Global Journal of Computer Science and Technology
Graphics & Vision
Volume 12 Issue 10 Version 1.0 June 2012
Type: Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals Inc. (USA)
Online ISSN: 0975-4172 & Print ISSN: 0975-4350

Implementation of Image Encoding Based on RGB and ARGB

By Fadi Al-Kalani & Mohamed Al Rabei

Delmon University for Science & Technology, Bahrain

Abstract - The study focuses on pixels while doing the process of implementation of image encoding. However, pixels can have enough power to protect the image and save the copyright of it. Images are nothing but set of pixels, each pixel can be considered as a box holding the colors' codes in a known sequence. Modifying those codes is kind of encoding. This study emphasizes the coding technique, the RIJNDAEL encryption and watermarking images.

Keywords : Image processing, image encoding, string encryption, watermark, image layers, RGB, ARGB, transparency.

GJCST-E Classification: FOR Code: I.4,E.3

Implementation of Image Encoding Based on RGB and ARGB

Fadi Al-Kalani [α] & Mohamed Al Rabei [α]

Abstract - The study focuses on pixels while doing the process of implementation of image encoding. However, pixels can have enough power to protect the image and save the copyright of it. Images are nothing but set of pixels, each pixel can be considered as a box holding the colors' codes in a known sequence. Modifying those codes is kind of encoding. This study emphasizes the coding technique, the RIJNDAEL encryption and watermarking images.

Keywords : Image processing, image encoding, string encryption, watermark, image layers, RGB, ARGB, transparency.

I. Introduction

The encoding process will be applied on 32-bit per pixel standards images (i.e. BMP, GIF, JPG and PNG).Pixels are encoded into four parts (RGBA) or (ARGB). (A) stands for Alpha, (R) for Red, (G) for Green and (B) for Blue.

ALPHA	*RED*	*GREEN*	*BLUE*
(0-7) bits	*(8-15) bits*	*(16-23) bits*	*(24-31) bits*

Fig. 1: The format of ARGB pixel

RGB was uniquely used as a base of pixels and to reproduce and present a broad array of colors while alpha has been added later to represent the transparency of the color [4].

Using RGB for encoding in this paper doesn't mean using old methods. RGB is still alive; it's the core of ARGB or even RGBAX. While alpha will be excluded from being processed in the first encoding method and will included later on.

As mentioned, the structure of a pixel (let us refer to it as color) is a set of 32 bits; 32 bits = 4 bytes. 4 bytes/ 4 sets {A, R, G & B} = 1 bytes for each set. Therefore; Only 3 bytes will be modified when using the RGB encoding method while 4 bytes will be used in the ARGB encoding method. Hence, the described encoding method is light compared to those which use 256 or more [5, 6].

Encoding and encryption are both routines performed on data; however the end results are quite different. In the case of encryption the purpose is to disguise the data such that it can't be read, except by the intended recipient. On the other hand, encoding is used merely to transform data into a more suitable format; it is the process of putting a sequence of objects (characters, letters, numbers, punctuation or any storage data type) into a specialized format for efficient transmission or storage.

Consequently, encoding and encryption can be integrated to secure data, to prevent others from reaching it. Encryption is needed to prevent hackers, and encoding to communicate, transmit or exchange data [9].

The implementation is accomplished in three integrated phases to get the full vision.

1. Encryption (used to encrypt the keys of encoding and decoding)
2. Encoding/Decoding methods
3. Watermarking the image.

II. Implementation

We have two directions in term of implementing our image encoding/ decoding. The first demonstrates the encoding processes. And the other is for the decoding. The process is described in the following figure:

a) *Encoding Processes*

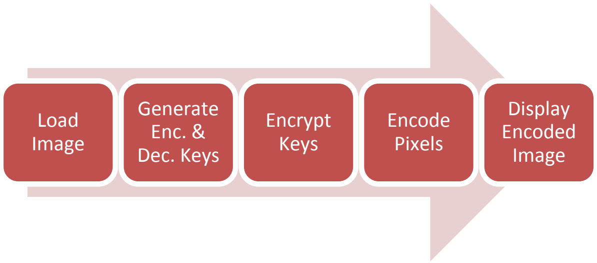

Fig. 2: Encoding process

As shown in figure 2, the first step, we start encoding by loading the image we'd like to encode. In the second step, we generate encoding key as well as decoding keys. These keys are strongly recommended to be saved safely. They are basically the sole of encoding and decoding. Hence, we will encrypt them in such a way that we empower their safety. Then encrypt the keys and start encoding each pixel in the image. Finally, we display the new encoded image (or save it into a file).

Author α : Fadi Al Kalani, Department of Computer Science, Delmon University for Science & Technology, Bahrain.
E-mail: fadek2000a@yahoo.com

b) Decoding Processes

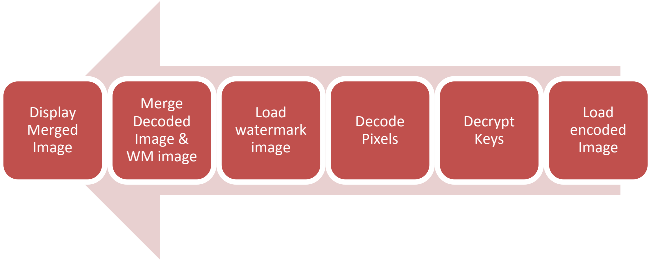

Fig. 3: Decoding process

Referring to figure 3 we realize that the decoding process starts by loading the encoded image, then decrypting the keys and verify them in the following step. While in the third step the image to decoded. The process ends by watermarking the image.

The three main important steps in both processes are:

1. Encryptions/Decryption the keys
2. Encoding/Decoding the image
3. Watermarking it.

c) Encryption/ Decryption

Generally, our methods of encoding images depend on a generated key of type integers. It contains 6 to 8 digits. And the same is for decryption. Thus makes it "unsafe". So we found that there is a need to encrypt them.

There are two kinds of encryption ciphers that use keys.

a. Single key encryption
b. Two keys based encryption

Two keys are used to strengthen the security; a public key which is used within the team to encrypt the data, and a private key which no data will be decrypted without it [2].

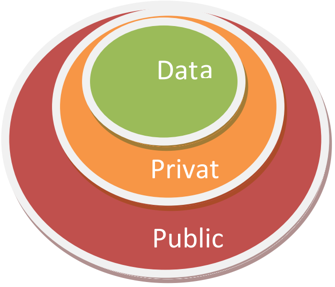

Fig. 4: Two Keys based Encryption

Similar to encryption ciphers, in fact the encoding process; explained through this paper is a single key encoding and another key – decoding key – is generated based on the first one. As a result, the encoding method that we develop supports both the Single and the double key encoding.

Practically, "Good encryption algorithms are hard to come by. They are exceptionally difficult to invent, and even when a new one is created, it is often quickly laden with patents and export restrictions, making it inconvenient or even impossible for others to reuse"[1].

Hence, we tried to use a very well-known encryption cipher called "RIJNDAEL Algorithm". RIJNDAEL is two ways encryption and used in keys decryption as well. It is one of the most powerful encryption solutions that serve the need in our research among several good encryption techniques of "System Security Cryptography" which is provided by .NET environment.

Most variables needed for encryption using RIJNDAEL class could have static values like hash algorithm, Initial vector, salt and others. These variables can be managed dynamically to empower encryption/ decryption. More encryption power in is better encoding [10].

d) Encoding / Decoding images

As mentioned previously, we have used two methods for encoding/ decoding images based on pixels.

i. RGB based encoding/decoding
ii. ARGB based encoding/decoding

Encoding using RGB and ARGB are almost the same in their structures. The only difference is that in the first one encodes the values of Red, Green and Blue whereas the second method encodes Alpha value - the transparency

The main function is "Encode". It's of type color (pixel). It takes source color and the encoding key and returns a new color to be replaced at the same position. Simply, the idea of the encoding is to rearrange the color in hexadecimal format with the help of the generated encoding key. Then convert the new hexadecimal value to color data type to present a new Red, Green and Blue or new (Alpha, Red, Green and Blue in case of ARGB).

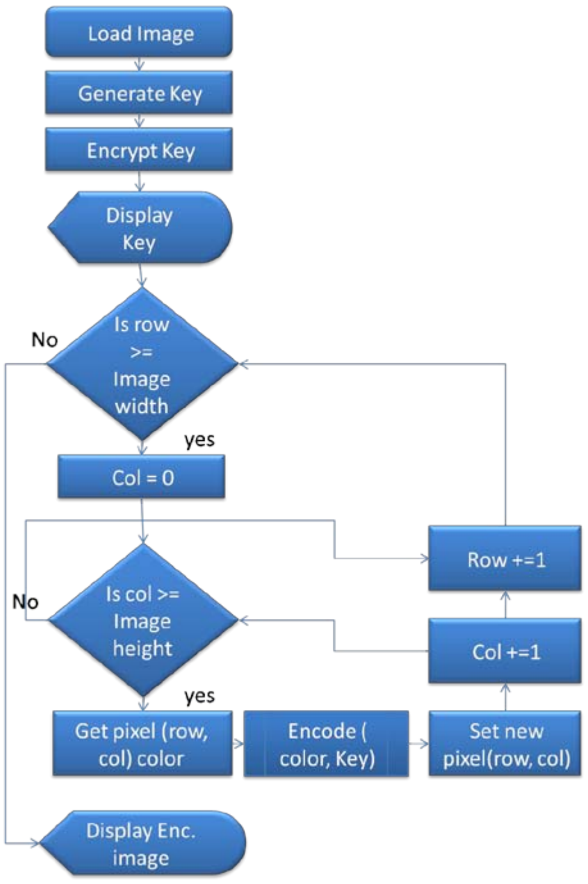

Fig. 5: Flowchart of encoding image

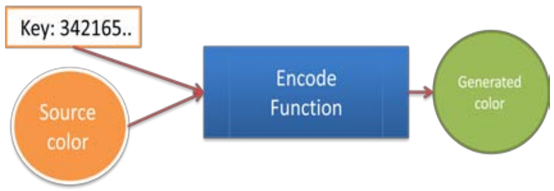

Fig. 6: Encode Functuin

The process used to decode the image is accomplished by generating a decoding key based on the key used in decoding. Based on the decryption key, we use the same function of encoding with different resources; the encoded image as well as the decoding key.

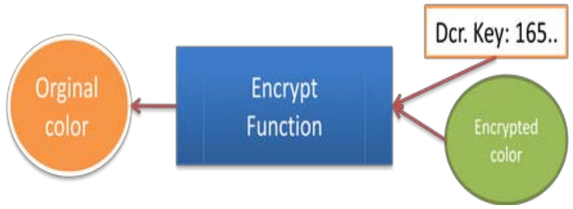

Fig. 7: Decryption Process

Due to the fact that this kind of encoding is based on independent small blocks (pixels), it cannot be classified as strong encoding. However, our challenge was to develop a method based on recoloring the pixels. But is that all? off course not. What remains is watermarking the image.

e) Adding Watermark

Adding watermark to an image is the process of embedding information (usually text or image) into an image in a way that it is difficult to remove [3]. Watermark is highly recommended for saving the copyrights especially in our case.

Watermark is not a part of encoding, but we took the decision to integrate it with our encoding system to add the flavor of security and to save the copyright [7, 8].

Programmatically, Adding watermark means merging two layers (or two images) one over the other and controlling the transparency of both based on the following formulas:

$$C_o = C_a\alpha_a + C_b\alpha_b(1-\alpha_a)$$

$$\alpha_o = \alpha_a + \alpha_b(1-\alpha_a)$$

Where C_o is the result of the operation [5].

C_a is the color of the pixel in Picture A; C_b is the color of the pixel in Picture B

α_a and α_b are the alpha of the pixels in Picture A and B respectively[6].

Programmatically, we can implement the previous equations easily like the following:

For the first formula:

$$C_o = (C_a\alpha_a/255) + (C_b\alpha_b(255-\alpha_a)/(255*255);$$

For the second formula:

$$\alpha_o = \alpha_a + (\alpha_b(255-\alpha_a)/255);$$

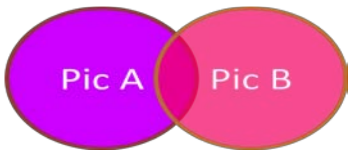

Fig. 8: Pic. A over Pic. B

Implementation of watermarks can be done in two steps.

I. Preparing the stamp image (signature).

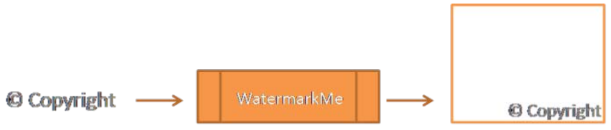

Fig. 9: Process line of preparing Signature image

In the preparation step, the size of the signature image is checked and compared to the decoded image, to assure that it fits the decoded image or needs resizing.

The next is to create a new empty image with the same size of the decoded image to draw the signature in down right corner or any elsewhere primary chosen.

I. Merging the signature and the decoded image.

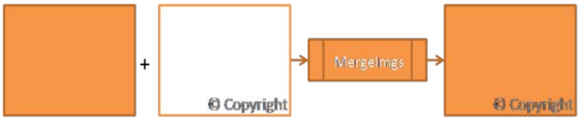

Fig. 10: Process line of merging Signature and decoded image

Merging both images together is merging each pixel of them based on the formulas mentioned before; within this process the transparency of the signature is to be set to be half of the real value of it.

As a result of our work, the following screenshots show the images before and after the encoding. They show the original image, and how it has been encoded using both methods RGB and ARGB.

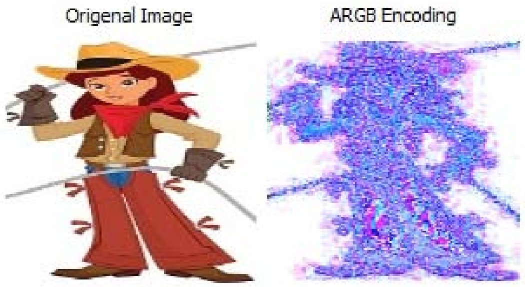

Fig. 11: An image before and after RGB encoding

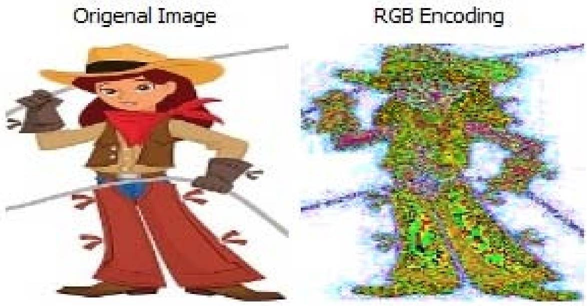

Fig. 12: An image before and after ARGB encoding

Due to the fact that this encoding method is based on encoding each pixel in separate, we clearly notice that it is not powerful for encoding plain colors that have same hexadecimal values for all bytes of the pixel like white color (A=FF, R=FF, G=FF, B=FF) as seen in the backgrounds in figure 11 and 12. Other than that it will work fine.

III. Conclusion

We tried in this study to test image encoding/decoding by changing the sequence of hexadecimal values of RGB and ARGB. Each pixel is encoded to have the same structure and different values. We provided the encoding method by generating public and private keys. However, we used a function to generate a decoding key as a gate for decoding. Because we have two keys, we worked on a fast and powerful encryption cipher called "RIJNDAEL Cipher". Finally, to strengthen the encoding we watermarked the decoded images.

References Références Referencias

1. Richard Heathfield, Lawrence Kirby, Et Al. (2002) C Unleashed.
2. Niels Ferguson, Bruce Schneier, Tadayoshi Kohno (2010). Cryptography Engineering: Design Principles and Practical Applications. Wiley;
3. Rafael C. Gonzalez, Richard E. Woods, Steven L. Eddins (2009). Digital Image Processing Using MATLAB, Gatesmark Publishing; 2nd edition,
4. Rafael C. Gonzalez, Richard E. Woods (2007). Digital Image Processing. Prentice Hall; 3rd edition,.
5. Alvy Ray Smith (1995), Image Compositing Fundamentals. Microsoft Tech Memo 4.
6. Alvy Ray Smith (1995). Alpha and the History of Digital Compositing. Microsoft Tech Memo 7,
7. Gengming Zhu, Nong Sang (2008), Watermarking Algorithm Research and Implementation Based on DCT Block. World Academy of Science, Engineering and Technology 45.
8. Chaelynne M. Wolak (2000), Digital Watermarking. A paper submitted in fulfillment of the requirements for DISS 780, Nova Southeastern University.
9. Tom St Denis (2007), Cryptography for Developers, Syngress.
10. Joan Daemen, Vincent Rijmen (2002), "The Design of Rijndael: AES - The Advanced Encryption Standard." Springer.