IMAGE RECONSTRUCTION AND TEXT EMBEDDING USING GRAPH CUT

[1]*Mahwish Bano , [2]Tariq Shah, [3]Romana Talat, and [4]T. M. Shah
[1]*Department of Mathematics, Quaid-e-Azam University, Islamabad, Pakistan
*(corresponding author)Tel: 92519263351, maham_dia@yahoo.com
[2]Department of Mathematics,Quaid-e-Azam University , Islamabad, Pakistan,
stariqshah@gmail.com
[3]Department of Computer Science, Bahria University,Islamabad, Pakistan
romanatalat@yahoo.com
[4]Department of Mathematics,, Air University ,Islamabad, Pakistan,
dr.tasneem@mail.au.edu.pk

ABSTRACT— *In the series of our previous work, the main focused was how to hide important messages/medical images into not objectionable images/pictures. Earlier works were based on genetic algorithms, Piece wise linear chaotic map and was found reasonable signal to noise ratio for the above algorithms.*

In continuation of similar work, we now introduced a new algorithm based on texture synthesis. The approach is known as graph cut for patching regions from a sample image/video transformed and copied to the output and then stitched together along optimal seams to generate a new output. The size of the patch is not chosen before, but instead a graph cut technique is used to determine the optimal patch region for any given offset between the input and the output texture. The graph cut technique for seam optimization is applicable in two and three dimensions. This graph cut technique of patching into original picture is very much applicable in image processing. We present a new idea to embed text into patching seam and placed in the original image. Although this technique is expensive in terms of image area usage but it is a new idea for text embedding and has to be optimized in the sense of original picture usages. We show results for regular, random and the natural images. We also demonstrate how this method can be used interactively.

Keywords- Text embeddind, image based randering, image processing, graph cut technique, patching, morphing**.**

1. Introduction

Sample based image texture synthesis method is needed to generate large realistic texture for rendering of complex graphic scenes. The concept of texture is defined as an infinite pattern that can be modeled by stationary stochastic processes. In this paper, we presented new algorithm which generate an infinite pattern from a small amount of trainee data using a small example patch embedded with text of the texture. We generate a large pattern with embedded text stochastically. This algorithm first searches for an appropriate location for the placement of the patch. It is then uses a graph cut technique to find the optimal region of patch embedded with text to transfer to the output. These approaches are not limited to spatial (image) texture, and include spatio-temporal (video) texture.

A generated texture should be perceptually similar to the example texture. This concept of perceptual similarity has been formulized as a Markov Random Field (MRF). The output texture is represented as grids of nodes depended on the similarity of their neighboring pixels in the input textures. This input texture is used for text embedding. The goal of texture synthesis restated as the solution for the nodes of the network. This formulation is known as machine learning. The primary contribution of this paper is an algorithm for texture synthesis with text embedded into it which after finding a good patch offset, compute the best patch seam. This algorithm works by reformulating as a minimum cost graph cut problem. Finally we have extended our previous text embedded into encrypted images based on graph cut algorithm. This graph cut technique is fairly used in image processing as a whole and synthesis specifically; First time we have introduced our text embedded into patterns which after synthesized would be placed in an output image. Further this technique can also be used in video synthesis technique.

2. Proposed Algorithm

- Take a target matrix in terms of adjacency matrix of size 256 x 256. This is called target matrix to be restructured
- Identify a cluster of pixels which is needed to be re-patched to get the target matrix repair.
- Consider another objective matrix of the same size from where a patch is extracted using graph cut technique. This patch of objective matrix is being used to embed text in it.
- The embedded text patch is placed in the target matrix to recover the original picture. This original picture is now embedded with text.
- To recover the text the patched portions remove from the original picture and text to be recovered.
- Embedding of text into patch can be done with common text embedding algorithm randomly.
- The algorithm is implemented into a sample picture (figures 8-11), where figure (8) is a target picture to be synthesized, figure (9) is the mask of input image. The text is to be embedded in the mask. Figure (10) is obtained after placing the mask into target image through graph cut technique and figure (11) is spread after the use of graph cut technique.

2.1 Details of Algorithm

The jest of the algorithm is the composition of the matching scene when place back to the target image. It has to recover the target image completely without superimposing pixels. The process of image composition is described in the following section.

2.1.1 Image composition Using Graph Cut

Once the best patch was found, we place composite matching scene into incomplete image. The input mask covers the area of incomplete image that the user specified but edges of mask might not be best for stitching two images when gradient of both images are very different to each other. This can be reduced by refining the input mask, we allow mask to leave from its original path and find its best place so that subsequent blending looks more persuasive. Instead of copying and pasting matched patch into hole, stitch it together with original image so that seem between matching patch and original image is less noticeable. This can be done by finding maximum flow/minimum cut via graph cut, it has played very important role in solving certain problem in vision. Growing number of publications in vision use graph-base energy minimization techniques for application like image segmentation, image restoration stereo, object recognition, Texture synthesis, shape reconstruction and others.

Graph we are building consider one node for each pixel that will be in final image. Each pixel is connected to its four neighbors. For solving graph cut problem, we need some quality measure for pixel from the original image and matched image. We assigned weights to all edges of pixel that is being considered the simplest measure is color difference between pair of pixels where s and t be two adjacent pixels in the overlap region. These weights actually decide where the best cut will be. In order to create flow inside graph we need two terminal source and sink as shown figure 1, all pixels under mask are connected to sink while all pixel at the maximum border of local context are connected to source terminal so flow must go through the border of input mask. For calculating maximum flow/ minimum cut, we have used library implementing Max-Flow Algorithm [11].

2.1.2 Background on Graph

A graph G= {V, E} consist of set of nodes V and set of edges E. Each node represent to a single pixel. In order to solve problems in vision using graph cut, we need two additional nodes source and sink are called terminals as shown figure 1. In the context of vision, terminals correspond to set of labels that can be assigned to each pixel that is being considered. George et al [13] were first to proposed the max-flow/min-cut algorithms from the combinatorial optimization can be used to minimize the energy function. The energy function proposed by George et al and other graph-based methods can be represented as equation 1.

$$\boldsymbol{E(L) = \sum_{p\in P} D_{p(L_p)} + \sum_{p,q\in N} V_{p,q}\,(L_p, L_q)} \quad \textbf{(1}$$

Where $L = \{L_p \mid p \in P\}$ is a labeling of image P. $D_p(.)$ is data penalty function that indicates label preference for pixel based on intensities. $V_{p,q}$ is an interaction potential that instigate spatial coherence by penalizing discontinuities between pixels. Normally there are two types of edges N-links and T-links. N-links connects pairs of pixel. Cost of n-links refers to a penalty for discontinuity between pixels and this cost can be derived from the $V_{p,q}$ term from equation 1. T-links basically are used to connect pixel with terminal nodes called source and sink, cost of t-links refers to a penalty for assigning a specific label to the pixel. In the next section we will describe Min-cut/Max-flow problem briefly

.

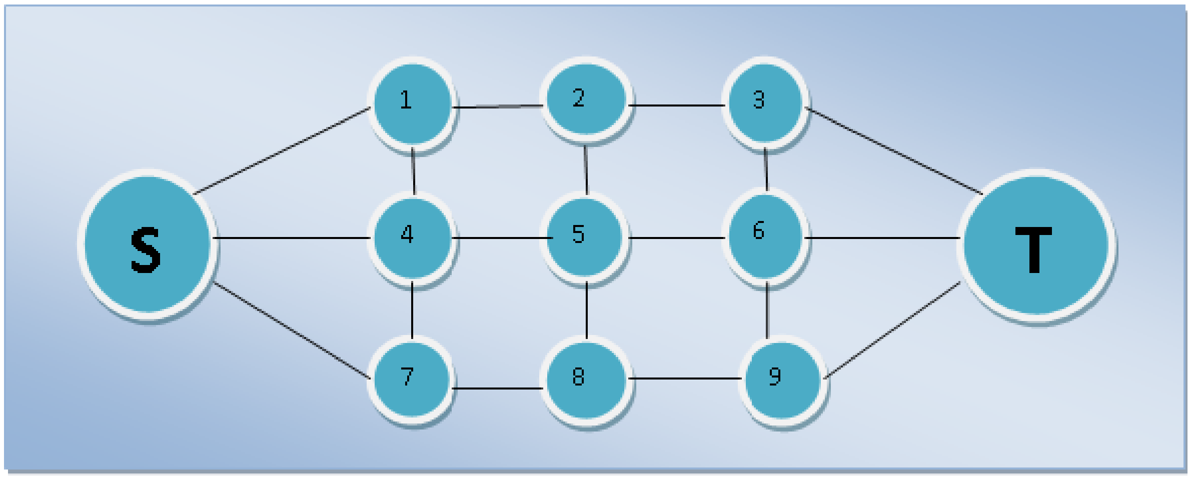

Figure 1: Example of graph with Source and Sink

2.1.3 Minimum-cut/Maximum- Flow Problem

An s/t cut CT with two terminals is partition of graph into subset S1 and S2 in such a way s belong to S1 and t belongs to S2. Figure 2 shows example of simple graph cut which show all nodes say Node1, Node2, Node4 and Node7 should belongs to subset S1 while Node3, Node5, Node6, Node8 and Node9 should belongs to subset S2. The cost of cut CT can be defined as “Sum of cost of all boundary edges say $\{p, q\}$ where $p \in S1$ and $q \in S2$. The basic purpose of minimum cut is to find a cut which has minimum cut among the all cuts. This problem can be solved by finding maximum flow from source S to sink T.

In other words we say that, maximum flow is like “maximum amount of water” that can be pass from source to sink by using edges refers to as “Pipes”. This can be done using most commonly used algorithm Ford and Fulkerson[12] states that maximum flow from source to sink actually saturates the edges that divide the graph into two subset S1 and S2 corresponding to minimum-cut. Hence maximum-Flow and minimum-Cut are equivalent. In simple words cut CT divides the graph into disjoint subset S1 and S2 and each subset containing only one terminal. Therefore, any cut corresponds to assigning a label to a pixel (node)

.

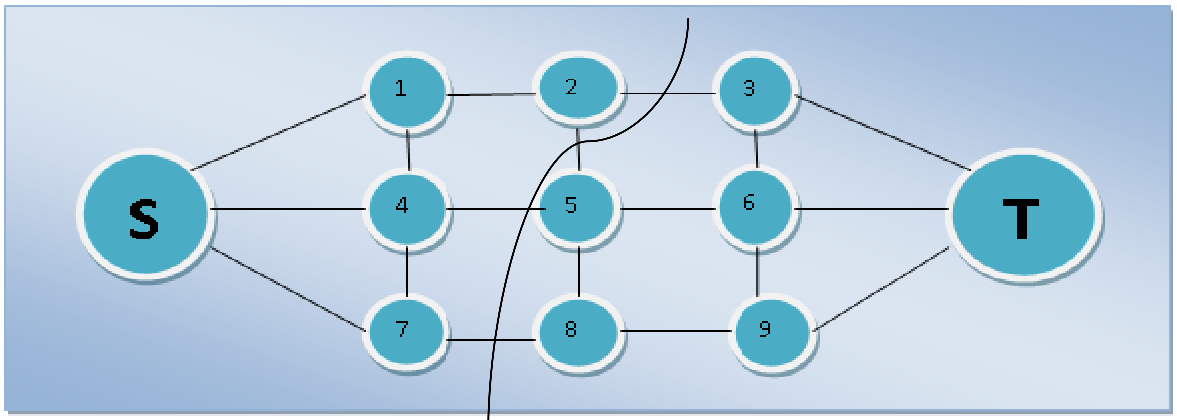

Figure 2: Example of cut on a Graph

2.2 Algorithm Overview

Figure 3 shows basic terminology, there are two non-overlapping search trees S1 and S2 connected with parent/root at source S and sink T respectively. In tree S1 and S2 all edges from its parent to children are non-saturated. The nodes that are not connected to any terminal node are called free nodes.

$S1 \subset V$, $S \subset S1$, $S2 \subset V$,$T \subset S2$, $S1 \cap S2 = \phi$

The nodes in tree S1 and S2 can b passive and active labeled as A and respectively, as shown in figure 3.The basic idea is active nodes allow to grow tree by getting new children from the free nodes. Active nodes are those which are on the border while passive nodes are internal nodes that cannot grow the tree because these nodes are completing blocked by other nodes. This algorithm is augmented –based, augmented path is found when an active node in one of the tree S1 or S2 finds a node that belongs to other tree. There are three stages that this algorithm iteratively repeats.

I. **Growth Stage:** In this stage search tree S1 and S2 grow by finding new children from set of free nodes until one node detects a node that belongs to other search tree and giving an S-T path. At growth stage search trees S1 and S2 elaborated. The active nodes search for non-saturated adjacent nodes and get new children from set of free nodes as represented by black nodes in figure 5.These new children become active node of respective tree, all neighbor of given active node are explored then active nodes become passive node. Growth stage terminated when active node finds a node that belongs to other tree and we get a path from S to T.

II. **Augmentation Stage:** In this stage, a found path during growth stage is augmented that actually break search tree into forest. Augmentation stage augments the path by pushing maximum flow so that some edges in the path become saturated. Some of the nodes in this stage become "Stray" as edges linking Stray nodes to their parents becomes saturated (having maximum flow).

III. **Adoption Stage:** In this stage tree S1 and S2 are restored. The main purpose of this stage is to restore the single-tree structure of tree S1 and S2 with roots S and T. Adoption stage find parent of "Stray" that should be from same tree S1 or S2 and connected with non-saturated edge, if adoption stage fails to find valid parent of "Stray" node then this node is removed and becomes free node This stage terminates when there is no "Stray" node and tree S1 and S2 are restored

.

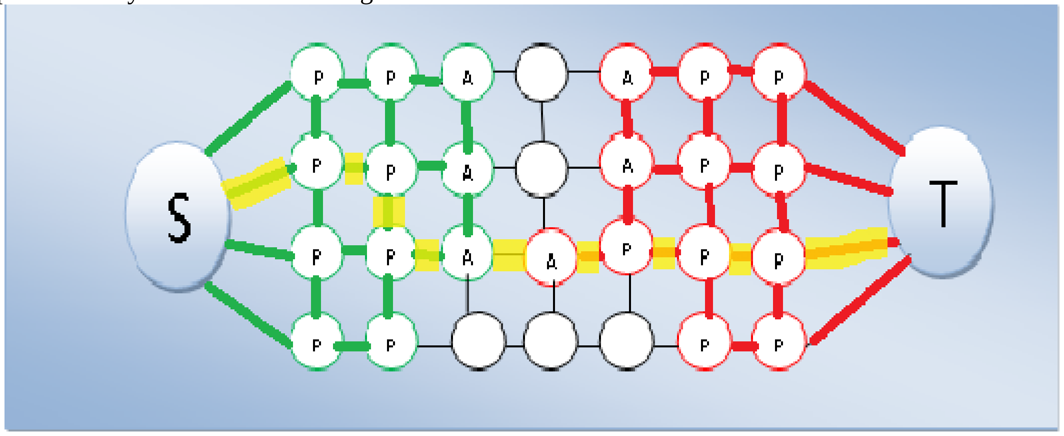

Figure 3: Example of search Tree S1 (Green nodes) and S2 (red nodes) after growth Stage when path is found from S to T .Free nodes are represented by black nodes. Active and passive nodes are labeled by A and P respectively.

```
Initialize:   S1= {S}   S2= {T}
A={S, T}   Stray=ϕ
While True
Grow S1 or S2 to find Augmenting
 Path "P" from S to T
If P =ϕ terminates
Augment on PATH
Restore Stray
End while
```

Figure 4: Structure of Algorithm

```
Find the bottleneck capacity C on path P
Update the residual graph $G_f$ by pushing
flow f through path P
For each edge $(q, r)$ in P that becomes
saturated
    If $flag(q) = flag(r) = S1$ then set
            $Parent(r) := \emptyset$   and
            $Stray := Stray \cup \{r\}$
  If  $flag(q) = flag(r) = S2$ then set
            $Parent(q) := \emptyset$  and
            $Stray := Stray \cup \{q\}$
```

Figure 5: Growth Stage

Figure 6: Adoption Stage

2.3 *Detail Implementation of Algorithm*

For augmenting path algorithm, Flow f and residual graph G_f (Residual graph have same topology as original graph, it only reflects residual capacity of edge given the amount of flow in that edge) should maintain. Let we have a graph $G = \{V, E\}$ .The structure of algorithm as shown figure 4. The detail of algorithm's stages Growth, Augmentation and Restore is described below.

It is very helpful to store content of search tree S1 and S2 using a flag that actually describe the relationship of each node q. so that

$$flag(q) = \begin{cases} S1 \text{ if } q \in S1 \\ S2 \text{ if } q \in S2 \\ \emptyset \text{ if } q \in FreeNode \end{cases}$$

If node q belongs to any tree then its content would be stored as $Parent(q)$. It is very important to mention it roots of search trees say source and sink, Stray and all free nodes don't have any parent. All augmenting path algorithm must maintain residual graph so $Res\_Cap(q \rightarrow r)$ shows the residual capacity of edge (q, r) if $flag(q) = S1$ or edge (r, q) if $flag(q) = S2$. The point that is very important all these edges should be non-saturated for node q to be a valid parent for its child r.

2.3.1 Growth Stage Implementation

At this stage Active node get new children from free nodes.

2.3.2 Augmentation Stage Implementation

During growth stage a path $S \rightarrow T$ has been found. The augmentation stage takes this path as an input and augments that path by pushing maximum flow from S to T so that some edges in path may become saturated. Initially, "Stray" is empty but at the end of this stage there may be some "stray "nodes because at least one edge become saturated in the given path.

2.3.3 Adoption Stage Implementation

In this stage all stray nodes are processed .Each node q being processed find a new parent within same tree. If node q finds valid parent that node will remain in the same tree but with different parent and if it does not find a valid parent then this node will remove from Stray and becomes a free node. All its children nodes becomes Stray.

The operation "Process q" contains following steps. First, find new valid parent of process q from all its neighbors. A parent r is valid parent if

$flag(q) = flag(r), Res\_Cap(r \rightarrow q) > 0$ and origin of r should be source or sink because during adoption stage some node may come from Stray in the search tree S1 and S2. If node q catches the legal parent then we assign Parent(q) = r and status of node q remains the same but if it does not find valid parent then node q is treated as free node and some more operations are needed to perform

a. Scan all neighbor of node q such that flag(r)=flag(q)
   i. If Res Cap>0 add r to the active set
   ii. If Parent(r) = q add r to the set of Stray and set Parent(r):=0
b. Flag(q) :=null, A := A – {q}

3 Experimental Result:

Figure 7: Augmentation Stage

After the experiment, outcomes have shown the strength of this algorithm as compare to the others [3]. We embedded and hide the text in actual or targeted image and got steganographed image. The peak signal to noise ratio (PSNR) of steganographed image is analyzed, the PSNR increased in this algorithm and visually, one cannot differentiate between original image or wrap image and steganographed image. Having implemented the Graph cut technique on known test plain images, we obtained SNR value listed below for testing algorithm.

Table 1. The SNR value of different pictures

Plain Image	256 x 256
Cameraman	0.4361
Rice	0.3621
Lena	0.4798
Tree	0.3123
APC	0.4291
Test Park	0.3982

Elain	0.4853
Truck	0.4663
Tiffany	0.4781
Pencils	0.2216
Couple	0.3672
Aerial	0.3144
Chemical Plant	0.4462
Moon Surface	0.3698

Table 2. Correlation of two adjacent pixels

Correlation	Horizontal	Vertical	Diagonal
Pepper	0.942848	0.945174	0.897210
Encrypted	0.000182	0.000357	0.004215
Cameraman	0.933475	0.959223	0.908663
Encrypted	0.000090	0.007362	0.003039
Lena	0.904267	0.906432	0.875651
Encrypted	0.000167	0.000342	0.004875
Pencils	0.863571	0.866551	0.824165
Encrypted	0.000159	0.000332	0.003452

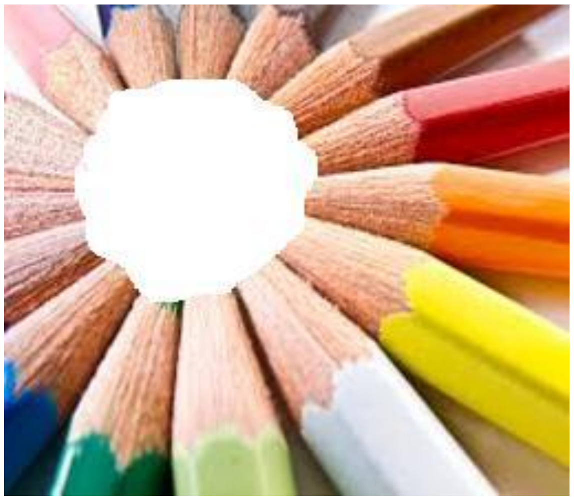

Figure 8:Input Image

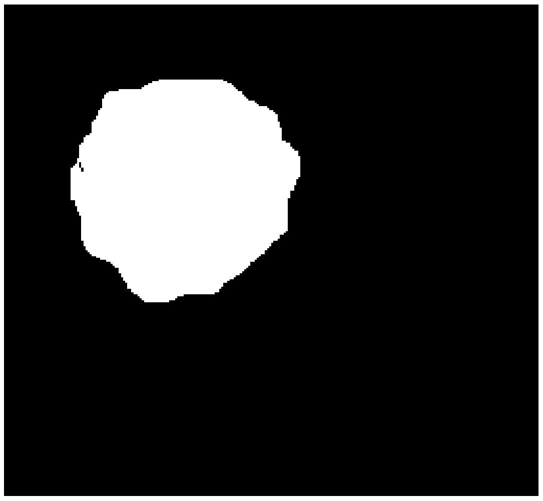

Figure 9:Mask of input Image

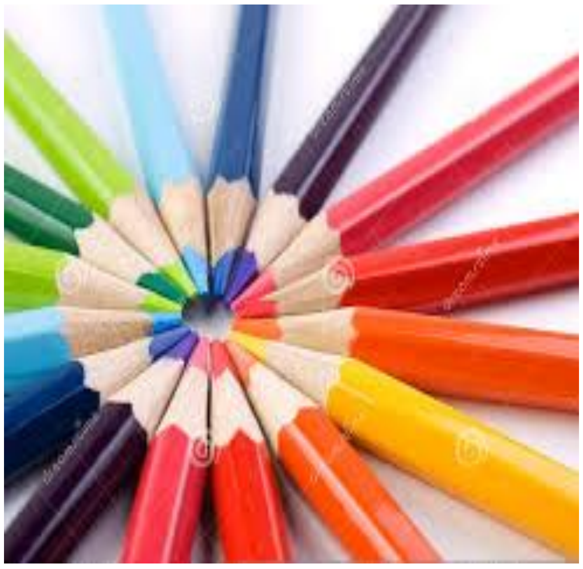

Figure 10:Matching Image

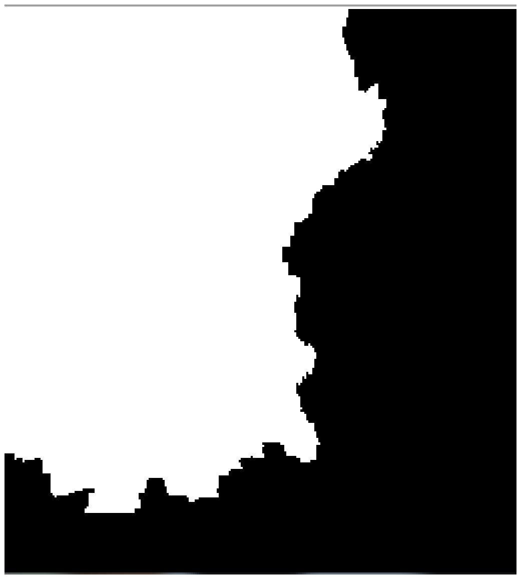

Figure 11: Mask After Graph cut

4 Conclusion

This paper is two-fold: one it has been described the details of graph cut technique to recover the original image with composing matches images and second the use of patch for the purpose of text hiding has been proposed. An example of using such algorithm has been demonstrated and analyses have also been carried out for a variety of known images and SNR were obtained.

Hiding text in such a manner is first time presented for a high quality security with double achievements. Further experiments would be done for saving space and cost effectiveness. Due to random security approach, it has ability to resist any kind of attacks. The scheme can be used on network communications.

REFERENCES

[1] Bano, M.,Shah, T. Shah, T. M. ``Genetic algorithm on Piecewise Linear Chaotic Map Based Image Encryption” approved for publication in Journal of Machine Learning and Cybernatics, 2015.

[2] Bano, M., Shah, T., Shah, T. M. `` Text Embedded into Encrypted Image Based on Genetic Algorithm on Piecewise Linear Chaotic Map”, submitted in Indian Journal of Science and Technology, 2015.

[3] H. Liu and X. Wang, ”Color image encryption based on one-time keys and robust chaotic maps”, Computers and Mathematics with Applications, vol. 59, no. 10, pp. 3320-3327, 2010.

[4] X. Tong and M. Cui, ”Image encryption with compound chaotic sequence cipher shifting dynamically”, Image and Vision Computing, vol. 26, no.6, pp. 843-850, 2008

[5] F. Sun, S. Liu, Z. Li, and Z. Lu, ” A novel image encryption scheme based on spatial chaos map”, Chaos, Solitons and Fractals, vol. 38, no. 3, pp 631-640, 2008.

[6] Z. Liu, Q. Guo, L. Xu, M. A. Ahmad, and S. Liu, ”Double image encryption by using iterative random binary encoding in gyratordomains”, Optics Express, vol. 18, no. 11, pp. 12033-12043,2010.

[7] G. Zhang and Q. Liu, ”A novel image encryption method based on total shuffling scheme”, Optics Communications, vol. 284, no. 12, pp. 2775-2780, 2011.

[8] C. E. Shannon, ’Communication theory of secrecy systems”, Bell Sytem Technical Journal, vol. 28, no. 4, pp. 656-715, 1949.

[9] H. Liu and X. Wang, ”Color image encryption using spatial bit-level per-mutation and high- dimension chaotic system”, Optics Communications, vol. 284, no. 16-17, pp. 3895-3903, 2011.

[10] X. Wang and C. Jin, ”Image encryption using game of life permutation and PWLCM chaotic system”, Optics Communications, vol. 285, no. 4, pp. 412-417, 2012.

[11] . Boykov, Yuri, and Vladimir Kolmogorov. "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision." *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 26.9 (2004): 1124-1137.

[12]. Ford, L. R., and Delbert Ray Fulkerson. *Flows in networks*. Vol. 1962. Princeton University Press: Princeton, 1962

[13]. Greig, D. M., B. T. Porteous, and Allan H. Seheult. "Exact maximum a posteriori estimation for binary images." *Journal of the Royal Statistical Society. Series B (Methodological)* (1989): 271-279.