모바일 GPU 기반의 고속 3차원 공간 정보 취득 기술

□ 정태현, 박준형, 박인규 / 인하대학교

요약

복잡한 알고리즘을 요구하는 3차원 공간 정보 취득 기술은 대부분 고성능의 하드웨어를 필요로 한다. 그러나 최근 스마트폰과 같은 모바일 플랫폼의 성능이 급격히 발전하면서 기존 알고리즘을 가속화해 온 디바이스로 이식하는 연구가 증가하고 있다. 이러한 추세에 따라 본 기고문은 플랫폼 제한 없는 GPU 병렬처리 프레임워크 OpenCL을 활용한 3차원 공간 정보 취득 기술의 가속화 방법을 소개하고자 한다. 본 고의 구성은 다음과 같다. 먼저 모바일 GPU 환경에서의 OpenCL 최적화 방법을 살펴본다. 이후 고전적인 기하학 기반의 스테레오 정합 알고리즘을 가속화한 방법을 소개한다. 마지막으로는 심층 신경망 네트워크와 가속화된 고전적 스테레오 알고리즘을 결합한 온 디바이스 친화적인 융합 알고리즘을 소개한다.

Ⅰ. 서 론

3차원 공간 정보는 스마트폰과 같은 온 디바이스 환경에서 증강현실, 얼굴 인식 등으로 광범위하게 활용된다. 이에 따라 최근 스마트폰 제조사는 공간 정보 취득이 가능한 RGB-D 카메라, LiDAR 센서, 다중 카메라 등의 특수한 장치가 내장된 기기를 선보이고 있다. 이중 다중 카메라는 특수한 센서 없이 영상만을 이용해 3차원 공간 정보를 취득할 수 있다는 점에서 큰 이점을 가진다.

기존의 영상 기반 3차원 공간 정보 취득 기술은 복잡한 알고리즘으로 인하여 대부분 고성능의 하드웨어를 필요로 하였다[19]. 그러나 최근 모바일 플랫폼의 성능이 급속도로 발전하면서 기존의 공간 정보 취득 알고리즘을 가속화하여 온 디바이스 환경으로 이식하는 연구가 증가하

※ 이 기고문은 2021년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임(2017-0-00142, 스마트기기를 위한 온 디바이스 지능형 정보처리 가속화 SW플랫폼 기술 개발, 2020-0-01389, 인공지능융합연구센터지원(인하대학교))

고 있다. 이러한 추세에 따라 플랫폼 제한 없이 GPGPU를 사용할 수 있도록 도와주는 OpenCL[20] 프레임워크가 온 디바이스 가속화에 있어 큰 주목을 받고 있다.

본 기고문에서는 OpenCL 기반의 GPU 병렬처리 과정을 통한 3차원 공간 정보 취득 기술의 가속화 방법을 소개하고자 한다. 이를 위하여 2장에서는 모바일 GPU에서의 OpenCL 최적화 기법을 살펴본다. 3장에서는 고전적인 기하학 기반의 스테레오 정합 알고리즘을 가속화여 디바이스 성능에 따라 유동적으로 선택할 수 있는 OpenCL 기반 프레임워크를 소개한다. 그리고 이를 응용한 3차원 복원 기법을 소개한다. 마지막으로 4장에서는 심층 신경망 네트워크와 가속화된 고전적 스테레오 알고리즘을 결합한 온 디바이스에 최적화된 공간 취득 모델을 소개한다.

Ⅱ. 모바일 GPU에서의 OpenCL 가속화 기법

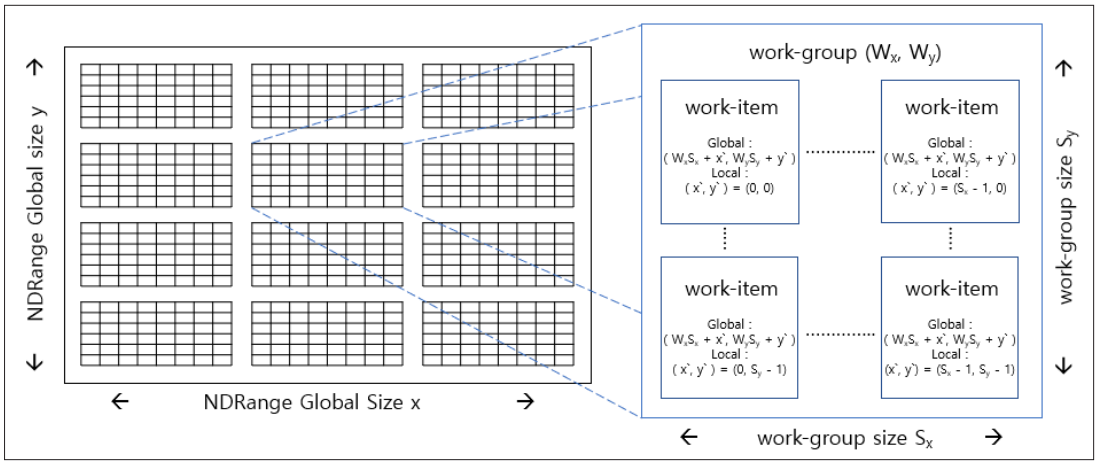
모바일 GPU 최적화 기법은 디바이스 메모리 관점과 호스트 단의 커널 함수 실행 관점으로 분류할 수 있다[15, 27].

• 메모리 구조에 따른 최적화 :

OpenCL의 명령어 중 MAX_WORK_GROUP_SIZE 값은 커널 실행 시 최대로 설정 가능한 워크그룹(work-group) 크기를 나타내며, 처리하는 디바이스의 메모리가 그 값의 배수일 때 성능이 극대화된다. 또한 OpenCL에서 Single Instruction Multiple Data(SIMD)[14] 연산은 각종 자료구조(char, int, half, float)에 대해 N(1, 2, 4, 8, 16)개의 데이터를 동시에 계산할 수 있다. 일반적으로 모바일 GPU의 경우 128bit로 4개의 데이터를 한꺼번에 처리하는데 최적화되어 있다. 이때 CPU 사양이 좋지 않은 모바일 환경에서는 디바이스 메모리 버퍼를 의미 없이 반복적으로 초기화하지 않는 것이 바람직하다.

• 커널 함수 실행 법에 따른 최적화 :

커널 함수는 이벤트 처리기를 이용한 동시 호출과 GPU 연산에 부적합한 조건문의 사용을 줄이는 것으



<그림 1> OpenCL 디바이스 메모리 구조

로 최적화가 가능하며, 모바일 환경에서는 이 두가지를 모두 적용하는 것이 유리하다. 이벤트 처리기를 이용한 동시 호출은 작업 우선순위가 없는 커널 함수에 적용할 수 있으며 순차적 호출에 비해 큰 이득을 얻을 수 있다. 커널 함수 내 조건문 사용을 줄이는 방법은 디바이스 메모리의 전역 오프셋과 전역 NDRange 크기를 조절해 경계선 문제와 관련된 조건문을 제거하거나 loop unrolling 기법을 사용하는 것 등이 있다.

• 모바일 GPU 최적화 기법의 실제 적용 예시 :

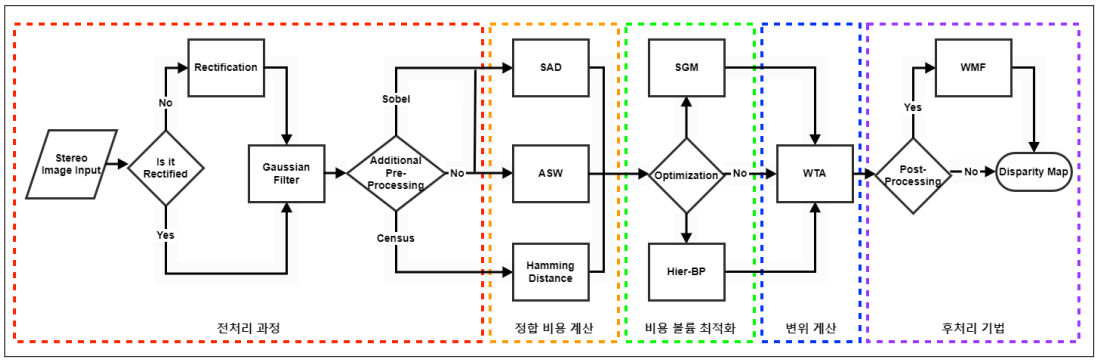
영상 기반의 병렬 알고리즘 구현 시 입력 영상의 너비와 높이를 전역 NDRange 크기로 정하고, 각 화소를 워크아이템으로 지정한다. 필터 적용 시 특정 화소에서 윈도우의 범위가 전역 크기를 넘어가는 경계 문제가 발생한다면 전역 오프셋과 전역 NDRange 크기를 조절하여 경계 조건의 조건문을 제거함으로 병렬 알고리즘의 성능 저하를 막을 수 있다. 만약 전역 NDRange 크기가 지정한 워크 그룹의 크기로 나누어 떨어지지 않을 때는 OpenCL 이벤트를 활용하여 명령어 큐에 대한 전역 크기를 워크그룹의 배수 부분과 나머지 부분으로 나눠 Task-parallel 모델을 구축해 알고리즘의 성능을 증가시킬 수 있다.

Ⅲ. 기하학 기반 공간 정보 취득 기술

다중 시점 영상이 주어지면 영상 간의 기하학적인 변위(disparity)를 계산하여 3차원 공간 정보 취득이 가능하다. 본 기고문에서는 다중 시점 영상을 수평 정렬된 좌/우 영상을 사용하는 양안 스테레오와 시점이 정렬되지 않은 다수의 영상을 사용하는 다중 뷰 스테레오로 나누어 소개하고자 한다. 먼저 양안 스테레오 정합 가속화 기법은 디바이스 성능에 따라 유동적으로 선택할 수 있는 OpenCL 기반 공간 정보 취득 프레임워크를 통해 소개한다[10]. 이후 다중 뷰 스테레오 정합 가속화 기법은 아래 프레임워크를 응용한 표면 요소(surfel) 기반의 3차원 형상 복원 방법을 통해 소개한다[27].

1. 양안 스테레오 정합 기법

양안 스테레오 정합 기법은 수평 정렬된 좌/우 영상이 입력으로 주어질 때, 화소의 횡적 변위를 계산하여 3차원 공간 정보를 취득하는 알고리즘이다[19]. Ivan은 (1)전처리 과정, (2)정합 비용 계산, (3)정합 비용 최적화, (4)변위 계산, (5)후처리 기법 다섯 단계로 나눠 디바이스 성능에 따라 유동적으로 선택할 수 있는 선택적



<그림 2> 모바일 GPU에서의 양안 스테레오 정합 프레임워크

프레임워크(〈그림 2〉)를 제안했다[10].

1) 전처리 과정

전처리 과정은 정합 비용 계산의 효과 증가를 위해 실행된다. 1차적으로 입력 영상을 카메라 인자를 통해 렌즈 왜곡을 보정한다. 이후 영상의 잡음 제거를 위해 가우시안 필터를 적용한다. 1차 전처리가 적용된 영상은 Sobel 필터 또는 Census 필터[25]를 이용하여 추가적인 전처리가 가능하다. 두 필터는 스테레오 정합 시 텍스처가 없는 영역의 변위 오류 문제를 해결하는데 매우 용이하다. 그러나 Census 필터는 GPU 병렬 처리에는 친화적이지 않으므로 본 프레임워크에서는 Sobel 필터를 이용한다. 전처리 과정에 사용된 카메라 인자, 가우시안 커널 등의 정적 변수는 모두 정적 메모리(static memory)에 저장하여 GPU 실행 시 메모리 대역폭(memory bandwidth)을 줄이고 데이터 접근 시간을 단축한다.

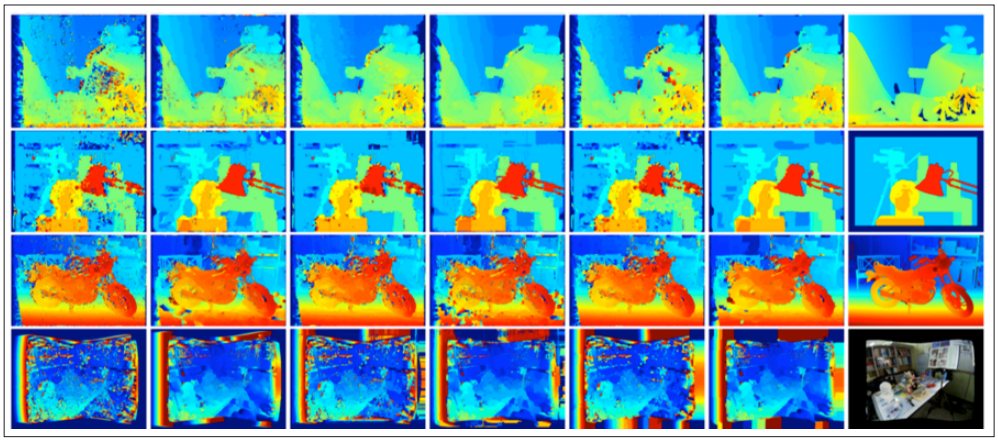
2) 정합 비용 계산

양안 스테레오에서 정합 비용 계산은 수평으로 한 화소씩 이동하며 좌/우 영상의 화소의 유사도를 계산하고, 동일 과정을 영상 전반에 걸쳐 수행해 영상 전체 해상도 크기의 비용 볼륨(cost volume)을 구축한다. 정합 비용 계산에는 sum of absolute distance(SAD)와 adaptive support weight(ASW)[24] 두 가지 알고리즘을 사용한다. SAD는 화소의 유사도를 단순히 비교하고, ASW는 좌/우 화소의 거리 차이와 밝기 차이에 따라 적응적으로 패널티를 주어 비용 볼륨을 구축한다는 차이점이 있다. 두 알고리즘은 윈도우의 폭과 높이에 따라 4개의 주 반복문을 영상의 너비와 높이에 각각 적용하여 실행한다. 본 프레임워크는 GPU에서 상위 2개의 반복문을 병렬적으로 수행하도록 했으며, fabs, hypot, abs_dif와 같은 OpenCL native 수학 함수를 사용하여 계산 시간을 단축했다.

3) 비용 볼륨 최적화

비용 볼륨 최적화 과정은 취득한 비용 볼륨의 신뢰도를 높이는 과정이다. [10]은 3가지 방법의 정합 비용 최적화 기법을 제안하며 이는 지역적 기법(local), 반전역적 기법(semi-global), 전역적 기법(global)으로 구성된다. 지역적 기법에서는 정합 비용을 개선하기 위한 부가적인 최적화 과정을 적용하지 않는다.

반전역적 최적화 기법으로는 경로 최적화에 적합한 동



<그림 3> 모바일 GPU에서의 양안 스테레오 정합 기술: 각 조합의 정성적 성능 평가

<표 1> 모바일 GPU에서의 양안 스테레오 정합 기술: 각 조합의 정량적 성능 평가

Metric & Algorithm		Books	Cone	Venus	Tsukuba	Dolls	Aloe	Average
MSE	SAD	37.11	30.64	16.51	0.72	36.74	15.58	12.4
	ASW	33.01	19.74	9.48	0.31	20.21	7.21	8.31
	SAD+SGM	**6.66**	8.18	4.62	0.61	32.72	13.62	6.1
	ASW+SGM	6.7	7.81	3.01	0.11	20.4	6.27	4.36
	SAD+Hier- BP	9.37	6.63	4.24	1.41	6.61	7.28	5.92
	ASW+Hier- BP	7.44	**4.91**	**2.92**	**0.07**	**5.57**	**3.76**	**4.11**
BP%	SAD	18.86	15.5	12.68	0.47	17.38	9.53	22.88
	ASW	16.8	11.28	5.45	0.24	11.5	4.63	14.99
	SAD+SGM	7.76	6.77	5.68	0.23	7.78	8.43	11.06
	ASW+SGM	7.98	**5.29**	**2.84**	0.079	6.14	**3.86**	7.36
	SAD+Hier- BP	7.98	8.1	4.03	0.38	**5.17**	10.8	6.07
	ASW+Hier- BP	**6.68**	6.96	3.01	**0.014**	5.25	5.5	**4.56**

<표 2> 다양한 GPU에서의 양안 스테레오 정합 수행 시간(왼쪽)과 디바이스별 CPU 대비 GPU 가속화 배율(오른쪽)

Algorithm	Processing Time[sec]			Speed Up		
	GTX1080Ti	Mali-G71 MP20	Mali T628 MP4	GTX 1080 Ti	Mali-G71 MP20	Mali T628 MP4
SAD	0.007	0.03	0.37	90.0x	108.5x	24.18x
ASW	0.079	0.5	19.72	132.03x	113.74x	52.93x
SGM	0.25	4.54	13.11	19.05x	5.08x	2.77x
Hier-BP	0.54	5.01	19.4	4.55x	3.79x	2.45x

적 프로그래밍 기반의 semi-global matching(SGM)[8] 알고리즘을 적용한다. SGM은 1차원 비용 볼륨에서 잘못된 정합 비용 계산에 페널티를 부여한다. SGM의 주요 원리는 비용 볼륨에 다양한 방향의 1차원 변위 최적화를 수행하는 것이다. 또한 SGM 기법은 방향 개수에 따라 처리 속도와 정확도 사이에서 균형 잡힌 절충점을 제공한다.

전역적 최적화 기법으로는 hierarchical belief propagation (Hier-BP)[22] 알고리즘을 적용한다. Hier-BP는 피라미드형으로 구축된 영상 층을 반복적으로 최적화하여 전역적으로 비용 볼륨을 최적화한다. 이를 사용한다면 잡음이 많고 텍스처가 없는 영역의 부정확한 정합 비용을 최적화할 수 있다. 전역적 기법은 비용 볼륨 전체에서 최적화를 진행하므로 상당한 비용과 복잡도를 요구해 모바일 GPU에서는 실시간 정합 비용을 계산하기는 어렵다. 하지만 타겟 GPU가 on-chip 공유 메모리를 사용할 수 있다면 효율적인 병렬처리가 가능하다.

4) 변위 계산

변위 계산은 비용 볼륨에서 깊이 영상(disparity map)을 얻는 과정이다. winner-takes-all(WTA) 알고리즘을 사용하며 GPU를 사용 시 모든 화소에 대한 최적 깊이를 병렬적으로 찾아 가속화를 할 수 있다.

5) 후처리 기법

변위 계산으로 구한 깊이 영상에 대해 후처리 과정으로 weighted median filter (WMF)[12]을 선택할 수 있다. WMF 사용 시 윤곽선 검출, 잡음 제거가 가능하다. 그러나 이미 반전역적, 전역적으로 비용 볼륨을 최적화했다면 해당 과정을 생략해 처리 시간을 단축할 수 있다.

실험 및 동작 검증 :

본 실험에서는 Samsung Galaxy Note 8(Mali-G71 MP20), Odroid XU4(Mali-T628 MP4) 그리고 PC (NVIDIA GTX 1080 Ti) 3종 플랫폼에서 GPU 프레임워크의 실행 결과를 비교한다. 제안하는 프레임워크를 평가하기 위해 참값을 포함한 Middlebury 스테레오 데이터셋[5, 16-18]을 활용하며, 실험 결과의 정량적 지표로는 mean square error(MSE) 와 bad pixel percentage(BP%)를 사용한다.

〈그림 3〉과 〈표 1〉은 각 알고리즘의 정성적, 정량적 평가의 결과를 나타낸다. SAD는 변위의 불연속적 구간에 취약하지만 ASW 알고리즘은 윤곽선 보존에 뛰어나며 텍스처가 없는 연속 구간 처리에 뛰어나다. 반전역적 기법 SGM과 전역적 기법 Hier-BP는 지역적 기법에 비해 잡음과 윤곽선 처리에 뛰어난 것을 확인할 수 있다. ASW와 Hier-BP의 조합을 통해 가장 정확한 깊이 영상을 추출할 수 있으며 이는 정량 평가로 확인할 수 있다.

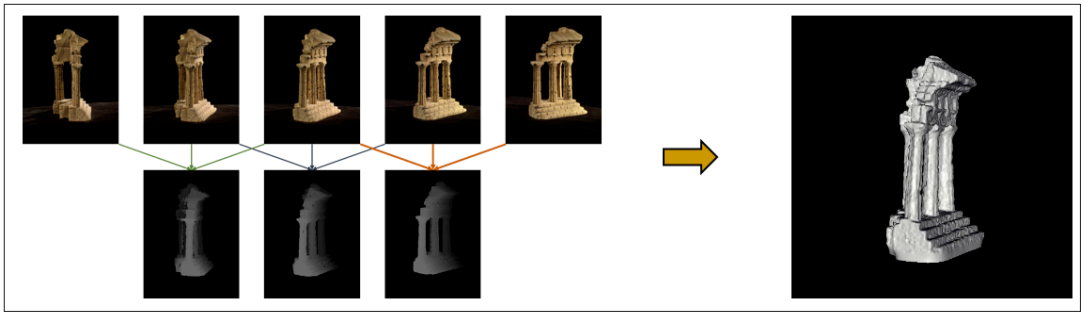
〈표2〉는 각 디바이스 별 알고리즘의 GPU 처리 속도, CPU 대비 GPU의 가속화 정도를 나타낸다. GPU 프레임워크 내에서는 SAD와 지역적 기법의 조합만이 모든 플랫폼에서 실시간 수준의 공간 정보 처리가 가능하다. SGM은 속도와 정확도 사이에서 균형 잡힌 결과를 얻는 절충안이 될 수 있으며 Hier-BP는 고사양 GPU에서만 실시간 처리가 가능하지만 프레임워크 중 가장 정확한 깊이 영상을 취득할 수 있다는 장점을 가진다.

2. 다중 뷰 스테레오 정합 기법

다중 뷰 스테레오 정합 기법은 정렬되지 않은 다수의 2차원 영상으로부터 3차원 공간 정보를 취득하는 알고리즘이다. 양안 스테레오 정합 기법의 모든 프레임워크를 공유하지만, 정합 비용 계산 과정에서 차이를 가진다. 참고문헌 [27]에서는 다중뷰 스테레오 정합 기법을 활용한 3차원 형상 복원 알고리즘을 제안하였다(〈그림 4〉).

1) 다중 뷰 스테레오 정합 비용 계산

다중 뷰 스테레오의 정합 비용 계산은 양안 스테레오 기법과 달리 시점 정렬이 되지 않아 횡적인 화소 정합 비교가 불가능하다. 따라서 정합 비용 계산을 위해서는



<그림 4> 모바일 GPU에서의 다중 뷰 스테레오 정합 기술: 3차원 형상 복원

인접한 영상에서 공간 정보를 취득하고자 하는 목표 영상으로 호모그래피 기반의 시점 보정 과정이 선행되어야 한다. 이후 plane sweeping stereo 알고리즘을 사용해 정합 비용을 계산하여 비용 볼륨을 구축한다. Plane sweeping stereo 알고리즘은 목표 영상에 대한 참조 영상을 알고 있는 상황에서 화소별로 독립적으로 계산하기 때문에 병렬처리가 가능하다.

2) 초기 표면 요소 모델 구축

다중 뷰 스테레오 정합 기법으로 구한 깊이 영상은 초기 표면 요소를 구축하는 데 사용된다[6]. 초기 표면 요소로는 깊이 영상의 3차원 볼륨 공간에 있는 모든 3차원 복셀(voxel)을 투영했을 때 2개 이상의 카메라로부터 보여지는 표면 요소만을 사용한다. 이때 각 영상에서 모든 3차원 복셀 투영 과정이 독립적으로 실행되므로, 초기 표면 요소 구축은 병렬처리가 가능하다.

3) 색상 일관성을 이용한 초기 표면 요소 모델 정제

구축한 초기 표면 요소는 각 인접 영상에 대해 색상 일관성(photo-consistency)이 유지되지 않는 아웃라이어(outlier)를 포함한다. 구축된 모든 초기 표면 요소들을 각 인접 영상에 투영하여 표면 요소의 화소 값과의 색상 일관성을 SAD 비용 함수로 계산한다. 이 과정은 지역적 최적화 과정으로 볼 수 있으며, 참조하는 영상의 각 표면 요소에 따라 독립적으로 실행되므로 병렬처리가 가능하다.

실험 및 동작 검증 :

실험에는 PC(NVIDIA GTX1090Ti), RK3399 보드(Mali T860 MP4)에서 다중 뷰 스테레오 응용 알고리즘을 구현한다. 실험 데이터로는 Middlebury 다중 뷰 데이터셋의 TempleRing[5, 19]을 사용한다. 〈표 3〉에서는 알고리즘의 단계별 CPU 실행 시간과 GPU 실행 시간을 비교한다. 두 디바이스에서 모두 Plane sweeping stereo와 초기 표면 요소 정제 단계에서 많은 시간이 소요되는 것을 확인할 수 있다. 이는 OpenCL을 통해 가속화되어 Plane sweeping stereo 단계에서는 PC의 경우 56배, 임베디드 보드의 경우 94배 가속화될 수 있다. 〈그림 4〉는 메쉬 형태로 복원해 3차원 렌더링한 결과이다[27].

Ⅳ. 심층 신경망 기반 공간 정보 취득 기술

최근 심층 신경망의 발전으로 학습 기반 모델을 통

<표 3> 다중 뷰 스테레오 정합 기법의 응용: 디바이스별 알고리즘 실행 시간 비교

Algorithm	PC			RK3399		
	CPU [msec]	GPU [msec]	Speedup Ratio	CPU [msec]	GPU [msec]	Speedup Ratio
Plane sweeping stereo SAD	1,572	28	56.14x	35,442	377	94x
Construct initial surfels	1	1	1.0x	19	6	3.16x
Refine initial surfels	194	6	32.33x	2,267	22	103.05x
sum	1,767	35	50.49x	37,728	405	93.16x

해 3차원 공간 정보를 취득하는 연구가 늘고 있다. 이는 고전적인 기하학 방식의 결과와 비교했을 때 잡음이 거의 없는 정교한 결과를 얻을 수 있다. 하지만 학습 기반 모델은 높은 하드웨어 사양을 요구하여 모바일 디바이스 환경에서 빠르게 실행되기에는 아직 무리가 있다. 이를 해결하기 위해 다양한 딥러닝 경량화 연구가 진행되고 있으며, 본 기고문에서는 OpenCL로 가속화된 기하학 기반의 알고리즘과 심층 신경망 네트워크를 결합한 온 디바이스 친화적인 융합 모델을 소개하고자 한다.

1. 단안 영상으로부터 3차원 깊이 영상 복원

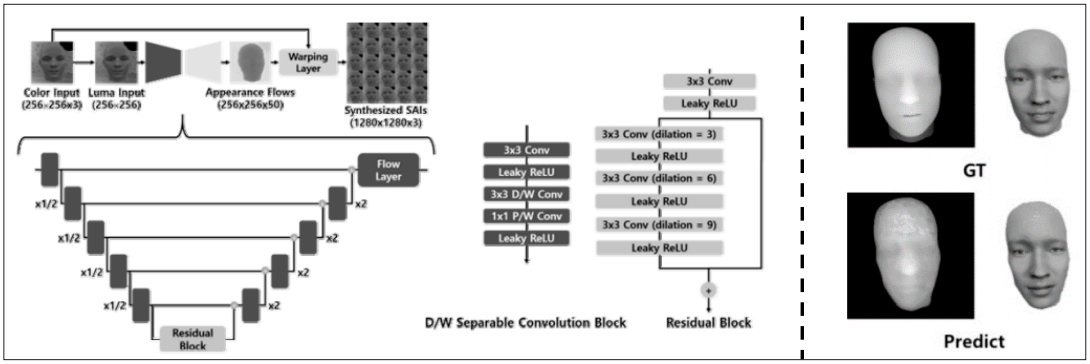
본 파트에서는 온 디바이스 환경에서 심층 신경망을 이용해 단안 영상으로부터 라이트필드 영상을 합성하고, OpenCL로 가속화된 기하학 기반의 라이트필드 정합 기법으로 깊이 영상을 복원하는 연구를 소개한다.

1) 라이트필드 합성 네트워크

참고문헌 [26]에서는 단안의 입력 영상으로부터 라이트필드 영상을 취득하기 위해 중심 영상으로부터 워핑하는 기법을 적용한 라이트필드 다중 뷰 영상 생성 기법을 제안하였다. 워핑을 위해서는 appearance flow가 필요하므로 단안 영상을 입력받아 인코더-디코더 구조를 통해 각 시점별 appearance flow를 한꺼번에 추정하는 네트워크를 설계했다. 추정된 appearance flow는 단안 입력 영상을 워핑하여 5x5개의 다중 뷰로 이루어진 sub-aperture image(SAI) 형태의 라이트필드를 출력한다. 본 모델의 네트워크는 〈그림 5〉와 같다. 모델은 온 디바이스 환경에서 수행할 있도록 모델 사이즈를 작게 유지하면서 깊은 네트워크를 구현하기 위해 depth-wise separable convolution, dilated convolution, residual block을 적용한다. 또한 본 모델은 모바일 기기와의 호환성을 지원하는 Caffe 딥러닝 프레임워크를 사용해 구현하였으며 모바일 기기 이식 시에는 모바일 ARM 프로세서에 최적화된 ARM compute library(ACL)[2]을 사용하는 Caffe-HRT[1] 프레임워크를 사용했다.

2) 라이트필드 응용 효과 구현

라이트필드 영상은 SAI들에 대한 특수한 다중 뷰 스테레오 정합을 통해 깊이 정보 추정이 가능하다. 라이트필드 영상의 전단 변환과 재배치 반복을 통해 라이트필드 정합 비용을 계산하는 함수인 cross angular entropy (CAE)를 통해 비용 볼륨을 생성한다. 비용 볼



<그림 5> 모바일 GPU에서의 단안 영상으로부터 3차원 깊이 영상 복원:라이트필드 합성 네트워크(왼쪽)와 깊이 영상(오른쪽)

<표 4> 라이트필드 합성 처리 시간(왼쪽)과 깊이 추정 처리시간(오른쪽)

Platform	라이트필드 합성[sec]	깊이영상 취득[sec]
PC (CPU)	35,442	-
PC (GPU)	19	0.543
RK3399 (CPU)	2,267	-
RK3399 (GPU)	2,267	13.623

륨으로부터 최종적인 깊이 영상은 WTA를 통해 취득할 수 있다.

실험 및 동작 검증 :

실험은 PC(NVIDIA GTX2080 Ti), 임베디드 보드(Mali-T860 MP4)에서 진행한다. 〈그림 5〉에서 얼굴 이미지에 대한 깊이 복원의 정성적 결과를 확인할 수 있다. 〈표 4〉는 플랫폼별 심층 신경망(라이트필드 합성)의 수행 시간과 OpenCL 기반 라이트필드 깊이 추정 과정의 처리 시간을 나타낸다. 라이트필드 합성은 PC 환경과 달리 모바일 CPU에서도 빠르게 동작한다. 이는 Caffe-HRT는 CPU와 GPU 환경 모두에서 병렬화가 구현되어 있지만, 일반적으로 모바일 GPU는 CPU에 비해 부족한 성능을 보이기 때문에 수행 시간에 차이가 없는 것으로 보인다[26].

2. 다중 영상으로부터 3차원 깊이 영상 복원

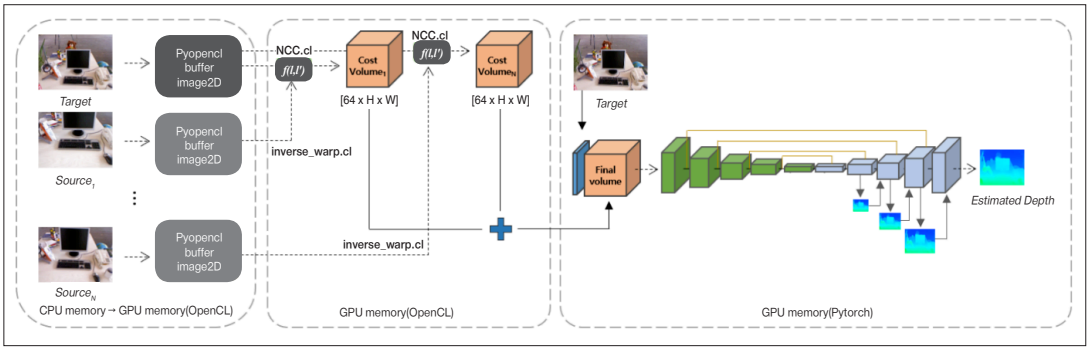
참고문헌 [12]에서는 소수의 입력 영상에 대해 기하학 기반으로 생성한 비용 볼륨과 CNN 기반의 인코더-디코더 구조의 신경망을 융합하여 온 디바이스에 친화적인 깊이 영상 복원 방법을 제안했다. 해당 방법은 비용 볼륨 구축과 심층 신경망을 통한 깊이 추정으로 나뉘며 전체 네트워크는 〈그림 6〉과 같다.

1) 비용 볼륨 구축

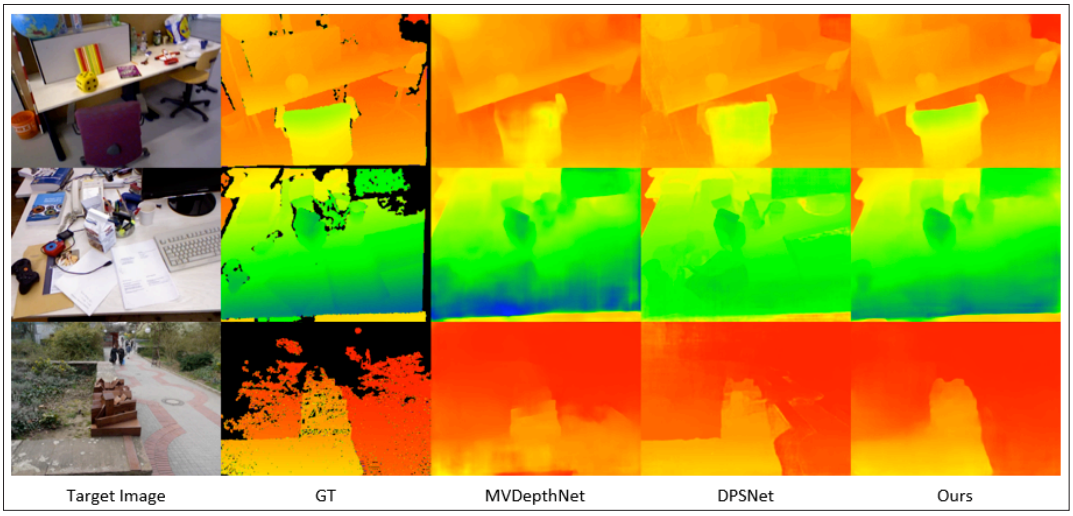
저자는 OpenCL로 가속화된 고전적인 기하학 기반의 다중뷰 스테레오 정합 기법을 통해 비용 볼륨을 구축한다. 이때 python을 사용하는 딥러닝 프레임워크와의 호환성을 위해 python 기반의 OpenCL 프레임워크인 PyOpenCL[4]을 사용했다. 또한 OpenCL 함수를 사용하면서 배치 학습이 가능하도록 커널 함수와 명령어 큐를 복수로 생성하는 기법을 사용하였다.

2) 학습 기반의 깊이 영상 취득

제안하는 네트워크는 깊이 정보를 취득하려는 목표(target) 영상과 여러 장의 참조(reference) 영상에서 계산된 비용 볼륨을 전달받아 CNN 기반의 심층 신경망을



<그림 6> 다중 영상으로부터 3차원 깊이 영상 복원: 다중 뷰 깊이 영상 복원 네트워크



<그림 7> 다중 영상으로부터 3차원 깊이 영상 복원: 깊이 영상의 정성적 성능 평가

통해 깊이 영상을 추론한다. 네트워크는 5개의 인코더 레이어와 5개의 디코더 레이어로 구성되어 있으며, 각 단에서 추출한 특징(feature)은 skip-connection을 통해 디코더 단으로 전달된다.

실험 및 동작 검증 :

실험은 DeMoN[21]에서 취합한 데이터 셋으로부터 훈련 및 검증 데이터셋을 구성했다. 본 모델은 PyTorch 딥러닝 프레임워크를 사용해 구현했으며 모바일 기기 이식을 위해 모바일 프로세서에 최적화된 PyTorch Mobile[4] 프레임워크를 사용했다. 〈그림 7〉에서는 다중뷰 깊이 영상 취득의 SOTA 기법인 MVDepthNet[23], DPSNet[9]과의 정성적 비교 평가 결과를 제시한다. 온 디바이스 친화적인 융합 알고리즘으로도 SOTA 수준의 성능을 보이는 것을 확인할 수 있다.

〈표 5〉는 제시하는 비용 볼륨 구축 과정의 수행 속도 이점을 확인하기 위해 역호모그래피 변환을 하는 신경망 기반의 PyTorch grid_sample 함수[11]와 inverse wrap 모듈의 속도를 비교한다. 실험 결과, 여러 해상도가 입력으로 주어지더라도 OpenCL 기반의 볼륨 구축 속도가 더욱 빠르며, 특히 입력 영상의 너비와 높이를 모두 워크그룹 크기(16, 16)의 배수로 설정한다면 메모리 구조에 따른 최적화가 가능하다.

<표 5> Inverse warp module과 grid_sample 함수 속도 비교

해상도	Inverse warp module [msec]	PyTorch grid sample function [msec]
480x352	313	1,987
640x480	118	2,027

V. 결 론

본 기고문에서는 모바일 GPU에서의 OpenCL 활용법과 온 디바이스 친화적인 기하학 기반, 심층 신경망

기반의 3차원 공간 정보 취득 기술을 살펴보았다. 기하학 기반의 다중뷰 스테레오 정합 알고리즘의 선택적 프레임워크와 함께 이를 응용한 3차원 복원 기법을 소개하였다. 심층 신경망 기반에서는 심층 신경망 네트워크와 가속화된 고전적 스테레오 알고리즘을 결합한 온 디바이스에 최적화된 공간 취득 모델을 소개하였다. 필자는 본 기고문이 추후 온 디바이스 환경에서 3차원 공간 정보 처리 기술의 가속화 연구에 많은 도움이 되기를 바란다.

참 고 문 헌

[1] Caffe-HRT, https://github.com/OAID/Caffe-HRT

[2] ARM Compute Library, https://developer.arm.com/

[3] PyTorch Mobile, https://pytorch.org/mobile/home/

[4] PyOpenCL, https://pypi.org/project/pyopencl/

[5] The Middlebury Computer Vision Pages, http://vision.middlebury.edu/mview/

[6] J. Y. Chang, H. S. Park, I. K. Park, K. M. Lee, and S. U. Lee, "GPU-Friendly Multi-View Stereo Reconstruction Using Surfel Representation and Graph Cuts," Computer Vision and Image Understanding, vol. 115, no. 5, pp. 620-34, 2011.

[7] D. Gallup, J. M. Frahm, P. Mordohai, Q. Yang, and M. Pollefeys, "Real-Time Plane-Sweeping Stereo with Multiple Sweeping Directions," Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp. 1-8, 2007.

[8] H. Hirschmuller, "Stereo Processing by Semiglobal Matching and Mutual Information," IEEE Trans. on Pattern Analysis and Machine Intelligence 30, no. 2, pp. 328-341, 2007.

[9] S. H. Im, H. G. Jeon, S. Lin, and I. S. Kweon, "DPSNet: End-to-End Deep Plane Sweep Stereo," Proc. International Conference on Learning Representations, 2019.

[10] A. Ivan, and I. K Park, "A Flexible and Configurable GPGPU Stereo Matching Framework," Multimedia Tools and Applications, vol. 79, no. 25, pp. 18367-86, 2020.

[11] M. Jaderberg, K. Simonyan, and A. Zisserman, "Spatial Transformer Networks," Proc. Advances in Neural Information Processing Systems, vol. 28, pp. 2017-25, 2015.

[12] Y. B. Jeon, and I. K. Park, "Deep Neural Network for Handcrafted Cost-Based Multi-View Stereo," Proc. International Workshop on Advanced Imaging Technology, 2021.

[13] Z. Ma, K. He, Y. Wei, J. Sun, and E. Wu, "Constant Time Weighted Median Filtering for Stereo Matching and Beyond," Proc. IEEE International Conference on Computer Vision, 2013.

[14] A. Munshi, B. Gaster, T. G. Mattson, and D. Ginsburg, OpenCL Programming Guide. Pearson Education, 2011.

[15] I. K. Park, Nitin Singhal, M. H. Lee, S. D. Cho, and Chris Kim, "Design and Performance Evaluation of Image Processing Algorithms on GPUs," IEEE Trans. on Parallel and Distributed Systems, vol. 22, no. 1, pp. 91-104, 2010.

[16] D. Scharstein, H. Hirschmüller, Y. Kitajima, G. Krathwohl, N. Nešić, X. Wang, and P. Westling, "High-Resolution Stereo Datasets with Subpixel-Accurate Ground Truth," Proc. German Conference on Pattern Recognition, 2014.

[17] D. Scharstein, and R. Szeliski, "High-Accuracy Stereo Depth Maps Using Structured Light," Proc. IEEE Conference on Computer Vision and Pattern Recognition, 2003.

[18] D. Scharstein, and R. Szeliski, "A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms," International journal of computer vision, vol. 47, no. 1, pp. 7-42, 2002.

[19] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski, "A Comparison and Evaluation of Multi-View Stereo Reconstruction Algorithms," Proc. IEEE Conference on Computer Vision and Pattern Recognition, 2006.

[20] J. E. Stone, D. Gohara, and G. Shi, "OpenCL: A Parallel Programming Standard for Heterogeneous Computing Systems," Computing in Science & Engineering, vol. 12, no. 3, pp. 66, 2020.
[21] B. Ummenhofer, H. Zhou, J. Uhrig, N. Mayer, E. Ilg, A. Dosovitskiy, and T. Brox, "DeMoN: Depth and Motion Network for Learning Monocular Stereo," Proc. IEEE Conference on Computer Vision and Pattern Recognition, 2017.
[22] Q. Yang, L. Wang, R. Yang, S. Wang, M. Liao, and D. Nister, "Real-Time Global Stereo Matching Using Hierarchical Belief Propagation," Proc. British Machine Vision Conference, 2006.
[23] Y. Yao, Z. Luo, S. Li, T. Fang, and L. Quan, "MVSNet: Depth Inference for Unstructured Multi-View Stereo," Proc. European Conference on Computer Vision, 2018.
[24] K. J. Yoon and I. S. Kweon, "Adaptive Support-Weight Approach for Correspondence Search," IEEE Trans. on pattern analysis and machine intelligence, vol. 28, no. 4, pp. 650-56, 2006.
[25] R. Zabih and J. Woodfill, "Non-Parametric Local Transforms for Computing Visual Correspondence," Proc. European Conference on Computer Vision, 1994.
[26] 박준형, 박인규, "온 디바이스 얼굴 라이트필드 합성 시스템," 전자공학회 논문지, vol. 58, no. 5, pp. 68-75, 2021년 5월.
[27] 전윤배, 박인규, "임베디드 GPU 에서의 병렬처리를 이용한 모바일 기기에서의 다중뷰 스테레오 정합," 방송공학회논문지, vol. 24, no. 6, pp. 1064-71, 2019년 11월.

필 자 소 개

정 태 현

- 2021년 2월 : 인하대학교 정보통신공학과 학사
- 2021년 3월 ~ 현재 : 인하대학교 전기컴퓨터공학과 석사과정
- 주관심분야 : 컴퓨터비전 및 그래픽스, deep learning, GPGPU

박 준 형

- 2020년 2월 : 한남대학교 전자공학과 학사
- 2020년 3월 ~ 현재 : 인하대학교 전기컴퓨터공학과 석사과정
- 주관심분야 : 컴퓨터비전(3D reconstruction), embedded GPGPU, deep learning

필 자 소 개

박 인 규

- *1995년 2월 : 서울대학교 제어계측공학과 학사*
- *1997년 2월 : 서울대학교 제어계측공학과 석사*
- *2001년 8월 : 서울대학교 전기컴퓨터공학부 박사*
- *2001년 9월 ~ 2004년 2월 : 삼성종합기술원 전문연구원*
- *2007년 1월 ~ 2008년 2월 : Mitsubishi Electric Research Laboratories 방문연구원*
- *2014년 9월 ~ 2015년 8월 : MIT Media Lab 방문부교수*
- *2018년 7월 ~ 2019년 6월 : University of California, San Diego (UCSD) 방문학자*
- *2004년 3월 ~ 현재 : 인하대학교 정보통신공학과 교수*
- *ORCID : https://orcid.org/0000-0003-4774-7841*
- *주관심분야 : 컴퓨터비전 및 그래픽스, deep learning, GPGPU*