

정규논문 (Regular Paper)

방송공학회논문지 제17권 제4호, 2012년 7월 (JBE Vol. 17, No. 4, July 2012)

<http://dx.doi.org/10.5909/JBE.2012.17.4.676>

깊이 카메라를 이용한 움직임 추정 방법

권순각^{a)*}, 김성우^{a)}

Motion Estimation Method by Using Depth Camera

Soon-kak Kwon^{a)*} and Seong-Woo Kim^{a)}

요약

동영상 부호화에서 움직임 추정은 구현 복잡도에 가장 크게 영향을 미친다. 본 논문에서는 깊이 카메라와 색상 카메라를 동시에 이용하여 움직임 추정의 복잡도를 줄이는 방법을 제안한다. 깊이 카메라로부터 얻어진 거리 정보로 동영상내 객체 정보를 얻고, 비슷한 거리에 있는 화소들을 그룹화하여 동일한 객체로 라벨링을 수행한다. 라벨링된 현재 및 참조 화면내에서 움직임 추정 블록을 세부분(배경, 객체내부, 경계)의 탐색영역으로 구분하여 적응적으로 판단한다. 현재 블록이 객체내부영역이면 참조 화면에서 객체내부영역에만 움직임 추정을 탐색하고, 배경영역이면 블록은 참조 화면에서 배경영역에만 탐색한다. 모의실험을 바탕으로 전역탐색방법에 비하여 제안된 방법은 움직임 추정 차신호가 동일하게 유지되면서 탐색 복잡도가 크게 줄어들음을 확인할 수 있다.

Abstract

Motion estimation in video coding greatly affects implementation complexity. In this paper, a reducing method of the complexity in motion estimation is proposed by using both the depth and color cameras. We obtain object information with video sequence from distance information calculated by depth camera, then perform labeling for grouping pixels within similar distances as the same object. Three search regions (background, inside-object, boundary) are determined adaptively for each of motion estimation blocks within current and reference pictures. If a current block is the inside-object region, then motion is searched within the inside-object region of reference picture. Also if a current block is the background region, then motion is searched within the background region of reference picture. From simulation results, we can see that the proposed method compared to the full search method remains the almost same as the motion estimated difference signal and significantly reduces the searching complexity.

Keyword : Motion estimation, Depth information, Object extraction, Labelling

1. 서론

동영상 압축표준으로는 ISO/IEC의 MPEG-4 Part 10과 ITU-U의 H.264 표준이 최근에 제정되었다^[1]. 또한 현재는 차세대 비디오 부호화 표준으로 HEVC(High Efficiency

a) 동의대학교 컴퓨터소프트웨어공학과 (Dept. of Computer Software Engineering, Donggeui University)

* 교신저자 : 권순각 (Soon-kak Kwon)

E-mail: skkwon@deu.ac.kr

Tel: +82-51-890-1727 Fax: +82-51-890-2629

· 접수일(2012년5월14일), 수정일(2012년6월13일), 게재확정일(2012년7월3일)

Video Coding)에 대한 표준화 작업이 진행중에 있다²⁴⁾. 이러한 동영상 압축표준에는 시간방향의 중복성을 제거하기 위하여 움직임 추정 부호화 기법을 사용하고 있다. 움직임 추정은 현재 화면내의 블록(공간방향으로 이웃한 화소들의 묶음)과 시간방향으로 이웃한 참조 화면내에서 가장 가까운 블록을 찾는 과정이다.

움직임 추정 과정에는 블록을 찾는 평가척도, 참조 화면내 탐색영역의 크기, 현재 블록의 크기 등이 고려되어야 한다. 블록을 찾는 평가척도로는 현재블록과 참조블록사이의 절대오차 신호의 합, 제곱오차 신호의 합 등이 사용된다. 구현의 복잡도 측면에서는 절대오차가 제곱오차에 비하여 더 간단하지만, 추정의 정확도 측면에서는 제곱오차가 절대오차에 비하여 더 정확하다. 움직임 추정의 복잡도는 참조 화면내 탐색영역의 크기에 비례하여 증가한다. 움직임 추정의 복잡도와 관련되는 탐색수는 현재 화면내 각 블록과 가장 가까운 블록을 참조 화면에서 찾기 위해서 탐색하는 총 회수를 의미하며, 탐색수는 수평방향의 탐색영역 크기와 수직방향의 탐색영역 크기의 곱에 정비례한다. 탐색영역 크기가 커질수록 움직임 추정의 복잡도는 커질 수밖에 없으나, 탐색영역 크기가 너무 작으면 추정의 정확도가 떨어질 수 있다. 따라서 동영상 부호화마다 적절한 탐색영역 크기 설정이 필요하다. 움직임 추정을 위한 현재 화면내 블록크기는 작을수록 추정의 정확도가 커지지만 추정에 필요되는 복잡도가 증가되고, 추가적으로 전송되는 움직임벡터가 증가되는 단점이 있다. H.264에서는 16x16, 8x8, 4x4 등으로 블록크기를 가변적으로 선택할 수 있도록 하고 있다.

움직임 추정은 동영상 부호화에서 구현 복잡도의 70% 이상을 차지하고 있으므로, 동영상 부호화 초창기부터 복잡도를 줄이는 고속 움직임 추정방법에 대한 연구가 진행되어져 왔다. 기존의 고속 움직임 추정방법은 탐색 패턴 방법^[5-7], 탐색 시작위치 설정 방법^[8,9], 탐색 조기 종료 방법^[10,11] 등이 있다. 탐색 패턴 방법은 3단계, 다이아몬드, 헥사곤 탐색 등으로 탐색영역을 모두 탐색하는 전역탐색방법을 사용하지 않고, 움직임 벡터 분포 특성을 고려하여 국한된 패턴에 따라 탐색하는 것으로서, 탐색수는 크게 줄일 수 있으나, 추정의 정확도가 떨어질 수 있는 단점이 있다. 탐색 시작위치 설정 방법은 동영상내 블록들 사이의 공간적 및

시간적 상관관계를 고려하여 탐색하는 시작 위치를 효율적으로 설정하여 시작 위치 주위의 일부 영역에 대해서만 탐색함으로써 추정의 복잡도를 줄이기 위한 것이다. 탐색 조기 종료 방법은 탐색시 평가척도의 기준치를 정해놓고, 그 기준치를 초과할 경우에는 그 당시의 참조블록의 비교를 조기에 종료하는 방법이다.

이러한 기존의 방법들은 색상 카메라만을 이용하여 밝기 정보를 얻고, 밝기정보간 움직임 추정을 수행한다. 그러나 최근에는 색상 카메라 뿐만아니라, 깊이 카메라를 동시에 이용한 다양한 응용분야가 대두되고 있다. 따라서, 본 논문에서는 깊이 카메라와 색상 카메라를 동시에 이용한 움직임 추정 방법을 제안한다. 깊이 카메라로부터 제공되는 거리정보로부터 비슷한 거리에 존재하는 화소들을 그룹화하고 라벨링하여 객체를 추출한다. 이러한 라벨링 화면으로부터, 배경만으로 구성된 배경영역, 객체만으로 구성된 객체내부영역, 객체와 배경이 같이 존재하는 경계영역으로 구분한다. 배경영역은 움직임 탐색을 수행하지 않고, 객체내부영역은 참조 라벨링 화면내 객체내부 영역으로, 경계영역은 참조 라벨링화면내 경계영역으로 탐색영역을 제한시켜줌으로써 움직임 추정 탐색수를 크게 줄여준다.

본 논문의 구성은 2장에서는 깊이 카메라를 이용하여 객체 간의 거리정보를 측정하고, 객체간 거리 정보로부터 움직임 추정을 간단히 구현하는 제안한 방법에 대해 살펴본다. 3장에서는 모의실험 및 결과를 분석하며 4장에서 결론에 대해 기술한다.

2. 깊이 정보를 이용한 제안된 움직임 추정방법

카메라가 고정된 상태에서 배경영역과 객체영역을 효과적으로 분리하고, 객체영역에 대해서만 움직임을 추정해줌으로써 움직임 추정의 복잡도를 줄일 수 있을 것이다. 배경과 객체 영역은 깊이 정보로부터 구분이 가능하다. 가장 먼 깊이 정보를 갖는 부분은 배경 부분이 되고, 그렇지 않는 부분은 객체영역이다. 객체 영역은 1개 이상으로 구분이 가능하다. 배경영역은 고정된 것으로서 화면의 변화가 없는 부분이므로 움직임 탐색이 필요없고, 현재 화면과 이전 화면의 같은 공간위치에 있는 화소의 차이값이 움직임 추정 차 신호

로 사용되고, 객체 부분은 두 개의 영역으로 구분된다. 배경과 객체가 같이 포함된 경계영역과 객체만으로 구성된 객체내부영역으로 구분되어 움직임 탐색이 이루어진다.

제안된 움직임 추정방법은 깊이 카메라와 색상 카메라를 동시에 이용하며, 깊이 카메라로부터 카메라와 객체 사이의 거리 정보를 얻고, 색상 카메라로부터 영상의 YUV 신호를 얻는다.

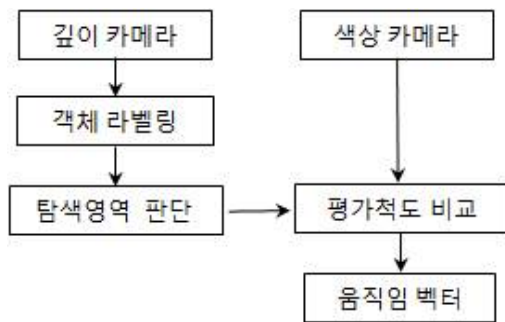


그림 1. 제안된 움직임 추정방법
Fig. 1. Proposed motion estimation method

2.1 깊이 정보의 객체 라벨링 및 탐색영역 판단

깊이 카메라는 비행시간(TOF: Time-of-Flight) 방식으로 작동하는 적외선 펄스 레이저로 구현되고 있다. 적외선 펄스

레이저는 적외선 레이저 다이오드를 포함하며, 다이오드에서 방출된 펄스광은 공간으로 조사되고 피사체의 표면에서 산란된 후에 적외선 펄스 레이저의 광검출기에서 검출할 때까지의 시간이 측정된다. 이 시간을 측정함으로써 객체와의 거리를 측정할 수 있다. 측정된 피사체의 거리 정보는 화면으로 구성되며, 화면은 각 화소(pixel) 별로 구성된다.

화면내 같은 객체는 비슷한 거리에 존재하기 때문에, 수십cm 이내 거리에 있는 화소들을 그룹화하고 라벨링하는 과정을 수행한다. 라벨링된 화면은 배경영역에는 '0', 첫 번째 객체영역에는 '1', 두 번째 객체영역에는 '2' 등으로 객체 수만큼의 크기 값으로 화소값이 저장된다. 이러한 라벨링 화면에 근거하여 움직임 추정에 필요한 두 개의 블록, 즉, 현재 라벨링화면의 움직임추정 블록과 참조 라벨링화면의 움직임탐색 블록은 배경, 경계, 객체내부영역의 세가지로 구분되어 판단된다.

배경영역은 라벨링 화면내 블록의 화소값이 모두 '0'으로 존재하기 때문에 다음과 같이 블록내 화소값을 모두 더하여 '0'이 되면 배경영역으로 판단한다.

$$\sum_{x=1}^{B_x} \sum_{y=1}^{B_y} L(x,y) = 0 \tag{1}$$

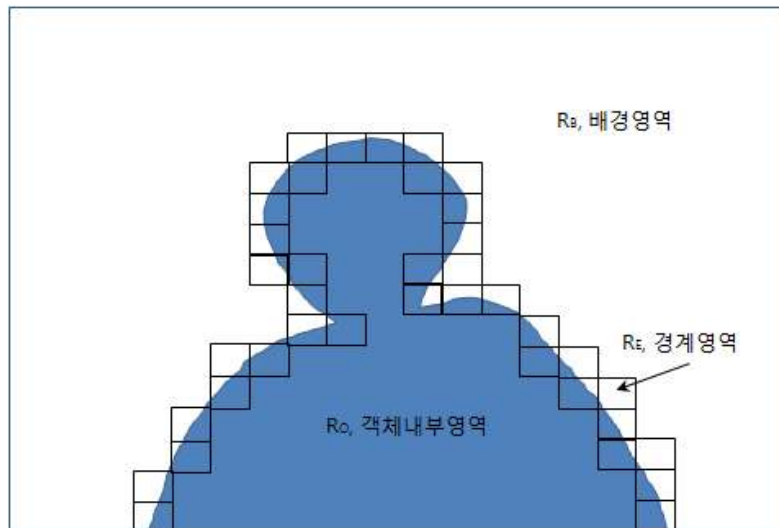


그림 2. 라벨링 화면에서 영역판단
Fig. 2. Region classification on labelling picture

여기서, $L(x, y)$ 은 블록내 수평방향으로 x 위치, 수직방향으로 y 위치에 있는 라벨링 화소값이고, B_x, B_y 는 각각 블록의 수평 및 수직방향의 크기이다.

객체내부영역은 라벨링 블록내 모든 화소값들이 '0'이 아니면서 모두 같은 값을 가지는 경우이므로, 배경영역으로 판단되지 않는 블록에 대하여 블록내 모든 화소값이 같은지를 비교하여 같으면 객체내부영역으로 판단한다.

경계영역은 라벨링 블록내 화소값들이 모두 같지 않는 경우이며, 배경영역과 객체내부영역으로 판단되지 않는 모든 블록을 경계영역으로 판단한다.

움직임 추정시에 현재 라벨링 화면과 참조 라벨링 화면이 필요하다. 현재 라벨링 화면은 움직임 추정하는 현재 색상 화면과 일치하는 것이고, 참조 라벨링 화면은 움직임 탐색에 필요되는 이전 또는 이후 시간에 존재하는 색상 화면과 일치하는 것이다. 현재 라벨링 화면은 움직임 추정의 블록크기가 결정되면, 현재 화면내에서 블록의 위치가 정해지며, 각 블록마다 탐색영역이 판단된다. 만약 현재 라벨링 화면내 하나의 블록이 배경영역으로 판단되면, 대부분 참조 라벨링 화면에서 탐색이 필요없으므로, 참조 블록에 대해서는 탐색영역을 판단하지 않는다. 만약 현재 블록이 객체내부영역 또는 경계영역으로 판단되면, 참조 라벨링 화면에서는 탐색블록마다 탐색영역을 판단한다.

2.2 색상 정보를 이용한 움직임 평가척도 비교

움직임 추정은 움직임 추정차 신호의 크기를 줄여서 부호화해야할 정보를 줄이기 위한 것이므로, 실제 움직임 추정에는 색상정보가 사용되어야 한다. 현재 및 참조의 색상 화면내 블록이 움직임 추정시 동시에 사용되며, 각각의 라벨링 화면내 블록의 탐색영역 판단에 근거하여 움직임 추정이 이루어진다. 현재 색상 화면내 블록은 대응되는 라벨링 블록의 세가지 탐색영역 판정에 따라서 다음과 같이 움직임 추정이 달라진다.

• 배경영역

현재 블록과 공간방향으로 같은 위치에 있는 참조 블록이 모두 배경영역으로 판정되면, 움직임 탐색은 이루어지

지 않고, 움직임 추정차 신호는 다음과 같이 공간적으로 같은 위치에 있는 화소값의 차이로 구해진다.

$$D(x, y) = I_C(x, y) - I_R(x, y) \quad (2)$$

여기서 $I_C(x, y)$ 는 현재 화면의 수평 x , 수직 y 위치에 있는 화소 값이고, $I_R(x, y)$ 는 참조 화면의 수평 x , 수직 y 위치에 있는 화소 값이다.

현재 블록은 배경영역이고, 공간방향으로 같은 위치에 있는 참조 블록이 배경영역이 아니면, 탐색영역내에 있는 모든 참조 블록에 대하여 배경영역인지를 판단한다. 배경영역으로 판단되는 참조블록에 대해서는 움직임 탐색이 수행된다. 그렇지 않는 참조블록에 대해서는 움직임 탐색이 수행되지 않는다.

• 객체내부영역

현재 블록이 객체내부영역이고 참조 라벨링 화면에서 탐색되는 각 블록이 배경 또는 경계영역으로 판정되면 움직임 탐색을 수행하지 않고, 객체내부영역으로 판정되면, 그 블록에 대하여 움직임 탐색이 수행된다. 움직임 탐색 과정은 현재 색상 화면내 블록과 참조 색상 화면내 탐색블록에 대하여 평가척도를 최소화하는 움직임 벡터(수평방향 V_x , 수직방향 V_y)를 얻는다. 다음 식은 절대오차의 합을 최소화 하는 평가척도에 대한 것이다.

$$\min \sum_{x=1}^{B_x} \sum_{y=1}^{B_y} |I_C(x, y) - I_R(x - V_x, y - V_y)|, \quad (3)$$

$$|V_x| < T_x, |V_y| < T_y, (V_x, V_y) \in R_O^R$$

여기서 T_x, T_y 는 수평 및 수직방향의 최대 탐색 크기 값이고, R_O^R 는 참조 라벨링 화면에서 객체내부영역에 속하는 부분을 의미한다. 또한, 절대오차 대신에 제곱오차를 사용하는 평가척도가 사용될 수도 있다.

위 식에서 얻어진 움직임 벡터로부터 다음 식처럼 움직임 추정 차신호를 구한다.

$$D(x, y) = I_C(x, y) - I_R(x - V_x, y - V_y) \quad (4)$$

만약 현재 화면에서 객체가 갑자기 나타나는 경우에는

참조 화면에 객체내부영역이 존재하지 않으므로 이 경우에는 움직임 탐색은 수행하지 않고, 움직임 벡터를 '0'으로 가정한다.

• 경계영역

현재 블록이 경계영역이고 참조 라벨링 화면에서 탐색되는 각 블록이 배경 또는 객체내부영역으로 판정되면 움직임 탐색을 수행하지 않고, 경계영역으로 판정되면, 그 블록에 대하여 움직임 탐색이 수행된다. 움직임 탐색은 현재 색상 화면내 블록과 참조 색상 화면내 탐색블록을 매칭하면서 다음 식과 같은 평가척도(절대오차의 합)를 최소화하는 움직임 벡터를 구한다.

$$\min \sum_{x=1}^{B_x} \sum_{y=1}^{B_y} |I_C(x,y) - I_R(x - V_x, y - V_y)|, \quad (5)$$

$$|V_x| < T_x, |V_y| < T_y, (V_x, V_y) \in R_E^R$$

여기서 R_E^R 는 참조 라벨링 화면에서 경계영역에 속하는 부분을 의미한다.

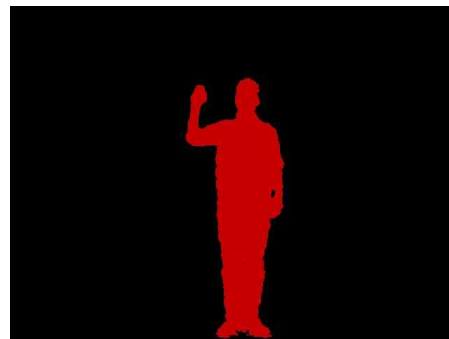
만약 현재 화면에서 객체가 갑자기 나타나는 경우에는 참조 화면에 경계영역이 존재하지 않으므로 이 경우에는 움직임 탐색은 수행하지 않고, 움직임 벡터를 '0'으로 가정한다.

3. 모의실험 및 결과

본 논문에서는 깊이 카메라로 Microsoft사에서 개발한 Kinect^[12]를 사용한다. Microsoft사는 PrimeSense사의 움직임 센서 도입, GestureTek사의 특허 라이선스 사용, 3DV Systems사와 Canesta사의 기업 인수 등을 통하여 게임분야에 적용하기 위한 Kinect를 개발하였다. Kinect 기술은 레이저를 이용해 사용자에게 적외선을 투사하고 나서 적외선 점(dot)들의 반사파 강도를 측정함으로써 이미지들을 나눈



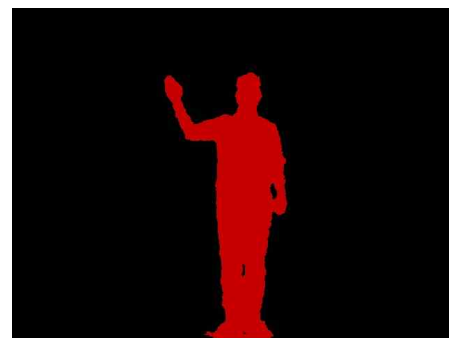
(a) 1번째 화면에 대한 색상신호



(b) 1번째 화면에 대한 라벨링신호



(c) 5번째 화면에 대한 색상신호



(d) 5번째 화면에 대한 라벨링신호

그림 3. Kinect로부터 얻어진 색상신호와 라벨링신호
Fig. 3. Color and labelling signals obtained from Kinect

다. 반사 강도가 약한 점들은 멀리서부터 반사된 것으로 간주되며 보다 강도가 센 점들은 전면에 있는 사용자로부터 온 것으로 추정한다. 그림3은 PrimeSense사의 오픈 라이브러리인 OpenNI^[13]를 이용하여 색상 영상과 거리정보로부터 객체정보를 얻어 라벨링해준 화면을 나타낸다.

각 화면은 수평방향으로 640화소, 수직방향으로 480화소로 구성되어있으며, 초당 30화면으로 총 90화면을 모의 실험에 사용하였다. 실험영상은 배경부분과 1개 객체부분을 담고 있으며, 객체부분에는 사람이 손을 흔들고 있다. 움직임 추정 블록크기는 수평 및 수직방향으로 각각 16화소, 8화소 두가지 경우이며, 탐색영역 크기는 수평 및 수직방향으로 각각 10화소, 15화소, 20화소 세가지 경우에 대하여 기존방법과 제안방법의 성능을 비교하였다. 두가지 방법은 모두 정수화소 단위까지 움직임을 추정한다. 그리고,

참조화면은 바로 이전화면 한 개만을 고려하였다. 표1에 나타낸 것처럼, 제안된 방법은 기존방법과 움직임 추정 차신호가 거의 동일한 수준이 되고, 화면당 탐색수가 10% 이내로 줄어드는 것을 알 수 있다.

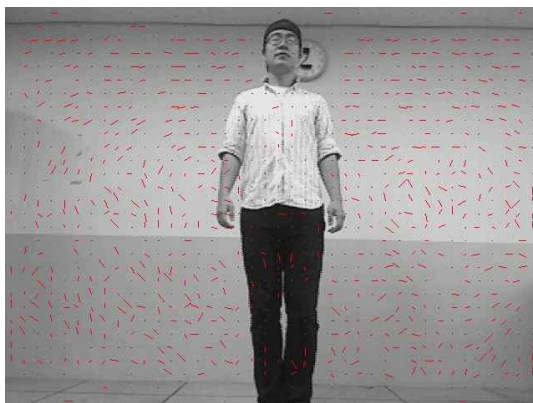
단순히 프레임 차이(Frame Difference)만을 적용하면, 움직임 추정에 대한 복잡도는 줄일 수 있지만, 평균제곱오차 신호가 증가될 수 있다. 표 1과 동일한 영상에 대해 프레임 차이 신호를 구하면, 평균제곱오차가 150.89로 움직임 추정 평균제곱오차에 비하여 크게 증가됨을 확인할 수 있었다.

움직임 추정 차신호 측면에서 기존방법에 비하여 제안방법이 아주 조금 증가하는 이유는 배경부분에서 제안방법은 움직임이 탐색되지 않으므로 움직임벡터가 '0'으로 판단된다. 그러나, 실제적으로는 움직임이 없지만, 잡음 또는 조명

표 1. 기존방법과 제안방법의 움직임 추정 성능 비교

Table 1. Performance comparison of motion estimation between conventional and proposed methods

추정블록크기	탐색영역 크기	전역탐색		제안방법	
		화면당 평균 탐색수	움직임추정 평균제곱오차	화면당 평균 탐색수	움직임추정 평균제곱오차
16	10	500200	22.45	51053	22.49
	15	1089000	22.33	106808	22.38
	20	1881744	22.13	179840	22.16
8	10	2046816	15.96	207387	16.01
	15	4423776	15.66	433809	15.72
	20	7660704	15.39	749641	15.45



(a) 전역탐색



(b) 제안된 방법

그림 4. 움직임 벡터 분포 특징 (추정블록크기=16, 탐색영역크기=10)

Fig. 4. Distribution characteristics of motion vectors (estimation block size = 16, searching region size = 10)

변화 등으로 인하여 전역탐색에 의해서는 움직임벡터가 '0'이 아닌 블록이 찾아지기 때문이다.

제안된 방법의 움직임 탐색수는 화면당 객체부분이 차지하는 비율에 영향을 받을 것이다. 배경부분에 비하여 객체부분이 차지하는 비율이 작아질수록 제안방법의 움직임 추정 탐색수는 비례하여 작아질 것이다. 또한, 양방향 움직임 추정일 경우에는 단방향 움직임 추정에 비하여 최소 2배 이상 움직임 추정의 복잡도가 증가되기 때문에 제안방법에 의한 탐색수 감소의 효능성이 증대될 수 있을 것이다. 또한, H.264처럼 16x16, 8x8, 4x4 등 다양한 가변적 블록크기에 대하여 움직임 추정이 동시에 수행된다면, 제안방법의 효능성이 더욱더 증대될 수 있을 것이다.

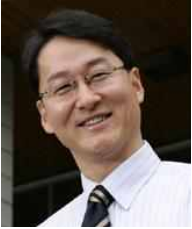
IV. 결론

객체의 윤곽선 정보간 움직임 추정은 기존 MPEG-4 Part2에서 제시되었지만, 정확한 객체 추출의 어려움 등으로 실현되지 못하였다. 본 논문은 깊이 카메라를 이용하여 객체를 추출하고, 객체의 라벨링 정보로부터 움직임 추정 시 탐색범위를 결정하는 방법을 제안하였다. 움직임 탐색은 현재 및 참조 화면의 라벨링 신호로부터 움직임 탐색여부가 판단된다. 현재와 참조블록이 모두 배경영역이면 움직임 탐색이 수행되지 않고, 현재블록이 객체내부영역이면 참조블록도 객체내부영역에 속하면 움직임 탐색이 수행하게 한다. 이러한 제안된 방법은 전역탐색방법에 비하여 거의 동일한 움직임 추정 차신호를 얻어내면서 탐색수를 약 10% 범위로 크게 감소시키는 것을 확인하였다. 본 논문에서 제시된 깊이 카메라를 이용한 움직임 추정방법은 현재까지의 국제표준 동영상 부호화기법 뿐만아니라, 현재 진행중인 HEVC 국제 표준 등에 적용 가능할 것이다. 또한, 깊이 카메라를 이용한 움직임 추정방법은 기존에 제시된 고속 움직임 추정방법과 접목이 가능하며, 이에 대한 추가 연구가 필요할 것이다.

참고 문헌

- [1] S.-k. Kwon, A. Tamhankar, K. R. Rao, "Overview of H.264/MPEG-4 Part 10", Journal of Visual Communications and Image Representation, Vol. 17, No. 2, pp. 186-216, 2006.
- [2] K. Ugur, K. Andersson, A. Fuldseth, G. Bjøntegaard, L. P. Endresen, J. Lainema, A. Hallapuro, J. Ridge, D. Rusanovskyy, C. Zhang, A. Norkin, C. Priddle, T. Rusert, J. Samuelsson, R. Sjöberg, Z. Wu, "High Performance, Low Complexity Video Coding and the Emerging HEVC Standard", IEEE Trans. Circuit Syst. Video Technology, Vol. 20, No. 12, pp.1688-1697, 2010.
- [3] W.-J. Han, J. Min, I.-K. Kim, E. Alshina, A. Alshin, T. Lee, J. Chen, V. Seregin, S. Lee, Y. M. Hong, M.-S. Cheon, N. Shlyakhov, K. McCann, T. Davies, J.-H. Park, "Improved Video Compression Efficiency through Flexible Unit Representation and Corresponding Extension of Coding Tools", IEEE Trans. Circuit Syst. Video Technology, Vol. 20, No. 12, pp.1709-1720, 2010.
- [4] F. Bossen, V. Drugeon, E. Francois, J. Jung, S. Kanumuri, M. Narroschke, H. Sasai, J. Sole, Y. Suzuki, T. K. Tan, T. Wedi, S. Wittmann, P. Yin, Y. Zheng, "Video Coding using a Simplified Block Structure and Advanced Coding Techniques", IEEE Trans. Circuit Syst. Video Technology, Vol. 20, No. 12, pp.1667-1675, 2010.
- [5] S. Zhu, K.-K. Ma, "A New Diamond Search Algorithm for Fast Block-Matching Motion Estimation", IEEE Trans. Circuit Syst. Video Technology, Vol. 10, No. 2, pp.287-290, 2000.
- [6] C. Zhu, X. Lin, L.-P. Chau, "Hexagon-based Search Pattern for Fast Block Motion Estimation", IEEE Trans. Circuit Syst. Video Technology, Vol. 12, No. 5, pp.349-355, 2002.
- [7] X. Jing, L. P. Chau, "An Efficient Three-step Search Algorithm for Block Motion Estimation", IEEE Trans. Multimedia, Vol. 6, No. 3, pp.435-438, 2004.
- [8] M. Ezhilarasan P. Thambidurai, "Simplified Block Matching Algorithm for Fast Motion Estimation in Video Compression", Journal of Computer Science, Vol. 4, pp. 282-289, 2008.
- [9] Y. Nie, K.-K. Ma, "Adaptive irregular pattern search with matching prejudgment for fast block-matching motion estimation", IEEE Trans. Circuit Syst. Video Technology, Vol. 15, No. 6, pp.789-794, 2005.
- [10] L. Yang, K. Yu, J. Li, S. Li, "An Effective Variable Block-size Early Termination Algorithm for H.264 Video Coding", IEEE Trans. Circuit Syst. Video Technology, Vol. 15, No. 6, pp.784 - 788, 2005.
- [11] J.-F. Yang, S.-H. Chang, C.-Y. Chen, "Computation Reduction for Motion Search in Low Rate Video Coders", IEEE Trans. Circuit Syst. Video Technology, Vol. 12, No. 10, pp.948 - 951, 2002.
- [12] <http://www.microsoft.com/en-us/default.aspx>
- [13] <http://openni.org>

저 자 소 개



권 순 각

- 1990년 : 경북대학교 전자공학과 공학사
- 1992년 : KAIST 전기및전자공학과 공학석사
- 1998년 : KAIST 전기및전자공학과 공학박사
- 1997년 ~ 1998년 : 한국전자통신연구원 연구원
- 1998년 ~ 2001년 : 기술보증기금 팀장
- 2001년 ~ 현재 : 동의대학교 컴퓨터소프트웨어공학과 교수
- 2003년 ~ 2004년 : University of Texas at Arlington 교환교수
- 2010년 ~ 2011년 : Massey University 교환교수
- 주관심분야 : 멀티미디어신호처리, 영상통신



김 성 우

- 1991년 : KAIST 전기및전자공학과 공학사
- 1993년 : KAIST 전기및전자공학과 공학석사
- 1999년 : KAIST 전기및전자공학과 공학박사
- 1999년 ~ 2001년 : 한국전자통신연구원 선임연구원
- 2002년 ~ 현재 : 동의대학교 컴퓨터소프트웨어공학과 부교수
- 2008년 ~ 2009년 : California State University at Fresno 교환교수
- 주관심분야 : 무선센서네트워크, 지능형로봇, IT융합시스템