Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2019) 27: 532 – 546

doi:10.3906/elk-1801-131

StegoGIS: A new steganography method using the geospatial domain

Ömer KURTULDU[1,*], **Mehmet DEMİRCİ**[2]

[1]Department of Information Systems, Informatics Institute, Gazi University, Ankara, Turkey
[2]Department of Computer Engineering, Faculty of Engineering, Gazi University, Ankara, Turkey

Received: 15.01.2018 • **Accepted/Published Online:** 30.06.2018 • **Final Version:** 22.01.2019

Abstract: Geographic data are used on a variety of computing devices for many different applications including navigation, tracking, location planning, and marketing. The prevalence of geographic data makes it possible to envision new useful applications. In this paper, we propose using geographic data as a medium for secret communication, or steganography. We develop a method called StegoGIS for hiding messages in geographic coordinates in the well-known binary of fast-moving objects and transmitting them secretly. We show that discovering this secret communication is practically impossible for third parties. We also show that a large amount of secret data can be transmitted this way, and that the receiver can easily extract the hidden message. We argue that StegoGIS offers a novel and practical secret communication method with minimal overhead.

Key words: Steganography, geospatial, open geospatial consortium, well-known binary

1. Introduction

Steganography or covered writing is the ancient art of hiding secret messages inside various media to transmit them without raising any suspicion [1]. Steganography is used for both legal purposes such as watermarking [2] as a protection against copyright infringement, secure transmission, and cultural residue protection and for illegal purposes such as spying. There are three main components in a steganographic transmission: the cover object, secret message, and secret key. The secret message can be in different forms such as text, image, or video. Typically, a secret key is used to encrypt the secret message. The cover object is generally chosen from common media files such as images [3, 4], torrent files [5], text [6], geographical objects [7], or network packets [8].

Geographical objects are composed of raster and vector data. Whereas raster data are based on a tessellation of image cells, vector data are based on vertices and paths. They can be categorized into three geometrical elements [9], which are points, lines, and polygons. Vector data are formed by spatial data, or in other words a sequence of coordinates describing the geographical locations of map objects, attribution data, and additional data. Vector data represent real-world features within a geographic information system (GIS) environment. GIS is used for processing geographical data as well as storing both the geometry and the attributes of these data.

The rapid development of the Internet makes it easy to exchange information via networks. In particular, emerging web-based applications make information available to both public and private clients. In addition, web services are developed to support interoperable machine-to-machine interaction over a network. Based on web

*Correspondence: omerkurtuldu@gmail.com

services, GIS capabilities can be used over the network for flexible and extensible applications. An international consensus organization named Open Geospatial Consortium (OGC) (http://www.opengeospatial.org) defines open standards for geospatial content and services for defense, environment, health, agriculture, meteorology, and many more areas. OGC standards are used to retrieve geographic data from a server or from multiple distributed servers into a single display for visual representation of the chosen georeferenced system.

The Spatial Reference System (SRS) (or Coordinate Reference System, CRS) is used to locate geographical entities based on coordinates. SRS/CRS provides ways to make transformations between different spatial reference systems as mentioned in OGC 06-103r4 [10], and 063r5-CRS [11] describes the coding of SRS/CRS as Well-Known Text (WKT) or Well-Known Binary (WKB). WKT can be read by both humans and computers, whereas WKB is utilized in SQL implementations of geometric objects in binary format. In this work, StegoGIS refers to the application of steganography in WKB.

The three most known services of OGC are Web Map Service (WMS), Web Feature Service (WFS,) and Web Coverage Service (WCS). WMS is an HTTP interface for requesting map images such as png, jpeg, gif, and tiff from one or more distributed geospatial databases, while WFS is an HTTP interface for requesting geographical vector features. WFS includes querying or retrieval of features from geospatial databases or files [12]. Transactional Web Feature Service (WFS-T) can be used to manipulate features through queries such as update feature. Geospatial data can be discovered, queried, or transformed using WFS. WCS can be used to retrieve geospatial data containing space and time information.

While GIS explains where geographic information is stored and how to integrate with geographic applications to exploit geographic data on the map, geospatial can be defined as the geographic location of features and may contain attribute data. Spatial data are usually stored as coordinates, topology, SRS, and type such as points, etc. These data can be mapped, accessed, manipulated, or analyzed in GIS. Spatial data represent not only location, but a collection of information connected to the location of objects.

Today, applications of GIS include defense, scientific research, commercial products, mobile applications, and social media. Increasing usage of GIS outside secure areas makes it vulnerable for steganography. Due to the huge amount of geographical map data processed and distributed all over the world, secret messages can be hidden in geographic data, particularly coordinates of vector data compatible with OGC standards.

In this paper, we examine how to exploit the coordinate information of geographical objects in the geospatial domain within the OGC WFS specification and propose a novel method for steganography. Our method replaces the real values of spatial data with slightly modified values to contain bits of a secret message while keeping coordinate deviations visually unrecognizable by humans. To embed secret messages, we chose airplanes because of the large number of flights and the availability of online tools for following geographic coordinates of airplanes.

This paper is organized as follows: Section 2 discusses related works on raster image and vector data steganography. Section 3 explains the proposed method for vector steganography based on OGC specification and the geospatial domain. Section 4 presents experimental results of the proposed method, and Section 5 summarizes the paper and lays out future research areas.

2. Related works

This section contains a discussion of works in the literature related to raster image steganography based on WMS, discrete cosine transformation (DCT), least significant bit (LSB), and vector data steganography.

A steganographic method associated with GIS jpeg images such as aerial photography was suggested in the study by Hebbes et al. [13]. In the proposed scenario, the output is required to be a compressed photo-realistic image, the format for which has been selected as JPEG. The cover image is divided into 8×8 pixel blocks and DCT is applied to each block. DCT coefficients are quantized by quantization tables. Finally, an entropy encoder is used to output a jpeg-compressed image. In order to embed data or image transformation information into the cover image, it is essential that the embedding process be lossless and that the message can be retrieved exactly although the image is compressed. In the embedding process, an irregular walk is used to increase security.

Klubsuwan and Mungsing [14] proposed a design and algorithm based on steganography on 3D Video GIS-Map, which defines a 3-dimensional video that contains geographical location on the earth. The cover object is an AVI file, which contains 3000 frames in BMP format. In the proposed method, the irregular pixel is selected in the second frame and the embedding process continues in sequential order for the other frames. The proposed algorithm processes sequentially to the LSB for DCT-LSB format. The secret data are embedded into the cover object by substituting one bit in every pixel.

GIS vector data is vulnerable against deterioration due to requirements of high fidelity and high precision. For this reason, applying traditional digital watermarking to GIS data will inevitably cause some loss of quality. Reversible watermarking or lossless data hiding is more suitable for vector maps. In a related study [7], a method of information disguising and reduction for GIS data was used to prevent illegal use of sensitive GIS information. The proposed method is based on the difference expansion principle and tested in only the shape file format but can be applied to others such as GML, E00, and MIF.

Based on context-sensitive texture adaptation along with GIS service data over a standards-based communication, a method was suggested to hide robust steganographic markings into the rendering process for GIS data by Wolthusen [15]. The proposed method starts with removing existing texture information using the Gaussian smoothing method and detecting edges using the Canny edge detector. Then segments with continuous edges are derived for the embedding process. Finally, the algorithm can be applied directly to image material.

Web services based on service-oriented architecture (SOA) can be used to set up a flexible and extensible GIS in order to provide quick response to clients or organizational requests. However, a large number of requests sent by clients is a drawback for GIS, and Yershov et al. [16] investigated this problem. The aim of the study was to decrease the amount of geographic requests by using extensions to existing GIS visual analysis tools and a GIS SOA synchronization mechanism related to steganography. The proposed method is based on the redundancy of visual information. In other words, the hiding of bits always happens in the raster image. In this way, it is claimed that the geospatial data streams will be kept, transmitted, and analytically processed with steganography.

Above, we review recent studies on steganography related to GIS data. Studies described in [13] and [14] are about GIS images but they are not related to geographical data. From this perspective, our study is different from [13] and [14]. Meanwhile, studies in [15] and [16] are about WMS images and similar to conventional image steganography. Our study focuses on WFS that contains vector GIS data. Lastly, [7] explains a method for changing vector GIS data based on the difference expansion principle to hide secret messages. This change occurs in the visible coordinate data, represented by ten-value digits. However, in our proposed method, changing GIS data occurs in the unusual GIS domain represented by hexadecimals.

3. Proposed method

As explained in the OpenGIS Implementation Standard [17], there exist multiple tables in a database schema supporting SQL implementation. These tables include features, geometry, and spatial reference information. Geometry information contains not only the coordinates of objects but also the reference system, dimension, etc. In this paper, we call the entirety of this geometry information the geospatial domain.

In this section, we explain how to design the proposed system architecture, how to choose cover points, how to initialize these points, and how to embed and extract hidden messages from the geospatial domain. The general block diagram of the system is illustrated in Figure 1. Each of the main steps is described in the following subsections.

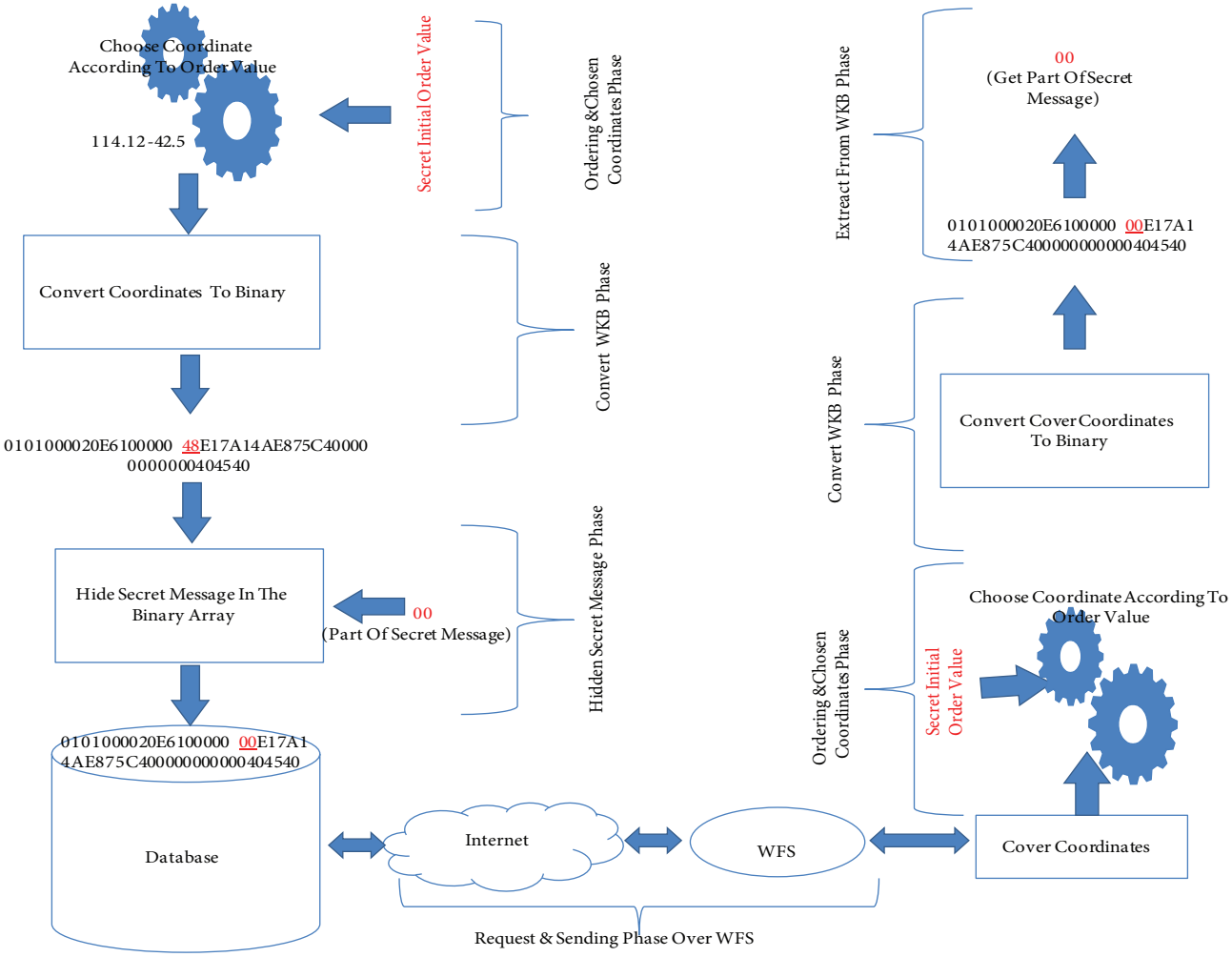

Figure 1. General block diagram.

3.1. System architecture

The system architecture is displayed in Figure 2. We use Postgresql with Postgis as the database system, Geoserver as the map server, and Java as the coding platform to hide and extract secret data. In this architecture, geospatial data are loaded from the database to the map server. To simulate airplane movement, we calculate

the airplane's next position every 1 min and refresh the table, which is connected to the map server, to publish up-to-date positions of airplanes to clients. When a client requests layer information from the map server, all the features are sent to the client over HTTP according to the requested type, such as JSON. All the requests can be made with HTTP URL written by the user manually, or with an application such as web pages or desktop software. To get the requested data, we use a Java application once every minute, which is a common action for monitoring platforms that are moving according to time series.

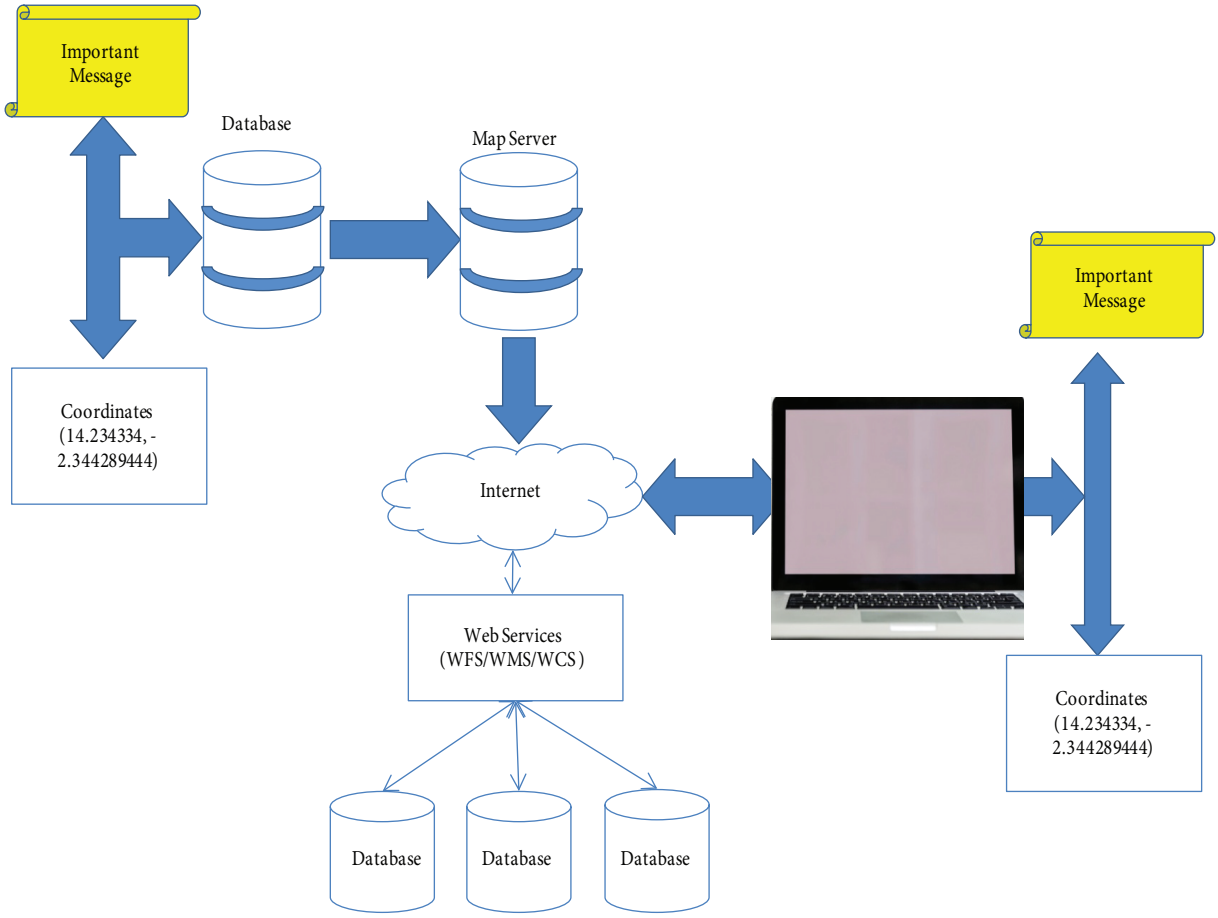

Figure 2. System architecture.

3.2. Unit selection

At any moment, there are about 10,000 flying airplanes in the world. All these airplanes can be exploited for sending secret data using steganography. Due to the high complexity of running simulations with 10,000 airplanes, we chose 1021 air routes in the USA airspace. In this section we answer the question of how to choose target airplanes in the right order. First of all, we partition all airplanes into clusters according to their distance from centers of clusters. We determine constant and noncommon positions for 5 centroids, and cluster iteration is chosen as 1 to decrease the probability of prediction. We use the k-means algorithm for this process. When we experiment with finding the clustering with the optimum group count, we determine the optimum number as 12 using sum squared error (SSE) analysis.

Algorithm 1 Point initializing and reordering algorithm.

```
1: procedure INITIALIZE CLUSTER VALUE(PointsList)
2:     center[1] ← Point(x, y)
3:     center[1]..center[5] ← Point(x1, y1)..Point(x5, y5)
4:     size ← PointsList.size
5:     for i=0..size do
6:         distance ← 1000000.0
7:         for j=1..5 do
8:             result ← Calculate − Distance − Between(PointsList[i], Center[j])
9:             if result < distance then
10:                Initialize − Point − Cluster − Value − As(j);
11:            end if
12:        end for
13:    end for
14: end procedure
15: procedure INITIALIZE ORDER VALUE(PointsList)
16:    f() ← f(secret − key)
17:    for i=0..size do
18:        Initialize − Point − Order − Value − As(PointsList[i], f().next);
19:    end for
20: end procedure
21: procedure REORDERING(PointsList)
22:    for i=0..size do
23:        for j=i+1..size do
24:            if PointList[i] > PointList[j] then
25:                swap(PointList[i], PointList[j]);
26:            end if
27:        end for
28:    end for
29: end procedure
```

We assign a cluster number to all air routes according to the minimum distance from centers of clusters. Lastly, we order all air routes by increasing complexity of prediction. We use the same air routes for embedding and extracting hidden data. The algorithm explained above is presented below in algorithmic form.

3.3. Embedding process

In this subsection, we explain the embedding process, which is a critical part of this study. First, we convert the airplane's position into the geospatial domain, which is defined in OpenGIS Implementation Specification for Geographic Information - Simple Feature Access - Part 2: SQL Option Version 1.2.1 [17]. This specifies an SQL schema that supports storage, retrieval, query, and update of geospatial features with simple geometry. Implementation can be different from one database to another but it is important that the logic is the same. Postgresql with postgis supports this specification and stores the position of tracks in the type of geometry. Apart from this, positions are represented by their hexadecimal equivalents.

In this form, position weights can be changed according to different implementations. In the chosen database between 19th-22nd and 34th-38th characters are used for small weights of positions in which the secret message can be embedded with tiny differences between the original and the modified position. As the position contains latitude and longitude information, we can embed 8 hexadecimal digits (32 bits or 4 bytes) in one air route position in the geospatial domain. Generally speaking, the geometry information is necessary for any map server without any latitude or longitude information. Hence, changing the long geometry type is unlikely to be caught by anybody. The embedding process is shown below.

Algorithm 2 Embedding algorithm.

```
procedure EMBEDDING SECRET DATA(OrderedPointsList, SecretDataAsHexadecimal)
    hexaData ← SecretDataAsHexadecimal.split − quad − tuple
    size ← OrderedPointsList.size
    embeddedCounter ← 0
    while embeddedCounter + 2 < hexaData.size do        ▷ until sending secret data is finished
        for i=0..size do
            geoSpatialDomain ← OrderedPointsList[i]
            HideFirstQuadTupleInLatitudeValue[embeddedCounter];
            HideSecondQuadTupleInLongitudeValue[embeddedCounter + 1];
            embeddedCounter ← embeddedCounter + 2
        end for
    end while
end procedure
```

3.4. Extraction process.

In this part, we explain the extraction process by which the secret data can be retrieved by the client or clients. Requests can be made using web pages or other applications at regular intervals. Another way is writing a special application to retrieve objects in the particular time series. For this purpose, we wrote a simple WFS client application for calling the WFS layer from Geoserver. Second, we parsed the received JSON format into a text file. Third, we clustered, assigned order value, and reordered in the same way as unit selection explained in Section 3.2. Fourth, all points of air routes are converted into the geospatial domain. Finally, the extraction process is performed by choosing special hexadecimal values from the type of geometry. This process is shown below.

4. Experimental evaluation

4.1. Simulation setup

We obtain our experimental results on a personal computer with an Intel i5-5200 2.2 GHz CPU with 8 GB memory. The database, map server, and client application are on the same computer. This architecture is demonstrated in Figure 3.

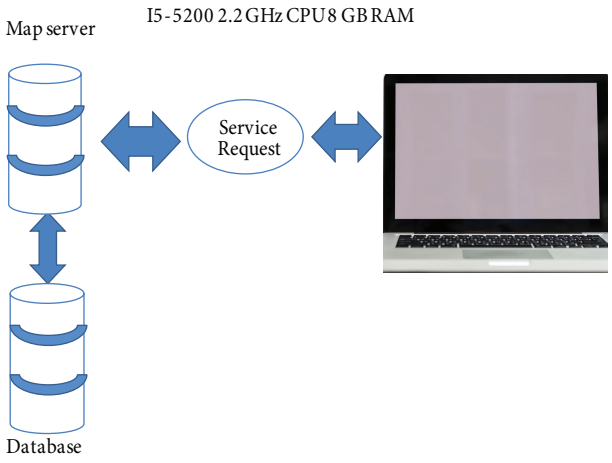

Figure 3. Simulation architecture.

The map server is connected to the database over an HTTP connection to retrieve geospatial information,

Algorithm 3 Extracting algorithm.

```
    procedure EXTRACTING SECRET DATA(url)
        embeddedCounter ← 0
 3:     hexaData ← 0
        while embeddedCounter + 2 < hexaData.size do          ▷ until receiving secret data is finished
            featureList ← GetWFSFromURL(url)
 6:         featureSize ← featureList.size
            procedure INITIALIZE CLUSTER VALUE(featureList)
                center[1]..center[5] ← Point(x1, y1)..Point(x5, y5)
 9:             for i=0..size do
                    distance ← 1000000.0
                    for j=1..5 do
12:                     result ← CalculateDistanceBetween(featureList[i], Center[j])
                        if result < distance then
                            InitializePointClusterValueAs(j);
15:                     end if
                    end for
                end for
18:         end procedure
            procedure INITIALIZE ORDER VALUE(featureList)
                f() ← f(secret − key)
21:             size ← featureList.size
                for i=0..size do
                    Initialize − Point − Order − Value − As(PointsList[i], f().next);
24:             end for
            end procedure
            procedure RE-ORDERING(featureList)
27:             size ← featureList.size
                for i=0..size do
                    for j=i+1..size do
30:                     if featureList[i] > featureList[j] then
                            swap(featureList[i], featureList[j]);
                        end if
33:                 end for
                end for
            end procedure
36:         for i=0..featureSize do
                geoSpatialDomain ← featureList[i]
                hexaData.add(Get4HexaFromLatitude[embeddedCounter]);
39:             hexaData.add(Get4HexaFromLongitude[embeddedCounter]);
            end for
        end while
42: end procedure
```

which is changing every minute to simulate airplane movements. The Java client application connects directly to the map server, and all requests by clients as well as responses by the map server are transmitted over HTTP. We use a Postgresql trigger to embed the secret message in the geographic object's geospatial domain. We produce secret messages and simulation airplane tracks using Java.

4.2. Evaluation of security

The probability of extracting the hidden message from the geographical unit's coordinates is calculated in Eq. (1), where P_T is defined as the probability of finding the correct order of units, P_A as the probability of finding the sequence of cluster centers, P_B as the probability of finding the number of clusters, and P_C as the probability of finding the order of chosen units.

$$P_T = P_A \times P_B \times P_C \tag{1}$$

We chose air routes over the USA as our test area and selected cluster centers arbitrarily. Hence, even if there are small alterations in the centers of clusters, sizes of clusters change. For this reason, whoever wants to extract the secret message from unit coordinates has to find out the correct centers of clusters with a small error, the upper limit of which was experimentally derived as 1.6 km. Therefore, assuming the area from 27 to 45 North and from -124 to -66 West is chosen as the target area, the distance between -124 and -66 West is 5691.25 km, and the distance between 27 and 45 North is 2001.41 km. If we accept the radius of the circle as 1.6 km, we can fit 1,111,250 circles into the given area. Then, by calculating the probability of guessing 5 correct centers in the correct order out of the given 1,111,250 centers, P_A can be found as in Eq. (2).

$$P_A = \frac{1}{1111250} \times \frac{1}{1111249} \times \frac{1}{1111248} \times \frac{1}{1111247} \times \frac{1}{1111246} \approx 5.9 \times 10^{-31} \tag{2}$$

We defined the number of clusters as 5, but this is not required. To illustrate the effect of cluster value, we utilize the sum of squares error (SSE) to compare cluster sizes. SSE is a statistical technique used in regression analysis, which is a mathematical approach to determining the dispersion of data points. Using SSE, we can determine how well a data series can be fitted to a function, which might help to explain how many airplane clusters should be generated. Eq. (3) represents the residual sum of squares (RSS), where y_i is the ith actual value of the variable to be predicted and $f(x_i)$ is the predicted value of y_i.

$$\sum_{n=1}^{n} (y_i - f(x_i))^2 \tag{3}$$

When we applied the chosen initial airplane's coordinates in R, the calculated result for the optimal number of clusters was 12, as shown in Figure 4.

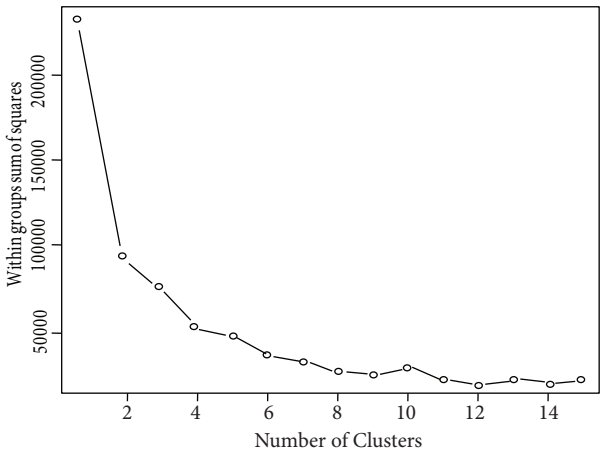

Figure 4. Finding appropriate K means group using SSE in R plot.

We used nonregular iteration for the proposed method to enhance security. We chose the iteration value as 1 since the optimality of clustering is not important in our case, and an adversary predicting an unconventional number of iterations will be less likely. However, if we used the regular pattern in R, cluster groups members are very different, as shown in Figure 5 and Figure 6.

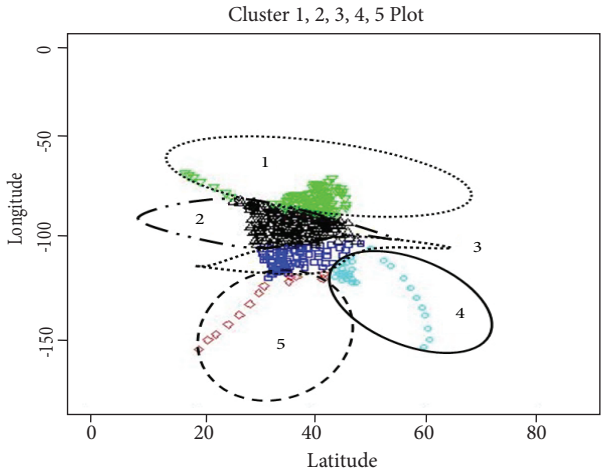

Figure 5. Proposed clusters plot.

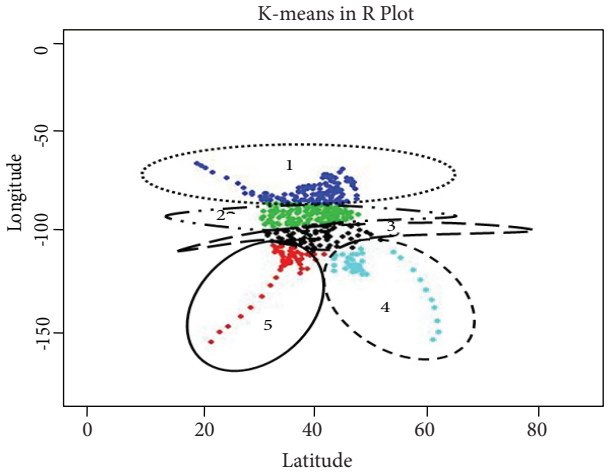

Figure 6. Common clusters plot.

We can calculate P_B as in Eq. (4).

$$P_B = \frac{1}{12} = 0.08\overline{3} = 8.\overline{3} \times 10^{-2} \tag{4}$$

It is an assumption that a long secret key value used in the embedding and extracting phases is known by both the sender and the receiver. The chosen long value is used to reorder the clusters so that the embedding and extraction phases can be done in the same order. Starting with Java 8, the unsigned long data type is represented by 64 bits and can store values between 0 and $2^{64} - 1$, so P_C can be calculated as in Eq. (5).

$$P_C = \frac{1}{2^{64}} \approx 5.4 \times 10^{-20} \tag{5}$$

Finally, we can calculate the total probability of prediction as in Eq. (6).

$$P_T \approx 5.9 \times 10^{-31} \times 8.\overline{3} \times 10^{-2} \times 5.4 \times 10^{-20} \approx 2.7 \times 10^{-51} \tag{6}$$

Therefore, we conclude that the probability of prediction is negligible. According to the website top500.org, Sunway TaihuLight, built in China, has been the fastest supercomputer in the world since June 2016. The number of operations per second for Sunway TaihuLight is expressed as 93.01 petaflops (quadrillions of floating-point operations per second). In other words, it can perform up to 9.301×10^{16} operations per second. The average amount of time to break our method using the fastest supercomputer can then be calculated as $\frac{1/2}{2.7 \times 10^{-51} \times 9.301 \times 10^{16}} \approx 2 \times 10^{33}$ seconds, or 6.3×10^{25} years, which is an unfeasibly long time, showing the strength of our method.

4.3. Evaluation of capacity

One of the big challenges for steganography methods is capacity [18]. In the simple image LSB technique, every pixel's bit value can be either 0 or 1. Unlike LSB, StegoGIS allows embedding 4 hexadecimal values into each latitude and longitude with a small distortion. Assuming 8 hexadecimal values can be hidden in each coordinate, the sum of 1021 secret messages equals 8168 hexadecimal values or 32,672 bits for each movement.

An airplane typically moves so fast that the secret message can be updated every 6 s. Hence, the transmission rate of the hidden message is 5.32 kbps for 1021 airplanes. Thus, in just 1 h, roughly 2.34 MB of data can be transmitted over the network.

Another measure of capacity in steganography methods is the amount of hidden data per one bit of the cover object. In this work, hidden data are embedded after the conversion to WKB, where latitude and longitude values are represented by 8 bytes each. Up to 2 bytes of hidden data can be embedded in each value, so the ratio of hidden data to cover data is 25% or 0.25 hidden bits per cover bit, which is a high ratio implying good capacity.

Various steganography methods developed for RGB images were compared in terms of capacity using the peak signal-to-noise ratio (PSNR) metric [19]. The highest capacity was observed for the method by Brisbane et al.[20] with a PSNR of 40 and a capacity of 6 bytes per pixel. An RGB pixel is 24 bytes, so the ratio is 25% or 0.25 secret bits per image bit. Hence, our method provides the same capacity as the highest-PSNR method reviewed in that work [19].

4.4. Evaluation of effectiveness

In our system, the coordinates of airplanes are different from their actual positions due to the secret data embedded into the hexadecimal values of coordinates. However, this difference must be kept to a minimum to make the existence of steganography practically undetectable. For this reason, we evaluated this positional variation between the actual coordinates and modified coordinates, as in a study by Szczypiorski and Tyl [21]. We simulated 1021 air routes 30 times; hence, 30,630 positions were evaluated.

Table. Comparison of deviation in meters for the proposed method and the alternative method of adding into coordinates value.

	Deviation for proposed method (m)	Deviation for adding method (m)
Sum	196.224	1417.013
Minimum	0.095	0.095
Maximum	0.135	412.179

As seen in the Table, the total, maximum, and minimum distances between actual and modified positions (i.e. deviations) are about 196.224 m, 0.095 m, and 0.135 m for the proposed method. Therefore, the proposed method's positional changes are minuscule and cannot be detected through visual analysis. However, if we add secret messages directly to the actual coordinates by overwriting the last few digits instead of using our proposed embedding method, we observe significant increases in the difference between actual and modified coordinates. Hence, our embedding method is better at producing low deviations.

As an additional measure of effectiveness, we focus on PSNR and MSE (mean squared error) values. A higher PSNR value implies less distortion in the cover medium/object. A high PSNR is also related to a low MSE value. To this end, we look at image steganography methods in the literature [19] providing secret message capacity comparable to that of our method. In that work, the highest PSNR value was reported as 40 dB [20], whereas the PSNR of our method is calculated as 233.7 dB for latitude (where maximum latitude is 90) and 233.4 dB for longitude (where maximum longitude is 180). Also, MSE values of our method are calculated as 3.46×10^{-20} for latitude and 1.50×10^{-19} for longitude. As we can see, PSNR is much higher for our method than for the other method in the literature. However, this does not necessarily mean that our method is superior, because steganography approaches and carrier media are different between the two

studies. If other methods for steganography in GIS are developed in the future, more meaningful comparisons will become possible. Nevertheless, even without such a comparison, the high PSNR of our method is a good indication of its effectiveness.

4.5. Evaluation of visual analysis

Visual analysis [22] is often used as a steganalysis tool to detect steganography. After the embedding phase, we explored whether StegoGIS objects are recognizable by visual analysis. To figure out the differences between original cover objects and StegoGIS objects, we selected one airplane's 30 tracks, which represent 1-min movements in an air route located north of New Orleans. We draw both the cover object's and the StegoGIS object's 30 positions with different map scales. Figures 7 and 8 display the cover object's path with the color blue and the StegoGIS object's path with the color red. The scale is too small for red and blue dots to be distinguished in the first six snapshots. Due to the overlap, only the red dots are shown. In the last two snapshots, we can see and distinguish both colors. This shows the ineffectiveness of visual analysis against our method since such a small deviation from the path will either not be recognized or will be attributed to natural factors and the inherent error in remote location tracking. Hence, we conclude that there is no observable

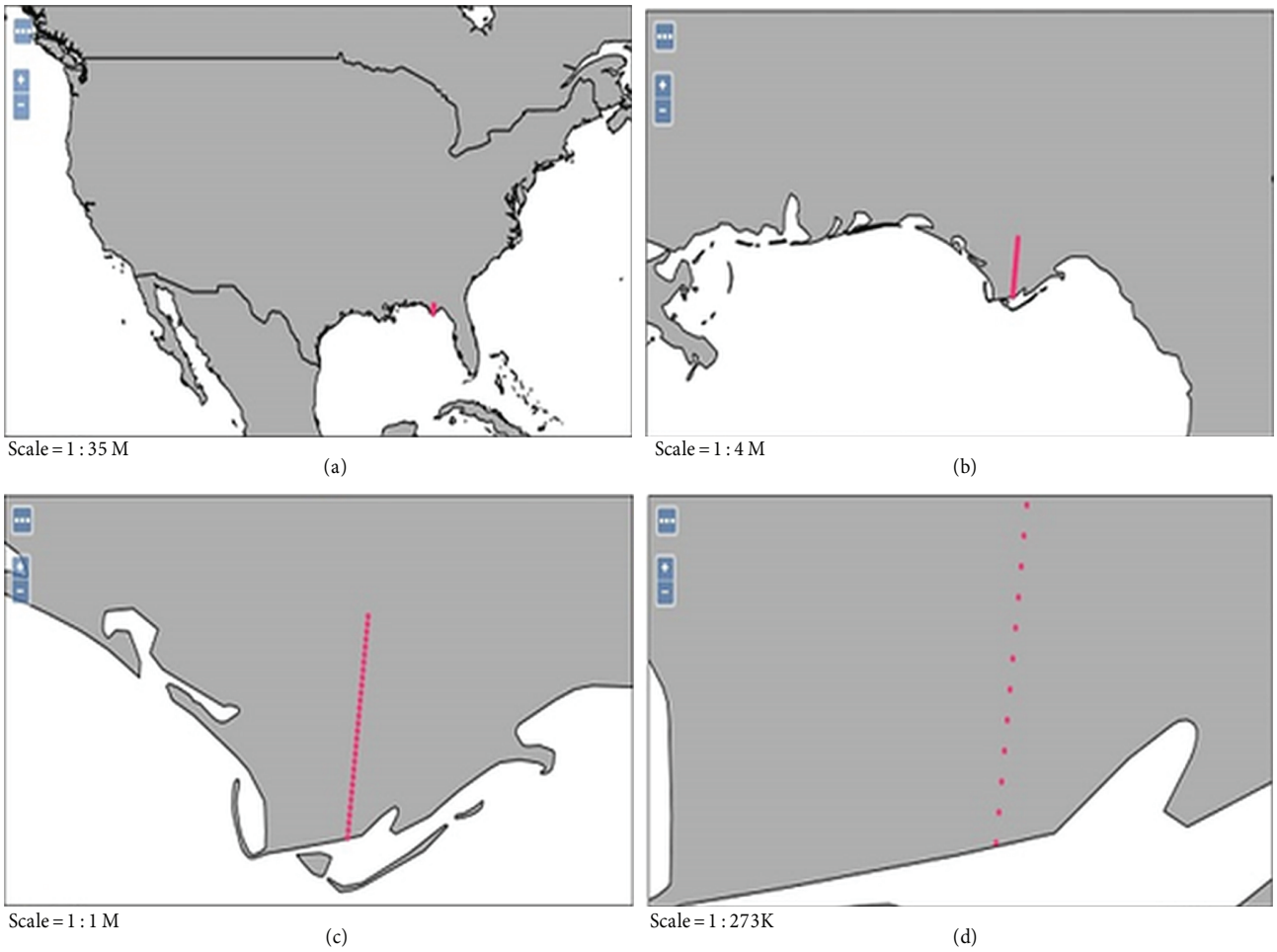

Figure 7. Comparison 1 with cover object (blue star) and StegoGIS object (red point) in small scales.

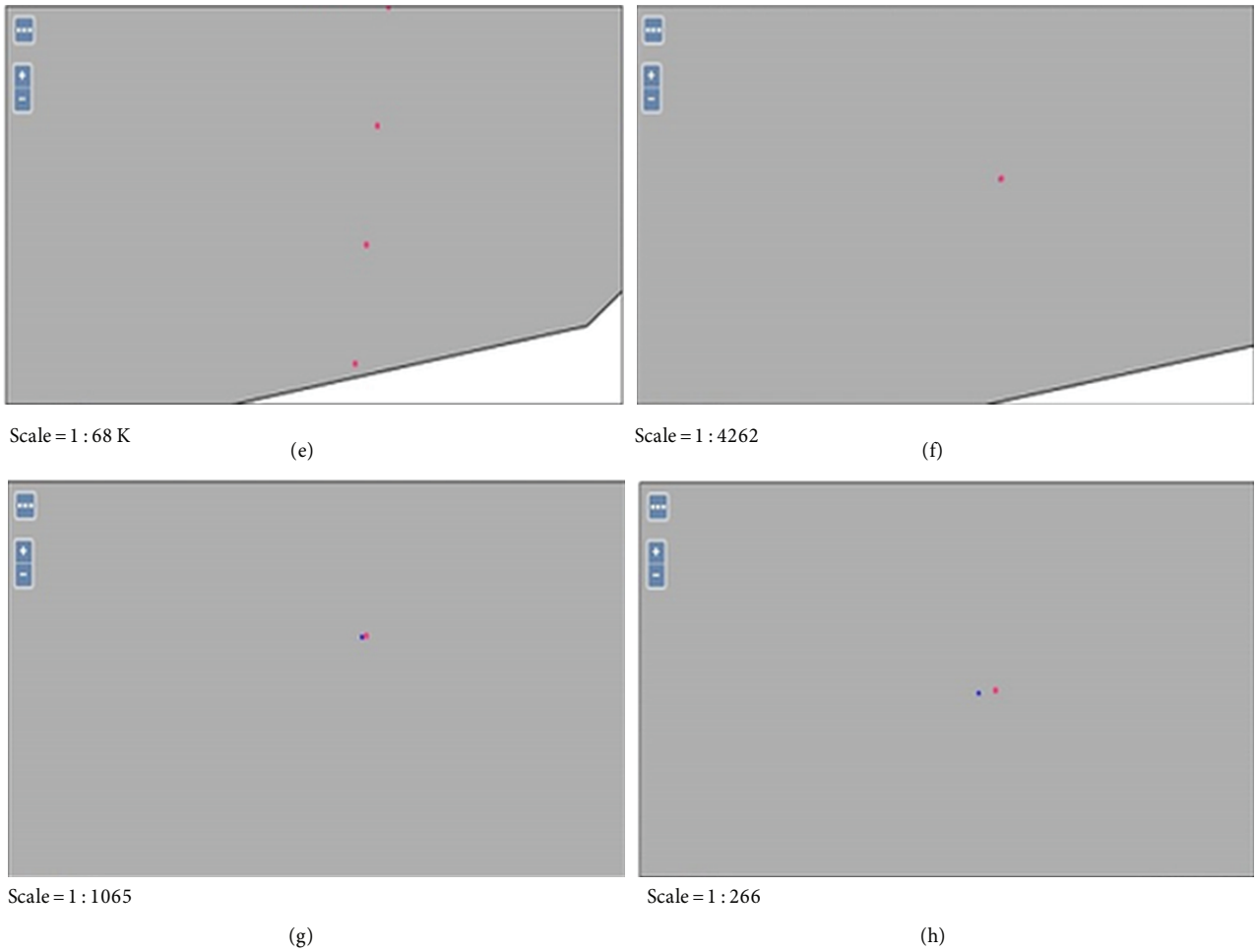

Figure 8. Comparison 2 with cover object (blue star) and StegoGIS object (red point) in big scales.

deviation in the paths until the scale gets to about 1:1000, and even that deviation is too insignificant to be recognized through visual analysis.

5. Conclusion

In this paper, we propose a novel steganography method utilizing the geospatial domain and provide its proof of concept. The hypotheses were tested with simulated airplane data following real airways over the USA. We chose simulated airplanes so as to be able to transfer the necessary amount of data over the network and improve security due to worldwide common usage. First, we collected data for the simulation and clustered them according to chosen special points over the USA. We then embedded hidden messages using conversion of coordinates of airplanes in the geospatial domain.

Our work demonstrates that a large amount of hidden data can be sent using fast-moving units such as airplanes over the Internet without raising any suspicion. If a hidden message is embedded in 10,000 airplanes' coordinates and updated once every 6 s, secret data can be transmitted with a rate of over 5 kbps. The differences between the normal path and the modified path of the airplanes, as calculated via variation of distance and observed via visual representation, do not suggest any significant and observable route change. In addition, we used nonstandard clustering and reordering of units to enhance the security of our method. Meanwhile, should

WFS-T be online, a receiver can send messages to a sender through HTTP over the Internet, as well. Therefore, bidirectional steganography can be used between the sender and the receiver. The arguments given above prove that steganography can be used in GIS.

Despite the abundance of steganalysis methods to uncover steganography, there is currently no effective method against steganography in the GIS domain. Traditional techniques rely on detecting pixel changes in images via probabilistic methods, identifying frequency anomalies in text, monitoring against potential misuse of padding areas in IP packets or deliberate network delays, and so on. However, messages embedded in binary format are harder to detect because changes will not be visible. As a result of this study, we suggest that periodically comparing transmitted binary values with actual coordinate values would be a prudent approach. Still, the presence of secret messages could be overlooked or considered an error because steganography is usually applied at unpredictable times and in limited scope. Additionally, in applicable cases, it would be useful to transmit coordinates with 2-4 decimal places to thwart steganography by making visual analysis more successful.

A number of areas for future research should be mentioned. First, performing steganalysis over GIS is essential to avoid misuse of geographic objects, so evaluations of existing and new steganalysis methods should be made in this domain. Second, this study can be applied over mobile vector objects such as polygon shapes. Third, bidirectional steganography can be implemented in GIS. Finally, different domains can be tested over GIS data to increase the amount of secret data in the cover objects using a combination of geospatial and spatial coordinates.

References

[1] Kurtuldu Ö, Arica N. A new steganography method using image layers. In: ISCIS 2008 23rd International Symposium on Computer and Information Sciences; 27–29 October 2008; İstanbul, Turkey. New York, NY, USA: IEEE. pp .1-4.

[2] Alla K, Shankar GG, Subrahmanyam GB. Secure transmission of authenticated messages using new encoding scheme and steganography. In: Proceedings of the Second International Conference on Computational Science, Engineering and Information Technology; 26–28 October 2012. New York, NY, USA: ACM. pp. 749-752.

[3] Arica N, Kurtuldu Ö. Image steganography by wavelet matching. J Electron Imaging 2009; 18: 033013.

[4] Wu KC, Wang CM. Steganography using reversible texture synthesis. In: IEEE TIP IEEE Transactions on Image Processing; 14 November 2014. New York, NY, USA: IEEE. pp. 130-139.

[5] Mazurczyk W, Szczypiorski K, Lubacz J. Four ways to smuggle messages through internet services. In: IEEE Spectrum; 2013. New York, NY, USA: IEEE. pp. 42-45.

[6] Ren W, Liu Y, Zhao J. Provably secure information hiding via short text in social networking tools. Tsinghua Sci Technol 2012; 17: 225-231.

[7] Li AB, Li SS, Lv GN. Disguise and reduction methods of GIS vector data based on difference expansion principle. Procedia Engineering 2012; 29: 1344-1350.

[8] Zielinska E, Mazurczyk W, Szczypiorski K. Trends in steganography. Commun ACM 2014; 57: 86-95.

[9] Niu XM, Shao CY, Wang XT. GIS watermarking: hiding data in 2D vector maps. SCI 2007; 58: pp.123–155.

[10] OpenGIS Implementation Standard for Geographic Information. OGC. Version 1.2.1.

[11] Geographic information Well known text representation of coordinate reference systems. OGC. Version 1.0.

[12] OGC Web Feature Service 2.0 Interface Standard With Corrigendum. OGC. Version 2.0.2.

[13] Hebbes L, Janjua F.Y., Livingstone D, Orwell J. A steganographic method for the secure embedding of GIS data streams into aerial photography. In: Joint IST Workshop on Mobile Future, 2006 and the Symposium on Trends in Communications; 24-27 June 2006. Bratislava, Slovakia: SympoTIC'06.

[14] Klubsuwan K, Mungsing S. Digital data security and hiding on virtual reality VDO 3DGIS-map. In: IAM2008 4th IEEE International Conference on Management of Innovation and Technology; 13–15 May 2008; Bangkok, Thailand. New York, NY, USA: IEEE. pp. 548-553.

[15] Wolthusen SD. Secure visualization of GIS data. In: IWIA 2006 IEEE Information Assurance Workshop; 21–23 June 2006; West Point, NY, USA. New York, NY, USA: IEEE. pp. 200-207.

[16] Yershov A, Zabiniako V, Semenchuk P. Using concatenated steganography for visual analysis in GIS SOA. In: ACSS 2012. pp.74-82.

[17] OpenGIS Implementation Standard for Geographic information - Simple Feature Access Part 2: SQL Option. OGC. Version 1.2.1.

[18] Hemalatha S, Dinesh Acharya U, Renuka A, Kamath PR. A secure and high capacity image steganography technique. arXiv preprint 2013. arXiv:1304.3629.

[19] Rabie T, Kamel I. High-capacity steganography: a global-adaptive-region discrete cosine transform approach. Multimed Tools Appl 2016; 76: 6473-6493.

[20] Brisbane G, Safavi-Naini R, Ogunbona P. High-capacity steganography using a shared colour palette. In: IEE Proceedings on Visual Image Signal Processings; 24 October 2005. pp. 787-792.

[21] Szczypiorski K, Tyl T. MoveSteg: A method of network steganography detection. Int J Electron Telecommun 2016; 62: 335-341.

[22] Watters PA, Martin F, Stripf HS. Visual steganalysis of LSB-encoded natural images. In: Proceedings of 3rd International Conference on Information Technology Applications; 1 August 2005; Sydney, Australia. New York, NY, USA: IEEE. pp. 746-751.