IJREAT International Journal of Research in Engineering & Advanced Technology, Volume 2, Issue 2, Apr-May, 2014
ISSN: 2320 – 8791 (Impact Factor: 1.479)
www.ijreat.org

Optimization in Lossless Data Hiding with the Advent of Line based Cubism like Image

Janani Gopalakrishnan[1], Kumari Gunja[2], Ms. M.Jeevitha[3]

[1, 2]Information Technology, Anand Institute of Higher Technology (Affiliated to Anna University), Chennai, Tamil Nadu, India

[3]Assistant Professor, Information Technology, Anand Institute of Higher Technology (Affiliated to Anna University), Chennai, Tamil Nadu, India

Abstract

New types of computer art, called line-based Cubism-like image, and a technique to create it automatically from a source image have been proposed. The method finds line segments in the source image by the Canny edge detection technique and the Hough transform, combines the nearby line segments, extends the remaining lines to the image boundaries, and re-color the created regions by their average colors, to create an abstract type of the original source image as the desired art image. By different threshold value number of art images are generated from a given cover image. By using outlier technique and automatic shape selection an optimum art image for hiding data among the generated art images is determined. An image similar to the selected art image is created for distinguishing the regions. Then, by utilizing the characteristics of the Cubism-like image creation process, a data hiding technique has been proposed. Data hiding with the minimal distortion is carried out skillfully by shifting the pixel's colors for the minimum amount of ±1 while keeping the average colors of the regions unchanged. The data embedding process is proved to be lossless and enhances the camouflage effect by theorems so that the cover image can be recovered perfectly after the embedded message data are extracted. Security enhancement measures are also adopted to prevent hackers from extracting embedded data correctly.

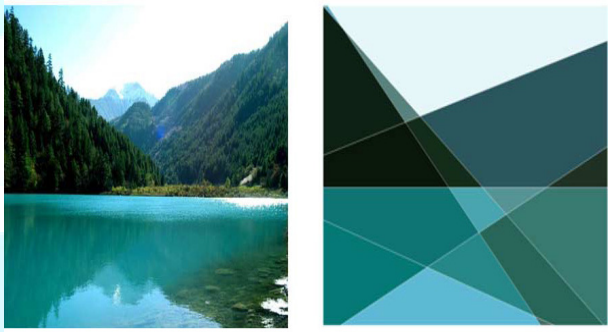
Keywords—*Computer art image, cover image, line-based Cubism-like image, reversible data hiding, stego-image, camouflage effect.*

I. INTRODUCTION

In recent years, the topic of automatic art image creation via the use of computers arouses interests of many people, and many methods have been proposed. One among them is line based cubism like image creation. Paintings of the Cubism style are dominated by lines and regions, like those shown in Fig. 1, to show abstractly a characteristic of the Cubism art—multiple-viewpoint. In this study, we try to imitate such line-type Cubism paintings to create automatically an abstract type of art image, called *line-based Cubism-like image*, from a given source image. Line segments in the source image are detected by appropriate image processing techniques, and prominent ones are keptafter removing noise line segments. Regions formed by the prominent lines then are created and the pixels in each region are recolored identically by the average color of the region. Two art images so created in this study are shown in Fig.2.

(a) (b) (c)

Fig. 1.Some computer-generated art images. (a) Image created by Hausner, (b) Image created by Haeberli. (c) Image created by Song.



(a) (b)

Fig 2. Example of line-based Cubism-like images created automatically in this study. (a) Source image. (b) Created art image from the source image.

It is desired to hide message data into the generated art image for various applications. It is also hoped that the characteristic of the art image creation process can be utilized effectively to carry out the data embedding work. The way of combining art image creation and data hiding is called *aesthetic data hiding*, a new idea of information hiding. Attracted by the art exhibited by the image, people hopefully will pay no attention to the hidden data in the art image and via this camouflage effect; the embedded data can be kept securely or transmitted covertly. Two criteria for designing data hiding techniques are *imperceptibility* of

distortion in the stego-image due to data embeddingand *recoverability* of the original cover image content fromthe stego-image. To achieve imperceptibility, a weakness of the human visual system in differentiating small color or grayscaledifferences is often utilized, e.g., by the least significant bit (LSB) modification scheme proposed by Chan and Cheng or by the contrast-keeping data embedding scheme proposed by Wu and Tsai .In this study a scheme of using the minimum color shifting of ±*1* for data embedding is proposed.

More specifically, in this study we hide message data in the automatically-generated Cubism-like image during the imagecreation process by shifting the colors of the pixels in the imageregions slightly for the *minimum* amounts of while keeping the average colors of the regions unchanged. In this way, the original art style of the image with uniform regions may be kept. The color differences in the resulting image are difficult to be found by a hacker because the human visual system is weakin discriminating small color changes. Also, by constraining the numbers of the embedded binary values of 0's and 1's to keep the average region colors unchanged, the data embedding process can be reversed so that lossless recovery of the cover image from the stego-image can be achieved, as proved by theorems in this study. In this way, the original art style in the cover image can be resumed. The security issue is also considered, and four enhancement measures are proposed.

II. PROPOSED LINE BASED CUBISM LIKE IMAGE CREATION PROCESS

A. Idea of Line-Based Cubism-Like Image Creation

Cubism artists transform a natural scene into geometric forms in paintings by breaking up, analyzing, and reassembling objects in the scene from multiple viewpoints. In addition, with the scene objects rearranged to intersect at random angles, each Cubism painting seems to be composed of intersecting lines and fragmented regions in an abstract style. The idea of the proposed art image creation technique is inspired by these concepts of the Cubism art.

Specifically, there are two major stages in the proposed line based Cubism-like image generation process—*prominent lineextraction* and *region recoloring*. In the first stage, at first we extract line segments from a given source image by edge detection and the Hough transform. Then, we conduct short line segment filtering and nearby line merging. In the second stage, at first we create regions in the image by extending the line segments to the image boundary to partition the image space. Then, we recolor the regions by the average region colors and whiten the boundaries of the regions.

B. *Algorithm for Line-Based Cubism-Like Image Creation*

The details of the above process are described as an algorithm in the following.

Algorithm 1: *line-based cubism-like image creation*

Input: a source image S, and two thresholds—the minimum line segment length L_{min} , and the minimum line distance D_{min}.
Output: a line-based Cubism-like image S_C.
Steps:
Stage 1—Prominent line extraction.

Step 1. (*Edge detection*) Apply Canny edge detection toimage S, resulting in a new image S'of edge points.

Step 2. (*Line segment detection*) Applying the Hough transformto S' to find a list of line segments $L_1,L_2,\ldots,L_m$ sorted according to their lengths, yielding a second newimage S" of the line type.

Step 3. (*Prominent line extraction*) Find prominent lines in S"by the following steps.

3.1 Select those line segments in S" with lengths larger than threshold L_{min} and discard the others, resulting in ashorter list of line segments $L'_1,L'_2\ldots L'_n$.

3.2 For all i=0 through n and all j=0 through n with $i \neq j$ and both L'_I and L'_j not deleted yet, compare L'_i and L'_j and if the distance between L'_i and L'_j is smaller than threshold D_{min}, then delete the shorter one of L'_i and L'_j.

After getting the art images based on different threshold values, an optimum art image in which the data embedding process can be carried out is determined. For selecting an optimum image out of created art images outlier technique is used. This scans the image horizontally and vertically to determine the number of lines detected and checks for the quadratics shapes formed by the line segments detected. This quadratic shapes gives information about the regions formed. Depending upon how many regions are formed and the embedding capacities of each region of the image, an image is selected which posses large amount of data embedding capacity .

Before carrying out data embedding process we generate a plain image which is similar to the selected art image. It basically contains only the prominent line segments of the selected image. To distinguish the regions effectively we recolor the regions with different colors. This helps in determining the pixels position in each region which corresponds to the original pixel position of the selected art image. The RGB value of each pixel of selected image is calculated with pixel position determined which helps in calculating average color of each region and performing average region recoloring.

Stage 2—Region recoloring.
Step 4. (*Line extension*) Extend each remaining line segment in S" to the image boundaries of S" .
Step 5. (*Region partitioning*) Partition S" into regions R_1, R_2…R_k by the extended lines.
Step 6. (*Region recoloring*) Recolor each region R_i in S" by the following steps with i=1, 2,…, k.

6.1 Compute the area A_i (in unit of pixel) of R_i and the average color (C_{ir}, C_ig, C_ib) of all the pixels in R_i .

6.2 Recolor each pixel in by (C_{ir}, C_ig, C_ib) .

Step 7. (*Line recoloring*) Recolor all region boundaries in S"by the white color.

Step 8. Take the final S" as the desired line-based Cubism-like image.

Two threshold values, namely, the minimum line segment length L_{min} and the minimum line distance D_{min}, are used in the above algorithm. They affect the flavor of the generated Cubism-like image according to our observation of the experimental results. After considering the mutual influence between the image size and the line segment length, we take one tenth of the image width as the initial values of L_{min} and D_{min} for use in Algorithm 1, and repeated executions of Algorithm1 by varying the values of L_{min} and D_{min} in our experiment.

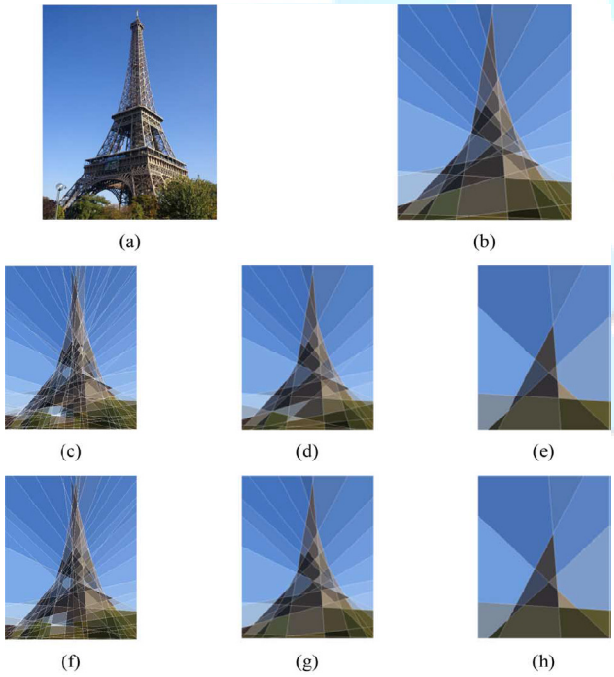


Fig.3. shows the experimental result with an image of Eiffel tower as input (a) source image with size 768 x 1024. (b)Initial D_{min}=102and Initial L_{min}=102.
(c) (D_{min},L_{min})= (51,51). (d) (D_{min},L_{min})=(51,102).
(e) (D_{min},L_{min})= (51,204). (f) (D_{min},L_{min})= (102,51).
(g)(D_{min},L_{min})= (102,102). (h) (D_{min},L_{min})=(102,204).
A seemingly better choice of the 7 images - (g).

Some results are shown in Fig. 3 from which we can see that a smaller initial value of L_{min} will cause extraction of more lines, which increases the complexity of the created image and gives an impression closer to the original image content. Contrarily, fewer lines will result from the use of a larger value of L_{min} , making the resulting image simpler and more abstract. The effect of varying the value of D_{min} is similar.

According to the above discussions, we see that different selections of the two threshold values L_{min} and D_{min} will result in totally different visual effects in the created art images. However, it is difficult to decide which result is better than the others because the decision is obviously dependent on people's individual feelings of art. Therefore, in this study we just offer a series of results yielded by the use of different sets of values of the two thresholds for the and choose the optimum image by using outlier technique and arty shapes detection process. The outlier detection is used in detecting the optimum segmented image from the images generated by finding out the images having minimum segmentation in order to perform the data embedding effectively.

The images which are segmented in large number are discarded since the data embedding becomes highly complex in such images. Shape simplification is a tool useful in Non-Photorealistic rendering from photographs, because it permits a level of abstraction otherwise unreachable. A variety of simple shapes (e.g. circles, triangles, squares, super ellipses and so on) are optimally fitted to each region within a segmented photograph. The system automatically chooses the shape that best represents the region; the choice is made via a supervised classifier so the "best shape" depends on the subjectivity of a user. The whole process is fully automatic, aside from the setting of two user variables to control the number of regions in a pair of segmentations.

Having segmented an image, we are able to fit a wide selection of shapes to each region. The data is first normalized by applying an appropriate transformation to put it into a canonical frame. The fitted geometric primitive is then simply obtained by taking the geometric primitive in the canonical frame and applying the inverse transformation to it. We are now able to optimally fit a collection of simple shapes to each region within a segmented image. The problem now is how to choose amongst them. Automatically selecting appropriate shape models is done using a supervised classification paradigm; a decision tree is learnt from a training set of regions which is then applied to new unseen data. The basis of a decision tree is that each feature can be used to make a decision that splits the data into smaller subsets; partitioning feature space into equivalence classes using axis parallel hyperplanes.It builds decision trees by selecting the most informative feature to split each subset. Entropy measure—Normalized

Information Gain—determines the effectiveness of each feature. The regions are described by a feature vector and are manually labeled into shape categories. These features are the basis for making the decision regarding which is the most appropriate model.

III DATA HIDING VIA LINE-BASED CUBISM-LIKE IMAGES

C. Idea of Proposed Data Hiding Technique

In the proposed Cubism-like image creation process described by Algorithm 1 above, one major step is to recolor the pixels in each image region with the average color of all the pixels in the region, resulting in an image visually looking like the source image. We take advantage of this step as well as a weakness of the human visual system in differentiating small color changes to design the data hiding technique in this study.

To be more specific, the proposed data hiding technique embeds message data into a cover cubism-like image by changing each pixel's color value in the cover image for the minimum amount of ±1in each color channel. As a result, people cannot tell the visual difference between the cover image and the stego image. This effect, in addition to that of attracting people by the artistic content of the Cubism-like image, gives the proposed data hiding technique a camouflage effect which arouses no suspicion from hackers. Furthermore, a reversible region recoloring scheme, which keeps the average color of each region unchanged, is designed as a substitute of the original recoloring process in Algorithm 1. This reversibility guarantees that we can extract the data embedded in the stego-image to restore the original content of the cover image losslessly. It is also noted that changing pixels' colors slightly while keeping average region colors unchanged, as proposed in this study, creates integrally a mosaic effect in the regions, which makes the stego-image look nearly identical to the cover image and thus enhances the camouflage effect of the proposed technique.

D. Principle of Lossless Data Embedding

In the proposed region recoloring process, when embedding a bit b into a pixel with color c, if is 0, then we decrement c by an integer value a, and if b is 1, then we increment c by a. After hiding message bits into the pixel's colors in a region by color shifting in this way, the region's average color will also be changed. It is found in this study that the property of rounding-off in integer computation may be utilized to modify this region recoloring process to keep the average region color unchanged, resulting in a reversible region recoloring process.

Lemma 1: The total numbers N_0 and N_1 of data bits of 0's and 1's, respectively, embeddable in an image region R with area A (in unit of pixel) by recoloring R by color shiftings of the amount ± a of without changing the average color C_0 of R in each of the three color channels is constrained by the following inequality:

$$-\frac{A}{2} \le (N_1 - N_0) \times a < \frac{A}{2}. \quad (1)$$

Theorem 1: When the data bits are embedded into the cover image to cause the minimum distortion in the mean square error (MSE) sense, the constraint of (1) becomes

$$-\frac{A}{2} \le N_1 - N_0 < \frac{A}{2}. \quad (2)$$

Corollary 1: When the stego-image is yielded with the minimum distortion by color shiftings of the amount of ±1 in region recoloring, the maximum numbers N_0 and N_1 of embedded 0's and 1's are constrained respectively by

$$\frac{A}{4} \le N_0 < \frac{3A}{4};\ \frac{A}{4} < N_1 \le \frac{3A}{4}. \quad (3)$$

And for the extreme case that the data bits are either all 0's or all 1's, the numbers N_0 or N_1 are constrained respectively by

$$N_0 \le \frac{A}{2};\ N_1 < \frac{A}{2}. \quad (4)$$

Theorem 2: Data embedding in a region by color shiftings of the amount of ±1 to embed 0's and 1's, respectively, under the constraint of (2) is lossless, i.e., the embedded data may be retrieved to resume the original stego-image perfectly.

E. Algorithm of Proposed Data Hiding Technique

The proposed data hiding process, which is based on thecreation process of the line-based Cubism-like image andthe lossless data embedding principle described previously,is composed of two stages—*data string randomization andembedding capacity computation; and data embedding*. In the first stage, at first we transform the data string to be hiddeninto a digit sequence of 0's and 1's and append an *ending pattern* (with at least one digit other than 0 and 1) to the endof the sequence to keep its length a multiple of three. By theending pattern, we can determine where the embedded data string ends in a sequence of extracted digits in the later data extraction process. Next, we try to obtain the information of two parameters of each region, namely, its area and average color, by performing Algorithm 1. Then, we use a secret key to randomize the order of the regions in the input image, and take the resulting sequence as the order for data hiding. For each region, in order to keep the average color of the region unchanged, we limit the embedded amount of message data bits in each region by the constraint of (2) in Theorem 1. In the secondstage, we embed the input data sequence by shifting the pixels colors for the amounts of ±1 according to the above-mentioneddata hiding order. After the data sequence is exhausted, theremight exist regions into which

no data is embedded. We dealfurther with these intact regions to keep the coloring style of allregions consistent. For this, we create a random binary stringof 0's and 1's with the bit numbers roughly constrained by (2), and use the same data embedding process to embed it into theintact regions. At the end, a stego-image is generated with theinput data string embedded imperceptibly. A detailed algorithmto implement these steps is given as follows.

Algorithm 2. Embedding a data string into a Cubism-like image created from a given image

Input: an image S, a secret key K_8, four random-number generator functions f_1, f_2, f_3, and f_4, and a message data string M in character form.
Output: a stego-image into which M is embedded.

Steps.
Stage 1—Data string randomization and embedding capacity computation.

Step 1. (*Randomizing and segmenting the data string*)Randomize and segment data string M by the followingsteps.

1.1 Transform M in character form into a binary string, and randomize the order of its bits to generate a new string M'using function f_1 with secret key K_8 as the seed.

1.2 Regard each bit of M' as a digit and append an endingpattern with at least one and no more than threeidentical digits d other than 0's and 1's to the end of M'to form a new digit sequence M" with its lengthbeing a multiple of three.

1.3 Divide M" into a sequence of 3-digit segments, m_1, m_2, .., m_N.

Step 2. (*Generating an art image and computing relatedparameters*) Generate an art image and compute theareas and average colors of the regions in the imageby the following steps.

2.1 Perform Algorithm 1 with S as the inputsource image to obtain an output artimage S' which has regions $R_1,R_2,\ldots,R_K$with areas $A_1,A_2,\ldots,A_K$ and average region colors (C_{1r},C_{1g}, C_{1b}), (C_{2r},C_{2g}, C_{2b}) ,…, (C_{Kr},C_{Kg}, C_{Kb}) respectively.

Step 3. (*Calculating the maximum data embedding capacityof each region*) Take sequentially an unprocessed R_I region in sequence *CS*, and compute the maximumdata embedding capacity Q_i of R_I by the followingsteps with the initial value of $Q_i$set to be zero.

3.1 Let (N_{r0}, N_{r1}), (N_{g0}, N_{g1}) and (N_{b0}, N_{b1}) denote the numbers of 0's and 1's embeddable in the R, G , and B color channels, respectively, in $R_I$with their initial values all set to be zero.

3.2 Take an *unprocessed* 3-digit m_i segment with digits d_r, d_g, d_b of data string M" and compute (N_{r0}, N_{r1}), (N_{g0}, N_{g1}) and (N_{b0}, N_{b1}) for d_r, d_g, d_b , respectively, in the following way:

(a) if d_r=0 , increment N_{r0} by 1; else, increment $N_{r1}$by 1;
(b) if d_g =0 , increment N_{g0} by 1; else, increment $N_{g1}$by 1;
(c) if d_b=0 , increment N_{b0} by 1; else, increment $N_{b1}$by 1.

3.3 If all of the following three inequalities hold:

$$-\frac{A'_i}{2} \le N_{r1} - N_{r0} < \frac{A'_i}{2};$$
$$-\frac{A'_i}{2} \le N_{g1} - N_{g0} < \frac{A'_i}{2};$$
$$-\frac{A'_i}{2} \le N_{b1} - N_{b0} < \frac{A'_i}{2},$$

then increase Q_i by 3; else, regard Q_i to have reached the maximum data embedding capacity for R_i according toTheorem 1 and go to Step 4.

3.4 If the data sequence M" is not exhausted, then go to Step 3.2 to repeat the above two steps.

Stage 2—Data embedding.

Step 4. (*Embedding the data*) Perform the following steps to embed data into region R_I of S_I.

4.1 Randomize the order of the pixels in R'$_i$ to generate an ordered pixel sequence HS = { P'$_1$, P'$_2$, …, P'$_t$ }using function f_3 with secret key K_8 and the indexi of $R_i$together as the seed.

4.2 Embed an unembedded 3-digit segment m_1 with digits d_r, d_g, d_b of sequence M" into an unprocessed pixel P'$_j$ with color values (C'$_{jr}$, C'$_{jg}$, C'$_{jb}$) taken sequentially from pixel sequence by the following steps.

(a) Obtain new color values (C"$_{jr}$, C"$_{jg}$, C"$_{jb}$) for P'$_j$ by modifying the original ones (C'$_{jr}$, C'$_{jg}$, C'$_{jb}$) in the following way for h= r , g , and b:

i. Increment C'$_{jh}$ by 1 if d_h = 1 ;
ii. decrement C'$_{jh}$ by 1 if d_h = 0;
iii. do nothing to C'$_{jh}$ if (an ending pattern digit).

(b) Recolor pixel P'$_1$by the new color values (C"$_{jr}$, C"$_{jg}$, C"$_{jb}$).
(c) Decrement the maximum data embedding capacity Q_i by 3.
(d) If Q_i is not equal to zero, then go to Step 4.2 to repeat the steps of (a) through (c) above.

Step 5. (*Ending of looping*) Repeat Steps 3 and 4 if sequence M" is not exhausted.

Step 6. (*Recoloring the intact regions*) Recolor each region $R'_K$with area A'_K in S', which has not been used fordata embedding so far, by the following steps.

6.1 Create a random digit string B with size $[A'_K/2]$ - 1 of 0's and 1's for R'_Kusing function f_4 with secret key K_8as the seed, and call B a *camouflage string*.

6.2 Perform Steps 3 through 5 to recolor the pixels of $R'_K$to embed camouflage stringB.

Step 7. Take the final image S" as the desired stego-image.

In the above algorithm, *wrap-around* problems in color values might occur in Steps of 4.2(a)-i and 4.2(a)-ii when the average region color value in any of the three color channels is 255 or 0. In such cases, we will obtain a stego-image with undesirable noise points. To avoid such extreme cases, we adjust the extreme average color values of 255 and 0 in any color channel to be 253 and 1, respectively, before data hiding.
Such slight color alternations in the generated stego-image will cause nearly no visual difference to human vision but will solve the wrap-around problem.

The lossless data embedding is composed of two stages data string randomization with embedding capacity computation and data embedding. In the first stage, at first we transform the data string to be hidden into a digit sequence of 0's and 1's and append an ending pattern to the end of the sequence to keep its length a multiple of three. Next, we obtain the information of two parameters of each region, namely, its area and average color. We use a secret key to randomize the order of the regions in the input image, and take the resulting sequence as the order for data hiding. In order to keep the average color of the region unchanged, we limit the embedded amount of message data bits in each region. In the second stage, we embed the input data sequence by shifting the pixels' colors for the amounts of ±1 according to the above-mentioned data hiding order. The security is provided by using a secret key for randomization of data string, region and for embedding data in intact regions.

F. Data Extraction Process

The proposed data extraction process, which is based on Theorem2, is basically a reverse version of the proposed data hiding process and consists of two stages—*embedded data extraction and data derandomization*. In the first stage, we recover the region recoloring sequence in the stego-image and obtain the area and the average color of each region in the stego-image. Based on the average color of each region, we retrieve the message data embedded in the stego-image by comparing it with the pixel's colors in the region. In the second stage, the retrieved data are derandomized to get the original message data using the secret key.

We adopt fourmeasures in Algorithm 2 to enhance the security of the proposed technique using a secret key:
(1) Randomization of the data string to be embedded;
(2) Randomization of the processing order ofthe regions;
(3) Randomization of the processing order of thepixels in each region;
(4) Embedding camouflage strings inintact regions to mislead a hacker to guess data in them erroneously.With these measures, the risk for the embedded data tobe stolen by a hacker is greatly reduced.

Algorithm 3: *extracting the hidden data string*

Input: a stego-image S ; and the secret key K_8 and three random number generators f_1, f_2, and f_3 used in Algorithm 2.
Output: the data string M embedded in S .

Steps.

Stage 1—Embedded data extraction.

Step1. (*Extracting the regions and related parameters*)Conduct region growing in a raster-scan order to find the regionsR_1 , R_2 ,… ,R_K in S , and compute their respective areasA_1 , A_2 ,… , A_K and average colors (C_{1r}, C_{1g}, C_{1b}), (C_{2r}, C_{2g}, C_{2b}) , …, (C_{Kr}, C_{Kg}, C_{Kb}) .

Step 2. (*Retrieving the region recoloring sequence*)
Randomize the initial raster-scan order of the regions R_1throughR_k to retrieve the region recoloring sequence CS = {R'_1, R'_2, …, R'_k} using f_2 with secret key K_8 as the seed, and change in accordance the orders of the areas and the average colors of the regions, resulting in the new ordered sequences of areas and average colors {A'_1, A'_2, …, A'_k} , and {(C'_{1r},C'_{1g},C'_{1b}), (C'_{2r},C'_{2g},C'_{2b}), …, (C'_{Kr},C'_{Kg},C'_{Kb})}, respectively.

Step 3. Create a message data sequence M' with an empty initial content.

Step 4. (*Extracting the embedded data*) Take an *unprocessed* region R'_i in CS and perform the following steps toextract the embedded data in R'_i to compose M'.

4.1 Randomize the initial raster-scan order of the pixels in R'_i to retrieve the pixel recoloring sequence HS = { P'_1, P'_2, …, P'_T }using function f_3 with secret key K_8 and the index of R'_I together as the seed.

4.2 Take sequentially an *unprocessed* pixel P'_j with color (C''_{jr}, C''_{jg}, C''_{jb}) in sequence .

4.3 Extract three embedded data digits d_r, d_g, and d_b from the difference values between the color(C''_{jr}, C''_{jg}, C''_{jb}) of pixel P'_j and the average color(C'_{ir}, C'_{ig}, C'_{ib}) of region R_i, respectively, by the following way, where h = r, g , and b :
(a) if $C''_{jh} < C'_{ih}$, then set tod$_h$ be 0;
(b) if $C''_{jh} > C'_{jh}$, then set to $d_h$be 1;
(c) if $C''_{jh} = C'_{ih}$, then set tod$_h$be d (an ending pattern digit).

4.4 If none of d_r, d_g, and $d_b$is d (the ending pattern digit), then appendd_r, d_g,d_b to M' and go to Step 4 to repeat Steps 4.1 through 4.3; else, ignore digits of d and append the remaining one(s) to M' .
Stage 2—Data derandomization.
Step 5. Reorder the digits in u M' using with secret key K_8 as the seed, regard the result as a binary string composed of 0's and 1's, and transform it into character form as the desired data string M .

Data extraction process, which is basically a reverse version of the proposed data hiding process.It consists of two stages are ***embedded data extraction*** *and* ***data de-randomization***. In the first stage, we recover the region re-coloring sequence in the stego-image and obtain the area and the average color of each region in the stego-image.Based on the average color of each region, we retrieve the message data embedded in the stego-image by comparing it with the pixels' colors in the region.In the second stage, the retrieved data are de-randomized to get the original message data using the secret key.

IV SYSTEM ARCHITECTURE

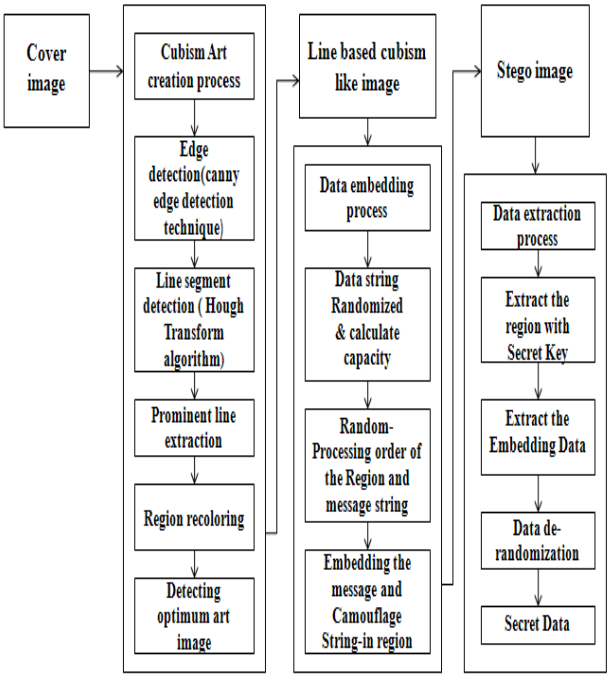


Fig.4 shows the system architecture of the proposed method.

The system architecture depicts the pictorial representation of the study proposed. The cover image is converted to a cubism art image through the Canny and Hough transform method which detects the edges and the line segment. Depending upon different threshold value a series of art images are yielded among which the optimum image is determined using art shape detection technique and outlier technique.foe embedding data we determine the area and embedding capacity of the regions and using random function and key as seed we randomize the region and message bits. Minimum amount of ±1 shifting in color channel of pixel is performed for embedding data. Retrieval of data is performed in the reverse order.

Fig 5 shows the experimental result of the theory proposed.

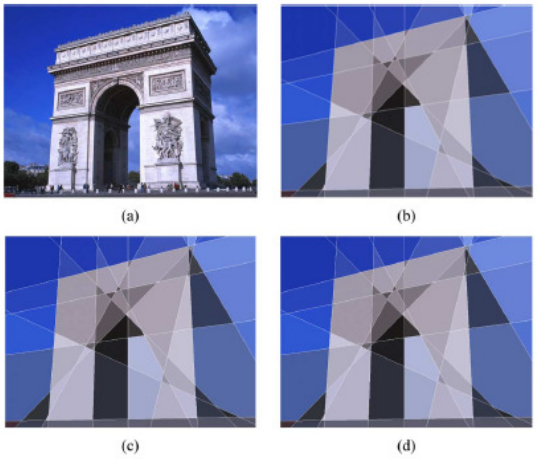


Fig.5 Experimental result. (a) Source image (cover image). (b) Generated Cubism-like image with no message data embedded. (c) Stego-image with color shiftings of 1. (d) Stego-image with color shiftings of 1 through 8.

V. CONCLUSION

In this paper, a new method of combining art image generation and data hiding to enhance the camouflage effect for various information hiding applications is proposed. At first, a new type of computer art, called line-based Cubism-like image,and a technique to create it automatically from a source imagehave been proposed. The method finds line segments in the source image by the Canny edge detection technique and the Hough transform, combines nearby line segments, extends the remaining lines to the image boundaries, and recolor the created regions by their average colors, to create an abstract type of the original source image as the desired art image. Then, by utilizing the characteristics of the Cubism-like image creation process, a data hiding technique has been proposed. Based on the minimum color shiftings of the values of , the technique embeds message data into the pixels of the regions of the generated art image while keeping the average region colors unchanged. The data embedding process is proved to be lossless by theorems so that the cover image can be recovered perfectly after the embedded message data are extracted.

The proposed method has several merits. First, it generates Cubism-like images as stego-images to distract the hacker's attention to the message data embedded in them. Also, by using the minimum color shiftings of to embed data bits, the resulting pixels' color differences between the generated. Cubism- like image and the stego-image are so small that a hacker will take no notice of the existence of the hidden data. Consequently, the proposed data hiding technique is very suitable for use in covert communication or secret keeping. Furthermore, four measures of randomization of the input message data and the processing order of them with a secret key and several random-number generating functions have been adopted in the proposed method. This enhances greatly the security of the proposed method.

REFERENCES

1. C. W. Lee and W. H. Tsai, "A lossless large-volume data hiding method based on histogram shifting using an optimal hierarchical block division scheme," J. Inform. Sci. Eng., vol. 27, no. 4, pp. 1265–1282, 2011.

2. Y. Z. Song, P. L. Rosin, P. M. Hall, and J. Collomosse, "Arty shapes" inProc. Computational Aesthetics in Graphics, Visualization &Imaging, Lisbon, Portugal, 2008, pp. 65–72

3. A. Hertzmann, "A survey of stroke-based rendering," IEEE Computer Graphics Applicat., vol. 23, no. 4, pp. 70–81, Jul./Aug. 2003.

4. Shan-Chun Liu and Wen-Hsiang Tsai,"Line-Based Cubism-Like Image—A New Type of Art Image and its Application to Lossless Data Hiding", IEEE Transactions,Volume 7,oct 2012.

5. C. K. Chan and L. M. Cheng, "Hiding data in images by simple LSB substitution," Pattern Recog., vol. 37, pp. 469–474, Mar. 2004.

6. Wikipedia and other online sources.