

The image filtering algorithm based on the enlarged pixel domain

Wang Weiqing^{1,a}, Yan Yuhua^{2,b}, Yu Rong^{3,c}

¹ Department of Information Management, Southwest University Rongchang Campus, Chongqing 402460, China

² Department of Computer, Chongqing Education Management School, Chongqing, China, 400066

³ Woman and Child Health Hospital of Rongchang, Chongqing 402460, China

^awwqlhy@163.com(corresponding author), ^b6919811@qq.com, ^c1042382293@qq.com

Keywords: filtering; algorithm; noise; image

Abstract. At the first, the algorithm unifies the all of the pixel domain within the rectangular pixels to (α, β) . At the same time, it gave threshold value γ of the gradient amplitude. Then it calculates the gradient amplitude ∇f of each pixel. Finally, by comparing ∇f and γ , the isolated noise pixel is identified. For the noise pixel, we deal it with the median-filtering algorithm. And for the non-noise pixel, we remain them unchanged. So the algorithm did not lose the information of the original image.

Introduction

During the image generation, transmission and storage process, an image can easily produce a variety of image noise, because the limitations of sensors and other factors. Currently, there are many image filtering algorithms [1-4]. At the same time image filtering is also widely applied in various industries [5-7]. Filtering algorithm is usually divided into linear filtering and nonlinear filtering. The linear filtering has low pass characteristics, after filtering the image is smoother, but the details and edge of the image became very vague. The typical algorithm of nonlinear filtering is median filtering algorithm. Median filtering can maintain a better effect for the isolated noise, however, the filtering capability will decline when the noise is a non-isolated pixel. Reference [2] combined the mean filtering with the adaptive median filtering, and improved the adaptive median filtering. It improved some of the filtering effect, but after filtering, each pixel is not the pixel before it is processed, so its information has been lost. This paper is on the basis of reference [2], proposed the image filtering algorithm based on the enlarged pixel domain. It only consider to filter the noises and the original information of non-noise pixels are reserved.

Algorithm theory

Vector gradient The basic idea of the image filtering algorithm based on the enlarged pixel domain is the identification and treatment of noise, among which the identification of noise is detecting the discontinuousness of gray value. If there is large difference gray value between a pixel and another, then we have reason to believe that the pixel is an isolated noise. The discontinuity of gray level is detected by the first derivative and second derivative. And the first derivative is the gradient. The

gradient of a two-dimensional function $f(x, y)$ is defined as the vector [8]: $\nabla f = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$ (0)

The amplitude of the vector is:

$$\nabla f = \text{mag}(\nabla f) = [G_x^2 + G_y^2]^{1/2} = [(\partial f / \partial x)^2 + (\partial f / \partial y)^2]^{1/2} \quad (1)$$

While calculating, it is often simplified as:

$$\nabla f = G_x^2 + G_y^2 \quad (2)$$

$$\text{or } \nabla f = |G_x| + |G_y| \quad (3)$$

The identification of an isolated noise In the space domain, assuming the size of an image is $m \times n$, and $f(i, j)$ (among which $i = 0, 1, 2, \dots, m-1$, and $j = 0, 1, 2, \dots, n-1$) is an any pixel before eliminating the noise in the image. In order to determine whether $f(i, j)$ is a noise or not, according to the equation 2-3, we calculate the gradient between this pixel and the pixel in its neighborhood domain. The neighborhood domain of this pixel is a 3×3 matrix, and its layout is as follows:

$$\begin{vmatrix} f(i-1, j-1) & f(i-1, j) & f(i-1, j+1) \\ f(i, j-1) & f(i, j) & f(i, j+1) \\ f(i+1, j-1) & f(i+1, j) & f(i+1, j+1) \end{vmatrix} \quad (4)$$

The gradient amplitudes of $f(i, j)$ and all of the pixels in the 3×3 matrix are as follows:

$$\nabla f_1 = |f(i, j) - f(i-1, j-1)| \quad (5)$$

$$\nabla f_2 = |f(i, j) - f(i-1, j)| \quad (6)$$

$$\nabla f_3 = |f(i, j) - f(i-1, j+1)| \quad (7)$$

$$\nabla f_4 = |f(i, j) - f(i, j-1)| \quad (8)$$

$$\nabla f_5 = |f(i, j) - f(i, j+1)| \quad (9)$$

$$\nabla f_6 = |f(i, j) - f(i+1, j-1)| \quad (10)$$

$$\nabla f_7 = |f(i, j) - f(i+1, j)| \quad (11)$$

$$\nabla f_8 = |f(i, j) - f(i+1, j+1)| \quad (12)$$

For the threshold value α , if the above gradient amplitude satisfies: $\sum_{i=1}^8 \nabla f_i \leq \alpha$ (13)

Then $f(i, j)$ is a non-noise pixel, it should be retained. Otherwise, $f(i, j)$ is a noise pixel, and it should be filtered.

Filtering the isolated noise pixel From the previous section we know that if the gradient of pixel $f(i, j)$ does not satisfy $\sum_{i=1}^8 \nabla f_i \leq \alpha$, then $f(i, j)$ is a noise pixel. Suppose $g(i, j)$ (among which $i = 0, 1, 2, \dots, m-1$, and $j = 0, 1, 2, \dots, n-1$) which is the pixel after being processed is corresponding to the original pixel $f(i, j)$. Then the value of $g(i, j)$ can be calculated out by the following equation:

$$g(i, j) = \frac{1}{8} \begin{vmatrix} f(i-1, j-1) & f(i-1, j) & f(i-1, j+1) \\ f(i, j-1) & f(i, j) & f(i, j+1) \\ f(i+1, j-1) & f(i+1, j) & f(i+1, j+1) \end{vmatrix} \quad (14)$$

The amplification principle of pixel threshold From the above analysis, it shows that, in order to determine whether $f(i, j)$ is a noise or not, we can the gradient of $f(i, j)$ and the pixels in its neighborhood domain according to the formula (5) - (12). According to whether the gradient amplitude satisfying (13), we can identify whether $f(i, j)$ is a noise or not. Therefore, in order to identify whether $f(i, j)$ is a noise, we should calculate $f(i, j)$ and the pixels in its 3×3 matrix. The range gap between the different 3×3 matrix in an image is very large, Therefore, it is very difficult to determine the threshold α . If your chosen threshold α is too smaller, the non-noise whose contrast is relatively large is mistakenly identified as a noise. And if your chosen threshold is too larger, the noise whose contrast is relatively small is mistakenly identified as a non-noise. Therefore, in the same image, we have to choose different threshold value for different 3×3 matrix domain. So the image processing will become more complex. By carefully analyzing we can find that the pixel range in different regions of the same image is quite different, but the pixel value in the same 3×3 matrix domain has only very small differences. So far, we can consider to magnify the pixel rang of different 3×3 matrix domains in the same image to the same pixel range. So we can select only one threshold for different 3×3 matrix domains in an image. The basic idea of magnification range is that: To unify the pixel $f(i, j)$ and the eight pixels in its 3×3 matrix domain that satisfy the equation (13) to the same range (α, β) . And its specific amplification is as follows:

1, To extract the extremum of a matrix domain: It scans all of the pixels $f(i, j)$ in a 3×3 matrix domain, then use (15) and (16) to respectively calculate the pixel range $(\alpha_{\min}, \alpha_{\max})$.

$$\alpha_{\max} = \max \begin{pmatrix} f(i-1, j-1) & f(i-1, j) & f(i-1, j+1) \\ f(i, j-1) & f(i, j) & f(i, j+1) \\ f(i+1, j-1) & f(i+1, j) & f(i+1, j+1) \end{pmatrix} \quad (15)$$

$$\alpha_{\min} = \min \begin{pmatrix} f(i-1, j-1) & f(i-1, j) & f(i-1, j+1) \\ f(i, j-1) & f(i, j) & f(i, j+1) \\ f(i+1, j-1) & f(i+1, j) & f(i+1, j+1) \end{pmatrix} \quad (16)$$

2, To extract the amplification factor: From the above formula (15) and (16), we can obtained the pixel range $(\alpha_{\min}, \alpha_{\max})$ in a 3×3 matrix domain. If the uniform range is (α, β) , then the amplification factor of the pixel range is as follows: $k = (\beta - \alpha) / (\alpha_{\max} - \alpha_{\min})$ (17)

Among which: $\alpha_{\max} \neq \alpha_{\min}$

3, To extract the new pixel value: $f(i, j)$ is the original value of any pixel in the 3×3 matrix domain that satisfies the formula (4). To identify the noises, its new value $g(i, j)$ is calculated as follows: $g(i, j) = f(i, j) * k - \alpha_{\min} * k$ (18)

4, To identify the noises: Assuming threshold of identifying noise is γ , by (18), if the $g(i, j)$ satisfies (19), (i, j) is a noise. Otherwise, it is a non-noise. $g(i, j) \geq \gamma$ (19)

Filter Design of filtering algorithm

Algorithm of noise identification Based on the above analysis, it unifies the pixel $f(i, j)$ and the eight pixels in its 3×3 matrix domain that satisfy the equation (13) to the same range (α, β) . By (19), we can determine whether it is a noise or not. The identifying algorithm of pseudocode is as follows:

```

1  cr[1...9] ← { GetPixel(i - 1, j - 1), GetPixel(i - 1, j), GetPixel(i - 1, j + 1), GetPixel(i, j - 1),
GetPixel(i, j), GetPixel(i, j + 1), GetPixel(i + 1, j - 1), GetPixel(i + 1, j), GetPixel(i + 1, j + 1) }
2  cr[1...9] ← { c[0].R, c[1].R, c[2].R, c[3].R, c[4].R, c[5].R, c[6].R, c[7].R, c[8].R }
3  cg[1...9] ← { c[0].G, c[1].G, c[2].G, c[3].G, c[4].G, c[5].G, c[6].G, c[7].G, c[8].G }
4  cb[1...9] ← { c[0].B, c[1].B, c[2].B, c[3].B, c[4].B, c[5].B, c[6].B, c[7].B, c[8].B }
5  kr ← MaxMin(cr[1...9], 9)
6  kg ← MaxMin(cg[1...9], 9)
7  kb ← MaxMin(cb[1...9], 9)
8  If (kr.max != kr.min)
    then fdr ← 1000 / (kr.max - kr.min)
    zxr ← kr.min * fdr;
    for k ← 0 to 9
        cr[k] ← cr[k] * fdr - zxr // To enlarge the value of the R of the pixel domain to 0-1000
9  if kg.max != kg.min
    then fdg ← 1000 / (kg.max - kg.min)
    zyg ← kg.min * fdg
    for k ← 0 to 9
        cg[k] ← cg[k] * fdg - zyg // To enlarge the value of the G of the pixel domain to 0-1000
10 if (kb.max != kb.min)
    then fdb ← 1000 / (kb.max - kb.min)
    zxb ← kb.min * fdb;
    for k ← 0 to 9
        cb[k] ← cb[k] * fdb - zxb // To enlarge the value of the B of the pixel domain to 0-1000
11 if (Abs(cr[4] - cr[0]) > yz && Abs(cr[4] - cr[1]) > yz && Abs(cr[4] - cr[2]) > yz && Abs(cr[4] -
cr[3]) > yz && Abs(cr[4] - cr[5]) > yz && Abs(cr[4] - cr[6]) > yz && Abs(cr[4] - cr[7]) > yz
&& Abs(cr[4] - cr[8]) > yz) then return true

```

```

12  else if (Abs(cg[4] - cg[0]) > yz && Abs(cg[4] - cg[1]) > yz && Abs(cg[4] - cg[2]) > yz &&
Abs(cg[4] - cg[3]) > yz && Abs(cg[4] - cg[5]) > yz && Abs(cg[4] - cg[6]) > yz && Abs(cg[4] -
cg[7]) > yz && Abs(cg[4] - cg[8]) > yz)    then  return true
13  else if (Abs(cb[4] - cb[0]) > yz && Abs(cb[4] - cb[1]) > yz && Abs(cb[4] - cb[2]) > yz &&
Abs(cb[4] - cb[3]) > yz && Abs(cb[4] - cb[5]) > yz && Abs(cb[4] - cb[6]) > yz && Abs(cb[4] -
cb[7]) > yz && Abs(cb[4] - cb[8]) > yz)    then  return true
14  else      return false

```

Among the algorithm, we will use a function that can return the maximum and minimum value. The function must return two values, so we use a structure to store the two values. The structure is defined as follows:

```

struct Maxm// Define a structure including the maximum and minimum

```

```

{ public int max; public int min; }

```

Accordingly, the algorithm of the maximum and minimum function is as follows:

```

1  function MaxMin(a[1...n], n) // Extract the maximum and minimum of the array
2      Maxm dx;
3      dx.max ← a[0];
4      dx.min ← a[0];
5      for i ← 1 to n
          if (a[i] > dx.max)    then dx.max ← a[i]
          else if (a[i] < dx.min) then dx.min ← a[i]
6      return dx

```

The image filtering algorithm based on the enlarged pixel domain Based on the above analysis, the basic idea of the image filtering algorithm based on enlarged pixel domain is: To identify whether $f(i, j)$ is a noise pixel or not. If it is a noise, then we use the pixel in its 3×3 matrix domain to calculate the noise. Otherwise, we will not change the pixel. So, the image filtering algorithm based on enlarged pixel domain is as follows:

```

1      for i ← 1 to Width - 2 // Width is the width of processed image
2      for j ← 1 to Height - 2 // Height is the height of processed image
3          if (zaosheng(i, j, 500, box1))
4              then c1 ← GetPixel(i - 1, j - 1)
                  c2 ← GetPixel(i - 1, j)
                  c3 ← GetPixel(i - 1, j + 1)
                  c4 ← GetPixel(i, j - 1)
                  c5 ← GetPixel(i, j + 1)
                  c6 ← GetPixel(i + 1, j - 1)
                  c7 ← GetPixel(i + 1, j)
                  c8 ← GetPixel(i + 1, j + 1)
                  kr ← (c1.R + c2.R + c3.R + c4.R + c5.R + c6.R + c7.R + c8.R) / 8)
                  kg ← (c1.G + c2.G + c3.G + c4.G + c5.G + c6.G + c7.G + c8.G) / 8)
                  kb ← (c1.B + c2.B + c3.B + c4.B + c5.B + c6.B + c7.B + c8.B) / 8)
                  SetPixel(i, j, Color.FromArgb(kr, kg, kb))

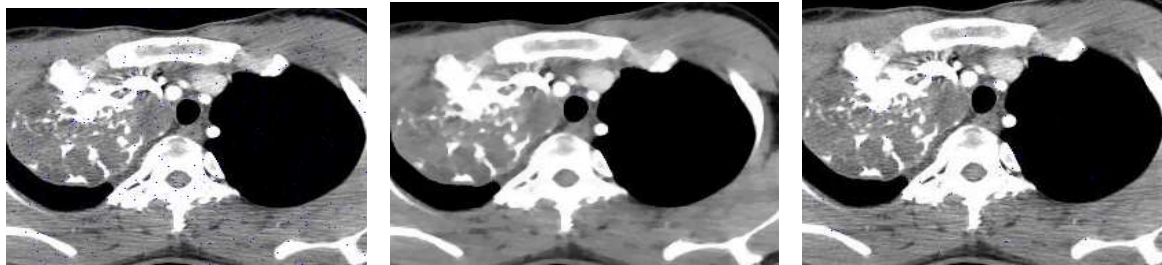
```

Experiment and comparative analysis of the algorithms

To verify the correctness and effectiveness of the filtering algorithm, we select this experimental environment: CPU is AMD Athlon (tm) 64 X2 Dual Core Processor 3600+, its frequency is 2GHz; the memory is 1G; Operation system is Microsoft Windows XP Professional Service Pack 3; the programming language is Visual C#.net 2005. The experimental image came from the network of CHINA MEDICAL IMAGING, and it is a lung image of a woman. The experimental results are as follows:

The figure 2-1 is the experimental results, in which, figure a is the original image including noises, figure b is its result after filtering by the median filtering algorithm, and figure c is its result after filtering by the image filtering algorithm based on the enlarged pixel domain.

It can be seen from Figure 2-1 that the image after being filtered by the median filtering algorithm is smoother, and the isolated noises, through its points in the neighborhood can be restrained. But at the same time, each pixel in the image itself is not the original pixel, and its value is the value of one of its neighborhood pixels. Therefore, some of the information in the image is damaged. At the same time there is a certain degree of damage in the edge in the image. It also brought some problems to the image segmentation. That is as shown in Figure 2-1 b. After being filtered by the image filtering algorithm based on the enlarged pixel domain that is put forward in this paper, the isolated noises of the image have been effectively removed, and the information of the non-noises is unchanged. Therefore, there is no information is damaged in the image, and there is no edge is damaged in the image. That is as shown in Figure 2-1 c.



a. Original image b. Median filtering algorithm c. Filtering algorithm of enlarging pixel domain

Fig 2-1 The filtering experiment

Conclusion

It is difficult to select different filtering thresholds in different image regions, because the contrasts of different regions are different. Therefore, the edges detected by the traditional detection algorithm are much fuzzy, at the same time, the original image has some distortions, because each non-noise pixel in the image is not its original pixel, and it is the average or median of pixels of its neighborhood domain. After being filtered by the image filtering algorithm based on the enlarged pixel domain that is put forward in this paper, the isolated noises of the image have been effectively removed, and the information of the non-noises is unchanged. Therefore, there is no edge is damaged in the image, and image segmentation is not affected. The research of this algorithm forms the base of the further research of image processing.

References

- [1] LI Shi-fei, WANG Ping, SHEN Zhen-kang. Noise removal model with anisotropic diffusion based on visual gradient.[J]. Computer Engineering and Applications. 2010,46(2):167-169.
- [2] WANG Xiao-kai, LI Feng. Improved adaptive median filtering.[J]. Computer Engineering and Applications. 2010,46(3):175-176,218
- [3] WANG Qu, CHEN Li, ZHOU Jin-yun. Morphological Correlator Based on Laplace filtering[J]. Opto—Electronic Engineering. 2010,37(8):140-144
- [4] YANG Yah, CHEN Qi-qiang, WAN Yi. Fast methods for estimating, detecting and filtering random valued impulse noise level.[J]. Journal of Lanzhou University(Natural Sciences. 2010,46(4):120-123
- [5] PAN Bo, FENG Jin—Fu, LI Qian, LI Bin. TRACKING ALGORITHM FOR MULTIMODAL / IRM ULTRASOUND FUSION[J]. J. Infrared Millim. Waves. 2010,29(3):230-235
- [6] LI Changzhong, FAN Liping, ZHANG Jiliang, MIAO Shengjun, WANG Yunfei. Application of Kalman filtering to high and steep slope deformation monitoring prediction of open pit mines[J]. Journal of University of Science and Technology Beijing. 2010,32(1):8-13.
- [7] You—YANG Cheng—su, HUANG Yong—xuan. A New De-noising Method for X—ray Image Using Fuzzy Weighted Median Filter[J]. ACTA PHOTONICA SINICA. 2010,39(8):1372-1376.
- [8] Rafael C.Gonzalez, Richard E.Woods, Steven L.Eddins write. Ruan Qiu-qi translate. Digital Image Processing Using MATLAB (M). Beijing: Publishing House of Electronics Industry. 2005