

# 시점의 공간 시간적 특징 기반 외삽 영상 홀 채움 기법의 개선

이 규철\*, 유지상<sup>o</sup>

## Improvement of Hole Filling Method of Extrapolation View Based on View-Spatial-Temporal Features

Gyu-Cheol Lee\*, Jisang Yoo<sup>o</sup>

### 요 약

본 논문에서는 외삽으로 가상 시점 영상 생성 시에 발생하는 홀 영역을 효과적으로 채우는 VST 기반의 홀 채움 기법을 제안한다. 기존의 VST 기법은 스테레오 영상을 이용하여 내삽 영상을 생성한다. 가상 시점 생성 시 발생하는 홀 영역을 채우기 위해 시점, 공간적, 시간적 유사성을 가지는 8개의 주변 영상을 탐색한다. 하지만 탐색하는 영상의 개수가 많을 뿐만 아니라 각각의 영상을 풀 서치 방식으로 탐색하기 때문에 속도가 매우 느리다는 단점을 가지고 있다. 외삽 시점에 적용하는 경우에는 내삽 시점보다 홀 영역이 더 많이 생성되기 때문에 VST 기법을 적용하면 시간이 더 많이 소요된다. 따라서 본 논문에서는 외삽 영상 생성 시 검색 범위를 줄여 계산량을 줄이는 동시에 성능은 유지하는 VST 기반의 홀 채움 기법을 제안한다. 먼저 참조 영상의 사용 통계를 분석하여 참조 영상의 개수를 줄였다. 두 번째로 참조 영상에서 블록 탐색 시 n-step 블록 탐색 기법을 적용하였다. 실험을 통해 제안하는 기법이 VST 기법에 비해 속도는 약 8.5배, PSNR은 약 0.78dB 상승한 것을 확인하였다.

**Key Words** : Autostereoscopy, Multi-view, Extrapolation, VST, Hole-filling

### ABSTRACT

In this paper, we propose a hole filling method based on VST that effectively fills the hole area generated by view extrapolation. The conventional VST method generates an interpolated view using a stereo image. In order to fill the hole region occurred when the virtual view is generating, eight images having a viewpoint, spatial, and temporal similarity to the reference image are searched. However, since the number of images to be searched is large and each image is searched by the full search method, the method has a disadvantage of low speed. In case of applying the VST method to view extrapolation, since the hole area is generated more than when view interpolation is applied, it takes more time. Therefore, we propose a new VST-based hole filling technique that reduces search range and computational complexity while maintaining the performance in view extrapolation. First, the number of reference images is reduced by analyzing the usage statistics of reference images. Second, n-step block search method is applied to block search in reference image. Experimental results show that the proposed method increases the speed by 8.5 times and the PSNR by 0.78dB compared with the VST method.

\* 본 연구는 과학기술정보통신부 및 정보통신기술진흥센터의 대학ICT연구센터육성지원사업의 연구결과로 수행되었음 (IITP-2017-2016-0-00288)

• First Author : Kwangwoon University Department of Electronic Engineering, gyucheol0116@gmail.com, 학생회원

o Corresponding Author : Kwangwoon University Department of Electronic Engineering, jsyoo@kw.ac.kr, 종신회원

논문번호 : KICS2017-04-132, Received April 30, 2017; Revised July 3, 2017; Accepted July 19, 2017

## I. 서 론

디지털 미디어 기술의 급속한 발전과 새로운 콘텐츠 요구가 커짐에 따라 실제와 같은 경험이 가능한 3차원 비디오 기술에 대한 연구가 활발히 진행되고 있다. 3차원 비디오 기술은 3D 영화, TV, 게임 등 다양한 분야에 적용되고 있다. 2009년에 개봉한 아바타를 선두로 많은 영화가 3D 형태로 개봉함으로써 관객들에게 역동적이고 생동감 있는 장면을 제공하고 있다. TV에서는 지상파 방송사가 2010년에 3DTV 본방송을 실시했으며 66번 채널을 통해 남아공 월드컵을 비롯하여 드라마, 다큐멘터리, 어린이 교육프로그램 등을 방송했다.

고정된 위치에 있는 다수의 시청자에게 또는 시청자의 위치가 바뀌더라도 입체감을 느낄 수 있는 다시점 3차원 기술도 개발되고 있다. 오토스테레오스코픽(auto-stereoscopic) 디스플레이는 사용자가 안경 없이도 입체감을 느끼고 사용자 위치에 해당하는 시점 영상을 제공한다. 다시점 영상을 획득하기 위해서는 각각의 시점에 대응되는 위치에 카메라를 설치하여 촬영하는 방법이 일반적이다. 하지만 다수의 카메라를 이용하는 방법은 카메라 환경 설정이 어렵다는 점과 비용이 많이 든다는 문제가 발생한다. 이에 대한 대안은 3D 워핑<sup>[1]</sup>으로 불리는 DIBR<sup>[2]</sup> 방식으로, 한 장의 컬러 영상과 대응하는 깊이 영상을 이용하여 기하학적 투영을 통해 가상시점 영상을 생성하는 것이다. DIBR<sup>[2]</sup> 방식은 효율적인 대역폭과 시점의 생성이 자유롭다는 점, 2D에서 3D로의 변환이 용이하다는 점 그리고 카메라를 많이 사용하지 않기 때문에 비용이 적게 드는 점 등의 장점을 가지고 있다. 또한 하나의 원본 영상을 이용하여 서로 다른 시점을 생성하기 때문에 생성한 시점들은 서로 대칭적이다. 이와 같은 이유로 MPEG 3D-HEVC의 차세대 표준에도 포함되어 있다.

DIBR<sup>[2]</sup> 기법에서 시점을 생성하는 방법은 먼저 참조 영상의 픽셀을 깊이 영상과 카메라 파라미터를 이용하여 세계 좌표계(world coordinate)로 투영시킨 후 가상 시점의 좌표계로 재투영한다. 하지만 이 방법은 시점을 생성할 때 현재의 시점에서는 보이지만 다른 시점에서 보이지 않는 영역을 완벽히 복원할 수 없기 때문에 가려짐 영역이라고 불리는 일종의 홀(hole)이 발생하게 된다. 뿐만 아니라 생성한 깊이 영상의 값이 잘못된 경우에도 홀이 발생한다. 일반적으로 홀을 채우는 것은 가려짐 영역에서 깊이 영상이나 참조 영상으로부터 이용할 정보가 없기 때문에 매우 어려운 문

제이다. 따라서 홀 채움 결과가 최종적으로 생성된 3D 콘텐츠의 품질을 좌우하게 된다.

홀 채움을 위해 많은 기법들이 제안되었다. L. Zhang<sup>[3]</sup>은 3D 워핑 시 수평 방향으로 깊이 값의 변화가 큰 영역에서 홀이 발생하기 때문에 가우시안 필터를 이용하여 깊이 영상을 부드럽게 만들어주는 전처리 과정을 적용하였다. 깊이 영상에 대한 가우시안 필터링은 깊이 값의 변화가 큰 영역을 부드럽게 하기 때문에 3D 워핑<sup>[1]</sup> 시 발생하는 홀의 크기를 줄일 수 있다. 홀의 크기가 작기 때문에 미디언 필터를 이용하여 쉽게 홀을 채울 수 있으나 깊이 영상을 가우시안 필터링하는 과정에서 기하학적인 왜곡이 발생한다.

M. Solh의 Hierarchical Hole-Filling(HHF)<sup>[4]</sup>은 먼저 홀이 발생한 영상을 5x5 가우시안 커널(kernel)을 이용하여 홀이 없어질 때까지 다운샘플링 한다. 일반적으로 3번 다운샘플링 하면 해상도는 낮지만 홀이 없어진 영상을 획득할 수 있다. 획득된 영상을 다시 원본 영상 크기로 업샘플링하여 홀 영역과 일대일 맵핑을 통해 홀을 채운다. HHF는 L. Zhang이 제안한 기법<sup>[3]</sup>과 달리 변이 영상의 전처리 과정 없이 홀을 채우기 때문에 기하학적인 왜곡이 발생하지 않는다. 하지만 업샘플링한 영상을 가지고 홀을 채우기 때문에 홀을 채운 영역에 블러 현상이 발생한다.

S. Choi<sup>[5]</sup>는 가상 시점 생성 후 발생하는 홀의 경계를 Random Walks<sup>[6]</sup>를 이용하여 전경과 배경으로 분리하고 홀을 배경으로 채워나갔다. 전경과 배경을 구분하기 때문에 홀을 효과적으로 채울 수 있으며 홀을 채울 패치(patch)를 탐색할 때 현재 프레임뿐만 아니라 앞 뒤 프레임을 모두 탐색하기 때문에 시간적 일관성이 보장된다는 장점이 있다. 하지만 Random Walks 기법<sup>[6]</sup>은 입력 영상 크기의 4배가 되는 행렬 연산을 수행하기 때문에 QHD (960x540) 크기의 영상에서 프레임 당 3시간이 소요될 정도로 시간이 오래 걸리는 단점을 가지고 있다.

Inpainting 기법<sup>[7]</sup>은 영상의 훼손된 영역을 복원하는데 사용되는 기법으로써 가상 시점 생성 시 발생하는 홀 영역을 채우기 위해서도 사용된다. 홀 영역의 경계에서부터 시작하여 경계의 범선에 가까울수록 경계 윤곽선 근접할수록 높은 가중치를 가지게 된다. 가중치가 높은 좌표부터 인접 픽셀들을 이용하여 홀을 채워나가며 홀을 다 채울 때까지 가중치 계산을 반복적으로 수행한다. 가상 시점의 홀을 채울 경우에는 깊이에 대한 파라미터를 추가하여 배경영역에 가중치를 더 주어 채워나간다. 높은 품질을 기대할 수 있지만 가중치를 계산하는 과정에서 오랜 시간이 소요되며

홀 영역이 뭉개져 채워진다는 단점이 존재한다.

L. Zhu가 제안한 기법<sup>[8]</sup>은 일반적으로 비디오는 서로 인접한 프레임 간 공간적(spatial), 시간적(temporal) 유사성을 가진다는 점을 착안한다. 추가적으로 스테레오 영상은 이 두 가지 유사성 외에도 부가적으로 시점(view)간의 유사성을 가지고 있다. VST(View-Spatial-Temporal) 기법<sup>[8]</sup>은 이러한 시점, 공간, 시간의 유사성을 활용하여 가상 시점을 생성한다. 현재 프레임에 기준으로 8개의 참조 영상에서 블록 매칭을 통해 가장 작은 비용을 갖는 블록을 탐색하여 홀을 채운다. VST 기법<sup>[8]</sup>은 블록 매칭을 기반으로 홀을 채우기 때문에 블러 현상이 발생하지 않으며 시점, 공간, 시간 유사성을 고려하기 때문에 생성된 가상 시점의 품질이 뛰어나다. 하지만 블록을 잘못 찾게 되면 에러를 전파하는 문제도 있다. 특히 변이 정보가 부정확한 경우에 가상시점을 생성하면 전경과 배경의 경계선 부근에서 에러가 발생할 가능성이 높다. 이 영역에 블록 매칭을 적용하게 되면 경계선 에러가 전파되어서 더 큰 에러를 초래한다. 또한 8개의 영상을 참조하여 적합한 블록을 탐색하는 시간이 너무 많이 소요된다. Full HD(1920 x 1080) 영상에서 한 프레임 당 50분 정도가 소요된다.

본 논문에서는 외삽 영상에서 발생하는 홀 영역을 채우기 위하여 VST<sup>[8]</sup> 기반의 홀 채움 기법을 제안한다. 기존 VST 기법<sup>[8]</sup>이 가지고 있는 많은 계산량으로 발생하는 속도 저하의 문제를 해결하기 위해 검색 영상의 범위를 줄이고, 효과적으로 홀을 채우기 위해 풀 서치 방식이 아닌 n-step 블록 탐색 기법을 통해 블록을 탐색한다. 또한 경계 영역에서 발생하는 에러를 줄이기 위해 변이 영상의 홀을 채울 때 전경이 아닌 배경 성분만을 이용하여 홀을 채움으로써 블록 매칭의 비용 계산 시 경계 영역에서 발생하는 에러를 차단한다.

본 논문의 구성은 다음과 같다. 2장에서는 VST 기법<sup>[8]</sup>에 대해 설명하고 3장에서는 제안하는 기법에 대한 설명을 한다. 4장에서는 실험 및 실험 결과를 분석하며 5장에서는 결론을 맺는다.

## II. VST 홀 채움 기법

일반적으로 비디오에서 이웃한 프레임은 공간적, 시간적 유사성을 가진다. 공간적 유사성이란 한 장의 프레임에서 이웃 픽셀 간 색상이 비슷한 것을 의미한다. 시간적 유사성이란 앞 뒤 프레임은 시간차가 거의 없기 때문에 비슷한 색상을 가지는 것을 의미한다. 스테레오 영상으로 구성된 비디오의 경우 이 두 가지 유

사성 이외에 시점간의 유사성을 가지고 있다. 시점간의 유사성이란 서로 다른 시점 영상 간 유사한 특징을 의미한다. VST (View-Spatial-Temporal) 홀 채움 기법은 시점(view), 공간(spatial), 시간(temporal) 등 세 가지의 유사성을 활용하여 가상 시점을 생성한다<sup>[8]</sup>. 그림 1은 VST 기법의 전체적인 흐름을 나타낸다.

VST 기법은 크게 네 단계로 구성되어 있다. 첫 번째는 marking holes 과정으로 가상 시점 생성 시 발생하는 홀들을 작은 홀과 큰 홀로 구분한다<sup>[8]</sup>. 일반적으로 가상 시점 생성은 실수 연산인 반면에 채투영하는 픽셀의 위치는 정수형이기 때문에 일대일 맵핑이 되지 않는 문제로 인하여 작은 홀들이 발생한다. 큰 홀들은 가상 시점 생성 시 참조 영상에서 가상 시점의 특징 영역에 대한 정보가 없기 때문에 발생한다. 홀들이 생성되는 원인이 다르기 때문에 각각 다른 기법을 적용하여 홀들을 제거해야 한다. VST 기법에서는 미디안 필터(median filter)를 이용하여 작은 홀과 큰 홀을 구분한다. 예를 들어 그림 2와 같이 3x3 크기의 미디안 필터를 적용했을 때 1차원으로 나열한 픽셀 배열의 중앙값이 홀이 아닌 경우는 작은 홀로 반대의 경우는 큰 홀로 구분한다. VST 기법을 실제로 적용할 경우에는 5x5 미디안 필터를 사용한다.

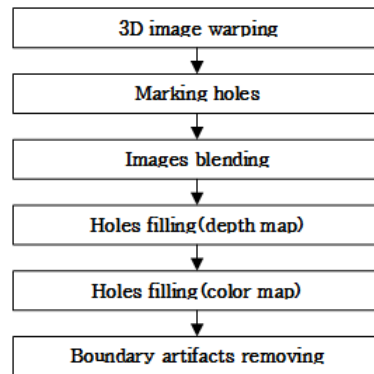


그림 1. VST 홀 채움 기법의 흐름도  
Fig. 1. Flow chart of VST hole filling method

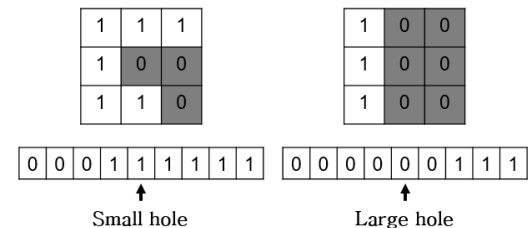


그림 2. Marking holes 과정  
Fig. 2. Process of marking holes

그림 3은 양방향 선형 보간법<sup>[9]</sup>을 이용하여 생성한 가상 시점 영상과 분류된 홀들을 보여준다. 사용한 영상은 Full HD (1920x1080) 크기를 가지고 있는 Avengers 영상이다. 그림 3(a)는 생성된 가상시점에서 발생한 검은색 의 홀을 그림 3(b)에서 marking holes 과정을 통해 홀들을 구분한 결과를 각각 보여준다. 빨간색은 큰 홀을 의미하고 파란색은 작은 홀들을 의미한다.

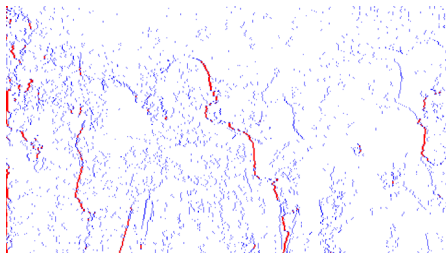
VST 기법의 두 번째 단계에서는 먼저 깊이 영상의 홀을 채운다<sup>[8]</sup>. 변이 영상은 칼라 영상보다 단순하며 높은 공간 유사성을 가지고 있기 때문에 기존에 많이 사용되는 식 (1)의 spatial weighted interpolation<sup>[7]</sup>을 사용하여 구한다.

$$D(u,v) = \frac{\sum_{(x,y) \in w(u,v)} \rho(x,y)d(x,y)D(x,y)}{\sum_{(x,y) \in w(u,v)} \rho(x,y)d(x,y)} \quad (1)$$

여기서  $w(u,v)$ 는 3x3 윈도우 내 픽셀을 의미한다.  $d(x,y)$ 는 윈도우 중앙 픽셀과 다른 픽셀 간의 유클리디언 거리를 의미하며  $D(x,y)$ 는 깊이 값을 나타낸다.  $\rho(x,y)$ 는  $(x,y)$  좌표의 픽셀의 홀 여부를 판단하는 파라



(a)



(b)

그림 3. 가상 시점과 Marking holes 결과 (a) 양방향 선형 보간법을 이용하여 생성한 가상 시점 영상 (b) Marking holes 결과  
Fig. 3. Virtual view generated by bidirectional linear interpolation (b) Result of marking holes

미터이며 홀인 경우는 0, 그렇지 않은 경우는 1이 된다.  $D'(u,v)$ 는 좌표  $(u,v)$ 에서의 변이 값이다.

그림 4는 spatial weighted interpolation<sup>[8]</sup>을 이용하여 깊이 영상의 홀을 채운 결과이다. 그림 4(a)는 홀 채움 전 가상 시점의 변이 영상이고 그림 4(b)는 홀 채움 후 결과이다. 변이 영상은 이웃 화소 간 유사성이 높기 때문에 spatial weighted interpolation<sup>[7]</sup> 방법으로 홀이 효과적으로 채워지는 것을 확인할 수 있다.

VST 기법의 세 번째 단계에서는 컬러 영상의 작은 홀과 큰 홀을 채운다<sup>[8]</sup>. 작은 홀은 실수 연산으로 인하여 발생한 홀이기 때문에 이웃 픽셀간의 유사성이 매우 높다. 따라서 깊이 영상과 동일하게 spatial weighted interpolation 방법을 이용하여 홀을 채운다. 큰 홀은 주로 가려짐 영역에 의하여 발생한다. 큰 홀도 작은 홀과 마찬가지로 주변 영역과의 유사성이 있지만 홀이 크기 때문에 공간 유사성 외에 시점 및 시간 유사성을 같이 고려하여 홀을 채운다. 그림 5는 큰 홀을 채우기 위해 참조하는 프레임들을 보여준다. V는 가상 시점, L은 좌 영상, R은 우 영상을 의미하며 k는 프레임을 의미한다. (k+1) 번째 프레임의 가상 시점,  $V_{k+1}$ 의 홀을 채우기 위해  $V_{k+1}$ 을 포함하여 총 8개의 영상에서 적합한 블록을 블록 매칭을 이용하여 탐색한다.

블록 매칭 전에 먼저 채울 홀의 우선순위를 정한다. 일반적으로 먼저 채울 블록은 이용 가능한 픽셀 정보



(a)



(b)

그림 4. 홀 채움 전과 후의 변이 영상 비교 (a) 홀 채움 전 (b) 홀 채움 후  
Fig. 4. Comparisons on the disparity maps before and after hole-filling, (a) Before hole-filling (b) After hole-filling

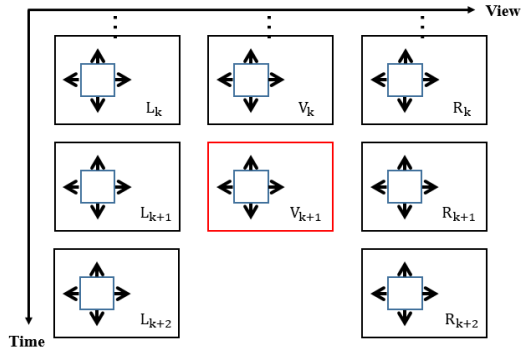


그림 5. 큰 홀을 채우기 위해 참조하는 프레임들  
Fig. 5. Frames referred to fill a large hole

가 많기 때문에 높은 우선순위를 가진다. 우선순위 비용은 블록 내의 홀이 아닌 픽셀의 비율과 컬러 영상과 깊이 영상의 텍스처의 구조를 이용하여 계산한다. 일반적으로 홀이 아닌 픽셀의 비율이 높고 컬러 영상과 깊이 영상의 텍스처 구조가 복잡할수록 채우기 쉽기 때문에 높은 우선순위를 가진다. 픽셀 좌표  $(u,v)$ 에 대한 우선순위 비용  $P(u,v)$ 는 식 (2)와 같이 표현된다.

$$P(u,v) = C(u,v)((1 - \lambda_D)I_D(u,v) + \lambda_D D_D(u,v)) \quad (2)$$

여기서  $\lambda_D$ 는 가중치 파라미터이다.  $C(u,v)$ 는 블록  $w(u,v)$ 에서 홀이 아닌 픽셀의 비율이며 식 (3)을 통해 계산된다.

$$C(u,v) = \frac{1}{N_{w(u,v)}} \sum_{(x,y) \in w(u,v)} \rho(x,y) \quad (3)$$

여기서  $N_{w(u,v)}$ 는 블록  $w(u,v)$ 의 픽셀의 개수이며  $\rho(x,y)$ 는 좌표  $(x,y)$ 에서 홀이면 0, 홀이 아니면 1을 나타낸다.

$ID(u,v)$ 는 블록  $w(u,v)$ 에서 YCbCr 컬러 공간의 변화량을 나타내며 식 (4)와 같이 표현된다.

$$I_D(u,v) = \sum_{(x,y) \in w(u,v)} \rho(x,y) |I(x,y) - I_{avg}(u,v)| \quad (4)$$

여기서  $I(x,y)$ 는 좌표  $(x,y)$  픽셀의 밝기 값을 의미하며  $I_{avg}(u,v)$ 는 블록  $w(u,v)$ 의 평균 밝기 값이다.

마찬가지로  $DD(u,v)$ 는 블록  $w(u,v)$ 에서 깊이 영상의 변화량을 나타내며 식 (5)와 같이 표현된다.

$$D_D(u,v) = \sum_{\substack{(x,y) \in \\ w(u,v)}} \rho(x,y) |D(x,y) - D_{avg}(u,v)| \quad (5)$$

여기서  $D(x,y)$ 는 좌표  $(x,y)$ 의 깊이 값을 의미하며  $D_{avg}(u,v)$ 는 블록  $w(u,v)$ 의 평균 깊이 값을 나타낸다.

먼저 채움 블록이 정해졌다면 어디에 있는 블록으로 채울지 탐색한다. 그림 5의 총 8개의 참조 영상(L, L1, L2, V, V1, R, R1, R2)에서 블록 매칭을 수행하며 블록 후보들 중에서 가장 낮은 비용을 갖는 블록을 찾아 홀을 채운다. 블록의 비용을 계산하는 수식은 식 (6)과 같다.

$$E(u + \Delta u, v + \Delta v, t, k) = \sum_{\substack{(x,y) \in \\ w(u,v)}} \rho(x,y,k) |I_k(x,y) - M_k(c, x + \Delta u, y + \Delta v, t)| + \lambda \sum_{\substack{(x,y) \in \\ w(u,v)}} \rho(x,y,k) |D_k(x,y) - M_k(d, x + \Delta u, y + \Delta v, t)| \quad (6)$$

여기서  $(\Delta u, \Delta v)$ 는 탐색 벡터를 의미하며  $\lambda$ 는 상수,  $E(u + \Delta u, v + \Delta v, t, k)$ 는 블록  $w(u,v)$ 에서  $t$ 번째 참조 프레임을 이용하여  $k$ 번째 프레임을 채울 때의 비용을 나타낸다.  $I_k(x,y)$ 와  $D_k(x,y)$ 는 각각 좌표  $(x,y)$ 에서  $k$ 번째 가상 컬러 영상과 가상 깊이 영상의 홀이 아닌 픽셀의 값을 의미한다.  $M_k(c, x, y, t)$ 와  $M_k(d, x, y, t)$ 는 좌표  $(x,y)$ 에서  $t$ 번째 컬러 후보 참조 프레임과 깊이 후보 참조 프레임의 픽셀 밝기 값을 나타낸다.

한 블록이 채워지면 채움 블록의 우선 순위를 다시 계산하고 홀이 다 채워질 때까지 위 과정을 반복하게 된다. 마지막 과정은 boundary artifacts removing이며 객체의 경계선에 발생하는 에러를 제거하기 위한 과정이다. 일반적으로 컬러 영상과 시차 영상은 정확하게 일치 하지 않는다. 따라서 진경이 배경으로 잘못 워핑되는 경우가 발생한다. 이러한 에러들은 블록 매칭 시 에러가 전파될 가능성을 높인다. 따라서 boundary noise에 해당되는 영역을 큰 홀이라고 취급하여 위에서 언급한 방법으로 홀을 채우게 된다.

### III. 제안하는 기법

#### 3.1 외삽 영상 생성 시 영상 참조 방법

VST 기법은 내삽 영상의 홀을 채우기 위해 그림 5와 같이 L, L1, L2, V, V1, R, R1, R2 총 8개의 참조 영상에서 적합한 블록을 탐색한다<sup>[8]</sup> 안하는 기법에서는 내삽(interpolated) 영상이 아닌 외삽(extrapolated) 영상에 홀 채움을 적용한다. 내삽 영상은 좌, 우 시점을 모두 이용하여 좌, 우 시점 사이에 있는 가상 시점



영상을 생성하는 반면에 외삽 영상은 좌, 우 영상 중 하나의 시점만을 이용하여 그 시점 바깥의 가상 시점을 생성한다. 따라서 외삽 영상에서 발생하는 홀의 크기는 두 시점을 이용하여 생성하는 내삽 영상보다 큰 특징이 있다.

외삽 영상의 홀 채움 시 내삽 영상과 달리 좌, 우 영상을 모두 참조할 필요가 없다. 예를 들어 좌 시점의 외삽 영상은 우 시점과는 관련이 없기 때문에 참조할 필요가 없다. 따라서 VST 기법을 외삽 영상에 적용할 경우 내삽 영상에 적용한 것처럼 8개의 참조 영상에서 블록을 탐색할 필요가 없다. 그림 6은 좌 시점의 외삽 영상 생성 시 기존의 VST 기법이 참조하는 영상의 비율을 보여준다. 참조 영상은 모두 8개이며 스테레오 영상의 좌 영상을 기준 시점으로 하여 외삽 시점의 영상을 생성한다. 참조 영상으로 좌 영상인 L, L1, L2를 사용한 비율이 약 62%이며 가상 시점의 영상 V, V1을 사용한 비율이 약 38%이다. 좌 영상으로 외삽 시점의 영상을 생성 할 경우, 우 영상은 사용하지 않는 것을 알 수 있다. 따라서 외삽 영상 홀 채움 시 기존의 VST 기법처럼 8개의 참조 영상을 활용하는 것이 아닌 기준 시점의 영상을 T, T1, T2(좌 영상 참조 시는 L, L1, L2, 우 영상 참조 시는 R, R1, R2)라고 하면 이들과 함께 가상 시점의 영상 V, V1만을 이용하는 것이 더 효율적이다.

VST 기법에서 블록 매칭은 상당히 많은 연산을 필요로 한다. 예를 들어 검색 범위가 75이고 윈도우의 크기가 15라고 가정하면 하나의 홀을 채우기 위하여  $15 \times 15 \times 75 \times 75$ , 약 130만 번의 연산이 필요하다. 따라서 연산량을 줄이기 위해서는 참조 영상의 개수를 줄일 필요가 있다. 그림 7은 VST 기법의 참조 영상 비율을 나타내는 그래프이다.

홀을 채울 경우에 T1은 55%, V는 30%, V1은 8%, T2는 5%, T는 2%를 참조 영상으로 사용한다. 결국 93%의 홀은 T1, V, V1을 참조하여 채울 수 있다. 또

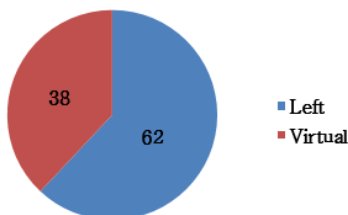


그림 6. 좌 시점의 외삽 영상 생성 시 블록 매칭에 사용된 참조 영상의 비율  
Fig. 6. The ratio of the reference image used for block matching in generating the left extrapolated image

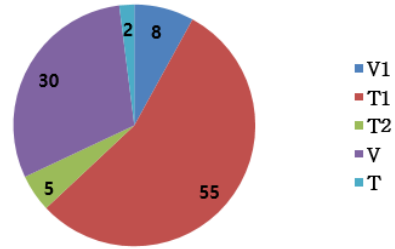


그림 7. VST 기법의 참조 영상 비율  
Fig. 7. Reference image ratio of VST method

한 T1은 공간 유사성, V는 시간 유사성, V1은 시점 유사성을 고려하므로 세 가지 기본 가정을 모두 유지할 수 있다. 따라서 T1, V, V1 영상만을 이용하여 블록 매칭을 진행하는 것이 효율적이라는 가정을 할 수 있다. 그림 8은 참조 영상의 개수에 따른 참조 영상의 활용 분포도를 나타낸다. 그림 8(a)는 5개의 참조 영상 T, T1, T2, V, V1을 이용하여 홀을 채웠을 때의 분포도이고 그림 8(b)는 T1, V, V1, 3개의 참조 영상을 이용한 경우의 분포도를 나타낸다. 파란색은 T1, 초록색은 V, 빨간색은 V1, 청록색은 T, 하얀색은 T2를 각각 의미한다.

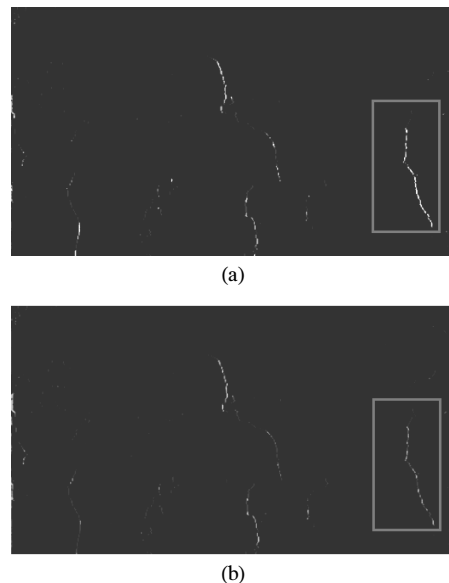


그림 8. 참조 영상 개수에 따른 참조 영상 분포도 (a) 참조 영상이 5개인 경우 (b) 참조 영상이 3개인 경우  
Fig. 8. Reference image distribution according to the number of reference images (a) Case with five reference images (b) Case with three reference images

### 3.2 배경영역을 이용한 변이 영상 홀 채움 기법 적용

기존의 VST 기법은 변이 영상에 식 (1)의 spatial weighted interpolation을 적용하여 홀을 채우며 공간 유사성으로 인하여 홀을 채운 영역의 주변에서 블러 현상이 발생한다<sup>8)</sup> 러한 블러는 블록 매칭 시 비용에 영향을 주게 되고 가상 시점 영상의 품질을 떨어트리는 원인이 된다.

일반적으로 홀은 전경과 배경의 경계선에서 나타나며 따라서 발생한 홀들은 배경으로 채워준다. 따라서 제안하는 기법에서는 spatial weighted interpolation 기법이 아닌 배경 영역을 이용하여 변이 영상의 홀을 채우게 된다. 그림 9는 spatial weighted interpolation 기법과 배경에 해당하는 변이 값만을 이용하여 홀을 채운 경우를 비교한 영상이다. 그림 9(a)는 양방향 선형 보간법을 이용하여 변이 영상의 시점을 이동시킨 영상이다. 그림 9(b)는 spatial weighted interpolation 기법<sup>7)</sup>을 이용하여 홀을 채운 영상이며 홀이 채워진 부분에서 블러 현상이 발생하는 것을 확인 할 수 있다. 그러나 그림 9(c)는 배경에 해당하는 변이 값들을 이용하여 홀을 채웠기 때문에 블러 현상이 발생하지 않는다.

그림 10(a)와 (b)는 Gyeback2 영상에 각각 spatial weighted interpolation 기법으로 생성한 외삽 영상과 제안하는 기법을 이용하여 생성한 외삽 영상을 각각 보여준다. Gyeback2 영상은 변이 영상이 비교적 정확하고 홀의 크기가 작은 편이기 때문에 두 기법간의 큰 차이가 존재하지 않는다. 그림 10(c)와 (d)는 Gyeback1 영상에 각각의 기법을 적용한 경우이다. 그림 10(c)와 (d)에서 초가집 지붕 부분을 비교해보면

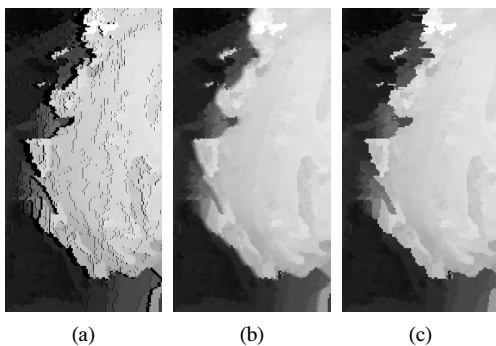


그림 9. 변이 영상의 홀 채움 영상 (a) 외삽한 변이 영상 (b) spatial weighted interpolation 기법 (c) 제안하는 변이 영상의 홀 채움 기법  
Fig. 9. Hole filled image of disparity image (a) Extrapolated disparity image (b) spatial weighted interpolation (c) proposed method

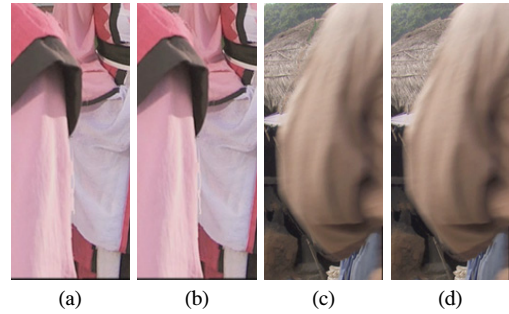


그림 10. 변이 영상 홀 채움 기법에 따른 외삽 시점 영상 (a) Spatial weighted interpolation (Gyeback2) (b) 배경을 이용한 홀 채움 기법 (Gyeback2) (c) Spatial weighted interpolation (Gyeback1) (d) 배경을 이용한 홀 채움 기법 (Gyeback1)

Fig. 10. Extrapolated view according to hole filling method of disparity image (a) Spatial weighted interpolation (Gyeback 2) (b) hole filling technique using background (Gyeback2) (c) Spatial weighted interpolation (Gyeback1) (d) Hall filling technique using background (Gyeback1)

상당히 큰 차이를 보여준다. 그림 10(c)의 경우 홀이 지붕으로 채워져야 하지만 배경에 존재하는 산으로 채워졌다. 그러나 그림 10(d)의 경우 올바르게 초가집 지붕으로 채워지는 것을 확인 할 수 있다.

Gyeback1은 변이 영상의 정확도가 매우 떨어지는 영상이다. 따라서 spatial weighted interpolation 기법<sup>7)</sup>을 사용하게 되면 이러한 부정확한 시차 값이 블록 매칭을 하는 과정에서 비용을 증가시키고 잘못된 블록으로 매칭되는 현상이 발생하게 된다. 따라서 제안하는 변이 영상 홀 채움 기법을 통해 배경 성분만을 이용해 홀을 채워줌으로써 블러의 영향을 줄일 수 있다.

### 3.3 N-step 블록 매칭 기법

앞에서 언급했듯이 VST 기법은 크게 네 단계의 과정을 거치게 된다. 이 중 가장 많은 연산량을 차지하는 것이 바로 블록 매칭 과정이다. 따라서 처리 속도를 향상시키기 위해서는 블록 매칭의 속도를 향상시키는 것이 매우 중요하다. 일반적으로 블록 매칭 기법은 exhaustive<sup>10)</sup>, n-step<sup>10)</sup> 방법 등이 존재한다. 기존의 VST 기법은 exhaustive 검색 방법을 사용한다<sup>8)</sup> 방법은 검색 범위의 모든 픽셀들을 탐색하는 방법으로 연산량이 많다. 예를 들어 검색 범위가 15라면 15×15, 총 225번의 연산을 수행해야 한다. 반면에 n-step 방법은 말 그대로 n 단계에 걸쳐 블록 매칭하는 방법으로 그림 11에 그 과정을 보여준다. 스텝 간격을 4라고 가정하면 1단계에서는 스텝 간격의 거리에 있는 블록들 간 매칭을 진행하여 최소가 되는 블록

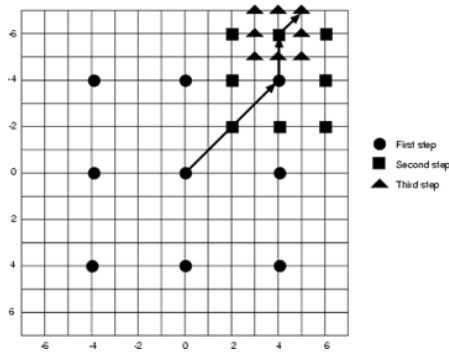


그림 11. N-step 과정  
Fig. 11. N-step process

을 찾는다. 2단계에서는 1단계에서 매칭한 블록을 기준으로 스텝 간격을 절반으로 줄여 매칭을 다시 진행한다. 이러한 과정을 스텝 간격이 1이 될 때까지 반복한다. 최종적으로 비용이 최소가 되는 블록을 최종 매칭 블록으로 결정한다. N-step 기법을 적용하면 연산량을 exhaustive 기법의 1/10 수준으로 줄일 수 있다.

#### IV. 실험 결과

본 논문에서는 제안하는 기법의 성능을 평가하기 위해 Avengers와 Gyeback1, Gyeback2 그리고 Snake 영상을 이용하여 실험을 진행하였다. 모든 영상은 Full HD 크기이며 프레임 수는 각각 50, 150, 100, 10 프레임이다. 변이 영상은 SGM을 이용하여 획득하였다. 그림 12는 실험 영상 및 변이 영상을 보여준다.

먼저 참조 영상의 개수에 따른 영상의 품질을 측정하였다. 블록을 탐색할 때 참조하는 프레임으로 T, T1, T2, V, V1을 참조할 때와 T1, V, V1을 참조할 때의 결과 영상을 그림 13에 비교하였다. 그림 13의 (a), (c), (e), (g)는 참조하는 영상을 색상을 달리하여 표현한 것이다. 그림 13의 (a), (e)는 T, T1, T2, V, V1을 참조한 경우이고 그림 13의 (c), (g)는 T1, V, V1을 참조한 경우이다. 파란색은 T1, 초록색은 V, 빨간색은 V1, 하얀색은 T2, 청록색은 T 영상을 참조하였음을 각각 의미한다. 그림 13의 (a)와 (c)는 T와 T2를 참조하는 것을 알 수 있다. 그림 13의 (b)와 (d) 그리고 (f)와 (g)는 홀을 채운 영상을 보여주며 큰 차이는 없는 것으로 확인된다. 표 1은 참조 영상 개수 및 블록 탐색 방법에 따른 처리 시간을 비교한 것이다. Exhaustive Search를 사용하는 경우 참조 영상을 5개 모두 사용하는 경우와 3개 사용하는 경우를 비교한 결과 3개를 사용하였을 때 처리 시간이 3/5 수준으로



그림 12. 실험에 사용한 영상 (a) Avengers 컬러 영상 (b) Avengers 변이 영상 (c) Gyeback1 컬러 영상 (d) Gyeback1 변이 영상 (e) Gyeback2 컬러 영상 (f) Gyeback2 변이 영상 (g) Snake 컬러 영상 (h) Snake 변이 영상  
Fig. 12. Images used in experiment (a) Avengers color image (b) Avengers disparity image (c) Gyeback1 color image (d) Gyeback1 disparity image (e) Gyeback2 color image (f) Gyeback2 disparity image (g) Snake color image (h) Snake disparity image

줄어들었다. 참조 영상 개수가 준 비율만큼 시간이 줄어든 것을 확인할 수 있다. N-step Search를 사용하였을 때 실험 영상 별로 시간이 대폭 줄어들었다. 참조 영상을 3개 사용한 경우를 비교해보면 N-step Search를 사용하는 경우가 Exhaustive Search를 사용하는 경우보다 Avengers는 10.3배, Gyeback1은 6.1배, Gyeback2는 10.6배, Snake는 8.3배 줄었으며 평균적으로 8.8배 빨라진 것을 확인할 수 있다.

객관적 화질 평가를 위해 Dancer 실험 영상을 사용하였다. 위에서 사용한 실험 영상들은 가상 시점에 대한 원본 영상이 없기 때문에 PSNR을 측정할 수 없다. 표 2는 참조 영상 개수 및 블록 탐색 방법에 따른 PSNR을 보여준다. 참조 영상 5개와 Exhaustive Search를 사용하는 경우 PSNR 수치는 제일 높다. 참조하는 영상의 개수가 많고 검색 범위가 커질수록 PSNR 수치가 증가하는 것을 알 수 있다. 하지만 PSNR의 증가폭에 비해 시간이 매우 많이 늘어나기 때문에 효율성이 떨어진다. 반면에 참조 영상 3개와



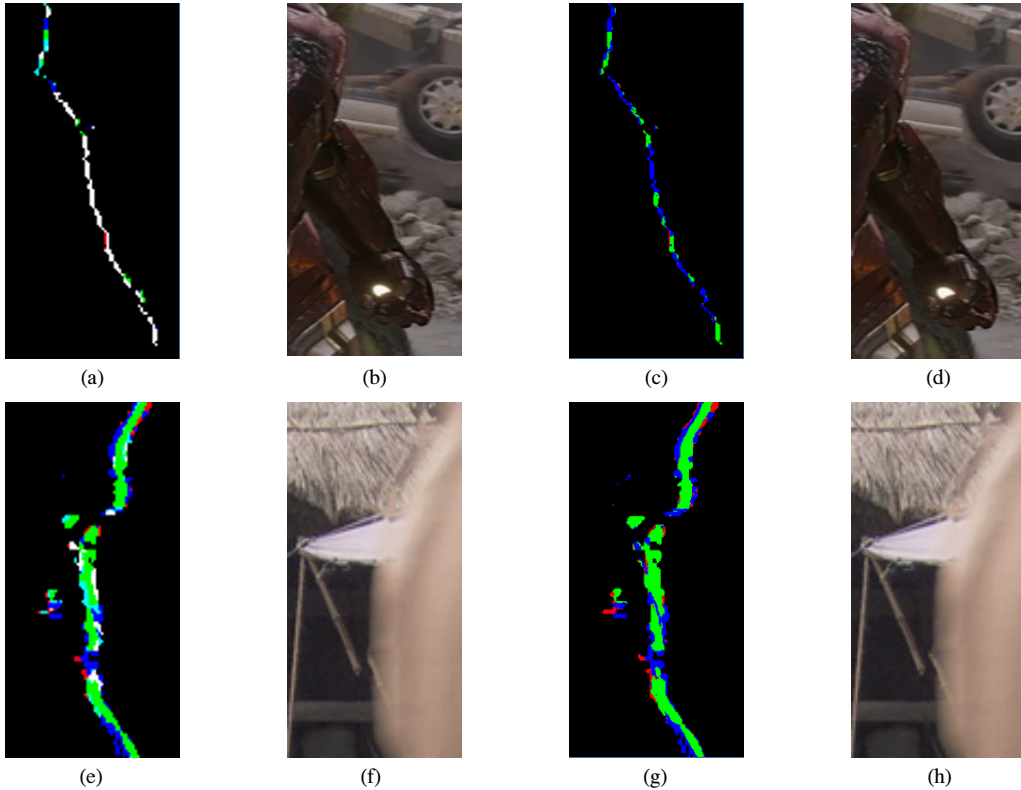


그림 13. 5개의 영상을 참조했을 때와 3개의 영상을 참조했을 때의 영상 비교 (a) Avengers 영상을 5개의 영상을 이용하여 홀 채움 시 블록을 가져오는 분포도 (b) 5개 영상을 이용한 Avengers 홀 채움 영상 (c) Avengers 영상을 3개의 영상을 이용하여 참조하였을 때 블록을 가져오는 분포도 (d) 3개 영상을 이용한 Avengers 홀 채움 영상 (e) Gyeback1 영상을 5개의 영상을 이용하여 참조하였을 때 블록을 가져오는 분포도 (f) 5개 영상을 이용한 Gyeback1 홀 채움 영상 (g) Gyeback1 영상을 3개의 영상을 이용하여 참조하였을 때 블록을 가져오는 분포도 (h) 3개 영상을 이용한 Gyeback1 홀 채움 영상

Fig. 13. Comparison of images when referring to 5 images and 3 images (a) Distribution that takes blocks when filling holes with 5 images (Avengers) (b) Hole-filling images using 5 images (Avengers) (c) Distribution that takes blocks when filling holes with 3 images (Avengers) (d) Hole-filling images using 3 images (Avengers) (e) Distribution that takes blocks when filling holes with 5 images (Gyeback1) (f) Hole-filling images using 5 images (Gyeback1) (g) Distribution that takes blocks when filling holes with 3 images (Gyeback1) (h) Hole-filling images using 3 images (Gyeback1)

표 1. 참조 영상 개수 및 블록 매칭 기법에 따른 처리 시간 비교 (단위: 분)

Table 1. Comparison of processing time according to the number of reference images and block matching method (unit: minute)

Method		Sequence			
Number of reference images	Block matching method	Avengers	Gyeback1	Gyeback2	Snake
5	Exhaustive Search	40.3	49.2	67.5	75.5
3	Exhaustive Search	24.7	29.3	40.2	45.6
5	N-step Search	3.8	5.6	7.2	8.8
3	N-step Search	2.4	4.8	3.8	5.5

표 2. 참조 영상 개수 및 블록 매칭 기법에 따른 PSNR 비교 (단위: dB)

Table 2. Comparison of PSNR according to the number of reference images and block matching method (unit: dB)

Method		PSNR
Number of reference images	Block matching method	
5	Exhaustive Search	34.18
3	Exhaustive Search	34.02
5	N-step Search	34.13
3	N-step Search	33.94

N-Step Search를 사용하는 경우에는 참조 영상 5개와 Exhaustive Search를 사용하는 경우와 비교하여 PSNR 수치가 0.24dB 감소하지만 속도는 15배 이상 빨라지기 때문에 효율성이 매우 좋다고 할 수 있다.

그림 14는 Dancer 영상에 대하여 제안하는 기법과 HHF<sup>[4]</sup> 그리고 Inpainting 기법<sup>[7]</sup>의 PSNR을 비교한 결과를 보여준다. 그림에서 가로 축은 프레임 번호를 의미하며 세로 축은 PSNR 값을 나타낸다. Dancer 영상은 CG영상으로 변이 영상이 매우 정확하다. 따라서 블록 매칭 시에 문제가 되는 경계선 에러가 존재하지 않기 때문에 제안하는 기법이 다른 기법보다 PSNR 값이 높게 측정되었다.

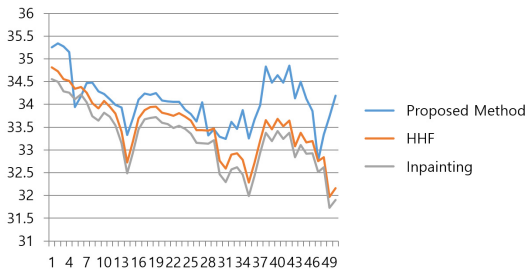


그림 14. Dancer 영상 PSNR 비교  
Fig. 14. PSNR Comparison of Dancer video

### V. 결론

본 논문에서는 기존 VST 기법을 근간으로 문제점을 개선한 새로운 홀 채움 기법을 제안하였다. 제안하는 기법에서는 기존 VST 기법의 연산량을 대폭 줄이면서 영상의 품질은 유지하는 방향으로 개선하였다. 먼저 기존 VST 기법이 홀을 채울 때 참조하는 블록을 통계적으로 분석하였다. 분석 결과 좌 영상으로 외삽 영상 생성 시 우 영상에서는 블록을 참조하지 않는 사실을 관측하였고, 따라서 8개의 영상 L, L1, L2, V, V1, R, R1, R2 중에서 L, L1, L2, V, V1에서 블록을 참조하는 비율을 확인한 결과 전체 홀 중 7%만이 L과 L2를 참조하였다. 따라서 블록 참조 후보는 L1, V, V1으로 결정하였다.

기존 VST 기법의 연산 속도가 느린 원인 중 하나는 풀 서치 방식의 블록 매칭 기법 때문인데, n-step 블록 매칭 기법을 적용하여 속도를 개선하였다. 풀 서치 방식은 검색 범위가 15라면 총 225번의 연산을 수행하는 반면에 n-step 방식은 총 25번의 연산만을 수행하기 때문에 훨씬 빠른 속도가 가능하다. 시간 측정 실험 결과 기존 VST 기법은 평균적으로 34.9분이 소

요되는 반면에 제안하는 기법은 4.1분이 소요되어 약 8.5배의 상승을 확인할 수 있었다. PSNR 측정 비교 실험에서도 제안하는 기법이 HHF<sup>[4]</sup>와 Inpainting<sup>[7]</sup> 등의 다른 기법과 비교하여 좋은 결과를 나타내었으며 주관적인 영상 품질 또한 개선되는 것을 확인하였다.

### References

- [1] W. Mark, L. McMillan, and G. Bishop, "Post-rendering 3D warping," in *Proc. 1997 Symp. Interactive 3D Graphics*, pp. 7-16, Rhode Island, USA, Apr. 1997.
- [2] C. Fehn, "Depth-image-based rendering (DIBR), compression, and transmission for a new approach on 3D-TV," *Int. Soc. for Optics and Photonics, Electron. Imaging 2004*, pp. 93-104, San Jose, USA, May 2004.
- [3] L. Zhang, W. J. Tam, and D. Wang, "Stereoscopic image generation based on depth images," *ICIP'04*, pp. 2993-2996, Singapore, Singapore, Oct. 2004.
- [4] M. Solh and G. AlRegib, "Hierarchical hole-filling for depth-based view synthesis in FTV and 3D video," *IEEE J. Sel. Topics in Sign. Process.*, vol. 6, no. 5, pp. 495-504, Jun. 2012.
- [5] S. Choi, B. Ham, and K. Sohn, "Space-time hole filling with random walks in view extrapolation for 3D video," *IEEE Trans. Image Process.*, vol. 22, no. 6, pp. 2429-2441, Mar. 2013.
- [6] F. Wang and D. P. Landau, "Efficient, multiple-range random walk algorithm to calculate the density of states," *Physical Rev. Lett.*, vol. 86, no. 10, pp. 2050-2053, Mar. 2001.
- [7] A. Criminisi, P. Perez, and K. Toyama, "Region filling and object removal by exemplar-based image inpainting," *IEEE Trans. Image Process.*, vol. 13, no. 9, pp. 1200-1212, Sept. 2004.
- [8] L. Zhu, Y. Zhang, M. Yu, G. Jiang, and S. Kwong, "View-spatial-temporal post-refinement for view synthesis in 3D video systems," *Sign.*

*Process.: Image Commun.*, vol. 28, no. 10, pp. 1342-1357, Nov. 2013.

- [9] M. S. Ko and J. S. Yoo, "Boundary noise removal and hole filling algorithm for virtual viewpoint image generation," *J. KICS*, vol. 37, no. 8A, pp. 679-688, Jul. 2012.
- [10] A. Barjatya, "Block matching algorithms for motion estimation," *IEEE Trans. Evolution Computation*, vol. 8, no. 3, pp. 225-239, 2004.
- [11] J. H. Yun, B. K. Bae, S. H. Park, H. Song, D. W. Kim, and J. S. Yoo "Fast disparity estimation method considering temporal and spatial redundancy based on a dynamic programming," *J. KICS*, vol. 33, no. 10C, pp. 787-797, Oct. 2008.
- [12] G. C. Lee and J. S. Yoo, "Real-time virtual-view image synthesis algorithm using Kinect camera," *J. KICS*, vol. 38, no. 5, pp. 409-419, May 2013.

**이 규 철 (Gyu-Cheol Lee)**



2013년 2월 : 광운대학교 전자공학과 학사  
2015년 2월 : 광운대학교 전자공학과 석사  
2015년 3월~현재 : 광운대학교 전자공학과 박사과정  
<관심분야> 영상처리, 영상압축, 영상인식

**유 지 상 (Jisang Yoo)**



1985년 2월 : 서울대학교 전자공학과 학사  
1987년 2월 : 서울대학교 전자공학과 석사  
1993년 5월 : Purdue Univ. EE, ph.D  
1997년 9월~현재 : 광운대학교 전자공학과 교수

<관심분야> 영상처리, 영상압축, 영상인식