

그래픽 하드웨어를 이용한 HD 영상의 실시간 스테레오 정합

오주현, *손광훈
 KBS 기술연구소, *연세대학교 전기전자공학과
 jhoh@kbs.co.kr, *khsohn@yonsei.ac.kr

Real-Time Stereo Matching of HD Video Using Graphics Hardware

Juhyun Oh and *Kwanghoon Sohn
 KBS Technical Research Institute
 *School of Electrical and Electronic Engineering, Yonsei University

요 약

최근 3DTV 의 급격한 활성화로 스테레오 영상 콘텐츠 제작이 크게 증가하고 있다. 스테레오 영상은 일반 2D 영상과 달리 깊이(depth)가 존재하므로 자막과 같은 그래픽의 삽입에서 그 깊이를 반드시 고려해야 한다. 또한 시각피로를 줄이기 위해 스테레오 촬영 시 영상의 변이맵(disparity map)을 실시간 관찰할 필요성도 요구되고 있다. 본 논문에서는 최신의 그래픽 하드웨어를 이용하여 듀얼스트림 HD 영상을 실시간으로 스테레오 정합하는 방법을 제안한다.

1. 서론

최근 3DTV 의 급격한 활성화로 스테레오 영상 콘텐츠 제작이 크게 증가하고 있다. KBS 기술연구소에서는 스테레오 영상에서 여러 변수에 의한 시각 피로도를 알아보기 위하여 3D 레퍼런스 콘텐츠를 제작한 바 있다 [1]. 그 중 그림 1 과 같은 문자그래픽 영상에서 고려해야 할 사항은 스테레오 영상에는 일반 2D 영상에서와 달리 깊이(depth) 개념이 존재하므로 그래픽의 삽입에 있어서 장면의 깊이(scene depth)를 고려할 필요가 있다는 것이다.



그림 1. 3DTV 문자그래픽의 예

일반적으로 인간의 깊이 지각 (depth cue) 중에서는 가림(occlusion)에 의한 지각이 스테레오에 의한 깊이

지각보다 크게 영향을 미치는 것으로 알려져 있다 [2]. 그런데 자막의 깊이를 고려하지 않고 2D 방송에서와 같이 단순히 오버레이할 경우, 이 두 가지 지각이 상충하여 시각적 피로감을 크게 증가시킬 수 있다. 가림 지각에 있어서 자막은 보통 영상 위에 오버레이하게 되므로 자막이 영상보다 앞에 있는 것으로 느끼게 된다. 이 때 자막의 변이(disparity)가 영상 변이보다 더 작아서 스테레오 지각에서는 자막이 영상보다 뒤에 있는 것으로 느끼게 되는 경우가 대표적인 사례이다.



그림 2. 소니의 Z Depth 콘솔

이러한 문제를 극복하는 가장 간단한 방법은 영상을 모두 음의 변이를 가지게 촬영하고, 자막을 변이가 0 인 화면 깊이(screen depth)에 놓는 것이다. 그러나 일반적인 문자발생기(character generator)의 경우 문자그래픽이 뒤의 배경 영상과 실시간으로 합성되는 경우가 많은데, 그 배경영상은 직접 제작하지 않은 외부 비디오 피드인 경우가 대부분이다. 따라서 전문가가 스테레오 영상을 계속

모니터링하면서 그림 2 와 같은 특수 장비 [3]를 사용하여 실시간으로 그래픽의 깊이를 조절하여야 하지만, 스테레오 영상 시청의 피로도를 감안할 때 이러한 작업을 사람이 지속적으로 하는 것은 불가능하므로 반드시 자동화할 필요가 있다.

그래픽의 삽입 문제 이외에도 시각피로를 줄이기 위해 스테레오 촬영 시 영상의 변이맵(disparity map)을 실시간 관찰할 필요성도 요구되고 있다. 그림 3 의 Cel-Scope 3D [4]는 depth budget 을 실시간으로 모니터링할 수 있게 해준다.

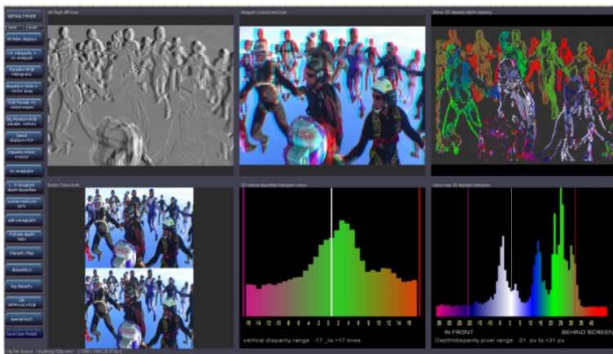


그림 3. Cel-Scope 3D

이러한 기능은 소니의 3D Box [5]에서도 일부 제공하고 있다. 그러나 이러한 상용 제품의 한계는 영상의 전체가 아닌 일부 영역에서만 깊이 정보를 제공한다는 것이다.

이러한 요구사항을 만족하기 위해서는 HD(high definition) 영상의 실시간 스테레오 정합을 구현할 필요가 있다. 스테레오 정합은 과거 수십 년간 컴퓨터비전 및 영상처리 분야에서 가장 활발하게 연구된 주제이다. 그러나 대부분의 기존 연구는 VGA 또는 SD(standard definition)급 해상도의 영상을 그 대상으로 하고 있으며, 완전한 HD 해상도(1920×1080)의 스테레오 영상에서 변이맵을 실시간으로 추출한 연구는 찾아보기 힘들다.

본 논문에서는 최신의 그래픽 하드웨어를 이용하여 듀얼스트림(dual stream) HD 영상을 실시간으로 스테레오 정합하는 방법을 제안한다. 본 논문의 구성은 다음과 같다. 2 절에서는 전체 시스템과 영상 입출력을 살펴보고, 3 절에서는 실시간 스테레오 정합을 위한 GPU(graphic processing unit) 기반 알고리즘을 소개한다. 4 절에서는 제안한 기법의 성능을 실험을 통해서 확인한다. 마지막으로 5 절에서 결론을 맺는다.

2. 시스템 구성

본 논문은 기존의 side-by-side, line-by-line 등과 같은 frame-compatible 한 제작방식이 아닌, 본격적인 듀얼스트림 HD SDI(serial digital interface) 스테레오 제작 시스템을 그 대상으로 한다. 아직 상용화된 듀얼스트림 스테레오 제작시스템이 많지 않은 관계로 본 논문의 실험은 파나소닉에서 출시한 3D 카메라 AG-3DA1 [6]만을 사용한다. 이 카메라는 그림 4 에서 볼 수 있듯이 동기화된 좌-우측 HD

SDI 영상 출력 신호를 제공한다.



(a)



(b)

(c)

그림 4. (a) 실험에 사용된 파나소닉 3D 카메라, (b) 촬영된 우영상, (c) 좌영상

3D 카메라의 출력을 PC 에서 처리하기 위해, Nvidia 의 Quadro SDI capture card [7]를 사용한다. 이 카드는 그림 5 에서 볼 수 있듯이 4 개의 HD-SDI 입력을 동시에 입력 받을 수 있으며, 입력된 비디오가 바로 Quadro GPU 의 텍스처 메모리로 전송된다는 점이 특징이다. 본 논문에서는 입력 1 과 2 를 3D 카메라의 좌-우 영상 입력으로 사용한다.

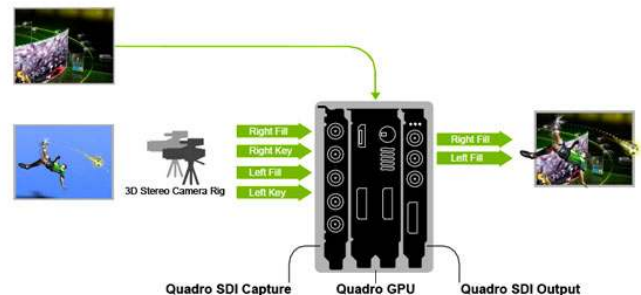


그림 5. Nvidia Quadro 디지털 비디오 파이프라인

3. GPU 기반 스테레오 정합

과거 오프라인 렌더링으로만 구현 가능했던 여러 가지

그래픽 효과들은 셰이딩 언어(shading language)의 등장으로 인해 GPU 상에서 실시간으로 구현할 수 있게 되었다. 대표적인 것이 Nvidia의 Cg(C for graphics)이다 [8]. 그림 6에서처럼, 이러한 프로그래밍 가능한 vertex processor와 fragment processor를 이용하면 GPU를 단순하고 고정적인 렌더링 외에 여러 가지 다양한 용도로 사용할 수 있게 된다.

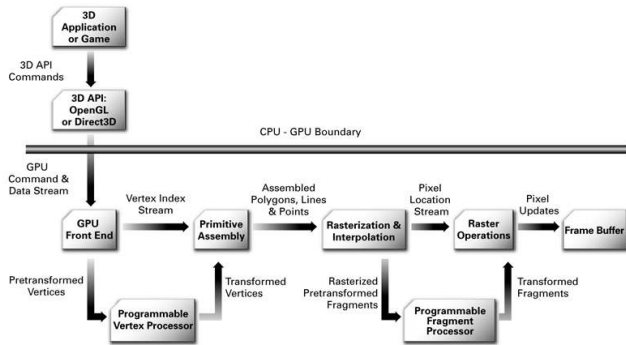


그림 6. 프로그래밍 가능한 그래픽 파이프라인

일반적인 GPGPU(General-Purpose GPU) 활용에 있어서 가장 큰 문제점은 그림 6의 'CPU-GPU Boundary'에서 볼 수 있듯이 CPU에서 GPU로 처리할 데이터를 보내고, GPU에서 처리한 데이터를 다시 CPU로 가져오는데 많은 시간적 오버헤드가 발생한다는 점이다. 그러나 본 논문에서 사용하는 Nvidia의 SDI Capture 카드는 비디오 데이터를 직접 텍스처 메모리에 기록해주기 때문에 이와 같은 오버헤드가 발생하지 않는 장점이 있다.

본 논문에서는 OpenGL과 Cg를 이용하여 스테레오 정합을 구현한다. 기본적인 방법은 [9,10]과 같이 vertex program을 이용하여 좌-우 영상을 변이만큼 이동(shift)하고, fragment program을 이용하여 비용(cost)의 계산과 축적(aggregation)을 수행한다. 평가할 N 번의 변이 값에 대하여 N 번의 렌더링을 통해 스테레오 정합이 이루어진다.

비용 축적에 있어서 본 논문에서는 (1)의 가중(weighted) SSD(sum of squared differences) 방법을 사용한다.

$$\sum_{s,t} w(s,t) \cdot \sqrt{(r_L - r_R)^2 + (g_L - g_R)^2 + (b_L - b_R)^2}$$

$$\text{where } w(s,t) = \frac{(1-l^2)}{(1+l^2)} \quad (1)$$

$$\text{and } l(s,t) = \sqrt{s^2 + t^2}$$

식 (1)에서 s, t 는 변이를 계산할 픽셀로부터 윈도우 내부의 다른 'support' 픽셀들까지의 offset으로서, 0~1 사이의 텍스처 좌표 단위이다. WIN은 SSD를 위한 윈도우의 크기를 나타낸다. r, g, b 는 픽셀의 컬러 값을 나타낸다. 그림 7은 본 논문에서 사용된, 텍스처 좌표 offset에 대한 가중치 그래프이다. 즉 윈도우 내부에서 현재 픽셀에 가까울수록 높은 가중치를 갖는다.

이와 같이 축적된 가중 SSD 값은 깊이 버퍼에, 해당 변이 값은 컬러 버퍼에 각각 저장된다. 최종적인 변이 선택(disparity selection)은 깊이 버퍼의 깊이 비교를 이용한 WTA(winner-take-all) 방법으로 구현한다. 즉 축적 비용이

가장 낮은 변이가 OpenGL의 깊이를 비교를 통해 컬러 버퍼에 자동으로 보이게 되는 구조이다.

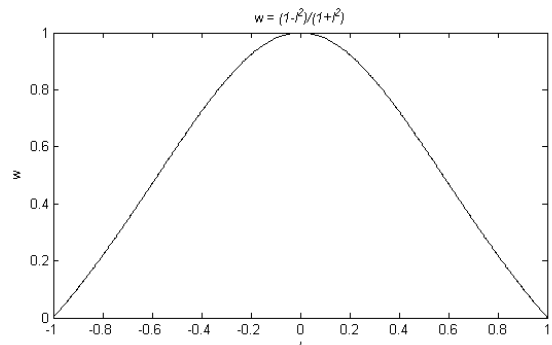


그림 7. SSD 가중치

4. 실험 결과

그림 8과 그림 9는 본 논문에서 제안한 방법으로 1920x1080 해상도의 HD 영상의 스테레오 정합을 실시간 구현한 결과로, 실험자가 의자에 앉아서 왼팔을 들고 있는 모습이다.



그림 8. 스테레오 정합 결과 (실험 1)

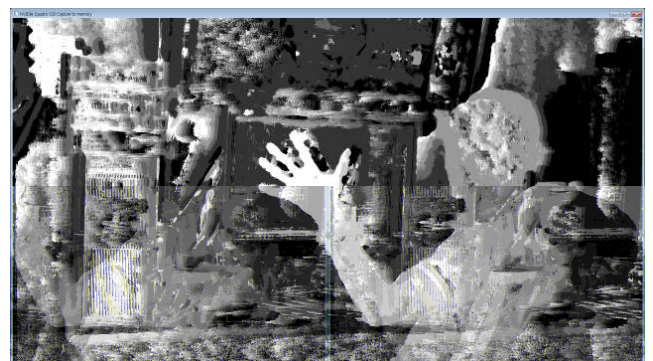


그림 9. 스테레오 정합 결과 (실험 2)

본 논문에서는 스테레오 카메라의 두 광축이 평행한 것으로 간주하여 별도의 스테레오 교정(rectification) 과정을

실시하지는 않았다. 다만 두 영상 사이에 전반적으로 1.5~2 픽셀 정도의 수직 오차가 관측되었으므로 이를 vertex program 에서 보상하였다.

실험에는 HP 사의 Z800 워크스테이션을 사용하였으며, 실험을 위한 알고리즘 변수 설정은 표 1 과 같이 하였다. SSD 스텝은 실시간 구현을 위하여 윈도우 안에서 건너 뛰는 픽셀 수를 말한다. 계산 속도는 대체적으로 15~25 fps 의 성능을 보였으며, 프로그램 실행 중 CPU 사용률은 15% 이내로 유지되었다.

표 1. 실험 변수 설정

	실험 1	실험 2
창 크기(픽셀)	15	25
SSD 스텝(픽셀)	5	2
변이 범위	[0,50]	[0,60]
변이 스텝	2	10

5. 결론

본 논문에서는 3D 콘텐츠 제작을 위한 실시간 스테레오 정합 알고리즘과 시스템을 제안하였다. 제안된 시스템은 듀얼 HD-SDI 비디오를 입력 받아 실시간으로 화면에 변이맵을 출력하는데 성공하였다. 본 논문은 기존 GPGPU 시스템에서 나타나는 CPU-GPU 메모리 전송 병목을 해소함으로써 더욱 효율적인 시스템을 구현하였다.

그러나 실험 결과에서 알 수 있듯이 현재는 단순한 SSD 와 WTA 기반 알고리즘을 적용함으로써 비실시간 방법들에 비해 변이맵의 품질이 좋지 못한 한계가 있다. 향후 좀 더 진보된 알고리즘을 적용하기 위해 CUDA[12] 등을 도입할 계획이다.

참고 문헌

- [1] 권재광, “3DTV 레퍼런스 콘텐츠 개발,” KBS 기술연구소 연구보고서 pp.158-164, 2010.
- [2] J. E. Cutting and P. M. Vishton, “Perceiving layout and knowing distances: The integration, relative potency, and contextual use of different information about depth,” *Handbook of perception perception and cognition, Vol 5: Perception of space and motion*, pp.69-117, Academic Press, 1995.
- [3] “Z Depth Overview,” <http://www.sonycreativesoftware.com/zdepth>, 2011.
- [4] “cel-scope3D,” <http://www.cel-soft.com/celscope3d/>, 2011.
- [5] “Sony 3D Box,” <http://pro.sony.com/bbsc/ssr/product-MPE200/>, 2011.
- [6] “Panasonic AG-3DA1,”

<http://pro-av.panasonic.net/en/3d/ag-3da1/index.html>, 2011.

- [7] “NVIDIA Quadro Digital Video Pipeline,” http://www.nvidia.com/object/quadro_dvp.html, 2011.
- [8] R. Fernando 외, “Cg 로 배우는 셰이더 프로그래밍,” 정보문화사, 2004.
- [9] R. Yang and M. Pollefeys, “Multi-resolution real-time stereo on commodity graphics hardware,” *Computer Vision and Pattern Recognition, Vol.1*, 2003.
- [10] R. Yang and M. Pollefeys, “Improved real-time stereo on commodity graphics hardware,” *Computer Vision and Pattern Recognition Workshop, Vol.3*, 2004.
- [11] “What is CUDA?,” http://www.nvidia.com/object/what_is_cuda_new.html, 2011.