



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2018-0102450
(43) 공개일자 2018년09월17일

(51) 국제특허분류(Int. Cl.)

G06F 17/11 (2006.01)

(52) CPC특허분류

G06F 17/11 (2013.01)

(21) 출원번호 10-2017-0029103

(22) 출원일자 2017년03월07일

심사청구일자 2017년03월07일

(71) 출원인

울산과학기술원

울산광역시 울주군 언양읍 유니스트길 50

(72) 발명자

정원기

울산광역시 울주군 언양읍 유니스트길 50

홍수민

울산광역시 울주군 언양읍 유니스트길 50

(74) 대리인

제일특허법인(유)

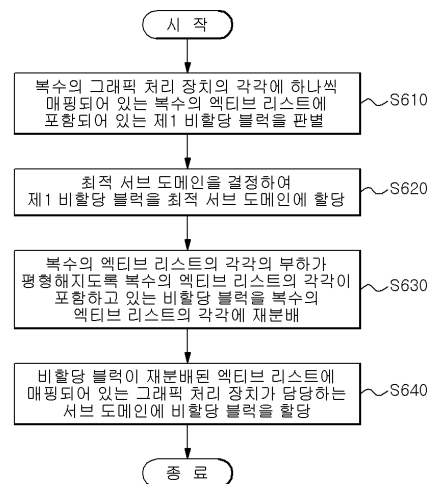
전체 청구항 수 : 총 11 항

(54) 발명의 명칭 FIM의 동적 서브 도메인 확장 방법 및 장치

(57) 요약

본 발명의 일 실시예에 따른 FIM의 동적 서브 도메인 확장 방법은 비할당 블록 중 복수의 그래픽 처리 장치의 각각에 하나씩 매핑되어 있는 복수의 액티브 리스트에 포함되어 있는 제1 비할당 블록을 판별하는 단계 및 복수의 블록 중 제1 비할당 블록과 인접하고 있는 블록이 속하는 서브 도메인을 기초로 복수의 서브 도메인 중 최적 서브 도메인을 결정하여 제1 비할당 블록을 최적 서브 도메인에 할당하는 단계를 포함한다.

대표도 - 도6



명세서

청구범위

청구항 1

아이코날 방정식이 계산될 도메인이 포함하는 복수의 블록 중 일부가 할당된 복수의 서브 도메인을 하나씩 담당하는 복수의 그래픽 처리 장치를 사용하는 컴퓨터 상에서 수행되고, 상기 복수의 그래픽 처리 장치가 상기 복수의 서브 도메인에 대하여 FIM(fast iterative method)을 수행한 후, 상기 복수의 블록 중 상기 복수의 서브 도메인에 할당되지 않은 상태인 비할당 블록을 상기 복수의 서브 도메인에 새로 할당시키는 FIM의 동적 서브 도메인 확장 방법에 있어서,

상기 비할당 블록 중 상기 복수의 그래픽 처리 장치의 각각에 하나씩 매핑되어 있는 복수의 액티브 리스트(active list)에 포함되어 있는 제1 비할당 블록을 판별하는 단계; 및

상기 복수의 블록 중 상기 제1 비할당 블록과 인접하고 있는 블록이 속하는 서브 도메인을 기초로 상기 복수의 서브 도메인 중 최적 서브 도메인을 결정하여 상기 제1 비할당 블록을 상기 최적 서브 도메인에 할당하는 단계를 포함하는

FIM의 동적 서브 도메인 확장 방법.

청구항 2

제1항에 있어서,

상기 비할당 블록에 상기 복수의 서브 도메인에 할당되지 않은 상태임을 나타내는 비할당 식별 정보를 매핑시켜 메모리에 저장된 도메인 매핑 테이블(domain mapping table)에 기록하는 단계; 및

상기 비할당 블록이 상기 최적 서브 도메인에 할당되는 경우 상기 최적 서브 도메인의 식별 정보를 매핑시켜 상기 도메인 매핑 테이블에 기록하는 단계를 더 포함하고,

상기 비할당 블록은 상기 도메인 매핑 테이블을 참조하여 판별되는

FIM의 동적 서브 도메인 확장 방법.

청구항 3

제1항에 있어서,

상기 제1 비할당 블록을 상기 최적 서브 도메인에 할당하는 단계는,

상기 제1 비할당 블록과 인접하고 있는 블록이 속하는 서브 도메인 중 같은 서브 도메인의 개수가 가장 많은 제1 서브 도메인을 상기 최적 서브 도메인을 결정하는 단계를 포함하는

FIM의 동적 서브 도메인 확장 방법.

청구항 4

제1항에 있어서,

상기 제1 비할당 블록을 상기 최적 서브 도메인에 할당하는 단계는,

상기 제1 비할당 블록과 인접하고 있는 블록이 속하는 서브 도메인 중 같은 서브 도메인의 개수가 가장 많은 제1 서브 도메인 및 상기 제1 비할당 블록과 인접하고 있는 블록이 속하는 서브 도메인을 담당하는 그래픽 처리 장치에 매핑된 액티브 리스트의 예상 부하를 기초로 상기 최적 서브 도메인을 결정하는 단계를 포함하는

FIM의 동적 서브 도메인 확장 방법.

청구항 5

제4항에 있어서,

상기 제1 비할당 블록을 상기 최적 서브 도메인에 할당하는 단계는,

상기 예상 부하를 계산하는 단계;

상기 액티브 리스트 중 상기 예상 부하가 가장 낮은 제2 액티브 리스트의 예상 부하에 대한 상기 제1 서브 도메인을 담당하는 그래픽 처리 장치에 매핑된 제1 액티브 리스트의 예상 부하의 비율을 계산하는 단계; 및

상기 비율이 소정의 임계치 이하이면 상기 제1 서브 도메인에 상기 제1 비할당 블록을 할당하고, 상기 비율이 소정의 임계치 이상이면 상기 제2 액티브 리스트와 매핑된 그래픽 처리 장치가 담당하는 제2 서브 도메인에 상기 제1 비할당 블록을 할당하는 단계를 포함하는

FIM의 동적 서브 도메인 확장 방법.

청구항 6

제5항에 있어서,

상기 예상 부하는,

상기 액티브 리스트가 포함하는 블록의 개수에 상기 액티브 리스트의 증가 상수를 곱하여 계산하는

FIM의 동적 서브 도메인 확장 방법.

청구항 7

제1항에 있어서,

상기 복수의 액티브 리스트의 각각이 포함하는 블록의 개수를 기초로 상기 복수의 액티브 리스트의 각각의 부하가 평형해지도록 상기 복수의 액티브 리스트의 각각이 포함하고 있는 상기 비할당 블록을 상기 복수의 액티브 리스트의 각각에 재분배하는 단계; 및

상기 비할당 블록이 재분배된 액티브 리스트에 매핑되어 있는 그래픽 처리 장치가 담당하는 서브 도메인에 상기 비할당 블록을 할당하는 단계를 더 포함하는

FIM의 동적 서브 도메인 확장 방법.

청구항 8

제7항에 있어서,

상기 비할당 블록을 상기 복수의 액티브 리스트의 각각에 재분배하는 단계는,

상기 복수의 액티브 리스트의 각각이 포함하는 블록의 개수의 표준 편차가 소정의 임계치 이하가 되도록 재분배하는

FIM의 동적 서브 도메인 확장 방법.

청구항 9

아이코날 방정식이 계산될 도메인이 포함하는 복수의 블록 중 일부가 할당된 복수의 서브 도메인을 하나씩 담당

하는 복수의 그래픽 처리 장치가 상기 복수의 서버 도메인에 대하여 FIM(fast iterative method)을 수행한 후, 상기 복수의 블록 중 상기 복수의 서버 도메인에 할당되지 않은 상태인 비할당 블록을 상기 복수의 서버 도메인에 새로 할당시키는 FIM의 동적 서버 도메인 확장 장치에 있어서,

상기 비할당 블록 중 상기 복수의 그래픽 처리 장치의 각각에 하나씩 매핑되어 있는 복수의 액티브 리스트(active list)에 포함되어 있는 제1 비할당 블록을 판별하는 판별부; 및

상기 복수의 블록 중 상기 제1 비할당 블록과 인접하고 있는 블록이 속하는 서버 도메인을 기초로 상기 복수의 서버 도메인 중 최적 서버 도메인을 결정하여 상기 제1 비할당 블록을 상기 최적 서버 도메인에 할당하는 할당부를 포함하는

FIM의 동적 서버 도메인 확장 장치.

청구항 10

제1항 내지 제8항의 방법을 프로세서가 수행하도록 하는 컴퓨터 판독 가능 기록매체에 저장된 프로그램.

청구항 11

제1항 내지 제8항의 방법을 프로세서가 수행하게 하는 명령어를 포함하는 프로그램이 기록된 컴퓨터 판독 가능 기록매체.

발명의 설명

기술 분야

[0001] 본 발명은 FIM(fast iterative method)의 동적 서버 도메인 확장 방법 및 장치에 관한 것으로서, 보다 자세하게는 복수의 그래픽 연산 장치를 사용하는 컴퓨터 상에서 FIM 모델을 이용한 아이코날 방정식(eikonal equation)을 계산함에 있어서 각 그래픽 연산 장치가 담당하는 서버 도메인을 동적으로 확장하는 FIM의 동적 서버 도메인 확장 방법 및 장치에 관한 것이다.

배경 기술

- [0003] 아이코날 방정식은 두 점간 광학적 최단 거리 또는 기하학적 최단 거리를 도출할 수 있도록 하는 방정식이다.
- [0004] 아이코날 방정식을 컴퓨터 상에서 수행하기 위해 가장 널리 쓰이던 FMM(fast marching method)은 아이코날 방정식을 풀기 위하여 그리드 노드(grid node) 간의 종속성에 따라 계산을 수행함으로써, 중복 계산의 가능성을 원천 봉쇄하였지만 종속성에 따라 그리드 노드를 순차적으로 계산하여야 한다는 특성상 복수의 스레드가 병렬적으로 계산을 수행하도록 하기 어려운 문제가 있었다.
- [0005] 이를 해결하기 위하여 FIM(fast iterative method)은 아이코날 방정식의 병렬 계산을 위해 그리드 노드간의 종속성과 상관없이 액티브 리스트(Active List)에 속한 그리드 노드들의 병렬적인 계산 방법을 제시하였다. (Won-Ki Jeong and Ross T. Whitaker. A fast iterative method for eikonal equations. SIAM J. Sci. Comput. Vol.30 No.5, pages 2512-2534, 2008.)
- [0006] 한편 기존에 복수의 그래픽 연산 장치를 사용하는 컴퓨터에서 FIM을 사용하는 경우 전체 도메인을 영역이 변하지 않는 여러 개의 서버 도메인으로 나눈 뒤 각각의 그래픽 연산 장치가 서로 다른 서버 도메인을 계산하도록 하였다.
- [0007] 그러나 이처럼 서버 도메인을 정적으로 분할하는 경우 FIM의 특성상 연산이 수행되는 데이터의 위치와 양이 동적으로 변하기 때문에 각 그래픽 연산 장치에 부하의 불균형이 발생한다. 이로 인하여 병렬적인 계산이 요구되는 FIM의 특성상 높은 병렬 효율을 기대할 수 없다.

선행기술문헌

특허문헌

[0009] (특허문헌 0001) 미국등록특허 제8,762,442호(2011.02.24 공고)

발명의 내용

해결하려는 과제

[0010] 본 발명의 실시예에서 해결하고자 하는 과제는 복수의 그래픽 연산 장치를 사용하는 컴퓨터에서 FIM을 사용하는 경우 각 그래픽 연산 장치가 담당하는 서브 도메인을 동적으로 확장하여 부하의 불균형을 방지하는 기술을 제공하는 것이다.

[0011] 다만, 본 발명의 실시예가 이루고자 하는 기술적 과제는 이상에서 언급한 과제로 제한되지 않으며, 이하에서 설명할 내용으로부터 통상의 기술자에게 자명한 범위 내에서 다양한 기술적 과제가 도출될 수 있다.

과제의 해결 수단

[0013] 아이코날 방정식이 계산될 도메인이 포함하는 복수의 블록 중 일부가 할당된 복수의 서브 도메인을 하나씩 담당하는 복수의 그래픽 처리 장치를 사용하는 컴퓨터 상에서 수행되고, 상기 복수의 그래픽 처리 장치가 상기 복수의 서브 도메인에 대하여 FIM(fast iterative method)을 수행한 후, 상기 복수의 블록 중 상기 복수의 서브 도메인에 할당되지 않은 상태인 비할당 블록을 상기 복수의 서브 도메인에 새로 할당시키는 본 발명명의 일 실시예에 따른 FIM의 동적 서브 도메인 확장 방법은 상기 비할당 블록 중 상기 복수의 그래픽 처리 장치의 각각에 하나씩 매핑되어 있는 복수의 액티브 리스트(active list)에 포함되어 있는 제1 비할당 블록을 판별하는 단계 및 상기 복수의 블록 중 상기 제1 비할당 블록과 인접하고 있는 블록이 속하는 서브 도메인을 기초로 상기 복수의 서브 도메인 중 최적 서브 도메인을 결정하여 상기 제1 비할당 블록을 상기 최적 서브 도메인에 할당하는 단계를 포함한다.

[0014] 이때 상기 비할당 블록에 상기 복수의 서브 도메인에 할당되지 않은 상태임을 나타내는 비할당 식별 정보를 매핑시켜 메모리에 저장된 도메인 매핑 테이블(domain mapping table)에 기록하는 단계 및 상기 비할당 블록이 상기 최적 서브 도메인에 할당되는 경우 상기 최적 서브 도메인의 식별 정보를 매핑시켜 상기 도메인 매핑 테이블에 기록하는 단계를 더 포함하고, 상기 비할당 블록은 상기 도메인 매핑 테이블을 참조하여 판별될 수 있다.

[0015] 또한 상기 제1 비할당 블록을 상기 최적 서브 도메인에 할당하는 단계는 상기 제1 비할당 블록과 인접하고 있는 블록이 속하는 서브 도메인 중 같은 서브 도메인의 개수가 가장 많은 제1 서브 도메인을 상기 최적 서브 도메인을 결정하는 단계를 포함할 수 있다.

[0016] 더하여 상기 제1 비할당 블록을 상기 최적 서브 도메인에 할당하는 단계는 상기 제1 비할당 블록과 인접하고 있는 블록이 속하는 서브 도메인 중 같은 서브 도메인의 개수가 가장 많은 제1 서브 도메인 및 상기 제1 비할당 블록과 인접하고 있는 블록이 속하는 서브 도메인을 담당하는 그래픽 처리 장치에 매핑된 액티브 리스트의 예상 부하를 기초로 상기 최적 서브 도메인을 결정하는 단계를 포함할 수 있다.

[0017] 한편 상기 제1 비할당 블록을 상기 최적 서브 도메인에 할당하는 단계는 상기 예상 부하를 계산하는 단계, 상기 액티브 리스트 중 상기 예상 부하가 가장 낮은 제2 액티브 리스트의 예상 부하에 대한 상기 제1 서브 도메인을 담당하는 그래픽 처리 장치에 매핑된 제1 액티브 리스트의 예상 부하의 비율을 계산하는 단계 및 상기 비율이 소정의 임계치 이하이면 상기 제1 서브 도메인에 상기 제1 비할당 블록을 할당하고, 상기 비율이 소정의 임계치 이상이면 상기 제2 액티브 리스트와 매핑된 그래픽 처리 장치가 담당하는 제2 서브 도메인에 상기 제1 비할당 블록을 할당하는 단계를 포함할 수 있다.

[0018] 아울러 상기 예상 부하는 상기 액티브 리스트가 포함하는 블록의 개수에 상기 액티브 리스트의 증가 상수를 곱하여 계산할 수 있다.

[0019] 또한 상기 복수의 액티브 리스트의 각각이 포함하는 블록의 개수를 기초로 상기 복수의 액티브 리스트의 각각의

부하가 평형해지도록 상기 복수의 액티브 리스트의 각각이 포함하고 있는 상기 비할당 블록을 상기 복수의 액티브 리스트의 각각에 재분배하는 단계 및 상기 비할당 블록이 재분배된 액티브 리스트에 매핑되어 있는 그래픽 처리 장치가 담당하는 서브 도메인에 상기 비할당 블록을 할당하는 단계를 더 포함할 수 있다.

[0020] 이때 상기 비할당 블록을 상기 복수의 액티브 리스트의 각각에 재분배하는 단계는 상기 복수의 액티브 리스트의 각각이 포함하는 블록의 개수의 표준 편차가 소정의 임계치 이하가 되도록 재분배할 수 있다.

[0021] 아이코날 방정식이 계산될 도메인이 포함하는 복수의 블록 중 일부가 할당된 복수의 서브 도메인을 하나씩 담당하는 복수의 그래픽 처리 장치가 상기 복수의 서브 도메인에 대하여 FIM(fast iterative method)을 수행한 후, 상기 복수의 블록 중 상기 복수의 서브 도메인에 할당되지 않은 상태인 비할당 블록을 상기 복수의 서브 도메인에 새로 할당시키는 본 발명의 일 실시예에 따른 FIM의 동적 서브 도메인 확장 장치는 상기 비할당 블록 중 상기 복수의 그래픽 처리 장치의 각각에 하나씩 매핑되어 있는 복수의 액티브 리스트(active list)에 포함되어 있는 제1 비할당 블록을 판별하는 판별부 및 상기 복수의 블록 중 상기 제1 비할당 블록과 인접하고 있는 블록이 속하는 서브 도메인을 기초로 상기 복수의 서브 도메인 중 최적 서브 도메인을 결정하여 상기 제1 비할당 블록을 상기 최적 서브 도메인에 할당하는 할당부를 포함한다.

발명의 효과

[0023] 본 발명의 실시예에 따르면, 복수의 그래픽 연산 장치를 사용하는 컴퓨터에서 FIM을 사용하는 경우 FIM의 데이터 연산에 맞추어 각 그래픽 연산 장치가 담당하는 서브 도메인을 동적으로 확장하여 각 그래픽 연산 장치간의 부하의 불균형을 줄일 수 있으며, 각 서브 도메인의 중첩 영역을 최소화하여 데이터 동기화의 비용을 줄일 수 있다.

도면의 간단한 설명

[0025] 도 1은 FIM의 계산 과정을 설명하기 위한 예시도이다.

도 2는 기존 복수의 그래픽 연산 장치를 사용하는 컴퓨터 상에서 FIM 모델을 이용하는 경우 아이코날 방정식이 계산될 도메인을 영역이 변하지 않는 여러 개의 서브 도메인으로 나누는 것을 설명하기 위한 예시도이다.

도 3은 서브 도메인을 정적으로 할당하는 경우 각 서브 도메인을 담당하는 그래픽 연산 장치에서 발생하는 부하의 불균형을 설명하기 위한 예시도이다.

도 4는 서브 도메인 간의 중첩영역에서 발생하는 데이터 동기화를 설명하기 위한 예시도이다.

도 5는 본 발명의 일 실시예에 따른 FIM의 동적 서브 도메인 확장 장치의 기능 블록도이다.

도 6은 본 발명의 일 실시예에 따른 FIM의 동적 서브 도메인 확장 방법의 프로세스를 도시하는 흐름도이다.

도 7은 본 발명의 일 실시예에 따라 비할당 블록과 인접하고 있는 블록이 속하는 서브 도메인을 기초로 최적 서브 도메인을 결정하는 것을 설명하기 위한 예시도이다.

발명을 실시하기 위한 구체적인 내용

[0026] 본 발명의 목적과 기술적 구성 및 그에 따른 작용 효과에 관한 자세한 사항은 본 발명의 명세서에 첨부된 도면에 의거한 이하의 상세한 설명에 의해 보다 명확하게 이해될 것이다. 첨부된 도면을 참조하여 본 발명에 따른 실시예를 상세하게 설명한다.

[0027] 본 명세서에서 개시되는 실시예들은 본 발명의 범위를 한정하는 것으로 해석되거나 이용되지 않아야 할 것이다. 이 분야의 통상의 기술자에게 본 명세서의 실시예를 포함한 설명은 다양한 응용을 갖는다는 것이 당연하다. 따라서, 본 발명의 상세한 설명에 기재된 임의의 실시예들은 본 발명을 보다 잘 설명하기 위한 예시적인 것이며 본 발명의 범위가 실시예들로 한정되는 것을 의도하지 않는다.

[0028] 도면에 표시되고 아래에 설명되는 기능 블록들은 가능한 구현의 예들일 뿐이다. 다른 구현들에서는 상세한 설명의 사상 및 범위를 벗어나지 않는 범위에서 다른 기능 블록들이 사용될 수 있다. 또한, 본 발명의 하나 이상의 기능 블록이 개별 블록들로 표시되지만, 본 발명의 기능 블록들 중 하나 이상은 동일 기능을 실행하는 다양한

하드웨어 및 소프트웨어 구성들의 조합일 수 있다.

[0029] 또한, 어떤 구성 요소들을 포함한다는 표현은 개방형의 표현으로서 해당 구성 요소들이 존재하는 것을 단순히 지칭할 뿐이며, 추가적인 구성 요소들을 배제하는 것으로 이해되어서는 안 된다.

[0030] 나아가 어떤 구성 요소가 다른 구성 요소에 연결되어 있다거나 접속되어 있다고 언급될 때에는, 그 다른 구성 요소에 직접적으로 연결 또는 접속되어 있을 수도 있지만, 중간에 다른 구성 요소가 존재할 수도 있다고 이해되어야 한다.

[0031] 또한 '제1, 제2' 등과 같은 표현은 복수의 구성들을 구분하기 위한 용도로만 사용된 표현으로써, 구성들 사이의 순서나 기타 특징들을 한정하지 않는다.

[0032] 이하에서는 도면들을 참조하여 본 발명의 실시예들에 대해 설명하도록 한다.

[0033] 아이코날 방정식은 두 점간 광학적 최단 거리 또는 기하학적 최단 거리를 도출할 수 있도록 하는 방정식으로 아래 수학식 1과 같다.

수학식 1

$$H(\mathbf{x}, \nabla \phi) = |\nabla \phi(\mathbf{x})|^2 - \frac{1}{f^2(\mathbf{x})} = 0 \quad \forall \mathbf{x} \in \Omega,$$

[0034]

[0035] 이때 Ω 은 계산을 수행할 대상이 되는 도메인을 의미하며, $\emptyset$ 는 도메인 상에서 특정 지점까지의 거리를 계산할 기준이 되는 지점이고, $\mathbf{x}$ 는 도메인 상에서의 특정 지점으로, $\nabla \phi(\mathbf{x})$ 는 지점 $\emptyset$ 로부터 지점 $\mathbf{x}$ 까지의 최단 거리를 의미하며, $f(\mathbf{x})$ 는 Ω 에서의 속도 함수이고, $H(\mathbf{x}, \nabla \phi(\mathbf{x}))$ 를 0으로 두었을 때 위 방정식의 해 $\nabla \phi(\mathbf{x})$ 를 구하여 지점 $\emptyset$ 로부터 지점 $\mathbf{x}$ 까지의 최단 거리를 구할 수 있다.

[0036] 이러한 아이코날 방정식을 컴퓨터 상에서 수행하기 위해서는 연속적으로 이루어진 도메인을 불연속적으로 나누어야 하기 때문에, 예를 들면 기준에 사용되는 Godunov upwind difference scheme을 통해 도메인을 불연속적으로 이루어진 그리드 노드(grid node)로 분할할 수 있다.

[0037] Godunov upwind difference scheme을 통해 도메인이 불연속적인 그리드 노드로 분할된 경우 도메인이 3차원이라면 아이코날 방정식은 아래 수학식 2와 같이 나타낼 수 있다.

수학식 2

$$g(\mathbf{x}) = \left[\frac{(U(\mathbf{x}) - U(\mathbf{x})^{\text{xmin}})^+}{h_x} \right]^2 + \left[\frac{(U(\mathbf{x}) - U(\mathbf{x})^{\text{ymin}})^+}{h_y} \right]^2 + \left[\frac{(U(\mathbf{x}) - U(\mathbf{x})^{\text{zmin}})^+}{h_z} \right]^2 - \frac{1}{f(\mathbf{x})^2},$$

[0038]

[0039] 이때 $U(\mathbf{x})$ 는 도메인 상에서 거리를 계산할 기준이 되는 지점 $\emptyset$ 로부터 그리드 노드 $\mathbf{x}=(i, j, k)$ 까지의 최단 거리를 의미하며, $U(\mathbf{x})^{\text{pmin}}$ 은 p축($p \in \{x, y, z\}$)상에서 $U(\mathbf{x})$ 와 인접한 두 개의 그리드 노드 중 더 짧은 거리에 있는 그리드 노드의 $U(\mathbf{x})$ 값을 의미한다. 또한 h_p 는 p축에 있어서 그리드 노드 간의 공간이며, $f(\mathbf{x})$ 는 속도 함수이고, $(n)^+ = \max(n, 0)$ 을 의미한다.

[0040] FIM(fast iterative method)은 아이코날 방정식을 컴퓨터 상에서 병렬적으로 계산하기 위해 액티브 리스트(active list)를 사용한다. 액티브 리스트에는 아이코날 방정식을 계산하고 있거나 계산해야 할 그리드 노드의 정보를 글로벌 메모리(global memory)에 저장하여, 복수의 스레드들이 액티브 리스트를 참조하여 액티브 리스트에 포함된 그리드 노드에 대해 아이코날 방정식을 계산함으로써 병렬적인 연산이 가능하도록 한다.

[0041] 도 1은 FIM의 계산 과정을 설명하기 위한 예시도이다.

[0042] 도 1의 (a)에 도시된 바와 같이, FIM 모델은 도메인 상에서 거리를 계산할 기준이 되는 씨드 노드(seed node)(101)와 인접한 그리드 노드(102, 103)를 액티브 리스트에 포함시켜 첫번째 연산을 통해 액티브 리스트에 포함된 그리드 노드의 아이코날 방정식을 계산하고, 도 1b와 같이 첫번째 연산이 끝나면 첫번째 연산에서 계산된 그리드 노드(102, 103)와 인접한 그리드 노드(104, 105, 106)를 액티브 리스트에 추가하여 액티브 리스트에 있는 그리드 노드에 대하여 아이코날 방정식을 계산하도록 한다. 이때 아이코날 방정식의 계산을 통해 특정 값에 수렴하게 되는 그리드 노드는 액티브 리스트에서 제외시키며, 이 과정을 모든 그리드 노드가 수렴하게 될 때까지 반복하여 아이코날 방정식의 계산을 완료한다.

[0043] 아래 표 1은 아이코날 방정식을 컴퓨터 상에서 계산하기 위한 FIM의 수도 코드를 나타낸 것이다.

표 1

FIM	
Input: Grid Ω , Solution U , Active list L	
/* Initialization */	
1	forall $x \in \Omega$ do
2	if x is a source node then
3	$U(x) \leftarrow 0$
4	add x to L
5	else
6	$U(x) \leftarrow \infty$
/* Compute new solutions for L */	
7	while L is not empty do
8	forall $x \in L$ do
9	$p \leftarrow U(x)$
10	$q \leftarrow$ solution of $g(x) = 0$
/* If not converged */	
11	if $p > q$ then
12	$U(x) \leftarrow q$
/* If converged */	
13	else
/* Check adjacent neighbor nodes for reactivation */	
14	forall x_{nb} adjacent to x do
15	if $U(x_{nb}) > U(x)$ and $x_{nb} \notin L$ then
16	$p \leftarrow U(x_{nb})$
17	$q \leftarrow$ solution of $g(x_{nb}) = 0$
18	if $p > q$ then
19	$U(x_{nb}) \leftarrow q$
20	add x_{nb} to L
21	remove x from L

[0044]

[0045] 표 1에 표기된 source node는 도메인 Ω 에서 특정 그리드 노드까지의 거리를 계산할 기준이 되는 씨드 노드를 의미한다. 따라서 씨드 노드 자체의 $U(x)$ 는 0 이며, 그 외의 그리드 노드의 아이코날 방정식의 해를 ∞ 로 초기 설정한 후, 씨드 노드로부터 인접한 그리드 노드를 액티브 리스트에 추가하면서 액티브 리스트에 포함된 그리드 노드에 대한 아이코날 방정식을 수행한다. 이때 이전에 계산한 아이코날 방정식의 해인 p 와 가장 최근 계산한 아이코날 방정식의 해 q 를 비교하여 특정 그리드 노드에서 $q-p < 0$ 인 경우 해당 그리드 노드에서의 아이코날 방정식의 해가 수렴한 것으로 간주하여 액티브 리스트에서 해당 그리드 노드를 제외하는 방식으로 모든 그리드 노드가 수렴할 때까지 반복하여 아이코날 방정식을 계산한다.

[0046] 도 2는 기존 복수의 그래픽 연산 장치를 사용하는 컴퓨터 상에서 FIM 모델을 이용하는 경우 아이코날 방정식이 계산될 도메인을 영역이 변하지 않는 여러 개의 서브 도메인으로 나누는 것을 설명하기 위한 예시도이다.

[0047] 도 2에 도시된 바와 같이, 기존에는 아이코날 방정식이 계산될 도메인을 다양한 방식으로 분할하여 복수 개의 서브 도메인으로 나눈 뒤 각각의 그래픽 연산 장치가 서로 다른 서브 도메인을 계산하도록 할 수 있다. 이때 각

서브 도메인에는 복수의 블록을 포함한다. 이때 블록은 복수의 그리드 노드의 집합으로 액티브 리스트는 블록을 저장하여 블록 단위로 FIM을 수행할 수 있다.

[0048] 즉, 하나의 블록 안에 포함된 그리드 노드에 대하여 도 1과 같이 FIM을 수행하고, 하나의 블록 안에 포함된 그리드 노드에 대한 FIM이 완료되면 인접하는 블록을 액티브 리스트에 추가하여 FIM을 수행할 수 있다. 이때 액티브 리스트는 그리드 노드 대신 계산 중이거나 계산이 되어야 할 블록에 대한 정보를 저장할 수 있다. 본 발명의 명세서에서는 블록 단위로 FIM을 수행하고 액티브 리스트에는 블록에 대한 정보가 저장되는 것을 기준으로 설명한다.

[0049] 한편, 도 2와 같이 서브 도메인을 정적으로 할당하는 경우 FIM의 특성상 연산이 수행되는 데이터의 위치와 양이 동적으로 변하기 때문에 각 그래픽 연산 장치에 부하의 불균형이 발생할 수 있으며, 이로 인하여 병렬 계산이 요구되는 FIM의 특성상 높은 병렬 효율을 기대할 수 없다. 도 3은 서브 도메인을 정적으로 할당하는 경우 각 서브 도메인을 담당하는 그래픽 연산 장치에서 발생하는 부하의 불균형을 설명하기 위한 예시도이다.

[0050] 예를 들어, 도 3의 (a)와 같이 4개의 사각형 모양으로 서브 도메인(310, 320, 330, 340)이 분할되어 있고, 각 서브 도메인 안에 2개의 블록이 각 액티브 리스트에 포함되어 있다고 가정한다. 이때 블록 간의 종속성 때문에 각 서브 도메인 마다 수행되는 데이터의 위치와 양이 다르므로, 블록간 종속도가 왼쪽 아래칸으로 향한다고 했을 때, 도 3의 (b)와 같이 서브 도메인 310에는 2개의 블록에 대한 연산이 끝나고 4개의 블록이 추가로 액티브 리스트에 추가될 수 있고, 서브 도메인 320 및 330에는 1개의 블록에 대한 연산이 끝나고 3개의 블록이 추가로 액티브 리스트에 추가될 수 있으며, 서브 도메인 340에는 여전히 2개의 블록에 대하여 연산 중일 수 있다. 이와 같은 경우, 도 3의 (c)와 같이 각 서브 도메인에서 연산이 완료된 블록 및 연산되어야 할 블록의 개수에 차이가 생겨, 각 그래픽 처리 장치 간에 부하의 불균형이 발생하고, 이로 인하여 병렬 계산이 요구되는 FIM의 특성상 높은 병렬 효율을 기대할 수 없게 된다.

[0051] 또한 도 4는 서브 도메인 간의 중첩영역에서 발생하는 데이터 동기화를 설명하기 위한 예시도이다. 복수의 그래픽 연산 장치를 사용하는 컴퓨터 상에서 FIM 모델을 이용하는 경우 서브 도메인은 각각 독립적으로 아이코날 방정식이 계산되므로 전체 도메인에 대하여 완벽하게 아이코날 방정식을 계산하기 위해서는 도 4와 같이 각 서브 도메인 간의 중첩 영역(halo)에서 동기화가 필요하다. 이때 각 서브 도메인 간 중첩 영역의 동기화에는 큰 연산량이 요구되기 때문에 중첩 영역을 최소화하는 것이 중요하다.

[0052] 이에, 본 발명의 실시예는 복수의 그래픽 연산 장치를 사용하는 컴퓨터에서 FIM을 사용하는 경우 각 그래픽 연산 장치가 담당하는 서브 도메인을 동적으로 확장하여 부하가 평형해지도록 하고, 각 서브 도메인의 중첩 영역을 최소화하는 기술로서 아래 도 5 내지 도 7과 함께 설명하도록 한다.

[0053] 도 5는 본 발명의 일 실시예에 따른 FIM의 동적 서브 도메인 확장 장치의 기능 블록도이다.

[0054] 본 발명의 일 실시예에 따른 FIM의 동적 서브 도메인 확장 장치는 아이코날 방정식이 계산될 도메인이 포함하는 복수의 블록 중 일부가 할당된 복수의 서브 도메인을 하나씩 담당하는 복수의 그래픽 처리 장치가 각 각각에 할당된 서브 도메인에 대하여 FIM을 수행한 후, 복수의 블록 중 복수의 서브 도메인에 할당되지 않은 상태인 비할당 블록을 복수의 서브 도메인에 새로 할당시켜 서브 도메인을 동적으로 확장시킬 수 있다.

[0055] 예를 들어, 최초의 FIM 수행시 복수의 그래픽 처리 장치의 각각에 하나씩 매핑되어 있는 복수의 액티브 리스트에는 아이코날 방정식에 의하여 도출될 거리의 기준이 되는 씨드 노드를 포함하는 블록만이 할당될 수 있으며, 이때 서브 도메인에 씨드 노드를 포함하는 블록만이 할당되도록 서브 도메인을 최초 분할할 수 있고, 각 서브 도메인에 대하여 FIM 계산이 완료되면 각 그래픽 처리 장치의 연산량을 고려하여 서브 도메인을 확장시킴으로써 전체 도메인에 대하여 아이코날 방정식의 계산이 완료될 때까지 FIM을 계속하여 수행하도록 할 수 있다.

[0056] 다만, 서브 도메인에 씨드 노드를 포함하는 블록만을 할당하는 것은 하나의 예시일 뿐, 서브 도메인에 속하지 않는 비할당 블록이 생기도록 다양한 방식에 따라 서브 도메인을 분할하고, 분할된 서브 도메인에 대하여 FIM이 수행된 이후에 서브 도메인을 확장시키는 방식이라면 본 발명의 실시예가 적용될 수 있다.

[0057] 도 5를 참조하면, 본 발명의 일 실시예에 따른 FIM의 동적 서브 도메인 확장 장치는 판별부 및 할당부를 포함한다.

[0058] 판별부는 비할당 블록 중 복수의 그래픽 처리 장치의 각각에 하나씩 매핑되어 있는 복수의 액티브 리스트에 포함되어 있는 제1 비할당 블록을 판별하고, 할당부는 복수의 블록 중 제1 비할당 블록과 인접하고 있는 블록이 속하는 서브 도메인을 기초로 복수의 서브 도메인 중 최적 서브 도메인을 결정하여 제1 비할당 블록을 최적 서

브 도메인에 할당한다.

- [0059] 이때 FIM의 동적 서브 도메인 확장 장치의 각 구성이 동작하는 구체적인 과정을 도 6과 함께 설명하기로 한다.
- [0060] 한편 상술한 실시예가 포함하는 판별부 및 할당부는 이들의 기능을 수행하도록 프로그램된 명령어를 포함하는 메모리, 및 이들 명령어를 수행하는 마이크로프로세서를 포함하는 연산 장치에 의해 구현될 수 있다.
- [0061] 도 6은 본 발명의 일 실시예에 따른 FIM의 동적 서브 도메인 확장 방법의 프로세스를 도시하는 흐름도이다. 도 6에 따른 FIM의 동적 서브 도메인 확장 방법의 각 단계는 도 5를 통해 설명된 FIM의 동적 서브 도메인 확장 장치에 의해 수행될 수 있다.
- [0062] 복수의 그래픽 연산 장치를 사용하는 컴퓨터 상에서 FIM 모델을 이용하여 아이코날 방정식을 계산하는 경우에는 도메인을 복수의 서브 도메인으로 나누어 각 그래픽 연산 장치가 하나의 서브 도메인에 대하여 FIM을 수행하도록 할 수 있고, 이를 위해 각 그래픽 연산 장치에는 FIM을 위한 액티브 리스트가 하나씩 매핑되어 있을 수 있다. 따라서 각 그래픽 연산 장치는 자신에게 매핑된 액티브 리스트를 참조하여 서브 도메인에 한하여 FIM을 수행한다. 이때 서브 도메인에는 아이코날 방정식이 수행될 그리드 노드가 포함된 복수의 블록을 포함하고, 액티브 리스트는 계산되어야 할 블록의 리스트를 저장하며, 그래픽 처리 장치는 글로벌 메모리(global memory)를 통해 액티브 리스트를 참조할 수 있다.
- [0063] 이때 본 발명의 일 실시예에 따른 FIM의 동적 서브 도메인 확장 방법은 복수의 그래픽 처리 장치가 각 그래픽 처리 장치에 할당된 서브 도메인에 대하여 FIM을 수행한 후, 복수의 블록 중 복수의 서브 도메인에 할당되지 않은 상태인 비할당 블록을 복수의 서브 도메인에 새로 할당시켜 서브 도메인을 동적으로 확장시킨다. 이때 본 발명의 일 실시예에 따른 FIM의 동적 서브 도메인 확장 방법은 각 그래픽 처리 장치의 부하 평형을 유지하고, 서브 도메인의 중첩 영역을 최소화하는 방향으로 서브 도메인을 확장시킬 수 있다.
- [0064] 우선, 판별부는 어느 서브 도메인에도 할당되지 않은 상태인 비할당 블록 중 어느 하나의 액티브 리스트에 포함되어 있는 제1 비할당 블록을 판별한다(S610).
- [0065] 이때 본 발명의 일 실시예에 따른 FIM의 동적 서브 도메인 확장 방법은 비할당 블록에 어느 서브 도메인에도 할당되지 않은 상태임을 나타내는 비할당 식별 정보를 매핑시켜 글로벌 메모리에 저장된 도메인 매핑 테이블(domain mapping table)에 기록하는 단계 및 비할당 블록이 특정 서브 도메인에 할당되는 경우 해당 서브 도메인의 식별 정보를 매핑시켜 도메인 매핑 테이블에 기록하는 단계를 포함할 수 있다. 이를 통해 S610 단계에서 판별부는 도메인 매핑 테이블을 참조하여 액티브 리스트에 포함된 비할당 블록을 판별할 수 있다.
- [0066] 한편, 복수의 그래픽 처리 장치가 자신에게 할당된 서브 도메인에 대하여 FIM을 완료한 이후 액티브 리스트는 도 1과 같이 계산이 완료된 블록에 근접하는 블록을 포함시키기 때문에, 서브 도메인과 인접하는 비할당 블록이 액티브 리스트에 포함될 수 있다. 따라서 액티브 리스트에 할당된 비할당 블록을 서브 도메인에 포함시켜 계산 영역을 확장해야 한다.
- [0067] 이를 위해, 할당부는 제1 비할당 블록과 인접하고 있는 블록이 속하는 서브 도메인을 기초로 최적 서브 도메인을 결정하여 제1 비할당 블록을 최적 서브 도메인에 할당한다(S620).
- [0068] 이때 도 7의 (a)를 참조하면 블록 701, 블록 702, 블록 703 과 같이 하나의 서브 도메인에만 인접하고 있는 블록의 경우 인접하는 서브 도메인에 할당하는 것이 계산의 종속성을 고려할 때 가장 이상적이므로 문제가 되지 않지만, 블록 704와 같이 두 개 이상의 서브 도메인에 인접하고 있는 블록의 경우 어느 서브 도메인에 해당 블록을 할당시켜야 부하의 평형 및 중첩 영역을 최소화할 수 있는지 문제가 된다.
- [0069] 이때 할당부는 제1 비할당 블록과 인접하고 있는 블록이 속하는 서브 도메인 중 같은 서브 도메인의 개수가 가장 많은 제1 서브 도메인을 최적 서브 도메인으로 결정할 수 있다.
- [0070] 다시 도 7의 (a)를 참조하면, 블록 704가 제1 비할당 블록에 해당하는 경우, 블록 704의 왼쪽 및 아래쪽의 2개의 블록은 제1 서브 도메인(710)에 속하며, 블록 704의 오른쪽의 1개의 블록은 제2 서브 도메인에 속하고, 블록 704의 위쪽 1개의 블록은 비할당 블록이다. 이때 도 7의 (b)는 블록 704를 제1 서브 도메인(710)에 할당한 모습이고, 도 7의 (c)는 블록 704를 제2 서브 도메인(720)에 할당한 모습이다.
- [0071] 도 7의 (b) 및 (c)에 도시된 바와 같이, 블록 704를 제1 서브 도메인(710)에 할당하게 되면 제1 서브 도메인(710)과 제2 서브 도메인(720)에서 중첩되는 영역이 3 지점이 발생하게 되고, 블록 704를 제2 서브 도메인(720)에 할당하게 되면 제1 서브 도메인(710)과 제2 서브 도메인(720)에서 중첩되는 영역이 4 지점이 발생하게 된

다. 따라서 블록 704이 인접하고 있는 블록들이 속하는 서브 도메인 중 개수가 가장 많은 제1 서브 도메인(710)에 할당하여 서브 도메인 간의 중첩 영역을 최소화할 수 있다.

[0072] 또한 할당부는 각 액티브 리스트의 부하가 평형해지도록 각 액티브 리스트가 포함하고 있는 비할당 블록을 재분배할 수 있다(S640).

[0073] 예를 들면, 할당부는 각 액티브 리스트가 포함하는 블록의 개수를 기초로 각 액티브 리스트의 각각의 부하가 평형해지도록 각 액티브 리스트가 포함하고 있는 비할당 블록을 재분배할 수 있다. 이를 위해 할당부는 각 액티브 리스트가 포함하는 블록의 개수의 표준 편차가 소정의 임계치 이하가 되도록 비할당 블록을 재분배할 수 있다.

[0074] 이후, 할당부는 비할당 블록이 재분배된 액티브 리스트에 매핑되어 있는 그래픽 처리 장치가 담당하는 서브 도메인에 재분배된 비할당 블록을 할당하여 서브 도메인을 확장할 수 있다(S640).

[0075] 한편, 상술한 실시예를 수도코드로 나타내면 아래 표 2와 같다.

표 2

Algorithm 1: DOMAIN ASSIGNMENT ALGORITHM

```

Input: Active List  $L$ , Domain Mapping Table  $T$ 
/*  $N$ : total number of sub-domains */
/*  $L_i$ : sub active list for sub-domain  $i$  ( $L = L_1 \cup L_2 \cup \dots \cup L_N$ ) */
/* 1. Block clustering for reducing halo communication */
/*  $P_i$  is the list of already assigned blocks in  $L_i$  */
/*  $Q_i$  is the list of unassigned blocks in  $L_i$  */
1 forall the  $i = 1$  to  $N$  do
2   forall the  $v \in L_i$  do
3     if  $T(v) = \text{unassigned}$  then
4        $j \leftarrow$  index of sub-domain adjacent to  $v$  that appears most
5       Insert  $v$  to  $Q_j$ 
6     else
7       Insert  $v$  to  $P_i$ 
8   /* 2. Load balancing between  $Q_1, \dots, Q_N$  */
9   if  $\text{stddev}(L_1, \dots, L_N) > \delta$  then
10    repeat
11      forall the  $i = 1$  to  $N$  do
12         $n_i \leftarrow$  size of  $P_i$  + size of  $Q_i$ 
13         $i_{\max} \leftarrow i$  if  $n_i \geq n_j$  for all  $j = 1, \dots, i-1$ 
14         $i_{\min} \leftarrow i$  if  $n_i \leq n_j$  for all  $j = 1, \dots, i-1$ 
15         $\Delta = n_{i_{\max}} - n_{i_{\min}}$ 
16        Load balancing by moving blocks from  $Q_{i_{\max}}$  to  $Q_{i_{\min}}$ 
17    until  $\Delta \leq \epsilon_1$ 
18  /* Update Domain Mapping Table and Collect Active List */
19  forall the  $i = 1$  to  $N$  do
20    forall the block  $v \in Q_i$  do
21       $T(v) \leftarrow i$ 
22     $L_i \leftarrow P_i \cup Q_i$ 
  
```

[0076]

[0077] 본 발명의 다른 실시예로 S620 단계에서 할당부는 제1 비할당 블록과 인접하고 있는 블록이 속하는 서브 도메인 중 같은 서브 도메인의 개수뿐만 아니라, 제1 비할당 블록과 인접하고 있는 블록이 속하는 서브 도메인을 담당하는 그래픽 처리 장치에 매핑된 액티브 리스트의 예상 부하를 추가적으로 고려하여 최적 서브 도메인을 결정할 수 있다.

[0078] 이를 위하여, S620 단계에서 할당부는 제1 비할당 블록과 인접하는 서브 도메인의 각 액티브 리스트에 대한 예상 부하를 계산할 수 있다(S621). 이때 예상 부하는 액티브 리스트가 포함하는 블록의 개수에 액티브 리스트의 증가 상수를 곱하여 계산할 수 있다. 이때 증가 상수는 액티브 리스트에 포함된 블록의 개수에 대한 비할당 블록의 개수의 비율, 또는 액티브 리스트에 포함된 블록의 개수에 대한 서브 도메인에 포함된 블록의 개수의 비율로 설정할 수 있으나, 이에 한정되지 않고 사용자의 설정에 따라 달라질 수 있다.

[0079] 이후, 액티브 리스트 중 예상 부하가 가장 낮은 제2 액티브 리스트의 예상 부하에 대한 제1 서브 도메인을 담당

하는 그래픽 처리 장치에 매핑된 제1 액티브 리스트의 예상 부하의 비율을 계산할 수 있다(S622). 이때 제2 액티브 리스트에 할당하게 되는 경우는 예상 부하가 낮은 액티브 리스트에 할당하게 되는 것이므로 부하의 평형 유지에 효과적이고, 제1 서브 도메인에 할당하게 되는 경우에는 도 7을 통해 상술한 바와 같이 중첩 영역의 최소화 효율적이다.

[0080] 따라서 S622 단계에서 계산된 비율이 소정의 임계치 이하이면 제1 및 제2 액티브 리스트 중 어느 액티브 리스트에 제1 비할당 블록을 할당하여도 부하의 평형에 큰 문제가 없다는 뜻이므로 중첩 영역의 최소화를 위해 제1 비할당 블록을 제1 서브 도메인에 할당할 수 있고, 비율이 소정의 임계치 이상이면 부하의 평형을 위해 제1 비할당 블록을 제2 액티브 리스트와 매핑된 그래픽 처리 장치가 담당하는 제2 서브 도메인에 할당할 수 있다(S623).

[0081] 이에 따른 실시예를 수도코드로 나타내면 아래 표 3과 같다.

Д 3

Algorithm 2: IMPROVED CLUSTERING ALGORITHM USING LOCAL LOAD BALANCING

```

1  /*  $N$ : total number of sub-domains */
2  Input: Active List  $L_1, \dots, L_N$ , Assigned Block List  $P_1, \dots, P_N$ 
3  Output: Unassigned block list  $Q_1, \dots, Q_N$ 
4  /*  $c_i$ : expected size of active list  $L_i$  after clustering */
5  for all the  $i = 1$  to  $N$  do
6     $c_i \leftarrow \alpha \times \text{size of } P_i$ 
7  for all the  $i = 1$  to  $N$  do
8    for all the  $v \in L_i$  do
9      if  $T(v) = \text{unassigned}$  then
10          $j \leftarrow$  index of sub-domain adjacent to  $v$  that appears most
11          $k \leftarrow$  index of sub-domain adjacent to  $v$  whose  $c$  is the smallest
12         if  $c_j/c_k < \varepsilon_2$  then
13             Insert  $v$  to  $Q_j$  // Default policy, increase clustering
14              $c_j = c_j + \beta$ 
15         else
16             Insert  $v$  to  $Q_k$  // Expected task loads are unbalanced
17              $c_k = c_k + \beta$ 

```

[0082]

[0083] 상술한 실시예에 따르면 복수의 그래픽 연산 장치를 사용하는 컴퓨터에서 FIM을 사용하는 경우 FIM의 데이터 연산에 맞추어 각 그래픽 연산 장치가 담당하는 서브 도메인을 동적으로 확장하여 각 그래픽 연산 장치간의 부하의 불균형을 줄일 수 있으며, 각 서브 도메인의 중첩 영역을 최소화하여 데이터 동기화의 비용을 줄일 수 있다.

[0084] 상술한 본 발명의 실시예들은 다양한 수단을 통해 구현될 수 있다. 예를 들어, 본 발명의 실시예들은 하드웨어, 펌웨어(firmware), 소프트웨어 또는 그것들의 결합 등에 의해 구현될 수 있다.

[0085] 하드웨어에 의한 구현의 경우, 본 발명의 실시예들에 따른 방법은 하나 또는 그 이상의 ASICs(Application Specific Integrated Circuits), DSPs(Digital Signal Processors), DSPDs(Digital Signal Processing Devices), PLDs(Programmable Logic Devices), FPGAs(Field Programmable Gate Arrays), 프로세서, 컨트롤러, 마이크로 컨트롤러, 마이크로 프로세서 등에 의해 구현될 수 있다.

[0086] 펌웨어나 소프트웨어에 의한 구현의 경우, 본 발명의 실시예들에 따른 방법은 이상에서 설명된 기능 또는 동작들을 수행하는 모듈, 절차 또는 함수 등의 형태로 구현될 수 있다. 소프트웨어 코드는 메모리 유닛에 저장되어 프로세서에 의해 구동될 수 있다. 상기 메모리 유닛은 상기 프로세서 내부 또는 외부에 위치하여, 이미 공지된 다양한 수단에 의해 상기 프로세서와 데이터를 주고 받을 수 있다.

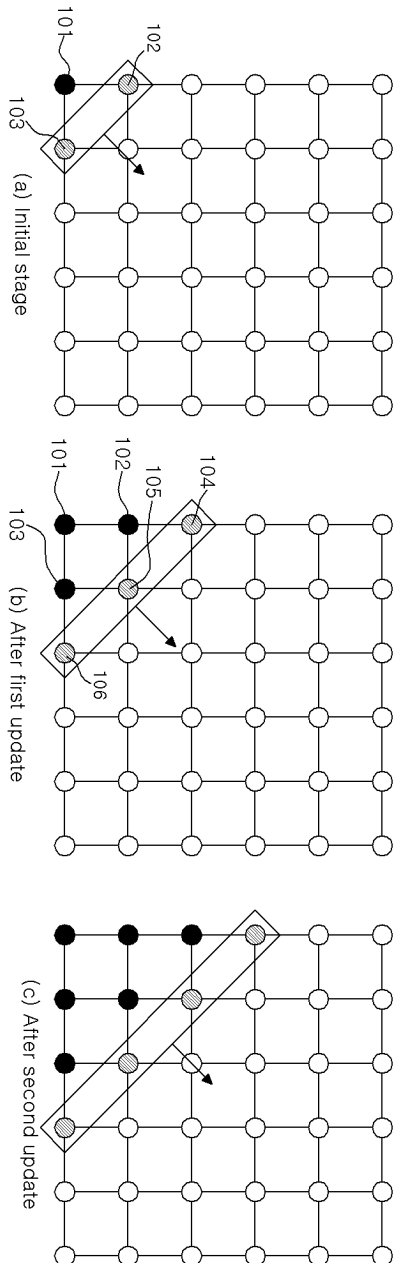
[0087] 이와 같이, 본 발명이 속하는 기술분야의 당업자는 본 발명이 그 기술적 사상이나 필수적 특징을 변경하지 않고서 다른 구체적인 형태로 실시될 수 있다는 것을 이해할 수 있을 것이다. 그러므로 이상에서 기술한 실시예들은 모든 면에서 예시적인 것이며 한정적인 것이 아닌 것으로서 이해해야만 한다. 본 발명의 범위는 상기 상세한 설명보다는 후술하는 특허청구범위에 의하여 나타내어지며, 특허청구범위의 의미 및 범위 그리고 그 등가개념으로부터 도출되는 모든 변경 또는 변형된 형태가 본 발명의 범위에 포함되는 것으로 해석되어야 한다.

부호의 설명

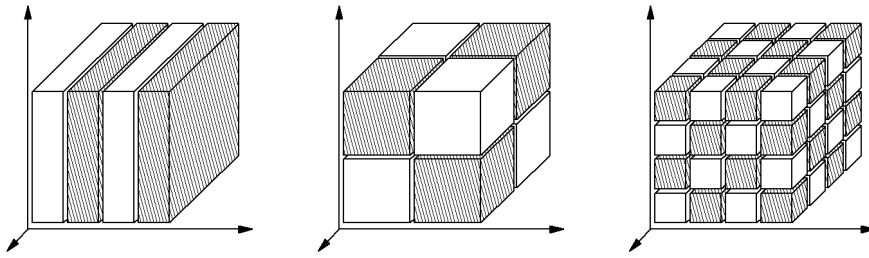
- [0089]
- 500: FIM의 동적 서브 도메인 확장 장치
- 510: 판별부
- 520: 할당부

도면

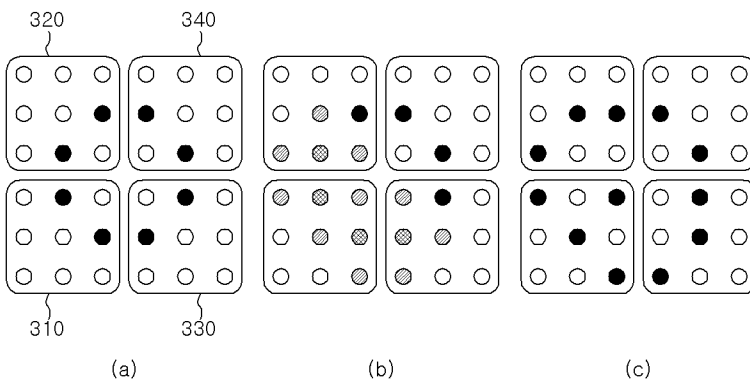
도면1



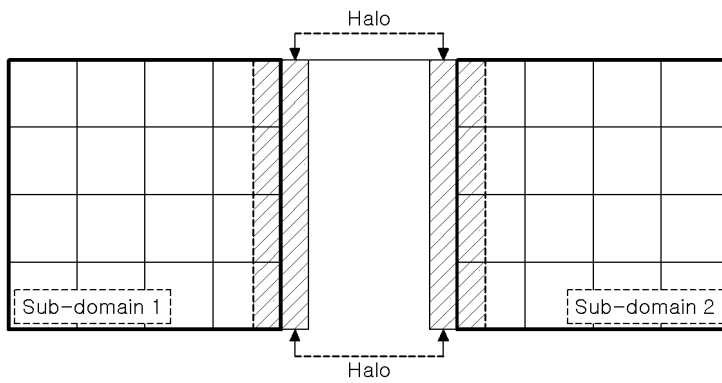
도면2



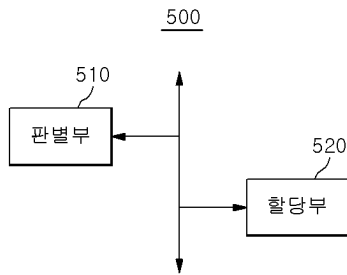
도면3



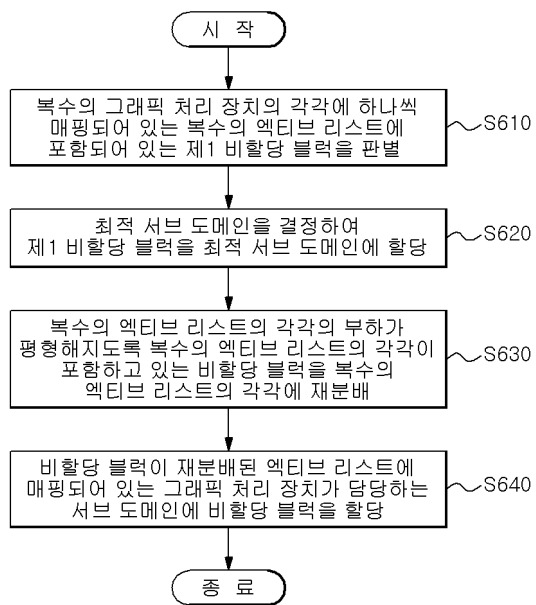
도면4



도면5



도면6



도면7

