

## 영상 감시를 위한 지도기반 감시영역 입력 방법

김재혁<sup>1\*</sup>, 맹승렬<sup>1</sup>  
<sup>1</sup>공주대학교 컴퓨터공학과

### A Map-Based Boundray Input Method for Video Surveillance

Jae-Hyeok Kim<sup>1\*</sup> and Seung-Ryol Maeng<sup>1</sup>

<sup>1</sup>Department of Computer Science and Engineering, Kongju National University

**요약** 본 논문에서는 영상감시 시스템을 위한 감시영역 입력방법을 제안한다. 이동물체의 침입 여부는 감시영역과 비교를 통해 이루어지므로 이를 입력하는 방법은 영상감시의 가장 기본적인 기능이다. 기존의 방법은 각 카메라 별로 획득된 영상을 기반으로 감시영역을 설정하기 때문에 감시영역의 크기, 카메라 수, 카메라 위치 등 감시환경의 변화에 대응하기 어려운 단점이 있다. 제안하는 입력방법은 위성지도를 기반으로 감시영역 전체를 다각형으로 정의하고, 이를 각 카메라 환경으로 변환하는 접근방법을 사용한다. 이 방법의 특징은 감시환경에 독립적이라는 점이다. 다각형의 정점의 수를  $n$ 이라 하면 침입탐지 수행시간은  $O(n)$ 의 복잡도를 보인다. 제안하는 방법의 타당성을 검증하기 위해 3D 시뮬레이션을 수행하였으며, 각 카메라 수와 위치에 무관하게 입력된 감시영역을 재사용할 수 있음을 확인하였다.

**Abstract** In this paper, we propose a boundary input method for video surveillance systems. Since intrusion of a moving object is decided by comparison of its position and the surveillance boundary, the boundary input method is a basic function in video surveillance. Previous methods are difficult to adapt to the change of surveillance environments such as the size of surveillance area, the number of cameras, and the position of cameras because those build up the surveillance boundary using the captured image in the center of each camera. In our approach, the whole surveillance boundary is once defined in the form of polygon based on the satellite map and transformed into each camera environment. Its characteristics is that the boundary input is independent from the surveillance environment. Given the position of a moving object, the time complexity of its intrusion detection shows  $O(n)$ , where  $n$  is the number of polygon vertices. To verify our method, we implemented a 3D simulation and assured that the input boundary can be reused in each camera without any redefinition.

**Key Words** : Computational Geometry, Coordinates Transformation, Satellite map, Video Surveillance

### 1. 개요

최근 영상감시는 국방, 안전, 보안 등 다양한 분야에 적용되고 있으며, 성능 면에서도 빠르게 발전하여 상황인식이 가능한 시스템이 보급되고 있다. 특히 입산금지, 침입탐지 등 특정지역의 경계를 침범하는 물체를 감시하는 시설보안 분야에서 영상감시 수요가 꾸준히 증가하고 있다.

영상감시 처리과정은 크게 타설 단계, 비디오 영상촬영, 감시영역 입력, 이동물체 탐지, 이동물체와 감시영역의 비교, 판단결과 출력으로 이루어진다. 이 중에서 감시영역은 침입여부를 판단하는 기준이 되므로 모든 카메라에서 정의되어야 한다. 그런데 카메라 수나 위치 등 카메라 환경은 쉽게 변경될 수 있기 때문에 환경에 독립적인 입력방법이 요구된다.

본 논문은 2012년도 미래창조과학부(구 교육과학기술부)의 재원으로 과학벨트기능지구 지원사업의 지원을 받아 수행된 연구임 (과제번호 2012K001564).

\*Corresponding Author : Jae-Hyeok Kim(Kongju National Univ.)

Tel: +82-10-9142-6950 email: widedev@kongju.ac.kr

Received November 13, 2013 Revised December 16, 2013 Accepted January 9, 2014

감시영역의 입력방법에 대해서는 많은 연구가 이루어 졌다고 할 수 없으나, 주로 한 대의 카메라를 사용하는 환경에서 연구되어 왔다. J.A.Freer는 카메라 영상과 같은 크기의 비트맵 이미지를 위협도에 따라 구분하고 가중치를 주어 단계별로 감시 수준을 달리하는 방법을 제안하였다[1]. 이 방법은 감시영상과 감시영역 데이터 간에 1:1 매핑이 가능하다는 장점이 있지만 카메라나 배경에 변동이 생길 경우 영역 데이터 전체를 수정해야 한다는 단점이 있다. 원종운의 연구와 오세찬의 연구에서는 철도 승강장 내의 안전선 영역을 다각형으로 정의하고 영역 내에서 사람 또는 열차의 움직임이 감지되는 경우를 판별한다[2][3]. 그 외에도 Axis나 I2V System 사 등이 공급하는 영상감시 제품은 감시영역을 카메라 영상 위에 표시하는 방법을 사용하며 세부적인 방법은 각각 특허로 출원되어 있다[4][5][6].

본 논문에서는 한 번의 감시영역 입력으로 카메라 수와 위치 등 카메라 환경의 변화에 독립적인 감시영역 입력방법을 제안한다. 제안하는 방법의 기본 아이디어는 기하도형은 쉽게 임의의 좌표계로 변환할 수 있다는 점을 이용하여 전체 감시영역을 입력하고 필요할 때마다 새로운 카메라 환경으로 변환하는 것이다. 이렇게 함으로서 기존 방법의 단점을 개선할 수 있다는 장점이 있다.

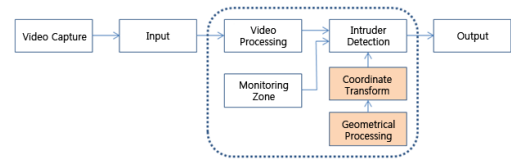
감시영역은 지도 위에 해당 영역의 경계를 다각형으로 편집함으로써 간단히 얻어진다. 여기서 지도의 좌표와 배율에 대한 정보는 감시 시스템에게 제공되어야 한다. 입력된 감시영역을 특정 카메라 환경으로 변환하는 작업의 핵심은 감시영역 다각형을 카메라 관점으로 변환하는 변환행렬을 계산하는 것이다. 변환행렬을 계산하기 위해서는 카메라 좌표계와 초점거리가 필요하며 본 논문에서는 보안시스템 설계 시 주어지는 것으로 가정한다. 다만 영상으로부터 카메라 파라미터를 찾는 문제는 오랜 연구과제이고 많은 연구결과가 발표되었으므로 이를 이용할 수 있다.

2장에서는 본 논문의 결과를 적용할 감시시스템 구성을 설명하고, 3장에서는 감시영역 다각형의 변환 행렬을 유도하며, 제 4장에서는 구현 알고리즘과 실험결과를 제시한다.

## 2. 영상감시시스템 구조와 다각형 입력

본 연구에서 제안하는 영상감시 시스템은 카메라, 통신모듈, 비디오서버, 감시 S/W로 구성되며, 영상감시 과정은 Fig. 1과 같이 영상촬영, 영상전송, 저장 및 영상처리, 침입탐지, 그리고 출력으로 이루어진다. 이는 일반적

인 영상감시 시스템과 크게 다르지 않다. 여기서 점선 부분은 침입탐지를 처리하는 S/W 부분으로 시스템마다 다르며 성능과 편의성을 좌우하는 중요한 요소이다.



[Fig. 1] Intrusion Detection Process in Video Surveillance System

영상감시의 최초의 시작점은 카메라이므로 카메라 선택은 전체 시스템의 성능에 중요한 역할을 한다. 영상감시의 목적과 비용에 따라 적절한 해상도와 성능의 카메라를 선택하는데, 최근 주밍과 방향전환이 가능한 PTZ 카메라가 보급되고 있으나 영상 감시 시스템에서는 보통 고정식 카메라가 사용된다. 이는 영상감시의 경우 감시지역이 넓고 오퍼레이터가 상주할 수 없는 환경이 대부분이기 때문이다. 감시 카메라는 적당한 위치에 특정 방향을 지향하도록 설치되며, 시설물의 전체보다는 일부 영역을 감시하도록 설치된다.

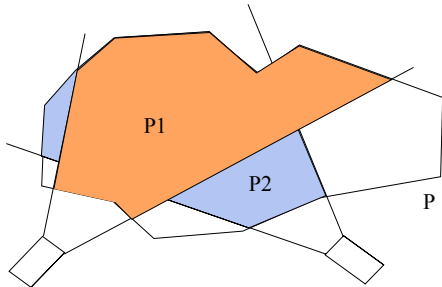
통신모듈은 카메라가 촬영한 영상을 서버에 전달하는 기능을 수행하며, 폐쇄회로 또는 IP기반의 통신 프로토콜을 사용한다. 고정식 카메라를 사용하는 경우, 카메라 제어가 불필요하므로 양방향 통신을 사용할 필요는 없다.

서버에 전달된 영상은 품질향상을 위한 전처리 과정을 거치게 되며, 이로부터 이동물체의 위치와 방향 등이 추출된다. 이동물체에 대한 정보 추출은 영상의 품질에 영향을 받으므로 이 단계는 시스템 성능에 중요하다. Fig. 1에서 보는 바와 같이 전처리 과정을 거친 영상으로부터 추출된 이동물체에 대한 정보는 최종적으로 감시영역과의 비교를 통해 침입여부를 판단하는데 사용된다. 따라서 감시영역을 입력하는 기능은 침입탐지 시스템을 구성하는 기본 요소임을 알 수 있다.

Fig. 1의 점선부분은 침입감시 시스템의 S/W를 구성하는 핵심부분으로 이동물체 정보와 감시영역 정보를 기반으로 침입탐지가 이루어지는 개념을 설명한다. 여기서 각 카메라는 자신의 관점에서 감시영역이 정의되어야함을 내포하고 있다.

Fig. 2에서 보는 바와 같이 입력된 다각형 P는 감시구역 전체를 정의하지만 카메라 관점으로 변환하게 되면 카메라는 위치에 따라 그 일부인 P1, P2만을 볼 수 있다. 입력된 전체 다각형은 카메라 관점에서 재구성되기 때문에 사용하는 감시영역을 재입력할 필요가 없다. 각 카메라

라 관점에서 재정의된 다각형은 침입여부를 판단하는 기준이 되며, 침입여부를 판단하는 문제는 폐곡선 내부·외부를 판별하는 계산기하학 문제가 된다[7-9].



[Fig. 2] redefining monitoring zone for each Camera Position

감시영역 다각형을 카메라 관점으로 변환하는 과정은 크게 세단계로 이루어진다. 첫째 대화형 편집기를 사용하여 디지털 지도위에 감시영역을 다각형으로 편집하고 정점의 리스트 얻는다. 둘째 스크린 좌표계 다각형을 실수좌표계로 변환한다. 셋째 마지막으로 다각형을 카메라 좌표계로 변환한다. 다음 장에서는 이러한 변환과정을 설명한다.

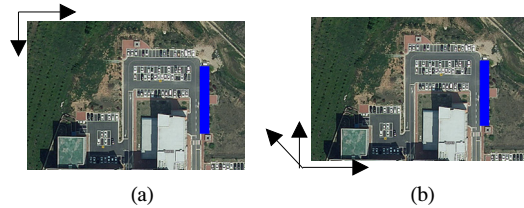
### 3. 감시 영역 입력을 위한 좌표계와 다각형의 변환

#### 3.1 감시 영역 다각형 정의

본 논문에서는 감시영역 정의를 위해 평면지도를 사용한다. 평면지도는 지표와 수직인 방향에서 촬영된 지도를 말하며, 구글 지도와 같은 위성지도가 대표적인 예이다. 위성지도는 대기권 밖에서 지표면과 수직인 방향으로 원근 투영되어 획득되지만 충분히 원거리에서 작은 지역을 촬영하는 것이므로 허용오차 범위에서 평행투영에 의해 촬영된 것으로 생각할 수 있다. 대화형 편집도구를 사용하여 평면지도 위에 감시영역을 다각형으로 표시하고 정점의 좌표를 배열에 저장하면 감시영역 설정작업은 종료된다.

Fig. 3(a)는 감시영역이 평면지도 위에 표시된 예를 보여준다. 편집도구는 감시 시스템에 영향을 주지 않으나 다각형의 표현은 좌표계와 깊은 관계를 가지므로 목적에 맞게 좌표계를 설정할 필요가 있다. 대화형 편집도구는 스크린 좌표계를 사용하는 반면 영상감시에서는 위치와 거리의 개념을 포함하므로 실수좌표계를 사용한다. 따라

서 정수좌표 다각형을 실수좌표로 변환하기 위해 2D 지도 좌표계를 도입한다.



[Fig. 3] Satellite Map and Map Coordinate  
(a) Polygon vertices in satellite map  
(b) Set map coordinate

스크린 좌표계의 임의의 픽셀을 지도좌표계 원점으로 설정할 수 있으나, 편의상 스크린 좌표계의 좌측하단을 지도좌표계 원점으로, 수평방향과 수직방향을 각각 U, V 축으로 설정한다. 윈도우 시스템의 스크린 좌표계는 좌측상단을 원점으로 우측방향을 x-축, 하단방향을 y-축으로 사용하기 때문에 먼저 정점의 y-좌표를 반대방향으로 변환하여야 한다. 세로방향 최대 픽셀 수를 j\_MAX라 할 때, 스크린 좌표계의 한 픽셀 p=(i, j)는 식 (u, v)=(i, -(j-j\_MAX)) 에 의해 지도 좌표계 한 픽셀 p(u, v)로 간단히 변환된다.

한편, 디지털 지도는 각 픽셀에 대해 GPS 좌표가 주어지므로 원점에 대한 GPS 좌표를 얻을 수 있고, 이를 O\_m이라 하자. 이 정보는 뒤에서 설명할 카메라 영상과의 정합에 이용된다. Fig. 3(b)는 위성지도 위의 지도좌표계를 보여준다. 디지털 지도의 가로와 세로는 뒤에 설명할 광역좌표계와 일치할 필요는 없으며, 벡터 U 또는 V가 광역좌표계 원점에 대해  $\Phi$  만큼 회전되었다고 가정하자. 이 정보는 후에 지도좌표를 광역좌표계로 변환하는데 사용된다.

디지털 지도는 한 픽셀에서 다른 픽셀까지의 거리를 미터(m)단위로 제공하므로 픽셀 당 실측거리(w)를 계산할 수 있고 이를 이용하여 식 (1)과 같이 정점 p=(u, v)를 미터(m)단위로 변환할 수 있다.

$$(u', v') = w \times (u, v) \quad (1)$$

여기서 w은 픽셀 거리를 나타낸다. 기본적으로 지도는 축척을 가지고 있으나 식 (1)에 의해 다각형 정점이 실거리로 변환된다.

#### 3.2 광역 좌표계 변환

광역좌표계는 절대좌표계로서 카메라 설치 장소를 설계하고, 감시영역 다각형을 측지 공간과 정합하기 위한

3D 실수좌표계이다. 편의상 지도 좌표계 원점  $O_m$ 의 GPS 좌표와 동일한 지점을 광역 좌표계 원점으로 설정하고, 이를  $O_g$ 라 하자. 그러나 반드시 지도좌표계 원점의 GPS 좌표와 광역좌표계 원점의 GPS 좌표가 일치할 필요는 없다. 두 좌표가 서로 다른 경우 기준에 제안된 지도와 GPS 정보를 일치시키는 방법을 이용하여 일치시킬 수 있으며 [10], 실거리를 측정하여 원점을 일치시킬 수 있다. 이 경우, 지도좌표계 원점으로부터 광역좌표계 원점까지의 벡터를  $D$ 이라 하자.

사용자는 카메라 위치를 설계하고 설치할 때 극방향을 따라 실측하는 것이 편리하기 때문에 광역좌표계 원점  $O_g$ 를 기준으로 정동방향 단위벡터를  $x$ -축, 정북방향 단위벡터를  $y$ -축,  $xy$ -평면에 수직인 벡터를  $z$ -축으로 각각 설정하고, 각각  $X$ ,  $Y$  그리고  $Z$ 라 하자. 여기서 지도좌표계는 광역좌표계에 대해  $\Phi$  만큼 회전되어 있다고 가정하자.

이제 지도좌표로 표현된 감시영역 다각형의 정점을 광역좌표계로 변환할 수 있다. 지도좌표계 다각형의 한 정점을  $p_m$ 이라하면  $p_m$ 은 식 (2)와 같이 광역좌표로 변환된다. 변환행렬은 광역좌표계를 기준으로 지도좌표계까지의 이동벡터, 지도좌표계의 회전변환이 포함된다.

$$p_g = M \cdot p_m \quad (2)$$

여기서  $M = [\cos \Phi, -\sin \Phi, 0, D_x], [\sin \Phi, \cos \Phi, 0, D_y], [0, 0, 1, D_z], [0, 0, 0, 1]$ 이고, 각 요소 벡터는 행벡터이다.

### 3.3 카메라 좌표계와 정점 변환

카메라 관점에서 경계영역을 인식하기 위해서는 광역좌표계로 변환된 경계영역 다각형은 카메라 좌표계로 변환되어야 한다. 변환행렬을 유도하기 위해서는 위치, 시점방향 등 카메라 파라미터가 필요한데, 영상으로부터 추출할 수도 있으나 여기서는 사용자가 입력하는 것으로 가정한다.

사용자가 설계한 위치를 카메라 좌표계 원점으로 설정하고, 이를  $O_c$ 라 하자. 그러면 카메라 좌표계 원점은 바로 카메라 좌표까지의 벡터  $D'$ 이 된다. 카메라의 시점벡터와 상향벡터가 주어질 때, 이를 각각  $z$ -축과  $y$ -축으로 설정하고, 오른손 법칙에 따라  $x$ -축을 결정하여 각각  $X'$ ,  $Y'$ ,  $Z'$ 이라 하자. 이들이 광역좌표계를  $x$ -축,  $y$ -축,  $z$ -축에 대해 각각  $\Theta_x$ ,  $\Theta_y$ ,  $\Theta_z$  만큼 회전되어 있다면 회전변환을 위한 행렬은 식 (3)과 같이 표현된다.

$$M(\Theta_x, \Theta_y, \Theta_z) = R(\Theta_x) \cdot R(\Theta_y) \cdot R(\Theta_z) \quad (3)$$

여기서  $R(\cdot)$ 은 회전행렬이다.

사용자는 카메라를 설치할 때 카메라의 방향을 측정하기 위해 다양한 측정기를 사용하는데 이들은 대부분 극 방향을 기준으로 측정한다. 또한 카메라 영상으로부터 시점벡터를 계산하는 여러 방법이 제안되어 있으나 본 논문의 범위를 벗어나므로 생략한다[11,12].

이를 종합하면, 광역좌표계 상의 한 정점  $p_g$  를 식 (4)와 같이 카메라 좌표계  $p_c$ 로 변환할 수 있다.

$$p_c = M(\Theta_x, \Theta_y, \Theta_z) \cdot T(D') \cdot p_g \quad (4)$$

여기서  $T(\cdot) = (D'_x, D'_y, D'_z, 0)^T$ 은 이동변환 행렬이다.

이제 감시 카메라에서 촬영된 이동물체와 감시영역 다각형과 비교를 통해 침범여부를 판단할 수 있다. 카메라는 특성에 따라 촬영영역이 결정되므로 만일 변환된 다각형을 렌더링 한다면 변환된 다각형은 일부만 카메라 필름에 나타난다. 필름평면에 나타나는 다각형은  $L = P(f) \cdot p_c$  로 계산할 수 있다. 여기서  $P(\cdot)$ 은 투영행렬이다. 다음 장에서는 지금까지 설명한 감시영역 입력방법을 구현하여, 각 카메라 관점으로 변환한 결과를 보여준다.

## 4. 구현 및 실험 결과

### 4.1 다각형 입력 알고리즘

제 3장에서 설명한 감시영역 다각형의 변환과정을 요약하면 Table 1과 같이 요약할 수 있다. 스크린좌표 다각형은 지도 좌표계로 변환되고, 지도좌표계 다각형은 광역좌표계로 변환되며, 광역좌표계 다각형은 카메라 좌표계로 차례로 변환된다.

[Table 1] Boundary Input Algorithm

#### Procedure boundary input for surveillance

input:  $S_s, V(i, j, t), C_p(x, y, z, \theta_1, \theta_2, \theta_3)$

output:  $S_c$

begin

Set up the 2D map coordinates  $M=(U, V)$   
 $(u', v') \leftarrow (u \times w, v \times w)$ ,  
 for  $p(u, v) \in S_s; S_M \leftarrow p(u', v')$

Set up the global coordinates  $G=(X, Y, Z)$

Get the displacement vector  $D$  from  $G$  to  $M$   
 and the orientation  $\Phi$  of  $M$  w.r.t.  $G$

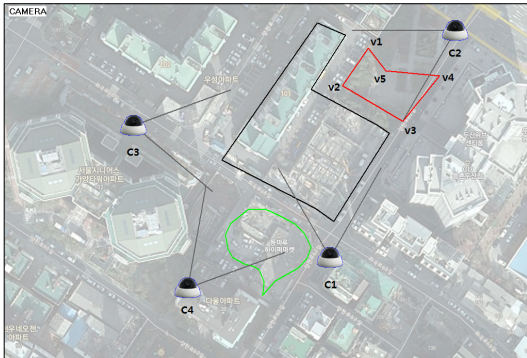
```

p(x, y) <- R(Φ)·T(D)·pT(u', v'),
SG <- p(x, y, 0), for p(u', v') ∈ SM
Set up the camera coordinates C=(X', Y', Z')
Get the displacement vector D' and
the orientation θ1, θ2, θ3 of a camera

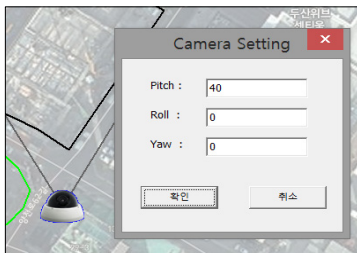
p'(x, y, z) <-
    M(f, θ3, θ2, θ1, D')·p(x, y, z),
for p(x, y, z) ∈ SG
SC <- p'(x,y,z), for p(x,y,z) ∈ SG
end
end Procedure
    
```

4.2 실험 결과

4.1 에서 제시한 알고리즘을 검증하기 위해 프로그램을 제작하여 3D 시뮬레이션을 진행하였다. 먼저, 감시 영역과 카메라 파라미터 데이터를 입력하기 위해 Fig. 4와 같이 위성 지도 위에 감시 영역과 카메라를 배치하였다. 감시 영역은 영역의 내·외부를 구분하기 위해 정점들의 연결리스트로 구성된 Simple Closed Curve Polygon 으로 정의하였고, 카메라는 화각과 높이를 고정 값으로 정의한 뒤 Fig. 5 와 같이 각 카메라의 Pitch, Roll, Yaw 값을 입력하도록 하였다.

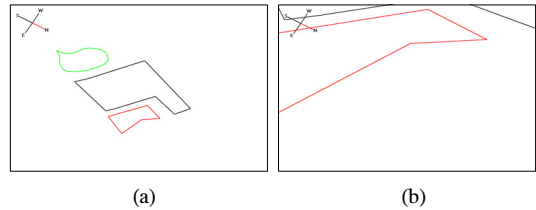


[Fig. 4] Input monitoring zone and camera information



[Fig. 5] Set camera parameter

입력된 영역 데이터와 카메라 파라미터를 기반으로 각 카메라 별로 알고리즘을 적용하여 감시 영역 폴리곤의 좌표들을 3D 좌표로 변환 후 스크린에 투영해 주었다. 이때 3D 투영 변환을 위해 OpenGL의 변환 과정을 참조하였다[13]. Fig. 6은 투영 변환 적용 유무에 따른 변환 결과의 차이를 보여주고 있다. Fig. 6(a)는 영역 다각형에 이동과 회전만을 적용하여 변환한 상태이기 때문에 카메라의 초점거리나 화각 등이 적용되지 않은 상태이다. 여기에 Fig. 6(b)와 같이 투시 투영 변환을 적용하여 카메라 시점에서의 변환이 완료된다.



[Fig. 6] Perspective Projection Transform  
(a) Before Transform (b) After Transform

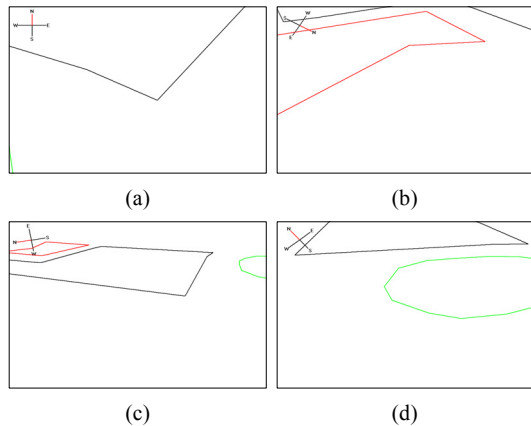
Table 2는 Fig. 5에서 카메라 C2가 5개의 정점 v1, v2, v3, v4, v5 로 구성된 감시영역 다각형을 바라볼 때 정점 v1, v2, v3, v4, v5 가 지도좌표, 광역좌표, 카메라좌표 순으로 변환된 결과를 보인다. 여기서 카메라의 화각과 높이는 각각 60도와 10m 이며, 카메라의 위치는 (876, 711), 카메라의 Pitch, Roll, Yaw 값은 각각 35도, 0도, 240도 이다.

[Table 2] Coordinate Transform for each vertex

	Input Coordinate	Map Coordinate	Global Coordinate	Camera Coordinate
v1	707, 677	176.75, 169.25	176.75, 169.25, 0.00	57.22, 85.37, 39.16
v2	657, 604	164.25, 151.00	164.25, 151.00, 0.00	12.98, 112.07, 55.46
v3	774, 535	193.50, 133.75	193.50, 133.75, 0.00	-100.58, 90.79, 41.78
v4	845, 623	211.25, 155.75	211.25, 155.75, 0.00	-103.07, 18.23, 20.23
v5	741, 633	185.25, 158.25	185.25, 158.25, 0.00	-0.23, 81.92, 37.62

Fig. 7 에서는 각 카메라 C1,C2,C3,C4의 시점에 대한 영역 다각형들의 변환 결과를 보여주고 있다. 실험 결과를 통해 지도에 정의한 영역 다각형이 카메라 파라미터

에 따라 각 카메라 시점에 맞게 잘 변환되었음을 확인할 수 있다. 이로써 본 논문의 제안방법이 기존의 연구들과 달리 다수의 카메라 환경에서 영역 데이터나 카메라 파라미터에 변동이 있을 경우, 하나의 데이터에서 변화한 값을 수정함으로써 변동 사항이 전체 카메라 시스템에 적용되기 때문에 시스템 관리 측면에서 더 효과적임을 확인하였다.



[Fig. 7] Transformation result for each camera  
(a) C1 (b) C2 (c) C3 (d) C4

## 5. 결론

감시 시스템의 보급이 증가하고 규모가 커짐에 따라 지능형 감시 시스템의 필요성이 증가하고 있다. 지능형 감시 방법 중 하나인 특정 영역 감시 또한 많은 연구가 진행되고 있고, 이러한 감시 시스템에서 영역 데이터 입력 방법은 성능에 많은 영향을 준다.

본 논문에서는 영역 감시를 위해 위성 지도에 다각형으로 영역을 정의하고 이를 카메라 파라미터를 이용하여 카메라 시점으로 변환하는 방법을 제안하였다. 제안하는 방법으로 실험한 결과, 평면 지도에 정의했던 영역 다각형이 카메라 시점에 맞게 변환이 이루어졌다.

이후 보안 감시 시스템을 완성하기 위해서는 영상 내에서 이동 물체를 인식하여 변환된 경계 영역으로의 접근과 침입을 탐지하는 방법과 장애물에 의해 감시 영역이 가려지는 경우에 대한 처리 방법들이 추가로 연구되어야 한다. 또한 카메라 영상과 변환된 영역 데이터 간의 좌표 오차를 최소화하기 위해 왜곡 보정과 좌표 보정 등의 프로세스가 추가로 적용되어야 한다.

## References

- [1] Freer, J. A, "Automatic video surveillance with intelligent scene monitoring and intruder detection", Security Technology, 30th Annual 1996 International Carnahan Conference, pp. 89-94, Oct, 1996
- [2] Jong-Un, Won, "A Study on the Adaptive Object Recognition in Safe Zone Monitoring", The Korean Society for Railway, 2010.
- [3] Sehchan Oh, "Train detection in railway platform area using image processing technology", Journal of the Korea Academia-Industrial cooperation Society, Vol. 13, No. 12, pp. 6098-6104, 2012  
DOI: <http://dx.doi.org/10.5762/KAIS.2012.13.12.6098>
- [4] Axis, "Axis Video Motion Detection", [www.axis.com/products/video\\_motion\\_detection/index.htm](http://www.axis.com/products/video_motion_detection/index.htm)
- [5] i2v system, "Video Intrusion Detection", [www.i2vsys.com/products.html](http://www.i2vsys.com/products.html)
- [6] Ilguk, Jo, "Digital Image Monitoring Apparatus Through Setting Virtual Tripzone and The Method", KIPO, 2011
- [7] Shimrat, M., "Algorithm 112 : Position of point relative to polygon" Communications of the ACM Volume 5 Issue 8, 1962.  
DOI: <http://dx.doi.org/10.1145/368637.368653>
- [8] R. Courant, H. Robbins, "What is Mathematics, Fourth Printing", pp. 267-269, Oxford University Press, 1969
- [9] Joseph O' Rourke, "Computational Geometry in C", Cambridge, 1998
- [10] Chan-Ung, Jun, "Positioning and Mapmatching Technique using DR/DGPS Integrated Kalman Filter", Chung-Ang University, 2003
- [11] Gallagher, A.C., "Using Vanishing Points To Correct Camera Rotation In Images", Computer and Robot Vision., Proceedings. The 2nd Canadian Conference on, pp. 460-467, May, 2005
- [12] Guo-Qing Wei, "Determining vanishing point and camera parameter: new approaches", Pattern Recognition, 1988., 9th International Conference on, pp. 450-452, Nov, 1988  
DOI: <http://dx.doi.org/10.1109/ICPR.1988.28263>
- [13] Edward Angel, "Interactive Computer Graphics 5th edition", Pearson Education, 2008

**김 재 혁**(Jae-Hyeok Kim)

[정회원]



- 2012년 2월 : 공주대학교 컴퓨터 공학과 학사
- 2012년 3월 ~ 현재 : 공주대학교 대학원 컴퓨터공학과 석사

<관심분야>

영상처리, 컴퓨터 그래픽스

---

**맹 승 렬**(Seung-Ryol Maeng)

[정회원]



- 2004년 2월 : KAIST 졸업 (공학 박사)
- 1994년 1월 ~ 현재 : 공주대학교 컴퓨터공학부 교수

<관심분야>

컴퓨터 그래픽스, 계산기하학