

論文94-31B-11-8

시공간 적응탐색에 의한 실시간 이동물체 추적

(Real-Time Tracking of Moving Object by Adaptive Search in Spatial-temporal Spaces)

金啓榮*, 崔瀟一*

(Gye Young Kim and Hyung Ill Choi)

要約

본 논문에서는 연속적으로 입력되는 영상에서 이동물체에 대한 정보를 추출하고 카메라의 시계(field of view)를 변경시키는 서보기기를 제어하여, 이동물체가 항상 영상의 중심에 위치하도록 하는 작업을 실시간에 수행하는 시스템에 대하여 기술한다.

영상은 방대한 양의 2차원적인 신호이므로 주어진 영상 전체에 필요한 연산(operation)을 수행하는 것은 방대한 계산시간을 필요로 한다. 특히, 메모리에서 화소들의 값을 처리기로 로딩(loading)하는데 소요되는 시간은 영상의 크기에 비례하여 기하급수적으로 증가한다. 따라서, 본 논문에서는 이러한 문제를 해소하고 이동물체에 대한 추적을 실시간에 수행하기 위하여 영상공간 및 시간공간을 선별적으로 탐색하는 방안에 초점을 맞춘다.

영상공간 및 시간공간을 선별적으로 탐색하면서 이동물체를 추적하기 위하여 본 논문에서는 물체가 카메라의 시계에 진입하는 것과 진입방향을 감지하는 방법, 진입방향에 따라 이동물체를 효과적으로 탐색하는 방법, 물체의 이동속도와 중심을 산출하는 방법, 서보기기의 성능과 물체의 속도 및 처리시간을 고려하여 서보기기를 제어하는 방법, 그리고 변화하는 속도의 물체를 추적하기 위하여 영상간의 시간간격(Δt)을 적당히 조절하는 방법에 대해서 기술한다.

Abstract

This paper describes the real-time system which, through analyzing a sequence of images, can extract motional information on a moving object and can control servo equipment to always locate the moving object at the center of an image frame.

An image is a vast amount of two-dimensional signal, so it takes a lot of time to analyze the whole quantity of a given image. Especially, the time needed to load pixels from a memory to processor increases exponentially as the size of an image increases. To solve such a problem and track a moving object in real-time, this paper addresses how to selectively search the spatial and time domain. Based on the selective search of spatial and time domain, this paper suggests various types of techniques which are essential in implementing a real-time tracking system. That is, this paper describes how to detect an entrance of a moving object in the field of view of a camera and the direction of the entrance, how to determine the time interval of adjacent images, how to determine nonstationary areas formed by a moving object and calculate velocity and position information of a moving object based on the determined areas, how to control servo equipment to locate the moving object at the center of an image frame, and how to properly adjust time interval(Δt) to track an object taking variable speed.

* 正會員, 崇實大學校 電子計算學科
(Department of Computer Science,
Soong Sil Univ.)

接受日字 : 1993年 11月 5日

*이 논문은 1994년 한국통신 장기기초과재(94-34) 연구지원비에 의하여 연구되었음

1. 서론

약간의 시차를 두고 연속적으로 입력되는 동영상(dynamic image)을 분석하여 실세계에 대한 정보를 추론하는 작업은 하드웨어 기술의 급속한 발전과 더불어 그 중요성이 강조되고 있다. 특히 이동물체를 탐지하고 추적하는 작업은 군사^[1,2] 및 산업 분야^[3,4,5]에 적용될 수 있는 매우 흥미있고 유용한 연구분야로서 이를 위한 많은 기법들이 문헌에 소개되고 있으며^[6,7], 일부에서는 이미 상품화 작업을 추진하고 있는 실정이다. 예를 들어, Oxford 대학의 로보틱스 연구 그룹에서는 스테레오 영상을 분석하여 이동물체에 대한 추적작업을 수행하는 YORICK 시스템^[8]을 발표하였다. 문헌에 소개된 이들 기법들은 기본적으로 다음과 같은 몇가지 유형으로 분류될 수 있다. 첫째는 연속적인 영상들을 분석하여 각 화소 또는 영역에 속도 벡터를 할당하고, 유사한 속도벡터를 갖는 화소 또는 영역들의 그룹을 감지하여 이동물체에 대한 추적작업을 수행하는 방법이다. 예를 들어, William 등은 점대점 대응관계로부터 속도벡터를 추출하는 방법^[9], Vincent는 현재영상에서 추출된 특징점과 이전영상에서 추출된 자취정보(tracking history)를 사용하여 이동물체를 추적하는 방법^[11], Chow와 Jian은 영역정보를 이용하여 후보영역간의 정합을 시도함으로써 이동물체를 추적하는 방법을 제안했다.^{[11][12]} 두 번째 유형은 연속적인 영상을 분석하여 광류(optical flow)를 계산하고 계산된 광류에 기초하여 물체의 이동속도 및 방향에 대한 정보를 추론하는 방법이다. 예를 들어, Horn과 Schunck는 영상내의 각 화소의 속도벡터의 변화는 완만하다는 제약조건을 사용하여 광류를 측정하는 방법^[13], Rognone 등은 측정된 광류와 완화법(Relaxation method)을 사용하여 이동물체를 추출하는 방법^[14], 그리고 [15]에서는 공간경사(spatial gradient)를 측정하는 방법을 개선하여 측정오차를 최소화하는 방법을 제안하였다. 세 번째 유형은 인접영상간의 명암차(difference of gray-values)에 의하여 형성된 차영상(difference images)들을 분석하여 물체의 이동형태에 대한 정보를 추론하는 방법이다. 예를 들어, Jain 등은 인접영상간의 명암차를 이용하여 정지영역과 이동영역을 분리하는 방법^[16] 주변처리기에서 차연산을 사용하여 물체의 동작정보를 추출하는 방법^[17], Liang-Hwa 등은 차연산과 적응예측자를 사용한 방법^[18], 그리고 [19]에서는 움직임 정보를 이용한 영상영역화 방법과 추적장을 적용하여 이동물체를 추적하는 방법을 제안하였다.

본 논문에서 제안하는 실시간 이동물체 추적방법은 세 번째 유형에 속하는 방법으로서, 카메라의 시계(FOV : field of view)에 진입하는 물체를 탐지하고 서보기기(servo equipment)를 구동하여 탐지된 물체에 대한 추적작업을 실시간에 수행하는 방법을 제안한다. 인접한 시점에서 촬영된 영상간의 차영상을 분석하여 물체의 이동에 관한 정보를 추론하기 위하여는 적절한 임계값(threshold value)을 이용하여 차영상에 대한 이진화 작업을 수행하여야 한다. 본 논문에서는 참고문헌 [17]에서 제시된 방법을 변형하여 이진화 작업을 수행한다. 그러나, 영상은 방대한 양의 2차원적인 신호이므로, 주어진 영상의 전체영역에 대하여 차연산(difference operation)을 수행하는 작업은 방대한 계산시간을 필요로 한다. 특히, 메모리에서 화소들의 값을 처리기로 로딩(loading)하는데 소요되는 시간은 영상의 크기에 비례하여 기하급수적으로 증가한다. 따라서, 본 논문에서는 이러한 문제를 해소하고 이동물체에 대한 추적을 실시간에 수행하기 위하여 영상공간 및 시간공간을 선별적으로 탐색하는 방안이 초점을 맞춘다.

본 논문의 구성은 다음과 같다. 제 I 절에서는 본 논문의 기본방향에 대하여 기술하였다. 제 II 절에서는 본 논문에서 제안하여 구현한 실시간 이동물체 추적시스템의 총체적인 구조와 각 구성요소의 기능에 대하여 기술한다. 인접한 세 영상을 분석하여, 즉 t_1 , t_2 , t_3 시점에서 획득된 영상을 분석하여 이동물체에 대한 추적을 수행하기 위하여 동기화 처리기, 동작정보 추출기 및 서보 제어정보 추출기의 기능에 대하여 기술하고, 시스템의 총체적인 제어흐름에 대하여 설명한다. 제 III 절에서는 카메라의 시계에 진입하는 물체를 감지하고 감지된 물체의 진입방향 및 연속적으로 입력되는 영상의 촬영간격(Δt)을 결정하는 동기화 작업을 실시간에 수행하는 방법에 대하여 기술한다. 특히, 주어진 영상공간의 제한된 영역들만을 분석하여 물체의 진입유무와 진입방향을 확인하는 방법과 시간축을 적응탐색(adaptive search)하여 물체의 이동속도와 조화를 이루는 시간간격(Δt)를 결정하는 방법을 제안한다.

제 IV 절에서는 영상공간과 시간축을 적응적으로(adaptively) 탐색하여 이동물체에 의하여 형성된 영역을 실시간에 탐지하는 작업을 기술한다. 일반적으로 추적하고자 하는 이동물체는 어느 정도의 크기를 갖고 있다는 전제하에서 가능한 빠른 시간안에 물체영역의 경계를 확인할 수 있는 방법을 제안한다. 제 V 절에서는 현재까지 추적작업이 진행되어온 상황을 나타내는 자취정보(tracking history)를 활용하

여 이동물체의 위치 및 속도를 추출하는 방법에 대하여 기술한다. 이동물체에 의하여 형성된 동작영역들 중에서 추적대상이 되는 영역을 선정하기 위하여 자취정보를 활용하는 방안을 제안하고, 선정된 동작영역들로부터 물체의 이동에 관한 정보를 추론하는 방안을 기술한다. 제 VI절에서는 이동물체에 대한 추적 작업을 수행하기 위하여 서보기기에 전달될 제어신호를 산출하는 방법을 기술한다. 물체의 이동에 관한 정보, 즉 이동물체의 위치 및 속도에 관한 정보를 이용하여 서보기기에 전달될 수직각(tilt angle)과 수평각(panning angle)을 산출하는 과정을 기술한다. 제 VII절에서는 본 논문에서 구현한 실시간 이동물체 추적시스템에 대한 실험결과 및 결과에 대한 분석을 기술한다.

II. 시스템 개요 및 각 구성요소의 기능

본 절에서는 카메라의 시계로 진입하는 물체를 탐지하고 이를 추적하는 실시간 이동물체 추적시스템의 총체적인 작업흐름과 각 구성요소의 기능에 대하여 기술한다. 그림 1은 본 논문에서 제안하는 시스템의 기본적인 구조를 보여준다. 영상 입력 장치는 카메라를 통하여 입력되는 영상을 계수화하여(ADC : Analog to Digital Converter) 버퍼(buffer)에 저장하는 장치이다. 계수화되어 저장된 데이터는 필요시 동기화 처리기와 동작정보 추출기에서 사용된다.

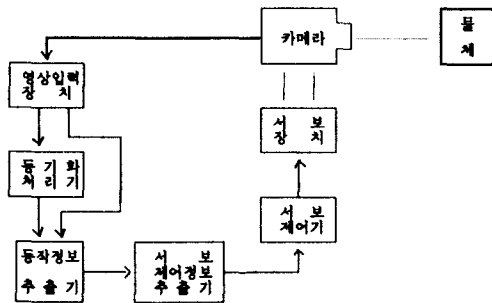


그림 1. 실시간 이동물체 추적시스템의 구성요소
Fig. 1. Components of real-time moving object tracking system.

동기화 처리기는 약간의 시차(Δt)를 두고 입력되는 두 영상간의 차연산을 통하여 카메라 시계에 물체의 진입여부와 진입방향을 감지하여 다음 처리기인 동작정보 추출기에 그 정보를 전달한다. 동기화 처리기는 초기에 이동물체의 진입을 확인하는 과정에서만 구동

되고, 일단 물체의 진입이 확인되면 이 물체에 대한 추적작업이 완료될 때까지 더 이상 구동되지 않는다. 동작정보 추출기는 동기화 처리기에서 전달된 정보에 따라 적당한 탐색방법으로 물체의 이동방향과 속도를 산출한다. 동기화 처리기는 두 영상간의 연산을 수행하지만 동작정보 추출기는 인접한 세 영상으로부터 이동물체의 위치와 속도를 추출한다. 서보 제어정보 추출기는 동작정보 추출기에서 계산된 물체의 이동속도와 처리시간 그리고 서보장치의 성능과의 관계를 고려하여 카메라의 이동방향과 이동정도를 계산하는 과정이다. 서보 제어기는 서보 제어정보 추출기에서 산출된 카메라의 이동방향으로 이동시간 만큼 제어신호를 보내는 장치이다. 서보장치는 카메라의 시계를 변화시킬 수 있는 장치로써 자유도(degree of freedom)는 2이다. 즉, 서보장치의 축을 기준으로하여 카메라의 시계를 상하 및 좌우로 변화시켜 현재 추적 중인 물체가 영상프레임의 중앙에 위치하도록 한다.

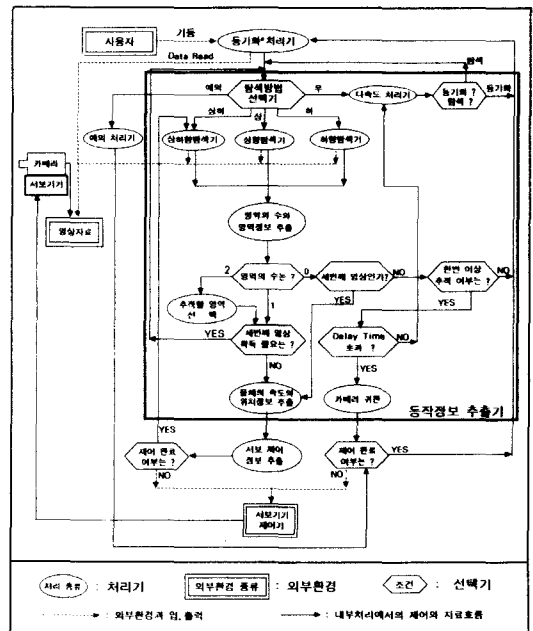


그림 2. 실시간 이동물체 추적시스템 구조도
Fig. 2. Structural organization of real-time moving object tracking system.

그림 2는 본 논문에서 제안하는 시스템의 제어절차와 자료흐름을 나타낸 구조도이다. 시스템을 구동시키면 동기화 처리기는 시간축 및 공간축을 적응탐색(adaptive search)하며 인접한 두 영상간의 차연산을 수행하여 동기화 정보를 추출한다. 즉, 물체의 진입여부를 결정하고 진입물체가 존재하는 경우에는 진

입방향을 판단하며 인접영상간의 시간간격 Δt 를 얼마로 할 것인가를 결정한다. 동기화 처리기의 처리결과에 따라 다음에 수행될 작업이 결정된다. 즉, 예외처리, 상향탐색, 하향탐색, 상하향탐색, 그리고 진입물체가 없는 경우 중에서 하나가 선택한다.

예외처리기에서는 카메라의 시계를 변경하는 작업을 수행한다. 즉, 사전에 설정된 방향으로 일정각도 만큼 카메라의 광축(optical axis) 방향을 변경시키고 동기화 처리를 다시 수행한다. 물체의 진입속도와 두 인접영상간의 시간간격이 조화를 이루지 못하면 동기화 처리기는 물체의 진입을 감지하지 못한다. 따라서, 동기화 처리기가 진입물체를 감지하지 못하는 경우에는 다속도 처리기에 의하여 두 입력영상간의 시간간격을 조정된 후 동기화 처리를 다시 수행한다. 다속도 처리기가 호출되는 경우는 동기화 작업을 위한 경우와 이동물체의 추적을 위한 경우로 분류될 수 있다. 다속도 처리기가 동기화를 위한 경우는 처리결과가 동기화 처리기에 전달되고, 이동물체의 추적을 위하여 사용되는 경우는 탐색방법 선택기에 전달된다.

상향탐색기는 영상의 하위에서 상위방향으로 진입한 이동물체의 탐색을 수행하고, 하향탐색기는 영상의 상위에서 하위방향으로 이동물체의 영역을 탐색하며, 상하향탐색기는 영상의 중앙에서 하위방향과 상위방향으로 이동물체의 영역을 탐색한다. 각 탐색기는 정의된 방향으로 인접한 두 영상을 스캔하면서 대응되는 화소간의 명암값을 비교하여 동작영역을 확인하고 각 동작영역의 특성을 추출한다. 즉, 차영상에서 물체의 이동에 의하여 형성된 영역을 추출하고 추출된 영역의 특성을 확인한다. 동작영역은 물체의 이동에 의하여 형성된 영역으로서 이에 대한 자세한 설명은 제 III 절에서 기술된다.

추출된 동작영역의 위치는 이동물체의 위치에 대한 정보를 제공하고, 동작영역의 위치변화에 대한 정보는 이동물체의 속도에 대한 중요한 정보를 제공한다. 그러나, 영상처리 과정에서 복수 개의 동작영역이 형성될 수 있으므로 이들에 대한 적절한 처리가 이루어져야 한다. 또한, 동작영역의 위치변화에 대한 정보를 추론하기 위하여는 전단계에서 분석된 정보를 활용하여야 한다. 본 논문에서 제안하는 시스템에서는 카메라가 정지된 상태에서 촬영된 인접한 세 시점의 영상(t_1 , t_2 , 그리고 t_3 시점에서 촬영된 영상)을 분석하여 이동물체에 대한 추적을 수행한다. 즉, 인접한 두 영상을 분석하여 추출된 동작영역의 수가 '0' 이면 자취정보(tracking history)를 이용하여 다음에 수행될 작업의 흐름을 결정한다. 즉, 시스템이 구동된 후 한번도 추적작업이 이루어지지 않은 경우에는

제어를 동기화 처리기로 보낸다. 반면에, 한번 이상 추적작업이 이루어진 경우에는 사전에 설정된 최대지연시간(maximal delay time: Δt_{max})의 초과 여부를 조사한다. 만약, 사전에 설정된 최대지연시간을 초과한 경우는 카메라를 초기상태로 귀환시키기 위하여 서보기기를 제어한다. 서보기기의 제어가 완료되면 제어를 동기화 처리기에 전달된다.

사전에 설정된 최대지연시간을 초과하지 않은 경우에는 인접영상간의 시간간격을 재조정하기 위하여 다속도 처리기를 구동한다. 이 경우는 탐색을 위한 경우이므로 다속도 처리기의 처리결과는 탐색방법 선택기에 전달되고, 탐색방법 선택기는 전에 탐색한 방법과 동일한 탐색방법을 선택한다. 그러나, 추출된 동작영역의 수가 '1'인 경우와 '2'인 경우는 다음 시점의 영상을 획득할 필요가 있는가를 확인하여 다음 작업을 결정한다. 즉, 동작영역의 추출이 t_1 및 t_2 시점의 영상을 분석하여 이루어진 경우에는 제어를 탐색방법 선택기로 보내고, 동작영역의 추출이 t_2 및 t_3 시점의 영상을 분석하여 이루어진 경우에는 이미 분석된 t_1 및 t_2 시점의 결과와 비교하여 동작정보를 추출한다. 추출된 동작정보는 서보기기에 대한 제어정보를 산출하기 위하여 사용되며, 산출된 제어정보에 의하여 서보기기가 제어된다.

본 논문에서는 차영상을 작성하기 위하여 카메라가 정지한 상태에서 인접영상들을 촬영한다. 즉, 배경부분(움직이지 않는 부분)이 변화하지 않는 상태에서 인접영상을 분석하여 이동물체의 위치 및 속도에 대한 정보를 추론하고, 추론된 정보에 기초하여 서보기기를 구동시킨다. 이처럼 카메라가 정지한 상태에서 차영상을 작성하면 촬영시간 동안 물체의 이동에 대한 추적작업을 수행하지 못하는 단점은 있으나, 차영상의 작성과정이 매우 단순해지는 잇점이 있다. 또한, 영상의 촬영시간은 매우 짧은 간격으로 이루어질 수 있으므로 카메라의 고정에 의하여 발생하는 정보의 손실은 최소화될 수 있으리라 여겨진다.

III. 시·공간 샘플링에 의한 실시간 동기화

본 절에서는 미리 설정된 영역을 감시하는 카메라의 시계에 진입하는 물체를 감지하고 감지된 물체의 진입방향 및 연속적으로 입력되는 영상의 촬영간격(Δt)을 결정하는 동기화 작업을 실시간에 수행하는 방법에 대하여 기술한다. 이동물체에 대한 추적작업을 실시간에 수행하기 위하여는 물체의 진입여부에 대한 판단이 매우 신속히 이루어져야 한다. 만약, 촬영된 영상들을 분석하는 시간이 길어져서 분석시간

중에 진입하는 물체가 발생하면 이 물체에 대한 추적 작업이 원만히 수행될 수 없다. 따라서, 본 논문에서는 주어진 영상공간의 제한된 영역들만을 분석하여 실시간 동기화 작업을 수행하는 방법을 제안한다.

진입하는 이동물체를 탐지하는 작업은 인접한 시점에서 촬영된 두 영상에 대한 차영상을 분석하여 수행된다. 즉, t_1 시점에서 촬영된 영상 $I(x, y; t_1)$ 과 t_2 시점에서 촬영된 영상 $I(x, y; t_2)$ 에 대한 차영상 $D(x, y; t_1 - t_2)$ 을 분석하여 물체의 이동에 의하여 형성된 동작영역을 확인하고, 확인된 동작영역에 기초하여 이동물체의 존재 및 이동방향에 대한 정보를 추론한다. 따라서, 영상촬영시점 t_1 과 t_2 간의 간격 Δt 를 얼마로 할 것인가 하는 문제는 이동물체에 대한 추적작업을 원만히 수행하기 위한 중요한 고려사항이다. 두 시점간의 간격이 매우 작고 이동물체가 천천히 움직이는 경우에는 차영상은 이동물체에 대한 충분한 정보를 추론할 수 없으며, 반면에 두 시점간의 간격이 물체의 이동속도에 비하여 상대적으로 큰 경우에는 비록 이동물체는 탐지된다고 하더라도 이에 대한 원만한 추적작업이 수행될 수 없다. 따라서, 본 논문에서는 시간축을 적응탐색(adaptive search)하여 물체의 이동속도와 조화를 이루는 시간간격 Δt 를 결정하는 방법을 제안한다.

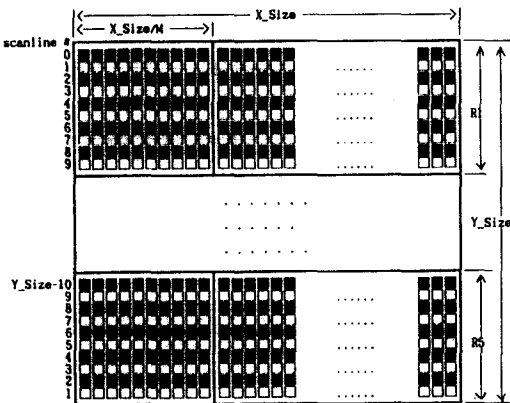


그림 3. 영상공간의 샘플링에 의한 런 구성

Fig. 3. Configuration of runs by sampling image plane.

본 논문에서는 주어진 영상의 각 열(row)에 속하는 화소들의 집단을 스캔라인(scanline)이라고 정의하고, 격간(every other scanline)으로 N 개의 스캔라인에 속하는 화소들의 집단을 하나의 런(run)으로 정의하며, 이동물체의 진입을 확인하기 위한 탐색공간을 5개의 런으로 제한한다. 영상의 가로축을 x 축

그리고 세로축을 y 축으로 정의하고, 가로축의 크기를 X_Size 그리고 세로축의 크기를 Y_Size 로 정의하자. 5개의 런들을 영상의 상단에서부터 시작하여 R_1, R_2, R_3, R_4, R_5 라고 정의하면, R_1 이 시작되는 스캔라인의 번호는 '0', R_2 가 시작되는 스캔라인의 번호는 ' $Y_Size/4$ ', R_3 는 ' $Y_Size/2$ ', R_4 는 ' $Y_Size 3/4$ ', 그리고 R_5 가 시작되는 스캔라인의 번호는 ' $Y_Size - 2N$ '이 된다. 그림 3은 한 영상에서 런을 정의하는 과정을 보여준다. 이 그림에서는 5개의 스캔라인들이 하나의 런을 형성하고 있다. 각 런은 이제 M 개의 셀(cell)들의 집합으로 재구성된다. 즉, 하나의 셀은 $N \times X_Size/M$ 개의 화소들의 집합으로 구성된다.

전시점(previous time)에서 촬영된 영상 $I(x, y; t_1)$ 의 각 셀과 현시점(current time)에서 촬영된 영상 $I(x, y; t_2)$ 의 대응되는 셀에 속하는 화소들을 비교하여 이들의 차가 미리 정한 임계값 TH_1 이상인 화소들의 수가 미리 정한 임계값 TH_2 보다 큰 셀은 동작셀(nonstationary cell)로 그렇지 않은 셀은 비동작(stationary cell)로 정의한다. 임계값 TH_1 을 너무 낮게 설정하면 잡음에 매우 민감해지는 결과를 초래하고, 너무 높게 잡으면 배경과 낮은 대조를 이루는 이동물체에 대한 탐지가 어렵게 된다. 본 논문에서는 참고문헌 [17][18]에서 사용한 8-10 정도의 값을 사용하였으며, 이는 여러 유형의 영상에 대하여 적용해본 결과 적절한 값을 확인할 수 있었다. 한편, TH_2 는 동작셀과 비동작셀을 판정하기 위하여 사용되는 임계값으로서, 이는 임계값 TH_1 과 매우 긴밀한 관계를 갖는다. 즉, TH_1 이 큰 값을 가지면 TH_2 는 상대적으로 낮은 값을 가져야 하며, TH_1 이 낮은 값을 가지면 TH_2 는 상대적으로 큰 값을 가져야 한다. 본 논문에서는 셀의 크기가 이동물체의 크기보다 상대적으로 작다는 전제하에서 셀 크기의 20% 정도를 TH_2 로 정의하였다.

동작셀을 포함하는 런은 동작런이 된다. 동기화 처리기는 5개의 런중에서 어떠한 런이 동작런인가를 확인하여 그 결과를 다음 단계의 처리에 전달한다. 만약, 모든 런이 동작런이면, 이는 이동물체의 진입방향을 확인할 수 없는 상태로 간주하고 예외 처리를 수행한다. 즉, 사전에 설정된 방향으로 설정된 각도 만큼 서보기기를 제어하고 동기화 처리를 다시 수행한다. 최상위의 런 R_1 이 동작런이고 다른 런들은 비동작런이면, 동작정보 추출기가 구동되어 하향탐색에 의한 이동물체의 탐지작업이 시작된다. 최하위의 런 R_5 가 동작런이고 다른 런들은 비동작런이면 동작정보 추출기는 상향탐색에 의한 이동물체의 탐지작업을 시

작하고, R_1 과 R_5 는 비동작런이고 기타의 런이 동작런이면 추적 프로세서는 상하향탐색에 의한 이동물체의 탐지작업을 시작한다.

그러나, 모든 런이 비동작런이면 이는 물체의 진입 또는 이동이 시점간의 간격 Δt 에 비하여 상대적으로 느린 경우로 간주하고 Δt 를 변화시키기 위하여 다속도 처리기를 구동한다. 다속도 처리기는 그림 4와 같은 세가지 유형의 시간간격을 이용하여 차영상을 작성할 것을 지시한다. 단시간 처리에서는 인접한 두 시점간의 간격을 미리 설정된 초기값 Δt_{init} 로 사용하고, 중시간 처리에서는 $4 \cdot \Delta t_{init}$ 를 두 시점간의 간격으로 사용하며, 장시간 처리에서는 $9 \cdot \Delta t_{init}$ 를 두 시점간의 간격으로 사용한다. 다속도 처리기에 의하여 Δt 를 변화시키는 과정은 다음과 같다. 먼저 단시간 처리에 의하여 물체의 진입 또는 이동에 대한 확인을 시도하고, 단시간 처리가 적절하지 못하다고 판단되면 중시간 처리 그리고 장시간 처리 순으로 진행한다.

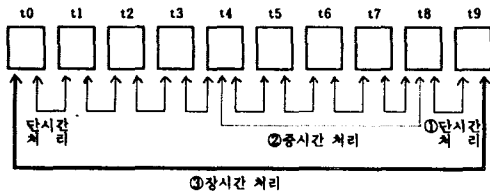


그림 4. 다속도 처리를 위한 시간축 샘플링
Fig. 4. Time sampling for multi-speed processing.

IV. 시 공간 적응탐색에 의한 이동물체 탐지

동기화 처리기에 의하여 물체의 진입이 확인되면, 이동물체에 의하여 형성된 영역을 탐지하는 작업을 시작한다. 이동물체에 의하여 형성된 영역을 탐지하는 작업은 동기화 처리에서와 같이 인접한 시점에서 촬영된 두 영상에 대한 차영상을 분석하여 수행된다. 그러나, 이러한 이동물체의 탐지작업을 실시간에 처리하기 위하여는 다음과 같은 두가지 문제를 고려하여야 한다. 첫째는 입력된 두 영상의 어떠한 영역에 대하여 차영상 $D(x, y; t_1, t_2)$ 을 구할 것인가 하는 문제이다. 즉, $D(x, y; t_1, t_2)$ 를 정의하기 위한 x 및 y 의 범위를 결정하는 문제이다. 일반적으로 영상은 방대한 양의 2차원적인 신호이므로, 주어진 영상의 전체 영역에 대하여 차연산(difference operation)을 수행하는 작업은 방대한 계산시간을 필요로 한다. 특

히, 메모리에서 화소들의 값을 처리기로 로딩 (loading)하는데 소요되는 시간은 영상의 크기에 비례하여 기하급수적으로 증가한다. 이러한 문제를 해결하기 위하여 본 논문에서는 차영상 $D(x, y; t_1, t_2)$ 가 정의되는 x 및 y 의 범위를 적응적으로(adaptively) 결정하는 방법을 제안한다. 이동물체에 대한 탐지작업을 실시간으로 수행하기 위하여 고려하여야 할 두번째 사항은 영상촬영시점 t_1 과 t_2 간의 간격 Δt 를 얼마로 할 것인가 하는 문제이다. 이는 동기화 작업에서도 중요시되었던 문제이다. 본 논문에서는 다속도 처리기를 이용하여 두 인접영상간의 시간간격을 물체의 이동속도에 적응적으로 조절한다.

이동물체에 의하여 형성된 영역을 탐지하는 작업은 두 가지 경우로 세분화될 수 있다. 첫째는 동기화 처리기에 의하여 물체의 진입이 확인된 후 처음으로 탐지작업을 시도하는 경우이고, 둘째는 이미 이동물체에 대한 추적작업이 진행중인 단계에서 탐지작업을 시도하는 경우이다. 첫번째 경우에는 동기화 처리기에서 전달받은 탐색방향에 대한 정보를 이용하여 탐색을 시도한다. 그러나, 이동물체에 대한 추적작업이 이미 진행중인 경우에는 현재 추적 중인 물체가 항상 영상프레임의 중앙에 위치하므로 상하향탐색을 시도한다.

이제 정의된 각 탐색방향으로 적응적인 탐색을 수행하여 이동물체에 의하여 형성된 영역을 결정하는 방법을 기술한다. 본 논문에서는 정의된 탐색방향으로 전시점에서 촬영된 영상과 현시점에서 촬영된 영상의 셀 단위 탐색을 시도하여 동작셀들을 추출한 후, 추출된 동작셀들을 분석하여 추적하고자 하는 물체영역을 결정한다. 일반적으로 추적하고자 하는 이동물체는 어느 정도의 크기를 갖고 있다는 전제하에서 가능한한 빠른 시간안에 물체영역의 경계를 확인할 수 있으면, 그 경계를 벗어난 지역에 대하여는 탐색작업 및 차연산을 생략할 수 있다. 따라서, 본 논문에서는 모든 런들을 조사하여 동작셀들을 추출하지 않고, 동작셀들을 추출하는 작업을 격런(every other run)으로 하여 동작셀을 포함하지 않는 런을 만날 때까지 진행한다. 예를 들어, 그림 5은 탐색방향이 상향탐색인 경우에 격런으로 탐색을 진행하며 동작셀들을 추출하는 작업을 보여준다. 즉, 이 예에서는 런 R_5 , 런 R_3 , 런 R_1 순으로 탐색을 수행하며 동작셀들을 추출한다. 격런으로 탐색을 진행하다가 동작셀을 포함하지 않는 런을 만나게되면, 이제 진행 방향을 역으로하여 이동물체의 경계를 확인한다. 즉, 그림 5의 예에서는 런 R_1 가 동작셀을 포함하지 않으므로, 이제 런 R_2 를 탐색하여 동작셀을 추출한 후 이

동물체에 의하여 형성된 영역의 경계를 확인한다.

물체의 이동에 의하여 형성된 동작셀들이 추출되면, 인접한 동작셀들로 구성된 동작영역을 추출한다. 예를 들어, 그림 5의 예에서는 세로선(■)으로 표시된 동작셀들이 하나의 동작영역을 형성하고, 가로선(■)으로 표시된 동작셀들이 또 다른 동작영역을 형성한다. 추출된 동작영역의 위치 및 동작영역의 갯수는 이동하고 있는 물체의 속도 및 형태에 대한 중요한 정보를 제공한다. 본 논문에서는 추출된 각 동작영역을 포함하는 최소포함사각형(MER: Minum Enclosing Rectangle)의 시작좌표와 끝좌표를 이용하여 동작영역의 위치를 나타낸다. 또한, 복수 개의 동작영역이 추출되는 경우에는 가장 크기가 큰 두 개의 영역에 대하여만 다음 단계의 처리를 수행한다.

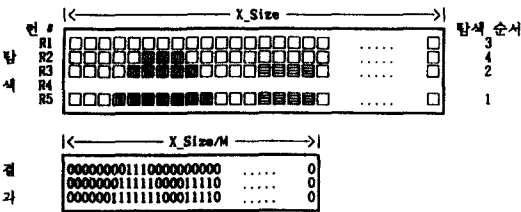


그림 5. 적응탐색에 의한 물체경계 추출(상향탐색)
Fig. 5. Extraction of moving object boundary using adaptive search. (upward search)

V. 자취정보를 이용한 동작정보 추출

이동물체에 대한 추적작업을 수행하기 위하여는 물체의 이동방향 및 속도에 대한 정보를 추출하여야 한다. 본 절에서는 현재까지 추적작업이 진행되어온 상황을 나타내는 자취정보(tracking history)를 활용하여 동작정보를 추출하는 작업에 대하여 기술한다. 동작정보를 추출하는 작업은 기본적으로 세 시점(t_1, t_2, t_3)에서 촬영된 영상들을 분석하여, 즉 $D(x, y: t_1-t_2)$ 와 $D(x, y: t_2-t_3)$ 를 분석하여 이루어진다.

두 시점간의 영상을 비교 분석하여 동작영역이 추출되면, 추출된 동작영역의 갯수에 대한 분석작업을 수행한다. 먼저 추출된 동작영역의 갯수가 '0'인 경우, 즉 동작영역이 없는 경우를 생각하자. 이러한 상태가 발생하는 경우는 추적시스템이 한번 이상 물체를 추적한 후에 발생하는 경우와 동기화 처리가 수행된 후 바로 다음에 발생하는 경우로 구분될 수 있다. 추적시스템이 한번 이상 물체를 추적한 후에 이 상태가 발생하는 경우는 이동물체가 이동하다가 정지하는 경우 또는

추적시스템이 이동물체에 대한 추적을 실패한 경우가 된다. 이러한 경우에는 추적시스템은 현재시간을 기록하고 다속도 처리기를 호출하여 시간간격 Δt 를 변경한 후 사전에 설정된 최대지연시간(maximal delay time)까지 추적작업을 계속 수행한다. 만약 이 시간이 경과할 때까지도 동작영역이 추출되지 않으면 카메라라는 처음의 위치로 복귀된다. 예를 들어, 그림 6은 $D(x, y: t_1-t_2)$ 에서는 동작영역이 추출되었으나 $D(x, y: t_2-t_3)$ 에서는 동작영역이 추출되지 않은 경우이다. 이 그림에서 가는선으로 표현된 직사각형은 t_1 에서의 물체의 위치를 나타내고 굵은선으로 표현된 직사각형은 t_2 및 t_3 에서의 물체의 위치를 나타낸다. 그림 6에서 $D(x, y: t_1-t_2)$ 에 의하여 추출된 최소포함사각형의 좌표를 (Start_x1, Start_y1)와 (End_x1, End_y1)라 할때 이동물체의 속도 (v_x, v_y), 위치 (Start_x0, Start_y0) 및 (End_x0, End_y0), 그리고 중심위치 (x_0, y_0)는 다음 식과 같이 구해진다.

$$\begin{aligned}
 v_x &= 0, & v_y &= 0 \\
 \text{Start_x}_0 &= \text{Start_x}_1, & \text{Start_y}_0 &= \text{Start_y}_1 \\
 \text{End_x}_0 &= \text{End_x}_1, & \text{End_y}_0 &= \text{End_y}_1 \\
 x_0 &= (\text{Start_x}_0 + \text{End_x}_0) / 2, & y_0 &= (\text{Start_y}_0 + \text{End_y}_0) / 2
 \end{aligned}
 \tag{1}$$

동작영역이 추출되지 않는 상황이 동기화 처리가 수행된 후 바로 다음에 발생하는 경우는 이동물체가 진입하다가 탈출한 경우이다. 이러한 경우에는 동기화 처리기를 재구동한다. 즉, 이동물체 추적시스템은 물체가 카메라의 시각범위에 진입하는 것을 탐지하는 작업을 수행한다.

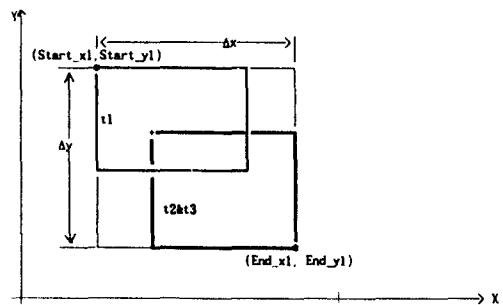


그림 6. $D(x, y: t_2-t_3)$ 에서는 동작영역이 추출되지 않은 경우
Fig. 6. Case where there does not exist nonstationary area in $D(x, y: t_2-t_3)$.

추출된 동작영역의 갯수가 복수 개인 경우에는 가장 크기가 큰 두 개의 영역중에서 자취정보와 일치하

는 하나의 영역을 선택하여 이에 대한 추적작업을 수행한다. 이러한 상황이 발생하는 이유는 첫째, 하나의 이동물체가 두 개의 분리된 영역을 생성하는 경우와 둘째, 이동물체의 갯수가 복수 개인 경우이다. 그러나, 두 시점간의 간격 Δt 가 충분히 작고 물체의 크기가 어느 정도 크면 실세계의 물체 중에서 첫번째 경우를 만족하는 물체는 거의 없다. 따라서, 본 논문에서는 복수 개의 동작영역은 복수 개의 이동물체에 의하여 형성되는 경우로 간주한다. 두 물체중에서 어떠한 물체를 추적대상으로 할 것인가에 대한 결정은 바로 전의 추적단계에서 확인된 물체의 이동속도 (α_x, α_y)와 서보제어를 위하여 소요된 시간에 기초하여 물체의 이동위치를 예측한 후, 이 예측된 위치를 포함하는 영역 또는 예측된 위치에 가장 가까운 동작영역을 추적대상으로 결정한다. 예측된 위치 ($x_p(t), y_p(t)$)를 산출하는 식은 다음과 같다.

$$\begin{aligned} x_p(t) &= v_x(t-1) \cdot (S_t + P_t) \\ y_p(t) &= v_y(t-1) \cdot (S_t + P_t) \end{aligned} \quad (2)$$

식 (2)에서 ($v_x(t-1), v_y(t-1)$)은 전단계 추적(previous tracking)에서 추출된 물체의 이동속도를 나타내고, S_t 와 P_t 는 전단계 추적에서 서보제어에 소요된 시간과 현단계 추적(current tracking)에서 지금까지 소요된 시간을 나타낸다. 이동물체의 예측위치를 나타내는 식이 식 (2)에서와 같은 단순한 형태, 즉 속도 곱하기 시간의 형태로 표현될 수 있는 이유는 본 논문에서 제안하는 추적시스템은 항상 현재 추적 중인 물체를 영상프레임의 중앙에 위치하도록 하기 때문이다.

추출된 동작영역의 갯수가 '1'인 경우 또는 예측정보에 의하여 추적대상이 되는 동작영역이 확인된 경우에는 $D(x, y; t_1-t_2)$ 에서 추출된 동작영역과 $D(x, y; t_2-t_3)$ 에서 추출된 동작영역이 공유지역(common area)을 갖는 경우와 공유지역을 갖지 않는 경우로 세분화한다. 그림 7은 동작영역들이 공유지역을 포함하는 예로서, 가선으로 표현된 직사각형은 t_1 에서의 물체의 위치를 나타내고, 굵은선으로 표현된 직사각형은 t_2 , 그리고 점선으로 표현된 직사각형은 t_3 에서의 물체의 위치를 나타낸다. 즉, $D(x, y; t_1-t_2)$ 에서 추출된 동작영역을 대표하는 최소포함사각형 $MER(t_1, t_2)$ 과 $D(x, y; t_2-t_3)$ 에서 추출된 동작영역을 대표하는 최소포함사각형 $MER(t_2, t_3)$ 은 공유지역을 포함한다. 그림 7에서와 같이 추출된 두 동작영역이 공유지역을 포함하는 경우는 두 동작영역이 동일한 물체의 이동에 의하여 형성된 것으로 간주한다. 이는 시점간의 간격 Δt 가 충분히 작다는 전제

하에서는 이동물체에 의하여 형성된 영역들은 반드시 공유지역을 포함하기 때문이다. $MER(t_1, t_2)$ 의 좌표를 $(Start_{x1}, Start_{y1})$ 와 (End_{x1}, End_{y1}) 이라 하고 $MER(t_2, t_3)$ 의 좌표를 $(Start_{x2}, Start_{y2})$ 와 (End_{x2}, End_{y2}) 라 하면, 물체의 이동속도는 식 (3)에서와 같이 두 최소포함사각형의 이동거리를 시간으로 나눈 값으로 정의된다.

$$\begin{aligned} v_x &= (Ch_{x1} + Ch_{x2}) / (2 \cdot \Delta t) & v_y &= (Ch_{y1} + Ch_{y2}) / (2 \cdot \Delta t) \\ Ch_{x1} &= Start_{x2} - Start_{x1} & Ch_{x2} &= End_{x2} - End_{x1} \\ Ch_{y1} &= Start_{y2} - Start_{y1} & Ch_{y2} &= End_{y2} - End_{y1} \end{aligned} \quad (3)$$

또한, 이동물체의 위치 및 중심위치는 식 (4)를 적용하여 산출된다.

$$\begin{aligned} x_o &= (End_{x_o} + Start_{x_o}) / 2 \\ y_o &= (End_{y_o} + Start_{y_o}) / 2 \\ Start_{x_o} &= Start_{x_2} - Ch_{x_2} & Start_{y_o} &= Start_{y_2} - Ch_{y_2} \\ End_{x_o} &= End_{x_2} & End_{y_o} &= End_{y_2} \end{aligned} \quad (4)$$

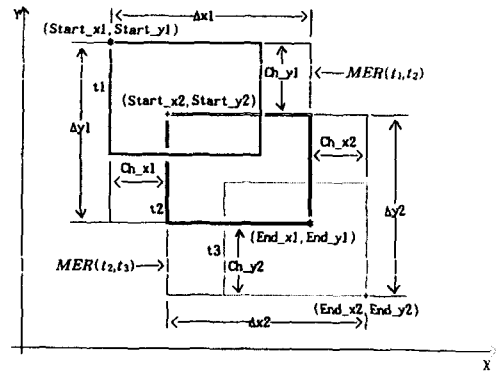


그림 7. 두 동작영역이 공유지역을 포함하는 경우
Fig. 7. Case where two nonstationary areas contain common area.

그림 8에서와 같이 $D(x, y; t_1-t_2)$ 에서 추출된 동작영역과 $D(x, y; t_2-t_3)$ 에서 추출된 동작영역이 공유지역(common area)을 갖지 않는 경우는 두 동작영역이 상이한 물체의 이동에 의하여 형성된 것으로 간주한다. 즉, 현재 추적중인 물체이외에 새로운 물체의 이동이 발생한 경우로 간주한다. 이러한 상황에서는 추적하던 물체를 계속 추적하기 위하여 식 (2)를 활용한다. 즉, 바로 전의 추적단계에서 확인된 물체의 이동속도 (v_x, v_y)와 서보제어를 위하여 소요된 시간에 기초하여 물체의 이동위치를 예측한 후, 이 예측된 위치를 포함하는 영역 또는 예측된 위치에 가장

가까운 동작영역을 추적대상으로 선택한다. 물체의 이동속도, 위치 및 중심위치는 선택된 동작영역에 식 (1)을 적용하여 산출된다.

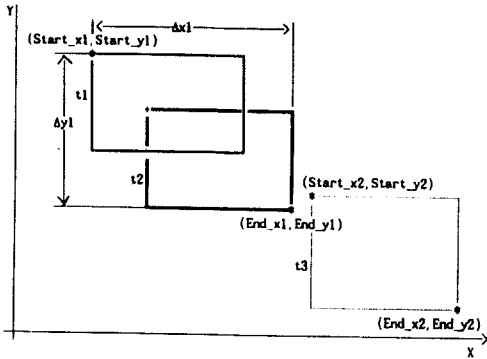


그림 8. 두 동작영역이 공유지역을 포함하는 않는 경우
Fig. 8. Case where 2 nonstationary areas do not contain common area.

Ⅶ. 서보기기 제어에 의한 이동물체 추적

산출된 동작정보, 즉 이동속도 (α_x, α_y)와 중심위치 (x_o, y_o)를 이용하여 이동물체에 대한 추적작업을 수행하기 위하여는 서보기기에 적당한 제어신호를 보내야 한다. 본 논문에서는 현재 추적 중인 물체가 영상 프레임의 중앙에 위치하도록 서보기기를 제어한다. 그림 9에서와 같이 영상프레임 좌표계의 원점을 (x_c, y_c), 이동물체의 중심좌표를 (x_o, y_o), 그리고 렌즈의 초점거리(focal length)를 f 라 하자.

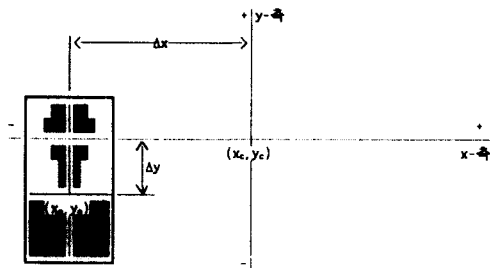


그림 9. ($\Delta x, \Delta y$)에 의한 제어신호 추출
Fig. 9. Extraction of control signal based on ($\Delta x, \Delta y$).

서보기기에 대한 제어는 수직방향의 이동정도를 나타내는 수직각(tilt angle δ)과 수평방향의 이동정도를 나타내는 수평각(panning angle θ)에 의하여 정

의될 수 있다. 그림 9에서 이동물체의 중심을 영상프레임 좌표계의 원점으로 이동하기 위하여는, 즉 x축으로 x 그리고 y축으로 y 만큼 이동하기 위하여 필요한 수평각과 수직각은 식 (5)과 식 (6)에 의하여 산출된다.

$$\Delta x = x_c - x_o$$

$$\theta = \tan^{-1}(\Delta x / f)$$

if ($\theta > 0$) Pan_Right(abs(θ))
if ($\theta < 0$) Pan_Left(abs(θ))

$$\Delta y = y_c - y_o$$

$$\delta = \tan^{-1}(\Delta y / f)$$

if ($\delta > 0$) Tilt_Up(abs(δ))
if ($\delta < 0$) Tilt_Down(abs(δ))

식 (5)과 식 (6)에 의하여 수직각과 수평각은 상호 독립적으로 산출될 수 있다. 그러나, 상품화되어 있는 대부분의 서보기기들은 그림 10에서와 같이 8 방향으로 제어가 가능하다. 즉, 수직 및 수평방향뿐 아니라 대각방향으로도 제어될 수 있다. 단, 대각방향으로 서보기기를 제어하기 위하여는 수직각과 수평각의 절대값이 같아야 한다. 표 1은 Δx 와 Δy 의 크기 및 부호에 따른 각 방향으로의 제어량을 나타낸다. 예를 들어, Δx 는 음수이고 Δy 는 양수이며 $|\Delta x|$ 가 $|\Delta y|$ 보다 큰 경우에는 서북방향으로 $|\alpha|$ 만큼 Tilt_up&Pan_left를 수행한 후 $|\theta|$ 만큼 Pan_left를 수행한다.

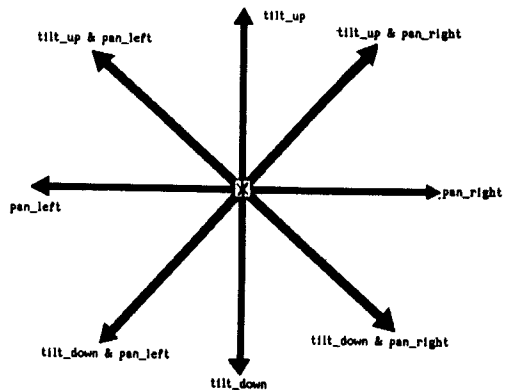


그림 10. 서보기기의 제어방향
Fig 10. Control orientation of servo equipment.

일반적으로 추적중인 물체는 제어신호를 산출하기 위하여 소요되는 시간중에도 이동을 계속한다. 따라

서, 표 1에서 사용된 Δx 와 Δy 는 이동물체의 중심위치 (x_0, y_0) 에 대한 정보뿐 아니라, 물체의 이동속도와 처리시간 그리고 서보기기의 성능을 고려하여 산출되어야 한다. 식 (7)은 물체의 이동속도와 처리시간 그리고 서보기기의 성능을 고려하여 f_0^* 변위(Δx 또는 Δy)를 계산하는 과정을 보여준다.

$$f_0^* = a \cdot t^* + b \tag{7}$$

표 1. x와 y에 의한 8방향 제어신호
Table 1. 8 directional control signals based on x and y.

Δx	Δy	$ \Delta x ? \Delta y $	제어 각도	제어 신호
움 (-)	움 (-)	<	$\alpha = \tan^{-1}(\Delta x/f)$ $\delta = \tan^{-1}((\Delta y - \Delta x)/f)$	Pan_Left_Tilt_Down(α) Tilt_Down(δ)
		>	$\alpha = \tan^{-1}(\Delta y/f)$ $\theta = \tan^{-1}((\Delta x - \Delta y)/f)$	Pan_Left_Tilt_Down(α) Pan_Left(θ)
움 (-)	앙 (+)	<	$\alpha = \tan^{-1}(\Delta x/f)$ $\delta = \tan^{-1}((\Delta y - \Delta x)/f)$	Pan_Left_Tilt_Up(α) Tilt_Up(δ)
		>	$\alpha = \tan^{-1}(\Delta y/f)$ $\theta = \tan^{-1}((\Delta x - \Delta y)/f)$	Pan_Left_Tilt_Up(α) Pan_Left(θ)
앙 (+)	움 (-)	<	$\alpha = \tan^{-1}(\Delta x/f)$ $\delta = \tan^{-1}((\Delta y - \Delta x)/f)$	Pan_Right_Tilt_Down(α) Tilt_Down(δ)
		>	$\alpha = \tan^{-1}(\Delta y/f)$ $\theta = \tan^{-1}((\Delta x - \Delta y)/f)$	Pan_Right_Tilt_Down(α) Pan_Right(θ)
앙 (+)	앙 (+)	<	$\alpha = \tan^{-1}(\Delta x/f)$ $\delta = \tan^{-1}((\Delta y - \Delta x)/f)$	Pan_Right_Tilt_Up(α) Tilt_Up(δ)
		>	$\alpha = \tan^{-1}(\Delta y/f)$ $\theta = \tan^{-1}((\Delta x - \Delta y)/f)$	Pan_Right_Tilt_Up(α) Pan_Right(θ)

식 (7)에서 a는 동작정보 추출단계에서 산출된 물체의 이동속도 (v_x 또는 v_y)를 나타내고, t^* 는 서보기기가 물체의 추적을 위하여 움직이는데 소요되는 시간을 나타내며, b는 이동속도 (v_x 또는 v_y)에 제어 신호를 산출하기 위하여 소요되는 처리시간 (P_t)을 곱한 값이다. 즉, 처리시간 동안에 예상되는 물체의 이동거리와 서보기기가 이동되는 동안에 예상되는 물체의 이동거리를 합한 값을 변위(x 또는 y)로 사용한다. 그림 11은 서보기기가 물체의 추적을 위하여 움직이는데 소요되는 시간 t^* 를 산출하는 과정을 보여준다.

그림 11에서 s는 서보기기의 이동속도를 나타내고, $f_s = s \cdot t$ 는 시간 t 동안에 서보기기가 이동하는 거리를 나타낸다. 현재 추적 중인 물체가 영상프레임의 중앙에 위치하도록 하기 위해서는 서보기기는 f_0^* 만큼 이동되어야 하므로, 즉 f_s 와 f_0^* 는 같은 값을 가져야 하므로, 서보기기가 물체의 추적을 위하여 움직이는데 소요되는 시간 t^* 는 식 (8)에 의하여 구해진다.

$$t^* = b / (s - a) \tag{8}$$

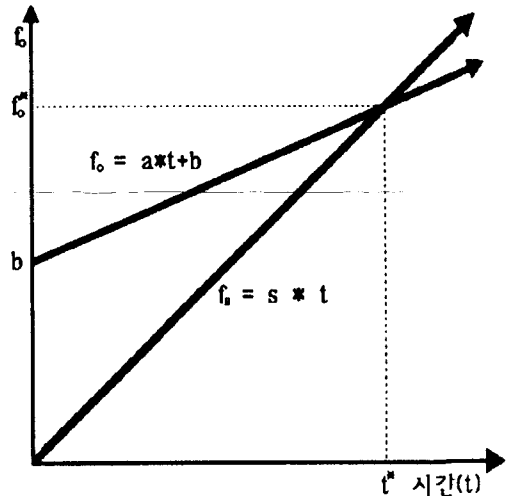


그림 11. 물체의 속도, 처리시간 그리고 서보기기 성능을 고려한 변위 계산

Fig. 11. Displacement calculation based on object speed, processing time, and servo speed.

VII. 실험결과 및 결론

이 절에서는 본 논문에서 구현한 실시간 이동물체 추적시스템의 동작원리를 설명하는 실험결과를 보이고 향후 연구방향에 대하여 기술한다. 추적시스템의 동작원리를 설명하는 실험은 주어진 입력영상에서 분석에 사용될 런(run)들을 추출하는 과정, 추출된 런들을 분석하여 현재 추적중인 물체의 위치 및 속도를 확인하는 과정, 그리고 확인된 물체를 카메라의 중심 위치로 이동시키는 서보기기의 제어과정을 중심으로 기술한다. 실험에 사용된 컴퓨터는 IBM-PC/386으로 CPU 속도는 56MHz이다. 이동물체에 대한 연속적인 입력영상들은 고정된 비디오 카메라(video camera)를 통하여 오프라인(offline)으로 1/30초 간격으로 촬영되었다. 영상입출력을 위하여 DT-2853 Frame Grabber를 사용했으며, 영상의 크기는 512 × 512이고 한 화소가 나타낼수 있는 밝기의 종류는 256이다. 구현을 위하여 사용한 언어는 Microsoft C이다. 사용된 서보기기는 국제 PAN&TILT로 초당 6도를 변화시킬 수 있다. 서보기기를 제어하는 구동부는 8개의 계전기(relay)를 이용하여 구축되었다. 즉, 8개의 계전기에 의하여 병렬포트(parallel port)의 각 단자와 서보기기의 각 단자를 연결하였으며, 각 계전기는 계산된 제어시간 동안 온-상태(on

state)를 유지하도록 하였다.

그림 12는 야외에서 촬영한 연속적으로 입력된 영상의 예이다. 시간이 경과하면서 좌측에서 우측으로 물체의 이동이 발생하고 있으며(t_0-t_{22}), 또한 일정 시간이 지난 후에는 우측에서 좌측으로 물체의 이동 역시 발생하고 있다($t_{18}-t_{22}$).

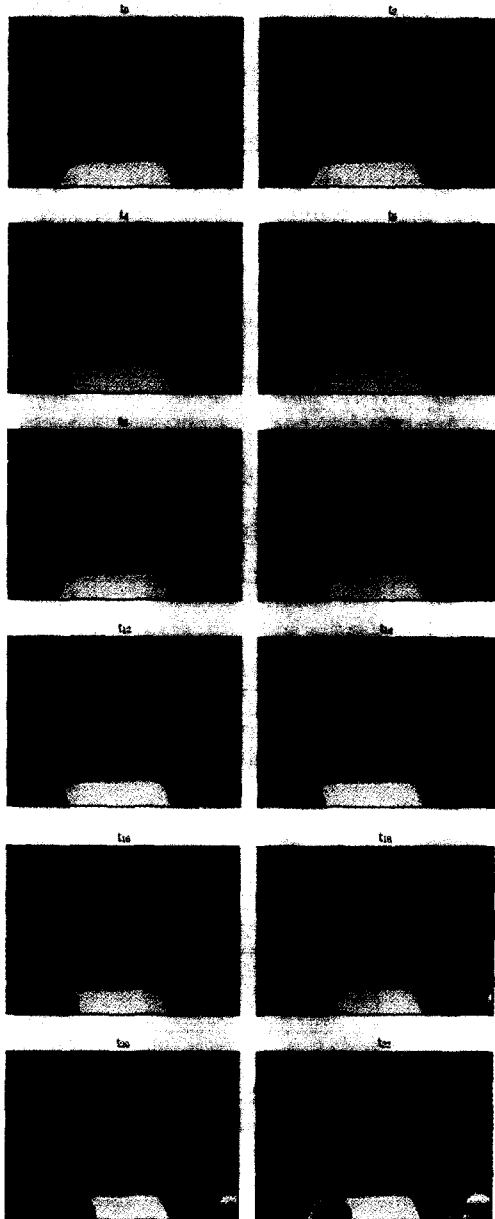


그림 12. 연속적인 입력영상의 예
Fig. 12. Example of image sequence.

그림 13은 t_0 시점의 영상과 t_1 시점의 영상을 사용하여 동기화 처리결과이다. 영상의 상단에서부터 시

작하여 5개의 런 R_1, R_2, R_3, R_4 , 그리고 R_5 의 위치를 흰선으로 표시하였다. 이 경우는 모든 런이 비동작런으로 판명되었다. 즉, 물체의 진입이 없거나 시간간격이 너무 적은 경우이다.

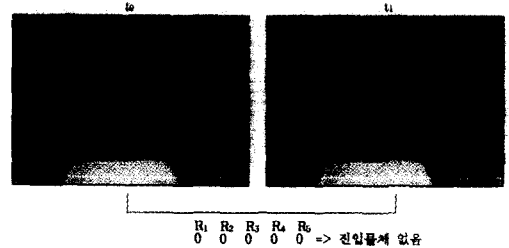


그림 13. 동기화 처리기의 실행 예
Fig. 13. Example of synchronous processor.

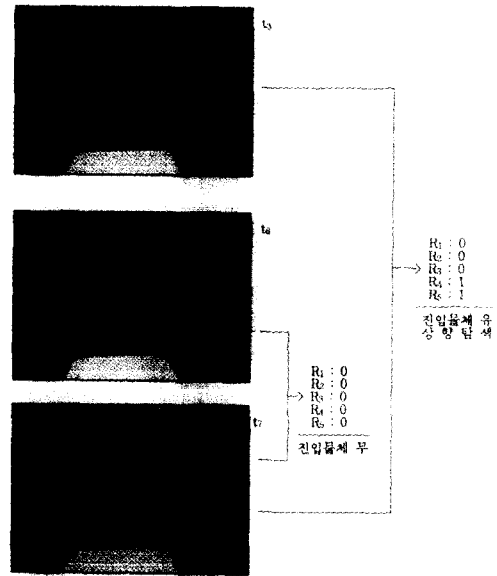


그림 14. 다속도 처리기의 실행 예
Fig. 14. Example of multi-speed processor.

그림 14는 다속도 처리기가 시간간격을 조정한 예이다. 이제 시간간격은 $4 \cdot \Delta t_{init}$ 로 조절되었다. 즉, t_3 에서 촬영된 영상과 t_7 에서 촬영된 영상을 비교분석하여 최하단에 위치한 런(R_5)이 동작런임을 확인하였다. 따라서, 동기화 처리기는 시간간격을 $4 \cdot \Delta t_{init}$ 로 탐색방향은 상향탐색으로 확정한 후 동작정보 추출기를 호출한다.

그림 15는 t_7, t_{11}, t_{15} 의 영상을 상향탐색한 결과이다. 이 결과로부터 추출된 속도정보는 $\Delta x =$

6.25(pixel)와 $\Delta y = 2.50(\text{pixel})$ 이고 중심위치는 $c_x = 110$ 와 $c_y = 329(\text{Pixel})$ 이다. 본 논문에서는 초점거리(f)의 계산을 위하여 참고문헌 [21]에서 제안한 방법을 사용하였으며, 본 실험영상에서는 $f = 54.23 \text{ mm}$ 의 고정된 값을 사용하였다. 계산된 물체

의 속도와 처리시간 그리고 서보기기의 성능을 이용하여 서보제어각도는 구해진다. 그림 15의 영상에 대하여 구해진 서보제어각도는 Pan_Left Tilt_Down = 0.42 (degree)와 Pan_Right = 0.07(degree) 이다.

그림 16에서는 앞에서 추출한 물체의 위치와 속도 정보를 이용하여 1/30초 후에 예측되는 위치와 1/30초 후의 영상과 겹친영상을 보인 것이다. 이 예에서는 물체가 수평이동하고 있으므로 예측된 영역과 추적대상이 되는 영역이 정확히 일치하고 있음을 알 수있다.

그림 17은 복수개의 물체가 나타난 경우이다. 즉, 복수개의 동작영역이 추출되는 경우이다. 이 경우 추적대상이 되는 동작영역을 선택하기 위하여 자취정보를 이용한다.

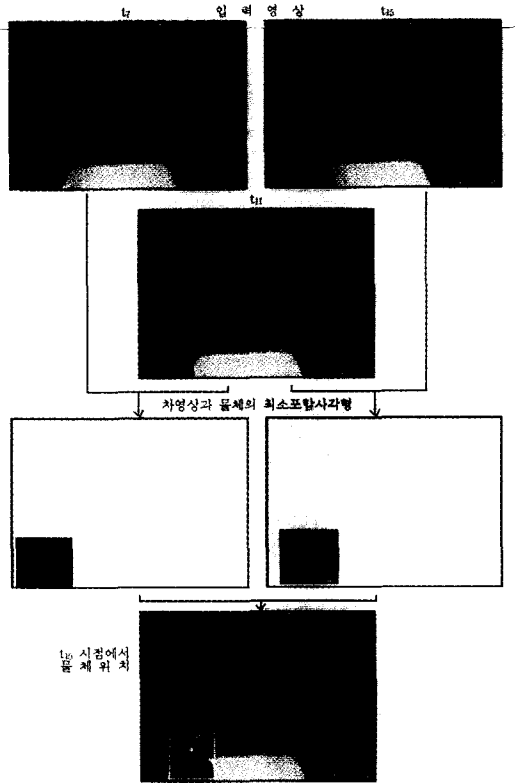


그림 15. 상향탐색의 결과
Fig. 15. Result of bottom-up search.

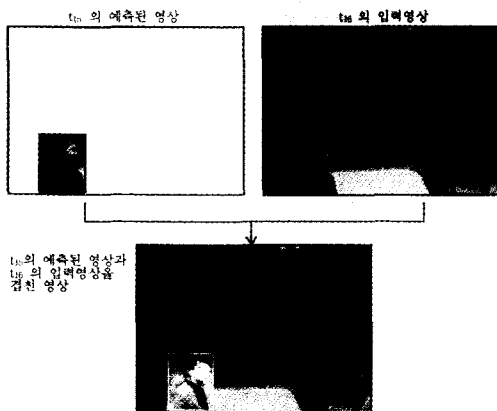


그림 16. 예측된 영상과 실영상 및 겹쳐진 영상
Fig. 16. Predicted image, input image, and overlapped image.

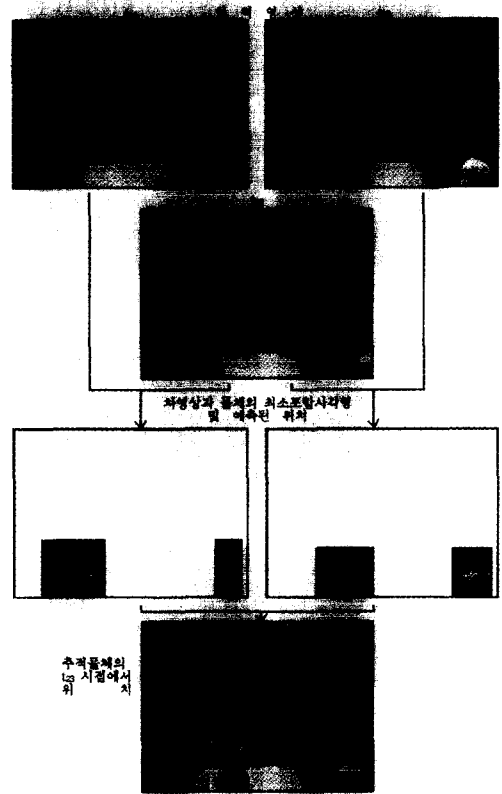


그림 17. 복수개의 물체와 예측된 위치
Fig. 17. Many objects and predicted position.

앞에서 추출한 물체속도 $\Delta x = 6.25(\text{pixel})$ 와 $\Delta y = 5.00(\text{pixel})$ 로부터 예측된 위치는 (+)이다. 따라서, 추적물체로 선택되는 동작영역은 (+)를 포함하는 동작영역이다.

본 논문의 구현환경에서 동기화 처리에 소요된 시

간은 약 0.17(sec)이고 동작영역 탐지 및 동작정보 추출을 위하여 소요된 시간은 약 0.55(sec)이다. 이 시간들은 영상정보를 영상획득기(frame grabber)에서 PC의 주기억장치로 전이하는 시간을 포함한 것이다. 따라서, 전이시간을 줄이기 위하여 영상획득기와 처리기가 같은 보드에 있을 경우 이 시간들은 약 0.06(sec)와 0.12(sec)로 현저히 줄어들어 빠르게 뛰어나는 물체도 추적할 수 있을 것으로 생각한다.

실시간 이동물체 추적시스템의 성능은 CPU의 처리시간 및 서보기기의 성능과 상관관계가 있다. 또한, 서보기기가 추적할 수 있는 물체의 속도는 카메라에서 물체까지의 상대거리에 영향을 받는다. 즉, 이동물체가 카메라에서 멀리 떨어져 있으면, 이 물체를 추적하기 위하여 필요한 서보제어각은 상대적으로 작을 값이 갖게 될 것이다. 식 (9)은 물체가 광축에 직교하는 방향으로 움직이는 경우에 물체의 속도, 물체와 카메라의 거리 및 서보기기의 성능과의 관계를 보여준다.

$$\theta \geq \tan^{-1}(O_s / O_d) \quad (9)$$

여기서, θ 는 서보기기가 단위시간당 움직일 수 있는 각도이고, O_s 는 물체의 속도이고, O_d 는 카메라에서 물체까지의 거리이다. 예를 들면, 초당 6도를 변화시킬 수 있는 서보기기로 10m 거리에 있는 물체를 추적할 때 추적할 수 있는 물체의 속도한계는 0.95m/sec이다. 이 속도는 성인의 보통의 보속이다. 현재의 기술로 가장 빠른 서보기기는 초당 30도를 변경할 수 있는 것으로 알려져 있다. 이 서보기기로 100m를 10초에 뛰는 사람을 추적하기 위해서는 카메라에서 물체까지의 거리를 약 17.32m 이상을 유지해야 한다. 처리시간(a)을 고려할 때 요구되는 서보기기의 성능은 다음 식 (10)과 같다.

$$\theta \geq \tan^{-1}((O_s * \alpha + O_s) / O_d) \quad (10)$$

초당 30도를 변경할 수 있는 서보기기로 100m를 10초에 뛰는 사람을 추적하기 위하여 처리시간이 0.3초 소요되면 카메라에서 물체까지의 거리는 약 21.52m 이상을 유지해야 한다. 서보기기의 정확성은 직류 스텝핑(DC stepping)으로 구성되어 있는 경우 오차율은 약 1/10⁶ 정도로 비교적 정확하다.²²

본 논문에서는 연속적으로 입력되는 영상에서 이동물체에 대한 정보를 추출하고 카메라의 시계를 변경시키는 서보기기를 제어하여 이동물체에 대한 추적작업을 실시간에 수행하는 시스템에 대하여 기술하였다.

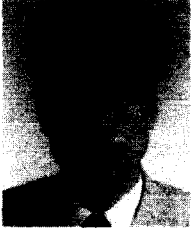
본 논문에서 제안하는 방법의 효율성을 높이기 위하여는 차 영상에 대한 이진화 작업을 환경변화에 적응적으로 대처하는 방안에 대한 연구가 요구되며 처리시간을 향상시키기 위하여 영상획득기와 처리기를 같은 보드에 두는 하드웨어 작업과 보다 빠른 처리를 위하여 병렬적으로 처리하는 연구가 요구된다.

參 考 文 獻

- [1] Alton L. Gilbert, Michal K. Giles, Gerald M. Flachs, Robert B. Rogers and Yee Hsun U, "A real-time video tracking system," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, PAMI-2, no. 1, pp. 47-56, January 1980.
- [2] R. Schalkoff and E. S. Mcvey . "A model and tracking algorithm for a class of video targets," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, PAMI-4, no. 1, pp. 2-10, 1982.
- [3] N. Hoose and L. G. Willumsen, "Automatically extracting traffic data from video tape using the CLIP4 parallel image processor," *Pattern Recognition Letters*, Vol. 6, pp. 199-213, 1987.
- [4] Toshinari Nonaka, Shigeru Kimura, and Shinji Ojawa, "A moving object extraction using adapted frame integration," *First Korea-Japan Joint Conference on Computer Vision*, pp. 186-191, 1991.
- [5] N. Moghadam Charkari and Hideo Mori, "A new approach for real time moving vehicle detection," *Proceedings of the 1993 IEEE/RSJ International Confrence on Intelligent Robots and System*, pp. 273-278, 1993.
- [6] T. Uno, M. Ejiri, and T. Tokunaga, "A method of real-time recognition of moving objects and its application," *Pattren Recognition*, Vol. 8, pp. 201-208, 1976.
- [7] W. N. Martin and J. K. Aggarwal,

- "Survey : dynamic scene analysis." *Computer Graphics and Image Processing*. Vol. 7, pp. 356-374, 1978.
- [8] David W. Murray, Philip F. McLauchlan, Ian D. Reid, and Paul M. Sharkey. "Reaction to Peripheral Image Motion using a Head/Eye Platform." *Fourth International Conference on Computer Vision*, pp. 403-411, 1993.
- [9] William B. Thompson, Pamela Leshleider, and Elizabeth R. Stuck. "Detecting moving objects using the rigidity constraint." *IEEE Transaction on Pattern Analysis and Machine Intelligence*, PAMI-15, No. 2, pp. 162-166, 1993.
- [10] Vincent S. S. Hwang. "Tracking feature points in time-varying images using an opportunistic selection approach." *Pattern Recognition*, Vol. 22, No. 3, pp. 247-256, 1989.
- [11] W. K. Chow and J. K. Aggarwal. "Computer analysis of planar curvilinear moving images." *IEEE Transaction Computer*, C-26, pp. 179-185, 1977.
- [12] Hsi Jian, Lung Fa Huang, and Z. Chen. "Multi-frame ship detection and tracking in a infrared image sequence." *Pattern Recognition*, Vol. 23, No. 7, pp. 785-798, 1990.
- [13] Berthold K. P. Horn and Brian G. Schunck. "Determining optical flow." *Artificial Intelligence*, Vol. 17, pp. 481-493, 1981.
- [14] A. Rognone, M. Campani, and A. Veri. "Detecting moving objects from optical Flow." *Pattern Recognition and Image Analysis*, Vol. 2, No. 1, pp. 109-111, 1991.
- [15] 김이한, 김성대. "새로운 공간경사를 사용한 시공간 경사법에 의한 운동경계 검출 및 이동 벡터 추정." *대한전자공학회 논문지 제 30권 B편, 제 2호*, pp. 157-166, 1993
- [16] R. Jain, D. Militer, and H. H. Nagel. "Separating non-stationary from stationary scene components in a sequence of real world TV-images." *Proc. 5th Int. Joint Conf. Artificial Intelligence*, pp. 612-618, 1977.
- [17] Ramesh Jain. "Extarction of motion information from peripheral processes." *IEEE Transaction on Pattern Analysis and Machine Intelligence*, PAMI-3, no. 5, pp. 489-503, September, 1981.
- [18] Liang-Hwa Chen and Shyang Chang. "A video tracking system with adaptive predictors." *Pattern Recognition*, Vol. 25, No. 10, pp. 1171-1180, 1992.
- [19] 황신환, 이성욱. "자동 영상 추적기 개발 연구." *대한전자공학회 논문지 제 25권 제 4호*, pp. 437-446, 1988
- [20] Hironobu Takahashi and Fumiaki Tomita. "Fast Region Labeling with Boundary Tracking." *IEEE ICIP '89*, pp. 369-373.
- [21] Robert M. Haralick. "Determining camera parameters from the perspective projection of a rectangle." *Pattern Recognition*, Vol. 22, No. 3, pp. 225-230, 1989.
- [22] Yasuhiko Dote. *Servo Motor and Motion Control Using Digital Signal Processors*, Prentice Hall, 4, 1990.

著 者 紹 介



金 啓 榮(正會員)

1964年 10月 1日生. 1990年 2月
송실대학교 전자계산학과 졸업(공학사). 1992年 2月 송실대학교 대학원 전자계산학과 졸업(공학석사). 1992年 8月 ~ 현재 송실대학교 대학원 전자계산학과 박사과정 재학중.

주관심 분야는 컴퓨터 비전, 영상처리, 동작이해, 무인 자동차 등임.



崔 澐 一(終身會員)

1956年 8月 16日生. 1979年 2月
연세대학교 전자공학과 졸업(공학사). 1982年 6月 미시간대학교 전산공학과 졸업(공학석사). 1987年 6月 미시간대학교 전산공학과 졸업(공학박사). 현재 송실대학교 전자계산학과 부교수 재직중.

주관심 분야는 컴퓨터 비전, 퍼지 및 신경망이론, 패턴인식, 지식기반시스템 등임.