การกำจัดสัญญาณรบกวนโดยใช้ตัวกรองสัญญาณเอ็นโทรปี

A Noise-Cleaning using a Entropy Filter

สาธิต อินทจักร์ และ กิตติ ไพฑูรย์วัฒนกิจ
คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ถนนฉลองกรุง เขตลาดกระบัง กรุงเทพฯ 10250
โทร. 326-7346-7 โทรสาร 326-7346-7
E-Mail: kisathit@kmitl.ac.th

บทคัดย่อ

ในบทความนี้จะนำเสนอ วิธีการกำจัดสัญญาณรบกวนที่ไม่เป็นเชิงเส้น โดยใช้ข้อมูลของจุดภาพในย่านใกล้เคียงมาวัดโอกาสการมีสัญญาณรบกวนด้วยค่าเอ็นโทรปี และเลือกประมวลผลเฉพาะบริเวณที่มีสัญญาณรบกวน จุดเด่นของวิธีนี้คือจุดภาพทั้งหมดจะไม่ถูกประมวลผล ซึ่งต่างจากตัวกรองสัญญาณที่ใช้กันอยู่ทั่วไป ที่ทุกจุดภาพจะต้องถูกกรองสัญญาณ ไม่ว่าจะมีหรือไม่มีสัญญาณรบกวนก็ตาม ทำให้ภาพที่ผ่านการกรองมีความคลาดเคลื่อนจากข้อมูลภาพเดิมสูง วิธีที่จะนำเสนอนี้จะสามารถกำจัดสัญญาณรบกวนได้ดี และยังคงความคมชัดของภาพไว้ใกล้เคียงข้อมูลภาพเดิมมากที่สุด สำหรับประสิทธิภาพของอัลกอริทึมที่นำเสนอ จะถูกนำมาประเมินผลทั้งการพิจารณาด้วยตาเปล่า และการเปรียบเทียบเชิงปริมาณ ซึ่งจากการทดลองที่ได้ประมวลผลกับสัญญาณรบกวนแกลเซียนนั้น ปรากฏว่าวิธีที่นำเสนอให้ผลลัพธ์ที่ดีกว่าการใช้ตัวกรองสัญญาณค่าเฉลี่ย 1.13 เท่า และค่ามัธยฐาน 1.06 เท่า โดยเทียบจากค่าอัตราส่วนสัญญาณรบกวนสูงสุด

Abstract

In this paper, a nonlinear removed noise algorithm based on the entropy concept is proposed. The neighborhood pixels are measured with the entropy value, if the value is high; the neighboring will be processed to remove noise. The dominant of our method, not every pixel in the images has to be processed that different from traditional methods, each pixel is filtered. Generally every pixel is replaced with a new value depending on the local context; neither the neighborhood has a noise, nor doesn't' have. So that the general filtering results are lack of a sharpening edge and more deviate from the original. Our method can remove impulse noise and preserving edges. The performance of the proposed algorithm is evaluated through subjective and objective criteria. From the experimental result, our method performs better than 1.13 of the traditional mean and 1.06 of the median when compare with peak signal to noise ratio for different types of noise.

1. คำนำ

การปรับปรุงภาพเป็นขั้นตอนสำคัญขั้นตอนหนึ่งในการประมวลผลภาพ อัลกอริทึมที่ใช้ในการปรับปรุงภาพแต่ละแบบจะถูกนำไปใช้ในวัตถุประสงค์ที่แตกต่างกัน ตามแต่จะนำไปใช้ในการปรับปรุงเพื่ออะไร เช่น ถ้าภาพมืดไปก็ใช้อัลกอริทึมการปรับภาพให้คมชัด หรือถ้าภาพมีสัญญาณยานรบกวนก็ใช้อัลกอริทึมในการกำจัดสัญญาณรบกวน เป็นต้น ปัญหาของการปรับปรุงภาพที่จะนำเสนอในที่นี้คือ การกรองสัญญาณรบกวนอิมพัลส์ (Impulse noise) ที่สามารถคงความคมชัดของขอบวัตถุในภาพเอาไว้ได้มากที่สุด แต่การกำจัดสัญญาณรบกวนกับการเน้นขอบภาพ (Edge Enchantment) ผลลัพธ์ที่ได้จากอัลกอริทึมของการปรับปรุงภาพในกรณีนี้มักจะขัดแย้งกัน เนื่องจากการกำจัดสัญญาณรบกวนมักจะทำให้ขอบของวัตถุในภาพหายไป และตัวกรองสัญญาณรบกวนโดยทั่วไป จะมองขอบของวัตถุที่มีความคมชัดมากๆ ว่าเป็นสัญญาณรบกวน

ถ้าเราพิจารณาผลลัพธ์ที่ได้จากอัลกอริทึมที่ใช้กรองสัญญาณรบกวนในปัจจุบัน เช่น ตัวกรองสัญญาณค่าเฉลี่ย (Mean filter) และตัวกรองสัญญาณมัธยฐาน (Median filter) [1] จะเห็นว่าความคมชัดของภาพจะหายไป ที่เป็นเช่นนี้ก็เพราะว่า อัลกอริทึมเหล่านี้จะกรองข้อมูลภาพทุกจุด ไม่ว่าบริเวณนั้นมีสัญญาณรบกวนหรือไม่ก็ตาม ยิ่งถ้าเป็นตัวกรองค่าเฉลี่ยด้วยแล้ว ผลลัพธ์ที่ได้ จะยิ่งมีคลาดเคลื่อนมากขึ้นตามขนาดของวินโดว์ที่ใช้ในการประมวลผล ส่วนตัวกรองมัธยฐานก็เช่นกัน แต่คุณภาพของผลลัพธ์ที่ได้จะดีกว่าตัวกรองแบบค่าเฉลี่ย เนื่องจากตัวกรองมัธยฐานเป็นตัวกรองสัญญาณแบบไม่เป็นเชิงเส้น [2]

ตัวกรองสัญญาณมัธยฐาน เป็นตัวกรองสัญญาณที่เกิดจากการประมาณค่าทางสถิติ ซึ่งสามารถกำจัดสัญญาณรบกวนชนิดอิมพัลส์ได้ดี แต่รายละเอียดภายในภาพจะหายไปบ้างพอสมควร ดังนั้นจึงมีผู้พัฒนาวิธีนี้เพื่อต้องการให้ภาพมีความคมชัดมากยิ่งขึ้น โดยการหาค่ามัธยฐานของตัวกรองสัญญาณแบบหลายระดับ (Multilevel Median Filter: MLMF [3]) แต่ตัวกรองสัญญาณนี้จะไม่สามารถกำจัดสัญญาณรบกวนที่ต่อกันเป็นเส้นสั้นๆ หรือที่เรียกว่า "Short Line" ได้ ดังนั้นจึงมีผู้ปรับปรุงต่ออีก คือ เป็นการกรองสัญญาณมัธยฐานแบบหลายขั้นตอน (Adaptive Multistage Median Filter: AMMF [4]) ซึ่งวิธีนี้ก็ยังคงรายละเอียดของภาพไว้ได้ดี แต่ยังคงมีปัญหา Short Line อยู่เช่นเดิม

RECEIVED 16 MAY, 1999
ACCEPTED 27 JULY, 2001

สำหรับแนวคิดที่ใช้สร้างตัวกรองสัญญาณชนิดนี้ขึ้นมาคือ ต้องการจะประมวลผลเฉพาะบริเวณที่มีสัญญาณรบกวนเท่านั้น แต่การที่จะพารามิเตอร์มาใช้วัดว่าบริเวณใดมีสัญญาณรบกวนหรือไม่คือ ใช้แนวคิดของค่าเอ็นโทรปี (Entropy) ซึ่งอาศัยค่าความน่าจะเป็นในแต่ละบริเวณเป็นตัวบ่งชี้ เมื่อข้อมูลภาพไม่ถูกประมวลผลทุกจุด ทำให้ผลลัพธ์ที่ได้มีความคมชัด ซึ่งบางครั้งเราไม่สามารถจำแนกถึงความแตกต่างด้วยการพิจารณาด้วยตาเปล่า ดังนั้นเราจึงใช้การวัดเชิงตัวเลขมายืนยัน เพื่อบอกถึงความมีประสิทธิภาพของอัลกอริทึม ซึ่งในบทความนี้ใช้วิธีการวัดเชิงตัวเลขสามวิธีด้วยกัน [5] คือ

1.1 ค่าความคลาดเคลื่อนสัมบูรณ์โดยเฉลี่ย (Mean Absolute Error: MAE) ซึ่งมีสมการดังนี้

$$MAE = \sum_{x=1}^{M}\sum_{y=1}^{N}\left|f(x,y) - g(x,y)\right|$$

เมื่อ $f(x,y)$ เป็นจุดภาพที่ตำแหน่ง (x, y) ที่ผ่านการกรองสัญญาณรบกวนแล้ว ส่วน $g(x,y)$ เป็นจุดภาพที่ตำแหน่งเดียวกันกับ $f(x,y)$ แต่เป็นภาพต้นแบบที่ไม่มีสัญญาณรบกวน ซึ่งภาพที่นำมาใช้มีขนาด M×N

1.2 ค่าความคลาดเคลื่อนกำลังสองโดยเฉลี่ย (Mean Square Error: MSE) ซึ่งสามารถคำนวณได้ดังนี้

$$MSE = \sum_{x=1}^{M}\sum_{y=1}^{N}\left(f(x,y) - g(x,y)\right)^2$$

1.3 อัตราส่วนของสัญญาณรบกวนสูงสุด (Peak Signal to Noise Ratio: PSNR)

$$PSNR = 10\log_{10}\left\{\frac{\sum_{x=1}^{M}\sum_{y=1}^{N}\left(f(x,y) - g(x,y)\right)^2}{\left(\max\{f(x,y)\}\right)^2}\right\} dB.$$

ทั้งข้อต่างๆ ที่จะนำเสนอต่อจากนี้ไปคือ แนวคิดในการสร้างตัวกรองสัญญาณด้วยค่าความน่าจะเป็น ซึ่งได้กล่าวถึงไว้ในหัวข้อที่ 2. ส่วนในหัวข้อที่ 3. จะอธิบายขั้นตอนการสร้างตัวกรองสัญญาณรบกวนด้วยค่าความน่าจะเป็น สำหรับหัวข้อที่ 4. จะแสดงผลลัพธ์ที่ได้ ซึ่งจะเปรียบเทียบกับตัวกรองสัญญาณเฉลี่ย และตัวกรองสัญญาณมัธยฐาน พร้อมกับอภิปรายถึงประสิทธิภาพ โดยใช้การประเมินผลเชิงตัวเลขเป็นบรรทัดฐาน ส่วนหัวข้อสุดท้ายจะสรุปผลพร้อมกับนำเสนอแนวทางที่จะพัฒนาต่อไป

2. การกรองสัญญาณรบกวน

สำหรับวิธีการกำจัดสัญญาณรบกวนที่นำเสนอนี้ จะใช้การเปรียบเทียบค่าความน่าจะเป็นของจุดภาพในย่านใกล้เคียง โดยจุดภาพจะถูกมองเป็นสัญญาณรบกวน เมื่อเปรียบเทียบกับจุดภาพบริเวณใกล้เคียงแล้วปรากฏว่ามีค่าแตกต่างกับจุดภาพอื่นอย่างมาก ดังนั้นค่าเทรชโฮลด์ที่ใช้เปรียบเทียบเฉพาะจุดภาพในบริเวณใดๆ เมื่อนำมาใช้ร่วมกับค่าความน่าจะเป็นในบริเวณเดียวกัน ก็สามารถใช้ระบุจุดภาพที่เป็นสัญญาณรบกวนได้ โดยดูจากค่าถ่วงน้ำหนักแล้วนำมาเปรียบเทียบกับค่าความน่าจะเป็น ถ้าจุดภาพที่ i; $(i = 0, 1, 2, \ldots, M \times N - 1)$ อยู่กึ่งกลางวินโดว์ โดยมีค่าของจุดภาพเป็น g_i ดังนั้นค่าที่ใช้เปรียบเทียบสำหรับจุดภาพที่ i ที่กำหนดขึ้นมาตามกฎของ Weber-Fechner [6] เป็นดังนี้

$$C_i = \frac{\left|g_i - \bar{g}_i\right|}{\bar{g}_i} \qquad (1)$$

ที่

$\bar{g}_i$ เป็นค่าเฉลี่ยของจุดภาพในวินโดว์ที่ i ดังแสดงในรูปที่ 1.

$$W_i = \begin{bmatrix} g(1,1) & g(1,2) & g(1,3) \\ g(2,1) & g_i(2,2) & g(2,3) \\ g(3,1) & g(2,3) & g(3,3) \end{bmatrix}$$

รูปที่ 1 วินโดว์ W_i ที่มีขนาด 3×3

ส่วน g_i เมื่อเทียบกับวินโดว์ W_i จะได้ว่า $g_i = g_i(2,2)$ และค่าความน่าจะเป็นที่จะใช้ในการเปรียบเทียบในบริเวณนี้คือ

$$P_i = \frac{C_i}{\sum_{i=1}^{n} C_i} \qquad (2)$$

ที่ n เป็นจำนวนสมาชิกในวินโดว์ W_i

ค่าความน่าจะเป็น P_i ในสมการที่ (2) สามารถนำไปประมาณค่าความน่าจะเป็นของจุดภาพที่จะเป็นสัญญาณรบกวนได้ เนื่องจากถ้าจุดภาพมีค่าใกล้เคียงกันค่าความน่าจะเป็น P_i ก็จะเข้าใกล้ศูนย์ หรือถ้าเราพิจารณาในเชิงข้อสารสนเทศก็สามารถกล่าวได้ว่า ระดับความไม่แน่นอน (Uncertainty) ที่จะหาจุดภาพที่เป็นสัญญาณรบกวนมีค่าเข้าใกล้ศูนย์ ดังนั้นเมื่อวัดความไม่แน่นอนของจุดภาพว่าจะเป็นสัญญาณรบกวนหรือไม่ ก็สามารถพิจารณาได้จากค่าเอ็นโทรปี ของจุดภาพในบริเวณนั้น ซึ่งสามารถกำหนดได้ดังนี้

$$H_i = -\sum_{i=1}^{n} P_i \log P_i \qquad (3)$$

ดังนั้นถ้าจุดภาพที่ i ในบริเวณใดที่เป็นสัญญาณรบกวนหรือ Isolated point ค่าเอ็นโทรปี H_i ก็จะมีค่าสูง เนื่องจากค่าความน่าจะเป็นในย่านนั้นมีค่าสูง แนวความคิดพื้นฐานที่ได้นำเสนอไปแล้วนี้ เป็นการแปลงค่าจุดภาพให้เป็นค่าความน่าจะเป็นที่แสดงถึงความแตกต่างในบริเวณใดบริเวณหนึ่ง เมื่อนำค่านี้ไปเปรียบกับค่า P_c ซึ่งเป็นค่าความน่าจะเป็นอีกตัวหนึ่งที่กำหนดขึ้นมาเป็นค่าเทรชโฮลด์ เพื่อใช้เป็นเกณฑ์ในการตัดสินใจ สำหรับข้อสมมุติฐานของค่า P_c ในวินโดว์ W_i จะอาศัยข้อเท็จจริงที่ว่าทุกจุดภาพมีโอกาสเป็นหรือไม่เป็นสัญญาณรบกวนได้เท่ากัน ดังนั้นค่า P_c จึงสามารถคำนวณได้จากการแจกแจงของยูนิฟอร์ม (Discrete Uniform Distribution)

ค่าความน่าจะเป็น P_i ในสมการที่ (2) จะถูกมองว่ามีโอกาสเป็นสัญญาณรบกวนได้สูงถ้าค่า H_i มีค่าสูง ซึ่งกรณีนี้แสดงให้เห็นว่าถ้าแต่ละวินโดว์ W_i มีพื้นที่ย่อยสองส่วนหรือมากกว่า ที่ค่าของจุดภาพในแต่ละส่วนแตกต่างกัน ดังนั้นจุดภาพ g_i มีโอกาสเป็นได้สูงที่จะถูกมองว่าเป็นสัญญาณ

รบกวนเมื่อค่าความน่าจะเป็น P_i มากกว่าหรือเท่ากับค่าความน่าจะเป็น P_c ดังนั้นในวินโดว์ W_i ก็จะถูกนำไปประมวลผล เพื่อกำจัดสัญญาณรบกวนโดยหาค่าเฉลี่ย หรือถ้าให้ดีกว่าก็ใช้ค่ามัธยฐาน แต่ถ้าในวินโดว์ W_i มีเฉพาะจุดภาพที่มีค่าใกล้เคียงกันค่า H_i ก็จะต่ำทำให้ค่า P_i น้อยกว่าค่า P_c ดังนั้นจุดภาพ g_i จะไม่ถูกประมวลผลเพื่อกำจัดสัญญาณรบกวน

3. อัลกอริทึม

ก่อนที่จะกล่าวถึงรายละเอียดของขั้นตอนการประมวลผลเพื่อกำจัดสัญญาณรบกวน ด้วยการใช้ค่าความน่าจะเป็นของจุดภาพในบริเวณใกล้เคียง ของเกริ่นนำเกี่ยวกับตัวแปรและฟังก์ชันต่างๆ ที่นำมาใช้ในการประมวลผล ซึ่งมีดังนี้

3.1 ข้อมูลภาพที่เป็นอินพุต $g(x,y)$ ซึ่งเป็นจุดภาพที่ตำแหน่ง (x,y) ภาพที่ใช้ในการประมวลผลเป็นภาพขนาด M×N ดังนั้นตัวชี้ตำแหน่งของจุดภาพ (x,y) จะแปรค่าไป โดย x = 0, 1, 2, ..., M-1 ส่วน y = 0, 1, 2, ..., N-1 สำหรับภาพที่นำมาใช้เป็นภาพเกรย์สเกลขนาด 8 บิต

3.2 วินโดว์ W_i เมื่อ i = 0, 1, 2, ..., (M×N-1) เป็นตัวจำกัดบริเวณที่จะประมวลผลภาพในย่านหนึ่งๆ โดย W_i จะเป็นเมทริกซ์ขนาด k×k ซึ่งในที่นี้เรากำหนดให้ k=3 ดังนั้นค่าความน่าจะเป็นที่ใช้ในการวัดว่าบริเวณใดมีสัญญาณรบกวนหรือไม่ ก็จะวัดจากวินโดว์นี้

3.3 MeanWindowOpteration เป็นฟังก์ชันที่ใช้ในการหาค่าเฉลี่ยที่ตำแหน่ง (x,y) ใดๆ ในวินโดว์ W_i และส่งค่าเฉลี่ยนั้นกลับมา

3.4 SumWindowOpteration เป็นฟังก์ชันที่ใช้หาค่าผลรวมของค่าเฉลี่ยที่คำนวณได้จากสมการที่ (1) ในวินโดว์ W_i ที่ตำแหน่ง (x,y) ใดๆ แล้วคำนวณค่าความน่าจะเป็นและค่าเอ็นโทรปีในสมการที่ (2) และ (3) ส่งค่าผลรวมนี้กลับมา

3.5 RemoveNoiseOperation เป็นฟังก์ชันที่ใช้ในการกำจัดสัญญาณรบกวน เมื่อตรวจพบว่าบริเวณนั้นมีสัญญาณรบกวน ซึ่งภายในฟังก์ชันนี้จะมีตัวกรองสัญญาณอยู่สามชนิดด้วยกัน สำหรับการเลือกใช้ขึ้นอยู่กับการส่งค่าตัวแปรให้กับตัวแปร f คือ

3.5.1. f = 0 ใช้ตัวกรองสัญญาณค่าเฉลี่ย

3.5.2. f = 1 ใช้ตัวกรองสัญญาณมัธยฐาน

3.5.3. f = 2 ใช้ตัวกรองสัญญาณมัธยฐานแบบหลายระดับ

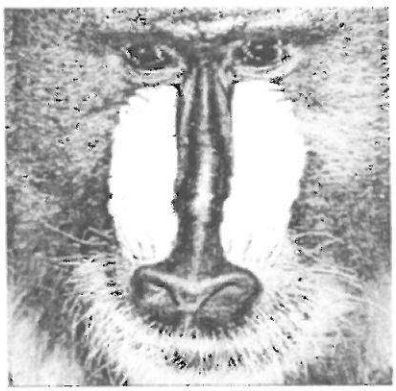
อัลกอริทึมที่ 3.1 แสดงขั้นตอนการกำจัดสัญญาณรบกวนด้วยการใช้เงื่อนไขของความน่าจะเป็น

```
void EntropyFilter()
{
  float   g̅(x,y) , gBar, p[2];
  for(int x=0; x<M; x++)
   for(int y=0; y<N; y++)
   {
     gBar = MeanWindowOpteration(x, y);
     g̅(x,y) = |g(x,y) − gBar| / gBar ;
   }
  for(x=0; x<M; x++)
  for(int y=0; y<N; y++)
  {
     p = SumWindowOpteration(x, y);
     p[1] = 1/p[1];
     if(p[0] > p[1])
       f(x,y) = RemoveNoiseOpteration(1);
     else
       f(x,y) = g(x,y);
  }
}
```

4. ผลการทดลองและเปรียบเทียบ

สำหรับการทดลองเราใช้ภาพมาตรฐาน Lena กับรูปลิงบาบูนที่มีขนาด 256×256 จุด มาเพิ่มสัญญาณรบกวนชนิดอิมพัลส์เข้าไปในระดับความหนาแน่นของสัญญาณต่างๆ กัน โดยสัญญาณรบกวนที่เพิ่มเข้ามามี 2 รูปแบบคือ

ก. รูป Lena



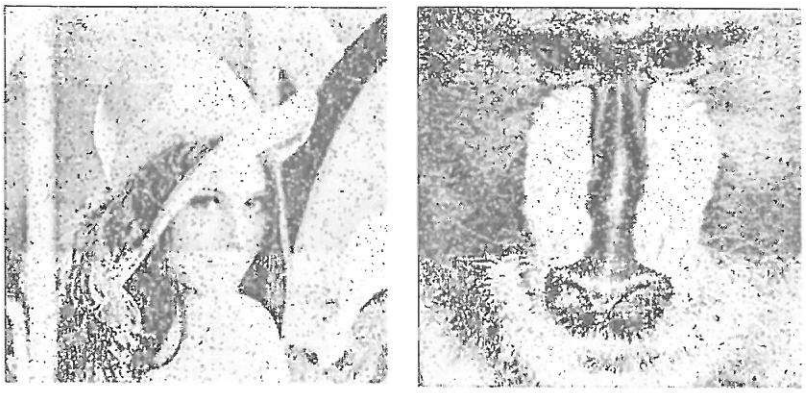
ข. ภาพลิงบาบูน

รูปที่ 2. ภาพต้นแบบ

4.1 การผนวกค่าสัญญาณรบกวนเข้ากับข้อมูลภาพเดิม หรือที่เรียกว่า "Additive Noise" ซึ่งสมการดังนี้

$$f(x,y) = \begin{cases} g(x,y) + n & \text{ด้วยความน่าจะเป็น } \frac{1}{2}p \\ g(x,y) - n & \text{ด้วยความน่าจะเป็น } \frac{1}{2}p \\ g(x,y) & \text{ด้วยความน่าจะเป็น } 1-p \end{cases}$$

เมื่อ n เป็นค่าคงที่ และ p เป็นค่าความน่าจะเป็น



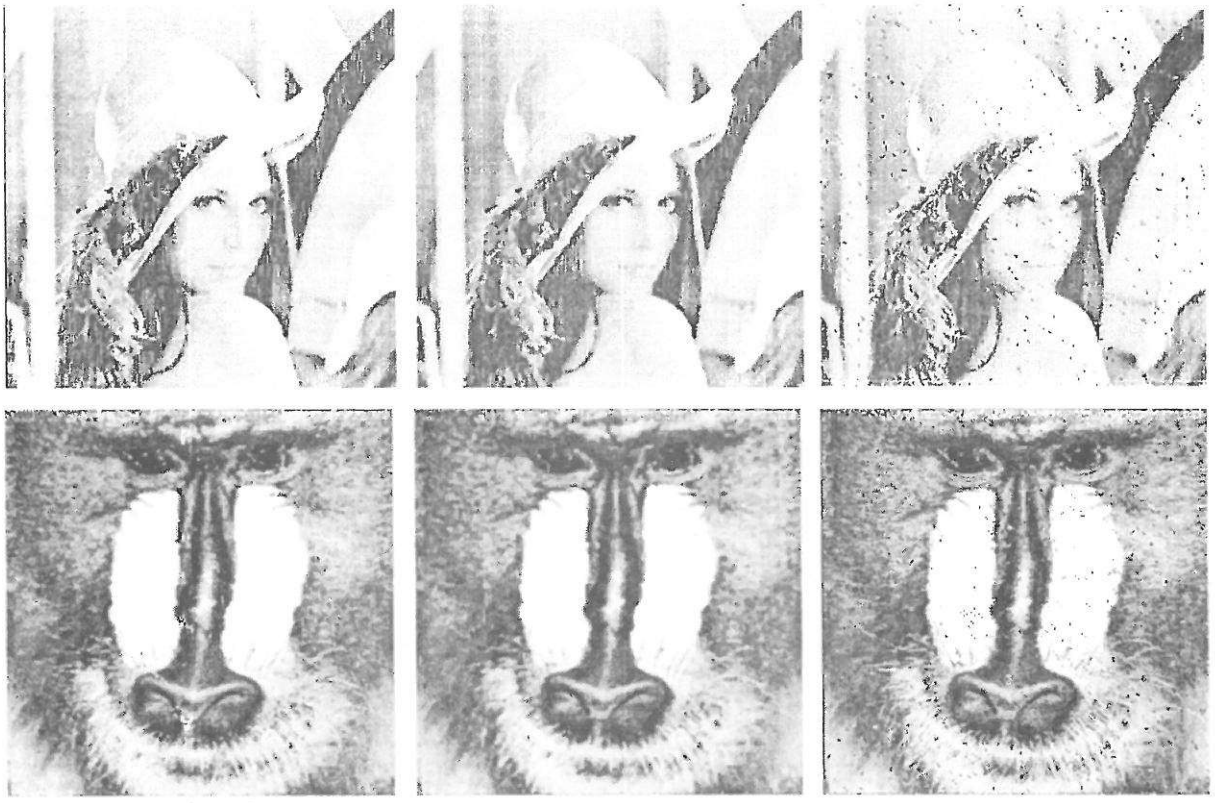
ก. Salt and Pepper ความหนาแน่น 10%

ข. Salt and Pepper ความหนาแน่น 20%

ค. Additive Noise ความหนาแน่น 10%

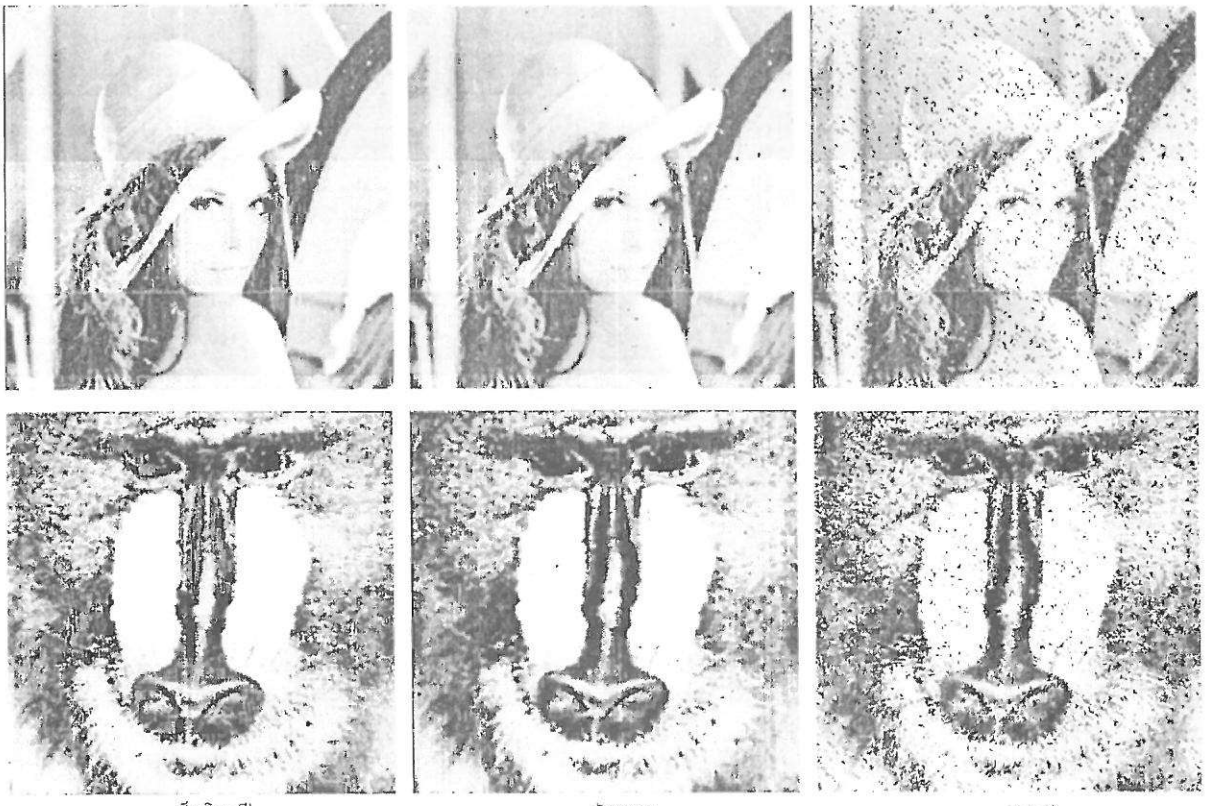
ง. Additive Noise ความหนาแน่น 20%

รูปที่ 3. ภาพที่เพิ่มสัญญาณรบกวนชนิด Salt and Pepper และ Additive Noise



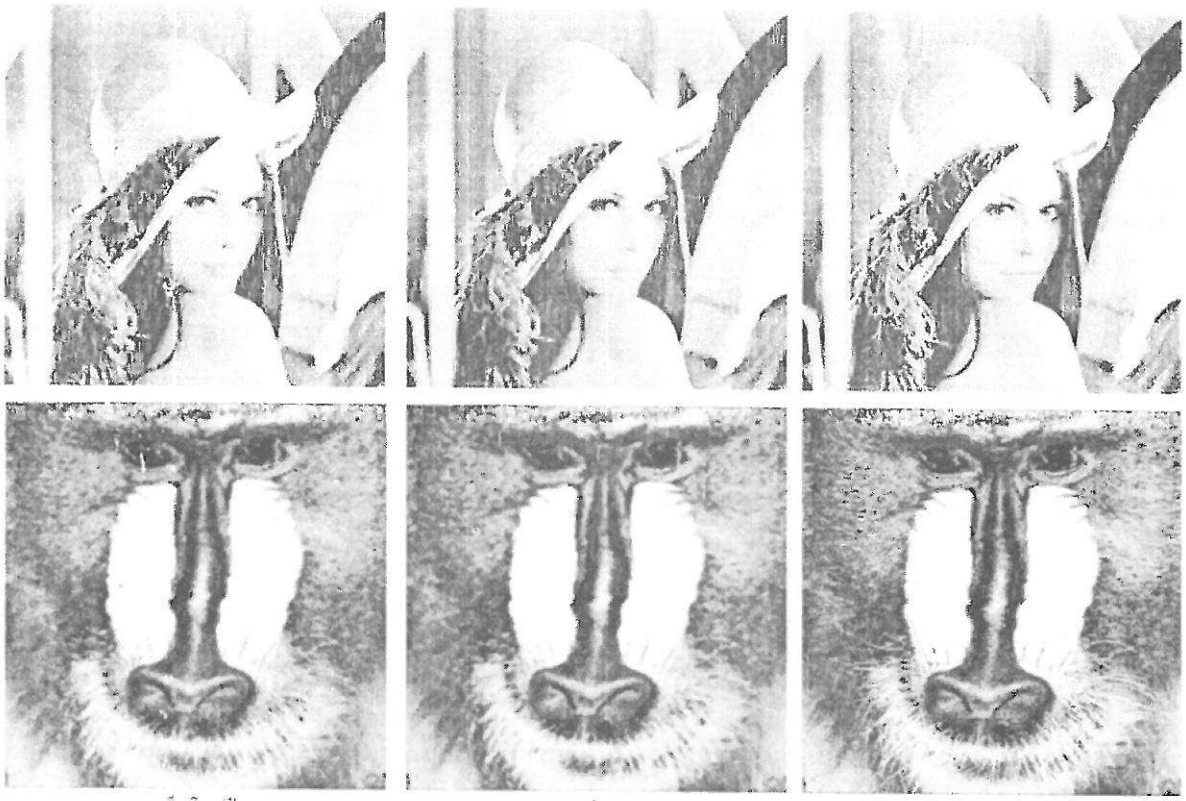
เอ็นโทรปี มัธยฐาน AMMF

ก. ผลลัพธ์ที่ได้จากการประมวลผลด้วยวิธีต่างๆ สำหรับสัญญาณรบกวน Salt and Pepper ที่มีความหนาแน่น 10%



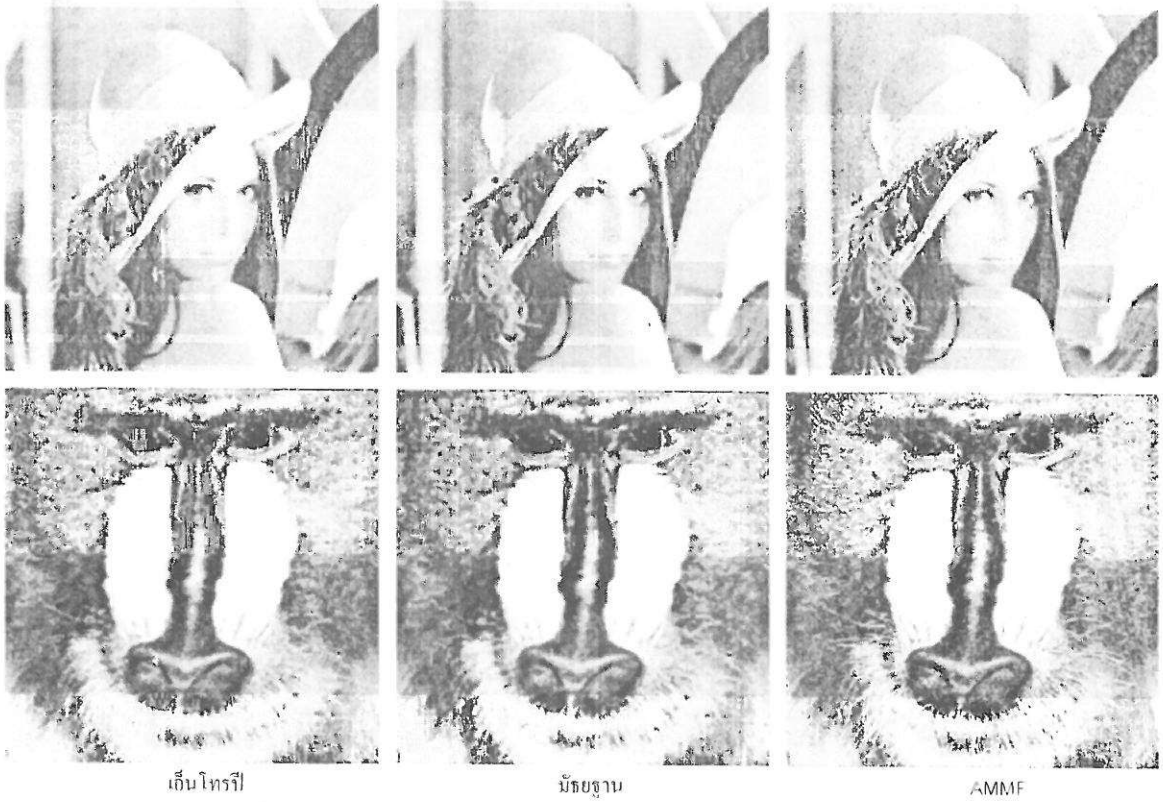
เอ็นโทรปี มัธยฐาน AMMF

ข. ผลลัพธ์ที่ได้จากการประมวลผลด้วยวิธีต่างๆ สำหรับสัญญาณรบกวน Salt and Pepper ที่มีความหนาแน่น 20%



เอ็นโทรปี มัธยฐาน AMMF

ก. ผลลัพธ์ที่ได้จากการประมวลผลด้วยวิธีต่างๆ สำหรับสัญญาณรบกวน Additive ที่มีความหนาแน่น 10%



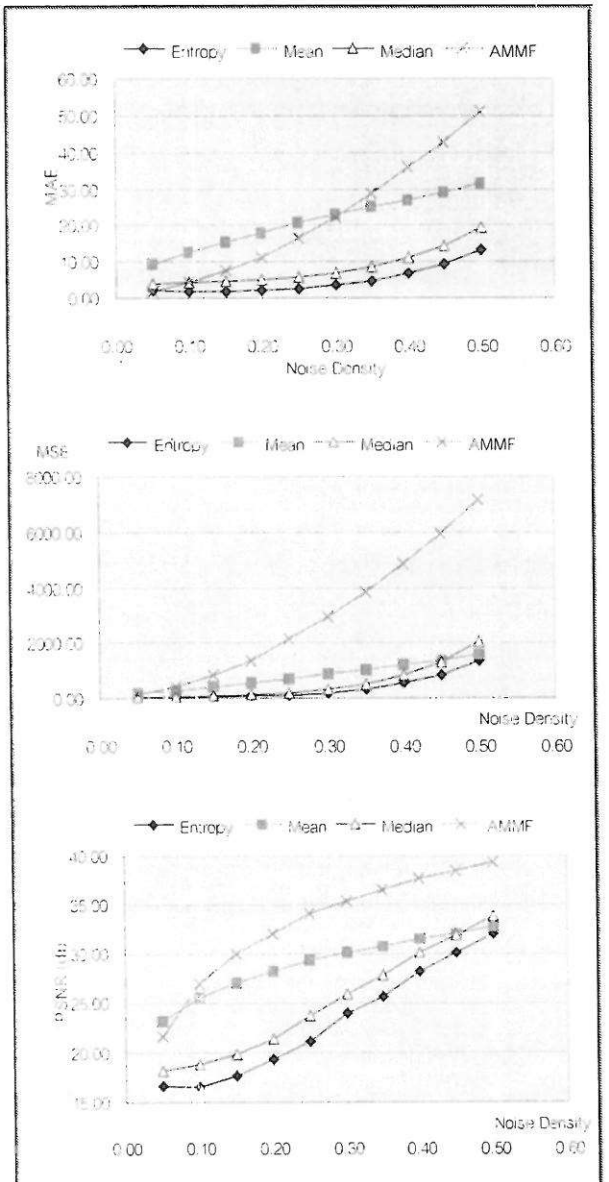
เอ็นโทรปี มัธยฐาน AMMF

ง. ผลลัพธ์ที่ได้จากการประมวลผลด้วยวิธีต่างๆ สำหรับสัญญาณรบกวน Additive ที่มีความหนาแน่น 20%

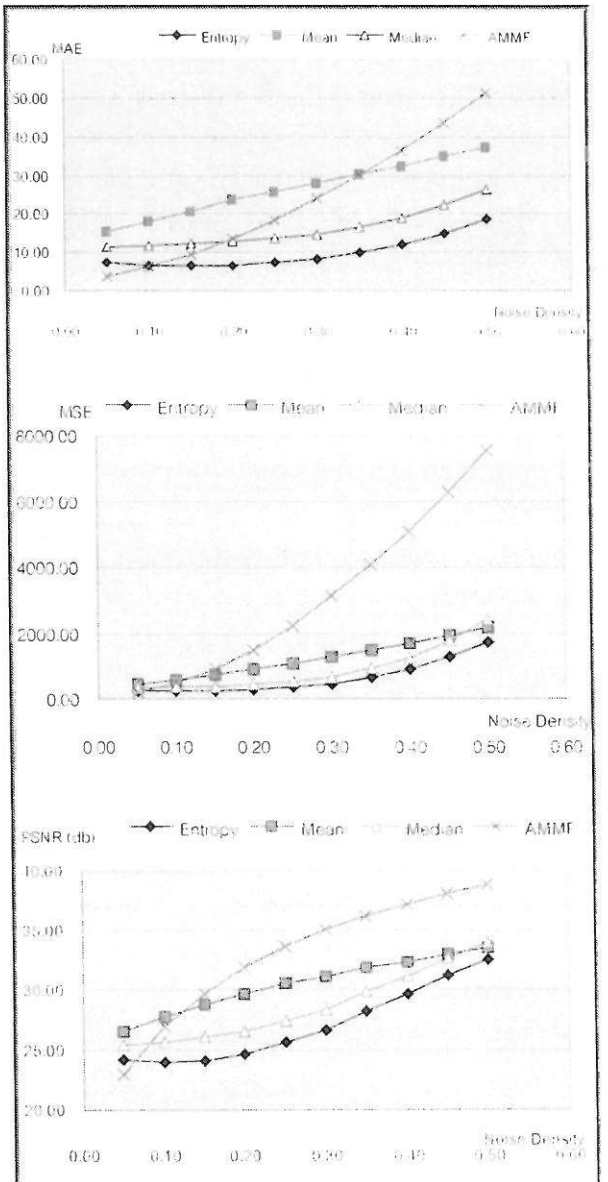
รูปที่ 4. ภาพผลลัพธ์ที่ได้จากตัวกรองสัญญาณชนิดต่างๆ ที่เพิ่มสัญญาณรบกวน Salt and Pepper และ Additive

ตารางที่ 1 ค่าความคลาดเคลื่อนชนิดต่างๆ ที่เกิดขึ้นในตัวกรองสัญญาณแต่ละแบบ เมื่อภาพ Lena มีสัญญาณรบกวนชนิด Additive ปนอยู่ที่ระดับความหนาแน่น 5-50%

ความหนาแน่นของ Noise	เอ็นโทรปี			ค่าเฉลี่ย			มัธยฐาน			AMMF		
	MAE	MSE	PSRN	MAE	MSE	PSRN	MAE	MSE	PSRN	MAE	MSE	PSNR
0.05	2.85	47.06	17.47	5.66	73.78	19.42	3.89	51.67	17.88	1.34	22.51	14.27
0.10	2.87	50.28	17.75	6.01	78.84	19.71	4.08	55.61	18.20	2.03	40.36	16.80
0.15	2.90	53.08	17.99	6.31	83.65	19.97	4.31	60.59	18.57	2.83	61.72	18.65
0.20	3.00	55.97	18.22	6.62	88.33	20.20	4.54	64.48	18.83	3.83	89.27	20.25
0.25	3.26	63.60	18.78	6.88	93.49	20.45	4.81	71.03	19.26	4.90	119.50	21.52
0.30	3.51	70.18	19.21	7.13	98.25	20.67	5.11	77.65	19.64	6.05	152.39	22.57
0.35	3.95	82.31	19.90	7.46	104.43	20.93	5.59	90.00	20.29	7.43	192.64	23.59
0.40	4.45	95.66	20.55	7.64	108.65	21.10	6.08	103.11	20.88	8.84	234.02	24.44
0.45	5.14	114.67	21.34	7.92	114.43	21.33	6.73	120.60	21.56	10.49	279.95	25.21
0.50	5.90	135.39	22.06	8.15	119.87	21.53	7.45	140.19	22.21	11.88	321.82	25.82



รูปที่ 5. ความคลาดเคลื่อน MAE, MSE และ PSNR ที่เกิดขึ้นในรูป Lena เมื่อเพิ่มสัญญาณรบกวน Salt and Pepper



รูปที่ 6. ความคลาดเคลื่อน MAE, MSE และ PSNR ที่เกิดขึ้นในรูปลิงบาบูน เมื่อเพิ่มสัญญาณรบกวน Salt and Pepper

4.2 การกลับค่าของข้อมูลเป็นค่าต่ำสุดหรือสูงสุดของสัญญาณ โดยจะขึ้นอยู่กับค่าความน่าจะเป็นที่ใช้กำหนดความหนาแน่นของสัญญาณรบกวน ซึ่งสัญญาณรบกวนชนิดนี้มีชื่อเฉพาะว่า "Salt and Pepper" มีสมการดังนี้

$$f(x,y) = \begin{cases} 0 & \text{ด้วยความน่าจะเป็น } \frac{1}{2}p \\ 255 & \text{ด้วยความน่าจะเป็น } \frac{1}{2}p \\ g(x,y) & \text{ด้วยความน่าจะเป็น } 1-p \end{cases}$$

สำหรับค่า 0 และ 255 เป็นค่าต่ำสุดและสูงสุดของสัญญาณภาพขนาด 8 บิต

สำหรับการประเมินผลการทดลองที่จะเสนอมีอยู่สองส่วนด้วยกันคือ ส่วนแรกเป็นการพิจารณาด้วยตาเปล่า โดยใช้รูปภาพสองรูป ที่แสดงไว้ในรูปที่ 2. โดยรูป 2ก. เป็นรูป Lena และรูป 2ข. เป็นรูปลิงบาบูน ซึ่งทั้งสองรูปเป็นภาพต้นแบบที่ยังไม่เพิ่มสัญญาณรบกวนเข้าไป ส่วนในรูปที่ 3. เป็นภาพที่เพิ่มสัญญาณรบกวนชนิด Additive (ที่ n=50) และ Salt and Pepper เข้าไป โดยสัญญาณรบกวนที่เพิ่มเข้าไปจะมีระดับความหนาแน่น 10% และ 20% ของข้อมูลภาพตามลำดับ ส่วนในรูปที่ 4. เป็นภาพที่ผ่านการกรองสัญญาณด้วยอัลกอริทึมที่เราเสนอ และได้แสดงเปรียบเทียบกับตัวกรองสัญญาณค่าเฉลี่ย, ตัวกรองสัญญาณมัธยฐาน ซึ่งตัวกรองสัญญาณทั้งสามแบบนี้จะประมวลผลในวินโดว์ขนาด 3×3 และตัวกรองสัญญาณมัธยฐานแบบหลายระดับ ที่ประมวลผลภายในวินโดว์ขนาด 5×5 (เนื่องจากเป็นวินโดว์ที่ขนาดเล็กที่สุดของวิธีนี้)

สำหรับการประเมินผลในส่วนที่สองเป็นการวัดเชิงตัวเลขที่นำเสนอในรูปของตารางและกราฟเส้น ดังตารางที่ 1. ที่แสดงค่าความคลาดเคลื่อน MAE, MSE และ PSNR เปรียบเทียบกันระหว่างตัวกรองสัญญาณที่เสนอ กับตัวกรองสัญญาณมัธยฐาน และตัวกรองสัญญาณมัธยฐานแบบที่มีหลายระดับ ที่เกิดขึ้นในรูป Lena เมื่อความหนาแน่นของสัญญาณรบกวน Additive ถูกเพิ่มเข้าไป 5-50% ของข้อมูลภาพ

ในรูปที่ 5. แสดงกราฟเส้นของค่าความคลาดเคลื่อน MAE, MSE และ PSNR ที่เกิดขึ้นในรูป Lena เมื่อเพิ่มสัญญาณรบกวนชนิด Salt and Pepper เข้าไปที่เปอร์เซ็นต์ความหนาแน่นของสัญญาณรบกวน 5-50% ของข้อมูลภาพ

รูปที่ 6. แสดงกราฟเส้นของค่าความคลาดเคลื่อน MAE, MSE และ PSNR ที่เกิดขึ้นในรูปลิงบาบูน เมื่อเพิ่มสัญญาณรบกวนชนิด Salt and Pepper เข้าไปที่เปอร์เซ็นต์ความหนาแน่นของสัญญาณรบกวน 5-50% ของข้อมูลภาพ

ในการพิจารณาถึงประสิทธิภาพของตัวกรองสัญญาณ เราจะแยกเป็นกรณีต่างๆ ดังนี้

- ความสามารถในการกำจัดสัญญาณรบกวน ตัวกรองสัญญาณเอ็นโทรปีสามารถกรองสัญญาณรบกวนทั้งชนิด Additive และ Salt and Pepper ได้ดีกว่าวิธีอื่น ดังที่เห็นได้จากภาพที่แสดงไว้ในรูปที่ 4. และการประเมินผลเชิงตัวเลขที่แสดงไว้ในตารางที่ 1. และรูปที่ 5. และ 6. แต่ในกรณีที่เปอร์เซ็นต์ของสัญญาณน้อยกว่า 10% AMMF จะสามารถกำจัดสัญญาณรบกวนได้ดีกว่าวิธีที่เสนอ แต่เมื่อเปอร์เซ็นต์ของสัญญาณรบกวนสูงกว่า 10% วิธีการที่นำเสนอจะคงสภาพเดิมของข้อมูลโดยแยกสัญญาณรบกวนออกไปได้ดีกว่า และเมื่อสังเกตในรูปที่ 4. จะเห็นว่าวิธี AMMF จะมีปัญหากับสัญญาณรบกวนที่ต่อกันเป็นเส้นสั้นๆ ซึ่งวิธีนี้จะไม่สามารถแยกสัญญาณรบกวนชนิดนี้ออกไปได้
- การคงรายละเอียดของข้อมูลเดิมเอาไว้ เนื่องจากวิธีที่นำเสนอไม่ได้ประมวลผลกับทุกจุดภาพทำให้การคงสภาพเดิมของข้อมูลของข้อมูล อย่างขอบของวัตถุภายในภาพหรือรายละเอียดอื่นๆ ไว้ได้สูงกว่าวิธีอื่น ดังจะเห็นได้จากค่าความคลาดเคลื่อนในรูปที่ 5 และ 6. โดยเฉพาะค่า MAE ที่สามารถใช้บอกประสิทธิภาพของตัวกรองสัญญาณ ถึงความสามารถในการคงรายละเอียดของข้อมูลเอาไว้ได้ เมื่อเปรียบเทียบทั้งสี่วิธีจะเห็นว่า ค่าความคลาดเคลื่อนของวิธีที่เสนอโดยรวมจะดีกว่าวิธีอื่น ยกเว้นกรณีที่มีเปอร์เซ็นต์ของสัญญาณรบกวนน้อยกว่า 10% วิธี AMMF จะคงรายละเอียดของข้อมูลได้ดีกว่า

5. สรุป

ตัวกรองสัญญาณรบกวนที่ใช้ค่าเอ็นโทรปีวัดว่าบริเวณใดมีสัญญาณรบกวนอยู่หรือไม่ เพื่อเลือกประมวลผลเฉพาะส่วนที่มีสัญญาณรบกวนเท่านั้น ทำให้ข้อมูลภาพสามารถรักษารายละเอียดต่างๆ ภายในภาพเอาไว้ได้สูง ซึ่งเป็นคุณสมบัติที่เด่นอีกอันหนึ่ง นอกจากความสามารถในการกำจัดสัญญาณรบกวน

ในส่วนของการกำจัดสัญญาณรบกวน วิธีที่เสนอสามารถกำจัดสัญญาณรบกวนได้ดีกว่าวิธี AMMF 1.22 เท่า, ดีกว่าตัวกรองสัญญาณค่าเฉลี่ย 1.13 เท่า และดีกว่าตัวกรองสัญญาณมัธยฐาน 1.06 เท่า (เทียบจากค่า PSNR) ดังนั้นประสิทธิภาพโดยรวมของวิธีที่เสนอจะดีกว่าทั้งสามวิธี ยกเว้นกรณีที่มีสัญญาณรบกวนน้อยๆ วิธี AMMF จะมีประสิทธิภาพที่ดีกว่า

เอกสารอ้างอิง

[1] สาธิต อินทจักร์. การประมวลภาพโดยใช้แบบจำลองฟัซซี่และนิวรอลเน็ตเวิร์ค. วิทยานิพนธ์. สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง. 2538.

[2] G. Qiu, "An Improved Recursive Median Filtering Scheme for Image Processing," IEEE Trans. on Image Processing, Vol. 5, No. 4, pp. 646-648, Apr. 1996.

[3] G. R. Arce and R. E. Foster, "Detail-Preserving Ranked-Order Based Filters for Image Processing," IEEE Trans. on ASSP, Vol. 37, No. 1, pp. 83-98, Jan. 1989.

[4] Xin Wang, "Adaptive Multistage Median Filter," IEEE Trans. on Signal Processing, Vol. 40, No. 4, pp. 1015-1017, Apr. 1992.

[5] W. K. Pratt, *Digital Image Processing*, 2nd ed. New York: Wiley, 1991.

[6] A. Beghdadi and A. Khellaf, "A Noise-Filtering Method using a Local Information Measure," IEEE Trans. on Image Processing, Vol. 6, No. 6, June 1997.