A New Approach to Video Sharing with steganographic effects +

Chang-Chou Lin (林長洲) and Wen-Hsiang Tsai (蔡文祥)

Department of Computer and Information Science,
National Chiao Tung University, Hsinchu, Taiwan, R. O. C.

Tel: (03) 5728368, E-mail: whtsai@cis.ncu.edu.tw

Abstract

A novel approach to secret video sharing with steganographic effects is proposed, in which the time and space redundancy characteristic of a continuous image sequence is utilized. First, frame differencing is carried out to decrease the image data amount coming from the static background where pixel values keep unchanged in successive frames. In each generated difference frame, some pixels are still with relative large gray-level values on the region where moving objects are located. Therefore, an extra quantization process is performed to decrease further the data amount that needs to be shared and embedded. Then, the quantized difference frame is encoded into a number of shares for a group of secret sharing participants. Moreover, a data hiding process is included for creating steganographic effects. Shares are embedded into the pre-selected image sequence before they are delivered to participants. Invaders cannot be aware of the existence of the shares easily. Examples are also included to illustrate the proposed scheme.

Keywords: Secret sharing, frame differencing, steganography, quantization, data hiding

1. Introduction

Secret keeping and protection is an important issue in many information security applications. One way is to share a secret among a group of participants. A well-known technique for this is the (k, n)-threshold method proposed by Shamir [1] and studied by several others [2-4], by which secret data can be encoded into n shares and distributed to n participants, and only when any k or more of the shares are collected can the secret be recovered.

Steganography is a kind of data hiding technique that provides another way of security keeping and protection. Numerous schemes have been developed to achieve the goal of data hiding [5-12]. Differing from the usual secret sharing schemes, which generate noisy data sets as shares that might be suspicious to invaders, the idea of embedding shares into meaningful contents is proposed in this study. This enhances the security protection effect. It requires the use of data hiding techniques after the secret sharing process.

In this paper, a scheme about sharing image sequences and embedding the generated shares into meaningful image sequences is proposed. Image sequences possess the characteristics of large data amount and high inter-frame correlation. Large data amount makes the size of the generated shares large and therefore the subsequent data hiding process hard to accomplish. The high inter-frame correlation can be utilized to decrease the data amount that needs to be shared and embedded,

and acquire a feasible result in quality in the meantime.

The rest of this paper is organized as follows. In section 2, we describe the proposed secret video sharing process. In Section 3, the proposed method of secret video recovery is introduced. Some experimental results are shown in Section 4. Finally, conclusions are given in Section 5.

2. Proposed Process of Share Encryption

In the proposed process of share encryption for n participants, a secret image sequence I is encrypted into n shares, which are then embedded into n cover image sequences selected in advance by the participants, resulting in n stego-image sequences. The entire process consists of four major steps: frame differencing, quantization, secret sharing, and data hiding.

Frame differencing is a common technique used in processing images with motion objects for the purpose of data size reduction. Each computed difference frame is usually with a relative smaller amount of data than the original frames because the gray-level values of the major portion of the difference frame belonging to the static image background mostly are close to zeros. On the other hand, the difference frames can be used to reconstruct the original image frames. These properties help subsequent secret sharing and data hide steps, which otherwise are difficult to carry out because of the enormity of the data amount. However, the gray-level values of the portion belonging to moving objects might still be large. Therefore, a quantization process is performed next to reduce further the data amount that need be shared.

Then the Shamir (k, n)-threshold scheme [1] is employed to encrypt the resulting difference frames into n shares. Finally, each participant's share is embedded into a participant-chosen image sequence by a data hiding scheme. The data hiding process creates an effect of steganography to the shares, which makes an invader unaware of the existence of the shares. It provides another way of security protection other than the secret sharing scheme. As to the process of decryption, its steps are similar to those executed in the encryption process except in a reverse order. In the remainder of this section, we describe first the details of the encryption process, which is divided into two parts, secret sharing and data hiding. The decryption process will be described in the next section. We denote the gray-level value of an image I at coordinates (x, y) by $I(x, y)$.

Algorithm 1: ***The process of secret sharing*****.**

Input: an image sequence with frames I, I_1, I_2, …, I_k.

Output: n sets of shares, which can be used to reconstruct I.

Steps.

Step 1. For $i = 1$ to $k-1$, perform the following steps.

1.1 (Frame differencing) Compute difference frame D_i by subtracting the gray-level value of each pixel in I_i from that of the corresponding one in I_{i+1}, i.e. compute

$$D_i(x, y) = I_{i+1}(x, y) - I_i(x, y). \quad (1)$$

1.2 (Quantization of frame difference values) Perform the following quantization to each pixel of D, resulting in a quantized difference frame, denoted as D':

a. For each pixel in D_i with gray-level values greater than 15, divide them by 16.

b. For each pixel in D_i with gray values smaller

than −15, divide them by −16.

c. For each pixel in D_i with gray-level values greater than −16 and smaller than 0, divide them by −1.

d. Otherwise, do nothing.

1.3 Establish a quantzation_table Q_i to record which condition the corresponding pixel belongs to.

1.4 According to the condition recorded in Q_i, compute

$$I_{i+1}'(x, y) = I_i(x, y) + D_i'(x, y) \times q_i(x, y), \quad (2)$$

where the value of $q_i(x, y)$ is decided by the following conditions:

a. $q_i(x, y)$ is 16 if condition a is recorded in Q_i.

b. $q_i(x, y)$ is -16 if condition b is recorded in Q_i.

c. $q_i(x, y)$ is -1 if condition c is recorded in Q_i.

d. $q_i(x, y)$ is 1 if condition d is recorded in Q_i.

1.5 Replace I_{i+1} by I_{i+1}'.

1.6 Encrypt D_i' with the Shamir (k, n)-threshold scheme into n shares, $S_{i1}, S_{i2}, \ldots, S_{in}$.

Step 2. Collect for each secret sharing participant the corresponding $k-1$ shares, with each share from a D_i'.

In Step 1.1, we do a frame differencing process. Because $D_i(x, y)$ is close to zero if $I_i(x, y)$ is a pixel on the static background, the amount of data for $D_i(x, y)$ usually decreases drastically than that for $I_{i+1}(x, y)$, which can be reconstructed from $I_i(x, y)$ and $D_i(x, y)$. Therefore, storing difference frame $D_i(x, y)$ instead of $I_{i+1}(x, y)$ is with the effect of space saving. Next, we perform a quantization process that reduces the data amount needed in the moving object whose gray values are relative large. Besides, we also want to control the quantized values to be positive, so an extra "divide −1" operation is executed for the negative values in D_i. These are summarized in Step 1.2. For the future reconstruction of D_i, we establish a quantization_table to record which operation we performed in Step 1.2. In Step 1.5 and 1.6, we reconstruct the value of I_{i+1}. First, we reconstruct the value of approximate D_i by $D_i'(x, y) \times q_i(x, y)$. The value of $q_i(x, y)$ which is decided by the content of $Q_i(x, y)$ is the same as that $D_i(x, y)$ divided in Step 1.2. Some distortion exists in this step, but we will show the distortion is acceptable in the experimental section. Then, the approximate next frame, $I_{i+1}'(x, y)$, can be computed by Eq. (2). The computed $I_{i+1}'(x, y)$, instead of $I_{i+1}(x, y)$, is further used to calculate the value of D_{i+1} because the participants can reconstruct $I_{i+1}'(x, y)$ only by the shares of D_i' in the decryption phase. Wrongly choosing $I_{i+1}(x, y)$ to calculate D_{i+1} makes

$$\begin{aligned} I_{i+2}(x, y) &= I_{i+1}'(x, y) + D_{i+1}(x, y) \\ &\fallingdotseq I_{i+1}'(x, y) + (I_{i+2}(x, y) - I_i(x, y)) \\ &= I_{i+2}(x, y) + (I_{i+1}'(x, y) - I_i(x, y)). \end{aligned}$$

An extra distortion $(I_{i+1}'(x, y) - I_i(x, y))$ is generated which will degrade the quality of reconstructed image sequence.

In Step 1.7, we introduce Shamir (k, n)-threshold scheme to share D_i', the key point to reconstruct the original image sequence. The details of this sharing process are described as follows. Based on a pre-selected secret integer value y and a threshold k, and by using the following $(k-1)$-degree polynomial

$$F(x) = y + m_1 \times x + m_2 \times x^2 + \cdots + m_{k-1} \times x^{k-1} \bmod p, \quad (3)$$

the generation of the n shares proceeds in the following way.

1. Choose y to be the value of $D_i'(x, y)$ that is to be shared.
2. Select the number k is to be no larger than n.
3. Choose p to be the nearest prime number

larger than $D_i'(x, y)$.

4. Choose $k - 1$ integer values $m_1, m_2, \ldots, m_{k-1}$ randomly in the range $[0, p]$.

5. For each participant a, compute a corresponding value of $F(a)$ by Equation (3).

6. Take $F(a)$ as a share S_{ai}.

Here we use modular arithmetic instead of real arithmetic as Shamir did. The set where all integers modulo a prime number p form a Galois field. In this field, we can reconstruct the polynomial $F(x)$ using an interpolation method in the recovery phase which will be described in the next section. From Step 1.2, we know that the value of $D_i'(x, y)$ is restricted within the range [0, 15], so we choose p as 17, the nearest prime number larger than $D_i'(x, y)$.

So far, we have accomplished a mechanism that not only shares a secret image sequence but also generates a small amount of share data for each participant. In our proposed scheme, the difference frames, which can be used to reconstruct original frames, record the portion of the static background with small storage space. Hence the information we must process decreases preliminarily. Moreover, an extra quantization process is applied to manipulate the remaining pixels with gray values out of the range [0, 15] and further decrease the amount of information that need be securely dealt with.

It is mentioned that the shares for each participant is meaningless in the form of image sequence, which may be suspicious to invaders. Therefore, a data hiding process is introduced after secret sharing is performed. The detailed algorithm lists below.

Algorithm 2: ***The process of data hiding*****.**

Input: n sets of shares, $S_{11}, S_{12}, \ldots, S_{1(k-1)}, S_{21}, S_{22}, \ldots, S_{2(k-1)}, \ldots, S_{n(k-1)}$.

Output: n pre-selected image sequences, $C_1, C_2, \ldots, C_n$ with length longer than $k-1$ where corresponding set of shares are embedded in.

Steps.

Step 1. For participant $a = 1$ to n and frame $i = 1$ to $k-1$, perform the following steps.

1.1 For $S_{ai}(x, y)$, acquire the value of $C_{ai}(x, y)$ and transform it into 17*ary*, which must be in the form $(pq)_{17}$. Compute

$$d = |S_{ai}(x, y) - q|. \quad (4)$$

a. If $q \le 8$ and $d > 8$, use $((p-1)S_{ai}(x, y))_{17}$ to substitute $(pq)_{17}$.

b. If $q \le 8$ and $d \le 8$, use $(pS_{ai}(x, y))_{17}$ to substitute $(pq)_{17}$.

c. If $q > 8$ and $d > 8$, use $((p+1)S_{ai}(x, y))_{17}$ to substitute $(pq)_{17}$.

d. If $q > 8$ and $d \le 8$, use $(pS_{ai}(x, y))_{17}$ to substitute $(pq)_{17}$.

Step 2. Deliver the n stego-image sequences where shares are embedded to n participants.

In Step 1, we try to embed shares generated in Algorithm 1 into the corresponding image sequence selected by the participant. From Eq. (3) with p is 17, it is guaranteed that the range of the value of $S_{ai}(x, y)$ falls in [0, 16]. Therefore, we first transform the gray value of the pixel in the cover image to in the form of $(pq)_{17}$. Then we embed the share into it by replacing the value of q with $S_{ai}(x, y)$, which does not exceed the least significant digit. Furthermore, we optimize the degree of distortion that arises from data embedding according to the evaluation of Eq. (4). The details list in Step 1.1. The difference between the embedding result and the original pixel value will not exceed 8. The reason is explained as follows.

1. For condition a: $q \le 8$ and $d > 8$ imply that $S_{ai}(x, y) > q$.

$(pq)_{17} - ((p-1)S_{ai}(x, y))_{17} = (((p-1)q)_{17} + 17 - ((p-1)S_{ai}(x, y))_{17}) = 17 + q - S_{ai}(x, y) < 17 + (-8) = 9$.

2. For condition b: $d \le 8$, the result is trivial.

3. For condition c: $q > 8$ and $d > 8$ imply that $S_{ai}(x, y) < q$.

$((p+1)S_{ai}(x, y))_{17} - (pq)_{17} = (pS_{ai}(x, y))_{17} + 17 - (pq)_{17} = 17 + S_{ai}(x, y) - q < 17 + (-8) = 9.4$.

For condition d:

$d \le 8$, the result is trivial.

Afterwards, we deliver the n stego-image sequences to accomplish the whole secret hiding process.

In this phase, we achieve the goal of steganography. It provides another way of security protection. Unlike utilizing a particular cipher algorithm to protect secret data from illicit access, the purpose of cover communication is to embed secret data in pre-selected meaningful images, called cover images, without creating visually perceptible changes to keep an invader unaware of the existence of the secret. Moreover, we manipulate our method to restrict the difference of the pixel value between the original and stego image under 8, while 31 is guaranteed by pure least significant bits replacement method.

3. Proposed Process of Decryption

In this section, we first describe the process of decryption, which is divided into two algorithms, and then explain the details.

Algorithm 3: ***The process of data extraction***.

Input: the n image sequences, $C_1, C_2, \ldots, C_n$ held by the n secret sharing participants.

Output: n sets of shares, $S_{11}, S_{12}, \ldots, S_{1(k-1)}, S_{21}, S_{22}, \ldots, S_{2(k-1)}, \ldots, S_{n(k-1)}$.

Steps.

Step 1. For participant $a = 1$ to n and frame $i = 1$ to $k-1$, perform the following steps.

1.1 Acquire the value of $C_{ai}(x, y)$ and transform it into 17*ary*, which must be in the form $(pq)_{17}$. Extract q as $S_{ai}(x, y)$.

In the above algorithm, we just extract the pixel value of each stego image and then transform it into the form of 17*ary*. The value in the least significant digit is what we want. After getting all the shares, a secret recovery process must be performed to restore the original secret image sequence.

Algorithm 4: ***The process of secret recovery***.

Input: n sets of shares, $S_{11}, S_{12}, \ldots, S_{1(k-1)}, S_{21}, S_{22}, \ldots, S_{2(k-1)}, \ldots, S_{n(k-1)}$, held by the n secret sharing participants, respectively; the quantization_table $Q_1, Q_2, \ldots, Q_{(k-1)}$; the first frame of I, I_1.

Output: the secret image sequence I.

Steps.

Step 1. For frame $i = 1$ to $k-1$, perform the following steps.

1.1 Reconstruct the value of $D_i'(x, y)$ using the interpolation method mentioned in [1] from the n corresponding shares, $S_{i1}, S_{i2}, \ldots, S_{in}$, held by the n participants.

1.2 According to the condition recorded in Q_i, compute

$I_{i+1}'(x, y) = I_i(x, y) + D_i'(x, y) \times q_i(x, y)$ as Eq. (2),

where the value of $q_i(x, y)$ is decided by the following conditions:

a. $q_i(x, y)$ is 16 if condition a is recorded in Q_i.

b. $q_i(x, y)$ is -16 if condition b is recorded in Q_i.

c. $q_i(x, y)$ is -1 if condition c is recorded in Q_i.

d. $q_i(x, y)$ is 1 if condition d is recorded in Q_i.

1.3 Replace I_{i+1} by I_{i+1}'.

Step 2. Combine in order all frames obtained in the last step to reconstruct the original secret image sequence.

After extraction of shares embedded in cover image sequences, we try to recover the frames of the secret image sequence one by one in Step 1. In Step 1.1, we recover the value of D_i' of each frame. The details are described as follows.

1. Collect at least k secret shares, which are assumed to be the first k without loss of generality, from the n ones to form a system of equations as follows:

$$\begin{aligned} F(1) &= y + m_1 \times 1 + m_2 \times 1^2 + \cdots + m_{k-1} \times 1^{k-1} \bmod 17, \\ F(2) &= y + m_1 \times 2 + m_2 \times 2^2 + \cdots + m_{k-1} \times 2^{k-1} \bmod 17, \\ &\vdots \\ F(k) &= y + m_1 \times k + m_2 \times k^2 + \cdots + m_{k-1} \times k^{k-1} \bmod 17. \end{aligned} \tag{5}$$

2. Use the Lagrange method to solve the k unknowns, m_1, m_2, …, m_{k-1}, and y, in the above k equations, and reconstruct the $(k-1)$-degree polynomial $F(x)$ described by Eq. (3). Note that the $F(i)$ in (5) with $1 \le i \le k$ are k known values collected from the k secret shares.

3. Construct $F(x)$ by the following formula:

$$\begin{aligned} F(x) = {} & F(x_1)\frac{(x-x_2)(x-x_3)\cdots(x-x_k)}{(x_1-x_2)(x_1-x_3)\cdots(x_1-x_k)} + F(x_2)\frac{(x-x_1)(x-x_3)\cdots(x-x_k)}{(x_2-x_1)(x_2-x_3)\cdots(x_2-x_k)} + \cdots \\ & + F(x_2)\frac{(x-x_1)(x-x_3)\cdots(x-x_k)}{(x_2-x_1)(x_2-x_3)\cdots(x_2-x_k)} \bmod 17 \end{aligned} \tag{6}$$

And take the secret value $C_1 = y$ to be $F(0)$.

Note that according to Shamir [1], if fewer than k secret shares are collected, the k unknowns cannot be solved and the desired y value cannot be reconstructed. After we get the value of D_i', we obtain correct $q_i(x, y)$ from Q_i and compute the next frame through the current frame as Eq. (2). Repeat these operations to get all the frames of the secret image sequence. Finally, we combine in order these frames to reconstruct the original image sequence.

4. Experimental results

In this section, some experimental results are shown to prove the feasibility of the proposed scheme. For ease of demonstration, we use gray image sequence to evaluate our scheme. But it is intuitively easy to extent our scheme to full color image sequence by just applying the operations we introduce in the last two sections on all RGB channels. We show as an example the effect of our scheme for the (2, 3)-threshold case here by some experimental results.

We first take an image sequence as the secret image sequence. Parts of it are shown in Fig.1 (a) through (d). After we performed the Algorithm 1 and 2, we got three sets of shares embedded in three pre-selected image sequences. Parts of one of the cover image sequence and the corresponding stego image sequence are shown in Fig.1 (e) through (h) and Fig.1 (i) through (l), respectively. The average PSNR of the stego image sequences is 34.24dB. In the phase of recovery, we extracted two of the three sets of shares as the Algorithm 3 described and recovered the secret as the Algorithm 4 did. Parts of the recovered image sequence corresponding to Fig.1 (a) through (d) are shown in Fig.1 (m) through (p). The average PSNR of the recovered image sequences is 32.96dB. Experimental results verify that the proposed scheme is feasible and the quality of the generated stego and recovered image sequences are visually acceptable.

5.Conclusions

A new scheme for secret image sequence sharing has been proposed, which can limit the size of the generated shares and possess steganography effects. Some characteristics of continuous image sequences are employed first to decrease the data that need to be processed. Then, the (k, n)-threshold function is adopted for a group of n participants to share the secret. Only k or more out of the n participants attend can the original image sequence be recovered. Finally, the proposed scheme is equipped with the capability of steganography. Participants can choose an image sequence, where the size of each frame is the same as that of the original image sequence, to embed the generated shares in it before delivering the shares. Furthermore, the quality of the recovered image sequence is feasible by experiments. This system is thus suitable applications where high security and efficiency is required.

REFERENCES

[1] A. Shamir, “How to share a secret,” *Communications of the Association for Computing Machinery*, vol. 22, no. 11, pp. 612-613, 1979.

[2] D. R. Stinson, “An explication of secret sharing schemes,” *Designs, Codes, and Cryptography*, vol. 2, pp. 357-390, 1992.

[3] C. C. Chang and H. C. Lee, “A new generalized group-oriented cryptoscheme without trusted centers,” *IEEE Journal on Selected Areas in Communications*, vol. 11, no. 5, pp. 725-729, 1993.

[4] C. C. Lin and W. H. Tsai, “Visual cryptography for gray-level images by dithering techniques,” *Pattern Recognition Letters*, vol. 24, no. 1-3, pp. 349-358, 2003.

[5] D. C. Wu and W. H. Tsai, “Data hiding in images via multiple-based number conversion and lossy compression,” *IEEE Transactions on Consumer Electronics*, vol. 44, no. 4, pp.1406-1412, 1998.

[6] C. T. Hsu and J. L. Wu, “Hidden digital watermarks in images,” *IEEE Trans. Image Process*, vol. 8, pp. 58-68, 1999.

[7] W. Bender, N. Morimoto, D. Gruhl, "Method and apparatus for data hiding in images," *U. S. Patent*, No. 5689587, 1997.

[8] S. Walton, "Image authentication for a slippery new age," *Dr. Dobb's Journal: Software Tools For The Professional Programmer*, vol. 20, pp. 18-26, 1995.

[9] L. M. Marvel and C. T. Retter, “Hiding information in images,” *Proceedings of International Conference on Image Processing*, vol. 2, pp. 396-398, 1998.

[10] T. Jamil, “Steganography: the art of hiding information in plain sight,” *IEEE Potentials*, vol. 18, no. 1, 1999.

[11] L. M. Marvel and C. T. Retter, “A methodology for data hiding using images,” *IEEE Military Communication Conference*, vol. 3, pp. 1044-1047.

[12] E. T. Lin and E. J. Delp, “A review of data hiding in digital images,” *The Image Proceeding, Image Capture Systems Conference*, pp. 274-278, 1999.

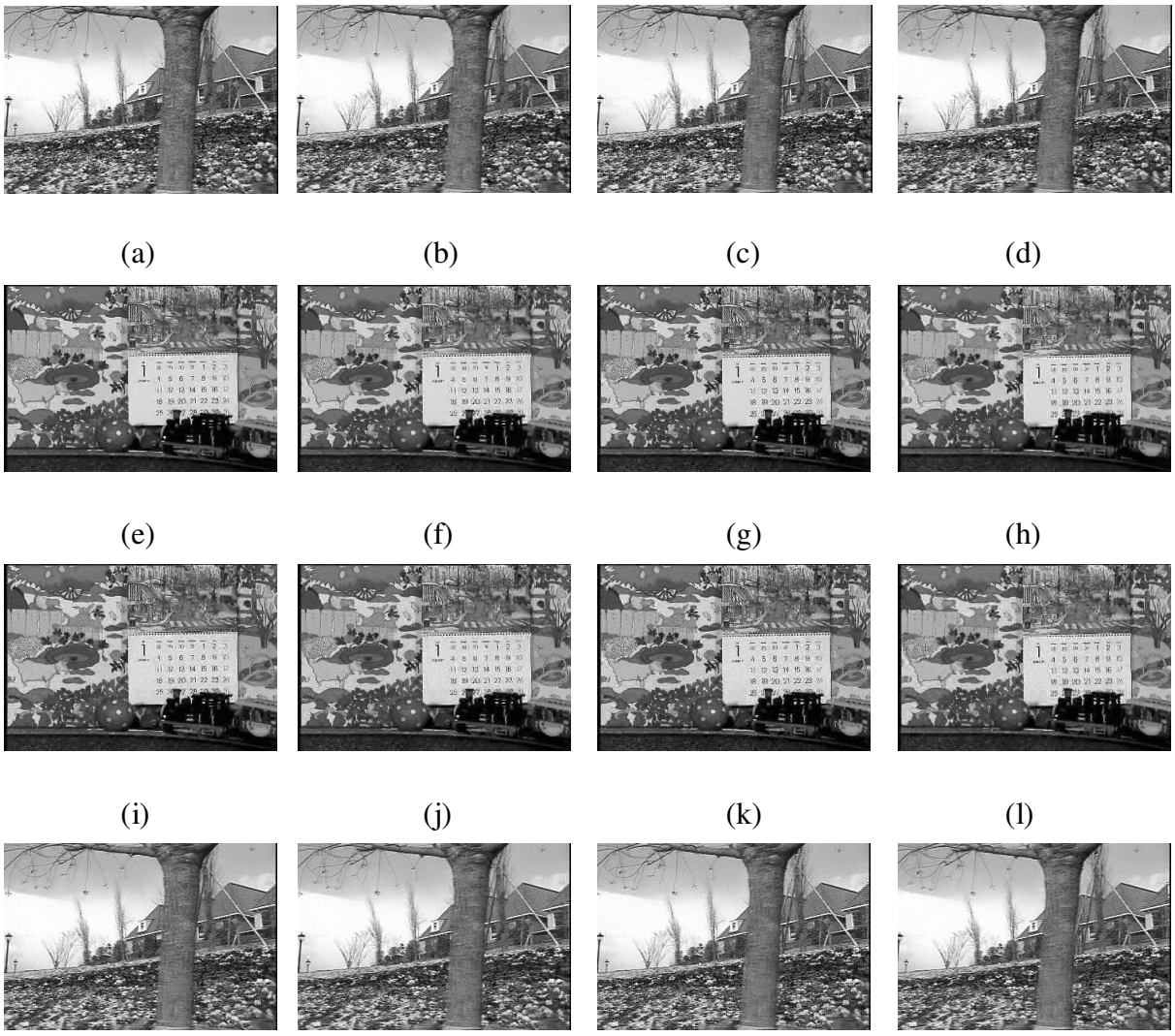

Fig. 1. (a) through (d) Parts of the secret image sequence. (e) through (h) Parts of the cover image sequence. (i) through (l) Parts of the stego image corresponding to (e) through (h). (m) through (p) Parts of the recovered image sequence corresponding to (a) through (d).