

## 화재 Simulation을 위한 Post-Processor 개발

Development of Post-Processor for Fire Simulation

○ 장재원<sup>1)</sup>, 허남건<sup>2)</sup>

J. Chang, N. Hur

When caught in a fire inside a building or a tunnel the generated smoke is the main cause of the bad visibility, which makes difficult for a person to find escape route. Therefore it is required to visualize the simulated results of smoke realistically from a viewpoint of a person caught in a fire. In the present study, demonstrated is a CFD post-processor which can visualize the objects through smoke from the results of CFD fire simulation.

### 1. 서론

화재란, 사람의 의도에 반하거나 고의에 의하여 발생하는 연소현상으로서 그 결과가 재해와 관련 되는 것을 말한다. 이런 화재에는 많은 물적 손실을 가져 올뿐만 아니라 많은 인명피해가 발생하게 된다. 특히 건물이나 터널 내부에서 화재가 발생할 경우 연소에 의해 발생한 연기는 안에 있던 사람의 시각 능력을 저하시키는 주원인으로 작용하여 대피 및 인명구조에 많은 어려움으로 작용한다.

그래서 과거 화재 시 건물이나 터널 내부의 연기 농도 분포를 알기 위해 많은 노력을 해왔다. 하지만 이제까지의 화재 Simulation 연구는 연기의 농도를 그래프형태로 나타내어 실제 화재 시 건물 내부에 있는 사람이 느끼는 시각 상태를 나타내는 데에는 한계가 있다.

본 연구에서는 건물이나 터널 내부에서의 화재 발생 시 연기의 농도와 사람의 시각능력과의 관계를 알아보고, 계산된 결과를 이용 내부의 연기 농도를 3차원 투명도를 사용하여 시각화하는 Post-Processor를 개발하였다. 또한 가상현실(Virtual Reality) 기술을 적용하여 시점의 변화에 연기 농도의 변화를 나타내었다.

프로그램 개발은 본 연구실에서 CFD Package 프로그램으로 개발중인 Turbo-3D에 기능을 삽입하는 방법으로 진행하였다.

### 2. Post-Processor

#### 2.1 개발 환경

1980년대 XT를 기반으로 DOS가 발표된 이후, 급속히 발달된 PC는 현재 1GHz대 CPU가 발표되어 보편화되었으며, GUI인 Windows도 보편화된 상태이다. 또한 PC는 지금도 계속 발달되고 있다. 보통 컴퓨터의 수명을 2년에서 3년 정도라고 생각 할 때 한 대에 수천만원하는 고가의 Workstation 보다는 비슷한 성능을 내는 저가형 PC가 더욱 효과적이다. 그래서 본 연구에서 개발한 프로그램은 OS Windows 2000 Professional Version, CPU Intel 600Hz, RAM: 128M에서 개발하였다.

Compiler는 Windows Programming의 가장 보편적으로 사용되는 Visual C 6.0과, MFC(Microsoft Foundation Class Library) 6.0을 사용하였다. 본 연구에서 개발된 프로그램의 3D 그래픽스 Library는 OpenGL v1.1을 사용하였다. 80년대 초반 그래픽스 기술의 필요에 의해 E&S와 Silicon Graphics에 의해 Graphics API로 개발이 시작된 OpenGL은 산업계에서 가장 광범위하게 사용되는 3D Graphics API로써 Unix상의 X Window 및 Windows NT, Windows 98등에서 사용이 가능하다. 또한 2D Graphic과는 달리 3차원 그래픽이 표준인 OpenGL은 RealTime Rendering, Shading, Texture Mapping등 다양한

1) 서강대학교 기계공학과 대학원 (121-742 서울시 마포구 신수동 1번지 Tel : 02-705-8637)

2) 서강대학교 기계공학과 (121-742 서울시 마포구 신수동 1번지 Tel : 02-705-8637)



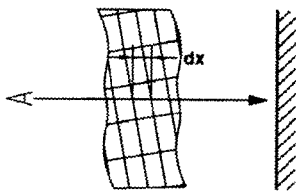
기능을 가지고 있으며 물체의 표현방법은 3차원 좌표 값을 기준으로 하고 있다.

### 2.2 Ray Trancing에 의한 연기 농도 적분

본 연구에서는 격자 내부의 연기 농도 분포를 3차원 투명으로 나타내었다. 각 격자내부의 연기 농도를  $c_i$ 라하고, 눈으로부터 물체 표면 또는 벽면에 위치한 격자의 중심을 연결하였을 때 나타나는 선을 통과하는 유체 격자의 개수를  $n$ , 이 선이 격자를 통과할 때 격자내부에 포함된 선의 길이를  $dx$ 로 하여 사람의 느끼는 연기의 농도의 강도를 Fig. 1과 다음 식으로 나타내었다.

$$S = \sum dx_i c_i \quad (1)$$

격자 내부를 통과하는 시선의 길이(dx)는 3차원 Cyrus Beck Algorithm을 사용하여 계산을 하였다. Cyrus Beck Algorithm은 한 꼭지점까지의 벡터를  $f$  라하고, 이 꼭지점에서부터 벡터  $P(t)$  위의 임의의 점까지의 벡터를  $P(t) - f$  라 하여 벡터  $P(t) - f$  와 한 면의 Inner Normal Vector( $n$ )의 내적 관계를 사용한 Algorithm이다.



Fluid Cell

Fig. 1 The Relationship between the Fluid Cell and the Line of Sight

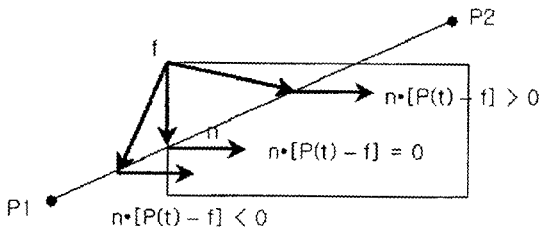


Fig. 2 Cyrus Beck Algorithm

이 두 벡터의 내적 관계는 Fig. 2와 같이 나타나며, 이 중 두 벡터의 내적이 Zero인 것을 이용하여 선과 면이 만나는 점을 계산하였다. 여기서 사용된 변수는 다음과 같다.

$n$  : 각 면의 Inner Normal Vector

$P(t)$  :  $P_1P_2$  Vector

### 2.3 연기의 농도와 시각과의 관계

화재가 발생한 건물이나 터널 안쪽에 있는 사람이 물건이나 벽을 바라보면 이색은 연기에 의해 뚜렷하게 보이지 않게 되는데, 이 정도를 가시도 함수(Visibility Function)  $f$ 로 정의하여 사용하였다. 여기서  $f$ 는 다음 식과 같이 Linear Function과 Exponential Function의 간단한 형태로 나타낼 수 있다.

Linear Function:

$$f = a \cdot S/Sr, \quad (S \leq Sr) \\ f = 1 \quad (S \geq Sr) \quad (2)$$

Exponential Function

$$f = 1 - \exp\left(-b \cdot \left(\frac{S}{Sr}\right)^d\right) \quad (3)$$

단, 여기서  $f$ 는 0과 1사이의 값을 가지며, 위의 식에서  $a, b, d$ 는 상수로써 빛의 강도나 연기의

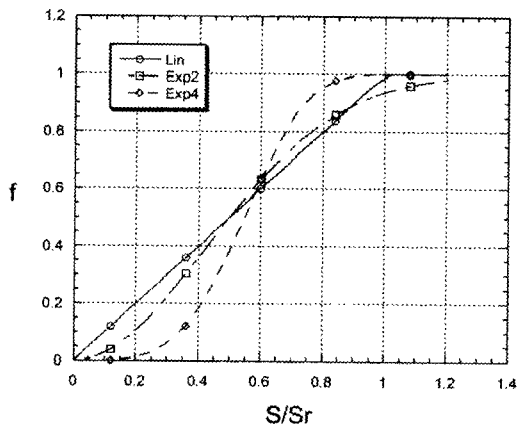


Fig. 3 Visibility Function



종류에 따라 결정하게된다. 본 연구에서는  $a = 0.6$ ,  $b = 1$ ,  $d = 2(\text{Exp}2)$  또는  $4(\text{Exp}4)$  을 사용하였다.  $f$  가 0일 때는 연기의 농도가 Zero로 모든 것이 뚜렷이 보이는 상태를 의미하며,  $f$  가 1이면 연기의 농도가 가장 강한 상태로 모든 것이 보이지 않게 되는 상태를 의미한다. Computer는 기본적으로 R(Red), G(Green), B(Blue) Color를 사용하여 모든 색을 나타내고 있다. 특히 OpenGL에서는 각 R, G, B에 0~1사이의 값을 사용하여 색을 나타내는데 예를 들어 Red는 R : 1, G : 0, B : 0, Green은 R : 0, G : 1, B : 0으로 나타낸다. 물건이나 벽면의 실제 색을  $C_r$ 라고 하고, 연기의 색을  $C_f$ 라고 할 때, 실제 내부에서 사람이 느끼는 색( $C_r'$ )은 다음 식과 같이 나타낸다.

$$C_r' = C_r + (C_f - C_r) f \quad (4)$$

이것을 각 R, G, B에 적용하면 다음 식과 같이 나타낼 수 있다.

$$\begin{aligned} R_r' &= R_r + (R_f - R_r) f \\ G_r' &= G_r + (G_f - G_r) f \\ B_r' &= B_r + (B_f - B_r) f \end{aligned} \quad (5)$$

위의 식들은 OpenGL Reference Manual의 glFog 함수를 참고하여 제작하였다.

### 2.4 프로그램 구성

본 프로그램의 화면 구성은 Fig. 4와 같이 아래쪽 명령창(Command window)과 위쪽의 표시창(Display window)로 구성하였다. 먼저 명령창은 command를 입력받기 위한 창으로 MFC의 CEditView Class를 계승받아 사용하였으며, 표시창은 기본적으로 CView Class를 계승받았으며 OpenGL를 사용 할 수 있도록 수정하였다. 또한 표시창의 경우 3차원 입체감을 주기 위해 Perspective Viewing을 사용하였으며, Fig. 5는 Perspective Viewing을 사용하여 터널 형태를 내부에서 본 그림이다. 모든 변수의 저장을 위해 CDocument Class 계승받아 사용하였다.

다른 프로그램과의 호환성을 위해 cell data을 읽어 오는 cread와 vertex data을 읽어 오는 vread, 계산결과 중 scale data를 읽어 오는 fread 명령을 제작하였다. 화면에 표시하기 위해서는 먼저 표시할 격자의 종류를 선택해야 한다. cplot 명령을 사용하여 선택된 격자를 화면에 표시하게 되는데, 이때 먼저 vertex data를 검색하여 각 좌표축의 최대, 최소 값을 계산하여 model의 크기를 결정한다. 다음에는 눈의 위치를 설정하고 실제 화면에 표시하게 된다. 또한 가상현실(Virtual Reality) 개념을 도입하여 키보드의 화살표 키를 이용하여 눈의 위치를 변화시킬 수 있도록 제작하였으며, 변화된 위치에서 사람이 느끼는 연기의 농도를 다시 계산하여 표시하게 하였다.

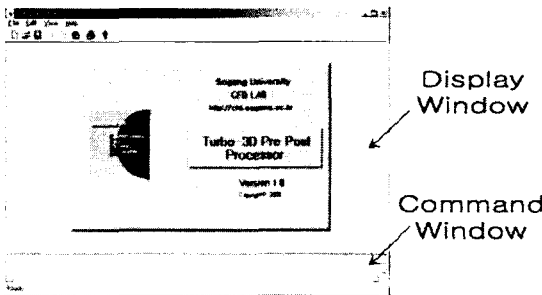


Fig. 4 Turbo-3D Pre-Post Processor

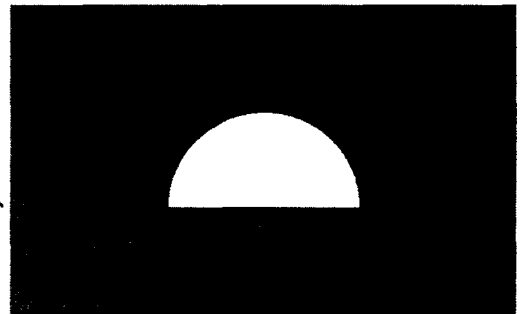


Fig. 5 Perspective Viewing

### 3. 결과 및 고찰

#### 3.1 적용예 : 등농도분포

먼저 프로그램을 Test하기 위해 가장 간단한 형태인 정육면체에 대해 내부의 연기 농도가 0.5로 일정한 경우에 적용해 보았다. 격자의 개수는 Fig. 6과 같이 각 축으로 10개씩 1000개이다. Test한 결과를 Fig. 7에 나타내었다. Fig. 7(a)는 연기의 농도가 Zero일 때이다. 이것은 다른 것과 비교하기 위해 제시하였으며, 면의 색은 R : 0, G : 0, B : 0 (검은색)이고 선의 색은 R : 1, G : 1, B : 1 (백색)로 연기의 색은 R : 1, G : 1, B : 1 (백색)로 하였다. Fig. 7(b)~Fig. 7(d)는 등농도분포일 때를 나타내었다. 이 경우에는 Fig. 7(a)와 다르게 선은 그리지 않았다. 3개의 그림 모두 농도가 일정하기 때문에 거리가 증가함에 따라 S의 값이 증가하여 연기의 영향이 증가하는 것을 잘 나타내고 있다. 또한 Fig. 3과 비교하여 먼저  $f$ 가 Linear한 Fig. 7(b)는 깊이변화에 따라 검은색 농도가 거의 일정하게 감소하는 것을 볼 수 있으며, Fig. 7(c)와 Fig. 7(d)는  $f$ 가 Exponential Function일 때이며, 깊어짐에 따라 검은색 농도가 Fig. 7(b)와 같이 감소하는 것을 볼 수 있다. Fig. 7(b)와 다른 점은 Exponential 함수를 사용하였기 때문에 눈에 가까운 물체는 Fig. 7(b)에 비해 더 선명하게 나타나며 멀어지면 더욱 흐려진다. 특히  $Exp4(d=4)$ 인 경우에는 그 차이가  $Exp2(d=2)$ 보다 더욱 심하다. 검은색의 농도 변화가 벽과 벽이 만나는 끝 쪽 모서리부분에서 급격히 변화는 것을 볼 수 있는데, 이것은 Fig. 3에서  $Exp4$ 의 기울기가 가장 급하기 때문에 나타난 현상이다.

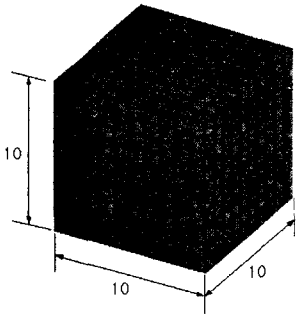
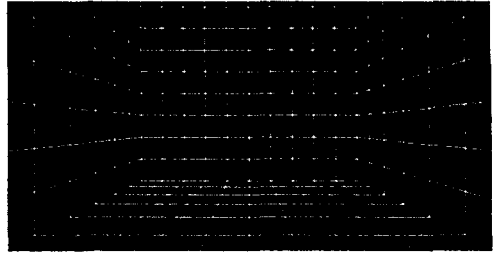
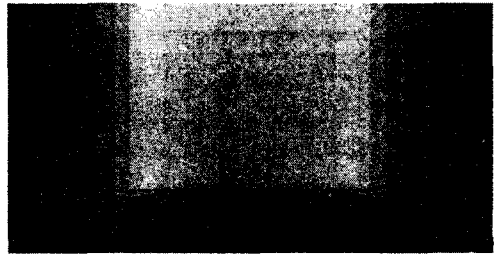


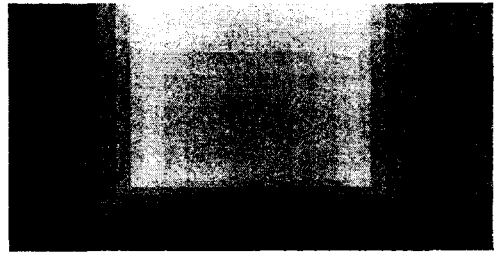
Fig. 6 Mesh for The Test Cell



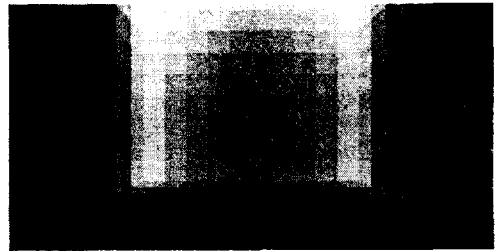
(a) Default View



(b) Linear Function( C = 0.5 )



(c) Exponential Function  
( d = 2, C = 0.5 )



(d) Exponential Function  
( d = 4, C = 0.5 )

Fig. 7 View for different visibility function



### 3.2 적용예 : 화재 해석결과

다음으로는 화재 해석결과에 적용해 보았다. Fig. 6과 같은 격자에 Fig. 8과 같이 아래쪽에 8개의 Fire Cell(1MW)을 설치하였다. 계산은 Star-CD를 사용하여 계산을 수행하였으며, Transient로 계산하여 시간이 40초일 때까지 계산하였다. 내부는 최초 공기로 채워져 있다고 가정하였다. 계산결과는 Fig. 9에 나타내었다. 화재가 발생한 중앙아래부터 연기가 위쪽으로 올라가면서 연기 기둥을 나타내는 것을 볼 수 있으며, 전체적으로 아래쪽보다는 위쪽의 연기 농도가 진하게 나타났다. Fig. 10과 Fig. 11은 Test한 결과이다. Fig. 10은 아래쪽에 모서리에서 본 그림이며 Fig. 11은 위쪽 모서리에서 본 그림이다. 그림들의 중앙에 연기기둥이 나타나는 것을 볼 수 있으며, Fig. 10의 경우 Exp4가 연기 기둥을 가장 잘 표현하고 있다. 반면에 Fig. 11에서는 Linear한 경우가 잘 표현하고 있다. 또한 위쪽에서 본 그림들이 아래쪽에서 본 그림들 보다 흐리게 나타나는 것은 Fig. 9에서와 같이 연기가 위쪽에 모이기 때문이다. 또한 Fig. 11(c)의 경우 중앙이 거의 흰색이 되는 것은 Exp4가 Fig. 3에서와 같이 일정수준 이후부터는  $f$ 의 값이 1로 수렴하기 때문이다.

### 4. 결론

본 연구는 건물이나 터널 내부에 화재 발생 시 내부의 연기 농도를 안에 있는 사람이 어떻게 느끼는지 알아보고자 3차원 투명도를 사용한 Post-Processor를 개발하였다. 3가지 Blending Factor를 정의하여 테스트해 보았다. 테스트한 결

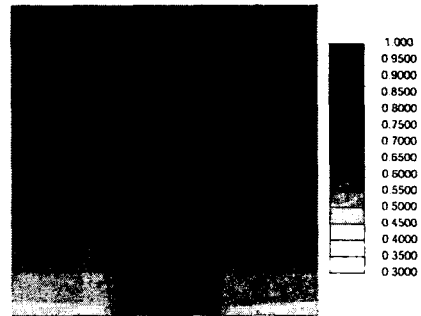


Fig. 9 Results showing the Smoke Concentration at section through fire cell



(a) Linear Function



(b) Exponential Function ( d = 2 )



(c) Exponential Function ( d = 4 )

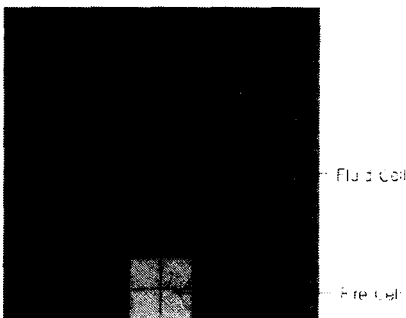
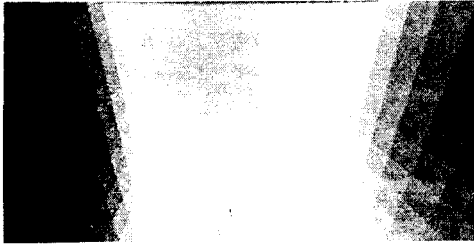
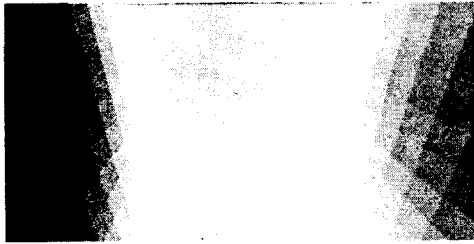


Fig. 8 Fire cell at the center of the domain

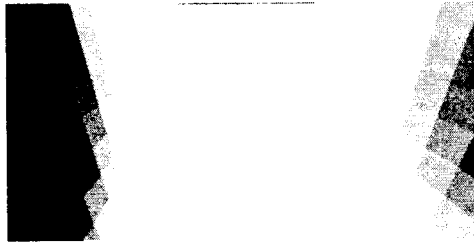
Fig. 10 Upward view for different visibility function



(a) Linear Function



(b) Exponential Function ( d = 2 )



(c) Exponential Function ( d = 4 )

Fig. 11 Downward view for different visibility function

과 Fig. 3 그래프를 잘 표현하고 있으며, 또한 계산 결과의 연기기둥 또한 잘 표현하고 있다.

제시한 visibility Function의 경우 Fig. 3에서 보는 것과 같이 S가 어느 이상의 값이 되면 f가 1이 되며, 이때 Linear한 경우에는 기울기가 급격히 변하기 때문에 사람의 시각과는 차이가 있다. 또한 사람의 시각은 가까운 곳에 있는 물체 더 뚜렷이 볼 수 있다는 점에서 S자 형태의 그래프를 나타내고 있는 Exponential Function을 사용하는 것이 물리적으로 더 타당성이 있다.

향후 빛의 영향과 연기 종류에 따른 가시도를 나타내는 계수의 결정이 필요하며, Cell의 개수가 증가 시 발행하는 Cell의 Search시간을 단축하기 위해 Search Algorithm의 개발 이 필요하다.

### 참고 문헌

- [1] David, F., "Procedural Elements for Computer Graphics," McGRAW-HILL (1988).
- [2] Cyrus, M., and Beck, J., "Generalized Two- and Three - Dimensional Clipping," Computers & Graphics, Vol. 3, (1978) p.22-28.
- [4] 권오승 외2인 "장대터널의 화재시물레이션 및 피난기술개발에 관한 연구," (주)대양설비기술단 (1999).
- [5] 허남건, "경량전철 시스템 안전프로그램 계획을 위한 화재 방재 대책수립", 경량전철시스템 기술개발사업 2차년도 연구결과보고서, 한국철도기술연구원(2000).
- [6] 이건우, "컴퓨터그래픽과 CAD," 영지문화사 (1990).