

บทที่ 1

แนะนำสู่การประมวลผลภาพดิจิทัล

Introduction to Digital Image Processing

1.1 แนะนำ

การมองเห็นของมนุษย์เป็นสิ่งที่สำคัญและเป็นกลไกการรับภาพที่ซับซ้อนอย่างหนึ่ง ซึ่งจะให้ข้อมูลที่มีความจำเป็นสำหรับใช้ในงานง่าย ๆ (ตัวอย่างเช่น การจดจำวัตถุ)และสำหรับงานที่มีความซับซ้อน(ได้แก่ การวางแผน การตัดสินใจ การค้นคว้าทางวิทยาศาสตร์ การพัฒนาทางด้านความคิด) ดังคำสาส์นของจีนกล่าวไว้ว่า "รูปภาพสามารถแทนคำได้เป็นพัน ๆ คำ" รูปภาพมีบทบาทมากสำหรับองค์กรต่าง ๆ เช่น หนังสือพิมพ์ โทรทัศน์ ภาพยนตร์ซึ่งได้ใช้ภาพ(ภาพนิ่ง ภาพเคลื่อนไหว)เป็นสื่อนำเสนอข้อมูลข่าวสารต่าง ๆ สิ่งที่น่าสนใจของข้อมูลที่เกี่ยวข้องกับการมองเห็นหรือข้อมูลภาพนั้นก็คือกระบวนการประมวลผลภาพ (Image Processing) โดยใช้ดิจิทัลคอมพิวเตอร์

ความพยายามทางด้านการประมวลผลภาพได้เริ่มขึ้นในปี 1964 ณ ห้องแลป Jet Propulsion (Pasadena California)ซึ่งได้นำการบวนการประมวลผลภาพมาใช้ในการพิจารณาภาพถ่ายดาวเทียมของดวงจันทร์ ต่อมาได้มีการตั้งสาขาทางวิทยาศาสตร์สาขาใหม่มีชื่อว่า Digital image processing หลังจากนั้นงานทางด้านการประมวลผลภาพก็พัฒนาขึ้นเรื่อย ๆ และใช้กันอย่างกว้างขวางสำหรับงานในหลาย ๆ ด้านตัวอย่างเช่นทางได้สื่อสารโทรคมนาคม การสื่อสารทางโทรทัศน์ ทางด้านการพิมพ์ ทางด้านการกราฟฟิก การแพทย์ และการค้นคว้าทางวิทยาศาสตร์

Digital image processing จะเกี่ยวกับการแปลงข้อมูลภาพให้อยู่ในรูปแบบข้อมูลดิจิทัล(Digital format)ซึ่งสามารถที่จะนำเอาข้อมูลนี้จัดผ่านกระบวนการต่าง ๆ ด้วยดิจิทัลคอมพิวเตอร์ได้ ในระบบของดิจิทัล อินพุตและเอาพุตของระบบจะอยู่ในรูปแบบดิจิทัลเท่านั้น

Digital image analysis จะเกี่ยวกับวิธีการอธิบายและการจดจำข้อมูลภาพดิจิทัล ซึ่งอินพุตของระบบจะเป็นข้อมูลภาพดิจิทัลและเอาพุตจะเป็นเครื่องหมายที่ใช้แทนข้อมูลภาพดิจิทัลเหล่านั้น ในการวิเคราะห์ภาพมีอยู่หลายวิธีด้วยกันที่ได้นำมาจากการทำงานของตามนุษย์(human vision)นั่นก็คืองานทางด้าน Computer Vision เป็นลักษณะเดียวกับ Digital image analysis นั่นเอง การมองเห็นของมนุษย์นับว่าเป็นกระบวนการที่ซับซ้อนซึ่งลักษณะเทคนิคโดยทั่ว ๆ ไปในกระบวนการ Digital image analysis และ Computer Vision จะค่อนข้างซับซ้อนเช่นกัน

1.2 รูปร่างของภาพ (Image Shape)

วัตถุที่มีอยู่ตามธรรมชาติและที่มนุษย์สร้างขึ้นมีรูปร่างที่แตกต่างกันไป ทั้งที่เป็นรูปทรงเรขาคณิตและไม่เป็นรูปทรงเรขาคณิต ในศาสตร์ของการประมวลผลภาพนั้น การกำหนดขอบเขตของภาพทุกภาพให้อยู่ในรูปสี่เหลี่ยม (Rectangular image model) เป็นวิธีที่นิยมใช้กันมากที่สุด เนื่องจากทำให้การอ่านภาพ การจัดเก็บข้อมูลภาพในหน่วยความจำ และการแสดงภาพออกทางอุปกรณ์ต่าง ๆ เป็นไปได้อย่างมีประสิทธิภาพ

การเก็บข้อมูลภาพลงหน่วยความจำของคอมพิวเตอร์สามารถทำได้โดยการจองหน่วยความจำของเครื่องไว้ในรูปของตัวแปรอะเรย์ (array) โดยค่าในแต่ละช่องของอะเรย์แสดงถึงคุณสมบัติของจุดภาพ (pixel) และตำแหน่งของช่องอะเรย์เป็นตัวกำหนดตำแหน่งของจุดภาพ

สมมติให้ Image เป็นตัวแปรแบบอะเรย์ขนาด $M \times N$ (M แถว และ N คอลัมน์) ที่ใช้เก็บภาพขนาด $M \times N$ จุด (M จุดในแนวนอน และ N จุดในแนวตั้ง) ค่าสี (หรือความสว่าง ในกรณีที่เป็นภาพ grey level) ของจุดภาพในแถวที่ 5 คอลัมน์ที่ 4 จะตรงกับค่าของ Image(5,4) จะเห็นว่าเราใช้ตำแหน่งของจุดภาพทั้งสองแกนเป็นตัวชี้ค่าข้อมูลในอะเรย์

จากการใช้หน่วยความจำเพื่อการเก็บภาพในลักษณะที่กล่าวมา เนื้อที่ในการเก็บภาพสามารถคำนวณได้จาก $M \times N \times g$ เมื่อ g เป็นจำนวนเต็มที่แทนจำนวนบิตของข้อมูลในแต่ละจุดภาพ ตัวอย่างถ้า g มีค่าเท่ากับ 8 บิต เราจะสามารถเก็บความแตกต่างของระดับสีที่เป็นไปสูงสุด 256 ระดับ ค่า M และ N จะเป็นตัวบอกถึงความละเอียดของภาพ สำหรับคอมพิวเตอร์ทั่วไปในระบบ VGA (Video Graphic Array) จะมีขนาด 640x480, 800x600 และ 1024x768 จุด เป็นต้น การกำหนดความละเอียดจะขึ้นอยู่กับงานที่จะใช้ ในงานบางอย่างใช้ความละเอียดแค่ 30 x 50 จุด ก็พอแล้วแต่ในงานบางชนิด ใช้ความละเอียดถึง 1000 x 1000 จุด ก็ยังไม่พอ

ปกติแล้วในการเก็บข้อมูลภาพโดยเครื่องมือต่าง ๆ จะเก็บตามมาตรฐานของโทรทัศน์ซึ่งมีอัตราส่วน x ต่อ y เท่ากับ 4:3 สำหรับเครื่องมือเก็บข้อมูลภาพที่ไม่เป็นไปตามอัตราส่วน 4:3 เมื่อนำภาพนี้ไปแสดงในจอภาพมาตรฐานจะทำให้ภาพที่แสดงนั้นมีขนาดของจุดภาพไม่เป็นสี่เหลี่ยมจัตุรัสเช่นในบางระบบอาจจะใช้ความละเอียดในการแสดงเท่ากับ 640 x 512 ซึ่งจะทำให้ขนาดของจุดภาพที่ได้มีขนาดของด้านกว้างมีความยาวมากกว่าด้านสูง ซึ่งลักษณะดังกล่าวนี้เป็นหัวข้อที่ต้องสนใจสำหรับการเขียนโปรแกรมทางด้านกราฟฟิกและการจัดการข้อมูล

จำนวนสีสูงสุดที่เป็นไปได้ของแต่ละจุดภาพขึ้นอยู่กับจำนวนบิตที่ใช้ เมื่อมีการกำหนดให้ขนาดของบิตต่อจุด มากขึ้นจะทำให้จำนวนของสีมากขึ้นด้วย ตัวอย่างเช่น

$$1 \text{ บิต} = 2^1 = 2 \text{ สี}$$

$$2 \text{ บิต} = 2^2 = 4 \text{ สี}$$

$$4 \text{ บิต} = 2^4 = 16 \text{ สี}$$

$$8 \text{ บิต} = 2^8 = 256 \text{ สี}$$

$$16 \text{ บิต} = 2^{16} = 65536 \text{ สี เป็นต้น}$$

สำหรับการแสดงข้อมูลภาพที่มีขนาด 1 บิตและ 8 บิตนั้นจะมีความทำงานที่ใกล้เคียงกันเนื่องจากหน่วยประมวลผลจะไม่สามารถจัดการกับข้อมูลที่เป็นบิตเดียว ๆ ได้ดังนั้นในการแสดงข้อมูลออกทางจอภาพตัวโปรเซสเซอร์จะทำการก๊อปปี้ข้อมูลทั้ง 8 บิต (1 Byte) ส่งให้กับจอภาพซึ่งในกรณีที่ Pixel มีขนาด 1 บิต เมื่อโปรเซสเซอร์จะทำงานกับบิตแรกที่ต้องการแล้วก็จะทำการก๊อปปี้ข้อมูลชุดใหม่ทันทีโดยที่ไม่เกี่ยวกับข้อมูลอีก 7 บิตที่เหลือส่วนในกรณี Pixel ที่มีขนาด 8 บิต โปรเซสเซอร์จะทำการก๊อปปี้ข้อมูลชุดใหม่ก็ต่อเมื่อโปรเซสเซอร์ทำงานกับทุกบิตแล้ว

ตัวอย่างสำหรับระบบที่มีความละเอียดเท่ากับ 800x600 และมีขนาด 16 บิตต่อ Pixel จะสามารถแสดงสีได้ทั้งหมด 65536 ระดับและต้องใช้เนื้อที่ในการเก็บเท่ากับ 800x600x16 บิต

1.3 มาตรฐานของสี

มาตรฐานของสีที่ใช้อยู่ในปัจจุบันมีอยู่หลายระบบด้วยกัน ทั้งนี้จะขึ้นอยู่กับการนำไปใช้ แต่โดยทั่วไปแล้วทุกมาตรฐานจะมีแนวคิดเดียวกันคือ การแทนจุดสีด้วยจุดที่อยู่ในสเปส 3 มิติ โดยจะมีแกนอ้างอิงสำหรับจุดสีนั้นในสเปสซึ่งแต่ละแกนจะมีความเป็นอิสระต่อกัน ตัวอย่างเช่นในระบบ RGB จะมีแกนสีคือ แกนสีแดง เขียว และน้ำเงินในระบบ HLS จะมีแกนเป็น ค่าสี(hue) ความสว่าง(lightness)และความบริสุทธิ์ของสี(saturation)

ตัวอย่างระบบสีที่นิยมใช้กันได้แก่ ระบบ RGB HSV (Hue Saturation Value) และ HLS (Hue Lightness Saturation)

1.3.1 ระบบสี RGB

ระบบสี RGB เป็นระบบสีที่เกิดจากการรวมกันของแสงสีแดง เขียวและน้ำเงินโดยมีการรวมกันแบบ Additive ซึ่งโดยปกติจะนำไปใช้ในจอภาพแบบ CRT (Cathode ray tube) ในการใช้งานระบบสี RGB ยังมีการสร้างมาตรฐานที่แตกต่างกันออกไปที่นิยมใช้งานได้แก่ RGB_{CIE} และ RGB_{NTSC}

ระบบสีแบบ RGB ของ CIE

เป็นระบบสีที่พัฒนาขึ้นโดย CIE (Commission International l 'Eclairage) ซึ่งอ้างอิงสีด้วยสีแดงที่ 700 nm สีเขียวเท่ากับ 546.1 nm และสีน้ำเงิน 435.8 nm

ระบบสีแบบ RGB ของ NTSC

เป็นระบบสีที่พัฒนาโดย NTSC (National Television System Committee) เพื่อใช้สำหรับการแสดงภาพของจอภาพแบบ CRT เป็นมาตรฐานสำหรับผู้ผลิตแบบ CRT ให้มีลักษณะเดียวกัน

1.3.2 ระบบสี HSV

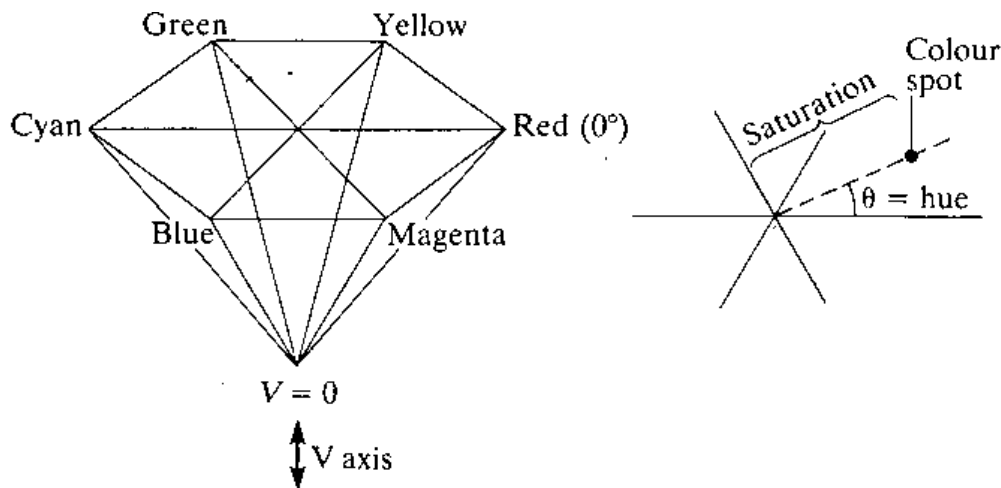
ระบบสี HSV (Hue Saturation Value) เป็นการพิจารณาสีโดยใช้ Hue Saturation และ Value ซึ่ง Hue คือค่าสีของสีหลัก(แดง เขียวและน้ำเงิน)ในทางปฏิบัติจะอยู่ระหว่าง 0 และ 255 ซึ่งถ้า Hue มีค่าเท่ากับ 0 จะแทนสีแดง และเมื่อ Hue มีค่าเพิ่มขึ้นเรื่อย ๆ สีก็จะเปลี่ยนแปลงไปตามสเปกตรัมของสีจนถึง 256 จึงจะกลับมาเป็นสีแดงอีกครั้ง ซึ่งสามารถแทนให้อยู่ในรูปขององศาได้ ดังนี้คือ สีแดง = 0 องศา สีเขียวเท่ากับ 120 องศา สีน้ำเงินเท่ากับ 240 องศา

Hue สามารถคำนวณได้จากระบบสี RGB ได้ดังนี้

$$red_h = red - \min(red, green, blue)$$

$$green_h = green - \min(red, green, blue)$$

$$blue_h = blue - \min(red, green, blue) \tag{1.1}$$



รูปที่ 1.2 แสดงระบบสี HSV

จากลักษณะโมเดลของระบบ Hue พบว่าจะมีค่าน้อยหนึ่งค่าที่จะเท่ากับ 0 แต่ถ้ามีสองค่าเท่ากับ 0 แล้ว hue จะเป็นมุมของสี(ค่าสี)มีค่าเป็นไปตามสีที่สามและถ้าทั้งสามสีมีค่าเท่ากับ 0 แล้วจะทำให้ไม่มีค่าของ Hue หรือสีที่ได้จะมีค่าเท่ากับสีขาวนั่นเอง ตัวอย่างเช่น จอภาพขาว-ดำ ถ้าเกิดมีสีใดสีหนึ่งมีค่าเท่ากับ 0 จะทำให้ค่าสีที่ได้เป็นไปตามสีที่เหลือ การให้น้ำหนักในการพิจารณาเมื่อสีแดงมีค่าเท่ากับ 0

$$\frac{(240 \times \text{blue}_h) + (120 \times \text{green}_h)}{\text{blue}_h + \text{green}_h} \quad (1.2)$$

Saturation คือความบริสุทธิ์ของสีซึ่งถ้า Saturation มีค่าเท่ากับ 0 แล้วสีที่ได้จะไม่มี Hue ซึ่งจะเป็นสีขาวล้วนแต่ถ้า Saturation มีค่าเท่ากับ 255 แสดงว่าจะไม่มีแสงสีขาวผสมอยู่เลย

Saturation สามารถคำนวณได้ดังนี้

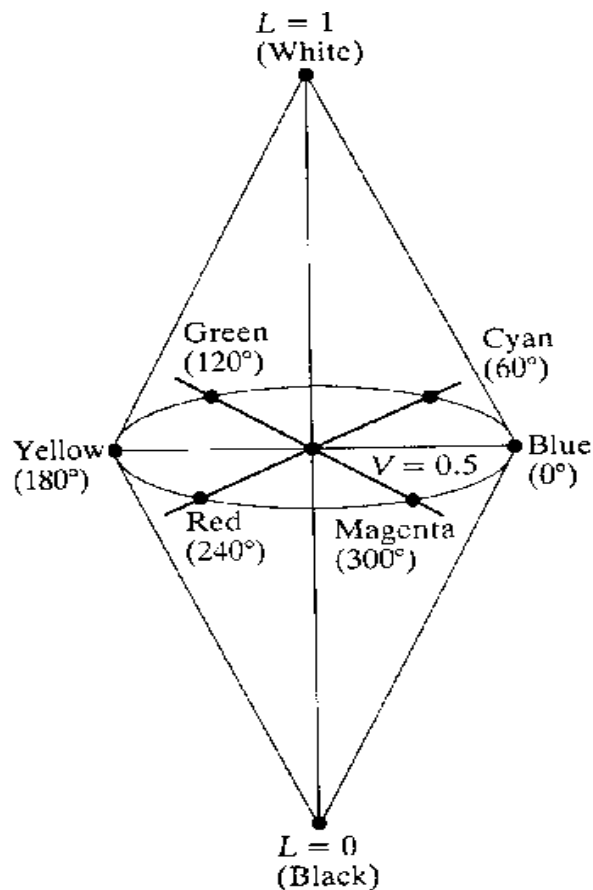
$$\text{Saturation} = \frac{\max(\text{red}, \text{green}, \text{blue}) - \min(\text{red}, \text{green}, \text{blue})}{\max(\text{red}, \text{green}, \text{blue})} \quad (1.3)$$

Value คือความสว่างของสี ซึ่งสามารถวัดได้โดยค่าความเข้มของความสว่างของแต่ละสีที่ประกอบกัน สามารถคำนวณได้จาก

$$\text{value} = \max(\text{red}, \text{green}, \text{blue}) \quad (1.4)$$

1.3.3 ระบบสีแบบ HLS

ระบบสีแบบ HLS (Hue lightness saturation) พัฒนาโดย Teletromix Incorporated จะมีลักษณะคล้ายกับ HSV ดังนั้นคือสีของระบบจะขึ้นอยู่กับ Hue Lightness และ Saturation



รูปที่ 1.3 แสดงระบบสี HLS

Hue คือค่าของสีหลักซึ่งมีสีน้ำเงินอยู่ที่ 0 องศา สีเขียวอยู่ที่ 120 องศา และสีแดงอยู่ที่ 240 องศา

Lightness คือค่าความสว่างซึ่งจะมีค่าเปลี่ยนแปลงตามแนวแกน L โดยที่ L = 0 จะเป็นสีดำ L = 1 จะเป็นสีขาว สามารถคำนวณได้ดังนี้

$$lightness = \frac{\max(red, green, blue) - \min(red, green, blue)}{2} \quad (1.5)$$

Saturation คือความบริสุทธิ์ของสีสามารถหาได้ดังนี้คือ

$$saturation = \begin{cases} \frac{\max(red, green, blue) + \min(red, green, blue)}{\max(red, green, blue) - \min(red, green, blue)} \text{ if } L \leq 0.5 \\ \frac{\max(red, green, blue) - \min(red, green, blue)}{2 - \max(red, green, blue) - \min(red, green, blue)} \text{ if } L > 0.5 \end{cases} \quad (1.6)$$

1.3.4 ระบบสีแบบ CMY

CMY (Cyan Magenta Yellow) เป็นระบบสีที่พัฒนาขึ้นมาใช้สำหรับการพิมพ์ภาพสีโดยมีสีหลักคือสี Cyan Magenta และ Yellow ซึ่งเรียกว่า Subtractive primaries Color (สีแดง เขียวและน้ำเงิน เรียกว่า Additive primaries Color) ระบบสีแบบ CMY สามารถหาได้โดยการนำเอาสีในระบบ RGB ลบกับสีขาวดังนี้คือ

$$\begin{aligned}C &= 1 - R \\M &= 1 - G \\Y &= 1 - B\end{aligned}\tag{1.7}$$

ระบบสี CMY จะนำไปใช้สำหรับการพิมพ์ภาพสีแต่ยังไม่ดีเท่าที่ควรเนื่องจากไม่ยังสามารถสร้างสีดำได้อย่างถูกต้อง ดังนั้นจึงมีการใช้ระบบ CMYK แทนโดย

$$\begin{aligned}K &= \min(C, M, Y) \quad K \text{ เป็นสีที่ 4 แทนสีดำ} \\C &= C - K \\M &= M - K \\Y &= Y - K\end{aligned}\tag{1.8}$$

1.3.5 ระบบสีแบบ YUV

ระบบสีแบบ YUV ใช้สำหรับโทรทัศน์แบบ PAL และ SECAM ซึ่งยังมีใช้อยู่ในหลาย ๆ ประเทศโดย Y คือค่าความสว่างของภาพ ส่วนสัญญาณ U และ V เป็นสัญญาณที่เก็บค่าสีของภาพ ต่อมาได้มีระบบ YIQ มาใช้แทนเนื่องจากพบว่าสัญญาณ I และ Q สามารถลด Bandwidth ได้มากกว่าสัญญาณ U และ V ในขณะที่ได้ภาพที่มีคุณภาพเท่ากัน

1.3.6 ระบบสีแบบ YIQ

เป็นระบบที่ใช้ใน TV Broadcasting สำหรับ NTSC ประโยชน์หลักก็เพื่อให้ใช้งานได้กับโทรทัศน์แบบขาวดำ โดยที่ y คือความสว่างของภาพ ส่วน I และ Q จะเป็นสัญญาณที่เข้ารหัสสีของภาพไว้ดังนั้นสำหรับโทรทัศน์ขาวดำ นั้นสามารถใช้ค่า Y ค่าเดียวก็สามารถได้ภาพที่สมบูรณ์

1.3.7 ระบบสีแบบ XYZ

เป็นระบบสีที่ CIE ได้กำหนดให้มีขึ้นเป็นมาตรฐานเนื่องจากในระบบสี RGB ยังไม่สามารถสร้างสีที่เป็นไปได้ทั้งหมดดังนั้นจึงได้มีตั้งระบบสี XYZ ซึ่งเป็นระบบสีที่สมมุติขึ้น

1.4 การแปลงค่าสีระหว่างระบบสีต่าง ๆ

การแปลงค่าสีระหว่างระบบสามารถทำได้โดยการใช้ Matrix ตัวอย่างเช่น การแปลงสีระหว่างระบบ RGB (ICE) กับระบบสีแบบ XYZ จะมีเมตริกสำหรับการแปลงดังนี้คือ

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.490 & 0.310 & 0.200 \\ 0.177 & 0.813 & 0.011 \\ 0.000 & 0.010 & 0.990 \end{bmatrix} \begin{bmatrix} R_{CIE} \\ G_{CIE} \\ B_{CIE} \end{bmatrix}$$

รูปที่ 1.4

ตัวอย่างโปรแกรม การแปลงสีจากระบบ RGB (CIE) ไปเป็น XYZ

```
int cieRGB_XYZ (r ,g ,b ,x ,y ,z ,N1 ,M1 ,N2 ,M2)
image r,g,b,x,y,z;
int N1,M1,N2,M2;
/* Subroutine to perform cieRGB to XYZ transform
r,g,b : input image buffers
x,y,z: transform buffers
N1, M1: upper left corner coordinates
N2, M2: lower right corner coordinates */
{
int i, j; double R,G,B; double X,Y,Z;
for(i = N1;i< N2;i++)
    for(j = M1;j<M2;j++)
    { R = (double)r[i][j]; G = (double)g[i][j]; B = (double)b[i][j];
      X = 0.490*R + 0.310*B + 0.200* B;
      Y = 0.177*R + 0.813*G + 0.011*B;
      Z = 0.010*G + 0.990*B;
      If(X>255.0) x[i][j]=255; else x[i][j]=(unsigned char)X;
      If(Y>255.0) y[i][j]=255; else y[i][j]=(unsigned char)Y;
      If(Z>255.0) z[i][j]=255; else z[i][j]=(unsigned char)Z;
    }
return(0);
}
```