CONSTRUCTION AND APPLICATIONS OF MESSAGE-RICH VIDEOS FOR PERVASIVE COMMUNICATION

[1]*Chi-Ju Wu* (吳季儒), *and* [2]*Wen-Hsiang Tsai* (蔡文祥)

[1]Institute of Multimedia Engineering
[2, 3] Department of Computer Science
National Chiao Tung University, Hsinchu, Taiwan
Emails: ing5698yi@hotmail.com, whtsai@cis.nctu.edu.tw

ABSTRACT

Many types of multimedia can be used to conduct pervasive communication, like advertisements, posters, videos on TVs, etc. Messages may be embedded into such multimedia by data hiding techniques, resulting in the so-called message-rich multimedia. The hidden message may be extracted by the use of smart phones, tablets, etc. In this study a new technique is proposed for creating a new type of message-rich multimedia, called message-rich video, via which people can exchange information without transmitting directly digital files such as videos and images. Message can be embedded in such videos and extracted by a display-and-imaging fashion using smart phones. For message embedding, the use of diagonal circular signals in the frequency domain is proposed, by which a 2-bit signal can be embedded in each video frame. For message extraction, a signal detection method based on the technique of moment-preserving thresholding using certain specially-designed masks is proposed. A message synchronization mechanism is also proposed to guarantee correct message extraction is also proposed. The feasibility of the proposed method is demonstrated by theoretical analyses and good experimental results.

Keywords: *message-rich video; data hiding; pervasive communication; display-and-imaging*

1. INTRODUCTION

With the development of mobile devices, cameras in such devices are more and more advanced; thereby people can record videos and images everywhere and anytime. Furthermore, people can exchange information with the identities existing in the environment conveniently [1]-[2]. For example, one can use the camera on a smart phone to scan a QR code on a merchandise item to obtain the detailed related information. Accordingly, Davis [3] proposed recently a new concept, called *signal-rich art*: the art that communicates its identity to context-aware devices through data hiding techniques, mainly, to realize *pervasive communication*. It is also mentioned in [3] that data hiding is the main technique used to implement signal-rich art applications in the past.

One technique regarded in this study as quite fit for the aim of signal-rich art is the data hiding technique which can survive *print-and-scan* operations [4]-[6]. In the usage of such a type of technique, an image with a message hidden in it is printed to be a hardcopy. Then, message extraction is conducted by scanning the hardcopy into a digital version and analyzing it to reveal the hidden message.

Inspired by the above ideas, it is desired in this study to design a system that allows people to exchange information directly by using the mobile device to capture videos displayed on TVs or computer screens. Specifically, we want to design a technique for creating a new type of video, by which, differently from traditional information transmission, people can exchange information mutually without transmitting *directly* digital files such as videos and images. Such a type of video, called *message-rich video* in this study, is a type of message-rich multimedia mentioned in [11]. Moreover, the message can be extracted by a *display-and-imaging* fashion, i.e., the video with the hidden message is displayed on a screen and a sequence of images of it is acquired with a smart phone, from which the hidden message is extracted finally. In this way, people can exchange messages robustly and possibly secretly without file transmission. Applications of such a new technique include: message transmission, covert communication, information hiding, QR code-like usages, and so on.

About related works for the purpose of pervasive communication, Lee and Tsai [7]-[9] have proposed several types of message-rich multimedia, including: 1) large-sized image; 2) encrypted image; 3) hardcopy of image. Specifically, Lee and Tsai in [7] proposed block-fitting and color-transformation techniques for hiding large-volume data into single images; in [8], proposed a data hiding method via message encryptions and spatial-correlation comparison; and in [9], proposed a new data hiding technique via so-called *message-rich code images* for automatic identification and data

[2] Wen-Hsiang Tsai is also with the Department of Information Communication at Asia University, Taiwan.

capture (AIDC) applications, in which a block luminance modulation technique for data hiding in the Y channel of the YUV color space and a spatial correction technique for identifying hidden messages were utilized to achieve message hiding and extraction.

In studies on message-rich multimedia, the main goal is to propose appropriate data hiding techniques to create various type of message-rich multimedia for pervasive communication. Fulfillment of this goal will be expected to enhance the state-of-art studies on data hiding techniques, yielding new visions of pervasive communication and further steps of extending its applications. In this study, we will concentrate on designing a proper method to transmit messages via videos by the *display-and-imaging* operation, hoping to achieve pervasive communication more flexibly.

The remainder of this paper is organized as follows. In Sec. 2, we give an overview of the proposed system as well as the proposed techniques for data hiding and creation of message-rich multimedia. In Sec. 3, the proposed method for message embedding against display-and-imaging attacks is described. The proposed method for embedding messages in videos with consideration of synchronization is described in Sec. 4. In Sec. 5, the proposed method for extracting the message and a technique for correction of erroneous messages are described. Methods for refining the message extraction result to improve the extraction accuracy and for improving the processing speed are also presented. In Sec. 6, experimental results together with some discussions about related issues are included. Finally, conclusions of the study and some suggestions for future works are given in Sec. 7.

2. SYSTEM DESIGN AND PROCESSES

The hardware configuration of the proposed system is shown in Fig. 1, which includes a monitor, a smart phone, and a mobile-device holder. Because the video which is captured for use by the proposed method might be long, we use the holder to hold the mobile device stably while doing the display-and-imaging work. Also, to adapt the proposed system to different environments, the holder can be used to adjust the height, position, and orientation of the mobile device easily for the purpose of covering the scope of the displayed video on the monitor.

Fig. 1. Hardware configuration of the proposed system.

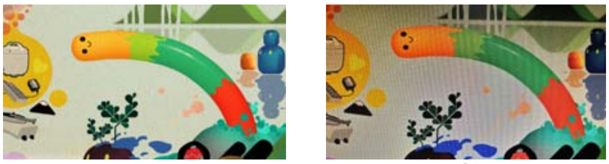
The operations of the proposed system include mainly a message-embedding process and a message-extraction process. The message-embedding process can take as input most types of videos which have different resolutions, frame rates, formats, etc. It can generate message-rich videos which can survive the display-and-imaging operation. At the beginning, the system gets the message for embedding as input and retrieves the frame rate of the source video. To achieve signal synchronization in message embedding and extraction, specially-designed signals are added to the original message as *initial and ending signals*, and the resulting message is transformed into a bit stream of ASCII codes according to the video frame rate. Similar to methods [4-6] which can resist the print-and-scan attack, the proposed message-embedding process replaces the low-frequency part of the frequency spectrum of each captured image frame with some predefined patterns which are generated according to the bit stream of the message's ASCII codes in order to resist the display-and-image operation.

Next, people can record the video displayed on the monitor using the camera of the mobile device fixed on the holder mentioned previously. Then, the recorded video is processed by the proposed message-extraction process to retrieve the message. In this stage, the system gets the video frame rate at first for the purpose of conducting signal synchronization. Next, the system extracts and then skips the initial signal of the video by analyzing the spectrum of each video frame's Y components. Then, the system starts to analyze the remaining frames until an ending signal is detected. In this process, *voting* and *noise-removal* techniques are used to retrieve the embedded ASCII codes. More details are described in the subsequent sections.

3. SIGNAL EMBEDDING AGAINST DISPLAY-AND-IMAGING OPERATIONS

3.1 Idea of Proposed Message Embedding Technique

In order to embed a message in a given video, the embedded message must "survive" the "attack" of the display-and-imaging operation. As shown in Fig. 2, unexpected moiré patterns can be found everywhere in the image frames resulting from display-and-imaging operations. Such patterns destroy the original data of a lot of pixels in the attacked image frame. This means that if we embed message data in the *spatial* domain of the image frame, the embedded data may be destroyed. It means also that if we want to apply the proposed system on different mobile devices in different environments, we should embed message signals which are robust enough to tolerate the differences between the original video and the recorded video.



(a) (b)

Fig. 2. Illustration of the effect of display-and-image operations. (a) An original cartoon image frame. (b) An image frame acquired of a desktop computer's screen, on which the display-and-imaging operations have been applied, includes unexpected moiré patterns.

Before discussing the proposed method, we briefly list some interesting findings by Solanki et al. [5] about the properties of the discrete Fourier transformation (DFT) of images here: 1) *frequency robustness*: high-frequency coefficients are more fragile than low-frequency and mid-frequency ones; 2) *effect on DFT magnitude spectrum*: high-magnitude DFT coefficients are preserved better than low-magnitude ones; 3) *effect on DFT phase spectrum*: for high-magnitude coefficients, the difference in phase of adjacent frequency locations is preserved during the print-and-scan operation.

Based on the above properties of the DFT coefficients, in this study we want to embed messages into image frames by inserting specially-designed patterns in the frequency spectra of the frames. Specifically, we want to make significant differences in the modified frequency spectra during message embedding, such as erasing high-magnitude coefficients in the spectra to form "holes," or adding high-magnitude ones to form "peaks," as described next.

3.2 Signal Embedding by Modifying Low-frequency Coefficients in Image Luminance Channel

In our experiments of message embedding in frequency spectra, at first we try to embed signals by adding high-magnitude values into the mid-frequency bands of the Y components of each input image frame as shown in Fig. 3. As can be seen from the images in the figure, after the display-and-imaging operation is applied, the embedded signals are reduced too much so that they cannot be detected easily.

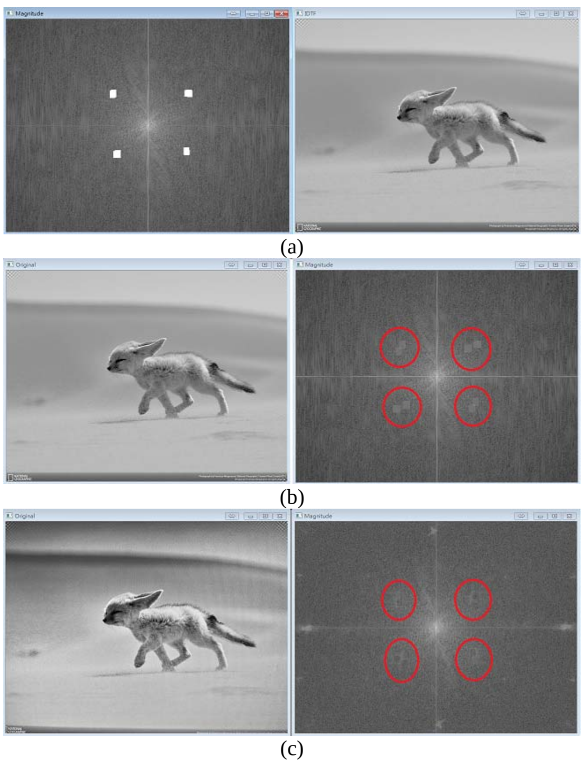


Fig. 3. Experimental results of adding high-magnitude values. (a) Spectrum of original image's Y component with high-magnitude values added into mid-frequency bands (left); and signal-embedded image after IDFT (right). (b) Signal-embedded image (left) and its spectrum (right). (c) Signal-embedded image on which display-and-imaging has been applied (left) and its spectrum (right).

From the above-mentioned experimental results, we see that the signals in mid-frequency bands in the frequency spectra cannot be preserved well to resist the display-and-imaging operation. So, we conducted another experiment. Firstly, it was noted that the low-frequency coefficients usually have high-magnitude values. So low-value signals were used to replace the low-frequency coefficients in the experiment. Such changes of low-frequency coefficient values destroys the values of a wide range of pixels in the resulting image as can be seen from Fig. 4, but the embedded low-value signals are stronger and so preserved after the attack as indicated by Fig. 4(b). Due to this phenomenon, the proposed method employs the signal-embedding technique of "erasing" the low-frequency coefficients with "erasing" meaning changing the value to be *zero*.

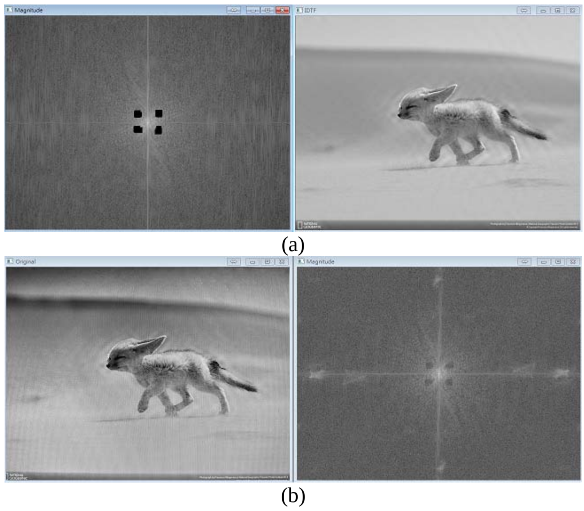


Fig. 4. Experimental results of embedding low-magnitude values to low-frequency coefficients. (a) Spectrum of original image's Y component with low-magnitude values added (left); and signal-embedded image after IDFT (right). (b) Signal-embedded image on which display-and-imaging has been applied (left) and its spectrum (right).

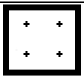
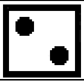
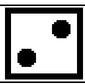
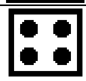
In conclusion, to overcome the "attacks" of display-and-imaging and compression operations, a technique of modifying the low-frequency coefficients of the input image's Y component by use of *circular holes* is proposed. Accordingly, we can determine whether an image frame is signal-embedded or not by the analysis of the low-band coefficients as described in the following.

4. MESSAGE EMBEDDING IN VIDEOS

4.1 Design of Patterns for Embedding Signals

According to the above-described idea of the proposed signal embedding technique, we can embed a 1-bit signal into a single frame by the use of a *four-corner signal consisting of four circles* as illustrated in Fig. 4. In order to improve the quantity of embedded signals in each frame, we separate the four-corner signal into two pairs of diagonal points and encode them in a way as shown in Table 1. Specifically, because the message can be transformed into a bit stream, the proposed patterns are designed to represent four different binary codes 00, 01, 10, and 11. Accordingly, an input message can be translated into a bit stream of binary codes which then are represented by corresponding patterns and embedded into video frames.

Table 1. Proposed design of patterns and their meanings.

	0(00)		1(01)
	2(10)		3(11)

The robustness of the patterns is an important factor for successful message extraction; without robustness against display-and-imaging operations, the embedded message cannot be extracted correctly. In Fig. 5, we show two examples of signal-embedded images after being attacked by such operations. As can be seen, the circles in the resulting spectrum images are still observable and so extractable as proved in our experiments in this study.

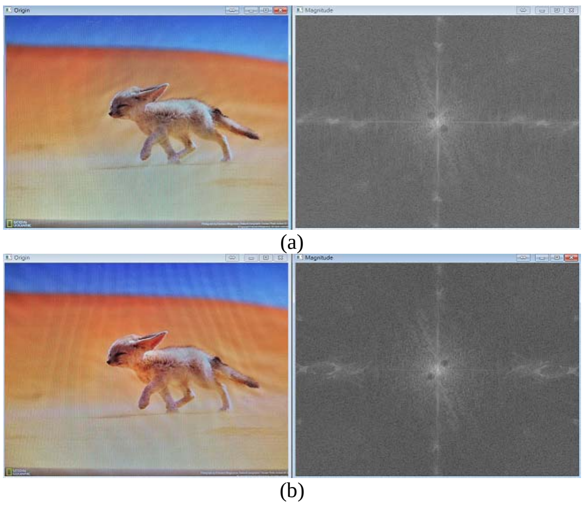


Fig. 5. The experimental results for checking the robustness of pattern signals “01” and “10.” (a) An image with signal 01 being embedded and being attacked by display-and-imaging and its spectrum. (b) Another example with signal 10 being embedded.

4.2 Synchronization in Message Embedding by Use of Special Signals

As a connection between signal embedding and extraction, it is necessary to let the message extraction process know where the embedded message begins in a recorded video. For this, we create an obvious starting point by the use of special-designed signals. Similarly, a signal specifying the ending of the recorded video is needed. An obvious ending signal is so designed as well in this study for terminating the message extraction process. The initial signal is designed to be “0033” in this study, by which we expect the double “3” to be detected successfully by the message extraction process. Similarly, the ending signal is designed to be “1333” in this study; accordingly, when the message extraction process detects a triple of “3,” it will terminate the message extraction process.

4.3 Synchronization in Message Embedding by Property of Video Frame Rate

The videos recorded by different mobile devices with different parameter settings may have different frame rates. This causes a synchronization problem for the proposed message extraction process. For example, if we embed a bit stream “0123” into a video whose frame rate is 30 fps and play and record it by the use of a 60-fps video camera, then the bit stream “0123” will be recorded as “00112233” as illustrated in Fig. 6, which is double in length. This means that if we do not correct the difference between the frame rates, the proposed message extraction process will not work correctly.

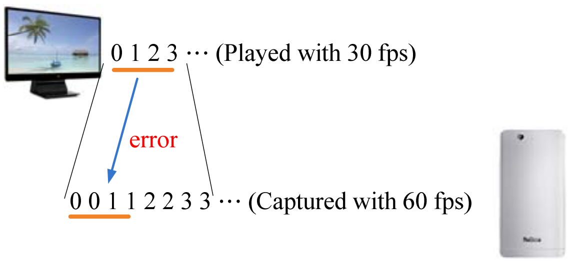


Fig. 6. Synchronization problem caused by different frame rates.

The basic idea proposed in this study to solve this problem is as follows. Firstly, it is noted that though the frame rate is different from one another, the content of the message-rich video displayed for inspection per second is the same. Also, because the video frame rate is fixed every second, we can embed *repeated* signals in a defined period of time, called *embedding period*, for synchronization. More specifically, no matter what the frame rate is, if we set the embedding period to be *S* second(s) and if the patterns of the message to be embedded are P_1, P_2, P_3, …, then we can embed P_1 in all frames of the first *S* second(s) and embed P_2 in all frames of the second *S* second(s), and so on; and the patterns so embedded in the recorded video can be extracted correctly by analyzing all frames every *S* second(s), as can be figured out. For example, as shown in Fig. 7, the signals “0123” are embedded into a sequence of frames of 4 seconds in length by embedding each signal in an embedding period of one second.

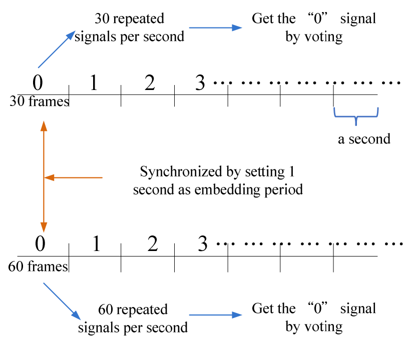


Fig. 7. Proposed Technique for synchronization using the property of video frame rate.

To extract the message accordingly, it is noted that the frame rate of a video can be obtained automatically by software. Thus, the proposed message extraction process can extract the signals by analyzing the content of each frame every second using the acquired frame rate. For example, for the case shown in Fig. 7, while the message embedding process embeds a pattern into the *original video* in every 30 frames, the message extraction process extracts a pattern by analyzing every 60 frames of the *recorded video* for synchronization.

In short, the embedding process and the extraction process can be synchronized by setting an appropriate *embedding period*, such as 0.5 second or 1 second, etc., without transmitting any extra information.

4.4 Proposed Algorithm for Message Embedding

Based on the discussions mentioned previously, an algorithm for embedding the message is described in this section. Recall that we separate the four-corner signals into two diagonal pairs to improve the quantity of message embedding. The robustness of the embedded signals is the most important here as mentioned previously, so we propose in this study the use of *circles of 9-pixel radii* as signals to replace low-frequency coefficients. Also, signal synchronization is solved in a way as described in the last section, by which the proposed system can deal with uses of different types of mobile devices and different formats of videos correctly for various application cases.

Algorithm 1. The message embedding process.

Input: a message M and a source video V into which M is to be embedded.

Output: a message-rich video M' containing the hidden message M.

Steps.

Step 1. Translate the characters of M into a bit stream B of ASCII codes.

Step 2. Insert the patterns of the initial signal “0033” into the beginning of B and those of the ending signal “1333” into the end of B, resulting in a new bit stream B'.

Step 3. Calculate the number D of signals which should be repeated in a pre-selected embedding period T according to the video frame rate R of the source video V obtained from the OpenCV by the formula: $D = R \times T$.

Step 4. Duplicate sequentially each bit of B' D times to get a new bit stream B''.

Step 5. Take an unprocessed frame F of V and process it by the following steps.

5.1 Convert F into a frame F' in the YUV color space and transform F' into a frequency spectrum F''.

5.2 Extract the beginning two bits of B'' and convert the binary number composed of them into a corresponding pattern P.

5.3 Embed P into F'' by erasing the corresponding low-frequency coefficients in F''.

5.4 Repeat Step 5 if V is not processed to the end; otherwise, perform the next step.

Step 6. Take the final V composed of frames with the modified frequency spectra, and transform it by the MPEG-4 codec or the H.264 codec into an AVI file as the desired output message-rich video M'.

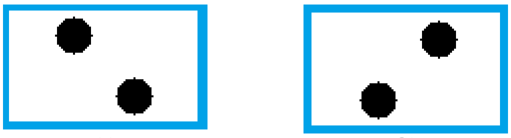
5. MESSAGE EXTRACTION WITH SIGNAL CORRECTION

5.1 Signal Detection by Moment-preserving Thresholding

In the beginning of message extraction, signals embedded in all video frames must be detected correctly. For this, the proposed technique checks the existence of the patterns in the frequency spectrum at first. Basically, if an *erased circle* (i.e., a “black” hole) exists in a spectrum, the frequency coefficients of the circle will be *lower* relatively in value than the other low-frequency coefficients. For this concept to be applicable for detecting embedded signals in the frequency spectrum, a threshold of coefficient values is needed. But a *fixed* threshold value is inapplicable here because the frequency spectrum of each YUV frame is different. A solution proposed in this study is to use the moment-preserving thresholding technique proposed by Tsai [10] which can determine *automatically* an *adaptive* threshold for each frame.

In more detail, with the adaptive thresholds, we can separate the frame spectrum into two parts: a *white* part with higher-frequency values and a *black* part with lower-frequency ones. Then, if the patterns exist, the circular regions with lower values will become black holes in the black part. Thus, we can determine which pattern is embedded in the current video frame by analyzing the structure of the black holes. In this way, the message signal detection problem is solved.

For block-hole analysis, a mechanism for deciding which pattern is embedded in the spectrum should be designed. For this, two masks are used in this study to detect circular holes at first. One mask, denoted as M_1, contains the left-top and right-bottom black circles as shown in Fig. 8(a); and the other mask, denoted as M_2, includes the right-top and left-bottom black circles as shown on Fig. 8(b). We use M_1 and M_2 to conduct an “OR” operation on the binary image I_b resulting from previously-mentioned spectrum thresholding. Then, we can get the number of the black pixels in the circular regions of the binary image I_b. Because of the property of the “OR” operation, the output image I_b' yielded by the “OR” operation will be white except the circular regions. In the circular regions, if the pixels of I_b are white, the corresponding pixels of I_b' are white, too. As a result, we can count the remaining black pixels easily. By the use of M_1 in this way, the number N_1 of black pixels in the left-top and right-bottom circular regions is known, so we can get a score S_1 by dividing N_1 by the area A of the circular regions in M_1, i.e., get $S_1 = N_1/A$. Similarly, we can get another score S_2 in a similar way. Finally, we determine which pattern is embedded by analyzing the values of S_1 and S_2 in the following way: 1) decide the embedded pattern to be "1" if S_1 is high and S_2 is low; 2) to be "2" if S_1 is low and S_2 is high; 3) to be "0" if both S_1 and S_2 are low; and 4) to be "3" if both S_1 and S_2 are high.



(a) (b)

Fig. 8. The masks for detecting patterns. (a) Mask for detecting left-top and right-bottom circular holes. (b) Mask for detecting right-top and left-bottom circular holes.

5.2 Proposed Technique for Message Correction by Noise Removal

In the beginning of the message-extraction process, the initial signal must be found. Because the initial signal is embedded in multiple frames, we *cannot* identify the initial signal just by a single frame. Thus, the message-extraction process must use an appropriate data structure to store the detected signals for later analysis. On the other hand, the number of video frames counted in each period of a second, according to our experience, varies slightly usually. That is, the so-called *frame rate* of the recorded video is just an *average* value. If the average frame rate is used for message extraction, the synchronization mechanism using the initial signal will become invalid because errors will accumulate gradually according to our experimental experience. For example, assume that the average frame rate is 58. It might be possible that the frame rate of the first half of the recorded video is 56 and that of the second half is 60. In this case, if the embedding period is taken to be 1 second, then using the frame rate of 58, the extraction process will analyze two extra signals in the first round of processing; and after that, the process will analyze four extra signals in the second round, because of the influence of the first processing round. It is in this way that errors accumulate, making the extraction process more and more inaccurate.

To solve the problem, we design a linear data structure for storing message signals and segmenting the sequence of message signals correctly. The structure stores detected patterns and their lengths of continuous segments as *signal lines*. For example, assume that the signals are "000 333 111 22222 11111 33333," the embedding period is 1 second, and the frame rate is 4 fps in average so that the *basic length* is 4. While the extraction process detects the first "0", the "0" will be stored as the first line segment whose value is 0 and whose length is 1. While reading the second "0," which is the same pattern as the first one, the first line segment is extended to have a length of 2, and so on. In this way, the first line segment will be extended to have a length of 3 at last. Next, while the extraction process detects the fourth signal "3" which is a pattern different from that of the first line segment, the extraction process will create another line segment to store the signal "3" initially. In the end, the extraction process will store six line segments with each line segment having only two integers as its attributes, one being the message signal value and the other the length of the segment. In contrast with the simple way of storing every detected signal, the extraction process stores signals with much lower cost.

Next, the lines are analyzed *sequentially* by the proposed message extraction process. To solve the unequal-length problem of the signals in the line segments, the proposed message-extraction process decides the signals of each line segment by comparing its length with the *basic length* mentioned previously. For example, for the above example, the length of the first line segment is 3, and the basic length is 4. The message-extraction process will divide the length of each signal line segment by the basic length and *round* the result to ignore any possible little length variation in each line segment. Accordingly, the first line segment will be regarded to a *normal* segment of signal "0" though its length is smaller than the basic length with a difference of 1.

In practice, the proposed message signal detection scheme as just described may, in addition, encounter noise in dealing with the embedded message. For example, the message signals "3333333333" may become "3333103333" after display-and-imaging "attacks." In dealing with this problem, we consider short segments in the signal line as noise. Accordingly, a noise removal algorithm is adopted in this study to remove such short noise segments. The aim is to rearrange the short segments into the right positions to construct a complete noise-free signal line. In more detail, at first we deal with simpler cases as follows: 1) a noise segment N between two segments A and B both with the *same* value; 2) a noise segment N between the two segments A and B with *different* values. In Case 1), obviously the original signal line is separated by a noise segment. So, we *straighten* the signal line by combining the segments A, B, and N to be a longer segment with the same value as that of A and B. In Case 2), the noise segment N is on the boundary of two different segments A and B. If A and B are *long enough*, we may regard the output signal to be not influenced by the short noise segment, and so combine N into either of A and B. However, if either or both of A and B are short segments, we may consider the algorithm to be a process of gathering and discarding noise segments. That is, if noise segments A, B, and N are all not too long, we may eliminate them *by an ascending order* of their lengths. The reason why we adopt an *ascending* order is that we want to avoid combining longer noise segments with shorter signal segments because it is more possible that longer signal segments may be separated by shorter noise segments, so that shorter segments should be eliminated at first.

5.3 Proposed Techniques for Speeding up Message Extraction Process

It is desired to speed up the signal detection process so that the proposed system can be used for realtime applications. For this, we studied the use of an invariance property of the DFT coefficients. First, it is seen that the scaling operations have almost no effect on the DFT coefficient. On the other hand, the smaller the frame size, the faster the computation of the proposed processes, as observed in our experiments. The reason is that the DFT operations take long time to accomplish, and they are related to the image size, as can be seen from the following DFT formula:

$$F(u,v) = \frac{1}{M \times N} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) e^{-j2\pi(\frac{ux}{M}+\frac{vy}{N})}. \qquad (1)$$

Accordingly, in the proposed method we reduce the image frame size before the message embedding and extraction processes are conducted. The ratio of reduction is selected according to the original frame size of the input video.

Also, it is obvious that we can speed up the processes by frame down-sampling (i.e., using less video frames). However, in the proposed method the most important thing is the correctness of the extracted message, so the down-sampling rate should not be too high; otherwise, the lengths of the resulting line segments as mentioned previous will be too short so that it becomes difficult to discriminate noise from normal message signals by the length feature. As the result, we choose to take one frame from every two as the down-sampling rule in this study.

5.4 Proposed Algorithm for Message Extraction

Based on the above discussions, an algorithm for extraction the message is described in the following.

Algorithm 2. The message extraction process.

Input: a video V with its frame size reduced and frame sequence down-sampled.

Output: a sequence M of characters representing the message embedded in V.

Steps.

Step 1. Convert each frame F of V into a version in the YUV color space, and transform its Y component into a frequency spectrum F'.

Step 2. Take a 100×60 *thresholding window W* around the origin of F', and binarize W to obtain a binary image B by the moment-preserving thresholding technique [10].

Step 3. Construct masks M_1 and M_2 as illustrated in Fig. 8 to detect message signals in B by the following steps.

3.1 Apply the “OR” operation to B with mask M_1 to yield an image O_1.

3.2 Count the number N_1 of black pixels in O_1, and calculate a score S_1 as $S_1 = N_1/A_1$ where A_1 is the area of the circular regions of M_1.

3.3 Apply the “OR” operation to B with mask M_2 to yield another image O_2.

3.4 Count the number N_2 of black pixels in O_2, and calculate another score S_2 as $S_2 = N_2/A_2$ where A_2 is the area of the circular regions of M_2.

Step 4. Analyze S_1 and S_2 in the way described in Sec. 5.1 using a threshold value t (taken to be 0.7 in our experiments) to determine the message signal S embedded in frame F, and put S sequentially into a line structure T.

Step 5. According to the algorithm described in Sec. 5.2, conduct the following two steps:

5.1 regard line segments in T with the lengths smaller than half of the basic length as noise and correct them;

5.2 convert each resulting line segment in T into signals by analyzing its length and value, resulting in a signal sequence T'

Step 6. Find the initial signal in T', convert the subsequent sequence of signals into ASCII codes until the ending signal in T' is detected, and take the extracted sequence M of characters as the desired output message.

6. EXPERIMENTAL RESULTS

In this section, we show some experimental results to prove the effectiveness of the proposed method.

A. Experimental results of embedding messages

At first, we show in Fig. 9 some screenshots of the frames of a video and the corresponding frames after the message signals of “CV” are embedded.

Fig. 9. Video frames in which the message “CV” is embedded. (a1) and (a2) The original video frames. (b1) and (b2) The signal-embedded video frames.

B. Experimental results of detecting signals

Next, we show an example of the results of signal detection by moment-preserving thresholding in Fig. 10, and the recognition rate of signal detection in Table 2.

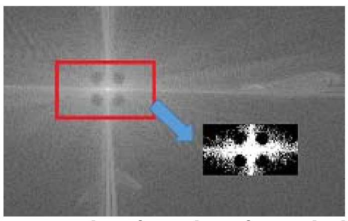


Fig. 10. An example of results of signal detection by moment-preserving thresholding.

C. Experimental results of extraction messages

Also, we show in Table 3 the rates of correctly-extracted message characters which were embedded in several videos.

Table 2. Experimental results of detecting signals by the proposed method, where **AS** means the number of all embedded signals; **CS** means the number of correctly-detected signals; **FS** means falsely-detected signals; and **RR** means the recognition rate.

Input video	Embedded message	AS	CS	FS	RR
1	PM7:00 EC129	1414	1403	11	99.22%
2	EC129	695	685	10	98.56%
3	Love You	1022	1019	3	99.70%
4	WuCJ	589	585	4	99.32%
5	I'm in 808	1236	1235	1	99.91%

Table 3. Recognition rates of the proposed message extraction process with embedding period 0.25s (**RC** = rate of correctness).

Input video	Embedded Message	Extracted Message	RC
1	It’s a Message-rich Video!	It’s a Message-rich Video!	98.8%
2	NCTU CVLab	NCTU CVLab	
3	I Love You, haha.	I Love You, haha.	
4	Chiao Tung	Chiio Tung	
5	Hey Hey! Let’s Go Fox!	Hey Hey! Let’s Go Fox!	

D. Experimental results of speeding up the message extraction process

Finally, we show in Table 4 the speedup rates of the proposed speed improving schemes. Comparing the data shown in the table, we see that the execution time is reduced to about 1/20 after speeding up.

Table 4. Experimental results of speeding up the message extraction process, where **RC** means whether the extracted message is correct by the speeded-up extraction process and the time unit is sec.

Input video	Execution Time	Method	RC
4	116.94	None	100%
	51.7246	Down-sampling	100%
	6.2332	Resizing	100%
	5.61157	Both	100%
Input video	**Execution Time**	**method**	**RC**
5	168.988	None	100%
	87.8472	Down-sampling	100%
	24.5968	Resizing	100%
	8.08563	Both	100%

7. CONCLUSIONS

In this study, we have proposed the use of message-rich video for pervasive communication. Through the designs of hardware and appropriate algorithms, several problems encountered in the creation of message-rich videos, including message embedding and extraction, have been solved so that such videos can be applied more easily. For message embedding, the use of diagonal circular signals has been proposed, by which the system can embed a 2-bit signal in each video frame. A synchronization mechanism by the uses of special signals and certain properties of the video frame rate has been proposed, by which the system can synchronize the message extraction work with the embedded message signals without transmitting any information. For message extraction, a signal detection method by moment-preserving thresholding based on the use of certain specially-designed masks has been proposed, by which the system can detect signals adaptively with a high recognition rate. Furthermore, schemes for speeding up the message extraction process by frame resizing and down-sampling have also been used. And the feasibility and effectiveness of all the proposed methods for solving the problems have been demonstrated by theoretical analyses and experimental results. In the future, more studies may be directed to improving the proposed method to increase the amount of data which can be embedded in a message-rich video and finding more robust signals for message embedding in videos with less distortions.

REFERENCES

[1] M. Wieser, “The computer for the 21st century,” *Scientific American, Special Issue on Communications, Computers, & Networks*, pp. 78–89, 1991.

[2] S. Poslad, *Ubiquitous computing: smart devices, Environments and Interactions*, John Wiley & Sons, Chichester, UK, 2009.

[3] B. Davis, “Signal rich art: enabling the vision of ubiquitous computing,” *Proc. of SPIE 7880: Media Watermarking, Security, & Forensics III*, N. D. Memon, J. Dittmann, A. M. Alattar, and E. J. Delp III, Eds., vol. 788002, San Francisco, CA, USA, Feb. 2011.

[4] Y. C. Chiu and W. H. Tsai, “Copyright protection by watermarking for color images against print-and-scan operations using coding and synchronization of peak locations in discrete Fourier transform domain,” *J. of Inform. Sci. & Eng.*, Vol. 22, No. 3, pp. 483-496.

[5] K. Solanki, et al, “‘Print and scan’ resilient data hiding in images,” *IEEE Tran. on Inform. Forensics & Security*, Vol. 1, pp. 464-478, 2006.

[6] F. Lefebvre, A. Gueluy, D. Delannay and B. Macq, “A print and scan optimized watermarking scheme,” *Proc. of IEEE 4th Workshop on Multimedia Signal Processing*, Cannes, France, pp. 511-516, 2001.

[7] Y. L. Lee and W. H. Tsai, “A new secure image transmission technique via secret-fragment-visible mosaic images by nearly-reversible color transformations,” *IEEE Trans. Circuits Syst. & Video Technol.*, vol. 24, no. 4, pp. 695-703, 2014.

[8] Y. L. Lee and W. H. Tsai, “Reversible data hiding by image encryptions and spatial correlation comparisons,” *J. of Inform. Sci. & Eng.*, accepted and to appear.

[9] Y. L. Lee and W. H. Tsai, “A new data transfer method via signal-rich-art code images captured by mobile devices,” *IEEE Trans. on Circuits & Systems for Video Technology*, submitted.

[10] W. H. Tsai, “Moment-preserving thresholding: a new approach,” *Computer Vision, Graphics, & Image Processing*, Vol. 29, pp. 377-393, 1985.

[11] Y. L. Lee. “Message-rich Multimedia – A New Tool for Pervasive Communication,” *Ph. D. Dissertation*, Institute of Computer Sci. & Eng., National Chiao Tung University, Hsinchu, Taiwan, November 2013.