

การใช้กระบวนการเชิงเส้นในการควอนไทซ์ข้อมูลแบบเวกเตอร์
A LINEAR APPROACH IN VECTOR QUANTIZATION



นายสุรเดช ตรีไตรลักษณ์
MR.SURADEJ TRETRILUXANA

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาดามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมไฟฟ้า

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2540

ISBN 974-621-989-8

ลิขสิทธิ์เป็นของบัณฑิตวิทยาลัย สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เลขหมู่.....

เลขทะเบียน.....28867

วัน, เดือน, ปี 1 0 พ.ย. 2540

เอกสารนี้เป็นเอกสารที่นำไปใช้หรือใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปะเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

A LINEAR APPROACH IN VECTOR QUANTIZATION



**A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE
MASTER OF ENGINEERING IN ELECTRICAL ENGINEERING
SCHOOL OF GRADUATE STUDIES
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
1997
ISBN 974-621-989-8**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น ลิขสิทธิ์ทางเว็บไซต์ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อวิทยานิพนธ์	การใช้กระบวนการเชิงเส้นในการควอนไทซ์ข้อมูลแบบเวกเตอร์
นักศึกษา	นายสุรเดช ตรีไตรลักษณะ
อาจารย์ผู้ควบคุมวิทยานิพนธ์	รศ.ดร.มนัส สังวรศิลป์
ระดับการศึกษา	วิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้า
ภาควิชา	วิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์
	สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
พ.ศ.	2540

บทคัดย่อ

กระบวนการควอนไทซ์ข้อมูลแบบเวกเตอร์เป็นกระบวนการหนึ่งที่ใช้กันมากในงานประมวลผลสัญญาณดิจิทัล และวิธีการของ LBG ก็เป็นวิธีการที่ขอมรับกันอย่างแพร่หลาย อย่างไรก็ตามวิธีการดังกล่าวใช้เวลามากในการประมวลผล โดยเฉพาะอย่างยิ่งเมื่อนำมาใช้กับข้อมูลจำนวนมาก เช่น การประมวลผลภาพ จึงมีความพยายามที่จะแบ่งแยกข้อมูลออกเป็นกลุ่มๆ ก่อนที่จะผ่านวิธีการดังกล่าว ที่เรียกว่า Classified Vector Quantization (CVQ) โดยกำหนดเกณฑ์ที่ใช้ในการแบ่งอย่างแน่นอน ประเด็นปัญหาก็คือเกณฑ์ที่กำหนดขึ้นมานั้นจะเหมาะสมกับข้อมูลชนิดหนึ่งๆ หากเราต้องการนำไปใช้กับข้อมูลกลุ่มอื่นก็ต้องแก้ไขกฎเกณฑ์ต้องสร้างกฎเกณฑ์ขึ้นมาใหม่ ซึ่งเป็นเรื่องที่ยุ่งยากมาก โดยเฉพาะกับข้อมูลที่มีความหลากหลาย วิทยานิพนธ์ฉบับนี้ได้ศึกษาถึงปัญหาดังกล่าวและนำเสนอวิธีแก้ไขโดยใช้หลักการ Polynomial Curve Fitting ที่มี Degree = 1 หรือที่เรียกว่า Linear Curve Fitting เป็นตัวแทนข้อมูลมาใช้ในการพิจารณาทำให้มองเห็นข้อมูลชัดเจนและแบ่งแยกได้ง่ายขึ้น

Thesis Title **A Linear Approach in Vector Quantization**
Student **Mr.Suradej Tretriluxana**
Thesis Advisor **Assoc.Prof.Dr.Manas Sangworasilp**
Level of Study **Master of Engineering in Electrical Engineering**
Department **Computer Engineering, Faculty of Engineering,**
King Mongkut's Institute of Technology Ladkrabang
Year **1997**

Abstract

Vector Quantization is widely used in digital signal processing and the process of LBG is a well known method used. However, this method consumes a large amount of time to compute all data especially in the case of image data. Therefore the datas are divided into groups before being processed by LBG and this procedure is called Classified Vector Quantization (CVQ). Data are classified according to an exact constrain. Problem is the limitation of constrain to only certain groups of data. Modified or new constrain must be created if new groups of data are applied. All these methods are difficult to handle. In order to overcome this problem, we introduce the principle of Polynomial Curve Fitting of degree one so it is called Linear Curve Fitting. It represents the data to make them easier to be classified later.

กิตติกรรมประกาศ

การจัดทำวิทยานิพนธ์ฉบับนี้ได้รับความสำเร็จลุล่วงไปด้วยดีก็เพราะได้รับการสนับสนุน และส่งเสริมทั้งกำลังใจและความกรุณาจากบุคคลหลายฝ่าย ผู้เขียนขอขอบพระคุณคุณพ่อ, คุณแม่ และครอบครัวที่คอยให้การสนับสนุนและกำลังใจเป็นอย่างมาก ขอขอบพระคุณ รศ.ดร.มนัส สังวรศิลป์ ในฐานะอาจารย์ที่ปรึกษาที่คอยให้ความดูแลเอาใจใส่อย่างสม่ำเสมอ ขอขอบคุณครู, อาจารย์ที่ได้สั่งสอนอบรม ขอขอบคุณดร.สมศักดิ์ ชุ่มช่วย, ดร.กิติพล ชิตสกุล, คุณจารุกุล สีม่วง, คุณศิริพงษ์ วงษ์การ เพื่อนอาจารย์และเพื่อนที่ไม่ได้เป็นอาจารย์ที่ไม่ได้กล่าวนาม ณ.ที่นี้ที่มีส่วนให้ความช่วยเหลือในสิ่งต่างๆ และท้ายสุดขอขอบคุณบัณฑิตวิทยาลัยที่ให้การสนับสนุนทุนสำหรับการจัดทำรูปเล่มวิทยานิพนธ์นี้

คุณค่าและประโยชน์อันพึงมีจากวิทยานิพนธ์ฉบับนี้ผู้วิจัยขอบแต่ผู้มีพระคุณและผู้ที่มีส่วน
ใจทุกท่าน

สุรเดช ตรีไตรลักษณ์

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VII
สารบัญภาพ.....	VIII
บทที่	
1. บทนำ.....	1
ความเป็นมาและความสำคัญของปัญหา.....	1
วัตถุประสงค์ของการวิจัย.....	1
สมมติฐานของการศึกษา.....	2
แนวคิดที่ใช้ในการวิจัย.....	2
ขอบเขตการวิจัย.....	2
2. ความเข้าใจพื้นฐานของกระบวนการเวกเตอร์ควอนไทซ์เซชัน.....	4
บทนำ.....	4
การควอนไทซ์แบบสเกลาร์ (Scalar Quantization).....	6
การควอนไทซ์แบบเวกเตอร์ (Vector Quantization).....	10
การวัดประสิทธิภาพของการควอนไทซ์แบบเวกเตอร์.....	12
ข้อได้เปรียบของการควอนไทซ์แบบเวกเตอร์.....	12
3. การออกแบบตารางเปรียบเทียบข้อมูล.....	17
บทนำ.....	17
ข้อกำหนดการออกแบบตารางเปรียบเทียบข้อมูล.....	17
วิธีการของ LBG.....	18
4. การประมาณค่าฟังก์ชัน โดยวิธี Least-Square.....	23
บทนำ.....	23
หลักการประมาณค่าแบบ Least-Square.....	23

สารบัญ(ต่อ)

บทที่	หน้า
การประมาณค่าโดยใช้สมการโพลิโนเมียลลำดับที่หนึ่ง.....	24
การประมาณค่าโดยใช้สมการโพลิโนเมียลลำดับที่ m	25
การประมาณค่าโดยใช้สมการเอ็กโพเนนเชียล.....	26
5. การประมาณค่าเชิงเส้นในกระบวนการควอนไทซ์แบบเวกเตอร์.....	27
บทนำ.....	27
การควอนไทซ์ข้อมูลเวกเตอร์เป็นกลุ่ม.....	27
ข้อจำกัดของวิธีการควอนไทซ์ข้อมูลเวกเตอร์เป็นกลุ่ม.....	29
การแบ่งกลุ่มข้อมูลโดยใช้การประมาณเชิงเส้น.....	30
ผลเปรียบเทียบปริมาณการคำนวณ.....	31
6. การทดลองและผลการทดลอง.....	33
บทนำ.....	33
ขอบเขตการทดลอง.....	33
ข้อมูลที่ใช้ในการทดลอง.....	33
การตรวจสอบและวัดผล.....	33
แผนภูมิลำดับการทดลอง.....	34
การทดลอง.....	36
การแปลงข้อมูลภาพต้นฉบับเป็น Input Vector.....	36
การหา Codebook ด้วยวิธีการ LBG's.....	37
VQ Encoder.....	40
การสร้างภาพข้อมูลกลับมาใหม่ (Reconstructed Image).....	41
การหาค่าประมาณเชิงเส้น (Linear Approximation).....	41
การแบ่งกลุ่มข้อมูล.....	43
การรวมค่า Codebook (Union Codebook).....	45
ผลการทดลอง.....	46
7. สรุปผลและแนวทางการพัฒนา.....	59

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

บทที่	หน้า
สรุปผลการทดลอง.....	59
ปัญหาที่พบ.....	60
แนวทางการพัฒนา.....	61
บรรณานุกรม.....	62
ภาคผนวก.....	63
ภาคผนวก ก Source Code ("M" files) ที่ใช้ในการทดลอง.....	64
ภาคผนวก ข ผลงานวิจัยที่ได้รับการตีพิมพ์.....	78
ประวัติผู้เขียน.....	79



สารบัญตาราง

หน้า

1 ตารางผลการทดลอง.....46



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการตีพิมพ์เท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

หน้า

1	แสดงลำดับการทำงานของการทำงานการเข้ารหัสที่ใช้ในการสื่อสาร.....	4
2	แสดงลำดับการทำงานของการทำงานการเข้ารหัสที่ใช้ในการเก็บข้อมูล.....	5
3	แสดงลำดับการทำงานของการทำงานการลดข้อมูลภาพ.....	5
4	แสดงตัวอย่างการควอนไทซ์ข้อมูลแบบยูนิฟอร์ม	7
5	แสดงความสัมพันธ์ระหว่าง eq และค่า f	8
6	แสดงผลเปรียบเทียบค่าความคลาดเคลื่อนเฉลี่ยระหว่างการควอนไทซ์แบบยูนิฟอร์มและการควอนไทซ์แบบ Lloyd-Max.....	9
7	แสดงการควอนไทซ์ข้อมูลแบบยูนิฟอร์มกับข้อมูลที่ไม่เป็นยูนิฟอร์มโดยการแปลงแบบ Nonlinear Transform.....	10
8	แสดงตัวอย่างของ VQ ที่มีจำนวนมิติ (N) = 2 และ Reconstruction Level (L) = 9.	12
9	แสดงข้อเปรียบเทียบระหว่าง VQ และ SQ กับข้อมูลที่มีความไม่อิสระแบบเชิงเส้น (Linear Dependence).....	13
10	แสดงข้อเปรียบเทียบระหว่าง VQ และ SQ กับข้อมูลที่มีความไม่อิสระแบบไม่เป็นเชิงเส้น (Nonlinear Dependence).....	14
11	แสดงข้อเปรียบเทียบระหว่าง VQ และ SQ กับข้อมูลที่เป็นอิสระต่อกัน (Statistically Independence).....	16
12	แสดงแผนภูมิการหาค่า Codebook ด้วยวิธี LBG's.....	19
13	แสดงลักษณะการประมาณค่าแบบ Least-Square.....	23
14(a)	แสดงแผนภูมิการควอนไทซ์แบบ LBG's.....	27
14(b)	แสดงแผนภูมิการควอนไทซ์ข้อมูลเวกเตอร์แบบเป็นกลุ่ม (CVQ).....	28
15	แสดงการหาลักษณะขอบภาพมาแบ่งกลุ่มข้อมูล.....	29
16	แสดงการใช้วิธี Linear Approach มาแบ่งกลุ่มข้อมูล.....	30
17	แสดงแผนภูมิการทดลองการควอนไทซ์แบบเวกเตอร์ด้วยวิธี LBG's.....	34
18	แสดงแผนภูมิการทดลองการควอนไทซ์แบบเวกเตอร์โดยวิธี Linear Approach.....	35
19	แสดงการแปลงภาพต้นฉบับเป็นข้อมูลเวกเตอร์.....	36
20	แสดงแผนภูมิการทดลองการหาค่า Codebook ด้วยวิธี LBG's.....	37

VIII

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ(ต่อ)

	หน้า
21 แสดงแผนภูมิการหาค่า Initial Codebook.....	38
22 แสดงตัวอย่างของ "S" Matrix.....	39
23 แสดงการหาค่าประมาณเชิงเส้น.....	41
24 แสดงการลดจำนวนมิติที่พิจารณาจาก 16 มิติเหลือ 2 มิติ.....	43
25 แสดงการจัดแบ่งข้อมูลที่ใช้ในการทดลอง.....	44
26 แสดงข้อมูลภาพต้นฉบับชื่อ "Camera".....	47
27 แสดงข้อมูลภาพต้นฉบับชื่อ "Cell".....	48
28 แสดงข้อมูลภาพต้นฉบับชื่อ "Dog".....	48
29 แสดงข้อมูลภาพต้นฉบับชื่อ "Girl".....	49
30 แสดงข้อมูลภาพต้นฉบับชื่อ "Lady".....	49
31 แสดงข้อมูลภาพต้นฉบับชื่อ "Lena".....	50
32 แสดงข้อมูลภาพต้นฉบับชื่อ "Plant".....	50
33 แสดงข้อมูลผลลัพธ์ภาพชื่อ "Camera" ที่ได้จากวิธีการ Linear Approach.....	51
34 แสดงข้อมูลผลลัพธ์ภาพชื่อ "Cell" ที่ได้จากวิธีการ Linear Approach.....	52
35 แสดงข้อมูลผลลัพธ์ภาพชื่อ "Dog" ที่ได้จากวิธีการ Linear Approach.....	52
36 แสดงข้อมูลผลลัพธ์ภาพชื่อ "Girl" ที่ได้จากวิธีการ Linear Approach.....	53
37 แสดงข้อมูลผลลัพธ์ภาพชื่อ "Lady" ที่ได้จากวิธีการ Linear Approach.....	53
38 แสดงข้อมูลผลลัพธ์ภาพชื่อ "Lena" ที่ได้จากวิธีการ Linear Approach.....	54
39 แสดงข้อมูลผลลัพธ์ภาพชื่อ "Plant" ที่ได้จากวิธีการ Linear Approach.....	54
40 แสดงข้อมูลผลลัพธ์ภาพชื่อ "Camera" ที่ได้จากวิธีการ LBG's.....	55
41 แสดงข้อมูลผลลัพธ์ภาพชื่อ "Cell" ที่ได้จากวิธีการ LBG's.....	56
42 แสดงข้อมูลผลลัพธ์ภาพชื่อ "Dog" ที่ได้จากวิธีการ LBG's.....	56
43 แสดงข้อมูลผลลัพธ์ภาพชื่อ "Girl" ที่ได้จากวิธีการ LBG's.....	57
44 แสดงข้อมูลผลลัพธ์ภาพชื่อ "Lady" ที่ได้จากวิธีการ LBG's.....	57
45 แสดงข้อมูลผลลัพธ์ภาพชื่อ "Lena" ที่ได้จากวิธีการ LBG's.....	58
46 แสดงข้อมูลผลลัพธ์ภาพชื่อ "Lena" ที่ได้จากวิธีการ LBG's.....	58

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

ความเป็นมาและความสำคัญของปัญหา

กระบวนการควอนไทซ์ข้อมูลแบบเวกเตอร์ (Vector Quantization) เป็นกระบวนการทางด้านการจัดการสัญญาณแบบหนึ่งที่มีความสนใจมากตั้งแต่ทศวรรษที่ 80 เป็นต้นมา มีงานวิจัยจำนวนมากที่กล่าวถึงหรือเกี่ยวข้องกับกระบวนการนี้ ก่อนหน้านี้วิธีดังกล่าวไม่ได้รับความนิยมนักเพราะติดขัดที่การหาค่าพารามิเตอร์ที่มีความสำคัญของกระบวนการที่เรียกว่า "ตารางเปรียบเทียบข้อมูล" (Codebook) ทำได้ยุ่งยาก ภายหลังมีกลุ่มนักวิจัยที่ประกอบด้วย Y. Linde, A. Bugo และ R.M. Gray สามารถคิดค้นวิธีการหาค่า Codebook ที่ดีขึ้น ต่อมาจึงเรียกวิธีที่คิดค้นนี้ว่า LBG's

แม้ว่าวิธี LBG's จะได้รับการยอมรับแต่ก็ยังมีข้อด้อยอยู่ เรื่องการใช้ทรัพยากรของระบบอย่างมากทั้งปริมาณการคำนวณและหน่วยความจำของระบบ ด้วยเหตุนี้จึงมีแนวคิดที่จะปรับปรุงวิธีการ LBG's นี้ให้ดีขึ้น ลดการใช้ทรัพยากรของระบบลง โดยเฉพาะปริมาณการคำนวณเพื่อให้การประมวลผลเป็นไปด้วยความรวดเร็ว

วัตถุประสงค์ของการวิจัย

งานวิจัยชิ้นนี้มีวัตถุประสงค์ต่อไปนี้

1. ปรับปรุงวิธีการ LBG's ให้ดีขึ้น ลดการใช้ทรัพยากรโดยลดปริมาณการประมวลผลลงเพื่อให้ระบบทำงานได้เร็วขึ้น
2. ทดลองใช้วิธีที่ได้ปรับปรุงใหม่นี้มาประยุกต์ใช้งานจริง ในที่นี้คือการนำมาประมวลผลภาพ (Image Processing) ให้ลดพื้นที่ที่ใช้จัดเก็บข้อมูลลง (Data Compression) และเปรียบเทียบผลลัพธ์ที่ได้กับการใช้วิธี LBG's โดยตรง
3. ขยายแนวคิดของการวิจัยเรื่องการควอนไทซ์ข้อมูลแบบเวกเตอร์ให้กว้างขวางขึ้น เสนอแนะแนวทางที่อาจใช้พัฒนางานด้านนี้ต่อไป

สมมติฐานของการศึกษา

เนื่องจากการควอนไทซ์ข้อมูลแบบเวกเตอร์เป็นการนำข้อมูลอินพุตทั้งหมดมาประมวลผลกับ "ค่าตารางเปรียบเทียบข้อมูล" ทั้งหมด เป็นสาเหตุให้มีปริมาณการประมวลผลอย่างมาก จึงตั้งข้อสมมติฐานว่าหากเราสามารถแบ่งข้อมูลอินพุตให้เป็นกลุ่มย่อยๆและจัดสัดส่วนของข้อมูลที่ต้องใช้ใน Codebook กับข้อมูลอินพุตให้เหมาะสม แล้วจึงประมวลในกลุ่มย่อยๆนั้นแทน น่าจะเป็นวิธีลดปริมาณการคำนวณรวมทั้งหมดลงได้และจะทำให้ความเร็วในการประมวลผลเพิ่มขึ้น

แนวคิดที่ใช้ในการวิจัย

แนวความคิดใหญ่ๆที่ใช้ในการวิจัยนี้มี 2 เรื่องคือ

1. แนวคิดเรื่องการแบ่งกลุ่มข้อมูลอินพุตออกเป็นกลุ่มย่อยๆและจัดสัดส่วนให้พอเหมาะกับจำนวนข้อมูลใน Codebook ได้กล่าวไว้ในเรื่อง "การควอนไทซ์ข้อมูลเวกเตอร์เป็นกลุ่ม" (Classified Vector Quantization : CVQ) ซึ่งตรงกับสมมติฐานที่ตั้งไว้ตอนแรกด้วย
2. การควอนไทซ์ข้อมูลแบบเวกเตอร์นี้จัดว่าเป็นการลดข้อมูลแบบหนึ่งที่มีการสูญเสีย (Lossy Compression) ดังนั้นแนวคิดที่จะพยายามแบ่งกลุ่มข้อมูลที่มีความใกล้เคียงกันให้อยู่ในกลุ่มเดียวกันจึงมีหลายวิธี และแนวคิดหนึ่งที่ถูกนำมาใช้ในงานวิจัยนี้คือ "การประมาณค่าเชิงเส้น" (Linear Approximation) โดยการแบ่งกลุ่มข้อมูลตามผลลัพธ์ที่ได้จากการประมาณค่าเชิงเส้นที่มีความใกล้เคียงกัน

ขอบเขตการวิจัย

ขอบเขตของงานวิจัยนี้อยู่ที่การนำเสนอการทดลองเปรียบเทียบระหว่างการควอนไทซ์ข้อมูลเวกเตอร์ด้วยวิธีเดิม (LBG's) กับวิธีที่ใช้การประมาณเชิงเส้นเข้ามาร่วม (Linear Approach) โดยทำการทดลองกับข้อมูลภาพขนาด 256 x 256 พิกเซล แต่ละพิกเซลมีความเข้มของระดับสีเทา 256 ระดับ (0-255) ซึ่งข้อมูลนี้จะถูกแปลงเป็นอินพุตเวกเตอร์ขนาด 16 มิติ จำนวน 4096 ตัวในเวลาต่อมา การทดลองจะกำหนดให้สร้าง Codebook ที่มีข้อมูลเวกเตอร์ภายใน 256 ตัว และเพื่อให้อึดหยุ่นกับข้อมูลหลายแบบ จึงใช้ภาพจำนวน 7 ภาพที่ต่างกันมาแทนข้อมูลอินพุตที่มีลักษณะต่างๆ และเนื่องจากวิธีการควอนไทซ์ข้อมูลแบบเวกเตอร์นี้ไม่สามารถประมาณผลการทำงานล่วงหน้าได้ การวัดผลจึงต้องวัดจากเวลาที่ใช้ทำงานแทนการวัดปริมาณการคำนวณ นอกจากนี้ยังวัดค่าความคลาดเคลื่อนของผลลัพธ์ที่ผ่านวิธีทั้งสองเปรียบเทียบกันด้วย

การทดลองทั้งหมดที่กล่าวมานี้จะพัฒนามาบนเครื่องไมโครคอมพิวเตอร์โดยใช้โปรแกรมคำนวณผลทางคณิตศาสตร์ Matlab[®] เวอร์ชัน 4.0 และวิเคราะห์ผลในขั้นสุดท้ายสำหรับเนื้อหารายละเอียดของรายงานได้แยกกล่าวไว้เป็นบทๆ ดังนี้

1. บทนำ กล่าวถึง ความเป็นมาและความสำคัญของปัญหา, วัตถุประสงค์ของการวิจัย, สมมติฐานของการศึกษา, แนวคิดที่ใช้ในการวิจัย และขอบเขตการวิจัย
2. ความเข้าใจพื้นฐานของกระบวนการเวกเตอร์ควอนไทซ์เซชัน คือการให้ความสนใจในภาพกว้างของพื้นฐานและประยุกต์ใช้งานของเวกเตอร์ควอนไทซ์เซชัน บทสรุปของที่มาตั้งแต่สเกลาร์ควอนไทซ์เซชัน, เวกเตอร์ควอนไทซ์เซชัน, การวัดประสิทธิภาพระบบและเปรียบเทียบข้อดีของการควอนไทซ์ข้อมูลแบบเวกเตอร์
3. การออกแบบตารางเปรียบเทียบข้อมูล อธิบายถึงหลักการออกแบบตารางเปรียบเทียบข้อมูล, เงื่อนไขที่เกี่ยวข้อง, อธิบายขั้นตอนของวิธีการ LBG's และชี้ให้เห็นถึงปัญหาที่เกิดขึ้น
4. การประมาณค่าฟังก์ชันโดยวิธี Least-Square กล่าวถึงทฤษฎีการประมาณค่าโดยวิธี Least-Square แม้ว่างานวิจัยนี้จะเลือกเอาการประมาณค่าเชิงเส้นมาใช้ก็ตามแต่เนื้อหาในบทนี้ยังได้กล่าวเสริมโดยสรุปของการประมาณค่าโดยใช้สมการโพลิโนเมียลลำดับที่ m และการประมาณค่าโดยใช้สมการเอ็กโพเนนเชียลด้วย
5. การประมาณค่าเชิงเส้นในกระบวนการควอนไทซ์แบบเวกเตอร์ เป็นการสรุปแนวคิดที่กล่าวไว้ในบทที่ 3 และ 4 มาใช้งาน โดยเริ่มจากการกล่าวถึงปัญหา, แนวทางพัฒนาที่ใช้และผลเปรียบเทียบความเป็นไปได้เชิงทฤษฎี
6. การทดลองและผลการทดลอง จะอธิบายรายละเอียดทุกขั้นตอนที่ใช้ในการทดลอง ตั้งแต่ ขอบเขตการทดลอง, ข้อมูลที่ใช้ในการทดลอง, การตรวจสอบและวัดผล, แผนภูมิลำดับการทดลอง, ขั้นตอนการทดลอง รวมทั้งผลลัพธ์ที่ได้เปรียบเทียบกันทั้งสองวิธี ทั้งในแง่ตัวเลขและผลลัพธ์ที่เป็นรูปภาพ
7. บทสรุปและแนวทางการพัฒนา

บทที่ 2

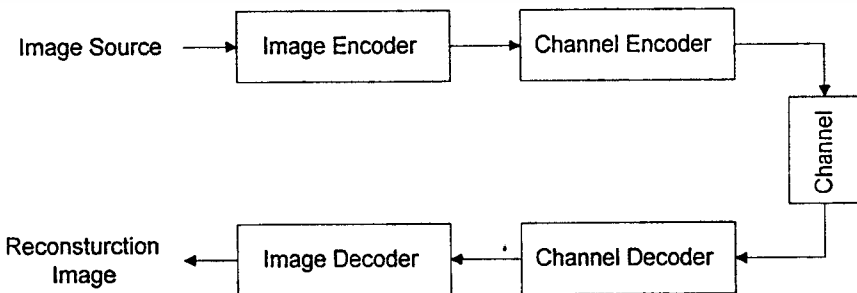
ความรู้พื้นฐานของกระบวนการเวกเตอร์ควอนไทซ์เซชัน

บทนำ

กระบวนการเวกเตอร์ควอนไทซ์เซชัน (Vector Quantization) หรือที่เรียกโดยย่อว่า "VQ" เป็นส่วนหนึ่งของ "การประมวลผลสัญญาณ" (Signal Processing) ได้มีการนำมาประยุกต์ใช้ในการประมวลผลภาพ (Image Processing) ที่เรียกว่า "การเข้ารหัสภาพ" (Image Coding) โดยมีวัตถุประสงค์เพื่อลดจำนวนบิตที่ใช้ในการเก็บข้อมูลและพยายามคงคุณภาพของภาพเดิมไว้ให้มากที่สุด อาจแบ่งแยกตามการใช้งานได้เป็น 2 ลักษณะ คือ การเข้ารหัสเพื่อลดแบนวิธของการรับ/ส่งข้อมูลที่นำไปใช้ประโยชน์ในงานสื่อสารต่างๆ ได้แก่ การส่งสัญญาณโทรทัศน์ การประชุมทางไกล (Video Conference) เป็นต้น และการเข้ารหัสเพื่อลดพื้นที่ที่ใช้ในการเก็บข้อมูลลง เพื่อนำไปใช้ในอุปกรณ์จัดเก็บข้อมูล (Data Storage Device), การจัดทำฐานข้อมูล เป็นต้น

นอกจากนั้นการเข้ารหัสภาพยังแบ่งได้อีก 2 แบบตามลักษณะผลลัพธ์คือ แบบที่เมื่อนำมาผ่านกระบวนการย้อนกลับแล้วให้ข้อมูลเหมือนเดิม เรียกว่าการลดข้อมูลแบบไม่สูญเสีย (Lossless Compression บางตำราเรียกว่า Information-preserving) ใช้กับข้อมูลที่มีความสำคัญมากๆ เช่น ภาพถ่ายจากอวกาศ และแบบที่เมื่อผ่านกระบวนการย้อนกลับแล้วให้ข้อมูลไม่เหมือนเดิม เรียกว่าการลดข้อมูลแบบมีการสูญเสีย (Lossy Compression) ใช้กับภาพที่เรายอมให้มีความผิดเพี้ยนได้โดยแลกกับจำนวนพื้นที่เก็บที่ลดลง งานในลักษณะนี้ได้แก่การส่งสัญญาณโทรทัศน์ เป็นต้น

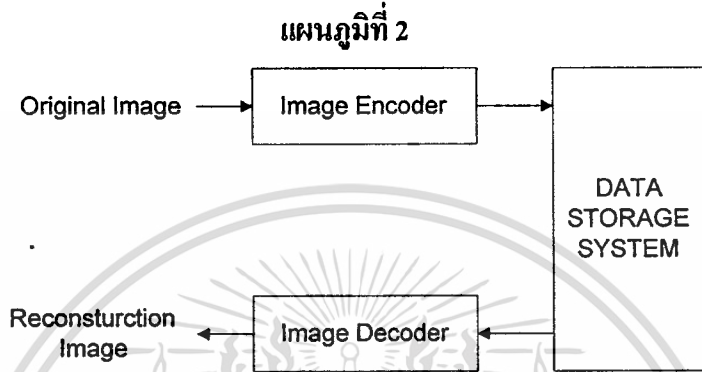
แผนภูมิที่ 1



แสดงลำดับการทำงานของการทำงานของการเข้ารหัสและถอดรหัสที่ใช้ในการสื่อสาร (Opened System)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

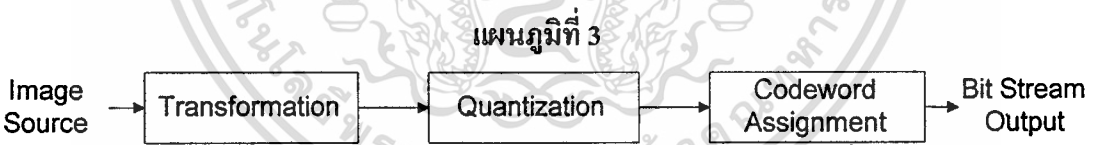
แผนภูมิที่ 1 แสดงลำดับการทำงานทั้งหมดของการสื่อสารส่งผ่านข้อมูลภาพ ตั้งแต่ถ่ายภาพต้นฉบับมาผ่านตัวเข้ารหัส (Image Coder) แปลงข้อมูลต่อโดย Channel Encoder และส่งรหัสที่ได้ผ่าน Channel เมื่อทางฝ่ายรับได้รับรหัสมาแล้วจะแปลงกลับโดยใช้ Channel Decoder และ Image Decoder ตามลำดับเพื่อ ได้ข้อมูลกลับเหมือนเดิม (Lossless) หรือ ใกล้เคียงของเดิม (Lossy) ต่อไป



แสดงลำดับการทำงานของการเข้ารหัสที่ใช้ในการเก็บข้อมูล (Closed System)

ตัวอย่างต่อมาในแผนภูมิที่ 2 เป็นการนำการเข้ารหัสมาช่วยลดพื้นที่การเก็บข้อมูลของระบบจัดเก็บข้อมูล (Data Storage System)

จะเห็นว่าส่วนสำคัญในตัวอย่างทั้งสองนั้นอยู่ที่การเข้า/ถอดรหัสข้อมูลนั่นเอง ในปัจจุบันนี้ ขั้นตอนการลดข้อมูลภาพจะแบ่งเป็น 3 ส่วนสำคัญ ดังแผนภูมิที่ 3



แสดงลำดับการทำงานของการลดข้อมูลภาพ

1. Transformation เป็นการเปลี่ยนรูปแบบข้อมูลต้นฉบับให้อยู่ในรูปแบบอื่นเพื่อเตรียมสำหรับการลดข้อมูลเช่น การแปลงข้อมูลจาก Time Domain ไปเป็น Frequency Domain ด้วยวิธีการแปลงแบบ Cosine, การแปลงแบบ Fourier หรือ การแปลงแบบ Wavelet เป็นต้น

2. Quantization เป็นขั้นตอนสำคัญในการลดพื้นที่ส่วนมากที่ใช้จัดเก็บข้อมูล โดยอาศัยหลักการใช้ตัวแทนข้อมูลที่มีความใกล้เคียงที่สุดแทนข้อมูลเดิม (เป็นการลดแบบ Lossy Compression)

3. Codeword Assignment ทำการลดข้อมูลซ้ำอีกครั้งแบบ Lossless Compression และนำส่วนที่เหลือส่งไปยัง Channel Decoder หรือเก็บลง Data Storage System ต่อไป

โดยทฤษฎีแล้วถ้าการปรับแต่งขั้นตอนทั้งสามนี้เป็นไปอย่างเหมาะสมจะช่วยจัดเก็บข้อมูลได้ใกล้เคียงต้นฉบับมากที่สุดโดยใช้พื้นที่จัดเก็บน้อยที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิพนธ์ให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิทยานิพนธ์ฉบับนี้มีเป้าหมายเพื่อนำเสนอการปรับปรุงเฉพาะในส่วนของการ Quantization ซึ่งเป็นส่วนสำคัญมากต่อกระบวนการทั้งหมดที่กล่าวมา โดยการนำวิธีการเดิมที่ยังมีข้อด้อยอยู่มาปรับปรุงให้มีประสิทธิภาพสูงขึ้น ดังจะอธิบายรายละเอียดในบทต่อไป

การควอนไทซ์แบบสเกลาร์ (Scalar Quantization)

การควอนไทซ์เซชันเริ่มมาจากการควอนไทซ์ข้อมูลแบบสเกลาร์ก่อน โดยหากเราสมมติให้ f เป็นฟังก์ชันของข้อมูลต่อเนื่อง (Continuous Function) เช่น ความเข้มของจุดภาพ, สัมประสิทธิ์ของการทรานฟอร์ม การควอนไทซ์ข้อมูลคือการแทนช่วงต่อเนื่องของค่าเหล่านี้ด้วยข้อมูลคงที่ที่ใกล้เคียงของเดิมที่สุด

การแบ่งระดับข้อมูลฟังก์ชัน f เป็น L ระดับ (Reconstruction Level) และเลือกเอาค่าใดค่าหนึ่งมาใช้แทนโดยให้มีค่าใกล้เคียงค่าเดิมของฟังก์ชัน f มากที่สุดเรียกว่า Amplitude Quantization หากนำวิธีการนี้มาใช้กับข้อมูลเดียว(สเกลาร์)จะเรียกว่า "สเกลาร์ควอนไทซ์เซชัน" แต่เมื่อนำมาใช้กับข้อมูลสเกลาร์ตั้งแต่ 2 ชุดขึ้นไปพร้อมกันก็เรียกว่า "เวกเตอร์ควอนไทซ์เซชัน"

เราอธิบายสเกลาร์ควอนไทซ์เซชันในรูปสมการได้ดังนี้

$$\hat{f} = Q(f) = r_i; \quad d_{i-1} < f \leq d_i \quad (1)$$

โดยที่ $Q(\cdot)$ Quantization Operator
 $r_i; 1 \leq i \leq L$ Reconstruction Level
 $d_i; 0 \leq i \leq L$ Decision Boundaries

จากสมการ (1) จะเห็นว่าค่าของ f อยู่ในช่วง d_{i-1} และ d_i จะถูกแทนที่ด้วยค่า r_i หรืออาจเขียนความสัมพันธ์ระหว่างค่าฟังก์ชัน f และค่าที่ได้จากการควอนไทซ์คือ

$$\hat{f} = Q(f) = f + e_q \quad (2)$$

โดยที่ e_q Quantization Error หรือ Quantization Noise

หรือ
$$e_q = \hat{f} - f \quad (3)$$

สังเกตว่าค่า e_q^2 ก็คือค่า Distortion, $d(f, \hat{f})$ ที่เป็นการวัดค่าความคลาดเคลื่อนระหว่างค่าของ f และค่าที่ได้ \hat{f} ที่นิยมใช้กัน

เมื่อทำการควอนไทซ์ข้อมูลทุกตัวแล้วและมาคิดค่านวณค่าคลาดเคลื่อนรวมทั้งหมดจะได้ค่าความคลาดเคลื่อนเฉลี่ย (Average Distortion : D)

$$D = E[d(f, \hat{f})] = \int_{f_0=-\infty}^{\infty} d(f_0, f) \cdot p_f(f_0) df_0 \quad (4)$$

โดยที่ $E[.]$: Expected Value, $p_f(f_0)$: Probability Density Function ของ f_0

ตัวอย่างการควอนไทซ์ข้อมูลพื้นฐานแบบหนึ่งคือการควอนไทซ์แบบยูนิฟอร์ม (Uniform Quantization) โดยให้ค่า Reconstruction Level เป็นแบบยูนิฟอร์มดังสมการ

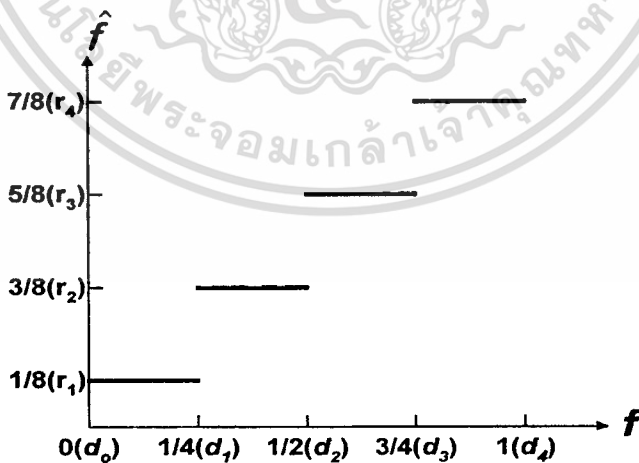
$$d_i - d_{i-1} = \Delta \quad \leq i \leq L \quad (5)$$

และ

$$r_i = \frac{d_i + d_{i-1}}{2} \quad \leq i \leq L \quad (6)$$

Δ ค่าความกว้างระดับ Reconstruction Level

รูปที่ 4

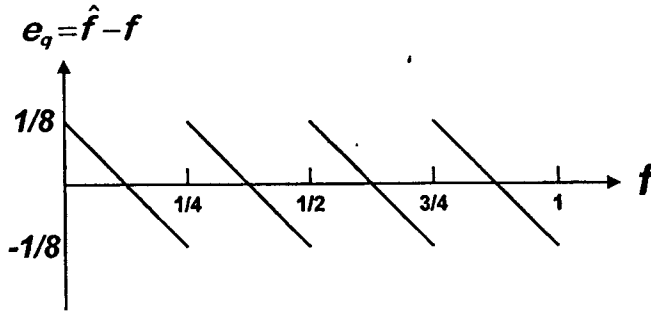


แสดงตัวอย่างการควอนไทซ์ข้อมูลแบบยูนิฟอร์ม

รูปที่ 4 แสดงตัวอย่างการควอนไทซ์ข้อมูลแบบยูนิฟอร์มที่มีค่า $L=4$ และ $0 \leq f \leq 1$ และรูปที่ 5 เป็นกราฟความสัมพันธ์ระหว่างค่า e_q และค่า f

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 5



แสดงความสัมพันธ์ระหว่าง e_q และค่า f

จากตัวอย่างแม้ว่าการควอนไทซ์แบบยูนิฟอร์มจะเป็นวิธีที่เข้าใจง่ายที่สุดแต่อาจไม่ใช่วิธีที่ดีที่สุด หากสมมติว่าข้อมูลส่วนใหญ่ไม่มีอยู่ในช่วง d_0-d_1 ก็ไม่มีความจำเป็นที่ต้องกำหนด r , ไว้ในช่วงนี้แต่ควรกำหนดไว้ในช่วงที่มีความหนาแน่นของข้อมูลสูงกว่าซึ่งจะทำให้ค่าความกว้างระดับ Reconstruction Level ไม่คงที่ เรียกวิธีการควอนไทซ์แบบนี้ว่า "การควอนไทซ์แบบไม่เป็นยูนิฟอร์ม" (Nonuniform Quantization)

วิธีการควอนไทซ์ที่ดีที่สุดได้จากการกำหนดเงื่อนไขให้ ค่าเฉลี่ยกำลังสองของความคลาดเคลื่อนต่ำที่สุด (Minimum Mean Square Error : MMSE) ส่งผลให้ค่าความคลาดเคลื่อนเฉลี่ยต่ำที่สุดด้วย

จาก (4)

$$D = E[d(\hat{f} - f)] = E[e_q^2] = E[(\hat{f} - f)^2]$$

$$D = \int_{f_0=-\infty}^{\infty} p_f(f_0)(\hat{f} - f_0)^2 df_0$$

กำหนดให้ \hat{f} มี Reconstruction Level = L

$$D = \sum_{i=1}^L \int_{f_0=d_{i-1}}^{d_i} p_f(f_0)(r_i - f_0)^2 df_0 \quad (7)$$

$$\frac{\partial D}{\partial r_k} = 0; \quad 1 \leq k \leq L \quad (8.1)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\frac{\partial D}{\partial d_k} = 0; \quad 1 \leq k \leq L \quad (8.2)$$

โดยที่ $d_0 = -\infty$ และ $d_L = \infty$ จาก (7), (8.1) และ (8.2)

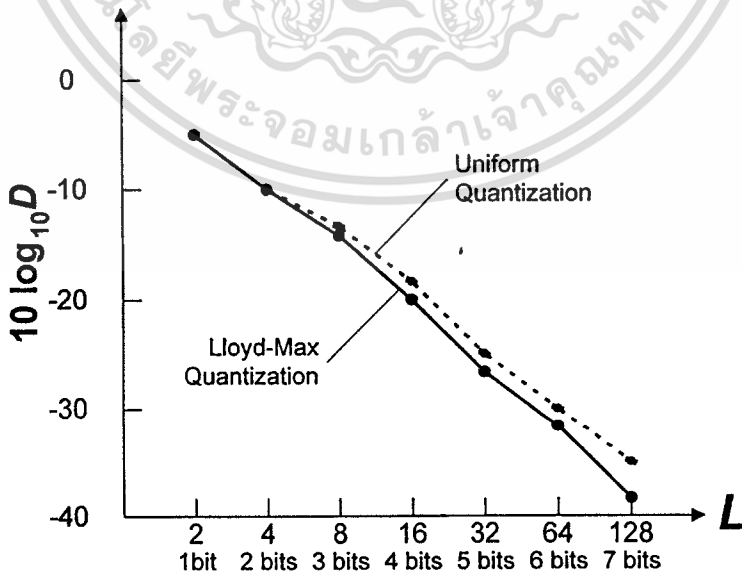
$$r_k = \frac{\int_{f_0=d_{k-1}}^{d_k} f_0 p_f(f_0) df_0}{\int_{f_0=d_{k-1}}^{d_k} p_f(f_0) df_0}; \quad \leq k \leq L \quad (9.1)$$

$$d_k = \frac{r_k + r_{k+1}}{2}; \quad \leq k \leq L \quad (9.2)$$

ยกเว้น $d_0 = -\infty$ และ $d_L = \infty$

สมการ (9.1) แสดงให้เห็นว่า r_k จะต้องอยู่ตรงกลางของค่า $p_f(f_0)$ ในช่วง $d_{k-1} \leq f_0 \leq d_k$ และสมการ (9.2) แสดงว่าค่า d_k เป็นค่ากึ่งกลางระหว่างจุดตัวแทนข้อมูลทั้งสองข้าง (r_k, r_{k+1}) ยกเว้น $d_0 = -\infty$ และ $d_L = \infty$ จึงจะทำให้ค่าความคลาดเคลื่อนต่ำที่สุด วิธีการนี้เรียกว่า "การควอนไทซ์แบบ Lloyd-Max" (Lloyd-Max Quantizer)

รูปที่ 6



แสดงผลเปรียบเทียบค่าความคลาดเคลื่อนเฉลี่ยระหว่างการควอนไทซ์แบบยูนิฟอร์ม

และการควอนไทซ์แบบ Lloyd-Max ที่จำนวน Reconstructed Level : L ต่างๆกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การควอนไทซ์แบบยูนิฟอร์มจะให้ได้ผลดีกับข้อมูลชนิดที่มีการกระจายแบบยูนิฟอร์มเท่านั้น หากเมื่อนำมาใช้กับการกระจายข้อมูลแบบที่ไม่เป็นยูนิฟอร์มแล้วผลที่ได้ย่อมด้อยกว่าการควอนไทซ์แบบ Lloyd-Max ดังแสดงในรูปที่ 6 ซึ่งเป็นผลเปรียบเทียบ "ค่าความคลาดเคลื่อนเฉลี่ย" ที่ได้รับจากทั้งสองวิธี กับข้อมูลที่มีการกระจายแบบ Gaussian สังเกตว่าวิธีการแบบ Lloyd-Max ให้ผลลัพธ์ค่า D ต่ำกว่าที่จำนวนบิต (จำนวน Reconstruction Level) เท่ากัน กล่าวโดยสรุปว่า "ยิ่งการกระจายข้อมูลไม่เป็นแบบยูนิฟอร์มมากเท่าใด การควอนไทซ์แบบ Lloyd Max ก็ยิ่งให้ผลดีกว่ามากขึ้น"

แผนภูมิที่ 7



แสดงการควอนไทซ์ข้อมูลแบบยูนิฟอร์มกับข้อมูลที่ไม่เป็นยูนิฟอร์มโดยการแปลงแบบ Nonlinear Transform

อย่างไรก็ดีหากข้อมูลนั้นได้รับการแปลงจาก "การกระจายแบบไม่เป็นยูนิฟอร์ม" ให้เป็น "การกระจายแบบยูนิฟอร์ม" ด้วยขั้นตอน "Nonlinear Transform" หรือ "Companding : $C[.]$ " แล้ว เช่น ถ้า f เป็นฟังก์ชันของข้อมูลที่ไม่เป็นยูนิฟอร์มถูกแปลงให้เป็นฟังก์ชัน g ที่เป็นแบบยูนิฟอร์มตามสมการ

$$g = C[f] \quad \text{Uniform}$$

การควอนไทซ์แบบยูนิฟอร์มกับผลที่ได้ อาจให้ผลลัพธ์ที่ดีกว่า แต่ค่าความคลาดเคลื่อนเฉลี่ย D จะเป็นของฟังก์ชัน g เอง ไม่ใช่ของข้อมูลโดยตรง (f)

$$D' = E[(\hat{g} - g)^2]$$

การควอนไทซ์แบบเวกเตอร์ (Vector Quantization)

การควอนไทซ์แบบเวกเตอร์ (Vector Quantization : VQ) หรือบางตำราเรียกว่า "Block Quantization" คือการนำข้อมูลที่เป็นสเกลาร์หลายๆตัวมาจัดรวมกันเป็นกลุ่ม (Block) หรือ "หน่วย" หน่วยหนึ่งและทำการควอนไทซ์ข้อมูลสเกลาร์ในหน่วยนั้นร่วมกัน

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กำหนดให้ $f = [f_1, f_2, \dots, f_N]^T$

f คือข้อมูลเวกเตอร์ที่มีขนาด N มิติ เกิดจาก f_i จำนวน N ตัวมาประกอบกัน

การควอนไทซ์ข้อมูลแบบเวกเตอร์คือการแทนที่เวกเตอร์ f ด้วยเวกเตอร์ r ที่มีจำนวนมิติเท่ากัน (N)

$$r = [r_1, r_2, \dots, r_N]^T$$

โดยที่ r เป็นเวกเตอร์ตัวหนึ่งในจำนวน L (Reconstruction Level) ตัวและมีความใกล้เคียงกับเวกเตอร์ f มากที่สุด สามารถเขียนให้อยู่ในรูปสมการได้ดังนี้

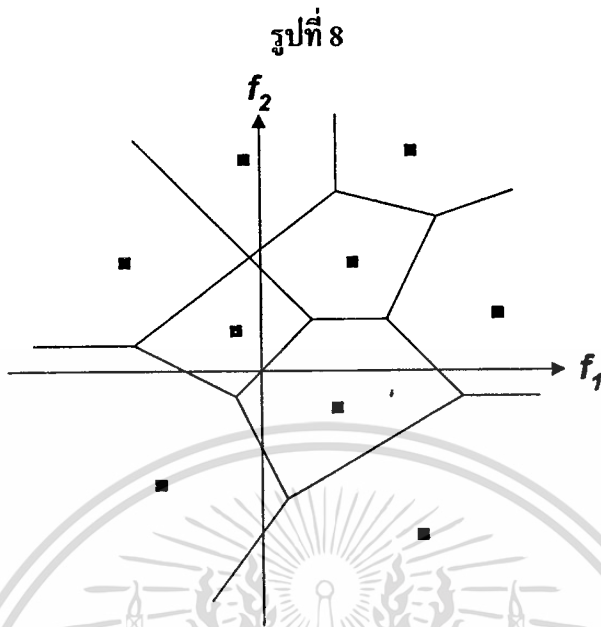
$$\hat{f} = VQ(f) = r_i; \quad f \in C_i, 1 \leq i \leq L$$

$VQ(\cdot)$ Vector Quantization Operation

r_i Reconstruction Vector ซึ่งเป็นค่าหนึ่งในจำนวน L ค่า

C_i กลุ่มข้อมูล (Cell) ที่ i

รูปที่ 8 เป็นตัวอย่างของ VQ ที่มีจำนวนมิติ (N) = 2 และ Reconstruction Level : $L = 9$ โดยจุดในช่องต่างๆคือค่า Reconstruction Vector ส่วนเส้นทึบคือขอบเขตของกลุ่มข้อมูล (Cell Boundaries) จะเห็นว่า"ขนาด" (Size) และ"รูปร่าง" (Shape) ของกลุ่มข้อมูลนั้นมีลักษณะต่าง ๆ กันเป็นข้อเปรียบเทียบชัดเจนกับการควอนไทซ์แบบสเกลาร์ที่สามารถเปลี่ยนแปลงได้เฉพาะ "ขนาด" เท่านั้น แต่"รูปร่าง"ไม่เปลี่ยนแปลงเพราะมีเพียงมิติเดียว



แสดงตัวอย่างของ VQ ที่มีจำนวนมิติ (N) = 2 และ Reconstruction Level (L) = 9

1. การวัดประสิทธิภาพของการควอนไทซ์แบบเวกเตอร์

การวัดประสิทธิภาพของ Vector Quantization : VQ มีหลักการเช่นเดียวกับการวัดประสิทธิภาพ Scalar Quantization : SQ คือการวัดค่าความคลาดเคลื่อนระหว่างฟังก์ชันเดิม f และฟังก์ชันที่สร้างขึ้นแทน \hat{f} เขียนเป็นสมการดังนี้

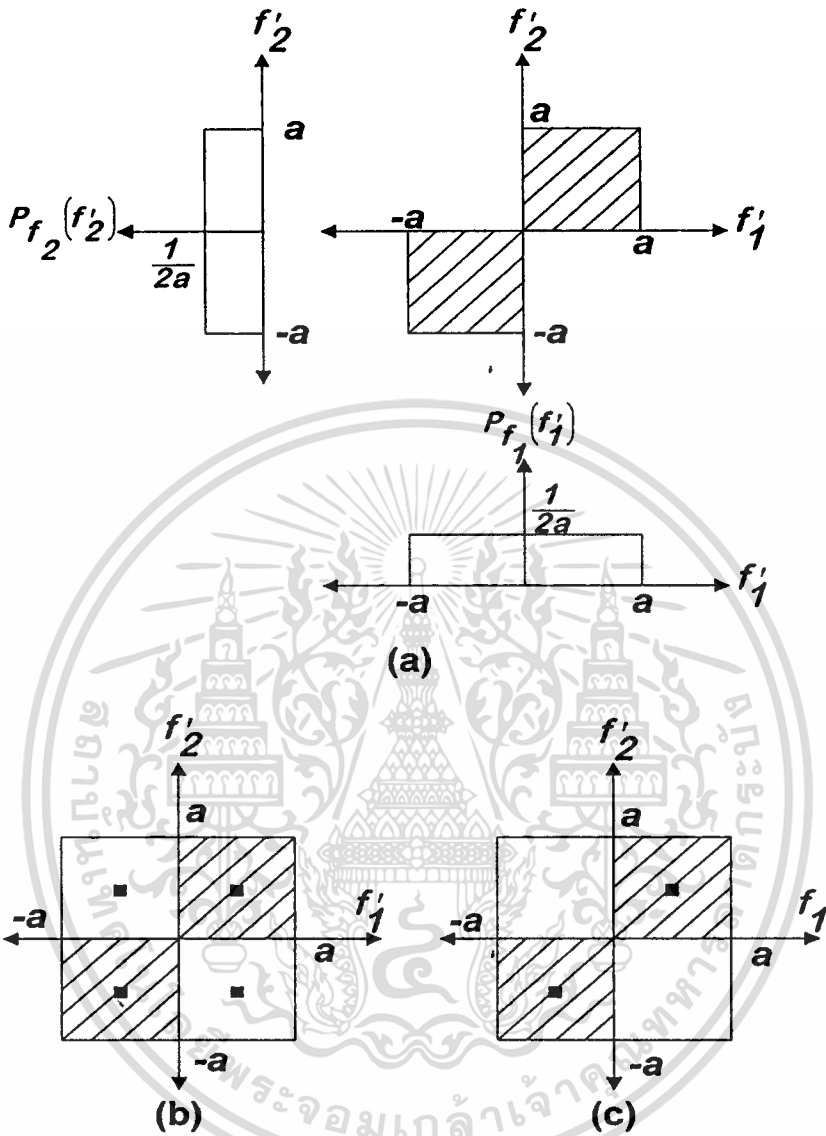
$$d(f, \hat{f}) = e_Q^T \cdot e_Q$$

$$e_Q : \text{Quantization Error} = \hat{f} - f = VQ(f) - f$$

2. ข้อได้เปรียบของการควอนไทซ์แบบเวกเตอร์

เมื่อเราเปรียบเทียบระหว่าง VQ และ SQ สิ่งที่เราเห็นได้ชัดก็คือการควอนไทซ์แบบเวกเตอร์สามารถลด "ค่าความคลาดเคลื่อนเฉลี่ย" ลงได้โดยที่ใช้จำนวน Reconstruction Level เท่ากับการควอนไทซ์แบบสเกลาร์ หรือยังคง "ค่าความคลาดเคลื่อนเฉลี่ย" ไว้ได้แต่ใช้จำนวน Reconstruction Level น้อยกว่าการควอนไทซ์แบบสเกลาร์ เพื่อความชัดเจนจึงขอยกตัวอย่างต่อไปนี้

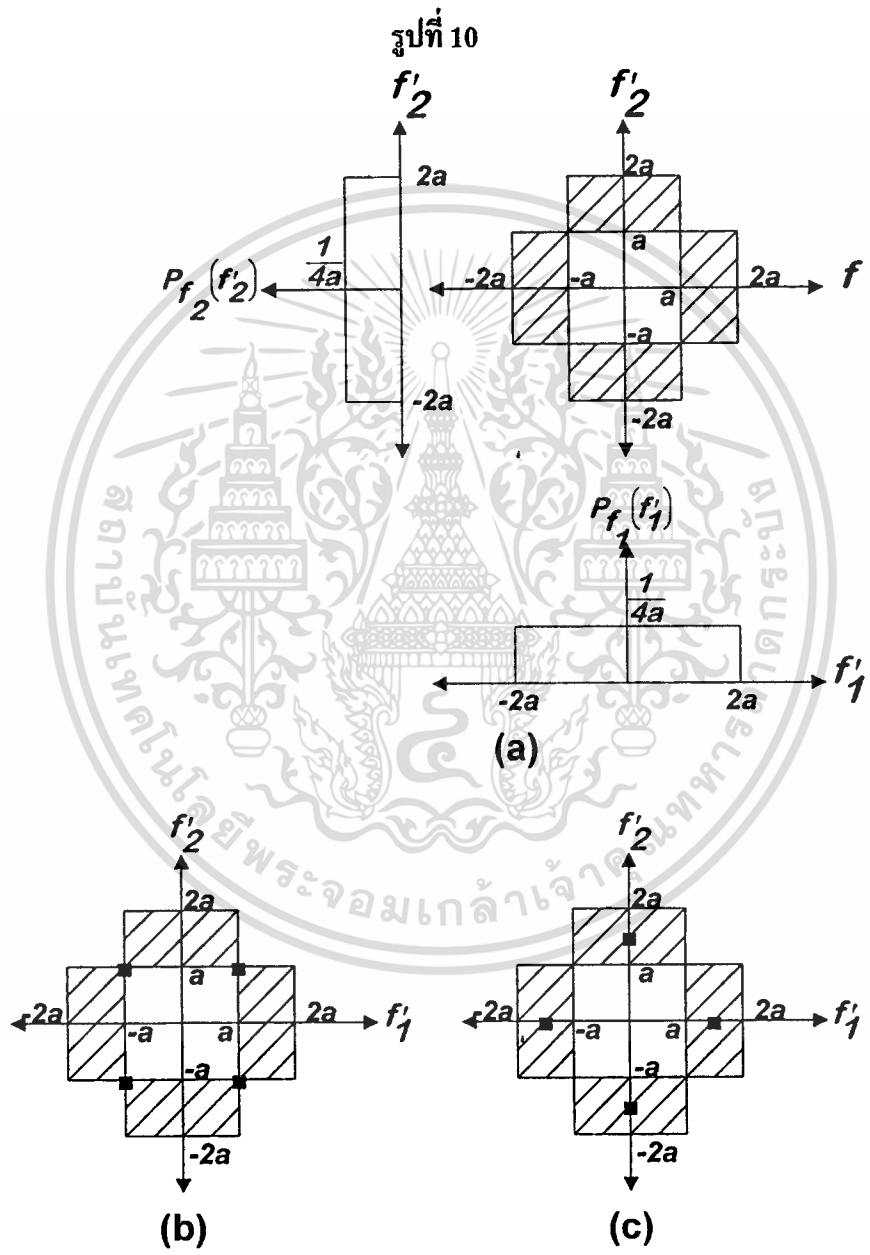
รูปที่ 9



แสดงข้อเปรียบเทียบระหว่าง VQ และ SQ กับข้อมูลที่มีความไม่อิสระเชิงเส้น (Linear Dependence)

ในรูปที่ 9 เป็นตัวอย่างของความไม่อิสระของข้อมูลแบบเชิงเส้น (Linear Dependence) โดยรูปที่ 9a เป็นกราฟความสัมพัทธ์ของฟังก์ชัน Joint Probability Density Function (pdf) $P_{f_1, f_2}(f'_1, f'_2)$ โดยพื้นที่ส่วนที่มีลายเส้นมีแอมพลิจูดเท่ากับ $1/2a^2$ และส่วนที่ไม่มีลายเส้นมีค่าแอมพลิจูดเท่ากับ "0" และมีค่า Marginal pdf คือ $p_{f_1}(f'_1)$ และ $p_{f_2}(f'_2)$ ดังในรูป เนื่องจากค่า $E[f_1, f_2] \neq E[f_1].E[f_2]$ แสดงว่า f_1 และ f_2 มีความไม่อิสระเชิงเส้นต่อกัน (Linear Dependence) หากนำข้อมูลฟังก์ชันทั้งสองมาควอนไทซ์แบบสเตลาร์แยกจากกันภายใต้เงื่อนไข MMSE โดยการควอนไทซ์แบบยูนิฟอร์ม กำหนดให้มี Reconstruction Level = 2 ต่อฟังก์ชัน จะได้ผลลัพธ์มีค่าเท่ากับ "a/2" และ "-a/2" เกิดขึ้นจำนวน 4 จุด (2x2) ใน 4 Quadrant ดังรูปที่ 9b สังเกตว่าจะมีค่าเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Reconstruction ที่ไม่ได้เป็นตัวแทนข้อมูล (ไม่ได้ใช้ประโยชน์) จำนวน 2 ตัว แต่หากนำข้อมูลชุดเดียวกันนี้มาควอนไทซ์แบบเวกเตอร์ก็จะได้ Reconstruction ที่เป็นตัวแทนข้อมูลอย่างถูกต้องเพียง 2 ตัว (รูปที่ 9c) กล่าวได้ว่าการควอนไทซ์ข้อมูลแบบเวกเตอร์ในกรณีนี้ทำให้ลดจำนวน Reconstruction Level ลงได้โดยที่"ค่าความคลาดเคลื่อน" ไม่เปลี่ยนแปลง



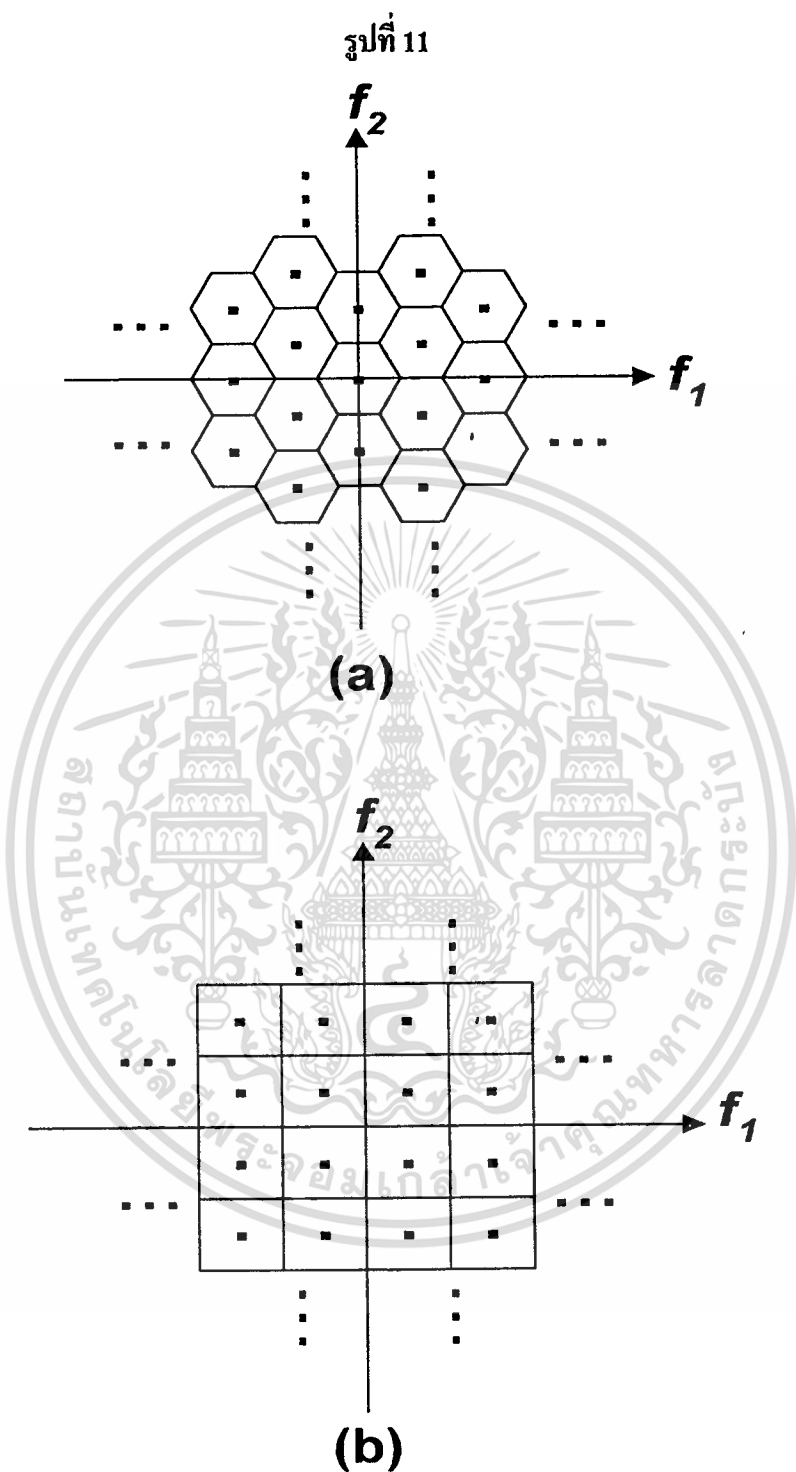
แสดงข้อเปรียบเทียบระหว่าง VQ และ SQ กับข้อมูลที่มีความไม่อิสระแบบไม่เป็นเชิงเส้น (Nonlinear Dependence)

รูปที่ 10a เป็นกราฟความสัมพันธ์ของฟังก์ชัน Joint Probability Density Function (pdf) ของ f_1 และ f_2 โดยพื้นที่ส่วนที่มีลายเส้นมีแอมพลิจูดเท่ากับ $1/(8a^2)$ และส่วนที่ไม่มีลายเส้นมีค่าไม่ต่ำกว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แอมพลิจูดเท่ากับ "0" , มีค่า Marginal pdf คือ $p_{f_1}(f_1)$ และ $p_{f_2}(f_2)$ ดังรูป จะได้ว่าค่า $E[f_1, f_2] = E[f_1] \cdot E[f_2]$ มีอิสระเชิงเส้นต่อกัน (Linearly Independence) แต่ค่า $p_{f_1, f_2}(f_1, f_2) \neq p_{f_1}(f_1) p_{f_2}(f_2)$ ซึ่งแสดงว่า f_1 และ f_2 ยังมีความไม่อิสระต่อกัน (Statistically Dependence) เมื่อรวมเงื่อนไขทั้งสองนี้ด้วยกันจึงสรุปว่า f_1 และ f_2 มีความไม่อิสระชนิดที่ไม่เป็นเชิงเส้นต่อกัน (Nonlinearly Dependence) การนำฟังก์ชันทั้งสองมาควอนไทซ์แบบสเตลาร์แยกจากกันภายใต้เงื่อนไข MMSE โดยการควอนไทซ์แบบยูนิฟอร์ม กำหนดให้มี Reconstruction Level = 2 ต่อฟังก์ชัน จะได้ผลลัพธ์มีค่าเท่ากับ "a" และ "-a" เกิดขึ้นจำนวน 4 จุด (2x2) ใน 4 Quadrant ดังรูปที่ 10b เทียบกับการควอนไทซ์แบบเวกเตอร์ที่ได้ผลลัพธ์จำนวน 4 จุดเช่นกันในรูปที่ 10c แต่เมื่อมาคำนวณค่าความคลาดเคลื่อนเฉลี่ย ค่าที่ได้จากการควอนไทซ์แบบสเตลาร์เท่ากับ " $a^2/2$ " ขณะที่ค่าที่ได้จากการควอนไทซ์แบบเวกเตอร์เท่ากับ " $5a^2/12$ " จึงกล่าวได้ว่ากรณีนี้การควอนไทซ์แบบเวกเตอร์ให้ "ค่าความคลาดเคลื่อนเฉลี่ย" น้อยกว่าการควอนไทซ์แบบสเตลาร์ เมื่อมีจำนวน Reconstruction Level เท่ากัน

ในบางกรณีเราสามารถตัดความสัมพันธ์ระหว่างข้อมูลภายในออกไปได้ด้วยวิธีการ Linear Transform เพื่อตัดความไม่อิสระแบบเชิงเส้นออกไป หรือใช้ Nonlinear Operation สำหรับตัดความไม่อิสระแบบไม่เป็นเชิงเส้น ส่งผลให้ข้อมูลที่ผ่านกระบวนการดังกล่าวไม่มีความสัมพันธ์ต่อกันเลย ตัวอย่างในรูปที่ 11 สมมุติว่า f_1 และ f_2 เป็นอิสระต่อกัน (Statistically Independence) และกำหนดให้มีค่า Reconstruction Level จำนวนมากๆ เมื่อเรานำข้อมูลนี้มาควอนไทซ์แบบสเตลาร์แล้วจะได้รูปร่างกลุ่มข้อมูล (Cell) มีลักษณะเป็นสี่เหลี่ยมจัตุรัสยาวด้านละ a หน่วยดังรูปที่ 11a แต่เมื่อนำมาควอนไทซ์แบบเวกเตอร์กลุ่มข้อมูลที่ได้จะมีลักษณะเป็น"รูปหกเหลี่ยม" ดังรูปที่ 11b ภายหลังจากการคำนวณแล้วพบว่า "ค่าความคลาดเคลื่อนเฉลี่ย" ที่ได้จากการควอนไทซ์แบบเวกเตอร์นี้ต่ำกว่าแบบที่ได้จากการควอนไทซ์แบบสเตลาร์อยู่ 4 % และเมื่อกำหนดให้ "ค่าความคลาดเคลื่อนเฉลี่ย" เท่ากัน VQ ก็ยังใช้จำนวน Reconstruction Level น้อยกว่า SQ อยู่ 2 % และจะดีกว่านี้หากจำนวนสเตลาร์ต่อเวกเตอร์ (มิติ) เพิ่มขึ้น ตัวอย่างนี้แสดงลักษณะของข้อมูลที่ไม่มีความสัมพันธ์กันซึ่งผลเปรียบเทียบระหว่างการควอนไทซ์ด้วยวิธีทั้งสองจะต่างกันน้อยมาก ซึ่งชี้ให้เห็นว่าการควอนไทซ์แบบเวกเตอร์นั้นเหมาะสมกับข้อมูลที่มีความสัมพันธ์กัน (ไม่ว่าจะเป็นเชิงเส้นหรือไม่เป็นเชิงเส้นก็ตาม) เช่น ความเข้มของจุดภาพ (Image Intensity) มากกว่าข้อมูลที่ไม่มีความสัมพันธ์กัน เช่น สัมประสิทธิ์ของการแปลงข้อมูลเป็นต้น

จากตัวอย่างทั้งหมดที่กล่าวมานี้สรุปได้ว่าการควอนไทซ์แบบเวกเตอร์จะให้ผลที่ดีกว่าการควอนไทซ์แบบสเตลาร์ไม่ว่าข้อมูลจะมีความสัมพันธ์กันในลักษณะใดก็ตาม



แสดงข้อเปรียบเทียบระหว่าง VQ และ SQ กับข้อมูลที่เป็นอิสระต่อกัน (Statistically Independence)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การออกแบบตารางเปรียบเทียบข้อมูล

บทนำ

จากบทที่ 2 ซึ่งแสดงให้เห็นถึงประโยชน์และข้อได้เปรียบของการควอนไทซ์ข้อมูลแบบเวกเตอร์ เมื่อพิจารณาแล้วจะพบว่าตัวแปรหรือค่าพารามิเตอร์ที่มีความสำคัญอย่างยิ่งของกระบวนการนี้ก็คือ "ตารางเปรียบเทียบข้อมูล" (Codebook) ภายในประกอบไปด้วย "ค่าตัวแทนข้อมูล" (Codeword) ซึ่งเป็นเวกเตอร์ข้อมูลที่ใช้แทนข้อมูลจริง (Reconstruction Vector) ความสำคัญของตารางเปรียบเทียบนี้ก็คือเป็นตารางเปรียบเทียบกับข้อมูลจริง (Input Vector) โดยตัวเข้ารหัส (Encoder) จะเลือกค่าตัวแทนข้อมูลที่มีค่าใกล้เคียงกับข้อมูลจริงมากที่สุด และจะส่งเฉพาะค่าของ "ดัชนี" (Index) ที่ชี้บอกตำแหน่งของค่าตัวแทนข้อมูลนั้นไปแทน ทางด้านตัวถอดรหัส (Decoder) จะนำค่าดัชนีนี้ไปชี้ตำแหน่งของค่าตัวแทนข้อมูลภายในตัวเอง (เป็นตารางชุดเดียวกับตัวเข้ารหัส) เพื่อนำค่าที่ถูกชี้ขึ้นไปแทนข้อมูลจริงต่อไป

จากกระบวนการที่กล่าวมาข้างต้นเป็นที่มาของเนื้อหาในบทนี้กล่าวคือ วิธีสร้างตารางเปรียบเทียบข้อมูลนี้ได้อย่างไร และมีวิธีใดที่ทำให้ค่าของตัวแทนข้อมูลนั้นมีค่าใกล้เคียงกับข้อมูลจริงมากที่สุด (มีค่าคลาดเคลื่อนน้อยที่สุด)

ข้อกำหนดการออกแบบตารางเปรียบเทียบข้อมูล

ก่อนปลายทศวรรษที่ 70 การนำวิธีการควอนไทซ์แบบเวกเตอร์ไปประยุกต์ใช้ในงานด้านต่างๆ เช่น การเข้ารหัสเสียง (Speech Coding) หรือ การประมวลผลภาพ (Image Processing) ยังมีน้อยมากเพราะสาเหตุการออกแบบตารางเปรียบเทียบข้อมูลที่ยุ่งยาก

กำหนดให้ f : Input Vector

และ r_i : Codeword ตำแหน่งที่ i

สมมติให้ Codebook นี้มีค่า Codeword จำนวน L ค่า

VQ : Vector Quantization Operation

C_i : กลุ่มข้อมูลกลุ่มที่ i

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การออกแบบ Codebook เพื่อให้ผลลัพธ์มี "ค่าความคลาดเคลื่อนเฉลี่ย" ต่ำที่สุดจะต้องเป็นไปตามเงื่อนไขสองประการดังนี้

1. ค่าของ Reconstruction Vector หรือ Codeword: r_i ตัวที่จะมาแทนข้อมูลจริงนั้น จะต้องเป็นค่าที่มีความคลาดเคลื่อนจากข้อมูลจริงตัวนั้นน้อยที่สุด เมื่อเทียบกับ Codeword ตัวอื่นที่เหลือใน Codebook นั้น (ในที่นี้มีจำนวนทั้งหมด L ค่า) เขียนเป็นสมการได้ดังนี้

$$VQ(f) = r_i \quad \text{โดยที่} \quad d(f, r_i) \leq d(f, r_j); j \neq i, 1 \leq i, j \leq L$$

2. ค่า Reconstruction Vector หรือ Codeword: r_i ซึ่งเป็นตัวแทนข้อมูลกลุ่ม (Cell) ที่ i จะต้องทำให้ค่าความคลาดเคลื่อนเฉลี่ยของข้อมูลกลุ่มนี้ต่ำที่สุด

$$\text{Minimum } E[d(f, r_i) | f \in C_i]$$

ค่า r_i ที่อยู่ภายใต้เงื่อนไขนี้เรียกว่า "จุดกึ่งกลางข้อมูล" (Centroid of C_i)

อธิบายได้ว่าเงื่อนไขข้อที่หนึ่งคือการควอนไทซ์ข้อมูล f โดยไม่สนใจกลุ่มข้อมูล C_i ใดๆ หรือกล่าวอีกนัยหนึ่งว่าเป็นการเปรียบเทียบข้อมูลอินพุตกับ Codeword ที่มีอยู่ทุกตัว และเงื่อนไขข้อที่สองก็คือการหาค่า Reconstruction Vector r_i หรือ Codeword เฉพาะกลุ่มข้อมูล (C_i) นั้นๆ ซึ่งเป็นการหาค่าตำแหน่ง Centroid ของกลุ่มนั่นเอง การคำนวณหาค่า Codebook ต้องอยู่ภายใต้เงื่อนไขทั้งสองข้อนี้จึงจะให้ผลดีที่สุด (Optimum Solution) อย่างไรก็ตามเงื่อนไขแต่ละข้อก็ไม่ได้เป็นอิสระต่อกัน เพราะค่า Codeword เองก็เป็นตัวกำหนดกลุ่มข้อมูล และกลุ่มข้อมูลนั้นก็เป็นตัวกำหนดค่า Codeword อีกที ดังนั้นการรู้เพียงค่าใดค่าหนึ่งอย่างเดียวก็น่าจะเพียงพอที่ใช้ในการคำนวณได้

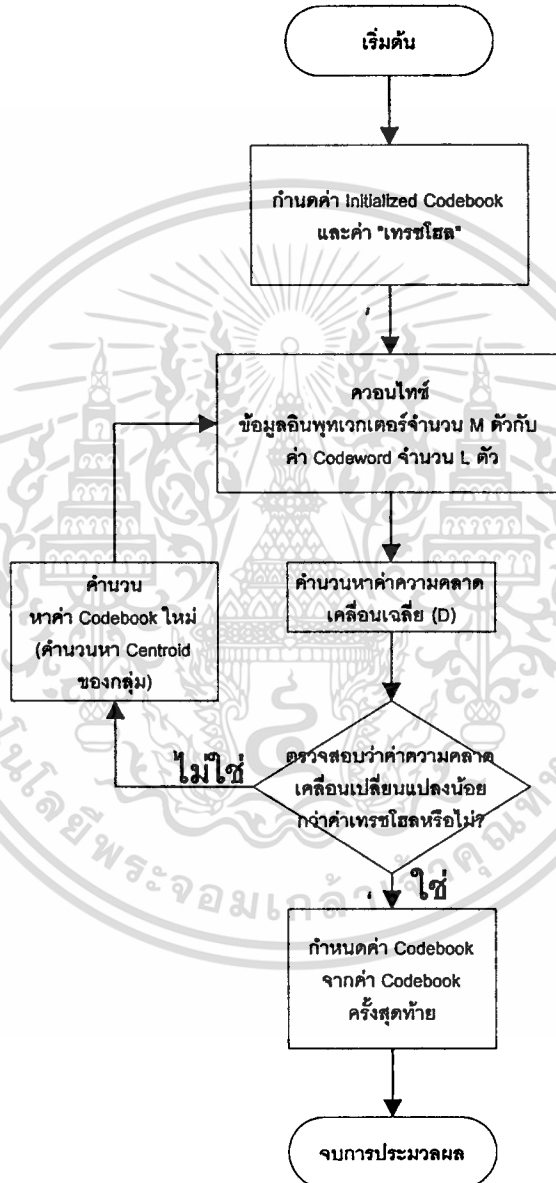
วิธีการของ LBG

จากเงื่อนไขที่กล่าวมานำไปสู่การคำนวณแบบซ้ำๆ (Iterative) เริ่มจากกำหนดค่าเริ่มต้นของ Codebook ก่อน แล้วจึงใช้เงื่อนไขข้อที่หนึ่งสำหรับแบ่งกลุ่มข้อมูล เมื่อได้ข้อมูลเป็นกลุ่มๆ แล้วก็ใช้เงื่อนไขข้อที่สองหาค่า Codebook ขึ้นมาใหม่ จากนั้นจึงวนกลับไปทำเงื่อนไขข้อที่หนึ่งอีกและทำอย่างนี้ต่อเนื่องไปเรื่อยๆ วิธีการนี้เรียกว่า "K-means algorithm" ถูกคิดค้นโดย Lloyd ในปี 1957 ซึ่งเขาได้อธิบายในลักษณะข้อมูลสเกลาร์ ($N=1$) ต่อมาในปี 1965 Forgy ได้นำมาอธิบายอีกในลักษณะข้อมูลแบบเวกเตอร์ จนในปีค.ศ. 1980 วิธีการนี้ถูกเรียกชื่อใหม่ว่า "LBG Algorithm" เพื่อเป็นเกียรติ

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับอาจารย์และบุคลากรในภาควิชาวิศวกรรมโทรคมนาคม คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แก่ Y. Linde, A. Buzo and R.M. Gray ซึ่งแสดงให้เห็นว่าวิธีการนี้สามารถนำมาใช้กับงานลักษณะต่างๆ ได้ โดยเฉพาะการเข้ารหัสเสียงและภาพ หลังจากนั้นมากก็มีงานวิจัยเรื่อง VQ ที่อ้างอิงงานวิจัยนี้ประยุกต์ใช้ในด้านอื่นอย่างแพร่หลาย

แผนภูมิที่ 12



แสดงแผนภูมิการทำงานหาค่า Codebook ด้วยวิธี LBG's

แผนภูมิที่ 12 แสดงแผนภูมิการทำงานของวิธี LBG อย่างละเอียด สมมุติว่าเรามี Input Vector f จำนวน M ตัว $f_i: 1 \leq i \leq M$ และกำหนดให้มี Codeword จำนวน L ตัว $r_j: 1 \leq j \leq L$ และโดยปกติแล้วค่า $M \gg L$ โดย M อาจมีค่าเป็น 10 เท่า, 50 เท่าของ L หรือมากกว่านั้นก็ได้ การคำนวณหาค่า Codeword นี้จะอยู่ภายใต้เงื่อนไขค่าความคลาดเคลื่อนเฉลี่ยต่ำที่สุด ใช้สูตรคำนวณคือด้านการคำนวณว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$D = \frac{1}{M} \sum_{i=1}^M d(f_i, \hat{f}_i)$$

ขั้นตอนมีดังนี้

1. กำหนดค่าเริ่มต้น (Initialize) ของ Reconstruction Vector : r_i และค่าเทรซโฮล : th ที่ใช้สำหรับตรวจสอบนัยสำคัญของการเปลี่ยนแปลงค่าความคลาดเคลื่อนเฉลี่ย

2. นำค่า Input Vector : f_i ที่ละค่ามาเปรียบเทียบกับค่า r_i ทุกค่า และตรวจสอบว่าค่า r_i ตัวใดให้ค่าความคลาดเคลื่อนต่ำที่สุด จากนั้นให้จัด f_i เข้ากลุ่มของ r_i ตัวนั้นๆ (ข้อนี้คือการปฏิบัติตามเงื่อนไขข้อที่หนึ่ง)

3. หลังจากข้อ 2 แล้วข้อมูลจะถูกแบ่งได้เป็น L กลุ่ม (เท่ากับจำนวน Codeword) คำนวณหาค่า Codeword นี้ใหม่ (Optimize) โดยการหาค่าเฉลี่ย (หรือตำแหน่ง Centroid) ของข้อมูลแต่ละกลุ่มใหม่ (ข้อนี้คือการปฏิบัติตามเงื่อนไขข้อที่สอง)

4. คำนวณหาค่าความคลาดเคลื่อนเฉลี่ย : D

5. ตรวจสอบเงื่อนไขว่าอัตราการเปลี่ยนแปลงค่าความคลาดเคลื่อนเฉลี่ยน้อยกว่าค่าเทรซโฮลที่กำหนดหรือไม่ หากยังคงมากกว่าก็ย้อนกลับมาทำตั้งแต่ข้อที่สองใหม่ แต่หากน้อยกว่าแล้วก็ให้ยุติการคำนวณและถือเอาค่า Codeword ครั้งสุดท้ายมาใช้งาน ได้

จะสังเกตว่า Algorithm ที่ใช้นี้ตั้งอยู่บนสมมุติฐานที่ว่าค่าความคลาดเคลื่อนเฉลี่ยในครั้งใหม่จะน้อยกว่าในครั้งก่อนหน้า (Convergence) ดังนั้นการกำหนดค่าเริ่มต้นของ Reconstruction Vector : r_i และค่าเทรซโฮล : th จะมีความสำคัญมากและมีผลต่อจำนวนรอบการคำนวณ และค่าความคลาดเคลื่อนที่ได้ ทั้งนี้ยังมีปัญหาใหญ่ที่เกิดจากการคำนวณลักษณะนี้ก็คือ ปริมาณการคำนวณมหาศาลที่ต้องใช้ในแต่ละรอบ ดังเช่น

กำหนดให้ N จำนวนสเกลาร์ (มิติ) ในแต่ละเวกเตอร์

R จำนวนบิต/มิติ ที่ต้องการ

L จำนวน Codeword

จะได้รับความสัมพันธ์ระหว่างตัวแปรทั้งสามดังนี้

$$L = 2^{NR}$$

สมมุติว่าเราต้องการให้ได้จำนวนบิต/มิติของข้อมูล (R) ภายหลังจากควอนไทซ์เท่ากับ 0.5 บิต/มิติ (หากมองในแง่ของการประมวลผลภาพเทียบได้กับการใช้พื้นที่เก็บข้อมูล 0.5 บิต/พิกเซล) และมีข้อมูลสเกลาร์จำนวน 16 ตัว (N) ในหนึ่งเวกเตอร์ ดังนั้นจำนวน Codeword (L) ที่ต้องสร้างขึ้นเท่ากับ $2^{0.5 \times 16} = 2^8 = 256$ ค่า ในการคำนวณ 1 รอบเราต้องเปรียบเทียบค่า Input Vector ทั้งหมด M ตัวกับค่า Codeword ทีละค่าทั้งหมด L ตัว เป็นการคำนวณหา "ค่ากำลังสองของความคลาดเคลื่อน" ($e_q^T \cdot e_q$) การเปรียบเทียบข้อมูลกัน 1 ชุดมีการคำนวณเท่ากับ N ครั้ง (การคำนวณ 1 ครั้งคือการคูณกัน 1 ครั้งและการบวกกัน 1 ครั้ง) ดังนั้นใน 1 รอบการคำนวณ (Iterative) ต้องใช้การคำนวณทั้งสิ้น

$$NML = NM \cdot 2^{NR} \text{ ครั้ง}$$

สมการนี้จะมีความหมายมากขึ้นเมื่อลองสมมุติตัวอย่างข้อมูลเวกเตอร์ขนาด 10 มิติ (N) กำหนดให้แต่ละมิติใช้พื้นที่ 2 บิต (R) และจำนวน Input Vector (M) มีจำนวนเป็น 10 เท่าของจำนวน Codebook (L) ปริมาณการคำนวณที่เกิดขึ้นในหนึ่งรอบคือ

$$NML = N(10L)L = 10NL^2 = 10N \cdot (2^{NR})^2 = 10N \cdot 2^{2NR}$$

$$\text{แทนค่า } N = 10 \text{ และ } R = 2$$

$$10N \cdot 2^{2NR} = 10(10) \cdot 2^{2(10)(2)} = 100 \times 2^{40} \approx 10^{14} \text{ (หนึ่งร้อย ล้านล้าน) ครั้ง}$$

นอกจากการคำนวณอย่างมหาศาลแล้ว พื้นที่ที่ใช้เก็บค่า Input Vector และ Codebook ก็เป็นปัจจัยที่ต้องพิจารณาเช่นกัน หากเราใช้พื้นที่ 1 หน่วยสำหรับเก็บข้อมูล 1 มิติ จะต้องใช้พื้นที่ MN หน่วย สำหรับเก็บ Input Vector และ LN หน่วย สำหรับเก็บ Codeword พื้นที่ทั้งหมดที่ใช้คือ

$$MN + LN = (M + L)N \approx MN ; M \gg L$$

$$\text{แทนค่า } N = 10, R = 2 \text{ และ } M = 10L$$

$$MN = 10LN = 10N \cdot 2^{NR} = 10(10) \cdot 2^{(10)(2)} = 100 \times 2^{20} \text{ หน่วย}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างการคำนวณทั้งสองนี้แสดงให้เห็นว่า VQ ไม่เหมาะในระบบขนาดเล็กที่มีหน่วยความจำและความเร็วในการประมวลผลจำกัด และแม้ว่าจะหาค่า Codebook ได้แล้ว การนำไปใช้งานก็ต้องเก็บค่า Codebook นี้ไว้ทั้งในส่วนของตัวเข้ารหัสและตัวถอดรหัสด้วย โดยใช้พื้นที่เก็บเท่ากับ

$$NL = N \cdot 2^{NR} \text{ หน่วย}$$

$$\text{แทนค่า } N = 10, R = 2$$

$$N \cdot 2^{NR} = 10 \times 2^{(10)(2)} = 10 \times 2^{20} \text{ หน่วย}$$

ขณะทำงานตัวเข้ารหัสจะต้องคำนวณเพื่อหาค่า "ดัชนี" ของ Codebook เพื่อส่งออกไปสำหรับ Input Vector 1 ตัวที่เข้ามาเท่ากับ

$$NL = N \cdot 2^{NR} \text{ ครั้ง}$$

ยังมีข้อมูลที่ต้องส่ง/เข้ารหัสมากขึ้นเท่าใด การคำนวณก็จะทวีคูณมากขึ้นเท่านั้น จึงเป็นเงื่อนไขอีกข้อที่ทำให้ตัวส่งหรือตัวเข้ารหัสต้องมีความเร็วในการประมวลผลมาก แต่สำหรับฝ่ายตัวรับหรือตัวถอดรหัสจะใช้เพียงนำค่า "ดัชนี" ที่ได้มาชี้ค่าของ Codeword นั้นๆ และส่งค่าออกไป ไม่ต้องอาศัยความเร็วการคำนวณเป็นพิเศษแต่อย่างใด

ตัวอย่างที่ชี้ให้เห็นทั้งหมดนี้สังเกตว่าตัวเลขการคำนวณหรือปริมาณหน่วยความจำที่ใช้จะแปรผันแบบเอ็กโพเนนเชียลกับค่าของ R และ N ดังนั้นการออกแบบระบบโดยคำนึงถึงตัวแปรทั้งสองจะช่วยลดตัวเลขต่างๆลงได้

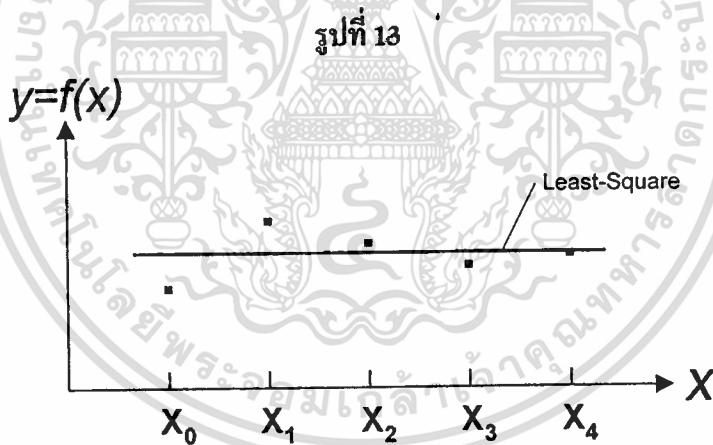
บทที่ 4

การประมาณค่าฟังก์ชันโดยวิธี Least-Square

บทนำ

แม้ว่าวิทยานิพนธ์ฉบับนี้จะใช้การประมาณค่าแบบสมการโพลิโนเมียลลำดับที่ 1 (1st Order Polynomial) มาใช้พิจารณาแทนข้อมูล แต่ผู้วิจัยได้ใส่เนื้อหาของการประมาณค่าแบบสมการโพลิโนเมียลลำดับที่ m และแบบเอ็กโพเนนเชียลไว้ด้วย ซึ่งอาจเป็นประโยชน์แก่ผู้ที่สนใจซึ่งสามารถพิจารณาข้อมูลในรูปแบบอื่นได้

หลักการประมาณค่าแบบ Least-Square



แสดงลักษณะการประมาณค่าแบบ Least-Square

สมมุติว่าเรามีความสัมพันธ์ของข้อมูลอยู่ n ชุดคือ

$$[x_i, f(x_i)]; i = 1, 2, 3, \dots, n$$

ถ้าเราประมาณฟังก์ชันของข้อมูลชุดนี้ในระหว่าง x_1 ถึง x_n ด้วยสมการ โพลิโนเมียล

$$p(x) = a_0 + a_1x + a_2x^2 + \dots + a_mx^m + \dots \quad (1)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะได้ผลต่างของค่าฟังก์ชันจริง $f(x_i)$ และค่าของฟังก์ชันที่ได้จากการประมาณ $p(x_i)$ ที่ตำแหน่ง x_i เป็น

$$r_i = f(x_i) - p(x_i); i = 1, 2, 3, \dots, n \quad (2)$$

ค่า r นี้เรียกว่า Residual คือส่วนที่เบี่ยงเบนไปจากค่าจริง และเมื่อนำค่า r_i ที่ตำแหน่งต่างๆ $i = 1, 2, 3, \dots, n$ มาหาค่าผลรวมของส่วนเบี่ยงเบนยกกำลังสองจะได้เป็น

$$Q = \sum_{i=1}^n r_i^2 = \sum_{i=1}^n [f(x_i) - p(x_i)]^2 \quad (3)$$

หลักการของวิธีนี้ก็คือการหาค่าสัมประสิทธิ์ a_j ($j = 1, 2, 3, \dots, n$) ตามแต่ลำดับของ โพลีโนเมียล $p(x_i)$ ที่ทำให้ Q มีค่าน้อยที่สุด

การประมาณค่าโดยใช้สมการโพลีโนเมียลลำดับที่หนึ่ง

สมมุติว่ามีความสัมพันธ์ของข้อมูลอยู่ n ชุดคือ

$$[x_i, f(x_i)]; i = 1, 2, 3, \dots, n$$

การประมาณข้อมูลด้วยสมการโพลีโนเมียลลำดับที่ 1 มีรูปแบบสมการดังนี้

$$p_1(x) = a_0 + a_1x \quad (4)$$

ค่าสัมประสิทธิ์ a_0 และ a_1 ที่ทำให้ค่า Q ในสมการที่ (3) มีค่าน้อยที่สุดได้จากการแก้สมการ Derivative ของ $Q(a_0, a_1)$ เทียบกับ a_0 และ a_1 แล้วให้เท่ากับศูนย์ดังนี้

$$\frac{\partial Q(a_0, a_1)}{\partial a_0} = -2 \sum_{i=1}^n [y_i - (a_0 + a_1x)] = 0$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หรือ
$$a_0 n + a_1 \sum_{i=1}^n x_i = \sum_{i=1}^n y_i \quad (5)$$

และ
$$\frac{\partial Q(a_0, a_1)}{\partial a_1} = -2 \sum_{i=1}^n x_i [y_i - (a_0 + a_1 x)] = 0$$

หรือ
$$a_0 \sum_{i=1}^n x_i + a_1 \sum_{i=1}^n x_i^2 = \sum_{i=1}^n y_i x_i \quad (6)$$

จาก (5) และ (6) แก้สมการหาค่า a_0 และ a_1 ได้ดังนี้

$$a_1 = \frac{n \sum_{i=1}^n y_i x_i - \sum_{i=1}^n x_i \sum_{i=1}^n y_i}{n \sum_{i=1}^n x_i^2 - \left(\sum_{i=1}^n x_i \right)^2} \quad (7.1)$$

$$a_0 = \frac{\sum_{i=1}^n y_i - a_1 \sum_{i=1}^n x_i}{n} \quad (7.2)$$

การประมาณค่าโดยใช้สมการโพลิโนเมียลลำดับที่ m

ลักษณะของสมการ โพลิโนเมียลลำดับที่ m จะเป็นดังนี้

$$p(x) = a_0 + a_1 x + a_2 x^2 + \dots + a_m x^m \quad (8)$$

การหาค่าสัมประสิทธิ์ที่ใช้หลักการเดียวกับแบบสมการ โพลิโนเมียลลำดับที่หนึ่งคือได้จากการแก้สมการที่เกิดจากการแทนค่า

$$\frac{\partial Q}{\partial a_j} = 0; j = 0, 1, 2, \dots, m \quad (9)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งจะได้ระบบสมการดังนี้

$$\begin{aligned} a_0 n + a_1 \sum x_i + \dots + a_m \sum x_i^m &= \sum y_i \\ a_0 \sum x_i + a_1 \sum x_i^2 + \dots + a_m \sum x_i^{m+1} &= \sum y_i x_i \\ &\dots \\ a_0 \sum x_i^m + a_1 \sum x_i^{m+1} + \dots + a_m \sum x_i^{2m} &= \sum y_i x_i^m \end{aligned}$$

เมื่อได้ชุดเมตริกซ์จำนวน $m+1$ สมการ และ $m+1$ ตัวแปรจึงแก้สมการหาคำตอบได้ด้วยวิธี Modify Cholesky Decomposition Method เพราะว่าสัมประสิทธิ์ของเมตริกซ์มีลักษณะสมมาตร

การประมาณค่าโดยใช้สมการเอ็กโพเนนเชียล

รูปแบบของสมการจะเป็นลักษณะไม่เป็นเชิงเส้นดังนี้

$$p_e(x) = a \cdot e^{bx} \quad (10)$$

จึงต้องแปลงรูปสมการก่อนโดยการ Take Log ทั้งสองข้าง

$$\ln p_e(x) = \ln a + bx \quad (11)$$

รูปแบบสมการจะเปลี่ยนเป็นแบบโพลิโนเมียลลำดับที่หนึ่งทันที โดย

$$p_1(x) = \ln p(x)$$

$$a_0 = \ln a$$

$$a_1 = b$$

จากนั้นจึงใช้วิธีการหาคำตอบโดยใช้วิธีการประมาณค่าโดยใช้สมการโพลิโนเมียลลำดับที่หนึ่งต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

การประมาณค่าเชิงเส้นในกระบวนการควอนไทซ์แบบเวกเตอร์

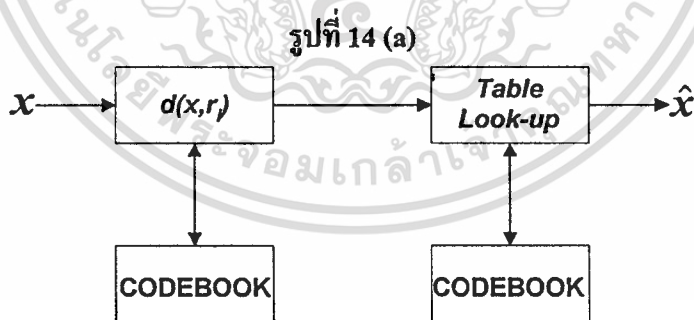
บทนำ

ทฤษฎีในบทที่ 3 ได้ชี้ให้เห็นถึงปัญหาที่เกิดจากการควอนไทซ์ข้อมูลแบบเวกเตอร์โดยทั่วไป นั่นคือ การใช้ทรัพยากรของระบบมากทั้งด้านความเร็วในการประมวลผลและการใช้หน่วยความจำ จากปัญหาเหล่านี้มีผู้คิดค้นวิธีใหม่ๆ ที่ปรับเปลี่ยน/แก้ไขเพื่อลดปริมาณการคำนวณลง วิธีหนึ่งที่ใช้กันก็คือ "การควอนไทซ์ข้อมูลเวกเตอร์เป็นกลุ่ม" (Classified Vector Quantization : CVQ)

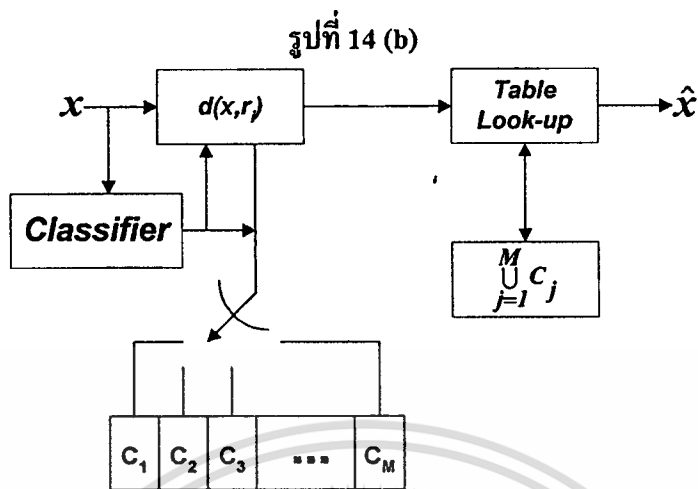
การควอนไทซ์ข้อมูลเวกเตอร์เป็นกลุ่ม

คือการจัดแบ่งข้อมูลออกเป็นกลุ่มๆ ก่อนเพื่อให้มีขนาดเล็กลง จากนั้นจึงนำข้อมูลแต่ละกลุ่มไปควอนไทซ์โดยวิธีการ LBG's อีกทีหนึ่ง

เพื่อเห็นภาพการทำงานที่ชัดเจนขอให้อธิบายรูปที่ 14 เป็น Block Diagram เปรียบเทียบการทำงานระหว่างวิธีการเดิม LBG's และวิธีการควอนไทซ์ข้อมูลเวกเตอร์เป็นกลุ่ม



แสดงแผนภูมิการควอนไทซ์แบบ LBG's



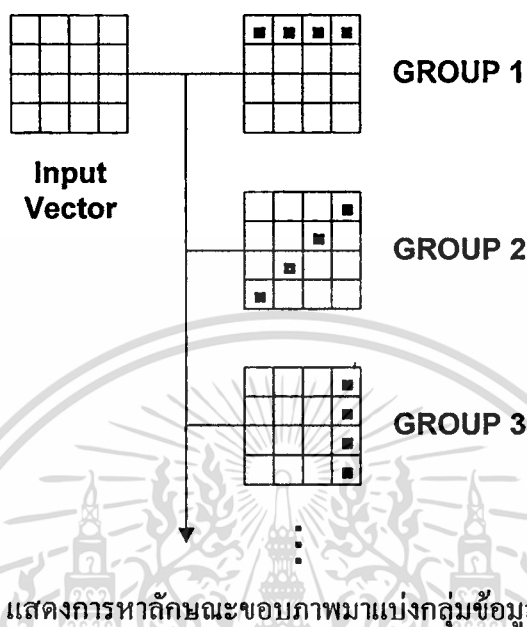
แสดงแผนภูมิการควอนไทซ์ข้อมูลเวกเตอร์แบบเป็นกลุ่ม (CVQ)

จากรูปแสดงว่าวิธีการทำงานของ LBG's นั้นต้องนำข้อมูลมาผ่านกระบวนการทั้งหมดเพื่อหา Codebook ขณะที่วิธีการ CVQ จะแบ่งข้อมูลที่มีลักษณะเหมือนหรือใกล้เคียงกันเป็นกลุ่มๆ แยกกันก่อน แล้วจึงทำกระบวนการ LBG's ภายในแต่ละกลุ่มย่อยเหล่านั้นอีกทีหนึ่ง วิธีนี้จะได้ประโยชน์มากเนื่องจากข้อมูลที่อยู่ภายในกลุ่มเดียวกันนั้นก็มีความใกล้เคียงกันอยู่ก่อนแล้วเมื่อนำมาหาค่า Codebook จึงใช้จำนวนรอบการคำนวณที่ลดลงและแต่ละรอบการคำนวณก็มีข้อมูลน้อยกว่าเพราะคำนวณเฉพาะข้อมูลภายในกลุ่มตัวเองเท่านั้น

ตัวอย่างเงื่อนไขที่ใช้จัดแบ่งกลุ่มข้อมูลขึ้นอยู่กับชนิดและรายละเอียดของข้อมูลนั้นๆ หากเป็นงานด้านการประมวลผลภาพอาจใช้วิธีการเช่น การหาขอบภาพ, การแบ่งเฉดสี เป็นต้น

รูปที่ 15 เป็น Block Diagram ตัวอย่างการหาลักษณะขอบภาพแบบต่างๆ มาเป็นเงื่อนไขการแบ่งข้อมูล

รูปที่ 15



ข้อจำกัดของวิธีการควอนไทซ์ข้อมูลเวกเตอร์เป็นกลุ่ม

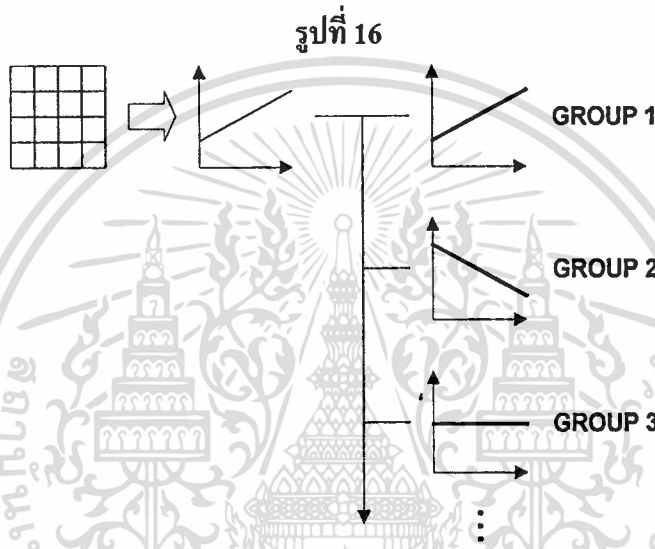
แม้ว่าการหาค่า Codebook ด้วยวิธีการควอนไทซ์ข้อมูลเวกเตอร์เป็นกลุ่มจะมีข้อดีซึ่งได้ปรับปรุงขึ้นจากวิธีการเดิมแล้ว แต่ก็ยังมีข้อจำกัดการนำไปใช้งานอยู่ กล่าวคือเงื่อนไขที่ใช้แบ่งข้อมูลนั้นจะใช้ได้กับข้อมูลหรือสัญญาณรูปแบบหนึ่งๆเท่านั้น เมื่อนำเงื่อนไขที่ใช้ได้กับข้อมูลชนิดหนึ่งไปใช้กับข้อมูลชนิดอื่นอาจใช้ไม่ได้ผลดี ตัวอย่างเช่นเราใช้การหาขอบภาพสำหรับการแบ่งกลุ่มข้อมูลภาพที่มีลักษณะเป็นรูปทรงเรขาคณิตได้ผลดี แต่เมื่อนำเงื่อนไขเดียวกันนี้ไปใช้กับข้อมูลภาพที่ไม่ใช่ลักษณะดังกล่าวก็อาจทำให้ไม่สามารถแบ่งแยกกลุ่มได้ หรือการใช้เงื่อนไขการกำหนดเฉดสีของภาพที่ระดับเทรชโวลต์ต่างๆสำหรับข้อมูลภาพแบบหนึ่ง เมื่อนำข้อกำหนดนี้ไปใช้กับข้อมูลอีกแบบหนึ่งที่มีขอบเขตของเฉดสีที่ต่างกันออกไป (อาจมากหรือน้อยกว่า) ถ้ายังคงต้องการให้การแบ่งกลุ่มเป็นไปได้อย่างเหมาะสมก็ต้องปรับเปลี่ยนค่าเทรชโวลต์ที่กำหนดไว้เดิมให้สอดคล้องกับข้อมูลใหม่เป็นต้น

นอกจากนี้เงื่อนไขบางอย่างที่ใช้แบ่งกลุ่มอาจเป็นอุปสรรคที่เพิ่มปริมาณการประมวลผลเสียเอง เช่นการแบ่งกลุ่มข้อมูลโดยเงื่อนไขการใช้ขอบภาพ หากการตรวจหาขอบภาพนั้นได้จากขั้นตอนที่ซับซ้อนแล้ว การที่จะได้ผลลัพธ์เพื่อมาแยกข้อมูลเป็นกลุ่มก็จะทำให้ใช้เวลามากขึ้นไปอีก

การแบ่งกลุ่มข้อมูลโดยใช้การประมาณแบบเชิงเส้น

วิธีการที่นำเสนอนี้เป็นความพยายามที่แก้ไขปัญหาคัดเลือกข้อมูลจากข้อจำกัดของการควอนไทซ์ข้อมูลเวกเตอร์เป็นกลุ่ม เพื่อที่จะสามารถใช้เงื่อนไขอันเดียวสำหรับข้อมูลทุกรูปแบบและไม่ต้องปรับแต่งค่าพารามิเตอร์ (เช่นค่าเทรซ โสด) เมื่อลักษณะข้อมูลมีการเปลี่ยนแปลงไป

การแบ่งกลุ่มข้อมูลด้วยการประมาณค่าแบบเชิงเส้นนี้ถือเป็นลักษณะหนึ่งของการควอนไทซ์ข้อมูลเวกเตอร์เป็นกลุ่ม อธิบายได้ตามรูปที่ 16



แสดงการใช้วิธี Linear Approach มาแบ่งกลุ่มข้อมูล

จากรูปเมื่อ Input Vector แต่ละตัวเข้ามา ก็จะถูกคำนวณหาค่าประมาณเชิงเส้น (Linear Approximation) จะได้สมการเส้นตรงที่ใกล้เคียงที่สุดใช้แทนเวกเตอร์นั้น และคำนวณอย่างนี้กับ Input Vector ทุกตัว เมื่อได้สมการของเวกเตอร์ทุกตัวแล้วก็จะคำนวณหาพิสัยของปลายแต่ละด้านของเส้นตรง (ดูรูปประกอบ) ทำให้เราทราบว่าพิสัยของปลายทางซ้ายกลุ่มเส้นตรงอยู่ในช่วงใด และพิสัยของปลายทางขวากลุ่มเส้นตรงอยู่ในช่วงใด และมองเห็นภาพการแบ่งกลุ่มได้โดยง่าย จากนั้นก็สามารถใช้วิธีการต่างๆ สำหรับแบ่งกลุ่มของช่วงพิสัยเหล่านี้ให้เหมาะสม และขั้นสุดท้ายก็แบ่งจำนวน Codeword ให้ได้สัดส่วนกับจำนวนข้อมูลแต่ละกลุ่ม

การแบ่งกลุ่มข้อมูลภายหลังผ่านกระบวนการ Linear Approach สามารถใช้วิธีการต่างได้มากมายแล้วแต่การพิจารณา วิทยานิพนธ์ฉบับนี้จะแสดงวิธีการแบ่งกลุ่มที่ใช้จริงของการทดลองในบทที่ 6 ต่อไป

ผลเปรียบเทียบปริมาณการคำนวณ

ต่อไปนี้เป็นตัวอย่างแสดงตัวเลขผลการเปรียบเทียบปริมาณการคำนวณของการควอนไทซ์แบบเวกเตอร์ทั้งแบบวิธีการเดิม LBG's และวิธีการใหม่ Linear Approach

จากทฤษฎีบทที่ 3 ที่แสดงวิธีการคำนวณหาปริมาณการคำนวณในแต่ละรอบของการควอนไทซ์แบบเวกเตอร์ของวิธีการ LBG's เท่ากับ

$$NML = NM2^{NR}$$

โดยที่ N จำนวนสเกลาร์/เวกเตอร์ (มิติ)
 M จำนวน Input Vector ทั้งหมด
 R จำนวนบิต/มิติ

หากเราสมมติว่าต้องใช้การคำนวณ 10 รอบจึงจะผ่านเงื่อนไข "ค่าความคลาดเคลื่อนเฉลี่ยของรอบการคำนวณรอบปัจจุบันน้อยกว่าค่าความคลาดเคลื่อนรอบก่อนหน้าต่ำกว่าค่าตัวเลข (เทรซโฮล) ที่ตั้งไว้" ดังนั้นปริมาณการคำนวณทั้งหมดเท่ากับ

$$10NM2^{NR} \text{ ครั้ง}$$

กำหนดให้ตัวแปรต่างๆมีค่าดังนี้

$$N = 10 \text{ สเกลาร์/เวกเตอร์ (มิติ)}$$

$$M = 8000 \text{ Vectors}$$

$$R = 0.5 \text{ บิต/มิติ}$$

ฉะนั้นการคำนวณด้วยวิธี LBG's จะใช้การคำนวณทั้งหมดเท่ากับ

$$10NM2^{NR} = (10)(10)(8000)2^{(10)(0.5)} = 2.56 \times 10^7 \text{ ครั้ง}$$

$$\text{โดยมีจำนวน Codeword : } L = 2^{NR} = 2^{(10)(0.5)} = 2^5 = 32 \text{ ตัว}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อนำข้อมูลชุดเดียวกันนี้มาคำนวณด้วยวิธี Linear Approach และแบ่งข้อมูลได้เป็น 8 กลุ่ม กลุ่มละเท่าๆกัน (1000 Vectors) และสัดส่วนของ Codeword ที่ได้รับแต่ละกลุ่มเท่ากับ $32/8 = 4$ ตัว สมมุติว่าแต่ละกลุ่มย่อยนี้ใช้การคำนวณ 10 รอบเท่ากัน ดังนั้นแต่ละกลุ่มย่อยจะมีปริมาณการคำนวณทั้งสิ้น

$$10NML = (10)(10)(1000)(4) = 4 \times 10^5 \text{ ครั้ง}$$

สังเกตว่าปริมาณการคำนวณครั้งนี้จะไม่ใช้สูตร $0NM2^{NR}$ เพราะค่า "L" ในที่นี้ไม่ใช่ได้มาจาก $L = 2^{NR}$ อีกแล้วแต่ได้จากการจัดสัดส่วนของ Codeword ต่อข้อมูล และปริมาณการคำนวณรวมของทุกกลุ่มเท่ากับ

$$8 \times 10NML = 8 \times 4 \times 10^5 = 3.2 \times 10^6 \text{ ครั้ง}$$

เปรียบเทียบกับปริมาณการคำนวณทั้งสองวิธีจะได้

$$\text{ลดลง} \quad \frac{2.57 \times 10^7}{3.2 \times 10^6} = 8 \text{ เท่า}$$

อย่างไรก็ดีในทางปฏิบัติแล้วตัวเลขดังกล่าวจะขึ้นอยู่กับค่าตัวแปรอื่นที่เกี่ยวข้องเช่น จำนวนกลุ่มข้อมูล, สัดส่วนของ Codeword ที่แบ่งให้แต่ละกลุ่ม, ค่าเทรชโฮลที่กำหนดไว้และจำนวนรอบที่ใช้จริง ซึ่งทำให้เราไม่สามารถประมาณตัวเลขล่วงหน้าออกมาได้ ดังนั้นในการวัดผลการทดลองจึงต้องวัดผลรวมของเวลาทั้งหมดที่ใช้มาเปรียบเทียบกัน

บทที่ 6

การทดลองและผลการทดลอง

บทนำ

บทนี้จะนำเสนอการทดลองและผลการทดลองเพื่อแสดงถึงข้อเปรียบเทียบตามที่อธิบายในบทก่อนหน้านี้ โดยจะบรรยายเป็นข้อๆ ไปตั้งแต่ขอบเขตการทดลองไปจนถึงผลการทดลองเพื่อความเข้าใจที่เป็นระเบียบ

ขอบเขตการทดลอง

การทดลองนี้เป็นการทดลองหา Codebook และนำ Codebook ที่ได้มาควอนไทซ์ข้อมูลเปรียบเทียบกันระหว่างวิธีของ LBG's และวิธีการที่พัฒนาขึ้นมาคือ Linear Approach โดยเปรียบเทียบทั้งในด้านเวลาที่ใช้ประมวลผลและ"ค่าความคลาดเคลื่อน"จากข้อมูลต้นแบบ

ข้อมูลที่ใช้ในการทดลอง

ข้อมูลที่นำมาใช้ทดลองได้จากการนำภาพขนาด 256x256 จุดและมีความเข้มของสีเทา (Gray Scale) 256 ระดับ (8 บิต) จำนวน 7 ภาพมาแทนข้อมูลในลักษณะต่างๆ ภาพทั้งหมดจะมีชื่อใช้เรียกอ้างอิงอยู่ใต้ภาพ (ภาพต้นฉบับแสดงอยู่ในส่วน "ผลการทดลอง" ท้ายบท)

การนำภาพมาเป็นข้อมูลในการทดลองเพราะเป็นการแสดงให้เห็นจริงว่าวิธีนี้สามารถนำมาใช้งานจริงได้ ทั้งสะดวกต่อการประมวลผลและวัดผลการทดลอง

การตรวจสอบและวัดผล

ในแต่ละการทดลองเราจะบันทึกผลลัพธ์ดังนี้

1. เวลาที่ใช้ในการประมวลผล (หน่วยเป็นวินาที) ซึ่งเป็นการวัดรวมทั้งหมดตั้งแต่ภายหลังจากแปลงข้อมูลภาพเป็น Input Vector แล้วไปจนกระทั่งการควอนไทซ์เสร็จสิ้นได้กลับมาเป็น Reconstruction Vector ซึ่งได้ครอบคลุมกระบวนการไว้ทั้งหมด แต่ไม่นับรวมถึง"การประมวลผลก่อนการทดลอง" (Pre-Processing) และ "การประมวลผลหลังการทดลอง" (Post-Processing) เพราะเป็นส่วนที่ขึ้นอยู่กับการนำไปประยุกต์ใช้ไม่เกี่ยวข้องกับการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ค่าความคลาดเคลื่อน เนื่องจากข้อมูลที่ใช้เป็นข้อมูลภาพดังนั้นการวัดค่าความคลาดเคลื่อนที่เหมาะสมและเป็นที่ยอมรับที่สุดก็คือการวัด "ค่าความคลาดเคลื่อนเฉลี่ยยกกำลังสอง" (Mean Square Error : MSE) โดยใช้สูตร

$$MSE = \frac{1}{N} \sum_{i=1}^N d(x_i - \bar{x}_i)^2$$

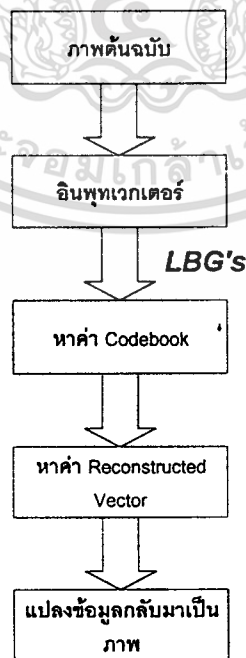
โดยที่ x_i : ข้อมูลภาพต้นฉบับ
 \bar{x}_i : ข้อมูลภาพผลลัพธ์
 N : จำนวนจุดภาพทั้งหมด

นอกจากนี้ในส่วนของวิธี Linear Approach ได้เพิ่มผลการนับจำนวนกลุ่มที่แบ่งได้ด้วยเพื่อเป็นข้อสังเกตประกอบการพิจารณาในสรุปผลการทดลอง

แผนภูมิลำดับการทดลอง

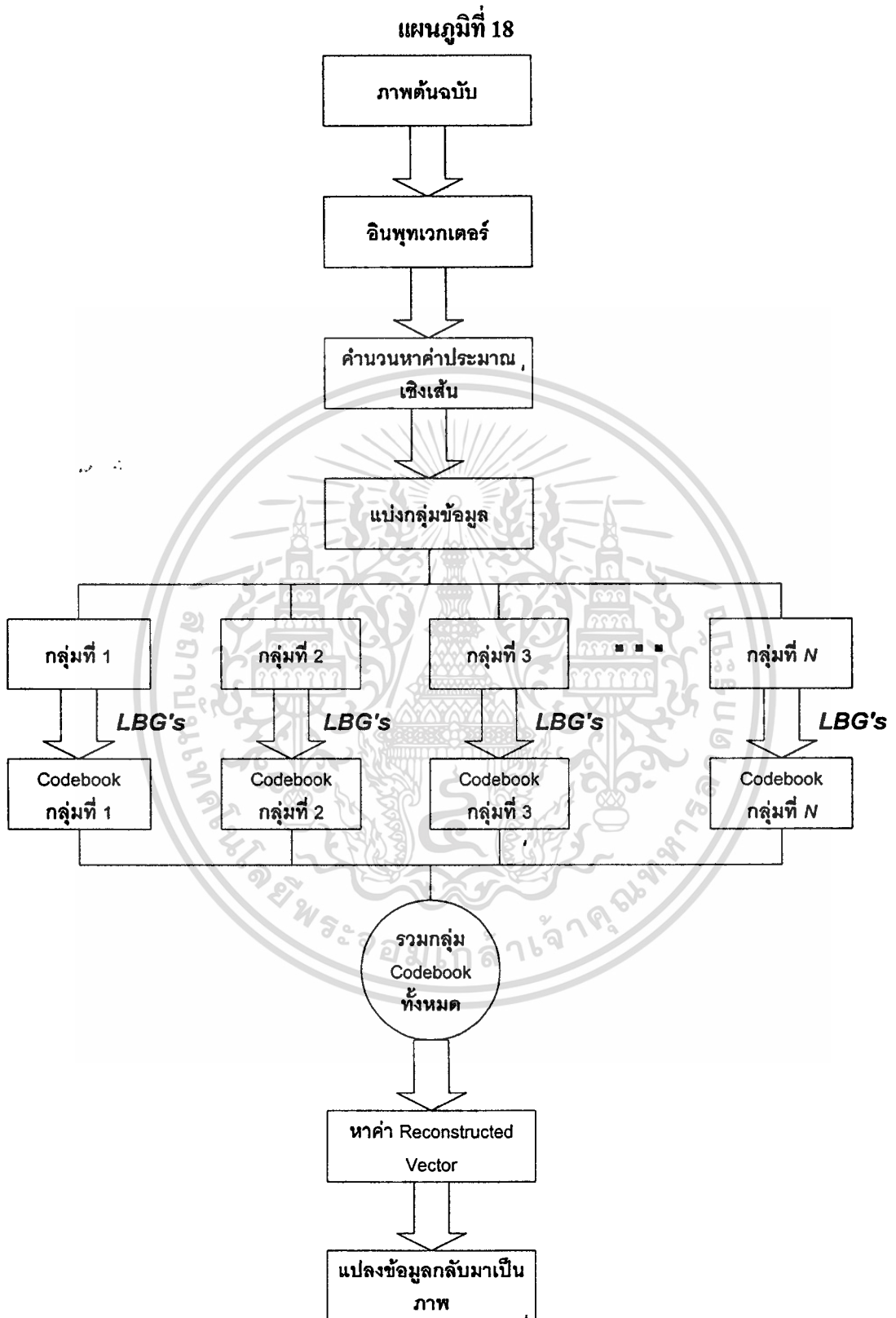
ตามที่ได้อธิบายในหัวข้อ"ขอบเขตการทดลอง"นั้น แผนภูมิที่ 17, 18 ต่อไปนี้เป็นภาพแสดงแผนภูมิลำดับการทดลองที่แยกเป็นสองส่วนคือ ภาพที่ 17 แสดงลำดับการทดลองของวิธีควอนไทซ์แบบเดิม (LBG's) และภาพที่ 18 แสดงแผนภูมิการทดลองของวิธีควอนไทซ์แบบ Linear Approach

แผนภูมิที่ 17



แสดงแผนภูมิการทดลองการควอนไทซ์แบบเวกเตอร์ด้วยวิธี LBG's

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



แสดงแผนภูมิการทดลองการควอนไทซ์แบบเวกเตอร์โดยวิธี Linear Approach

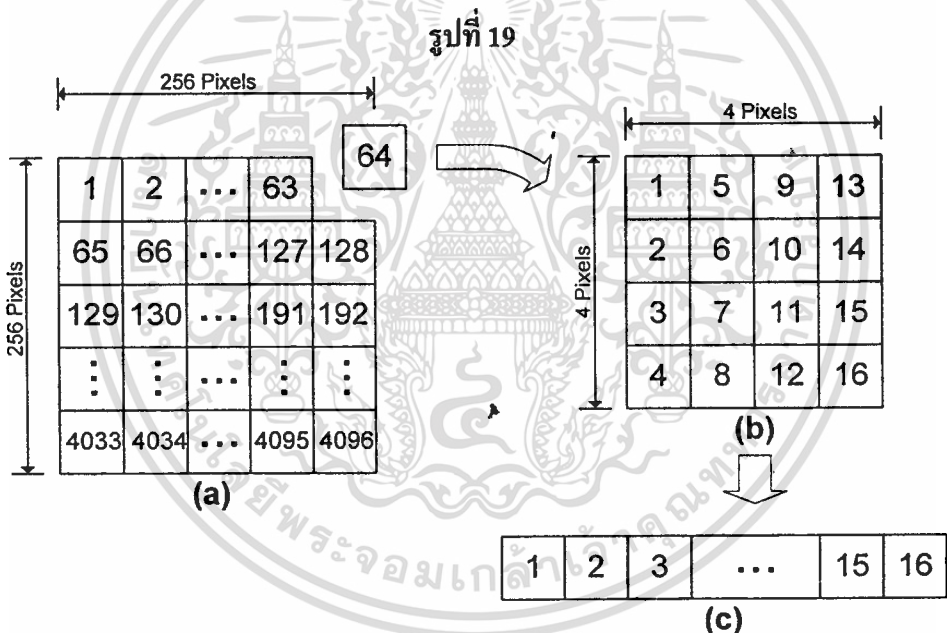
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลอง

การทดลองนี้พัฒนาขึ้นโดยใช้โปรแกรมคำนวณทางคณิตศาสตร์ MATLAB[®] เวอร์ชัน 4.2 ทำงานบนเครื่องไมโครคอมพิวเตอร์ที่มีหน่วยประมวลผล "Pentium[®]" ความเร็ว 100 MHz และหน่วยความจำขนาด 24 Mbytes (ตัวโปรแกรมที่ใช้ได้ลงพิมพ์ไว้ในส่วน "ภาคผนวก ก" ของวิทยานิพนธ์ฉบับนี้แล้ว)

การอธิบายอย่างละเอียดจะอธิบายตามลำดับแผนภูมิการทำงานในแผนภูมิที่ 17 และ 18 โดยเริ่มด้วยวิธี LBG's เมื่อเสร็จสิ้นแล้วจึงอธิบายในส่วนของวิธี Linear Approach เฉพาะที่เพิ่มเติมจากวิธีแรก เป็นข้อๆดังนี้

1. การแปลงข้อมูลภาพต้นฉบับเป็น Input Vector



แสดงการแปลงภาพต้นฉบับเป็นข้อมูลเวกเตอร์

ลักษณะของภาพที่นำมาใช้เป็นข้อมูลมีคุณสมบัติต่อไปนี้

1. มีขนาด 256x256 จุดภาพ
2. เป็นภาพที่มีค่าความเข้มของสีเทา (Gray Scale) 256 ระดับ (0-255) หรือใช้พื้นที่เก็บ 8 บิต/จุดภาพ

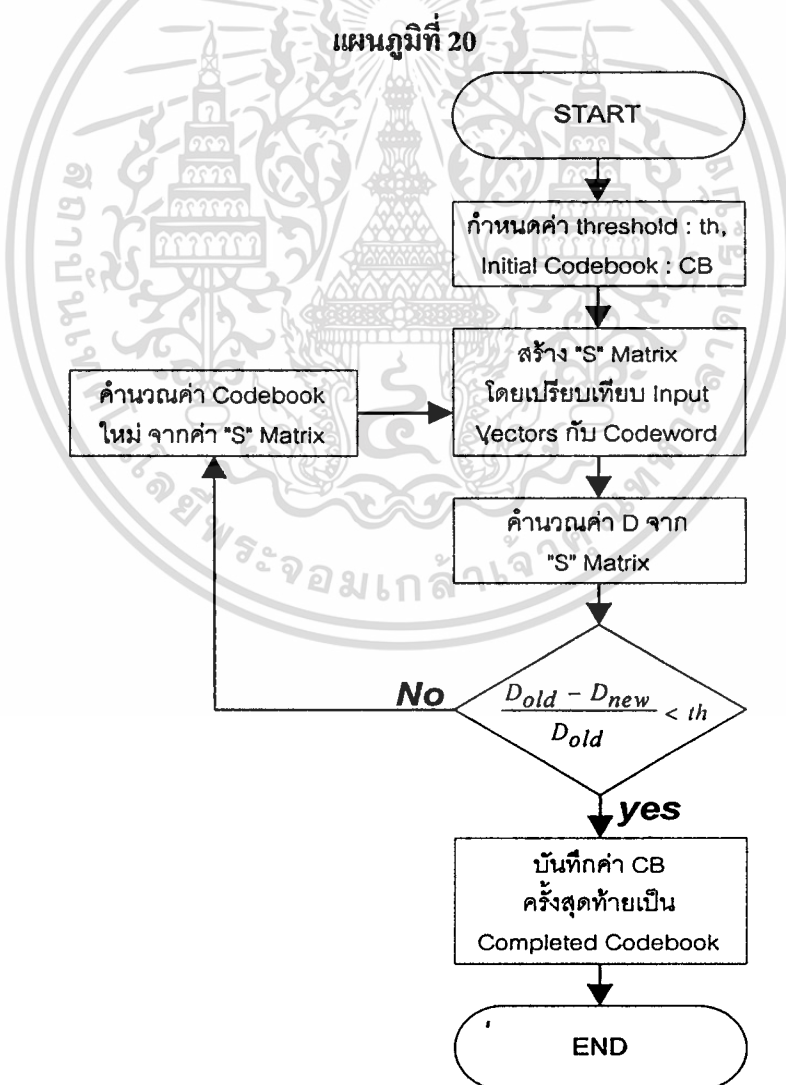
3. ความแตกต่างของภาพแต่ละภาพที่ใช้เมื่อแปลงเป็น Input Vector แล้วก็คือความแตกต่างของสัญญาณข้อมูลนั่นเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 19 แสดงขั้นตอนการทำงาน เริ่มจากเมื่อดึงภาพที่ต้องการเข้ามาในหน่วยความจำคอมพิวเตอร์ในรูปแบบเมทริกซ์ขนาด 256×256 แล้ว จะตัดแบ่งภาพนั้นเป็นกลุ่มย่อยขนาด 4×4 จุด เรียงจากซ้ายไปขวาได้เป็นจำนวน $(256 \times 256) / (4 \times 4) = 4096$ กลุ่ม แต่ละกลุ่มข้อมูลที่เกิดขึ้นจะถูกจัดเรียงใหม่อีกเป็นเมทริกซ์ขนาด "1x16" หรือเวกเตอร์ขนาด 16 มิติตามรูปจนครบทุกกลุ่ม สุดท้ายจะได้เป็นเมทริกซ์ใหม่ที่มีขนาด 4096×16 เมื่อจบสิ้นขั้นตอนนี้เราจะได้อำพารามิเตอร์ที่ใช้ในการทดลอง 2 ตัว คือ จำนวน Input Vector : $M = 4096$ ตัว และ จำนวนสเกลาร์ต่อเวกเตอร์ : $N = 16$ มิติ

2. การหา Codebook ด้วยวิธีการ LBG's

การหา Codebook ด้วยวิธีการ LBG's นี้มีใช้ทั้งในสองการทดลอง แต่ต่างกันที่จำนวน Input Vector และจำนวน Codeword ที่ใช้ดังที่ได้อธิบายในบทที่ 5 มาแล้ว



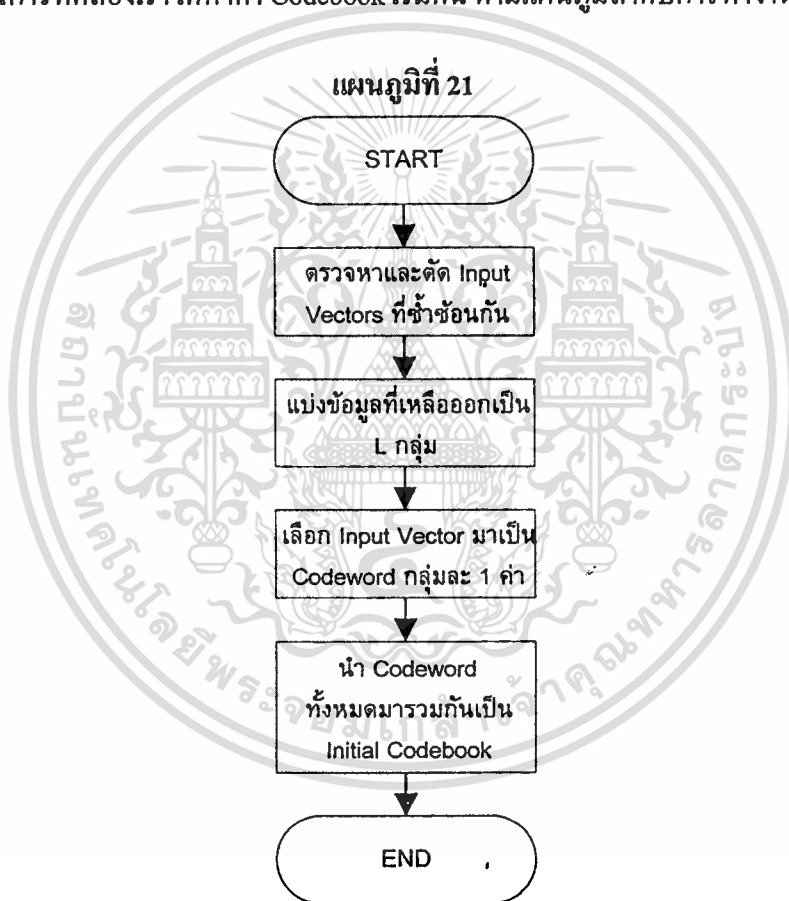
แสดงแผนภูมิการทดลองการหาค่า Codebook ด้วยวิธี LBG's

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากแผนภูมิที่ 20 เป็นแผนภูมิลำดับการทำงาน อธิบายเป็นข้อๆ ได้ดังนี้

2.1 เริ่มจากการตั้งค่าเทรซโฮลด์ที่เป็นใช้เงื่อนไขยุติการคำนวณ การทดลองนี้ตั้งไว้ที่ 0.05

2.2 กำหนดค่า Codebook เริ่มต้น (Initialized Codebook) ดังที่ได้กล่าวไว้แล้วว่าค่านี้เป็นตัวแปรหนึ่งที่สำคัญมีผลต่อจำนวนรอบการทำงาน เพราะฉะนั้นการกำหนดค่า Codebook เริ่มต้นที่ดีจะช่วยในเรื่องการประมวลผลด้วย โดยทั่วไปแล้วค่า Initialized Codebook กำหนดขึ้นจากค่าใดๆก็ได้ แต่การกำหนดค่าที่เหมาะสมนั้นควรมีลักษณะที่ได้อยู่สองอย่างคือ ข้อมูลที่เป็น Codeword ใน Codebook เริ่มต้นแต่ละตัวต้องไม่เหมือนกัน และควรได้มาจากการสุ่มเลือกแบบกระจายจาก Input Vector เอง ในการทดลองเราได้หาค่า Codebook เริ่มต้น ตามแผนภูมิลำดับการทำงานในภาพที่ 21



แสดงแผนภูมิการหาค่า Initial Codebook

เริ่มจากนำ Input Vector ที่ได้จากการแปลงข้อมูลภาพต้นฉบับมาตรวจหา Vector ที่เหมือนกัน และตัดออกให้เหลือเฉพาะแต่ Vector ที่ไม่เหมือนกันเท่านั้น แบ่งข้อมูลที่ได้ออกเป็นกลุ่ม กลุ่มละเท่าๆกัน จำนวนกลุ่มเท่ากับจำนวน Codeword ที่ต้องการสร้าง แล้วจึงเลือกเอาค่าหนึ่งค่าใดในกลุ่มมาเป็น Codeword กลุ่มละ 1 ตัว (การทดลองจริงได้เลือกเอาค่าแรกของกลุ่มมาใช้) สุดท้ายเรา

จะได้จำนวน Codeword เท่ากับที่เราต้องการสร้าง, ค่าที่ได้แต่ละตัวไม่ซ้ำกันและกระจายจาก Input Vector

2.3 การสร้าง "S" Matrix : คือเมตริกซ์ที่เป็นตัวบอกว่า Input Vector ตัวใดมีค่าใกล้เคียงกับ Codeword ตัวใดมากที่สุด โดยมีลักษณะตัวอย่างดังนี้ (รูปที่ 22 ประกอบ)

รูปที่ 22

5	7	14	57	...
3	10	25	30	...
17	5	9	18	...
...
13	3	11	98	...

แสดงตัวอย่างของ "S" Matrix

- ◆ เป็นเมตริกซ์ที่มีจำนวนแถวเท่ากับจำนวน Codeword
- ◆ ค่าในคอลัมน์ที่หนึ่งของแต่ละแถวเป็นตัวบอกจำนวนว่ามีจำนวน Input Vector จำนวนเท่าใดที่มีค่าใกล้เคียงกับค่า Codeword ตัวที่ "ตำแหน่งแถว" นั้นมากที่สุด
- ◆ ข้อมูลตั้งแต่คอลัมน์ที่สองขึ้นไปของแต่ละแถว เป็นตัวบอกค่าลำดับของ Input Vector ที่ใกล้เคียงกับ Codeword ของแถวนั้นที่สุด

ตัวอย่างเช่น ข้อมูลแถวที่ 2 ของ "S" Matrix ประกอบไปด้วยตัวเลขจากคอลัมน์ที่หนึ่งดังนี้ "3, 10, 25 และ 30" ตามลำดับ หมายความว่า "มีข้อมูลจำนวน 3 ตัว (ค่าในคอลัมน์ที่ 1) ที่ใกล้เคียงที่สุดกับค่า Codeword ตัวที่ 2 ("S" Matrix แถวที่ 2) คือ Input Vector ตัวที่ 10, 25 และ 30 ตามลำดับ (ค่าในคอลัมน์ที่ 2-4)" เป็นต้น

2.4 คำนวณ "ค่าความคลาดเคลื่อนเฉลี่ย" โดยนำค่า Input Vector แต่ละตัวมาเปรียบเทียบกับค่าที่ใกล้เคียงค่าของมันมากที่สุด (ดูจาก "S" Matrix) และนำมารวมกันหาค่าเฉลี่ยทั้งหมดดังนี้

$$D = \frac{1}{M} \sum_{i=1}^M d(f_i, r_i)$$

โดยที่ f_i : Input Vector
 r_i : ค่า Codeword ที่ใกล้เคียงที่สุด
 M : จำนวน Input Vector ทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5 ตรวจสอบเงื่อนไข : ขั้นตอนนี้เป็นตรวจสอบว่าจะยุติการคำนวณหรือต้องคำนวณใหม่ต่อไปโดยดูจากเงื่อนไข

$$\frac{D_{OLD} - D_{NEW}}{D_{OLD}} < threshold$$

โดยที่ D_{OLD} : ค่าความคลาดเคลื่อนเฉลี่ยครั้งก่อนหน้า
 D_{NEW} : ค่าความคลาดเคลื่อนเฉลี่ยครั้งปัจจุบัน
 th : ค่าเทรชโฮลที่กำหนดในตอนต้น

หากผลการคำนวณทางซ้ายมีน้อยกว่าค่าเทรชโฮลที่กำหนดไว้แต่แรกทางขวามือคือเงื่อนไขเป็นจริงการคำนวณก็จะยุติลงและให้นำค่า Codebook ที่ได้ครั้งสุดท้ายไปใช้งาน แต่หากเงื่อนไขไม่เป็นจริง (ผลการคำนวณยังมากกว่าค่าเทรชโฮล) ก็ต้องคำนวณใหม่ต่อไป

สังเกตว่าการกำหนดค่าเทรชโฮลให้มีค่าต่ำจะทำให้ต้องทำให้ใช้จำนวนรอบการคำนวณมากเพื่อจะทำให้เงื่อนไขเป็นจริง และค่าเทรชโฮลนี้ก็ได้เป็นตัวกำหนดโดยตรงให้ค่าความคลาดเคลื่อนเฉลี่ยน้อยลงแต่อย่างไร

2.6 หากเงื่อนไขในข้อ 2.5 ยังไม่เป็นจริงจะต้องคำนวณหาค่า Codebook ชุดใหม่ ซึ่งเป็นการคำนวณหาตำแหน่งจุดกึ่งกลางของข้อมูล (Centroid of Cell) ใหม่ โดยการนำข้อมูล Input Vector แต่ละกลุ่ม (แต่ละแถวของ "S" Matrix) มาบวกกันแล้วหารด้วยจำนวนข้อมูลแต่ละกลุ่ม (ตัวเลขในคอลัมน์แรกของ "S" Matrix) เมื่อทำครบทุกกลุ่มก็จะได้ค่า Codebook ชุดใหม่ที่ต้องนำไปผ่านขั้นตอนตั้งแต่ข้อ 2.3 ใหม่อีกครั้ง

หากเงื่อนไขในข้อ 2.5 เป็นจริงแล้วให้ยุติการทำงานและนำค่า Codebook ครั้งสุดท้ายบันทึกไว้เพื่อนำไปใช้งานจริงต่อไป ค่า Codebook ครั้งสุดท้ายเรียกว่า "Completed Codebook"

3. VQ Encoder

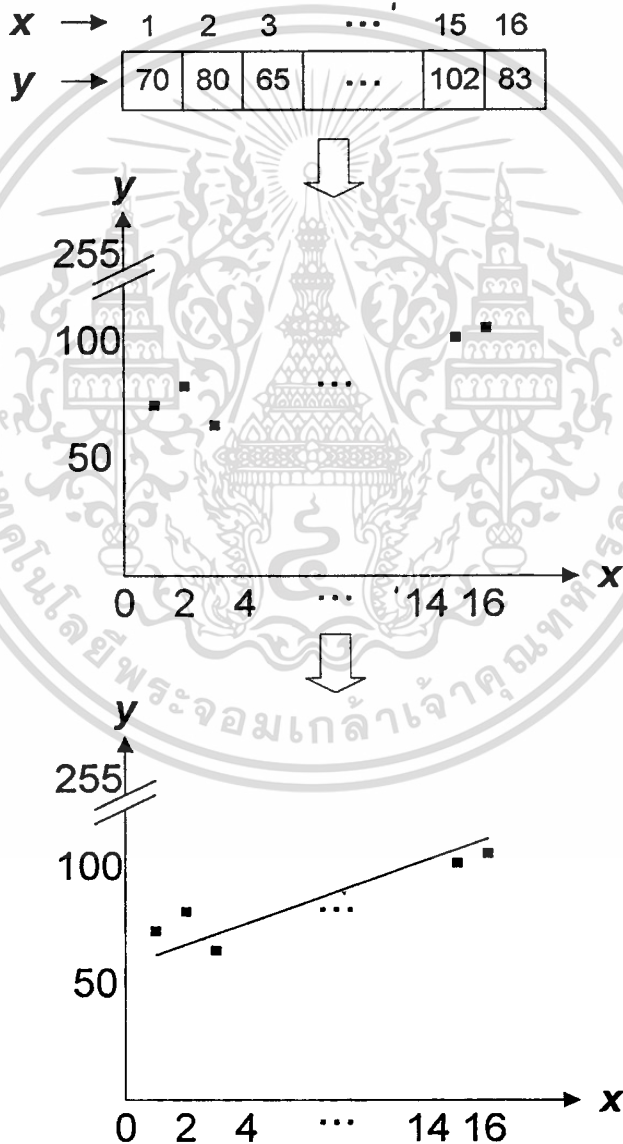
กระบวนการนี้คือการนำค่า Completed Codebook ที่ได้มาใช้ในกระบวนการควอนไทซ์แบบเวกเตอร์จริง การทำงานจะเป็นลักษณะเดียวกับการสร้าง "S" Matrix ต่างกันตรงที่เมื่อตรวจหาค่า Codeword ที่ใกล้เคียงกับ Input Vector แล้วก็จะส่งค่า "ดัชนี" ของ Codeword ตัวนั้นออกไปแทนและทำเช่นเดียวกันนี้จนครบ Input Vector ทุกตัว

4. การสร้างภาพข้อมูลกลับมาใหม่ (Reconstructed Image)

ค่า "ดัชนี" ที่ได้จากขั้นตอน VQ Encoder จะถูกนำกลับมาสร้างเป็นเวกเตอร์ขนาด 4096×16 ก่อนซึ่งเกิดจากการที่ VQ Decoder เทียบค่า "ดัชนี" กับ Codebook ที่เก็บไว้ และเรียงกลับมาเป็นรูปภาพขนาด 256×256 จุด ย้อนกลับวิธีการแปลงข้อมูลภาพต้นฉบับเป็น Input Vector (ดูรูปที่ 19)

5. การหาค่าประมาณเชิงเส้น (Linear Approximation)

รูปที่ 23



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนนี้ใช้เฉพาะในวิธี Linear Approach เท่านั้น โดยนำข้อมูลที่เป็น Input Vector แล้ว จากข้อ 1 มาหาค่าประมาณเชิงเส้นแบบ Least Square (ดูรูปที่ 23) โดย x คือค่าลำดับของมิติในแต่ละเวกเตอร์ และ y คือค่าข้อมูลจริงของมิตินั้นตามสมการ

$$y = p_1(x) = a_0 + a_1x$$

และคำนวณหาค่าสัมประสิทธิ์ a_0 และ a_1 จากสูตร

$$a_1 = \frac{n \sum_{i=1}^n y_i x_i - \sum_{i=1}^n x_i \sum_{i=1}^n y_i}{n \sum_{i=1}^n x_i^2 - \left(\sum_{i=1}^n x_i \right)^2}$$

$$a_0 = \frac{\sum_{i=1}^n y_i - a_1 \sum_{i=1}^n x_i}{n}$$

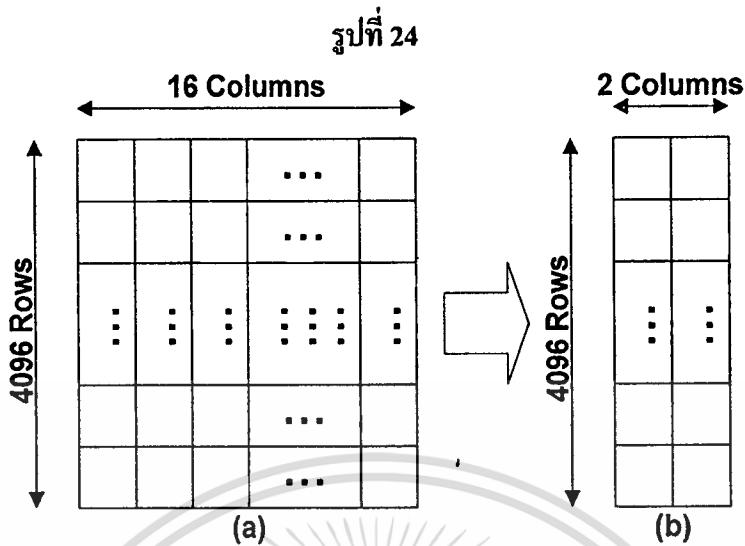
ค่า n ในที่นี้คือ 16

ในทางปฏิบัติโปรแกรม MATLAB[®] มีคำสั่งที่ใช้หาค่าสัมประสิทธิ์ทั้งสองตัวนี้ได้โดยตรง และให้คำตอบที่ถูกต้องเหมือนกับวิธีที่อธิบายในบทที่ 4 ทุกประการ
ค่าสัมประสิทธิ์ทั้งสองนั้นจะถูกนำมาใช้คำนวณค่าขอบเส้นตรงทั้งสองด้านตามสูตร

$$\text{ขอบเส้นตรงทางซ้าย} = a_0 + a_1$$

$$\text{ขอบเส้นตรงทางขวา} = a_0 + 16a_1$$

ค่าขอบซ้ายและขวาของเส้นตรงที่ได้แต่ละเส้นจะถูกเก็บไว้ในเวกเตอร์ขนาด 2 มิติ (คอลัมน์) คอลัมน์ที่ 1 เก็บค่าขอบซ้าย และคอลัมน์ที่ 2 เก็บค่าขอบขวา (ดูรูปที่ 24)



แสดงการลดจำนวนมิติที่พิจารณาจาก 16 มิติเหลือ 2 มิติ

6. การแบ่งกลุ่มข้อมูล (Classification)

เวกเตอร์ที่ได้จากการประมาณค่าเชิงเส้นจะมีขนาด 4096×2 (เป็นการลดมิติที่ใช้แบ่งจาก 16 มิติเหลือ 2 มิติ) ส่วนจำนวน Codebook ที่กำหนดให้สร้างในการทดลองนี้ได้มาจากข้อกำหนดให้ค่าพื้นที่ที่ใช้เก็บข้อมูล/มิติ (R) เท่ากับ 0.5 บิต/มิติ เพราะฉะนั้นจำนวน Codeword ได้จากสูตร

$$L = 2^{NR} = 2^{(16)(0.5)} = 2^8 = 256 \text{ ตัว}$$

การแบ่งกลุ่มจะแบ่งตามความหนาแน่นของข้อมูลโดยปฏิบัติตามขั้นตอนนี้

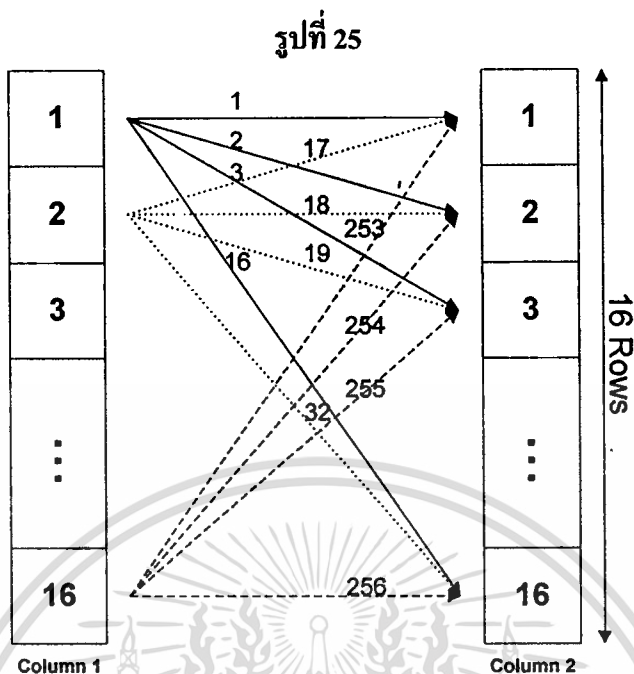
6.1 แยกจำนวน Codeword ออกเป็น 2 กลุ่ม กลุ่มละ $\sqrt{L} = \sqrt{256} = 16$ ตัว

6.2 แบ่งข้อมูลคอลัมน์ซ้ายออกเป็น 16 กลุ่ม (เท่ากับค่าที่คำนวณได้จากข้อ 6.1) ด้วยสมการ

$$(\text{ค่าสูงสุดของคอลัมน์ซ้าย} - \text{ค่าต่ำสุดของคอลัมน์ซ้าย}) / 16$$

6.3 ทำเช่นเดียวกับข้อ 6.2 กับคอลัมน์ขวามือ

6.4 หลังจากแบ่งกลุ่มข้อมูลทั้งสองด้าน เราจะได้รูปของข้อมูลที่จะถูกจัดกลุ่มดังรูปที่ 25



แสดงการจัดแบ่งข้อมูลที่ใช้ในการทดลอง

การแบ่งข้อมูลเป็น 256 กลุ่ม (เท่ากับจำนวน Codeword) จะกำหนดตามเงื่อนไขเหล่านี้

ถ้าปลายเส้นตรงด้านซ้ายอยู่ในกลุ่มทางซ้ายที่ 1 และปลายเส้นตรงด้านขวาอยู่ในกลุ่มทางขวาที่ 1 *Input Vector* ของเส้นตรงนั้นจะถูกจัดอยู่ในกลุ่มที่ 1

ถ้าปลายเส้นตรงด้านซ้ายอยู่ในกลุ่มทางซ้ายที่ 1 และปลายเส้นตรงด้านขวาอยู่ในกลุ่มทางขวาที่ 2 *Input Vector* ของเส้นตรงนั้นจะถูกจัดอยู่ในกลุ่มที่ 2

ถ้าปลายเส้นตรงด้านซ้ายอยู่ในกลุ่มทางซ้ายที่ 1 และปลายเส้นตรงด้านขวาอยู่ในกลุ่มทางขวาที่ 3 *Input Vector* ของเส้นตรงนั้นจะถูกจัดอยู่ในกลุ่มที่ 3

...

ถ้าปลายเส้นตรงด้านซ้ายอยู่ในกลุ่มทางซ้ายที่ 2 และปลายเส้นตรงด้านขวาอยู่ในกลุ่มทางขวาที่ 1 *Input Vector* ของเส้นตรงนั้นจะถูกจัดอยู่ในกลุ่มที่ 17

...

ถ้าปลายเส้นตรงด้านซ้ายอยู่ในกลุ่มทางซ้ายที่ 16 และปลายเส้นตรงด้านขวาอยู่ในกลุ่มทางขวาที่ 16 *Input Vector* ของเส้นตรงนั้นจะถูกจัดอยู่ในกลุ่มที่ 256

6.5 จัดสัดส่วน Codeword ให้เหมาะสม ในทางปฏิบัติแล้วเราไม่สามารถแบ่งจำนวน

Codeword ให้แต่ละกลุ่มได้กลุ่มละ 1 ตัวเพราะไม่อาจกำหนดได้ว่าข้อมูลจะถูกแบ่งเท่ากันทุกกลุ่ม เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยามให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีก้นำไปใช้

จะมีบางกลุ่มมีข้อมูลหนาแน่นมากและบางกลุ่มมีข้อมูลหนาแน่นน้อยและบางกลุ่ม ไม่มีข้อมูลเลย เพื่อที่จะปรับแต่งข้อมูลให้ถูกต้อง เราจะต้องกำหนดค่าอัตราส่วนระหว่างจำนวน Input Vector ต่อจำนวน Codeword : R ดังนี้

$$R = \text{จำนวน Input Vector} / \text{จำนวน Codeword} = 4096/256 = 16$$

และจำนวน Codeword ของแต่ละกลุ่มเท่ากับ

$$\text{จำนวน Codeword} = \text{Round}(\text{จำนวนข้อมูลในกลุ่ม} / R)$$

โดยที่ $\text{Round}()$: การปัดเศษทางคณิตศาสตร์

และหากกลุ่มใดมีข้อมูลน้อยเกินไปจนเมื่อคำนวณแล้วค่าจำนวน Codeword เท่ากับ "0" ก็ จะนำข้อมูลที่อยู่ในกลุ่มเหล่านั้นมารวมกันเป็นกลุ่มใหม่กลุ่มเดียว ในการทดลองเราตั้งชื่อกลุ่มนี้ว่า "Left" และจำนวน Codeword ที่จัดสรรให้กับกลุ่มนี้คือส่วนที่เหลือจากที่จัดสรรให้กลุ่มอื่นหมดแล้ว

เมื่อเสร็จสิ้นแล้วจะพบว่าจำนวนกลุ่มทั้งหมดจะลดลงจากที่กำหนดไว้เดิม 256 กลุ่ม และแต่ละกลุ่มก็จะมีจำนวน Codeword ได้เป็นสัดส่วนกับ Input Vector แต่ละกลุ่มอย่างเหมาะสม

7. รวมค่า Codebook กลับคืน (Union Codebook)

ข้อมูลที่แยกเป็นกลุ่มและสัดส่วนของ Codebook ที่ได้รับจะถูกนำไปหาค่า "Completed Codebook" ด้วยวิธีการ LBG's (ข้อ 2) หลังจากนั้นก็นำ Codebook ที่ผ่านกระบวนการแล้วของแต่ละกลุ่มมารวมกัน (Union) ให้เป็น "Completed Codebook" ของระบบอีกที

ผลการทดลอง

ตารางที่ 1

ชื่อภาพ	ตัวแปรที่วัดผล	วิธี LBG's	วิธี Linear Approach
Camera	เวลาประมวลผล (sec)	1711.0	262.11
	ค่าความคลาดเคลื่อนเฉลี่ย	129.1499	154.0807
	จำนวนกลุ่ม (กลุ่ม)	1	24
Cell	เวลาประมวลผล (sec)	1123.4	313.46
	ค่าความคลาดเคลื่อนเฉลี่ย	11.6609	12.6932
	จำนวนกลุ่ม (กลุ่ม)	1	25
Dog	เวลาประมวลผล (sec)	1562.8	171.20
	ค่าความคลาดเคลื่อนเฉลี่ย	55.1567	68.8412
	จำนวนกลุ่ม (กลุ่ม)	1	53
Girl	เวลาประมวลผล (sec)	1633.4	135.50
	ค่าความคลาดเคลื่อนเฉลี่ย	60.8323	68.0900
	จำนวนกลุ่ม (กลุ่ม)	1	55
Lady	เวลาประมวลผล (sec)	1735.6	231.01
	ค่าความคลาดเคลื่อนเฉลี่ย	61.4725	74.3472
	จำนวนกลุ่ม (กลุ่ม)	1	29
Lena	เวลาประมวลผล (sec)	1664.4	163.17
	ค่าความคลาดเคลื่อนเฉลี่ย	67.0454	82.1272
	จำนวนกลุ่ม (กลุ่ม)	1	51
Plant	เวลาประมวลผล (sec)	1613.7	143.57
	ค่าความคลาดเคลื่อนเฉลี่ย	109.0966	125.7395
	จำนวนกลุ่ม (กลุ่ม)	1	43

ตารางผลการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพที่ 26 - 32 เป็นภาพที่ใช้เป็นข้อมูลสำหรับการทดลอง (ชื่อภาพที่ใช้เรียกอ้างอิงปรากฏ
อยู่ใต้ภาพนั้นๆ)

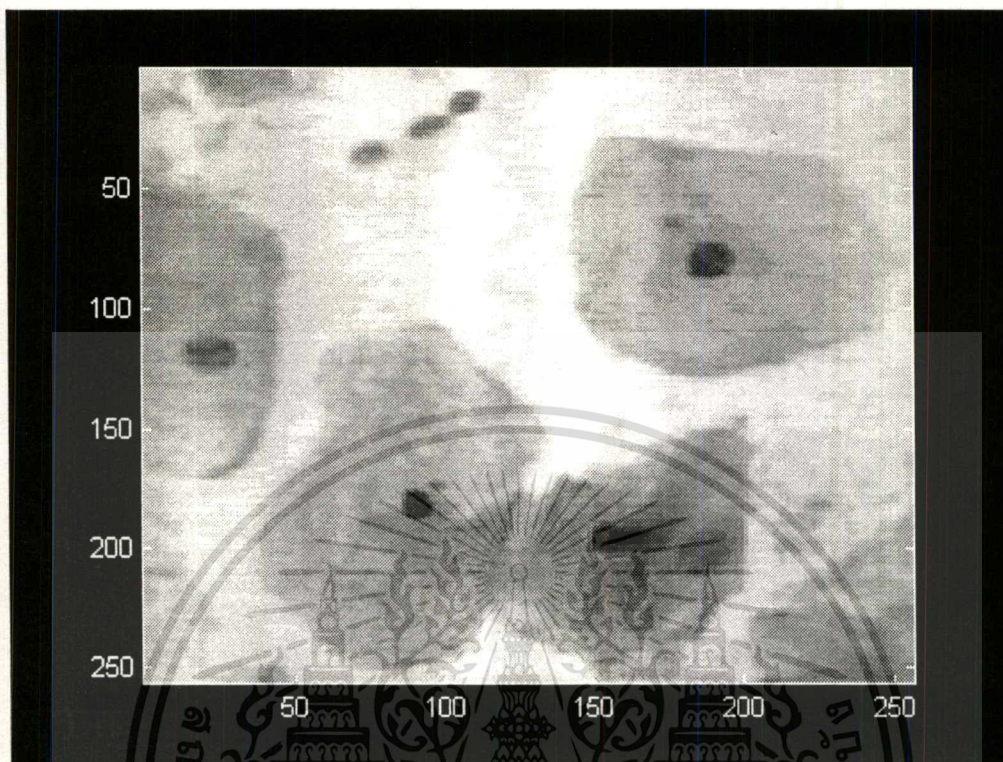
ภาพที่ 26



แสดงข้อมูลภาพต้นฉบับชื่อ "Camera"

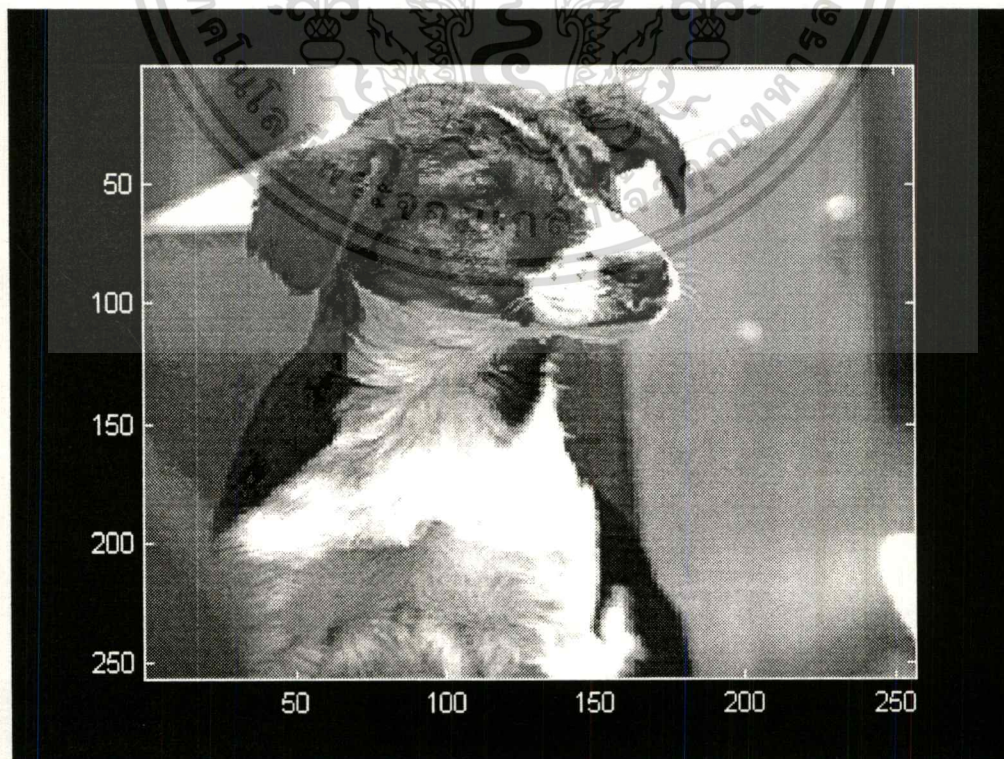
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพที่ 27



แสดงข้อมูลภาพต้นฉบับชื่อ "Cell"

ภาพที่ 28



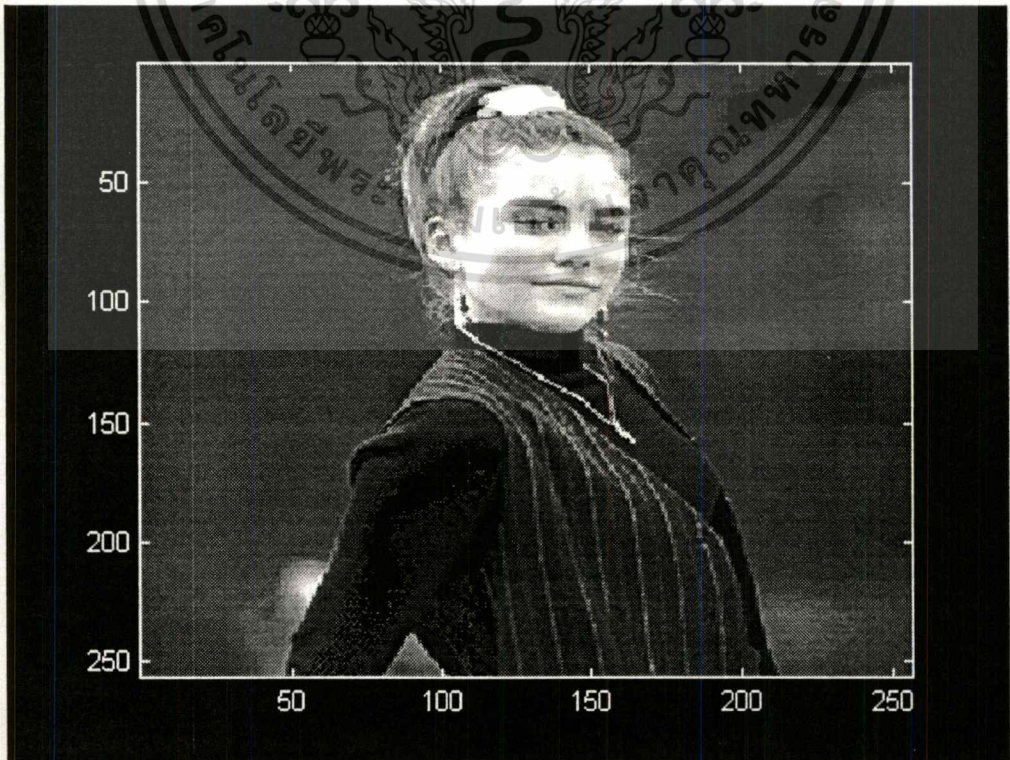
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับแสดงข้อมูลภาพต้นฉบับชื่อ "Dog" อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพที่ 29



แสดงข้อมูลภาพต้นฉบับชื่อ "Girl"

ภาพที่ 30



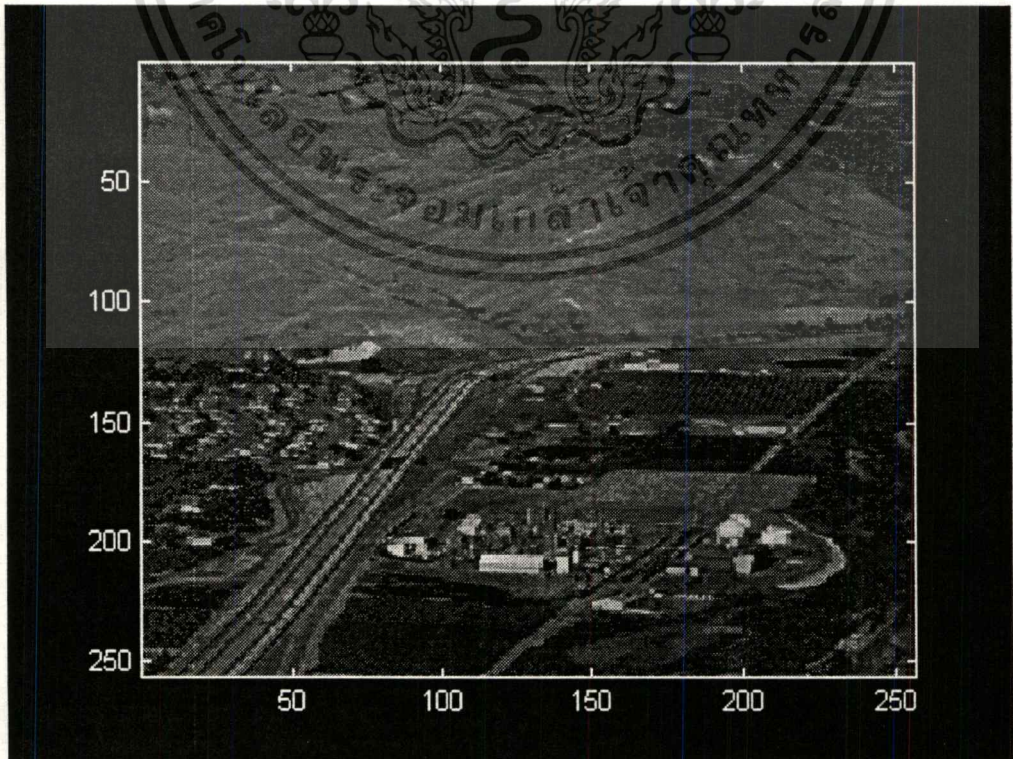
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับแสดงข้อมูลภาพต้นฉบับชื่อ "Lady" อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพที่ 31



แสดงข้อมูลภาพต้นฉบับชื่อ "Lena"

ภาพที่ 32



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับแสดงข้อมูลภาพต้นฉบับชื่อ "Plant" อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพที่ 33 - 39 เป็นภาพที่ได้จากการผ่านกระบวนการ Linear Approach

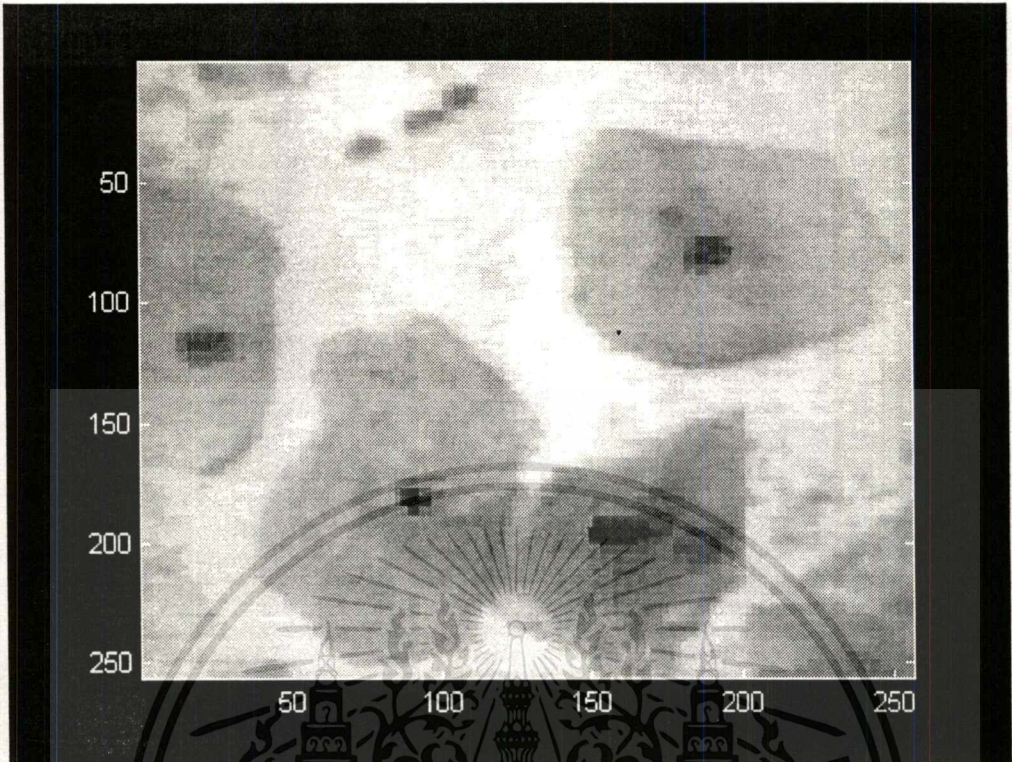
ภาพที่ 33



แสดงข้อมูลผลลัพธ์ภาพชื่อ "Camera" ที่ได้จากวิธีการ Linear Approach

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพที่ 34



แสดงข้อมูลผลลัพธ์ภาพชื่อ "Cell" ที่ได้จากวิธีการ Linear Approach

ภาพที่ 35



แสดงข้อมูลผลลัพธ์ภาพชื่อ "Dog" ที่ได้จากวิธีการ Linear Approach

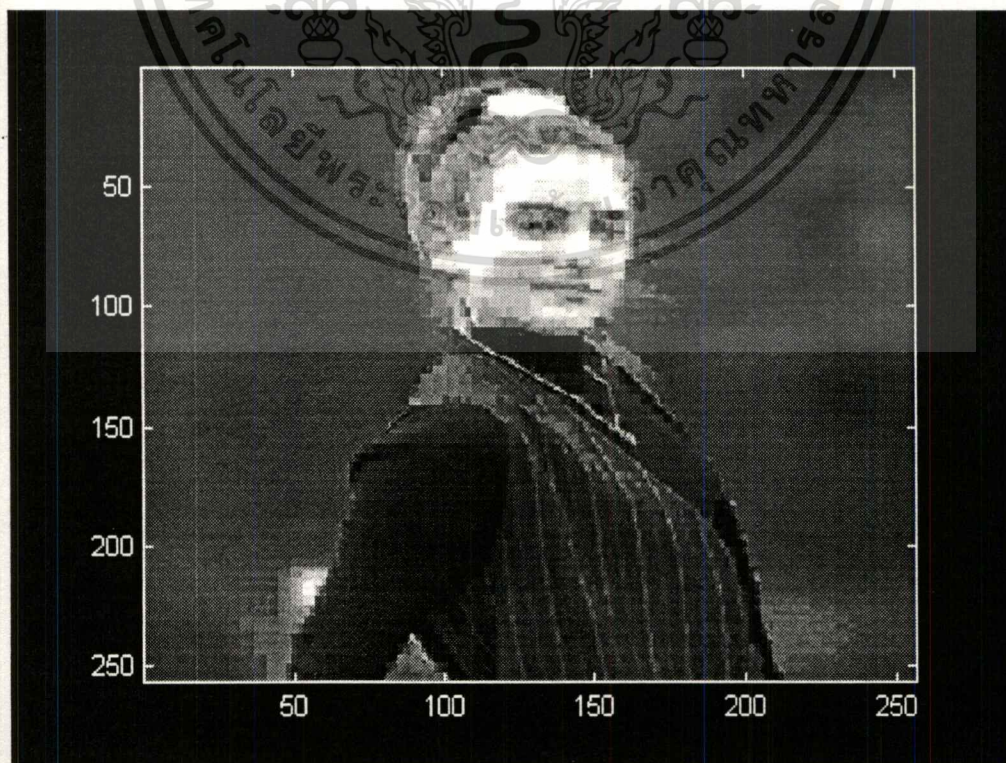
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพที่ 36



แสดงข้อมูลผลลัพธ์ภาพชื่อ "Girl" ที่ได้จากวิธีการ Linear Approach

ภาพที่ 37



เอกสารนี้เป็นเอกสารแสดงข้อมูลผลลัพธ์ภาพชื่อ "Lady" ที่ได้จากวิธีการ Linear Approach ในประเด็นด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพที่ 38



แสดงข้อมูลผลลัพธ์ภาพชื่อ "Lena" ที่ได้จากวิธีการ Linear Approach

ภาพที่ 39



แสดงข้อมูลผลลัพธ์ภาพชื่อ "Plant" ที่ได้จากวิธีการ Linear Approach

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการศึกษาเท่านั้น เมื่อผู้เช่าได้เห็น ใบโฆษณาโครงการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพที่ 40 - 46 เป็นภาพที่ได้จากการผ่านกระบวนการ LBG's (วิธีเดิม)

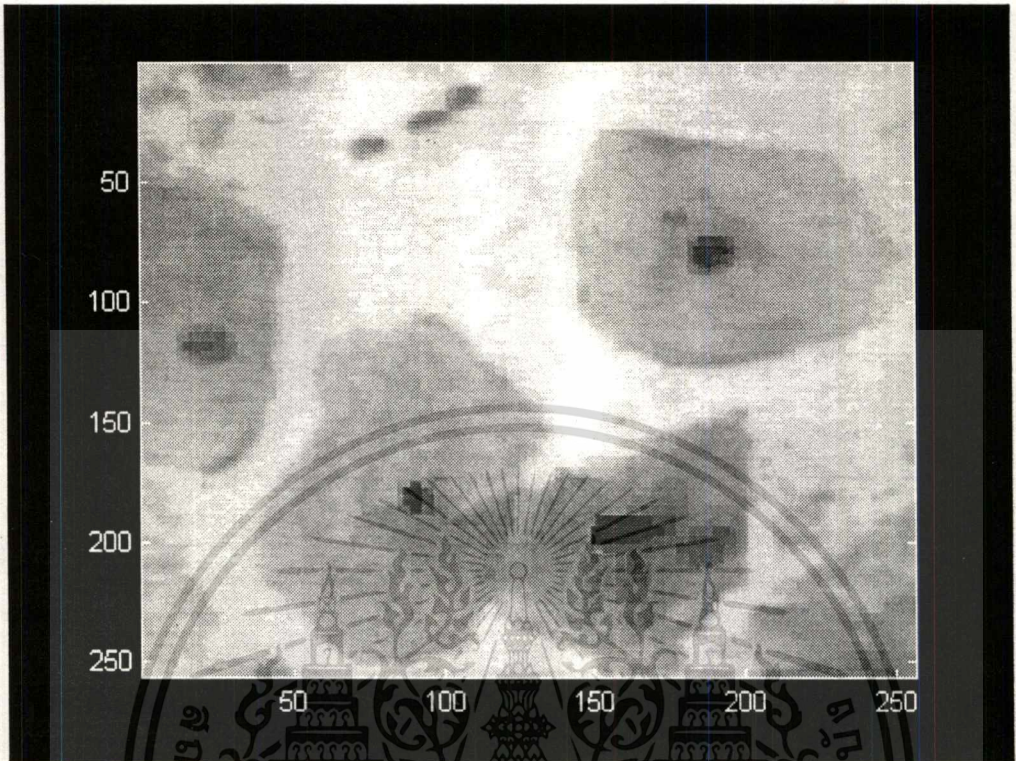
ภาพที่ 40



แสดงข้อมูลผลลัพธ์ภาพชื่อ "Camera" ที่ได้จากวิธีการ LBG's

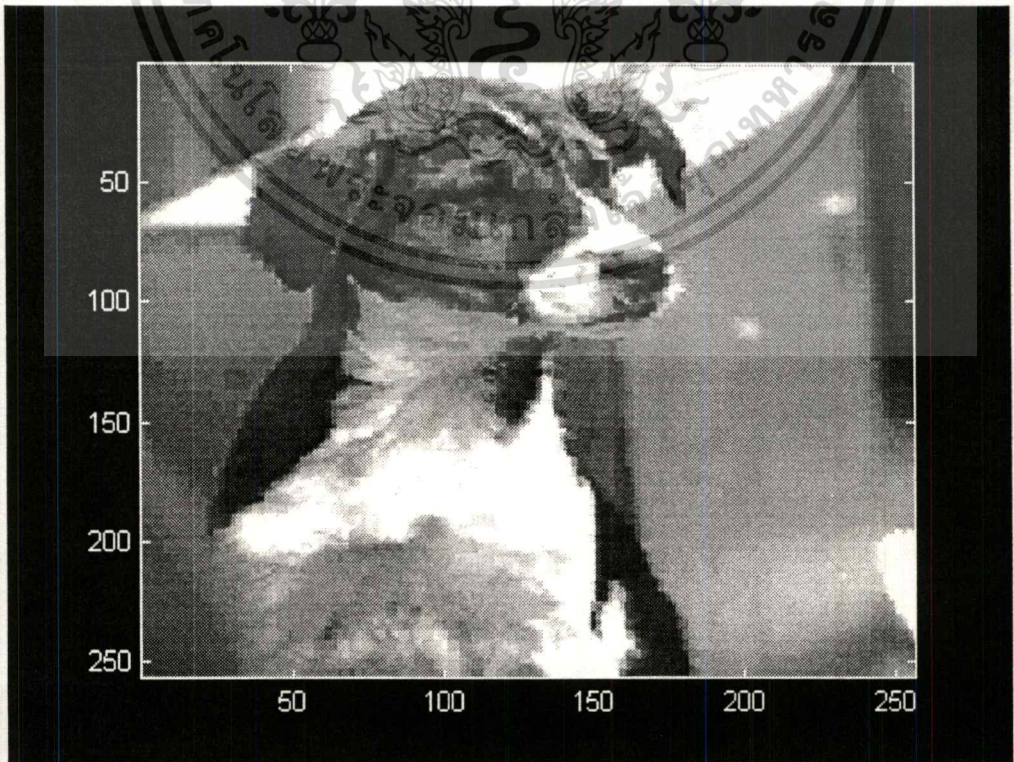
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพที่ 41



แสดงข้อมูลผลลัพธ์ภาพชื่อ "Cell" ที่ได้จากวิธีการ LBG's

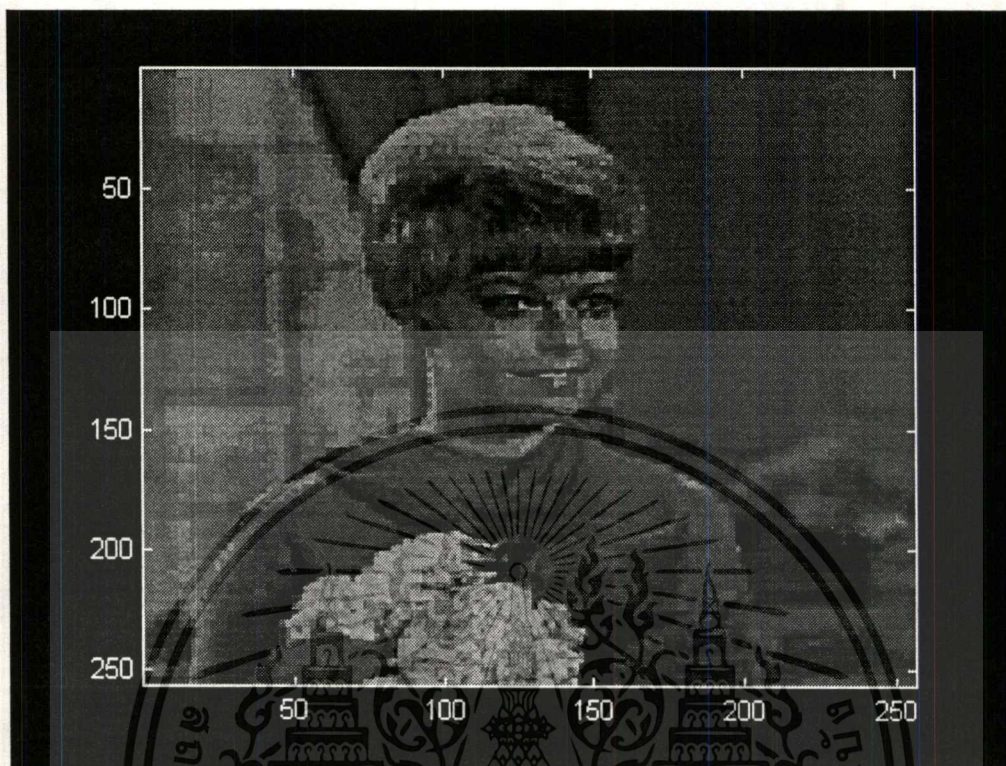
ภาพที่ 42



แสดงข้อมูลผลลัพธ์ภาพชื่อ "Dog" ที่ได้จากวิธีการ LBG's

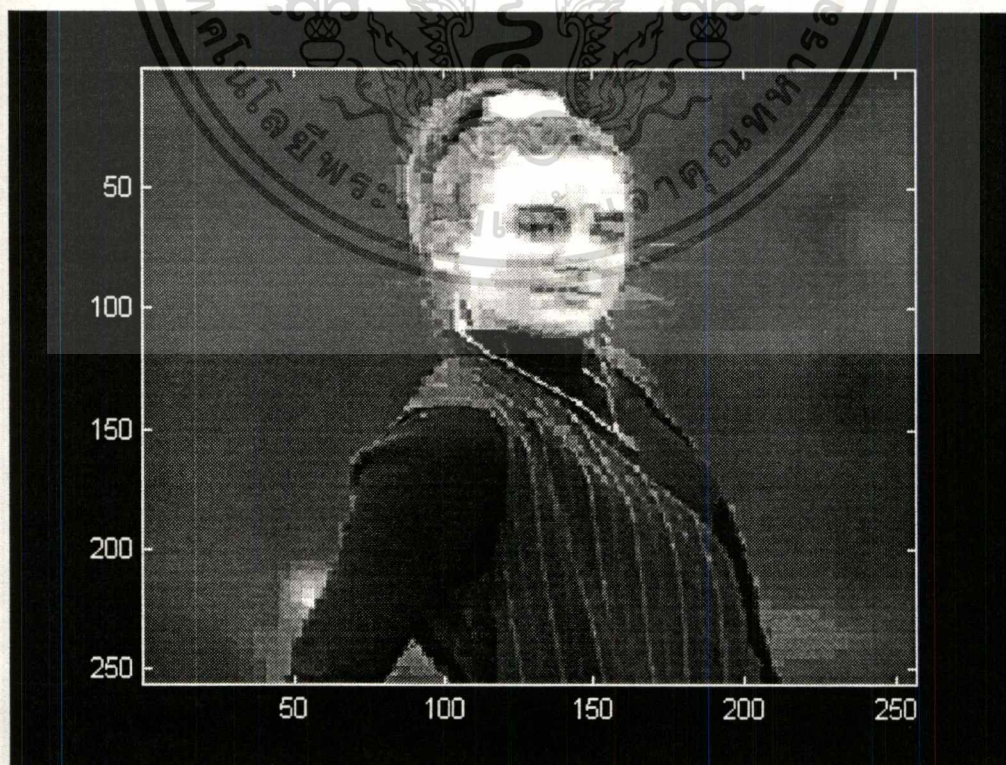
เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพที่ 43



แสดงข้อมูลผลลัพธ์ภาพชื่อ "Girl" ที่ได้จากวิธีการ LBG's

ภาพที่ 44



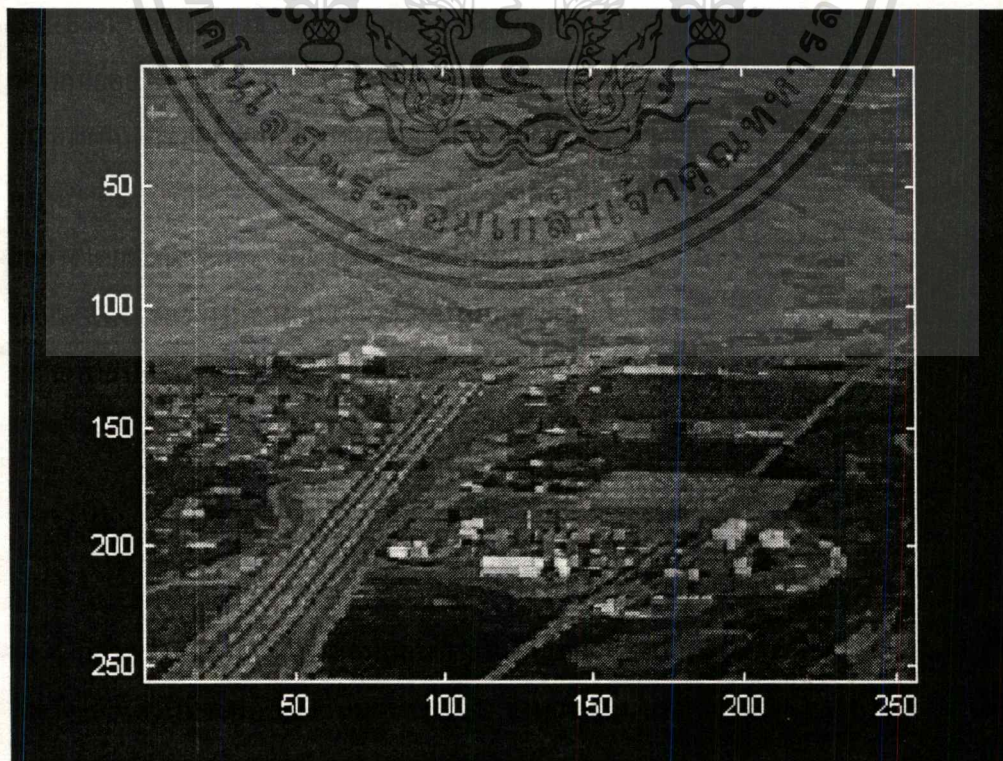
เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ภาพชื่อ "Lady" ที่ได้จากวิธีการ LBG's ไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพที่ 45



แสดงข้อมูลผลลัพธ์ภาพชื่อ "Lena" ที่ได้จากวิธีการ LBG's

ภาพที่ 46



เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์เพื่อใช้ในการศึกษาเท่านั้น เมื่อผู้ขาดเห็นไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 7

สรุปผลและแนวทางการพัฒนา

สรุปผลการทดลอง

จากผลการทดลองที่ได้เราสามารถสรุปเป็นข้อๆ ได้ดังนี้

1. เมื่อเปรียบเทียบตัวเลขแสดงเวลาที่ใช้ในการประมวลผลแบบ Linear Approach กับแบบ LBG's แล้วจะเห็นว่าแตกต่างกันอย่างมีนัยสำคัญ ผลการทดลองของภาพที่ใช้เวลาใกล้เคียงกันมากที่สุดคือภาพ "Cell" มีอัตราส่วนของเวลาที่ใช้ประมวลผลเท่ากับ $1123.4/313.46 \approx 3.58$ เท่า และผลการทดลองของภาพที่ใช้เวลาแตกต่างกันมากที่สุดคือภาพ "Girl" มีอัตราส่วนของเวลาที่ใช้ประมวลผลเท่ากับ $1633.4/135.50 \approx 12.05$ เท่า จึงอาจสรุปได้ว่าการใช้การประมาณค่าเชิงเส้นในกระบวนการควอนไทซ์ข้อมูลเวกเตอร์สามารถลดเวลาการประมวลผลลงได้

2. แม้ว่าค่าความคลาดเคลื่อนเฉลี่ย ที่ได้จากวิธี Linear Approach มีค่าสูงกว่าที่ได้จากวิธี LBG's โดยตรง ดูผลการทดลองจากภาพ "Camera" ที่มีความแตกต่างกันมากที่สุดเท่ากับ $154.0807 - 129.1499 = 24.9308$ ขณะที่ผลการทดลองจากภาพ "Cell" มีค่าใกล้เคียงกันมากที่สุดเท่ากับ $12.6932 - 11.6609 = 1.0323$ ทั้งนี้เป็นผลมาจาก "Codebook" ที่ได้จากวิธี Linear Approach นั้นเกิดจากการรวมกัน (Union) ของ Codebook ที่ได้จากกลุ่มย่อยไม่ใช่เกิดจากการประมวลผลข้อมูลอินพุททั้งหมดแบบที่ได้จากวิธี LBG's ดังนั้นเมื่อนำมาเป็นตารางเปรียบเทียบข้อมูลจริงก็อาจมีเวกเตอร์บางค่า (Codeword) ที่ยังให้ค่าความคลาดเคลื่อนที่สูงกว่าอยู่เล็กน้อย ซึ่งผลต่างของค่าความคลาดเคลื่อนในระดับนี้ก็เป็นที่ยอมรับได้ในงานหลายอย่าง โดยเฉพาะเมื่อคำนวณในลักษณะอัตราส่วนสัญญาณต่อสัญญาณรบกวน (Signal per Noise Ratio : SNR) จะยังให้ตัวเลขที่แตกต่างกันน้อยลงไปอีก เช่นในการทดลองนี้ค่าอัตราส่วนสัญญาณต่อสัญญาณรบกวนหาได้จาก

$$SNR = 10 \log_{10} \left(\frac{255^2}{MSE} \right) (dB)$$

3. จากทฤษฎีที่ได้กล่าวไว้ในบทที่ 4 ถึงความพยายามที่แบ่งข้อมูลอินพุทเป็นกลุ่มย่อยหลายๆ กลุ่ม (ให้เท่ากับจำนวน Codeword) ที่สร้างได้ แต่ในทางปฏิบัติ (ในการทดลอง) จำนวนกลุ่มจะลดลงเนื่องจากจำนวนอินพุทเวกเตอร์ในบางกลุ่มมีน้อยเกินต้องถูกนำมาขบรวมกับกลุ่มอื่น ยิ่งการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทดลองโคมีน้อยกลุ่มและแต่ละกลุ่มมีจำนวนอินพุทเวกเตอร์มาก (เกิดจากนำหลายกลุ่มมารวมกัน) ก็จะทำให้การประมวลผลของแต่ละกลุ่มช้าซึ่งมีผลให้เวลารวมในการประมวลผลเพิ่มขึ้น จากผลการทดลองของภาพ "Camera" มีจำนวนกลุ่ม 24 กลุ่มมีอัตราส่วนของเวลาที่ใช้ประมวลผลเท่ากับ $1711.0/262.11 \approx 6.53$ เท่าและผลการทดลองของภาพ "Cell" มีจำนวนกลุ่ม 25 กลุ่มมีอัตราส่วนของเวลาที่ใช้ประมวลผลเท่ากับ 3.58 เท่า สรุปได้ว่าจำนวนกลุ่มข้อมูลมีผลต่อเวลาที่ใช้ในการประมวลผลด้วย

4. แม้ว่าการทดลองนี้จะไม่ได้แสดงให้เห็นถึงความแตกต่าง แต่ในทางทฤษฎีก็สามารถตรวจสอบได้ว่าการจัดเรียงข้อมูลจาก Block ของรูปภาพขนาด 4×4 ให้เป็นอินพุทเวกเตอร์ (1×16) จะมีความสัมพันธ์กับการประมวลผลของวิธี Linear Approach ด้วย ยกตัวอย่างเช่นหากเราเรียงข้อมูลของ Block จากซ้ายไปขวาจะแตกต่างจากการเรียงข้อมูลจากบนลงล่าง (ยกเว้น Block ข้อมูลนั้นมีลักษณะเป็นเมตริกซ์ที่สมมาตร) เพราะเมื่อนำเวกเตอร์ที่ได้จากการเรียงนั้นมาหาค่าประมาณเชิงเส้นจะให้ค่าที่แตกต่างกันโดยสิ้นเชิงและทำให้การแบ่งกลุ่มของอินพุทเวกเตอร์ทั้งหมดเปลี่ยนแปลงไปด้วย

ปัญหาที่พบ

การทดลองครั้งนี้ได้ประสบปัญหาอยู่บางประการ ขอชี้แจงเป็นข้อๆดังนี้

1. ในการทดลองกับข้อมูลที่มีลักษณะคล้ายคลึง/เหมือนกันจำนวนมาก เช่นการทดลองกับภาพ "Cell" จะเกิดเหตุการณ์ที่ข้อมูลจำนวนมากมารวมกันหนาแน่นในบางกลุ่ม (จำนวนน้อย) และกลุ่มอื่นที่เหลืออยู่จะมีข้อมูลน้อยเกินไป ไม่สามารถจัดสรร Codeword ให้ได้จะต้องถูกจับมารวมกันในกลุ่ม "Leaf" จนผลสุดท้ายจะแบ่งกลุ่มได้เพียงไม่กี่กลุ่มทำให้ใช้เวลาการประมวลผลของแต่ละกลุ่มนาน จึงให้ผลออกมาดีกว่าวิธี LBG's ไม่มาก

2. มีข้อมูลบางลักษณะที่อาจให้ค่าประมาณเชิงเส้นเท่ากัน โดยที่ข้อมูลจริงแตกต่างกัน เช่น ข้อมูลในลักษณะเป็นคาบ (Periodic) และสมมาตร (Symetry) เช่น สัญญาณ Sine Wave ข้อมูลแบบนี้จะทำให้การใช้วิธี Linear Approach ให้ข้อผิดพลาดเพราะข้อมูลแบบนี้ที่มีความถี่หรือแอมพลิจูดต่างกัน (แสดงว่าสัญญาณต่างกัน) จะถูกนำมาอยู่ในกลุ่มเดียวกันและทำให้เกิดค่าความคลาดเคลื่อนได้ง่าย เป็นลักษณะข้อมูลที่ต้องให้ความสนใจเป็นพิเศษ

3. ปัญหาอื่นที่พบในทางปฏิบัติ เช่นในระหว่างประมวลผลซึ่งทำงานบนโปรแกรม Windows ซึ่งสามารถทำงานหลายอย่างพร้อมกันได้ หากมีการสั่งให้ทำงาน โปรแกรมอื่นขึ้นมา (Multitasking) จะทำให้เวลาการประมวลผลเพิ่มขึ้นเนื่องจากงานอื่นที่เพิ่มเข้ามา ดังนั้นการในทดลองจึงต้องระงับงานอื่นไว้ทั้งหมดไม่ว่าจะเป็นการติดต่อ Network หรือการขยับ Mouse ก็ตามเพื่อให้ได้ผลที่เที่ยง
แน่นอนที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แนวทางการพัฒนา

จากการทดลองครั้งนี้ก่อให้เกิดแนวทางที่จะพัฒนาด้านนี้ต่อไปอีกหลายเรื่องดังนี้

1. การแบ่งกลุ่มข้อมูล จากการทดลองที่ผ่านมาเราได้จัดแบ่งกลุ่มข้อมูลโดยแบ่งออกเป็น 2 ด้าน (2 คอลัมน์) ด้านละเท่าๆกัน (16 ส่วน) ซึ่งการแบ่งลักษณะนี้เป็นการแบ่ง โดยกว้าง โดยให้ขนาดแต่ละกลุ่มเท่ากันโดยมาพิจารณาจำนวนอินพุทเวกเตอร์ในภายหลัง หากมีการพัฒนาให้แต่ละกลุ่มมีขนาดไม่เท่ากันแต่ค่านึงถึงจำนวนอินพุทเวกเตอร์แต่ละกลุ่มใกล้เคียงกันแทนก็จะทำให้ผลการทดลองดีขึ้น
2. งานวิจัยชิ้นนี้ได้ใช้หลักการประมาณค่าเชิงเส้นในการแบ่งกลุ่ม ดังนั้นหากมีการใช้งานในลักษณะอื่นๆที่เราารู้รูปแบบของสัญญาณเราก็อาจใช้การประมาณค่าแบบอื่นได้เพื่อให้การประมาณค่ามีความใกล้เคียงกับข้อมูลจริงมากที่สุด (เช่นตัวอย่างการประมาณค่าในบทที่ 4)
3. ปัจจัยสำคัญอย่างหนึ่งในการควอนไทซ์ข้อมูลเวกเตอร์แบบ LBG's นี้ก็คือค่า Initialized Codebook ซึ่งหากเราสามารถกำหนดค่านี้นี้ให้ใกล้เคียงกับข้อมูลตั้งแต่แรกแล้วก็จะช่วยลดจำนวนรอบการคำนวณลงได้มาก
4. การจัดแบ่งจำนวน Codebook ตามสัดส่วนของข้อมูลอินพุทเวกเตอร์แต่ละกลุ่มก็เป็นเรื่องที่น่าจะปรับปรุงได้โดยอาจพิจารณากฎเกณฑ์แบบใหม่ที่ไม่จำเป็นตัวเป็นค่าคงที่ (เช่นในการทดลอง) เพื่อจัดให้เกิดเหมาะสมขึ้นได้
5. หากวิธีการควอนไทซ์แบบเวกเตอร์นี้ได้รับการปรับปรุงจนมีประสิทธิภาพที่ดีขึ้นแล้ว เราอาจออกแบบแผงวงจรรวมที่ประมวลผลการควอนไทซ์นี้โดยตรงได้เพื่อช่วยเร่งความเร็วขึ้นไปอีก

บรรณานุกรม

1. JAE S.LIM, "Two-Dimensional Signal and Image Processing", Prentice Hall, 1990
2. Allen Gersho, Robert M.Gray, "Vector Quantization and Signal Compression", Kluwer Academic Publishers, 1992
3. Roger J. Clarke, "Signal Processing and its Application : Digital Compression of Still Images and Video", Academic Press, 1995
4. The Math Works Inc., "The Student Edition of MATLAB[®] version 4 : User's Guide", Prentice Hall , 1995
5. S. Tretriluxana, M. Sangworaslip, P. Koosirivanichakorn, S. Choomchuay, "A Linear Approach in Vector Quantization", RESTEC'96, Regional Symposium on Telecommunications Electronics Circuits and Systems, Faculty of Engineering King Mongkut's Institute of Technology Ladkrabang Bangkok, Thailand, pp c45-c48, 6-8 July, 1996
6. S. Tretriluxana, M. Sangworaslip, S. Srihawatanakul, S. Choomchuay, "A Linear Approach in Vector Quantization", ROVPIA'96, International Conference on Robots, Vision and Parallele Processing for Industrial Automation, Ipoh, Perak, Malaysia, vol.2 of 2, pp. 912-915, 28-30 November, 1996
7. N.M. Nasrabadi, R.A. King, "Image Coding Using Vector Quantization : A Review", IEEE Trans. Commun., vol.38, pp. 957-971, Aug. 1988



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก

Source Code ("M" files) ที่ใช้ในการทดลอง

โปรแกรมที่ใช้พัฒนานี้จะแยกเป็น 2 ส่วนคือตั้งแต่หน้า 65-72 เป็นส่วนที่คำนวณวิธี Linear Approach และหน้า 73-77 เป็นส่วนที่คำนวณวิธี LBG's ทั้งสองส่วนนี้อาจมีชื่อไฟล์ที่ซ้ำกันแต่การทำงานจะข้อแตกต่างกันยกเว้นโปรแกรมดังต่อไปนี้

1. DATA2PIC.M
2. PIC2DATA.M
3. MSE.M

ซึ่งใช้ร่วมกันในทั้งสองวิธีจึงได้พิมพ์ไว้เพียงส่วนเดียวเท่านั้น



ART.M

```
function [cb,time]=art(s,left,n,data)
% Usage : [cb time]=art(s,left,n,data)
% VQ by ART's Alogithm Fn.
% s : "S" matrix for group of data
% left : "S" matrix for ungroup data
% n : number of codewords equal to "n" in "CLASSIFY"
% data : Data from "PIC2DATA"
% cb : Complete Codebook for all data
% time : Processing Time
```

```
tic;
[row col]=size(data);
ratio=ceil(row/n);
[row col]=size(s);
cb=[ ];
c=0;
for a=1 : row
    b=round((s(a,1))/ratio);
    c=c+b;
    cb=[cb;make_cb(b,s(a,:),data)];
end
cb=[cb;make_cb(n-c,left,data)];
time=toc;
```

AVGDIST.M

```
function D=avgdist(cb,data,s)
% Usage : D=avgdist(cb,data,s)
% Average Distortion "D" Calculation Fn.
% D : Average Distortion
% cb : Codebook
% data : Data in this group
% s : "S" matrix for the data
```

```
[row col]=size(cb);
D=0;
for a=1 : row
    for b=2 : s(a,1)+1
        D=D+sum((data(s(a,b),:)-cb(a,:)).^2);
    end
end
temp=sum(s);
D=D/temp(1,1);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CHK_MIN.M

```
function y=chk_min(cb, cb2, data)
% Usage : y=chk_min(cb, cb2, data)
% Function for check minimum error between DATA (1x16) and CB (256x16)
% New Algorithm for finding minimum distortion
% If there is more than one row make minimum distortion,
% this function detect only the FIRST ROW !
% MINIMUM SUM of (Ai-Bi)^2 = MINIMUM SUM of (Bi^2 - 2AiBi) ; when Ai is all the same!
% y : The First Row of 'cb' that make minimum distortion (Scalar)
% cb : Codebook (row x 16)
% cb2 : cb*cb' (row x 1)
% data : 1 row of data (1 x 16 matrix)

data=data';
AX=cb*data;
[result y]=min(cb2-(2*AX));
```

CLASSIFY.M

```
function [s,left,time] =classify(x, n)
% Usage : [s left time]=classify(x, n)
% Classify input data to "n" codewords and store it in "S" matrix
% s : "S" matrix
% left : "S" matrix for group of vector which less than "th"
% x : Input Data (row x 2)
% n : number of codewords to be created
% *** it mube be square root to an integer ***
```

```
tic
n2=n^0.5;
if n2 ~= round(n2)
    s=-1;
    left=-1;
else
    peak=[max(x) min(x)];
    delta_l=(peak(1,1)-peak(1,3))/n2;
    delta_r=(peak(1,2)-peak(1,4))/n2;
    edge_l=peak(1,3);
    edge_r=peak(1,4);
    for a=1 : n2-1
        edge_l(a+1)=a * delta_l + peak(1,3);
        edge_r(a+1)=a * delta_r + peak(1,4);
    end
    edge_l(n2+1)=peak(1,1);
    edge_r(n2+1)=peak(1,2);
```

```
[row col]=size(x);
d=zeros(n,1);
for a=1 : row
    loopbrk=0;
```

```
for b=1 : n2
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if x(a,1) <= edge_l(b+1)
    for c=1 : n2
        if x(a,2) <= edge_r(c+1)
            index=(b-1)*n2 + c;
            d(index,d(index,1)+2)=a;
            d(index,1)=d(index,1)+1;
            loopbrk=1;
            break;
        end
    end
    if loopbrk == 1
        break;
    end
end
end
end

```

```

th=ceil(row/n);
a=1;
left=[ ];
for b=1 : n
    if d(b,1) >= th
        s(a,:)=d(b,:);
        a=a+1;
    elseif d(b,1) > 0 & d(b,1) < th
        for c=2 : d(b,1)+1
            left=[left,d(b,c)];
        end
    end
end
[row col]=size(left);
left=[col,left];
end
time=toc;

```

DATA2PIC.M

```

function pic=data2pic(cb,data,x,y,mode)
% Usage : pic=data2pic(cb,data,x,y,mode)
% Conver Data to Picture Function
% pic : Complete Picture
% data : Raw Data from "PIC2DATA" Fn.
% cb : Complete Codebook
% x : X-size
% y : Y-size
% mode : Mode of Data Reconstruction
% if mode = "1" : Restore data in "Left-to-Right" format
% if mode = "2" : Restore data in "Top-to-Buttom" format

```

```

[row col]=size(data);
if rem(x,4) ~= 0 | rem(y,4) ~= 0 | x*y/16 ~= row
    pic=-1;
else

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

s=modify_s(cb,data);
[row col]=size(cb);
if mode == 1
    for a=1 : row
        for b=1 : 4
            for c=1 : 4
                temp(b,c)=cb(a,(b-1)*4+c);
            end
        end
        for d=2 : s(a,1)+1
            x_pic=(ceil(s(a,d)*4/x)-1)*4+1;
            y_pic=rem(s(a,d)-1,x/4)*4+1;
            pic(x_pic : x_pic+3,y_pic : y_pic+3)=temp;
        end
    end
elseif mode == 2
    for a=1 : row
        for b=1 : 4
            for c=1 : 4
                temp(c,b)=cb(a,(b-1)*4+c);
            end
        end
        for d=2 : s(a,1)+1
            x_pic=(ceil(s(a,d)*4/x)-1)*4+1;
            y_pic=rem(s(a,d)-1,x/4)*4+1;
            pic(x_pic : x_pic+3,y_pic : y_pic+3)=temp;
        end
    end
end
end
end

```

MAKE_CB.M

```

function cb=make_cb(n,s,x)
% Usage : cb=make_cb(n,s,x)
% Make classify codebook fn.
% cb=complete codebook for each group
% n : no of codebook to be made
% s : One row of "S" matrix tell which data will be group
% x : Input data (row x 16) from "PIC2DATA" fn.
% loop : Iteration Loop = 500 "Internal Parameter"
% e : Minimum Chage = 0.005 (0.5%) "Internal Parameter"

```

```

% ---Build New Data---

```

```

x2=[ ];
for a=2 : s(1,1)+1
    x2=[x2;x(s(1,a),:)];
end

```

```

% ---Convert to Linear Curve Fitting---

```

```

for a=1 : s(1,1)
    x3(a,:)=polyfit(0:15,x2(a,:),1);
end

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

% ---Check Redundancy---
x4=x3(1,:);
count=1;
index=1;
for a=2 : s(1,1)
    [row col]=size(x4);
    redun=0;
    for b=1 : row
        if x3(a,:)==x4(b,:)
            redun=1;
            break;
        end
    end
    if redun ~= 1
        x4(row+1,:)=x3(a,:);
        count=count+1;
        index(count)=a;
    end
end

% ---Initial Codebook---
if count <= n
    for a=1 : count
        cb(a,:)=x2(index(1,a),:);
    end
else
    delta=floor(count/n);
    for a=1 : n
        cb(a,:)=x2(index(1,(a-1)*delta+1),:);
    end
end

% ---Make Final Codebook---
loop=500;
e=0.005;
old_avgd=-1;
for a=1 : loop
    s2=modify_s(cb,x2);
    new_avgd=avgdist(cb,x2,s2);
    if old_avgd > 0 & (old_avgd-new_avgd)/old_avgd < e
        break;
    end
    cb=optmz_cb(x2,s2);
    old_avgd=new_avgd;
end
end

```

MODIFY_S.M

```

function s=modify_s(cb, data)
% Usage : s=modify_s(cb, data)
% Function Generate new "S" Matrix
% s : "S" Matrix Output

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
% cb : Codebook
% data : Input Data for the "cb"
```

```
[row col]=size(cb);
for a=1 : row
    cb2(a,:)=cb(a,:)*cb(a,:);
end
s=zeros(row,1);
[row col]=size(data);
for a=1 : row
    index=chk_min(cb, cb2, data(a,:));
    s(index,s(index,1)+2)=a;
    s(index,1)=s(index,1)+1;
end
```

```
function err=mse(org,restore)
% Usage : err=mse(org,restore)
% Mean Square Error Function
% err : Mean Square Error
% org : Original Picture (M x N matrix)
% restore : Restore Picture (M x N matrix)

if size(org) ~= size(restore)
    err=-1;
else
    [row col]=size(org);
    err=sum(sum((org-restore).^2))/(row*col);
end
```

OBTMZ_CB.M

```
function cb=optmz_cb(data,s)
% Usage : cb=optmz_cb(data,s)
% Optimize Codebook Function
% cb : New Codebook
% data : Data in this group
% s : "S" Matrix
```

```
[row col]=size(s);
for a=1 : row
    buffer=0;
    for b=2 : s(a,1)+1
        buffer=buffer+data(s(a,b),:);
    end
    cb(a,:)=round(buffer/s(a,1));
end
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ORDER_1.M

```
function [y,time]=order_1(x)
% Usage : [y time]=order_1(x)
% Linear (1st order) curve fitting Function
% And Adjust the border within 0-255
% y : Data Output (row x 2)
% time : Processing time
% x : Data Input (row x 16)
```

```
tic
[row col]=size(x);
if col ~= 16
    y=-1;
else
    for b=1 : row
        y(b,:)=polyfit(0:15,x(b,:),1);
        temp=[y(b,2) y(b,1)*15+y(b,2)];
        y(b,:)=temp;
        if y(b,1) > 255
            y(b,1)=255;
        elseif y(b,1) < 0
            y(b,1)=0;
        end
        if y(b,2) > 255
            y(b,2)=255;
        elseif y(1,2) < 0
            y(b,2)=0;
        end
    end
end
end
time=toc;
```

PIC2DATA.M

```
function data=pic2data(pic,mode)
% Usage : data=pic2data(pic,mode)
% Convert 'M x N' Picture ----> 'Row x 16' matrix
% data : Raw data = 'Row x 16' matrix
% pic : image from file read
% mode : mode for Conversion
% if mode = "1" : Collect data from Left To Right
% if mode = "2" : Collect data from Top To Bottom
```

```
if mode == 1
    pic2=pic;
    [row col]=size(pic2);
    if rem(row,4) ~= 0 | rem(col,4) ~= 0
        data=-1;
    else
        index=0;
        for a=1 : 4 : row-3
            for b=1 : 4 : col-3
```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

temp=pic2(b:b+3,a:a+3);
temp=temp(:);
index=index+1;
data(index,:)=temp';
end
end
end
elseif mode == 2
[row col]=size(pic);
if rem(row,4) ~= 0 | rem(col,4) ~= 0
data=-1;
else
index=0;
for a=1 : 4 : row-3
for b=1 : 4 : col-3
temp=pic(a:a+3,b:b+3);
temp=temp(:);
index=index+1;
data(index,:)=temp';
end
end
end
end
end

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AVGDIST.M

```
function D=avgdist(cb,data,s)
% Usage : D=avgdist(cb,data,s)
% Average Distortion "D" Calculation Fn.
% D : Average Distortion
% cb : Codebook
% data : Data in this group
% s : "S" matrix for the data
```

```
[row col]=size(cb);
D=0;
for a=1 : row
    for b=2 : s(a,1)+1
        D=D+sum((data(s(a,b),:)-cb(a,:)).^2);
    end
end
temp=sum(s);
D=D/temp(1,1);
```

CHK_MIN.M

```
function y=chk_min(cb, cb2, data)
% Usage : y=chk_min(cb, cb2, data)
% Function for check minimum error between DATA (1x16) and CB (256x16)
% New Algorithm for finding minimum distortion
% If there is more than one row make minimum distortion,
% this function detect only the FIRST ROW !
% MINIMUM SUM of (Ai-Bi)^2 = MINIMUM SUM of (Bi^2 - 2AiBi) ; when Ai is all the same!
% y : The First Row of 'cb' that make minimum distortion (Scalar)
% cb : Codebook (row x 16)
% cb2 : cb*cb' (row x 1)
% data : 1 row of data (1 x 16 matrix)

data=data';
AX=cb*data;
[result y]=min(cb2-(2*AX));
```

INIT_CB.M

```
function [cb1,time]=init_cb(data,n,mode)
% Usage : [cb1 time]=init_cb(data,n,mode)
% Initial Codebook Function
% cb1 : First Codebook (Initialized)
% time : Processing Time
% data : Raw data from "PIC2DATA"
% n : Number of Codeword to be created
% mode : mode of create initial codebook
% if mode = "1" : Sampling independent inputs
```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

% if mode = "2" : Select first independent inputs

```

tic
[row col]=size(data);
if (col ~= 16)
    cb=-1;
elseif mode == 1
    % ---Check Redundancy---
    x=data(1,:);
    count=1;
    index=1;
    loop=row;
    for a=2 : loop
        [row col]=size(x);
        redun=0;
        for b=1 : row
            if data(a,:) == x(b,:)
                redun=1;
                break;
            end
        end
        if redun == 0
            count=count+1;
            x(count,:)=data(a,:);
            index(count)=a;
        end
    end
    % ---Initial Codebook---
    if count <= n
        for a=1 : count
            cb1(a,:)=data(index(1,a),:);
        end
    else
        delta=floor(count/n);
        for a=1 : n
            cb1(a,:)=data(index(1,(a-1)*delta+1),:);
        end
    end
elseif mode == 2
    loop=row;
    cb1(1,:)=data(1,:);
    count=1;
    for a=2 : loop
        [row col]=size(cb1);
        redun=0;
        for b=1 : row
            if data(a,:) == cb1(b,:)
                redun=1;
                break;
            end
        end
        if redun == 0
            count=count+1;
            cb1(count,:)=data(a,:);
        end
    end
    if count == n
        break;
    end
end

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    end
  end
end
time=toc;

```

LBG.M

```

function [cb,time,epoch]=lbg(cb1,data)
% Usage : [cb time epoch]=lbg(cb1,data)
% Make classify codebook fn. (LBG's)
% cb : complete codebook
% time : processing time
% epoch : loop of iterative
% cb1 : Initial codebook from "INIT_CB"
% data : Raw Data from "PIC2DATA"
% loop : Iteration Loop = 500...<Internal Parameter>
% e : Minimum Change = 0.005 (0.5%)...<Internal Parameter>

tic
cb=cb1;
loop=500;
e=0.005;
old_avgd=-1;
for epoch=1 : loop
  s=modify_s(cb,data);
  new_avgd=avgdist(cb,data,s);
  if old_avgd > 0 & (old_avgd-new_avgd)/old_avgd < e
    break;
  end
  cb=optmz_cb(data,s);
  old_avgd=new_avgd;
end
time=toc;

```

MAKE_CB.M

```

function cb=make_cb(n,s,x)
% Usage : cb=make_cb(n,s,x)
% Make classify codebook fn.
% cb=complete codebook for each group
% n : no of codebook to be made
% s : One row of "S" matrix tell which data will be group
% x : Input data (row x 16) from "PIC2DATA" fn.
% loop : Iteration Loop = 500 "Internal Parameter"
% e : Minimum Change = 0.005 (0.5%) "Internal Parameter"

```

```

% ---Build New Data---

```

```

x2=[ ];

```

```

for a=2 : s(1,1)+1

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

x2=[x2;x(s(1,a,:))];
end

% ---Convert to Linear Curve Fitting---
for a=1 : s(1,1)
    x3(a,:)=polyfit(0:15,x2(a,:),1);
end

% ---Check Redundancy---
x4=x3(1,:);
count=1;
index=1;
for a=2 : s(1,1)
    [row col]=size(x4);
    redun=0;
    for b=1 : row
        if x3(a,:)==x4(b,:)
            redun=1;
            break;
        end
    end
    if redun ~= 1
        x4(row+1,:)=x3(a,:);
        count=count+1;
        index(count)=a;
    end
end

% ---Initial Codebook---
if count <= n
    for a=1 : count
        cb(a,:)=x2(index(1,a,:));
    end
else
    delta=floor(count/n);
    for a=1 : n
        cb(a,:)=x2(index(1,(a-1)*delta+1,:));
    end
end

% ---Make Final Codebook---
loop=500;
e=0.005;
old_avgd=-1;
for a=1 : loop
    s2=modify_s(cb,x2);
    new_avgd=avgdist(cb,x2,s2);
    if old_avgd > 0 & (old_avgd-new_avgd)/old_avgd < e
        break;
    end
    cb=optmz_cb(x2,s2);
    old_avgd=new_avgd;
end
end

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MODIFY_S.M

```
function s=modify_s(cb, data)
% Usage : s=modify_s(cb, data)
% Function Generate new "S" Matrix
% s : "S" Matrix Output
% cb : Codebook
% data : Input Data for the "cb"
```

```
[row col]=size(cb);
for a=1 : row
    cb2(a,:)=cb(a,:)*cb(a,:);
end
s=zeros(row,1);
[row col]=size(data);
for a=1 : row
    index=chk_min(cb, cb2, data(a,:));
    s(index,s(index,1)+2)=a;
    s(index,1)=s(index,1)+1;
end
```

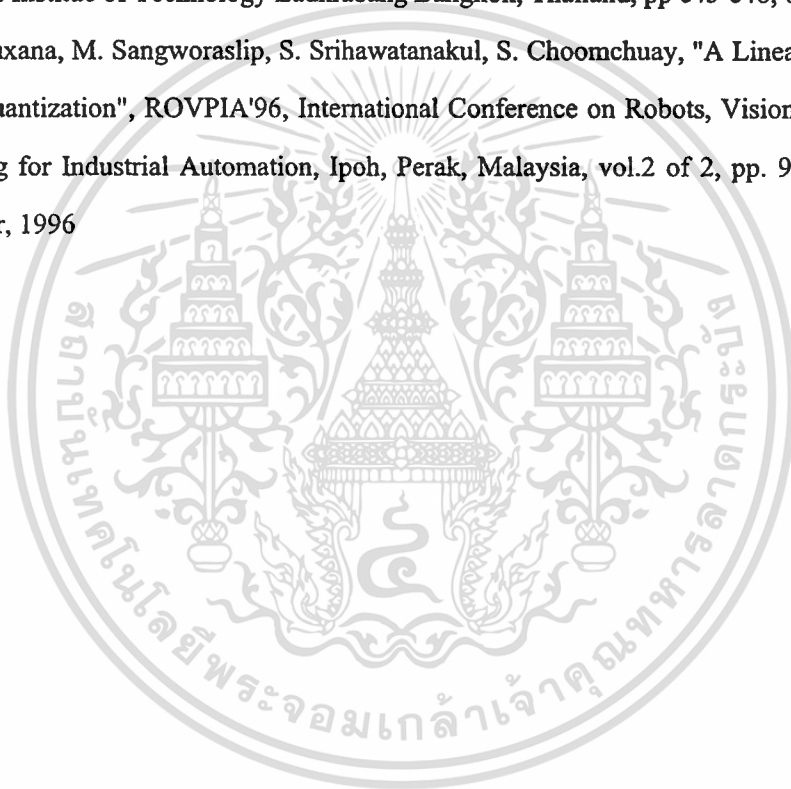
OBTMZ_CB.M

```
function cb=optmz_cb(data,s)
% Usage : cb=optmz_cb(data,s)
% Optimize Codebook Function
% cb : New Codebook
% data : Data in this group
% s : "S" Matrix
```

```
[row col]=size(s);
for a=1 : row
    buffer=0;
    for b=2 : s(a,1)+1
        buffer=buffer+data(s(a,b),:);
    end
    .cb(a,:)=round(buffer/s(a,1));
end
```

ภาคผนวก ข
ผลงานวิจัยที่ได้รับการตีพิมพ์

1. S. Tretriluxana, M. Sangworaslip, P. Koosirivanichakorn, S. Choomchuay, "A Linear Approach in Vector Quantization", RESTEC'96, Regional Symposium on Telecommunications Electronics Circuits and Systems, Faculty of Engineering King Mongkut's Institute of Technology Ladkrabang Bangkok, Thailand, pp c45-c48, 6-8 July, 1996
2. S. Tretriluxana, M. Sangworaslip, S. Srihawatanakul, S. Choomchuay, "A Linear Approach in Vector Quantization", ROVPIA'96, International Conference on Robots, Vision and Parallele Processing for Industrial Automation, Ipoh, Perak, Malaysia, vol.2 of 2, pp. 912-915, 28-30 November, 1996



ประวัติผู้เขียน

นายสุรเดช ตรีไตรลักษณะ เกิดวันที่ 18 สิงหาคม 2513 ที่จังหวัดกรุงเทพมหานคร สำเร็จ การศึกษาระดับประถมและมัธยมศึกษาที่โรงเรียนอัสสัมชัญธนบุรี กรุงเทพมหานคร และระดับ ปริญญาตรีวิศวกรรมศาสตรบัณฑิต (อิเล็กทรอนิกส์) จากสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณ ทหารลาดกระบังในปีการศึกษา 2535 ภายหลังจากสำเร็จการศึกษาได้เข้ารับราชการต่อที่สถาบัน แห่งนี้ ปัจจุบันดำรงตำแหน่งอาจารย์ประจำภาควิชาอิเล็กทรอนิกส์ คณะวิศวกรรมศาสตร์ สถาบัน เทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

