

일반논문 (Regular Paper)

방송공학회논문지 제19권 제2호, 2014년 3월 (JBE Vol. 19, No. 2, March 2014)

<http://dx.doi.org/10.5909/JBE.2014.19.2.1>

ISSN 2287-9137 (Online) ISSN 1226-7953 (Print)

계층적 결합형 양방향 필터를 이용한 실시간 깊이 영상 보정 방법

신 동 원^{a)}, 호 요 성^{a)‡}

Real-time Depth Image Refinement using Hierarchical Joint Bilateral Filter

Dong-Won Shin^{a)} and Yo-Sung Ho^{a)‡}

요 약

본 논문에서는 결합형 양방향 필터를 이용하여 깊이 영상을 실시간으로 보정하는 방법을 제안한다. 제안한 방법은 Kinect 깊이 카메라로부터 얻은 깊이 영상의 화질을 실시간으로 향상시키기 위해 GPU 내의 상수 메모리와 2차원 영상 처리에 적합한 텍스처 메모리를 사용한다. 또한, 단일 화소에 대한 결합형 양방향 필터 연산을 각 GPU 스레드(thread)에 할당한 다음 병렬로 처리하여 계산량을 현저히 감소시킨다. 그리고 깊이 영상의 품질을 더욱 높이기 위해 CUDA를 이용해 구현한 결합형 양방향 필터를 계층형 구조로 반복적으로 수행하여 폐색 영역이 채워진 깊이 영상을 얻을 수 있다. 실험 결과를 통해, 제안한 실시간 깊이 영상 보정 방법이 깊이 영상의 주관적 화질을 향상시키고, 초당 55 화면의 속도로 동작하는 것을 확인했다.

Abstract

In this paper, we propose a method for real-time depth image refinement. In order to improve the quality of the depth map acquired from Kinect camera, we employ constant memory and texture memory which are suitable for a 2D image processing in the graphics processing unit (GPU). In addition, we applied the joint bilateral filter (JBF) in parallel to accelerate the overall execution. To enhance the quality of the depth image, we applied the JBF hierarchically using the compute unified device architecture (CUDA). Finally, we obtain the refined depth image. Experimental results showed that the proposed real-time depth image refinement algorithm improved the subjective quality of the depth image and the computational time was 260 frames per second.

Keyword : depth map, refinement, joint bilateral filter, CUDA, hierarchical approach

a) 광주과학기술원 정보통신공학부 (School of Information and Communication)

‡ Corresponding Author : 호요성(Yo-Sung Ho)
E-mail: hoyo@gist.ac.kr
Tel: +82-62-715-2258

· Manuscript received January 13, 2014 Revised March 18, 2014
Accepted March 19, 2014

1. 서 론

최근 3차원 비디오 시스템은 초고해상도를 특징으로 하는 UHD TV, 초고속 인터넷 망을 이용하여 제공되는 양방

향 서비스인 IPTV 시스템 등과 더불어 중요한 차세대 방송 시스템 중의 하나로 각광받고 있다. 3차원 비디오 시스템은 3차원 비디오 장면 정합, 압축, 렌더링 등의 다양한 주요 요소로 구성되어 있으며, 3차원 비디오 콘텐츠는 스테레오 카메라, 다시점 카메라, 깊이 카메라와 같은 다양한 영상 제작 도구들을 이용하여 획득할 수 있다. 그리고 이 콘텐츠들은 3차원 비디오 압축 방법을 통해 부호화되어 영상출력 장치에 전달되고, 이러한 장치들은 전달된 영상 정보를 DIBR (depth image-based rendering) 기술을 통해 재현해서 영상을 화면에 출력한다^[1].

DIBR 기술은 동일한 시점의 색상 영상과 깊이 영상을 이용해 가상의 시점을 합성할 수 있다. 깊이 영상은 3차원 장면에서 객체와 카메라와의 거리를 나타내는 정보를 포함하고 있기 때문에, 깊이 영상을 이용하면 사용자가 원하는 가상 시점에서의 위치를 계산할 수 있다. 가상 시점 영상을 생성하기 위해서는 첫번째로 색상 영상의 모든 화소들을 카메라 매개변수와 깊이 영상을 통해 세계 좌표계 (world coordinate)로 역사상 (back-projection) 시킨다. 그런 다음, 3차원 공간으로 이동된 이 화소들을 다시 가상 시점의 영상 평면으로 사상한다. 이러한 과정을 3차원 워핑이라 한다^[2].

현재에 이르러 Microsoft 사의 Kinect 깊이 카메라가 가격적인 가격으로 출시됨에 따라 일반 가정에서도 3차원 비디오를 쉽게 접할 수 있게 되었다. 앞서 소개한 DIBR 기술을 사용해 Kinect 깊이 카메라로부터 획득한 색상 영상과 깊이 영상을 사용하여 3차원 장면을 쉽게 복원할 수 있다. 하지만 그림 1(b)에서 볼 수 있듯이, Kinect 깊이 카메라로부터 획득한 깊이 영상은 깊이값 추정에 실패한 영역들이 많이 포함하고 있다. 이러한 영역은 Kinect 깊이 카메라에 장착된 적외선 송신기와 수신기의 물리적인 거리 차이로 인한 폐색 영역 발생, 객체의 경계 불일치 등 다양한 센서 잡음 때문에 발생한다. 그림 1(a)와 그림 1(b)에서 볼 수 있듯이, 원본 깊이 영상에서 책상 위와 물체 주변에 검은색으로 표현되어 있는 센서 잡음을 확인할 수 있다. 이러한 깊이 영상의 오차는 3차원 비디오 시스템의 부호화, 복호화 등의 계산 과정에서 오차가 그대로 전파되기 때문에 재현단계 이르렀을 때 양질의 합성 영상을 시청자에게 제공할 수 없게 만든다.

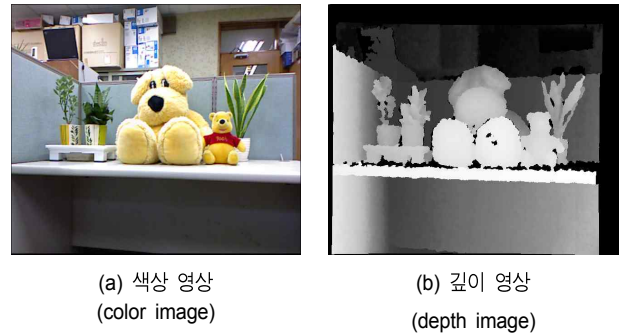


그림 1. Kinect 카메라로부터 획득한 영상
Fig 1. Image from Kinect camera

깊이 영상 오차를 제거하기 위해 최근 다양한 방법이 연구되고 있다. 이 가운데 대표적인 방법으로 결합형 양방향 필터 (joint bilateral filter, JBF)를 들 수 있다^[3]. 이 방법은 표적 화소와 주변 화소 간의 거리 차이를 반영하는 공간 (spatial) 필터와 화소값 차이를 반영하는 범위 (range) 필터를 사용하는데, 범위 필터를 보조 영상, 즉, 색상 영상의 것으로 대체함으로써 색상 영상의 경계 정보를 깊이 영상에 그대로 반영한다. 따라서, 두 영상 간의 객체 경계 불일치 문제를 효과적으로 제거함과 동시에 깊이값 오차를 보정할 수 있다.

본 논문에서는 결합형 양방향 필터를 기반으로 하여 실시간으로 깊이 오차를 제거하는 방법을 제안한다. 제안한 방법은 Kinect 깊이 카메라로부터 획득한 색상 영상을 이용하여 깊이 영상에 대해 결합형 양방향 필터를 적용한다. 이 필터는 성능을 결정하는 다양한 인자와 Gaussian 함수와 같은 복잡한 수식이 포함되는 필터이기 때문에, 이를 실시간으로 처리하도록 구현하기 위해 CUDA (compute unified device architecture) 병렬 프로그래밍을 통해 각 화소에 대한 계산을 동시에 처리함으로써 수행시간을 줄인다. 하지만, 영상 내에 넓게 존재하는 폐색 영역을 채우기 위해서는 결합형 양방향 필터를 반복적으로 수행해야 한다. 따라서, 본 논문에서는 계층적 결합형 양방향 필터 구조를 통해 폐색 영역을 고속으로 채우는 방법을 사용한다. 해상도에 따라 계층을 나누어 하위 계층에서 상위 계층까지 반복적으로 결합형 양방향 필터를 수행하면 대부분의 홀이 채워진 깊이 영상을 얻을 수 있다.

본 논문의 구성은 다음과 같다. 제 2장에서는 CUDA를 이용한 결합형 양방향 필터와 이를 계층적으로 구현한 방법에 대해서 기술한다. 3장에서는 CUDA의 내부에서 계산 속도를 높이는 방법에 대해 자세히 살펴본다. 4장에서는 CPU와 GPU로 구현한 결합형 양방향 필터에 대해 주관적 화질과 계산시간을 비교하고 객관적인 화질 향상에 대한 평가를 위해 bad pixel rate를 측정하는 실험을 수행했다. 5장에서는 이러한 내용을 바탕으로 결론을 기술한다.

II. 제안하는 방법

1. CUDA를 이용한 결합형 양방향 필터

기존에는 중앙 처리 장치(central processing unit, CPU)를 이용한 순차적인 프로그래밍이 널리 사용되었으나, 최근에는 그래픽 처리 장치(graphic processing unit, GPU)를

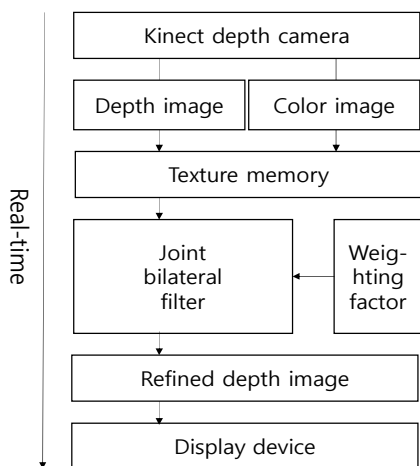


그림 2. CUDA를 이용한 결합형 양방향 필터의 순서도
 Fig 2. Flowchart of joint bilateral filter using CUDA

이용한 병렬 프로그래밍이 많은 관심을 받고 있다. 특히, NVIDIA사에서 CUDA라는 C언어 기반의 병렬처리 라이브러리를 배포하면서 연산 시간 단축에 크게 기여하고 있다^[4].

제안하는 방법에서는 Kinect로부터 얻은 동일한 장면에 대한 깊이 영상과 색상 영상을 앞서 설명한 CUDA를 이용

하여 구현된 결합형 양방향 필터를 통해 실시간으로 보정한다. 그림 2는 CUDA를 이용한 결합형 양방향 필터의 순서도를 나타낸다.

먼저 Kinect로부터 얻은 깊이 영상과 색상 영상을 2차원 영상 처리에 적합하게 설계된 CUDA의 텍스처 메모리에 적재한 다음, 이 영상에 대한 결합형 양방향 필터 계산을 병렬 수행한다. 단일 화소에 대한 결합형 양방향 필터 계산을 GPU의 스레드 함수 하나에 할당하고 영상에 존재하는 화소의 개수만큼 스레드를 동시에 수행함으로써 속도를 개선시킨다.

또한, 계산 시간을 줄이기 위해서 결합형 양방향 필터 내부에서 계산할 가중치 인자를 스레드 함수 실행 이전에 미리 계산해두고, 그 값이 필요할 때에만 가져다 쓰는 방법을 사용했다. 그리고 이 가중치 인자는 GPU 내부의 다른 메모리들보다 상대적으로 읽기 접근 속도가 빠른 상수 메모리에 저장하여 수행 속도를 더욱 향상시킨다.

가중치 인자를 계산할 때에는 커널의 크기가 커짐에 따라 연산량이 급격하게 증가한다. 이러한 2차원 Gaussian 함수는 1차원 함수의 곱으로 분할할 수가 있기 때문에, 이를 이용하여 보다 빠른 처리 속도를 보여주는 필터를 구현한다^[5].

결합형 양방향 필터의 수식을 살펴보면 식 (1)과 같다.

$$D_o(x, y) = \frac{\sum_{u \in U_p} \sum_{v \in V_p} W(u, v) D_i(u, v)}{\sum_{u \in U_p} \sum_{v \in V_p} W(u, v)} \quad (1)$$

여기서 D_i 는 입력 깊이 영상을 나타내고 W 는 가중치 인자이다. 이것을 (u, v) 에 위치한 화소에 대해서 각각을 계산하고 가중치 인자의 총합으로 나누는 정규화를 거쳐서 D_o 를 계산한다. u 와 v 는 필터 커널 내부의 특정 위치를 나타내는 벡터 u_p 와 v_p 의 원소이다. 따라서 다음과 같이 나타낼 수 있다.

$$\vec{u}_p = \{x - r, \dots, x + r\} \quad (2)$$

$$\vec{v}_p = \{y - r, \dots, y + r\} \quad (3)$$

수식 (2)와 수식 (3)에서 각각의 기호를 그림으로 설명하면 그림 3과 같이 나타낼 수 있다. 검은색으로 표시된 격자는 영상의 화소들을 나타내고 바깥쪽의 두꺼운 선으로 표시된 사각형은 필터 커널을 나타낸다. 필터 커널에서 (u,v) 는 인접 화소의 위치를 나타내고 (x,y) 는 중심 화소의 위치를 나타낸다. 또한 r 은 필터 커널의 반지름을 나타낸다. 앞에서 설명한 벡터 u_p 와 v_p 는 수평과 수직의 위치에 나타냈다.

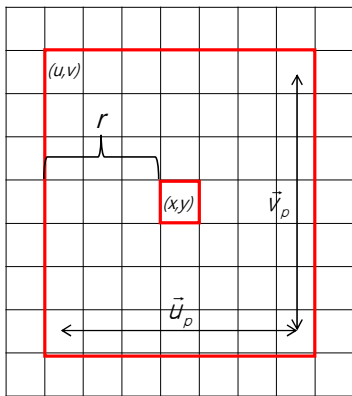


그림 3. 기호에 대한 설명
Fig 3. Description on a symbol

또한 가중치 인자 W 는 다음과 같이 정의된다.

$$W(u,v) = \begin{cases} 0 & , \text{ if } D_i(u,v) = 0 \\ g(u,v) \cdot f(u,v) & , \text{ otherwise} \end{cases} \quad (4)$$

$f(u,v)$ 는 거리 차이에 따라 결정되는 공간 필터를 나타내고, $g(u,v)$ 는 필터 커널에서 표적 화소와 주변 화소와의 색상 차이에 따라 결정되는 범위 필터를 나타낸다. 이 두 함수는 모두 Gaussian 함수를 이용하여 다음과 같이 정의된다.

$$g(u,v) = \exp\left\{-\frac{|I(x,y) - I(u,v)|^2}{2\sigma_R^2}\right\} \quad (5)$$

$$f(u,v) = \exp\left\{-\frac{(x-u)^2 + (y-v)^2}{2\sigma_D^2}\right\} \quad (6)$$

여기서 σ_R 과 σ_D 는 각각의 Gaussian 함수에 대한 표준 편차를 의미한다. 이 값에 따라 Gaussian 분포의 폭이 결정된다.

2. 계층적 접근법

결합형 양방향 필터를 한번 수행한 결과로는 Kinect로부터 얻은 원본 깊이 영상의 모든 부분을 채울 수 없다. 깊이 값이 서로 다른 부분임에도 불구하고 같은 값으로 채워지거나 경계가 다소 불분명한 부분이 존재한다. 따라서 원본 깊이 영상에서의 깊이 값이 없는 폐색영역을 모두 채우기 위해서 결합형 양방향 필터를 계층적으로 수행하는 접근 방법을 사용할 수 있다^[6]. 계층적 접근 방법은 그림 4와 같은 방법으로 수행된다.

계층 1은 원래 크기의 색상 영상과 깊이 영상을 가지고 있다. 계층의 숫자가 높아질수록 하위 계층이며 하위 계층으로 갈수록 영상의 크기가 1/2로 줄어든다. 예를 들어 원본 영상의 크기가 640x480이면 계층 2에서의 영상의 크기는 320x240이 된다. 또한 계층 3에서는 160x120으로 줄어든다. 그림 4에서 아래 계층으로 갈수록 해상도가 낮아지는 것을 알 수 있다. 영상의 크기는 하위 계층으로 갈수록 작아지지만 결합형 양방향 필터의 크기는 모든 계층에서 동일한 크기를 사용한다. 또한 본 논문의 실험에서는 하위계층의 제한을 계층 5까지로 제한을 했다.

그림 5는 계층 사이의 깊이 영상 보정 과정을 자세하게 표현한 그림이다. 먼저 각 계층의 영상을 생성한 뒤에 가장 하위 계층에서 그 계층의 색상 영상과 깊이 영상을 이용하여 결합형 양방향 필터를 수행한다. 이 결과로 나온 깊이 영상을 상위 계층의 깊이 영상에서 깊이 값이 없는 폐색 영역에 복사한다. 위 계층과 아래 계층의 영상 크기가 2배의 차이를 가지므로 위 계층의 깊이 영상에 2개의 화소 당 1개의 화소 씩 아래 계층의 깊이 화소 값으로 채워진다. 이렇게 갱신된 깊이 영상과 해당 계층의 색상 영상을 이용하여 다시 결합형 양방향 필터를 수행한다. 이 결과로 나온 보정된 깊이 영상을 이용하여 다시 상위 계층의 깊이 영상에서 깊이 값이 없는 폐색 영역에 복사하고 결합형 양방향 필터를 수행하는 일련의 과정을 가장 상위 계층에 도달할

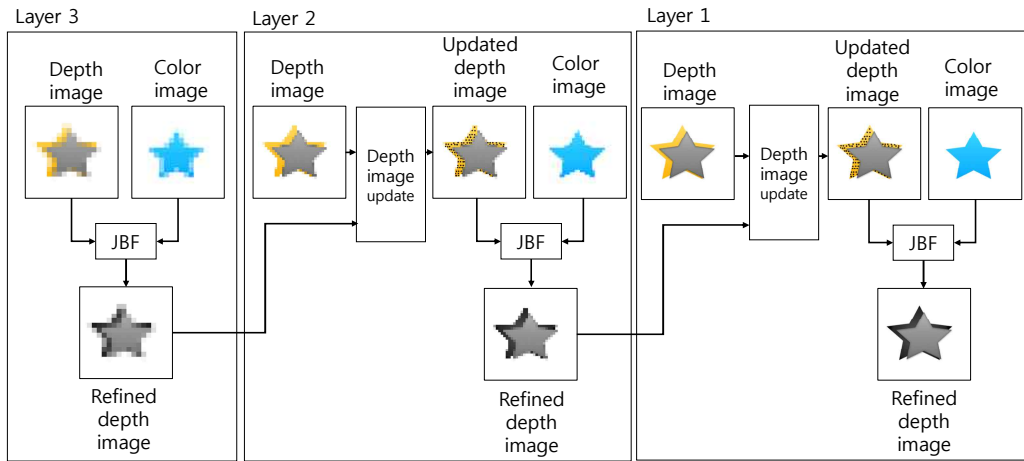


그림 4. 계층적 접근법의 순서도
 Fig 4. Flowchart of hierarchical approach

때까지 반복하여 수행하면 최종적으로 대부분의 폐색 영역이 채워진 깊이 영상을 얻을 수 있다. 그림 6은 계층 수를 늘려감에 따라 계층적 접근법을 수행한 결과이다. 계층 수가 늘어날수록 깊이 영상의 화질이 개선되는 것을 확인했다. 먼저 1개의 계층을 사용한 깊이 영상에서 좌상단의 물체에 대해 찌그러진 형태의 깊이 정보를 획득했다^[7]. 하지만 4개의 계층을 사용한 깊이 영상에서는 물체 원형 그대로의 깊이 정보를 획득했다. 또한 Kinect 깊이 카메라의 시야각 때문에 영상의 테두리 부분에서 깊이 값을 얻지 못하는 부분이 존재했는데 이러한 부분에서도 계층 수를 늘려감에 따라 적절한 깊이 값으로 채워진 영상을 획득했다. 그리고 탁자의 오른쪽 영역에서 생긴 홀 부분도 탁자에 대한 깊이 값으로 채워주고 있는 것을 알 수 있다.

III. CUDA

CUDA는 GPU를 이용한 대용량의 데이터에 대한 병렬 계산을 하는데 기반이 되는 통합 개발 환경이다. 본 논문에서는 CUDA를 이용하여 앞서 소개한 결합형 양방향 필터의 연산을 병렬적으로 수행함으로써 계산 시간을 줄인다.

1. 상수 메모리

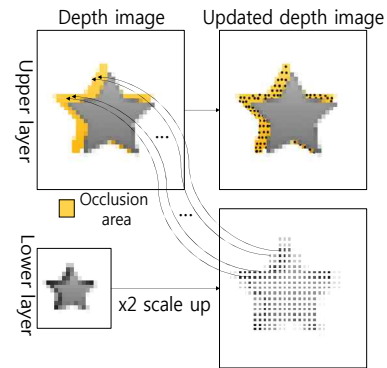


그림 5. 계층 사이의 깊이 영상 보정 과정
 Fig 5. Depth map refinement between levels

제안한 방법은 연산 속도 향상을 위해 결합형 양방향 필터에서 사용되는 가중치 인자들을 미리 계산한 뒤 그 값을 CUDA에서 제공하는 상수 메모리에 저장하고 실제 계산 수행 시에는 이를 참조하여 쓰는 방법을 사용했다. 상수 메모리는 DRAM에 있는 데이터를 읽기 전용으로 사용하며 캐시를 지원하는데 최초로 읽어오는 데이터는 DRAM에서 가져오기 때문에 GPU클럭으로 최소 400, 최대 500 사이클이 소요되지만 한번 캐시에 올라온 값을 반복하여 재사용할 때는 레지스터와 동일한 속도로 사용할 수 있게 되어 데이터에 빠르게 접근할 수 있도록 도와준다. 이러한 방법

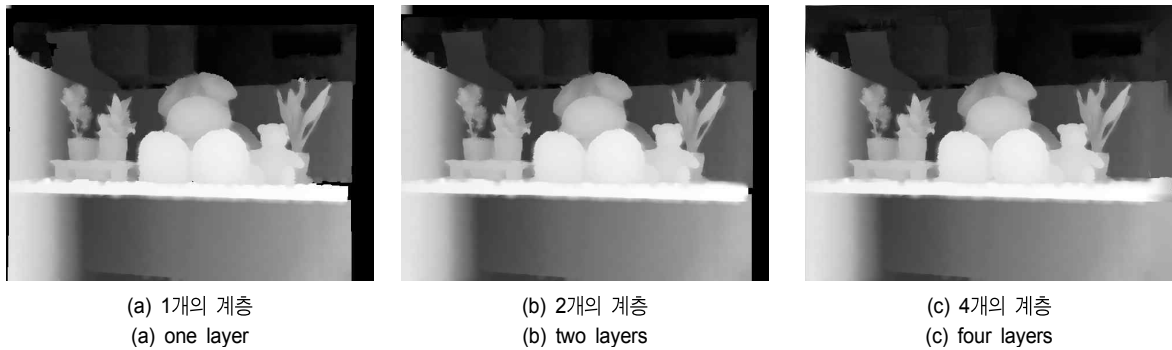


그림 6. 계층 수의 증가에 따른 결과 깊이 영상
Fig 6. Result depth map with increase

을 사용하여 본 논문에서는 상수 메모리와 미리 계산된 가중치 인자를 이용해서 속도를 향상시킨다.

2. 텍스처 메모리

CUDA에서 2차원 영상 처리와 관련하여 유용하게 쓰일 수 있는 것이 바로 텍스처 메모리이다. 텍스처 메모리는 오프칩인 DRAM에 대한 메모리 요청을 줄여줌으로써 매우 효과적인 대역폭을 제공할 수 있다. 다시 말해, 텍스처 캐시는 공간 구역성을 자주 드러내는 메모리 접근 패턴을 가진 그래픽스 어플리케이션을 위해 자주 사용된다.^[8] 여기서 공간 구역성이란 일단 하나의 기억 장소가 참조되면 그 근처의 기억 장소가 계속 참조되는 경향이 있는 특성을 의미한다. 2차원 영상 처리 어플리케이션에서는 이러한 공간 구역성이 잘 드러나는 경우를 많이 찾아볼 수 있기 때문에 CUDA의 텍스처 메모리는 2차원 영상 처리에 있어서 매우 우수한 성능을 보인다. 따라서, 제안하는 방법에서는 텍스처 메모리에 색상 영상과 깊이 영상을 미리 적재하고 결합형 양방향 필터의 계산 시에 빠르게 접근 할 수 있도록 함으로써 실시간의 성능을 낼 수 있도록 설계를 했다.

IV. 실험 결과

먼저 제안한 깊이 영상 보정 방법을 Intel Xeon CPU(2.53Ghz)로 수행된 영상과 Geforce GTX Titan GPU

로 수행된 영상을 그림 7과 그림 8에 나타냈다. 영상의 해상도는 색상 영상과 깊이 영상 모두 640x480이다. Kinect에서 기본적으로 획득한 영상의 해상도는 색상 영상이 640x480이고 깊이 영상이 320x240이지만 OpenNI의 depth generator를 사용하여 깊이 영상에 대해서 640x480으로 업샘플링한 영상을 사용했다. 그리고 결합형 양방향 필터에서 σ_R 의 값은 5, σ_D 의 값은 30으로 설정했다. 여기서는 결합형 양방향 필터를 한번 수행한 것을 기준으로 삼고 각각 CPU와 GPU로 구현하여 결과를 도출했다.

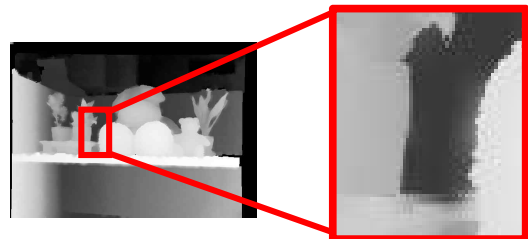


그림 7. CPU를 이용하여 계산된 영상
Fig 7. Depth map computed by using CPU

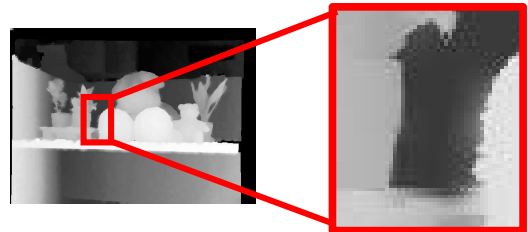


그림 8. GPU를 이용하여 계산된 영상
Fig 8. Depth map computed by using GPU

각 영상의 오른쪽에 영상의 일부를 확대하여 나타냈다. 확대된 두 영상을 비교하여 보았을 때 전혀 화질의 차이가 존재하지 않는 것을 확인할 수 있다. 두 결과가 완전히 같기 때문에 차이 영상을 구해보면 영상 전체가 완전히 검은색인 영상을 얻게 된다. 하지만 수행 시간의 관점에서 비교를 해보았을 때 GPU 프로그래밍을 이용하여 필터를 수행한 것이 CPU 프로그래밍을 이용했을 때보다 현저하게 빠른 것을 확인할 수 있었다. 표 1은 다양한 영상 샘플에 대해서 CPU와 GPU를 이용하여 계산한 수행 시간을 나타낸 것이다. 영상은 컴퓨터 비전 분야에서 잘 알려진 middlebury 사이트에서 제공하는 영상을 사용했고 그 중에 venus,

표 1. CPU와 GPU기반의 알고리즘 수행시간 비교
 Table 1. Algorithm running time comparison between CPU and GPU

Sample (size)	CPU(ms)	GPU(ms)	times
Venus (434x383)	103148	19.93	5175.51
	100727	19.99	5038.86
	102403	20.17	5076.99
	102309	20.09	5092.53
	102078	20.07	5086.09
Tsukuba (384x288)	68343	13.46	5077.48
	68778	13.33	5159.64
	67574	13.35	5061.72
	67981	13.4	5073.20
	66483	13.31	4994.96
Cones (450x375)	115536	21.46	5383.78
	118697	20.78	5712.07
	119631	20.98	5702.14
	115784	20.66	5604.25
	118248	20.99	5633.54

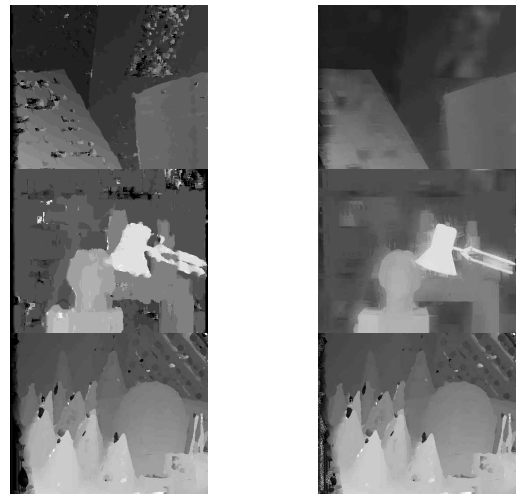
tsukuba, cones의 총 3개의 영상 시퀀스에 대해서 실험을 수행했다. 각 영상에서 실험 결과를 통해 GPU를 사용한 경우 약 5000배 이상의 수행 시간 단축 효과가 있음을 확인했다. 결합형 양방향 필터에서 σ_R 의 값은 15로 설정하고 σ_D 의 값은 venus와 tsukuba 영상의 경우에는 10, cones 영상의 경우에는 0.6으로 설정하고 실험했다.

다음으로 화질향상에 대한 평가를 위해 bad pixel rate(BPR)를 측정하는 실험을 수행했다. 스테레오 매칭 알고리즘인 SSD 방법을 통해 얻은 깊이 영상과 여기에 계

표 2. Bad pixel rate 비교
 Table 2. Comparison of bad pixel rate

Sample	Bad pixel rate(%)	
	Before JBF	After JBF
Venus	12.02	7.96
Tsukuba	8.9	6.43
Cones	16.39	16.32

층적 결합형 양방향 필터를 수행시켜 나온 깊이 영상을 ground truth 영상과 비교하여 bad pixel rate를 측정해 보았다. 그림 9와 표 2는 그 결과를 보여주고 있다. Venus와 tsukuba 영상에서는 확연히 BPR이 줄어드는 것을 확인할 수 있다. 하지만 cones 영상에 대해서는 원래의 영상과 비슷한 수치를 보인다.



(a) 계층적 결합형 양방향 필터 적용 전 (b) 계층적 결합형 양방향 필터 적용 후
 (a) Before hierarchical JBF (b) After hierarchical JBF

그림 9. 스테레오 매칭 결과에 계층적 결합형 양방향 필터 적용
 Fig 9. Hierarchical JBF to result of stereo matching

V. 결론

본 논문에서는 CUDA를 이용한 결합형 양방향 필터를 이용하여 실시간으로 보정된 깊이 영상을 획득하는 방법에 대해서 살펴보았다. 또한 깊이 영상의 화질을 더욱 더 향상시키기 위해서 계층별로 영상의 해상도를 조절하여 색상 영상과 깊이 영상을 구한 다음, 하위 계층에서 상위 계층으

로 올라가면서 결합형 양방향 필터를 수행하여 최종적으로 모든 부분의 홀 영역이 채워진 깊이영상을 구하는 계층적 접근 방법도 사용했다. 원본 깊이 영상과 제안한 방법의 결과로 나온 깊이 영상의 화질을 비교하여 보았을 때 제안한 방법이 깊이 영상의 주관적 화질을 향상시킨 것을 확인했다. 또한, CPU로 수행된 방법과 GPU로 수행된 방법을 비교하여 보았을 때, 화질 면에서는 동일한 화질 향상 수준을 보이지만 수행 시간 면에서는 GPU 프로그래밍을 이용하여 결합형 양방향 필터를 수행한 것이 CPU 프로그래밍을 이용했을 때보다 현저하게 빠른 것을 확인했다. 실험 결과는 초당 55 장면을 처리할 수 있는 속도를 나타냈는데 이는 카메라로부터 입력되는 깊이 영상을 실시간으로 처리하기에 충분한 속도임을 확인했다. 화질향상에 대한 객관적인 평가를 위해서 스테레오 매칭의 결과로 얻은 깊이 영상과 여기에 계층적 결합형 양방향 필터를 적용하여 얻은 깊이 영상에 대해서 bad pixel rate를 비교하여 본 논문에서 제안한 방법이 더 낮은 bad pixel rate를 나타냄을 확인했다. 이를 통해 제안한 방법이 물체의 경계를 잘 맞추고 있어 3차원 물체를 더 잘 표현하고 있는 것을 확인했다.

참 고 문 헌 (References)

- [1] C. Fehn, "Depth-image-based rendering (DIBR), Compression and Transmission for a New Approach on 3-D TV," Proc. of SPIE Conference Stereoscopic Displays and Virtual Reality Systems, vol. 5291, pp. 93-104, Jan. 2004.
- [2] K. Jung, Y. Park, J. Kim, H. Lee, K. Yon, N. Hur, and J. Kim, "2D/3D Mixed Service in T-DMB System using Depth Image Based Rendering," Proc. of International Conference on Advanced Communication Technology, vol. 3, pp. 1868-1871, Feb. 2008.
- [3] J. Kopf, M. F. Cohen, D. Lischinski, and M. Uyttendaele, "Joint Bilateral Upsampling," ACM Transactions on Graphics, vol. 26, no. 3, pp. 1-5, July 2007.
- [4] NVIDIA, "CUDA C Programming Guide," document PG-02829-001_v5.0, Oct. 2012.
- [5] S. Hwang, "Image Processing Programming by Visual C++," Hanbit Media Inc, pp. 366-370, 2007.
- [6] C. Richardt, C. Stoll, N. Dodgson, H. Seidel and C. Theobalt, "Coherent Spatiotemporal Filtering, Upsampling and Rendering of RGBZ Videos," Computer Graphics Forum, vol 31, pp. 247-256, May. 2012.
- [7] D. Shin, S. Lee and Y. Ho, "Real-time Depth Map Refinement using Joint Bilateral Filter", Fall Conference Proceeding of The Korean Society of Broadcast Engineers, Dec. 2013.
- [8] J. Sanders and E. Kandrot, "CUDA by example", Addison-Wesley Professional, p. 116, 2010.
- [9] <http://vision.middlebury.edu/stereo/data/>

저 자 소 개



신 동 원

- 2013년 금오공과대학교 컴퓨터공학과 졸업(학사)
- 2013년~현재 : 광주과학기술원 정보통신공학부 석사과정
- 주관심분야 : 3D 영상처리, 실감방송



호 요 성

- 1981년 서울대학교 공과대학 전자공학과 졸업(학사)
- 1983년 서울대학교 대학원 전자공학과 졸업(석사)
- 1989년 Univ. of California, Santa Barbara, Dept. of Electrical and Computer Engineering.(박사)
- 1983년~1995년 한국전자통신연구소 선임연구원
- 1990년~1993년 미국 Philips 연구소, Senior Research Member
- 1995년~현재 광주과학기술원 정보통신공학과 교수
- 주관심분야 : 디지털 신호처리, 영상 신호 처리 및 압축, 디지털 TV와 고선명 TV, 멀티미디어 시스템, MPEG 표준, 3차원 TV, 실감방송