



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2016년05월04일
 (11) 등록번호 10-1618384
 (24) 등록일자 2016년04월28일

- (51) 국제특허분류(Int. Cl.)
 G06F 15/16 (2006.01) G06F 15/17 (2006.01)
- (21) 출원번호 10-2010-7024957
- (22) 출원일자(국제) 2009년03월27일
 심사청구일자 2014년03월04일
- (85) 번역문제출일자 2010년11월05일
- (65) 공개번호 10-2011-0021755
- (43) 공개일자 2011년03월04일
- (86) 국제출원번호 PCT/US2009/038642
- (87) 국제공개번호 WO 2009/139963
 국제공개일자 2009년11월19일
- (30) 우선권주장
 12/119,415 2008년05월12일 미국(US)
- (56) 선행기술조사문헌
 US06252579 B1*
 US06268855 B1*
 US06473101 B1*
 *는 심사관에 의하여 인용된 문헌

- (73) 특허권자
 마이크로소프트 테크놀로지 라이선싱, 엘엘씨
 미국 워싱턴주 (우편번호 : 98052) 레드몬드 원
 마이크로소프트 웨이
- (72) 발명자
 사울, 엘튼
 미국 98052-6399 워싱턴주 레드몬드 원 마이크로
 소프트 웨이
 슈미에더, 로버트 윌헬름
 미국 98052-6399 워싱턴주 레드몬드 원 마이크로
 소프트 웨이
- (74) 대리인
 제일특허법인

전체 청구항 수 : 총 17 항

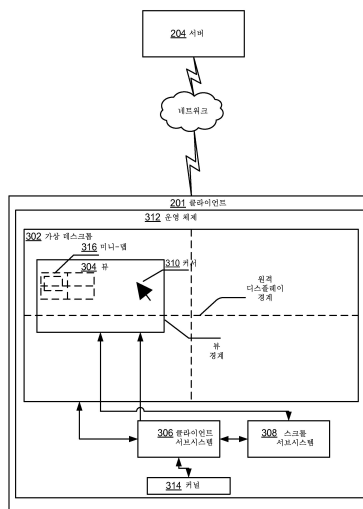
심사관 : 박미정

(54) 발명의 명칭 **가상 데스크톱 뷰 스크롤**

(57) 요약

본 개시의 예시적인 실시예들은 클라이언트 상의 가상 데스크톱의 뷰를 스크롤하는 것에 관한 것에 관련된다. 클라이언트는 디스플레이 드라이버를 갖는 서버와 통신할 수 있다. 서버는 이미지들을 나타내는 신호들을 클라이언트에 전송하도록 구성될 수 있고 클라이언트는 가상 데스크톱 내에 이미지들을 드로잉하도록 구성될 수 있다. 클라이언트는 가상 데스크톱의 일 부분을 포함하는 가상 데스크톱의 뷰를 렌더링하고, 사용자 입력을 추적하며, 사용자 입력에 응답하여 뷰를 패닝하도록 구성될 수 있다.

대표도 - 도3



명세서

청구범위

청구항 1

가상 데스크톱을 탐색(navigation)하는 방법으로서,

원격 컴퓨터로부터, 경계 박스(bounding box)를 기술하는 정보를 수신하는 단계- 상기 경계 박스는 가상 데스크톱의 크기를 정의하고, 상기 가상 데스크톱의 크기는 상기 원격 컴퓨터 상의 제1 디스플레이 드라이버의 제1 디스플레이 해상도 정보와 상기 원격 컴퓨터 상의 제2 디스플레이 드라이버의 제2 디스플레이 해상도 정보와 상기 제1 디스플레이 드라이버에 의해 생성된 이미지들이 상기 제2 디스플레이 드라이버에 의해 생성된 이미지들에 대해 어떻게 디스플레이되는지를 정의하는 관계에 기초함 -

상기 경계 박스를 기술하는 상기 수신된 정보에 따라 상기 가상 데스크톱을 생성하는 단계,

상기 원격 컴퓨터로부터 수신된 그래픽을 상기 가상 데스크톱에 드로잉하는 단계,

상기 가상 데스크톱의 뷰(view)를 렌더링하는 단계- 상기 뷰는 경계들(edges)에 의해 정의되며 상기 가상 데스크톱의 일부분을 보여줌 -,

상대적 커서 이동 정보를 수신하는 단계,

상기 상대적 커서 이동 정보와 상기 뷰의 경계들로부터, 커서가 상기 뷰의 경계에 도달했음을 판정하는 단계,

상대적 커서 이동의 방향으로 상기 가상 데스크톱의 상기 뷰를 스크롤하는 단계,

상기 뷰를 정의하는 경계들과 상기 제1 디스플레이 해상도 정보 및 상기 제2 디스플레이 해상도 정보를 정의하는 경계들로부터, 상기 뷰의 경계가 상기 제1 디스플레이 해상도 정보의 경계에 도달하였음을 판정하는 단계,

상기 커서의 속도가 미리 정해진 값보다 크다는 판정에 응답하여, 상기 가상 데스크톱의 뷰를 상기 제2 디스플레이 해상도 정보와 연관된 먼쪽 경계(a far edge)로 스크롤하는 단계

를 포함하는 방법.

청구항 2

제1항에 있어서,

상기 상대적 커서 이동 정보를 수신하는 단계 이후에, 상기 상대적 커서 이동 정보에 선형 가속 인자(linear accelerating factor)를 적용하는 단계를 더 포함하는 방법.

청구항 3

제1항에 있어서,

상기 상대적 커서 이동 정보를 수신하는 단계 이후에,

상기 커서의 속도를 판정하는 단계,

동적 가속 인자(dynamic accelerating factor)를 생성하는 단계- 상기 동적 가속 인자는 상기 커서의 속도의 함수임 - 및

상기 상대적 커서 이동 정보에 상기 동적 가속 인자를 적용하는 단계

를 더 포함하는 방법.

청구항 4

제1항에 있어서,

상기 상대적 커서 이동 정보를 수신하는 단계 이후에,

상기 커서의 좌표 정보와 상기 제1 디스플레이 드라이버 및 상기 제2 디스플레이 드라이버를 정의하는 경계들로부터, 상기 커서가 상기 제1 디스플레이 드라이버의 경계에 도달했음을 판정하는 단계, 및 햅틱 응답(haptic response)을 생성하는 단계를 더 포함하는 방법.

청구항 5

제1항에 있어서, 상기 가상 데스크톱의 뷰를 상기 제2 디스플레이 해상도 정보와 연관된 먼쪽 경계로 스크롤하는 단계는 상기 가상 데스크톱의 뷰를 상기 먼쪽 경계로 점프시키는 단계를 포함하는 방법.

청구항 6

가상 데스크톱을 탐색하는 컴퓨터-판독가능 명령어들을 포함하는 컴퓨터 판독가능 저장 매체로서, 원격 컴퓨터로부터, 경계 박스를 기술하는 정보를 수신하는 명령어- 상기 경계 박스는 가상 데스크톱의 크기를 정의하고, 상기 가상 데스크톱의 크기는 상기 원격 컴퓨터 상의 제1 디스플레이 드라이버의 제1 디스플레이 해상도 정보와 상기 원격 컴퓨터 상의 제2 디스플레이 드라이버의 제2 디스플레이 해상도 정보와 상기 제1 디스플레이 드라이버에 의해 생성된 이미지들이 상기 제2 디스플레이 드라이버에 의해 생성된 이미지들에 대해 어떻게 디스플레이되는지를 정의하는 관계에 기초함 - 상기 경계 박스를 기술하는 상기 수신된 정보에 따라 상기 가상 데스크톱을 생성하는 명령어, 상기 원격 컴퓨터로부터 수신된 그래픽을 상기 가상 데스크톱에 드로잉하는 명령어, 원격 컴퓨터로부터 수신된 그래픽을 묘사하는 가상 데스크톱의 뷰를 렌더링하는 명령어- 상기 뷰는 픽셀 범위(a range of pixels)에 의해 정의되고 상기 가상 데스크톱은 원격 컴퓨터의 디스플레이 해상도에 대응하는 픽셀 범위에 의해 정의됨 - 상대적 커서 이동 정보를 수신하는 명령어, 상기 상대적 커서 이동 정보와 상기 뷰를 정의하는 상기 픽셀 범위로부터, 커서가 상기 뷰의 경계에 도달했음을 판정하는 명령어, 상대적 커서 이동의 방향으로 가상 데스크톱의 뷰를 스크롤하는 명령어, 상기 커서의 속도가 미리 정해진 값보다 크다는 판정에 응답하여, 상기 가상 데스크톱의 뷰를 상기 제2 디스플레이 해상도 정보와 연관된 먼쪽 경계로 스크롤하는 명령어를 포함하는 컴퓨터 판독가능 저장 매체.

청구항 7

제6항에 있어서, 상기 가상 데스크톱을 복수의 구역으로 나누는 명령어, 각각의 구역을 키보드의 키에 할당하는 명령어, 특정의 구역에 할당된 키의 선택을 수신하는 명령어, 및 상기 키의 선택에 응답하여 상기 뷰를 상기 특정의 구역 상의 중앙에 오도록 하는 명령어를 더 포함하는 컴퓨터 판독가능 저장 매체.

청구항 8

제6항에 있어서, 상기 뷰에 미니-맵을 렌더링하는 명령어를 더 포함하고, 상기 미니-맵은 상기 가상 데스크톱의 스케일링된 이미

지와 상기 뷰의 스케일링된 이미지를 보여주는

컴퓨터 판독가능 저장 매체.

청구항 9

제8항에 있어서,

상기 미니-맵은 상기 가상 데스크톱의 와이어프레임 이미지(wireframe image)와 상기 뷰의 와이어프레임 윤곽선(wireframe outline)을 포함하는

컴퓨터 판독가능 저장 매체.

청구항 10

제8항에 있어서,

사용자 입력을 상기 원격 컴퓨터로 전송하는 명령어- 상기 원격 컴퓨터는 상기 사용자 입력에 응답하여 프로그램을 실행하도록 구성됨 -,

상기 프로그램의 사용자 인터페이스를 나타내는 그래픽을 수신하는 명령어, 및

상기 프로그램의 사용자 인터페이스의 스케일링된 이미지로 상기 미니-맵을 갱신하는 명령어를 더 포함하는 컴퓨터 판독가능 저장 매체.

청구항 11

제10항에 있어서,

상기 가상 데스크톱 상에서의 상기 사용자 인터페이스의 위치를 나타내는 좌표를 수신하는 명령어,

사용자 입력을 수신하는 명령어, 및

상기 사용자 입력에 응답하여 상기 뷰를 상기 사용자 인터페이스 상의 중앙에 오도록 하는 명령어를 더 포함하는 컴퓨터 판독가능 저장 매체.

청구항 12

컴퓨터 시스템으로서,

원격 컴퓨터로부터, 경계 박스를 기술하는 정보를 수신하는 회로- 상기 경계 박스는 가상 데스크톱의 크기를 정의하고, 상기 가상 데스크톱의 크기는 상기 원격 컴퓨터 상의 제1 디스플레이 드라이버의 제1 디스플레이 해상도 정보와 상기 원격 컴퓨터 상의 제2 디스플레이 드라이버의 제2 디스플레이 해상도 정보와 상기 제1 디스플레이 드라이버에 의해 생성된 이미지들이 상기 제2 디스플레이 드라이버에 의해 생성된 이미지들에 대해 어떻게 디스플레이되는지를 정의하는 관계에 기초함 -

상기 경계 박스를 기술하는 상기 수신된 정보에 따라 상기 가상 데스크톱을 생성하는 회로,

상기 원격 컴퓨터로부터 수신된 그래픽을 상기 가상 데스크톱에 드로잉하는 회로,

가상 데스크톱의 뷰를 렌더링하는 회로- 상기 가상 데스크톱은 먼저 원격 컴퓨터의 디스플레이 해상도에 대응하는 제1 픽셀 범위에 의해 정의되고 상기 뷰는 제2 픽셀 범위에 의해 정의됨 -

커서 이동 정보를 수신하는 회로,

상기 커서 이동 정보와 상기 제2 픽셀 범위로부터, 커서가 상기 제2 픽셀 범위의 경계에 도달했는지를 판정하는 회로,

커서 이동의 방향으로 상기 뷰를 스크롤하는 회로, 및

상기 커서의 속도가 미리 정해진 값보다 크다는 판정에 응답하여, 상기 가상 데스크톱의 뷰를 상기 제2 디스플레이 해상도 정보와 연관된 먼쪽 경계로 스크롤하는 회로

를 포함하는 컴퓨터 시스템.

청구항 13

제12항에 있어서,

상기 뷰에 미니-맵을 렌더링하는 회로를 더 포함하고, 상기 미니-맵은 상기 가상 데스크톱의 스케일링된 이미지와 상기 뷰의 스케일링된 이미지를 보여주는

컴퓨터 시스템.

청구항 14

제13항에 있어서,

상기 가상 데스크톱 내에서 생성된 프로그램의 사용자 인터페이스의 픽셀 좌표를 수신하는 회로, 및

상기 미니-맵 내의 상기 프로그램의 사용자 인터페이스에 대한 식별자를 생성하는 회로를 더 포함하는 컴퓨터 시스템.

청구항 15

제12항에 있어서,

상기 뷰를 스크롤하는 회로는 상기 커서 이동 정보에 선형 가속 인자(linear accelerating factor)를 적용하는 회로를 더 포함하는

컴퓨터 시스템.

청구항 16

제12항에 있어서,

상기 뷰를 스크롤하는 회로는,

상기 커서의 속도를 판정하는 회로,

동적 가속 인자(dynamic accelerating factor)를 생성하는 회로 - 상기 동적 가속 인자는 상기 커서의 속도의 함수임 - 및

상기 커서 이동 정보에 상기 동적 가속 인자를 적용하는 회로를 더 포함하는

컴퓨터 시스템.

청구항 17

제12항에 있어서,

상기 커서가 상기 제1 디스플레이 드라이버의 디스플레이 해상도의 특정의 경계에 도달했는지를 판정하는 회로 - 상기 특정의 경계는 상기 제2 디스플레이 드라이버의 특정의 경계에 근접해 있음 -, 및

상기 뷰를 상기 제2 디스플레이 드라이버에 대응하는 상기 가상 데스크톱의 영역 상의 중앙에 오도록 하는 회로를 더 포함하는

컴퓨터 시스템.

청구항 18

삭제

청구항 19

삭제

청구항 20

삭제

발명의 설명

배경 기술

- [0001] 원격 컴퓨팅 시스템은 사용자가 원격 컴퓨터에 의해 호스팅되는 리소스에 액세스할 수 있게 해준다. 이러한 시스템에서, 필수 구성요소들은 클라이언트와 서버이다. 이러한 시스템에서, 서버는 프로그램들을 실행하고 사용자 인터페이스를 나타내는 신호들을, TCP/IP 프로토콜과 같은 통신 프로토콜을 사용하여, 연결되어 있는 클라이언트들로 전송한다. 각각의 연결되어 있는 클라이언트에는 세션, 예를 들면, 일련의 리소스들을 포함하는 실행 환경이 제공된다. 각각의 클라이언트는 사용자 입력을 나타내는 신호를 서버로 전송하고, 서버는 적절한 세션에 사용자 입력을 적용한다.
- [0002] 일반적으로, 클라이언트가 서버에 연결될 때, 서버의 디스플레이 드라이버(들)은 일정한 디스플레이 해상도로 사용자 인터페이스를 생성할 수 있다. 어떤 상황들에서는, 서버의 디스플레이 해상도가 클라이언트의 디스플레이 해상도보다 높을 수 있다. 이러한 아키텍처에서는 세션에 대한 사용자 인터페이스가 서버에 의해 생성되고 클라이언트에서는 단지 렌더링되기만 하기 때문에, 클라이언트에서 세션이 어떻게 보이는지를 변경하려면 클라이언트가 서버의 디스플레이 설정을 수정해야 한다.
- [0003] 그러나, 어떤 상황들에서는, 클라이언트에 대응하기 위해 디스플레이 설정을 수정하는 것이 바람직하지 않을 수 있다. 예를 들어, 클라이언트가 사용자에게 기술 지원을 제공하거나 사용자와 공동으로 프로젝트를 수행하기 위한 목적으로 원격 연결되어 있는 동안, 서버의 디스플레이는 사용자 인터페이스를 렌더링하도록 구성될 수 있다. 클라이언트에 대응하기 위해 서버의 디스플레이 설정이 조정된다면, 서버 운영자의 사용자 경험은 떨어질 것이다. 다른 상황에서는, 클라이언트의 최적 해상도가 서버의 디스플레이 드라이버에 의해 지원되지 않을 수도 있다. 이 상황에서는, 디스플레이 설정을 수정하는 것이 어쩌면 서버에 연결된 물리적 디스플레이를 손상시킬 수 있다.
- [0004] 현재의 기술은 사용자 인터페이스의 뷰를 스크롤 하기 위해 스크롤 막대, 예를 들면, 그래픽 위젯(graphical widget)을 제공한다. 스크롤 막대가 이동됨에 따라, 사용자 인터페이스의 뷰가 패닝(panning)된다. 하지만, 이 기술은 번거롭고 정상적인 워크플로를 방해한다. 일례로, 사용자 인터페이스의 한쪽 끝에서 다른 쪽 끝으로 파일을 드래그하려는 사용자는 사용자 인터페이스(UI)를 묘사한 부분의 경계로 파일을 드래그하고, 스크롤 막대를 이동시킨 후, 원하는 위치로 파일을 드래그해야 한다. 다른 일례에서, 여러 대의 물리적 모니터에 걸쳐 있는 사용자 인터페이스로 작업을 하는 사용자는, 그들의 관심의 초점을 변경할 때, 그들의 커서를 한 모니터에서 다른 모니터로 자연스럽게 이동시킨다. 사용자 인터페이스의 다른 부분들을 보기 위해 사용자가 스크롤 막대를 이동시켜야 하는 것은 이러한 자연스러운 워크플로를 방해한다. 따라서, 사용자 인터페이스를 탐색하는 보다 자연스럽게 직관적인 방법을 제공하는 것이 바람직할 것이다.

발명의 내용

과제의 해결 수단

- [0005] 본 발명의 예시적인 실시예는 원격 컴퓨터의 디스플레이 드라이버의 디스플레이 해상도를 포함하는 원격 컴퓨터로부터 수신된 기하 형태 정보(geometry information)로부터, 가상 데스크톱을 생성하는 단계, 원격 컴퓨터로부터 수신된 그래픽을 가상 데스크톱에 드로잉하는 단계, 가상 데스크톱의 뷰를 렌더링하는 단계 - 뷰는 경계들(edges)에 의해 정의되며 가상 데스크톱의 일부분을 보여줌 -, 및 상대적 커서 이동 정보를 수신하는 단계(이들로 제한되지 않음)를 포함하는 방법을 기술한다. 전술한 내용에 부가하여, 다른 측면들이 본 발명의 일부를 구성하는 특허청구범위, 첨부 도면 및 본문에 기술되어 있다.
- [0006] 본 발명의 예시적인 실시예는 원격 컴퓨터로부터 수신된 그래픽을 묘사하는 가상 데스크톱의 뷰를 렌더링하는 명령어들 - 뷰는 픽셀 범위에 의해 정의되고 가상 데스크톱은 원격 컴퓨터의 디스플레이 해상도에 대응하는 픽셀 범위에 의해 정의됨 -, 상대적 커서 이동 정보를 수신하는 명령어들, 상대적 커서 이동 정보와 뷰를 정의하는 픽셀 범위로부터, 커서가 뷰의 경계에 도달했는지를 판정하는 명령어들, 상대적 커서 이동의 방향으로 가상 데스크톱의 뷰를 스크롤하는 명령어들(이들로 제한되지 않음)을 포함하는 컴퓨터 판독가능 명령어들을 포함하는 컴퓨터-판독가능 저장 매체를 기술하고 있다. 전술한 내용에 부가하여, 다른 측면들이 본 발명의 일부를 구성하는 특허청구범위, 첨부 도면 및 본문에 기술되어 있다.
- [0007] 본 발명의 예시적인 실시예는 가상 데스크톱의 뷰를 렌더링하는 회로 - 이 가상 데스크톱은 원격 컴퓨터의 디스

플레이 해상도에 대응하는 제1 픽셀 범위에 의해 정의되고, 뷰는 제2 픽셀 범위에 의해 정의됨 -, 커서 이동 정보를 수신하는 회로, 커서 이동 정보와 제2 픽셀 범위로부터, 커서가 제2 픽셀 범위의 경계에 도달했는지를 판정하는 회로, 및 커서 이동의 방향으로 뷰를 스크롤하는 회로(이들로 제한되지 않음)를 포함하는 컴퓨터 시스템을 기술하고 있다. 전술한 내용에 부가하여, 다른 측면들이 본 발명의 일부를 구성하는 특허청구범위, 첨부 도면 및 본문에 기술되어 있다.

[0008] 당업자라면 본 발명의 하나 이상의 다양한 측면들이 본 명세서에서 언급되는 본 발명의 측면들을 실시하는 회로 및/또는 프로그래밍을 포함할 수 있지만, 이들로 제한되지 않고, 이 회로 및/또는 프로그래밍이 시스템 설계자의 설계 선택들에 따라 본 명세서에 언급된 측면들을 실시하도록 구성된 하드웨어, 소프트웨어 및/또는 펌웨어의 거의 모든 조합일 수 있다는 것을 잘 알 수 있다.

[0009] 이상은 요약이며, 따라서 필요에 의해 상세가 간략화, 일반화 및 생략되어 있다. 당업자라면 이 요약이 단지 예시적인 것이고 결코 제한하기 위한 것이 아니라는 것을 잘 알 것이다.

도면의 간단한 설명

[0010] 도 1은 본 발명의 측면들이 구현될 수 있는 예시적인 컴퓨터 시스템을 나타낸 도면.

도 2는 본 발명의 측면들이 구현될 수 있는 예시적인 터미널 서버를 나타낸 도면.

도 3은 본 발명의 측면들이 구현될 수 있는 예시적인 클라이언트를 나타낸 도면.

도 4는 본 발명의 측면들을 실시하는 예시적인 플로우차트.

도 5는 도 4의 동작 절차(400)의 대안의 실시예를 나타낸 도면.

도 6은 도 5의 동작 절차(400)의 대안의 실시예를 나타낸 도면.

도 7은 도 6의 동작 절차(400)의 대안의 실시예를 나타낸 도면.

도 8은 본 발명의 측면들을 실시하는 예시적인 플로우차트.

도 9는 도 8의 동작 절차(800)의 대안의 실시예를 나타낸 도면.

도 10은 도 9의 동작 절차(800)의 대안의 실시예를 나타낸 도면.

도 11은 도 10의 동작 절차(800)의 대안의 실시예를 나타낸 도면.

도 12는 본 발명의 측면들을 실시하는 예시적인 플로우차트.

도 13은 도 12의 동작 절차(1200)의 대안의 실시예를 나타낸 도면.

도 14는 도 13의 동작 절차(1200)의 대안의 실시예를 나타낸 도면.

도 15는 도 12의 동작 절차(1200)의 대안의 실시예를 나타낸 도면.

도 16은 도 15의 동작 절차(1200)의 대안의 실시예를 나타낸 도면.

발명을 실시하기 위한 구체적인 내용

[0011] 본 발명의 다수의 실시예들이 컴퓨터 상에서 실행될 수 있다. 도 1 및 이하의 설명은 본 발명이 구현될 수 있는 적당한 컴퓨팅 환경에 대한 간략하고 전반적인 설명을 제공하기 위한 것이다. 당업자라면 도 1의 컴퓨터 시스템이, 어떤 실시예들에서, 도 2의 서버(204) 및 도 3의 클라이언트(201)를 실시할 수 있다는 것을 잘 알 것이다. 이러한 예시적인 실시예들에서, 서버(204)와 클라이언트(201)는 도 1에 설명된 구성요소들의 일부 또는 전부를 포함할 수 있으며, 어떤 실시예들에서는, 서버(204)와 클라이언트(201)가 본 발명의 특정 측면들을 구현하도록 구성된 회로를 포함할 수 있다. 일례로, 본 명세서 전반에 걸쳐 사용되는 용어 회로는 펌웨어 또는 스위치에 의해 기능(들)을 수행하도록 구성된 특수 하드웨어 구성요소를 포함할 수 있다. 다른 예시적인 실시예들에서, 용어 회로는 기능(들)을 수행하는 동작을 하는 논리를 구현하는 소프트웨어 명령어들에 의해 구성되는, 범용 처리 장치, 메모리, 기타 등등을 포함할 수 있다. 회로가 하드웨어와 소프트웨어의 조합을 포함하는 예시적인 실시예들에서, 구현자는 논리를 구현하는 소스 코드를 작성할 수 있으며, 소스 코드는 범용 처리 장치에 의해 처리될 수 있는 기계-판독가능 코드로 컴파일될 수 있다. 당업자라면 하드웨어, 소프트웨어, 또는 하드웨어/소프트웨어 조합 간에 거의 차이가 없는 수준까지 최신 기술이 발전해왔다는 사실을 잘 알 수 있기 때문에, 특정 기능

들을 실시하기 위해 소프트웨어를 선택할지 하드웨어를 선택할지는 구현자에게 맡겨진 설계 선택이다. 보다 구체적으로는, 당업자라면 소프트웨어 프로세스가 동등한 하드웨어 구조로 변환될 수 있고, 하드웨어 구조 자체가 동등한 소프트웨어 프로세스로 변환될 수 있다는 것을 잘 알 것이다. 따라서, 소프트웨어 구현을 선택할지 하드웨어 구현을 선택할지는 설계 선택의 하나로서 구현자에게 맡겨져 있다.

[0012] 이제 도 1을 참조하면, 예시적인 범용 컴퓨팅 시스템이 나타나어져 있다. 범용 컴퓨팅 시스템은 처리 장치(21), 시스템 메모리(22), 및 시스템 메모리를 비롯한 다양한 시스템 구성요소들을 처리 장치(21)에 결합시키는 시스템 버스(23)를 포함하는 종래의 개인용 컴퓨터(20), 기타 등등을 포함할 수 있다. 시스템 버스(23)는 메모리 버스 또는 메모리 제어기, 주변 장치 버스, 및 각종의 버스 아키텍처 중 어느 것이라도 사용하는 로컬 버스를 비롯한 몇가지 유형의 버스 구조 중 어느 것이라도 될 수 있다. 시스템 메모리는 판독 전용 메모리(ROM)(24) 및 랜덤 액세스 메모리(RAM)(25)를 포함한다. 시동 중과 같은 때에 개인용 컴퓨터(20) 내의 구성요소들 간의 정보 전송을 돕는 기본 루틴들이 들어 있는 기본 입/출력 시스템(26)(BIOS)은 ROM(24)에 저장되어 있다. 개인용 컴퓨터(20)은 하드 디스크(도시 생략)로부터 판독을 하고 그에 기록을 하는 하드 디스크 드라이브(27), 이동식 자기 디스크(29)로부터 판독을 하거나 그에 기록을 하는 자기 디스크 드라이브(28), 그리고 CD-ROM 또는 기타 광 매체 등의 이동식 광 디스크(31)로부터 판독을 하거나 그에 기록을 하는 광 디스크 드라이브(30)를 더 포함할 수 있다. 어떤 예시적인 실시예들에서, 본 발명의 측면들을 구현하는 컴퓨터 실행가능 명령어들은 ROM(24), 하드 디스크(도시 생략), RAM(25), 이동식 자기 디스크(29), 광 디스크(31), 및/또는 처리 장치(21)의 캐시에 저장될 수 있다. 하드 디스크 드라이브(27), 자기 디스크 드라이브(28) 및 광 디스크 드라이브(30)는 각각 하드 디스크 드라이브 인터페이스(32), 자기 디스크 드라이브 인터페이스(33), 및 광 드라이브 인터페이스(34)에 의해 시스템 버스(23)에 연결되어 있다. 이 드라이브들 및 그와 연관된 컴퓨터 판독가능 매체는 개인용 컴퓨터(20)에 대한 컴퓨터 판독가능 명령어, 데이터 구조, 프로그램 모듈 및 기타 데이터의 비휘발성 저장을 제공한다. 본 명세서에 기술된 예시적인 환경이 하드 디스크, 이동식 자기 디스크(29), 및 이동식 광학 디스크(31)를 이용하고 있지만, 당업자라면 자기 카세트, 플래시 메모리 카드, DVD(digital video disk), 베르누이 카트리지, RAM(random access memories), ROM(read only memories), 기타 등등과 같은 컴퓨터에 의해 액세스 가능한 데이터를 저장할 수 있는 다른 유형의 컴퓨터-판독가능 매체들도 예시적인 운영 환경에서 사용될 수 있다는 것을 잘 알 것이다.

[0013] 운영 체제(35), 하나 이상의 응용 프로그램들(36), 기타 프로그램 모듈들(37) 및 프로그램 데이터(38)를 비롯한 다수의 프로그램 모듈들이 하드 디스크, 자기 디스크(29), 광 디스크(31), ROM(24) 또는 RAM(25)에 저장될 수 있다. 사용자는 키보드(40) 및 포인팅 장치(42) 등의 입력 장치를 통해 개인용 컴퓨터(20)에 명령 및 정보를 입력할 수 있다. 다른 입력 장치(도시 생략)로는 마이크, 조이스틱, 게임 패드, 위성 안테나, 스캐너, 기타 등등이 있을 수 있다. 이들 및 기타 입력 장치는 종종 시스템 버스에 연결된 직렬 포트 인터페이스(46)를 통해 처리 장치(21)에 연결되지만, 병렬 포트, 게임 포트 또는 USB(universal serial bus) 등의 다른 인터페이스에 의해 연결될 수도 있다. 디스플레이(47) 또는 기타 유형의 디스플레이 장치도 비디오 어댑터(48) 등의 인터페이스를 통해 시스템 버스(23)에 연결될 수 있다. 디스플레이(47)에 부가하여, 개인용 컴퓨터는 통상적으로 스피커 및 프린터 등의 다른 주변 출력 장치(도시 생략)를 포함하고 있다. 도 1의 예시적인 시스템은 또한 호스트 어댑터(55), SCSI(Small Computer System Interface) 버스(56), 및 SCSI 버스(56)에 연결된 외부 저장 장치(62)도 포함하고 있다.

[0014] 개인용 컴퓨터(20)는 원격 컴퓨터(49) 등의 하나 이상의 원격 컴퓨터와의 논리적 연결을 사용하여 네트워크화된 환경에서 동작할 수 있다. 원격 컴퓨터(49)는 다른 개인용 컴퓨터, 서버, 라우터, 네트워크 PC, 피어 장치 또는 기타 통상의 네트워크 노드일 수 있고, 통상적으로 개인용 컴퓨터(20)와 관련하여 상기한 구성요소들 중 다수 또는 그 전부를 포함할 수 있지만, 도 1에는 메모리 저장 장치(50)만이 도시되어 있다. 도 1에 도시된 논리적 연결은 근거리 통신망(LAN)(51) 및 원거리 통신망(WAN)(52)을 포함할 수 있다. 이러한 네트워킹 환경은 사무실, 전사적 컴퓨터 네트워크, 인트라넷 및 인터넷에서 흔한 것이다.

[0015] LAN 네트워킹 환경에서 사용될 때, 개인용 컴퓨터(20)는 네트워크 인터페이스 또는 어댑터(53)를 통해 LAN(51)에 연결될 수 있다. WAN 네트워킹 환경에서 사용될 때, 개인용 컴퓨터(20)는 통상적으로 인터넷 등의 WAN(52)을 통해 통신을 설정하는 모뎀(54) 또는 기타 수단을 포함할 수 있다. 내장형이거나 외장형일 수 있는 모뎀(54)은 직렬 포트 인터페이스(46)를 통해 시스템 버스(23)에 연결될 수 있다. 네트워킹된 환경에서, 개인용 컴퓨터(20) 또는 그의 일부와 관련하여 도시된 프로그램 모듈들은 원격 메모리 저장 장치에 저장될 수 있다. 도시된 네트워크 연결이 예시적인 것이고 컴퓨터들 간에 통신 링크를 설정하는 기타 수단이 사용될 수 있다는 것을 잘 알 것이다. 또한, 본 발명의 수많은 실시예들이 컴퓨터화된 시스템에 특히 적합할 것이라고 기대되는 반면에,

본 문서 내의 어떤 내용도 본 발명을 이러한 실시예들로 제한하기 위한 것이 아니다.

[0016] 이제 도 2 및 도 3을 참조하면, 이들 도면은 본 발명의 측면들이 구현될 수 있는 예시적인 환경을 개괄적으로 나타낸 것이다. 당업자라면 도 2와 도 3에 기술된 예시적인 요소들이 본 발명을 설명하기 위한 동작 프레임워크 (operational framework)를 제공하기 위해 도시되어 있다는 것을 잘 알 것이다. 따라서, 일부 실시예들에서, 각각의 환경의 물리적 레이아웃이 서로 다른 구현 방식들에 따라 다를 수 있다. 따라서, 예시적인 동작 프레임워크는 단지 예시로서 취급되어야 하고, 결코 특허청구범위를 제한하지 않는다. 당업자라면 이하의 설명이 소개하는 것이고 도 2와 도 3에 나타낸 요소들이 도 4 내지 도 16에 기술된 동작 절차들의 설명 내에 더 자세하게 기술되어 있다는 것을 잘 알 것이다.

[0017] 일반적으로, 도 2는 본 발명의 측면들을 포함하도록 구성될 수 있는 서버 환경에 대한 상위 레벨 개요를 나타낸 것이다. 동 도면을 참조하면, 터미널 서버 (terminal server)를 실시하도록 구성된 회로를 포함할 수 있는 서버 (204)가 나타내어져 있거나, 다른 실시예들에서, 서버(204)는 원격 데스크톱 연결을 지원하도록 구성된 회로를 포함할 수 있다. 도시된 일례에서, 서버(204)는 클라이언트들을 연결하기 위해, 세션 1 내지 세션 N(N은 2보다 큰 정수임)과 같은 하나 이상의 세션들을 생성하도록 구성되어 있을 수 있다. 간단히 설명하면, 본 발명의 예시적인 실시예들에서, 세션은 일반적으로 서버(204)의 커널(214)과 상호작용하도록 구성된 복수의 서브 시스템들(예를 들어, 소프트웨어 코드)에 의해 실시되는 동작 환경을 포함할 수 있다. 예를 들어, 세션은 데스크톱 창과 같은 사용자 인터페이스를 인스턴스화하는 프로세스, 창 내에서 마우스 이동을 추적하는 서브시스템들, 아이콘 위에서의 마우스 클릭을 프로그램의 인스턴스를 실행하는 명령으로 변환하는 서브시스템들, 기타 등등을 포함할 수 있다. 예를 들어, 서버(204)가 네트워크 연결을 통해 클라이언트(201)로부터 연결 요청을 수신하면, 서버(204)에 의해 사용자마다 세션이 생성될 수 있다. 일반적으로, 연결 요청은 먼저, 예를 들면, 서버(204)의 회로에 의해 실행될 수 있는 전송 논리(210)에 의해 처리될 수 있다. 전송 논리(210)는, 몇몇 실시예들에서, 연결 메시지들을 수신하고 이들을 엔진(212)으로 전달하도록 구성될 수 있는 네트워크 어댑터, 펌웨어 및 소프트웨어를 포함할 수 있다. 도 2에 나타낸 바와 같이, 몇몇 실시예들에서, 전송 논리(210)는 각각의 세션에 대한 프로토콜 스택 인스턴스들(protocol stack instances)을 포함할 수 있다. 일반적으로, 각각의 프로토콜 스택 인스턴스는 사용자 인터페이스 출력을 클라이언트로 라우팅하고, 클라이언트로부터 수신된 사용자 입력을 그의 세션과 연관된 세션 코어(session core)(244)로 라우팅하도록 구성될 수 있다.

[0018] 도 2의 개괄적인 설명을 계속하면, 본 발명의 몇몇 예시적인 실시예들에서, 엔진(212)은 세션에 대한 요청을 처리하고, 각각의 세션의 기능을 확인하며, 세션에 대한 일련의 물리적 리소스들을 할당하고 세션에 대한 프로토콜 스택 인스턴스를 인스턴스화함으로써 세션을 생성하도록 구성될 수 있다. 몇몇 실시예들에서, 엔진(212)은 위에서 언급한 동작 절차들의 일부를 구현할 수 있는 특수 회로 구성요소들에 의해 실시될 수 있다. 예를 들어, 몇몇 예시적인 실시예들에서, 회로는 메모리와, 엔진(212)을 실시하는 코드를 실행하도록 구성된 프로세서를 포함할 수 있다. 도 2에 나타낸 바와 같이, 어떤 경우들에, 엔진(212)은 연결 요청을 수신할 수 있고, 예를 들면, 라이선스가 이용가능하고 요청에 대해 세션이 생성될 수 있는지를 판정할 수 있다. 서버(204)가 원격 데스크톱 기능을 포함하는 원격 컴퓨터인 상황에서, 엔진(212)은 라이선스를 확인함이 없이 연결 요청에 응답하여 세션을 생성하도록 구성될 수 있다. 도 2에 나타낸 바와 같이, 세션 관리자(216)는 엔진(212)으로부터 메시지를 수신하도록 구성될 수 있으며, 그 메시지에 응답하여, 세션 관리자(216)는 또한 세션 식별자를 테이블에 추가하고, 메모리를 세션 식별자에 할당하며, 세션 식별자에 할당된 메모리에 서브시스템 프로세스들의 시스템 환경 변수 및 인스턴스들을 생성할 수 있다.

[0019] 도 2에 나타낸 바와 같이, 세션 관리자(216)는 세션 코어(244)와 같은 커널 모드 부분(kernel mode part)을 포함할 수 있는 런타임 서브시스템(240)과 같은 환경 서브시스템을 인스턴스화할 수 있다. 일례로, 실시예에서의 환경 서브시스템은 서비스들 중 일부를 응용 프로그램에 노출시키고, 운영 체제(214)의 커널에 대한 액세스 포인트를 제공하도록 구성되어 있다. 예시적인 실시예들에서, 런타임 서브시스템(240)은 프로세스와 스레드의 실행을 제어할 수 있고, 세션 코어(244)는 스레드에 메모리를 할당하고, 스레드가 실행될 시간을 스케줄링하라는 요청을 커널(214)의 실행부(executive)에 보낼 수 있다. 일 실시예에서, 세션 코어(244)는 그래픽 디스플레이 인터페이스(GDI)(246), 보안 서브시스템(250) 및 입력 서브시스템(252)을 포함할 수 있다. 이러한 실시예들에서, 입력 서브시스템(252)은 세션과 연관된 프로토콜 스택 인스턴스를 통해서 클라이언트(201)로부터 사용자 입력을 수신하고 이 입력을 적절한 세션의 세션 코어(244)로 전송하도록 구성되어 있을 수 있다. 몇몇 실시예들에서, 사용자 입력은 절대적 및/또는 상대적 마우스 이동 명령, 마우스 좌표, 마우스 클릭, 키보드 신호, 조이스틱 이동 신호 등을 나타내는 신호를 포함할 수 있다. 사용자 입력(예를 들어, 아이콘에서 마우스를 두 번 클릭)이 세션 코어(244)에 의해 수신될 수 있으며, 입력 서브시스템(252)은 두 번 클릭과 연관된 좌표에 아이콘이

위치하는지를 판정하도록 구성될 수 있다. 이어서, 입력 서브시스템(252)은 아이콘과 연관된 응용 프로그램의 프로세스를 실행할 수 있는 런타임 서브시스템(240)으로 통지를 보내도록 구성되어 있을 수 있다.

[0020] 클라이언트(201)로부터 입력을 수신하는 것에 부가하여, 드로잉 명령(draw command)이 응용 프로그램 및/또는 데스크톱으로부터 수신되고 GDI(246)에 의해 처리될 수 있다. 일반적으로, GDI(246)는 그래픽 개체 드로잉 명령들을 생성할 수 있는 프로세스를 포함할 수 있다. 이 예시적인 실시예에서, GDI(246)는 원격 디스플레이 서브시스템(254)으로 자신의 출력을 전달하도록 구성될 수 있고, 원격 디스플레이 서브시스템(254)에서 그 명령들이 세션에 연결된 디스플레이 드라이버에 맞게 포맷팅된다. 어떤 예시적인 실시예들에서, 하나 이상의 물리적 디스플레이가, 예를 들어, 원격 데스크톱 상황에서, 서버(204)에 연결되어 있을 수 있다. 이러한 예시적인 실시예들에서, 원격 디스플레이 서브시스템(254)은 원격 컴퓨터 시스템의 디스플레이 드라이버(들)에 의해 렌더링되는 드로잉 명령을 미러링하고, 미러링된 정보를 세션과 연관된 스택 인스턴스(stack instance)를 통하여 클라이언트(201)로 전송하도록 구성될 수 있다. 다른 예시적인 실시예에서, 서버(204)가 터미널 서버인 경우, 원격 디스플레이 서브시스템(254)은 서버(204)에 물리적으로 연결된 디스플레이와 연관되어 있지 않을 수 있는 가상 디스플레이 드라이버(들)을 포함하도록 구성될 수 있다, 예를 들어, 서버(204)가 헤드리스(headless)로 실행될 수 있다. 이 실시예에서, 원격 디스플레이 서브시스템(254)은 하나 이상의 가상 디스플레이들에 대한 드로잉 명령들을 수신하고, 드로잉 명령들을 세션과 연관된 스택 인스턴스를 통하여 클라이언트(201)로 전송하도록 구성될 수 있다. 본 발명의 실시예에서, 원격 디스플레이 서브시스템(254)은 각각의 디스플레이 드라이버마다 디스플레이 해상도를 확인하고, 예를 들어, 가상 디스플레이들과 연관된 가상 디스플레이 드라이버(들)의 디스플레이 해상도 또는 물리적 디스플레이들과 연관된 디스플레이 드라이버들의 디스플레이 해상도를 확인하고, 연관된 프로토콜 스택 인스턴스를 통하여 패킷들을 클라이언트(201)로 라우팅하도록 구성될 수 있다.

[0021] 몇몇 예시적인 실시예들에서, 세션 관리자(216)는 그에 부가하여 세션 로그온과 로그오프를 처리하도록 구성될 수 있는 세션의 세션 식별자와 연관된 로그온 프로세스의 인스턴스를 인스턴스화할 수 있다. 이러한 예시적인 실시예들에서, 로그온 프로세스와 연관된 그래픽 사용자 인터페이스를 나타내는 드로잉 명령들은 클라이언트(201)로 전송될 수 있고, 이 때 클라이언트(201)의 사용자는, 예를 들면, 사용자 이름/암호의 조합, 스마트카드 식별자, 및/또는 생체인식 정보와 같은 계정 식별자를 로그온 화면에 입력할 수 있다. 이 정보는 서버(204)로 전송될 수 있으며 엔진(212)과 세션 코어(244)의 보안 서브시스템(250)으로 라우팅될 수 있다. 일례로, 어떤 예시적인 실시예들에서, 엔진(212)은 사용자 계정이 라이선스와 연관되어 있는지 여부를 판정하도록 구성될 수 있으며, 보안 서브시스템(250)은 세션에 대한 보안 토큰을 생성하도록 구성될 수 있다.

[0022] 이제 도 3을 참조하면, 도 3은 본 발명의 측면들이 구현될 수 있는 예시적인 클라이언트(201)를 나타낸 것이다. 일반적으로, 도 3의 예시적인 클라이언트(201)는 인터넷을 통하여 서버(204)와 연결되어 동작할 수 있다. 일 실시예에서, 클라이언트(201)는 사용자 입력을 서버(204)로 보내고 서버(204)가 생성한 사용자 인터페이스 정보를 디스플레이하도록 구성된 하드웨어에 의해 실시되는 컴퓨터 터미널(computer terminal)일 수 있다. 다른 실시예에서, 클라이언트(201)는 도 1의 개인용 컴퓨터(20)의 요소들과 유사한 요소들을 포함하는 컴퓨터일 수 있다. 이 예시적인 실시예에서, 클라이언트(201)는 운영 체제(312)를 실행하도록 구성된 회로와 터미널의 기능을 에뮬레이트하도록 구성된 회로를 포함할 수 있다. 당업자라면 운영 체제(312)를 실행하도록 구성된 회로도 역시 터미널을 에뮬레이트하도록 구성될 수 있다는 것을 잘 알 수 있다. 도 3의 개괄적인 설명을 계속하면, 클라이언트(201)는, 몇몇 실시예들에서, 실행 서비스(executive service)의 일부를 노출시킬 수 있는 클라이언트 서브시스템(306), 및 API 호출을 커널(314)의 커널 모드 호출로 변환하는 DLL을 포함할 수 있다. 한 실시예에서, 클라이언트 서브시스템(306)은 클라이언트(201)와 연결될 수 있는 디스플레이에 대한 디스플레이 드라이버를 포함할 수 있다. 클라이언트 서브시스템(306)은 서버(204)로부터 드로잉 명령들을 수신하고 또 디스플레이 드라이버에게 가상 데스크톱(302)에 이미지를 드로잉하라고 지시하도록 구성될 수 있다. 가상 데스크톱(302)이 메모리에 드로잉되면, 디스플레이 드라이버는 물리적 디스플레이 상에 가상 데스크톱(302)의 뷰를 렌더링하도록 구성될 수 있다, 예를 들면, 클라이언트 서브시스템(306)은 디스플레이 드라이버에게 가상 데스크톱(302)의 뷰(304)를 가져다가 이를 디스플레이 상에 렌더링하도록 지시할 수 있다. 도 3의 개괄적 설명을 계속하면, 몇몇 실시예들에서, 클라이언트 서브시스템(306)의 일부분 또는 클라이언트 서브시스템(306)과 상호작용하는 별도의 서브시스템일 수 있는 스크롤 서브시스템(scrolling subsystem)(308)이 나타내어져 있다. 스크롤링 서브시스템(308)은 일반적으로 여러 변수들을 추적함으로써 언제 가상 데스크톱(302)의 뷰(304)를 스크롤해야 하는지를 판정하도록 구성될 수 있다.

[0023] 이하는 프로세스들의 구현을 나타낸 일련의 플로우차트이다. 이해의 편의를 위해, 흐름도들은 처음의 플로우차트들이 전반적인 “큰 그림”의 관점에서 구현을 제시하는 식으로 구성되어 있다. 당업자라면 본 명세서에 이용

되는 제시 방법(예를 들면, 전반적인 뷰를 제시하는 플로우차트(들)의 제시로 시작하고 나서 추가적 및/또는 보다 자세한 내용을 후속하는 플로우차트들에서 제공함)이 일반적으로 다양한 동작 절차의 빠르고 쉬운 이해를 가능하게 해준다는 것을 잘 알 것이다.

[0024] 이제 도 4를 참조하면, 동작(400) 내지 동작(412)을 포함하는 본 발명의 측면들을 실시하는 예시적인 플로우차트가 나타내어져 있다. 예시적인 동작(400)은 동작 절차를 시작하고, 동작(402)은, 원격 컴퓨터의 디스플레이 드라이버의 디스플레이 해상도를 포함하는 원격 컴퓨터로부터 수신된 기하 형태 정보로부터, 가상 데스크톱을 생성하는 것을 나타낸다. 일례로, 도 3과 함께 도 2를 참조하면, 클라이언트(201)는 신호, 예를 들어, 서버(204)로부터의 기하 형태 정보를 나타내는 하나 이상의 정보 패킷을 수신하도록 구성될 수 있으며, 도 3의 클라이언트 서브시스템(306)은 메모리 내에 가상 데스크톱(302)을 생성하기 위해 그 정보를 처리하도록 구성될 수 있다. 예시적인 실시예에서, 상기 기하 형태 정보는 서버(204) 디스플레이 드라이버의 디스플레이 해상도를 정의할 수 있으며, 클라이언트 서브시스템(306)은 가상 데스크톱(302), 예를 들면, 서버(204)와 동일한 디스플레이 해상도를 갖는 비트맵을 생성할 수 있다.

[0025] 본 발명의 실시예에서, 서버(204)의 원격 디스플레이 서브시스템(254)은, 예를 들어, 서버(204)의 디스플레이 드라이버(들)와 연관된 디스플레이 픽셀의 양과 동일한 양의 픽셀을 포함하는 경계 상자(bounding box)를 생성하는 것으로 기하 형태 정보를 생성하도록 구성될 수 있다. 특정의 일례에서, 4개의 디스플레이 드라이버가 서버(204)에 의해 인스턴스화될 수 있고, 각각의 드라이버는 1280 x 1024 픽셀의 해상도를 가질 수 있다. 이 일례에서, 원격 디스플레이 서브시스템(254)은 5120 x 1024 픽셀의 기하 형태를 갖는 경계 상자를 생성할 수 있다. 원격 디스플레이 서브시스템(254)은 그에 부가하여 아이콘의 위치, 기타를 각각의 디스플레이 드라이버와 관련된 좌표로부터 경계 상자의 좌표로 변환하고 기하 형태를 전송 논리(210)를 통하여 클라이언트(201)로 보내도록 구성될 수 있다. 이 일례에서, 클라이언트(201)는 기하 형태로부터 경계 상자를 생성하도록 구성될 수 있다. 특정의 일례를 계속 살펴보면, 클라이언트 서브시스템(306)은 신호를 수신하고 디스플레이 드라이버에게 메모리 내에 5120 x 1024 픽셀의 크기로 가상 데스크톱(302)을 생성하라고 지시하도록 구성될 수 있다. 이 일례에서, 클라이언트 서브시스템(306)은 그 다음에 경계 상자 내에서의 그래픽 위치에 대해 비트맵으로 그래픽을 드로잉하도록 디스플레이 드라이버를 구성할 수 있다. 다른 특정의 일례에서, 원격 디스플레이 서브시스템(254)은 서버(204)에 연결된 디스플레이 드라이버가 2048 x 1536 픽셀의 디스플레이 해상도를 출력하도록 구성되었는지를 판정하도록 구성될 수 있다. 이 일례에서, 원격 디스플레이 서브시스템(254)은 디스플레이 해상도를 나타내는 기하 형태 정보를 전송 논리(210)를 통해서 클라이언트(201)로 전송하도록 구성될 수 있다. 이 특정의 일례에서, 클라이언트 서브시스템(306)은 신호를 수신하고 메모리 내에 2048 x 1536 픽셀의 크기를 갖는 가상 데스크톱(302)을 생성하도록 구성될 수 있으며, 클라이언트 서브시스템(306)은 경계 상자 내에서의 그래픽 위치에 대해 비트맵으로 그래픽을 드로잉하도록 디스플레이 드라이버를 구성할 수 있다.

[0026] 도 4의 설명을 계속하면, 동작(404)은 원격 컴퓨터로부터 수신된 그래픽을 가상 데스크톱에 드로잉하는 것을 나타낸다. 예를 들어, 이전의 일례를 계속 살펴보면, 클라이언트 서브시스템(306)은 신호, 예를 들어, 원격 디스플레이 서브시스템(254)으로부터의 드로잉 명령들을 나타내는 하나 이상의 패킷을 수신하도록 구성될 수 있다. 그 명령들은 클라이언트 서브시스템(306)에 의해 처리될 수 있으며, 클라이언트(201)의 디스플레이 드라이버는 가상 데스크톱(302) 내에 이미지를 드로잉하도록 구성될 수 있다. 특정의 일례에서, 복수의 아이콘들을 각각의 아이콘의 서버(204)에 대한 위치 정보와 함께 사용하여, 데스크톱 환경을 정의하는 신호가 서버(204)로부터 수신될 수 있다. 클라이언트 서브시스템(306)은 이 정보를 처리하고 디스플레이 드라이버에게 경계 상자에 대한 데스크톱 및 아이콘을 드로잉하도록 지시할 수 있다.

[0027] 도 4의 설명을 계속하면, 동작(406)은 가상 데스크톱의 뷰를 렌더링하는 것을 나타내며, 이 때 뷰는 경계들에 의해 정의되고 가상 데스크톱의 일부분을 나타낸다. 예시적인 동작 절차의 설명을 계속하면, 그래픽이 서버(204)로부터 수신되고 가상 데스크톱(302)에 드로잉된 후에, 클라이언트 서브시스템(306)의 디스플레이 드라이버는 가상 데스크톱(302)의 뷰(304)를 렌더링하도록 구성될 수 있다. 예를 들면, 본 발명의 실시예에서, 가상 데스크톱(302) 전체가 메모리에 드로잉될 수 있는 반면에, 디스플레이 드라이버는, 그래픽의 일부가 가상 데스크톱(302)의 현재 뷰 내에 위치하는지 여부에 기초하여, 클라이언트(201)에 연결된 디스플레이 상에 그래픽의 일부만을 렌더링할 수 있다. 본 발명의 실시예에서, 클라이언트(201)의 디스플레이 드라이버는 가상 데스크톱(302)의 디스플레이 해상도보다 작은 특정의 디스플레이 해상도를 가질 수 있다, 예를 들어, 가상 데스크톱(302)의 크기가 2048 x 1536 픽셀일 수 있는 반면에, 클라이언트(201)의 디스플레이 드라이버는 단지 800 x 600 픽셀의 디스플레이 해상도를 렌더링할 수 있을 수 있다. 이 일례에서, 클라이언트 서브시스템(306)은 뷰(304)의 디스플레이 해상도를 정의하는 정보 및 가상 데스크톱(302)에 대한 뷰의 좌표를 수신하도록 구성될 수 있으며,

이미지를 뷰(304)의 크기 내에 나타나게 렌더링하도록 디스플레이 드라이버를 구성할 수 있다.

[0028] 도 4의 설명을 계속하면, 동작(408)은 상대적 커서 이동 정보를 수신하는 것을 나타낸다. 예를 들면, 이전의 일례에 부가하여, 스크롤 서브시스템(308)은 클라이언트(201)의 운영 체제(312)로부터 상대적 마우스 이동 정보를 수신하도록 구성될 수 있다. 예를 들어, 일 실시예에서, 운영 체제(312)는 뷰(304) 내에서의 커서(310), 예를 들어, 비트맵의 이동을 추적하도록 구성될 수 있다. 몇몇 실시예들에서, 커서(310)는 핫스팟(hotspot)이라고 불리는 단일-픽셀 점, 예를 들어, 운영 체제(312)가 추적하여 커서(310)의 위치로서 인식하는 점을 포함할 수 있다. 이 예시적인 실시예에서, 스크롤 서브시스템(308)은 물리적 입력 장치(예를 들어, 터치패드나 마우스)가 움직인 방향을 식별해주는, 예를 들어, 입력 메시지들을 미리 정해진 간격으로 수신하도록 구성될 수 있다. 예를 들어, 일 실시예에서, 클라이언트(201)의 운영 체제(312)는 입력 장치(예를 들어, 마우스나 터치패드)가 물리적으로 움직일 때마다 추적하도록 구성될 수 있다. 이 일례에서, 입력 메시지는 입력 장치가 오른쪽 위로 움직인 경우에 (+1,+1)과 같은 정보를 포함할 수 있다. 대안의 실시예에서, 스크롤 서브시스템(308)은 가상 데스크톱(302)에 대한 커서(310)의 좌표를 식별해주는 절대적 커서 이동 정보를 추적하도록 구성될 수 있다. 이 일례에서, 스크롤 서브시스템(308)은 위치 정보로부터 좌표를 계산하도록 구성될 수 있다.

[0029] 도 4의 설명을 계속하면, 동작(410)은, 상대적 커서 이동 정보와 뷰의 경계들로부터, 커서가 뷰의 경계에 도달했는지를 판정하는 것을 나타낸다. 예를 들어, 본 발명의 일 실시예에서, 스크롤 서브시스템(308)은 가상 데스크톱(302)에 대한 뷰(304)의 픽셀 범위 및 커서(310)의 현재 위치를 식별해주는 정보를 수신하도록 구성될 수 있다. 스크롤 서브시스템(308)은 그에 부가하여, 예를 들면, 절대적 마우스 이동 정보 및/또는 상대적 마우스 이동 정보를 포함할 수 있는 메시지를 수신하도록 구성될 수 있다. 스크롤 서브시스템(308)은 가상 데스크톱(302)에 대한 커서(310)의 위치를 확인하고 그 위치를 가상 데스크톱(302)에 대한 뷰(304)를 정의하는 픽셀 범위와 비교함으로써, 예를 들면, 마우스 이동 메시지가 수신될 때마다, 가상 데스크톱(302)에 대한 커서(310)의 x 및 y 좌표를 가상 데스크톱(302)에 대한 뷰(304)의 경계들을 정의하는 x 및 y 좌표와 비교함으로써, 커서(310)가 뷰(304)의 경계에 도달했는지 여부를 판정할 수 있다. 특정의 일례에서, 스크롤 서브시스템(308)은 가상 데스크톱에서의 커서(310)의 좌표(예를 들어, 픽셀 400, 픽셀 300) 및 가상 데스크톱의 뷰를 정의하는 픽셀들(예를 들어, x축에서 픽셀 0 내지 800과 y축에서 픽셀 0 내지 600)을 포함하는 테이블을 포함할 수 있다. 사용자가 마우스를 움직임에 따라, 커서(310)는, 예를 들어, 뷰(304)를 가로질러 좌에서 우로 이동할 수 있다. 스크롤 서브시스템(308)은 커서(310)의 픽셀 위치를 증분시킬 수 있으며, 각각의 증분에 대해, 픽셀이 경계(예를 들어, 이 일례에서, x축 픽셀 800)에 도달했는지를 검사한다. 스크롤 서브시스템(308)이 x축 경계에 도달했다고(예를 들어, 커서(310)가 픽셀 점 800, 300에 위치한다고) 판정하는 경우에, 스크롤 서브시스템(308)은 클라이언트 서브시스템(306)으로 메시지를 보내도록 구성될 수 있다.

[0030] 도 4의 설명을 계속하면, 동작(412)은 상대적 커서 이동의 방향으로 가상 데스크톱의 뷰를 스크롤하는 것을 나타낸다. 예를 들어, 이전의 일례에 부가하여, 스크롤 서브시스템(308)이 뷰(304)의 경계에 도달했다고 판정하는 경우에, 스크롤 서브시스템(308)은 상대적 마우스 이동의 방향으로 뷰(304)를 패닝하라는 신호를 클라이언트 서브시스템(306)으로 보내도록 구성될 수 있다. 예를 들어, 특정의 실시예에서, 스크롤 서브시스템(308)은 커서(310)가 뷰(304)의 오른쪽 경계에 위치했을 때 마우스가 x-방향으로 +1 만큼 움직였다는 것을 나타내는 신호를 수신할 수 있다. 스크롤 서브시스템(308)은 뷰를 x-축 상에서 +1 픽셀만큼 패닝하라는 메시지를 클라이언트 서브시스템(306)으로 보내도록 구성될 수 있다. 이 일례에서, 커서(310)가 x-방향으로 한번더 +1 픽셀만큼 움직일 수 있고, 스크롤 서브시스템(308)은 커서(310)가 새로운 경계 상에 있는지를 판정할 수 있다. 따라서, 마우스가 또 다시 우측으로 움직인다면, 스크롤 서브시스템(308)은 뷰(304)를 또 다시 패닝하라는 신호를 클라이언트 서브시스템(306)으로 보내도록 구성될 수 있다.

[0031] 이제 도 5 내지 도 7을 참조하면, 이들 도면은 도 4의 동작 절차(400)의 대안의 실시예들을 나타낸 것이다. 도 5는 부가의 동작 절차들(514 및 516)과 도 4의 동작 절차(400)에 대한 미세 조정(518)을 포함하는 도 4의 대안의 실시예를 나타낸 것이다. 동작(514)을 참조하면, 동작(514)은 상대적 커서 이동 정보에 선형 가속 인자(linear accelerating factor)를 적용하는 것을 나타낸다. 예를 들어, 본 발명의 실시예에서, 스크롤 서브시스템(308)은 상대적 커서 이동 정보를 수신하고, 상대적 커서 이동 정보를 스케일링 값과 곱하며, 스케일링된 값에 의해 정의된 양만큼 뷰(304)를 스크롤하라는 신호를 클라이언트 서브시스템(306)으로 전송하도록 구성될 수 있다. 예를 들어, 예시적인 실시예에서, 뷰(304)를 상대적 마우스 이동과 같은 양보다 많은 수의 픽셀만큼 패닝하는 것이 바람직할 수 있다, 예를 들어, 클라이언트(201)가 터치패드를 포함하는 예시적인 실시예들에서, 가상 데스크톱(302)의 한 쪽에서 다른 쪽으로 뷰(304)를 이동시키기 위해, 사용자는 터치패드를 가로질러 자신의 손가락을 터무니없이 여러 번 스크롤해야 할지도 모른다. 따라서, 일 실시예에서, 스크롤 서브시스템(308)은 상대

적 이동 정보를 고정된 값과 곱하고 또 클라이언트 서브시스템(306)에게 뷰(304)를 스케일링된 양만큼 이동하라고 지시하도록 구성될 수 있다. 특정의 일례에서, 스크롤 서브시스템(308)은 커서 위치가 뷰(304)의 오른쪽 경계를 정의하는 픽셀 위에 있는지를 판정하도록 구성될 수 있다. 이 일례에서, 사용자가 마우스를 x-방향으로 +1만큼 움직이면, 스크롤 서브시스템(308)은 50과 같은 고정된 숫자를 픽셀 이동에 곱하고, 뷰(304)를 x-방향으로 +50 픽셀만큼 움직이려는 메시지를 클라이언트 서브시스템(306)으로 보낼 수 있다.

[0032] 도 5의 설명을 계속하면, 도 5는, 그에 부가하여, 커서의 속도를 구하는 것, 동적 가속 인자를 생성하는 것 - 동적 가속 인자는 커서의 속도의 함수임 -, 및 상대적 커서 이동 정보에 동적 가속 인자를 적용하는 것을 나타내는 동작(516)을 나타내고 있다. 예를 들어, 본 발명의 실시예에서, 스크롤 서브시스템(308)은 상대적 커서 이동 정보를 수신하고, 이를 동적 스케일링 값과 곱하며, 스케일링된 값에 의해 정의된 양만큼 뷰(304)를 스크롤하라는 신호를 클라이언트 서브시스템(306)으로 전송하도록 구성될 수 있다. 예를 들어, 일 실시예에서, 커서(310)가 뷰(304)를 가로질러 빠르게 움직이는 경우에 뷰(304)를 많은 수의 픽셀만큼 패닝하는 것이 바람직할 수 있다. 일례로, 사용자가 뷰(304)를 가로질러 커서(310)를 움직인다면, 사용자가 가상 데스크톱(302)의 먼쪽 끝에 있는 무엇인가를 조작하는 데 관심이 있는 것으로 추정될 수 있다. 이 일례에서, 스크롤 서브시스템(308)은 사용자의 의도가 뷰(304)를 많은 양의 픽셀만큼 패닝하는 것이라고 추정하도록 구성될 수 있다. 이 상황에서, 스크롤 서브시스템(308)은 커서(310)가 뷰(304)를 가로질러 움직일 때의 커서(310)의 속도에 의존하는 스케일링 값을 생성하도록 구성될 수 있다. 일례로, 스크롤 서브시스템(308)은 마지막 64개의 샘플에 걸쳐 커서(310)의 좌표를 추적하도록 구성될 수 있다. 샘플링 레이트(sampling rate)는 일정할 수 있으며, 스크롤 서브시스템(308)은 커서(310)의 평균 속도를 구하도록 구성될 수 있다. 이 일례에서, 스크롤 서브시스템(308)은 커서(310)의 평균 속도에 기초하여 스케일링 값을 증가시키고, 클라이언트 서브시스템(306)에게 뷰(304)를 스케일링된 양만큼 패닝하라고 지시하도록 구성될 수 있다. 특정의 일례에서, 스케일링 값이 속도에 따라 기하급수적으로 증가할 수 있다. 따라서, 커서(310)가 뷰(304)에서 빠르게 움직일수록, 클라이언트 서브시스템(306)은 뷰(304)를 더 많이 스크롤할 것이다. 특정의 실시예에서, 커서(310)는 뷰(304)를 가로질러 왼쪽에서 오른쪽으로 이동할 수 있다. 이러한 특정의 일례에서, 스크롤 서브시스템(308)은 커서(310)의 좌표가 x-방향으로 샘플당 +10만큼씩 변하고 있음을 나타내는 다수의 메시지를 수신할 수 있다. 이러한 특정의 일례에서, 스크롤 서브시스템(308)은 e^{10} 을 적용하여 동적 가속 인자를 구하고, 클라이언트 서브시스템(306)에게 뷰(304)를 x-방향으로 22,026 픽셀만큼 이동시키라고 지시하도록 구성될 수 있다. 동일하거나 다른 실시예들에서, 스크롤 서브시스템(308)은 403과 같은 최대 동적 값을 설정하도록 구성될 수 있으므로, 이 상황에서는 클라이언트 서브시스템(306)은 수천 픽셀을 스크롤하여 뷰(304)를 엄청나게 변화시키지 않을 수도 있다.

[0033] 대안의 실시예에서, 스크롤 서브시스템(308)은 커서(310)의 속도를 구하고, 스크롤 서브시스템(308)이 뷰(304)의 경계들로서 간주하는 픽셀 범위를 변경하도록 구성될 수 있다. 이 일례에서, 스크롤 서브시스템(308)은 커서(310)의 속도를 검출하고, 커서 속도가 증가함에 따라 작아지는 뷰(304) 내에서 경계들로서 기능하는 경계 상자를 생성할 수 있다. 일례로, 800 x 600 픽셀을 포함하는 뷰(304)에서, 뷰(304)의 경계들이 x-방향으로는 픽셀 0과 픽셀 800이고, y-방향으로는 픽셀 0과 픽셀 600일 수 있다. 커서(310)의 속도가 경계들에서 +1로서 검출되면, 스크롤 서브시스템(308)은 경계들을 동일하게 유지할 수 있다. 하지만, 속도가 x-방향에서 +10으로 검출되면, 스크롤 서브시스템(308)은 경계들을 x-방향으로는 픽셀 0과 픽셀 720으로 또 y-방향으로는 픽셀 0과 픽셀 540으로 구성할 수 있다. 이 일례에서, 뷰(304)의 크기가 여전히 800 x 600이지만, 스크롤 동작이 뷰(304)의 물리적 경계에 도달하기 전에 행해지도록 구성될 수 있다. 이 일례에서, 커서(310)가 빠르게 움직이면, 클라이언트 서브시스템(306)은 커서(310)가 뷰(304)의 물리적 경계에 도달하기 전에 뷰를 패닝하도록 디스플레이 드라이버를 구성할 수 있다. 이 대안의 구현은 뷰(304)를 더 빠르게 패닝하도록 클라이언트 서브시스템(306)을 구성하기 위하여 다른 가속 기법들과 결합하여 사용될 수 있다.

[0034] 도 5의 설명을 계속하면, 도 5는 그에 부가하여 동작(402)의 미세 조정을 나타내고 있다. 미세 조정(518)은, 원격 컴퓨터로부터 수신된 기하 형태 정보로부터, 가상 데스크톱을 생성하는 것을 나타내며, 이 때 기하 형태 정보는 경계 상자와 원격 컴퓨터의 디스플레이 레이아웃을 정의하는 정보를 포함하며, 또한 경계 상자의 디스플레이 해상도는 원격 컴퓨터의 제1 및 제2 디스플레이 드라이버의 디스플레이 해상도 정보로부터 계산된다. 예를 들어, 본 발명의 실시예에서, 서버(204)의 원격 디스플레이 서브시스템(254)은 제1 디스플레이 드라이버 및 제2 디스플레이 드라이버와 관련된 픽셀 범위들의 합과 동일한 픽셀 범위를 정의하는 경계 상자를 생성하도록 구성될 수 있다. 이러한 예시적인 실시예들에서, 원격 디스플레이 서브시스템(254)은, 예를 들어, 제2 디스플레이 드라이버의 좌표를 경계 박스에 재매핑하도록 구성될 수 있다, 예를 들면, 제1 디스플레이 드라이버가 1280 x 1024의 디스플레이 해상도를 갖는 경우에, 800 x 600의 픽셀 범위를 갖는 제2 디스플레이 상의 픽셀(0,0)은 픽셀(1280,0)에 재매핑될 수 있다. 이 일례에서, 원격 디스플레이 서브시스템(254)은 서버(204) 상의 디스플레이

드라이버들의 논리적 레이아웃에 따라 각각의 디스플레이 드라이버의 픽셀들의 합을 포함하는 경계 박스를 생성하도록 구성될 수 있다. 일례로, 제1 디스플레이 드라이버가 논리적으로 제2 디스플레이 드라이버의 왼쪽에 이미지를 렌더링하도록 구성되어 있는 경우, 원격 디스플레이 서브시스템(254)에 의해 생성된 경계 상자는 2080 x 1024 픽셀의 해상도를 가질 수 있다. 다른 일례에서, 제1 디스플레이 드라이버가 논리적으로 제2 디스플레이 드라이버의 위쪽에 이미지를 렌더링하도록 구성되어 있는 경우, 원격 디스플레이 서브시스템(254)에 의해 생성된 경계 상자는 1280 x 1624 픽셀의 크기를 가질 수 있다. 이러한 상황들에서, 원격 디스플레이 서브시스템(254)은 크기를 나타내는 신호를 전송 논리(210)을 통해 클라이언트(201)로 전송하도록 구성될 수 있다. 이러한 일례에서, 클라이언트 서브시스템(306)은 경계 상자의 픽셀 범위를 포함하는 가상 데스크톱(302)을 생성하도록 구성될 수 있다. 경계 상자의 픽셀 범위를 전송하는 것에 추가하여, 원격 디스플레이 서브시스템(254)은 그에 추가하여 서버(204)의 디스플레이 드라이버들의 레이아웃을 전송하도록 구성될 수 있다. 일례로, 1280 x 1024의 디스플레이 해상도를 갖는 디스플레이 드라이버가 논리적으로 800 x 600의 픽셀 범위를 갖는 제2 디스플레이 드라이버의 왼쪽에 위치하는 경우, 가상 데스크톱의 픽셀들은 동일한 레이아웃을 포함하도록 구성될 수 있다, 예를 들면, 제2 디스플레이 드라이버의 픽셀들이 경계 상자의 픽셀 (1280,0)에서 시작하도록 재매핑될 수 있다. 한편, 1280 x 1024의 디스플레이 해상도를 갖는 디스플레이 드라이버가 논리적으로 800 x 600의 픽셀 범위를 갖는 제2 디스플레이 드라이버의 오른쪽에 위치하는 경우, 제1 디스플레이 드라이버의 픽셀들이 경계 상자의 픽셀 (800,0)에서 시작하도록 재매핑될 수 있다.

[0035] 이제 도 6을 참조하면, 도 6은 부가의 동작(620)을 포함하는 도 5의 동작 절차(400)의 대안의 실시예를 나타낸 것이다. 동작(620)에 관해서는, 이 동작은 원격 장치의 제1 및 제2 디스플레이 드라이버의 디스플레이 해상도 정보를 수신하는 것과, 디스플레이 해상도 정보로부터 제1 및 제2 디스플레이 드라이버의 경계를 구하는 것을 나타낸다. 예를 들어, 본 발명의 실시예에서, 서버(204)의 원격 디스플레이 서브시스템(254)은 제1 디스플레이 드라이버 및 제2 디스플레이 드라이버와 관련된 픽셀 범위들의 합과 동일한 픽셀 범위를 포함하는 경계 상자를 생성하도록 구성될 수 있다. 경계 상자를 클라이언트(201)로 전송하는 것에 추가하여, 원격 디스플레이 서브시스템(254)은 각각의 디스플레이 드라이버의 기하 형태를 나타내는 신호를 전송 논리(210)를 통하여 클라이언트(201)로 전송하도록 구성될 수 있다. 이 예시적인 실시예에서, 클라이언트 서브시스템(306)은 각각의 디스플레이 드라이버의 기하 형태를 나타내는 하나 이상의 패킷을 수신하고, 디스플레이 드라이버들의 경계들과 연관된 가상 데스크톱(302) 내의 픽셀 위치들을 구하도록 구성될 수 있다. 이 예시적인 실시예에서, 클라이언트 서브시스템(306)은 이 정보를 스크롤 서브시스템(308)으로 전달하도록 구성될 수 있다.

[0036] 이제 도 7을 참조하면, 도 7은 부가의 동작(722 및 724)을 포함하는 도 6의 동작 절차(400)의 대안의 실시예를 나타낸 것이다. 동작(722)에 관해서는, 이 동작(722)은, 뷰를 정의하는 경계들과 제1 및 제2 디스플레이 드라이버를 정의하는 경계들로부터, 뷰의 경계가 제1 디스플레이 드라이버의 경계에 도달했는지 판정하는 하위 동작, 커서의 속도가 미리 정해진 값보다 큰 지를 판정하는 하위 동작, 및 가상 데스크톱의 뷰를 제2 디스플레이 드라이버와 연관된 먼쪽 경계로 스크롤하는 하위 동작을 나타낸다. 일례로, 본 발명의 일 실시예에서의 이전의 일례에 추가하여, 스크롤 서브시스템(308)은 뷰(304)를 제2 디스플레이의 픽셀들과 연관되어 있는 먼쪽 경계로 패닝하도록 구성될 수 있다. 예시적인 구현에서, 스크롤 서브시스템(308)은 뷰(304)의 경계가 제1 디스플레이와 관련된 경계에 도달하고 커서(310)가 미리 정해진 속도로 움직이고 있는 경우에, 이 점프 동작을 수행하도록 구성될 수 있다. 예를 들어, 이 일례에서, 스크롤 서브시스템(308)은 가상 데스크톱(302)에 대한 뷰(304)의 경계들, 서버(204)의 디스플레이 드라이버들의 경계들을 정의하는 가상 데스크톱(302)에 대한 픽셀 값들, 및 커서(310)의 속도를 식별해주는 정보를 수신하도록 구성될 수 있다. 이 예시적인 실시예에서, 스크롤 서브시스템(308)은 뷰(304)의 경계가 원격 디스플레이 드라이버의 경계와 만나는 때를 검출하도록 구성될 수 있다. 스크롤 서브시스템(308)이 경계에 도달했다면, 예를 들어, 뷰(304)의 경계가 픽셀 1280에 있고 제1 예시적인 디스플레이 드라이버의 해상도가 픽셀 1280에서 끝나는 것으로 판정하는 경우, 스크롤 서브시스템(308)은 커서(310)의 속도를 구하도록 구성될 수 있다. 커서(310)의 속도가 미리 정해진 임계값(예를 들어, 샘플당 +15 픽셀)보다 크다면, 스크롤 서브시스템(308)은 사용자가 전체 화면을 점프하기를 원하는 것으로 판정하고, 서버(204)의 원격 디스플레이 드라이버와 관련된 먼쪽 경계(예를 들어, 가상 데스크톱(302)의 픽셀 1624)로 뷰(304)를 이동시키라는 신호를 클라이언트 서브시스템(306)에게 보내도록 구성될 수 있다. 일 실시예에서, 사용자가 서로 다른 디스플레이들과 관련된 픽셀 범위들에서 응용 프로그램들의 최대화된 창 크기를 갖는 경우에, 이 동작이 바람직할 수 있다. 사용자가 디스플레이들 사이에서 왔다갔다 점프하기를 원한다면, 사용자는 마우스를 특정 속도로 움직일 수 있고 스크롤 서브시스템(308)은 디스플레이 전체에 걸쳐 가상 데스크톱의 뷰(304)를 '범프'할 수 있다.

[0037] 대안의 실시예에서, 가상 데스크톱(302)의 먼쪽 경계로 점프하는 대신에, 뷰(304)는 뷰(304)의 디스플레이 해상도와 동일한 일정 범위의 픽셀들만큼 점프하도록 구성될 수 있다. 예를 들어, 이 예시적인 실시예에서, 스크롤

서브시스템(308)은 가상 데스크톱(302)에 대한 뷰(304)의 경계들 및 커서(310)의 속도를 식별해주는 정보를 수신하도록 구성될 수 있다. 이 예시적인 실시예에서, 스크롤 서브시스템(308)은 커서(310)가 뷰(304)의 특정의 경계에 도달하는 때를 검출하도록 구성될 수 있다. 스크롤 서브시스템(308)이 특정의 경계에 도달하였고 커서(310)의 속도가 미리 정해진 임계값(예를 들어, 샘플당 +15 픽셀)보다 큰 것으로 판정하는 경우, 스크롤 서브시스템(308)은 사용자가 뷰(304)를 점프하기를 원하는 거스로 판정하고, 뷰의 오른쪽 경계를 왼쪽으로 800 픽셀만큼 옮기라는 신호를 클라이언트 서브시스템(306)으로 보내도록 구성될 수 있다.

[0038] 도 7의 설명을 계속하면, 동작(724)은, 커서의 좌표 정보와 제1 및 제2 디스플레이 드라이버를 정의하는 경계들로부터, 커서가 제1 디스플레이 드라이버의 경계에 도달했는지를 판정하는 것과 햅틱 응답(haptic response)을 생성하는 것을 나타낸다. 일례로, 본 발명의 실시예에서의 이전의 일례에 부가하여, 스크롤 서브시스템(308)은 햅틱 응답[예를 들어, 클라이언트(201)나 주변장치의 구성요소에 의해 처리될 때, 진동을 생성하는 신호]을 생성하도록 구성될 수 있다. 일례로, 마우스나 터치패드는 스크롤 서브시스템(308)으로부터 수신되는 전기적인 신호에 응답하여 질량(mass)을 회전시키는 모터를 포함하도록 구성될 수 있다. 이 일례에서, 질량을 회전시키는 동작에 의해 마우스의 하우징 또는 랩톱이 흔들리게(ramble) 된다. 이 일례에서, 스크롤 서브시스템(308)은 뷰(304)의 위치를 식별해주는 정보, 예를 들어, 커서(310)의 위치를 정의하는 가상 데스크톱(302)에 대한 픽셀 값들과 서버(204)의 디스플레이 드라이버들의 경계들을 정의하는 가상 데스크톱(302)에 대한 픽셀 값들을 수신하도록 구성될 수 있다. 이 예시적인 실시예에서, 스크롤 서브시스템(308)은 커서(310)의 위치가 원격 디스플레이 드라이버의 경계와 만나는 때를 검출하도록 구성될 수 있다. 스크롤 서브시스템(308)이 경계에 도달했다고, 예를 들어, 2개의 디스플레이 드라이버가 1280 x 1024 및 800 x 600인 실시예에서 커서(310)의 위치가 픽셀 1280 위에 있다고 판정하는 경우, 스크롤 서브시스템(308)은 햅틱 서브시스템에게 사용자 입력 장치를 진동시키라고 지시하는 신호를 마우스와 같은 사용자 입력 장치로 보내도록 구성될 수 있다.

[0039] 이제 도 8을 참조하면, 동작(800) 내지 동작(808)을 포함하는 본 발명의 측면들을 실시하는 예시적인 플로우차트가 나타내어져 있다. 본 발명의 실시예에서, 도 8의 예시적인 동작 절차는 컴퓨터-관독가능 저장 매체에 저장된 컴퓨터-관독가능 명령어들로 구현될 수 있다. 컴퓨터-관독가능 명령어들은 클라이언트(201)의 프로세서에 의해 처리될 수 있으며 이 프로세서는 동작 절차들을 수행할 수 있다. 이제 도 8을 참조하면, 예시적인 동작(800)은 동작 절차를 시작하고, 동작(802)은 원격 컴퓨터로부터 수신된 그래픽을 그리는 가상 데스크톱의 뷰를 렌더링하는 것을 나타내며, 이 때 뷰는 픽셀 범위에 의해 정의되고, 가상 데스크톱은 원격 컴퓨터의 디스플레이 해상도에 대응하는 픽셀 범위에 의해 정의된다. 일례로, 클라이언트 서브시스템(306)은 디스플레이 드라이버에게 클라이언트(201)의 메모리 내에 가상 데스크톱(302)을 생성하고, 서버(204)로부터 수신된 그래픽을 가상 데스크톱(302) 상에 드로잉하라고 지시하도록 구성될 수 있다. 이 예시적인 실시예에서, 클라이언트(201)의 디스플레이 드라이버는 픽셀 범위(예를 들어, 800 x 600)에 의해 정의되는 가상 데스크톱(302)의 뷰(304)를 렌더링하도록 구성될 수 있다. 예를 들면, 본 발명의 실시예에서, 가상 데스크톱(302) 전체가 디스플레이 드라이브에 의해 메모리에 드로잉되는 반면에, 디스플레이 드라이버는, 그래픽의 일부가 가상 데스크톱(302)의 현재 뷰 내에 위치하는지 여부에 기초하여, 클라이언트(201)에 연결된 디스플레이 상에 그래픽의 일부분을 렌더링할 수 있다. 일 실시예에서, 클라이언트(201)의 디스플레이 드라이버는 가상 데스크톱(302)의 디스플레이 해상도보다 작은 특정의 디스플레이 해상도를 가질 수 있다, 예를 들어, 가상 데스크톱(302)의 크기가 2048 x 1536 픽셀일 수 있는 반면에, 클라이언트(201)의 디스플레이 드라이버는 단지 800 x 600 픽셀의 디스플레이 해상도를 렌더링할 수 있을 수 있다. 이 일례에서, 클라이언트 서브시스템(306)은 뷰(304)의 디스플레이 해상도를 정의하는 정보 및 가상 데스크톱(302)에 대한 뷰의 좌표를 수신하도록 구성될 수 있으며, 이미지를 뷰(304)의 크기 내에 나타나게 렌더링하도록 디스플레이 드라이버를 구성할 수 있다.

[0040] 도 8의 설명을 계속하면, 도 8은 그에 부가하여 상대적 커서 이동 정보를 수신하는 것을 나타내는 동작(804)를 나타내고 있다. 예를 들면, 이전의 일례에 부가하여, 스크롤 서브시스템(308)은 클라이언트(201)의 운영 체제(312)로부터 상대적 마우스 이동 정보를 수신하도록 구성될 수 있다. 예를 들어, 일 실시예에서, 운영 체제(312)는 뷰(304) 내에서의 커서(310), 예를 들어, 비트맵의 이동을 추적하도록 구성될 수 있다. 몇몇 실시예들에서, 커서(310)는 핫스팟(hotspot)이라고 불리는 단일-픽셀 점, 예를 들어, 운영 체제(312)가 추적하여 커서(310)의 위치로서 인식하는 점을 포함할 수 있다. 이 예시적인 실시예에서, 스크롤 서브시스템(308)은 물리적 입력 장치(예를 들어, 터치패드나 마우스)가 움직인 방향을 식별해주는, 예를 들어, 입력 메시지들을 미리 정해진 간격으로 수신하도록 구성될 수 있다. 예를 들어, 일 실시예에서, 클라이언트(201)의 운영 체제(312)는 입력 장치(예를 들어, 마우스나 터치패드)가 물리적으로 움직일 때마다 추적하도록 구성될 수 있다. 이 일례에서, 입력 메시지는 입력 장치가 오른쪽 위로 움직인 경우에 (+1,+1)과 같은 정보를 포함할 수 있다. 대안의 실시예에서, 스크롤 서브시스템(308)은 커서(310)의 좌표를 식별해주는 절대적 커서 이동 정보를 추적하고 가상 데스크

톱(302)에 대한 커서의 이동을 계산하도록 구성될 수 있다.

[0041] 도 8의 설명을 계속하면, 도 8은 그에 부가하여, 상대적 커서 이동 정보와 뷰를 정의하는 픽셀 범위로부터, 커서가 뷰의 경계에 도달했는지를 판정하는 것을 나타내는 동작(806)을 나타내고 있다. 예를 들어, 본 발명의 일 실시예에서, 스크롤 서브시스템(308)은 가상 데스크톱(302)에 대한 뷰(304)의 픽셀 범위 및 커서(310)의 현재 위치를 식별해주는 정보를 수신하도록 구성될 수 있다. 스크롤 서브시스템(308)은 그에 부가하여, 예를 들면, 절대적 마우스 이동 정보 및/또는 상대적 마우스 이동 정보를 포함할 수 있는 메시지를 수신하도록 구성될 수 있다. 스크롤 서브시스템(308)은 가상 데스크톱(302)에 대한 커서(310)의 위치를 확인하고 그 위치를 가상 데스크톱(302)에 대한 뷰(304)를 정의하는 픽셀 범위와 비교함으로써, 예를 들면, 마우스 이동 메시지가 수신될 때마다, 가상 데스크톱(302)에 대한 커서(310)의 x 및 y 좌표를 가상 데스크톱(302)에 대한 뷰(304)의 경계들을 정의하는 x 및 y 좌표와 비교함으로써, 커서(310)가 뷰(304)의 경계에 도달했는지 여부를 판정할 수 있다. 특정의 일례에서, 스크롤 서브시스템(308)은 가상 데스크톱에서의 커서(310)의 좌표(예를 들어, 픽셀 400, 픽셀 300) 및 가상 데스크톱의 뷰를 정의하는 픽셀들(예를 들어, x축에서 픽셀 0 내지 800과 y축에서 픽셀 0 내지 600)을 포함하는 테이블을 포함할 수 있다. 사용자가 마우스를 움직임에 따라, 커서(310)는, 예를 들어, 뷰(304)를 가로질러 좌에서 우로 이동할 수 있다. 스크롤 서브시스템(308)은 커서(310)의 픽셀 위치를 증분시킬 수 있으며, 각각의 증분에 대해, 픽셀이 경계(예를 들어, 이 일례에서, x축 픽셀 800)에 도달했는지를 검사한다. 스크롤 서브시스템(308)이 x축 경계에 도달했다고(예를 들어, 커서(310)가 픽셀 점 800, 300에 위치한다고) 판정하는 경우에, 스크롤 서브시스템(308)은 클라이언트 서브시스템(306)으로 메시지를 보내도록 구성될 수 있다.

[0042] 도 8의 설명을 계속하면, 도 8은 그에 부가하여 상대적 커서 이동의 방향으로 가상 데스크톱의 뷰를 스크롤하는 것을 나타내는 동작(808)을 나타내고 있다. 예를 들어, 이전의 일례에 부가하여, 스크롤 서브시스템(308)이 뷰(304)의 경계에 도달했다고 판정하는 경우에, 스크롤 서브시스템(308)은 상대적 마우스 이동의 방향으로 뷰(304)를 패닝하라는 신호를 클라이언트 서브시스템(306)으로 보내도록 구성될 수 있다. 예를 들어, 특정의 실시예에서, 스크롤 서브시스템(308)은 커서(310)가 뷰(304)의 오른쪽 경계에 위치했을 때 마우스가 x-방향으로 +1 움직였다는 것을 나타내는 신호를 수신할 수 있다. 스크롤 서브시스템(308)은 뷰를 x-축 상에서 +1 픽셀만큼 패닝하라는 메시지를 클라이언트 서브시스템(306)으로 보내도록 구성될 수 있다. 이 일례에서, 커서(310)가 x-방향으로 한번더 +1 픽셀만큼 움직일 수 있고, 스크롤 서브시스템(308)은 커서(310)가 새로운 경계 상에 있는지를 판정할 수 있다. 따라서, 마우스가 또 다시 우측으로 움직인다면, 스크롤 서브시스템(308)은 뷰(304)를 또 다시 패닝하라는 신호를 클라이언트 서브시스템(306)으로 보내도록 구성될 수 있다.

[0043] 이제 도 9 내지 도 11을 참조하면, 이들 도면은 도 8의 동작 절차의 대안의 실시예들을 나타낸 것이다. 이제 도 9를 참조하면, 도 9는 부가의 동작 절차들(910 및 912)을 포함하는 도 8의 동작 절차(800)를 나타낸 것이다. 동작(910)은 가상 데스크톱을 다수의 구역으로 나누는 것, 각각의 구역을 키보드의 키에 할당하는 것, 특정의 구역에 할당된 키의 선택을 수신하는 것, 키의 선택에 응답하여 뷰를 특정의 구역의 중앙에 오도록 하는 것을 나타낸다. 예를 들어, 본 발명의 실시예에서, 서버(204)의 원격 디스플레이 서브시스템(254)은 가상 데스크톱의 기하 형태를 정의하는 신호를 생성하도록 구성될 수 있으며, 서버(204)는 기하 형태 정보를 나타내는 신호를 클라이언트(201)로 전송하도록 구성될 수 있다. 이 예시적인 실시예에서, 클라이언트 서브시스템(306)은 메모리 내에 가상 데스크톱(302)을 생성하고 기하 형태를 나타내는 신호를 스크롤 서브시스템(308)으로 전송하도록 구성될 수 있다. 이 예시적인 실시예에서, 스크롤 서브시스템(308)은 가상 데스크톱(302)을 다수의 구역으로 나누도록 구성될 수 있고, 이 때 각각의 구역에는 픽셀 범위가 할당될 수 있다. 가상 데스크톱이 다수의 구역으로 나누어지면, 구역의 식별자가 테이블에 저장되고 키보드 상의 키와 연관될 수 있다. 일례로, 가상 데스크톱(302)이 4개의 구역으로 나누어지는 경우에, 키 '9', '7', '1' 및 '3'이 가상 데스크톱(302)의 4개의 사분면에 대응하는 데 사용될 수 있다. 이 일례에서, 스크롤 서브시스템(308)이, 예를 들어, 컨트롤 키가 눌러져 있는 동안에, 키 '9', '7', '1' 및 '3'을 포함하는 통지를 수신하면, 스크롤 서브시스템(308)은 클라이언트 서브시스템(306)에게 뷰(304)의 중심을 선택된 구역의 중앙을 정의하는 픽셀에 위치시키라고 지시하는 신호를 클라이언트 서브시스템(306)으로 보내도록 구성될 수 있다.

[0044] 도 9의 설명을 계속하면, 동작(912)은 뷰에 미니-맵(mini-map)을 렌더링하는 것을 나타내며, 이 때 미니-맵은 가상 데스크톱의 스케일링된 이미지 및 뷰의 스케일링된 이미지를 나타낸다. 예를 들어, 본 발명의 실시예에서, 클라이언트 서브시스템(306)은 뷰(304)가 렌더링되고 있는 가상 데스크톱(302)의 일부분을 나타내기 위해 가상 데스크톱(302) 및 뷰(304)의 스케일링된 이미지를 보여주는 미니-맵(316)을 생성하도록 구성될 수 있다. 예를 들어, 미니-맵(316)을 포함하는 실시예에서, 클라이언트 서브시스템(306)은 가상 데스크톱(302)을 정의하는 픽

셀 범위, 및 가상 데스크톱(302)에 대한 뷰(304)를 정의하는 픽셀 범위를 수신하도록 구성될 수 있다. 이 예시적인 실시예에서, 클라이언트 서브시스템(306)은 뷰(304)와 가상 데스크톱(302) 둘다에 대한 픽셀 범위를 나누는 것 및 가상 데스크톱(302)에서의 뷰(304)의 중심의 픽셀 좌표에 대응하는 스케일링된 뷰의 중심을 구하는 것에 의해 미니-맵(316)을 생성하도록 구성될 수 있다. 이 예시적인 실시예에서, 클라이언트 서브시스템(306)은 뷰(304)의 위치를 뷰(304)의 일부분 위에 디스플레이하는 미니-맵(316)을 렌더링하도록 클라이언트(201)의 디스플레이 드라이버를 구성할 수 있다. 이 예시적인 실시예에서, 뷰(304)가 스크롤될 때, 뷰(304)를 정의하는 픽셀 범위가 갱신될 수 있으며, 클라이언트 서브시스템(306)은 뷰(304)가 변경된 것을 반영하는 갱신된 미니-맵(316)을 렌더링하도록 클라이언트(201)의 디스플레이 드라이버를 구성할 수 있다.

[0045] 본 발명의 다른 실시예에서, 미니-맵(316)은, 스케일링된 뷰와 스케일링된 가상 데스크톱(302)에 부가하여, 서버(204)로부터 수신된 스케일링된 이미지를 렌더링하도록 구성될 수 있다. 일례로, 서버(204)로부터 수신된 각각의 창 및/또는 아이콘 드로잉 명령은 가상 데스크톱(302) 내에서의 픽셀 위치와 연관될 수 있다. 이 예시적인 실시예에서, 클라이언트 서브시스템(306)은 창 및/또는 아이콘을 스케일링하고, 스케일링된 가상 데스크톱(302) 내에서 스케일링된 창 및/또는 아이콘의 픽셀 위치를 구하도록 구성될 수 있다. 이 예시적인 실시예에서, 클라이언트 서브시스템(306)은 스케일링된 아이콘 및/또는 창을 포함하는 미니-맵(316)을 뷰(304)의 일부분 상에 렌더링하도록 클라이언트(201)의 디스플레이 드라이버를 구성할 수 있다.

[0046] 이제 도 10을 참조하면, 도 10은 동작 절차의 미세 조정(1014) 및 부가의 동작 절차(1016)를 포함하는 도 9의 동작 절차(800)의 대안의 실시예를 나타낸 것이다. 미세 조정(1014)을 참조하면, 이는 뷰에 미니-맵(mini-map)을 렌더링하는 것을 나타내며, 이 때 미니-맵은 가상 데스크톱의 와이어프레임 이미지(wireframe image) 및 뷰의 와이어프레임 윤곽선(wireframe outline)을 포함한다. 예를 들어, 본 발명의 실시예에서, 미니-맵(316)은 뷰와 가상 데스크톱의 단색 또는 반투명 이미지를 뷰(304) 위에 렌더링 할 수 있지만, 일 실시예에서는, 계산 능력을 절감하기 위해서, 클라이언트 서브시스템(306)이 뷰(304)와 가상 데스크톱(302)의 와이어프레임을 생성하도록 구성될 수 있다. 일례로, 실시예에서, 와이어프레임은 픽셀 범위로 정의되는 뷰와 가상 데스크톱의 경계를 식별함으로써 생성되는 시각적 표현(visual presentation)을 포함할 수 있다.

[0047] 도 10의 설명을 계속하면, 동작(1016)은 사용자 입력을 원격 컴퓨터로 전송하는 것 - 원격 컴퓨터는 사용자 입력에 응답하여 프로그램을 실행하도록 구성되어 있음 -, 프로그램에 대한 사용자 입력을 나타내는 그래픽을 수신하는 것, 그리고 프로그램의 사용자 인터페이스의 스케일링된 이미지로 미니-맵을 갱신하는 것을 나타낸다. 예를 들어, 본 발명의 실시예에서, 클라이언트(201)의 네트워크 어댑터는 가상 데스크톱(302)에 대한 커서(310)의 좌표 및 마우스 두 번 클릭과 같은 사용자 입력을 나타내는 하나 이상의 패킷을 전송하도록 구성될 수 있다. 이 일례에서, 도 2를 참조하면, 사용자 입력은 세션과 연관된 스택 인스턴스에 의해 수신되어, 입력 서브시스템(252)으로 전송될 수 있다. 입력 서브시스템(252)은 커서(310)의 핫스팟의 좌표가 프로그램에 대한 아이콘이 위치한 곳인지를 판정할 수 있다. 세션 코어(244)는 식별된 프로그램에 대한 핸들을 포함하는 신호를 런타임 서브시스템(240)으로 보낼 수 있다. 런타임 서브시스템(240)은 프로그램에 대한 하나 이상의 스레드를 인스턴스화할 수 있는 프로세스를 실행할 수 있다. 스레드는 서버(204)의 프로세서에 의해 처리되도록 서버 운영 체제(214)의 커널(214)에 의해 스케줄링될 수 있으며, 프로그램을 나타내는 사용자 인터페이스가 GDI(246)에 의해 생성될 수 있다. 이 일례에서, 원격 디스플레이 서브시스템(254)은 사용자 인터페이스를 나타내는 드로잉 명령을 수신하고, 이 명령을 세션의 스택 인스턴스로 라우팅하도록 구성될 수 있다. 그 후에, 전송 논리(210)는 패킷을 클라이언트(201)로 전송할 수 있다. 이제 도 3을 참조하면, 클라이언트 서브시스템(306)이 패킷을 수신하면, 클라이언트 서브시스템(306)은 사용자 인터페이스의 좌표를 식별하고 가상 데스크톱(302)에 사용자 인터페이스를 드로잉하도록 구성될 수 있다. 이 일례에서, 클라이언트 서브시스템(306)은 가상 데스크톱(302)을 정의하는 픽셀 범위, 가상 데스크톱(302)에 대한 뷰를 정의하는 픽셀 범위, 및 프로그램의 사용자 인터페이스를 정의하는 픽셀 범위를 수신하도록 구성될 수 있다. 이 예시적인 실시예에서, 클라이언트 서브시스템(306)은 가상 데스크톱의 픽셀 범위 및 프로그램의 스케일링된 사용자 인터페이스의 픽셀 범위를 나눔으로써 미니-맵(316)을 생성하고 프로그램의 스케일링된 사용자 인터페이스의 중앙을 결정하도록 구성될 수 있다. 이 예시적인 실시예에서, 클라이언트 서브시스템(306)은 프로그램의 스케일링된 사용자 인터페이스 또는, 예를 들면, 프로그램의 스케일링된 사용자 인터페이스의 와이어프레임을 포함하는 미니-맵(316)을 뷰(304)의 일부분 상에 렌더링하도록 클라이언트(201)의 디스플레이 드라이버를 구성할 수 있다. 다른 실시예에서, 클라이언트 서브시스템(306)은, 사용자 인터페이스가 가상 데스크톱(302) 상의 어디에 렌더링되었는지를 사용자가 빠르게 식별할 수 있게 해주기 위해서, 미리 정해진 시간 동안 스케일링된 사용자 인터페이스의 색상을 변경하도록 구성될 수 있다.

[0048] 이제 도 11을 참조하면, 도 11은 가상 데스크톱 상의 사용자 인터페이스의 위치를 나타내는 좌표를 수신하는

것, 사용자 입력을 수신하는 것, 및 사용자 입력에 응답하여 뷰를 사용자 인터페이스의 중앙에 오도록 하는 것을 나타내는 동작 절차(1118)를 포함하는 도 10의 동작 절차(800)의 대안의 실시예를 나타낸 것이다. 예를 들어, 일 실시예에서, 새 창 또는 프로그램의 사용자 인터페이스가 클라이언트 서브시스템(306)에 의해 드로잉될 때, 창 또는 사용자 인터페이스의 좌표를 식별해주는 통지가 스크롤 서브시스템(308)으로 보내질 수 있다. 이 실시예에서, 좌표를 나타내는 핸들이 테이블에 저장될 수 있으며, 키 또는 마우스 버튼과 같은 사용자 입력은 이 핸들과 연관될 수 있다. 이 예시적인 실시예에서, 스크롤 서브시스템(308)은 사용자 입력과 관련한 통지를 수신하고, 수신된 사용자 입력으로부터, 핸들과 연관된 키 또는 마우스 단추가 선택되었는지를 판정하도록 구성될 수 있다. 선택된 경우에, 스크롤 서브시스템(308)은 새 창 또는 프로그램의 사용자 인터페이스의 좌표 상에 뷰(304)의 중앙이 오도록 하라고 클라이언트 서브시스템(306)에게 지시하는 신호를 클라이언트 서브시스템(306)으로 보낼 수 있다. 이 일례에서, 클라이언트 서브시스템(306)은 그 메시지를 수신하고 디스플레이 드라이버에게 뷰(304)를 중앙에 오게 하라고 지시할 수 있다.

[0049] 이제 도 12를 참조하면, 도 12는 동작들(1200, 1202, 1204 및 1206)을 포함하는 본 발명의 측면들을 실시하는 예시적인 플로우차트를 나타낸 것이다. 본 발명의 실시예에서, 컴퓨터 시스템은 도 12의 동작 절차들을 실시하도록 구성된 회로, 예를 들어, 하드웨어와 소프트웨어의 조합을 포함할 수 있다. 이제 도 3과 함께 도 12를 참조하면, 동작(1200)은 동작 절차를 시작하고, 동작(1202)은 가상 데스크톱의 뷰를 렌더링하는 것을 나타내며, 여기에서 가상 데스크톱은 먼저 원격 컴퓨터의 디스플레이 해상도에 대응하는 제1 픽셀 범위에 의해 정의되고, 뷰는 제2 픽셀 범위에 의해 정의된다. 예를 들어, 본 발명의 실시예에서, 클라이언트(201)는 커널(314)을 갖는 운영 체제(312)를 포함하도록 구성될 수 있다. 커널(314)은 클라이언트(201)의 하드웨어를 관리하고 프로세서에서 실행되는 스레드를 스케줄링하도록 구성될 수 있다. 운영 체제(312)는 사용자 모드와 커널 모드, 예를 들면, 제한된 수의 권한을 갖는 응용 프로그램이 실행되는 비특권 프로세서 모드(non-privileged processor mode)와 모든 메모리 및 CPU 명령어들이 발행될 수 있는 프로세서에서의 특권 코드 실행 모드(privileged mode of code execution)를 포함하도록 구성될 수 있다. 이 일례에서, 커널(314)은, 서버(204)로부터 드로잉 명령을 수신하고 디스플레이 드라이버에게 드로잉 명령에 따라 메모리에 저장된 가상 데스크톱(302)에 드로잉하도록 지시함으로써, 원격 데스크톱 환경을 실시하는 하나 이상의 프로세스를 포함할 수 있는 클라이언트 서브시스템(306)과 인터페이스 할 수 있다. 위에 설명된 것과 유사하게, 가상 데스크톱(302)은 클라이언트 서브시스템(306)에 의해 생성될 수 있으며, 가상 데스크톱(302)의 픽셀 범위는 서버(204)의 디스플레이 드라이버의 디스플레이 해상도에 대응할 수 있다. 예를 들어, 이 실시예에서, 서버(204)는 1280 x 1024의 디스플레이 해상도를 생성하도록 구성되어 있는 디스플레이 드라이버를 포함할 수 있다. 이 일례에서, 가상 데스크톱(302)은 1280 x 1024의 디스플레이 해상도를 포함하도록 생성될 수 있다. 다른 실시예에서, 서버(204)는 각각이 1280 x 1024의 디스플레이 해상도를 생성하는 2개의 디스플레이 드라이버를 포함할 수 있다. 이 일례에서, 가상 데스크톱(302)은 2560 x 1024의 디스플레이 해상도를 포함하도록 구성될 수 있다. 서버(204)의 디스플레이 해상도가 클라이언트(201)에서 동일하게 유지되는 예시적인 실시예에서, 클라이언트 서브시스템(306)은 비용이 많이 드는 크기 조정 및 재매핑 동작을 수행하지 않고 이미지를 렌더링할 수 있다, 즉 가상 데스크톱(302)의 기하 형태가 서버(204)의 디스플레이 드라이버와 동일하기 때문에, 서버(204)에서 좌표(1000, 900)에 위치하는 아이콘이 아이콘의 크기 조정 및 재매핑 없이 클라이언트(201)의 동일한 좌표에 드로잉될 수 있다.

[0050] 가상 데스크톱(302)을 드로잉하는 것에 부가하여, 클라이언트 서브시스템(306)은 디스플레이 드라이버에게 가상 데스크톱(302)의 뷰(304)를 렌더링하라고 지시하도록 구성될 수 있다. 이 예시적인 실시예에서, 뷰(304)는 제2 픽셀 범위에 의해 정의될 수 있으며, 클라이언트(201)의 디스플레이 드라이버는 가상 데스크톱(302)의 보여진 부분에 이미지를 렌더링하도록 구성될 수 있다. 이 예시적인 실시예에서, 뷰(304)를 정의하는 픽셀 범위가 가상 데스크톱(302)을 정의하는 픽셀 범위보다 작을 수 있다. 예를 들어, 일 실시예에서, 클라이언트(201)가 1280 x 1024 픽셀들을 렌더링할 수 있는 스크린을 갖는 랩톱 컴퓨터일 수 있는 반면, 서버(204)는 각각이 1280 x 1024 픽셀을 렌더링하는 다수의 모니터들을 갖는다[따라서, 서버(204)는 본질적으로 5120 x 1024의 실면적을 갖는다]. 다른 일례에서, 뷰(304)는 클라이언트(201)의 데스크톱 상의 조절가능한 창(adjustable window)일 수 있다. 따라서, 일 실시예에서, 서버(204)와 클라이언트 둘다가 1280 x 1024의 디스플레이 해상도를 가질 수 있지만, 뷰(304)가 창의 형태로 되어 있기 때문에, 뷰가 바로 그 때 600 x 400 픽셀을 포함할 수 있고, 클라이언트(201)는 나머지 픽셀들에 의해 렌더링되는 다른 프로그램의 사용자 인터페이스를 가질 수 있다.

[0051] 본 발명의 일 실시예에서, 뷰(304)를 정의하는 픽셀 범위가 가상 데스크톱(302)을 정의하는 픽셀 범위에 매핑될 수 있다. 일례로, 클라이언트 서브시스템(306)은 뷰(304)의 현재의 픽셀 범위를 나타내는 정보를 포함하는 테이블을 포함하도록 구성될 수 있다. 특정의 일례에서, 가상 데스크톱의 뷰가 800 x 600의 픽셀 범위를 갖고 있고 뷰(304)가 가상 데스크톱(302)의 최우측 상단 부분을 렌더링하고 있는 경우, 테이블은 뷰(304)의 현재의 픽셀

범위가 x-축으로는 픽셀 1280부터 480까지이고 y-축으로는 1024부터 424까지라는 것을 나타내는 정보를 포함할 수 있다. 다른 구현에서, 테이블은 단지 뷰의 우측 상단 픽셀(1280,1024)과 같은 하나의 점만을 포함할 수 있고, 경계를 정의하는 픽셀들을 계산하기 위하여 이 점과 뷰(304)의 디스플레이 해상도를 이용할 수 있다.

[0052] 도 12의 설명을 계속하면, 도 12는 커서 이동 정보를 수신하는 것을 나타내는 동작(1204)를 포함하고 있다. 예를 들면, 이전의 일례에 부가하여, 스크롤 서브시스템(308)은, 본 발명의 일 실시예에서, 클라이언트(201)의 운영 체제(312)로부터 커서 이동 정보를 수신하도록 구성될 수 있다. 예를 들어, 일 실시예에서, 운영 체제(312)는 커서(310)의 이동을 추적하도록 구성될 수 있다. 일 실시예에서, 커서(310)는 핫스팟(hotspot)이라고 불리는 단일-픽셀 점, 예를 들어, 운영 체제(312)가 추적하여 커서(310)의 위치로서 인식하는 점을 포함할 수 있다. 이 예시적인 구현에서, 스크롤 서브시스템(308)은 가상 데스크톱(302)에 대한 커서(310)의 절대 위치를 식별해주는 입력 메시지를 수신하고 가상 데스크톱(302)에 대해 커서가 이동한 방향을 결정하기 위하여 이 위치를 커서(310)의 이전 위치와 비교하도록 구성될 수 있다. 대안의 실시예에서, 입력 메시지는 상대적 마우스 이동 정보를 포함할 수 있다. 예를 들어, 일 실시예에서, 클라이언트(201)의 운영 체제(312)는 입력 장치(예를 들어, 마우스나 터치패드)가 물리적으로 움직일 때마다 추적하도록 구성될 수 있다. 이 일례에서, 입력 메시지는 입력 장치가 오른쪽 위로 움직인 경우에 (+1,+1)과 같은 정보를 포함할 수 있다. 다른 예시적인 실시예에서, 스크롤 서브시스템(308)은 절대적 및 상대적 마우스 이동 정보 둘다를 수신하도록 구성될 수 있다.

[0053] 도 12의 설명을 계속하면, 도 12는, 커서 이동 정보와 제2 픽셀 범위로부터, 커서가 제2 픽셀 범위의 경계에 도달했는지를 판정하는 것을 나타내는 동작(1206)을 포함하고 있다. 예를 들어, 본 발명의 일 실시예에서, 스크롤 서브시스템(308)은 가상 데스크톱(302)에 대한 뷰(304)의 픽셀 범위 및 커서(310)의 현재 위치를 식별해주는 정보를 수신하도록 구성될 수 있다. 스크롤 서브시스템(308)은 그에 부가하여, 예를 들면, 절대적 마우스 이동 정보 및/또는 상대적 마우스 이동 정보를 포함할 수 있는 메시지를 수신하도록 구성될 수 있다. 스크롤 서브시스템(308)은 가상 데스크톱(302)에 대한 커서(310)의 위치를 확인하고 그 위치를 가상 데스크톱(302)에 대한 뷰(304)를 정의하는 픽셀 범위와 비교함으로써, 예를 들면, 마우스 이동 메시지가 수신될 때마다, 가상 데스크톱(302)에 대한 커서(310)의 x 및 y 좌표를 가상 데스크톱(302)에 대한 뷰(304)의 경계들을 정의하는 x 및 y 좌표와 비교함으로써, 커서(310)가 뷰(304)의 경계에 도달했는지 여부를 판정할 수 있다. 특정의 일례에서, 스크롤 서브시스템(308)은 가상 데스크톱에서의 커서(310)의 좌표(예를 들어, 픽셀 400, 픽셀 300) 및 가상 데스크톱의 뷰를 정의하는 픽셀들(예를 들어, x축에서 픽셀 0 내지 800과 y축에서 픽셀 0 내지 600)을 포함하는 테이블을 포함할 수 있다. 사용자가 마우스를 움직임에 따라, 커서(310)는, 예를 들어, 뷰(304)를 가로질러 좌에서 우로 이동할 수 있다. 스크롤 서브시스템(308)은 커서(310)의 픽셀 위치를 증분시킬 수 있으며, 각각의 증분에 대해, 픽셀이 경계(예를 들어, 이 일례에서, x축 픽셀 800)에 도달했는지를 검사한다. 스크롤 서브시스템(308)이 x축 경계에 도달했다고(예를 들어, 커서(310)가 픽셀 점 800, 300에 위치한다고) 판정하는 경우에, 스크롤 서브시스템(308)은 클라이언트 서브시스템(306)으로 메시지를 보내도록 구성될 수 있다.

[0054] 도 12의 설명을 계속하면, 도 12는 커서 이동의 방향으로 뷰를 스크롤하는 것을 나타내는 동작(1206)을 포함하고 있다. 예를 들어, 이전의 일례에 부가하여, 스크롤 서브시스템(308)이 뷰(304)의 경계에 도달했다고 판정하는 경우에, 스크롤 서브시스템(308)은 마우스 이동의 방향으로 뷰(304)를 패닝하라는 신호를 클라이언트 서브시스템(306)으로 보내도록 구성될 수 있다. 예를 들어, 특정의 실시예에서, 스크롤 서브시스템(308)은 마우스가 x-축으로 (-10,0)만큼 이동했다는 것을 나타내는 신호를 수신할 수 있다. 스크롤 서브시스템(308)은 커서(310)의 위치를 변경하고 그 위치를 뷰(304)의 경계들을 정의하는 픽셀들과 비교하도록 구성될 수 있다. 이 일례에서, 뷰(304)의 좌측 경계에 도달했을 수 있으며, 스크롤 서브시스템(308)은 가상 데스크톱(302)의 뷰(304)를 움직이라고 지시하는 메시지를 클라이언트 서브시스템(306)으로 보내도록 구성될 수 있다. 특정의 일례에서, 뷰(304)는 800 x 600 픽셀의 해상도를 포함할 수 있으며, 가상 데스크톱(302)의 최우측 상단 부분을 보여줄 수 있다. 이 일례에서, 클라이언트 서브시스템(306)은 뷰, 예를 들면, x-축에서 1280부터 480까지, y-축에서 1024부터 424까지를 정의하는 좌표를 저장하는 테이블을 포함한다. 이 예시적인 실시예에서, 스크롤 서브시스템(308)이 뷰(304)가 (-10,0)만큼 이동해야 하는 것으로 판정하는 경우, 클라이언트 서브시스템(306)은 x-축으로는 1270부터 470까지, y-축으로는 1024부터 424까지의 픽셀 범위를 렌더링하라고 디스플레이 드라이버에게 지시하도록 구성될 수 있다.

[0055] 이제 도 13 내지 도 16을 참조하면, 이들 도면은 도 12의 동작 절차(1200)에 대한 예시적인 미세 조정들 및 부가의 동작 절차들을 나타낸 것이다. 이제, 도 13의 설명을 계속하면, 도 13은 뷰에 미니-맵(mini-map)을 렌더링하는 것을 나타내는 부가의 동작(1308)을 포함하는 도 12의 동작 절차를 나타내며, 이 때 미니-맵은 가상 데스크톱의 스케일링된 이미지 및 뷰의 스케일링된 이미지를 나타낸다. 예를 들어, 본 발명의 실시예에서, 클라이언

트 서브시스템(306)은 뷰(304)가 보여지고 있는 가상 데스크톱(302)의 일부분을 나타내기 위해 가상 데스크톱(302) 및 뷰(304)의 스케일링된 이미지를 보여주는 미니-맵(316)을 생성하도록 구성될 수 있다. 예를 들어, 미니-맵(316)을 포함하는 실시예에서, 클라이언트 서브시스템(306)은 가상 데스크톱(302)을 정의하는 픽셀 범위, 및 가상 데스크톱(302)에 대한 뷰(304)를 정의하는 픽셀 범위를 수신하도록 구성될 수 있다. 이 예시적인 실시예에서, 클라이언트 서브시스템(306)은 뷰(304)와 가상 데스크톱(302) 둘다에 대한 픽셀 범위를 나누는 것 및 가상 데스크톱(302)에서의 뷰(304)의 중심의 픽셀 좌표에 대응하는 스케일링된 뷰의 중심을 구하는 것에 의해 미니-맵(316)을 생성하도록 구성될 수 있다. 이 예시적인 실시예에서, 클라이언트 서브시스템(306)은 뷰(304)의 위치를 뷰(304)의 일부분 위에 디스플레이하는 미니-맵(316)을 렌더링하도록 클라이언트(201)의 디스플레이 드라이버를 구성할 수 있다. 이 예시적인 실시예에서, 뷰(304)가 스크롤될 때, 뷰(304)를 정의하는 픽셀 범위가 갱신될 수 있으며, 클라이언트 서브시스템(306)은 뷰(304)가 변경된 것을 반영하는 갱신된 미니-맵(316)을 렌더링하도록 클라이언트(201)의 디스플레이 드라이버를 구성할 수 있다.

[0056] 본 발명의 다른 실시예에서, 미니-맵(316)은, 스케일링된 뷰와 스케일링된 가상 데스크톱(302)에 부가하여, 서버(204)로부터 수신된 스케일링된 이미지를 렌더링하도록 구성될 수 있다. 일례로, 서버(204)로부터 수신된 각각의 창 및/또는 아이콘 드로잉 명령은 가상 데스크톱(302) 내에서의 픽셀 위치와 연관될 수 있다. 이 예시적인 실시예에서, 클라이언트 서브시스템(306)은 창 및/또는 아이콘을 스케일링하고, 스케일링된 가상 데스크톱 내에서 스케일링된 창 및/또는 아이콘의 픽셀 위치를 구하도록 구성될 수 있다. 이 예시적인 실시예에서, 클라이언트 서브시스템(306)은 스케일링된 아이콘 및/또는 창을 포함하는 미니-맵(316)을 뷰(304)의 일부분 상에 렌더링하도록 클라이언트(201)의 디스플레이 드라이버를 구성할 수 있다.

[0057] 본 발명의 또 다른 실시예에서, 미니-맵(316)은 뷰와 가상 데스크톱의 단색 또는 반투명 이미지를 뷰(304) 위에 렌더링 할 수 있지만, 일 실시예에서는, 계산 능력을 절감하기 위해서, 클라이언트 서브시스템(306)이 뷰(304)와 가상 데스크톱(302)의 와이어프레임을 생성하도록 구성될 수 있다. 일례로, 실시예에서, 와이어프레임은 픽셀 범위로 정의되는 뷰와 가상 데스크톱의 경계를 식별함으로써 생성되는 시각적 표현(visual presentation)을 포함할 수 있다.

[0058] 이제 도 14를 참조하면, 도 14는 가상 데스크톱에서 생성된 프로그램의 사용자 인터페이스에 대한 픽셀 좌표를 수신하는 것, 및 미니-맵 내에 프로그램의 사용자 인터페이스에 대한 식별자를 생성하는 것을 나타내는 부가의 동작(1410)을 포함하는 도 13의 동작 절차를 나타낸다. 예를 들어, 본 발명의 일 실시예에서, 클라이언트(201)의 네트워크 어댑터는 프로그램의 사용자 인터페이스에 대한 좌표를 식별해주는 신호를 서버(204)로부터 수신하도록 구성될 수 있다. 클라이언트 서브시스템(306)은 그 신호를 수신하고, 클라이언트(201)의 디스플레이 드라이버에게 가상 데스크톱(302)의 어떤 부분에 사용자 인터페이스가 드로잉되었는지를 신속하게 판정하는 데 사용될 수 있는 식별자를 미니-맵(316) 상에 렌더링하라고 지시할 수 있다. 일례로, 서버(204)의 런타임 서브시스템(240)은 프로그램의 프로세스와 스레드를 생성하도록 구성될 수 있다. 이 예시적인 실시예에서, 원격 디스플레이 서브시스템(254)은 프로세스가 시작할 때의 통지와 프로그램에 대한 사용자 인터페이스의 좌표를 식별해주는 정보를, 예를 들어, GDI(246)로부터 수신하도록 구성될 수 있다. 원격 디스플레이 서브시스템(254)은 픽셀 범위, 또는 새롭게 생성된 사용자 인터페이스에 대한 픽셀 점을 식별해주는 메시지를 생성하고, 전송 논리(210)를 통해 클라이언트(201)로 신호를 보내도록 구성될 수 있다. 클라이언트 서브시스템(306)은 메시지를 수신하고 스케일링된 사용자 인터페이스 또는, 예를 들면, 스케일링된 사용자 인터페이스의 와이어프레임을 포함하는 미니-맵(316)을 뷰(304)의 일부분 상에 렌더링하라고 클라이언트(201)의 디스플레이 드라이버에게 지시하도록 구성될 수 있다. 클라이언트 서브시스템(306)은 이어서, 사용자 인터페이스가 가상 데스크톱(302) 상의 어디에 렌더링되었는지를 사용자가 빠르게 식별할 수 있게 해주기 위해서, 미리 정해진 시간 동안 스케일링된 사용자 인터페이스의 색상을 변경하도록 디스플레이 드라이버를 구성할 수 있다.

[0059] 이제 도 15를 참조하면, 도 15는 부가의 동작들(1512 및 1514) 및 세부 조정(1516)을 포함하는 도 12의 동작 절차를 나타낸 것이다. 도 12의 설명을 계속하면, 동작(1512)은 커서 이동 정보에 선형 가속 인자를 적용하는 것을 나타낸다. 예를 들어, 본 발명의 실시예에서, 스크롤 서브시스템(308)은 커서 이동 정보를 수신하고, 이를 스케일링 값과 곱하며, 스케일링된 값에 의해 정의된 양만큼 뷰(304)를 스크롤하라는 신호를 클라이언트 서브시스템(306)으로 전송하도록 구성될 수 있다. 예를 들어, 일 실시예에서, 뷰(304)를 상대적 마우스 이동의 양보다 많은 수의 픽셀만큼 패닝하는 것이 바람직할 수 있다. 예를 들어, 클라이언트(201)가 터치패드를 포함하는 예시적인 실시예들에서, 가상 데스크톱(302)의 한 쪽에서 다른 쪽으로 뷰(304)를 이동시키기 위해, 사용자는 터치패드를 가로질러 자신의 손가락을 터무니없이 여러 번 스크롤해야 할지도 모른다. 따라서, 일 실시예에서, 스크롤 서브시스템(308)은 상대적 이동 정보를 고정된 값과 곱하고 또 클라이언트 서브시스템(306)에게 뷰(304)를 스케

일링된 양만큼 이동하라고 지시할 수 있다. 특정의 일례에서, 스크롤 서브시스템(308)은 커서 위치가 뷰(304)의 오른쪽 경계를 정의하는 픽셀 위에 있는지를 판정하도록 구성될 수 있다. 이 일례에서, 사용자가 마우스를 x-방향으로 +1만큼 움직이면, 스크롤 서브시스템(308)은 50과 같은 고정된 숫자를 픽셀 이동에 곱하고, 뷰(304)를 x-방향으로 +50 픽셀만큼 움직이려는 메시지를 클라이언트 서브시스템(306)으로 보낼 수 있다.

[0060] 도 15의 설명을 계속하면, 도 15는, 그에 부가하여, 커서의 속도를 구하는 것, 동적 가속 인자를 생성하는 것 - 동적 가속 인자는 커서의 속도의 함수임 -, 및 커서 이동 정보에 동적 가속 인자를 적용하는 것을 나타내는 동작(1514)을 나타내고 있다. 예를 들어, 본 발명의 실시예에서, 스크롤 서브시스템(308)은 상대적 커서 이동 정보를 수신하고, 이를 동적 스케일링 값과 곱하며, 스케일링된 값에 의해 정의된 양만큼 뷰(304)를 스크롤하라는 신호를 클라이언트 서브시스템(306)으로 전송하도록 구성될 수 있다. 예를 들어, 일 실시예에서, 커서(310)가 뷰(304)를 가로질러 빠르게 움직이는 경우에 뷰(304)를 많은 수의 픽셀만큼 패닝하는 것이 바람직할 수 있다. 일례로, 사용자가 뷰(304)를 가로질러 커서(310)를 움직인다면, 사용자가 가상 데스크톱(302)의 먼쪽 끝에 있는 무엇인가를 조작하는 데 관심이 있는 것으로 추정될 수 있다. 이 일례에서, 스크롤 서브시스템(308)은 사용자의 의도가 뷰(304)를 많은 양의 픽셀만큼 패닝하는 것이라고 추정하고 그에 따라 반응하도록 구성될 수 있다. 이 상황에서, 스크롤 서브시스템(308)은 커서(310)가 뷰(304)를 가로질러 움직일 때의 커서(310)의 속도에 의존하는 스케일링 값을 생성하도록 구성될 수 있다. 예를 들어, 이 일례에서, 스크롤 서브시스템(308)은 마지막 64개의 샘플에 걸쳐 커서(310)의 좌표를 추적하도록 구성될 수 있다. 이 일례에서, 샘플링 레이트(sampling rate)는 일정할 수 있으며, 스크롤 서브시스템(308)은 커서(310)의 평균 속도를 구하도록 구성될 수 있다. 이 일례에서, 스크롤 서브시스템(308)은 커서(310)의 평균 속도에 기초하여 스케일링 값을 증가시키고, 클라이언트 서브시스템(306)에게 뷰(304)를 스케일링된 양만큼 패닝하라고 지시하도록 구성될 수 있다. 예를 들어, 특정의 예시적인 실시예에서, 스케일링 값이 속도에 따라 기하급수적으로 증가할 수 있다. 따라서, 커서(310)가 뷰(304)에서 빠르게 움직일수록, 클라이언트 서브시스템(306)은 뷰(304)를 더 많이 스크롤할 것이다. 특정의 실시예에서, 커서(310)는 뷰(304)를 가로질러 왼쪽에서 오른쪽으로 이동할 수 있다. 이 일례에서, 스크롤 서브시스템(308)은 커서(310)의 좌표가 x-방향으로 샘플당 +10만큼씩 변하고 있음을 나타내는 다수의 메시지를 수신할 수 있다. 이 실시예에서, 스크롤 서브시스템(308)은 e^{10} 을 적용하여 동적 가속 인자를 구하고, 클라이언트 서브시스템(306)에게 뷰(304)를 x-방향으로 22,026 픽셀만큼 이동시키라고 지시하도록 구성될 수 있다. 동일하거나 다른 실시예들에서, 스크롤 서브시스템(308)은 403과 같은 최대 동적 값을 설정하도록 구성될 수 있으며, 따라서 이 상황에서는 클라이언트 서브시스템(306)은 수천 픽셀을 스크롤하여 뷰(304)를 엄청나게 변화시키지 않을 수도 있다.

[0061] 도 15의 설명을 계속하면, 도 15는 그에 부가하여 도 12의 동작 절차의 미세 조정(1516)을 나타내고 있으며, 여기서 제1 픽셀 범위는 원격 컴퓨터의 제1 디스플레이 드라이버의 디스플레이 해상도와 제2 디스플레이 드라이버의 디스플레이 해상도에 대응하고, 또한 여기서 제1 디스플레이 드라이버의 디스플레이 해상도는 제1 경계들에 의해 정의되고, 제2 디스플레이 드라이버의 디스플레이 해상도는 제2 경계들에 의해 정의된다. 예를 들어, 본 발명의 실시예에서, 서버(204)는 적어도 2개의 디스플레이 드라이버를 포함하는 원격 디스플레이 서브시스템(254)을 포함할 수 있다. 이 일례에서, 각각의 디스플레이 드라이버는 그 자신의 디스플레이 해상도를 가질 수 있으며, 자신의 디스플레이 상의 이미지를 나타내는 신호를 전송 논리(210)을 통하여 클라이언트(201)로 전송하도록 구성될 수 있다. 이 예시적인 실시예에서, 제1 디스플레이 드라이버는 1280 x 1024 픽셀의 디스플레이 해상도로 동작하도록 구성될 수 있고, 제2 디스플레이 드라이버는 800 x 600 픽셀의 해상도로 동작할 수 있다. 이 예시적인 실시예에서, 도 3의 클라이언트 서브시스템(306)은 각각의 디스플레이에 대한 개별적인 데이터 스트림을 수신하고, 디스플레이 해상도들의 합을 포함하는 가상 데스크톱(302)을 생성하며, 가상 데스크톱(302)과 일치하도록, 예를 들어, 제2 디스플레이 드라이버의 픽셀 범위를 재매핑하도록 구성될 수 있다. 이 예시적인 실시예에서, 클라이언트 서브시스템(306)은 가상 데스크톱(302)에서 각각의 디스플레이에 대한 경계들을 구하기 위하여 제1 및 제2 디스플레이 해상도를 정의하는 픽셀 범위를 사용할 수 있다. 일례로, 클라이언트 서브시스템(306)은 각각의 디스플레이의 픽셀 범위를 가상 데스크톱(302) 전체의 픽셀 범위와 연관시키는 테이블을 포함할 수 있다. 제1 디스플레이 드라이버의 디스플레이 해상도가 1280 x 1024이고 제2 디스플레이 드라이버의 디스플레이 해상도가 800 x 600인 일 실시예에서, 디스플레이 드라이버의 물리적 레이아웃이 논리적으로 제1 디스플레이의 우측에 제2 디스플레이를 갖도록 설정되는 경우, 테이블은 제2 디스플레이 드라이버의 픽셀들을, 가상 데스크톱(302)의 x-축으로는 픽셀 번호 1280 내지 2080에, 그리고, 예를 들어, y-축으로는 제2 디스플레이 드라이버의 수직 정렬에 따라 0 내지 600 또는 424 내지 1024 픽셀에 매핑할 수 있다.

[0062] 이제 도 16을 참조하면, 도 16은 커서가 제1 디스플레이 드라이버의 디스플레이 해상도의 특정의 경계 - 특정의 경계는 제2 디스플레이 드라이버의 특정의 경계에 근접해 있음 - 에 도달했는지를 판정하는 것과, 뷰를 제2 디스플레이 드라이버에 대응하는 가상 데스크톱의 영역의 중앙에 위치시키는 것을 나타내는 부가의 동작(1618)을

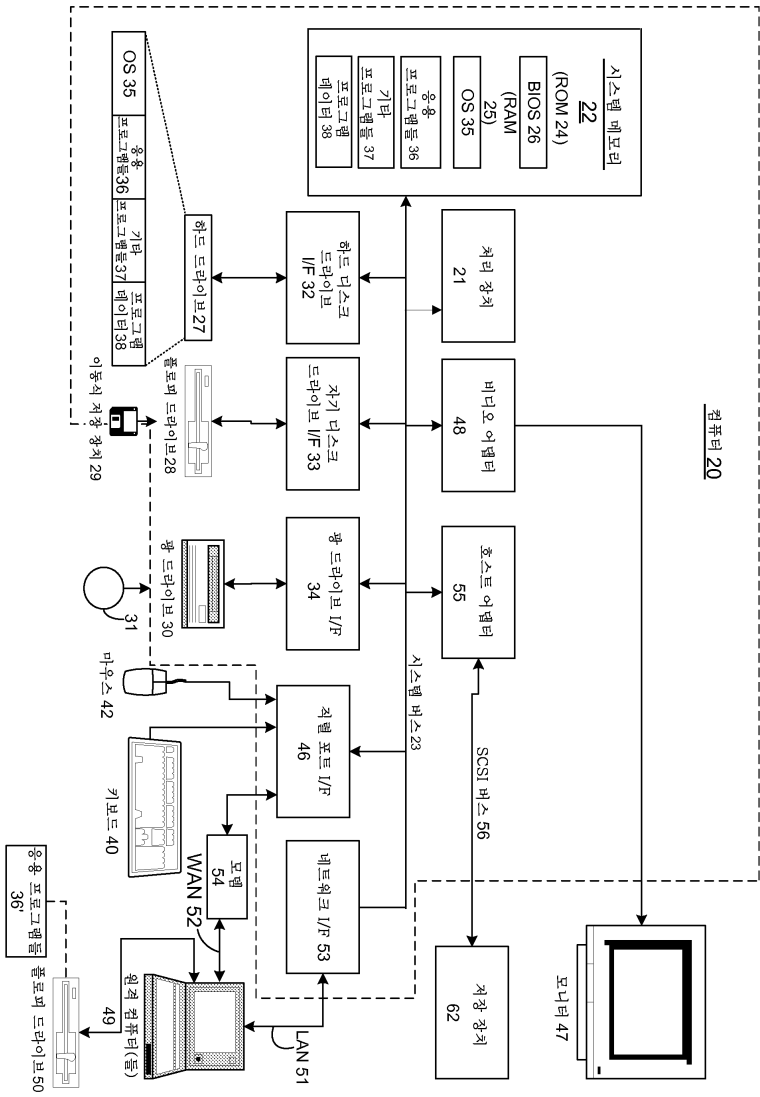
포함하는 도 15의 동작 절차의 대안의 실시예를 나타낸 것이다. 예를 들어, 본 발명의 실시예에서, 스크롤 서비스시스템(308)은 가상 데스크톱(302)의 매핑된 픽셀 범위와 관련된 각각의 디스플레이 드라이버의 픽셀 범위들을 포함하도록 구성될 수 있다. 이 예시적인 실시예에서, 스크롤 서비스시스템은 클라이언트(201)의 운영 체제(312)으로부터 커서 이동 정보를 수신하고, 가상 데스크톱(302)과 관련된 커서(310)의 좌표를 저장하도록 구성되어 있을 수 있다. 이 예시적인 실시예에서, 가상 데스크톱(302)과 관련된 커서(310)의 위치가 추적될 수 있으며, 커서(310)가 디스플레이 드라이버의 경계, 예를 들어, x축 방향으로 픽셀 1280에 도달하면, 스크롤 서비스시스템(308)은 디스플레이 드라이버에게 제2 디스플레이에 대응하는 가상 데스크톱 픽셀들 상에 중심이 오도록 픽셀 범위를 디스플레이하라고 지시하는 메시지를 클라이언트 서비스시스템(306)에게 보내도록 구성될 수 있다.

[0063] 전술한 상세한 설명은 여러 일례들 및/또는 동작 흐름도를 통하여 시스템 및/또는 프로세스들의 다양한 실시예를 기술하였다. 이러한 블록도들 및/또는 일례들이 하나 이상의 기능 및/또는 동작을 포함하는 한, 당업자라면 이러한 블록도들 또는 일례들 내에서의 각각의 기능 및/또는 동작이 광범위한 하드웨어, 소프트웨어, 펌웨어 또는 거의 모든 이들의 조합에 의해, 개별적으로 및/또는 공동으로, 구현될 수 있다는 것을 잘 알 것이다.

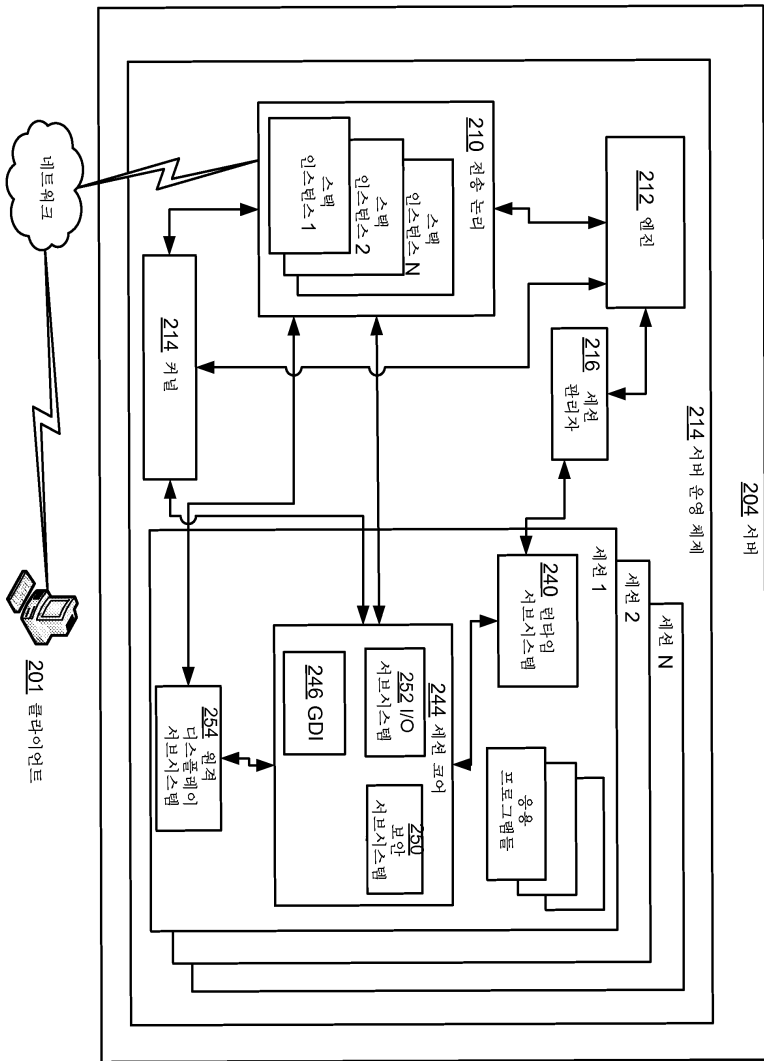
[0064] 본 명세서에 기술된 발명 대상의 특징의 측면들이 도시되고 설명되었지만, 본 명세서의 개시 내용에 기초하여, 본 명세서에 기술된 발명 대상 및 그의 광의의 측면들을 벗어나지 않고 변경들 및 수정들이 이루어질 수 있으며 따라서, 첨부된 특허청구범위가 본 명세서에 기술된 발명 대상의 진정한 사상 및 범위 내에 있는 이러한 모든 변경들 및 수정들을 본 발명의 범위 내에 포괄한다는 것이 당업자에게는 명백할 것이다.

도면

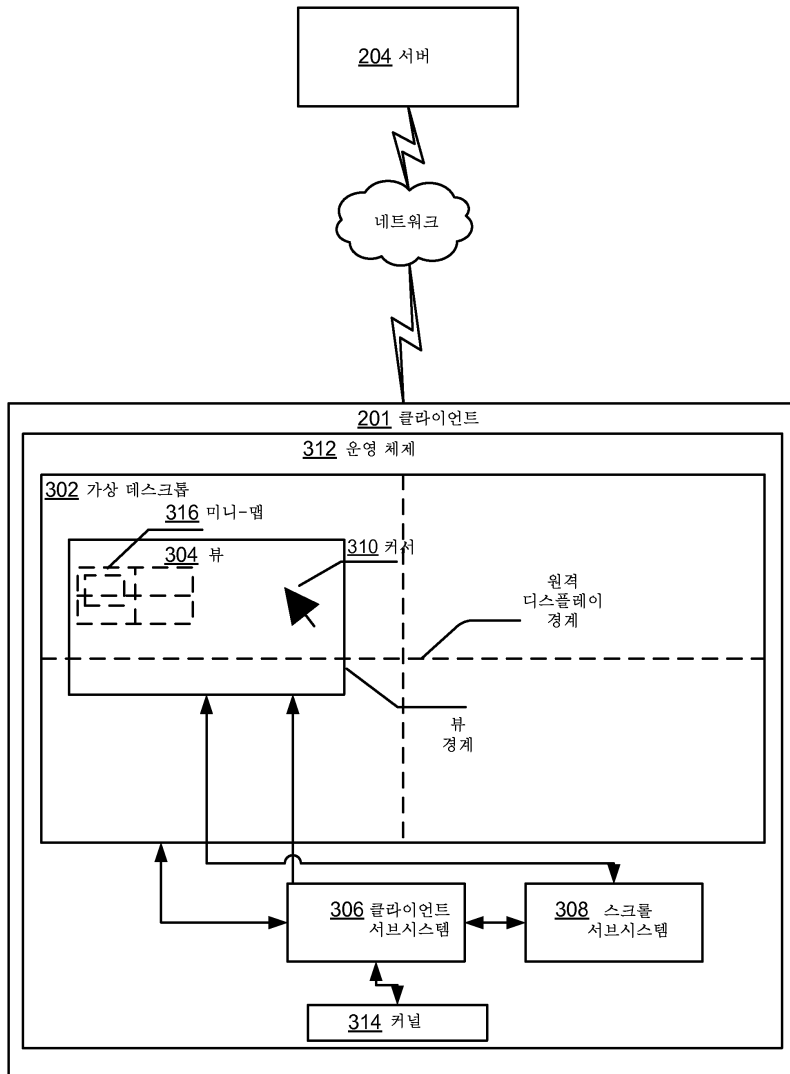
도면1



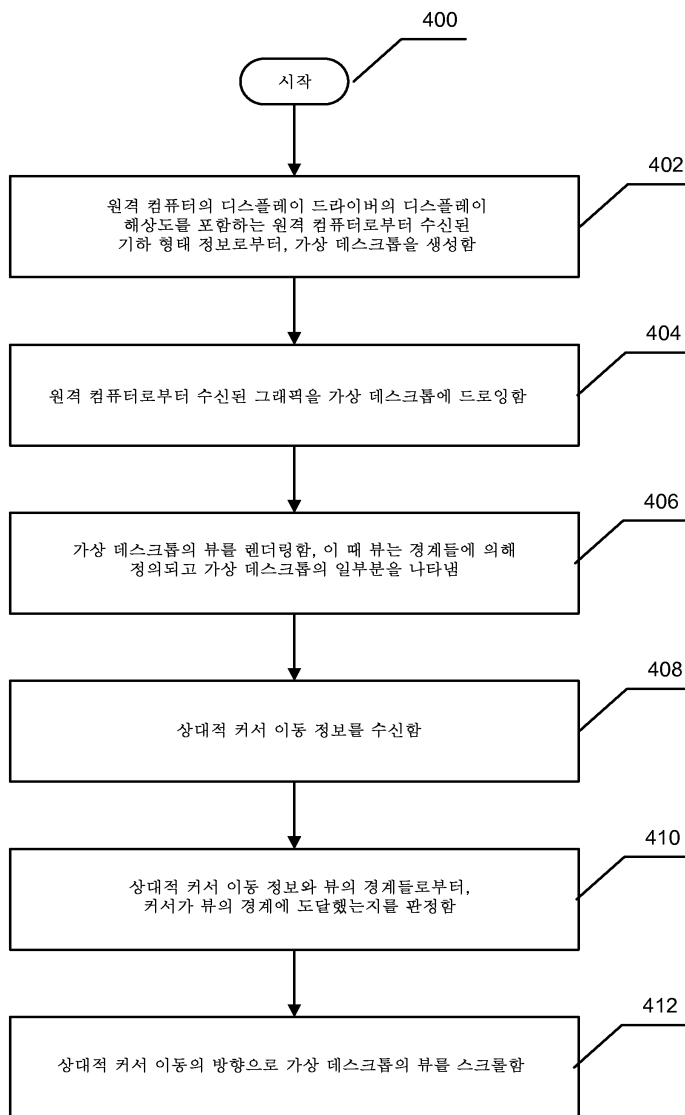
도면2



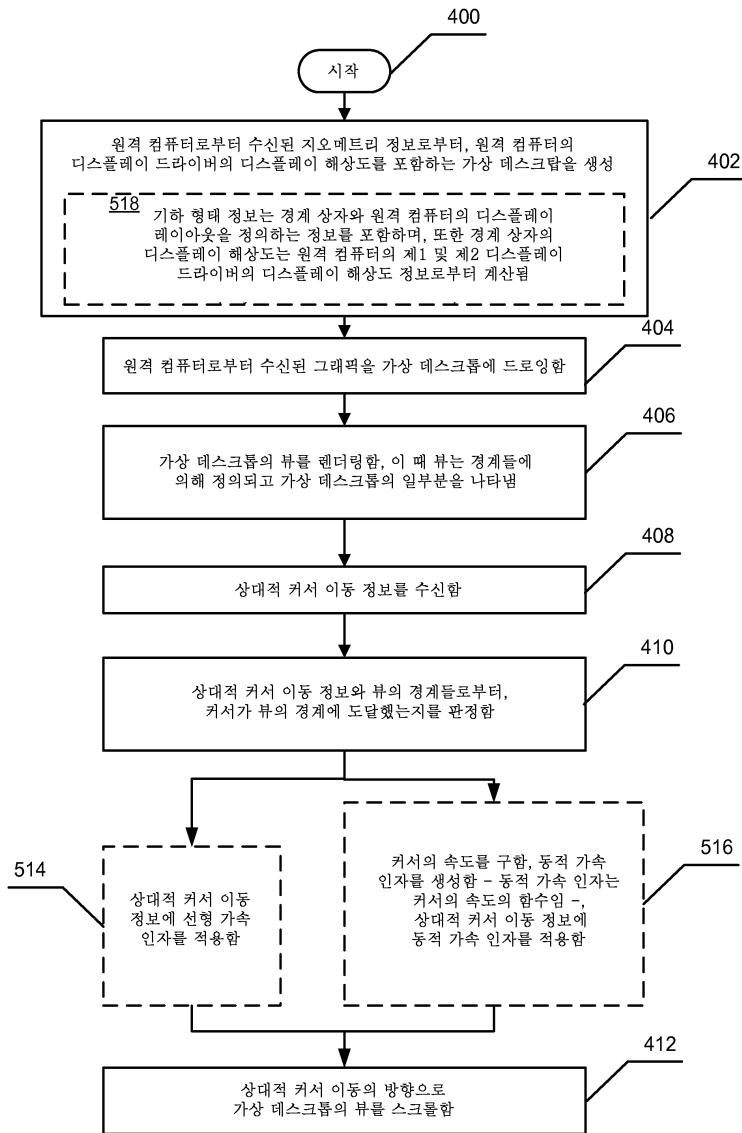
도면3



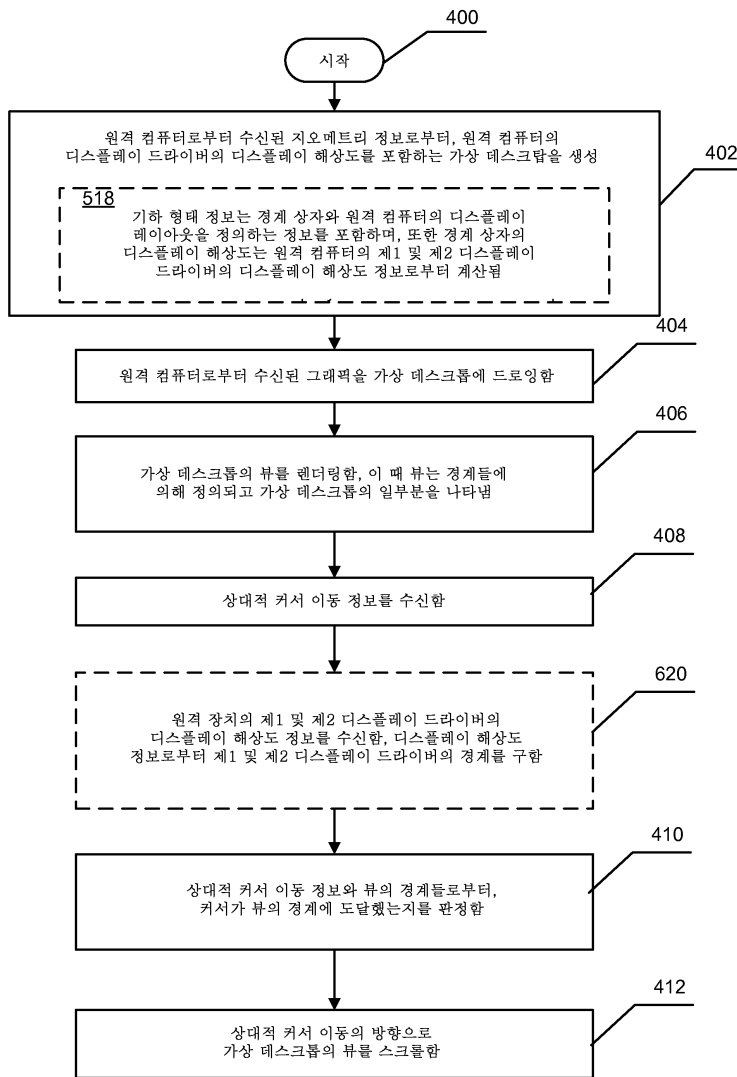
도면4



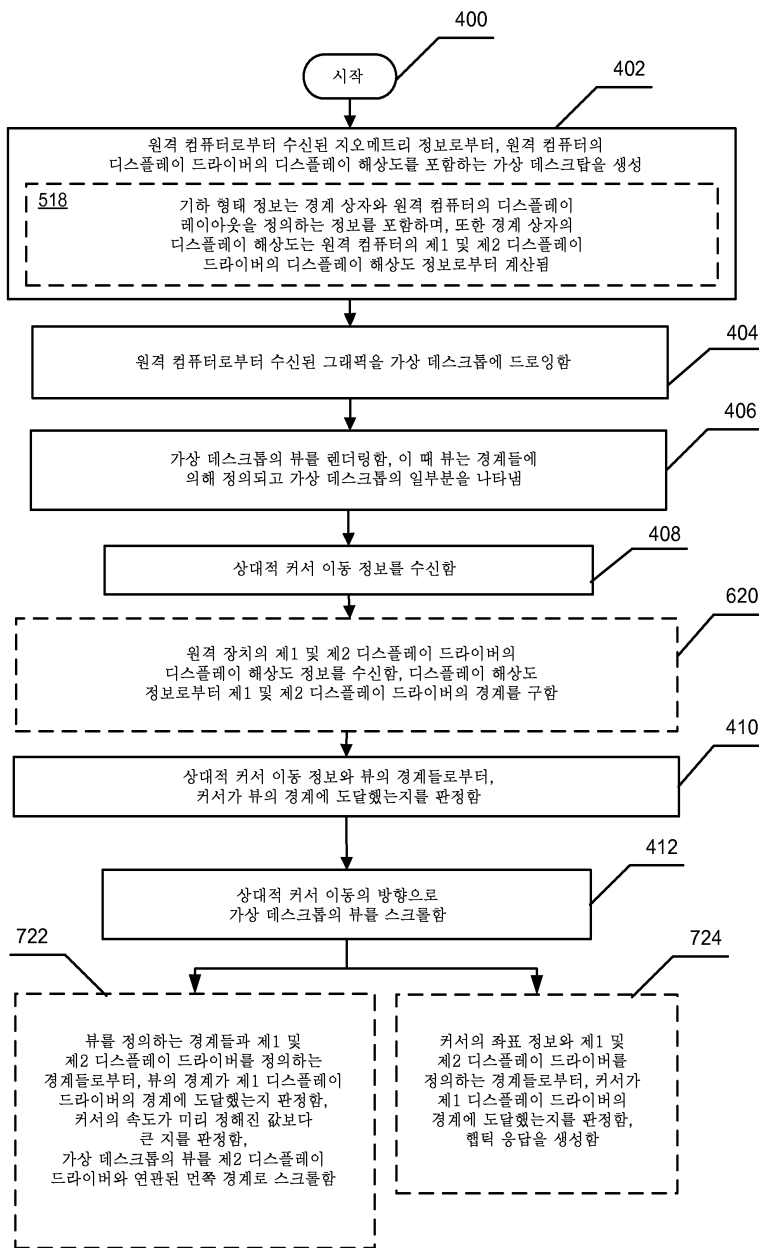
도면5



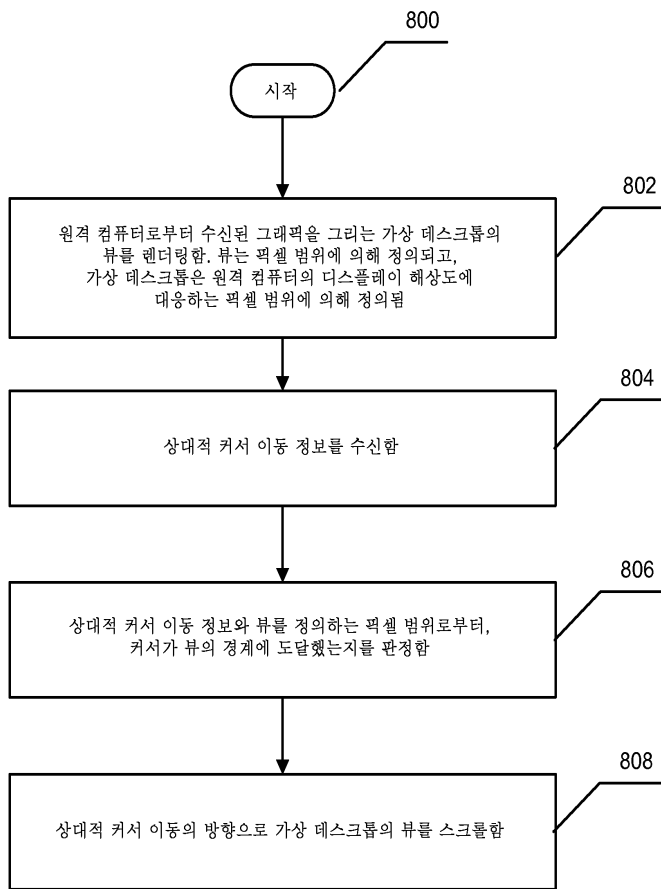
도면6



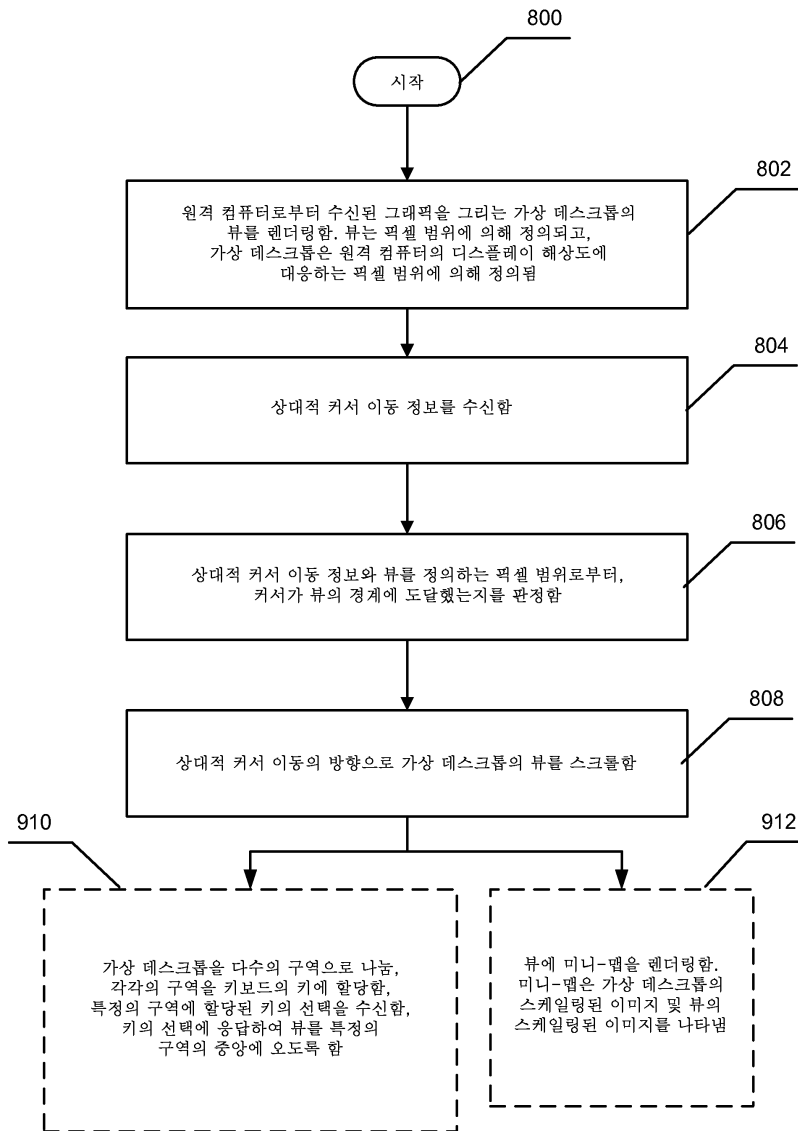
도면7



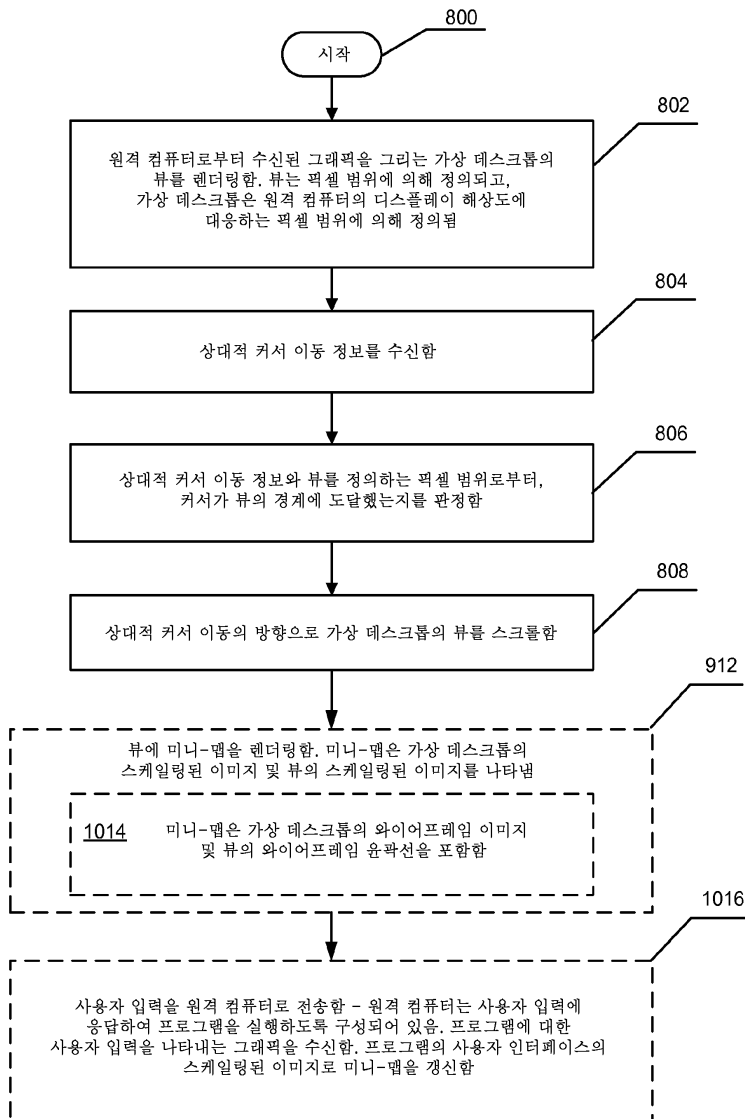
도면8



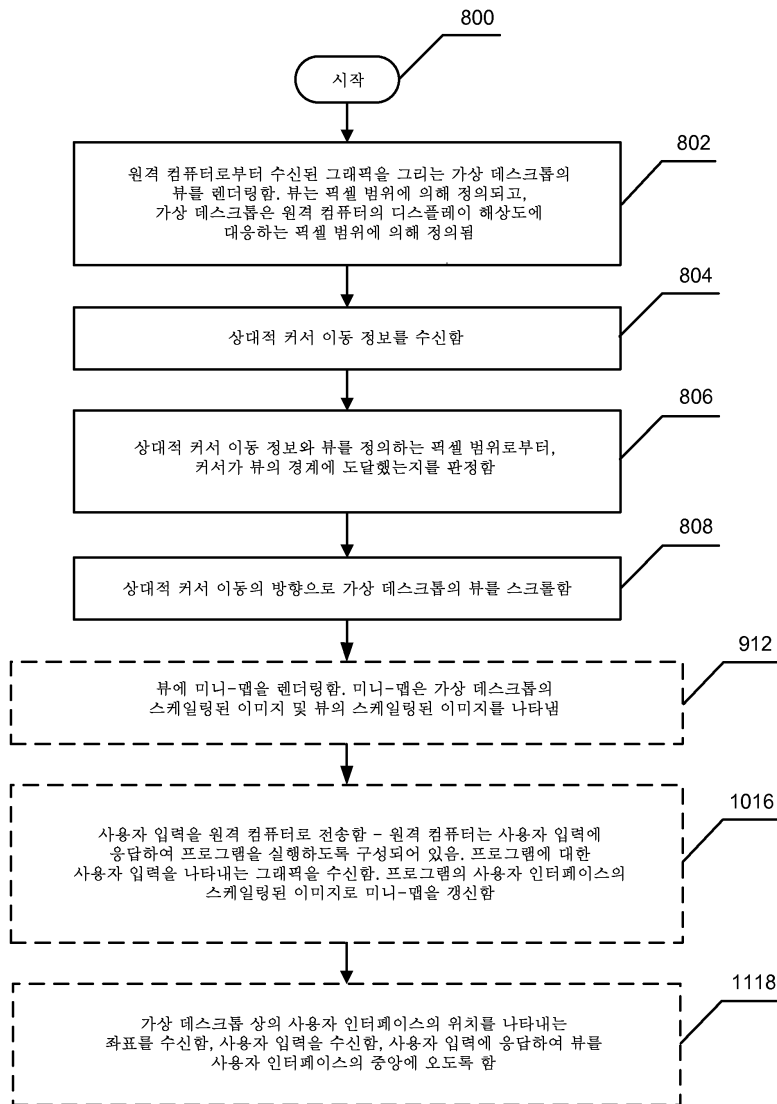
도면9



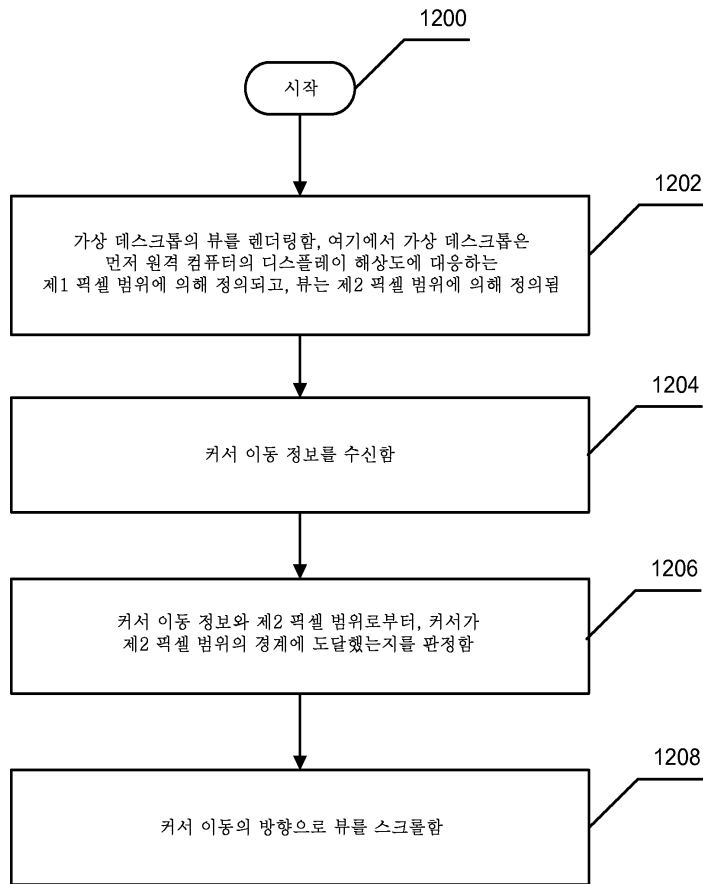
도면10



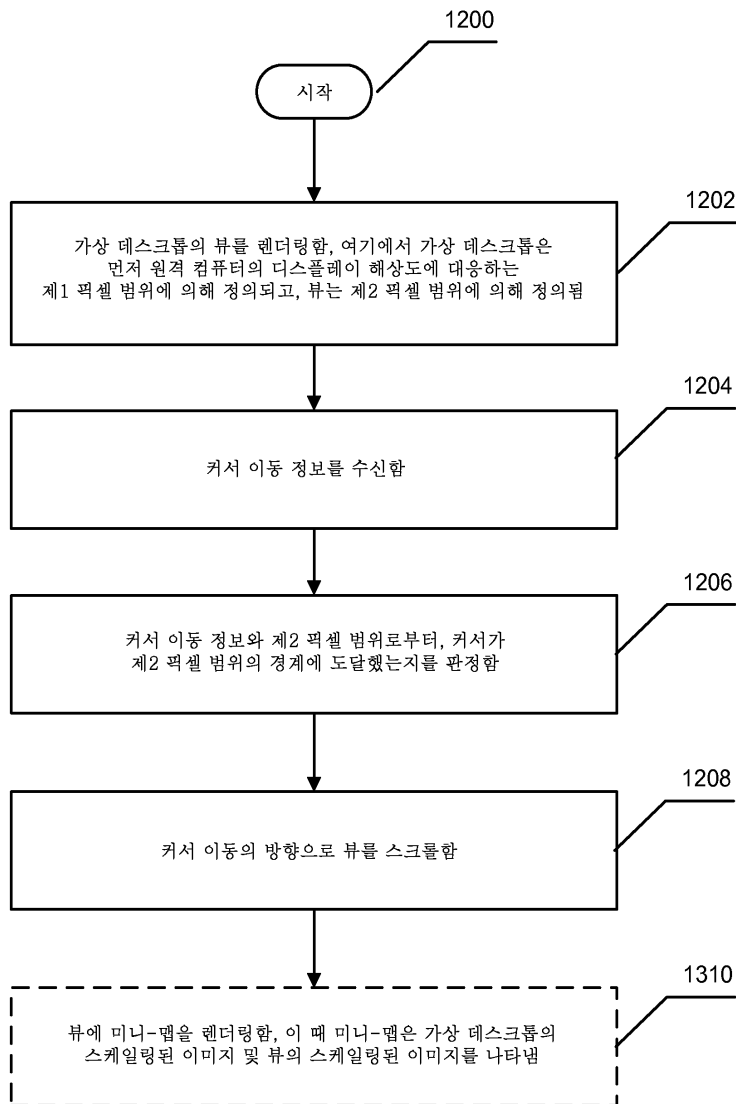
도면11



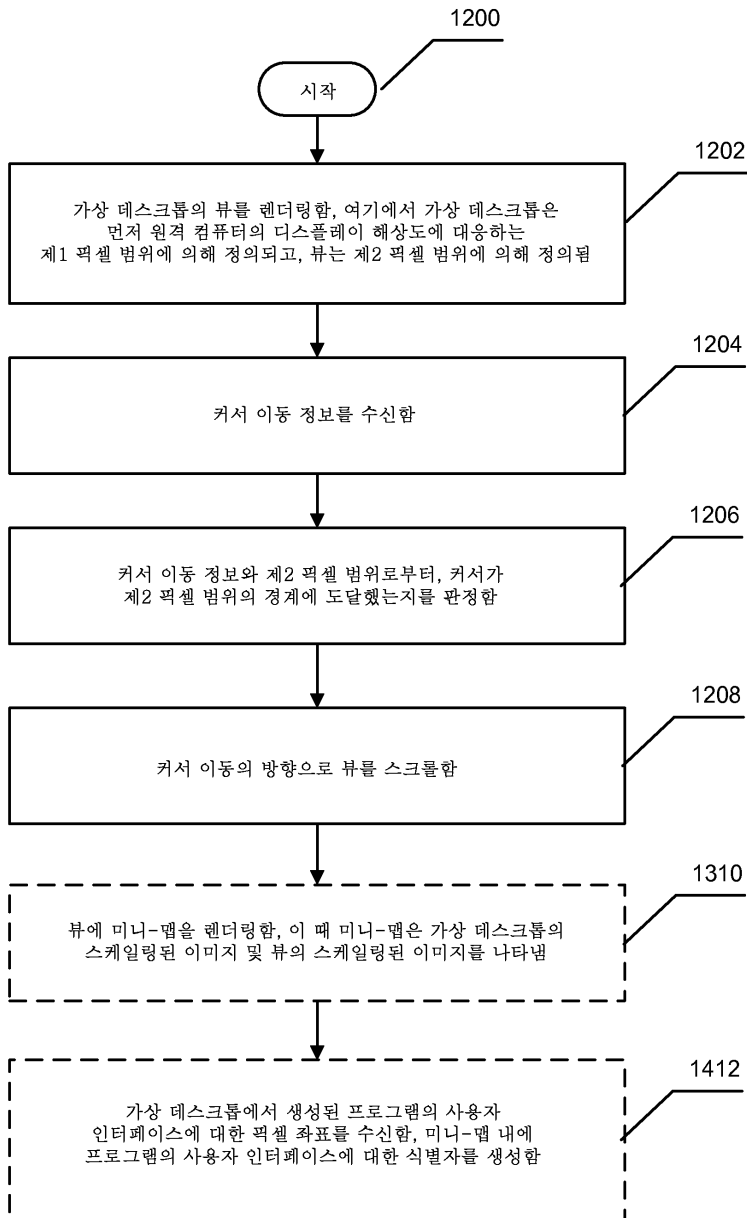
도면12



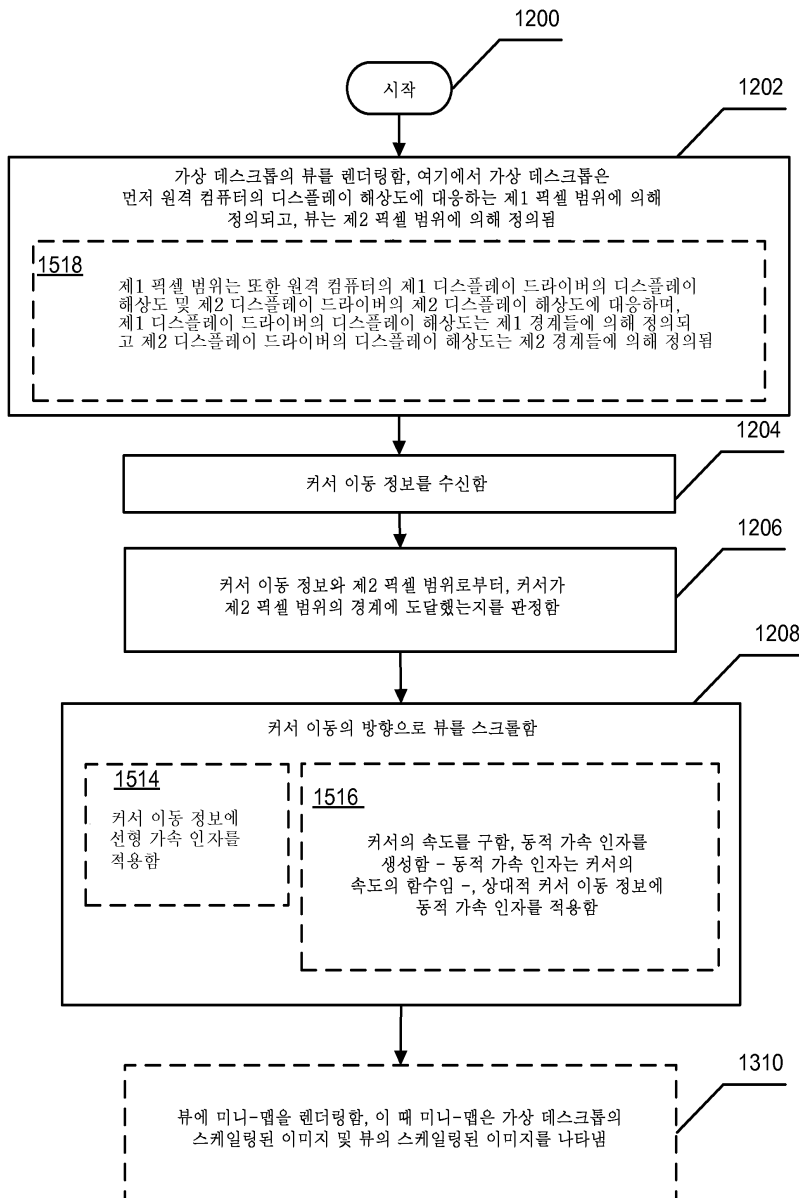
도면13



도면14



도면15



도면16

