this pixel values. At reverse process, these pixel values should be adding with pixel values in the 'Bn'. Then replace all values which are less than or equal to 255 with zeroes. Then add these pixel values with pixels of blended image. Now getting actual pixel values that are formed during the operation Cn=plus (An, Bn). Now we can extract 'An' by subtracting Cn and Bn (Bn must keep as a key). Then Pixel bits of both 'An' and 'Bn' are rotated four positions right. Interchange the segments of 'Bn'. Then remove all the zeroes that are added at the end of the arrays for making equal size arrays during the blending process. Now getting the original 'An' and 'Bn'. From this can extract the actual input images. From the above processes, we can clearly see that there is no pixels lose during the blending process and reverse processes.

IV. RESULTS

The outcomes showed beneath are uses of the proposed methodology to mix images. The first two models have considered two-color images and by mixing they have created one resultant image. The third model has considered three color images as input and created one output image. Obviously all results are extraordinary and not demonstrating element of input image like the current mixing mechanisms. Any number of images can be mixed to yield one output in this manner lessening the capacity of essential and private images in a single corner

Fig 5: hiding two similar images as a single image.

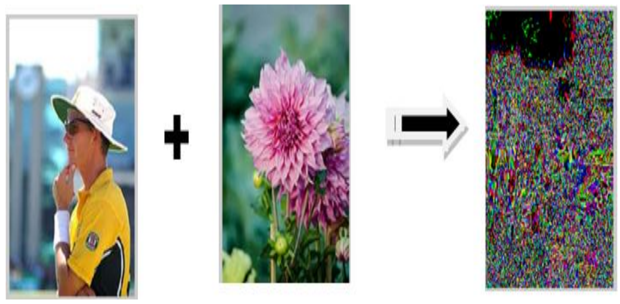

Fig 6: hiding two dissimilar images as a single image.

Fig 7: hiding three dissimilar images as a single image

V. CONCLUSION

In this paper, a new technique is proposed for hiding color images. The mixing procedure conceals the significance of input images. The work can be connected to keep confidentiality on image information and to safeguard image during transmission too. The strategy is tried for taking three color images however can be reached out to a huge number of color images for mixing. Hence the principle subject of concealing images is effectively actualized.

ACKNOWLEDGEMENT

The work is supported by RUSA 2.0, Alagappa University for submitting thesis and getting a degree in this University, India.

REFERENCES

1. R.Johnson Suthakar et al, International Journal of Computer Science and Mobile Computing, Vol.3 Issue.11, November- 2014, pg. 469-476
2. Shutao Li, Xudong Kang, Leyuan Fang, Jianwen Hu, Haitao Yin Pixel-level image fusion: A survey of the state

of the art http://dx.doi.org/10.1016/j.inffus.2016.05.004 1566-2535/© 2016 Elsevier

3. https://homepages.inf.ed.ac.uk/rbf/HIPR2/copyrght.htm
4. Ross T. Whitaker, Member, IEEE "A Level-Set Approach to Image Blending",ieee transactions on image processing, vol. 9, no. 11, november 2000
5. https://pdfs.semanticscholar.org/a8b8/8d7271df1c52e028c932cf8f7459 92ab36c0.pdf
6. International Journal of Computer Applications (0975 –8887) Volume 36–No.5, December 201119
7. A. M. Nickfarjam , H. Ebrahimpour-komleh & A. Pourshabanan Najafabadi "Image hiding using neighborhood similarity", 6th Conference on Information and Knowledge Technology (IKT 2014), May 28-30, 2014, Shahrood University of Technology, Tehran, Iran
8. Lin-Yu Tseng ; Yung-Kuan Chan ; Yu-An Ho ; Yen-Ping Chu "Image Hiding with an Improved Genetic Algorithm and an Optimal Pixel Adjustment Process", 2008 Eighth International Conference on Intelligent Systems Design and Applications, Publisher: IEEE, doi: 10.1109/ISDA.2008.235
9. International Journal of Pattern Recognition and Artificial Intelligence Vol. 16, No. 04, pp. 399-416 (2002)

AUTHORS PROFILE

B.Rahul,Mphil Research Scholar
Department of Computer Science
Alagappa University,
Karaikudi, Tamil Nadu
Rahul.rahul.b9@gmail.com
Mob: 7907745191

Dr.S.S.Dhenakaran, Professor, is working in the Dept. of Computer Science, Alagappa University, and Tamil Nadu State, India for the past three decades. He has completed M.Sc(Mathematics), PGDOR, MCA,M.Phil, Ph.D degrees. He has experience in teaching and research in the discipline of Computer Science. His area of interest is information security, design of algorithms and data mining techniques. He has published more than 130 papers in international Journals and Conferences.