

적응적 신축 움직임 추정 방법

장원석[†], 권오준^{**}, 권순각^{***}

Adaptive Zoom Motion Estimation Method

Won-Seok Jang[†], Oh-Jun Kwon^{**}, Soon-Kak Kwon^{***}

ABSTRACT

We propose an adaptive zoom motion estimation method where a picture is divided into two areas based on the distance information with a depth camera : the one is object area and the other is background area. In the proposed method, the zoom motion is only applied to the object area except the background area. Further, the block size of motion estimation for the object area is set to smaller than that of background area. This adaptive zoom motion estimation method can be reduced at the complexity of motion estimation and can be improved at the motion estimation performance by reducing the block size of the object area in comparison with the conventional zoom motion estimation method. Based on the simulation results, the proposed method is compared with the conventional methods in terms of motion estimation accuracy and computational complexity.

Key words: Zoom, Motion Estimation, Object Region, Motion Block Sizes

1. 서 론

동영상 압축표준 기법으로 MPEG-2, H.264[1], HEVC[2-4] 등이 있으며, 이러한 기법은 시간적인 중복성을 제거하기 위하여 움직임 추정 및 보상 방법이 사용되고 있다. 움직임 추정은 계산적인 복잡도와 추정의 정확도를 동시에 고려하여 블록 정합 기법이 가장 많이 사용된다. 블록 정합 기법은 움직임 추정 과정에서 블록을 찾는 평가척도, 참조 화면 내 탐색 영역의 크기, 현재 블록의 크기 등이 고려되어야 한다. 블록을 찾는 평가척도로는 현재 블록과 참조블록 사이의 절대오차 신호의 합, 제곱오차 신호의 합 등이 사용된다. 구현의 복잡도 측면에서는 절대오차가 제곱오차에 비하여 더 간단하지만, 추정의 정확도 측

면에서는 제곱오차가 절대오차에 비하여 더 정확하다. 움직임 추정의 복잡도는 참조 화면내 탐색영역의 크기에 비례하여 증가한다. 움직임 추정의 복잡도와 관련되는 블록크기는 현재 화면내 각 블록과 가장 가까운 블록을 참조 화면에서 찾기 위하여 탐색하는 블록의 크기를 의미하며, 블록의 크기가 작아질수록 움직임 추정의 복잡도는 커질 수밖에 없으나, 블록의 크기가 너무 작으면 움직임 벡터의 수가 증가한다. 따라서 적절한 블록 크기 설정이 필요하다.

대부분의 블록단위 움직임 추정은 공간적인 평행 이동의 움직임에 대해서는 우수한 움직임 추정 성능을 나타내지만, 카메라 방향으로의 신축 움직임에는 움직임 추정의 성능이 좋지 못하다. 이러한 신축 움직임을 추정하기 위하여 영상의 현재 화면에 대해

* Corresponding Author : Soon-Kak Kwon, Address: (614-714) Eomgang-ro 995, Busanjin-gu, Busan, Korea, TEL : +82-51-890-1727, FAX : +82-51-890-2629, E-mail : skkwon@deu.ac.kr

Receipt date : Jun. 30, 2014, Revision date : Jul. 3, 2014
Approval date : Jul. 4, 2014

[†] Dept. of Computer Software Engineering, Dongeui University (E-mail : 13562@deu.ac.kr)

^{**} Dept. of Computer Software Engineering, Dongeui University (E-mail : ojkwon@deu.ac.kr)

^{***} Dept. of Computer Software Engineering, Dongeui University

* This work was supported by Dong-eui University Grant.(2014AA226)

참조 화면으로부터, 확대 및 축소에 해당되는 신축 비율을 정확하게 추출해야 한다. 이러한 신축 비율은 모든 경우가 가능하며, 가능한 신축비율을 모두 적용하기에는 구현상 불가능하다. 따라서, 구현의 복잡도를 줄이기 위한 방법에 대한 연구가 진행되어 왔다. 다중 참조 화면을 이용한 신축 움직임 추정 방법[5], 블록기반의 고속 스케일링을 이용한 확대 및 축소 움직임 추정 방법[6], 색상 카메라와 깊이 카메라를 동시에 이용하여 간단하게 신축 움직임을 추정하는 신축 정보를 이용한 깊이 카메라를 이용한 움직임 추정 방법[7] 등이 제안되었다.

깊이 카메라를 이용한 신축 움직임 추정 방법은 깊이 카메라로부터 카메라와 객체간의 거리 정보를 얻고, 현재화면내 움직임 추정 블록과 참조 화면 내 탐색 블록의 평균 거리 값을 비교하며, 거리 값의 비율에 따라 신축 비율을 결정한다. 신축 비율에 따라 참조 블록을 확대 또는 축소하고, 현재블록에 대한 움직임 추정 차 신호를 구한다. 따라서, 신축 비율 계산이 한번에 이루어지기 때문에 신축 움직임 추정의 계산량과 움직임 추정의 정확도를 동시에 고려한 측면에서 가장 타당한 방법이 될 수 있다. 그러나, 영상전환 또는 영상의 응용분야에서와 같이 사람 객체가 있고, 뒤 배경이 있는 영상에 대하여 영상의 화면 전체로 신축 비율을 계산하는 것은 계산의 복잡도를 증대시킬 수 있다. 객체 영역은 신축 움직임 추정을 적용하는 것이 바람직하지만, 배경 영역은 신축 움직임이 존재하지 않기 때문에 신축 움직임 추정을 적용하는 것이 바람직하지 않다.

본 논문에서는 깊이 카메라를 이용하여 기존에 제안한 신축 움직임 추정에서 거리 정보를 이용하여 배경과 객체를 구분하고, 객체 영역에 대해서만 신축 움직임 추정을 적용하는 적응적 방법을 제안한다. 또한, 객체 영역과 배경 영역의 움직임 추정의 블록 크기를 서로 다르게 적용하여 객체 영역에서 신축 움직임 추정의 정확도를 높여준다.

본 논문의 구성으로 2장에서는 기존의 신축 움직임 추정방법에 대해 살펴보고, 3장에서는 제안하는 적응적 신축 움직임 추정방법에 대해 설명한다. 4장에서는 모의실험 및 결과를 바탕으로 제안하는 신축 움직임 추정의 정확도와 복잡도를 기존 방법과 비교한다. 마지막으로 5장에서 결론을 맺는다.

2. 기존의 신축 움직임 추정 방법

2.1 다중 참조 화면을 이용한 신축 움직임 추정 방법

다중 참조 화면을 이용한 신축 움직임 추정 방법 [5]은 여러 장의 참조 화면 중 현재 화면의 블록과 가장 유사한 화면 영역을 참조 화면으로 결정하여 예측하는 것으로서, 다중 참조 화면을 확장하여 참조 화면들을 확대 또는 축소시키는 방법을 사용한다.

블록 정합 이동성 및 신축 움직임 추정을 구현하는 가장 직접적인 방법은 예측용 신축 참조 화면을 포함하는 참조 화면 세트를 확장하는 것이다. Fig. 1에서 시간 축(Time)은 다중 참조 화면내 기존 시간적 참조 화면을 나타내고 있으며, 신축 축(Zoom in/out)은 각 시간적 참조 화면의 신축 참조 화면 세트를 나타낸다. 시간 축은 두 개 대상의 움직임을 나타낸다. 구름은 오른쪽으로 카메라를 향해 이동하고 있으며(확대 효과), 태양은 상대적으로 고정적이며 시간 축을 따라 축소 효과를 나타낸다. 이 경우, 다중 참조 화면은 구름과 태양 모두에 대해 적절한 예측을 제공하지 못하지만 보여 지는 것과 같이 신축 참조 화면은 더 나은 움직임 예측을 제공하게 된다. 그러나, 이 방법은 참조 화면들을 확대 또는 축소하는 과정에서 신축 비율을 정하기 어렵고, 다수의 참조 화면이 만들어져야 하는 문제점이 있다.

2.2 깊이 카메라를 이용한 신축 움직임 추정 방법

깊이 카메라와 색상 카메라를 동시에 이용하여 신축 움직임을 추정하는 방법[7]이다. 색상 카메라로부터 색상정보, 깊이 카메라로부터 거리 정보를 획득하고, 움직임 추정을 위한 현재 화면내 하나의 블록에 대한 색상 정보와 신축정보의 평균을 얻는다. 참조 화면내 탐색 블록에 대한 색상정보와 깊이정보의 평균값을 얻은 후, 현재 블록과 참조 블록에 대한 거리정보의 평균값으로부터 신축 비율을 계산한다.

$$s = \left(\frac{\bar{R}_c}{\bar{R}_r} \right)^\beta \quad (1)$$

여기서, \bar{R}_c 는 현재 화면의 블록의 평균 거리정보, \bar{R}_r 는 참조 화면의 블록의 평균 거리정보이고, β 는 상수값이다.

식(1)에서 구한 신축 비율은 현재블록과 참조블록의 거리 비율에 직접적으로 관계되고, $\bar{R}_c > \bar{R}_r$ 이면,

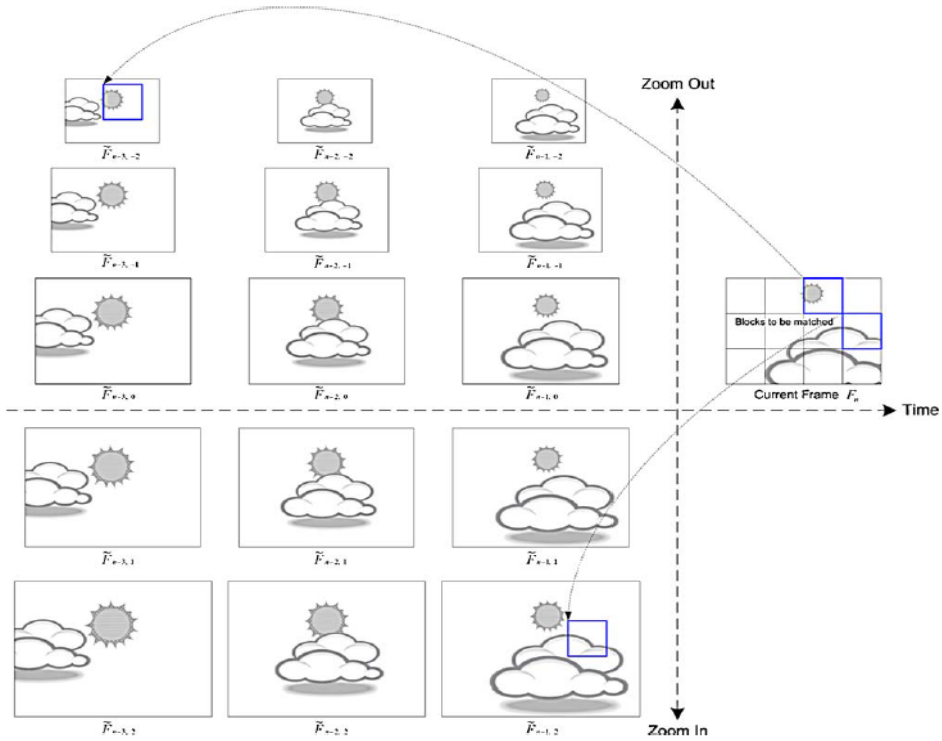


Fig. 1. Zoon motion estimation using multiple reference pictures.

$S > 1$ 로 되어 참조블록은 확대되고, $\bar{R}_f < \bar{R}_c$ 이면, $S < 1$ 로 되어 참조블록은 축소된다. 이렇게 구해진 신축 비율을 이용 참조블록에 적용하여 확대 또는 축소하여 움직임 추정 차신 호를 얻는다. 그러나, 이 방법은 다중 참조 화면을 이용한 신축 움직임 추정에 비하여 계산의 복잡도가 현저하게 줄어들지만, 객체와 배경을 분리 하지 않고, 전체 화면을 같은 블록 크기로 움직임 추정하기 때문에 객체와 배경이 같이 포함된 영상에서는 비효율적이다. 배경의 경우에는 움직임이 작기 때문에 차신호가 작을뿐 만아니라, 신축을 적용할 필요가 없다.

3. 제안하는 적응적 신축 움직임 추정 방법

객체와 배경 모두를 신축 적용하여 움직임 추정 차 신호를 구할 수 있지만, 배경의 경우 움직임 차 신호가 작으므로 블록 크기를 크게 하여 신축 움직임을 적용하거나 신축 움직임을 적용하지 않으며, 객체의 경우에는 블록크기를 작게 하여 신축 움직임을 적용하는 것이 더 효율적이다. Fig. 2는 본 논문에서

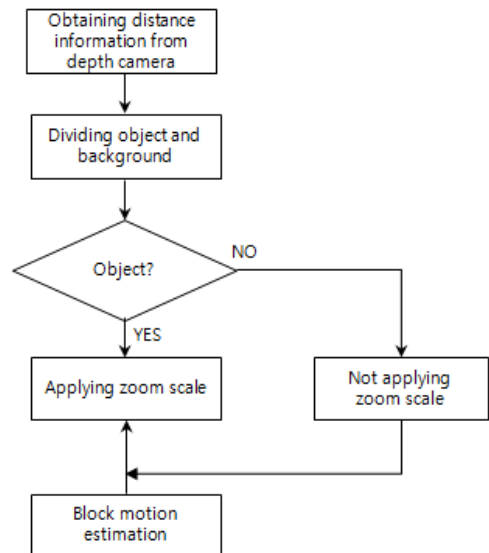


Fig. 2. Flowchart for adaptive zoom motion estimation.

제안하는 적응적 신축 움직임 추정 방법에 대한 알고리즘 흐름도이다. 먼저, 깊이 카메라로부터 제공되는 거리정보를 이용하여 현재 화면의 기준블록이 배경

영역인지 또는 객체 영역인지를 판단한다. 배경 영역은 구현의 간단화를 위하여 신축 움직임 탐색을 수행하지 않게 하고, 객체 영역에 대해서만 신축 움직임 탐색을 수행하게 한다. 이를 바탕으로 움직임 추정 작업의 복잡도를 줄일 수 있다.

3.1 객체와 배경의 분리 방법

깊이 카메라로부터 구한 거리 정보를 사용하여 객체와 배경을 분리한다. 거리 정보를 이용하여 객체와 배경을 분리 하는 방법은 두가지가 있다.

3.1.1 거리 정보의 특정 임계치 이용

깊이 카메라를 이용하면 카메라로부터의 거리 정보를 알 수 있다. 카메라로부터 가장 먼 거리 정보의 근사값을 임계값으로 두고 먼 거리는 배경으로 판단하고, 가까운 거리는 객체로 판단하는 방법이다. 이 방법은 간단한 구현과 계산이 단순한 장점이 있다. 객체 수가 작거나 배경이 단순할 경우 아주 좋은 방법이 될 것이다. 동일 객체는 대체로 유사한 간격의 거리를 가지고 있기 때문에 깊이 정보 값도 대체로 동일하다. 이 점을 이용해 깊이 정보 값을 가지고 있는 깊이 정보 영상에서 유사한 깊이 정보 값을 갖고 있는 단위마다 묶는 라벨링 알고리즘을 이용하게 되면 깊이 정보 영상에서 객체를 추출할 수 있게 된다.

$$R_O < R_T < R_B \tag{2}$$

여기서 R_T 는 임계값으로서 카메라로부터 배경까지의 거리에서 배경과 가까운 거리를 정한다. R_T 보다 큰 값은 배경으로 판단하고, 작은 값은 객체로 판별한다.

Fig. 3은 화면의 깊이 값을 파악하고 깊이 값이 임계치를 넘어 가면 해당 깊이 값을 255로 변환하여 배경 영역으로 판단하고, 임계값을 넘지 않을 경우 0으로 변환하여 객체 영역으로 판단한 경우를 보인다.

3.1.2 신축 비율을 이용

특정 임계치를 이용한 방법은 배경이 돌출되거나 객체가 여러가지일 경우 잘못 인식될 수 있다. 신축 비율을 이용한 방법은 배경과 객체 분리에 참조 화면과 현재 화면을 모두 이용하여 배경과 객체를 분리하는 방법이다. 움직임 추정에 사용되는 현재 블록의 크기가 정해지면, 현재 블록 위치의 거리 정보를 이

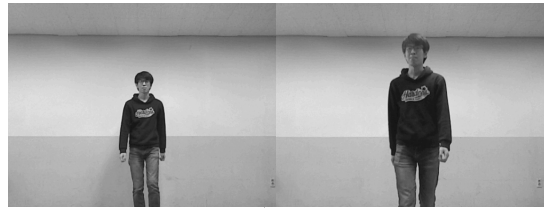


Fig. 3. Object extraction using depth information.



용하여 거리 정보의 평균값을 구한다. 참조 블록의 경우도 같은 방법으로 거리 정보의 평균을 구한다. 이렇게 구해진 현재블록의 거리 평균값과 참조블록의 거리 평균값을 이용하여 신축 비율을 구하는데, 신축 비율은 식 (1)을 이용한다. 구해진 신축 비율이 다음과 같은 범위에 속하면 배경 영역으로 판단한다.

$$T_1 < S_B < T_2 \tag{3}$$

여기서, T_1 과 T_2 는 상수 값으로서 $T_1 < T_2$ 이다.

또한, 다음과 같은 조건이 만족되면 객체 영역으로 판단한다.

$$S_O < T_1 \text{ 또는 } S_O > T_2 \tag{4}$$

Fig. 4는 S 값이 $T_1=0.8$ 초과하거나 $T_2=1.3$ 미만일 경우에, 한 화면에 대하여 신축비율을 선으로 표시한 예를 나타낸다.



Fig. 4. Zoom rate for a picture.

3.2 적응적 신축 움직임 추정

객체와 배경을 분리하고 난 후, 객체의 경우에만 신축 움직임 추정 방법을 적용하여 움직임 추정 차신호를 구한다. 배경영역은 움직임 추정의 블록 크기를 가능한 크게 설정한다. H.264와 HEVC 동영상 부호화 기법에서는 움직임 추정시에 블록크기를 가변적으로 사용할 수 있도록 허용하고 있다. 따라서, 배경영역은 가능한 큰 블록 크기인 수평방향 16화소, 수직방향 16화소로 설정한다. 객체 영역은 신축 움직임 추정의 정확도를 높이기 위하여 배경보다 작은 수평방향 8, 수직방향 8화소 크기를 사용한다. 또한, 현재 블록이 객체 영역과 배경 영역이 같이 포함되는 경계 영역에서도 8×8 크기 단위로 객체 및 배경을 판단하여 신축 움직임 추정을 적용한다.

Fig. 5에서 나타난 것처럼 화면에서 객체와 배경을 분리하면 객체 영역, 배경 영역, 경계 영역으로 나눌 수 있다. 배경 영역과 객체 영역이 겹치는 경계 영역의 경우 객체 영역이다. 객체 영역의 경우에는 조금씩 움직이기 때문에 신축비율 변화로 판단할 수 있지만 경계 영역의 경우에는 거리정보를 이용해서 객체영역으로 판단하기에는 무리가 있다. 블록 크기 안에 단 몇 개의 화소만이 객체 영역이면 그 위치의 거리정보로는 신축비율을 이용한 방법에만 의존하기 힘들다. 그러므로 거리정보의 특정 임계값을 이용한 객체 분리 방법도 병행되어야 한다. 임계값을 이용하여 블록안의 거리값들을 비교하고 그 안에 객체 영역 거리값이 있다면 그 블록을 경계 영역으로 판단한다.

경계 영역의 경우에는 현재 블록이 경계 블록이라 판단되면 16블록 크기에서 4개의 8블록 크기로 나눈다.

다. 여기서 다시 4개의 블록을 임계값을 이용하여 블록안의 거리값을 비교하고 객체 영역의 거리값이 있는지 판단한다. 만약 블록안의 거리값 전체가 배경 거리 값이라면 신축을 적용하지 않고 움직임 추정을 진행하고, 만약 객체 영역의 거리 값이 존재 한다면 신축을 적용하여 신축 비율만큼 참조 블록을 확대 또는 축소하여 움직임 추정을 진행한다. 배경과 객체를 분리하여 신축 움직임 추정 방법을 적용하면, 배경 부분의 움직임 추정 과정에서 블록의 확대 및 축소 과정을 생략할 수 있기 때문에 신축 움직임 추정의 복잡도를 줄 일 수 있다.

4. 모의실험 및 결과

모의실험을 통하여 객체 영역의 신축 움직임 추정 방법에 대한 성능을 살펴본다. 실험 영상으로는 한 개의 객체와 배경으로 구성된 것으로서 자체적으로 Kinect 깊이카메라를 사용하여 제작하였다. Fig. 6은 모의실험에 사용한 영상중에서 8개의 화면을 나타낸다. 화면은 수평방향으로 640화소, 수직방향으로 480화소로서 총 17화면으로 구성된다. 움직임 추정은 H.264의 구조를 사용하였다. 움직임 추정 정확도를 평가하기 위하여 현재 화면과 움직임 보상된 화면 사이의 평균제곱차(MSE)값을 사용하였다. 움직임 추정 블록크기는 객체의 경우 수평 및 수직방향으로 8화소이며, 배경영역의 경우 16화소를 적용하였다. 탐색영역의 크기는 수평 및 수직방향으로 70화소이다.

Fig. 7은 화면 전체 영역의 거리값 평균과 객체 영역만의 거리값 평균을 나타내고 있다. 객체 영역은

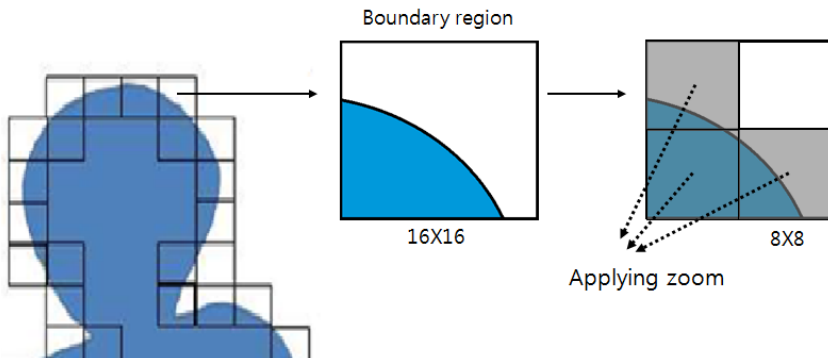


Fig. 5. Object detection at boundary regions.

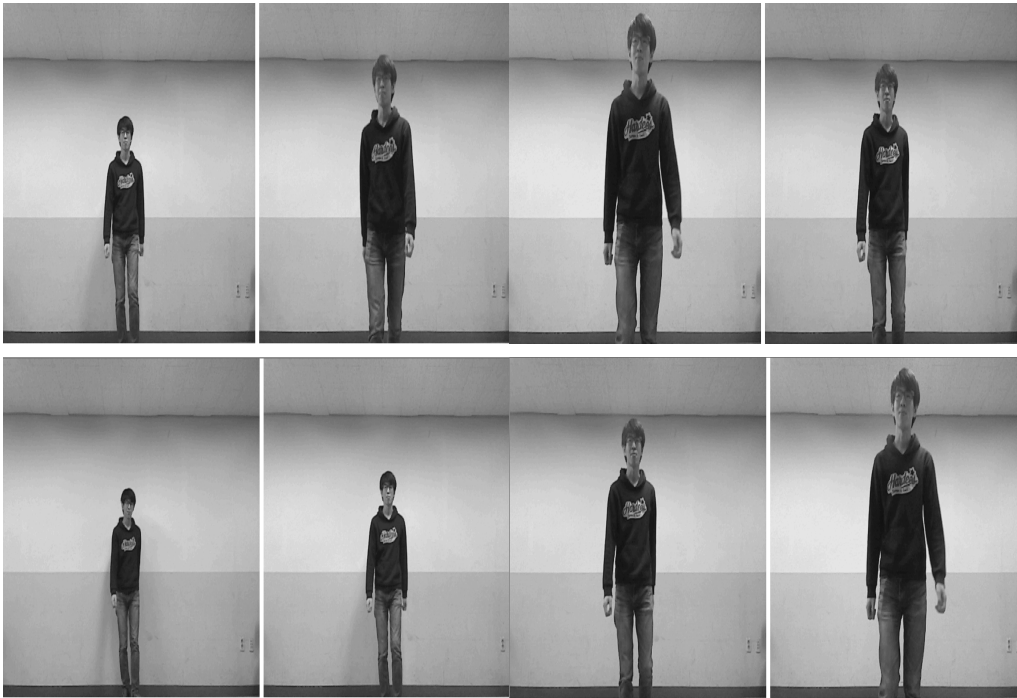


Fig. 6. Pictures used in simulation,

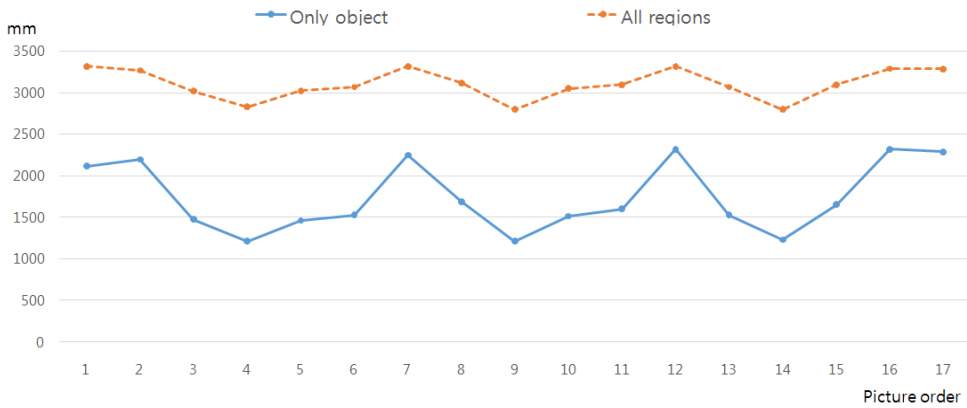


Fig. 7. Average values of distances from camera,

카메라로부터 배경보다 가까이 있기 때문에 거리값이 작은 것을 알 수 있다.

Fig. 8은 기존의 신축을 적용하지 않은 16×16 블록 움직임 추정방법(No Zoom)에 대한 움직임 추정 MSE와 신축을 적용한 16×16 블록 방법(Conventional Zoom)에 대한 MSE를 제안하는 적응적 신축 움직임 추정 방법(Adaptive Zoom)과 비교하여 나타낸다. 기존의 신축을 적용하지 않은 16×16블록 MSE와 신축을 적용한 16×16블록 MSE를 보면 신축 움직임

이 많은 화면, 즉 화면의 객체 거리값 차이가 많이 나는 경우 MSE값이 큰 차이를 나타내는 것을 확인할 수 있다.

또한, 객체와 배경을 분리하고 객체영역에서 움직임 추정 블록크기를 작게하는 적응적 신축 움직임 추정 방법은 기존 신축 움직임 추정 방법보다 MSE 값이 더 줄어들음을 알 수 있다. 16×16 블록 크기로 전역으로 신축 적용한 방법에 비하여 배경은 신축 적용하지 않고 객체만 8×8블록 크기로 적용한 방법이 움

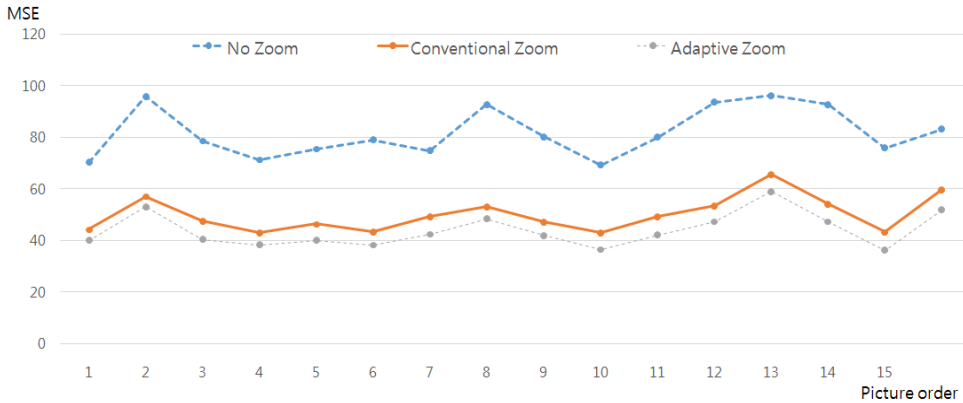


Fig. 8. Comparison of motion estimation MSE.

직입 추정 정확도에서 개선된 것이다. 배경의 경우 움직임이 없기 때문에 신축을 적용하나 적용하지 않으나 큰 차이가 없다. 배경영역의 블록은 신축을 적용하지 않으므로 움직임 추정의 계산량이 크게 줄어든다. 만약 객체 영역의 블록 크기를 작게 하여 신축을 적용한다면 전체영역을 신축 적용한 것과 비슷한 계산량으로 더 좋은 차 값을 얻을 수 있다. 즉 기존 신축 움직임 추정방법과 계산량은 비슷하지만 객체 영역의 블록 크기를 줄임으로써 더 좋은 움직임 추정 예측 값을 얻을 수 있다.

블록 정합 과정에서 객체와 배경을 판단하는데 신축 비율 및 거리값을 이용 하여 객체영역(경계영역 포함) 배경영역으로 구분한다.

Table 1은 블록 정합과정에서 선택된 객체영역의 블록과 배경영역의 블록을 나타내고 있다. 객체 영역의 블록을 살펴보고, 객체가 화면에서 차지하는 비율을 나타내고 있다. 16×16크기의 블록일 경우 전체 블록 수는 1600개이다. 한 개의 16×16블록은 4개의 8×8 블록으로 나눈다.

5. 결 론

블록정합 움직임 추정에는 신축 움직임을 추정하기 어렵다. 깊이 카메라는 카메라와 객체 사이의 거리 정보를 알려주며, 이러한 거리 정보로부터 화면의 객체 영역과 배경 영역을 알 수 있다. 객체 영역과 배경 영역을 분리하여 신축 움직임 추정 방법을 적용하면 배경영역을 신축 적용하지 않기 때문에 계산량이 줄어든다. 객체영역의 블록크기를 줄여 기존 신축 움직임 추정 방법과 비슷한 계산량으로 움직임 추정

Table 1. Block occupation of object and background regions

picture order	object (8×8)	background (16×16)	object percentage(%)
1	1696	1176	26
2	1972	1107	30
3	1940	1115	30
4	1792	1152	28
5	1648	1178	25
6	1948	1113	30
7	1844	1139	28
8	2100	1075	32
9	1808	1148	39
10	1728	1168	36
11	1660	1185	25
12	1864	1134	29
13	2076	1081	32
14	1836	1141	28
15	1648	1188	25
16	1804	1179	28

의 정확도가 좋아지는 것을 확인할 수 있었다.

REFERENCE

[1] S.K. Kwon, A. Tamhankar, and 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, and et al., "High Performance, Low Complexity Video Coding and the Emerging HEVC Standard," *IEEE Transactions on Circuit Systems 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, and et al., "Improved Video Compression Efficiency through Flexible Unit Representation and Corresponding Extension of Coding Tools," *IEEE Transactions on Circuit Systems Video Technology*, Vol. 20, No. 12, pp. 1709-1720, 2010.
- [4] F. Bossen, V. Drugeon, E. Francois, J. Jung, S. Kanumuri, M. Narroschke, and et al., "Video Coding using a Simplified Block Structure and Advanced Coding Techniques," *IEEE Transactions on Circuit Systems Video Technology*, Vol. 20, No. 12, pp. 1667-1675, 2010.
- [5] L.M. Po, K.M. Wong, K.W. Cheung, and K.H. Ng, "Subsampled Block-Matching for Zoom Motion Compensated Prediction," *IEEE Transaction on Circuit Systems Video Technology*, Vol. 20, No. 11, pp. 1625-1637, 2010.
- [6] H.S. Kim, J.H. Lee, C.K. Kim, and B.G. Kim, "Zoom Motion Estimation using Block- Based Fast Local Area Scaling," *IEEE Transactions on Circuit Systems Video Technology*, Vol. 22, No. 9, pp. 1280-1291, 2012.
- [7] S.K. Kwon, Y.H. Park, and K.R. Kwon, "Zoom Motion Estimation using Depth Information," *Journal of Korea Multimedia Society*, Vol. 16, No. 2, pp. 131-137, 2013.



장 원 석

2002년 3월~2006년 2월 경상대학교 해양생산학과 졸업
 2012년 3월~현재 동의대학교 컴퓨터소프트웨어공학과 석사과정
 관심분야: 컴퓨터응용, 멀티미디어 신호처리



권 오 준

1986년 2월 경북대학교 전자공학과 졸업
 1992년 2월 충남대학교 전산학과 석사
 1998년 2월 포항공과대학교 전산학과 박사

1986년 1월~2000년 2월 한국전자통신연구원 선임연구원
 2000년 3월~현재 동의대학교 컴퓨터소프트웨어공학과 교수
 관심분야: 정보보안, 컴퓨터네트워크, 패턴인식



권 순 각

1990년 2월 경북대학교 전자공학과 졸업
 1992년 2월 KAIST 전기및전자공학과 석사
 1998년 2월 KAIST 전기및전자공학과 박사

1997년 3월~1998년 8월 한국전자통신연구원 연구원
 1998년 9월~2001년 2월 기술신용보증기금 기술평가센터 팀장
 2003년 9월~2004년 8월 Univ. of Texas at Arlington 교환 교수
 2010년 9월~2011년 8월 Massey University 교환 교수
 2001년 3월~현재 동의대학교 컴퓨터소프트웨어공학과 교수
 관심분야: 멀티미디어 신호처리, 영상통신