(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2018-0070438
(43) 공개일자 2018년06월26일

(51) 국제특허분류(Int. Cl.)
H04N 19/119 (2014.01) *H04N 19/122* (2014.01)
H04N 19/176 (2014.01) *H04N 5/14* (2006.01)
(52) CPC특허분류
H04N 19/119 (2015.01)
H04N 19/122 (2015.01)
(21) 출원번호 10-2017-0084036
(22) 출원일자 2017년07월03일
심사청구일자 2017년07월03일
(30) 우선권주장
1020160171831 2016년12월15일 대한민국(KR)

(71) 출원인
경희대학교 산학협력단
경기도 용인시 기흥구 덕영대로 1732 (서천동, 경희대학교 국제캠퍼스내)
(72) 발명자
허의남
경기도 용인시 기흥구 이현로29번길 86-21 201동401호(보정동,e편한세상대림아파트)
이진섭
경기도 수원시 권선구 탑동로57번길 66-15, 한성빌라 A동 301호
(뒷면에 계속)
(74) 대리인
김등용, 김홍석

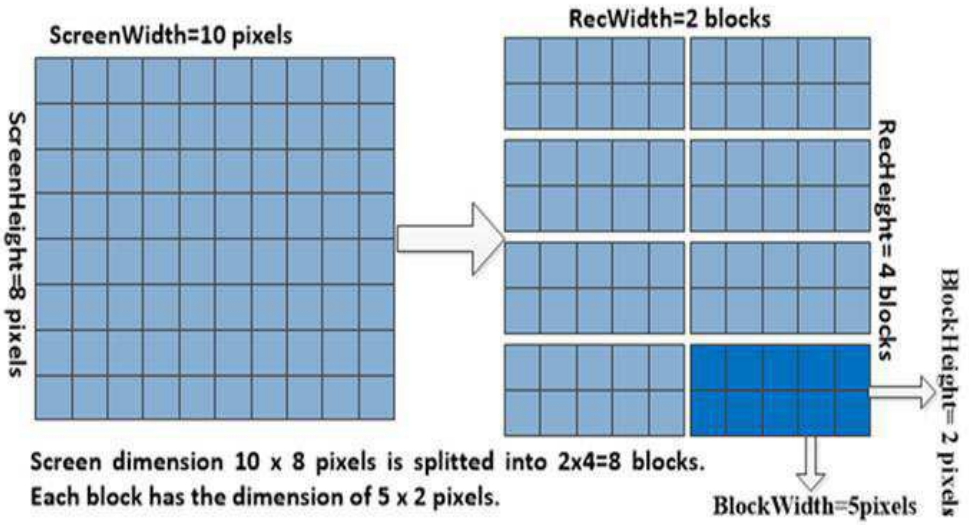
전체 청구항 수 : 총 20 항

(54) 발명의 명칭 **데스크톱 스크린을 비중첩 동일 크기 블록으로 분할하기 위한 동적 분할 알고리즘**

(57) 요 약

화면의 분할을 이미지나 비디오의 압축, 고효율 비디오 코팅(HEVC), 비디오나 이미지의 블록 Classification, 원격 데스크톱 전송 등 다양한 어플리케이션에서 활용된다. 여기에 제시된 방법은 이미지나 데스크톱 스크린을 중복되지 않는 동일한 크기의 블록으로 분할하는데 적용 가능하다.

대 표 도 - 도1



(52) CPC특허분류
H04N 19/176 (2015.01)
H04N 5/145 (2013.01)
(72) 발명자
박준영
경기도 용인시 기흥구 덕영대로 1732 국제캠퍼스 전자정보대학 331호 (서천동,경희대학교)

레익 아부
경기도 용인시 기흥구 덕영대로 1732 경희대학교 국제캠퍼스 전자정보대학 331호

이 발명을 지원한 국가연구개발사업
과제고유번호 1711035224
부처명 미래창조과학부
연구관리전문기관 정보통신기술진흥센터
연구사업명 대학ICT연구센터육성 지원사업
연구과제명 실시간 모바일 클라우드 서비스 플랫폼 연구개발
기 여 율 1/1
주관기관 경희대학교 산학협력단
연구기간 2016.01.01 ~ 2016.12.31

명 세 서

청구범위

청구항 1

데스크톱 스크린을 비중첩 동일 크기 블록으로 분할하기 위한 동적 분할 방법에서,

화면에 대한 화면 높이(SH: ScreenHeight)와 화면 너비(SW: ScreenWidth)에 관한 입력 정보를 수신하는 입력 정보 수신 과정;

상기 화면 높이(SH) 및 상기 화면 너비(SW)의 최대 공약수(GCD: Greatest Common Divisor)를 연산하는 최대 공약수 연산 과정; 및

상기 화면의 너비와 높이를 각각 상기 GCD보다 작은 값을 갖는 사이즈의 블록으로 분할하여, 상기 화면의 높이 방향과 너비 방향의 블록 개수(RecH: RecHeight, RecW: RecWidth)를 결정하는 블록 사이즈 결정 과정을 포함하는, 동적 분할 방법.

청구항 2

제1 항에 있어서,

상기 블록 개수에 따라 분할된 각 블록에 대한 픽셀 정보 매트릭스인 제1 픽셀 정보 매트릭스를 생성하는 제1 픽셀 정보 매트릭스 생성 과정을 더 포함하고 ,

상기 각 블록에 대한 픽셀 정보 매트릭스는,

$$RM = \begin{bmatrix} B_{11} & B_{12} & \cdots & \cdots & B_{1RecW} \\ B_{21} & B_{22} & \cdots & \cdots & B_{2RecW} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ B_{RecH1} & \cdots & \cdots & \cdots & P_{RecH\,RecW} \end{bmatrix}$$ 와 같이 결정되는, 동적 분할 방법.

청구항 3

제2 항에 있어서,

상기 각 블록 내부의 픽셀에 대한 픽셀 정보 매트릭스인 제2 픽셀 정보 매트릭스를 생성하는 제2 픽셀 정보 매트릭스 생성 과정을 더 포함하고 ,

n행의 k번째 블록은 B_{nk} 로 표현되고, 상기 B_{nk} 에 대한 제2 픽셀 정보 매트릭스 BM_{nk}는,

$$BM_{nk} = \begin{bmatrix} P_{[(n-1)*BH+1][(k-1)*BW+1]} & P_{[(n-1)*BH+1][(k-1)*BW+2]} & \cdots & \cdots & P_{[(n-1)*BH+1][(k-1)*BW+BW]} \\ P_{[(n-1)*BH+2][(k-1)*BW+1]} & P_{[(n-1)*BH+2][(k-1)*BW+2]} & \cdots & \cdots & P_{[(n-1)*BH+2][(k-1)*BW+BW]} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ P_{[(n-1)*BH+BH][(k-1)*BW+1]} & P_{[(n-1)*BH+BH][(k-1)*BW+2]} & \cdots & \cdots & P_{[(n-1)*BH+BH][(k-1)*BW+BW]} \end{bmatrix}$$ 와 같이 결정되는, 동적 분할 방법.

청구항 4

제3 항에 있어서,

상기 분할된 각 블록에 대하여, 상기 제1 픽셀 정보 매트릭스에 스케일링이나 프로세싱을 위한 서로 다른 상수 또는 서로 다른 매트릭스 연산을 적용하고, 상기 각 블록 내의 픽셀에 대하여, 상기 제2 픽셀 정보 매트릭스에

스케일링이나 프로세싱을 위한 동일한 상수 또는 동일한 매트릭스 연산을 적용하는 이미지 프로세싱 과정; 및

상기 동일하거나 서로 다른 상수 또는 매트릭스 연산이 적용된 제1 및 제2 픽셀 정보 매트릭스에 의해 스케일링 또는 프로세싱된 픽셀을 포함하는 이미지를 표시하는 이미시 표시 과정을 더 포함하는 것을 특징으로 하는, 동적 분할 방법.

청구항 5

데스크톱 스크린을 비중첩 동일 크기 블록으로 분할하기 위한 동적 분할 방법에서,

화면에 대한 화면 높이(SH: ScreenHeight)와 화면 너비(SW: ScreenWidth)에 관한 입력 정보를 수신하는 입력 정보 수신 과정;

상기 화면 높이(SH) 및 상기 화면 너비(SW)의 최대 공약수(GCD: Greatest Common Divisor)와 상기 GCD의 약수들(GCDdivs)을 연산하는 최대 공약수 및 약수 연산 과정; 및

상기 화면의 너비와 높이를 상기 약수들(GCDdivs) 중 하나의 사이즈의 블록으로 분할하여, 상기 화면의 높이 방향과 너비 방향의 블록 개수(RecH: RecHeight, RecW: RecWidth)를 결정하는 블록 사이즈 결정 과정을 포함하는, 동적 분할 방법.

청구항 6

제5 항에 있어서,

상기 입력 정보 수신 과정에서, 최대 블록의 수(MaxDB)에 대한 정보를 더 수신하고,

상기 블록 사이즈 결정 과정은,

상기 약수들(GCDdivs)의 개수만큼 인덱스(i)를 증가시키면서, 상기 블록 개수(RecH, RecW)를 결정하는 제1 결정 과정;

상기 블록 개수(RecH, RecW)의 곱에 해당하는 전체 블록 개수가 상기 최대 블록의 수(MaxDB) 이상인지를 판단하는 블록 개수 판단 과정; 및

상기 전체 블록 개수가 상기 최대 블록의 수(MaxDB) 미만이면, 상기 제1 결정 과정에서 결정된 상기 블록 개수(RecH, RecW)를 반환하도록 결정하는 제2 결정 과정을 포함하는, 동적 분할 방법.

청구항 7

제5 항에 있어서,

상기 입력 정보 수신 과정에서, 원하는 블록 사이즈 (BS: block size) 정보를 더 수신하고,

상기 블록 사이즈 결정 과정은,

상기 약수들(GCDdivs)의 개수만큼 인덱스(i)를 증가시키면서, 상기 블록 개수(RecH, RecW)를 결정하는 제1 결정 과정;

상기 블록 개수(RecH, RecW)의 곱에 해당하는 전체 블록 개수가 최대 블록의 수(MaxDB) 이상인지를 판단하는 블록 개수 판단 과정 – 상기 최대 블록의 수(MaxDB)는 ceil(SH/BS) * ceil (SW/BS)로 결정되고, ceil()는 소수점에 대한 올림 함수임 – ; 및

상기 전체 블록 개수가 상기 최대 블록의 수(MaxDB) 미만이면, 상기 제1 결정 과정에서 결정된 상기 블록 개수(RecH, RecW)를 반환하도록 결정하는 제2 결정 과정을 포함하는, 동적 분할 방법.

청구항 8

제5 항에 있어서,

상기 블록 개수(RecH, RecW)는 상기 화면의 종횡비(axial ratio)에 비례하여 결정되고, 상기 블록 각각은 정사각형 형태인 것을 특징으로 하는, 동적 분할 방법.

청구항 9

데스크톱 스크린을 비중첩 동일 크기 블록으로 분할하기 위한 동적 분할 방법에서,

화면에 대한 화면 높이(SH: ScreenHeight), 화면 너비(SW: ScreenWidth), 제1 블록 개수(DHeight, DWidth) 또는 제1 블록 사이즈(BH: BlockHeight, BW:BlockSize)에 대한 정보를 수신하는 입력 정보 수신 과정;

상기 화면 높이(SH)와 상기 화면 너비(SW)에 대한 약수들(WidthDivs, HeightDivs)을 연산하는 약수 연산 과정;

상기 연산된 약수들(HeightDivs, WidthDivs) 중에서 상기 제1 블록 개수(DHeight, DWidth)에 가장 가까운 약수(HeightDivs[I], WidthDivs[J])들을 탐색하는 약수 탐색 과정;

상기 화면의 너비와 높이를 각각 상기 탐색된 약수(HeightDivs[I], WidthDivs[J])에 해당하는 개수로 분할하여, 상기 화면의 높이 방향과 너비 방향의 제2 블록 개수(RecH: RecHeight, RecW: RecWidth)를 결정하는 블록 사이즈 결정 과정을 포함하는, 동적 분할 방법.

청구항 10

제9항에 있어서,

상기 약수 탐색 과정 이후에,

상기 제1 블록 개수(DHeight, DWidth)의 곱이 최대 블록의 수(MaxDB) 이상인지를 판단하는 블록 개수 판단 과정; 및

상기 제1 블록 개수(DHeight, DWidth)의 곱이 상기 최대 블록의 수(MaxDB) 이상이면, 상기 탐색된 약수(Heightdivs[I], Widthdivs[J])들의 인접한 약수들(HeightDivs[I±k], WidthDivs[J±k]) 중에서 하나를 선택하는 인접 약수 선택 과정을 더 포함하고,

상기 블록 사이즈 결정 과정은,

상기 화면의 너비와 높이를 각각 상기 선택된 약수에 해당하는 개수로 분할하여, 상기 화면의 높이 방향과 너비 방향의 제2 블록 개수(RecH: RecHeight, RecW: RecWidth)를 결정하는, 동적 분할 방법.

청구항 11

제10 항에 있어서,

상기 입력 정보 수신 과정에서, 상기 제1 블록 개수(DHeight, DWidth)에 대한 정보가 수신되고,

상기 최대 블록의 수(MaxDB)는 상기 제1 블록 개수(DHeight, DWidth)의 곱으로 결정되는, 동적 분할 방법.

청구항 12

제10 항에 있어서,

상기 입력 정보 수신 과정에서, 제1 블록 사이즈(BH: BlockHeight, BW:BlockSize)에 대한 정보가 수신되고,

상기 제1 블록 개수(DHeight, DWidth)는 각각 ceil(SH/BH) 및 ceil(SW/BH)로 결정되고, ceil()는 소수점에 대한 올림 함수이고,

상기 최대 블록의 수(MaxDB)는 상기 제1 블록 개수(DHeight, DWidth)의 곱으로 결정되는, 동적 분할 방법.

청구항 13

데스크톱 스크린을 비중첩 동일 크기 블록으로 분할하기 위한 동적 분할 장치에서,

화면에 대한 화면 높이(SH: ScreenHeight)와 화면 너비(SW: ScreenWidth)에 관한 입력 정보를 수신하는 인터페이스; 및

상기 인터페이스로부터 수신한 상기 화면 높이(SH) 및 상기 화면 너비(SW)의 최대 공약수(GCD: Greatest Common Divisor)와 상기 GCD의 약수들(GCDdivs)을 연산하고, 상기 화면의 너비와 높이를 상기 약수들(GCDdivs) 중 하나의 사이즈의 블록으로 분할하여, 상기 화면의 높이 방향과 너비 방향의 블록 개수(RecH: RecHeight, RecW: RecWidth)를 결정하는 제어부를 포함하는, 동적 분할 장치.

청구항 14

제13 항에 있어서,

상기 제어부는, 상기 분할된 각 블록에 대하여, 상기 제1 픽셀 정보 매트릭스에 스케일링이나 프로세싱을 위한 서로 다른 상수 또는 서로 다른 매트릭스 연산을 적용하고, 상기 각 블록 내의 픽셀에 대하여, 상기 제2 픽셀 정보 매트릭스에 스케일링이나 프로세싱을 위한 동일한 상수 또는 동일한 매트릭스 연산을 적용하고,

상기 동일하거나 서로 다른 상수 또는 매트릭스 연산이 적용된 제1 및 제2 픽셀 정보 매트릭스에 의해 스케일링 또는 프로세싱된 픽셀을 포함하는 이미지를 표시하는 디스플레이를 더 포함하는, 동적 분할 장치.

청구항 15

제13 항에 있어서,

상기 인터페이스는, 최대 블록의 수(MaxDB)에 대한 정보를 더 수신하고,

상기 제어부는, 상기 약수들(GCDdivs)의 개수만큼 인덱스(i)를 증가시키면서, 상기 블록 개수(RecH, RecW)를 결정하고, 상기 블록 개수(RecH, RecW)의 곱에 해당하는 전체 블록 개수가 상기 최대 블록의 수(MaxDB) 이상인지를 판단하고, 그리고 상기 전체 블록 개수가 상기 최대 블록의 수(MaxDB) 미만이면, 상기 결정된 블록 개수(RecH, RecW)를 반환하도록 결정하는, 동적 분할 장치.

청구항 16

제13 항에 있어서,

상기 인터페이스는, 원하는 블록 사이즈 (BS: block size) 정보를 더 수신하고,

상기 제어부는, 상기 약수들(GCDdivs)의 개수만큼 인덱스(i)를 증가시키면서, 상기 블록 개수(RecH, RecW)를 결정하고, 상기 블록 개수(RecH, RecW)의 곱에 해당하는 전체 블록 개수가 최대 블록의 수(MaxDB) 이상인지를 판단 – 상기 최대 블록의 수(MaxDB)는 ceil(SH/BS) * ceil (SW/BS)로 결정되고, ceil()는 소수점에 대한 올림 함수임 – 하고, 그리고 상기 전체 블록 개수가 상기 최대 블록의 수(MaxDB) 미만이면, 상기 결정된 블록 개수(RecH, RecW)를 반환하도록 결정하는, 동적 분할 장치.

청구항 17

데스크톱 스크린을 비중첩 동일 크기 블록으로 분할하기 위한 동적 분할 장치에서,

화면에 대한 화면 높이(SH: ScreenHeight), 화면 너비(SW: ScreenWidth), 제1 블록 개수(DHeight, DWidth) 또는 제1 블록 사이즈(BH: BlockHeight, BW:BlockSize)에 대한 정보를 수신하는 인터페이스; 및

상기 인터페이스로부터 수신한 상기 화면 높이(SH)와 상기 화면 너비(SW)에 대한 약수들(WidthDivs, HeightDivs)을 연산하고, 상기 연산된 약수들(HeightDivs, WidthDivs) 중에서 상기 제1 블록 개수(DHeight, DWidth)에 가장 가까운 약수(HeightDivs[I], WidthDivs[J])들을 탐색하고, 그리고 상기 화면의 너비와 높이를 각각 상기 탐색된 약수(HeightDivs[I], WidthDivs[J])에 해당하는 개수로 분할하여, 상기 화면의 높이 방향과 너비 방향의 제2 블록 개수(RecH: RecHeight, RecW: RecWidth)를 결정하는 제어부를 포함하는, 동적 분할 장치.

청구항 18

제17 항에 있어서,

상기 제어부는, 상기 제1 블록 개수(DHeight, DWidth)의 곱이 최대 블록의 수(MaxDB) 이상인지를 판단하고, 상기 제1 블록 개수(DHeight, DWidth)의 곱이 상기 최대 블록의 수(MaxDB) 이상이면, 상기 탐색된 약수(Heightdivs[I], Widthdivs[J])들의 인접한 약수들(HeightDivs[I±k], WidthDivs[J±k]) 중에서 하나를 선택하고, 그리고 상기 화면의 너비와 높이를 각각 상기 선택된 약수에 해당하는 개수로 분할하여, 상기 화면의 높이 방향과 너비 방향의 제2 블록 개수(RecH: RecHeight, RecW: RecWidth)를 결정하는, 동적 분할 장치.

청구항 19

제17 항에 있어서,

상기 인터페이스는, 상기 제1 블록 개수(DHeight, DWidth)에 대한 정보를 수신하고,

상기 제어부는, 상기 최대 블록의 수(MaxDB)를 상기 제1 블록 개수(DHeight, DWidth)의 곱으로 결정하는, 동적 분할 장치.

청구항 20

제17 항에 있어서,

상기 인터페이스는, 제1 블록 사이즈(BH: BlockHeight, BW:BlockSize)에 대한 정보를 수신하고,

상기 제어부는, 상기 제1 블록 개수(DHeight, DWidth)를 각각 ceil(SH/BH) 및 ceil(SW/BH)로 결정하고, 상기 최대 블록의 수(MaxDB)를 상기 제1 블록 개수(DHeight, DWidth)의 곱으로 결정하는, 동적 분할 장치.

발명의 설명

기 술 분 야

[0001] 본 발명은 데스크톱 스크린을 비중첩 동일 크기 블록으로 분할하기 위한 동적 분할 알고리즘에 관한 것이다.

배 경 기 술

[0002] 기존의 기술은 고정된 크기(16×16, 8×8등)의 작은 블록을 설정하고 그에 대한 분할 방법을 적용한다. 이미지 또는 동영상의 압축, 분류, 스크린 컨텐츠 코딩을 다루는 많은 연구에서 화면을 8×8, 16×16, 32×32 사이즈의 블록으로 나누는 방법이 사용된다. 동일하게, 가상 데스크톱 프로토콜이나 카메라 어플리케이션 역시 분할을 필요로 한다. 추가적으로, 몇몇 VNC-RFB 프로토콜을 이용하는 원격 데스크톱 Quality on Experience 프로비저닝 연구에서도 고정된 블록사이즈로 화면을 분할하는 것이 사용된다. 동일한 블록 크기의 분할이 갖는 이점은 스케일링이나 프로세싱에 동일한 상수와 수식을 모든 블록에 공통적으로 사용할 수 있다는 점이다.

[0003] 화면에 나타난 영상의 다양한 상황에 따라 동적으로 블록의 크기를 조절하는 것이 가능 하다면 이는 보다 효율적 영상 압축 및 처리의 방법이 될 수 있다.

선행기술문헌

특허문헌

[0004] (특허문헌 0001) 대한민국 공개특허 제10-2007-0007382호(2007.01.15)

발명의 내용

해결하려는 과제

[0005] 본 특허에서는 상기의 문제를 해결하기 위해 동적 화면 분할을 위한 알고리즘을 제안한다.

[0006] 첫 번째 알고리즘은 중첩되지 않는 가장 큰 정사각형의 블록으로 분할하기 위한 알고리즘이다.

[0007] 두 번째 알고리즘 역시 정사각형으로 블록을 분할하는 방법이다.

과제의 해결 수단

[0008] 화면의 분할을 이미지나 비디오의 압축, 고효율 비디오 코팅(HEVC), 비디오나 이미지의 블록 Classification, 원격 데스크톱 전송 등 다양한 어플리케이션에서 활용된다. 여기에 제시된 방법은 이미지나 데스크톱 스크린을 중복되지 않는 동일한 크기의 블록으로 분할하는데 적용 가능하다.

[0009] 본 발명의 일 양상에 따른 데스크톱 스크린을 비중첩 동일 크기 블록으로 분할하기 위한 동적 분할 방법은, 화면에 대한 화면 높이(SH: ScreenHeight)와 화면 너비(SW: ScreenWidth)에 관한 입력 정보를 수신하는 입력 정보 수신 과정; 상기 화면 높이(SH) 및 상기 화면 너비(SW)의 최대 공약수(GCD: Greatest Common Divisor)를 연

산하는 최대 공약수 연산 과정; 및 상기 화면의 너비와 높이를 각각 상기 GCD보다 작은 값을 갖는 사이즈의 블록으로 분할하여, 상기 화면의 높이 방향과 너비 방향의 블록 개수(RecH: RecHeight, RecW: RecWidth)를 결정하는 블록 사이즈 결정 과정을 포함하고, 특정 어플리케이션에 대한 최적의 분할을 찾아 영상처리의 효율을 높이며, 상황에 맞게 자동적으로 적용할 수 있다.

[0010] 일 실시 예에서, 상기 블록 개수에 따라 분할된 각 블록에 대한 픽셀 정보 매트릭스인 제1 픽셀 정보 매트릭스를 생성하는 제1 픽셀 정보 매트릭스 생성 과정을 더 포함할 수 있다. 이때, 상기 각 블록에 대한 픽셀 정보 매트릭스는,

$$RM = \begin{bmatrix} B_{11} & B_{12} & \cdots & \cdots & B_{1RecW} \\ B_{21} & B_{22} & \cdots & \cdots & B_{2RecW} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ B_{RecH1} & \cdots & \cdots & \cdots & P_{RecH\,RecW} \end{bmatrix}$$

와 같이 결정될 수 있다.

[0011] 일 실시 예에서, 상기 각 블록 내부의 픽셀에 대한 픽셀 정보 매트릭스인 제2 픽셀 정보 매트릭스를 생성하는 제2 픽셀 정보 매트릭스 생성 과정을 더 포함할 수 있다. 이때, n행의 k번째 블록은 B_{nk} 로 표현되고, 상기 B_{nk} 에 대한 제2 픽셀 정보 매트릭스 BM_{nk}는,

[0012]
$$BM_{nk} = \begin{bmatrix} P_{[(n-1)*BH+1][(k-1)*BW+1]} & P_{[(n-1)*BH+1][(k-1)*BW+2]} & \cdots & \cdots & P_{[(n-1)*BH+1][(k-1)*BW+BW]} \\ P_{[(n-1)*BH+2][(k-1)*BW+1]} & P_{[(n-1)*BH+2][(k-1)*BW+2]} & \cdots & \cdots & P_{[(n-1)*BH+2][(k-1)*BW+BW]} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ P_{[(n-1)*BH+BH][(k-1)*BW+1]} & P_{[(n-1)*BH+BH][(k-1)*BW+2]} & \cdots & \cdots & P_{[(n-1)*BH+BH][(k-1)*BW+BW]} \end{bmatrix}$$

와 같이 결정될 수 있다.

[0013] 일 실시 예에서, 상기 분할된 각 블록에 대하여, 상기 제1 픽셀 정보 매트릭스에 스케일링이나 프로세싱을 위한 서로 다른 상수 또는 서로 다른 매트릭스 연산을 적용하고, 상기 각 블록 내의 픽셀에 대하여, 상기 제2 픽셀 정보 매트릭스에 스케일링이나 프로세싱을 위한 동일한 상수 또는 동일한 매트릭스 연산을 적용하는 이미지 프로세싱 과정을 더 포함할 수 있다. 또한, 상기 동일하거나 서로 다른 상수 또는 매트릭스 연산이 적용된 제1 및 제2 픽셀 정보 매트릭스에 의해 스케일링 또는 프로세싱된 픽셀을 포함하는 이미지를 표시하는 이미지 표시 과정을 더 포함할 수 있다.

[0014] 본 발명의 다른 양상에 따른, 데스크톱 스크린을 비중첩 동일 크기 블록으로 분할하기 위한 동적 분할 방법은, 화면에 대한 화면 높이(SH: ScreenHeight)와 화면 너비(SW: ScreenWidth)에 관한 입력 정보를 수신하는 입력 정보 수신 과정; 상기 화면 높이(SH) 및 상기 화면 너비(SW)의 최대 공약수(GCD: Greatest Common Divisor)와 상기 GCD의 약수들(GCDdivs)을 연산하는 최대 공약수 및 약수 연산 과정; 및 상기 화면의 너비와 높이를 상기 약수들(GCDdivs) 중 하나의 사이즈의 블록으로 분할하여, 상기 화면의 높이 방향과 너비 방향의 블록 개수(RecH: RecHeight, RecW: RecWidth)를 결정하는 블록 사이즈 결정 과정을 포함한다.

[0015] 일 실시 예에서, 상기 입력 정보 수신 과정에서, 최대 블록의 수(MaxDB)에 대한 정보를 더 수신할 수 있다. 한편, 상기 블록 사이즈 결정 과정은, 상기 약수들(GCDdivs)의 개수만큼 인덱스(i)를 증가시키면서, 상기 블록 개수(RecH, RecW)를 결정하는 제1 결정 과정; 상기 블록 개수(RecH, RecW)의 곱에 해당하는 전체 블록 개수가 상기 최대 블록의 수(MaxDB) 이상인지를 판단하는 블록 개수 판단 과정; 및 상기 전체 블록 개수가 상기 최대 블록의 수(MaxDB) 미만이면, 상기 제1 결정 과정에서 결정된 상기 블록 개수(RecH, RecW)를 반환하도록 결정하는 제2 결정 과정을 포함할 수 있다.

[0016] 일 실시 예에서, 상기 입력 정보 수신 과정에서, 원하는 블록 사이즈 (BS: block size) 정보를 더 수신할 수 있다. 한편, 상기 블록 사이즈 결정 과정은, 상기 약수들(GCDdivs)의 개수만큼 인덱스(i)를 증가시키면서, 상기 블록 개수(RecH, RecW)를 결정하는 제1 결정 과정; 상기 블록 개수(RecH, RecW)의 곱에 해당하는 전체 블록 개수가 최대 블록의 수(MaxDB) 이상인지를 판단하는 블록 개수 판단 과정 – 상기 최대 블록의 수(MaxDB)는 ceil(SH/BS) * ceil (SW/BS)로 결정되고, ceil()는 소수점에 대한 올림 함수임 – ; 및 상기 전체 블록 개수가 상기 최대 블록의 수(MaxDB) 미만이면, 상기 제1 결정 과정에서 결정된 상기 블록 개수(RecH, RecW)를 반환하도

록 결정하는 제2 결정 과정을 포함할 수 있다.

[0017] 일 실시 예에서, 상기 블록 개수(RecH, RecW)는 상기 화면의 종횡비(axial ratio)에 비례하여 결정되고, 상기 블록 각각은 정사각형 형태인 것을 특징으로 할 수 있다.

[0018] 본 발명의 또 다른 양상에 따른, 데스크톱 스크린을 비중첩 동일 크기 블록으로 분할하기 위한 동적 분할 방법은, 화면에 대한 화면 높이(SH: ScreenHeight), 화면 너비(SW: ScreenWidth), 제1 블록 개수(DHeight, DWidth) 또는 제1 블록 사이즈(BH: BlockHeight, BW:BlockSize)에 대한 정보를 수신하는 입력 정보 수신 과정; 상기 화면 높이(SH)와 상기 화면 너비(SW)에 대한 약수들(WidthDivs, HeightDivs)을 연산하는 약수 연산 과정; 상기 연산된 약수들(HeightDivs, WidthDivs) 중에서 상기 제1 블록 개수(DHeight, DWidth)에 가장 가까운 약수(HeightDivs[I], WidthDivs[J])들을 탐색하는 약수 탐색 과정; 및 상기 화면의 너비와 높이를 각각 상기 탐색된 약수(HeightDivs[I], WidthDivs[J])에 해당하는 개수로 분할하여, 상기 화면의 높이 방향과 너비 방향의 제2 블록 개수(RecH: RecHeight, RecW: RecWidth)를 결정하는 블록 사이즈 결정 과정을 포함한다.

[0019] 일 실시 예에서, 상기 약수 탐색 과정 이후에, 상기 제1 블록 개수(DHeight, DWidth)의 곱이 최대 블록의 수(MaxDB) 이상인지를 판단하는 블록 개수 판단 과정; 및 상기 제1 블록 개수(DHeight, DWidth)의 곱이 상기 최대 블록의 수(MaxDB) 이상이면, 상기 탐색된 약수(Heightdivs[I], Widthdivs[J])들의 인접한 약수들(HeightDivs[I±k], WidthDivs[J±k]) 중에서 하나를 선택하는 인접 약수 선택 과정을 더 포함할 수 있다. 한편, 상기 블록 사이즈 결정 과정은, 상기 화면의 너비와 높이를 각각 상기 선택된 약수에 해당하는 개수로 분할하여, 상기 화면의 높이 방향과 너비 방향의 제2 블록 개수(RecH: RecHeight, RecW: RecWidth)를 결정할 수 있다.

[0020] 일 실시 예에서, 상기 입력 정보 수신 과정에서, 상기 제1 블록 개수(DHeight, DWidth)에 대한 정보가 수신될 수 있다. 한편, 상기 최대 블록의 수(MaxDB)는 상기 제1 블록 개수(DHeight, DWidth)의 곱으로 결정될 수 있다.

[0021] 일 실시 예에서, 상기 입력 정보 수신 과정에서, 제1 블록 사이즈(BH: BlockHeight, BW:BlockSize)에 대한 정보가 수신될 수 있다. 한편, 상기 제1 블록 개수(DHeight, DWidth)는 각각 ceil(SH/BH) 및 ceil(SW/BH)로 결정되고, ceil()는 소수점에 대한 올림 함수이고, 상기 최대 블록의 수(MaxDB)는 상기 제1 블록 개수(DHeight, DWidth)의 곱으로 결정될 수 있다.

[0022] 본 발명의 일 양상에 따른, 데스크톱 스크린을 비중첩 동일 크기 블록으로 분할하기 위한 동적 분할 장치는, 화면에 대한 화면 높이(SH: ScreenHeight)와 화면 너비(SW: ScreenWidth)에 관한 입력 정보를 수신하는 인터페이스; 및 상기 인터페이스로부터 수신한 상기 화면 높이(SH) 및 상기 화면 너비(SW)의 최대 공약수(GCD: Greatest Common Divisor)와 상기 GCD의 약수들(GCDdivs)을 연산하고, 상기 화면의 너비와 높이를 상기 약수들(GCDdivs) 중 하나의 사이즈의 블록으로 분할하여, 상기 화면의 높이 방향과 너비 방향의 블록 개수(RecH: RecHeight, RecW: RecWidth)를 결정하는 제어부를 포함한다.

[0023] 일 실시 예에서, 상기 제어부는, 상기 분할된 각 블록에 대하여, 상기 제1 픽셀 정보 매트릭스에 스케일링이나 프로세싱을 위한 서로 다른 상수 또는 서로 다른 매트릭스 연산을 적용하고, 상기 각 블록 내의 픽셀에 대하여, 상기 제2 픽셀 정보 매트릭스에 스케일링이나 프로세싱을 위한 동일한 상수 또는 동일한 매트릭스 연산을 적용할 수 있다. 한편, 상기 동일하거나 서로 다른 상수 또는 매트릭스 연산이 적용된 제1 및 제2 픽셀 정보 매트릭스에 의해 스케일링 또는 프로세싱된 픽셀을 포함하는 이미지를 표시하는 디스플레이를 더 포함할 수 있다.

[0024] 일 실시 예에서, 상기 인터페이스는, 최대 블록의 수(MaxDB)에 대한 정보를 더 수신할 수 있다. 한편, 상기 제어부는, 상기 약수들(GCDdivs)의 개수만큼 인덱스(i)를 증가시키면서, 상기 블록 개수(RecH, RecW)를 결정하고, 상기 블록 개수(RecH, RecW)의 곱에 해당하는 전체 블록 개수가 상기 최대 블록의 수(MaxDB) 이상인지를 판단하고, 그리고 상기 전체 블록 개수가 상기 최대 블록의 수(MaxDB) 미만이면, 상기 결정된 블록 개수(RecH, RecW)를 반환하도록 결정할 수 있다.

[0025] 일 실시 예에서, 상기 인터페이스는, 원하는 블록 사이즈 (BS: block size) 정보를 더 수신할 수 있다. 한편, 상기 제어부는, 상기 약수들(GCDdivs)의 개수만큼 인덱스(i)를 증가시키면서, 상기 블록 개수(RecH, RecW)를 결정하고, 상기 블록 개수(RecH, RecW)의 곱에 해당하는 전체 블록 개수가 최대 블록의 수(MaxDB) 이상인지를 판단 – 상기 최대 블록의 수(MaxDB)는 ceil(SH/BS) * ceil (SW/BS)로 결정되고, ceil()는 소수점에 대한 올림 함수임 – 하고, 그리고 상기 전체 블록 개수가 상기 최대 블록의 수(MaxDB) 미만이면, 상기 결정된 블록 개수(RecH, RecW)를 반환하도록 결정할 수 있다.

[0026] 본 발명의 다른 양상에 따른, 데스크톱 스크린을 비중첩 동일 크기 블록으로 분할하기 위한 동적 분할 장치는, 화면에 대한 화면 높이(SH: ScreenHeight), 화면 너비(SW: ScreenWidth), 제1 블록 개수(DHeight, DWidth) 또

는 제1 블록 사이즈(BH: BlockHeight, BW:BlockSize)에 대한 정보를 수신하는 인터페이스; 및 상기 인터페이스로부터 수신한 상기 화면 높이(SH)와 상기 화면 너비(SW)에 대한 약수들(WidthDivs, HeightDivs)을 연산하고, 상기 연산된 약수들(HeightDivs, WidthDivs) 중에서 상기 제1 블록 개수(DHeight, DWidth)에 가장 가까운 약수(HeightDivs[I], WidthDivs[J])들을 탐색하고, 그리고 상기 화면의 너비와 높이를 각각 상기 탐색된 약수(HeightDivs[I], WidthDivs[J])에 해당하는 개수로 분할하여, 상기 화면의 높이 방향과 너비 방향의 제2 블록 개수(RecH: RecHeight, RecW: RecWidth)를 결정하는 제어부를 포함한다.

[0027] 일 실시 예에서, 상기 제어부는, 상기 제1 블록 개수(DHeight, DWidth)의 곱이 최대 블록의 수(MaxDB) 이상인지를 판단하고, 상기 제1 블록 개수(DHeight, DWidth)의 곱이 상기 최대 블록의 수(MaxDB) 이상이면, 상기 탐색된 약수(Heightdivs[I], Widthdivs[J])들의 인접한 약수들(HeightDivs[I±k], WidthDivs[J±k]) 중에서 하나를 선택하고, 그리고 상기 화면의 너비와 높이를 각각 상기 선택된 약수에 해당하는 개수로 분할하여, 상기 화면의 높이 방향과 너비 방향의 제2 블록 개수(RecH: RecHeight, RecW: RecWidth)를 결정할 수 있다.

[0028] 일 실시 예에서, 상기 인터페이스는, 상기 제1 블록 개수(DHeight, DWidth)에 대한 정보를 수신할 수 있다. 한편, 상기 제어부는, 상기 최대 블록의 수(MaxDB)를 상기 제1 블록 개수(DHeight, DWidth)의 곱으로 결정할 수 있다.

[0029] 일 실시 예에서, 상기 인터페이스는, 제1 블록 사이즈(BH: BlockHeight, BW:BlockSize)에 대한 정보를 수신할 수 있다. 한편, 상기 제어부는, 상기 제1 블록 개수(DHeight, DWidth)를 각각 ceil(SH/BH) 및 ceil(SW/BH)로 결정하고, 상기 최대 블록의 수(MaxDB)를 상기 제1 블록 개수(DHeight, DWidth)의 곱으로 결정할 수 있다.

발명의 효과

[0030] 본 발명에 따른 데스크톱 스크린을 비중첩 동일 크기 블록으로 분할하기 위한 동적 분할 알고리즘에 의하면 특정 어플리케이션에 대한 최적의 분할을 찾아 영상처리의 효율을 높이며, 상황에 맞게 자동적으로 적용할 수 있는 방법이다.

도면의 간단한 설명

[0031] 도 1 은 스크린 분할의 예시이다.

도 2는 RM 매트릭스도 같은 종횡비를 가지는 생성된 블록이다.

도 3은 예제 이미지를 두 가지의 다른 수의 블록 개수로 분할한 것을 나타낸다.

도 4는 본 발명의 실시예에 따른 상세한 흐름도이다.

도 5는 본 발명의 실시예에 따른 예제 이미지이다.

도 6a는 알고리즘 1에 따른 동적 분할 방법의 흐름도를 도시한다.

도 6b는 알고리즘 2 또는 알고리즘 4에 따른 동적 분할 방법의 흐름도를 도시한다.

도 6c는 알고리즘 3 또는 알고리즘 5에 따른 동적 분할 방법의 흐름도를 도시한다.

도 7은 본 발명에 따른 동적 분할 장치 (또는 이미지 표시 장치)의 세부 구성을 도시한다.

발명을 실시하기 위한 구체적인 내용

[0032] 상술한 본 발명의 특징 및 효과는 첨부된 도면과 관련한 다음의 상세한 설명을 통하여 보다 분명해 질 것이며, 그에 따라 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자가 본 발명의 기술적 사상을 용이하게 실시할 수 있을 것이다. 본 발명은 다양한 변경을 가할 수 있고 여러 가지 형태를 가질 수 있는바, 특정 실시 예들을 도면에 예시하고 본문에 상세하게 설명하고자 한다. 그러나 이는 본 발명을 특정한 개시형태에 대해 한정하려는 것이 아니며, 본 발명의 사상 및 기술범위에 포함되는 모든 변경, 균등물 내지 대체물을 포함하는 것으로 이해되어야 한다. 본 명세서에서 사용한 용어는 단지 특정한 실시 예들을 설명하기 위해 사용된 것으로, 본 발명을 한정하려는 의도가 아니다.

[0033] 각 도면을 설명하면서 유사한 참조부호를 유사한 구성요소에 대해 사용한다.

[0034] 제1, 제2등의 용어는 다양한 구성요소들을 설명하는데 사용될 수 있지만, 상기 구성요소들은 상기 용어들에 의해 한정되어서는 안 된다. 상기 용어들은 하나의 구성요소를 다른 구성요소로부터 구별하는 목적으로만 사용된

다.

[0035] 예를 들어, 본 발명의 권리 범위를 벗어나지 않으면서 제1 구성요소는 제2 구성요소로 명명될 수 있고, 유사하게 제2 구성요소도 제1 구성요소로 명명될 수 있다. "및/또는" 이라는 용어는 복수의 관련된 기재된 항목들의 조합 또는 복수의 관련된 기재된 항목들 중의 어느 항목을 포함한다.

[0036] 다르게 정의되지 않는 한, 기술적이거나 과학적인 용어를 포함해서 여기서 사용되는 모든 용어들은 본 발명이 속하는 기술분야에서 통상의 지식을 가진 자에 의해 일반적으로 이해되는 것과 동일한 의미가 있다.

[0037] 일반적으로 사용되는 사전에 정의되어 있는 것과 같은 용어들은 관련 기술의 문맥상 가지는 의미와 일치하는 의미를 가지는 것으로 해석되어야 하며, 본 출원에서 명백하게 정의하지 않는 한, 이상적이거나 과도하게 형식적인 의미로 해석되지 않아야 한다.

[0038] 이하의 설명에서 사용되는 구성요소에 대한 접미사 "모듈", "블록" 및 "부"는 명세서 작성의 용이함만이 고려되어 부여되거나 혼용되는 것으로서, 그 자체로 서로 구별되는 의미 또는 역할을 갖는 것은 아니다.

[0039] 이하, 본 발명의 바람직한 실시 예를 첨부한 도면을 참조하여 당해 분야에 통상의 지식을 가진 자가 용이하게 실시할 수 있도록 설명한다. 하기에서 본 발명의 실시 예를 설명함에 있어, 관련된 공지의 기능 또는 공지의 구성에 대한 구체적인 설명이 본 발명의 요지를 불필요하게 흐릴 수 있다고 판단되는 경우에는 그 상세한 설명을 생략한다.

[0040] 이하, 본 발명에 따른 데스크톱 스크린을 비중첩 동일 크기 블록으로 분할하기 위한 동적 분할 방법 및 장치에 대해 살펴보기로 하자. 이와 관련하여, 본 발명에 따른 동적 분할 방법 및 장치에서의 화면 분할에 대해 살펴보면 다음과 같다.

[0041] 화면 분할 (Screen Splitting)

[0042] 말한바와 같이 화면 이미지, 동영상 또는 데스크톱의 스크린을 동일한 사이즈의 중복되지 않는 블록으로 분할할 필요가 있다. 만약, 화면의 넓이가 ScreenWidth(SW) x ScreenHeight(SH) 면, 분할 프로세싱은 그것을 Outer Rectangle이라는 이름의 RecWidth x RecHeight로 분할된다.

[0043] Rectangle과 블록간의 관계에서 ScreenWidth = BlockWidth*RecWidth, ScreenHeight = BlockHeight*RecHeidht 라는 것을 이용해 원본 스크린의 넓이를 얻는다.

[0044] 도 1 은 본 발명에 따른 스크린 분할의 예시이다. 도 1에 도시된 바와 같은 스크린 분할과 관련하여 수학적 분석은 다음과 같다.

[0045] 수학적 분석(Mathematical Analysis)

[0046] 스크린의 넓이를 SW×SH, 스크린의 너비를 SW, 높이를 SH라할 때, 화면의 픽셀 정보 매트릭스 FM은 수학식 1과 같다.

[0047] [수학식 1]

$$FM = \begin{bmatrix} P_{11} & P_{12} & \cdots & \cdots & P_{1SW} \\ P_{21} & P_{22} & \cdots & \cdots & P_{2SW} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ P_{SH1} & \cdots & \cdots & \cdots & P_{SHSW} \end{bmatrix}$$

[0048]

[0049] 스크린을 중첩되지 않는 블록 (크기는 BW×BH로 표시)로 분할하려고 한다면 블록의 너비와 높이는 스크린의 너비와 높이를 나머지 없이 나눠 떨어져야한다. 이를 수식으로 표현하면 수학식 2와 같다.

[0050] [수학식 2]

[0051] $$BH \mid SH \quad \text{and} \quad BW \mid SW$$

[0052] 또한, 분할 후의 높이방향과 너비방향의 블록 개수는 수학식 3, 수학식 4와 같이 표현할 수 있다.

[0053] [수학식 3]

[0054] $$\mathrm{Rec}H = \frac{SH}{BH}$$

[0055] [수학식 4]

[0056] $$\mathrm{Rec}W = \frac{SW}{BW}$$

[0057] 따라서 생성된 블록의 총 개수는 수학식 5와 같다.

[0058] [수학식 5]

[0059] $$TB = \mathrm{Rec}H * \mathrm{Rec}W$$

[0060] 이때 생성된 각 블록은 수학식 6과 같이 매트릭스 RM으로 표현할 수 있다.

[0061] [수학식 6]

[0062] $$RM = \begin{bmatrix} B_{11} & B_{12} & \cdots & \cdots & B_{1\mathrm{Rec}W} \\ B_{21} & B_{22} & \cdots & \cdots & B_{2\mathrm{Rec}W} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ B_{\mathrm{Rec}H1} & \cdots & \cdots & \cdots & P_{\mathrm{Rec}H\,\mathrm{Rec}W} \end{bmatrix}$$

[0063] 사각 매트릭스의 각 요소 B_{xx} 는 분할 결과 생성된 블록들을 의미한다. N행의 K번째 블록은 B_{nk} 로 표현되며, 이 블록안에 속한 화면의 픽셀 데이터는 수학식 7과 같은 행렬식으로 구할 수 있다.

[0064] [수학식 7]

[0065] $$BM_{nk} = \begin{bmatrix} P_{[(n-1)*BH+1][(k-1)*BW+1]} & P_{[(n-1)*BH+1][(k-1)*BW+2]} & \cdots & \cdots & P_{[(n-1)*BH+1][(k-1)*BW+BW]} \\ P_{[(n-1)*BH+2][(k-1)*BW+1]} & P_{[(n-1)*BH+2][(k-1)*BW+2]} & \cdots & \cdots & P_{[(n-1)*BH+2][(k-1)*BW+BW]} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ P_{[(n-1)*BH+BH][(k-1)*BW+1]} & P_{[(n-1)*BH+BH][(k-1)*BW+2]} & \cdots & \cdots & P_{[(n-1)*BH+BH][(k-1)*BW+BW]} \end{bmatrix}$$

[0066] B_{11}에 대한 예시는 수학식 8과 같다.

[0067] [수학식 8]

[0068] $$BM_{11} = \begin{bmatrix} P_{11} & P_{12} & \cdots & \cdots & P_{1BW} \\ P_{21} & P_{22} & \cdots & \cdots & P_{2BW} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ P_{BH1} & \cdots & \cdots & \cdots & P_{BHBW} \end{bmatrix}$$

[0069] 수학에서 사용하는 두 개 이상의 정수의 최대 공약수(Greatest Common Divisor)는 나머지 없이 두 수를 나누는 가장 큰 양의 정수이다. 최대공약수(GCD)를 스크린의 너비와 높이인 SW와 SH의 공약수 집합이라 할 때, 생성 가능한 가장 큰 블록은 GCD × GCD의 크기를 갖는다.

[0070] 이를 수학식 2와 결합하면 수학식 9와 같이 표현될 수 있다.

[0071] [수학식 9]

[0072] $$GCD \mid SH \text{ and } GCD \mid SW$$

[0073] 또한, BW=BH=GCD를 통해 수학식 3과 4를 아래 수학식 10, 11로 표현할 수 있다.

[0074] [수학식 10]

[0075] $$RecH = \frac{SH}{BH} = \frac{SH}{GCD}$$

[0076] [수학식 11]

[0077] $$RecW = \frac{SW}{BW} = \frac{SW}{GCD}$$

[0078] 수학식 10, 11을 이용하면 블록의 가로 개수와 세로 개수의 비율을 수학식 12와 같이 표현할 수 있다.

[0079] [수학식 12]

[0080] $$\frac{RecW}{RecH} = \frac{SH / GCD}{SW / GCD} = \frac{SH}{SW}$$

[0081] 따라서, 수학식 12와 같이 블록 개수의 종횡비는 스크린 크기의 종횡비와 동일하다.

[0082] 스크린을 동일한 크기의 블록으로 중첩없이 분할하기 위해 스크린의 너비와 높이의 최대공약수를 찾아야 한다. 수학식 13과 같이 벡터 GCDdivs는 1에서 GCD까지의 모든 약수를 포함한다.

[0083] [수학식 13]

[0084] $$GCDdivs = divisors(GCD) = [d_1, d_2, \ldots\ldots\ldots\ldots d_n]$$

[0085] 이때, d1 = 1이고 dn=GCD이다.

[0086] GCD의 정의에 따라, GCDdivs에 속한 모든 약수 d_i는 스크린의 너비와 높이를 각각 나머지 없이 나눌 수 있다. 이는 수학식 14와 같이 표현할 수 있다.

[0087] [수학식 14]

[0088] $$d_i \mid SH \text{ and } d_i \mid SW$$

[0089] 수학식 14에 따라 다시, 너비와 높이 방향의 블록의 개수의 비율은 수학식 15와 같이 스크린 원본의 종횡비를 보존한다.

[0090] [수학식 15]

[0091] $$\frac{RecH}{RecW} = \frac{SH / d_i}{SW / d_i} = \frac{SH}{SW}$$

[0092] MaxDB는 사용자가 원하는 블록 개수의 최대 범위이며 분할 시 수학식 16을 만족해야 한다.

[0093] [수학식 16]

[0094] $$TB <= MaxDB$$

[0095] TB는 분할 결과 생성된 블록의 총 개수이며, MaxDB와 가장 가까운 값을 가져야 한다.

[0096] 또 다른 분할 방법으로, 너비 방향과 높이 방향에 원하는 블록의 개수(DWidth와 DHeight)를 결정한다. 이때는 수학식 17과 같이 정사각형이 아닌 형태의 블록을 생성될 수 있다.

[0097] [수학식 17]

[0098] $$MaxDB = DHeight * DWidth$$

[0099] SH 및 SW에 대한 약수 벡터는 수학식 18과 같이 HeightDivs 및 수학식 19와 같이 WidthDivs라고 하며 아래 수학식 18, 19와 같이 정의된다.

[0100] [수학식 18]

[0101] $$HeightDivs = [Hd_1, Hd_2, ..., Hd_n]$$

[0102] [수학식 19]

[0103] $$WidthDivs = [Wd_1, Wd_2, ..., Wd_n]$$

[0104] Where, $Hd_1=Wd_1=1$, $Hd_n=SH$, $Wd_n=SW$

[0105] 이때, 분할 방법의 총 개수는 Length(WidthDivs)*Length(HeightDivs)이다.

[0106] 너비, 높이 방향의 블록 개수의 모든 쌍(DHeight, DWidth)에 대해서, 수학식 17을 만족하는 쌍을 아래 수학식 20 내지 24에 정의된 대로 찾을 필요가 있다.

[0107] [수학식 20]

[0108] $$I = Loc[\min\{abs(Hd_i - DHeight)\}]$$

[0109] [수학식 21]

[0110] $$J = Loc[\min\{abs(Wd_j - DWidth)\}]$$

[0111] [수학식 22]

[0112] $$RecH_i = HeightDivs(I)$$

[0113] [수학식 23]

[0114] $$RecW_j = WidthDivs(J)$$

[0115] [수학식 24]

[0116] $$TB_1 = RecH_i * RecW_j$$

[0117] 하지만, 찾은 쌍이 수학식을 만족하는 최적의 쌍인지 알 수 없다. 따라서 찾은 쌍의 주변 값을 이용한 블록 개수를 계산한다.

[0118] [수학식 25]

[0119] $$TB_2 = RecH_{i-1} * RecW_j$$

[0120] [수학식 26]

[0121] $$TB_3 = RecH_i * RecW_{j-1}$$

[0122] [수학식 27]

[0123] $$TB_4 = RecH_{i+1} * RecW_j$$

[0124] [수학식 28]

[0125] $$TB_5 = RecH_i * RecW_{j+1}$$

[0126] [수학식 29]

[0127] $$TB_6 = RecH_{i-1} * RecW_{j-1}$$

[0128] 그런 다음 MaxDB를 초과하지 않는 가장 가까운 TBi를 발견하면, 해당 사각형 넓이가 결정된다.

[0129] 분할 알고리즘(Splitting Algorithms)

[0130] 이 장에서는 스크린 분할을 형식 알고리즘으로 구성한다. 예시로 사용하는 'flower.jpg' 이미지 파일은 240 X 160이다.

[0131] (http://layek.itrrc.com/files/flower.jpg)

[0132] 분할은 원하는 블록 수, 블록 크기, 사각형의 모양 등과 같이 다양한 선택을 제공한다. 직사각형을 선택할 경우에는 외부 직사각형의 종횡비가 어떻게 돼야 하는지에 대한 선택이 다시 발생하기도 한다.

[0133] 가장 간단한 예시로, 너비와 높이라는 두 가지 정보만 선택한다고 가정한다. 예시 파일은 3:2 비율을 가지고 있고 분할은 3*2 = 6 블록으로 분할될 수 있다. 여기서 RM 매트릭스도 같은 종횡비를 가지며, 생성된 블록은 도 2와 같이 80 X 80크기의 정사각형이다. 즉, 도 2는 본 발명에 따른 화면과 같은 종횡비를 갖도록 생성된 블록을 나타낸다.

[0134] 알고리즘 1은 화면의 종횡비를 유지한 상태로 스크린을 가장 큰 넓이의 사각형 블록으로 분할하는 알고리즘이다. 위에 제시된 수학식 1 - 8은 수학적 분석의 일반 매개변수를 정의한다. 그 다음, 수학식 9 - 12를 통해 화면 크기를 나머지 없이 나눌 수 있는 GCD를 찾는다. 다음으로, 가장 큰 사각형 블록을 얻기 위해, 앞에서 얻은 GCD를 블록의 크기로 사용하고 그에 따라 너비, 높이 방향의 블록의 개수(RecHeight, RecWidth)를 계산한다.

[0135] 알고리즘에서 변수 GCD는 화면 너비와 높이의 최대 공약수의 값을 나타낸다. 예시에서의 GCD는 gcd(240, 160)=80이다.

[0136] 예외 경우로, 너비와 높이의 GCD가 1이면 분할이 없다.

[0137] 이외에, 보다 작은 크기의 사각형 블록, 예를 들어 40X40 블록 사이즈(RecWidth = 6, RecHeight = 4, TB = 24), 20X20 블록 사이즈(RecWidth = 10, RecHeight = 8, TB = 96)등이 선택 될 수도 있다.

[0138] 이 방법에서는 다른 매개변수인 블록 총 개수(MaxNumberOfBlacks)를 이용하여 분할 후 블록 수에 대한 상한값(MaxDB)을 설정한다.

[0139] 예외 처리로써 원하는 최대 블록(MaxDB) 수가 가장 큰 블록(TB) 수보다 작으면 전체 화면 자체를 반환한다.

[0140] 알고리즘 2 는 수학식 13 - 수학식 16을 기반으로 상세한 접근법을 보여준다. 도 3은 예제 이미지를 두 가지의 다른 수의 블록 개수로 분할한 것을 나타낸다. 여기서 두 분할 방법 모두 원본 이미지의 종횡비가 240 : 160 = 3 : 2, 6 : 4 = 15 : 10 = 3:2 으로 3:2를 유지하는 것을 알 수 있다. GCDdiv 의 모든 약수는 스크린의 너비와 높이를 모두 나누어 정사각형 블록으로 분할할 수 있다. 즉, 도 3은 본 발명에서, 예제 이미지를 두 가지의 다른 수의 블록 개수로 분할한 것을 나타낸다.

[0141] 요청하는 정사각형 블록의 수가 증가하게 되면 더 작은 크기의 블록으로 나누어지게 된다. 이때 step 6과 같이 최대 블록의 수(MaxDB)를 초과하지 않는 블록의 수로 나눌 수 있도록 해야 한다. 알고리즘의 최종 단계는 GCD가 1인 예외 상황을 처리하는 것이다.

[0142] 앞의 두 방법으로는, 스크린의 종횡비를 유지하는 정사각형 모양의 블록이 만들어진다. 예시를 통해 보면 이미지 사각형 분할은 6 (3×2), 24 (6×4), 96 (12×8), 150 (15×10), 384 (24×16), 600 (30×20), 1536 (48×32), 2400 (60×40), 9600 (120×80), 38400 (240×160)개의 블록들로 나누어진다.

[0143] 그러나 사용자는 정사각형 이외의 분할을 요청하기도 한다. 이를 위한 방법으로 수학식 17 - 수학식 29의 분석을 기반으로, 알고리즘 3은 단계별 임의 분할 과정을 나타내며, 상세한 흐름도는 도 4에 나와 있다. 즉, 도 4는

본 발명의 실시 예에 따른 상세한 흐름도이다.

[0144] 한편, 도 5는 본 발명의 실시 예에 따른 예제 이미지이다.

[0145] 도 5와 같이, 예제 이미지는 1×16, 16×1, 5×6, 15×16 블록으로 나눌 수 있다. 본 발명의 목표는 중복되지 않게 균등한 사이즈의 블록으로 경계와 최대한 근접하게 나누는 것이다. 예를 들어 너비 18, 높이 14만큼 블록을 나누게 되면 전체 블록 수는 18x14 = 252 블록이 되는 식이다. 이때 252는 최대 블록의 수(MaxDB)를 나타낸다. 이 때 약수 벡터는 수학식 30, 31과 같다.

[0146] [수학식 30]

[0147] WidthDivs=약수(240)=[1,2,3,4,5,6,8,10,12,15,16,20,24,30,40,48,60,80,120,240]

[0148] [수학식 31]

[0149] HeightDivs=약수(160)=[1,2,4,5,8,10,16,20,32,40,80,160]

[0150] 이러한 경우 우선 벡터 WidthDivs 와 HeighDivs 사이에서 18과 14에 가장 가까운 약수를 찾는다. 이 경우에는 16이 선택되고 양 방향에서 총 256(16x16)개의 블록(TB)을 만들고 이는 총수(MaxDB)인 252를 넘어가게 된다. 이후 알고리즘은 약수 벡터들 사이에서 15×10, 15×16, 15×20, 16×10, 16×20, 20×10, 20×16, 20×20 과 같은 가까운 블록 크기의 쌍을 확인한다. 위 쌍들 중에서 알고리즘은 15x16을 선택하였다 (도 5(d) 참조) 따라서 240개의 블록을 생성하고, 이는 252개를 넘지 않는다.

[0151] 만약 총 블록의 수나 양 방향에 존재하는 블록의 수 대신에 블록의 크기가 주어지는 경우 간단한 변형을 통해 알고리즘 2 혹은 알고리즘 3을 이용할 수 있다.

[0152] 알고리즘 4와 알고리즘 5가 방금 언급한 절차에 대한 것을 나타낸다.

[0153] 데스크톱 모션 탐지를 학습 연구로 사용하고 다양한 실험 평가를 통해 우리는 이 분할 전략이 분류 성능에 주는 영향을 알아보았다. 몇가지 동적/애니메이션 실행 및 정적/문서 실행 영역을 배열한 후 서로 다른 분할 환경에서 성능을 측정하고 평가하였다. 이 결과로 블록 수가 증가하고 비율이 일정하지는 않지만 비율 편차가 감소할수록 성능이 증가함을 알 수 있다. 한편 블록 수의 증가는 자원 사용량을 엄청나게 증가시키는 반면에 블록 수를 동일하게 유지하는 비율만 조절해 준다면 시스템 자원 사용량에 큰 영향을 미치지 않게 된다. 여기에 제안하는 분할 알고리즘이 많은 어플리케이션에 적용 가능하고 쉽게 자동화 될 수 있다. 이는 런타임 검증을 통해 특정 시나리오에 적합한 분할 방법을 찾을 수 있다. 예를 들어 원격 데스크톱 제공 솔루션은 동적 분할을 통합하여 더 나은 QoE를 지원할 수 있다.

[0154] 한편, 도 6a 내지 도 6c는 이상에서 전술된 본 발명에 따른 알고리즘 1 내지 알고리즘 5에 따른 동적 분할 방법의 흐름도를 도시한다. 한편, 이러한 동적 분할 방법은 데스크톱 스크린을 포함하는 이미지 처리 장치(또는 동적 분할 장치)의 제어부 또는 기타 구성요소에 의해 수행될 수 있다. 예를 들어, 이러한 동적 분할 방법은 이미지 처리 장치의 제어부에 의해 수행될 수 있다.

[0155] 구체적으로, 도 6a는 알고리즘 1에 따른 동적 분할 방법의 흐름도를 도시한다. 도 6a에 도시된 바와 같이, 알고리즘 1에 따른 동적 분할 방법은 입력 정보 수신 과정(S110), 최대 공약수 연산 과정(S120), 및 블록 사이즈 결정 과정(S130)을 포함한다. 또한, 상기 동적 분할 방법은 제1 픽셀 정보 매트릭스 생성 과정(S140), 제2 픽셀 정보 매트릭스 생성 과정(S150), 이미지 프로세싱 과정(S160) 및 이미시 표시 과정(S170)을 더 포함한다.

[0156] 한편, 알고리즘 1에 따른 동적 분할 방법은 아래의 표 1과 도 1에 표시된 바와 같은 알고리즘에 따라 수행된다.

표 1

[0157]

Algorithm1: SplitAspectRatio(ScreenHeight, ScreenWidth)
Input: Height and Width of the screen
Output: Number of screen blocks in height direction(RecHeight) and in width direction (RecWidth)

1: GCD=GreatestCommonDivisor(ScreenHeight, ScreenWidth)
2: RecHeight = ScreenHeight/GCD
3: RecWidth = ScreenWidth/GCD

[0158] 먼저, 입력 정보 수신 과정(S110)에서, 화면에 대한 화면 높이(SH: ScreenHeight)와 화면 너비(SW: ScreenWidth)에 관한 입력 정보를 수신한다. 다음으로, 최대 공약수 연산 과정(S120)에서, 상기 화면 높이(SH) 및 상기 화면 너비(SW)의 최대 공약수(GCD: Greatest Common Divisor)를 연산한다. 다음으로, 블록 사이즈 결정 과정(S130)에서, 상기 화면의 너비와 높이를 각각 상기 GCD보다 작은 값을 갖는 사이즈의 블록으로 분할하여, 상기 화면의 높이 방향과 너비 방향의 블록 개수(RecH: RecHeight, RecW: RecWidth)를 결정한다.

[0159] 한편, 제1 픽셀 정보 매트릭스 생성 과정(S140)에서, 상기 블록 개수에 따라 분할된 각 블록에 대한 픽셀 정보 매트릭스인 제1 픽셀 정보 매트릭스를 생성한다. 다음으로, 제2 픽셀 정보 매트릭스 생성 과정(S150)에서, 상기 각 블록 내부의 픽셀에 대한 픽셀 정보 매트릭스인 제2 픽셀 정보 매트릭스를 생성한다. 이때, 상기 각 블록에 대한 픽셀 정보 매트릭스는 전술된 수학식 6과 같이 결정될 수 있다. 또한, n행의 k번째 블록은 B_{nk} 로 표현되고, 상기 B_{nk} 에 대한 제2 픽셀 정보 매트릭스 BM_{nk}는 전술된 수학식 7과 같이 결정될 수 있다.

[0160] 한편, 이미지 프로세싱 과정(S160)에서, 상기 분할된 각 블록에 대하여, 상기 제1 픽셀 정보 매트릭스에 스케일링이나 프로세싱을 위한 서로 다른 상수 또는 서로 다른 매트릭스 연산을 적용할 수 있다. 또한, 이미지 프로세싱 과정(S160)에서, 상기 각 블록 내의 픽셀에 대하여, 상기 제2 픽셀 정보 매트릭스에 스케일링이나 프로세싱을 위한 동일한 상수 또는 동일한 매트릭스 연산을 적용할 수 있다. 이러한, 서로 다른 상수 또는 서로 다른 매트릭스 연산을 적용하는 과정과 동일한 상수 또는 동일한 매트릭스 연산을 적용하는 과정은 독립적으로 수행되거나, 또는 어느 한 과정의 결과를 이용하여 나머지 과정을 수행할 수 있다. 또한, 이러한 과정들이 독립적으로 수행되는 경우에도, 프로세서의 개수 또는 프로세싱 방법에 따라 병렬적으로 수행되거나, 또는 순차적으로 수행될 수 있다. 또한, 이미시 표시 과정(S170)에서, 상기 동일하거나 서로 다른 상수 또는 매트릭스 연산이 적용된 제1 및 제2 픽셀 정보 매트릭스에 의해 스케일링 또는 프로세싱된 픽셀을 포함하는 이미지가 표시되도록 제어한다.

[0161] 구체적으로, 도 6b는 알고리즘 2 또는 알고리즘 4에 따른 동적 분할 방법의 흐름도를 도시한다. 도 6b에 도시된 바와 같이, 알고리즘 2 또는 알고리즘 4에 따른 동적 분할 방법은 입력 정보 수신 과정(S210), 최대 공약수 및 약수 연산 과정(S220), 및 블록 사이즈 결정 과정(S230)을 포함한다. 한편, 도 6b에 표시되지 않았지만, 도 6a에서의 제1 픽셀 정보 매트릭스 생성 과정(S140), 제2 픽셀 정보 매트릭스 생성 과정(S150), 이미지 프로세싱 과정(S160) 및 이미시 표시 과정(S170)이 도 6b에 더 포함될 수 있다.

[0162] 한편, 알고리즘 2 또는 알고리즘 4에 따른 동적 분할 방법은 아래의 표 2 또는 표 3과 도 2 또는 도 4에 표시된 바와 같은 알고리즘에 따라 수행된다.

표 2

[0163]

```
Algorithm2: SplitAspectRatioGivenBlocks(ScreenHeight, ScreenWidth, MaxNumberOfBlocks)
Input: Height and Width of the screen and the maximum number of blocks
Output: Number of screen blocks in height direction (RecHeight) and in width direction
(RecWidth)
 1: GCD=GreatestCommonDivisor(ScreenHeight , ScreenWidth)
 2: GCDdivs=divisors(GCD)
 3: for i=1:length(GCDdivs)
 4:     RecHeight=vidHeight/GCDdivs(i)
 5:     RecWidth=vidWidth/GCDdivs(i)
 6:     if (MaxNumberOfBlocks>=RecHeight*RecWidth)
 7:         return
 8:     endif
 9: end of step 3 loop
10: RecHeight=RecWidth= 1
```

표 3

[0164]

```
Algorithm4: SplitAspectRatioGivenBlocks(ScreenHeight, ScreenWidth, BS)
Input: Height and Width of the screen and the Desired Block Dimension
Output: Number of screen blocks in height direction (RecHeight) and in width direction (RecWidth)
  1: MaxNumberOfBlocks= ceil(ScreenHeight/BS)*ceil( ScreenWidth/BS)
2: GCD=GreatestCommonDivisor(ScreenHeight , ScreenWidth)
  3: GCDdivs=divisors(GCD)
  4: for i=1:length(GCDdivs)
  5:     RecHeight=vidHeight/GCDdivs(i)
  6:     RecWidth=vidWidth/GCDdivs(i)
  7:     if (MaxNumberOfBlocks>=RecHeight*RecWidth)
  8:           return
  9:     endif
 10: end of step 3 loop
 11: RecHeight=RecWidth= 1
```

[0165] 먼저, 입력 정보 수신 과정(S210)에서, 화면에 대한 화면 높이(SH: ScreenHeight)와 화면 너비(SW: ScreenWidth)에 관한 입력 정보를 수신한다. 한편, 표 2에 도시된 바와 같이, 알고리즘 2에 따르면, 입력 정보 수신 과정(S210)에서, 최대 블록의 수(MaxDB)에 대한 정보를 더 수신할 수 있다. 또한, 표 3에 도시된 바와 같이, 알고리즘 4에 따르면, 입력 정보 수신 과정(S210)에서, 원하는 블록 사이즈 (BS: block size) 정보를 더 수신할 수 있다.

[0166] 다음으로, 최대 공약수 및 약수 연산 과정(S220)에서, 상기 화면 높이(SH) 및 상기 화면 너비(SW)의 최대 공약수(GCD: Greatest Common Divisor)와 상기 GCD의 약수들(GCDdivs)을 연산한다. 다음으로, 블록 사이즈 결정 과정(S230)에서, 상기 화면의 너비와 높이를 상기 약수들(GCDdivs) 중 하나의 사이즈의 블록으로 분할하여, 상기 화면의 높이 방향과 너비 방향의 블록 개수(RecH: RecHeight, RecW: RecWidth)를 결정한다.

[0167] 한편, 블록 사이즈 결정 과정(S230)은, 제1 결정 과정 (S231), 블록 개수 판단 과정(S232), 제2 결정 과정(S233), 및 제3 결정 과정(S234)을 더 포함할 수 있다. 알고리즘 2에 따라, 표 2를 참조하면, 제1 결정 과정(S231)에서, 상기 약수들(GCDdivs)의 개수만큼 인덱스(i)를 증가시키면서, 상기 블록 개수(RecH, RecW)를 결정할 수 있다. 또한, 블록 개수 판단 과정(S232)에서, 상기 블록 개수(RecH, RecW)의 곱에 해당하는 전체 블록 개수가 상기 최대 블록의 수(MaxDB) 이상인지를 판단할 수 있다. 또한, 제2 결정 과정(S233)에서, 상기 전체 블록 개수가 상기 최대 블록의 수(MaxDB) 미만이면, 상기 제1 결정 과정(S231)에서 결정된 상기 블록 개수(RecH, RecW)를 반환하도록 결정할 수 있다. 한편, 제3 결정 과정(S234)에서, 상기 전체 블록 개수가 상기 최대 블록의 수(MaxDB) 이상이면, 임의의 수의 블록 개수를 반환하도록 결정할 수 있다. 이때, 임의의 수의 블록 개수는 RecH = 1, 및 RecW =1 로 결정되어, 분할되지 않은 전체 화면이 결정될 수 있다.

[0168] 한편, 알고리즘 4에 따라, 표 3을 참조하면, 제1 결정 과정(S231)에서, 상기 약수들(GCDdivs)의 개수만큼 인덱스(i)를 증가시키면서, 상기 블록 개수(RecH, RecW)를 결정할 수 있다. 또한, 블록 개수 판단 과정(S232)에서, 상기 블록 개수(RecH, RecW)의 곱에 해당하는 전체 블록 개수가 최대 블록의 수(MaxDB) 이상인지를 판단할 수 있다. 이때, 상기 최대 블록의 수(MaxDB)는 ceil(SH/BS) * ceil (SW/BS)로 결정되고, ceil()는 소수점에 대한 올림 함수일 수 있다. 또한, 제2 결정 과정(S233)에서, 상기 전체 블록 개수가 상기 최대 블록의 수(MaxDB) 미만이면, 상기 제1 결정 과정에서 결정된 상기 블록 개수(RecH, RecW)를 반환하도록 결정할 수 있다. 한편, 제3 결정 과정(S234)에서, 상기 전체 블록 개수가 상기 최대 블록의 수(MaxDB) 이상이면, 임의의 수의 블록 개수를 반환하도록 결정할 수 있다. 이때, 임의의 수의 블록 개수는 RecH = 1, 및 RecW =1 로 결정되어, 분할되지 않은 전체 화면이 결정될 수 있다.

[0169] 한편, 알고리즘 2 또는 알고리즘 4에서, 상기 블록 개수(RecH, RecW)는 상기 화면의 종횡비(axial ratio)에 비례하여 결정되고, 상기 블록 각각은 정사각형 형태인 것을 특징으로 할 수 있다.

[0170] 구체적으로, 도 6c는 알고리즘 3 또는 알고리즘 5에 따른 동적 분할 방법의 흐름도를 도시한다. 도 6b에 도시된 바와 같이, 알고리즘 3 또는 알고리즘 5에 따른 동적 분할 방법은 입력 정보 수신 과정(S310), 약수 연산 과정

(S320), 약수 탐색 과정(S325), 및 블록 사이즈 결정 과정(S330)을 포함한다. 한편, 도 6c에 표시되지 않았지만, 도 6a에서의 제1 픽셀 정보 매트릭스 생성 과정(S140), 제2 픽셀 정보 매트릭스 생성 과정(S150), 이미지 프로세싱 과정(S160) 및 이미시 표시 과정(S170)이 도 6c에 더 포함될 수 있다.

[0171] 한편, 알고리즘 3 또는 알고리즘 5에 따른 동적 분할 방법은 아래의 표 3 또는 표 5와 도 3 또는 도 5에 표시된 바와 같은 알고리즘에 따라 수행된다.

표 4

[0172]

Algorithm3: SplitArbitrary (ScreenHeight, ScreenWidth , DHeight , DWidth)
Input: Height and Width of the screen, Number of desired blocks toward height and width
Output: Number of screen blocks in height direction (RecH) and in width direction (RecW)

1. MaxDB= DHeight * DWidth
2. HeightDivs=divisors(ScreenHeight)
3. WidthDivs=divisors(ScreenWidth)
4. Find the divisor HeightDivs[I] in HeightDivs vector which is closest to DHeight
5. Find the divisor WidthDivs[J] in WidthDivs vector which is closest to DWidth
6. Calculate the Total Blocks, TB=HeightDivs[I]*WidthDivs[J], then replace the values of I,J with the pairs (I-1,J), (I,J-1), (I+1,J), (I,J+1), (I-1,J-1) and compute the corresponding TBs. Discard the values where any of the element in the pair does not exist.
7. Finally, find the TB that is closest to but not exceeds MaxNumberOfBlocks and select the corresponding height and width divisor as the values of RecH and RecW.

표 5

[0173]

Algorithm5: SplitArbitraryGivenBlocksDimension (ScreenHeight, ScreenWidth , BH, BW)
Input: Height and Width of the screen, desired Block Height and Width
Output: Number of screen blocks in height direction (RecH) and in width direction (RecW)

1. DHeight= ceil(ScreenHeight/BH)
2. DWidth = ceil(ScreenHeight/BW)
3. MaxDB= DHeight * DWidth
4. HeightDivs=divisors(ScreenHeight)
5. WidthDivs=divisors(ScreenWidth)
6. Find the divisor HeightDivs[I] in HeightDivs vector which is closest to DHeight
7. Find the divisor WidthDivs[J] in WidthDivs vector which is closest to DWidth
8. Calculate the Total Blocks, TB=HeightDivs[I]*WidthDivs[J], then replace the values of I,J with the pairs (I-1,J), (I,J-1), (I+1,J), (I,J+1), (I-1,J-1) and compute the corresponding TBs. Discard the values where any of the element in the pair does not exist.
9. Finally, find the TB that is closest to but not exceeds MaxNumberOfBlocks and select the corresponding height and width divisor as the values of RecH and RecW.

[0174] 먼저, 입력 정보 수신 과정(S310)에서, 화면에 대한 화면 높이(SH: ScreenHeight), 화면 너비(SW: ScreenWidth), 제1 블록 개수(DHeight, DWidth) 또는 제1 블록 사이즈(BH: BlockHeight, BW:BlockSize)에 대한 정보를 수신한다. 이때, 표 4에 도시된 바와 같이, 알고리즘 3에 따르면, 입력 정보 수신 과정(S310)에서, 화면에 대한 화면 높이(SH: ScreenHeight), 화면 너비(SW: ScreenWidth), 및 제1 블록 개수(DHeight, DWidth) 에 대한 정보를 수신한다. 반면에, 표 5에 도시된 바와 같이, 알고리즘 5에 따르면, 입력 정보 수신 과정(S310)에서, 화면에 대한 화면 높이(SH: ScreenHeight), 화면 너비(SW: ScreenWidth), 및 제1 블록 사이즈(BH: BlockHeight, BW:BlockSize)에 대한 정보를 수신한다.

[0175] 다음으로, 약수 연산 과정(S320)에서, 상기 화면 높이(SH)와 상기 화면 너비(SW)에 대한 약수들(WidthDivs, HeightDivs)을 연산한다. 다음으로, 약수 탐색 과정(S325)에서, 상기 연산된 약수들(HeightDivs, WidthDivs) 중에서 상기 제1 블록 개수(DHeight, DWidth)에 가장 가까운 약수(HeightDivs[I], WidthDivs[J])들을 탐색한다. 다음으로, 블록 사이즈 결정 과정(S330)에서, 상기 화면의 너비와 높이를 각각 상기 탐색된 약수(HeightDivs[I], WidthDivs[J])에 해당하는 개수로 분할하여, 상기 화면의 높이 방향과 너비 방향의 제2 블록 개수(RecH: RecHeight, RecW: RecWidth)를 결정한다.

[0176] 한편, 표 4 또는 표 5에 따른 알고리즘 3 또는 알고리즘 5에 따른 동적 분할 방법은 블록 개수를 판단하여 최대

블록 수 이상이면 인접한 조합을 선택하는 다음과 같은 기술적 특징을 갖는다. 즉, 약수 탐색 과정(S325) 이후에, 상기 제1 블록 개수(DHeight, DWidth)의 곱이 최대 블록의 수(MaxDB) 이상인지를 판단하는 블록 개수 판단 과정(S326)을 더 포함한다. 이때, 최대 블록의 수(MaxDB)는 상기 제1 블록 개수(DHeight, DWidth)의 곱으로 결정될 수 있다.

[0177] 또한, 상기 제1 블록 개수(DHeight, DWidth)의 곱이 상기 최대 블록의 수(MaxDB) 이상이면, 상기 탐색된 약수(Heightdivs[I], Widthdivs[J])들의 인접한 약수들(HeightDivs[I±k], WidthDivs[J±k]) 중에서 하나를 선택하는 인접 약수 선택 과정(S327)을 더 포함한다. 예를 들어, 인접한 약수들은 (I-1,J), (I,J-1), (I+1,J), (I,J+1), (I-1,J-1)의 인덱스 중 하나를 가질 수 있다. 또한, 블록 사이즈 결정 과정(S330)에서, 상기 화면의 너비와 높이를 각각 상기 선택된 약수에 해당하는 개수로 분할하여, 상기 화면의 높이 방향과 너비 방향의 제2 블록 개수(RecH: RecHeight, RecW: RecWidth)를 결정할 수 있다. 한편, 블록 개수 판단 과정(S326)에서 상기 제1 블록 개수(DHeight, DWidth)의 곱이 최대 블록의 수(MaxDB) 미만이면, 블록 사이즈 결정 과정(S330)이 바로 수행될 수 있다. 또는, 블록 개수 판단 과정(S326)에서 상기 제1 블록 개수(DHeight, DWidth)의 곱이 최대 블록의 수(MaxDB) 이상이면, 인접 약수 선택 과정(S327) 이후에 블록 사이즈 결정 과정(S330)이 수행될 수 있다.

[0178] 한편, 표 5에 따른 알고리즘 5를 참조하면, 제1 블록 사이즈(BH: BlockHeight, BW:BlockSize)에 대한 정보만 수신되고, 제1 블록 개수(DHeight, DWidth)에 대한 정보는 수신되지 않고 별도록 계산된다. 이때, 상기 제1 블록 개수(DHeight, DWidth)는 표 5에 표시된 바와 같이, 각각 ceil(SH/BH) 및 ceil(SW/BH)로 결정되고, ceil()는 소수점에 대한 올림 함수이다.

[0179] 이상에서는, 본 발명에 따른 동적 분할 방법에 대해 살펴보았다. 아래에서는, 이러한 동적 분할 방법을 수행하는 동적 분할 장치 (또는 이미지 표시 장치)에 대해 살펴보기로 하자. 한편, 전술된 동적 분할 방법에 대한 내용은 동적 분할 장치에 대한 내용과 결합되어 이용될 수 있음은 물론이다.

[0180] 이와 관련하여, 도 7은 본 발명에 따른 동적 분할 장치 (또는 이미지 표시 장치)의 세부 구성을 도시한다. 도 7을 참조하면, 동적 분할 장치 (또는 이미지 표시 장치)(700)는 인터페이스(710), 제어부(720), 메모리(730) 및 디스플레이(740)를 포함한다. 여기서, 인터페이스(710)는 물리적인 인터페이스뿐만 아니라, 프로그램 (또는 계층 간)의 가상 인터페이스를 포함할 수 있다. 예를 들어, 인터페이스(710)는 응용 프로그램 (application program)간의 인터페이스인 API (Application Program Interface)일 수 있다. 여기서, 제어부(720)는 임의의 프로세서일 수 있고, 소프트웨어 및/또는 하드웨어를 제어하는 프로세서로서, 응용 프로그램을 제어하는 임의의 프로세서와 이러한 기능을 수행하는 임의의 프로그램 모듈을 포함할 수 있다.

[0181] 알고리즘 1에 따른 동적 분할 방법 (및/또는 이미지 표시 방법)을 수행하는 동적 분할 장치 (또는 이미지 표시 장치)(700)에 대해 살펴보면 다음과 같다. 인터페이스(710)는 화면에 대한 화면 높이(SH: ScreenHeight)와 화면 너비(SW: ScreenWidth)에 관한 입력 정보를 수신하도록 구성된다. 제어부(720)는 상기 화면 높이(SH) 및 상기 화면 너비(SW)의 최대 공약수(GCD: Greatest Common Divisor)를 연산하도록 구성된다. 또한, 제어부(720)는 상기 화면의 너비와 높이를 각각 상기 GCD보다 작은 값을 갖는 사이즈의 블록으로 분할하여, 상기 화면의 높이 방향과 너비 방향의 블록 개수(RecH: RecHeight, RecW: RecWidth)를 결정하도록 더 구성될 수 있다.

[0182] 또한, 제어부(720)는 상기 블록 개수에 따라 분할된 각 블록에 대한 픽셀 정보 매트릭스인 제1 픽셀 정보 매트릭스를 생성하도록 더 구성될 수 있다. 여기서, 상기 각 블록에 대한 픽셀 정보 매트릭스는 전술된 수학식 6과 같이 결정될 수 있다. 또한, 제어부(720)는 상기 각 블록 내부의 픽셀에 대한 픽셀 정보 매트릭스인 제2 픽셀 정보 매트릭스를 생성하도록 더 구성될 수 있다. 여기서, n행의 k번째 블록은 B_{nk} 로 표현되고, 상기 B_{nk} 에 대한 제2 픽셀 정보 매트릭스 BM_{nk}는, 전술된 수학식 7과 같이 결정될 수 있다.

[0183] 또한, 제어부(720)는 상기 분할된 각 블록에 대하여, 상기 제1 픽셀 정보 매트릭스에 스케일링이나 프로세싱을 위한 서로 다른 상수 또는 서로 다른 매트릭스 연산을 적용 및/또는 상기 각 블록 내의 픽셀에 대하여, 상기 제2 픽셀 정보 매트릭스에 스케일링이나 프로세싱을 위한 동일한 상수 또는 동일한 매트릭스 연산을 적용하도록 더 구성될 수 있다.

[0184] 메모리(730)는 제어부(720)에 의해 생성 및/또는 처리된 각종 데이터 및/또는 매트릭스에 관한 정보를 저장하도록 구성된다.

[0185] 디스플레이(740)는 제어부(720)와 연동하여, 상기 동일하거나 서로 다른 상수 또는 매트릭스 연산이 적용된 제1 및 제2 픽셀 정보 매트릭스에 의해 스케일링 또는 프로세싱된 픽셀을 포함하는 이미지를 표시한다.

[0186] 다음으로, 알고리즘 2 또는 알고리즘 4에 따른 동적 분할 방법 (및/또는 이미지 표시 방법)을 수행하는 동적 분할 장치 (또는 이미지 표시 장치)(700)에 대해 살펴보면 다음과 같다.

[0187] 인터페이스(710)는 전술한 바와 같이, 화면에 대한 화면 높이(SH: ScreenHeight)와 화면 너비(SW: ScreenWidth)에 관한 입력 정보를 수신하도록 구성된다. 한편, 알고리즘 2에 따른 동적 분할 방법을 수행하는 경우, 인터페이스(710)는 최대 블록의 수(MaxDB)에 대한 정보를 더 수신하도록 구성될 수 있다. 반면에, 알고리즘 4에 따른 동적 분할 방법을 수행하는 경우, 인터페이스(710)는 원하는 블록 사이즈 (BS: block size) 정보를 더 수신하도록 구성될 수 있다.

[0188] 제어부(720)는 상기 인터페이스로부터 수신한 상기 화면 높이(SH) 및 상기 화면 너비(SW)의 최대 공약수(GCD: Greatest Common Divisor)와 상기 GCD의 약수들(GCDdivs)을 연산하도록 구성된다. 또한, 제어부(720)는 상기 화면의 너비와 높이를 상기 약수들(GCDdivs) 중 하나의 사이즈의 블록으로 분할하여, 상기 화면의 높이 방향과 너비 방향의 블록 개수(RecH: RecHeight, RecW: RecWidth)를 결정하도록 구성된다.

[0189] 또한, 제어부(720)는 상기 분할된 각 블록에 대하여, 상기 제1 픽셀 정보 매트릭스에 스케일링이나 프로세싱을 위한 서로 다른 상수 또는 서로 다른 매트릭스 연산을 적용 및/또는 상기 각 블록 내의 픽셀에 대하여, 상기 제2 픽셀 정보 매트릭스에 스케일링이나 프로세싱을 위한 동일한 상수 또는 동일한 매트릭스 연산을 적용하도록 더 구성될 수 있다. 이때, 디스플레이(740)는 상기 동일하거나 서로 다른 상수 또는 매트릭스 연산이 적용된 제1 및 제2 픽셀 정보 매트릭스에 의해 스케일링 또는 프로세싱된 픽셀을 포함하는 이미지를 표시할 수 있다.

[0190] 한편, 알고리즘 2에 따른 동적 분할 방법을 수행하는 경우, 제어부(720)는, 상기 약수들(GCDdivs)의 개수만큼 인덱스(i)를 증가시키면서, 상기 블록 개수(RecH, RecW)를 결정하고, 상기 블록 개수(RecH, RecW)의 곱에 해당하는 전체 블록 개수가 상기 최대 블록의 수(MaxDB) 이상인지를 판단하고, 그리고 상기 전체 블록 개수가 상기 최대 블록의 수(MaxDB) 미만이면, 상기 결정된 블록 개수(RecH, RecW)를 반환하도록 결정하도록 더 구성될 수 있다.

[0191] 한편, 알고리즘 4에 따른 동적 분할 방법을 수행하는 경우, 제어부(720)는, 상기 약수들(GCDdivs)의 개수만큼 인덱스(i)를 증가시키면서, 상기 블록 개수(RecH, RecW)를 결정하고, 상기 블록 개수(RecH, RecW)의 곱에 해당하는 전체 블록 개수가 최대 블록의 수(MaxDB) 이상인지를 판단 – 상기 최대 블록의 수(MaxDB)는 ceil(SH/BS) * ceil (SW/BS)로 결정되고, ceil()는 소수점에 대한 올림 함수임 – 하고, 그리고 상기 전체 블록 개수가 상기 최대 블록의 수(MaxDB) 미만이면, 상기 결정된 블록 개수(RecH, RecW)를 반환하도록 결정하도록 더 구성될 수 있다.

[0192] 다음으로, 알고리즘 3 또는 알고리즘 5에 따른 동적 분할 방법 (및/또는 이미지 표시 방법)을 수행하는 동적 분할 장치 (또는 이미지 표시 장치)(700)에 대해 살펴보면 다음과 같다.

[0193] 인터페이스(710)는 전술한 바와 같이, 화면에 대한 화면 높이(SH: ScreenHeight)와 화면 너비(SW: ScreenWidth)에 관한 입력 정보를 수신하도록 구성된다. 한편, 알고리즘 3에 따른 동적 분할 방법을 수행하는 경우, 인터페이스(710)는 제1 블록 개수(DHeight, DWidth)에 대한 입력 정보를 더 수신하도록 구성된다. 반면에, 알고리즘 5에 따른 동적 분할 방법을 수행하는 경우, 인터페이스(710)는 제1 블록 사이즈(BH: BlockHeight, BW:BlockSize)에 대한 입력 정보를 더 수신하도록 구성된다.

[0194] 제어부(720)는 상기 인터페이스(710)로부터 수신한 상기 화면 높이(SH)와 상기 화면 너비(SW)에 대한 약수들(WidthDivs, HeightDivs)을 연산하도록 구성된다. 또한, 제어부(720)는 상기 연산된 약수들(HeightDivs, WidthDivs) 중에서 상기 제1 블록 개수(DHeight, DWidth)에 가장 가까운 약수(HeightDivs[I], WidthDivs[J])들을 탐색하도록 더 구성될 수 있다. 또한, 제어부(720)는 상기 화면의 너비와 높이를 각각 상기 탐색된 약수(HeightDivs[I], WidthDivs[J])에 해당하는 개수로 분할하여, 상기 화면의 높이 방향과 너비 방향의 제2 블록 개수(RecH: RecHeight, RecW: RecWidth)를 결정하도록 더 구성될 수 있다.

[0195] 또한, 제어부(720)는, 상기 제1 블록 개수(DHeight, DWidth)의 곱이 최대 블록의 수(MaxDB) 이상인지를 판단하고, 상기 제1 블록 개수(DHeight, DWidth)의 곱이 상기 최대 블록의 수(MaxDB) 이상이면, 상기 탐색된 약수(Heightdivs[I], Widthdivs[J])들의 인접한 약수들(HeightDivs[I±k], WidthDivs[J±k]) 중에서 하나를 선택하도록 더 구성될 수 있다. 이때, 상기 제어부는, 상기 최대 블록의 수(MaxDB)를 상기 제1 블록 개수(DHeight, DWidth)의 곱으로 결정할 수 있다. 또한, 제어부(720)는, 상기 화면의 너비와 높이를 각각 상기 선택된 약수에 해당하는 개수로 분할하여, 상기 화면의 높이 방향과 너비 방향의 제2 블록 개수(RecH: RecHeight, RecW: RecWidth)를 결정하도록 더 구성될 수 있다.

[0196] 한편, 알고리즘 5에 따라, 상기 인터페이스(710)가 제1 블록 사이즈(BH: BlockHeight, BW:BlockSize)에 대한 정보를 수신하면, 상기 제어부(720)는, 상기 제1 블록 개수(DHeight, DWidth)를 각각 ceil(SH/BH) 및 ceil(SW/BH)로 결정하고, 상기 최대 블록의 수(MaxDB)를 상기 제1 블록 개수(DHeight, DWidth)의 곱으로 결정할 수 있다.

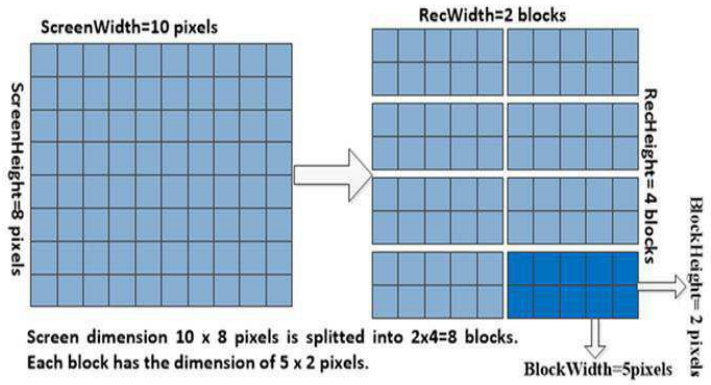
[0197] 이상에서, 데스크톱 스크린을 비중첩 동일 크기 블록으로 분할하기 위한 동적 분할 방법 (이미지 처리/표시 방법)과 이러한 방법을 수행하는 동적 분할 장치 (이미지 처리/표시 장치)에 대해 살펴보았다. 한편, 전술된 동적 분할 방법 (이미지 처리/표시 방법)은 전술된 단계의 순서에 따라 한정되는 것이 아니라, 필요에 따라 각 단계의 순서를 변경하여 자유롭게 실시될 수 있음은 물론이다.

[0198] 본 발명의 적어도 일 실시예에 따른 데스크톱 스크린을 비중첩 동일 크기 블록으로 분할하기 위한 동적 분할 알고리즘에 의하면 특정 어플리케이션에 대한 최적의 분할을 찾아 영상처리의 효율을 높이며, 상황에 맞게 자동적으로 적용할 수 있다는 장점이 있다.

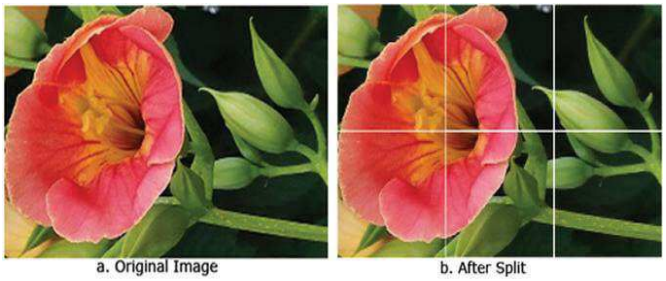
[0199] 소프트웨어적인 구현에 의하면, 본 명세서에서 설명되는 절차 및 기능뿐만 아니라 각각의 구성 요소들은 별도의 소프트웨어 모듈로도 구현될 수 있다. 상기 소프트웨어 모듈들 각각은 본 명세서에서 설명되는 하나 이상의 기능 및 작동을 수행할 수 있다. 적절한 프로그램 언어로 쓰여진 소프트웨어 어플리케이션으로 소프트웨어 코드가 구현될 수 있다. 상기 소프트웨어 코드는 메모리에 저장되고, 제어부(controller) 또는 프로세서(processor)에 의해 실행될 수 있다.

도면

도면1



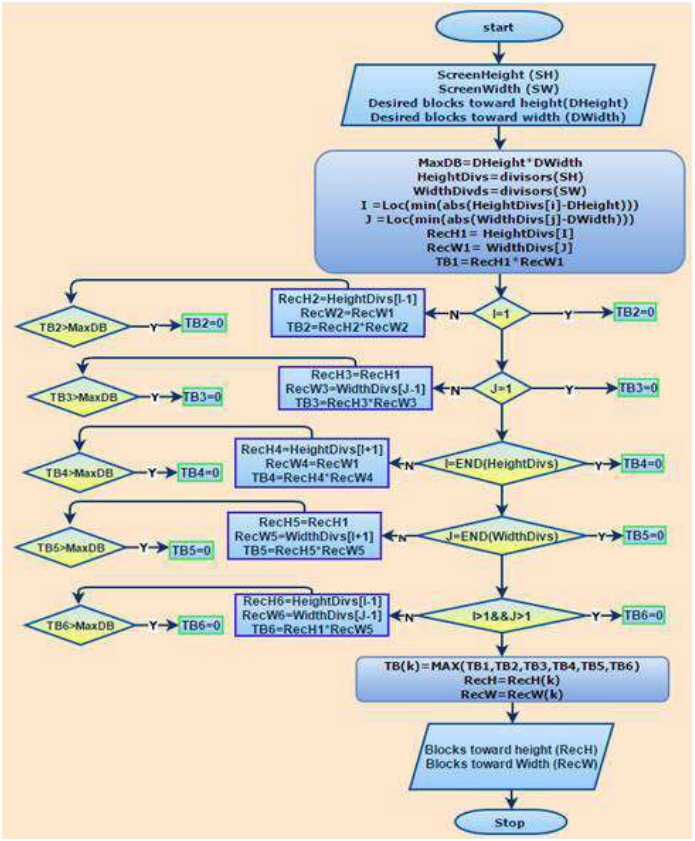
도면2



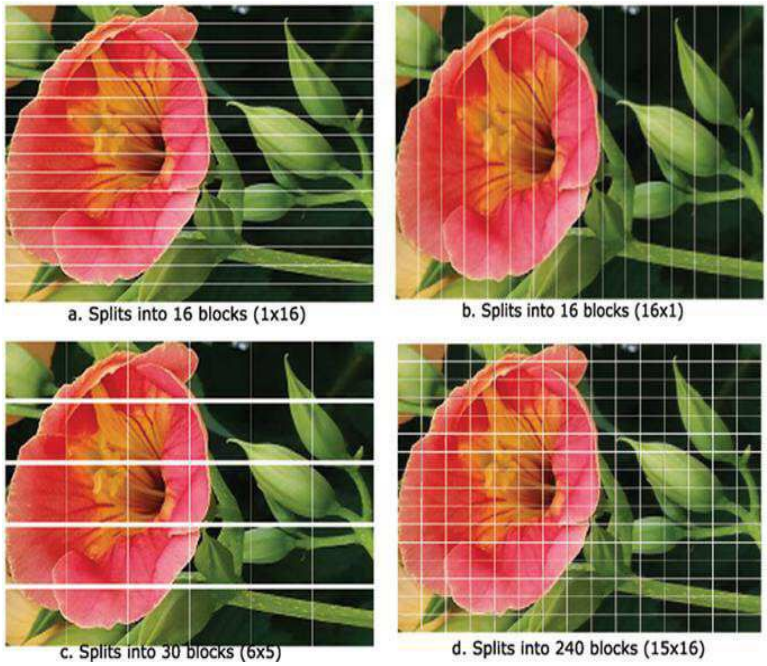
도면3


a. Generated 24 (6*5) square shaped blocks when MaxBlocks is set to 30
b. Generated 150 (15*10) square shaped blocks when MaxBlocks is set to 180

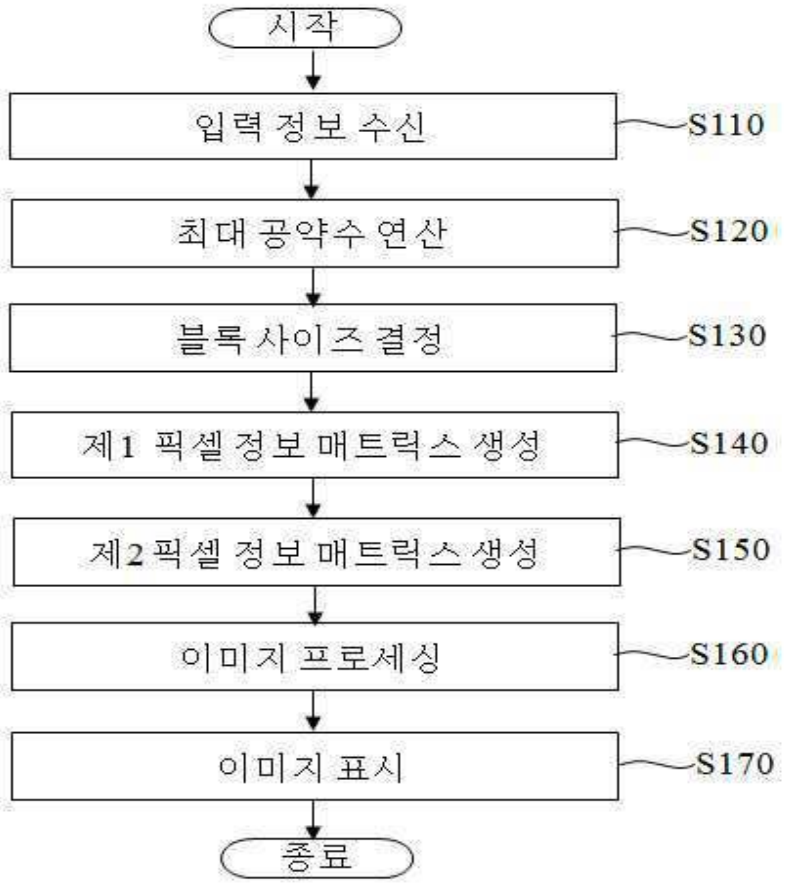
도면4


start
ScreenHeight (SH)
ScreenWidth (SW)
Desired blocks toward height(DHeight)
Desired blocks toward width (DWidth)
MaxDB=DHeight*DWidth
HeightDivs=divisors(SH)
WidthDivds=divisors(SW)
I =Loc(min(abs(HeightDivs[i]-DHeight)))
J =Loc(min(abs(WidthDivs[j]-DWidth)))
RecH1= HeightDivs[I]
RecW1= WidthDivs[J]
TB1=RecH1*RecW1
I=1
RecH2=HeightDivs[I-1]
RecW2=RecW1
TB2=RecH2*RecW2
TB2>MaxDB
TB2=0
J=1
RecH3=RecH1
RecW3=WidthDivs[J-1]
TB3=RecH3*RecW3
TB3>MaxDB
TB3=0
I=END(HeightDivs)
RecH4=HeightDivs[I+1]
RecW4=RecW1
TB4=RecH4*RecW4
TB4>MaxDB
TB4=0
J=END(WidthDivs)
RecH5=RecH1
RecW5=WidthDivs[J+1]
TB5=RecH5*RecW5
TB5>MaxDB
TB5=0
I>1&&J>1
RecH6=HeightDivs[I-1]
RecW6=WidthDivs[J-1]
TB6=RecH1*RecW5
TB6>MaxDB
TB6=0
TB(k)=MAX(TB1,TB2,TB3,TB4,TB5,TB6)
RecH=RecH(k)
RecW=RecW(k)
Blocks toward height (RecH)
Blocks toward Width (RecW)
Stop
Y
N

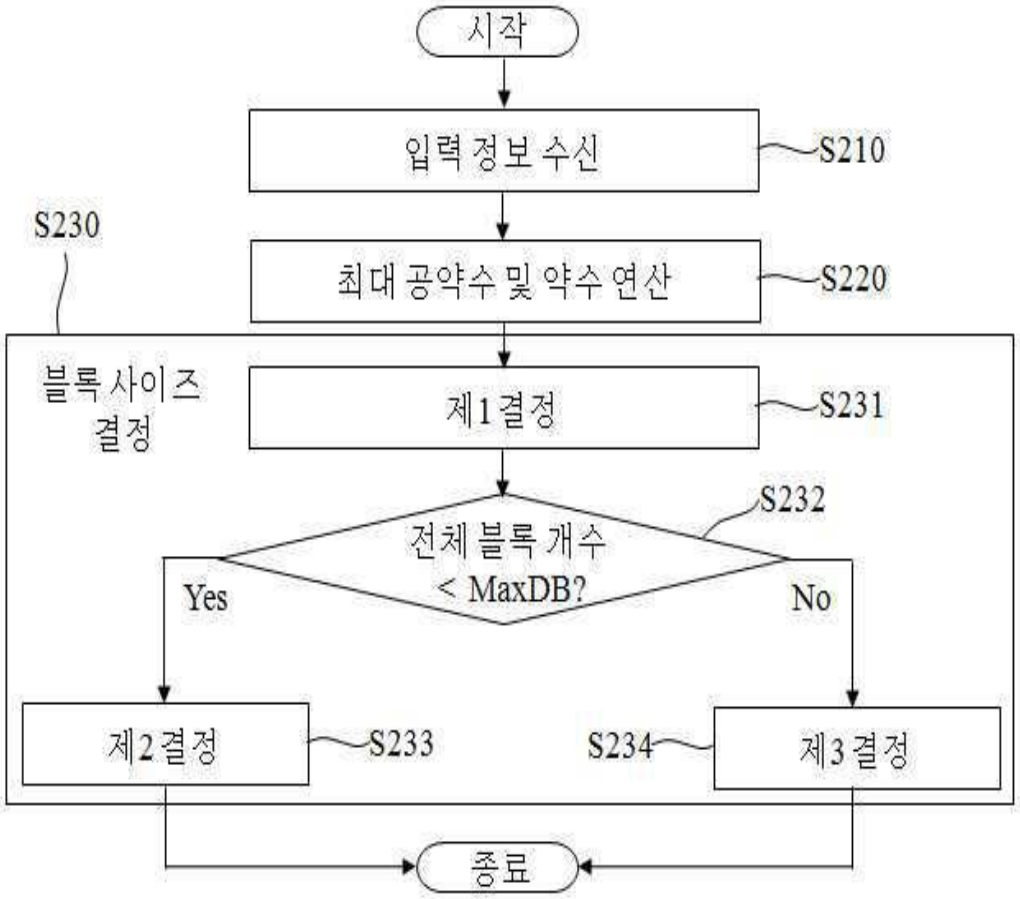
도면5


a. Splits into 16 blocks (1x16)
b. Splits into 16 blocks (16x1)
c. Splits into 30 blocks (6x5)
d. Splits into 240 blocks (15x16)

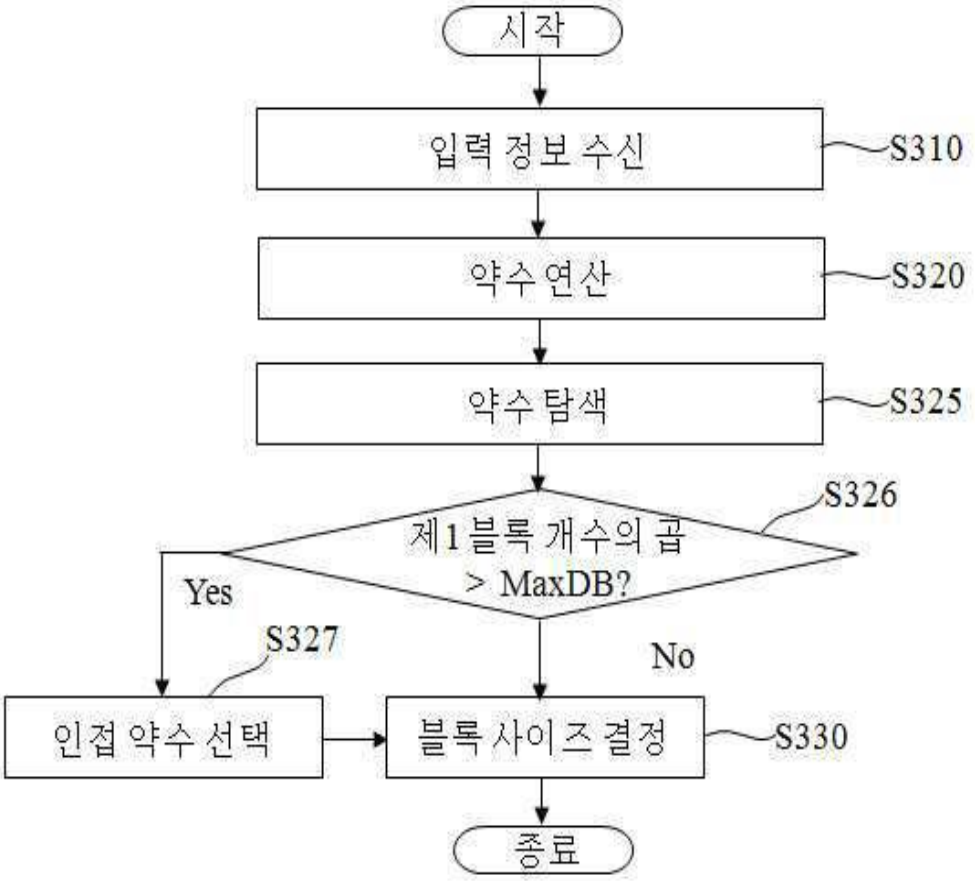
도면6a


시작
입력 정보 수신
S110
최대 공약수 연산
S120
블록 사이즈 결정
S130
제1 픽셀 정보 매트릭스 생성
S140
제2 픽셀 정보 매트릭스 생성
S150
이미지 프로세싱
S160
이미지 표시
S170
종료

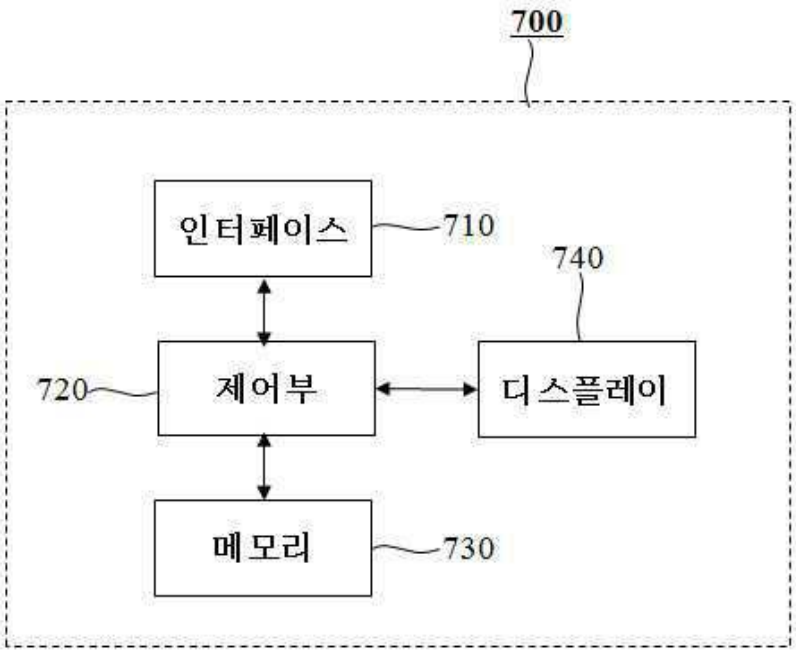
도면6b


시작
입력 정보 수신
S210
S230
최대 공약수 및 약수 연산
S220
블록 사이즈 결정
제1 결정
S231
S232
전체 블록 개수 < MaxDB?
Yes
No
제2 결정
S233
S234
제3 결정
종료

도면6c


시작
입력 정보 수신
S310
약수 연산
S320
약수 탐색
S325
제1블록 개수의 곱 > MaxDB?
S326
Yes
No
S327
인접 약수 선택
블록 사이즈 결정
S330
종료

도면7


700
인터페이스
710
740
720
제어부
디스플레이
메모리
730