

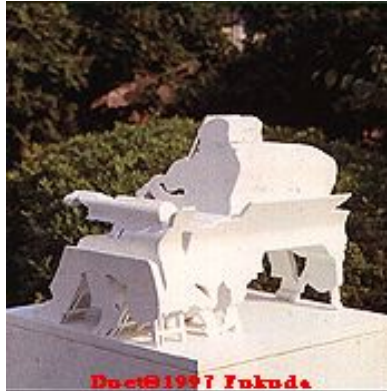
# 1 การรับรู้จากภาพ

เมื่อตาซ้ายเห็นลูกบอลพุ่งไปทางขวา และตาขวาเห็นลูกบอลพุ่งไปทางซ้าย กล้ามเนื้อคอถูกสั่งงานให้ส่ายหัวหลบทันที นี่เป็นเพียงตัวอย่างเล็กน้อย ที่แสดงให้เห็นถึงความสำคัญของระบบประสาทตา ที่มีต่อการใช้ชีวิตอย่างปกติ ในสภาวะแวดล้อมที่ไม่หยุดนิ่ง ประสาทตานั้นเป็นประสาทรับรู้ ที่มีความสำคัญที่สุดอย่างหนึ่งของมนุษย์ เราอาศัยประสาทตาในการประกอบกิจวัตรประจำวันแทบทุกชนิด ข้อมูลที่ได้จากการมองเห็นเป็นส่วนสำคัญในการตัดสินใจว่าจะทำอะไร อย่างไร ซึ่งนอกจากมนุษย์แล้ว สัตว์หลายชนิดก็สามารถรับรู้ด้วยประสาทตาที่ไม่แตกต่างกับเรามากนัก นี่เป็นหลักฐานทางธรรมชาติที่แสดงให้เห็นถึงความสำคัญและความสามารถของการรับรู้จากภาพ สำหรับหุ่นยนต์แล้ว ข้อมูลเกี่ยวกับสิ่งแวดล้อม นับว่ามีความสำคัญเป็นอย่างยิ่งต่อความสำเร็จของการปฏิบัติงานตามเป้าหมายที่วางแผนไว้ ในบทที่ผ่านมา เราได้ศึกษาการรับรู้ของหุ่นยนต์จากอุปกรณ์เซนเซอร์แบบพื้นฐาน ในบทนี้เราจะขอกล่าวถึงการรับรู้โดยผ่านเซนเซอร์อีกชนิดหนึ่ง นั่นก็คือกล้องรับภาพ

ในปัจจุบันก็ยังไม่มีการเข้าใจหลักการทำงานของการประมวลผลภาพที่ได้จากระบบประสาทตาอย่างสมบูรณ์ อย่างไรก็ตาม ก็มีความพยายามในการสร้างระบบที่มีความสามารถในการวิเคราะห์และประมวลผลภาพ โดยเป้าหมายหลักประการหนึ่งก็คือการสร้างระบบที่สามารถบอกได้ว่าภาพสองมิติที่ได้รับมานั้นเป็นภาพของวัตถุใด มีลักษณะในสามมิติอย่างไร ศาสตร์ที่ว่ามีชื่อเรียกกันว่า computer vision จะเห็นได้ว่าเป้าหมายของศาสตร์แขนงนี้นั้นตรงกันข้ามกับ computer graphics ซึ่งมีวัตถุประสงค์หลักที่จะสร้างภาพเหมือนสองมิติจากแบบจำลองของวัตถุที่กำหนดให้ ในขณะที่ปัญหาของ computer graphics นั้นมีคำตอบที่น่าพอใจ ถึงขนาดที่ทำให้เราอาจแยกไม่ออกว่าฉากหนึ่งๆ จากภาพยนตร์เป็นผลงานของคอมพิวเตอร์หรือเปล่า แต่สำหรับ computer vision เราคงต้องรออีกนานพอสมควรสำหรับคำตอบที่จะเป็นที่น่าสนใจในระดับเดียวกัน ทั้งนี้ส่วนหนึ่งก็เพราะธรรมชาติของปัญหาเองที่ต้องการกู้คืนข้อมูลสามมิติจากข้อมูลเข้าสองมิติ ซึ่งเป็นไปไม่ได้นอกจากมีข้อกำหนดเสริมที่เพียงพอ ดังสำนวนที่ว่า “A picture is worth a thousand words” ดูภาพของวัตถุชิ้นเดียวกันจากสามมุมมองในรูปแบบที่ 1.1 และรูปที่ 1.2



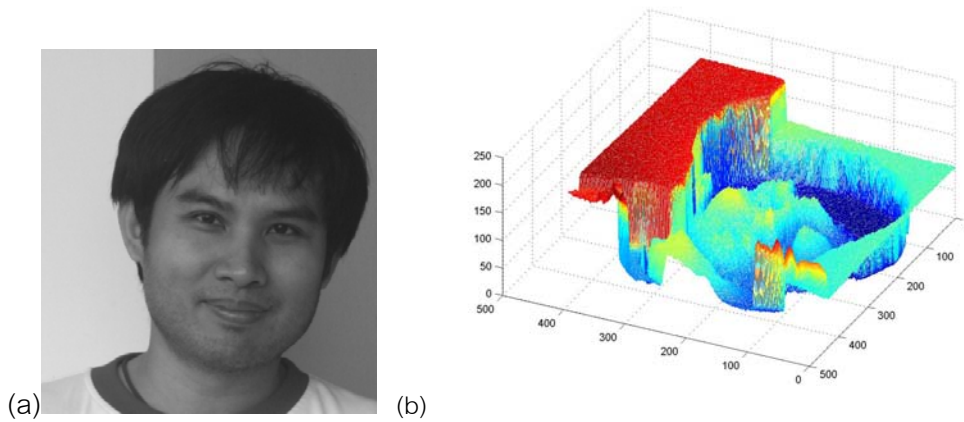
รูปที่ 1.1 รูปของวัตถุเดียวกันจากสองมุม



รูปที่ 1.2 รูปของวัตถุในรูปที่ 1.1 ในอีกมุมมอง

## 1.1 ภาพ (image)

เรารับรู้โลกจริงสามมิติด้วยการรับแสงที่สะท้อนจากสิ่งแวดล้อมผ่านเลนส์แก้วตา และมาตกกระทบที่บริเวณสองมิติบนส่วนของตาที่เรียกว่าเรตินา แสงที่มาตกกระทบที่เรตินานี้เองเป็นข้อมูลเข้าของการประมวลผลเพื่อให้ได้ความเข้าใจในตัวตนของโลกจริง เราเรียกแสงที่ตกกระทบที่เรตินาว่าภาพ ในบทนี้ภาพคือรูปแบบของแสงบนบริเวณสองมิติที่รับได้ด้วยประสาทตาหรืออุปกรณ์รับภาพอื่นๆ ในทางปฏิบัติเราจะพิจารณาระดับความเข้มของแสงหรือระดับพลังงาน ในช่วงความถี่หนึ่งๆ เท่านั้น ถึงแม้ว่าจะไม่สามารถเก็บรายละเอียดของแสงที่เข้ามาได้ทั้งหมด (อาจยังมีพลังงานในช่วงความถี่อื่นๆ ของแสง และไม่สามารถบอกระดับความเข้มของแสงที่ความถี่ต่างๆ) แต่ก็ เป็นวิธีที่ธรรมชาติเลือกใช้ในระบบประสาทตา เช่นประสาทตาของเราเลือกที่รับแสงในสามช่วงความถี่คือแดง เขียว และน้ำเงิน จะเห็นได้ว่าประสาทตาของเรารับรู้สิ่งแวดล้อมโดยใช้แสงเป็นตัวกลาง สิ่งแวดล้อมสามมิติสร้างภาพฉายด้วยแสงลงบนเรตินาสองมิติ ข้อมูลของแสงบนแต่ละจุดของเรตินามีเพียงความเข้มของแสงในช่วงความถี่หนึ่งเท่านั้น ถึงแม้ว่าข้อมูลโลกจริงจะถูกลดทอนลงไปมาก แต่เราก็สามารถที่จะรับรู้ความเป็นสามมิติได้อย่างมีประสิทธิภาพ นับเป็นความน่าทึ่งของธรรมชาติ จากที่กล่าวมาแล้วเราอาจเขียนให้ชัดเจนว่าภาพก็คือฟังก์ชัน  $i(x, y)$  ที่บอกค่าความเข้มของแสงเป็นจำนวนจริง ณ พิกัด  $(x, y)$  ในบริเวณของภาพ ตัวอย่างในรูปที่ 1.3 แสดงภาพเด็กและรูปแบบของความเข้มของแสงสองมิติของภาพนี้ ที่บริเวณมุมบนซ้ายของภาพจะเป็นสีขาว ซึ่งก็คือความเข้มแสงมากที่สุดสอดคล้องกับบริเวณของกราฟที่มีค่าความเข้มของแสงสูงสุด

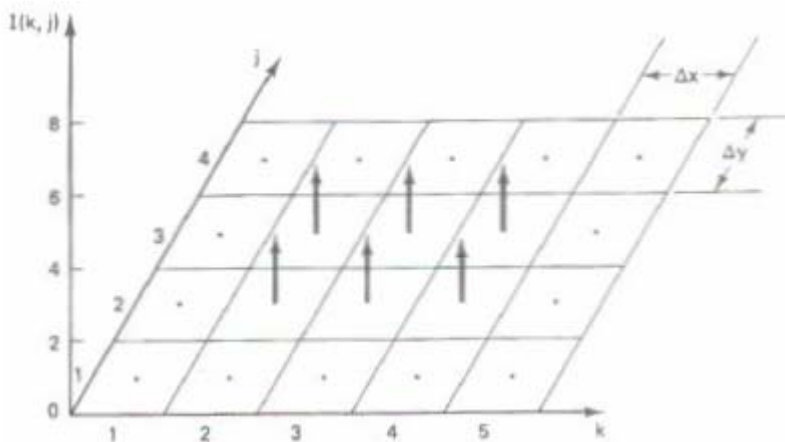


รูปที่ 1.3 ภาพและรูปแบบความเข้มของแสง

ในการที่จะนำรูปแบบของความเข้มแสงไปใช้ในการคำนวณด้วยคอมพิวเตอร์นั้น เราต้องมีวิธีเปลี่ยนข้อมูลแอนะล็อกเป็นข้อมูลดิจิทัลอย่างมีระบบ วิธีที่นิยม คือการใช้แถวลำดับสองมิติ (two dimensional array) ของจำนวนเต็ม ในการแทนรูปแบบของระดับความเข้มของแสง โดยเมื่อสมมติให้  $i(x, y)$  เป็นฟังก์ชันอะนาล็อกที่แทนความเข้มของแสงที่พิกัด  $(x, y)$  ของภาพ เราจะใช้ค่าความเข้มของแสงในแถวลำดับสองมิติที่แถว  $k$  และหลัก  $j$  ดังนี้

$$I_a(k, j) = \frac{\int_0^{\Delta x} \int_0^{\Delta y} i[(k-1)\Delta x + x, (j-1)\Delta y + y] dy dx}{\Delta x \Delta y}$$

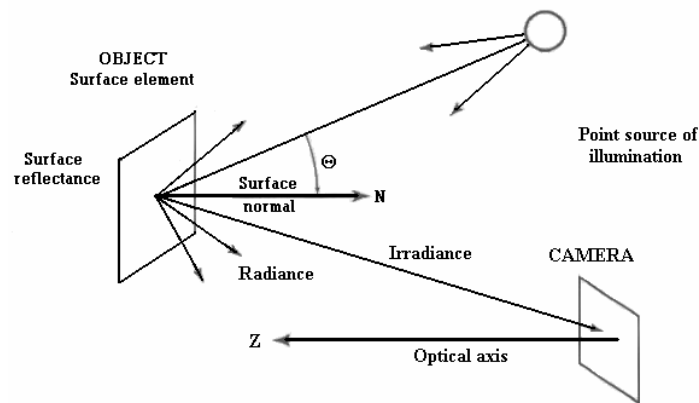
โดยภาพได้ถูกแบ่งออกเป็นตารางเล็กๆ ที่มีขนาด  $\Delta x \times \Delta y$  (ดูรูปที่ 1.4) และค่าแสดงระดับความเข้มของแสง  $I_a(k, j)$  ที่ได้ก็จะถูกเก็บเป็นไบนารี จะมีความละเอียดเท่าไรขึ้นกับจำนวนบิตที่ใช้



รูปที่ 1.4 แถวลำดับสองมิติที่แทนความเข้มของแสง

## 1.2 อุปกรณ์รับภาพ

เพื่อให้จะได้ภาพที่สามารถนำมาวิเคราะห์หาข้อมูลที่มีประโยชน์ เราต้องมีวิธีการบันทึกภาพที่เหมาะสม หลักการคร่าวๆ ของการบันทึกภาพก็เหมือนกับการทำงานของกล้องถ่ายภาพที่เรารู้จักกันดี นั่นก็คือแสงจากต้นทางที่ตกกระทบวัตถุจะสะท้อนและผ่านส่วนรวมแสง โดยแสงที่ผ่านส่วนรวมแสงจะไปตกกระทบที่ส่วนรับภาพ จากนั้นภาพที่ได้ก็จะถูกนำไปประมวลผลต่อไป เราสามารถแยกประเภทของอุปกรณ์รับภาพ ด้วยลักษณะของส่วนรวมแสงและส่วนรับภาพ ต่อไปนี้จะขอกล่าวถึงอุปกรณ์รับภาพแบบต่างๆ โดยจะขอเริ่มจากอุปกรณ์รับภาพธรรมชาติที่ทุกคนรู้จักดีเป็นอันดับแรก

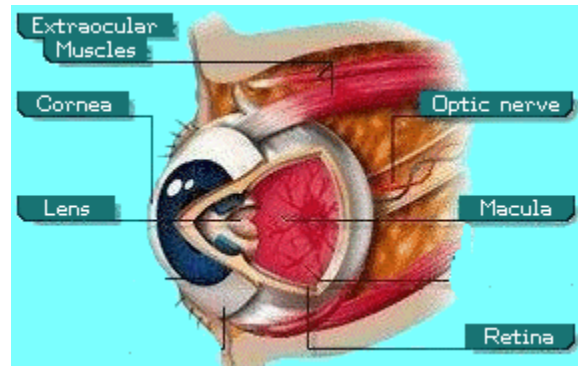


รูปที่ 1.5 ทางเดินของแสงเข้าสู่กล้องถ่ายภาพ

### 1.2.1 ตา

ตาเป็นอวัยวะของมนุษย์ที่ทำหน้าที่รับภาพโดยมีส่วนประกอบดังรูปที่ 1.6 คือ

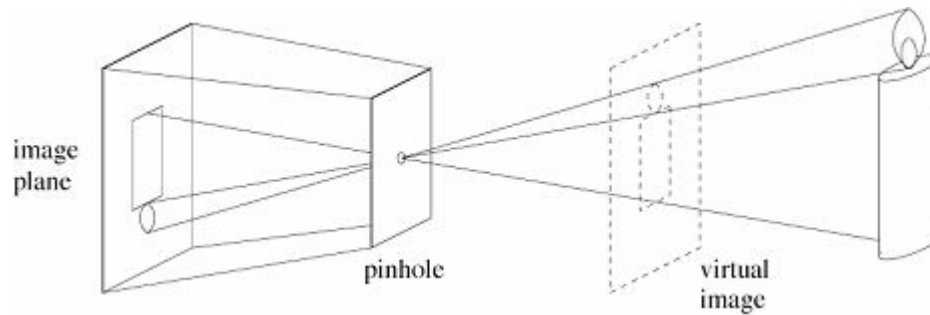
- Extraocular Muscles เป็นกล้ามเนื้อทำหน้าที่ควบคุมการเคลื่อนไหวของลูกตาเพื่อให้อยู่ในทิศทางที่จะรับภาพที่ต้องการได้ เหมือนการปรับมุมกล้องให้ได้พอเหมาะ
- Cornea ทำหน้าที่ เป็นส่วนควบคุมปริมาณแสงที่เข้าสู่ตาให้เหมาะสม เหมือนกับการปรับรูรับแสงของกล้องถ่ายรูปที่จะให้แสงเข้า
- Lens ทำหน้าที่ เป็นส่วนรวมแสงเหมือนเลนส์กล้อง
- Retina เป็นฉากรับแสง ประกอบไปด้วยเซลล์ไวแสงแบบต่างๆ เหมือนฟิล์มในกล้องถ่ายรูป
- Macula ทำหน้าที่ปรับให้เกิดความคมชัดของภาพที่ retina มาเหมือนการปรับ f-stop ของกล้องถ่ายรูป
- Optic nerve ทำหน้าที่ส่งต่อข้อมูลที่รับมาสู่สมอง



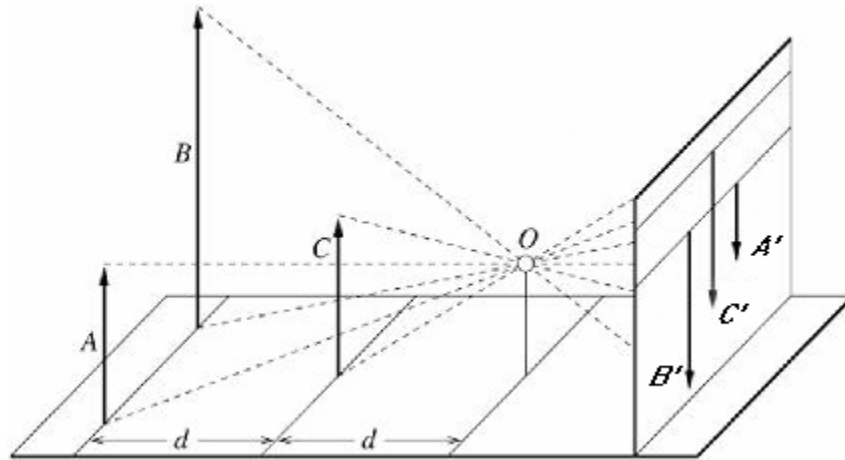
รูปที่ 1.6 ตามนุษย์

### 1.2.2 กล้องรูเข็ม (pinhole camera)

กล้องรูเข็มเป็นอุปกรณ์รับภาพที่อาศัยรูเล็กๆ เพียงรูเดียวให้แสงสามารถลอดผ่านได้ โดยแสงที่กระทบวัตถุจะลอดผ่านรูนี้เป็นเส้นตรง และไปกระทบกับฉากทำให้เกิดเป็นภาพของวัตถุตั้งแสดงในรูปที่ 1.7 ภาพที่ปรากฏที่ฉากจะเป็นภาพกลับข้าง และจะมีขนาดเล็กลงเมื่อวัตถุเคลื่อนห่างรูเข็มออกไป เช่นในรูปที่ 1.8 จะเห็นได้ว่า A และ C มีความสูงเท่ากัน แต่ ภาพของ A บนฉาก จะเล็กกว่าภาพบนฉากของ B ทั้งนี้เพราะ A อยู่ห่างรูเข็มมากกว่า C

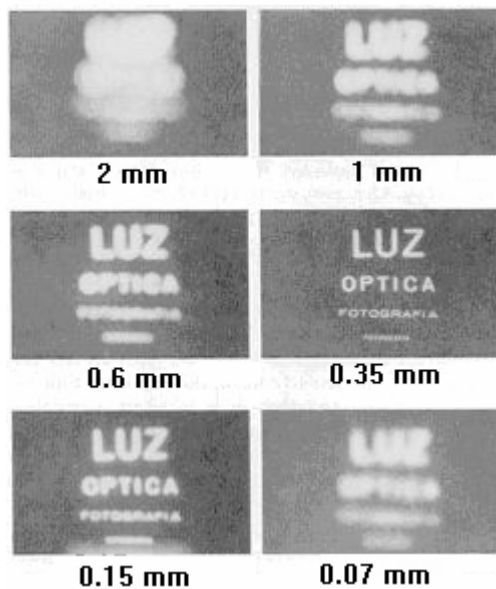


รูปที่ 1.7 กล้องรูเข็ม



รูปที่ 1.8 ภาพที่ได้จากกล้องรูเข็มที่มีรูที่จุด O

กล้องรูเข็มเป็นอุปกรณ์ชิ้นแรกๆ ของมนุษย์ที่ใช้ในการเก็บภาพ ข้อเสียของอุปกรณ์ชิ้นนี้ก็คือ ความเข้มแสงที่กระทบวัตถุมีผลต่อความชัดเจนของภาพที่ได้เป็นอย่างมาก ขนาดของรูของกล้องที่ไม่เหมาะสม ก็เป็นปัจจัยที่ทำให้ภาพพร่ามัวได้ จะเห็นได้จากรูปที่ 1.9 ที่แสดงภาพที่ได้จากกล้องรูเข็มซึ่งขนาดเส้นผ่านศูนย์กลางของรูต่างๆ กัน โดยสำหรับรูขนาด 2mm, 1mm และ 0.6mm เราจะได้ภาพที่พร่ามัว ไม่ชัดเจนเพราะขนาดของรูใหญ่เกินไป ถ้าแสงจากแต่ละจุดของวัตถุผ่านรูเป็นเส้นตรงหลายเส้น ทำให้เกิดภาพของวัตถุหลายภาพซ้อนกัน ภาพที่ได้จึงมัว ไม่ชัดเจน แต่ถ้ารูของกล้องมีขนาดเล็กเกินไปเช่น 0.15mm และ 0.07mm จะทำให้ภาพพร่ามัวเพราะรูของกล้อง มีขนาดเล็กพอที่จะทำให้เกิดการเลี้ยวเบนของแสง (diffraction) ทำให้แสงที่ผ่านรูกระจายออก และเกิดภาพบนฉากที่พร่ามัว

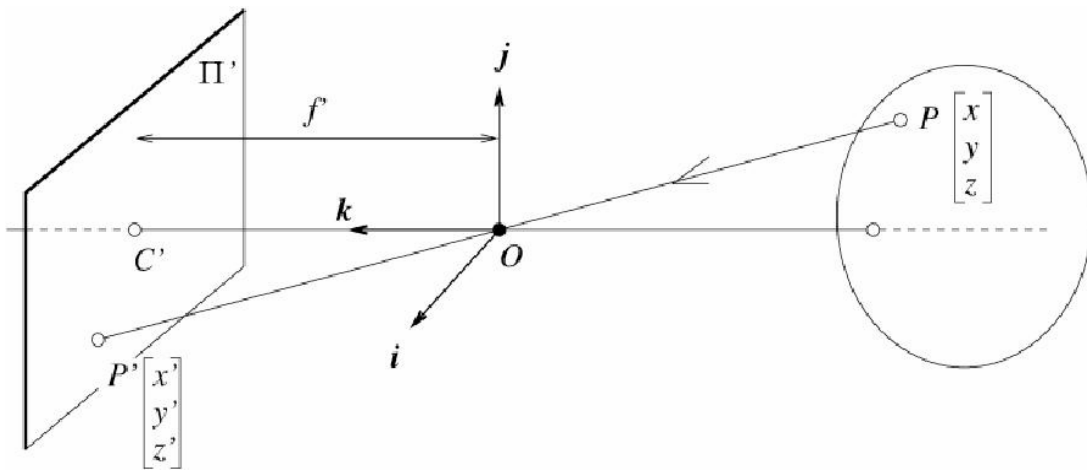


รูปที่ 1.9 ภาพจากกล้องรูเข็มที่ขนาดของรูต่างๆ

เมื่อเราสมมติให้รูเข็มมีขนาดเล็กเป็นจุด เราสามารถคำนวณได้ว่า แสงสะท้อนจากจุดบนวัตถุจะไปกระทบฉากที่ไหน เราเรียกรายการฉายของแสงผ่านจุดรูเข็มที่กำหนดนี้ว่า perspective projection ในรูปที่ 1.10 เราสามารถคำนวณตำแหน่งของ  $P'$  ซึ่งเป็นภาพฉายของ  $P$  ผ่านรูเข็มได้โดยใช้สูตร

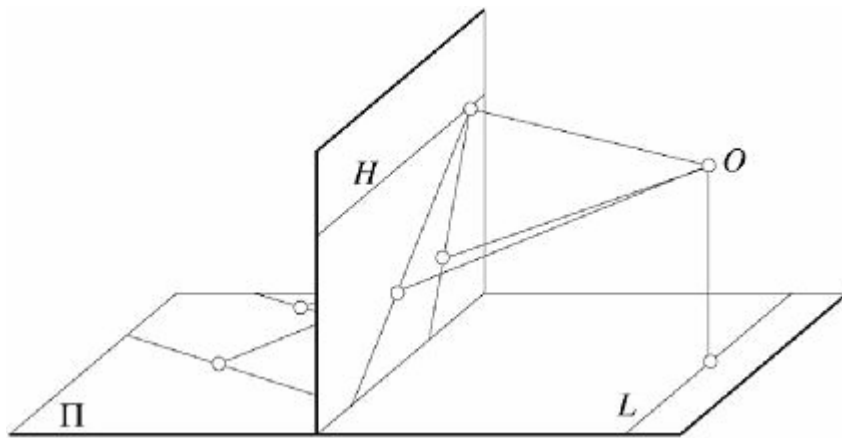
$$P' = \begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = f \begin{pmatrix} -x/z \\ -y/z \\ 1 \end{pmatrix}$$

โดยแกน  $k$  เป็นแกนผ่านจุดรูเข็มและตั้งฉากกับฉากรับแสง



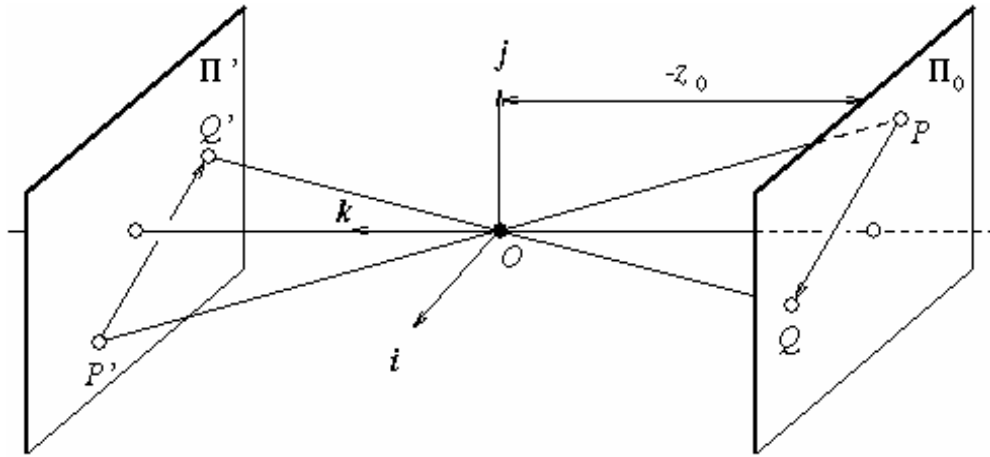
รูปที่ 1.10 เส้นทางเดินของแสงผ่านรูเข็ม

ที่น่าสนใจอย่างหนึ่งคือ perspective projection ของเส้นขนานบนระนาบหนึ่งจะเป็นเส้นที่ตัดกันที่จุดๆ หนึ่ง (นึกถึงเวลาเรามองทางรถไฟที่ทอดยาวออกไป) และไม่ว่าจะเป็นเส้นขนานคู่ไหนบนระนาบนี้ จุดตัดของ perspective projection ของมันจะอยู่บนเส้นตรงเดียวกันเสมอ โดยที่ระนาบที่ผ่านเส้นตรงนี้กับศูนย์กลางการ projection จะขนานกับระนาบของเส้นขนานทั้งหลาย ดูรูปที่ 1.11



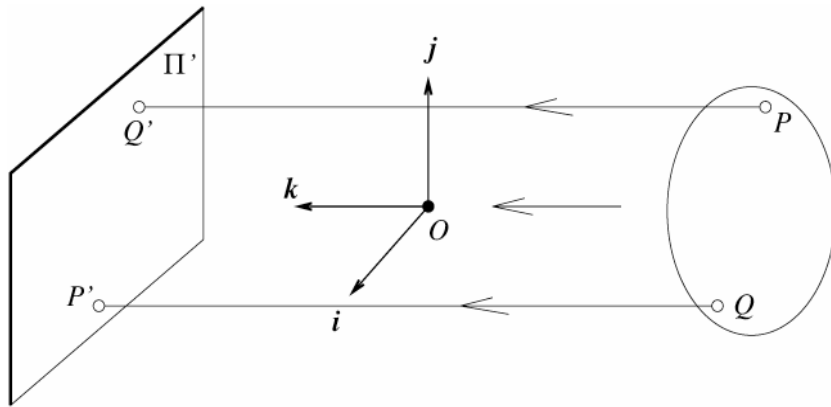
รูปที่ 1.11 perspective projection ของเส้นขนาน

ในกรณีที่วัตถุมีความแตกต่างของความลึกน้อยมาก เมื่อเทียบกับระยะห่างจากศูนย์กลางการ projection เราอาจกำหนดให้ทุกจุดบนวัตถุใช้ระยะห่างจากศูนย์กลางเท่ากันหมดเป็นค่าคงที่ค่าหนึ่ง ผลที่ได้คือ projection ที่สะดวกในการคำนวณ เรียกว่า weak perspective projection ในรูปที่ 1.12 ทุกจุดบนวัตถุถูกกำหนดให้ใช้ระยะห่าง  $-z_0$  จากจุดดูขี้ม



รูปที่ 1.12 weak perspective projection

กรณีพิเศษของ weak perspective projection คือในกรณีที่เราไม่สนใจความลึกของวัตถุในแนวแกน  $k$  เลย นั่นคือทุกจุดบนวัตถุจะมีภาพที่เกิดจากการฉายขนานกับแกน  $k$  ดังรูปที่ 1.13 เราเรียกการฉายแบบนี้ว่า orthographic projection

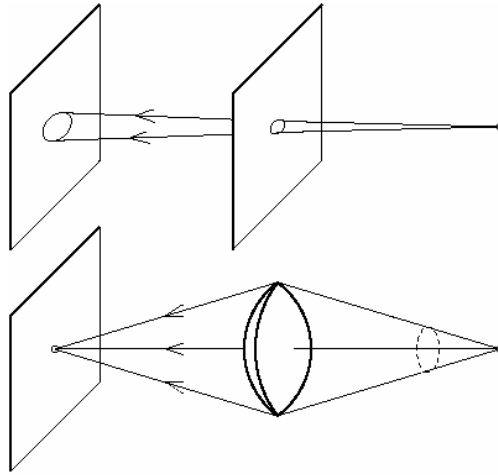


รูปที่ 1.13 orthographic projection



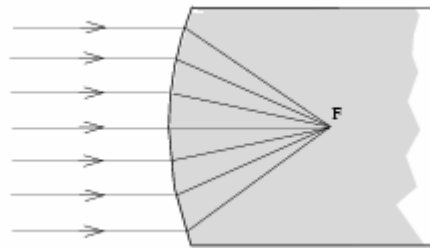
### 1.2.3 เลนส์

ปริมาณแสงที่ผ่านรูของกล้องรูเข็มไปยังฉากมีจำกัด เพราะการขยายขนาดรูให้ใหญ่เพื่อให้ได้ปริมาณแสงที่เพิ่มขึ้น อาจทำให้ได้ภาพมัว ปริมาณแสงที่จำกัดนี้เป็นสาเหตุให้ได้ภาพที่มีมืดและไม่ชัดเจน เราสามารถแก้ปัญหานี้ได้ด้วยการใช้เลนส์เป็นอุปกรณ์รวมแสง โดยใช้หลักการหักเหของแสง (refraction) แทนการให้แสงรอดผ่านรู ดังรูปที่ 1.14



รูปที่ 1.14 การรวมแสงผ่านรูและการหักเหโดยเลนส์

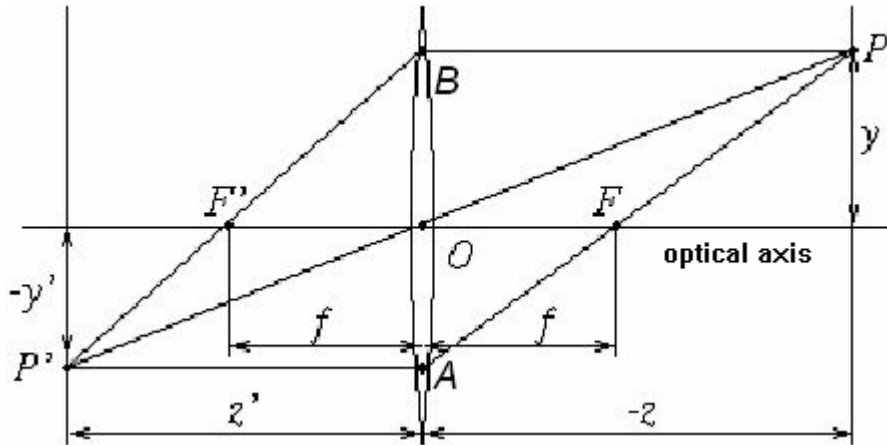
การทำงานของเลนส์ สามารถอธิบายได้ด้วยกฎของสเนล (Snell's law) ซึ่งกล่าวถึงการหักเหของแสง เมื่อเดินทางผ่านตัวกลางที่มีความทึบแสงต่างกัน สำหรับแท่งแก้วใสที่มีผิวเป็นส่วนโค้งกลม (spherical surface) ลำแสงขนานซึ่งเดินทางในอากาศและตกกระทบแท่งแก้วตามรูปที่ 1.15 จะหักเหและมาพบกันที่จุด  $F$  เราเรียกจุด  $F$  นี้ว่าจุดโฟกัส และเรียกระยะห่างที่ใกล้สุดจากจุดนี้ไปยังผิวของแก้วว่าระยะโฟกัส



รูปที่ 1.15 การหักเหของแสงเมื่อผ่านแท่งแก้วผิวโค้งกลม

เลนส์ก็คือแท่งแก้วที่มีผิวโค้งกลมทั้งสองด้าน เส้นทางของลำแสงที่ผ่านเลนส์จึงสามารถคำนวณได้โดยใช้กฎของสเนล แต่ในกรณีของเลนส์บาง (ความหนาของเลนส์น้อยมากเมื่อเทียบกับรัศมีความโค้ง) การคำนวณเส้นทางเดินของแสงทำได้ง่ายขึ้น โดยอาศัยหลักที่ว่า ลำแสงทั้งหมดจากจุดๆ หนึ่ง เมื่อเดินทางผ่านเลนส์แล้วจะไปรวมกันเป็นจุดที่อีกฝั่งเสมอ โดยจุดรวมนี้สามารถคำนวณได้ จากหลักที่ว่า ลำแสงที่พุ่งไปยังศูนย์กลางของเลนส์จะไม่มีหักเห และลำแสงขนานกับ optical axis จะทะลุผ่านเลนส์มาตัดกับ optical axis ที่ระยะโฟกัส จุดตัดที่ได้ของลำแสง

สองลำนี้คือจุดที่ลำแสงทั้งหมดมารวมกันในอีกฝั่งหนึ่งของเลนส์ เราสามารถวาดรูปแสดงเส้นทางเดินของแสงจากจุด  $P$  ผ่านเลนส์ซึ่งมีระยะโฟกัส  $f$  และไปกระทบฉากเป็นจุดที่  $P'$  ได้ดังรูปที่ 1.16



รูปที่ 1.16 เส้นทางเดินของแสงเมื่อผ่านเลนส์

และได้ความสัมพันธ์ของระยะต่างๆ คือ

จากรูปสามเหลี่ยมคล้าย  $P'AB$  กับ  $F'OB$

$$\frac{f}{z'} = \frac{y}{y - y'}$$

จากรูปสามเหลี่ยมคล้าย  $PAB$  กับ  $FOB$

$$\frac{f}{-z} = \frac{-y'}{y - y'}$$

$$\frac{f}{z} = \frac{y'}{y - y'}$$

$$\frac{f}{z} + 1 = \frac{y'}{y - y'} + 1 = \frac{y}{y - y'}$$

นั่นก็คือ

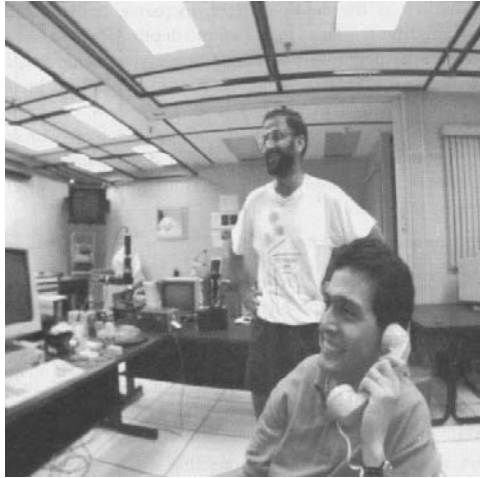
$$\frac{f}{z'} = \frac{f}{z} + 1$$

หรือ

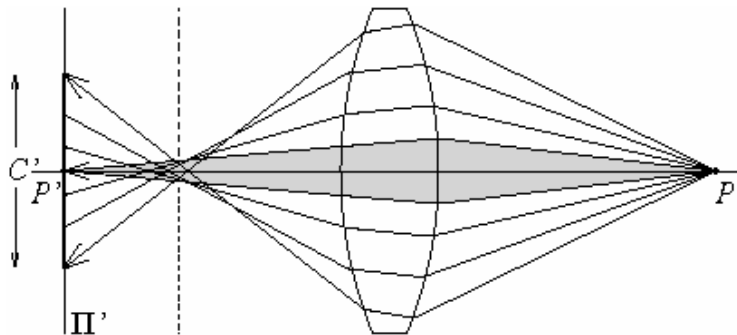
$$\frac{1}{z'} - \frac{1}{z} = \frac{1}{f}$$

ถึงแม้การใช้เลนส์จะช่วยแก้ปัญหาเรื่องปริมาณแสงได้ แต่การใช้เลนส์เป็นอุปกรณ์รับภาพก็ยังมีข้อจำกัดบางประการที่สำคัญได้แก่ การบิดเบือนของภาพในแนวรัศมี (radial distortion) ดังจะเห็นได้จากด้านบนของรูปที่ 1.17 แสดงขอบที่เป็นเส้นตรงบนเพดานแต่กลับกลายเป็นเส้นโค้งในภาพที่ได้จากเลนส์ อย่างไรก็ตามเรามีวิธีในการปรับแก้การบิดเบือนโดยใช้วิธีที่เรียกว่า camera calibration นอกจากการบิดเบือนในแนวรัศมีแล้ว ความสามารถในการรวมแสงที่ต่างกันของส่วนขอบเลนส์และส่วนกลางเลนส์ยังทำให้เกิดภาพที่ไม่ชัดเจน (ดังแสดงในรูปที่ 1.18 จะเห็นว่าแสง

จากจุด P ที่ผ่านขอบเลนส์จะไม่โฟกัสที่จุด P' บนฉากรับแสง) ความไม่ชัดเจนในกรณีนี้เรียกกันว่า spherical aberration ด้วยคุณสมบัติของเลนส์ดังกล่าวภาพที่ได้จากจึงไม่เหมือนภาพจากกล้องรูเข็ม



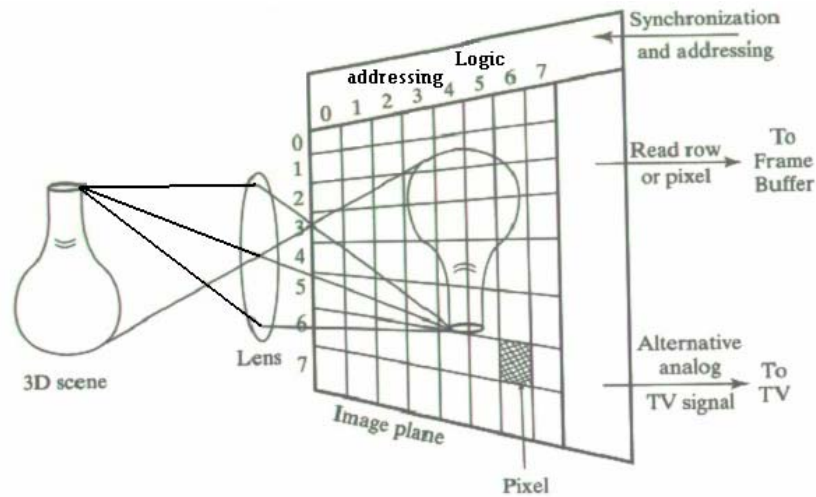
รูปที่ 1.17 การบิดเบือนของภาพตามแนวรัศมี



รูปที่ 1.18 Spherical aberration

#### 1.2.4 กล้อง CCD

กล้องถ่ายภาพรูปบันทึกภาพลงบนฟิล์ม โดยอาศัยปฏิกิริยาเคมี ที่ทำให้เกิดการเปลี่ยนแปลงบนส่วนของฟิล์มที่ได้รับแสง แต่ภาพที่บันทึกบนแผ่นฟิล์มนั้น ไม่สามารถที่จะนำไปประมวลผลด้วยคอมพิวเตอร์โดยตรงได้ กล้อง CCD (charged coupled device) ประกอบด้วยส่วนรับแสง และส่วนบันทึกภาพ ซึ่งเป็นวงจรรวมที่ประกอบไปด้วยเซลล์เล็กๆ เรียงอยู่อย่างมีระเบียบ (รูปที่ 1.19) โดยแต่ละเซลล์จะมีความไวแสงและเปลี่ยนแสงที่มากกระทบให้อยู่ในรูปของสัญญาณไฟฟ้าพร้อมที่จะส่งไปยังคอมพิวเตอร์เพื่อประมวลผล



รูปที่ 1.19 กล้อง CCD

ด้วยการพัฒนางจรรวมที่เป็นไปอย่างรวดเร็ว วงจรรวม CCD ขนาดเล็ก สามารถบันทึกรายละเอียดได้สูงขึ้นไปอย่างมาก (รูปที่ 1.20) ดังที่จะเห็นได้ว่ามีกล้องดิจิทัลในท้องตลาด ที่เรียกว่าเมกกาพิกซ์เซล (megapixel) สามารถบันทึกภาพด้วยเซลล์ไวแสงจำนวนนับล้านเซลล์ นอกจากรายละเอียดที่สูงขึ้น วงจรรวม CCD ในปัจจุบันยังมีความไวแสงที่สูงขึ้นมากอีกด้วย นี่เป็นปัจจัยหนึ่งที่ทำให้กล้อง CCD ในปัจจุบันไม่ต้องอาศัยเลนส์รวมแสงที่มีขนาดใหญ่หรือสามารถทำงานได้ดีแม้ด้วยกล้องแบบรูเข็มเท่านั้น



รูปที่ 1.20 กล้อง CCD ขนาดจิ๋ว โดย Fujitsu ที่มีความละเอียด 352 x 288

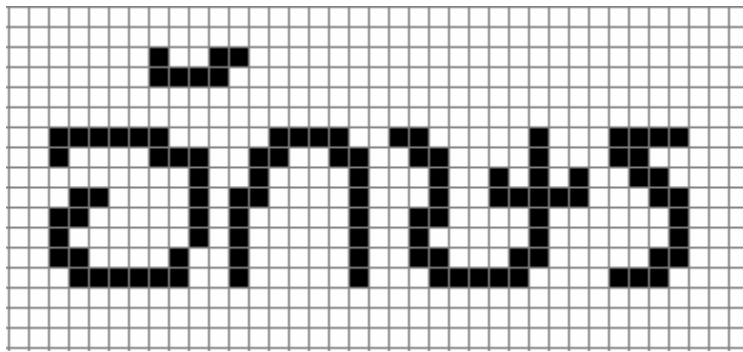
### 1.3 การวิเคราะห์และประมวลผลภาพ

การวิเคราะห์ประมวลผลภาพและการรับรู้จากภาพ เป็นศาสตร์ที่ครอบคลุมเนื้อหากว้างขวางและลึกซึ้ง เกินกว่าที่จะกล่าวถึงได้ครบถ้วนในบทนี้ เพื่อให้ผู้เรียนได้เกิดความเข้าใจถึงลักษณะของปัญหาที่สามารถประยุกต์ใช้เทคนิคต่างๆ จากศาสตร์แขนงนี้ จะขอยกตัวอย่างการประมวลผลภาพแบบพื้นฐาน โดยจะขอกล่าวถึงการประมวลผล

ภาพไบนารี (binary image) ซึ่งเข้าใจได้ง่ายก่อน แล้วจึงจะกล่าวถึงการประมวลผลภาพเกรย์สเกล (gray scale image) ที่มีความซับซ้อนขึ้นเป็นลำดับต่อไป

### 1.3.1 ภาพไบนารี

ภาพไบนารีก็คือภาพซึ่งในแต่ละพิกเซลจะมีความเข้มของแสงได้สองแบบเท่านั้น คือสว่างกับมืด โดยอาจเขียนแทนได้ด้วยเลข 0 และ 1 ด้วยความเข้มของแสงเพียงสองระดับ ภาพไบนารีจึงมีข้อจำกัดที่จะนำมาใช้แสดงภาพต่างๆ ไปได้แต่อย่างไรก็ตามการมีความเข้มของแสงเพียงสองระดับก็ทำให้การประมวลผลทำได้ง่ายมีประสิทธิภาพ การประมวลผลภาพไบนารีนำไปใช้มากในการประมวลผลเอกสาร การประมวลผลภาพในอุตสาหกรรมที่ต้องการความเร็วสูง เช่นการนับจำนวนชิ้นส่วนที่อยู่บนสายพานที่กำลังเคลื่อนที่เป็นต้น การมีความเข้มเพียงสองระดับในภาพไบนารี ทำให้สามารถที่จะเลือกพิจารณาให้ความเข้มระดับหนึ่งแทนภาพของสิ่งที่เราสนใจโดยเราจะขอเรียกพิกเซลที่มีความเข้มระดับนี้ว่าพิกเซลภาพ และความเข้มอีกระดับแทนพื้นหลังโดยพิกเซลที่มีความเข้มระดับนี้จะถูกเรียกว่าพิกเซลพื้นหลัง ในทางปฏิบัติในการพิจารณาภาพไบนารี เราจะไม่สนใจแต่ละพิกเซลแยกกันไป แต่เราจะสนใจกลุ่มของพิกเซลที่อยู่ติดกัน หรือที่เรียกว่าพิกเซลเพื่อนบ้าน เช่นกลุ่มของพิกเซลที่เรียงกันเป็นตัวอักษร (รูปที่ 1.21) สำหรับพิกเซลที่อยู่ในรูปแบบตารางสี่เหลี่ยมนั้น การตีความว่าพิกเซลใดอยู่ติดกันทำได้สองแบบหลักๆ คือ



รูปที่ 1.21 ภาพไบนารีแสดงกลุ่มของพิกเซลภาพที่ประกอบเป็นตัวอักษร

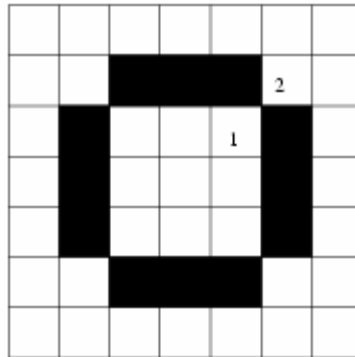
- (1) แบบสี่เพื่อนบ้าน (4-connectedness): จะถือว่าแต่ละพิกเซลอยู่ติดกับสี่พิกเซลที่อยู่ด้านบน, ล่าง, ซ้าย และขวา เช่นในรูปที่ 1.22 เราจะถือว่าพิกเซล  $p$  อยู่ติดกับพิกเซล 0, 2, 4 และ 6 แต่ไม่ติดกับพิกเซล 1, 3, 5 และ 7 สังเกตได้ว่าในบางครั้งเราอาจใช้ชื่อทิศในการเรียกพิกเซลรอบๆ เช่นเรียกพิกเซล 2 ว่าพิกเซลเหนือของพิกเซล  $p$  เป็นต้น

3	2	1
4	$p$	0
5	6	7

รูปที่ 1.22 พิกเซล  $p$  และพิกเซลรอบๆ

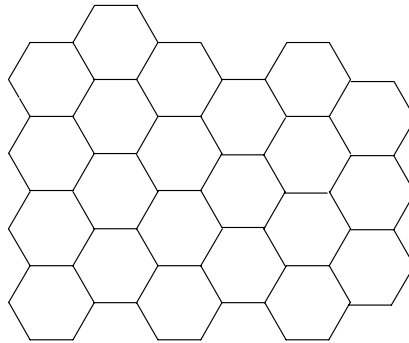
- (2) แบบแปดเพื่อนบ้าน (8-connectedness): จะถือว่าแต่ละพิกเซลอยู่ติดกับทั้งแปดพิกเซลที่อยู่ล้อมรอบ เช่น ในรูปที่ 1.22 พิกเซล  $p$  อยู่ติดกับพิกเซล 0,1,2,3,4,5,6 และ 7

นิยามของการอยู่ติดกันของพิกเซลทั้งสองแบบที่กล่าวมาเป็นที่นิยมใช้ในการประมวลผลภาพไบนารี แต่มีสิ่งที่จะต้องระวังคือมีความไม่สอดคล้องบางประการกับเรขาคณิตที่เราคุ้นเคย กล่าวคือจากทฤษฎีเส้นโค้งจอร์แดน (Jordan's curve theorem) เส้นโค้งบนระนาบที่ลากเป็นวงปิดจะแบ่งระนาบออกเป็นสองส่วนที่ไม่ติดกัน คือส่วนที่อยู่ภายในวงและส่วนที่อยู่นอกวง แต่สำหรับภาพไบนารีหากเราลองกำหนดให้ใช้นิยามการติดกันแบบแปดเพื่อนบ้านแล้วลองลากเส้นให้เป็นวงเช่นในรูปที่ 1.23 จะเห็นได้ว่าพิกเซลที่ประกอบเป็นเส้นที่ลากนี้ไม่ได้แบ่งพิกเซลที่เหลือให้เป็นสองส่วนที่ไม่ติดกันเลย (สังเกตพิกเซล 1 กับ 2 ของพื้นที่หลังที่ถือว่าติดกัน) ในลักษณะเดียวกันหากเราใช้การติดกันแบบสี่เพื่อนบ้านกับภาพไบนารีในรูปที่ 1.23 ภาพนี้ก็จะประกอบไปด้วยเส้นสี่เส้นที่ไม่ติดกันและไม่มีการปิดเกิดขึ้น แต่ว่ากลับทำให้พิกเซลที่เหลือ (พิกเซลพื้นหลัง) แยกกันเป็นสองกลุ่มที่ไม่ติดกัน



รูปที่ 1.23 ภาพไบนารีของเส้นที่ลากเป็นวง

หากความไม่สอดคล้องนี้ไม่ทำให้เกิดปัญหากับงานที่ทำก็ไม่เป็นไร แต่หากจำเป็นต้องแก้ไขให้สอดคล้อง วิธีแก้ปัญหาวินัยหนึ่งก็คือการเปลี่ยนไปใช้พิกเซลที่เป็นรูปหกเหลี่ยมแทนพิกเซลสี่เหลี่ยม (รูปที่ 1.24) และกำหนดให้พิกเซลที่มีขอบร่วมกันเป็นพิกเซลที่ติดกัน อีกวิธีที่ทำได้โดยยังคงรูปแบบพิกเซลสี่เหลี่ยมก็คือการพิจารณาการติดกันของพิกเซลพื้นหลังแบบสี่เพื่อนบ้าน แต่ใช้การติดกันแบบแปดเพื่อนบ้านสำหรับพิกเซลที่แทนภาพของสิ่งที่เราสนใจ



รูปที่ 1.24 พิกเซลแบบหกเหลี่ยม

### 1.3.1.1 การนับจำนวนกลุ่มของพิกเซลที่อยู่ติดกัน

ดังที่ได้กล่าวไปแล้วว่า โดยทั่วไปเราไม่สนใจพิกเซลเดี่ยวๆ แยกกันไป แต่เราจะสนใจกลุ่มของพิกเซลที่อยู่ติดกัน กลุ่มของพิกเซลเหล่านี้อาจเป็นภาพของตัวอักษรในเอกสาร หรืออาจเป็นภาพของสิ่งที่เราสนใจอื่นๆ ก็ได้ ข้อมูลที่เราต้องการทราบอยู่บ่อยๆ อย่างหนึ่งก็คือ จำนวนกลุ่มของพิกเซลที่อยู่ติดกัน เช่นจำนวนตัวอักษรในภาพเอกสาร หรือจำนวนชิ้นส่วนที่อยู่บนสายพาน การที่เราจะทราบจำนวนดังกล่าวได้ เราต้องสามารถแยกพิกเซลออกเป็นกลุ่มๆ ที่ติดกัน เรานิยามกลุ่มของพิกเซลที่อยู่ติดกันในลักษณะคล้ายๆ กับนิยามของ connected component จากทฤษฎีกราฟ นั่นก็คือทุกๆ คู่ของพิกเซลในกลุ่มเดียวกันจะต้องมีเส้นทางเชื่อมที่ประกอบไปด้วยพิกเซลอื่นๆ ในกลุ่มที่เรียงต่อกัน โดยต้องไม่มีเส้นทางระหว่างคู่พิกเซลต่างกลุ่ม วิธีนับที่จะเสนอนี้ ทำงานโดยแจกหมายเลขกลุ่มให้กับแต่ละพิกเซล พิกเซลที่อยู่คนละกลุ่มจะได้หมายเลขต่างกัน ในอัลกอริทึมนี้ ภาพไบนารีที่จะนำมาประมวลผล ถูกเก็บในแถวลำดับสองมิติ  $A[1..n][1..m]$  ที่มีขนาด  $n \times m$  โดยพิกเซลภาพจะมีค่าเป็น -1 และพิกเซลพื้นหลังมีค่าเป็น 0 การทำงานของอัลกอริทึมจะเป็นดังนี้

imageLabel(A)

Input: ภาพไบนารีในแถวลำดับสองมิติ  $A[1..n][1..m]$  ที่พิกเซลภาพมีค่า -1 และพิกเซลพื้นหลังเป็น 0

Output: จำนวนกลุ่มพิกเซล และเลขที่กลุ่มของแต่ละพิกเซลในแถวลำดับ A

```

1:     k = 1
2:     for (i=1; i<=n; i++)
3:         for (j=1; j<=m; j++) {
4:             ถ้า  $A[i][j] == -1$  จึง assignLabel(A, i, j, k)
5:             k = k+1
6:         }
7:     return k

```

โดย assignLabel ที่ทำหน้าที่แจกหมายเลขกลุ่ม k เป็นดังนี้

assignLabel(A,i,j,k)

Input: ภาพไบนารีในแนวลำดับสองมิติ A ธรรมชาติ i, j ซึ่พิกเซลเริ่มต้น และหมายเลขกลุ่มที่จะแจก

Output: กลุ่มของพิกเซลที่มีพิกเซล A[i][j] จะได้หมายเลข k

- 1: สร้าง queue เปล่า Q
- 2: เพิ่มคู่อันดับ (i,j) ใน Q
- 3: while (Q ยังไม่ว่าง) {
- 4:     ดึงคู่อันดับแรกออกจาก Q สมมุติว่ามันคือ (a,b)
- 5:     แจกหมายเลขกลุ่ม k ให้พิกเซลภาพ A[a][b] โดยทำ A[a][b] = k
- 6:     สำหรับทุกๆ พิกเซลภาพ A[c][d] == -1 ที่อยู่ติดกับพิกเซล A[a][b]  
      ให้เพิ่มคู่อันดับ (c,d) ใน Q }

### 1.3.1.2 ตัวบอกลักษณะเชิงพื้นที่

เราสามารถบรรยายรูปร่างของกลุ่มพิกเซลที่อยู่ติดกันด้วยตัวบอกลักษณะเชิงพื้นที่ ตัวบอกลักษณะเชิงพื้นที่ของกลุ่มพิกเซล  $R$  ตัวแรกที่จะกล่าวถึงก็คือโมเมนต์ (moment) ซึ่งสามารถคำนวณได้ดังนี้

$$m_{k,j} = \sum_{(x,y) \in R} x^k y^j$$

โดยที่  $(x, y)$  คือพิกัดของแต่ละพิกเซลในกลุ่ม  $R$  และค่า  $k+j$  ก็คือค่าอันดับของโมเมนต์นั่นเอง โมเมนต์ที่อันดับต่ำๆ จะให้ค่าที่มีความหมายเช่น

$$A = m_{0,0} = \sum_{(x,y) \in R} x^0 y^0$$

ให้ค่าจำนวนพิกเซลที่อยู่ในกลุ่ม ซึ่งถ้ากำหนดให้แต่ละพิกเซลมีพื้นที่หนึ่งหน่วย ค่า  $A$  ที่ได้ก็คือพื้นที่ของกลุ่มพิกเซล  $R$  นอกจากใช้หาพื้นที่แล้ว เรายังสามารถหาจุดศูนย์กลางมวลของกลุ่มพิกเซล  $R$  ได้จากการคำนวณโมเมนต์อันดับ 1 นั่นคือเราจะได้พิกัดของศูนย์กลางมวลเป็น  $(x_c, y_c) = (m_{1,0}, m_{0,1})$  นอกจากโมเมนต์ที่กล่าวถึงยังมีค่าโมเมนต์รอบศูนย์กลาง (central moment) ซึ่งให้ค่าที่ไม่ขึ้นกับตำแหน่งของกลุ่มพิกเซล (ขึ้นกับรูปร่างเท่านั้น) โมเมนต์รอบศูนย์กลางคำนวณได้ดังนี้

$$\mu_{k,j} = \sum_{(x,y) \in R} (x - x_c)^k (y - y_c)^j$$

จริงๆ แล้วโมเมนต์รอบศูนย์กลางก็คือโมเมนต์ธรรมดาที่กล่าวมาแล้วเมื่อเราเลื่อนกลุ่มพิกเซล  $R$  ให้ศูนย์กลางมวลของมันมาอยู่ที่จุดกำเนิด ด้วยค่าโมเมนต์รอบศูนย์กลางนี้เราสามารถคำนวณแกนกลางของกลุ่มพิกเซล  $R$  ได้ โดย



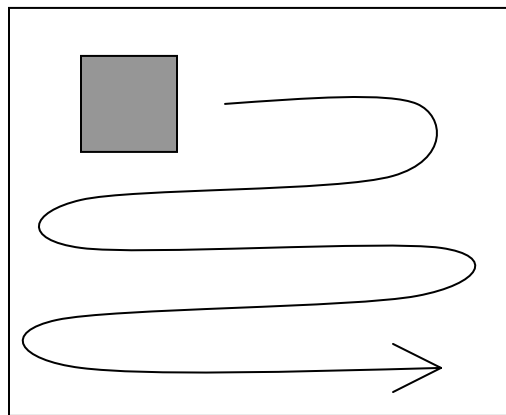
แกนกลางของกลุ่มพิกเซลนี้คือแกนที่ให้ค่าโมเมนต์อินเนอร์เซีย (moment of inertia) ต่ำสุด แกนกลางนี้จะวางตัวตามทางยาวของกลุ่มพิกเซลและผ่านจุดศูนย์กลางมวลของกลุ่มพิกเซล โดยทำมุมกับแกน x เท่ากับ

$$\frac{1}{2} \text{atan2}(\mu_{1,1}, \mu_{2,0} - \mu_{0,2})$$

เราสามารถใช้อัตลักษณ์เชิงพื้นที่นี้ช่วยในการตรวจสอบอย่างคร่าวๆ ว่าภาพที่ได้รับนั้นมีโอกาสเป็นภาพของวัตถุต้นแบบหรือไม่ โดยการเปรียบเทียบตัวบ่งชี้เชิงพื้นที่ของภาพที่ได้กับภาพของวัตถุต้นแบบ ทั้งนี้ภาพที่ต้องการตรวจสอบนั้นต้องถูกถ่ายในสภาวะเดียวกันกับภาพของวัตถุต้นแบบ นอกจากนี้วิธีนี้เราก็อาจใช้ภาพต้นแบบในการเปรียบเทียบโดยตรงกับภาพที่ต้องการตรวจสอบโดยไม่ต้องคำนวณตัวบ่งชี้เชิงพื้นที่ก่อน เราเรียกวิธีนี้ว่าการจับคู่กับแม่แบบ (template matching)

### 1.3.1.3 การจับคู่กับแม่แบบ

การใช้แม่แบบเป็นส่วนประกอบสำคัญของหลายๆ เทคนิคในการประมวลผลภาพ แม่แบบก็คือภาพ ซึ่งโดยทั่วไปจะนำไปวางทับกับภาพที่พิจารณาและทำการประมวลผลโดยใช้ค่าของพิกเซลของภาพและของแม่แบบที่อยู่ในตำแหน่งเดียวกัน ในที่นี้เราจะเสนอการจับคู่กับแม่แบบซึ่งก็คือการตรวจสอบว่ามีส่วนของภาพที่ตำแหน่งใดที่เหมือนกับแม่แบบ โดยหลักการแล้วการจับคู่กับแม่แบบก็เหมือนกับการค้นแบบอนุกรม นั่นก็คือเราจะไล่เปรียบเทียบไปเรื่อยๆ ตามลำดับจนหมด แต่เนื่องจากภาพของเราเป็นสองมิติการไล่เปรียบเทียบก็จะทำโดยเลื่อนแม่แบบจากซ้ายไปขวาและจากบนลงล่างทีละพิกเซลจนครบทั้งภาพ (รูปที่ 1.25) ส่วนการตรวจสอบว่าส่วนของภาพนั้นมีความเหมือนกับแม่แบบเท่าไรนั้น ในกรณีภาพไบนารีเราอาจใช้วิธีนับว่ามีพิกเซลที่มีค่าตรงกับแม่แบบกี่พิกเซล ตำแหน่งของการเลื่อนแม่แบบที่ได้จำนวนพิกเซลที่ตรงกับแม่แบบมากที่สุดก็คือตำแหน่งที่มีภาพเหมือนแม่แบบปรากฏอยู่ ในอัลกอริทึมต่อไปนี้จะตรวจสอบว่าแม่แบบ P มีความเหมือนกับภาพ A ที่ตำแหน่งใดมากที่สุด บรรทัดที่ 5 ในรหัสเทียมเป็นการนับจำนวนพิกเซลที่มีค่าตรงกับแม่แบบ



รูปที่ 1.25 การจับคู่กับแม่แบบ

templateMatching(A,P)

Input: ภาพไบนารีในแถวลำดับสองมิติ  $A[1\dots n][1\dots m]$  ที่พิกเซลภาพมีค่า 1 และพิกเซลพื้นหลังเป็น 0  
ภาพแม่แบบไบนารีในแถวลำดับสองมิติ  $P[1\dots r][1\dots q]$  โดย  $r < n$  และ  $q < m$

Output: ตำแหน่ง  $x$  และ  $y$  ในภาพ  $A$  ที่เหมือนกับแม่แบบมากที่สุด

```

1:   maxPix = 0;
2:   for (i=1; i<=n-r; i++)
3:       for (j=1; j<=m-q; j++) {
4:           cnt = 0;
5:           for (k=1; k<=r; k++) for (l=1; l<=q; l++) cnt += A[i+k][j+l] * P[k][l];
6:           if (cnt > maxPix) {maxPix = cnt; x = i; y=j; }
6:       }
7:   return x and y

```

### 1.3.2 ภาพเกรย์สเกล

ถึงแม้ว่าภาพไบนารีจะสามารถนำมาประมวลผลและจัดเก็บได้สะดวก ภาพที่รับมาจากกล้องรับภาพนั้นโดยทั่วไปจะไม่ได้อยู่ในแบบไบนารี แต่อยู่ในแบบเกรย์สเกลซึ่งบอกรูปแบบของความเข้มของแสงในแต่ละพิกเซล โดยความเข้มของแสงนี้มีได้มากกว่าสองระดับ ถึงแม้จะมีความซับซ้อนกว่าภาพไบนารี ภาพเกรย์สเกลก็สามารถให้ข้อมูลได้กว้างขวางลึกซึ้งยิ่งกว่า ในส่วนนี้ขอเสนอวิธีประมวลภาพเกรย์สเกลเบื้องต้น

#### 1.3.2.1 Thresholding

ในบางครั้งภาพเกรย์สเกลที่พิจารณามีความเด่นชัดของวัตถุในภาพที่แยกตัวออกจากพื้นหลังอย่างชัดเจน ในกรณีนี้เราอาจแยกภาพของวัตถุออกจากพื้นหลังโดยการเปลี่ยนภาพเกรย์สเกลให้เป็นภาพไบนารี แต่ภาพเกรย์สเกลมีระดับความเข้มของแสงได้มากกว่าสองระดับดังนั้นเราจึงต้องมีวิธีเลือกพิกเซลที่ระดับความเข้มใดจะถูกกำหนดให้เป็นพิกเซลพื้นหลัง และพิกเซลที่ระดับความเข้มใดจะถูกกำหนดให้เป็นพิกเซลภาพ วิธีที่นิยมใช้วิธีหนึ่งก็คือการกำหนดค่าความเข้มของแสงค่าหนึ่งที่ใช้แยกแยะ เช่นหากความเข้มของแสงของพิกเซลที่พิจารณามีค่าน้อยกว่าค่านี้นั้นพิกเซลนั้นก็เป็นพิกเซลพื้นหลัง มิฉะนั้นพิกเซลนั้นก็คือเป็นพิกเซลภาพ ค่าสำหรับแยกประเภทของพิกเซลที่ตัวนี้เราเรียกว่าค่าเทรชโฮลด์ (threshold) เราสามารถหาค่าเทรชโฮลด์นี้ได้ด้วยการพิจารณาฮิสโตแกรม (histogram) ของความเข้มของแสง ฮิสโตแกรมนี้ก็คือกราฟที่แจกแจงจำนวนพิกเซลที่ความเข้มของแสงต่างๆ กัน ตัวอย่างในรูปที่ 1.26 แสดงภาพและฮิสโตแกรมของมัน จะเห็นได้ว่าเมื่อพื้นหลังและภาพวัตถุสามารถแยกกันได้อย่างชัดเจน ฮิสโตแกรมจะเป็นกราฟเหมือนภูเขาสองลูก เราสามารถเลือกค่าความเข้มของแสงที่แบ่งภูเขาสองลูกเพื่อใช้เป็นค่าเทรชโฮลด์

## รูปที่ 1.26 ภาพและฮิสโตแกรม

## 1.3.2.2 การมัวภาพ (image blurring)

เมื่อพูดถึงภาพมัว หลายคนคงนึกถึงภาพที่ขาดคุณภาพ แต่จริงๆ แล้วการมัวภาพมีประโยชน์ที่สำคัญอย่างหนึ่งก็คือ การลดผลกระทบของสัญญาณรบกวนในภาพ ซึ่งทำให้สามารถนำภาพที่ได้ไปประมวลผลต่อไปอย่างมีประสิทธิภาพ วิธีง่ายที่สุดวิธีหนึ่งของการมัวภาพก็คือการแทนค่าความเข้มของพิกเซลนั้นด้วยค่าเฉลี่ยของมันกับค่าความเข้มของพิกเซลที่อยู่รอบๆ เราสามารถมัวภาพด้วยการหาค่าเฉลี่ยนี้โดยการใช้แม่แบบ ซึ่งในกรณีนี้เรานิยมเรียกแม่แบบว่า หน้ากาก (mask) โดยวิธีทำก็คือเราจะเลื่อนหน้ากากไปเรื่อยๆ บนภาพที่จะมัวจากซ้ายไปขวาและจากบนลงล่างที่ละพิกเซล (เหมือนการจับคู่กับแม่แบบ) โดยทุกครั้งที่เลื่อนก็จะคำนวณผลรวมหน้ากาก ผลรวมที่ได้จะถูกนำไปเขียนลงที่พิกเซลภาพผลลัพธ์ (แยกเป็นอีกภาพต่างหาก) ที่พิกัดตรงกับพิกัดของภาพตั้งต้นตรงจุดกึ่งกลางหน้ากากในขณะนั้น สมมุติว่าภาพที่ทำการมัวคือ  $A[1..m][1..n]$  และหน้ากากมีขนาด  $3 \times 3$  คือ  $P[-1, \dots, 1][-1, \dots, 1]$  ผลรวมหน้ากากขณะจุดกึ่งกลางหน้ากากเลื่อนมาอยู่ที่  $A[i][j]$  ก็คือ

$$\begin{array}{lll} A[i-1][j-1] * P[-1][-1] + & A[i-1][j] * P[-1][0] + & A[i-1][j+1] * P[-1][1] + \\ A[i][j-1] * P[0][-1] + & A[i][j] * P[0][0] + & A[i][j+1] * P[0][1] + \\ A[i+1][j-1] * P[1][-1] + & A[i+1][j] * P[1][0] + & A[i+1][j+1] * P[1][1] \end{array}$$

จะเห็นได้ว่าหาก  $P[i][j]$  ใดๆ ของหน้ากากมีค่าเท่ากับ  $1/9$  ผลลัพธ์ที่ได้ก็คือค่าเฉลี่ยของพิกเซลกึ่งกลางกับพิกเซลรอบๆ รหัสเทียมต่อไปนี้จะบรรยายวิธีการใช้หน้ากากในการมัวภาพ

imageMasking(A,P,B)

Input: ภาพเกรย์สเกลในแถวลำดับสองมิติ  $A[1..m][1..n]$

หน้ากากในแถวลำดับสองมิติขนาด  $(k+1) \times (k+1)$   $P[-k...k][-k...k]$

Output: ภาพเกรย์สเกลผลลัพธ์ในแถวลำดับสองมิติ  $B[1..m][1..n]$

```
1:   for (i=1; i<=m; i++)
2:       for (j=1; j<=n; j++)
3:           B[i][j] = sumMask(A,P,i,j)
4:   return B
```

sumMask(A,P,i,j)

```
1:   sum = 0;
2:   for (q=-k; q<=k; q++)
3:       for (r=-k; r<=k; r++)
4:           sum = sum + A[i+q][j+r] * P[q][r]
5:   return sum
```