

반전된 Depth 영상을 이용한 실시간 Gaussian Hole-Filling Algorithm

안양근*, 홍지만*

Real-time Gaussian Hole-Filling Algorithm using Reverse-Depth Image

Yangkeun Ahn *, Jiman Hong *

요약

기존의 Stereoscropy영상의 생성 방법은 2개의 촬영용 렌즈를 일정한 간격으로 띄워놓고 같은 피사체를 촬영하여 좌우 시점의 영상을 생성하는 방법을 이용하게 된다. 하지만 Stereoscropy 카메라를 이용한 3차원 TV 수신에 경우에 좌시점과 우시점에 대한 2개의 영상을 모두 동시에 전송해야 하기 때문에 대역폭의 부담이 크다. 이에 보다 효율적인 여러 방법들에 대한 논의가 이루어지고 있다. 그중 DIBR(Depth Image Based Rendering)은 한장의 영상과 이에 대한 Depth정보를 이용하여 좌시점과 우시점의 영상을 생성하기 때문에 전송 대역폭을 줄일 수 있으며, 이러한 점으로 인하여 기존의 Static Scene에서 DIBR Image 생성에 대한 다양한 Algorithm이 연구되어 왔다. 본 논문에서는 반전된 Depth 영상을 이용하여 자연스럽게 Hole을 채움과 동시에 주변 배경의 왜곡 또한 최소화하는 Gaussian Hole-Filling 방법을 제안하려 한다. 또한 각 Algorithm들의 성능을 비교, 계산하여 각 Algorithm들의 효율성을 분석하였다.

▶ Keyword : Realtime DIBR, DIBR, 3DTV, Depth image, Hole-Filling

Abstract

Existing method of creating Stereoscropy image, creates viewpoint image from the left and right by shooting one object with 2 lens in certain distance. However, in case of 3-D TV using Stereoscropy camera, the necessity to transmit 2 viewpoint images from the left and right simultaneously, increases the amount of bandwidth. Various and more effective alternatives are

• 제1저자 : 안양근 • 교신저자 : 홍지만
• 투고일 : 2012. 04. 10, 심사일 : 2012. 05. 03, 게재확정일 : 2012. 05. 29.
* 숭실대학교 컴퓨터학과(School of Computing, Soongsil University)

under discussion. Among the alternatives, DIBR(Depth Image Based Rendering) creates viewpoint images from the left and right using one image and its Depth information, thus decreasing the amount of transmitted bandwidth. For this reason, there have been various studies on Algorithm to create DIBR Image in existing Static Scene. In this paper, I would like to suggest Gaussian Hole-filling solution, which utilizes reverse-depth image to fill the hole naturally, while minimizing distortion of background. In addition, we have analyzed the effectiveness of each Algorithm by comparing and calculating its functions.

▶ Keyword : Realtime DIBR, DIBR, 3DTV, Depth image, Hole-filling

1. 서 론

최근 기존의 TV에서 한 차원 진보한 3D TV가 등장하였다. 2개의 영상을 합성하여 입체감을 나타낼 수 있는 3D TV는 현재 가정으로까지 점점 활성화되고 있는 추세이다. 또한 이 과정에서 많은 사용자들이 3D 인터랙티브 기술들을 경험하며, 기존의 TV에서 느끼기 힘들었던 입체감과 입장감을 느끼려는 사용자 또한 늘어남에 따라서 3D TV의 폭 또한 확대될 것으로 전망되고 있다 [1]. 따라서 보다 효율적인 여러 방법들에 대한 필요성이 인식되고 있으며, 관련 연구들은 현재 학계보다 다양한 기업들에 의해 논의가 이루어지고 있다.

DIBR(Depth Image-Based Rendering)은 한장의 컬러 영상과 그에 대응하는 깊이 영상을 이용하여 가상의 좌우 영상을 생성하여 시청자에게 깊이감을 제공해주는 기법으로, 이를 이용함으로써 제한된 입력영상으로부터 다양한 영상을 생성하여 3D TV에서 입체감을 높일 수 있다. 때문에 이를 이용하여 시청자가 원하는 입의 시점의 영상을 생성할 수 있다는 장점이 있다 [2]. 때문에 현재 DIBR의 수행 과정에서 깊이 영상과 컬러 영상으로부터 다시점(좌우) 영상을 생성하기 위해 여러가지 알고리즘들이 제안되고 있다 [3-5].



그림 1. DIBR Algorithm에서 발생하는 Hole 영역
Fig. 1. Hole area in DIBR Algorithm

하지만 이렇게 DIBR을 이용하여 가상 시점의 새로운 이미지를 만들 경우, 색상 정보와 깊이 정보가 없는 새로운 영역이 발생, 위 새로운 영역을 Hole이라 하고, Hole이 생긴 영역은 후에 Software적으로 보간 해줘야 한다. 때문에 이러한 보간 과정은 다양하게 연구되고 있다 [6-8]. 이러한 처리 과정을 Hole-filling이라 부르고 본 논문에서는 이러한 Hole을 채우는 여러 DIBR Algorithm들을 실시간 영상에서 비교하려 한다.

우선 일반적인 방법으로는 Warping 후 생기는 Hole을 인접한 pixel을 참조하여 채우는 인접 픽셀 복사 방식이 있다. 이는 수평방향으로 pixel을 탐색해서 Hole에서 배경과 가장 가까운 픽셀을 하나 정한 다음 그 픽셀을 수평 방향으로 Hole을 모두 채우는 방식이다. Static scene에서 이 방법을 사용할 경우, 특별한 연산이 적용되지 않기 때문에 가장 빠른 속도로 Hole을 채울 수 있는 방법이지만 Hole의 크기가 커질 경우 Hole을 채운 영역이 어색하게 표현되기가 쉽고, 이는 객체가 이동한 자리가 육안으로 쉽게 노출된다는 단점이 있다.

이와 비슷한 방법으로 주변 영역의 pixel을 늘어뜨려 Hole을 채우는 인접 영역 확장 방식이 있다. 알고리즘의 흐름은 인접 픽셀 복사 방식과 유사하게 진행되지만, 인접 픽셀 복사 방식이 단순히 하나의 픽셀을 가지고 Hole을 채운다면 이 방식은 Hole의 크기만큼의 영역을 잡아 그 영역을 2배로 늘려 Hole과 함께 기존의 영역을 채우는 방식이다. 인접 영역 확장 방식은 인접 픽셀 복사 방식보다는 보다 자연스러운 Hole-filling이 가능하지만, 위 방법 또한 Hole을 채운 영역이 어색해진다는 단점이 있다.

또한 Zhang은 Static Scene에서 DIBR 영상의 잡음이나 왜곡을 줄이는 대안으로서 Depth map을 Gaussian-smoothing 하는 방법을 제안하였다 [9]. Zhang은 Gaussian-smoothing은 대칭(Symmetric)과 비대칭(Asymmetric)으로 나누어 적용하였고, 이를 이용하여 Image warping 후 발생하는 Hole의 기하학적 잡음을 줄일 수 있었다. 위 방법은 객체가 이동한 Hole 영역을 자

연스럽게 채울 수 있는 장점이 있는 반면에, 주변 배경이 왜곡되는 단점을 가지고 있다. DIBR은 Real-time TV에 적용하기 위해서는 객체의 품질과 주변 배경 영상의 품질을 동시에 유지시켜주는 것이 중요하다.

본 논문에서는 반전된 Depth 영상을 이용하여 보다 자연스러운 결과물을 얻을 수 있는 Reverse-Gaussian 방법을 제시하여 Warping된 객체의 Hole-filling과 더불어 주변 배경의 왜곡을 최소화 하였다. 또한 이를 앞서 제시한 여러 알고리즘들과 함께 실시간 영상에 적용하여 결과를 분석하였고 품질을 비교하였다. 보다 정확한 비교를 위해 우선 기존의 각 알고리즘의 적용 과정을 기술하였고 그 후 본 논문의 Algorithm인 Reverse Gaussian 방법을 기술하였다. 그 후 각 알고리즘들의 품질과 연산 속도에 관한 결과를 분석하였다.

II. 실시간 DIBR System 구성

1. 실시간 3D 영상 생성 시스템 플랫폼 구성도

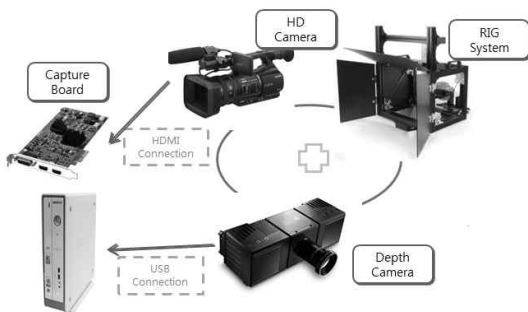


그림 2. HW 시스템 플랫폼 구성도
Fig. 2. HW system platform layout

DIBR image를 생성하기 위해서는 원본 color영상과 이에 해당하는 depth영상이 필요하다. 또한 이를 위한 H/W 설계는 알고리즘과 함께 매우 중요한 부분이고 이에 대한 연구는 DIBR 전용 chip 설계로도 이뤄지고 있다 [10]. 그림 1은 본 논문에서 실험한 H/W 시스템의 구성을 보여주고 있다.

PC에 내장 되어있는 capture board는 ADC(Analog to Digital Converter) 회로가 장착 되어있어 매 프레임마다 비디오카메라의 아날로그 데이터인 비디오 영상을 디지털화하여 개인용 컴퓨터(PC)로 전송하는 역할을 한다. 이렇게 보내진 image를 컴퓨터에 저장하고, 이를 이용하여 DIBR 이미지를 생성할 수 있게 된다.

또한 color영상을 추출 할 수 있는 camera 1대와 depth 영상을 얻기 위한 depth 카메라 1대가 필요한데, 이를 위하여 sony사의 HVR-Z5N HD 카메라 1대, Depth를 추출하기 위한 PMD CamCube 2.0 Depth 카메라를 사용하였다. 위 두 카메라는 서로 독립된 2개의 카메라이지만, 같은 하나의 영상에 대한 정보를 가지고 있어야 하기 때문에, 보이는 영상이나 거리, 영상의 초점거리 등 대응이 이루어져야 한다. 따라서 환경을 구성하는 과정에서 HD 카메라와 Depth 카메라의 광축 일치를 위한 특수 거울(Hot Mirror)을 장착한 RIG system을 설계하였다. 따라서 두 카메라를 고정하여 두 카메라가 정확하게 같은 초점 유지가 가능하였다.



(a) (b)



(c)



(d)

그림 3. HW 실험환경 (a) Sony HD 카메라, (b) PMD Depth 카메라, (c) Rig 시스템 (d) 시스템 종합 환경

Fig. 3. HW test environment (a)Sony HD camera, (b)PMD Depth camera, (c) Rig system (d) Overall System environment

2. 실시간 3D 영상 생성과정

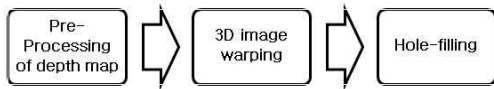


그림 4. DIBR System 구현과정
Fig. 4. DIBR system realization process

그림 4는 일반적으로 실시간 DIBR 시스템을 구현하는데 필요한 세 단계의 과정을 보여주고 있다. 첫 번째 단계인 Pre-processing of depth map 단계는 Image Warping 전 여러 데이터를 가공하는 단계이다. 위 과정에서 보다 정확한 Warping과 Hole-filling 과정을 위한 가장 대표적인 전처리로는 Zero Parallax Setting(이하 ZPS)이 있다. Positive Parallax 상태에서는 스테레오 평면에 맺힌 두 점이 각각 좌, 우 눈으로 들어와 뇌에서 하나로 합쳐져서 스테레오 평면보다 들어가 보이게 된다. 반면에 Negative Parallax는 스테레오 평면에 맺힌 두 점이 각각 반대쪽 좌, 우 눈으로 들어오고, 뇌에서 하나로 합쳐져서 스테레오 평면보다 앞으로 튀어나와 보이게 된다. 마지막으로 Zero Parallax는 스테레오 평면에 정확한 초점을 맺으며 양안 시차가 없는 것을 말하며 입체감이 생기지 않는다. 8bit depth map에서는 표현할 수 있는 최대치인 256을 이용하여 시청자의 시각에서 가까운 거리를 뜻하는 Z_{near} 을 255로, 먼 거리를 의미하는 Z_{far} 를 0으로 기준을 잡는다. 이를 이용한 Z_c 는 아래의 식 (1)로 결정하게 된다.

$$Z_c = (Z_{near} - Z_{far}) / 2 \tag{1}$$

ZPS를 시청자가 결정하여 이를 전처리로 이용하면 Depth Map의 기준을 설정할 수 있게 되는데, 이를 이용하면 시청자가 원하는 임의 시점의 영상을 생성할 수 있다.

다음으로 영상을 warping하는 과정을 거치게 되는데 위 과정은 획득한 Depth data에서 배경에 해당하는 Depth 영역과 객체에 해당하는 Depth 영역을 분리시키기 위해 필요한 과정이다. 이는 입체감을 주고자 하는 객체를 배경에서 떼어내어 이동하는 과정을 말한다. 두 눈의 viewpoint가 다르기 때문에 Warping시 Left-eye, Right-eye에 해당하는 영상을 따로 생성해야 하는데, 이 과정을 거치게 되면 완성된 영상에서 객체가 입체감 있게 보이는 것을 확인할 수 있다. 또한 객체를 이동시키는 정도에 따라서 얼마나 배경과 멀리 떨어져 있게 보이는지를 조절이 가능하다.

영상이 이처럼 이동하게 되면, 기존에 존재하지 않았던 빈 공간이 생기는데 이 공간을 Hole이라 하며, 보간법을 통해 Hole을 채워줘야 한다. 이 과정을 Hole-filling이라 한다. 이와 같이 DIBR 영상 생성과정은 기본적으로 위의 3단계를 거쳐야 생성될 수 있다.

III. 기존 DIBR Algorithm

1. 인접 픽셀 복사 방식과 인접 영역 확장 방식

3D-Image Warping 단계는 가상으로 이동하고자 하는 객체를 좌 또는 우로 이동시키는 단계이다. 본 논문에서 검증한 4가지 방법은 크게 Gaussian-smoothing을 사용하는 방법과 사용하지 않는 방법으로 나눌 수 있다. 그중 처음 2가지 방법은 Gaussian-smoothing을 사용하지 않고 주변의 pixel의 값을 이용하여 Hole을 채우는 방법이다. 인접 픽셀 복사 방식은 먼저 객체의 해당되는 pixel을 고정한 후, 이동하고자 하는 크기만큼 이동하고자 하는 방향으로 픽셀을 복사한다. 때문에 연산 과정이 매우 단순하여 처리 속도가 빠르다. 하지만 이는 픽셀의 강제적 이동이기 때문에, Depth 영상의 품질이 좋지 않을 경우 깨끗한 Warping 영상을 얻을 수 없게 된다.

인접 픽셀 복사 방식이 단순히 하나의 픽셀을 가지고 Hole을 채운다면 인접 영역 확장 방식은 Hole의 크기만큼의 영역을 잡아 그 영역을 2배로 늘려 Hole과 함께 기존의 영역을 채우는 방식이다.

$$Hole(X, Y) = Color((X' - X)/2 + X, Y) \tag{2}$$

식 2을 예를 들어 설명하면 다음과 같다. 그림 5에서 Hole의 X좌표가 H1(15,y), H2(16,y)이며 인접 컬러 영역은 C1(13,y), C2(14,y)일 경우 오른쪽으로 홀을 채울 때 결과 이미지의 R(13,y), R(14,y), R(15,y), R(16,y)라 하자. R(13,y)의 데이터는 C(13,y)의 x좌표 13에서 R(13,y)의 x좌표인 13을 빼서 2로 나눈 결과 값에 R(13,y)의 x좌표 13을 더하면 13이 나오게 되어 H(13,y)는 C(13,y)의 데이터가 들어가게 된다. 마찬가지로 R(14,y)의 경우에도 이 공식대로 대입하면 C(13,y)의 데이터가 들어가게 되며 R(15,y)에는 C(14,y)가, R(16,y)에는 C(14,y)의 값이 들어가게 되어 Hole이 채워지게 된다.

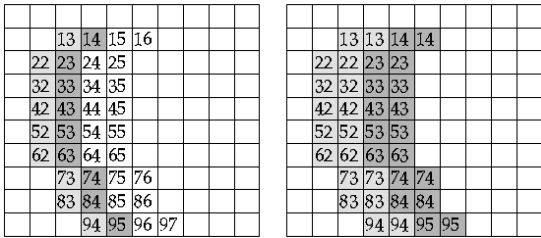


그림 5. 인접 영역 확장 방식 Algorithm
Fig. 5. Adjacent area expansion Algorithm

그림 5에서 숫자로만 표시되어 있는 부분은 실제 Hole 영역에 해당하며, 진한 회색과 연한 회색 영역은 이 Hole의 크기만큼 영역이 잠긴 Hole의 인접 영역을 나타낸다. 여기서 인접 영역 확장 방식을 이용하게 되면, Warping 후 채워지는 Hole에서 회색영역이 2배로 확장되어 오른쪽과 같이 Hole의 영역의 2배 크기정도가 채워지게 된다. 이는 일반적으로 인접 픽셀 복사 방식에 비해 Hole에 대한 자연스러운 품질을 기대할 수 있다. 하지만 위 방식 또한 픽셀의 강제적 이동과 보간법을 사용하기 때문에, 해당 픽셀들을 확장하는 과정에서 부자연스러운 경계면이 나타날 수 있다. 아래의 그림 6에서는 두 방식의 차이점에 대해 표현하고 있다.

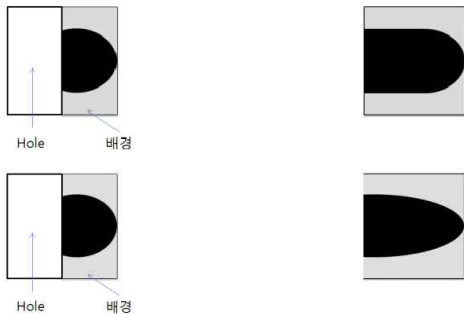


그림 6. 인접 픽셀 복사 방식(위) 인접 영역 확장 방식(아래) 의 비교
Fig. 6. Comparison Copying adjacent pixel(top) with Adjacent area expansion(bottom)

2. Gaussian Filter를 활용한 DIBR Algorithm

실시간 Gaussian DIBR Algorithm을 위해서는 먼저 기존의 Depth map을 Gaussian-smoothing하는 과정이 필요하다 [11]. 이 과정은 후에 Image warping시 발생하는 기하학적 잡음을 줄일 수 있고, 새로운 합성 영상의 화질을 향상시킬 수 있다. 본 논문에서는 식 (3)을 이용하여 Gaussian filter $g(x, \sigma)$ 를 적용한 depth image를 생성할 수 있다.

$$g(x, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left\{-\frac{x^2}{\sigma^2}\right\}, \quad \left(\text{for } -\frac{w}{2} \leq \frac{w}{2}\right) \quad (3)$$

w 는 window size를, σ 는 표준편차를 가리키며 σ 에 따라 depth 영상의 smoothing 정도를 결정하게 된다. 최종적으로 Smoothing depth image $S(x, y)$ 를 구하기 위해 식 (4)을 사용한다.

$$S(x, y) = \frac{\sum_{v=-w/2}^{w/2} \left\{ \sum_{\mu=-w/2}^{w/2} (s(x-\mu, y-v)g(\mu, \sigma_\mu))g(v, \sigma_v) \right\}}{\sum_{v=-w/2}^{w/2} \left\{ \sum_{\mu=-w/2}^{w/2} g(\mu, \sigma_\mu)g(v, \sigma_v) \right\}} \quad (4)$$



그림 7. Gaussian Filter를 적용한 Smooth the depth image
Fig. 7. Smooth the depth image with Gaussian Filter

다음으로 영상에 Gaussian Filter를 적용 시킨 후 Warping을 통해 객체를 이동시켜야 한다. 이 경우에는 세 번째 단계인 Hole-Filling 과정도 동시에 같이 진행된다. 이 방식의 경우에는 Gaussian Filter를 사용하기 때문에 알고리즘 연산에 많은 처리 비용이 들어가게 된다. 하지만 처리 속도는 떨어지지만 Depth 영상의 품질이 다소 떨어지더라도 좋은 품질의 Warping 알고리즘을 얻을 수 있다는 장점이 있다. 본 연구에서는 식 (5)와 같은 Gaussian Warping 공식을 사용한다. x_l 과 x_r 은 좌영상과 우영상 각각을 가리키며 t_x 는 두 가상 카메라 사이의 거리이다. x_c 와 Z 는 Depth 영상과 Color 카메라 영상을 통해 결정된다. 이렇게 계산된 가상의 좌시점와 우시점 영상의 차이($x_l - x_r$)를 통하여 대상이 ZPS로부터 얼마나 떨어져 있는지 확인할 수 있다. Z 가 줄어 들수록 차이는 작아지게 된다 [12].

$$x_l = x_c + \frac{t_x}{2} \frac{f}{Z}, \quad x_r = x_c - \frac{t_x}{2} \frac{f}{Z} \quad (5)$$

Hole-Filling 과정에서 Gaussian-Smoothing을 비대칭으로 적용하게 되면 보다 더 효과적으로 왜곡을 억제할 수 있다. 이는 인간의 뇌가 수직선으로 입체감을 인지한다는 점을 이용하여 수직선을 비대칭하게 Smoothing하는 방법이다 [13].



그림 8. 대칭 Depth image와 비대칭 Depth Image의 비교
Fig. 8. Comparison Symmetric Depth Image with Asymmetric Depth Image

IV. Reverse-Gaussian DIBR Algorithm

1. Amplitude Data를 이용한 잡음제거

일반적으로 첫 번째 단계인 Pre-processing of depth map 단계는 Color 카메라와 Depth 카메라로부터 얻은 데이터를 가공하는 단계이다. 위 과정에서 Color 카메라와 Depth 카메라의 해상도가 다르며, Depth 영상의 경우 그 데이터에 많은 잡음이 있어 Static Scene과 달리 실시간 DIBR Algorithm에서는 이를 바로 사용하기에는 문제가 있다. 또한 본 논문에서 제시하는 Reverse-Gaussian Algorithm의 경우에는 보다 정확한 Warping과 Hole-filling 과정을 위해서 잡음에는 둔감하고 실제 객체에 서만 반응하는 Depth Image가 필요하다. 따라서 본 실험에서는 Depth Camera와 사물들에 거리에 따른 Amplitude Data를 Depth Camera의 적외선 Data와 합성하였다.



그림 9. Amplitude Depth 데이터를 이용한 객체 추출
Fig. 9. Object extracted with Amplitude Depth data

먼저 Depth 카메라로부터 Data를 수신한다. 하지만 이렇게 얻은 데이터는 빛에 의한 잡음과 적외선 빛의 반사로 인한 잡음이 포함되어있다. 반면에 Amplitude Data는 빛에 의한 잡음은 적지만 객체의 명암을 살릴 수 없다. 따라서 먼저 원하는 범위의 데이터만 뽑아낸 후, Amplitude 데이터와의 합성을 통하여 Warping하고자 하는 객체 이외의 빛과 반사에 의해 생긴 나머지 잡음들을 제거한다. 마지막으로 Color 카메라와 해상도를 맞추기 위해 영상의 크기를 변환한다.

Depth 영상의 크기를 변환하는 과정에서는 Algorithm에 최적화하기 위한 960x1080의 크기로 영상의 크기를 resize한다. 960x1080이 최적의 해상도인 이유는 Full HD의 영상의 경우 1920x1080이기 때문이다. 실시간 DIBR Algorithm에서는 좌영상과 우영상을 생성하여 출력하여야 한다. 따라서 좌영상과 우영상 각각 960x1080의 크기로 resize하여 후에 화면에 출력하게 된다. Color 카메라 영상 역시 Depth 카메라 영상과 동일한 크기로 resize한다. 이렇게 함으로써 DIBR 알고리즘을 적용할 수 있는 가공된 Depth 영상과 Color 영상을 얻게 된다.

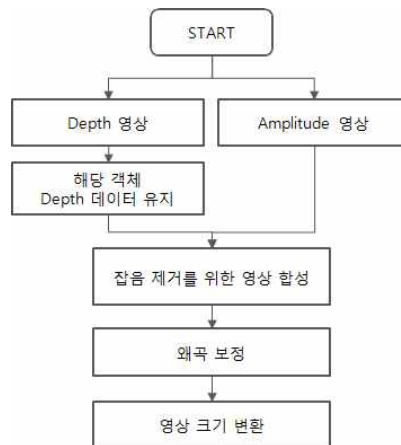


그림 10. Reverse-Gaussian Depth 영상의 전처리 과정
Fig. 10. Pre-processing procedure of Depth image

2. Reverse-Gaussian Depth 영상 생성

DIBR 영상을 생성하기 위해서는 앞서 설명한 바와 같이 Depth 영상이 필요하다. Reverse-Gaussian Algorithm을 위해서 먼저 추출된 객체의 Depth 영상을 이진화 과정을 거쳐 반전하게 된다. 이렇게 Depth 영상을 반전시키는 이유는 후에 Gaussian Equation을 적용할 때, 객체의 영역이 기존 방식과 달리 팽창되지 않고 축소가 되기 때문에 배경 영역이 커지게 되기 때문이다. Depth 영상에서 주변 배경의 영역을

확장시킴으로 인하여 후의 Warping후 생기게 되는 Hole 영역의 왜곡 범위가 축소되는 효과를 가져오게 된다.



그림 11. 이진화 과정을 거친 객체의 Depth Image 반전
Fig. 11. Reverse-Depth Image after Binary Equation

다음으로 영상에 Gaussian Filter를 적용해야 한다. 그림 12에서 확인할 수 있듯이 Filter를 거친 Reverse-Depth 영상은 객체의 크기가 축소된 것을 확인할 수 있다. 이는 반대로 객체 주변 배경의 영역이 확장되었다는 것을 의미한다. 축소된 객체의 영역은 후에도 그대로 축소되어 손실되는 것이 아니라, 이후 영상의 Warping 과정과 Hole-filling 과정을 통해 다시 복원하게 된다.



그림 12. Gaussian Filter를 적용한 반전된 Depth Image
Fig. 12. Reverse-Gaussian Depth Image Using Gaussian Filter

3. Reverse-Gaussian Hole-Filling

다음으로 영상의 객체를 Warping해야 한다. Gaussian Algorithm과 마찬가지로 위 방법 또한 Hole-Filling 과정과 동시에 진행되게 된다. 위에서 적용한 Reverse-Gaussian 이미지를 활용하여 Gaussian-Warping 공식을 적용하여 객체의 배경을 확장시킨 후, 객체의 크기를 줄인 color 영상을 얻어내게 된다. 이러한 과정을 거친 중간 결과 영상의 Hole은, 원본에 비해 왜곡으로 심하게 일그러지지만, 주변의 왜곡은 크게 감소하게 된다. 그림 13는 Hole이 심하게 일그러져 왜곡된 중간 결과 영상을 보여주고 있다.



그림 13. Reverse-Gaussian을 통해 얻은 중간 결과 영상
Fig. 13. Interim Result Image of Reverse-Gaussian DIBR

마지막으로 위와 같이 얻어진 왜곡된 영상에, 기존의 객체를 그대로 덮어씌운다. 이때 영상에서 객체가 이동한 거리만큼의 픽셀을 옮기는 과정을 거친 후 덮어씌워 왜곡된 객체를 복원해주게 된다. 이는 이전의 왜곡된 영상이, 객체가 상대적으로 축소된 Reverse-Gaussian Depth영상을 사용하여 생성된 영상이기 때문에 덮어씌우는 과정이 매끄럽게 이루어질 수 있다. 위 방법을 사용하면 결과적으로 Hole은 배경의 확장으로 인해 문제없이 복원 가능하다.



그림 14. Reverse-Gaussian을 통해 얻은 최종 결과 영상
Fig. 14. Final Result Image of Reverse-Gaussian DIBR



그림 15. 원본 영상(좌)과 결과영상(우)의 비교
Fig. 15. Comparison Original Image with Result Image

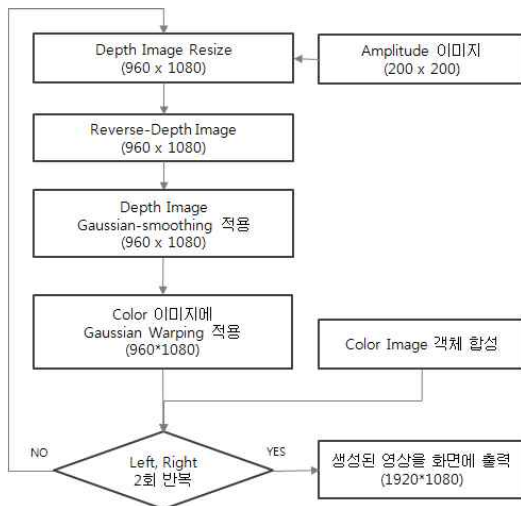


그림 16. Reverse-Gaussian Algorithm의 흐름도
Fig. 16. Flowchart of Reverse-Gaussian Algorithm

IV. 실험 결과

1. 화질 평가 방법

4개의 알고리즘을 서로 비교하기 위해서 측정값으로 FPS(Frame Per Second)와 PSNR(Peak Signal-to-noise ratio)값을 활용하였다 [14]. PSNR은 신호가 가질 수 있는 최대 전력에 대한 잡음의 전력을 나타낸 것이며, 주로 영상 또는 동영상 손실 압축에서 화질 손실 정보를 평가 할때 사용된다. 따라서 서로 다른 두 영상에 대한 차이를 수치나 단위, 정량적으로 느끼고 알 수 있도록 숫자로 나타내기 위한 값을 알 수 있다.

$$\begin{aligned}
 PSNR &= 10 \cdot \log_{10} \left(\frac{MAX_I^2}{MSE} \right) \\
 &= 20 \cdot \log_{10} \left(\frac{MAX_I}{\sqrt{MSE}} \right) \quad (6)
 \end{aligned}$$

2. 각 Algorithm의 실험결과

식 (6)를 이용하여 PSNR 결과 값을 도출할 수 있다. 여기서 MAXI 는 해당 영상의 최대값으로, 해당 채널의 최대 값에서 최소값을 빼서 구할 수 있다. MSE는 해당 영상의 변화량을 나타내는 것으로, 무손실 영상의 경우에는 MSE가 0 이기 때문에 PSNR은 정의되지 않는다. 본 실험에 사용된 영상들은 8bit 그레이스케일영상이기 때문에 다음 경우에는 255 (255 - 0)가 된다. PSNR의 단위는 dB이며 로그스케일에서 측정하기 때문에, 손실이 적을수록 높은 값을 가진다. 또한 본 실험에서는 강아지 인형을 사용하여 실험결과를 얻었다.



그림 17. 실험에 사용된 Color Image와 Depth Image
Fig. 17. Color Image and Depth Image used in the test

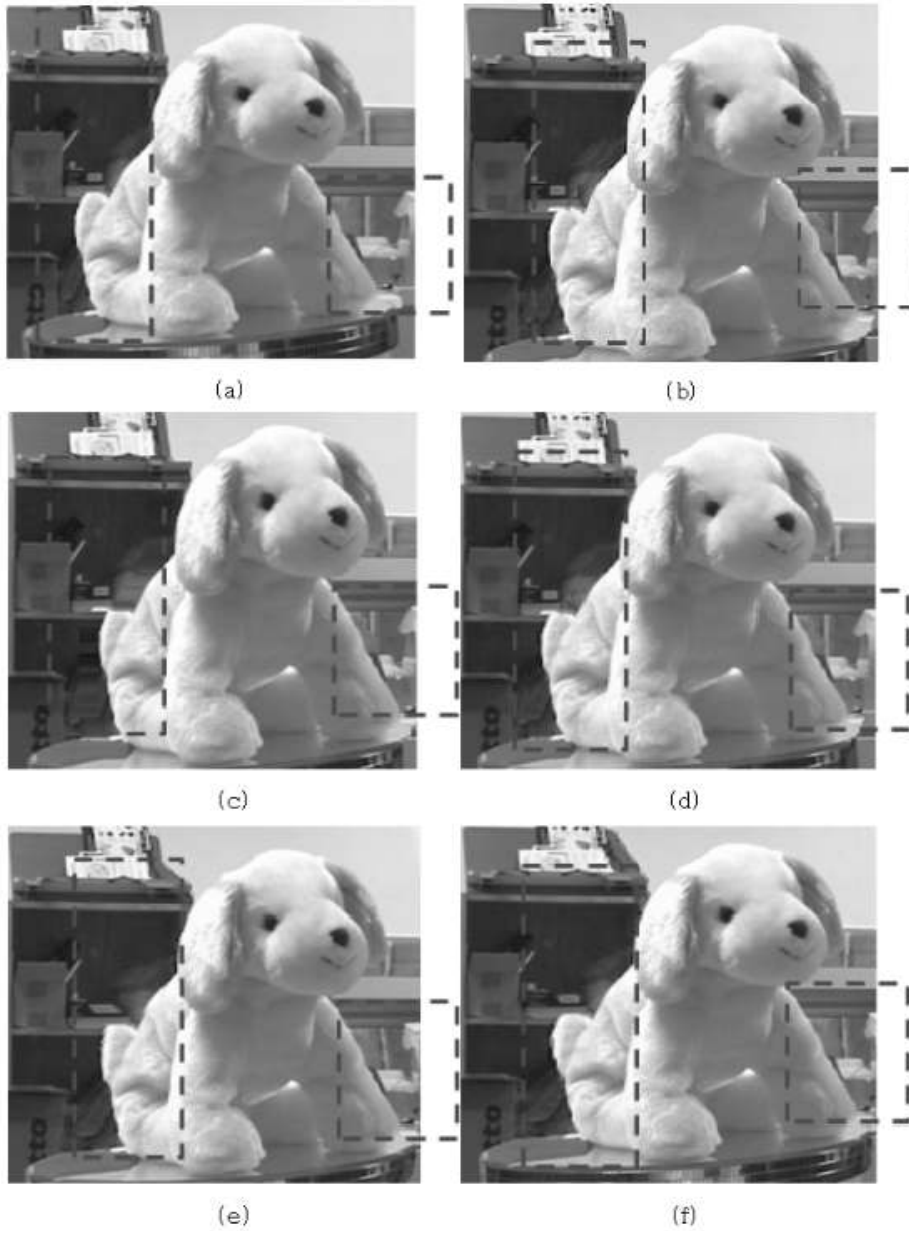


그림 18. 실험에 사용된 Algorithm들의 최종 영상

(a) 원본 이미지, (b) Reverse-Gaussian, (c)인접 픽셀 복사,

(d)인접 영역 확장 복사, (e) 대칭 Gaussian-Smoothing, (f) 비대칭 Gaussian-Smoothing

Fig. 18. Final image of Algorithm used in the test (a) Original image, (b) Reverse-Gaussian, (c) Copying adjacent pixel, (d) Adjacent area expansion, (e) Symmetric Gaussian-Smoothing (f) Asymmetric Gaussian-Smoothing

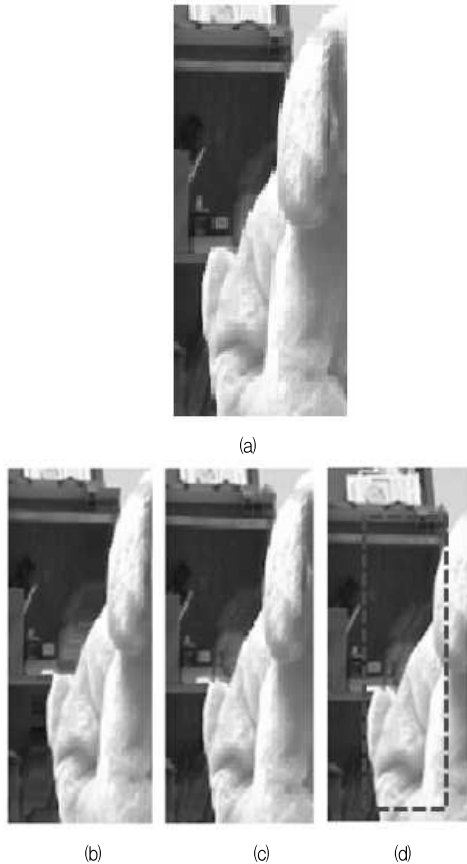


그림 19. 실험에 사용된 Algorithm들의 최종 영상 (a) 원본 이미지, (b) 인접 픽셀 복사, (c) 인접 영역 확장 복사 (d) Reverse-Gaussian
 Fig. 19. Final image of Algorithm used in the test (a) Original image, (b) Copying adjacent pixel, (c) Adjacent area expansion, (d) Reverse-Gaussian

실험결과 이미지를 살펴보면 5개의 이미지의 품질을 실제 육안으로 비교한 결과를 나타내고 있다. 그림 18에서 알 수 있듯이 Gaussian 계열의 알고리즘들이 우수한 품질의 이미지를 보여주었고, 인접 픽셀 복사 방식이 가장 품질저하가 되었음을 확인할 수 있었다. 또한 Gaussian 계열의 알고리즘 중에서도 본 논문에서 제시하는 Reverse-Gaussian Algorithm은 Hole의 변형이 적음과 동시에 선반의 왜곡도 일어나지 않았음을 확인할 수 있다. 점선으로 표현된 부분은 실시간 DIBR에 각 Algorithm을 적용하였을 때 중점적으로 비교해야 할 부분이다. 기존의 Depth 영상에 Gaussian Filter를 사용하는 Algorithm들의 경우 배경의 창틀과 책장이 늘어지는 현상이 발생하였음을 확인할 수 있었다. 이처럼 Gaussian Filter를 사용하여 DIBR을 수행하면 주변 배경

픽셀에 영향을 주게 되지만, (b)에서 확인한 것과 같이 반전된 Depth map과 원본 영상을 사용함으로써 육안으로 비교할 때 훨씬 더 자연스럽게 보이는 것을 확인할 수 있었다.

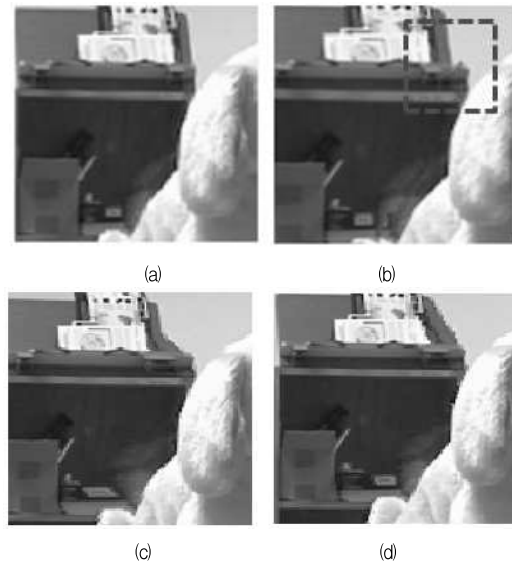


그림 20. 실험에 사용된 Algorithm들의 최종 영상 (a) 원본 이미지, (b) Reverse-Gaussian, (c) 대칭 Gaussian-Smoothing, (d) 비대칭 Gaussian-Smoothing
 Fig. 20. Final image of Algorithm used in the test (a) Original image, (b) Reverse-Gaussian, (c) Symmetric Gaussian-Smoothing (d) Asymmetric Gaussian-Smoothing

또한 이를 보다 상세하게 확대한 그림 19에서 확인할 수 있듯이, 점선으로 표시한 Reverse-Gaussian과 달리 인접 픽셀 복사 방법과 인접 영역 확장 복사 방법은 배경의 왜곡은 없었지만 위의 그림과 같이 Warping후 Hole이 채워지는 과정에서 품질저하가 일어남을 확인할 수 있었다.

Gaussian 계열의 Algorithm들을 확대한 그림 20의 경우 Warping후 Hole이 채워지는 과정에서는 보다 자연스러운 결과를 확인할 수 있다. 하지만 주변 배경에 왜곡이 생긴다는 단점이 있다. 위의 그림 20과 같이 배경의 선반이 Warping하는 방향으로 번지는 현상이 일어나 "Puppy" 객체 뒤의 선반배경의 품질저하가 발생하게 된다. 비대칭 Gaussian-smoothing의 경우 이러한 배경 왜곡을 대칭 Gaussian-smoothing에 비해 많이 개선되었지만 개선된 배경의 선반 또한 왜곡이 심한 것을 확인할 수 있다. 이에 비해 Reverse-Gaussian은 점선으로 표시한 선반의 왜곡이 최소화되었음을 확인하였다.

표 1. "Puppy" DIBR Image의 PSNR[dB] 비교
Table 1. PSNR analysis of "Puppy" DIBR Image

이동한 Pixel 개수 사용 Algorithm	1	2	3	4	5
인접 픽셀 복사 방식	30.9045	27.3110	25.6008	24.4572	23.6243
인접 영역 확장 복사 방식	30.9305	27.3247	25.6125	24.4692	23.6370
대칭 가우시안	31.0997	27.5547	25.7246	24.5572	23.6994
비대칭 가우시안	31.4786	27.6453	25.8569	24.5971	23.7201
Reverse-Gaussian	30.9981	26.5954	25.6962	24.5173	23.6551

표 2. "Puppy" DIBR Image의 FPS(Frame Per Second) 비교
Table 2. FPS analysis of "Puppy" DIBR Image

이동한 Pixel 개수 사용 Algorithm	1	2	3	4	5
인접 픽셀 복사 방식	14.5019	14.6714	13.8754	13.9668	14.4951
인접 영역 확장 복사 방식	14.2951	13.7177	13.9841	14.0922	13.7701
대칭 가우시안	6.7232	7.0740	6.9903	7.0359	6.9507
비대칭 가우시안	3.7210	3.6873	3.7477	3.7011	3.8078
Reverse-Gaussian	6.4212	6.4441	6.3412	6.5102	6.4611

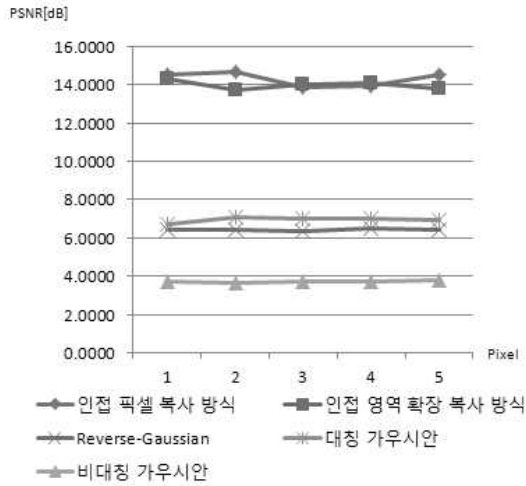


그림 22. 실험에 사용된 Algorithm들의 FPS 비교
Fig. 22. FPS of Algorithm used in the test

표 1과 그림 21은 본 논문에서 실험한 4개의 알고리즘의 PSNR결과이다. PSNR의 경우 높을수록 좋은 품질의 이미지를 의미하기 때문에 Warping하는 Pixel이 많으면 많을수록 원 영상과 다른 영상이 되기 때문에 전체적인 PSNR 결과는 낮아지게 됨을 알 수 있다. 그리고 각 Algorithm의 PSNR을 비교한 결과 비대칭 Gaussian-Smoothing Algorithm이 가장 좋은 품질을 얻은 것을 확인할 수 있다. 반면에 원본에 비해 가장 품질저하가 심한 영상은 인접 픽셀 복사 방식을 확인할 수 있다. 이를 통해 인접 픽셀 복사 방식은 Hole을 채울 수 있는 특별한 연산이 적용되지 않기 때문에 Hole을 채운 영역이 어색하게 표현된다는 것을 확인할 수 있었다. Reverse-Gaussian은 육안으로는 뛰어난 품질을 보여주지만 실제 PSNR 수치에서는 원본과 합성하는 과정을 거치기 때문에 기존 Gaussian 계열의 Algorithm에 비해 PSNR값이 낮게 나옴을 확인할 수 있었다.

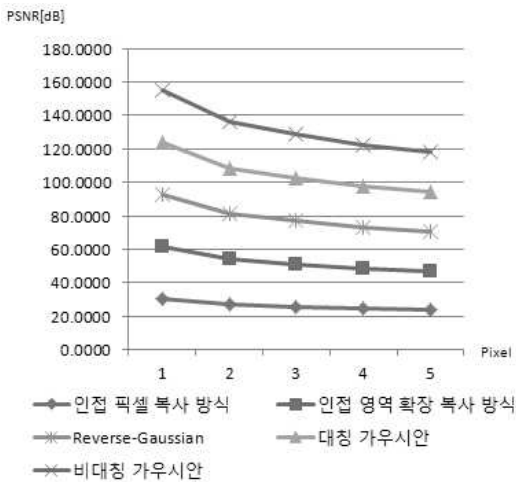


그림 21. 실험에 사용된 Algorithm들의 PSNR 비교
Fig. 21. PSNR of Algorithm used in the test

표 2와 그림22는 본 논문에서 실험한 4개의 알고리즘의 FPS(Frame Per Second)를 나타낸 결과이다. FPS가 높으면 높을수록 빠른 알고리즘을 의미하기 때문에 인접 픽셀 복사 방식이 가장 좋은 결과를 보여주는 것을 알 수 있다. Reverse-Gaussian은 Gaussian 연산을 거쳐야 하기 때문에, 픽셀을 참조하는 방식에 비해 FPS가 낮음을 확인할 수 있었다. 하지만 기존의 품질이 가장 좋았던 비대칭 Gaussian Algorithm에 비해 FPS 면에서 빠른 결과를 보여주면서 배경의 왜곡 또한 없는 결과를 확인할 수 있었다.

V. 결 론

우리는 지금까지 다양한 DIBR Algorithm을 실시간 영상에 적용하여 그 성능을 비교, 분석하였다. 또한 기존의 정지 영상과 다르게 실시간 영상에서 발생하는 Depth 영상의 잡음제거, 영상 크기 맞춤, 알고리즘 최적화 등을 통하여 실험 결과의 신뢰도를 향상시켰다.

본 논문에서 제시한 Reverse-Gaussian Algorithm은 품질에 있어서 다른 Algorithm들에 비해 뛰어난 모습을 보여주었다. 인접 픽셀 복사 방식은 가상 시점 영상을 생성하는데 가장 빠른 처리 속도를 가지며 이동 크기에 상관없이 항상 같은 처리 속도를 가지게 되지만, Hole 주변의 영상의 품질저하가 가장 심하며 Depth 이미지에 의존적이라는 단점을 확인하였다. 따라서 높은 품질의 Depth 이미지가 필요하다. 또한 인접 영역 확장 복사 방식도 인접 픽셀 복사 방식보다는 Hole에 대한 자연스러운 품질을 가지지만 어떤 Hole이든 인접 영역의 2배를 늘리기 때문에 때때로 저 품질의 결과를 가져왔다. 본 Algorithm은 이러한 Hole의 품질저하를 막을 수 있음을 확인하였다.

대칭 Gaussian Algorithm의 경우 비대칭 Gaussian 보다는 빠르지만 다른 방식에 비해서는 느린 처리 속도를 보여주었다. 또한 Sigma 값에 의해 처리 속도가 확연하게 달라지며 Sigma값은 이동 거리가 커질수록 그 값이 증가해야 적절한 결과를 얻어 낼 수 있다. 가장 큰 문제가 된 배경의 왜곡을 막기 위해 비대칭 Gaussian 방식을 사용하였지만, 마찬가지로 주변부의 왜곡을 확인할 수 있었다. 이러한 점으로 보아 본 Algorithm은 Gaussian에서 문제가 되는 주변부 수직 왜곡을 해결 할 수 있어 품질에 있어 매우 효율적임을 확인하였다.

본 연구의 성능분석은 3D TV에서의 DIBR 뿐만이 아니라 현재 활발한 연구 활동 중인 3D 체감 게임기, 3D Video 등 다양한 응용분야에서 적용될 수 있을 것이다.

참고문헌

- [1] C. Fehn, R. Barre, and R. S. Pastoor, "Interactive 3-DTV concepts and key technologies," Proc. of the IEEE, vol. 94, no. 3, pp. 524-38, Mar. 2006.
- [2] C. Fehn, E. Cooke, O. Schreer, and P. Kauff, "3D Analysis and Image-Based Rendering for Immersive TV Applications," Signal Processing: Image Communication 17, pp. 705-15, Oct. 2002.
- [3] J. Flack, P. Harman, S. Fox, "Low bandwidth stereoscopic image encoding and transmission", Proceedings of SPIE Conference on Stereoscopic Displays and Virtual Reality Systems X, Vol. 5006, pp. 206-214, CA, U.S.A., Jan. 2003.
- [4] C. Fehn, "Depth-image-based rendering (DIBR), compression and transmission for a new approach on 3D-TV", Proceedings of SPIE Conference on Stereoscopic Displays and Virtual Reality Systems XI, Vol. 5291, pp. 93-104, CA, U.S.A., Jan. 2004.
- [5] Wan-Yu Chen, Yu-Lin Chang, Shyh-Feng Lin, Li-Fu Ding, and Liang-Gee Chen, "Efficient Depth Image Based Rendering with Edge Dependent Depth Filter and Interpolation."Multimedia and Expo, ICME 2005, pp. 1314- 1317, July 2005,
- [6] Y. Mori, N. Fukushima, T. Yendo, T. Fujii, and M. Tanimoto, "View generation with 3D warping using depth information for FTV," Signal Processing: Image Communication, vol. 24, pp. 65-2, 2009.
- [7] A. Smolic, K. Muller, K. Dix, P. Merkle, P. Kauff, and T. Wiegand, "Intermediate view interpolation based on multi-viewvideo plus depth for advanced 3D video systems," Proc. of ICIP, pp. 2448-451, 2008.
- [8] C. Fehn, "A 3D-TV approach using depth-image-based rendering (DIBR)", Proceedings of VIIP 03, Benalmadena, Spain, Sept. 2003.
- [9] L. Zhang and W. Tam, "Stereoscopic image generation based on depth images for 3D TV," IEEE Trans. Broadcast., vol. 51, no. 2, pp. 191-99, June 2005.
- [10] Wan-Yu Chen, Yu-Lin Chang, Hsu-Kuang Chiu, Shao-Yi Chien, and Liang-Gee Chen "Real-Time Depth Image based Rendering Hardware Accelerator for Advanced Three

Dimensional Television System', Proc. of the ICME IEEE, pp. 2069-2072, 9-12 July 2006

[11] W. J. Tam, G. Alain, L. Zhang, T. Martin, R. Renaud, "Smoothing depth maps for improved stereoscopic image quality", Proceedings of SPIE Conference on Three-dimensional TV, Video, and Display III, Vol. 5599, pp.162-172, Philadelphia, U.S.A., Oct. 2004.

[12] A. Woods, T. Docherty, R. Koch, "Image distortions in stereoscopic video systems", Proceedings of SPIE Conference on Stereoscopic Displays and Applications, pp. 36-48, San Jose, CA, USA, Feb. 1993.

[13] W. J. Tam, L. Zhang, "Non-uniform smoothing of depth maps before image-based rendering". Proceedings of SPIE Conference on Three-dimensional TV, Video, and Display III, Vol. 5599, pp.173-183, Philadelphia, U.S.A., Oct. 2004.

[14] ITU: 'Objective perceptual video quality measurement techniques for digital cable', ITU-T Recommendation J.144, Geneva, Switzerland, March 2003

저 자 소 개



안 양 근
 1994 : 한국해양대학교 제어계측공학과 공학사
 1997 : 한국해양대학교 제어계측공학과 공학석사
 현 재 : 전자부품연구원 책임연구원
 관심분야 : 시스템소프트웨어, 컴퓨터비전, HCI
 Email : ykahn@keti.re.kr



홍 지 만
 1994 : 고려대학교 컴퓨터공학과 공학사
 1997 : 서울대학교 컴퓨터공학과 공학석사
 2003 : 서울대학교 컴퓨터공학과 공학박사
 현 재 : 숭실대학교 컴퓨터학과 교수
 관심분야 : 시스템소프트웨어, 임베디드시스템
 Email : jimhan@ssu.ac.kr

