

컴퓨터 그래픽 이미지에서의 새로운 Anti-Aliasing 알고리즘 설계

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

Design of A New Anti-Aliasing Algorithm for Computer Graphics Images

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

요 약

자연에 존재하는 연속량의 데이터는 디지털화 될 때 숫자로 변환된다. 이 때 부딪치는 일반적인 문제는 완전히 통합되지 않았거나 잘못 통합된 표본 추출의 결과로 정보를 잃어 버리는 앨리어싱 (Aliasing) 현상이 일어나게 된다. 현재 일반 사용자가 사용하는 PC기반의 모니터 해상도는 96PPI(Pixel Per Inch)밖에 제공하지 못한다. 이러한 저해상도에서 컴퓨터 그래픽 이미지 객체를 디스플레이 할 때 앨리어싱 현상이 자주 발생한다. 현재 PC기반의 그래픽 하드웨어 구조의 변형없이 저해상도 모니터 상에서 앨리어싱 현상이 발생하는 이미지를 기존의 계산 방법을 감소시키면서 앨리어싱 현상을 제거 하는 기법이 필요하다. 본 연구에서는 이와 같이 앨리어싱 현상을 감소시켜 이미지 데이터의 품질을 향상시킬 수 있는 개선된 Anti-Aliasing 알고리즘을 제안한다.

1. 서론

현재 사용중인 일반 PC의 모니터에 객체를 디스플레이 할 때 객체의 해상도는 96PPI(Pixel Per Inch)이다. 그러나 사람의 눈은 훨씬 높은 해상도를 가지고 있기 때문에 96PPI의 샘플 해상도가 사람의 눈으로 보여질 때 해상도가 낮기 때문에 Aliasing이 일어나게 된다. Aliasing이란 데이터에 있는 정보를 캡처하기에 불충분한 해상도를 갖는 데이터를 Sampling 할 때 원래의 이미지에 대한 정보를 잃어 버리는 현상이다. 대표적인 현상으로는 계단(Band)을 들 수 있다. 해상도를 증가시키면 시킬수록 Aliasing 현상이 줄어든다. 예를 들면 영화 토이스토리를 만든 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 새로운 Anti-Aliasing 방법을 제안한다.

논문의 구성은 2장에서 Anti-Aliasing 방법을 알아보고 3장에서는 제안된 Anti-Aliasing 방법을 4장에서는 제안한 Anti-Aliasing 기법을 이용한 실험 내용 및 결과를 서술하고, 기존의 방법들과 비교분석 한다. 5장에서 논문의 결론과 향후 연구과제를 제시한다.

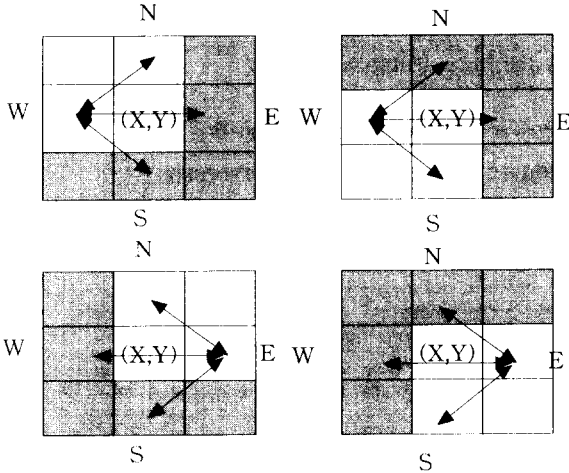
2. 기존의 Anti-Aliasing 방법

현재 그래픽 이미지에서 Aliasing 현상을 줄이는 대표적인 Anti-Aliasing의 방법으로 Prefiltering과 Postfiltering 방법이 있다. Prefiltering 방법으로는 그래픽 이미지의 인접한 픽셀 값들의 평균값을 이용하는 방법과 Mask Filtering을 이용하는 방법이 주로 사용된다. 그러나 이 방법은 간단하지만 Aliasing 효과를 많이 줄 일수 없다. 그래서 일반적으로 많이 사용하는 것이 Postfiltering의 방법(Supersampling)인데 이것은 샘플링의 해상도를 높이는 방법으로 H/W적인 해상도를 증가시켜 Aliasing 현상이 줄어들게 한다. 그러나 H/W적인 방법은 복잡하고 어려움이 따른다. 그래서 일반적으로 정상 해상도에서 정상 크기의 두배 이상 이미지의 표본을 만들고 이 이미지에 low-pass 필터를 적용하여 Aliasing을 일으킬 고 빈도 정보를 제거한다. 여기서 말하는 low-pass 필터란 이웃 픽셀의 대칭적 평균을 취함으로써 순화 또는 흐리게 하는 필터이다. 고밀도 정보가 제거된 확대 이미지의 크기를 조절하여 원래의 정상크기의 크기로 이미지로 변환 시킨다.

3. 제안된 Anti-Aliasing 알고리즘

본 논문에서 기존의 Anti-Aliasing의 방법보다 Aliasing 효과를 더 줄 일수 있는 새로운 Postfiltering의 알고리즘을 제안한다. 제안한 알고리즘을 살펴보면 다음과 같다.

- 그래픽 원 이미지에서 Anti-Aliasing 시킬 윤곽선을 추출한다.
- 윤곽선 안쪽은 Black으로 바깥쪽은 White로 한다.
- 원 이미지를 두 배 이상으로 확대한다.
- 확대된 이미지를 스캔 라인시 Find Edge Orientation 조사 아래의 조건 중 하나를 만족하는지 검사한다.(단, X,Y의 좌표값은 현재 Scan된 픽셀(white))



```

if(West==white) && (North==white) && (South==black) &&(East==black)
if(West==white) && (South==white) && (North==black) &&(East==black)
if(East==white) && (North==white) && (South==black) &&(West==black)
if(East==white) && (South==white) && (North==black) &&(West==black)
    
```

그림 1. 각 Find Edge Orientation 및 검색 조건식

- 위 조건식이 만족하면 West 또는 East 방향으로 그리고 North 또는 South 방향으로 각각 단계 계수를 구한다 단, 최대 단계는 5단계이다. 구해진 두 방향의 단계 계수가 높은 방향에 계산된 가중치 값(256/단계계수)을 단계 계수가 높은 방향으로 누적하여 적용한다.

예를 들어 첫 번째 조건을 만족한다면 각각 West, North 방향으로 West는 아래 픽셀의 값이 Black이 마지막이 될 때까지, North는 오른쪽 픽셀의 값이 Black의 마지막이 될 때까지 단계 계수를 검사 한다. 검사된 두 방향의 단계계수가 높은 곳의 방향에 계산된 가중치 값(256/(단계계수+2))을 단계 계수가 높은 방향으로 누적하여 적용한다. 적용되는 알고리즘과 그림은 다음과 같다.

```

int east,west,north,south,center;
int count,west_count, north_count;
int i,j,index,value;
    
```

```

if((center==255)&&(west==255)&&(north==255)&&
(south==0) && (east==0)){
west_count=0;
north_count=0;
    
```

```

for(i=1;i<5;i++) {
index=x-i;
if(index>=0)

if(m_opening[y+1][index]!=0)
west_count++;
}

for(i=1;i<5;i++) {
index=y-i;
if(index>=0)

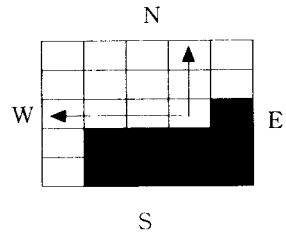
if(m_opening[index][x+1]!=0)
north_count++;
}

if(west_count >= north_count ) {
value=count=(int)(256.0/(lcount+2));

for(j=0;j<=west_count;j++) {
m_opening[y][x-j]=count;
count=count + value;
}
}
else {
value=count=(int)(256.0/(north_count+2));

for(j=0;j<=north_count;j++) {
m_opening[y-j][x]=count;
count=count + value;
}
}
}
    
```

알고리즘 1. 첫 번째 조건에 대한 알고리즘



(West 방향 Level수 > North 방향 Level수)

그림 2. 첫 번째 방향 검사의 예

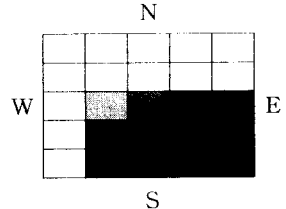


그림 3. 가중치 값을 적용한 예

- 이미지의 크기를 축소하여 정상의 이미지로 변환한다.

4. 실험 내용 및 결과

본 논문에서 제안한 기법을 Text와 Polygone 에 적용해 실험하였다. 그림 4, 5의 실험 결과 중 왼쪽은 원래의 이미지이고, 오른쪽의 이미지는 본 논문에서 제안한 Anti-Aliasing 알고리즘을 적용한 이미지를 보여주고 있다.

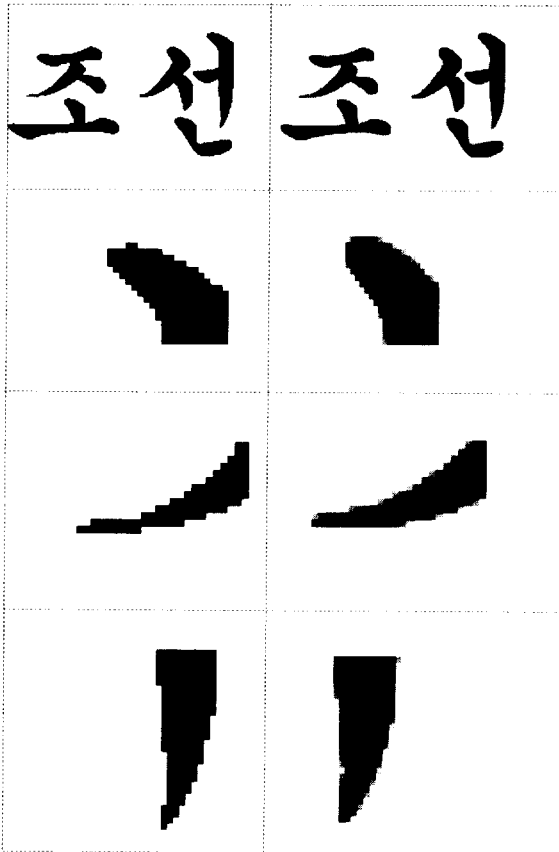


그림 4. Text 이미지를 Anti-Aliasing 한 예

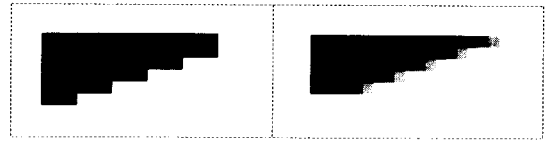
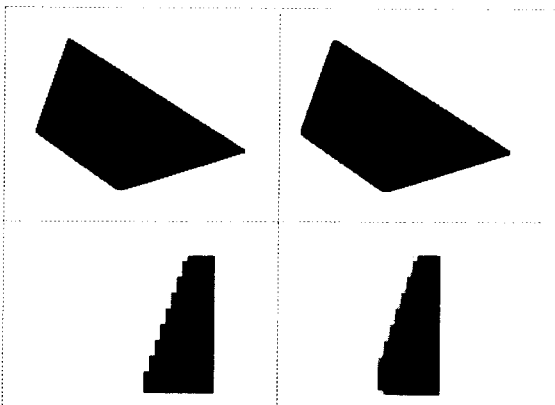


그림 5. Polygon 이미지를 Anti-Aliasing 한 예

5. 결론 및 향후 연구과제

본 논문에서 제안한 새로운 Anti-Aliasing 알고리즘은 Aliasing 현상이 일어나는 컴퓨터 그래픽 이미지에서 복잡한 기존의 계산 방법을 감소 시켰고, 급하게 기울어진 사선의 Aliasing 효과를 제거하는데 효과적이었다. 향후 연구과제로는 PC 기반에서 그래픽 이미지의 변형을 최소화하면서 High-quality, High-performance한 Anti-Aliasing 방법을 연구해야 할 것이다.

참고문헌

- [1] Roger S. Pressman "Software Engineering A Practitiners' Approach" 3Rd Ed. Mcgra Whilll
- [2] Foley, Et. Al., Computer Graphics: Principles And Practice, 2Nd Edition, Addiaion-Wesley, Nov. 1993.
- [3] Catmull, E., "A Hidden-Surface Algorithm With Anti-Aliasing" SIGGRAPH 78, pp.6-11.
- [4] Capenter, L., "The A-buffer, an Antialiased Hidden Surface Method, "SIGGRAPH 84, pp.103-108.
- [5] 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.
- [6] 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.