
객체추적을 위한 적응적 정합 블록을 이용한 블록정합 알고리즘

김진태* · 안수홍** · 오정수***

Block Matching Algorithm Using an Adaptive Matching Block for Object Tracking

Jin-tea Kim* · Soo-hong Ahn** · Jeong-su Oh***

이 논문은 2010년도 한서대학교 교비 학술연구 지원 사업에 의하여 연구되었음

요 약

블록정합 기법을 이용한 객체추적에서 크기가 다양하고 수시로 변하는 객체를 추적하기 위해 고정 정합블록을 사용하는 것은 적합하지 못하다. 본 논문은 동적 환경을 위한 적응적 정합블록을 정의하고, 이를 위한 블록정합 알고리즘을 제안한다. 정합블록은 42×42 화소의 넓은 영역에 10×10 화소의 주 블록과 6×6 화소의 부 블록 8개로 구성되고, 영역 중심에 위치한 주 블록은 객체 블록으로 사용되고, 영역의 외곽에 위치한 부 블록은 객체블록을 위한 후보 블록으로 사용된다. 제안된 알고리즘은 부 블록에서 이전 10 프레임의 움직임 벡터를 이용해 객체블록을 추출하고, 주 블록과 추출된 객체블록을 이용해 블록정합을 수행한다. 성능 평가를 위한 실험들은 제안된 알고리즘이 정합블록에서 유효한 객체블록만을 적절히 추출하고, 자유로운 움직임을 갖는 객체를 영상의 중심 영역에 유지시켜 주는 것을 보여주고 있다.

ABSTRACT

In object tracking using the block mating algorithm, it is not proper to use a fixed matching block to track an object of which size may be various and can be changed at any time. This paper defines an adaptive matching block for the dynamic environment and proposes a block matching algorithm for it. The matching block is composed of a main-block of 10×10 pixels and 8 sub-blocks of 6×6 pixels in a wide area of 42×42 pixels, the main-block located its center is used as an object block, and the sub-blocks located its boundary are used as candidates for the object block. The proposed algorithm extracts the object blocks from the sub-blocks by using their motion vectors for 10 previous frames and performs the block matching with the main block and them. The experiments for perform estimation show that the proposed algorithm extracts just valid object blocks from the matching block and keeps an object having free movement in image center area.

키워드

적응적 정합블록, 블록정합, 객체추적, 움직임벡터

Key word

adaptive matching block, block matching, object tracking, motion vector

* 정회원 : 한서대학교 컴퓨터정보공학과

** 정회원 : 코아로직

*** 종신회원 : 부경대학교 이미지시스템공학과(ojs@pknu.ac.kr)

접수일자 : 2010. 09. 29

심사완료일자 : 2010. 10. 13

I. 서 론

컴퓨터, 네트워크 기술의 발전에 의해 다양한 영역에서 CCD 카메라가 활용되고 있다. 그러나 CCD 카메라들은 시계(field of view, FOV)가 제한되어 움직이는 객체들은 짧은 시간 내에 카메라의 시계 즉 취득 영상에서 벗어날 수 있고, 감시자/관찰자로서 인간은 집중력과 주의력의 한계로 지속적인 객체 감시/관찰이 어렵다. 그래서 CCD 카메라들은 정해진 패턴을 따라 움직여 넓은 시계를 확보하거나 추가적인 다수 감시자/관찰자와 장비를 투입하고 있고, 최근에는 지능형 감시카메라 시스템에 관한 연구가 진행되고 있다[1-4].

지능형 감시 카메라 시스템은 움직이는 객체의 위치를 추출하고, 위치에 따른 카메라 제어신호를 생성하여 카메라 시선을 움직이는 객체를 따라가므로 카메라 시계 제한을 거의 받지 않고, 객체를 카메라 시계의 중심에 유지시켜 주므로 감시자/관찰자의 한계를 해소할 수 있다. 객체추적은 프레임 단위로 움직이는 객체의 위치를 검출하는 것으로 거리 벡터에 대한 통계값을 이용하는 칼만필터(Kalman filter) 기법, 현 프레임 화소들을 다음 프레임의 8 방향 화소들과 비교하여 가장 비슷한 영역을 정하여 추적하는 광학흐름(optical flow) 기법, 현 프레임의 일정 블록이 다음 프레임의 일정 영역 내에서 가장 유사한 블록을 찾아 추적하는 블록정합(block matching) 기법이 있다[5-7].

본 논문에서는 블록정합 기법을 이용한 객체추적에서 고정 크기 정합블록(matching block)의 장단점을 분석하고, 고정 크기 정합블록이 갖는 문제를 해결하기 위한 적응적인 정합블록을 정의하고, 이에 적합한 블록정합 알고리즘을 제안한다. 정합블록은 중심에 항상 객체로 인식되는 작은 블록과 8 근방에 객체가 될 수 있는 더 작은 블록들로 42×42의 넓은 영역에 구성되고, 블록정합은 객체로 인식된 블록들만으로 수행된다. 성능 평가 실험을 통해 제안된 알고리즘은 정합블록에서 배경블록을 제거하고 객체블록만을 적절히 선택하고, 제안된 알고리즘과 팬/틸트 카메라 제어를 통해 자유롭게 움직이는 객체를 영상의 중심에 유지시켜 주는 것을 확인할 수 있다.

II. 객체추적을 위한 블록정합 기법

본 장에서는 객체추적을 위한 블록정합을 기술하고, 정합블록 크기가 객체추적에 주는 영향을 살펴본다.

2.1 블록정합 기법

블록정합 기법은 동영상 압축에서 널리 사용되는 기법으로 현 프레임의 정합블록에 대해 이전 프레임의 지정된 영역 내에서 가장 유사한 블록을 찾는 것이다. 반면 객체추적에서는 현 프레임의 객체로 지정된 정합블록에 대해 다음 프레임의 지정된 영역 내에서 가장 유사한 블록을 찾아 객체의 움직임을 나타내는 것이다. 여기서 유사 블록을 찾는 지정된 영역을 탐색영역(search range), 정합블록과 유사블록 사이의 변위를 움직임 벡터(motion vector)라 한다. 블록간 유사도는 SAD (Sum of abstract difference), MAD(Mean absolute difference), MSD(Mean square difference) 등이 사용되며 본 논문에서는 크기가 다른 블록들이 정합에 사용되어 식 (1)의 MAD을 사용한다.

$$MAD = \frac{1}{n^2} \sum_{i=1}^n \sum_{j=1}^n |B_f(i,j) - B_{f+1}(x+i,y+j)| \quad (1)$$

여기서 n 은 블록의 크기이고, B_f 과 B_{f+1} 는 각각 현 프레임의 정합블록과 다음 프레임의 후보 유사블록이고, (x,y) 는 정합블록과 후보 유사블록 사이의 변위이다. 본 논문에서 블록정합의 계산량을 줄이기 위해 PDE (partial distortion elimination)를 이용한 나선형 탐색(spiral search)을 수행한다[8].

2.2 정합블록 크기의 영향

동영상 압축에서 이상적인 블록정합을 위한 전체 조건이 정합블록의 모든 화소는 동일한 움직임을 갖는 것이고, 이는 정합블록이 하나의 객체로 구성됨을 의미한다. 그래서 동영상 압축에서 전체 조건과 블록정합을 위한 계산량, 부가정보 등을 고려하여 16×16 화소의 블록을 보편적으로 사용한다. 한편, 객체추적에서는 추적하고자 하는 객체 크기가 다양하게 발생할 수 있고, 수시로 변할 수 있다. 그래서 이상적인 객체추적을 위해 정합블록은 객체 크기에 맞추어 적응적으로 결정되어야 하지만 이는 현실적으로 어렵다. 효율적인 객체추적을 위해

정합블록 크기에 따른 블록정합과 객체추적에 대한 영향을 확인한다.

그림 1은 큰 객체와 작은 객체에서 큰 정합블록과 작은 정합블록의 관계를 보여주고 있다. 그림 (a)의 큰 객체에서 큰 정합블록은 충분한 객체 정보를 포함할 수 있어 적절한 객체추적이 예상된다. 작은 정합블록은 완전히 객체만을 포함해도 배경과 구분되는 충분한 정보를 갖기 어렵고, 점선으로 표시된 유사 배경이 주변에 포함되면 잘못된 객체추적이 예상된다. 그림 (b)의 작은 객체에서 작은 정합블록은 충분한 객체 정보를 가질 수 있어 적절한 객체추적이 예상된다. 큰 정합블록은 객체보다 큰 배경을 포함하여 배경 추적이 예상된다. 결과적으로 이상적인 정합블록은 가능한 많은 객체를 포함하면서 그 객체가 배경과 구분되는 충분한 정보를 갖고 있어야 한다. 또한 블록 크기가 커질수록 정합을 위한 계산량을 크게 증가하고 실시간 객체추적을 어렵게 한다.

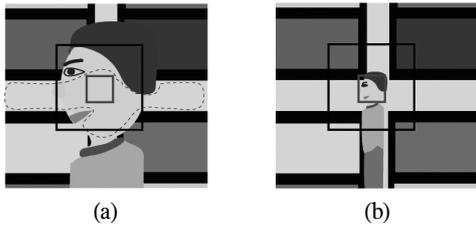


그림 1. 객체와 정합블록의 관계
(a) 큰 객체 (b) 작은 객체

Fig. 1 Relation between objects and matching blocks
(a) Large object (b) Small object

III. 제안된 객체추적 알고리즘

본 장에서는 효율적인 객체추적을 위한 적응적 정합 블록을 이용한 개선된 블록정합 알고리즘과 카메라 제어에 대해 기술한다.

3.1 객체추적을 위한 블록정합

앞에서 언급한 것처럼 정합블록과 객체가 일치할 때 이상적인 객체추적이 이루어진다. 그런데 추적하는 대상 객체에 따라, 객체와 카메라 사이의 거리에 따라 객체 크기는 수시로 변하므로 최적의 객체추적을 위한 적절한 정합블록 크기를 결정하는 것은 어렵다. 따라서 다양한

크기의 객체에 적응적으로 대응하며 효율적으로 객체를 추적할 수 있는 정합블록과 블록정합 알고리즘이 요구된다.

그림 2는 제안된 정합블록으로 42×42 화소의 넓은 영역에 작은 블록들을 그림처럼 배치한다. 중심에 위치한 10×10 화소의 주 블록(B0)은 고정된 객체블록이 되고, 경계 영역에 위치한 6×6 화소의 부 블록들(B1~B8)은 객체를 위한 후보 블록들이다. 넓은 영역에 작은 블록들로 구성되어 큰 객체는 물론 작은 객체에 대응하도록 한다. 주 블록과 객체로 선택된 부 블록이 실질적인 정합블록이 되고, 앞 장에서 제시한 고정 정합블록의 문제를 해결할 수 있다.

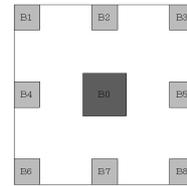


그림 2. 제안된 정합블록들
Fig. 2 Proposed matching blocks

제안된 정합블록을 이용한 블록정합은 그림 3처럼 수행된다. “정합블록”블록에서는 정합을 위한 블록들 B0~B8을 지정한다. 추적을 시작하는 프레임에서는 마우스로 선택한 위치에 주 블록 중심을 두고 주변 블록을 설정하고, 이후 프레임에서는 블록정합의 결과로 결정된 객체 움직임에 따라 정합블록이 설정된다.

“움직임?”블록에서는 spiral PDE의 시작점 정합오차인 MAD(0,0)를 통해 정합블록의 움직임을 판단한다. MAD(0,0)가 임계값(MAD_TH) 이하이면 움직임이 없는 것으로 움직임 벡터는 (0,0)가 된다. 여기서 임계값 MAD_TH은 동적 객체가 없는 영상에서 주변 환경 영향에 의해 순간적으로 변화되는 평균 명암이다. 본 실험 환경에서는 평균 4정도 변화가 발생 하였고, 실험 환경에 따라 다소 차이가 있을 것이다.

“spiral PDE”블록에서는 움직임이 있는 블록들에 대해 순차적으로 spiral PDE를 수행한다.

“움직임 정보”블록에서는 정합블록들과 객체의 움직임 벡터를 결정한다. 블록들의 움직임 벡터는 블록들의 독립된 움직임으로 다음 프레임에서 객체블록을 결정하기 위한 정보로 사용된다. 객체의 움직임 벡터는 이전

프레임에서 결정된 객체블록들로 구성된 객체의 움직임 벡터로 객체의 움직임 정보이고 카메라 제어를 위한 정보로 사용한다. “유사 움직임”블록에서는 부 블록의 움직임 벡터를 통해 움직임 방향 모드를 결정하고, 주 블록과 유사 움직임인지 결정한다. 움직임 방향은 그림 4 처럼 9개 모드를 가지고, 블록의 움직임 방향은 9개 모드 중 가장 인접한 방향을 선택한다. 주 블록의 움직임에 대한 유사 움직임은 자신의 방향모드와 인접한 방향모드이다. 즉, 주 블록의 방향모드가 D1이면 D1, D2, D8의 부 블록은 주 블록과 유사 움직임을 갖는 것이다.

“객체블록 결정”블록에서는 부 블록들을 객체블록으로 포함시킬 것인가를 결정한다. 부 블록들이 주 블록의 움직임과 유사 움직임을 일정 프레임동안 유지할 때 부 블록을 객체블록으로 포함시킨다. 일정 프레임 유지 이전 10 프레임 동안 6회 이상 유지하는 것을 의미한다. 10 프레임 이전까지는 객체블록을 결정할 정보가 부족하므로 주 블록을 객체로 사용한다. 또한 부 블록이 D0이면 움직이지 않는 객체일 수 있지만 배경일 수도 있으므로 객체블록 결정에서 무시된다.

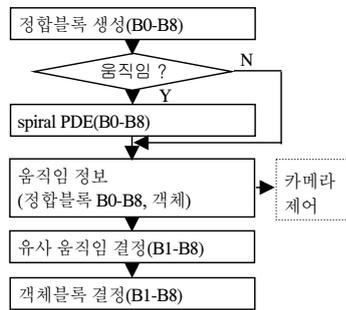


그림 3. 객체 추적을 위한 블록정합
Fig. 3 Block matching for object tracking

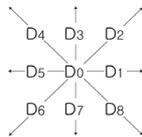


그림 4. 방향 모드들
Fig. 4 Direction modes

3.2 카메라 제어

카메라는 다이와(daiwa) DMP23이고, 선행연구를 토대로 15Hz의 프레임율과 1o당 40/66.67의 팬/틸트 이동

데이터를 카메라 제어를 위해 사용한다[4]. 카메라 제어는 그림 3의 “움직임 정보”블록의 객체 움직임 벡터를 통해 영상 중심의 객체 위치(cObj_P)를 계산하고, 제어 신호에 의해 카메라 시선을 이동하여 객체가 영상 중심에 위치하도록 한다. 객체 위치에 따른 팬/틸트 데이터 (Pan_D, Tilt_D)는 그림 5의 카메라와 영상의 기하학적인 관계에서 영상 중심에 대한 수직/수평 변위(vDis, hDis)에 의해 식 (2)로 얻어진다.

$$\begin{aligned} Pan_D &= \tan^{-1}\left(\frac{vDis}{CamtoImL}\right) \times 40 \\ Tilt_D &= \tan^{-1}\left(\frac{hDis}{CamtoImL}\right) \times 66.67 \end{aligned} \tag{2}$$

여기서 CamtoImL는 카메라와 영상 사이의 가상 거리이고, 팬/틸트 데이터는 40으로 제한한다. 이때 카메라 시선의 이동 벡터(View_V)는(vDis, hDis)이고 (12.3, 12)을 초과 할 수 없다.

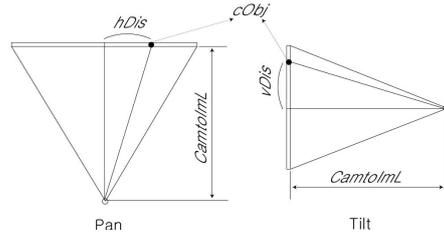


그림 5. 카메라와 영상의 기하학적인 관계
Fig. 5 Geometric relation between a camera and an image

본 논문의 객체추적은 객체 위치를 확인하고 카메라 시선을 이동시키고 있어 카메라 시선 이동 후 객체 위치에 대한 정보와 실제 객체 위치(reObj_P)는 다르다. 다음 프레임의 블록정합을 위해 객체 위치가 수정되어야 한다. 수정된 객체 위치 성분(reObj_Ph, reObj_Pv)은 카메라 시선 이동을 위해 사용된 팬/틸트 데이터를 이용해 식 (3)에 의해 계산된다.

$$\begin{aligned} reObj_Ph &= \tan\left(\frac{Pan_D}{40}\right) \times CamtoImL \\ reObj_Pv &= \tan\left(\frac{Tilt_D}{66.67}\right) \times CamtoImL \end{aligned} \tag{3}$$

벡터 연산으로는 식 (4)와 같이 표시할 수 있다.

$$reObj_P = cObj_P - View_V \tag{4}$$

그림 6은 객체추적 과정을 도식화하여 보여주고 있다. 첫 번째 그림에서는 제안된 블록정합을 통해 이전 프레임의 객체($pObj$)를 현 프레임에서 찾아 그 위치($cObj_P$)를 확인하고, 객체의 위치에 따라 이동되는 카메라 시선 벡터($View_V$)을 보여주고 있고, 두 번째 그림에서는 카메라 시선 이동에 따른 객체 위치 보상으로 객체 위치가 $cObj_P$ 에서 $reObj_P$ 로 수정되는 것을 보여주고 있다.

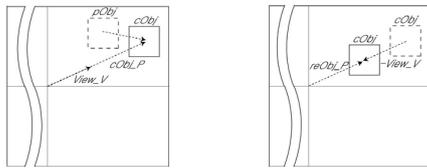


그림 6. 도식화된 객체추적
Fig. 6 Diagrammatized object tracking

IV. 실험 결과 및 고찰

실험은 일반 연구실 환경에서 다이와사의 DMP 23 돔 카메라를 이용해 영상을 취득하고, RS-232C를 이용해 카메라를 제어하며 수행되었다. 제안된 알고리즘의 성능 평가를 위해 정합블록에 따른 객체추적 실험과 자유롭게 움직이는 객체에 대해 카메라 제어를 포함한 객체추적 실험이 수행되었다.

첫 번째 실험에서는 42×42 의 큰 정합블록, 10×10 의 작은 정합블록, 제안된 정합블록을 이용해 객체추적 성능을 비교한다. 동등한 성능 평가를 위해 640×480 의 824 프레임 동영상을 미리 생성하고 객체추적을 수행하였다. 동영상에는 그림 7과 같이 영상 중심에서 실험자가 좌우 4회의 왕복운동을 수행하고, 실험자가 정면을 바라볼 때 얼굴영역에 마우스로 클릭하면 정합블록을 생성하고 객체추적을 시작한다.

그림 8은 객체추적 과정에서 선택된 객체 블록(검정 사각형)을 보여주고 있다. (a)는 큰 정합블록이고 블록 내에 객체가 많이 존재하지만 배경 역시 무시할 수 없을 정도로 존재하는 것을 보여준다. 정합블록에 포함된 일정 크기의 배경은 객체추적의 실패의 원인이다. (b)는 작은 정합블록이고 블록 내에 객체가 대부분 점유하지만 배경과 뚜렷하게 구분이 되지 않는 적은 정보를 보여주

고 있다. 배경과 구분되지 않는 적은 객체정보 역시 객체추적 실패의 원인이다. (c)는 제안된 알고리즘에 의해 선택된 객체블록들이고 배경 영역의 블록들을 적절히 제거시키고 다수의 블록들이 충분한 객체를 포함하고 있는 것을 보여준다. 결과적으로 적절한 객체추적을 가능하게 한다.



그림 7. 객체추적 실험 환경
Fig. 7 Environment for simulation

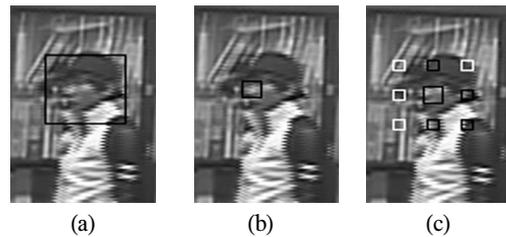


그림 8. 정합블록의 비교 (a) 큰 정합블록 (b) 작은 정합블록 (c) 제안된 정합블록
Fig. 8 Comparison of MB (a) Large MB (b) Small MB (c) Proposed MB

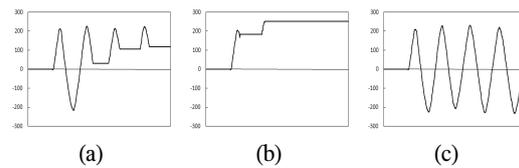


그림 9. 객체추적의 비교 (a) 큰 정합블록 (b) 작은 정합블록 (c) 제안된 정합블록
Fig. 9 Comparison of object tracking (a) Large MB (b) Small MB (c) Proposed MB

그림 9는 이동 객체의 추적 결과를 비교하고 있다. (a)에서는 큰 정합블록이 일정 시간동안 정상적으로 객체추적을 수행하다가 실패하는 것을 보여주고 있고, (b)에서는 작은 정합블록이 짧은 시간 내에 객체추적이 실패하는 것을 보여주고 있고, (c)에서는 제안된 정합블록이

주어진 환경에서 실험자의 움직임에 따라 정확히 객체 추적을 수행하는 것을 보여주고 있다. 수직 값은 영상 중심을 기준으로 한 객체의 수평 좌표이다.

그림 10은 객체추적이 실패하는 순간의 영상이다. (a)에서는 정합블록 내에 배경 영상이 많이 포함되면서 배경이 중요 내용으로 작용하고 배경을 추적하는 것을 보여주고 있고, (b)에서는 객체가 순간적으로 빠르게 움직여 객체 영역이 점선으로 표시된 탐색영역을 벗어나 유사 배경을 객체로 인식하여 배경을 추적하는 것을 보여주고 있다. 반면 제안된 알고리즘은 객체블록과 배경블록을 적절히 구분하고 객체추적을 성공적으로 수행하는 것을 아래 그림에 보여주고 있다.



그림 10. 실패한 객체 추적
(a) 큰 정합블록 (b) 작은 정합블록
Fig. 10 Failed object tracking
(a) Large MB (b) Small MB

실험자가 카메라와 일정거리를 유지하여 객체 크기가 거의 변하지 않는 실험과 같이 실험자가 카메라에 접근을 반복하여 객체 크기가 변하는 실험에서도 유사한 객체추적 결과를 보여주었다. 그림 11은 실험에서 객체 크기가 다르지만 검은 사각형으로 표시된 객체를 적절히 분리하는 것을 보여준다.

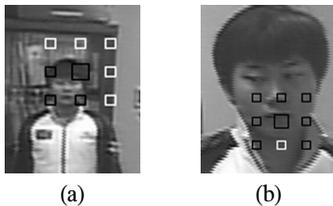


그림 11. 선택된 객체블록
(a) 작은 객체 (b) 큰 객체
Fig. 11 Selected object blocks
(a) Large object (b) Small object

이는 객체 크기가 변하는 실험에서 객체추적이 성공할 수 있는 이유이다.

두 번째 실험에서는 자유롭게 이동하는 실험자를 제안된 객체추적 알고리즘으로 추적하는 실험이다. 실험자는 실험실 전 영역을 자유롭게 움직여 카메라의 전체 FOV에 타나날 수 있도록 하지만 제안된 블록정합 알고리즘과 카메라 제어에 의해 대상 객체는 그림 12처럼 영상 중심 (0,0)에서 수평방향으로 -23 화소에서 20 화소 사이, 수직방향으로 -21 화소에서 1 화소 사이에 유지 시키고 있는 것을 보여주고 있다. 주어진 실험환경에서 수직 방향 움직임을 생성하기 어려워 수직 화소값이 작게 나타나고 있다. 두 실험 결과들은 제안된 객체추적 알고리즘이 객체추적을 적절히 수행하는 것을 의미한다.

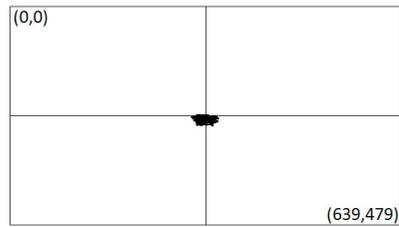


그림 12. 객체 추적 결과
Fig. 12 Result of object tracking

표 1은 계산량을 비교하고 있다. 객체추적이 실패하여 배경을 추적하는 경우 움직임이 없는 것으로 인식한다. 그래서 추적을 수행하지 않는 경우가 발생하여 계산량 비교가 어렵다. 표의 계산량은 세 정합블록이 정상적으로 동작하는 62 프레임의 계산량을 큰 정합블록을 기준으로 상대적인 크기로 비교하고 있다. 제안된 정합블록이 큰 정합블록의 장점을 유지하면서도 계산량을 29.5%로 줄여주고 있다.

표 1. 계산량 비교
Table 1 Comparison of a amount of computation

큰 정합블록	작은 정합블록	제안된 정합블록
100%	7.48%	29.50%

V. 결 론

블록정합 기법을 이용한 객체추적에서 객체 크기가 다양하고 수시로 변하기 때문에 객체추적에 고정 크기의 정합블록은 적합하지 않다. 큰 정합블록의 경우 블록 내에 객체 정보가 많지만 배경 정보 또한 다수 존재하게 되어 지속적인 추적을 어렵게 하고, 작은 정합블록의 경우 블록 내에는 대부분의 객체 정보가 존재하지만 배경과 유사한 단순한 객체이거나 객체 움직임이 조금 빠르면 적절한 객체를 검출하지 못해 지속적인 추적을 어렵게 한다. 본 논문에서는 큰 영역에 다수의 작은 블록들로 구성된 적응적 정합블록을 정의하고, 블록들 중 객체로 판단되는 블록만으로 블록정합을 수행하는 객체추적 알고리즘을 제안하였다. 제안된 알고리즘은 객체 크기가 유지되거나 자유롭게 변하는 영상에서 정합블록에서 배경블록을 제거하고 객체블록만을 적절히 선택하면서 효율적으로 객체추적을 수행하는 것을 보여주고 있고, 제안된 블록정합 알고리즘과 팬/틸트 카메라 제어를 통해 카메라의 FOV 영역에서 자유롭게 움직이는 객체를 수평 -23~20 화소, 수직 -21~1 화소 내에 유지시키는 것을 보여주고 있다. 또한 제안된 정합블록은 계산량을 크게 줄여주고 있다.

참고문헌

[1] I. D. Raid, D. W. Murray, "Active Tracking of Foveated Feature Clusters using Affine Structures," *International Journal of Computer Vision*, vol. 10, pp. 41~60, 1996.

[2] D. Comaniciu, V. Ramesh, and P. Meer, "Real-Time Tracking of Non-Rigid Objects Using Mean Shift," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, vol. II, pp. 142-149, 2000.

[3] 백인호, 임재현, 박경주, 백준기, "Pan-Tilt-Zoom 카메라를 이용한 파노라마 배경 생성과 객체 추적," *전자공학회 논문지 SP편*, vol. 45, no. 3, pp. 239-247, 2008.

[4] 김진태, 안수홍, 오정수, "CCD 카메라의 자동 제어를 위한 신호 생성," *한국화상학회*, vol. 15, no. 2, pp. 1-8, 2009.

[5] Robert Grover Brown, Patrick Y. C. Hwang, *Kalman Filtering*, 1999.

[6] Horn, B.K.P., and Schunck, B., "Determining Optical Flow," *Artificial Intelligence*, vol. 17, pp. 185-203, 1981.

[7] K. Hariharakrishnan and D. Schonfeld, "Fast object tracking using adaptive block matching," *IEEE Transactions on Multimedia*, vol. 7, no. 5, pp. 853-859, 2005.

[8] ITU-T Recommendation H.263 software implementation, Digital Video Coding Group at Telenor R&D, 1995.



김진태(Jin-Tae Kim)

1987년: 중앙대학교 전자공학사
 1989년: 중앙대학교 전자공학석사
 1993년: 중앙대학교 전자공학박사
 1995년 ~ 현재 : 한서대학교
 컴퓨터정보공학과 교수

※ 관심분야: 영상처리, 디지털 포렌직



안수홍(Soo-hong Ahn)

2008년 부경대학교 화상정보공 학부
 학사
 2010년 부경대학교 대학원
 이미지시스템공학과 석사

2010년~현재 코아로직

※ 관심분야: 디지털영상처리, 영상검색



오정수(Jeong-su Oh)

1992년 중앙대학교 대학원 전자
 공학 석사
 2001년 중앙대학교 첨단영상대학원
 영상공학과 박사

2002년~현재 부경대학교 이미지시스템공학과

※ 관심분야: 디지털영상처리, 영상검색