(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2013년01월02일
(11) 등록번호 10-1217492
(24) 등록일자 2012년12월26일

(51) 국제특허분류(Int. Cl.)
G01S 3/80 (2006.01)
(21) 출원번호 10-2012-0001530
(22) 출원일자 2012년01월05일
심사청구일자 2012년01월05일
(56) 선행기술조사문헌
L. G. Silveira Jr. 외 3명, "A GPU Implementation of the SRP-PHAT Sound Source Localization Algorithm", Proc. 12th IWAENC, (2010.08)

(73) 특허권자
고려대학교 산학협력단
서울 성북구 안암동5가 1
(72) 발명자
육동석
서울 용산구 서빙고동 신동아아파트 1동 102호
이태우
서울특별시 중구 서애로1길 34, 711호(묵정동, 묵정아파트)
(74) 대리인
특허법인충현

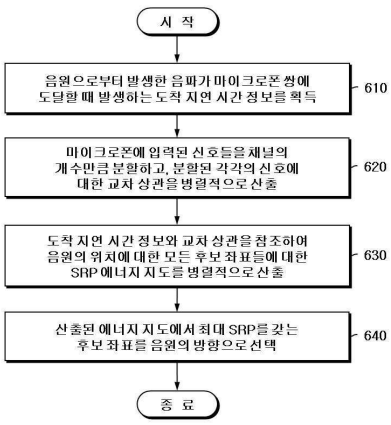
전체 청구항 수 : 총 9 항 심사관 : 윤지영

(54) 발명의 명칭 **병렬 처리를 이용한 음원 위치 추정 장치 및 방법**

(57) 요 약

본 발명은 병렬 처리를 이용한 음원 위치 추정 장치, 방법 및 그 방법을 기록한 기록매체에 관한 것으로, 본 발명에 따른 음원의 위치를 추정하는 방법은, 음원으로부터 발생한 음파가 마이크로폰 쌍에 도달할 때 발생하는 도착 지연 시간 정보를 획득하고, 마이크로폰에 입력된 신호들을 마이크로폰의 채널의 개수만큼 분할하고 마이크로폰 채널의 개수만큼의 처리기를 통해 각각의 분할된 신호에 대한 교차 상관을 병렬적으로 산출하고, 도착 지연 시간 정보와 교차 상관을 참조하여 음원의 위치에 대한 모든 후보 좌표의 개수만큼의 처리기를 통해 모든 후보 좌표들에 대한 SRP 에너지 지도를 병렬적으로 산출하며, 산출된 에너지 지도에서 최대 SRP를 갖는 후보 좌표를 음원의 방향으로 선택한다.

대 표 도 - 도6



특허청구의 범위

청구항 1

복수 개의 마이크로폰(microphone)을 구비하여 음원의 위치를 추정하는 방법에 있어서,

음원(sound source)으로부터 발생한 음파(sound wave)가 상기 마이크로폰 쌍(pair)에 도달할 때 발생하는 도착 지연 시간(time difference of arrival, TDOA) 정보를 획득하는 단계;

상기 마이크로폰에 입력된 신호들을 상기 마이크로폰의 채널의 개수만큼 분할하고, 상기 마이크로폰 채널의 개수만큼의 처리기를 통해 상기 분할된 각각의 신호에 대한 교차 상관(cross correlation)을 병렬적으로 산출하는 단계;

상기 획득된 도착 지연 시간 정보와 상기 산출된 교차 상관을 참조하여 상기 음원의 위치에 대한 모든 후보 좌표의 개수만큼의 처리기를 통해 상기 모든 후보 좌표들에 대한 SRP(steered response power) 에너지 지도(energy map)를 병렬적으로 산출하는 단계; 및

상기 산출된 에너지 지도에서 최대 SRP를 갖는 후보 좌표를 상기 음원의 방향으로 선택하는 단계;를 포함하는 방법.

청구항 2

제 1 항에 있어서,

상기 교차 상관을 산출하는 단계는,

상기 마이크로폰에 입력된 신호들을 상기 마이크로폰의 채널의 개수만큼 분할하고, 상기 마이크로폰 채널의 개수만큼의 처리기를 통해 소정 윈도우(window)를 병렬적으로 곱셈하는 단계;

상기 처리기가 상기 윈도우가 적용된 신호들을 병렬적으로 고속 푸리에 변환(Fast Fourier Transform, FFT)하는 단계;

상기 처리기가 상기 고속 푸리에 변환된 신호들에 대해 병렬적으로 복합 곱셈(conjugated multiplication) 및 PHAT 가중치(phase transform weighting)를 적용하는 단계; 및

상기 처리기가 상기 복합 곱셈 및 PAHT 가중치가 적용된 신호들을 역 고속 푸리에 변환(Inverse Fast Fourier Transform, IFFT)하는 단계;를 포함하는 방법.

청구항 3

제 1 항에 있어서,

상기 처리기는,

멀티 쓰레드(multi-thread) 또는 멀티 프로세스(multi-process)이고,

병렬 처리하고자 하는 대상 신호의 개수만큼 생성되어 해당 작업을 동시에 수행하는 것을 특징으로 하는 방법.

청구항 4

제 1 항에 있어서,

상기 처리기는 GPU(graphic processing unit) 기반의 마이크로 프로세서(micro-processor)를 이용하여 구현되는 것을 특징으로 하는 방법.

청구항 5

제 1 항에 있어서,

상기 SRP 에너지 지도를 산출하는 단계는,

상기 음원의 위치에 대한 모든 후보 좌표의 개수만큼의 처리기가 상기 획득된 도착 지연 시간 정보와 상기 산출

된 교차 상관을 이용하여 상기 후보 좌표에 대한 SRP 값을 병렬적으로 누적하는 단계; 및

상기 처리기가 모든 후보 좌표들에 대해 상기 SRP 값의 누적 과정을 반복함으로써 전체 후보 좌표에 대한 SRP 에너지 지도를 생성하는 단계;를 포함하는 방법.

청구항 6

제 5 항에 있어서,

상기 SRP 에너지 지도를 구성하는 블록(block)의 수는 상기 모든 후보 좌표의 개수를 블록당 최대 처리기의 수로 제산한 결과를 올림(round-up)한 값으로 결정되는 것을 특징으로 하는 방법.

청구항 7

제 1 항에 있어서,

최대 SRP를 갖는 후보 좌표를 선택하는 단계는,

상기 모든 후보 좌표의 개수 및 상기 SRP 에너지 지도를 구성하는 블록의 수에 기초하여 상기 산출된 에너지 지도에서 최대 SRP를 갖는 후보 좌표를 병렬적으로 탐색하는 단계; 및

상기 탐색 결과 발견된 후보 좌표를 상기 음원의 방향으로 결정하는 단계;를 포함하는 방법.

청구항 8

제 1 항에 있어서,

상기 도착 지연 시간 정보는, 상기 복수 개의 마이크로폰 중에서 선택된 고유한 마이크로폰 쌍에 대해 미리 산출되어 소정 테이블에 저장된 후, 상기 도착 지연 시간 정보를 획득하는 단계에서 독출되며, 상기 교차 상관을 위한 인덱스(index)로 변환되는 것을 특징으로 하는 방법.

청구항 9

제 1 항 내지 제 8 항 중에 어느 한 항의 방법을 컴퓨터에서 실행시키기 위한 프로그램을 기록한 컴퓨터로 읽을 수 있는 기록매체.

명 세 서

기 술 분 야

[0001] 본 발명은 음원의 위치를 추정하는 기술에 관한 것으로, 보다 자세하게는 마이크로폰을 통해 입력된 음파에 공간 내의 어느 위치로부터 발생한 것인지를 추정함에 있어서 이러한 음원의 위치를 병렬 처리를 통해 추정하는 장치, 방법 및 그 방법을 기록한 기록매체에 관한 것이다.

배 경 기 술

[0002] 인간과 로봇 간의 상호작용(Human-Robot interaction, HRI) 은 다양한 센서들을 통해 인간과 로봇 간의 자연스러운 상호작용을 촉진하는 기술을 의미한다. 이러한 HRI 기술과 관련하여 사람의 음성 또는 음파 신호를 이용하는 다양한 어플리케이션들이 등장하고 있다. 예를 들어, 사람의 음성을 인식하여 기기를 조작하거나 잡음 환경 내에서 특정 음성 신호를 추출하여 활용하는 등의 활용 방법들이 바로 그것이다. 이러한 음성 신호를 입력받아 특정 명령을 수행하는 과정에서 가장 먼저 수행되어야 하는 것이 바로 음원의 위치를 정확하게 파악하는 일이다.

[0003] 이러한 음원 위치 추정(Sound source localization, SSL)은 음성-기반 HRI의 주요한 기술 요소들 중 하나이며, 음원 위치 추정을 통해 음원의 방향을 식별할 수 있다. 이 때, 빔포밍(Beamforming) 기술은 공간 상에 존재하는 다양한 환경 잡음을 억제하는데 사용되며, 이러한 빔포밍의 전처리 과정에서 음원의 방향에 대한 추정이 요구된다. 이러한 이유에서 음원 위치 추정은 강인한 음성-기반 HRI의 핵심 기술에 해당한다. 이하에서 인용되는 비특허문헌은 가정용 로봇 환경에서의 음원 위치 추정 방법을 소개하고 있다.

[0004] SRP-PHAT(steered response power - phase transform)은 반향(reverberation)에 대한 강인함 때문에 널리 연구되어 왔다. 그러나, SRP-PHAT는 매우 많은 수의 후보 음원 위치를 연산해야만 하기 때문에 실시간으로 실행될

수 없다는 점이 문제점으로 지적되어 있다. 따라서, 이러한 문제점을 극복하기 위한 새로운 기술이 요구되고 있다.

선행기술문헌

비특허문헌

[0005] (비특허문헌 0001) K. Kwak and S. Kim, "Sound source localization with the aid of excitation source information in home robot environments," IEEE Trans. Consumer Electron., vol. 54, no. 2, pp. 852-856, May 2008.

발명의 내용

해결하려는 과제

[0006] 본 발명의 실시예들이 해결하고자 하는 기술적 과제는 종래의 SRP-PHAT 기술이 많은 연산으로 인해 처리에 많은 시간이 소모되며, 그 결과 실시간 실행이 불가능하다는 문제점을 해결하고, 단일 코어의 프로세서에 기반하여 동작하도록 설계되었는 바 성능 향상에 있어 한계가 존재하는 비효율을 극복하고자 한다.

과제의 해결 수단

[0007] 상기 기술적 과제를 해결하기 위하여, 본 발명의 일 실시예에 따른 음원의 위치를 추정하는 방법은, 음원(sound source)으로부터 발생한 음파(sound wave)가 마이크로폰(microphone) 쌍(pair)에 도달할 때 발생하는 도착 지연 시간(time difference of arrival, TDOA) 정보를 획득하는 단계; 상기 마이크로폰에 입력된 신호들을 상기 마이크로폰의 채널의 개수만큼 분할하고, 상기 마이크로폰 채널의 개수만큼의 처리기를 통해 상기 분할된 각각의 신호에 대한 교차 상관(cross correlation)을 병렬적으로 산출하는 단계; 상기 획득된 도착 지연 시간 정보와 상기 산출된 교차 상관을 참조하여 상기 음원의 위치에 대한 모든 후보 좌표의 개수만큼의 처리기를 통해 상기 모든 후보 좌표들에 대한 SRP(steered response power) 에너지 지도(energy map)를 병렬적으로 산출하는 단계; 및 상기 산출된 에너지 지도에서 최대 SRP를 갖는 후보 좌표를 상기 음원의 방향으로 선택하는 단계;를 포함한다.

[0008] 상기된 음원의 위치를 추정하는 방법에서, 상기 교차 상관을 산출하는 단계는, 상기 마이크로폰에 입력된 신호들을 상기 마이크로폰의 채널의 개수만큼 분할하고, 상기 마이크로폰 채널의 개수만큼의 처리기를 통해 소정 윈도우(window)를 병렬적으로 곱셈하는 단계; 상기 처리기가 상기 윈도우가 적용된 신호들을 병렬적으로 고속 푸리에 변환(Fast Fourier Transform, FFT)하는 단계; 상기 처리기가 상기 고속 푸리에 변환된 신호들에 대해 병렬적으로 복합 곱셈(conjugated multiplication) 및 PHAT 가중치(phase transform weighting)를 적용하는 단계; 및 상기 처리기가 상기 복합 곱셈 및 PAHT 가중치가 적용된 신호들을 역 고속 푸리에 변환(Inverse Fast Fourier Transform, IFFT)하는 단계;를 포함한다.

[0009] 상기된 음원의 위치를 추정하는 방법에서, 상기 처리기는, 멀티 쓰레드(multi-thread) 또는 멀티 프로세스(multi-process)이고, 병렬 처리하고자 하는 대상 신호의 개수만큼 생성되어 해당 작업을 동시에 수행한다. 또한, 상기 처리기는 GPU(graphics processing unit) 기반의 마이크로 프로세서(micro-processor)를 이용하여 구현될 수 있다.

[0010] 상기된 음원의 위치를 추정하는 방법에서, 상기 SRP 에너지 지도를 산출하는 단계는, 상기 음원의 위치에 대한 모든 후보 좌표의 개수만큼의 처리기가 상기 획득된 도착 지연 시간 정보와 상기 산출된 교차 상관을 이용하여 상기 후보 좌표에 대한 SRP 값을 병렬적으로 누적하는 단계; 및 상기 처리기가 모든 후보 좌표들에 대해 상기 SRP 값의 누적 과정을 반복함으로써 전체 후보 좌표에 대한 SRP 에너지 지도를 생성하는 단계;를 포함한다.

[0011] 상기된 음원의 위치를 추정하는 방법에서, 최대 SRP를 갖는 후보 좌표를 선택하는 단계는, 상기 모든 후보 좌표의 개수 및 상기 SRP 에너지 지도를 구성하는 블록의 수에 기초하여 상기 산출된 에너지 지도에서 최대 SRP를 갖는 후보 좌표를 병렬적으로 탐색하는 단계; 및 상기 탐색 결과 발견된 후보 좌표를 상기 음원의 방향으로 결정하는 단계;를 포함한다.

[0012] 상기된 음원의 위치를 추정하는 방법에서, 상기 도착 지연 시간 정보는, 상기 복수 개의 마이크로폰 중에서 선

택된 고유한 마이크로폰 쌍에 대해 미리 산출되어 소정 테이블에 저장된 후, 상기 도착 지연 시간 정보를 획득하는 단계에서 독출되며, 상기 교차 상관을 위한 인덱스(index)로 변환된다.

[0013] 한편, 이하에서는 상기 기재된 음원의 위치를 추정하는 방법을 컴퓨터에서 실행시키기 위한 프로그램을 기록한 컴퓨터로 읽을 수 있는 기록매체를 제공한다.

발명의 효과

[0014] 본 발명의 실시예들은 SRP-PHAT의 각 단계를 분리하고, GPU와 같은 가속 프로세서를 이용하여 분리된 각 단계를 병렬적으로 동시에 처리함으로써 많은 연산에도 불구하고 소요 시간을 단축시키고, 그 결과 음원의 위치를 실시간으로 추정할 수 있으며, CPU 뿐만 아니라 GPU와 같은 가속 프로세서를 병행하여 활용함으로써 해당 기기의 성능을 극대화할 수 있다.

도면의 간단한 설명

[0015] 도 1은 본 발명의 실시예들이 채택하고 있는 음원 위치 추정의 병렬 처리를 위해 SRP-PHAT(steered response power - phase transform)을 4 개의 단계로 분리하는 아이디어를 제시한 도면이다.

도 2는 본 발명의 실시예들이 채택하고 있는 TDOA(time difference of arrival) 테이블의 산출 과정을 도시한 도면이다.

도 3은 순차적인 실행 과정에 따라 PHAT 가중치를 통해 교차 상관을 산출하는 과정을 도시한 도면이다.

도 4는 후보 좌표에 대해 TDOA 테이블을 참조하여 SRP를 산출하는 과정을 도시한 도면이다.

도 5는 본 발명의 실시예들이 도 1의 기본 아이디어를 개량하여 GPU에 기반한 SRP-PHAT 과정을 도시한 도면이다.

도 6은 본 발명의 일 실시예에 따른 복수 개의 마이크로폰(microphone)을 구비하여 음원의 위치를 추정하는 방법을 도시한 흐름도이다.

도 7은 본 발명의 실시예들이 채택하고 있는 PHAT 가중치 적용 동작과 교차 상관을 병렬적으로 처리하는 과정을 설명하기 위한 도면이다.

도 8은 도 7의 교차 상관을 처리함에 있어서 쓰레드 블록(thread block)의 차원(dimension)을 예시한 도면이다.

도 9는 도 7 및 도 8의 병렬적인 교차 상관의 전체 절차를 CUDA 의사코드(pseudocode)로서 구현한 것을 예시한 도면이다.

도 10은 병렬적인 방식에 따라 전체 SRP 에너지 지도를 산출하는 과정을 도시한 도면이다.

도 11은 도 10의 에너지 지도에 대한 블록 및 쓰레드의 차원을 예시한 도면이다.

도 12는 도 10 및 도 11의 병렬적인 SRP 에너지 지도 산출 과정을 CUDA 의사코드로서 구현한 것을 예시한 도면이다.

도 13은 최대 SRP 좌표를 탐색하는 과정을 도시한 도면이다.

도 14는 도 13의 탐색을 위한 쓰레드 블록의 차원을 예시한 도면이다.

도 15는 도 13 및 도 14의 최대 SRP 좌표를 탐색하는 과정을 CUDA 의사코드로서 구현한 것을 예시한 도면이다.

발명을 실시하기 위한 구체적인 내용

[0016] 본 발명의 실시예들을 설명하기에 앞서 실시예들이 구현되는 환경에 대해 개괄적으로 소개하고, 실시예들이 공통적으로 채용하고 있는 기본 아이디어를 제시하고자 한다.

[0017] 앞서 간략히 소개한 바와 같이 SRP-PHAT(steered response power - phase transform)은 반향(reverberation)에 대한 강인함 때문에 널리 연구되어 왔다. 그러나, SRP-PHAT는 매우 많은 수의 후보 음원 위치를 연산해야만 하기 때문에 실시간 처리에 문제점을 노출하여 왔다.

[0018] 따라서, 이하에서 기술되는 본 발명의 실시예들에서, SRP-PHAT는 GPU와 같은 가속 프로세서를 활용할 수 있도록 병렬화된다. 또한, 본 발명의 실시예들은 변화하는 마이크로폰의 수에 대응하여 데이터의 병렬성을 가지며, GPU

기반의 SSL 어플리케이션에 직접 적용할 수 있는 구체적인 알고리즘을 제시하고자 한다.

[0019] GPU 기반의 병렬화를 이용하여 음원의 위치를 추정함에 있어서, 높은 성능을 달성하기 위해서는 다음의 2 개의 조건이 만족되어야만 한다. 첫 번째 조건은 데이터의 병렬성(parallelism)이다. 이 때의 알고리즘은 하나의 명령어로서 다중 데이터를 처리할 수 있어야만 한다. 두 번째 조건은 암달의 법칙(Amdahl's law)이다. 암달의 법칙은 알고리즘의 성능을 향상시키기 위해서는 전체 알고리즘 내에서 순차적인 방식으로 처리되는 부분이 작아야만 한다고 안내하고 있다. 즉, 이러한 순차적인 방식으로 처리되는 부분을 최소화하기 위해 본 발명의 실시예들은 가능한 모든 수행 과정을 병렬화하여 처리하고자 한다.

[0020] 이하에서는 우선 SRP-PHAT에 대해 약술하도록 한다.

[0021] SRP-PAHT는 모든 후보 좌표들의 SRP(steered response power)를 산출함으로써 최대 SRP를 통해 음원의 방향을 결정한다. 3차원 구면 좌표 $\vec{x}_{\rho,\theta,\phi}$ 가 주어진 경우, PHAT-가중치 SRP는 다음의 수학식 1과 같이 산출된다.

수학식 1

[0022]
$$P_n(\vec{x}_{\rho,\theta,\phi}) = \sum_{k=0}^{M-2}\sum_{l=k+1}^{M-1}\int_{-\infty}^{\infty} \Psi_{kl}(\omega)M_k(\omega)M_l^*(\omega)e^{j\omega(\tau(\vec{x}_{\rho,\theta,\phi},k)-\tau(\vec{x}_{\rho,\theta,\phi},l))}d\omega$$

[0023] 여기서, M은 마이크로폰의 개수를 나타낸다. n-번째 프레임에서, SRP 산출은 모든 고유한 마이크로폰 쌍(pair)인 (k, l)에 대해 수행된다. $M_k(\omega)$ 는 k-번째 마이크로폰 신호의 고속 푸리에 변환(fast Fourier transform, FFT) 값이고, $M_k(\omega)M_l^*(\omega)$ 은 (k, l) 쌍 간의 교차 스펙트럼(cross-spectrum)이다. $\tau(\vec{x}_{\rho,\theta,\phi}, k)$ 및 $\tau(\vec{x}_{\rho,\theta,\phi}, l)$ 는 $\vec{x}_{\rho,\theta,\phi}$ 로부터 각각 k 및 l까지의 TDOA이다. $\Psi_{kl}(\omega)$ 는 $1/\left|M_k(\omega)M_l^*(\omega)\right|$ 인 PHAT 가중치 함수이다.

수학식 2

[0024]
$$\hat{x}_s^n = \arg\max_{\vec{x}_{\rho,\theta,\phi}} P_n(\vec{x}_{\rho,\theta,\phi})$$

[0025] 이상의 수학식 2에서 볼 수 있듯이, SRP는 모든 후보 좌표들에 대해 산출되고, n-번째 프레임에서 최대 SRP $P_n(\vec{x}_{\rho,\theta,\phi})$ 를 갖는 좌표 $\vec{x}_{\rho,\theta,\phi}$ 의 방향 θ, ϕ 이 음원의 방향으로 추정된다.

[0026] 이제, 이상에서 설명한 SRP-PHAT가 단일 코어(single core) CPU에서 어떻게 구현되는지를 설명하도록 한다.

[0027] SRP-PHAT 는 크게 3 개의 단계로 분리된 방법을 이용하여 구현될 수 있다. 즉, 역 사상(inverse mapping), 교차 상관(cross correlation) 및 SRP 탐색이 그것이다. 그러나, 앞서 설명한 바와 같이 본 발명의 실시예들은 병렬처리를 통해 음원의 위치 추정의 성능을 향상시키고자 하므로, 이러한 병렬화에 적합한 구조를 제공하기 위해 이상의 3 개의 단계를 다음과 같은 4 개의 SRP-PHAT 단계로 수정할 것을 제안한다.

[0028] 도 1은 본 발명의 실시예들이 채택하고 있는 음원 위치 추정의 병렬 처리를 위해 SRP-PHAT(steered response power - phase transform)을 4 개의 단계로 분리하는 아이디어를 제시한 도면으로서, TDOA 테이블 독출(load)(110), 교차 상관(120), SRP 에너지 지도 산출(130) 및 최대 SRP 좌표 탐색(140)의 단계들을 포함한다. 이하에서 각각의 단계들을 구체적으로 설명한다.

[0029] 도 2은 본 발명의 실시예들이 채택하고 있는 TDOA(time difference of arrival) 테이블의 산출 과정을 도시한 도면으로서, 도 1에서 독출된 TDOA 테이블(110)이 어떠한 방식에 의해 사전에 구성되는지를 설명하고 있다.

[0030] 도 2를 참조하면, 구면 좌표에 대해 고유한 마이크로폰 쌍 간의 이상적인(ideal) TDOA가 산출되어, TDOA 테이블

내에 일 요소로서 저장되고 있음을 보여주고 있다. 이러한 과정은 모든 고유한 마이크로폰 쌍에 대한 연산이 완료될 때까지 잇달아 반복되게 된다. 구면 좌표를 이용하여 이상적인 TDOA 값을 산출하는 과정은 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자에게 자명한 것으로서, SRP-PHAT의 병렬화에 집중하고 있는 본 발명의 본질을 흐릴 우려가 있으므로 여기서는 생략한다.

[0031] 이상과 같이 모든 고유한 마이크로폰 쌍에 대한 이상적인 TDOA 값들은 음원의 위치에 대한 모든 후보 좌표들보다 미리 산출되는 것이 바람직하다. 따라서, 이상적인 TDOA 값들은 실질적으로 SRP-PHAT를 수행하고자 하는 프로그램이 동작하기 이전에 산출되어 특정 공간(예를 들어, TDOA 테이블과 같은 형태의 저장 공간이 될 수 있다.)에 저장되어 필요시 독출되어 활용된다.

[0032] 다음으로, 교차 상관(Cross Correlation)이 수행된다.

[0033] 교차 상관은 3 단계로 구성되는데, 고속 푸리에 변환(FFT), PHAT 가중치를 통한 곱셈(multiplication) 및 역 고속 푸리에 변환(inverse FFT)이 그것이다. M 개의 마이크로폰들이 주어진 경우, 이러한 교차 상관은 매 프레임마다 ${}_{M}C_{2}$ 쌍에 대해 산출되며, 이들은 보통 모든 후보 좌표가 산출될 때 변경없이 참조될 수 있다. 따라서, 이러한 교차 상관 동작이 매 프레임마다 한 번 수행되면, 이러한 단계는 다른 단계들과 분리될 때 더 좋은 성능이 발휘될 수 있을 것이다.

[0034] 도 3은 순차적인 실행 과정에 따라 PHAT 가중치를 통해 교차 상관을 산출하는 과정을 도시한 도면이다.

[0035] 도 3에서, 우선 해밍 윈도우(Hamming window)가 모든 채널에 대해 곱해진다. 본 실시예에서는 해밍 윈도우를 활용하였으나, 해밍 윈도우 이외에도 다양한 윈도우 함수가 활용될 수 있다. 그런 다음, FFT와 역 FFT가 각각의 프레임에 대해 순차적으로 수행됨으로써, 복합 곱셈(conjugated multiplication)과 PHAT 가중치(weighting)가 프레임의 개별 요소에 대해 적용된다. 도 3은 복합 곱셈 및 PHAT 가중치 쓰레드가 ch1-번째 및 ch2-번째 프레임의 i-번째 요소에 적용되는 것을 도시하고 있다.

[0036] 이제, SRP 에너지 지도(Energy Map)가 산출된다. 앞서 설명한 바와 같이 ${}_{M}C_{2}$개의 고유한 마이크로폰 쌍의 교차 상관 및 TDOA 테이블은 각각 이전 단계(120, 110)에서 산출되었다.

[0037] 도 4는 후보 좌표에 대해 TDOA 테이블을 참조하여 SRP를 산출하는 과정을 도시한 도면으로서, SRP 산출은 TDOA 테이블로부터 후보 좌표에 대한 각각의 쌍의 TDOA 값을 참조함으로써 교차 상관 값들을 누적하기 위한 과정으로서 이해될 수 있다.

[0038] 도 4의 상단은 교차 상관 배열에서 TDOA(x,y)-번째 요소를 참조함으로써 후보 좌표의 SRP(y)가 누적되는 것을 보여주고 있다. 이러한 동작은 후보 좌표의 SRP를 산출하기 위해 열(row)을 따라 순차적으로 반복된다. 또한, 모든 후보 좌표에 대한 전체 SRP 에너지 지도를 산출하기 위해 후보 좌표에 대해 SRP 동작이 순차적인 방식으로 반복된다.

[0039] 마지막으로, 최대 SRP 좌표가 탐색된다. 앞서 간략히 소개한 바와 같이, 본 발명의 실시예들은 병렬화 알고리즘을 위해 최대 SRP 좌표를 탐색하는 모듈을 SRP 탐색 단계로부터 분리하였다. 보다 구체적인 방법은 이후, 도 13 및 도 14를 통해 설명하도록 한다.

[0040] 이제, 이상과 같은 SRP-PHAT를 병렬적으로 처리하는 방법에 대해 설명하도록 한다. 이하에서 기술될 본 발명의 실시예들은 마이크로폰의 개수 및 후보 좌표들에 대응하여 데이터 병렬성을 갖는 GPU 기반의 SRP-PAHT에 대한 일반화된 프레임워크를 제안하고 있으며, CUDA 의사코드(pseudocodes)를 통해 쓰레드 블록(thread block)의 차원을 고려한다. 여기서, CUDA(compute unified device architecture)는 GPU에 기반한 프로그래밍 툴킷(toolkit)의 일례로서 활용되었다.

[0041] 앞서 언급한 바와 같이, GPU 프로그래밍에서 높은 성능을 달성할 수 있도록 암달의 법칙 및 데이터의 병렬성을 고려하여 전체 알고리즘을 병렬화하도록 한다. 다만, 도 1에서 설명한 것과는 달리, GPU 기반의 병렬 처리에서는 호스트(host)와 디바이스(device) 간에 추가적인 메모리 복사 동작이 필요하다.

[0042] 도 5는 본 발명의 실시예들이 도 1의 기본 아이디어를 개량하여 GPU에 기반한 SRP-PHAT 과정을 도시한 도면으로서, 110 단계 내지 140 단계는 도 1과 동일하므로, 여기서는 새롭게 추가된 510 단계 및 520 단계에 집중하여 설명하도록 한다.

[0043] 도 5에 도시된 바와 같이, 메모리 복사 동작을 교차 상관 단계 이전(510)과 메모리 복사 시간을 감소시키기 위해 최대 SRP 좌표를 탐색하는 마지막 단계 이후(520)에 배치하였다. 즉, 510 단계에서는 호스트로부터 디바이스

로 메모리 복사가 수행되고, 520 단계에서는 디바이스로부터 호스트로 메모리 복사가 수행되는 것이 바람직하다. 데이터 병렬성은 단일 명령 다중 데이터(single instruction to multiple data, SIMD)의 어플리케이션을 위해 조심스럽게 고려되어야 한다.

[0044] 쓰레드 블록(thread block)의 차원을 구성하는 방법 역시 중요하다. 왜냐하면, 이러한 과정은 실행 시간 중의 커널의 성능에 영향을 미칠 수 있기 때문이다. 특히, 만약 쓰레드 블록의 차원이 복잡하다면, 복잡한 인덱싱(indexing) 방법이 사용되어야만 한다. 이러한 복잡한 인덱싱은 커널에 있어서 조건 표현의 사용을 요구함으로써 그 성능을 저하시키게 될 것이다. 또한, 만약 쓰레드가 적절하게 분배되지 못할 경우에도, CUDA의 런타임 스케줄링을 방해되므로 성능이 저하될 수 있다.

[0045] 도 6은 본 발명의 일 실시예에 따른 복수 개의 마이크로폰(microphone)을 구비하여 음원의 위치를 추정하는 방법을 도시한 흐름도이다. 각각의 단계들은 적어도 하나의 프로세서(processor)를 이용하여 구현되는 것이 바람직하며, 이하에서 기술될 일련의 처리 과정을 수행하기 위해 상기된 프로세서 이외에 데이터를 임시로 저장하기 위한 메모리(memory) 또는 캐시(cache)와 같은 별도의 저장 공간이 활용될 수 있다. 비록, 이러한 처리 과정은 프로세서, 메모리 또는 캐시와 같은 하드웨어에 기반하여 동작하나, 이들을 제어하고 연산을 수행하기 위한 소프트웨어 코드가 활용될 수 있음은 당연하다. 따라서, 이하에서는 이러한 하드웨어 및 소프트웨어의 결합으로 구현된 수행 주체를 음원 위치 처리 장치라고 명명하였다.

[0046] 210 단계에서, 음원 위치 처리 장치는 음원(sound source)으로부터 발생한 음파(sound wave)가 마이크로폰 쌍(pair)에 도달할 때 발생하는 도착 지연 시간(time difference of arrival, TDOA) 정보를 획득한다.

[0047] 220 단계에서, 음원 위치 처리 장치는 상기 마이크로폰에 입력된 신호들을 상기 마이크로폰의 채널의 개수만큼 분할하고, 상기 마이크로폰 채널의 개수만큼의 처리기를 통해 상기 분할된 각각의 신호에 대한 교차 상관(cross correlation)을 병렬적으로 산출한다.

[0048] 230 단계에서, 음원 위치 처리 장치는 210 단계를 통해 획득된 도착 지연 시간 정보와 220 단계를 통해 산출된 교차 상관을 참조하여 상기 음원의 위치에 대한 모든 후보 좌표의 개수만큼의 처리기를 통해 상기 모든 후보 좌표들에 대한 SRP(steered response power) 에너지 지도(energy map)를 병렬적으로 산출한다.

[0049] 240 단계에서, 음원 위치 처리 장치는 230 단계를 통해 산출된 에너지 지도에서 최대 SRP를 갖는 후보 좌표를 상기 음원의 방향으로 선택한다.

[0050] 이상의 과정들에서, 처리기는 소프트웨어적인 측면에서 멀티 쓰레드(multi-thread) 또는 멀티 프로세스(multi-process)로서 구현될 수 있으며, 병렬 처리하고자 하는 대상 신호의 개수만큼 생성되어 해당 작업을 동시에 수행하는 역할을 한다. 또한, 이러한 처리기는 하드웨어적인 측면에서 GPU(graphic processing unit) 기반의 마이크로 프로세서(micro-processor)를 이용하여 구현될 수 있을 것이다. 물론, 병렬 처리를 위해 GPU 이외의 다양한 멀티 프로세서(multi-processor)가 활용될 수 있음은 당연하다.

[0051] 이제, 데이터 병렬성 및 쓰레드 블록의 차원을 고려한 SRP-PHAT의 각각 4개의 단계에 대한 일반화된 병렬화 프레임워크(framework)를 설명하도록 한다.

[0052] (1) TDOA 테이블

[0053] 앞서 도착 지연 시간(TDOA) 정보가 미리 산출되어 테이블에 저장될 수 있음을 설명하였다. 보다 구체적으로, 도착 지연 시간 정보는, 복수 개의 마이크로폰 중에서 선택된 고유한 마이크로폰 쌍에 대해 미리 산출되어 특정 저장 공간(TDOA 테이블을 의미한다.)에 저장된 후, 도착 지연 시간 정보를 획득하는 단계에서 독출되며, 교차 상관을 위한 인덱스(index)로 변환되는 것이 바람직하다.

[0054] TDOA 값들은 초기화 단계에서 독출되는데, 이 때 모든 TDOA 값들은 MAX_{TDOA}만큼 더해져, 교차 상관을 위한 인덱스로서 직접 사용될 수 있다. 이러한 연산은 각각의 참조 인덱스가 교차 상관 요소를 참조하는 양의 인덱스로서 재할당될 수 있도록 도와준다. 그런 다음, TDOA 테이블은 디바이스의 전역 메모리(global memory)에 복사된다.

[0055] (2) 교차 상관(Cross Correlation)

[0056] 도 7은 본 발명의 실시예들이 채택하고 있는 PHAT 가중치 적용 동작과 교차 상관을 병렬적으로 처리하는 과정을

설명하기 위한 도면으로서, 이하에서는 편의상 FFT 및 역 FFT를 CUDA 툴킷으로서 제공되는 CUFFT로 대체하여 설명하도록 한다. 곱셈 및 PHAT 가중치 처리 동안, 하나의 프레임의 각 요소는 동일한 쓰레드를 이용하여 산출된다. 이러한 쓰레드는 복합 곱셈과 PHAT 가중치 처리를 산출한다.

[0057] 도 7을 참조하면, 해밍 윈도우가 M 개의 채널 데이터(마이크로폰의 수에 대응된다.)에 적용된다. 해밍 윈도우는 초기화 단계에서 산출되고, 디바이스의 전역 메모리에 복사된다. 그러면, 채널 데이터의 각 요소는 쓰레드를 이용하여 각각 대응하는 윈도우 요소와 곱해진다.

[0058] 도 8은 도 7의 교차 상관을 처리함에 있어서 쓰레드 블록(thread block)의 차원(dimension)을 예시한 도면으로, 도 8의 (a)에 도시된 바와 같이, 블록과 쓰레드의 차원은 각각 프레임 크기 및 M에 해당한다. 그러면, 포워드 CUFFT(forward CUFFT)가 M 개의 채널 데이터에 대해 산출되고, 또한 역 CUFFT가 $_{M}C_{2}$개의 고유한 마이크로폰 쌍에 대해 일괄 작업(batch process)으로서 산출된다. 복합 곱셈과 PHAT 가중치를 산출하는 모든 쓰레드는 하나의 프레임의 각각의 요소들에 동시에 적용될 수 있다. 이러한 점에서, FFT의 프레임 크기는 일반적으로 2의 지수로서 결정되고 워프 크기(warp size)가 CUDA에서의 작업 스케줄링의 단위로 주어지기 때문에, 블록 및 쓰레드의 차원은 도 8의 (b)에 도시된 바와 같이, 각각 N_F(frame size)/N_W(warp size) 및 N_W(warp size)으로써 결정될 수 있다.

[0059] $_{M}C_{2}$개의 쌍에 대해 산출된 $MAX_{TDOA}\times2+1$의 교차 상관 결과는 이후에 설명할 도 10에 도시된 바와 같이 재배치되고, TDOA 테이블 값으로부터 직접 참조될 수 있도록 변수 d_CC에 복사된다. 이러한 동작은 다중 채널 데이터의 교차 상관 결과가 공유 메모리를 이용하여 제한된 크기로 산출될 수 있도록 한다. 이것은 또한 커널 인덱싱 동작으로 단순화된다. 교차 상관을 복사하기 위한 블록 및 쓰레드의 차원은 도 8의 (c)에 도시된 바와 같이 각각 $_{M}C_{2}$ 및 MAX_{TDOA}이다.

[0060] 도 9는 도 7 및 도 8의 병렬적인 교차 상관의 전체 절차를 CUDA 의사코드(pseudocode)로서 구현한 것을 예시한 도면으로서, 도 7 내지 도 9를 참조하면, 교차 상관을 산출하는 단계는 다음과 같이 요약될 수 있다.

[0061] 우선, 마이크로폰에 입력된 신호들을 마이크로폰의 채널의 개수만큼 분할하고, 마이크로폰 채널의 개수만큼의 처리기를 통해 해밍 윈도우(Hamming window)를 병렬적으로 곱셈한다. 다음으로, 처리기는 해밍 윈도우가 적용된 신호들을 병렬적으로 고속 푸리에 변환(Fast Fourier Transform, FFT)하고, 고속 푸리에 변환된 신호들에 대해 병렬적으로 복합 곱셈(conjugated multiplication) 및 PHAT 가중치(phase transform weighting)를 적용한다. 이어서, 처리기는 복합 곱셈 및 PAHT 가중치가 적용된 신호들을 역 고속 푸리에 변환(Inverse Fast Fourier Transform, IFFT)하게 된다.

[0062] (3) SRP 에너지 지도(Energy Map)

[0063] 도 10은 병렬적인 방식에 따라 전체 SRP 에너지 지도를 산출하는 과정을 도시한 도면으로서, 순차적인 SRP 에너지 지도 산출 과정에서 SRP가 병렬적인 방식으로 산출된다는 점이 특화되었다.

[0064] 병렬화 구현에 있어서, TDOA 테이블의 요소들은 열(row) 방향으로 동시에 산출될 수 없다. 왜냐하면, 역 방향에서의 수행되는 동작은 쓰레드들 간의 충돌을 야기하는 오버래핑(overlapping) 누적으로 이어지기 때문이다. 따라서, 정확도에 아무런 손실이 없는 가장 작은 쓰레드가 열에 대한 후보 좌표를 위한 SRP 산출이 될 수 있다. 그러면, 이러한 쓰레드는 동시에 모든 후보 좌표에 적용될 수 있다.

[0065] 도 10에서, 쓰레드는 모든 고유한 마이크로폰 쌍에 대한 후보 좌표에 대해 $MAX_{TDOA}\times2+1$의 교차 상관 값의 TDOA(x,y)-번째 값을 누적한다. 이러한 쓰레드는 모든 후보 좌표에 대해 동시에 적용된다. 도 11은 도 10의 에너지 지도에 대한 블록 및 쓰레드의 차원을 예시한 도면으로서, 블록 및 쓰레드의 차원은 각각 ceil(N_C/N_{MTB}) 및 N_{MTB}로써 설정될 수 있다. 여기서, N_C는 후보 좌표의 개수이고, N_{MTB}는 블록당 최대 쓰레드의 개수이며, ceil()은 올림(round-up) 함수를 의미한다.

[0066] 이상과 같은 방식으로 차원을 설정하는 것은 다음의 2 가지 이유 때문이다. 첫째, 이와 같이 함으로써 공유 메모리에 로드된 데이터의 사용을 최대화할 수 있다. 둘째, N_{MTB}는 보통 2의 지수로써 결정되므로 다음 단계에 대해 기능적으로도 효과적이다.

[0067] 도 12는 도 10 및 도 11의 병렬적인 SRP 에너지 지도 산출 과정을 CUDA 의사코드로서 구현한 것을 예시한 도면

이다. 도 12에서, 최초에 각각의 쓰레드가 모든 쌍의 $MAX_{TDOA}\times 2+1$ 교차 상관 값을 공유 메모리에 로드하는 것을 보여준다. 그런 다음, 쓰레드들 간의 충돌을 피하기 위해 SRP는 열 방향으로 순차적으로 산출된다. 이러한 점에서, 유효한 범위 바깥에 놓인 쓰레드는 생략된다.

[0068] 도 10 내지 도 11를 참조하면, SRP 에너지 지도를 산출하는 단계는 다음과 같이 요약될 수 있다. 우선, 음원의 위치에 대한 모든 후보 좌표의 개수만큼의 처리기가 앞서 획득된 도착 지연 시간 정보와 앞서 산출된 교차 상관을 이용하여 후보 좌표에 대한 SRP 값을 병렬적으로 누적한다. 이러한 처리기는 모든 후보 좌표들에 대해 SRP 값의 누적 과정을 반복함으로써 전체 후보 좌표에 대한 SRP 에너지 지도를 생성한다. 이 때, SRP 에너지 지도를 구성하는 블록(block)의 수는 모든 후보 좌표의 개수를 블록당 최대 처리기의 수로 제산한 결과를 올림(round-up)한 값으로 결정될 수 있다.

[0069] (4) 최대 SRP 좌표(Maximum SRP Coordinates)

[0070] 도 13은 최대 SRP 좌표를 탐색하는 과정을 도시한 도면으로서, 최대 SRP를 갖는 후보 좌표를 선택하는 단계는 다음과 같이 요약될 수 있다. 우선, 모든 후보 좌표의 개수 및 SRP 에너지 지도를 구성하는 블록의 수에 기초하여 앞서 산출된 에너지 지도에서 최대 SRP를 갖는 후보 좌표를 병렬적으로 탐색한다. 그런 다음, 탐색 결과 발견된 후보 좌표를 음원의 방향으로 결정한다.

[0071] 보다 구체적으로, 병렬화 구현에 있어서 최대 SRP 좌표를 탐색하기 위해 다수의 내부 단계들이 사용되었고, 이러한 단계들의 수는 N_C 및 N_{MTB}에 의해 결정된다. 만약 N_C가 $(N_{MTB})^{n-1} < N_C \leq (N_{MTB})^n$의 범위를 가지면, 단계들의 수는 n으로 결정된다.

[0072] 예를 들어, 단계들의 수가 2로 결정되었다고 가정하면, 전체 과정은 도 13에 도시된 바와 같이 수행된다. 첫 번째 단계에서, N_{MTB} 요소들이 블록과 최대값으로 연결되고, 인덱스는 인덱스 절약(index saving) 동작의 감소 이후에 각각 변수 d_max_SRPs 및 d_max_indices에 저장된다. 두 번째 단계에서, d_max_SRPs에 대한 최대 값이 다시 탐색되고, 그 인덱스는 cache(1)에 저장된다. 마지막으로, 최대 SRP 좌표의 전역 인덱스가 cache(1) 및 d_max_indices를 역참조(de-referencing)함으로써 산출되게 된다. 만약 $ceil(N_C/N_{MTB})$ 가 2보다 크다면, 최대 SRP 좌표의 전역 인덱스는 다수의 단계들의 재귀적인 역참조에 의해 산출되어야만 한다.

[0073] 도 14는 도 13의 탐색을 위한 쓰레드 블록의 차원을 예시한 도면으로서, 도 14의 (a)에 도시된 바와 같이 첫 번째 단계 동안 각각 $ceil(N_C/N_{MTB})$ 및 N_{MTB}로서 결정되고, 도 14의 (b)에 도시된 바와 같이 두 번째 단계 동안 1 및 $ceil(N_C/N_{MTB})$로서 결정된다. 도 15는 도 13 및 도 14의 최대 SRP 좌표를 탐색하는 과정을 CUDA 의사코드로서 구현한 것을 예시한 도면이다.

[0074] 상기된 본 발명의 실시예들에 따르면, SRP-PHAT의 각 단계를 독립적인 4 개의 단계로 분리하고, GPU와 같은 가속 프로세서를 이용하여 분리된 각 단계를 병렬적으로 동시에 처리함으로써 많은 연산에도 불구하고 소요 시간을 단축시키고, 그 결과 음원의 위치를 실시간으로 추정할 수 있으며, CPU 뿐만 아니라 GPU와 같은 가속 프로세서를 병행하여 활용함으로써 해당 기기의 성능을 극대화할 수 있다.

[0075] 한편, 본 발명은 컴퓨터로 읽을 수 있는 기록 매체에 컴퓨터가 읽을 수 있는 코드로 구현하는 것이 가능하다. 컴퓨터가 읽을 수 있는 기록 매체는 컴퓨터 시스템에 의하여 읽혀질 수 있는 데이터가 저장되는 모든 종류의 기록 장치를 포함한다.

[0076] 컴퓨터가 읽을 수 있는 기록 매체의 예로는 ROM, RAM, CD-ROM, 자기 테이프, 플로피디스크, 광 데이터 저장장치 등이 있으며, 또한 캐리어 웨이브(예를 들어 인터넷을 통한 전송)의 형태로 구현하는 것을 포함한다. 또한, 컴퓨터가 읽을 수 있는 기록 매체는 네트워크로 연결된 컴퓨터 시스템에 분산되어, 분산 방식으로 컴퓨터가 읽을 수 있는 코드가 저장되고 실행될 수 있다. 그리고 본 발명을 구현하기 위한 기능적인(functional) 프로그램, 코드 및 코드 세그먼트들은 본 발명이 속하는 기술 분야의 프로그래머들에 의하여 용이하게 추론될 수 있다.

[0077] 이상에서 본 발명에 대하여 그 다양한 실시예들을 중심으로 살펴보았다. 본 발명에 속하는 기술 분야에서 통상의 지식을 가진 자는 본 발명이 본 발명의 본질적인 특성에서 벗어나지 않는 범위에서 변형된 형태로 구현될 수 있음을 이해할 수 있을 것이다. 그러므로 개시된 실시예들은 한정적인 관점이 아니라 설명적인 관점에서 고려되어야 한다. 본 발명의 범위는 전술한 설명이 아니라 특허청구범위에 나타나 있으며, 그와 동등한 범위 내에 있

는 모든 차이점은 본 발명에 포함된 것으로 해석되어야 할 것이다.

부호의 설명

[0078] 110 : TDOA 테이블 독출 120 : 교차 상관 산출

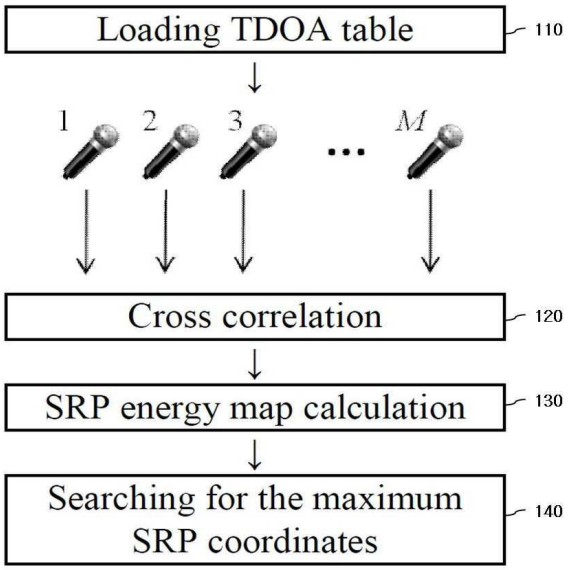
130 : SRP 에너지 지도 산출 140 : 최대 SRP 좌표 탐색

510 : 메모리 복사 (호스트로부터 디바이스로)

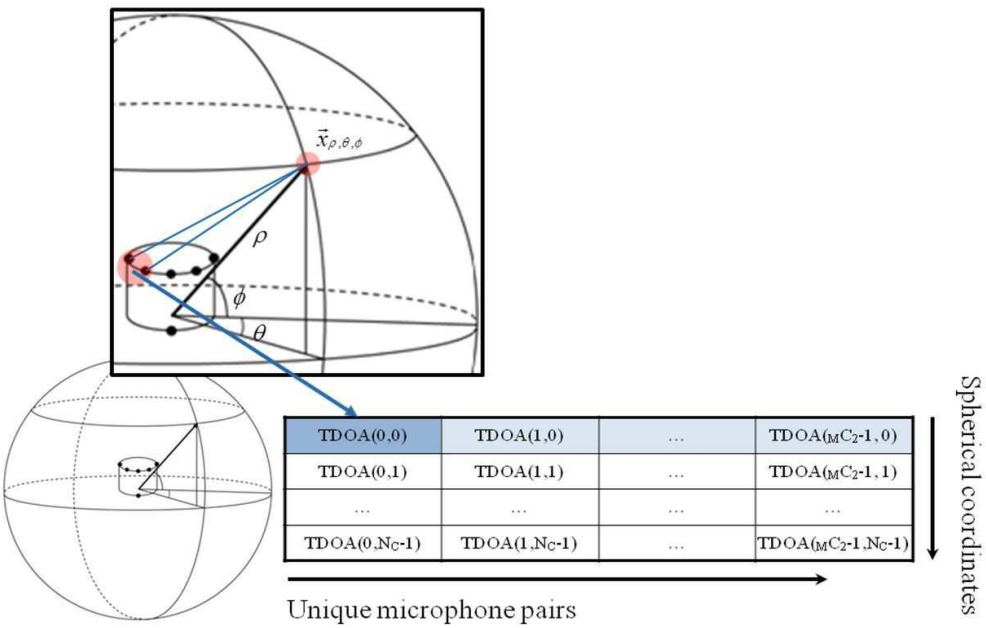
520 : 메모리 복사 (디바이스로부터 호스트로)

도면

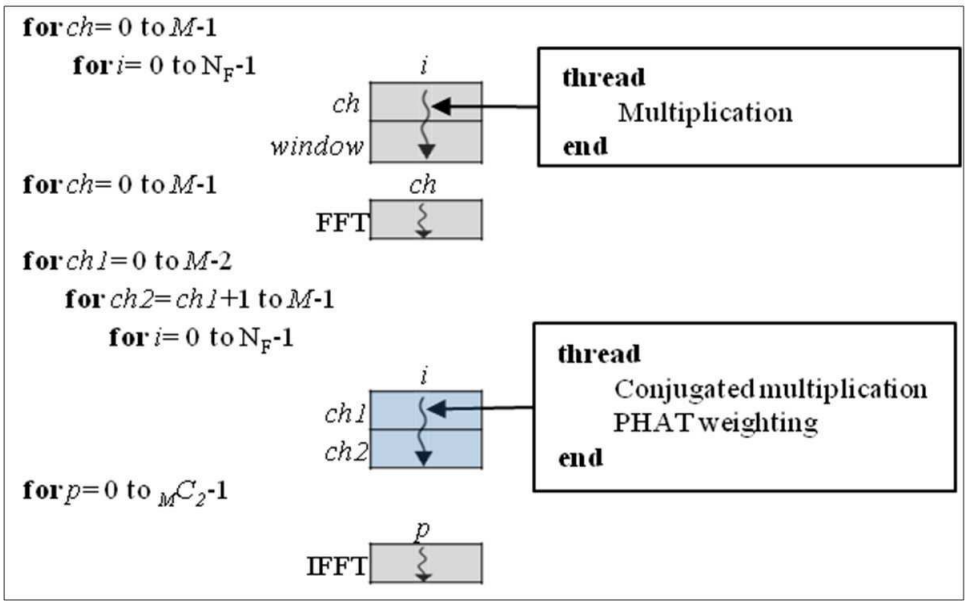
도면1



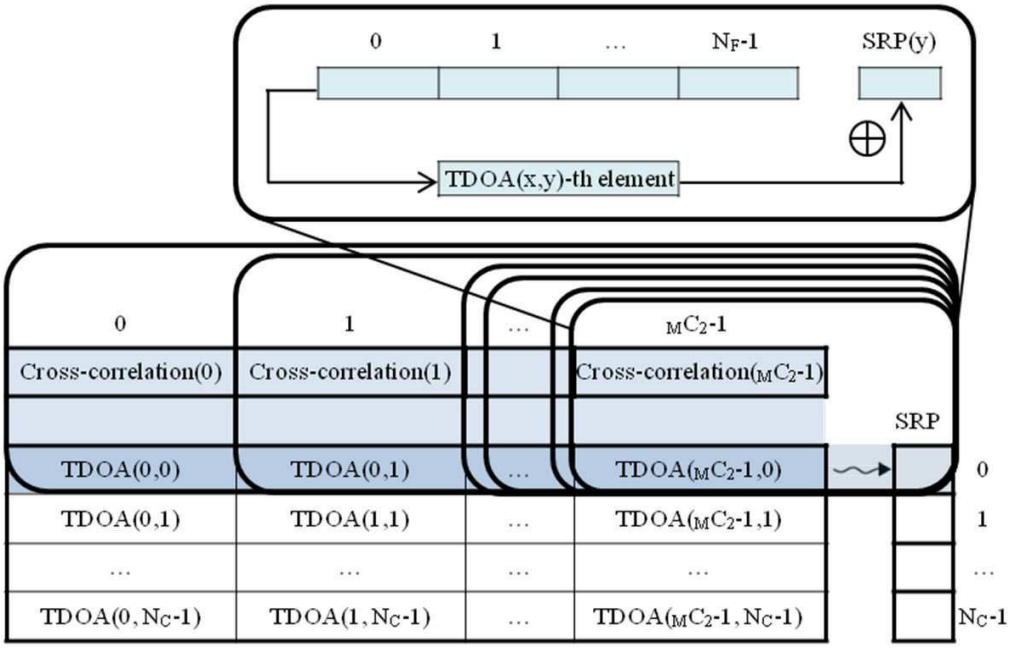
도면2



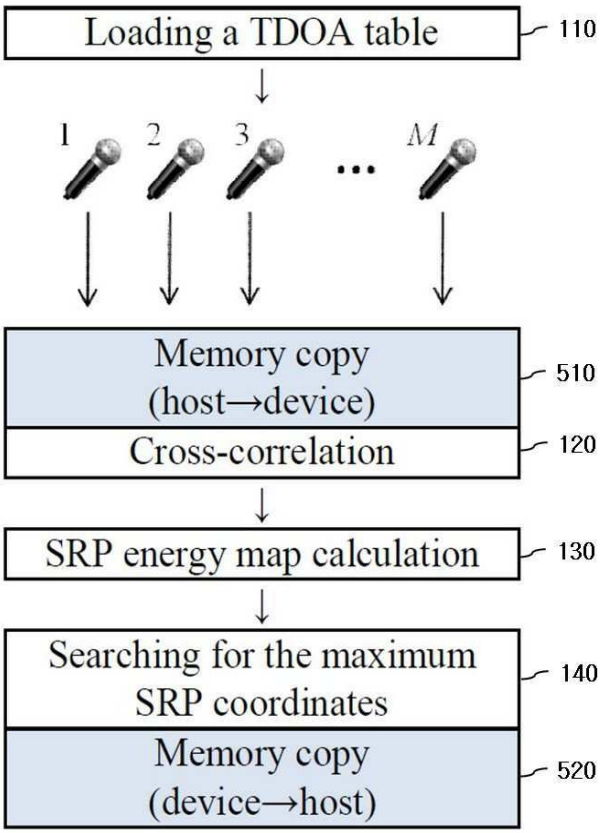
도면3


for ch= 0 to M-1
for i= 0 to N_F-1
i
ch
window
thread
Multiplication
end
for ch= 0 to M-1
ch
FFT
for ch1= 0 to M-2
for ch2= ch1+1 to M-1
for i= 0 to N_F-1
i
ch1
ch2
thread
Conjugated multiplication
PHAT weighting
end
for p= 0 to MC2-1
p
IFFT

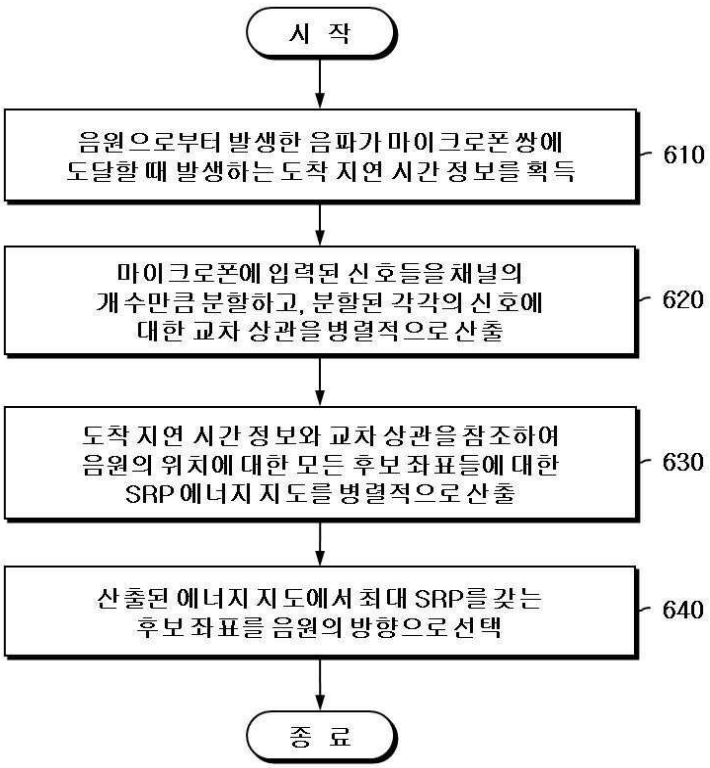
도면4


0
1
…
N_F-1
SRP(y)
TDOA(x,y)-th element
0
1
…
MC2-1
Cross-correlation(0)
Cross-correlation(1)
Cross-correlation(MC2-1)
SRP
TDOA(0,0)
TDOA(0,1)
…
TDOA(MC2-1,0)
0
TDOA(0,1)
TDOA(1,1)
…
TDOA(MC2-1,1)
1
…
…
…
…
…
TDOA(0,NC-1)
TDOA(1,NC-1)
…
TDOA(MC2-1,NC-1)
NC-1

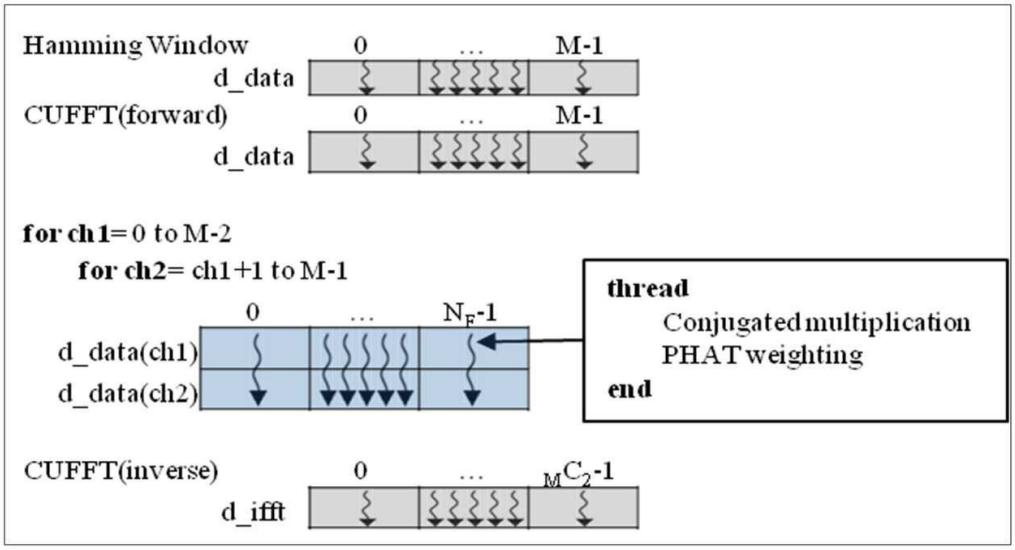
도면5


Loading a TDOA table
110
1
2
3
...
M
Memory copy
(host→device)
510
Cross-correlation
120
SRP energy map calculation
130
Searching for the maximum
SRP coordinates
140
Memory copy
(device→host)
520

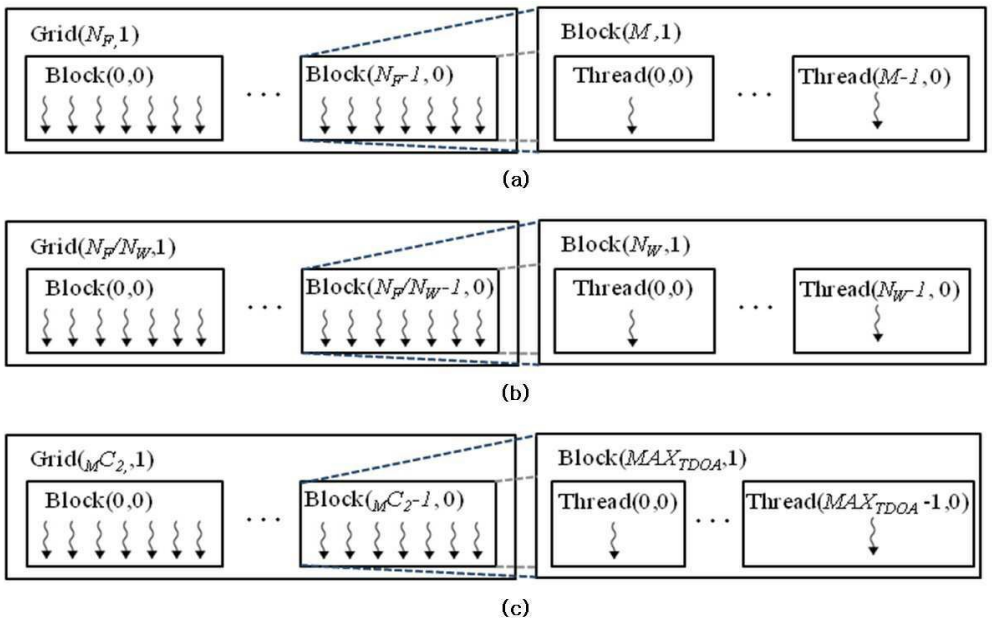
도면6


시 작
음원으로부터 발생한 음파가 마이크로폰 쌍에
도달할 때 발생하는 도착 지연 시간 정보를 획득
610
마이크로폰에 입력된 신호들을 채널의
개수만큼 분할하고, 분할된 각각의 신호에
대한 교차 상관을 병렬적으로 산출
620
도착 지연 시간 정보와 교차 상관을 참조하여
음원의 위치에 대한 모든 후보 좌표들에 대한
SRP 에너지 지도를 병렬적으로 산출
630
산출된 에너지 지도에서 최대 SRP를 갖는
후보 좌표를 음원의 방향으로 선택
640
종 료

도면7


Hamming Window
0 … M-1
d_data
CUFFT(forward)
0 … M-1
d_data
for ch1= 0 to M-2
for ch2= ch1+1 to M-1
0 … N_F-1
d_data(ch1)
d_data(ch2)
thread
Conjugated multiplication
PHAT weighting
end
CUFFT(inverse)
0 … $_MC_2$-1
d_ifft

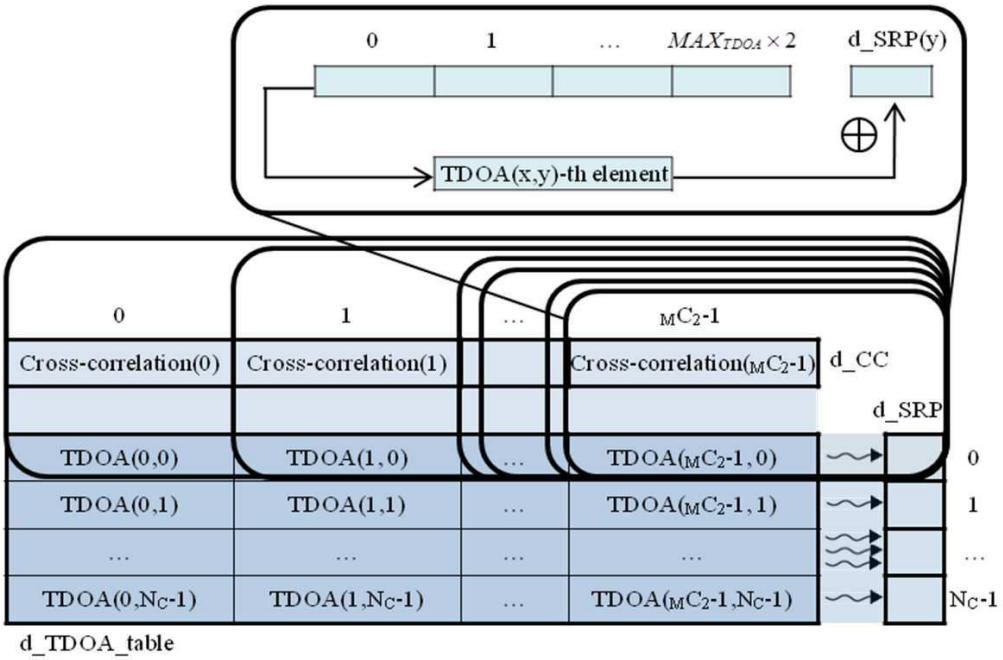
도면8


Grid(N_F,1)
Block(0,0)
Block(N_F-1,0)
Block(M,1)
Thread(0,0)
Thread(M-1,0)
(a)
Grid(N_F/N_W,1)
Block(0,0)
Block(N_F/N_W-1,0)
Block(N_W,1)
Thread(0,0)
Thread(N_W-1,0)
(b)
Grid($_MC_2$,1)
Block(0,0)
Block($_MC_2$-1,0)
Block(MAX_{TDOA},1)
Thread(0,0)
Thread(MAX_{TDOA}-1,0)
(c)

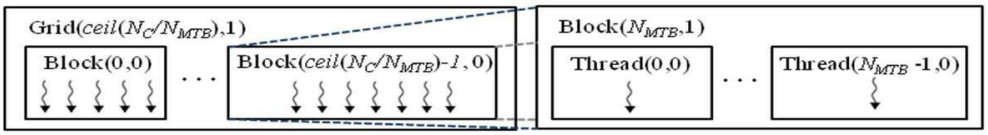
도면9

```
N_F = FRAME SIZE, N_W = WARP SIZE
BX = BLOCKIDX.X, TX = THREADIDX.X
PAIR_SIZE = MAX_TDOA*2+1
WINDOW<<< N_F, M >>>( D_DATA, D_WINDOW)
    D_DATA[TX*N_F+BX] = D_WINDOW[BX] * D_DATA[TX*N_F+BX]
END
CUFFT( M, D_DATA, FORWARD )
PI=0;
FOR CH1 = 0 TO M-2
  FOR CH2 = CH1+1 TO M-1
   MULWITHPHAT<<< N_F/N_W, N_W>>>(D_DATA, CH1, CH2, PI, D_IFFT)
        MUL = CPLXMUL( D_DATA[CH1*N_F + BX*N_W + TX],
                  CPLXCONJ(D_DATA[CH2*N_F + BX*N_W + TX]) )
            MUL = MUL/CPLXABS(MUL)
            D_IFFT[PI*N_F + BX*N_W + TX] = MUL
   END
   PI = PI+1
  END
END
CUFFT( M, D_IFFT, INVERSE)
COPYCROSSCORRELATION<<< MC2, MAX_TDOA>>>( D_TDOA_TABLE,
                                D_CC, D_SRP)
    D_CC[BX*PAIR_SIZE + TX] = D_IFFT[BX*N_F + N_F-MAX_TDOA + TX].X
    D_CC[BX*PAIR_SIZE + MAX_TDOA+1] = D_IFFT[BX*N_F].X
    IF TX==0
        D_CC[BX*PAIR_SIZE + MAX_TDOA] = D_IFFT[BX*N_F].X
    END
END
```

도면10



도면11



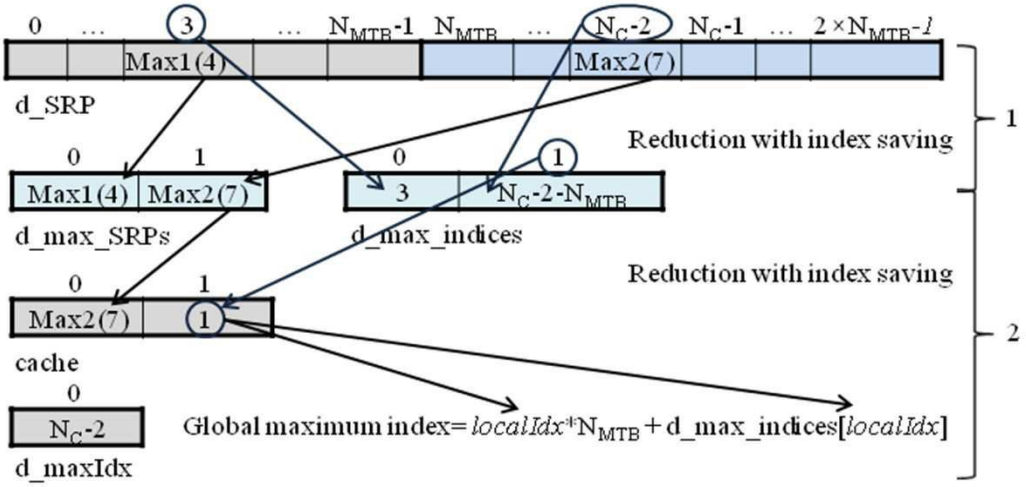
도면12

M= THE NUMBER OF MICROPHONES
N_C= THE NUMBER OF CANDIDATE COORDINATES
N_{MTB}= THE NUMBER OF MAXIMUM THREADS PER BLOCK
PAIR_SIZE= MAX_{TDOA}*2+1
BX= THREADIDX.X, TX= THREADIDX.Y
__SYNCTHREADS() : WAIT UNTIL ALL THREADS ARE FINISHED
CEIL(): ROUND UP

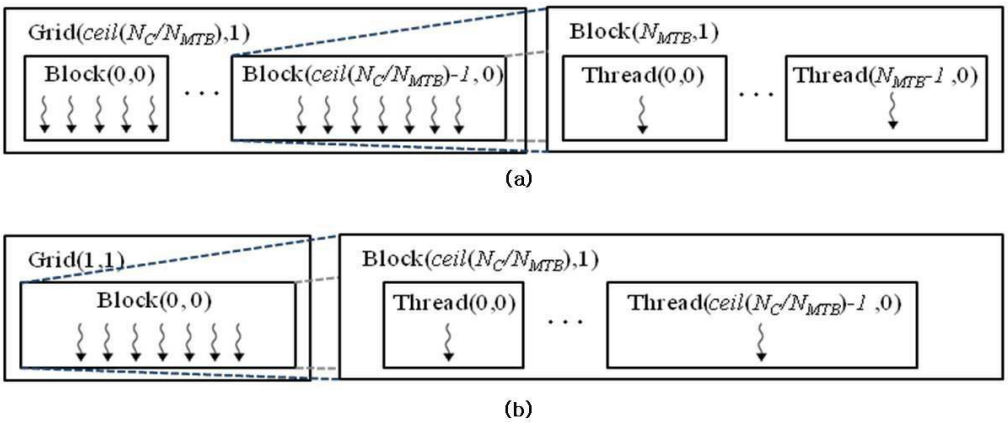
```
CALCSRP<<<CEIL(NC/NMTB), NMTB>>>( D_TDOA_TABLE,  D_CC,  D_SRP)
    __SHARED__ CC[ PAIR_SIZE*MC2]
    IF TX < MC2
        FOR I= 0 TO PAIR_SIZE
            CC[ TX*PAIR_SIZE+I]= D_CC[ TX*PAIR_SIZE+I];
        END
    END
  __SYNCTHREADS()

    IF BX AND TX EXCEED VALID RANGE
        STOP
    END
    FOR I= 0 TO MC2
        TDOA= D_TDOA_TABLE[ BX*MC2*NMTB + TX*MC2 + I]
        SRP= SRP + CC[ I*PAIR_SIZE + TDOA]
    END
    D_SRP[ BX*NMTB + TX] = SRP;
END
```

도면13


0 … 3 … N_MTB-1 N_MTB … N_C-2 N_C-1 … 2×N_MTB-1
Max1(4)
Max2(7)
d_SRP
Reduction with index saving
1
0 1
Max1(4) Max2(7)
d_max_SRPs
0 1
3 N_C-2-N_MTB
d_max_indices
Reduction with index saving
0 1
Max2(7) 1
cache
0
N_C-2
d_maxIdx
Global maximum index= localIdx*N_MTB + d_max_indices[localIdx]
2

도면14


Grid(ceil(N_C/N_MTB),1)
Block(0,0)
Block(ceil(N_C/N_MTB)-1,0)
Block(N_MTB,1)
Thread(0,0)
Thread(N_MTB-1,0)
(a)
Grid(1,1)
Block(0,0)
Block(ceil(N_C/N_MTB),1)
Thread(0,0)
Thread(ceil(N_C/N_MTB)-1,0)
(b)

도면15

```
N_C = THE NUMBER OF CANDIDATE COORDINATES
N_MTB = THE NUMBER OF MAXIMUM THREADS PER BLOCK
BX = BLOCKIDX.X, TX = THREADIDX.X
CEIL(): ROUND UP
__SYNCTHREADS() : WAIT UNTIL ALL THREADS ARE FINISHED
FINDMAX<<<CEIL(N_C/N_MTB), N_MTB>>>(D_SRP,D_MAX_SRPS,D_MAX_INDICES, D_MAXIDX)
    __SHARED__ CACHE[ N_MTB]
    IF BX AND TX EXCEED VALID RANGE
        STOP
    END
    CACHE[TX] = D_SRP[ BX*N_MTB + TX]
    __SYNCTHREADS()
    I = N_MTB/2
    WHILE I != 0                                  /// REDUCTION WITH INDEX-SAVING
        IF TX < I
            IF CACHE[TX] < CACHE[TX+I]
                CACHE[TX] = CACHE[TX+I]
                CACHE[TX+I] = TX+I
            ELSE
                CACHE[TX+I] = TX
            END
        END
    END
    __SYNCTHREADS()
    I/=2
    IF TX==0
        D_MAX_SRPS[BX] = CACHE[0]
    D_MAX_INDICIES[BX] = DE-REFERENCED MAXIMUM INDEX FROM CACHE
    END
END
FINDMAX2<<<1, CEIL(N_C/N_MTB)>>>(D_MAX_INDICES,D_MAX_SRPS,D_MAXIDX)
    __SHARED__ CACHE[CEIL(N_C/N_MTB)]
    CACHE[TX] = D_MAX_SRPS[TX]
    __SYNCTHREADS()
    I = CEIL(N_C/N_MTB)/2
    REDUCTION WITH INDEX-SAVING
    __SYNCTHREADS()
    I/=2
    IF TX==0
        GET LOCALIDX FROM CACHE BY RECURSIVE DE-REFERENCING
        D_MAXIDX[BX] = LOCALIDX*N_MTB + D_MAX_INDICES[LOCALIDX]
    END
END
```