

오차 확산 방법의 절삭 오차 문제

Truncation Error Problem of Error Diffusion Method

조청운*

Cheung-Woon Jho*

요 약

오차 확산 방법은 디지털 하프토닝 방법 중 하나로 현재 처리하는 픽셀의 양자화 오차를 아직 처리하지 않은 주변 이웃 화소에 확산시켜 변환된 흑백 영상의 품질을 높이는 방법이다. 이러한 방법은 발생한 양자화 오차를 주변으로 분산하는 과정에서 부분적으로 오차의 합이 증가하거나 감소하는 경우가 발생하게 된다. 본 논문에서는 오차 확산의 대표적인 방법인 Floyd-Steinberg 방법과 Jarvis-Judice-Ninke의 방법, Stucki의 방법, Shiau-Fan의 방법을 분석하고 이에 대한 해결 방법을 제시한다.

Abstract

The error diffusion method is one of the digital halftoning methods that diffuses quantization errors of current processing pixel to neighboring pixels and get a high-quality black-white image. This method has the problematic case which partially increase or decrease summation of diffused errors in the process of diffusing the quantization error. In this paper, we analyze Floyd-Steinberg method, Jarvis-Judice-Ninke method, Stucki method, and Shiau-Fan method as a representative case of error diffusion methods and propose a solution method of this problem.

Key words : halftoning, Error Diffusion, quantization errors, Truncation Error

I. 서 론

디지털 하프토닝은 매우 제한된 톤으로 원래 이미지의 연속적인 톤을 재현하기 위한 기술이다. 흑백 출력 장치는 물론 컬러 출력 장치에도 디지털 하프토닝을 이용하여 인쇄물을 만들어 낸다. 또한 300dpi가 넘는 고해상도 디스플레이를 사용하는 스마트폰이나 스마트 패드의 이용자 증가와 전자책의 보급으로 인해 디지털 하프토닝의 활용은 매우 다양하고 중요하다고 할 수 있다.

이러한 디지털 하프토닝은 매우 다양한 방법들이 있으며 대표적인 방법으로는 점의 크기로 톤을 표현하는 방법인 AM(amplitude modulated) 디지털 하프토닝과 고정된 점으로 점간의 간격을 조정하여 톤을 표현하는 방법인 FM(frequency modulated) 하프토닝이 있다[1].

FM 디지털 하프토닝 방법 중 오차 확산(error-diffusion) 방법은 영상을 일정한 방법으로 스캔하면서 흑백의 이진 영상으로 출력하고 원래 영상과의 차이를 계산한다. 이 차이를 양자화 오차(quantization er-

* 동서대학교 디지털콘텐츠학부 디지털영상제작전공

· 제1저자 (First Author) : 조청운

· 투고일자 : 2011년 9월 20일

· 심사(수정)일자 : 2011년 9월 20일 (수정일자 : 2011년 10월 24일)

· 게재일자 : 2011년 10월 30일

ror)라고 하는데 이 값을 아직 처리하지 않은 주변 픽셀로 확산시킴으로써 영상의 품질을 향상시킨다.

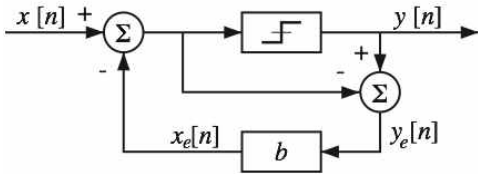


그림 1. 오차 확산 알고리즘의 다이어그램
Fig. 1. Diagram of error diffusion algorithm

오차 확산 하프토닝 방법은 각각의 화소마다 발생하는 양자화 오차를 주변화소에 손실없이 확산시킨다는 가정으로 다양한 확산 필터들이 개발되어 있으나 양자화 오차가 확산되는 과정에서 발생하는 절삭 오차(truncation error)가 부분적으로 발생할 수 있다. 이러한 오차는 기존의 연구에서 고려된 적이 없으나 하프토닝의 결과에 영향을 줄 수 있는 요소이며 기존의 오차 확산 방법의 결과도 이러한 절삭 오차가 주는 영향에 대해 분석할 필요가 있다. 본 논문에서는 이러한 확산 절삭 오차를 분석하고 해결 방법을 제시하고자 한다.

II장에서는 일반적인 오차 확산 하프토닝에 대해 설명하고 III장에서는 본 논문에서 제기하는 확산 절삭 오차에 대한 정의와 이러한 절삭 오차가 대표적인 확산 오차 방법들에서 어떠한 형태로 나타나는지를 분석하고 이에 대한 해결 방법을 제시한다. IV장에서는 기존 방법의 대표적인 문제점을 나타내는 예에 대해 기존 방법과 제안하는 방법의 결과를 비교한다. V장에서는 결론과 함께 향후 연구 방향을 제시한다.

II. 오차 확산 하프토닝

오차 확산에 의한 하프토닝 방법은 그림1에 보인 것처럼 연속적인 톤으로 된 입력 영상 $x[n]$ 을 흑백으로 된 출력 영상 $y[n]$ 으로 변환시키는 과정이며 현재 처리하는 화소에서 발생한 양자화 오차 $y_e[n]$ 를 오차 확산 필터 b_i 에 의해 이웃 화소로 확산시킨다. 이렇게 각 이웃 화소로 확산된 $x_e[n]$ 는 입력 영상의 원래 밝기인 $x[n]$ 과 함께 합산하여 다음 화소의 처리에

사용한다. 이렇게 오차 확산 방법은 흑백 영상으로 변환하는 과정에서 발생하는 양자화 오차를 주변 화소에 분산시킴으로써 결과물을 향상시키는 하프토닝 방법이다.

$$y[n] = \begin{cases} 255, & \text{if } (x[n] + x_e[n]) > T \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

$$y_e[n] = y[n] - (x[n] + x_e[n]) \quad (2)$$

$$x_e[n] = \sum_{i=1}^M b_i \cdot y_e[n-1], \text{ where } \sum_{i=1}^M b_i = 1 \quad (3)$$

영상의 밝기 단계는 가장 일반적인 8비트 영상을 가정하여 0부터 255까지의 256단계의 정수로 밝기가 표현됨을 가정한다.

이러한 오차 확산 방법은 Floyd-Steinberg에 의해 처음 제안되었고 4개의 이웃 화소에 오차를 확산한다[2]. 이를 개선하기 위해 Jarvis와 Judice, Ninke에 의해 확산되는 이웃화소를 12개로 늘려 제한하였으며 [3], Stucki 또한 12개의 이웃화소에 새로운 오류확산 필터를 사용하는 방법을 제안하였다[4]. Shiau-Fan은 이웃화소를 2줄로 제한하여 6개로 줄인 오류 확산 필터를 사용하기도 하였다[5].

또한 이미지를 위에서 아래로 스캔하는 전통적인 방법과 다르게 프랙탈 커브를 이용하여 하나의 이웃 화소에 오차를 확산하는 방법[6]이나 미로 생성 방법을 이용한 방법[7]도 오차 확산 방법을 사용한다.

III. 확산 절삭 오차

3-1 확산 절삭 오차

오차확산 하프토닝에서 각각의 화소마다 발생하는 양자화 오차를 이웃화소로 분산시킬 때 사용하는 비율인 b_i 는 각기 다른 오차 확산 알고리즘에 따라 고유의 각기 다른 계수의 필터를 사용하게 되는데 이때 주변에 확산되는 오차의 값이 더 커지거나 작아지지 않도록 전체의 합이 항상 1이 되어야 한다.

$$\sum_{i=1}^M b_i = 1 \quad (4)$$

따라서 이웃 화소로 확산된 양자화 오차를 모두 더하면 원래 발생한 양자화 오차와 같아야 하므로 항상 다음을 만족해야 한다.

$$\sum_{i=1}^M b_i \cdot y_e[n-1] = y_e[n-1] \quad (5)$$

그러나 양자화 오차의 값에 따라 위의 조건을 만족하지 않는 경우가 발생한다. 이는 0보다 크고 1보다 작은 b_i 를 양자화 오차와 곱하고 다시 정수형으로 변환하는 과정에서 반올림 후 절삭(truncation)으로 인해 원래 양자화 오차값과는 다른 경우가 발생한다. 이러한 오차에 대해서 본 논문에서 양자화 오차가 확산되는 과정에서 절삭에 의해 발생하는 오차이므로 ‘확산 절삭 오차’라고 부르기로 한다.

이러한 확산 절삭 오차는 양자화 오차의 모든 값에 대해 항상 발생하는 것이 아니라 사용하는 확산 필터의 종류와 양자화 오차의 값에 따라 일부의 경우에만 발생한다. 그러나 이러한 확산 절삭 오차는 대부분의 오차 확산 방법에서 발생하며 실험 예에서 보인 것처럼 영상의 결과에도 영향을 주는 것으로 나타났다. 다음 절에서는 대표적인 네 가지 오차 확산 방법의 확산 절삭 오차의 발생에 대해 분석한다.

3-2 대표적인 오차 확산 방법에 대한 분석

본 논문에서는 확산 오차 방법의 대표적인 방법인 Floyd-Steinberg 방법을 비롯하여 Jarvis-Judice-Ninke의 방법, Stucki의 방법, Shiao-Fan의 방법에 대해 확산 절삭 오차를 분석한다. 그림 2에서 그림 5까지 나타낸 그래프는 각 방법들의 확산 절삭 오차를 나타낸 것으로 가로축은 확산될 양자화 오차(A)를 나타내고 세로축은 실제로 확산된 값(B)과 이에 대한 차이(B-A)를 나타내었다. 양자화 오차는 실제로 확산된 값과 비교하기 위하여 사선형태의 그래프로 표시하였다. 이상적인 경우 확산된 값(B)의 그래프가 사선으로 나타낸 그래프와 일치하여야 하나 실제로 분산된 오차의 합(B)은 확산 절삭 오차로 인해 값이 더 크거나 작은 경우가 발생함을 그래프에서 확인할 수 있다.

한편 그래프에서 가로축에 나타낸 양자화 오차의 값은 각 방법의 오차 확산 필터의 공통 분모값으로 제한하여 나타냈으며 각 방법의 확산 절삭 오차의 주

기와 같다. 공통 분모값이 오차 확산 필터의 주기가 되는 이유는 다음과 같다.

예를 들어 Floyd-Steinberg 방법에서 양자화 오차가 35인 경우를 보자. Floyd-Steinberg의 오류 확산 필터는 $\{7/16, 3/16, 5/16, 1/16\}$ 로 모두 16이라는 공통분모를 갖는다. 발생한 양자화 오차를 7/16만큼 확산될 때 35는 $16 \times 2 + 3$ 으로 16으로 나눈 몫과 나머지 형태로 표현할 수 있으므로 확산되는 오차는 다음과 같다.

$$35 \cdot (7/16) = (16 \times 2 + 3) \cdot (7/16) = 7 \cdot 2 + 3 \cdot (7/16)$$

확산 절삭 오차는 분수로 표현되는 부분에서 발생하는 오차에 의해 발생하는데 16으로 나눈 나머지는 항상 16보다 작으므로 확산 절삭 오차의 주기도 16이 된다. 즉, 오차 확산 필터의 공통 분모값이 주기가 된다.

3-2-1 Floyd-Steinberg 방법의 분석

그림 2에 나타낸 Floyd-Steinberg 방법은 양자화 오차를 16으로 나눈 나머지가 1인 경우와 7, 8, 9, 15인 경우에 각각 절삭 오차가 발생한다. 각각의 경우를 분석해 보면 다음과 같다.

- 나머지가 1인 경우: 양자화 오차 1을 오차 확산 필터의 값인 $\{7/16, 3/16, 5/16, 1/16\}$ 으로 분산시키면 소수점으로 각각 0.4375, 0.1875, 0.3125, 0.0625가 되고 이를 반올림하여 절삭하면 정수로 모두 0, 0, 0, 0이 된다. 원래 양자화 오차의 값을 고려하면 7/16인 곳에 1이 확산되고 나머지는 0이어야하지만 어느 곳으로도 양자화 오차가 확산되지 않는다.
- 나머지가 7인 경우: 양자화 오차 7을 오차 확산 필터의 값인 $\{7/16, 3/16, 5/16, 1/16\}$ 으로 분산시키면 소수점으로 각각 3.0625, 1.3125, 2.1875, 0.4375가 되고 이를 반올림하여 절삭하면 3, 1, 2, 0이 되어 합하면 6이 된다. 확산되어야 하는 양자화 오차는 7이지만 실제로는 모두 반올림이 안 되었기 때문에 1이 모자라게 된다.
- 나머지가 8인 경우: 양자화 오차 8을 오차 확산 필터의 값인 $\{7/16, 3/16, 5/16, 1/16\}$ 으로 분산시키면 소수점으로 각각 3.5, 1.5, 2.5, 0.5가 되고

이를 반올림하여 절삭하면 4, 2, 3, 1이 되어 합하면 10이 된다. 확산되어야 하는 양자화 오차는 8이지만 실제로는 모두 반올림이 되었기 때문에 2가 증가하게 된다.

- 나머지가 9인 경우: 양자화 오차 9를 오차 확산 필터의 값인 {7/16, 3/16, 5/16, 1/16}으로 분산시키면 소수점으로 각각 3.9375, 1.6875, 2.8125, 0.5625가 되고 이를 반올림하여 절삭하면 4, 2, 3, 1이 되어 합하면 10이 된다. 확산되어야 하는 양자화 오차는 9이지만 실제로는 모두 반올림이 되었기 때문에 1이 증가하게 된다.
- 나머지가 15인 경우: 양자화 오차 15를 오차 확산 필터의 값인 {7/16, 3/16, 5/16, 1/16}으로 분산시키면 소수점으로 각각 6.5625, 2.8125, 4.6875, 0.9375가 되고 이를 반올림하여 절삭하면 7, 3, 5, 1이 되어 합하면 16이 된다. 확산되어야 하는 양자화 오차는 15이지만 실제로는 모두 반올림이 되었기 때문에 1이 증가하게 된다.



그림 2. Floyd-Steinberg 방법의 분석
Fig. 2. Analysis of Floyd-Steinberg method

3-2-2 Jarvis-Judice-Ninke, Stucki, Shiau-Fan 오차 방법의 분석

Jarvis-Judice-Ninke에 의한 방법은 오차 확산 필터에 12개의 요소를 사용하였으며 {7/48, 5/48, 3/48, 1/48}의 네 가지 값을 각각 2군데, 4군데, 4군데, 2군데에 중복 사용한다. 따라서 오차가 발생할 경우 2배나 4배로 증폭되어 Floyd-Steinberg 방법보다 더 큰 확

산 절삭 오차를 나타낸다. 이를 그림 3에 나타내었으며 Jarvis-Judice-Ninke의 오차 확산 필터가 48의 공통 분모를 가지므로 48의 주기를 보인다. 최대로 큰 오차는 나머지가 23인 경우 -5를, 나머지가 24인 경우 6의 확산 절삭 오차를 나타내어 양자화 오차에는 1밖에 차이가 나지 않지만 분산된 오차의 합에서는 11이나 차이가 난다.



그림 3. Jarvis-Judice-Ninke 방법의 분석
Fig. 3. Analysis of Jarvis-Judice-Ninke method

Stucki 방법의 경우도 Jarvis-Judice-Ninke 방법과 유사하게 12개의 요소를 사용하며 {8/42, 4/42, 2/42, 1/42}의 네 가지 값을 각각 2군데, 4군데, 4군데 2군데에 중복 사용한다. 오차 발생 범위는 -3에서 3 사이로 Jarvis-Judice-Ninke에 비해서는 평균적으로 적게 발생한다.



그림 4. Stucki 방법의 분석
Fig. 4. Analysis of Stucki method

Shiau-Fan의 방법은 5개의 요소를 사용하며 {8/16, 4/16, 2/16, 1/16}을 사용하며 1/16만 두 군데에 중복해

서 사용한다. 확산 절삭 오차는 최대로 1 발생하지만 평균 50%가 발생하여 Floyd-Steinberg보다 발생 빈도가 높다.



그림 5. Shiao-Fan 방법의 분석
Fig. 5. Analysis of Shiao-Fan method

3-3 제안하는 해결 방법

확산 절삭 오차에 대해 본 논문에서 제시하는 문제 해결 방법은 발생한 양자화 오차에 따라 각 이웃 화소로 분산될 값을 테이블 형태로 미리 만들어 놓는 방법이다. 2절에서 설명한 것처럼 오차 확산 필터의 공통 분모값을 주기로 확산되는 값의 소수점의 부분이 반복되므로 주기만큼만 테이블로 만들면 된다. 주기를 이용해서 테이블의 크기를 줄일 수 있는 이유는 아래와 같이 설명할 수 있다.

각각의 오차 확산 방법에서 오차 확산 필터를 정의할 때 분수 형태로 설계하기 때문에 공통된 분모를 갖는 형태로 표현할 수 있다. 예를 들어 Floyd-Steinberg 방법의 경우 오차 확산 필터는 그림 6에 보인 것처럼 7/16, 5/16, 3/16, 1/16으로 16이라는 공통 분모를 갖는다. 따라서 어떤 값의 양자화 오차값이 발생하더라도 다음과 같은 형태로 표시할 수 있다.

$$x = 16n+m$$

여기서 x는 양자화 오차값을 나타내며, n은 x를 16으로 나눈 정수 몫, m은 x를 16으로 나눈 나머지다. 여기에 y/16의 형태로 된 오차 확산 필터 값을 곱하면 다음과 같은 형태가 된다.

$$(16n+m)*(y/16) = yn + (y/16)*m$$

따라서 yn은 두 정수의 곱이므로 직접 계산하고 (y/16)*m 은 m이 0에서 15까지 사이에서 존재하므로 (y/16)을 0에서 15까지 곱한 값을 테이블로 미리 만들어서 m으로 참조하면 된다.

표 1. Floyd-Steinberg 방법에서 양자화 오차에 따른 확산 값의 비교

Table 1. Diffusion value comparison for quantization error in Floyd-Steinberg method.

양자화 오차(A)	오류 확산 필터에 의해 계산된 값	분산된 오차의 합(B)	확산절삭 오차(B-A)	확산 절삭 오차가 없도록 수정한 값
1	0, 0, 0, 0	0	-1	1, 0, 0, 0
2	1, 0, 1, 0	2	0	1, 0, 1, 0
3	1, 1, 1, 0	3	0	1, 1, 1, 0
4	2, 1, 1, 0	4	0	2, 1, 1, 0
5	2, 1, 2, 0	5	0	2, 1, 2, 0
6	3, 1, 2, 0	6	0	3, 1, 2, 0
7	3, 1, 2, 0	6	-1	3, 1, 2, 1
8	4, 2, 3, 1	10	2	3, 1, 3, 1
9	4, 2, 3, 1	10	1	4, 1, 3, 1
10	4, 2, 3, 1	10	0	4, 2, 3, 1
11	5, 2, 3, 1	11	0	5, 2, 3, 1
12	5, 2, 4, 1	12	0	5, 2, 4, 1
13	6, 2, 4, 1	13	0	6, 2, 4, 1
14	6, 3, 4, 1	14	0	6, 3, 4, 1
15	7, 3, 5, 1	16	1	6, 3, 5, 1
16	7, 3, 5, 1	16	0	7, 3, 5, 1

본 논문에서 사용한 테이블의 값은 다음과 같은 두 가지 조건을 만족하도록 하여 구성하였다.

- $i > j$ 이면 $table[i][k] > table[j][k]$ 가 되도록 한다.
- 양자화 오차값이 네 군데로 확산된 오차들의 합과 같아지도록 한다.

$$i = \sum_{k=0}^3 table[i][k]$$

이러한 조건을 유지하면서 새로 수정된 값은 표 1의 가장 오른쪽 열인 ‘수정된 정수’에 나타내었다.

이렇게 16*4=64개의 정수 테이블을 만들어 놓고 계산에 사용한다. 값을 참조하는 방법은 그림 6에 나타난 방법으로 계산한다.

이렇게 확산해야 할 오차 값이 정해지면 실수를 곱하고 다시 정수로 변환하는 과정 없이 테이블에 있

는 값을 참조하기만 하면 된다. 이러한 방법은 기존의 절삭오차 발생을 해결하는 동시에 소수점 계산을 할 필요가 없어진다는 장점을 갖는다.

```
float fs_a = 7/16, fs_b = 3/16, fs_c = 5/16,
      fs_d = 1/16; // Floyd Steinberg mask value

for (y = 0; y < image_height; y++)
{
    for (x = 0; x < image_width; x++)
    {
        new_bw = gray_image[x][y] > middle_gray
                ? white : black;
        bw_image[x][y] = new_bw;
        quant_err = gray_image[x][y] - new_bw;
        gray_image[x+1][y] +=
            (int)(fs_a*(float)quant_err +
                0.5);
        gray_image[x-1][y+1] +=
            (int)(fs_b*(float)quant_err +
                0.5);
        gray_image[x][y+1] +=
            (int)(fs_c*(float)quant_err +
                0.5);
        gray_image[x+1][y+1] +=
            (int)(fs_d*(float)quant_err +
                0.5);
    }
}
```

그림 6. 기존 방법의 의사 코드
Fig. 6. Pseudo code of previous method

```
int table[16][4]= {
{ 0, 0, 0, 0 }, { 1, 0, 0, 0 }, { 1, 0, 1, 0 }, { 1, 1, 1, 0 },
{ 2, 1, 1, 0 }, { 2, 1, 2, 0 }, { 3, 1, 2, 0 }, { 3, 1, 2, 1 },
{ 3, 1, 3, 1 }, { 4, 1, 3, 1 }, { 4, 2, 3, 1 }, { 5, 2, 3, 1 },
{ 5, 2, 4, 1 }, { 6, 2, 4, 1 }, { 6, 3, 4, 1 }, { 6, 3, 5, 1 }
};

for (y = 0; y < image_height; y++)
{
    for (x = 0; x < image_width; x++)
    {
        new_bw = gray_image[x][y] > middle_gray
                ? white : black;
        bw_image[x][y] = new_bw;
        quant_err = gray_image[x][y] - new_bw;
        err_mod = quant_err%16;
        err_div = quant_err/16;
        gray_image[x+1][y] += ta-
            ble[err_mod][0]+err_div*7;
        gray_image[x-1][y+1] += ta-
            ble[err_mod][1]+err_div*3;
        gray_image[x][y+1] += ta-
            ble[err_mod][2]+err_div*5;
        gray_image[x+1][y+1] += ta-
            ble[err_mod][3]+err_div;
    }
}
```

그림 7. 제안하는 방법의 의사코드
Fig. 7. Pseudo code of proposed method

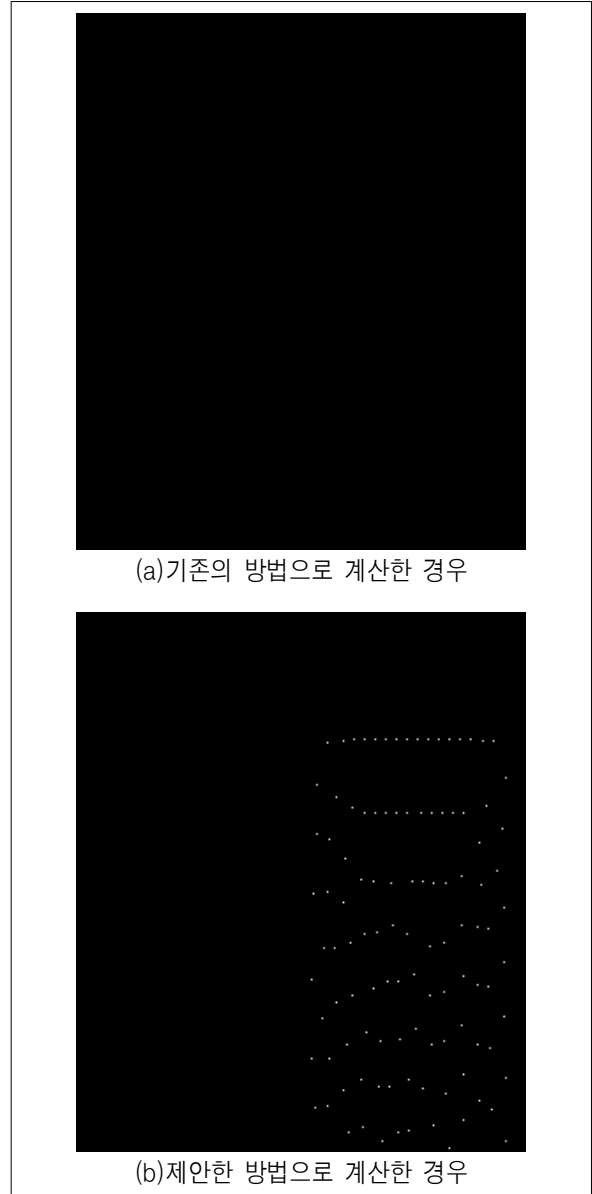


그림 8. Floyd-Steinberg 방법에 대한 기존 방법과 제안한 알고리즘의 비교

Fig. 8. Comparison of previous algorithm and proposed algorithm for Floyd-Steinberg method

IV. 실험 결과

본 논문에서 분석한 오차 확산 방법의 분산 절삭 오차가 발생하는 문제점의 가장 대표적인 예는 최대 밝기가 255인 8비트 영상에서 영상 전체가 밝기 1인 영상이다. 하프토닝의 결과에 흰색 화소가 전혀 발생하지 않는다는 점이다. 밝기가 0 인 경우에는 하프토닝 결과도 전체가 흰색 화소가 전혀 없이 원본 영상

과 같은 영상을 얻게 된다. 그러나 영상의 밝기가 1(1/255)인 경우에는 하프토닝을 할 경우 발생한 양자화 오차가 주변 화소로 분산됨으로써 흰색 화소가 발생해야만 한다. 그러나 양자화 오차를 주변 화소들에 분산 시키는 과정에서 반올림으로 인해 양자화 오차가 주변 화소 어느 곳에도 확산되지 않는다. 따라서 밝기가 0으로만 이루어진 경우 동일하게 흰색 화소가 전혀 발생하지 않는 영상을 얻게 된다(그림 8의 (a)). 반면 이러한 문제점을 해결하기 위해 본 논문에서 제안한 방법의 경우에는 밝기에 비례하여 흰색 화소가 발생한다(그림 8의 (b)). 이러한 문제점은 Stucki의 방법에서는 밝기가 2인 경우에도 Jarvis-Judice-Ninke의 방법에서는 밝기가 3인 경우까지도 하프토닝 결과에 흰색 화소가 발생하지 않는다. 이러한 결과는 절삭오차로 발생하는 대표적인 예가 된다.

V. 결 론

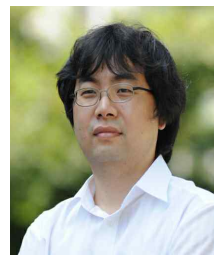
본 논문에서는 오차 확산 방법에 기반한 디지털 하프토닝에서 양자화 오차를 주변 이웃 픽셀들에 분산시키는 과정에서 정수 반올림 계산 과정에서 확산 절삭 오차가 발생하는 경우가 있음을 보였다. 또한 다양한 오차 확산 하프토닝 방법에 대해서 이를 분석하고 이러한 절삭 오차를 해결하기 위한 방법으로 테이블에 기반한 계산 방법을 제시하였다.

본 논문에서는 절삭 오차로 인한 문제점의 하나로 실험에 사용한 예를 제시하였다. 그러나 절삭 오차로 인해 발생할 수 있는 다른 다양한 문제점에 대한 연구가 필요할 것으로 보인다. 오차 확산 알고리즘에 대한 분석도 본 논문에서 분석한 알고리즘 이외에 다양한 오차 확산 방법에 대한 분석이 필요하다. 또한 본 논문에서 제시한 테이블은 기존의 확산되는 값을 참조하여 조건에 맞게 조정하여 만든 것으로 알고리즘에 의해 계산된 것이 아니다. 향후 이러한 확산 값을 자동으로 계산할 수 있는 일반적인 알고리즘의 개발도 필요한 연구로 제시한다.

참 고 문 헌

- [1] D. Lau and G. Arce, *Modern Digital Halftoning*, Marcel Dekker, New York NY, USA 2001.
- [2] Floyd and L. Steinberg, "An Adaptive Algorithm for Spatial Gray Scale," *SID Symposium*, pp. 36-37, 1975.
- [3] J. F. Jarvis, C. N. Judice, and W. H. Ninke, "A Survey of Techniques for the Display of Continuous Tone Pictures on Bilevel Displays", *Computer Graphics and Image Processing*, Vol.5, 1976, pp.13-40.
- [4] P. Stucki, "Image Processing for Document Reproduction", *Advances in Digital Image Processing*, Plenum Press, New York, 1979, pp.177-218.
- [5] J. Shiao and Z. Fan. "A Set of Easily Implementable Coefficients in Error Diffusion with Reduced Worm Artifacts", *SPIE*, vol. 2658 pp. 222-225, Jan. 1996.
- [6] I. H. Witten and R. M. Neal, "Using Peano Curves for Bilevel Display of Continuous Tone Images," *IEEE Computer Graphics and Applications*, pp. 47-52, May 1982.
- [7] 조청운, "미로 생성 알고리즘을 이용한 디지털 하프토닝", *한국향행학회논문지*, 제13권 제6호, pp. 984-990, 2009.

조 청 운 (趙靑雲)



1992년 2월 : 중앙대학교 전자계산학과(공학사)

1994년 2월 : 중앙대학교 컴퓨터공학과(공학석사)

2000년~2003년 : 중앙대학교 첨단영상대학원 영상공학과 연구교수

2004년~현재 : 동서대학교 디지털콘텐츠학부 조교수
관심분야 : 디지털 하프토닝, 비사실적 렌더링, 컴퓨터 그래픽스