

# 동적 블러연산을 이용한 새로운 Anti-Aliasing 알고리즘 설계

김명신\*, 지용준, 이성태, 김판구, 이윤배  
조선대학교 대학원 전자계산학과 멀티미디어 시스템 연구실  
e-mail : mskim@infoman.chosun.ac.kr

Design of A New Anti-Aliasing Algorithm Using  
Dynamic Block Operation

Myoung-Sin Kim\*, Young-Jun Ji, Sung-Tae Lee, Pan-Koo Kim, Yun-Bae Lee  
Multimedia System Lab, Dept. of Computer Science,  
Graduate School, Chosun University

요약

본 논문은 컴퓨터 그래픽 이미지의 데이터를 디지털화 하는 과정에서 Aliasing으로 인하여 손실된 Pixel 정보에 대해 동적 블러연산으로 분할 연산하고 벡터 양자화, Gaussian 함수를 이용하여 손실된 정보들을 보간하여 해상도가 높아진 영상을 얻을 수 있는 새로운 Anti-Aliasing 알고리즘을 제시한다.

Anti-Aliasing의 효과를 더욱 시각적으로 분별 할 수 있도록 하기 위해 Gray 레벨의 이미지로 실험을 하였고, 현재 Graphic을 지원하는 하드웨어 구조의 PC 기반에 변화 없이 적용할 수 있고, 이미지의 질감을 더욱 부드럽게 향상 시킬수가 있다.

## 1. 서론

현재 그래픽 이미지에서 Anti-Aliasing은 그래픽에 적용되는 가장 중요한 요소중 하나이다. 사용 중인 일반 PC의 모니터에 객체를 디스플레이할 때 객체의 해상도는 96PPI(Pixel Per Inch)이다. 그러나 사람의 눈은 훨씬 높은 해상도를 볼수 있다. 그런데, 96PPI의 샘플 해상도는 사람의 눈으로 볼수 있는 해상도보다 낮기 때문에 Aliasing이 일어나게 된다.

Aliasing이란 데이터에 있는 정보를 캡처하기에 불충분한 해상도를 갖는 데이터를 Sampling할 때 원래의 이미지에 대한 정보를 놓치는 현상이다. 대표적인 것으로는 계단(Band) 현상을 들 수 있는데 이러한 현상은 해상도를 증가시키면 시킬수록 계단 현상이 줄어든다. 예를 들어, 영화에서 사용되는 스틸 이미지(정지 화면)들은 실제로 가로, 세로 해상도가 3000×3000이 넘는 이미지를 렌더링(Rendering)해 필름에 담은 것이다. 즉, 일반 PC에서는 해상도가 낮기 때문에 본 논문에서는 Aliasing 효과를 줄일수 있는 컴퓨터 그래픽 이미지 처리를 위한 동적 블러 방법을 이용한 새로운 Anti-Aliasing 알고리즘을 제안한다.

논문의 구성은 2장에서 기존에 사용중인 Anti-Aliasing 방법을 알아보고, 3장에서는 동적 블러 연산을 이용한 Anti-Aliasing 방법을 4장에서는 가우시안 함수를 적용한 실험 내용 및 결과를 서술하고, 기존의 방법들과 비교분석 한다. 5장에서 결론과 향후 연구과제를 제시한다.

## 2. 기존의 Anti-Aliasing 방법

계단 현상을 줄이는 대표적인 Anti-Aliasing의 방법으로 Prefiltering과 Postfiltering방법이 있다.

Prefiltering 방법으로는 그래픽 이미지의 인접한 픽셀 값들의 평균값을 이용하는 방법과 Mask Filtering을 이용하는 방법이 주로 사용된다. 그러나 이 방법은 간단하지만 Aliasing효과를 크게 줄일 수 없다. 그래서 일반적으로 Anti-Aliasing으로 많이 사용하는 것이 Postfiltering의 방법(Supersampling)이다. 이 방법은 샘플링의 해상도를 높이는 방법으로 H/W적인 해상도를 높여 감으로써 계단 현상을 줄어 들게 한다. 일반적으로 정상 해상도에서 정상 크기의 두 배인 이미지의 표본을 만들고 이미지에 Low-pass 필터를 적용하여 Aliasing을 일으키는 고빈도 정보를 제거한다. Low-pass 필터란 이웃 픽셀의 대칭적 평균을 취함으로써 순화 또는 흐리게 하는 필터이다. 고밀도 정보가 제거된 확대 이미지의 크기를 조절하여 원래의 정상의 크기 이미지로 변환시키는 방법이다.

## 3. 동적 블러 연산

본 논문에서 기존의 Anti-Aliasing의 방법보다 Aliasing 효과를 더 줄 일수 있는 동적 블러 연산을 이용한 방법을 제안한다. 제안한 내용의 효과적인 분석을 위하여 Gray Level 이미지를 사용하여 연구 하였다. 일반적인 이미지 영상에는 Aliasing 효과를 가지고 있기 때문에 그림 2 처럼 Stair 현상이 보

이는 블럭이 있을 수가 있다.

이러한 블럭을 부드럽게 보이기 위해서 적용되는 방법이 Anti-Aliasing인데, Stair 현상을 처리하기 위해서는 가장 먼저 화면을 확대하고, 동적 영역으로 블러킹하는 방법을 이용해 영역들을 보간하면서 확대된 이미지의 정보를 가지고 동적 블러킹을 한다.

먼저, 확대하는 방법으로는 최근접 이웃 보간법이나 평균값을 이용한 보간법을 적용하면 더욱 좋은 질감을 얻을수 있다. 그중 최근접 이웃 보간법은 근접한 한 픽셀의 값을 가지고 확대 영상에서 할당받지 못한 빈 픽셀에 바로 옆 픽셀의 데이터 값을 매칭 시키는 방법이고, 평균값을 이용한 보간법은 이웃된 4개나 그 이상의 픽셀 등의 합을 가지고 보간한다.

보간된 영상에서의 동적 블럭 단위로 구분하기 위해 에지 검출을 하는데, 구배 연산자(Gradient Operator)와 콤파스 연산자(Compass Operator)를 이용하면 이미지 내에 존재하는 방향과 크기를 알수 있으며 이미지 영역의 에지를 먼저 검출하고, 검출된 에지의 벡터로 적용된 값을 가지고 블럭을 검출할 수 있다.

영상의 기울기를 그래디언트(Gradient)라고 하는데, 다음과 같이 수학적으로 정의할 수 있다.

$$\nabla F = \left| \begin{matrix} G_x \\ G_y \end{matrix} \right| = \left| \begin{matrix} \left( \frac{\delta f}{\delta x} \right) \\ \left( \frac{\delta f}{\delta y} \right) \end{matrix} \right|$$

$$\nabla f = \text{mag}(\nabla F) = \sqrt{G_x^2 + G_y^2}$$

$$\therefore \alpha(X, Y) = \tan^{-1} \left( \frac{G_y}{G_x} \right)$$

(단, Gx는 x축 편미분, Gy는 y축 편미분)

먼저 기울기를 알아낸 다음, 영상의 이미지는 좌우 상하로 픽셀이 그려지기 때문에, 현재 X축과 Y축의 값에 의해 방향성이 구해진다. 구해진 방향의 값을 가지고 이전 블럭의 위치 정보 값을 비교해 X가 0이면 수평으로 진행이 되고, Y가 0이면 수직으로 진행이 된다.

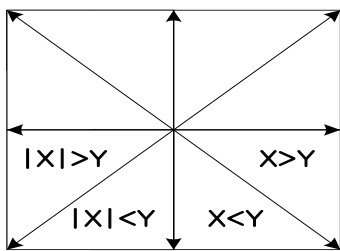


그림 1. 벡터의 방향성 검사(8 방향)

다음 기준에 존재하는 벡터의 방향성을 8 방향으로 비교하여, 각 방향에 가중치를 주어 블럭 크기를

계산한다.

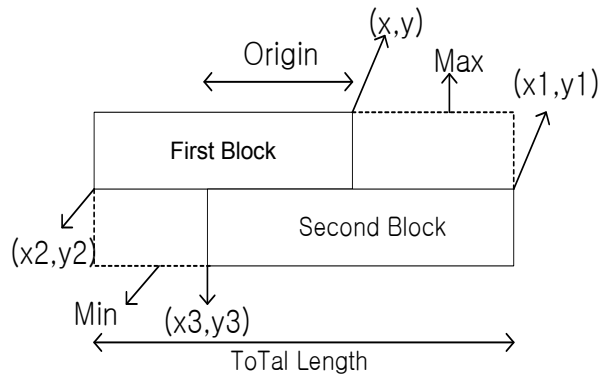


그림 2. 동적 블러킹 처리과정

구배 연산자를 이용한, Edge Detection을 통해 First 블럭의 시작점을 구하고, First 블럭의 끝점 (X,Y)과 , Second 블럭의 끝점(X1,Y1)을 가지고, 기울기를 알아내고, Max 값을 구하는데, 수식과 블럭도는 다음과 같다.

· Total Length = First 블럭의 시작 + Second 블럭의 끝점

· Max =  $\alpha(X, Y)$ 와 (X,Y), (X1,Y1)

· Min =  $\alpha(X, Y)$ 와 (X2,Y2), (X3,Y3)

· Offset(i) = Max;(Offset은 길이를 저장하는 임시 변수)

Min의 값은 3가지의 경우로 나눌 수가 있는데 Min의 기울기가 양수, 음수 그리고 0일 경우로 나눌 수가 있다.

· Min > 0 일 경우 - Second의 시작점은 First의 시작점보다 작아진다.

· Min = 0 일 경우 - Second의 시작점과 First의 시작점은 같아진다.

· Min < 0 일 경우 - Second의 시작점은 First의 시작점보다 커진다.

· Second Block 시작점 = Total Length - Min;

· Origin = Total Length - Max - Min;(Origin은 First 블럭과 Second 블럭의 공통부분)

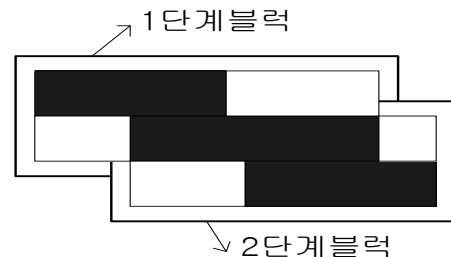


그림 3. 연산과정을 거친 블럭

이처럼 이미지를  $n \times n$ 개의 Pixel들로 이루어져 있는 여러 개의 블록으로 분할을 시켜 놓고 검색된 블록과 관련 정보들을 블록 테이블에 저장시켜 놓는다. 이때 잘못된 에지 검출로 인한 블록 제거도 필요하다. 만약 한 블록이 여러 개의 블록을 담고 있다면 블록의 패턴은 아마도 우리가 매칭 시키려고 했던 Vector 블록과 다른 값들이 들어가게 될 것이다. 동적 블록을 만들어 낸 영상 데이터에 블록 단위로 벡터 양자화를 거치는 과정을 통해 모자이크의 정보를 가진 블록 영상 정보가 되는데 여기서 얻은 정보를 가지고 처리를 하면 High-Quality가 일어난다. 이 때, 블록이 겹치지 않는 예외 조건을 주어야 블록 테이블에 있는 정보를 가지고 빠르게 블록 매칭하여, Aliasing된 Offset 영역에 Anti-Aliasing을 적용할 수 있다. 즉, 블록들을 이미지에 적용하여 동일한 영역을 덮어씌우고, 영역을 구분하기 위해서 Offset값을 이용하여 Vector를 매핑 시킨다.

#### 4. 실험 및 분석

##### 4.1 Gaussian 함수의 적용

인간의 눈은 Rasterizing된 라인의 큰 변화에 매우 민감하기 때문에 날카로운 에지의 부분에 약간의 Blurring 효과를 주는 것이 Anti-Aliasing의 매우 중요한 기능이다. 선택된 Rasterizing Cluster 블록 Offset에 Halfton 효과를 내기 위해 점진적인 침전 현상을 나타내는 방법을 적용한다.

확대된 영상을 가지고 선택된 블록의 Offset이 존재하는 블록을 미리 만들어진 블록 테이블과 매칭하면서  $n \times n$ 의 벡터 양자화를 하고 양자화된 영상 블록의 Offset에 가우시안 Mask를 적용시켜 High Quality의 블록으로 만들 수 있다.

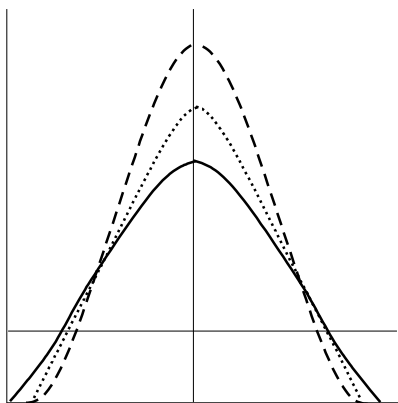


그림 4. 가우시안 분포부분

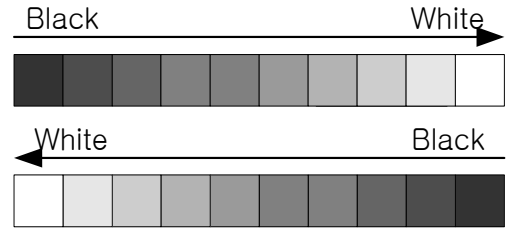


그림 5. Offset영역에 적용될 Halfton 영상 Offset의 크기를 가중치 값( $\sigma$ )으로 놓고 가우시안 분포 정도를 구한다.

가우시안 함수로는 LoG(Laplacian of Gaussian)을 자주 사용하는데 계산시간이 많이 걸리기 때문에 가우시안에 대한 차연산자인 DoG(difference of Gaussians)가 LoG의 근사값이므로 DoG에 Offset의 비율을 가중치로 주어 사용한다.

$$LoG(x,y) = \frac{1}{\pi \sigma^4} \left[ 1 - \frac{(x^2 + y^2)}{2\sigma^2} \right] e^{-\frac{(x^2 + y^2)}{2\sigma^2}}$$

$$DoG(x,y) = \frac{e^{-\frac{(x^2 + y^2)}{2\sigma_1^2}}}{2\sigma_1^2} - \frac{e^{-\frac{(x^2 + y^2)}{2\sigma_2^2}}}{2\sigma_2^2}$$

DoG를 사용하는 장점으로서는 가중치의 값에 변화를 줄 수 있기 때문에 Offset을 더 넓게나 좁게 조정할 수 있다.

위 식에서 얻어진 결과를 가지고 Offset 블록에 Halfton 효과를 단계적으로 주면 아래의 그림 6과 같은 현상이 일어난다.

Edge Detection방법으로 얻어진 정보를 가지고 계단과 같은 두 개의 수평바 길이와 Offset과 M의 길이를 가지고 Offset 영역에 가우시안을 적용하면 된다.

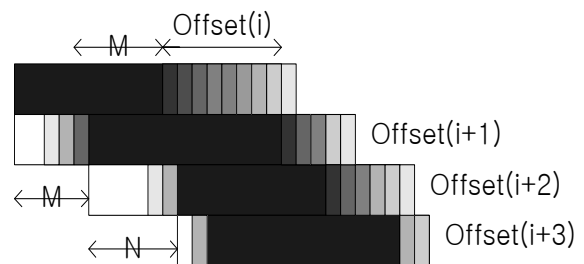


그림 6. Anti-Aliasing 연산 후 블록

마지막 단계로는 원래 영상으로의 손실을 줄이면서 축소를 해야 하는데 축소 비율은 원래 영상의 크기로 고정시키고 이웃 영상의 정보와 비교하여 확대했을 때 적용된 방법과 같은 방법으로 보간을 시켜준다. 이와 같은 과정을 거친 후 확대된 Text 영상 데이터의 결과를 보면 사선뿐만 아니라 직선 그리고 경계 에지 부분에 존재했던 이미지의 Aliasing의 효과들이 제거되고 부드럽게 Anti-Aliasing의 효과로 바뀐 영상의 데이터를 볼 수 있을 것이다.

#### 4.2 실험 결과 및 성능 분석

본 논문에서 제안한 동적 블록 연산 방법을 하기 위해, Text(영문, 한글)와 도형을 가지고 실험을 하였고, 그 결과는 다음과 같다.

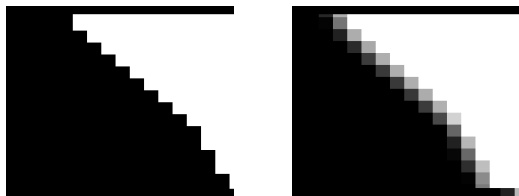


그림 8. Text 영상의 Aliasing 영상과 Anti-Aliasing된 영상

그림 7의 왼쪽 영상은 일반적인 사선에 적용되는 Rasterizing 즉, Aliasing 효과를 나타낸 것이고, 본 논문에서 제안된 알고리즘을 통해 Anti-Aliasing 효과가 나타난 것을 오른쪽 영상이다.



그림 8. 영문 알파벳에 적용된 Aliasing 영상

그림 8은 영문 “A”라는 대문자 내에 존재하는 Band 현상을 2배로 확대했을 때 보인 영상으로 Aliasing 된 영역을 볼 수 있다.

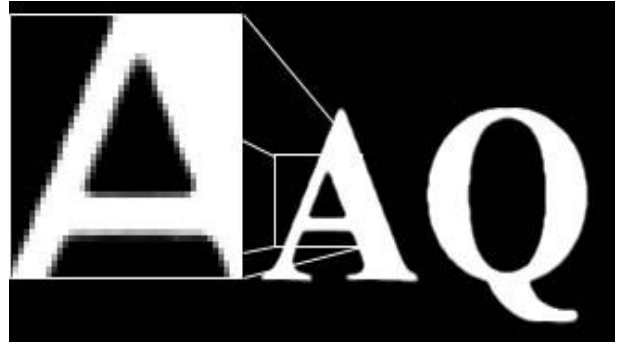


그림 9. 영문 알파벳에 Anti-Aliasing된 영상

그림 9는 그림 8과 같은 원래 영상에서 동적 블록 연산을 이용해 Anti-Aliasing 된 영상이다. 그림 8과 그림 9를 비교해보면 A라는 대문자 내부에 존재하는 Band 현상이 제거 된 것을 볼 수 있다.



그림 10. 한글에 적용된 Aliasing 영상

그림 10은 한글에서 Aliasing 된 궁서체 “카”의 어근인 “ㄱ”의 부분을 확대한 모습이다. 궁서체를 Sample로 선택한 이유는 예기치 못한 기울기가 가장 많이 존재하는 서체이기 때문이다.



그림 11. 한글에 Anti-Aliasing 된 영상

그림 11은 그림 10과 같은 영상을 가지고 본 논문에서 제안한 알고리즘을 적용한 영상인데 “ㄱ”의 Band 현상이 그림 10 보다는 Smoothing하게 Aliasing이 줄어 든 것을 시각적으로도 판별할 수 있다.

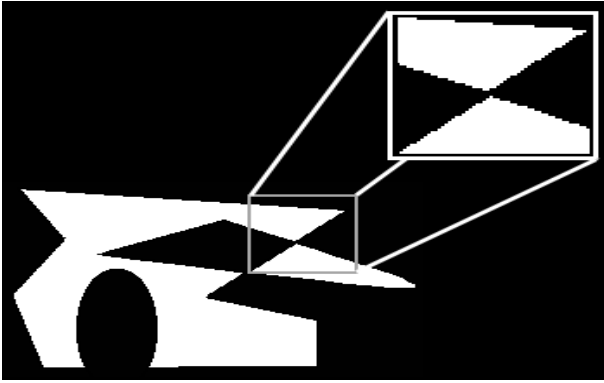


그림 12. 도형의 Aliasing 영상

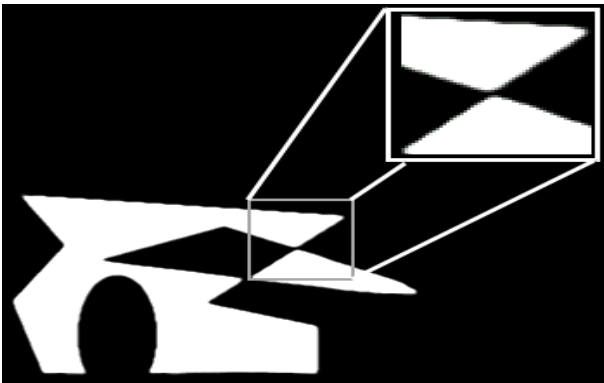


그림 13. 도형에 Anti-Aliasing 된 영상

Anti-Aliasing 효과는 일반적으로 Text에만 적용되는 것이 아니라, 여러 가지의 도형이나 이미지 영상에도 적용되는 것으로 그림 12와 그림 13의 도형을 가지고 비교해 보면 Rasterizing에서 손실된 Pixel을 보간 된다는 사실을 알 수 있다.

## 5. 결론 및 향후 연구과제

본 논문에서 제안된 동적 블록 연산방법이 컴퓨터 그래픽스 이미지에 적용되는 Aliasing 효과를 소프트웨어적으로 제거하는데 효율적인 방법이다.

제안된 알고리즘을 Text와 도형에 적용하여 실험한 결과, 동적 블록 연산 방법은 매우 단계적이고 적당한 방법임을 증명하였는데, 그 특징을 요약하면 다음과 같다.

- (1) 복잡한 계산성의 감소.
  - : 동적인 블록 단위로 처리를 하기 때문에 Anti-Aliasing 하기 위한 연산 시간의 감소.
- (2) 급하게 기울어진 사선 효과적 적용.
  - : Gradient, Compass 연산자를 이용해 기울어진 사선이나 수평, 수직, 직선 등을 정확하게 Anti-Aliasing 할 수 있는 정보를 추출한다.
- (3) Gaussian 함수의 적용.

: Offset 부분을 추출해내고 Offset 값을 가중치로 Gaussian 함수에 적용하여 Anti-Aliasing 효과를 주었다.

향후 연구과제로는 Anti-Aliasing 효과를 2D 그래픽 이미지에서 3D 그래픽 이미지로의 확대 적용하는 기법을 연구해야 할 것이다.

## 참고문헌

- [1] Yuan-Hau Yeh, Chen-Yi Lee, "A New Anti-Aliasing Algorithm for Computer Graphics Images." ICIP 1999, vol 2, pp.442-446.
- [2] Roger S. Pressman "Software Engineering A Practitiners' Approach" 3Rd Ed. Mcgra Whill1
- [3] Foley, Et. Al., Computer Graphics: Principles And Practice, 2Nd Edition, Addiaion-Wesley, Nov. 1993.
- [4] Catmull, E., "A Hidden-Surface Algorithm With Anti-Aliasing" SIGGRAPH 78, pp.6-11.
- [5] Capenter, L., "The A-buffer, an Antialiased Hidden Surface Method, "SIGGRAPH 84, pp.103-108.
- [6] David B. Kirk, "Unsolved Problems and Opportunities for High-quality, High-performance 3D Graphics on a PC Platform, "Eurographics/Siggraph Graphics Hardware Workshop, August 1988, pp.11-13.
- [7] P.H. Gregson, "Using angular dispersion of gradient direction for detection edge ribbons", IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 15, pp 682-696, Jul. 1993.