การอำพรางข้อมูลบนพื้นฐานของความถี่โดยใช้ตารางการแบ่งนับที่ถูกประมาณค่าในช่วงขนาด 32x32 และผลการแปลงโคไซน์แบบไม่ต่อเนื่อง

นายนที วงศ์อุไร

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาการคอมพิวเตอร์และเทคโนโลยีสารสนเทศ ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์
คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2555

FREQUENCY-BASED STEGANOGRAPHY USING 32X32 INTERPOLATED QUANTIZATION TABLE AND DISCRETE COSINE TRANSFORM

Mr. Natee Vongurai

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Science Program in
Computer Science and Information Technology
Department of Mathematics and Computer Science
Faculty of Science
Chulalongkorn University
Academic Year 2012

Thesis Title	FREQUENCY-BASED STEGANOGRAPHY USING 32X32 INTERPOLATED QUANTIZATION TABLE AND DISCRETE COSINE TRANSFORM
By	Mr. Natee Vongurai
Field of Study	Computer Science and Information Technology
Thesis Advisor	Assistant Professor Suphakant Phimoltares, Ph.D.

Accepted by the Faculty of Science, Chulalongkorn University in Partial Fulfillment of the Requirements for the Master's Degree

....Dean of the Faculty of Science

(Professor Supot Hannongbua, Dr. rer. nat.)

THESIS COMMITTEE

....Chairman

(Assistant Professor Saranya Maneeroj, Ph.D.)

....Thesis Advisor

(Assistant Professor Suphakant Phimoltares, Ph.D.)

....External Examiner

(Saichon Jaiyen, Ph.D.)

นที วงศ์อุไร : การอำพรางข้อมูลบนพื้นฐานของความถี่โดยใช้ตารางการแบ่งนับที่ถูกประมาณค่าในช่วงขนาด 32x32 และผลการแปลงโคไซน์แบบไม่ต่อเนื่อง (FREQUENCY-BASED STEGANOGRAPHY USING 32X32 INTERPOLATED QUANTIZATION TABLE AND DISCRETE COSINE TRANSFORM)
อ.ที่ปรึกษาวิทยานิพนธ์หลัก : ผู้ช่วยศาสตราจารย์ ดร.ศุภกานต์ พิมลธเรศ, 56 หน้า.

การอำพรางข้อมูลในภาพเป็นศิลปะของการซ่อนความมีอยู่ของสารสนเทศในภาพ การอำพรางข้อมูลสามารถจำแนกเป็นสองชนิดในรูปแบบของโดเมนเชิงพื้นที่และโดเมนความถี่ การอำพรางข้อมูลบนพื้นฐานความถี่โดยใช้ผลการแปลงโคไซน์แบบไม่ต่อเนื่องและเจเพ็กถูกนำเสนอในการศึกษานี้ บล็อกที่ใหญ่ขึ้นขนาด 32x32 จุดภาพจะถูกใช้ร่วมกับตารางแบ่งนับขนาด 32x32 ที่สอดคล้องกันซึ่งถูกสร้างโดยเทคนิคการประมาณค่าในช่วงเสมือนพหุนามกำลังสามแทนการใช้บล็อกขนาด8x8จุดภาพร่วมกับตารางแบ่งนับขนาด8x8ผลการทดลองแสดงว่าเวลาในการคำนวณลดลงและความจุของสารลับเพิ่มขึ้นในขณะที่ยังคงคุณภาพและขนาดของภาพเจเพ็กที่มีการซ่อนสารลับแล้ว

ภาควิชา คณิตศาสตร์และวิทยาการคอมพิวเตอร์ ลายมือชื่อนิสิต

สาขาวิชา วิทยาการคอมพิวเตอร์และเทคโนโลยีสารสนเทศ ลายมือชื่อ อ.ที่ปรึกษาวิทยานิพนธ์หลัก

ปีการศึกษา 2555

5373612823: MAJOR COMPUTER SCIENCE AND INFORMATION TECHNOLOGY
KEYWORDS: DISCRETE COSINE TRANSFORM/ QUANTIZATION TABLE/ SPLINE INTERPOLATION

NATEE VONGURAI: FREQUENCY-BASED STEGANOGRAPHY USING 32X32 INTERPOLATED QUANTIZATION TABLE AND DISCRETE COSINE TRANSFORM. ADVISOR: ASSISTANT PROFESSOR SUPHAKANT PHIMOLTARES, Ph.D., 56 pp.

Image steganography is the art of hiding the existence of the information in the image. The steganography can be categorized into two types in spatial domain and frequency domain. In this study, the frequency based image steganography using Discrete Cosine Transformation (DCT) and JPEG is proposed. Instead of using 8x8-pixel blocks with the 8x8-pixel quantization table, a larger block of size 32x32 will be used with a corresponding 32x32-pixel quantization table created by cubic spline interpolation technique. The experimental results show that the computation time is reduced and the capacity of the secret messages is increased while maintaining the quality and the size of JPEG stego-image.

Department: Mathematics and Computer Science Student's Signature ....

Field of Study: Computer Science and Information Technology Advisor's Signature ....

Academic Year: 2012

ACKNOWLEDGEMENTS

I would like to acknowledge my thesis advisor, Assistant Professor Dr. Suphakant Phimoltares, at the department of Mathematics, faculty of Science, for helping me in many things. He did encourage me to finish the thesis and also give me the knowledge.

CONTENTS

Page

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

CHAPTER I

INTRODUCTION

Recently, sending the information via the internet is very popular because this method is fast and convenient compared to other methods. However, despite the advantage, there is also a security issue that needs to be concerned during the sending process. The information in the network can be sniffed off. Therefore, some information may need to be protected in order to ensure that no one opens it. There are many ways to protect such information. Among them, encryption in the cryptography is a popular scheme, which will transform the plain information into secret code for which some key or password is needed to decrypt or open. However, with this kind of protection, the existence of secret message might attract the attention of people especially hacker.

In order to avoid this problem, steganography is used. Steganography is a way to hide the secret message in the media such as image, audio or video. In other words, the existence of the secret message is unavailable. This means that the unwanted parties cannot see the difference between a cover image which is an original image that hasn't been embedded the secret messages and a stego-image including the secret messages. Unlike the cryptography, steganography focuses on difficulty of an existence of secret messages while cryptography focuses on making the secret messages not able to understand except the sender and receiver. In addition, steganography is similar to digital watermarking but the difference is that the main purpose of digital watermarking is to prevent removal of the content in the watermarked data without considering its existence [1].

For the image steganography, there are two types including spatial domain method and frequency domain method. For the spatial domain method, the secret messages will be hidden in the pixels of cover image. For example, from Chan et al.'s research [2], data are hidden by using simple LSB (the least significant bit) with an optimal pixel adjustment process. In case of the frequency domain the cover image will first be transformed from spatial domain to frequency domain before embedding the secret messages. In addition, for the spatial domain method the capacity of the secret messages that can be embedded is greater than that of frequency domain method, however, it is easier to be detected with the Human Vision System (HVS).

In this work, frequency domain method is chosen. Initially, the transformation from spatial domain to frequency domain is applied to an image with the advantage of the characteristic of HVS that is sensitive to the low frequency range and insensitive to high frequency range. Once the image is transformed into frequency domain, the high frequency range can be discarded. In addition, there are various transform techniques used in steganography works such as DCT, DFT and DWT. However, many image algorithms use DCT because unlike DFT, there is no need to work with the imaginary part. In this work, DCT is used to transform cover image into frequency domain. Also, the image file format used in this thesis is based on JPEG. The advantage of JPEG is its compression efficiency in high quality. With this reason, it is widely used over the internet. Therefore, using JPEG as the image format file will reduce the chance to be suspected.

In image steganography, the capacity of the secret messages that can be embedded and the quality of the stego-image are considered as the important factors. It is pointless if more secret messages can be embedded but the quality is degraded such that it can easily be detected. Likewise, if more secret messages can be embedded and the quality is good but the computation time is significantly increased then it would not be practical used. Moreover, the size of stego-image should be also considered since if the size is increased too much then it can easily be suspected. Therefore, for this work, four significant factors in our consideration are capacity, quality, stego-image's size and computation time.

This thesis consists of five sections starting with the introduction. Next, Section 2 reviews about the related work. Furthermore, Section 3 reviews the theory used in this work. Subsequently, Section 4 is about our proposed method to embed the secret messages. In Section 5 the experiments will be conducted together with the discussion about the results. Finally, the conclusion will be in the Section 6.

1.1Objectives

The main objectives of this study are the following:

1. To improve the capacity of the secret messages that can be embedded in the cover image
2. To reduce the computation time used in the process
3. To maintain the quality of the stego-image
4. To reduce the size of the stego-image

1.2 Scope

1. In this work, only gray scale image of size 512x512 is used as a cover image.
2. Frequency domain is mainly considered in this study and only Discrete Cosine Transformation (DCT) is used to transform the image to frequency domain.
3. Size of pixel-block does not exceed than 32x32 pixels.
4. The size of the stego-image (KB), capacity of the secret messages (bits), Peak Signal to Noise Ratio (PSNR) and computation time are used as the evaluation criteria of this work.

1.3 Research methodology

The cover image is partitioned into 32x32 non-overlapping pixels blocks. After that, DCT is applied to transform each block into frequency domain. Each DCT block is then quantized by 32x32 interpolated quantization table. Next, Secret messages are embedded. Finally, employ JPEG encoding to each block and the JPEG stego-image is obtained.

In order to archive the objectives above, the following tasks will be stated below with the detail of the work.

- Feasibility study: First the related work and related knowledge were studied to find the feasibility of the work.
- Theoretical Study: The theoretical background that related to our work was studied.
- Plan and design: After the theoretical study is finished, the system and process were planned and designed
- Algorithm design and analysis: The algorithms were written according to plan and design task.
- Experiment setup: After the algorithms task is finished, the experiments were conducted
- Collect and improve results: The results were collected, evaluated, and improved.
- Documentation: writing the thesis

Table 1.1: Task schedule

Step	Description (Thesis started in March, 2011)	Month sequence																	
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	Feasibility Study	—	—	—	—														
2	Theoretical Study			—	—	—	—												
3	Plan and design				—	—	—	—	—										
4	Algorithm design and analysis						—	—	—	—	—	—	—						
5	Experiment setup										—	—	—	—					
6	Collect and improve results													—	—	—	—		
7	Documentation																—	—	—

1.4 Benefits

The expected outcomes are the new method and larger quantization table that

- Reduce the computation time.
- Increase the capacity of the secret messages that can be embedded.
- Maintain the image quality .
- Reduce the size of JPEG stego-image.

CHAPTER II

RELATED WORK

As mentioned earlier, image steganography can be divided into 2 types which are the spatial domain and the frequency domain.

2.1 Jpeg-Jsteg

In the recent survey, Jpeg-Jsteh is famous JPEG steganography software. It divides a cover image into many 8x8-pixel blocks. In embedding process, the secret messages are embedded into the least significant bits of the quantized DCT coefficients whose values are not -1, 0, or 1. However, due to the quantization process, many of the DCT quantized coefficients equal to the zero. Therefore, the capacity of Jpeg-Jsteg is quite limited. In addition, most of the DCT quantized coefficients whose values are not equal to -1, 0, 1 are located in the region of lower frequency. Embedding the secret messages in that region causes degradation in image quality. In addition, for the Jpeg-Jsteg, the quantization table used in quantization process is the standard quantization table (table 2.1) that use in the JPEG standard.

Table 2.1 The standard quantization table

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

The embedding algorithm of Jpeg-Jsteg is shown as follows

Begin

1. Partition a cover image into non-overlapping 8x8-pixel blocks (Figure 2.1).

120	118	117	115	113	103	97	109
136	142	149	154	146	94	44	40
152	152	150	139	121	84	49	30
161	152	135	105	80	80	82	57
143	144	140	125	107	87	74	80
120	134	151	164	161	104	58	104
135	127	121	139	153	128	106	130
165	125	83	88	113	149	175	150

Figure 2.1 Example of a 8x8-pixel block

2. Apply DCT transformation to each block, starting from left to right and top to bottom. The DCT coefficients are obtained (Figure 2.2).

-85.375	157.53583	-37.688544	8.3310555	25.125	-15.856658	2.7576987	0.5572381
-60.085802	87.127991	-47.376592	-20.562611	-8.7885486	1.0020068	2.9865897	2.0581272
35.507345	-100.10512	27.419864	30.961921	7.7388992	0.6085992	-4.2740485	-0.5819617
7.0381278	-27.806318	-37.274823	-40.023015	28.52202	-23.230648	10.991767	-2.3279168
-15.625	-45.601381	92.968436	17.401702	-32.625	25.018053	-7.7959083	0.478931
-3.2473912	-16.824626	19.329987	-3.7328031	-2.9207438	2.3708411	0.2163796	-0.3283963
1.122362	-6.9482956	1.9759515	-3.3278221	0.7181147	-0.6031264	1.0801362	0.1508914
-1.3885579	-2.9440283	2.8337764	1.0200711	-0.1937895	0.5899321	-0.4549103	0.0241823

Figure 2.2 Example of the DCT coefficient

3. Quantize each DCT coefficient block by dividing each block with the standard 8x8-pixel quantization table and then rounding off. In addition, this is the step where the lossy compression occurred. The quantized DCT coefficients are obtained (Figure 2.3).

-5	14	-4	1	1	0	0	0
-5	7	-3	-1	0	0	0	0
3	-8	2	1	0	0	0	0
1	-2	-2	-1	1	0	0	0
-1	-2	3	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Figure 2.3 Example of the quantized DCT coefficient

4. Embed the secret messages into each quantized DCT coefficient block. The secret messages are embedded into the least significant bit of each quantized coefficient whose value is not -1, 0 and 1 (Figure 2.4).

-4	15	-4	1	1	0	0	0
-5	8	-2	-1	0	0	0	0
4	-8	3	1	0	0	0	0
1	-3	-2	-1	1	0	0	0
-1	-1	2	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Figure 2.4 Example of the embedded quantized DCT coefficient

5. Employ JPEG encoding algorithms. In this step, the stego-image JPEG is obtained.

End

Figure 2.5 shows the overview process of the Jpeg-Jsteg's embedding process.

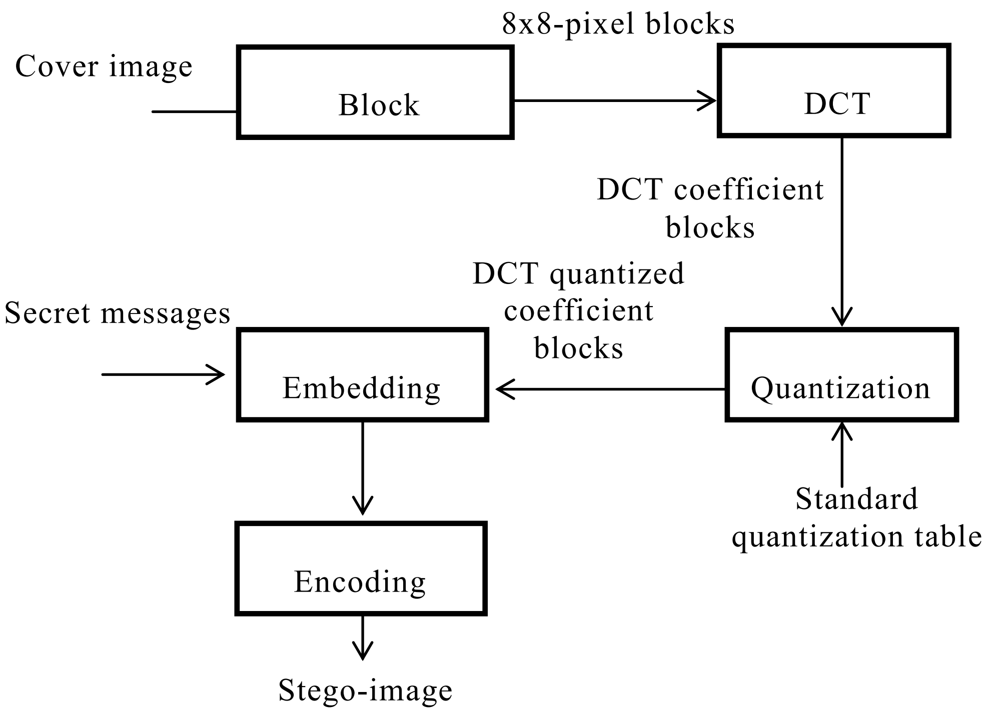


Figure 2.5 The block diagram of Jpeg-Jsteg embedding process

The extracting algorithms of Jpeg-Jsteg is shown as follows

Begin

1. Decode JPEG file by employing JPEG decoding algorithms.
2. Extract the secret messages from each block by collecting the information from the least significant bit of each embedded quantized coefficient.
3. De-quantize each block by multiplying that with the standard quantization table. The DCT coefficients are now obtained (Figure 2.6).

-64	165	-40	16	24	0	0	0
-60	96	-28	-19	0	0	0	0
56	-104	48	24	0	0	0	0
14	-51	-44	-29	51	0	0	0
-18	-22	74	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Figure 2.6 Example of the DCT coefficient

4. Apply the inverse DCT (IDCT) to each 8x8-pixel DCT coefficient block. The 8x8-pixel blocks are now obtained (Figure 2.7).

142	121	118	132	124	97	96	118
147	146	146	143	128	100	71	53
159	165	156	131	110	89	53	16
163	158	141	115	92	74	56	43
143	133	134	140	119	81	68	79
123	118	140	173	162	114	88	96
136	125	128	147	155	142	127	122
166	139	102	87	113	152	166	157

Figure 2.7 Example of a 8x8-pixel block

5. Merge all 8x8-pixel blocks and obtain the cover image.

End

Figure 2.8 below shows the overview process of the Jpeg-Jsteg's embedding process.

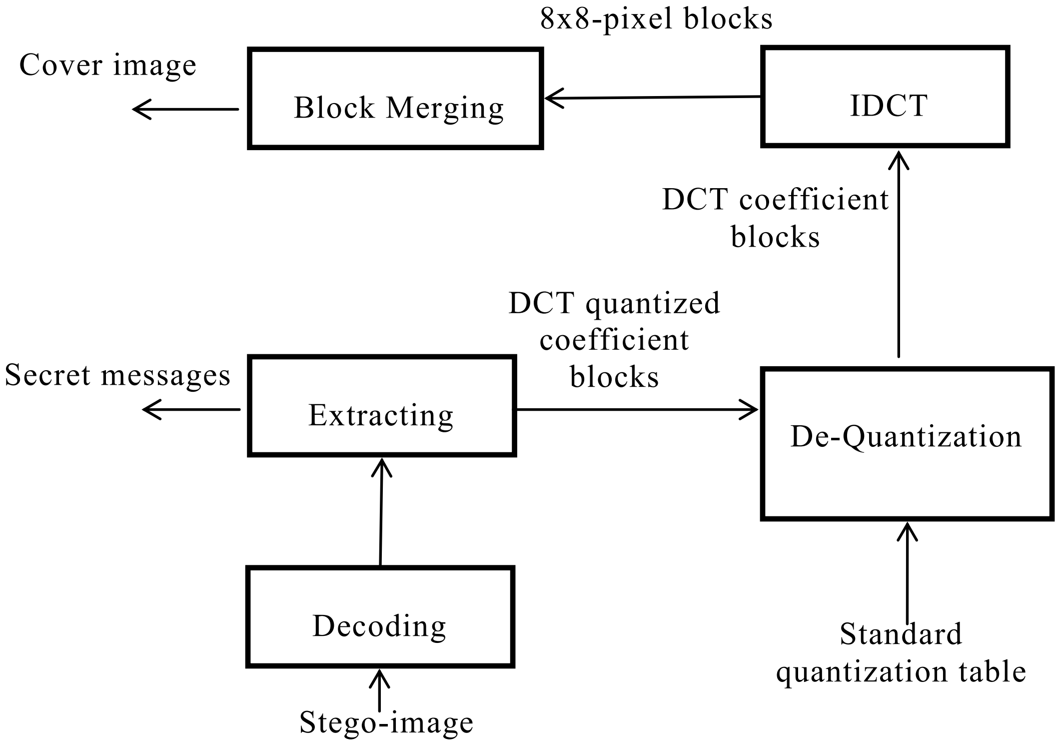


Figure 2.8 The block diagram of Jpeg-Jsteg's extracting process

2.2 Chang's method

Chang et al. [3] proposed a novel steganographic method based on JPEG. Likely, a cover image is divided into 8x8-pixel blocks. To improve the capacity of the secret messages and maintain the image quality, the secret messages are embedded in the region of middle frequency of the DCT quantized coefficient and the 8x8 quantization table is modified such that the pixels in the region of middle frequency of the quantization table is set to be 1s (table2.2). In addition, the secret messages are embedded into the two least significant bits in each middle coefficient.

Table 2.2 Chang's modified quantization table

16	11	10	16	1	1	1	1
12	12	14	1	1	1	1	55
14	13	1	1	1	1	69	56
14	1	1	1	1	87	80	62
1	1	1	1	68	109	103	77
1	1	1	64	81	104	113	92
1	1	78	87	103	121	120	101
1	92	95	98	112	100	103	99

The embedding algorithm of Chang's method is shown as follows

Begin

1. Partition a cover image into non-overlapping 8x8-pixel blocks (Figure 2.1).

2. Apply DCT transformation to each block, starting from left to right and top to bottom (Figure 2.2)

3. Quantize each DCT coefficient block by dividing each block with the Chang's modified quantization table (Table 2.2) and then rounding off. The quantized DCT coefficients are now obtained (Figure 2.9).

-5	14	-4	1	25	-16	3	1
-5	7	-3	-21	-9	1	3	0
3	-8	27	31	8	1	0	0
1	-28	-37	-40	29	0	0	0
-16	-46	93	17	0	0	0	0
-3	-17	19	0	0	0	0	0
1	-7	0	0	0	0	0	0
-1	0	0	0	0	0	0	0

Figure 2.9 Example of the quantized DCT coefficient in Chang's method

4. Embed the secret messages into each quantized DCT coefficient block. The secret messages are embedded into the two least significant bits of each quantized coefficient n the middle part of Chang's modified quantization table that has value '1' (in the middle frequency part) starting from left to right and top to bottom. The result is shown below (Figure 2.10).

-5	14	-4	1	25	-16	2	1
-5	7	-3	-23	-9	1	3	0
3	-8	25	30	10	1	0	0
1	-27	-38	-37	28	0	0	0
-15	-46	95	16	0	0	0	0
-3	-18	17	0	0	0	0	0
3	-7	0	0	0	0	0	0
-2	0	0	0	0	0	0	0

Figure 2.10 Example of the embedded quantized DCT coefficient in Chang's method

5. Employ JPEG encoding algorithms.

End

Figure 2.11 shows the overview of Chang's embedding process.

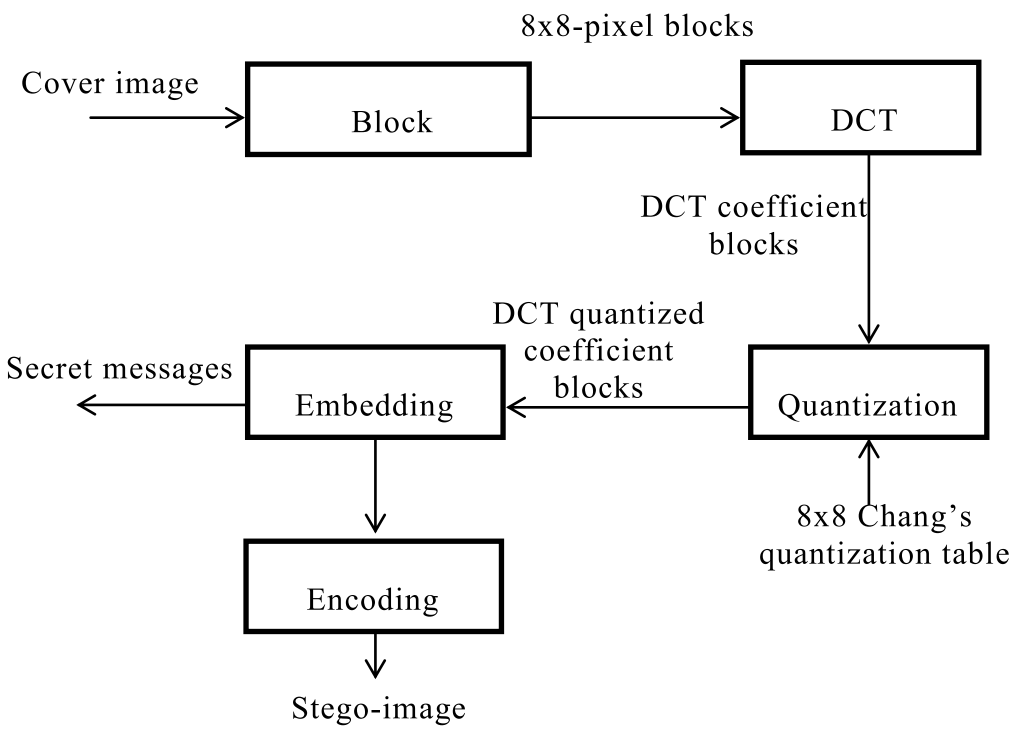


Figure 2.11 The block diagram of Chang's embedding process

The extracting algorithms of Chang's method is shown as follows

Begin

1. Decode JPEG file by employing JPEG decoding algorithms.

2. Extract the secret messages from each block by collecting the information from the two least significant bit of each embedded quantized coefficient starting from left to right and top to bottom (Same order as in the embedding process).

3. De-quantize each block by multiplying that with the Chang's modified quantization table. Figure 2.12 shows an example of the DCT coefficient in Chang's method.

-80	154	-40	16	25	-16	2	1
-60	84	-42	-23	-9	1	3	0
42	-104	25	30	10	1	0	0
14	-27	-38	-37	28	0	0	0
-15	-46	95	16	0	0	0	0
-3	-18	17	0	0	0	0	0
3	-7	0	0	0	0	0	0
-2	0	0	0	0	0	0	0

Figure 2.12 Example of the DCT coefficient in Chang's method

4. Apply the inverse DCT (IDCT) to each 8x8-pixel DCT coefficient block. The 8x8-pixel blocks are now obtained (Figure 2.13).

127	114	113	121	122	104	98	117
133	143	153	152	137	98	55	38
146	156	152	133	118	91	50	25
162	153	127	99	89	80	69	66
147	144	136	128	112	83	70	82
123	133	152	170	155	107	80	89
134	126	124	142	152	132	116	120
165	125	81	85	126	153	159	161

Figure 2.13 Example of the 8x8-pixel block in Chang's method

5. Merge all 8x8-pixel blocks and obtain the cover image.

End

The block diagram in Figure2.14 shows the overview of Chang's extracting process.

2.3 Almohammad's method

Almohammad et al. [4] proposed a steganographic method based upon blocks of size 16x16 pixels and modified 16x16-pixel quantization table with the same technique used by Chang et al (table2.3). They found that their method can embed more secret messages than the method based on 8x8-pixel blocks. Additionally, the computation time of their method is a bit less than Chang et al.'s method as well.

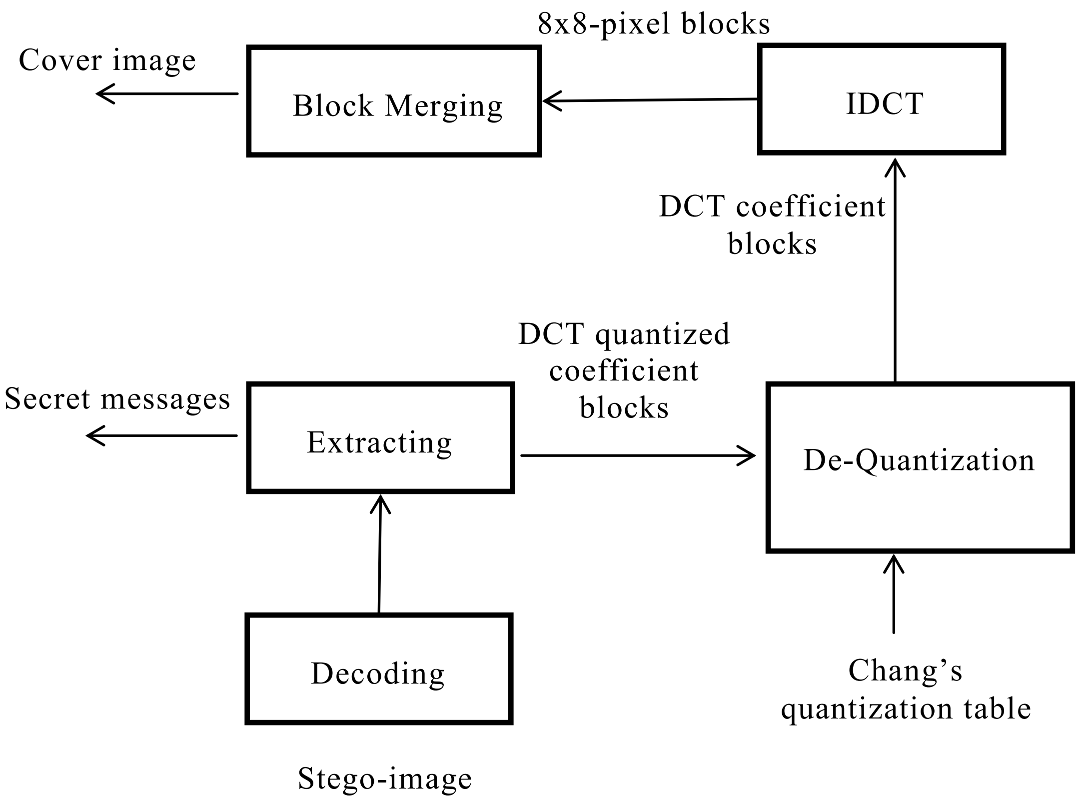


Figure 2.14 The block diagram of Chang's extracting process

Table 2.3 16x16 Quantization table

16	8	7	6	6	1	1	1	1	1	1	1	1	1	1	1
7	7	6	6	1	1	1	1	1	1	1	1	1	1	1	30
7	6	6	1	1	1	1	1	1	1	1	1	1	1	30	28
6	8	1	1	1	1	1	1	1	1	1	1	1	32	25	29
8	1	1	1	1	1	1	1	1	1	1	1	32	35	32	38
1	1	1	1	1	1	1	1	1	1	1	35	40	42	40	35
1	1	1	1	1	1	1	1	1	1	35	44	42	40	35	31
1	1	1	1	1	1	1	1	1	35	44	44	50	53	52	45
1	1	1	1	1	1	1	1	31	34	44	55	53	52	45	39
1	1	1	1	1	1	1	31	34	40	41	47	52	45	52	50
1	1	1	1	1	1	30	32	36	41	47	52	54	57	50	46
1	1	1	1	1	36	32	36	44	47	52	57	60	60	55	50
1	1	1	1	36	39	42	44	48	52	57	61	60	60	55	51
1	1	1	39	42	47	48	46	49	57	56	55	52	51	54	51
1	1	41	46	47	48	48	49	53	56	53	50	51	52	51	50
1	43	47	47	48	48	49	57	57	56	50	52	52	51	50	50

The embedding algorithm of Almohammad's method is shown as follows

Begin

1. Partition a cover image into non-overlapping 16x16-pixel blocks. Figure 2.15 shows a 16x16-pixel block.

122	76	33	57	91	80	67	80	95	77	62	91	117	75	42	98
116	77	40	61	90	78	65	78	92	75	61	87	110	71	41	91
93	76	61	73	87	73	59	68	81	67	55	68	80	54	35	62
69	81	95	92	83	64	50	58	69	62	54	52	50	39	30	31
74	104	131	108	75	54	44	53	66	70	73	71	65	43	24	24
81	121	157	120	71	49	42	53	71	85	95	95	87	58	33	33
56	100	141	114	73	49	41	63	89	96	97	88	79	85	90	68
31	71	111	104	85	61	47	71	100	107	104	85	74	112	144	103
31	48	74	102	121	100	73	64	71	111	146	123	99	123	145	107
38	35	46	100	147	130	95	57	45	114	183	164	129	128	130	103
36	42	54	82	105	91	73	61	67	115	165	164	149	138	126	107
33	55	77	66	50	42	45	73	106	122	135	147	153	137	118	112
25	60	90	64	33	28	44	98	149	144	124	117	111	97	89	109
25	65	102	73	36	30	49	114	172	157	120	90	68	60	67	106
46	80	112	94	66	48	44	71	103	118	120	92	67	76	95	114
70	95	120	117	101	69	41	25	30	74	116	103	85	104	125	118

Figure 2.15 A 16x16-pixel block

2. Apply DCT transformation to each block, starting from left to right and top to bottom. The DCT coefficients are now obtained (Figure 2.16).

-716	-121	-6.2	99.5	-108	-136	-20.2	-0.27	8.31	-15.3	-10.5	-4.69	13.2	4.08	1.47	4.48
-112	185	15.3	-12.6	17	102	14.3	23.2	58.2	-32.1	16	-9.51	-4.88	-3.43	-0.457	-0.53
-77.2	63.6	10.3	20.4	100	27.5	5.81	-2.96	66.3	-48.4	24.3	-11.2	5.66	-9.71	2.56	-1.65
81.2	-181	-66.5	-12.8	-4.56	90.9	74.3	39.1	43.9	-8.02	6.76	-17.2	-6.75	-4.89	-0.998	-0.41
66.7	-33.	53.1	-75.1	-21.7	10.7	62.1	33	-12.5	28.9	-14.8	-5.22	-7.26	1.23	-0.737	0.073
-1.64	-7.82	-25.3	68.5	96.6	-44.8	-92.5	-21.4	23	-40.2	9.37	11.8	8.89	-2.45	1.68	-1.17
29.3	11	61.2	27.8	-82.8	46.1	32.9	-14.5	-2.43	5.25	10.4	-9.73	-1.84	1.26	-0.579	0.505
1.74	30.3	-60.2	3.57	2.76	-13.3	24.1	-5.72	1.92	1.21	3.59	-5.58	2.76	-1.35	0.581	-0.409
-18.1	-13.8	32.2	-15.8	2.85	-1.84	-21.1	7.46	-8.44	3.62	-6.41	5.89	-1.79	1.84	-0.239	0.176
-18.8	-4.76	-8.6	-12.9	22.8	-15.9	-16.3	5.76	-3.87	2.23	-6.29	4.83	0.123	0.073	0.0835	0.117
-0.971	11.3	-10.9	-8.94	-12.3	1.97	14.7	1.43	-2.78	5.22	-0.592	-1.6	-1.22	-0.22	0.32	0.065
-1.12	1.43	6.61	7.24	-8.08	3.74	0.393	-3.51	-1.36	-0.196	2.16	0.033	0.816	0.669	-0.29	0.093
-1.06	4.61	-3.56	4.8	6.11	-3.47	-7.01	-1.29	1.88	-3.71	1.08	0.934	0.468	-0.34	0.216	0.441
2.21	-2.35	3.5	-2.06	-2.87	1.47	4.54	0.468	-1.6	1.1	-0.65	-0.46	-0.57	-0.15	-0.075	-0.11
1.3	-1.8	-2.3	0.856	1.64	1.34	-0.657	0.236	1.94	-1.07	0.744	0.226	0.232	-0.16	-0.417	0.119
-2.62	-1.5	1.28	-0.73	0.22	0.71	0.714	0.127	-0.028	0.502	-0.68	-0.53	-0.23	-0.24	-0.21	0.342

Figure 2.16 A 16x16-pixel DCT coefficient

3. Quantize each DCT coefficient block by dividing each block with the 16x16 quantization table (Table 2.3) and then rounding off. The quantized DCT coefficients are now obtained (Figure 2.17).

-45	-15	-1	17	-18	-136	-20	0	8	-15	-11	-5	13	4	1	4
-16	26	3	-2	17	102	14	23	58	-32	16	-10	-5	-3	0	0
-11	11	2	20	100	27	6	-3	66	-48	24	-11	6	-10	0	0
14	-23	-67	-13	-5	91	74	39	44	-8	7	-17	-7	0	0	0
8	-33	53	-75	-22	11	62	33	-13	29	-15	-5	0	0	0	0
-2	-8	-25	68	97	-45	-92	-21	23	-40	9	0	0	0	0	0
29	11	61	28	-83	46	33	-14	-2	5	0	0	0	0	0	0
2	30	-60	4	3	-13	24	-6	2	0	0	0	0	0	0	0
-18	-14	32	-16	3	-2	-21	7	0	0	0	0	0	0	0	0
-19	-5	-9	-13	23	-16	-16	0	0	0	0	0	0	0	0	0
-1	11	-11	-9	-12	2	0	0	0	0	0	0	0	0	0	0
-1	1	7	7	-8	0	0	0	0	0	0	0	0	0	0	0
-1	5	-4	5	0	0	0	0	0	0	0	0	0	0	0	0
2	-2	3	0	0	0	0	0	0	0	0	0	0	0	0	0
1	-2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
-3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 2.17 A 16x16-pixel quantized DCT coefficient

4. Embed the secret messages into each quantized DCT coefficient block. The secret messages are embedded into the two least significant bits of each quantized coefficient n the middle part of 16x16 quantization table that has value '1' (in the middle frequency part) (Figure 2.18).

-45	-15	-1	17	-18	-135	-20	2	9	-15	-9	-7	15	5	2	6
-16	26	3	-2	17	101	14	23	56	-31	18	-9	-8	-3	2	0
-11	11	2	21	103	25	6	-1	67	-47	27	-11	4	-11	0	0
14	-23	-66	-13	-5	89	75	36	47	-7	6	-20	-5	0	0	0
8	-35	54	-74	-24	9	62	35	-13	29	-14	-5	0	0	0	0
-1	-7	-26	71	96	-47	-90	-24	21	-39	10	0	0	0	0	0
28	9	61	30	-84	45	33	-14	-4	5	0	0	0	0	0	0
1	30	-60	5	1	-14	24	-7	1	0	0	0	0	0	0	0
-18	-16	33	-15	2	-4	-23	5	0	0	0	0	0	0	0	0
-18	-8	-11	-15	22	-16	-15	0	0	0	0	0	0	0	0	0
-3	10	-12	-11	-11	2	0	0	0	0	0	0	0	0	0	0
-4	1	5	6	-8	0	0	0	0	0	0	0	0	0	0	0
-3	5	-2	4	0	0	0	0	0	0	0	0	0	0	0	0
1	-3	2	0	0	0	0	0	0	0	0	0	0	0	0	0
0	-3	0	0	0	0	0	0	0	0	0	0	0	0	0	0
-3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 2.18 A 16x16-pixel quantized DCT coefficient after embedded

5. Employ JPEG encoding algorithms.

End

Figure 2.19 shows the overview of Almohammad's embedding process.

The extracting algorithms of Almohammad's method is shown as follows

Begin

1. Decode JPEG file by employing JPEG decoding algorithms.
2. Extract the secret messages from each block by collecting the information from the two least significant bit of each embedded quantized coefficient starting from left to right and top to bottom (Same order as in the embedding process).
3. De-quantize each block by multiplying that with the 16x16 quantization table. The DCT coefficients are obtained (Figure 2.20).

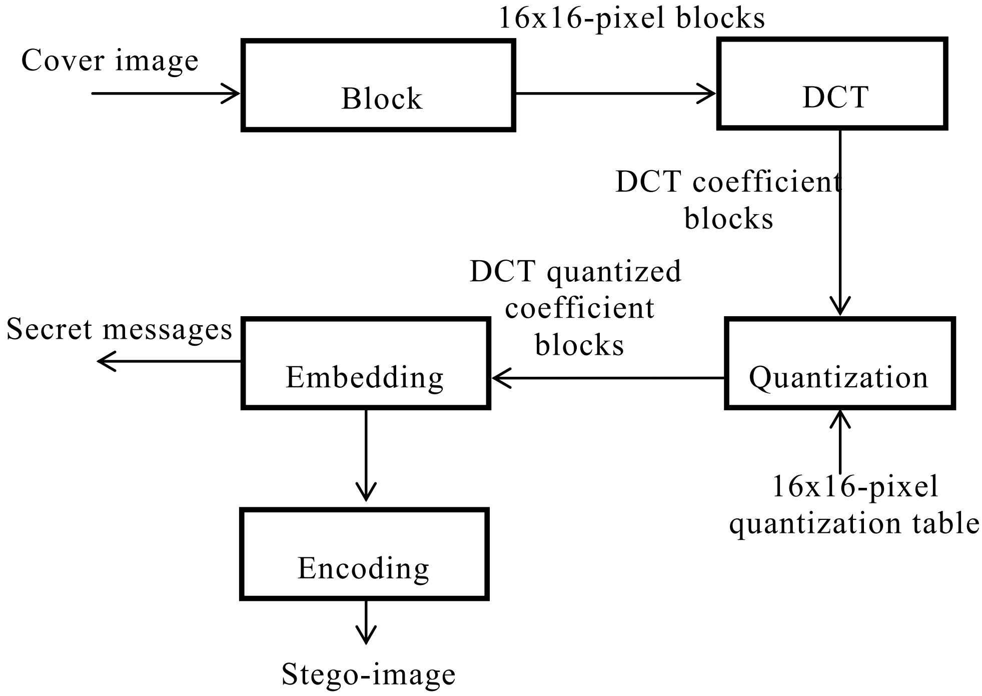


Figure 2.19 The block diagram of Almohammad's embedding process

-720	-120	-7	102	-108	-135	-20	2	9	-15	-9	-7	15	5	2	6
-112	182	18	-12	17	101	14	23	56	-31	18	-9	-8	-3	2	0
-77	66	12	21	103	25	6	-1	67	-47	27	-11	4	-11	0	0
84	-184	-66	-13	-5	89	75	36	47	-7	6	-20	-5	0	0	0
64	-35	54	-74	-24	9	62	35	-13	29	-14	-5	0	0	0	0
-1	-7	-26	71	96	-47	-90	-24	21	-39	10	0	0	0	0	0
28	9	61	30	-84	45	33	-14	-4	5	0	0	0	0	0	0
1	30	-60	5	1	-14	24	-7	1	0	0	0	0	0	0	0
-18	-16	33	-15	2	-4	-23	5	0	0	0	0	0	0	0	0
-18	-8	-11	-15	22	-16	-15	0	0	0	0	0	0	0	0	0
-3	10	-12	-11	-11	2	0	0	0	0	0	0	0	0	0	0
-4	1	5	6	-8	0	0	0	0	0	0	0	0	0	0	0
-3	5	-2	4	0	0	0	0	0	0	0	0	0	0	0	0
1	-3	2	0	0	0	0	0	0	0	0	0	0	0	0	0
0	-3	0	0	0	0	0	0	0	0	0	0	0	0	0	0
-3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 2.20 A 16x16-pixel DCT coefficient

4. Apply the inverse DCT (IDCT) to each 16x16-pixel DCT coefficient block. The 16x16-piexel blocks are now obtained (Figure 2.21)

119	75	34	57	91	74	64	79	94	78	63	91	120	74	41	100
122	78	40	60	91	78	66	76	90	77	65	86	107	71	44	91
95	74	60	74	87	72	60	67	79	68	54	64	78	56	35	61
68	79	95	94	81	62	52	60	72	62	49	50	55	40	26	35
76	105	131	107	73	54	45	50	66	71	72	74	64	41	25	23
81	123	154	117	73	51	41	49	70	85	96	97	81	57	39	25
52	102	142	113	72	47	42	65	90	97	95	85	79	87	88	66
32	70	109	104	86	58	49	73	95	108	108	83	78	114	136	108
33	45	73	102	124	101	70	62	68	110	148	123	98	124	142	107
36	32	52	97	143	131	90	59	52	111	178	165	128	130	133	100
37	39	55	79	105	95	73	62	66	112	167	169	149	137	127	106
33	55	74	66	54	40	44	75	104	122	138	147	152	135	117	116
24	64	91	65	33	22	44	104	152	146	122	112	115	95	85	109
26	68	99	73	39	30	53	113	165	157	123	88	70	57	67	104
48	79	110	94	65	48	44	70	106	120	121	89	64	73	93	113
72	92	121	118	98	72	36	22	35	69	114	102	83	108	122	115

Figure 2.21 A 16x16-pixel block

5. Merge all 16x16-pixel blocks and obtain the cover image.

End

The following block diagram (Figure 2.22) shows the overview of Almohammad's extracting process.

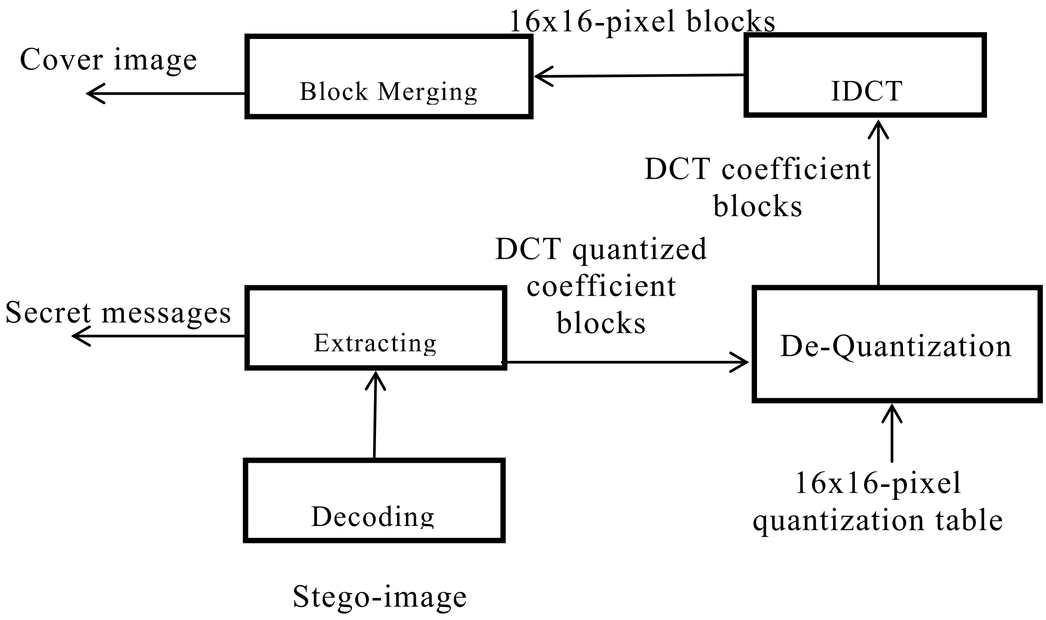


Figure 2.22 The block diagram of Almohammad's extracting process

2.4 Chan's method

In this work, Chan et al. [2] proposed the spatial domain steganography technique by using the substitution of the least significant bit (LSB) with the optimal pixel adjustment process (OPAP) technique applied with the stego-image. Chan conducted the experiment to compare the image quality (PSNR) of the simple LSB substitution technique compared with the simple LSB substitution with the OPAP. The results showed that the image quality of Chan's method improved significantly compared to the simple LSB substitution method with an additional low computational complexity. The example of the stego-image with the secret messages of size 512x256 pixels of simple LSB and Chan's method shown in figure 2.23 below.

(a)Simple LSB substitution method (b) Chan's method

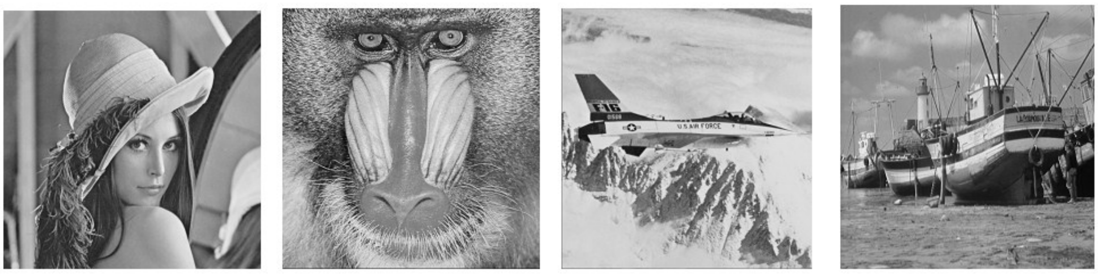
Figure 2.23 'Lena' stego-images

2.5 Nag's method

In this work, Nag et al. [5] proposed the novel steganography based on the frequency domain method by using the Discrete Wavelet Transform (DWT) and Huffman encoding. The difference of this method compared to the proposed method is, in this method, DWT is used to transform the cover image from spatial domain into the frequency domain instead of using DCT.

Also, before embedding, the secret messages will be encoded by Huffman encoding. Lastly, the secret messages will be embedded in the high frequency coefficients which, in the proposed method, the secret messages will be embedded in the middle frequency coefficients. The experimental results show that the image quality of the stego-images (PSNR) is better than some existing steganography based on DWT. The figure 2.24 below shows the stego-images of Nag's method compared with the original cover images.

(a) Original cover images



(b) Stego-images of Nag's method

Figure 2.24 Original cover images and Stego-images of Nag's method

2.6 Monro's optimal quantization table

Monro et al. [6] proposed the method for creating the optimal quantization table to be used in the JPEG compression process. The aim of this study is try to create the quantization table that will minimize the root mean square error (RMS) between the cover image and the reconstructed image at the desired compression ratio. Monro found the optimal quantization table by using simulate annealing. In addition, the following equation is the three parameters equation model proposed by Monro to find the 8x8 optimal quantization table.

$$Q_{(x,y)} = A + Dz^F \qquad (1)$$

where $A = -1.29 + 2.04C_R$

$D = -2.37 + 0.315C_R$

$F = 4.33 - 0.218C_R + 0.00384C_R^2$

and C_R = desired compression ratio

The experiment shows that the RMS obtained from using Monro's 8x8 quantization table is less than the RMS obtained from using the standard quantization table.

For the conclusion, in this section, there are 5 frequency domain steganography methods including DCT and DWT. Furthermore, there is one spatial domain method and a method to create the quantization table.

CHAPTER III

THEORETICAL FOUNDATION

This thesis is about image steganography which is related to the image processing. In addition, image steganography can be broadly classified into 2 types. The first one is spatial domain and the second one is frequency domain which will be used in this work. Moreover, this image processing work is based on the JPEG which is the image file format that work with the DCT. Therefore, DCT and JPEG will be reviewed in this chapter followed by the interpolation technique related to build the quantization table.

3.1 Discrete Cosine Transform

3.1.1 Forward Discrete Cosine Transform

Discrete Cosine Transform or DCT is a technique that is used to transform the signal or image from the spatial domain to the frequency domain. It is widely used in many applications including JPEG standard [7]. It works by separating the image into the parts of different frequencies in term of sum of the cosine function. In this work, image is used to work with the DCT therefore 2 dimensional DCT will be used. The following equation is the 2-d DCT equation of a M-by-N matrix C

$$D(i,j) = a(i)a(j)\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} C(x,y)\cos\left[\frac{\pi(2x+1)i}{2M}\right]\cos\left[\frac{\pi(2y+1)i}{2N}\right] \quad (1)$$

where $a(i) = \frac{1}{\sqrt{M}}$, i = 0,

$a(i) = \sqrt{\frac{2}{M}}$, $1 \le i \le M-1$,

and $a(j) = \frac{1}{\sqrt{N}}$, j = 0,

$a(j) = \sqrt{\frac{2}{N}}$, $1 \le j \le N-1$

In addition, if the matrix C is N-by-N matrix, the equation (1) can be written in the form of the DCT matrix equation. The following equation is the DCT matrix equation.

$$D(i,j) = S(i,j)C(i,j)S'(i,j) \qquad (2)$$

Where $S(i,j) = \frac{1}{\sqrt{N}}$, i = 0

$$S(i,j) = \sqrt{\frac{2}{N}} cos\left[\frac{\pi(2j+1)i}{2M}\right] ,1 \leq i \leq N-1$$

$S'(i,j)$ is the inverse of matrix S

The computation using the DCT matrix equation is sometimes faster than using the DCT equation especially if large number of small DCTs is considered to compute because the matrix S needs to be determined only once. Figure 3.1 and 3.2 show the matrix S of size 8x8 and invert of matrix S size 8x8 respectively.

0.3536	0.3536	0.3536	0.3536	0.3536	0.3536	0.3536	0.3536
0.4904	0.4157	0.2778	0.0975	-0.0975	-0.2778	-0.4157	-0.4904
0.4619	0.1913	-0.1913	-0.4619	-0.4619	-0.1913	0.1913	0.4619
0.4157	-0.0975	-0.4904	-0.2778	0.2778	0.4904	0.0975	-0.4157
0.3536	-0.3536	-0.3536	0.3536	0.3536	-0.3536	-0.3536	0.3536
0.2778	-0.4904	0.0975	0.4157	-0.4157	-0.0975	0.4904	-0.2778
0.1913	-0.4619	0.4619	-0.1913	-0.1913	0.4619	-0.4619	0.1913
0.0975	-0.2778	0.4157	-0.4904	0.4904	-0.4157	0.2778	-0.0975

Figure 3.1 A 8x8 Matrix

0.3536	0.4904	0.4619	0.4157	0.3536	0.2778	0.1913	0.0975
0.3536	0.4157	0.1913	-0.0975	-0.3536	-0.4904	-0.4619	-0.2778
0.3536	0.2778	-0.1913	-0.4904	-0.3536	0.0975	0.4619	0.4157
0.3536	0.0975	-0.4619	-0.2778	0.3536	0.4157	-0.1913	-0.4904
0.3536	-0.0975	-0.4619	0.2778	0.3536	-0.4157	-0.1913	0.4904
0.3536	-0.2778	-0.1913	0.4904	-0.3536	-0.0975	0.4619	-0.4157
0.3536	-0.4157	0.1913	0.0975	-0.3536	0.4904	-0.4619	0.2778
0.3536	-0.4904	0.4619	-0.4157	0.3536	-0.2778	0.1913	-0.0975

Figure 3.2 An Inverse 8x8 Matrix s

3.1.2 Inverse discrete cosine transform

The equation of the inverse discrete cosine transform (IDCT) is the same as the DCT matrix equation except that the inverse DCT matrix S' is first multiplied by matrix C and then multiplied by the DCT matrix S. The following equation is the IDCT matrix equation.

$$I(i,j) = S'(i,j)C(i,j)S(i,j) \qquad (3)$$

where $S(i,j) = \frac{1}{\sqrt{N}}$, i = 0

$$S(i,j) = \sqrt{\frac{2}{N}} cos\left[\frac{\pi(2j+1)i}{2M}\right] \; ,1 \le i \le N-1$$

$S'(i,j)$ is the inverse of matrix S

3.2 JPEG

JPEG or Joint Photographic Expert Group is the famous image format file that has been widely used over the internet. The reason that it is very popular is because its efficiency for compressing the size of image while still gets the good quality image. In this work, JPEG compression is used to compress the cover image so that the size of image is reduced. In addition, using JPEG as the image file format for our stego-image will reduce the chance to be suspected. The JPEG compression process is a lossy image compression in which some parts of the image are actually lost.

The overview of the JPEG process is that first the image is divided into a blocks of pixels. After that, The DCT is then applied to each block of pixels. Next, each block will be quantized by using the quantization table and coded [8]. Example of the JPEG encoding process with the 8x8 block of pixels is shown below.

1. A cover image is divided into many 8x8-pixel blocks (Figure 3.3).

162	162	162	161	162	157	163	161
162	162	162	161	162	157	163	161
162	162	162	161	162	157	163	161
162	162	162	161	162	157	163	161
162	162	162	161	162	157	163	161
164	164	158	155	161	159	159	160
160	160	163	158	160	162	159	156
159	159	155	157	158	159	156	157

Figure 3.3 An example of 8x8-pixel block

2. Since DCT is designed to work with the pixel value from -128 to 127, the 8x8-pixel block will be leveled shift by subtracting with value 128 (Figure 3.4).

34	34	34	33	34	29	35	33
34	34	34	33	34	29	35	33
34	34	34	33	34	29	35	33
34	34	34	33	34	29	35	33
34	34	34	33	34	29	35	33
36	36	30	27	33	31	31	32
32	32	35	30	32	34	31	28
31	31	27	29	30	31	28	29

Figure 3.4 Leveled shift 8x8-pixel block

3. Apply the DCT to each block by using the equation (2) to get the DCT coefficients (Figure3.5).

259.5	4.7682	3.2403	-0.1991	0.2500	-0.5538	-4.5893	5.6384
7.9473	-0.7878	0.5547	-4.9323	1.9602	2.9784	-3.7971	3.3221
-5.0349	-0.2973	-1.5517	1.7249	-0.6764	-0.4524	1.8498	-2.2002
2.2618	1.1390	1.7038	0.9242	-0.7606	-1.3686	0.2142	1.14223
-0.9999	-1.1497	-0.3430	-1.3596	1.7500	1.1188	-1.4814	-0.6728
1.2106	0.4561	-1.802	-0.1060	-2.0115	0.7096	1.6866	0.7551
-1.7028	0.2649	3.0998	1.6354	1.6332	-2.2545	-1.1982	-0.9011
1.3259	-0.4152	-2.3825	-1.5945	-0.8845	1.9693	0.5531	0.6539

Figure 3.5 DCT coefficients

4. Quantization by dividing each DCT coefficient by the quantization table and then round to the nearest integer. In addition, this is the step where the lossy compression occurred. Also the compression ratio can be adjusted by multiplying or dividing the standard quantization table by a number. However, if the compression ratio is increased, the quality of the image will be degraded. In the other hand, if the compression ratio is decreased, the quality of the image will be upgraded. After the DCT coefficient is quantized by the standard quantization table, the quantized DCT coefficients are gotten (Figure 3.6). From the figure 3.6, it is notice that most of the coefficients are discarded. The remain coefficients are on the upper left corner because the upper left corner is the low frequency part mainly for the natural image. Also, the energy is concentrated on the lower frequency part. In addition, for the Human Vision System (HVS), HVS is more sensitive to lower frequency and insensitive to high frequency. This is why most of the coefficients in the high frequency are discarded.

16	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Figure 3.6 Quantized DCT coefficients

5. Employ the Jpeg entropy coding in the zigzag pattern, The JPEG file is obtained.

Example of JPEG decoding process is shown below.

1. Decode the JPEG file
2. De-quantize the quantized DCT coefficient by multiplying with the standard quantization table. The DCT coefficient is obtained (Figure 3.7)

256	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Figure 3.7 DCT coefficient

3. Apply the inverse DCT (IDCT) to the DCT coefficient and round to the nearest integer, the 8x8-block is obtained (Figure 3.8).

34	34	34	34	34	34	34	34
34	34	34	34	34	34	34	34
33	33	33	33	33	33	33	33
32	32	32	32	32	32	32	32
32	32	32	32	32	32	32	32
31	31	31	31	31	31	31	31
30	30	30	30	30	30	30	30
30	30	30	30	30	30	30	30

Figure 3.8 A 8x8 pixels block

4. Merge all the 8x8 blocks, the decompressed image is obtained.

3.3 Interpolation

From the JPEG process, it can be seen that the quantization table is needed in order to do the quantization step. However, there is only standard quantization table for the block size of 8x8. Therefore, in order to create the quantization table for the other block size, interpolation technique will be used.

Interpolation is a method for estimating the value of new data points that lie between the known data points by using mathematical equation [9]. There are various types of interpolation such as Newton’s interpolation or Spline Interpolation. For the spline interpolation, the polynomial equations are determined for each interval between two known data points. After that, the values between the known data points can be estimated using those equations. Moreover, there are three types of spline interpolation grouped by the order of polynomial used.

1. Linear spline: using first-order polynomials equation for estimating the value of new data points that lie between the known points
2. Quadric spline : using second-order polynomials equation for estimating the value of new data points that lie between the known points

3. Cubic spline : using third-order polynomials equation for estimating the value of new data points that lie between the known points

In this work, cubic spline interpolation will be used to build the quantization table. The form of the third-order polynomial equation is

$$f(x) = ax^3 + bx^2 + cx + d \tag{4}$$

Also for the known n points, there are n-1 intervals. Therefore, there are n-1 third-order polynomials equations. In addition, for each third-order polynomials equation, there are four unknown coefficients which are a, b, c and d in the equation (4). Thus in order to solve n-1 third-order polynomials equations, the 4n-4 coefficients needed to be determined by using the 4 properties of the cubic spline

1. The third-order polynomial equations of each interval i will connect with the starting point of the interval i+1. It can be concluded that

$$f_i(x_{i+1}) = f_{i+1}(x_{i+1}) \tag{5}$$

2. The function f(x) will interpolate all the point. It can be conclude that

$$f_i(x_i) = y_i \tag{6}$$

3. Slope or f'(x) will be continuous for each connecting point between each interval from i=1 to i = n-2. It can be concluded that

$$f_i{}'(x_{i+1}) = f'_{i+1}(x_{i+1}) \tag{7}$$

4. $f''(x)$ will be equal to zero at the first and the last point and will be continuous for each connecting point between each interval from i=1 to i= n-2. It can be concluded that

$$f_1''(x_1) = 0 \text{ and } f_{n-1}''(x_{n-1}) = 0 \quad (8)$$

$$f_i''(x_{i+1}) = f_{i+1}''(x_{i+1}) \quad (9)$$

From 4 properties and the equation (5) – (9), the 4n-4 unknown coefficients can be determined.

3.4 Peak Signal to Noise Ratio (PSNR)

Peak Signal to Noise Ratio (PSNR) is a common method for measuring the quality of compressed image. In this work, it will be used as a quality measurement between the cover image and the stego-image. The higher the PSNR means the better of the stego-image's quality. The following equation is the PSNR equation for the MxN-gray level image.

$$PSNR = 10\log_{10}\frac{255^2}{MSE} \quad (10)$$

Where MSE or Mean Square error can be found from the following equation

$$MSE = \frac{1}{M}\frac{1}{N}\sum_{i=1}^{M}\sum_{j=1}^{N}[C(i,j) - S(i,j)]^2 \quad (11)$$

Where $C(i,j)$ is the pixel value of the cover image

$S(i,j)$is the pixel value of the stego-image.

CHAPTER IV

PROPOSED METHOD

In this work, a larger block size with the interpolated quantization table will be used to prove that the computation time and the size of the stego-image should decrease while capacity and quality should increase. However, there is also a limitation in computer memory space required more for computation during the process when a larger block size is used. Therefore, blocks of size 32x32 are chosen to create JPEG image steganography. By using 32x32-pixel blocks, a corresponding 32x32-pixel quantization table is needed in the quantization. However, there is no standard quantization table for this size. Therefore, the 32x32-pixel quantization table will be created by using interpolation technique.

4.1 Interpolated quantization table

In order to build the 32x32-pixel blocks, the standard 8x8-pixel quantization table is initially selected to be the based quantization table (Table. 2.1). Next, cubic spline interpolation technique is applied to build the 16x16-pixel interpolated quantization table by estimating the unknown data between two adjacent pixels.

Again, by using our 16x16-pixel interpolated quantization table as the based quantization table, 32x32-pixel interpolated quantization table is obtained

Finally, using Chang's technique, the 32x32-pixel table is modified such that the values in the middle part of proposed table will be changed to 1.

The following steps are to build the 32x32 interpolated quantization table.

1. The standard quantization table is used as a based quantization table in which the data in the standard quantization table is considered as the known data points.

2. The known data points from the standard quantization table is then repositioned to the 16x16 based quantization table before interpolation (value ‘0’ in this table is stand for the unknown data points that needed to be determined by cubic spline interpolation).

Table 4.1 16x16 based quantization table before interpolation

16	0	11	0	10	0	16	0	24	0	40	0	51	0	0	61
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	12	0	14	0	19	0	26	0	58	0	60	0	0	55
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	13	0	16	0	24	0	40	0	57	0	69	0	0	56
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	17	0	22	0	29	0	51	0	87	0	80	0	0	62
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	0	22	0	37	0	56	0	68	0	109	0	103	0	0	77
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
24	0	35	0	55	0	64	0	81	0	104	0	113	0	0	92
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
49	0	64	0	78	0	87	0	103	0	121	0	120	0	0	101
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
72	0	92	0	95	0	98	0	112	0	100	0	103	0	0	99

3. After the 16x16 based quantization table before interpolation is obtained. The unknown data points (value ‘0’ in table 4.1) between the known data points are now ready to be determined and replaced by using cubic spline interpolation technique (Table 4.2)

Table 4.2 16x16 based interpolated quantization table

16	13	11	10	10	13	16	19	24	32	40	46	51	54	57	61
12	12	12	12	13	15	16	16	22	39	56	60	55	52	52	57
12	12	12	12	14	17	19	20	26	42	58	63	60	57	54	55
13	12	12	13	15	18	22	26	33	44	55	62	65	64	60	55
14	13	13	14	16	19	24	31	40	49	57	65	69	69	65	56
14	14	15	16	18	20	24	33	45	59	69	74	73	70	65	58
14	15	17	19	22	25	29	37	51	71	87	89	80	72	66	62
16	17	19	23	29	36	43	48	59	82	102	103	92	81	73	69
18	19	22	28	37	48	56	59	68	89	109	112	103	93	84	77
20	21	27	36	45	54	61	66	74	90	106	112	109	103	95	85
24	26	35	46	55	60	64	71	81	93	104	111	113	110	103	92
35	39	48	58	66	70	74	81	92	102	111	117	117	114	108	98
49	56	64	72	78	82	87	94	103	113	121	123	120	115	109	101
59	67	75	80	85	89	94	101	109	117	123	123	119	114	108	102
67	78	84	88	91	94	99	106	113	116	117	116	113	110	106	101
72	85	92	95	95	95	98	107	112	107	100	100	103	105	104	99

4. After the 16x16 based quantization table before interpolation is obtained. The unknown data points (value '0' in table 4.1) between the known data points are now ready to be determined and replaced by using cubic spline interpolation technique (Table 4.2).

5. The known data points from the standard quantization table is then repositioned to the 32x32 based quantization table before interpolation (value '0' in this table is stand for the unknown data points that needed to be determined by cubic spline interpolation) (Table 4.3).

6. Determine the unknown data points from the table 4.3 (value '0') by using cubic spline interpolation technique (Table 4.4).

7. Use Chang's technique to modify the middle frequency that will be embedded the secret messages into value '1' (Table 4.5)

Table 4. 3 The 32x32 based quantization table before interpolation

16	0	13	0	11	0	10	0	10	0	13	0	16	0	19	0	24	0	32	0	40	0	46	0	51	0	54	0	57	0	0	61
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	12	0	12	0	12	0	13	0	15	0	16	0	16	0	22	0	39	0	56	0	60	0	55	0	52	0	52	0	0	57
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	12	0	12	0	12	0	14	0	17	0	19	0	20	0	26	0	42	0	58	0	63	0	60	0	57	0	54	0	0	55
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	12	0	12	0	13	0	15	0	18	0	22	0	26	0	33	0	44	0	55	0	62	0	65	0	64	0	60	0	0	55
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	13	0	13	0	14	0	16	0	19	0	24	0	31	0	40	0	49	0	57	0	65	0	69	0	69	0	65	0	0	56
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	14	0	15	0	16	0	18	0	20	0	24	0	33	0	45	0	59	0	69	0	74	0	73	0	70	0	65	0	0	58
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	15	0	17	0	19	0	22	0	25	0	29	0	37	0	51	0	71	0	87	0	89	0	80	0	72	0	66	0	0	62
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	0	17	0	19	0	23	0	29	0	36	0	43	0	48	0	59	0	82	0	102	0	103	0	92	0	81	0	73	0	0	69
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	0	19	0	22	0	28	0	37	0	48	0	56	0	59	0	68	0	89	0	109	0	112	0	103	0	93	0	84	0	0	77
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	0	21	0	27	0	36	0	45	0	54	0	61	0	66	0	74	0	90	0	106	0	112	0	109	0	103	0	95	0	0	85
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
24	0	26	0	35	0	46	0	55	0	60	0	64	0	71	0	81	0	93	0	104	0	111	0	113	0	110	0	103	0	0	92
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
35	0	39	0	48	0	58	0	66	0	70	0	74	0	81	0	92	0	102	0	111	0	117	0	117	0	114	0	108	0	0	98
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
49	0	56	0	64	0	72	0	78	0	82	0	87	0	94	0	103	0	113	0	121	0	123	0	120	0	115	0	109	0	0	101
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
59	0	67	0	75	0	80	0	85	0	89	0	94	0	101	0	109	0	117	0	123	0	123	0	119	0	114	0	108	0	0	102
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
67	0	78	0	84	0	88	0	91	0	94	0	99	0	106	0	113	0	116	0	117	0	116	0	113	0	110	0	106	0	0	101
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
72	0	85	0	92	0	95	0	95	0	95	0	98	0	107	0	112	0	107	0	100	0	100	0	103	0	105	0	104	0	0	99

Table 4.4 The 32x32 interpolated quantization table

16	14	13	12	11	10	10	10	10	11	13	15	16	17	19	21	24	28	32	36	40	43	46	49	51	53	54	55	57	58	60	61
13	13	12	12	12	12	12	12	12	13	14	15	15	16	16	18	22	28	36	44	50	54	55	54	53	52	52	52	53	55	57	59
12	12	12	12	12	12	12	12	13	14	15	16	16	16	16	18	22	30	39	49	56	60	60	58	55	53	52	52	52	53	55	57
12	12	12	12	12	12	12	13	14	15	16	17	17	17	17	19	23	31	41	51	58	62	62	60	57	55	54	53	52	53	54	56
12	12	12	12	12	12	12	13	14	16	17	18	19	19	20	22	26	33	42	51	58	62	63	62	60	58	57	55	54	54	54	55
12	12	12	12	12	12	12	13	14	16	18	19	21	22	23	25	29	35	43	50	56	60	62	63	63	62	61	59	57	56	55	55
13	12	12	12	12	12	13	14	15	16	18	20	22	24	26	29	33	38	44	50	55	59	62	64	65	65	64	62	60	58	57	55
14	13	12	12	12	13	13	14	15	17	19	21	23	26	29	32	37	41	46	50	55	59	63	66	67	68	67	65	63	61	58	55
14	13	13	13	13	13	14	15	16	17	19	21	24	27	31	35	40	45	49	53	57	61	65	68	69	69	69	68	65	63	60	56
14	14	14	14	14	14	15	16	17	18	19	21	24	28	32	37	43	48	54	58	62	66	69	70	71	71	70	68	65	63	60	57
14	14	14	14	15	15	16	17	18	19	20	21	24	28	33	39	45	52	59	65	69	72	74	74	73	72	70	68	65	63	61	58
14	14	14	15	16	17	17	18	20	21	22	23	25	29	34	40	48	56	65	72	78	81	81	79	76	73	70	68	65	63	61	60
14	14	15	16	17	18	19	20	22	24	25	27	29	32	37	43	51	61	71	80	87	90	89	85	80	76	72	69	66	64	63	62
15	15	16	17	18	19	21	23	25	27	30	33	35	38	42	47	55	65	77	88	95	98	96	92	86	80	76	72	69	67	66	65
16	16	17	18	19	21	23	26	29	32	36	40	43	45	48	52	59	70	82	94	102	105	103	98	92	86	81	77	73	71	70	69
17	17	18	19	20	22	25	29	33	38	42	47	50	52	54	57	64	74	86	98	107	110	109	104	98	92	87	82	78	76	74	73
18	18	19	20	22	25	28	32	37	43	48	53	56	58	59	62	68	77	89	100	109	113	112	108	103	98	93	88	84	81	79	77
19	19	20	22	24	28	32	36	41	46	52	56	59	61	63	66	71	80	90	100	108	112	113	110	107	102	98	94	90	87	84	81
20	20	21	23	27	31	36	41	45	50	54	58	61	64	66	69	74	81	90	99	106	110	112	111	109	106	103	99	95	92	89	85
21	21	23	26	30	36	41	46	50	53	57	59	62	65	68	72	77	84	91	98	104	108	111	112	111	109	107	104	99	96	93	89
24	24	26	30	35	41	46	51	55	58	60	62	64	67	71	76	81	87	93	99	104	108	111	113	113	112	110	107	103	100	96	92
29	29	31	36	41	46	52	56	60	63	64	66	68	71	75	80	86	92	97	102	106	110	113	115	115	114	112	110	106	103	99	95
35	36	39	43	48	53	58	62	66	68	70	72	74	77	81	86	92	97	102	107	111	115	117	118	117	116	114	111	108	105	102	98
42	45	48	52	56	61	65	69	72	74	76	78	81	84	88	93	98	103	108	113	116	119	121	120	119	117	115	112	109	106	103	100
49	52	56	60	64	68	72	75	78	80	82	84	87	90	94	98	103	108	113	118	121	123	123	122	120	118	115	112	109	107	104	101
54	58	62	66	70	73	77	79	82	84	86	88	91	94	98	102	106	111	116	120	123	124	124	122	120	118	115	112	109	106	104	102
59	63	67	71	75	78	80	83	85	87	89	91	94	97	101	105	109	113	117	121	123	124	123	121	119	117	114	111	108	106	104	102
63	68	73	76	80	82	84	86	88	90	92	94	97	100	104	108	111	114	117	120	121	121	120	119	117	115	112	110	107	105	103	102
67	73	78	81	84	86	88	90	91	92	94	96	99	102	106	110	113	115	116	117	117	117	116	115	113	112	110	108	106	104	103	101
69	76	81	84	87	89	91	92	93	94	95	97	100	103	107	111	113	114	114	113	113	112	112	111	110	109	108	107	105	104	102	100
71	78	84	87	89	92	93	94	94	95	95	97	99	103	107	111	113	113	111	109	107	106	106	107	107	107	107	106	105	103	102	100
72	79	85	89	92	94	95	95	95	95	95	96	98	102	107	111	112	110	107	103	100	99	100	101	103	104	105	105	104	103	101	99

Table 4.5 The 32x32 interpolated quantization table with Chang's technique

16	14	13	12	11	10	10	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
13	13	12	12	12	12	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	59
12	12	12	12	12	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	55	57
12	12	12	12	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	53	54	56
12	12	12	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	54	54	54	55
12	12	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	59	57	56	55	55
13	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	64	62	60	58	57	55
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	68	67	65	63	61	58	55
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	69	69	69	68	65	63	60	56
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	70	71	71	70	68	65	63	60	57
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	74	74	73	72	70	68	65	63	61	58
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	81	81	79	76	73	70	68	65	63	61	60
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	87	90	89	85	80	76	72	69	66	64	63	62
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	88	95	98	96	92	86	80	76	72	69	67	66	65
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	82	94	102	105	103	98	92	86	81	77	73	71	70	69
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	74	86	98	107	110	109	104	98	92	87	82	78	76	74	73
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	68	77	89	100	109	113	112	108	103	98	93	88	84	81	79	77
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	66	71	80	90	100	108	112	113	110	107	102	98	94	90	87	84	81
1	1	1	1	1	1	1	1	1	1	1	1	1	1	66	69	74	81	90	99	106	110	112	111	109	106	103	99	95	92	89	85
1	1	1	1	1	1	1	1	1	1	1	1	1	65	68	72	77	84	91	98	104	108	111	112	111	109	107	104	99	96	93	89
1	1	1	1	1	1	1	1	1	1	1	1	64	67	71	76	81	87	93	99	104	108	111	113	113	112	110	107	103	100	96	92
1	1	1	1	1	1	1	1	1	1	1	66	68	71	75	80	86	92	97	102	106	110	113	115	115	114	112	110	106	103	99	95
1	1	1	1	1	1	1	1	1	1	70	72	74	77	81	86	92	97	102	107	111	115	117	118	117	116	114	111	108	105	102	98
1	1	1	1	1	1	1	1	1	74	76	78	81	84	88	93	98	103	108	113	116	119	121	120	119	117	115	112	109	106	103	100
1	1	1	1	1	1	1	1	78	80	82	84	87	90	94	98	103	108	113	118	121	123	123	122	120	118	115	112	109	107	104	101
1	1	1	1	1	1	1	79	82	84	86	88	91	94	98	102	106	111	116	120	123	124	124	122	120	118	115	112	109	106	104	102
1	1	1	1	1	1	80	83	85	87	89	91	94	97	101	105	109	113	117	121	123	124	123	121	119	117	114	111	108	106	104	102
1	1	1	1	1	82	84	86	88	90	92	94	97	100	104	108	111	114	117	120	121	121	120	119	117	115	112	110	107	105	103	102
1	1	1	1	84	86	88	90	91	92	94	96	99	102	106	110	113	115	116	117	117	117	116	115	113	112	110	108	106	104	103	101
1	1	1	84	87	89	91	92	93	94	95	97	100	103	107	111	113	114	114	113	113	112	112	111	110	109	108	107	105	104	102	100
1	1	84	87	89	92	93	94	94	95	95	97	99	103	107	111	113	113	111	109	107	106	106	107	107	107	107	106	105	103	102	100
1	79	85	89	92	94	95	95	95	95	95	96	98	102	107	111	112	110	107	103	100	99	100	101	103	104	105	105	104	103	101	99

4.2 Embedding process

For the embedding process, in this work, a 512x512-pixel gray scale image is used as a cover image with the proposed 32x32 pixel block size and proposed interpolation quantization table. For the first step, a cover image is divided into the non-overlapping proposed block size which is 32x32-pixel blocks. After that 32x32-pixel blocks will be quantized by the proposed 32x32 interpolated quantization table with the Chang's technique and rounded to the nearest integer. Next, the secret messages will be embedded into the two least significant bits of the middle frequency coefficients (value '1' in the quantization table). Lastly, the DCT quantized coefficients will be coded then the stego-image is obtained.

The embedding algorithm is shown as follows.

Begin

1. Partition a cover image into non-overlapping 32x32-pixel blocks.
2. Apply DCT transformation to each block, starting from left to right and top to bottom.
3. Quantize each DCT coefficient block by dividing each block with the proposed 32x32 quantization table (Table 4.5) and then rounding to the nearest integer.
4. Embed the secret messages into each quantized DCT coefficient block. The secret messages are embedded into the two least significant bits of each quantized coefficient in the middle part of 32x32 quantization table that has value '1' (in the middle frequency part) starting from left to right and top to bottom.
5. Employ JPEG encoding algorithms. In this step, the stego-image is obtained

End

Figure 4.1 shows the overview of the embedding process

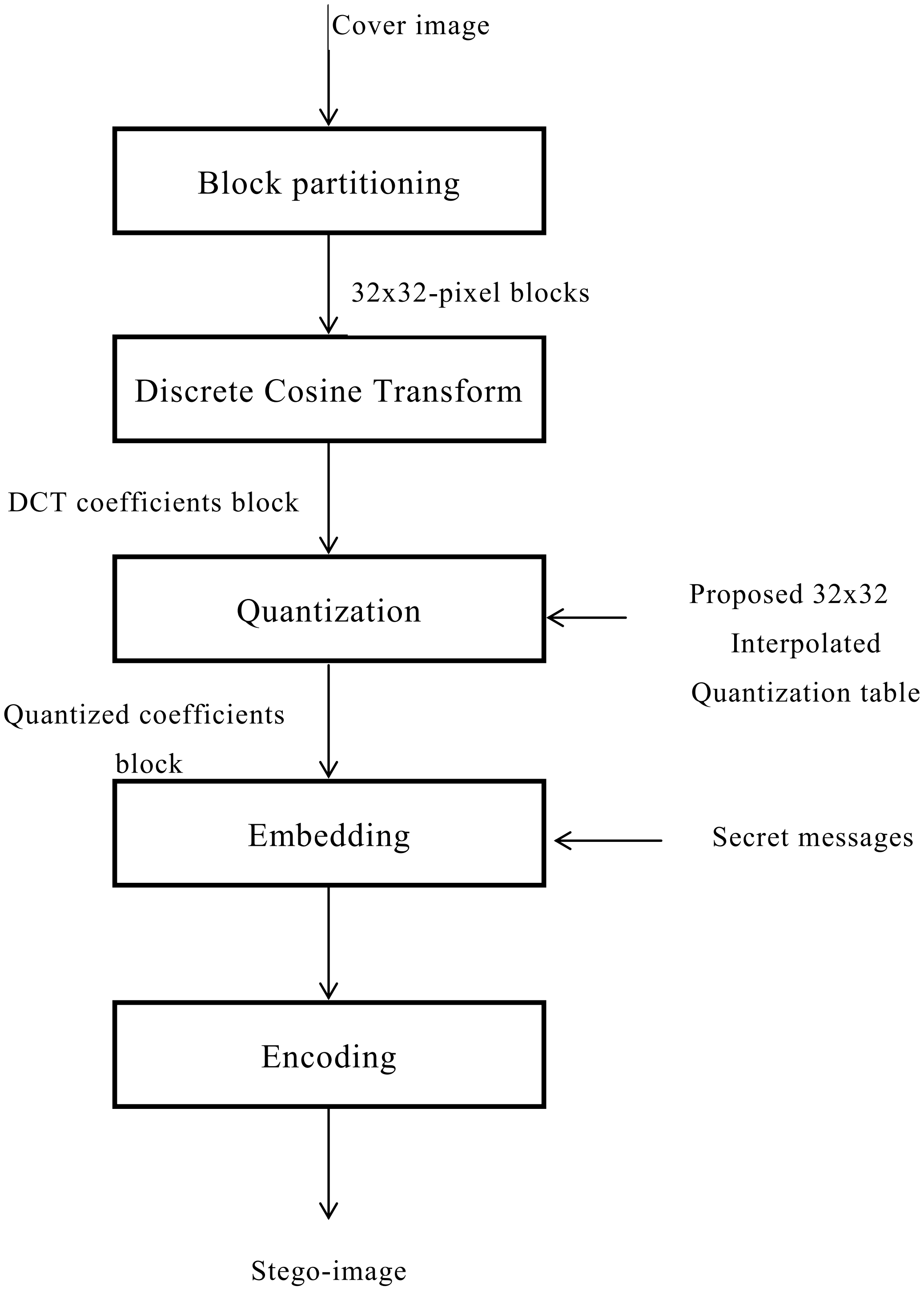


Figure 4.1 The block diagram of embedding process

4.3 The extracting process

For the extracting process, in this work, a stego-image is decoded and then extract the secret messages by collecting the information from the two least significant bits of each embedded coefficient (value '1' in the quantization table) starting from left to right and top to bottom. After that, the DCT quantized coefficients are now de-quantized by multiplying with the proposed 32x32-pixel interpolated quantization table (Table 4.5). Next, apply the IDCT to the each 32x32 pixel DCT coefficients blocks. Lastly, merge all 32x32-pixel blocks and obtain a cover-image.

The extracting algorithm is shown as follows

Begin

1. Decode JPEG file by employing JPEG decoding algorithms.

2. Extract the secret messages from each block by collecting the information from the two least significant bits of each embedded quantized coefficient starting from left to right and top to bottom. (Same order as in the embedding process).

3. De-quantize each block by multiplying that with the proposed 32x32-pixel quantization table (Table 4.5).

4. Apply the inverse DCT (IDCT) to each 32x32-pixel DCT coefficient block.

5. Merge all 32x32-pixel blocks and obtain the cover image.

End

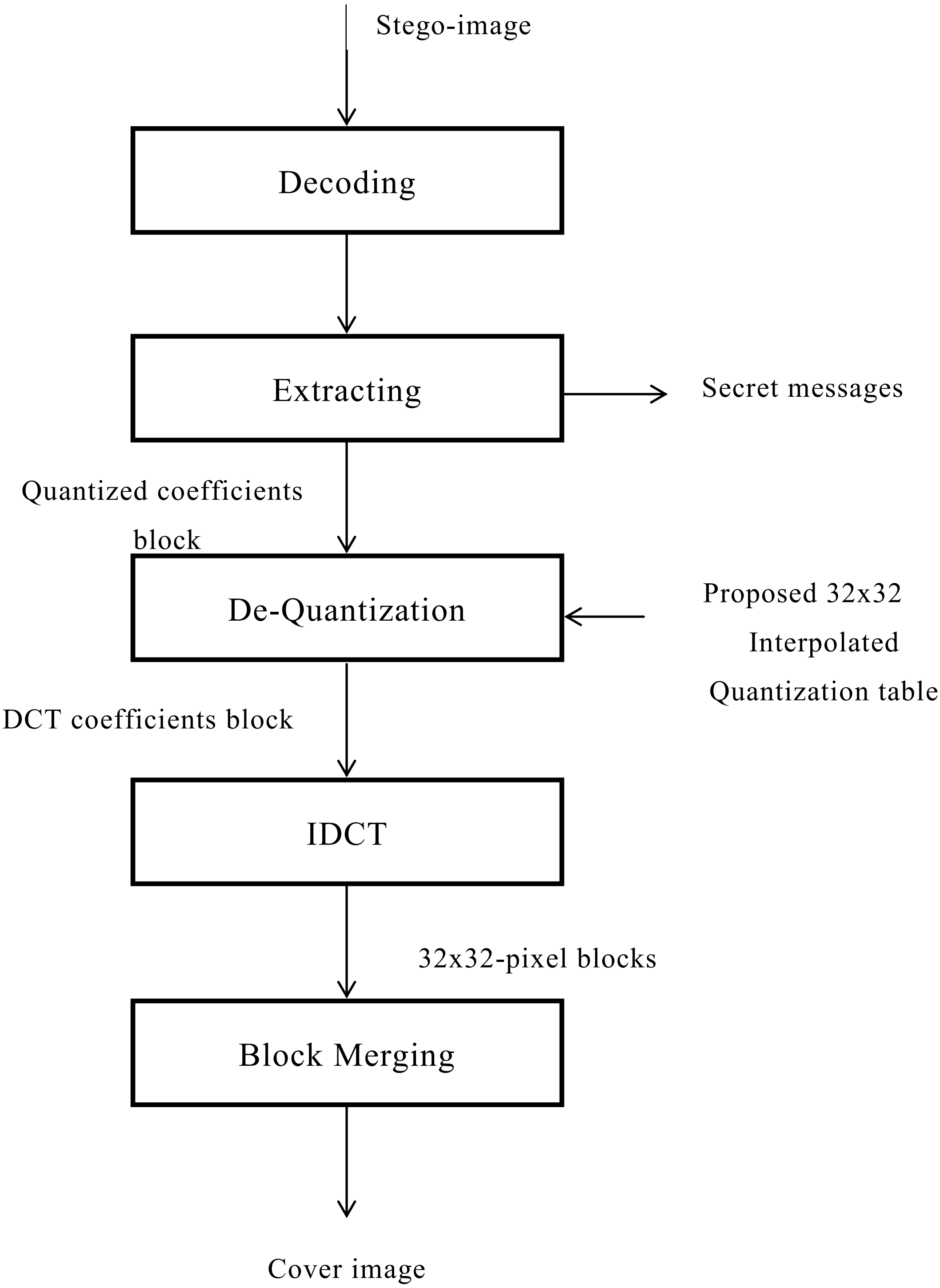


Figure 4.2 The block diagram of extracting process

CHAPTER V

EXPERIMENTAL RESULTS & DISCUSSION

In this chapter, the experiments are executed on MATLAB R2011B windows 7, CPU Core i5 with 4 GB of ram. Four 512x512-pixel gray level benchmark images which are Lena, Pepper, Baboon and House were used as cover images. Four criteria, consisting of 1) capacity of the secret messages that can be embedded, 2) quality of the stego-image, 3) size of the stego-image, and 4) the computation time consumed during the process, are used to measure the performance of our method compared to Chang's and Almohammad's methods.

In addition, in order measure the performance of the interpolated quantization table, the experiments are executed to measure the performance between the 16x16 interpolated quantization table (Table 4.2) and Almohammad's 16x16 quantization table (Table 2.3) . The compression ratio and quality of the stego-images will be measured by using both tables to compare the results.

Lastly, the experiments are executed to compare the performance of the proposed method by three different amounts of the capacity of the secret messages embedded.

5.1 Capacity

The capacity will be measured by the number of bits that can be embedded into a cover image.

- For Chang's method, in each 8x8-pixel block, there are 26 quantized coefficients that can be embedded and each quantized coefficient can be embedded two bits. For 512x512 image, there are (512x512)/(8x8) = 4096 blocks. Therefore, the capacity for Chang's method is 4096x52 = 212992 bits.

- For Almohammad's method, 16x16-pixel blocks are considered and, in one block, there are 121 quantized coefficients that can be embedded. Therefore, the capacity is (121x2) x [(512x512)/ (16x16)] = 247808 bits.
- For the proposed method, in one block, there are 500 embeddable quantized coefficients so the secret messages that can be embedded are 500x2 = 1000 bits. Therefore, with the total of 256 blocks obtained from (512x512)/ (32x32), the capacity is 1000x256 = 256000 bits.

In addition, the result of this work will perform much better than that of the other two existing methods if a larger cover image is used instead. Table 5.1 below shows the capacity in bits of three methods.

Table 5.1: The Capacity of The 512x512 Stego-images in Bits

Method	**512x512-pixel stego-image**
Chang's	212992
Almohammad's	247808
Proposed method	256000

5.2 Size

The sizes of four original images are 256 KB. The compression ratio is shown in Table 5.2

Table 5.2: The Compression Ratio of The Stego-images

Method	**Image**			
	Lena	***Pepper***	***Baboon***	***House***
Chang's	2.96:1	2.86:1	2.46:1	3.73:1
Almohammad's	2.85:1	2.74:1	2.35:1	3.69:1
Proposed method	3.02:1	2.92:1	2.47:1	3.92:1

The compression ratios of proposed method are higher than those of both Chang's method and Almohammad's method. This is caused by two reasons.

1. Among these three methods, entries in the middle frequency part of quantization table are defined to '1's and mostly the compression usually occurs from quantization in high frequency part. This means that the proposed method yields higher compression ratio than the others due to using larger block size causing larger high frequency part.
2. The entries' values in the high frequency part of our quantization table are higher than those of Almohammad's method. Basically, the higher the entries' values, the higher the compression ratio is. Hence, this makes the proposed method achieve high compression ratio.

5.3 Peak Signal to Noise Ratio (PSNR)

Peak Signal to Noise Ratio (PSNR) is a common method for measuring the quality of compressed image. Table.5.3 shows PSNR for three methods.

Table 5.3: PSNR of the stego-images (db)

Method	**Image**			
	Lena	***Pepper***	***Baboon***	***House***
Chang’s	38.8	36.2	39.5	43.2
Almohammad’s	40	36.6	42.4	46.4
Proposed method	39.6	36.4	42.3	46.6

From the result, even though, in our method, more secret messages can be embedded, our images’ qualities are still outperform those of Chang’s method and are very close to those of Almohammad’s method. Also, in figure. 5.1, 5.2 , 5.3 and 5.4 , the stego-images of proposed method are shown and compared with the original cover images and the stego-image of Chang and Almohammad’s method, it is very difficult to distinguish a difference with the HVS.

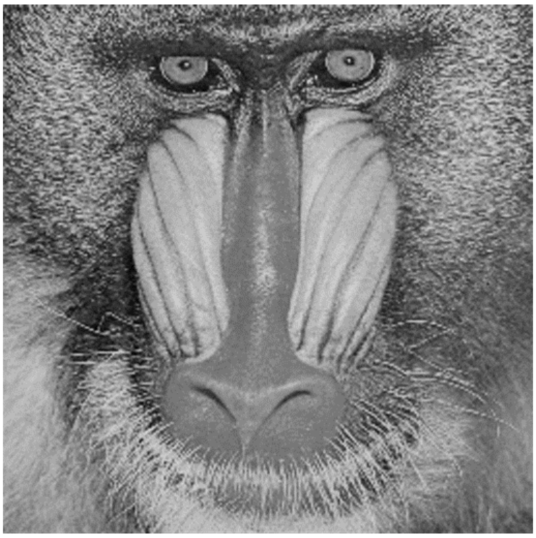
(a) Original ‘Lena’ image

(b) Our stego ‘Lena’ image

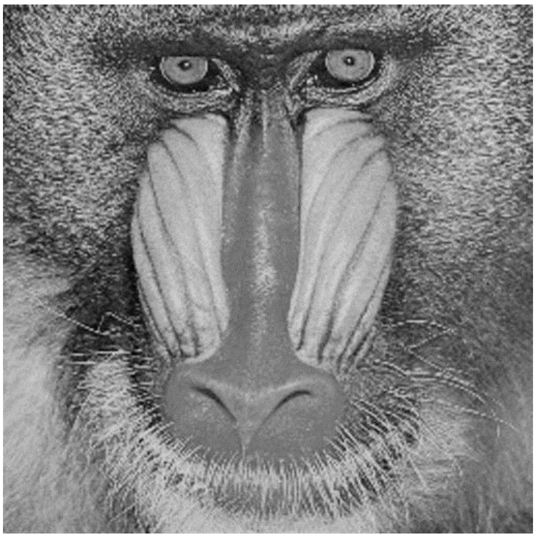
(c) Almohammad’s stego ‘Lena’ image

(d) Chang’s stego ‘Lena’ image

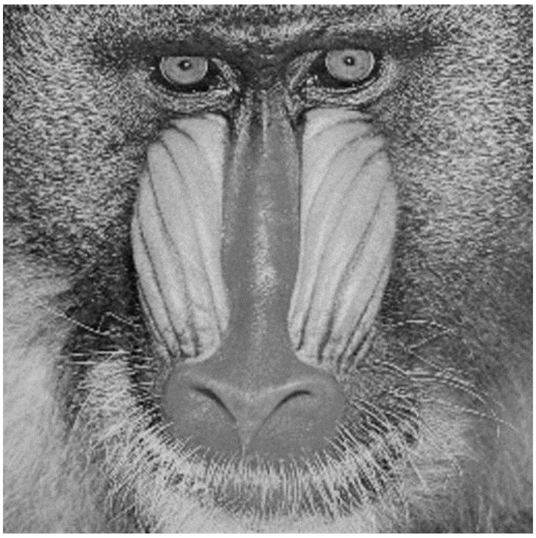
Figure 5.1 ‘Lena’ images



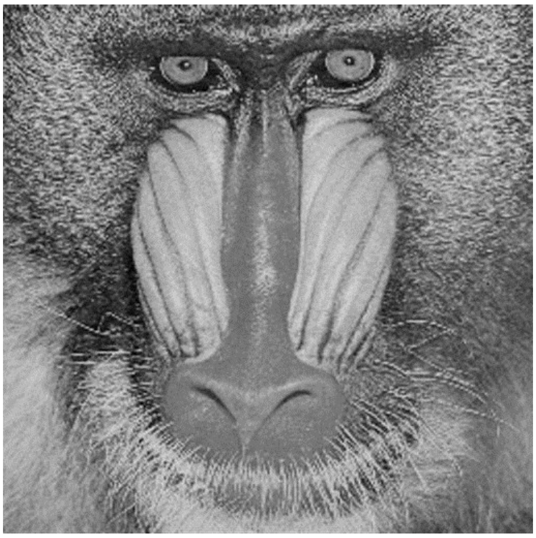
(a) Original ‘Baboon’ image



(b) Our stego ‘Baboon’ image



(c) Almohammad’s stego ‘Baboon’ image



(d) Chang’s stego ‘Baboon’ image

Figure 5.2 ‘Baboon’ images

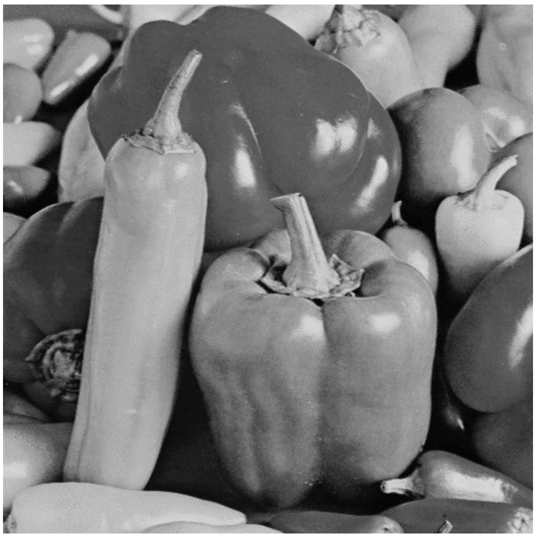
(a) Original ‘House’ image

(b) Our stego ‘House’ image

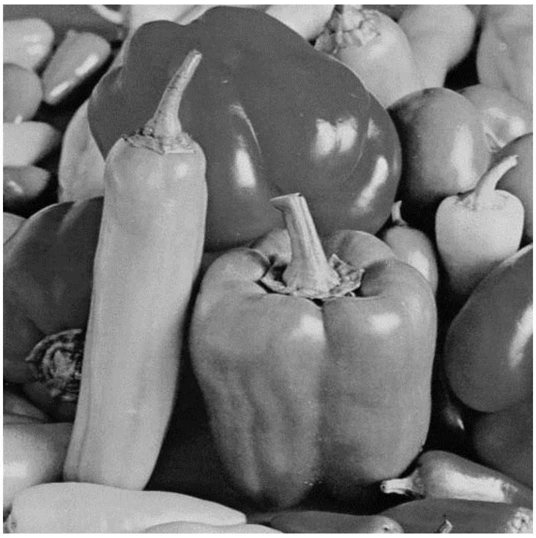
(c) Almohammad’s stego‘House’ image

(d) Chang’s stego ‘House’ image

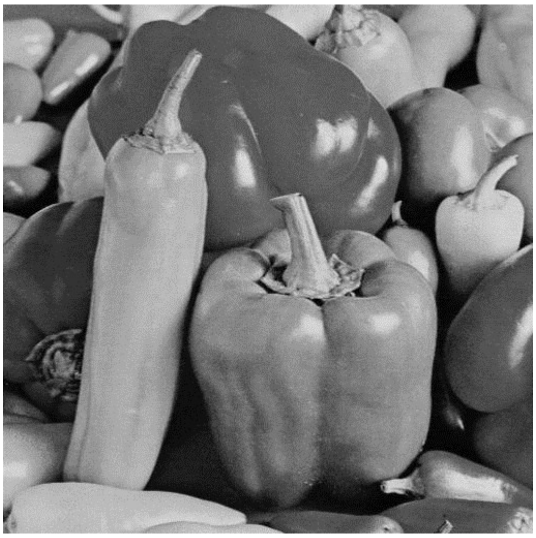
Figure 5.3 ‘House’ images



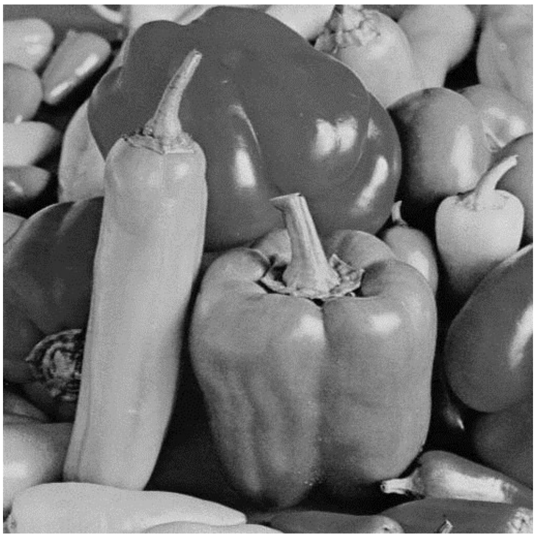
(a) Original ‘Peppers’ image



(b) Our stego ‘Peppers’ image



(c) Almohammad’s stego‘Peppers’ image



(d) Chang’s stego ‘Peppers’ image

Figure 5.4 ‘Peppers’ images

5.4 Computation time

As the result shown in the table.5.4, the proposed method's computation time is less than that of Chang"s method. Moreover, it is also faster than Almohammad"s method. Even though, in proposed method, the secret messages to be embedded are more than the other two methods, the time consumed by this method is still the least. In addition, with the 32x32-pixel block size, the number of the blocks to be calculated is reduced hence the computation time is decreased. Furthermore, higher difference between the result of the proposed method and that of the others two methods is gained if the larger image is used as a cover image.

Table 5.4: The computation time (s)

Method	**Image**			
	Lena	***Pepper***	***Baboon***	***House***
Chang's	1.36	1.35	1.40	1.33
Almohammad's	0.55	0.53	0.54	0.50
Proposed method	0.34	0.33	0.35	0.32

5.5 Performance of the interpolated quantization table

Table 5.5 and 5.6 show the compression ratio and quality of the four stego-images that use the 16x16 interpolated quantization table (Table 4.2) and Almohammad's 16x16 quantization table (Table 2.3) .

Table 5.5: The compression ratio of the stego-images

Quantization table	**Image**			
	Lena	***Pepper***	***Baboon***	***House***
Almohammad's	2.85:1	2.74:1	2.35:1	3.69:1
16x16 Interpolated	2.94:1	2.86:1	2.41:1	3.74:1

From the result, it can be seen that the compression ratios of the stego-images that use 16x16 interpolated quantization table are greater than those of Almohammad's quantization table. This is because the values in the high frequency part in the 16x16 interpolated quantization table are higher than those of Almohammad's and mostly the compression usually occurs from quantization in high frequency part.

Table 5.6: The PSNR of the stego-images

Quantization table	**Image**			
	Lena	***Pepper***	***Baboon***	***House***
Almohammad's	40	36.6	42.4	46.4
16x16 Interpolated	39.43	36.4	41.39	45.50

From table 5.6 , it can be seen that, the quality of the stego-images that use 16x16 interpolated quantization table are a bit less than those of Almohammad's. This is due to the overall values in the Almohammad's quantization table are lower than those of 16x16 interpolated quantization table so that the quality of the stego-images after compression are a bit greater.

5.6 Performance of the proposed method with the 3 different amounts of the capacity.

In order to compare the performance of three differences capacity embedded, 50 images are tested with the proposed method by embedding the secret messages for 247296, 251904, and 256000 bits, respectively. After that, the size, the quality and the computation time of each stego-image will be measured to compare the results. Moreover, the computation time barely depends upon the three other criteria.

Table 5.7: The average quality and size of the stego-images with three different amounts of the capacity of the secret messages embedded

Capacity (bits)	**Average size (KB)**	**Average PSNR**
247296	104.345	34.784
251904	105.291	34.849
256000	106.236	34.871

From the results, it can be seen that the average size and average PSNR increase a little bit when the capacity is increased. However, the increasing in the percentage of the capacity is greater than that of size and PSNR. Therefore the 256000 capacity is selected in the proposed method.

CHAPTER VI

CONCLUSION

In this work, the frequency based steganography using block size of 32x32 pixels with the interpolated quantization table and DCT is proposed. The interpolated 32x32-pixel quantization table is created by using the cubic spline interpolation technique. Once quantization table is obtained, the embedding process of the secret messages is ready. Initially, a cover image is transformed into frequency domain by using DCT and the secret messages are then embedded after the quantization process.

To measure the performance of the proposed method with the interpolated quantization table, the experiments were conducted to compare our method with Chang's and Almohammad's methods. The results showed that the proposed model can embed the secret messages more than the others. In case of compression ratio, our model outperforms the others as well whilst the model is better than Chang's method and close Almohammad's method in term of image quality. Also, the proposed method's computation time taken is less than that of the other methods. Under such criteria, the proposed model can reduce the computation time and increase the capacity while maintaining the size and image quality.

REFERENCES

[1] Cheddad, A., Condell, J., Curran, K., and Kevitt, P.M. Digital image steganography: survey and analysis of current methods. Signal Processing 90(3) (March 2010) : 727-752.

[2] Chan, C.K., and Cheng, L.M. Hiding data in images by simple LSB substitution. Pattern Recognition 37(3) (March 2004) : 469 – 474.

[3] Chang, C.C., Chen, T.S., and Chung, L.Z. A steganographic method based upon JPEG and quantization table modification. Information Science 141(2) (March 2002) : 123-138.

[4] Almohammad, A., Hierons, R.M., and Ghinea, G. High capacity steganographic method based upon JPEG. IEEE 3rd International Conference on Availability, Reliability and Security (March 2008) : 544-549.

[5] Nag, A., Biswas, S., Sarkar, D., and Sarkar, P.P. A novel technique for image steganography based on DWT and huffman encoding. International Journal of Computer Science and Security 4(6) (2011) : 561-570.

[6] Monro, D.M., and Sherlock, B.G. Optimal quantisation table for DCT image compression. IEEE Proceeding on Vision, Image and Signal Processing 143(1) (February 1996) : 10-14.

[7] Watson, A.B. Image compression using the Discrete Cosine Transform. Mathematical Journal 4(1) (1994) : 81-88.

[8] Wallace, G.K. The JPEG still picture compression standard. Communication of the ACM 34(4) (April 1991) : 30-44.

[9] Gilat, A., and Subramaniam, V. Numerical methods for engineers and scientists: an introduction with applications using MATLAB. Hoboken, NJ : Wiley, 2007.

BIOGRAPHY

Mr. Natee Vongurai was born at Bangkok, Thailand in 1987. He received the bachelor degree in engineering from Chulalongkorn University and is studying the master degree at Chulalongkorn University in Computer Science and Information Technology program, department of Mathematics.