

깊이 영상 부호화에서 신축 움직임 추정 방법

이동석[†], 권순각^{**}

Zoom Motion Estimation Method for Depth Video Coding

Dong-Seok Lee[†], Soon-Kak Kwon^{**}

ABSTRACT

In this paper, we propose a method of the zoom motion estimation for the depth video coding. The proposed method calculates the zoom ratio using the average of the depth values in the current block and in the reference block. It resizes the reference block by the zoom ratio and interpolates the reference block to size of the current block. It compares the current block with the reference block that is obtained by subtracting the average of pixels from the current block to the reference block in order to find the reference block that is the best closest one to the current block. The results of the simulation for the proposed method show that the motion estimation errors are significantly reduced.

Key words: Depth Video Coding, Zoom Motion Estimation, Depth Information

1. 서 론

깊이 카메라의 등장으로 영상을 이용한 움직임 검출에서 기존 색상 영상에서의 한계를 극복하고, 물체의 거리 정보를 통해 정확한 움직임 정보를 얻을 수 있게 되었고, 이를 통해 객체를 검출, 추적하여 이벤트를 제어하는 방법을 제공할 수 있다. 깊이 영상을 통해 객체를 검출하는 방법과 검출된 객체의 움직임을 추적하여 이벤트를 제공하는 방법 등이 폭 넓게 연구되고 있다[1-7].

색상 영상은 부호화를 통하여 저장, 통신, 방송 등의 다양한 응용분야로 활용되고 있다. 하지만 깊이 영상의 부호화에 대한 연구가 현재까지는 폭 넓게 이루어지지 않고 있다. 현재까지 연구된 깊이 영상의 부호화방법으로는 객체 기반의 부호화 방법[8-11]과

촬영 영역의 표면 정보를 모델링하여 깊이 영상을 부호화하는 방법[12-13] 등이 연구되었다. 하지만 이러한 방법들은 정지 영상에 대한 화면내 부호화 방법으로, 연속적으로 촬영되는 깊이 영상을 활용하기 위해서는 화면간 부호화 방법이 연구되어야 할 필요가 있다.

화면간 부호화 방법의 대표적인 방법으로는 현재 프레임과 근접한 참조 프레임으로부터 움직임 추정을 수행하여 추정 프레임과 현재 프레임간의 차신호를 부호화하는 방법이 있다. 색상 영상의 부호화 표준인 MPEG-2, H.264/AVC, H.265/HEVC 등에서는 이러한 움직임 추정을 통한 부호화 방법을 정의하고 있다. 색상 영상의 화면간 부호화 방법에 쓰이고 있는 움직임 추정 방법에서는 카메라의 수평 이동이나 영상 내 객체의 수평 이동의 경우에는 우수한 성능을

※ Corresponding Author: Soon-Kak Kwon, Address: (47340) Eomgang-ro 176, Busanjin-gu, Busan, Korea, TEL: +82-51-890-1727, FAX: +82-505-182-6905, E-mail: skkwon@deu.ac.kr

Receipt date: Sep. 5, 2017, Revision date: Oct. 10, 2017
Approval date: Oct. 14, 2017

[†] Dept. of Computer Software Engineering, Donggeui University (E-mail: ulsan333@gmail.com)

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

※ This research was supported by The Leading Human Resource Training Program of Regional New industry through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT and future Planning(No. 2017045434).

보이지만, 객체가 카메라 방향으로 움직이는 경우와 카메라가 객체 방향으로 움직이는 경우 등 영상 내 객체의 신축이 일어난 경우에는 움직임 추정의 성능이 좋지 못하다. 이는 움직임 추정에 있어서 신축을 고려하기에는 가능한 신축 비율의 개수가 많기 때문에 계산 복잡도가 너무 커지기 때문이다. 이에 따라 색상 영상의 신축 움직임 추정에 있어서 계산복잡도를 낮추기 위한 여러 연구가 이루어져왔다. 여러 신축 비율의 참조 영상을 생성하여 신축 움직임을 추정하는 방법[14], 블록기반의 고속 스케일링을 이용한 확대 및 축소 움직임 추정 방법[15], 계산 효율, 정확도 등을 고려한 반복적 최소-자승 추정에 대한 다양한 개선방법[16] 등이 제안되었다. 또한, 깊이 카메라를 이용하여 객체의 신축 비율을 추정하여 색상 영상의 움직임 추정 및 움직임 보상을 하는 방법[17,18]이 제안되었다.

앞에서 살펴본 신축 움직임 추정방법은 깊이 영상을 이용하지만, 실제로는 색상 영상의 움직임 추정을 효율적으로 수행하는 용도로 깊이 영상이 사용되었다. 본 논문은 깊이 영상의 자체를 부호화하기 위하여 시간 방향의 움직임 추정을 효율적으로 수행하는 방법을 제안한다. 깊이 영상을 부호화하기 위한 움직임 추정에서는 영상 내 객체가 신축이 일어난 경우 객체의 크기와 함께 깊이 화소값(깊이 영상에서는 카메라와 객체사이의 거리와 관련된 값)도 변한다는 특징이 있다. 따라서 깊이 영상 부호화에서 움직임 추정을 수행하기 위해서는 객체의 신축에 따른 깊이 화소값의 변화도 고려하여 움직임 추정을 수행하여야 한다.

본 논문에서 제안하는 깊이 영상 부호화용 움직임 추정 방법은 먼저 현재 블록의 화소 평균과 탐색 영역 내 참조 블록의 화소 평균을 계산한다. 현재 블록과 참조 블록의 평균들로부터 신축 비율을 계산하고, 신축 비율을 이용하여 참조 블록 크기를 변경한다. 그 후 변경된 크기를 가지는 참조 블록을 현재 블록의 크기로 보간한다. 그 후 현재 블록과 참조 블록에 대해 각각의 화소 평균을 빼 새로운 블록들을 생성하고, 그 블록들의 오차를 움직임 추정 평가 척도로 사용한다. 제안한 방법을 이용하여 깊이 영상에서의 신축 움직임 추정을 수행할 수 있다.

2. 색상 영상 부호화에서 신축을 통한 움직임 추정 방법

색상 영상에서 신축 움직임 추정 방법 중 여러 신축 비율의 참조 영상을 생성하여 신축 움직임을 추정하는 방법[14]은 시간상의 각각의 참조 영상들에 대해 다양한 신축 비율로 신축한 참조 영상을 Fig. 1처럼 생성한다. 그 후 현재 영상의 각각의 블록마다 참조 영상들 중 제일 유사한 블록영역을 참조블록으로 하여 신축 추정을 하는 방법이다. 이 때 Fig. 1에서 구름과 태양이 시간상으로 다른 방향으로 움직이고, 신축 비율도 달라지지만, 본 방법을 이용함으로써 각 블록마다 정확한 참조 블록을 찾아낼 수 있다. 하지만 이 방법은 신축 움직임을 추정하기 위해 각 참조 영상마다 여러 신축 비율에 대한 참조 영상을 새로 생성해야 하기 때문에 메모리 소모가 크고, 신축 움직임 추정에 있어 정확도를 높이기 위해서는 많은 신축 비율에 대해서 참조 영상이 생성되고 이를 참조하는 과정이 추가되어야 하므로 계산 복잡도가 커진다는 문제점이 있다.

블록기반의 고속 스케일링을 이용한 확대 및 축소 움직임 추정 방법[15]은 신축 움직임 추정을 하기 위해 참조 블록의 크기를 신축 비율에 맞춰 확대 또는 축소하여 참조 블록을 탐색한 후, 참조 블록에 대해 현재 블록의 크기로 보간을 수행하여 움직임 추정을 하는 방법이다. 이 때 Fig. 2와 같이 기존 움직임 추정에서 쓰인 움직임 벡터(MV: Motion Vector)와 더불어 신축 비율을 나타내는 신축 벡터(ZV: Zoom Vector)를 사용하여 움직임 추정을 수행한다. 또한 신속한 신축 움직임 추정을 하기 위해 참조 블록을 탐색

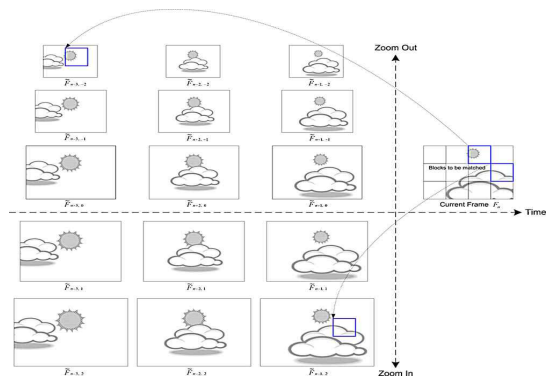


Fig. 1. Method of motion estimation using multiple zoom reference frames.

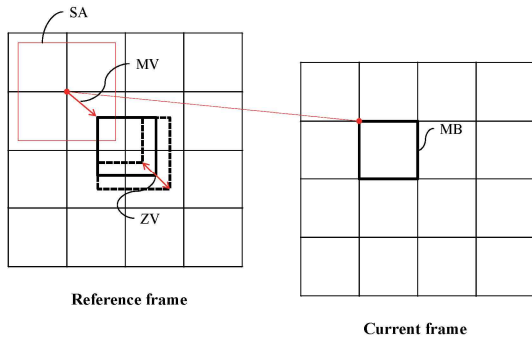


Fig. 2. Motion estimation using zoom vector.

하는 과정에서 기존 다이아몬드 탐색 방법[19]을 응용하여 신축 비율을 Z축으로 보고 Fig. 3과 같이 3차원 다이아몬드 탐색을 수행하여 탐색 횟수를 줄인다. 이 방법은 기존 신축 움직임 추정 방법보다 탐색 횟수를 줄였지만, 여전히 신축비율을 결정하기 위해서는 다양한 신축 비율에 대해 탐색을 수행하여야 하는 문제가 있다.

색상 영상에 대해 신축 움직임 추정에 필요한 신축 비율을 추정하기 위해 카메라와 객체의 거리를 측정할 수 있는 깊이 카메라를 이용하는 방법[16]에서는 영상 내 객체가 신축이 일어났을 때 객체와 카메라의 거리와 영상에서의 객체의 크기의 관계를 이용하여 신축 움직임 추정을 수행할 때 신축 비율을 결정한다. Fig. 4는 이 방법에 대한 흐름도이다.

이 때 객체와 카메라간의 거리 R 과 객체가 차지하는 화소의 개수 P 의 관계는 식 (1)과 같이 나타낼 수 있다.

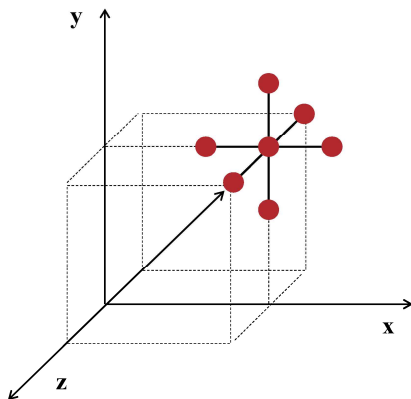


Fig. 3. Method of searching reference block by 3D-Diamond search pattern.

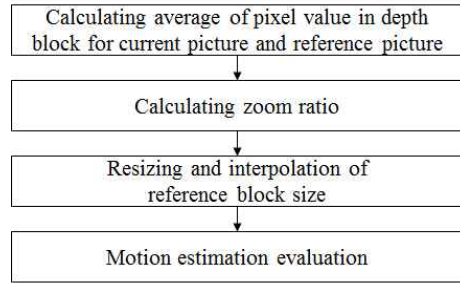


Fig. 4. Flowchart of motion estimation method through depth image.

$$P = \frac{\alpha}{R^\beta} + \gamma \tag{1}$$

현재 블록 내 객체가 차지하는 화소의 개수와 카메라와의 거리를 각각 P_c, R_c 라 하고, 참조 블록 내 객체가 차지하는 화소의 개수와 카메라와의 거리를 P_r, R_r 라 할 때, 만약 식 (1)에서 γ 가 충분히 작다면, 현재 블록과 참조 블록 내 객체가 차지하는 화소의 비로 표현할 수 있는 신축 비율 s 는 식 (2)와 같이 근사할 수 있다.

$$s = \frac{P_c}{P_r} = \left(\frac{\alpha}{R_c^\beta} + \gamma \right) / \left(\frac{\alpha}{R_r^\beta} + \gamma \right) \cong \left(\frac{R_r}{R_c} \right)^\beta \tag{2}$$

이 때 객체의 거리 R 은 깊이 화면 내 해당 블록 영역의 화소의 평균 \bar{R} 로 표현할 수 있다. 이를 이용하여 신축 비율 s 는 현재 블록 내의 깊이 화소의 평균인 \bar{R}_c 와 참조 블록 내의 깊이 화소의 평균인 \bar{R}_r 로 식 (3)과 같이 구할 수 있다.

$$s = \left(\frac{\bar{R}_r}{\bar{R}_c} \right)^\beta \tag{3}$$

그 후 신축 비율 s 에 따라 참조 블록의 크기를 신축한 후 현재 블록의 크기로 보간하여 움직임 추정을 수행한다. 이 방법은 기존 신축 움직임 추정과 달리 신축 비율에 대한 탐색을 수행할 필요 없이 탐색 영역 내 참조 블록에 대해 신축 비율을 바로 구하여 움직임 추정을 할 수 있다.

3. 깊이 영상 부호화에서 신축을 통한 움직임 추정 방법

깊이 영상 부호화에서 신축 움직임 추정 방법은 Fig. 5와 같이 이루어진다. 움직임 추정의 단위로 블

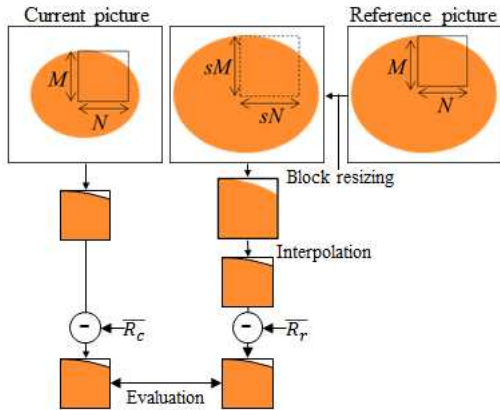


Fig. 5. Zoom motion estimation method for depth video coding.

록은 $M \times N$ 크기로, 탐색 영역은 $m \times n$ 로 가정한다. 먼저 현재 블록의 화소 평균 \bar{R}_c 와 탐색 영역 내 참조 블록의 화소 평균 \bar{R}_r 를 계산한다. 그 후 \bar{R}_c 와 \bar{R}_r 로부터 식 (3)을 통해 신축 비율 s 를 계산한다. 그 후 참조 블록 크기를 폭과 너비에 각각 s 를 곱한 $sM \times sN$ 크기로 변경한다. 그 후 변경된 크기를 가지는 참조 블록을 현재 블록의 크기로 보간한다.

현재 블록과 참조 블록의 오차를 측정하여 움직임 추정에 대한 평가척도를 적용하고 현재 블록과 가장 가까운 참조 블록을 찾고, 움직임 벡터와 신축 벡터를 결정한다.

이 때 객체에 신축이 발생할 경우 객체의 크기가 변하는 것과 더불어 객체 내의 화소 값도 변한다. Fig. 6은 깊이 화면 간 객체의 움직임에 따른 화소 값의 변동의 예를 보여준다. 객체가 신축되지 않고 좌우로만 움직인다면 객체 영역 내의 화소 값의 변동은 생기지 않는다. 하지만 객체가 신축되는 경우에는

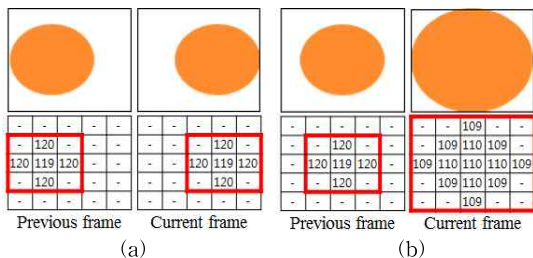


Fig. 6. Change in pixel value by motion in depth video: (a) Motion without zoom and (b) Motion with zoom.

객체 영역내의 값이 화소 값의 변화가 발생한다.

따라서 현재 블록과 참조 블록의 화소값을 그대로 이용하여 움직임 추정에 대한 평가척도에 이용한다면 신축으로 인해 발생된 화소값의 변화로 인해 올바른 움직임 추정이 어렵다. Fig. 7에서는 신축이 일어난 객체에 대해 신축 비율을 통해 참조 블록의 크기를 변경하고, 변경된 크기의 참조 블록을 현재 블록의 크기로 보간함으로써 객체의 형태적 특징은 일치시켰다. 하지만 신축으로 인한 화소값의 변경으로 현재 블록과 참조 블록간의 화소 값이 다른 것을 알 수 있다. 올바른 움직임 추정을 위해서는 이러한 신축으로 인해 발생된 화소값의 변화를 고려하여 현재 블록과 참조 블록을 비교해야 한다. 객체의 신축이 일어날 경우, 객체 내 전체 화소는 신축이 일어나기 전의 화소 값으로부터 신축 거리에 따라 값이 변하는 것을 알 수 있다. 이러한 신축 거리는 현재 블록과 참조 블록 내 화소 평균들을 통해 알 수 있다. 이를 이용하여 신축의 영향을 제거하여 움직임 추정의 평가를 수행하기 위해 다음과 같은 과정을 수행한다. 현재 블록의 각 값이 화소의 값에서 \bar{R}_c 를 뺀 새로운 블록을 생성한다. 그 후 움직임 추정을 위해 참조 블록을 탐색한 후 탐색된 참조 블록에서도 마찬가지로 \bar{R}_r 를 뺀 새로운 블록을 생성한다. 그 후 블록의 평균 값을 뺀 두 블록간의 SAE(Sum Absolute Error) 혹은 SSE(Sum Square Error)를 계산하여 움직임 추정에 대한 평가척도로 이용한다. 그 후 현재 블록과 제일 유사한 참조 블록을 찾아 움직임 벡터와, 신축 비율의 정보를 가지는 신축 벡터를 결정한다.

이 때 현재 블록이 객체 영역이 아닌 배경 영역일 경우에는 객체의 움직임이 없기 때문에 움직임 추정을 수행하지 않는 것이 효율적이다. 움직임 추정을

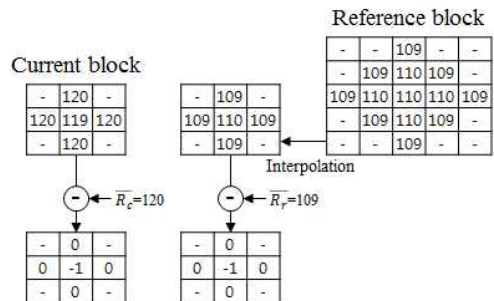


Fig. 7. Evaluation of motion estimation in depth image.

수행하고자 하는 블록이 배경 영역 내에 있을 경우, 현재 화면과 참조 화면 간의 화소값들의 변화는 없고, 신축 비율 s 는 1과 매우 가까운 값이 계산된다. 이를 이용하여 배경 영역을 검출하기 위해서 탐색을 수행하기 전 단계에서 참조 화면 내 현재 블록과 같은 위치에 존재하는 블록에 대해 신축 비율 s 를 구한 후 s 가 $T_1 < s < T_2$ 범위에 존재할 경우 배경 영역으로 판단하고, 이 경우 움직임 추정을 수행하지 않는 방법을 적용할 수 있다[18].

4. 실험 결과 및 고찰

4.1 직접 촬영영상에 대한 신축 움직임 추정 실험 결과

마이크로소프트가 제작한 Kinect V2 깊이 카메라를 이용하여 깊이 영상 부호화에서 신축을 통한 움직임 추정 방법의 성능을 측정하였다. 이 때 Kinect V2는 ToF(Time of Flight) 방식으로 깊이를 측정하는 깊이 카메라이다. 이 때 깊이 영상 해상도로 512×424 를 사용하였다. 이 때 배경으로 판단하는 임계치 T_1, T_2 는 각각 0.95, 1.05를 적용하였다.

모의실험을 수행하기 위해 각각 다른 모양의 객체를 촬영한 영상에서 각 객체의 거리차이가 일어난 두 화면을 선택하여 Fig. 8-10과 같이 현재 화면과 참조 화면으로 이용하였다. Fig. 8은 상자 모양의 객체를 촬영한 영상이고, Fig. 9는 원통 모양의 객체를 촬영한 영상, Fig. 10은 실제 사람을 대상으로 촬영한 영상이다. 각 화면에서의 평균 \bar{R}_c 와 \bar{R}_r 는 Fig. 8의 경우 1581과 2224, Fig. 9의 경우 2083과 1221, Fig. 10의 경우에는 1502과 2427이다.

Fig. 8-10의 깊이 화면에 대하여 색상화면을 위하여 기존 H.264[20], H.265 등의 표준에 적용된 움직임 추정 방법과 본 논문에서 제안한 신축 움직임 추정

방법을 적용한 결과를 비교하였다. 이 때 블록 크기는 8×8 을 적용하였고, 탐색 영역은 25×25 를 적용하였다. 또한 식 (3)에서의 β 는 0.9를 적용하였다.

움직임 추정을 수행하였을 때, 움직임 벡터는 Fig. 11과 같이 나타났다. 이 때 깊이 영상에 기존 움직임 추정 방법을 적용하였을 때는 Fig. 11 (a)와 같이 실제 객체의 움직임 방향으로 움직임 추정을 하지 못하는 것을 확인할 수 있다. 반면 제안된 신축 움직임 추정 방법을 적용하였을 경우에는 Fig. 11 (b)와 같이 움직임 방향대로 움직임 추정이 되었음을 확인할 수 있다. 기존 방법을 통해 움직임 추정을 한 경우는 탐색 영역 내의 참조 블록들 중 블록 화소 평균 \bar{R}_c 의 값이 제일 작은 참조 블록으로 움직임 추정을 하는 경향을 보였다. 이는 Fig. 11 (a)에서 대체적으로 움직임 벡터가 한 지점으로 뭉쳐있는 것을 통해 확인할 수 있다. 반면 제안된 방법을 사용하였을 때는 움직임 추정으로 인해 생성된 움직임 벡터는 Fig. 11 (b)와 같이 신축 움직임 방향과 일치하는 결과를 보였다.

움직임 추정 정확도를 측정하기 위해 움직임 추정을 통해 예측한 화면과 원 화면 간의 MSE를 구하여 추정 오차를 구하였다. 그 결과 제안된 신축 움직임

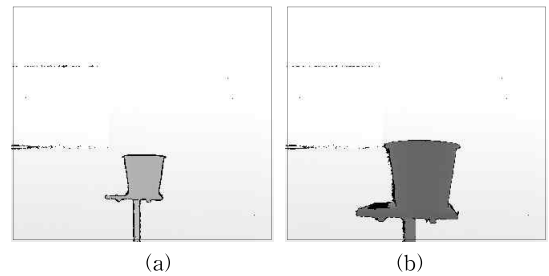


Fig. 9. Depth pictures for cylinder object: (a) Current picture and (b) Reference picture.

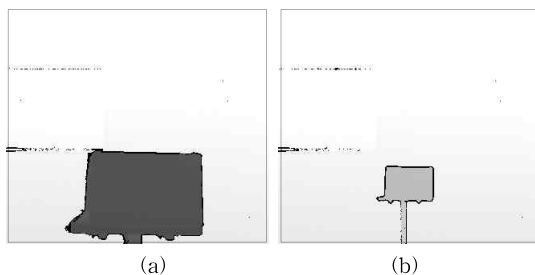


Fig. 8. Depth pictures for box object: (a) Current picture and (b) Reference picture.

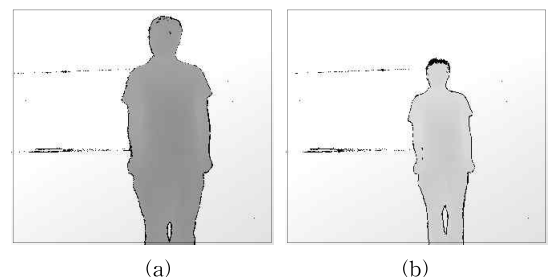


Fig. 10. Depth pictures for person object: (a) Current picture and (b) Reference picture.

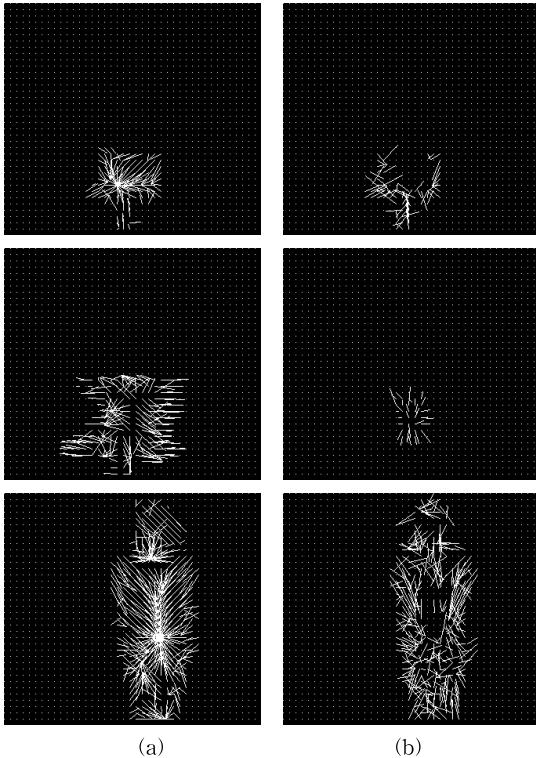


Fig. 11. Motion vector: (a) Conventional method and (b) Proposed method.

Table 1. Error between current picture and predicted picture by motion estimation

Motion estimation method	Source		
	box	cylinder	person
Conventional method	2278.72	2670.00	3027.60
Proposed method	19.56	26.82	25.27

추정방법을 적용하였을 경우에 기존 움직임 추정 방법에 비해 추정 오차가 눈에 띄게 줄어들었음을 보였다. 이 때 색상 영상에서는 화소의 범위인 0~255로 제한된 반면, 깊이 영상에서의 화소의 단위는 거리를 나타내는 mm 단위으로써 본 실험 환경에서는 화소의 범위가 최대 3000mm이기 때문에 깊이 영상을 본 실험에서 측정된 MSE는 색상 영상에서의 움직임 추정

Table 2. Estimation error in depth videos

Motion estimation method	Source				
	desk	kitchen	meeting room	table	round table
Conventional method	710.67	1008.72	1423.89	637.27	863.99
Proposed method	88.67	46.82	98.79	58.17	108.73

에서의 MSE보다 크게 나온다.

4.2 공개된 깊이 영상에 대한 신축 움직임 추정 실험 결과

움직임 추정 방법을 적용한 결과를 실험하기 위해 온라인에 공개된 깊이 영상을 사용하였다[21]. 이 깊이 영상들은 카메라가 여러 방향으로 이동하면서 객체를 촬영한 깊이 영상으로 본 실험의 목적인 신축 움직임 추정의 성능을 평가하기에 적당한 영상이다. 실험에 사용된 깊이 영상은 Microsoft사의 Kinect 깊이 카메라로 촬영되었으며, 320×240의 해상도로 촬영되었다. 각 깊이 영상들은 1초에 20프레임씩 촬영되었다.

신축 움직임이 포함된 깊이 영상에 대해 본 논문에서 제안한 신축 움직임 방법을 적용하여 그 결과를 분석하였다. 이 때 블록 크기는 16×16을 적용하였고, 탐색 영역은 15×15를 적용하였다. 또한 식 (3)에서의 β는 0.9를 적용하였다.

깊이 영상에 대해 움직임 추정 정확도를 비교하기 위해 Fig. 12의 깊이 영상들의 각 프레임 화면에 기존 움직임 추정 방법과 제안된 신축 움직임 추정 방법을 적용하여 MSE를 구하였다. 이 때 참조 화면으로는 현재 화면의 바로 직전 프레임의 화면을 사용하였다. Table 2는 깊이 영상들의 MSE의 평균을 구한 결과이다. 제안된 방법을 이용하여 움직임 추정을 한 결과 추정 오차가 눈에 띄게 개선되었음을 보였다.

각 실험영상에서 영상 초반의 50프레임에 대해 움직임 추정에서 신축을 적용한 경우와 적용하지 않았을 경우의 추정 오차를 Fig. 13과 같이 측정하였다. 이 때 움직임 추정에서 신축을 적용하지 않았을 경우에는 영상에서 신축이 발생하였을 경우 추정 오차가 급격히 높아졌지만, 신축을 적용하여 움직임 추정을 하였을 경우에는 신축 움직임이 발생하더라도 추정 오차가 높아지지 않았다.

또한 탐색 영역의 크기를 달리하여 Fig. 14와 같이 추정 오차를 비교하였다. 그 결과 탐색 영역의 크기가 10 또는 15가 될 때 까지는 탐색 영역의 크기가

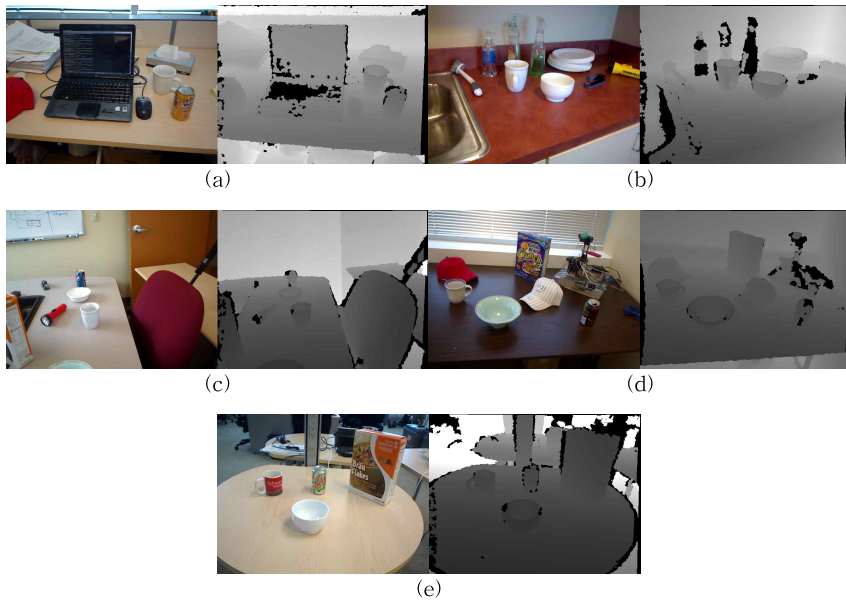


Fig. 12. Data set of depth video for simulation of zoom motion estimation for depth video coding: (a) desk, (b) kitchen, (c) meeting room, (d) table, and (e) round table.

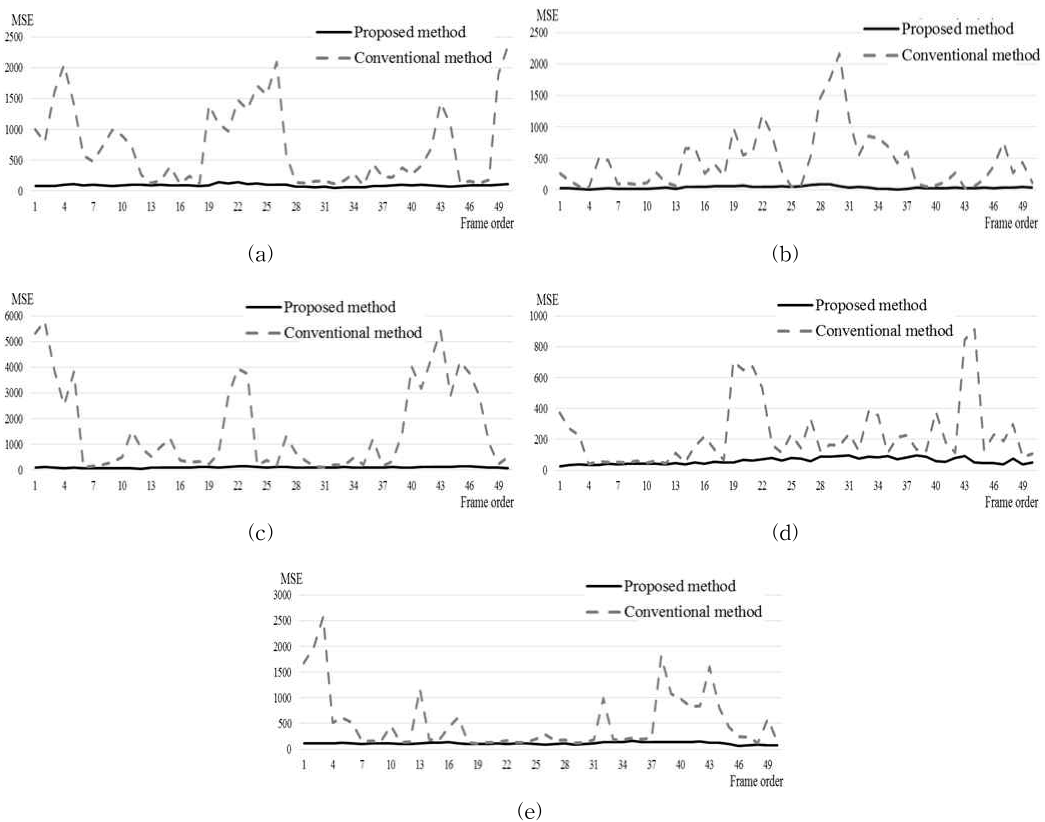


Fig. 13. Estimation error by frame order: (a) desk, (b) kitchen, (c) meeting room, (d) table, and (e) round table.

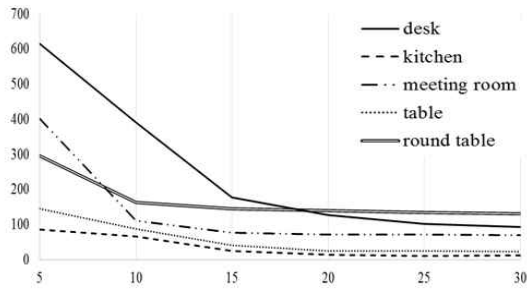


Fig. 14. Estimation error by search range.

증가할수록 움직임 추정 성능이 대폭 개선되었지만, 그 이상의 탐색 영역 크기에 대해서는 움직임 추정 성능 개선의 정도가 줄어들었다.

5. 결 론

본 논문에서는 깊이 영상에서 움직임 추정을 수행할 때 발생하는 신축 움직임을 추정하기 위해 탐색 블록과 현재 블록의 각각의 깊이의 평균을 구하고 블록 내 각 화소마다 블록 평균을 빼어 움직임 추정을 수행하는 방법을 제안하였다. 제안된 방법을 통해 움직임 추정을 수행한 결과로 제안된 신축 움직임 추정 방법을 사용하지 않았을 경우에 비해 움직임 추정 오차의 객관적인 평가척도인 MSE가 크게 줄었음을 확인하였다. 특히 제안 방법은 깊이 영상 내에서 신축 움직임이 발생하였을 때에 움직임 벡터가 신축 방향과 거의 일치하게 찾아졌다. 또한 탐색 영역의 크기가 증가할수록 움직임 추정 성능이 좋아지지만, 탐색 영역의 크기가 최대 특징치를 초과할 경우에는 움직임 추정 개선이 이루어지지 않았다.

REFERENCE

[1] D.S. Lee and S.K. Kwon, "A Recognition Method for Moving Objects Using Depth and Color Information," *Journal of Korea Multimedia Society*, Vol. 19, No. 4, pp. 681-688, 2016.

[2] S.K. Kwon and D.S. Lee, "Correction of Perspective Distortion Image Using Depth Information," *Journal of Korea Multimedia Society*, Vol. 18, No. 2, pp. 106-112, 2015.

[3] J. Preis, M. Kessel, M. Werner, and C. Lin-

nhoff-Popien, "Gait Recognition with Kinect," *Proceeding of the First Workshop on Kinect in Pervasive Computing*, pp. P1-P4, 2012.

[4] R.M. Salinas, R.M. Carnicer, F.J.M. Cuevas, and A.C. Poyato, "Depth Silhouettes for Gesture Recognition," *Pattern Recognition Letters*, Vol. 29, No. 3, pp. 319-329, 2008.

[5] P. Suryanarayan, A. Subramanian, and D. Mandalapu, "Dynamic Hand Pose Recognition Using Depth Data," *Proceeding of 20th International Conference on Pattern Recognition*, pp. 3105-3108, 2010.

[6] M. Zollhoefer, M. Martinek, G. Greiner, M. Stamminger, and J. Suessmuth, "Automatic Reconstruction of Personalized Avatars from 3D Face Scans," *Computer Animation and Virtual Worlds*, Vol. 22, No. 2-3, pp. 195-202, 2011.

[7] J. Tong, J. Zhou, L. Liu, Z. Pan, and H. Yan, "Scanning 3D Full Human Bodies Using Kinects," *IEEE Transactions on Visualization and Computer Graphics*, Vol. 18, No. 4, pp. 643-650, 2012.

[8] F. Jäger, "Simplified Depth Map Intra Coding with An Optional Depth Lookup Table," *Proceeding of 2012 International Conference on 3D Imaging*, pp. 1-4, 2012.

[9] S. Liu, P. Lai, D. Tian, and C.W. Chen, "New Depth Coding Techniques with Utilization of Corresponding Video," *IEEE Transactions on Broadcasting*, Vol. 57, No. 2, pp. 551-561, 2011.

[10] B.T. Oh, J. Lee, and D.S. Park, "Depth Map Coding Based on Synthesized View Distortion Function," *IEEE Journal of Selected Topics in Signal Processing*, Vol. 5, No. 7, pp. 1344-1352, 2011.

[11] S.H. Park and J.S. Yoo, "Depth Map Pre-processing Using Gaussian Mixture Model and Mean Shift Filter," *Journal of the Korea Institute of Information and Communication Engineering*, Vol. 15, No. 5, pp. 1155-1163, 2011.

[12] D.S. Lee and S.K. Kwon, "Improvement of Depth Video Coding by Plane Modeling," *Journal of the Korea Industrial Information Systems Research*, Vol. 21, No. 5, pp. 11-17, 2016.

[13] S.K. Kwon, D.S. Lee, and Y.H. Park, "Depth Video Coding Method for Spherical Object," *Journal of the Korea Industrial Information Systems Research*, Vol. 21, No. 6, pp. 23-29, 2016.

[14] L.M. Po, K.M. Wong, K.W. Cheung, and K.H. Ng, "Subsampled Block-Matching for Zoom Motion Compensated Prediction," *IEEE Transactions on Circuit Systems for Video Technology*, Vol. 20, No. 11, pp. 1625-1637, 2010.

[15] 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 for Video Technology*, Vol. 22, No. 9, pp. 1280-1291, 2012.

[16] S. Sorwar, M. Murshed, and L. Dooley, "Fast Global Motion Estimation Using Iterative Least-Square Estimation Technique," *Processing of International Conference on Information Communications and Signal processing*, pp. 282-286, 2003.

[17] S.K. Kwon, Y.H. Park, and K.R. Kwon, "Zoom Motion Estimation Method by Using Depth Information," *Journal of Korea Multimedia Society*, Vol. 16, No. 2, pp. 131-137, 2013.

[18] W.S. Jang, O.J. Kwon, and S.K. Kwon, "Adaptive Zoom Motion Estimation Method," *Journal of Korea Multimedia Society*, Vol. 17, No. 8, pp. 915-922, 2014.

[19] S. Zhu and K. Ma, "A New Diamond Search Algorithm for Fast Block-matching Motion Estimation," *IEEE Transactions on Image Processing*, Vol. 9, No. 2, pp. 287-290, 2000.

[20] S.K. Kwon, A. Tamhankar, and K.R. Rao, "Overview of H.264/MPEG-4 Part 10," *Journal of Visual Communication and Image Representation*, Vol. 17, No. 2, pp. 186-216, 2006.

[21] K. Lai, L. Bo, X. Ren, and D. Fox, "A Large-Scale Hierarchical Multi-view RGB-D Object Dataset," *Processing of IEEE International Conference on Robotics and Automation*, pp. 1817-1824, 2011.



이 동 석

2015년 2월 동의대학교 컴퓨터소프트웨어공학과 졸업
 2017년 2월 동의대학교 컴퓨터소프트웨어공학과 석사
 2017년 3월~현재 동의대학교 컴퓨터소프트웨어공학과 박사과정

관심분야 : 멀티미디어신호처리, 영상인식



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 방문 교수
 2010년 9월~2011년 8월 Massey University 방문 교수
 2001년 3월~현재 동의대학교 컴퓨터소프트웨어공학과 교수

관심분야 : 멀티미디어신호처리, 영상통신