

# DEM을 이용한 실영상기반 가상표적의 폐색처리기법

차 정 희\* · 장 효 종\*\* · 김 계 영\*\*\*

## 요 약

실 세계 영상에 가상표적을 효과적으로 전시하여 현실감을 높이기 위하여 먼저 두 세계를 정합한 후 폐색영역을 산출하여 가상객체의 위치를 결정하는 것이 필수적이다. 본 논문에서는 실 영상위에 지정된 경로에 따라 가상표적을 이동시킬 때 발생하는 폐색문제를 해결하는 새로운 방법을 제안한다. 이를 위해 먼저 실험 영역의 DEM을 이용하여 3차원 가상세계를 생성하고 이를 CCD 카메라 영상과 시각적 단서를 이용하여 정합한다. 또한 스네이크 알고리즘과 픽킹 알고리즘을 이용하여 영상에서 폐색 처리될 지점의 3차원 정보를 산출하고 표적이동시 이를 이용하여 폐색문제를 해결하는 방법을 제안하였다. 실험에서는 부분적 폐색이 발생하는 환경에서 제안한 방법의 유효성을 입증하였다.

키워드 : DEM, 가상표적, 스네이크, 픽킹, 폐색

## Resolving Occlusion Technique of Virtual Target on Real Image using DEM

Jeong-Hee Cha<sup>†</sup> · Hyo-Jong Jang<sup>\*\*</sup> · Gye-Young Kim<sup>\*\*\*</sup>

### ABSTRACT

For virtual target to be displaying on real image realistically, it is essential to determine the location of the virtual object together with producing the occlusions area after registering two world. In this paper, we propose the new method to solve occlusions which happens during virtual target moves according to the simulated route on real image. For this purpose, we first construct three dimensional virtual world by DEM of experimental area and register CCD camera image on it by visual clues. Next, we also propose a method to solve the occlusion using snake and picking algorithm which can extract the three dimensional information of the position happening occlusion in the image and can use it when target moves that area. In the experiment, we proved the effectiveness of the proposed method in the environment which a partial occlusions happens.

Key Words : DEM, Virtual Target, Snake, Picking, Occlusion.

### 1. 서 론

증강현실(Augmented Reality)이란 실세계 영상에 3D 가상정보들을 실시간으로 정합하여 관찰자로 하여금 실세계에 대한 현실감을 증대시키고 이해를 돕는 가상현실의 한 분야로 현재 군사, 의료, 교육, 건설, 게임, 방송 등 다양한 분야에서 연구와 활용이 이루어지고 있다. 증강현실은 광학(optical) 또는 영상(video) 기반 시스템으로 디자인 될 수 있는데 광학기반 시스템은 사용자가 HMD(Head-Mounted Display)의 반투명 스크린에 가상의 물체를 투영하여 사용자에게 정보를 제공하는 것이고 영상 기반 시스템은 카메라로부터 영상을 얻은 다음 이 영상에 가상의 정보를 합성하

는 것이다. 증강현실은 일반적으로 영상 정합, 상호작용,

실시간의 3가지 특징을 가지고 있다[1]. 영상정합은 사용자가 보는 실세계 대상 물체와 관련된 가상객체의 위치가 일치하는 것을 말한다. 또한 상호작용이란 가상 객체와 실 영상 내 객체들의 병합이 주위 환경과 어우러져서 현실감 있게 전시되어야 한다는 것으로 관찰자의 위치나 시선의 변화에 따라 폐색 영역을 결정하거나 충돌을 감지하는 등 관찰자의 위치를 추적하여 가상객체를 다시 렌더링 하는 것을 말한다. 마지막으로 실시간은 영상정합과 상호작용이 실시간으로 이루어지는 것을 말한다. J. P. Mellor는 원 모양의 특징점들을 추적하여 카메라의 투영행렬을 실시간으로 얻어 내고 실제 물체 영상과 3차원 그래픽으로 모델링된 가상물체를 정합하여 전시하는 연구를 수행하였고[2], G. Simon, M-O Berger 등은 영상에서 점과 곡선 등의 2차원 특징점들을 추적하면서 통계학적인 방법을 응용하여 각 2차원 특징점들을 3차원 모델의 특징점들과 정합시키고 두 대의 카메라

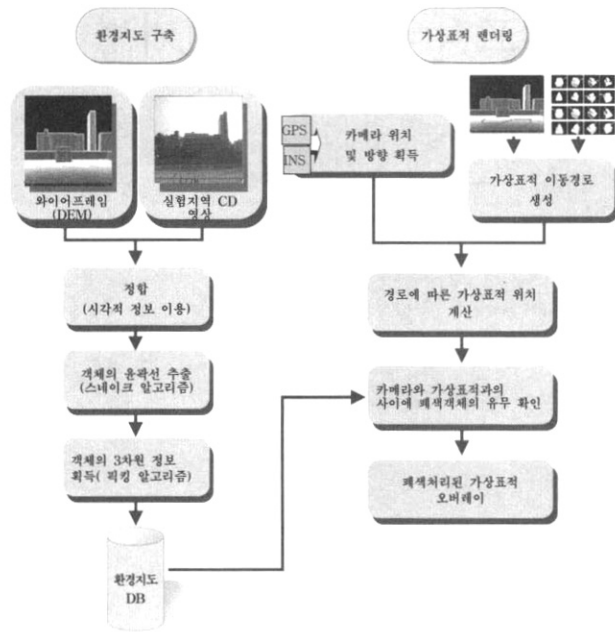
\* 이 논문은 2006년 정부(교육인적자원부)의 재원으로 한국학술진흥재단의 지원을 받아 수행된 연구(KRF-2006-005-J03801)임.

† 준 회원 : 숭실대학교 정보미디어기술연구소 전문연구원

\*\* 준 회원 : 숭실대학교 컴퓨터학과 박사과정

\*\*\* 종신회원 : 숭실대학교 컴퓨터학과 교수

논문접수 : 2006년 9월 6일, 심사완료 : 2006년 10월 24일



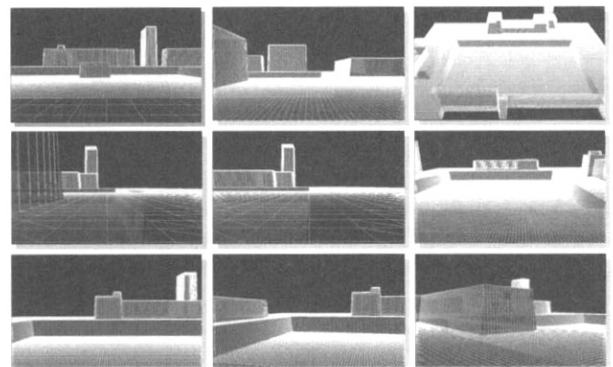
(그림 1) 전체 시스템 개요도

라를 사용하여 폐색문제를 해결하였다.[3] D. Koller와 G. Klinker등은 검은 사각형을 특징점으로 사용하여 카메라를 보정하고 특징점을 추적하여 그 결과를 이용하였으며[4], U. Neumann등은 퍼지 이론과 다중해상도 접근법을 증강현실에 도입하였다[5]. 그러나 실영상과 가상객체의 현실감 있는 결합은 실제 환경 안에서 가상객체가 물리적으로 그럴듯한 방식으로 행동하는 것을 요구하는데 관측자에 더 가까운 물체가 시선에 더 멀리 있는 가상객체를 가리거나 객체에 의해 실영상의 물체가 덮여지는 폐색문제를 해결하기 위해서는 두 세계간의 정합이 더욱 정확하게 이루어진 후 실제 장면의 깊이에 대하여 가상객체의 깊이를 비교하는 작업이 필요하다. 또한 구축된 맵의 정확성이나 밀도가 폐색 경계선 추정에 충분하지 않은 경우에는 폐색 영역을 결정하기 어렵다. 따라서 본 논문에서는 정합문제를 해결하기 위해 실험 영역의 DEM을 이용하여 3차원 와이어프레임을 생성하고 이를 CCD 카메라 영상과 시각적 단서를 이용하여 정합하는 방법을 제안한다. 또한 맵의 밀도에 관계없이 경계선을 정확히 추정하여 폐색문제를 해결하기 위해 스네이크(Snake) 알고리즘과 픽킹(Picking) 알고리즘을 이용하여 폐색지점의 3차원 정보를 산출하고 이를 가상표적의 이동시 사용하였다. 본 논문의 구성은 다음과 같다. 2절에서는 DEM을 이용하여 와이어프레임을 생성하고 이를 2차원 영상과 정합하는 과정에 대해 설명하고 3절에서는 스네이크 알고리즘과 픽킹 알고리즘을 이용한 객체 윤곽선 추출과 3차원 정보획득과정에 대해 설명한다. 4절에서는 가상 표적의 이동경로 생성과 최소포함사각형을 이용한 폐색 영역후보 추출에 대해서 5절에서는 실험결과에 대해 설명하고 6절에서는 결론을 맺는다.

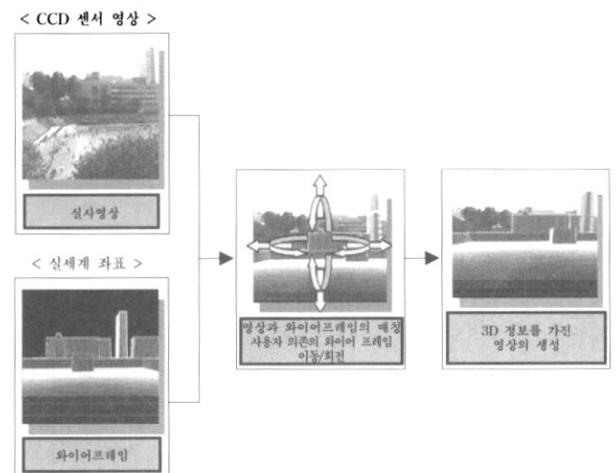
## 2. CCD 카메라 영상과 DEM의 정합을 통한 3차원 정보 획득

2차원 CCD 영상과 실제계 좌표를 매핑하기 위해 지형정보인 DEM(Digital Elevation Model)을 이용한다. DEM은 일정한 간격마다 X, Y 좌표로 표현되는 위,경도와 높이에 대한 정보를 가지고 있는 지형의 형상을 나타내는 자료로 일정크기의 격자로 저장되는 격자방식의 DEM, 높이가 같은 지점을 연속적으로 연결하여 만든 등고선 방식, 단층에 의한 프로파일 방식 등이 있다. 실험에 사용된 DEM은 실험지역을 300M×300M로 제한하여 1M 간격마다 2차원 좌표에 대한 고도정보를 갖도록 제작한 격자방식의 DEM이다. DEM의 데이터를 읽어 이를 각 4각형의 정점으로 메쉬를 생성하여 3차원의 깊이 정보를 가진 와이어 프레임(그림 2)와 같이 생성하였다[6][7].

메쉬(mesh)구조의 와이어프레임(wireframe)을 센서영상 위에 오버레이(overlay)시켰을 때 정합여부를 확인할 수 있도록 (그림 3)과 같이 영상의 상, 하, 좌, 우로 이동하여 오차를 줄이고 정합한다.



(그림 2) DEM을 이용한 와이어프레임 생성



(그림 3) 3차원 정보획득을 위한 영상과 와이어프레임의 정합

이렇게 시각적인 단서를 이용하여 정합한 때의 카메라의 초기위치를 기준으로 하고 차량 이동으로 생긴 위치 변화는 GPS(Global Positioning System:위성항법장치)를 이용하여 와 이어프레이밍 내에서 차량의 위치를 수시로 갱신하도록 하였다.

### 3. 스네이크 알고리즘과 픽킹 알고리즘을 이용한 객체 윤곽선 추출과 3차원 정보획득

논문에서는 영상 객체의 윤곽선을 얻어내기 위해 물체의 에너지 최소점을 탐색하는 스네이크 알고리즘[8][9]을 이용하였다.

#### 3.1. 스네이크를 이용한 객체의 윤곽선 추출

스네이크 알고리즘은 사용자가 입력한 스네이크 포인트로부터 에너지함수를 반복적으로 최소화하는 방향으로 움직이면서 객체의 윤곽을 찾아내는 방법으로 에너지 함수는 식(1)과 같다. 계산되는 에너지 함수는 이산공간상에서 이루어지기 때문에 각 에너지 함수의 매개변수는 영상 내에서의 각 정점의 좌표 값이 된다. 여기서  $v(s)$ 는 스네이크 포인트를 의미하며  $v(s)=(x(s),y(s))$ 로  $x(s)$ 와  $y(s)$ 는 해당 스네이크 포인트의 이미지상의  $x$ 와  $y$ 의 위치값을 나타낸다. 또한  $\alpha, \beta, \gamma$  는 가중치로 본 논문에서는  $\alpha=1.0, \beta=0.4, \gamma=2.0$ 의 값을 취하였다.

$$E_{snake} = \int_0^1 (\alpha E_{cont}(v(s)) + \beta E_{curve}(v(s)) + \gamma E_{\epsilon}(v(s))) ds \quad (1)$$

첫 번째 항은 폐색 영역을 둘러싸고 있는 스네이크 정점들의 연속성을 나타내는 에너지 함수이고 이 항을 이용하여 각 스네이크 정점간의 간격을 등 간격으로 조정한다[10]. 수식은 다음과 같다.

$$E_{cont,i} = |v_s(s)|^2 = \left| \frac{d(v(s))}{ds} \right|^2 \approx |d - |v_i - v_{i-1}||^2 \quad (2)$$

두 번째 항은 스네이크가 이루는 곡선의 완만성을 제어하는 에너지 함수로 곡률이 큰 부분에서 값이 커지므로 코너

점의 검출이 가능하다. 코너로 인식되는 부분에서는  $\beta$ 값을 0으로 설정하여 코너점에서의  $E_{curve}$  값이 전체 에너지 함수에 영향을 끼치지 않도록 한다.

$$E_{curve,i} = |v_{ss}(s)|^2 = \left| \frac{d^2(v(s))}{ds^2} \right|^2 = |v_{i-1} - 2v_i + v_{i+1}|^2 \quad (3)$$

마지막으로 식(4)의  $E_{image}$ 는 영상 특징에 대한 함수로 Max는 스테이크 정점 주변의 화소 들 중 가장 큰 밝기 값의 기울기 강도를 나타내며 Min은 최소 밝기 값의 기울기 강도를 그리고  $\nabla I(v(s))$ 는 이미지 밝기 값의 기울기 강도를 나타낸다.

$$E_{image} = \frac{(Max - \nabla I(v(s)))}{(Max - Min)} \quad (4)$$

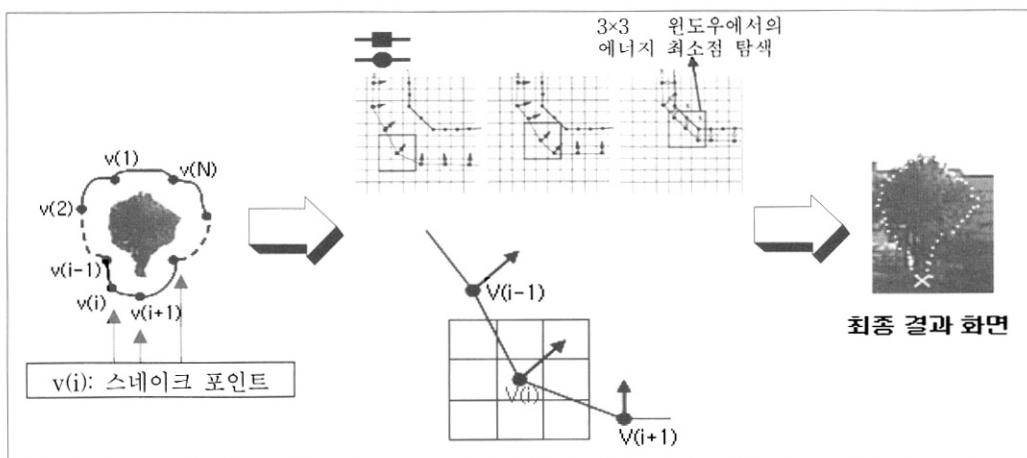
에너지 함수들은 모두 1에서 0의 값을 지니도록 정규화 된다.

<표 1>은 논문에서 사용한 스네이크 알고리즘의 수행과정을 정리한 표이다.

<표 1> 스네이크 탐색 알고리즘

```

Index Arithmetic is modulo n.
do
// i = 스네이크 포인트 (1 = first point, n = last point)
for i=1 to n
    E_min = BIG
    m_i = 3 by 3 // m is size of neighborhood
    for j=1 to m_i
        E_snake(v_j) = alpha E_cont(v_j) + beta E_curve(v_j) + gamma(v_j)
        if E_snake(v_j) < E_min then
            E_min = E_snake(v_j)
            j_min = j
    Move point v_j to location of j_min
until (탐색 완료)
    
```



(그림 4) 스네이크 탐색의 예

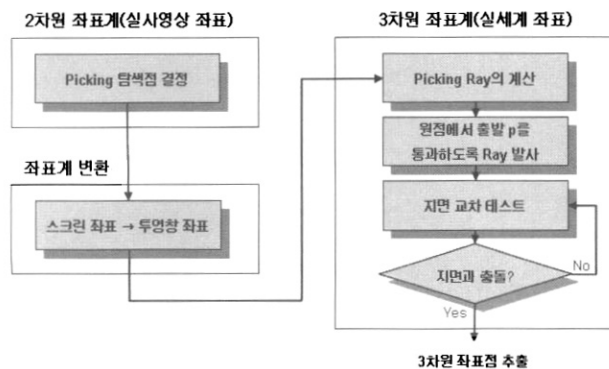
사용된 알고리즘은 (그림 4)와 같이 각 정점  $v(i)$ 에 3화소×3화소의 윈도우를 설정하고 윈도우내에서 이전 정점과 다음 정점 들 간의 연속성, 곡률, 에지 강도 등을 고려하여 에너지가 가장 최소화되는 위치를 찾아 정점을 이동시키는 에너지 최소화 알고리즘을 탐색이 완료될 때까지 반복 수행함으로써 윤곽선을 추출해 간다.

3.2. 픽킹 알고리즘을 이용한 3차원 정보 획득

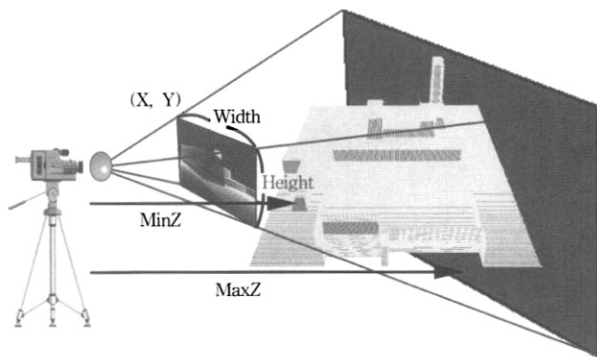
추출한 정점의 3차원 정보를 얻기 위해 본 논문에서는 3차원 그래픽스 기법 중에 잘 알려진 픽킹(picking) 기법을 사용하였는데 이것은 2차원 영상의 해당지점과 일치하는 DEM으로 생성된 3차원 와이어프레임과의 충돌점을 찾아내어 그 지점의 3차원 정보를 획득할 수 있도록 해준다. (그림 5)은 논문에서 제안한 픽킹 알고리즘을 이용한 3차원정보 획득 과정의 단계별 순서도이다.

3.2.1 스크린에서 뷰포트로의 변환

픽킹 탐색점은 2차원 영상에서 추출한 객체 정점의 최하 단점이다. 3차원 영상의 렌더링(Rendering) 과정에서 투영 변환한 도형을 실제로 표시하는 직사각형의 영역인 스크린 좌표계는 실제로 마우스가 존재하는 좌표계로 픽킹을 행하기 위해서 이것을 3차원의 실제 지형이 존재하는 뷰포트 좌표계로 변화시켜주어야 할 필요가 있다. (그림 6)은 뷰포트의 개념도이다.



(그림 5) 픽킹 알고리즘의 단계별 순서도



(그림 6) 뷰포트 개념도

먼저 뷰포트를 스크린으로 변화시켜주는 변환 행렬인 식 (5)를 사용하여 먼저 2차원의 스크린에서 3차원의 투영창으로 변환하는 식을 얻어내도록 한다.

$$\begin{bmatrix} \frac{Width}{2} & 0 & 0 & 0 \\ 0 & -\frac{Height}{2} & 0 & 0 \\ 0 & 0 & MaxZ - MinZ & 0 \\ X + \frac{Width}{2} & Y + \frac{Height}{2} & MinZ & 1 \end{bmatrix} \quad (5)$$

여기서 MinZ 와 MaxZ는 각각 버퍼값의 최소값과 최대값으로 0에서 1사이로 정의하였고 X, Y, Width, Height 는 각각 뷰포트를 포함하는 윈도우와의 상대적인 뷰포트 사각형을 정의하였다.

식(5)를 이용하면 스크린상의 한 점  $s=(s_x, s_y)$ 와 대응하는 뷰포트상의 한 점  $p=(p_x, p_y, p_z)$  간의 관계는 식(6)과 같이 된다.

$$(s_x \ s_y \ s_z \ 1) = (p_x \ p_y \ p_z \ 1) \begin{bmatrix} \frac{Width}{2} & 0 & 0 & 0 \\ 0 & -\frac{Height}{2} & 0 & 0 \\ 0 & 0 & MaxZ - MinZ & 0 \\ x + \frac{Width}{2} & y + \frac{Height}{2} & MinZ & 1 \end{bmatrix} \quad (6)$$

위 식을 이용하여 스크린의 x, y 좌표인  $s_x$  와  $s_y$  를 각각 얻을 수 있다.

$$s_x = p_x \left( \frac{Width}{2} \right) + X + \frac{Width}{2} \quad (7)$$

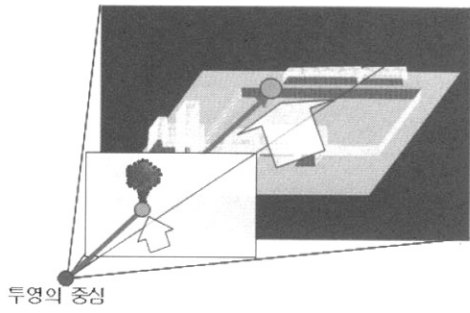
$$s_y = -p_y \left( \frac{Height}{2} \right) + Y + \frac{Height}{2} \quad (8)$$

$s_x$  와  $s_y$  는 알고 있는 값이고 얻고자 하는 값은  $p_x$  ,  $p_y$  ,  $p_z$  이므로 이것에 관한 식 즉 투영창의 관계식으로 변환을 하면 식(9)와 식(10)와 같다.

$$p_x = \frac{2s_x - 2X - Width}{Width} \quad (9)$$

$$p_y = \frac{-2s_y + 2Y + Height}{Height} \quad (10)$$

투영창은 3차원이므로 깊이 값(Z)이 필요하나 깊이가 버퍼가 가까울 수록 1에 멀 수록 0에 가까운 좌표계에서는 투영창이 평면 Z=1 과 일치한다는 점을 이용하여  $p_z = 1$  으로 정의할 수 있다. 이제 구하고자 하는 점과 지표면과의 충돌점을 구해야 하므로 투영창에서 지표면을 향해 점차 길이를 늘여가는 광선(Ray)을 이용한다. 광선의 일반식은 다음과 같다.



(그림 7) 광선과 DEM의 충돌점 픽킹의 예

$$p(t) = p_0 + tu \Rightarrow p_0 = (0, 0, 0) \quad (11)$$

위에서 u는 방향벡터로 식(12)와 같다.

$$u = p - p_0 = (p_x, p_y, 0) - (0, 0, 0) = p \quad (12)$$

이제 식(11)에서 t 값을 점차적으로 늘려 지면과의 충돌점을 찾아내면(t: 실험에선 0.01씩 증가) 그 위치가 구하고자 하는 2차원 좌표와 대응되는 3차원 좌표이다. (그림 7)에서 화살표가 가리키는 폐색영역의 가장 하단 위치가 구하고자 하는 기준점이고 이 점이 실질적으로 2차원 영상의 3차원 공간에서의 위치 값이 된다.

### 3.2.2 3차원 정보 생성

충돌점인 기준점은 실제 DEM상의 3차원 좌표를 지닌 한

점이나 객체 윤곽선으로 표시된 스테이크의 다른 정점은 충돌점이 없어 3차원 좌표를 얻을 수 없다. 따라서 본 논문에서는 기준점을 이용하여 2차원 영상과 3차원 DEM 사이에 하나의 비례관계를 구하고 이를 이용하여 다른 정점의 3차원 좌표를 얻어낸다.

(그림 8)은 2차원과 3차원의 두 정점간의 비례관계를 보여주는 그림이다.

앞에서 구한 기준점 좌표를 이용하여 (그림 8)에서와 같이 화면의 중심을 지나는  $p_m$ 을 구하기 위해 먼저  $t'$ 을 구한다.

픽킹 광선에 의해 t 값이 주어지므로 주어진 t 값과  $y_B$ 를 이용하여 식(13)과 같이  $\theta_B$ 를 구하고 이  $\theta_B$ 를 이용해 식(14)와 같이  $t'$ 을 구한다.

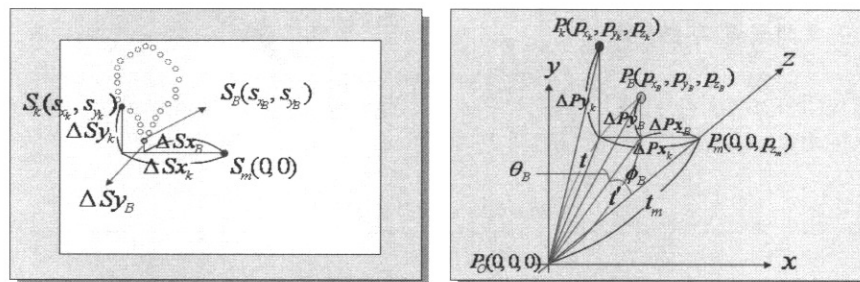
먼저 주어진 t 값과  $y_B$ 를 이용하여  $\theta_B$ 를 구하면

$$\theta_B = \sin^{-1}\left(\frac{\Delta P y_B}{t}\right) \quad (13)$$

이고 이 식을 이용하여  $t'$ 의 식을 구하면

$$t' = |t_B| \cos(\theta_B) \quad \text{단, 여기서 } t' = |t'| \quad (14)$$

$t_m$ 을 구하기 위해 식(15)와 같이  $t'$ 과  $t_m$ 의 사이각을 먼저 구한다.



(그림 8) 2차원과 3차원의 두 정점간의 비례관계

- $S_m$ : 스테이크 포인트의 최하단 포인트(기준 포인트)
- $S_B$ : 스테이크 포인트의 최하단 포인트(기준 포인트)
- $\Delta S_B$ :  $S_m$ 에서  $S_B$ 까지의 변화량 ( $\Delta S_B = (\Delta S_{x_B}, \Delta S_{y_B})$ )
- $S_k$ : 스테이크 포인트에서의 기준포인트를 제외한 임의의 포인트
- $\Delta S_k$ :  $S_m$ 에서  $S_k$ 까지의 변화량 ( $\Delta S_k = (\Delta S_{x_k}, \Delta S_{y_k})$ )
- $P_m$ :  $P_B$ 를 화면의 중심을 지나는 3D 공간상의 직선으로 투영시킨 한 포인트 ( $P_{x_m} = 0, P_{y_m} = 0$ )
- $P_B$ :  $S_B$ 의 3D 대응점
- $\Delta P_B$ :  $P_m$ 에서  $P_B$ 까지의 변화량 ( $\Delta P_B = (\Delta P_{x_B}, \Delta P_{y_B}, \Delta P_{z_B})$ )
- $P_k$ : 스테이크 포인트에서의 기준포인트를 제외한 임의의 포인트
- $\Delta P_k$ :  $P_m$ 에서  $P_k$ 까지의 변화량 ( $\Delta P_k = (\Delta P_{x_k}, \Delta P_{y_k}, \Delta P_{z_k})$ )
- $t$ :  $\overrightarrow{P_C P_B}$ ,  $t_m$ :  $\overrightarrow{P_C P_m}$
- $t'$ : t를 xz평면으로 투영시킨 한 벡터
- $\theta_B$ :  $\angle t t'$
- $\phi_B$ :  $\angle t' t_m$

$$\phi_B = \tan^{-1}\left(\frac{\Delta Px_B}{t'}\right) \quad (15)$$

구해진  $\phi_B$ 를 이용해 구하고자 하는  $t_m$ 을 구하면 식(16)과 같다.

$$t' = |t_m| \cos(\phi_B) \quad |t_m| = \frac{|t'|}{\cos(\phi_B)} \quad \text{단, 여기서 } t_m = |t_m| \quad (16)$$

$t_m = p_{z_m}$ 이므로  $P_m = (0, 0, t_m)$ 으로 정의할 수 있다.

이제 (그림 8)의 2차원 스크린 뷰 와 3차원 공간 좌표간의 관계를 식으로 나타낼 수 있는데 이것으로 2차원 스테이크 정점의 대응점인  $p_k$ 를 구하도록 한다.

$$\Delta S_B : \Delta P_B = \Delta S_k : \Delta P_k \quad (17)$$

$$\Delta Sx_B : \Delta Px_B = \Delta Sx_k : \Delta Px_k \quad \Delta Px_k = \frac{\Delta Px_B \times \Delta Sx_k}{\Delta Sx_B} \quad (18)$$

$$\Delta Sy_B : \Delta Py_B = \Delta Sy_k : \Delta Py_k \quad \Delta Py_k = \frac{\Delta Py_B \times \Delta Sy_k}{\Delta Sy_B} \quad (19)$$

그러므로 구하고자 하는 각 스테이크 정점의 3차원 공간 대응점인  $\Delta P_k = (\Delta Px_k, \Delta Py_k)$ 를 얻을 수 있다.

#### 4. 가상표적의 이동경로 생성과 최소포함사각형(MER)을 이용한 폐색객체 후보선택

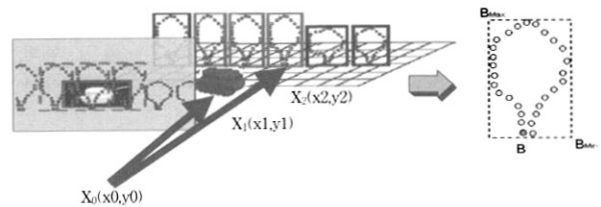
제안한 폐색처리 알고리즘을 실험하기 위해 논문에서는 가상표적의 이동경로를 생성하였는데 <표 2>는 가상표적의 이

동경로를 생성하고 이동에 따른 표적의 방향 변화와 모양 변화, 표적의 3차원 위치를 설정하는 속도(pseudo) 코드이다.

알고리즘을 살펴보면 먼저 교관이 설정한 표적의 시작점과 끝점을 저장하고 이 경로가 이루는 각도를 계산하여 각도 변화에 따라 표적의 방향과 모양을 갱신한다. 그리고 표적의 이동변화에 따라 해당하는 3차원 좌표를 제안한 알고리즘에 의해 결정한다.

실제영상과 가상표적의 현실감 있는 결합은 실제 환경 안에서 두 객체가 물리적으로 그럴듯한 방식으로 보여지는 것을 요구하는데 영상내의 객체에 의해 가상표적이 가려지거나 가상표적에 의해 영상 내 객체가 가려지는 것을 자연스럽게 실행시키는 것을 말한다. 그러나 폐색처리를 위해 학습과정에서 추출된 영상 내 모든 객체와 가상표적의 비교 연산을 하게 되면 수행시간이 많이 걸리는 문제가 있으므로 시간단축을 위해 본 논문에서는 카메라의 위치와 방향은 GPS와 INS를 이용하여 얻은 후 카메라 시야각내에 존재하는 객체의 최소포함사각형(MER: Minimum Enclosing Rectangle) 영역정보와 가상표적의 최소포함사각형을 비교하여 폐색객체 후보를 찾고 이를 처리하는 방법을 제안하였다.

객체의 최소포함사각형이란 (그림 9)의 오른쪽 그림과 같이 객체를 포함할 수 있는 최소사각형을 의미하는데 카메라



(그림 9) 최소포함사각형과 거리를 이용한 폐색객체 결정 가상 표적사이의 폐색영역

(표 2) 가상표적의 이동경로생성 알고리즘

```

// 경로 생성
for (int i=1; i<pDoc->m_nTankPosCount; i++)
{
    fPath.x = TankPos[i-1].x; // 경로의 Start Point 와 Dest Point 를 저장
    fPath.z = TankPos[i-1].y;
    fPath2.x = TankPos[i].x;
    fPath2.z = TankPos[i].y;

    fAngle = aTan(fPath2.x - fPath.x, fPath2.z - fPath.z); // 두 Point 가 이루는 각을 계산
    TankDir((fAngle+90) / 22.5); // 22.5도 씩 가상표적의 방향을 결정. 전방위를 향해야 하므로 +90

    // Start Point에서 Dest Point로 이동
    do {
        fDist = // 남은 이동거리 계산
        fPath.x += 이동속도 * Sin(fAngle); // 이동위치 저장
        fPath.z += 이동속도 * Cos(fAngle);

        X = fPath.x;
        Z = fPath.z;

        fPath.y = 해당 지표면의 높이;

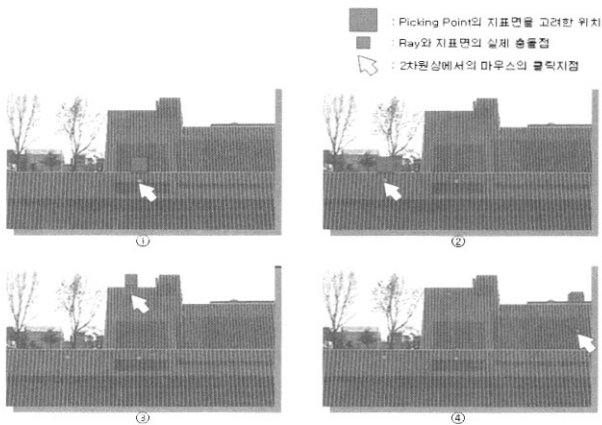
        Move(X, fPath.x); // 표적의 와이어프레임 상에서의 위치를 Set
        Move(Y, fPath.y);
        Move(Z, fPath.z);

    } while(이동할 거리가 남아있는동안);
}
    
```

영상내의 객체들과 가상표적의 최소포합사각형을 비교하여 겹쳐지는 영역이 있는 객체를 결정한다. 또한 결정된 객체와 가상표적은 카메라를 기준으로 어느 정도 일직선상에 놓이는 점을 이용하여 객체와 가상표적까지의 거리를 각각 구해 이 값을 이용하여 가상표적과 카메라사이에 객체가 존재하는지 없는지를 판단하였다.

5. 실험결과

(그림 10)은 마우스를 이용해 2차원 영상의 여러 위치에 서 제안한 픽킹 알고리즘을 실험한 것이다. 뷰포트의 중심



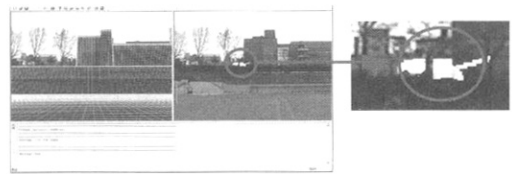
(그림 10) Ray와 DEM의 충돌점



(a) 61 Frame



(b) 102 Frame



(c) 155 Frame



(d) 301 Frame

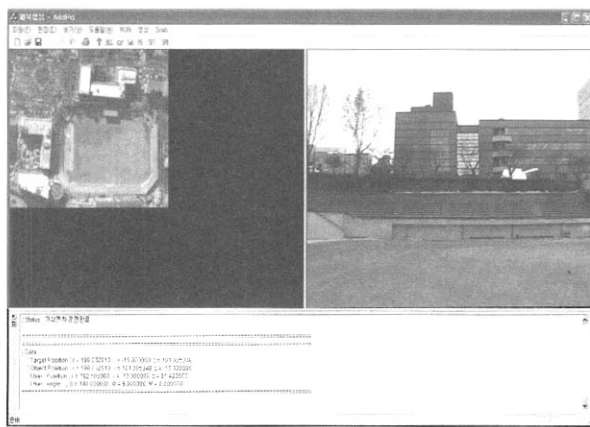
[그림 12] 가상표적의 이동 및 폐색 실험 결과

에서 마우스의 클릭 포인트를 향해 광선을 늘어 해당 지점의 높이가 DEM의 높이보다 낮아질 경우 충돌점으로 간주하여 픽킹점으로 표시하였는데 그림에서는 지표면을 고려한 위치에 물체를 사각형으로 표시하였다.

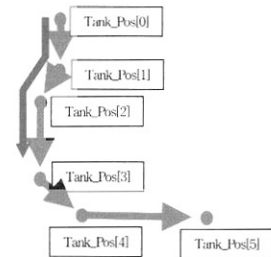
(그림 11)의 실험결과는 설정된 이동경로(b)를 따라 가상표적이 폐색처리 되면서 움직이는 모습(a)이다. 또한 (c)는 표적이 이동함에 따라 변화된 모습을 영상위에 전시하기 위해 생성한 다양한 가상표적의 모습이다.

(그림 12)는 경로를 따라 이동하는 가상표적의 모습을 프레임별로 나타낸 것이다. 프레임이 증가하면서 전차와 객체사이에 폐색처리가 되는 것을 볼 수 있다.

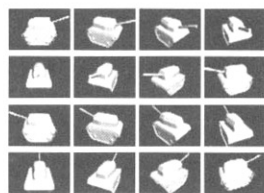
<표 3>은 (그림 12)의 실험에서 폐색처리를 위해 가상표적과 비교할 영상 내 객체를 선택하기 위해 추출된 스네이크 정점들을 사용한 경우와 제안한 최소포합사각형을 이용할 경우의 속도를 비교한 것이다. 제안한 방법의 경우 처리속도가 1.671배 정도 감소하여 성능향상에 기여하였다.



(a) 가상표적의 이동경로 설정 및 실험결과



(b) 설정된 가상표적의 이동경로



(c) 22.5도로 구분된 표적의 이동 방향에 따른 모습

(그림 11) 가상표적의 경로 생성 및 폐색처리

<표 3> 연산 처리속도의 비교

방법	프레임 수	비교한 객체 수	처리시간(초)	초당 프레임 수
스네이크 정점들 이용	301	10	112	2.687
최소포합사각형 이용	301	10	67	4.492

## 6. 결론 및 향후 연구방향

사용 언어는 Microsoft사의 Visual C++6.0을 사용하였으며 DEM은 송실대학교 운동장을 기준으로 300M×300M의 크기로 작성하여 사용하였다. 본 논문에서는 실 영상위에 지정된 경로에 따라 가상표적을 이동시킬 때 발생하는 폐색 문제를 효율적으로 해결하기 위하여 DEM을 이용하여 3차원 가상세계를 생성하고 이를 카메라 영상과 시각적 단서를 이용하여 정합하였다. 또한 스테이크 알고리즘과 픽킹 알고리즘을 이용하여 원래의 형상에 가까운 객체를 추출하여 폐색 처리될 지점의 3차원 정보를 산출하였다. 또한 폐색 처리 시 속도향상을 위해 객체의 최소포함사각형(MER) 영역의 3차원 정보를 사용하는 방법을 제안하였으며 실험을 통해 제안한 방법의 유효성을 입증하였다. 그러나 연구를 통해 연산 처리속도의 개선 및 폐색 처리에 의한 현실감이 증가하였으나 실험에 사용한 간이실험장치의 불안정으로 카메라의 정확한 위치와 방향을 설정하는데 어려움이 있었으며 실영상으로 사용된 2차원 CCD 영상위에 3차원 가상표적을 전시하였으므로 입체감의 부조화가 존재하였다. 따라서 향후에는 보다 현실감 있는 영상을 얻기 위해 와이어프레임에 2차원 실영상을 텍스처로 사용하여 입체적이고 현실감있는 가상현실(VR: Virtual Reality)을 구축하고 여기에 가상표적을 전시하여 보다 정확하면서도 실감도가 떨어지지 않는 훈련 영상을 구현해야 하겠다. 또한 조명에 강건하고 보다 빠른 객체 추출방법을 연구해야 하며 충돌에 따른 표적의 상호작용을 추가하는 것도 해결해야 할 과제이다.

## 참고 문헌

[1] R. Azuma, A Survey of Augmented Reality, Teleoperators and Virtual Environments, Vol. 6, No. 4, pp. 355-385, August, 1997.

[2] J. P. Mellor, Enhanced Reality Visualization in a Surgical Environment, AI Technical Report No. 1544, MIT, January 1995.

[3] G. Simon and MO Berger, A Two-Stage Robust Statistical Method for Temporal Registration from Features of Various Type, Proc. International Conference on Computer Vision, Bombay, India, pp. 261-266, 1998.

[4] D. Koller, G. Klinker, E. Rose, DE Breen, RT Whitaker, and M. Tuceryan., Real-time VisionBased Camera Tracking for Augmented Reality Applications. In ACM Symposium on Virtual Reality Software and Technology, pages 87-94.

[5] J. Yong Noh and U. Neumann. Expression cloning. In SIGGRAPH'01, pages 277-288, 2001.

[6] Stefan Growe, Peter Schulze, Ralf TÖnjes, 3D Visualization and Evaluation of Remote Sensing Data, Computer Graphics International '98 Hannover, Germany, June 22-26, 1998.

[7] E. Chen, Quicktime VR—an image-based approach to virtual environment navigation, Proc. of SIGGRAPH, 1995.

[8] N.E. Davison, H. Eviatar, R.L. Somorjai, Snakes simplified, Pattern Recognition 33, pp.1651-1664 (2000)

[9] Lilian Ji, Hong Yan, Attractable snakes based on the greedy algorithm for contour extraction, Pattern Recognition 35, pp.791-806 (2002)

[10] Chun Leung Lam, Shiu Yin Yuen, An unbiased active contour algorithm for object tracking, Pattern Recognition 19, pp.491-498 (1998)

## 차 정 희



e-mail : pelly@vision.ssu.ac.kr

1983년 2월 송실대학교 전자계산학과(공학사)  
1983~1998 신용보증기금 전산실 근무  
2001년 8월 동국대학교 컴퓨터공학과(공학석사)  
2005년 8월 송실대학교 컴퓨터학과(공학박사)  
2006년 6월~현재 송실대학교 정보미디어

기술연구소 전문연구원

관심분야: 컴퓨터(로봇)비전, 영상처리, 생체인식, 증강현실, 3D 모델링 등

## 장 호 중



e-mail : ozjhj@vision.ssu.ac.kr

2001년 송실대학교 컴퓨터학부(학사)  
2003년 송실대학교 대학원 컴퓨터학과(석사)  
2003년~현재 송실대학교 컴퓨터학과  
박사과정

관심분야: 음성처리, 컴퓨터비전, 영상처리, 패턴인식, 영상 검색등

## 김 계 영



e-mail : gykim@computing.soongsil.ac.kr

1990년 송실대학교 전자계산학과(학사)  
1992년 송실대학교 대학원 컴퓨터학과(석사)  
1996년 송실대학교 대학원 컴퓨터학과(박사)  
1996년~1997년 한국전자통신연구원(Post Doc.)

1997년~2001년 한국전력공사 전력연구원(선임연구원)

2001년~현재 송실대학교 컴퓨터학과 교수

관심분야: 컴퓨터비전, 패턴인식, 생체인식, 증강현실, 영상 및 신호처리 등