(19)대한민국특허청(KR)
(12) 등록특허공보(B1)

(51) 。Int. Cl.
G06T 7/00 (2006.01)
G06T 5/40 (2006.01)

(45) 공고일자 2006년08월22일
(11) 등록번호 10-0613673
(24) 등록일자 2006년08월10일

(21) 출원번호 10-2005-0045384
(22) 출원일자 2005년05월30일

(65) 공개번호
(43) 공개일자

(73) 특허권자 안영하
강원도 춘천시 교동 113-106 203호

(72) 발명자 안영하
강원도 춘천시 교동 113-106 203호

김백섭
강원도 춘천시 퇴계동 그린타운아파트 103동 1404호

(74) 대리인 민혜정

심사관 : 이상헌

(54) 영상인식을 이용한 자동 에이전트시스템

요약

본 발명은 영상인식을 이용한 자동 에이전트시스템에 관한 것으로, 보다 상세히는 영상인식에 대한 지식을 바탕으로 모니터로 출력되는 영상의 패턴을 분석하여 그에 따른 적절한 반응을 자동적으로 일으키는 영상인식을 이용한 자동 에이전트 시스템에 관한 것이다.

본 발명의 영상인식을 이용한 에이전트시스템은, 타겟 프로그램을 구비하는 타겟부; 상기 타겟부로부터 수신된 타겟 어플리케이션을 분석하여 현재의 이벤트 상태를 인식하고, 그 인식된 결과와 도메인지식을 가지고 어떠한 작업을 처리할 것인지를 결정하는 에이전트부; 상기 에이전트부로부터 호출되어, 상기 에이전트부로 부터 수신된 데이터를 이진화를 하며, 이진화된 관심영역의 문자열에서 각 문자를 분리하며 또한 사물을 배경으로부터 분리하고, 분리된 문자열의 해당문자를 인식하며 분리된 사물의 해당사물을 인식하는 인식기부;를 구비하는 영상인식을 이용한 에이전트 시스템에 있어서, 상기 에이전트부는, 상기 타겟부로 부터 입력되는 타겟 프로그램의 이벤트 상태를 에이전트간 통신언어(ICL)로 해석하는 ICL부; 상기 ICL부로 부터 타겟 프로그램의 이벤트 상태를 ICL로 해석한 결과를 수신하여 인식하고자 하는 타겟 프로그램의 이벤트를 사용자에 의해 설정된 키워드에 해당하는 영상 및 문자를 인식하여 매치된 트리거의 액션을 보내는 이벤트 관리자부; 상기 이벤트 관리자부에 의해 인식해야 할 내용이 정해지면 필요한 영역만큼만 캡쳐(Capture)하여 상기 인식기부로 전송하는 캡쳐어부; 상기 캡쳐어부에서 캡쳐하여 상기 인식기부로 보낸 것에 대한 응답을 상기 인식기부로부터 수신하고, 상기 이벤트 관리자부의 판단에 따라 타겟부로 출력하는 입력 관리자부;를 구비하는 것을 특징으로 한다.

또한 상기 인식기부는, 상기 에이전트로부터 수신된 데이터로부터 관심영역을 결정하며, 결정된 관심영역을 중요도에 따라 순위를 매기는 ROI부; 상기 ROI부로 부터 관심영역내의 데이터를 임계치(Thresholding value)를 이용하여 이진화하는 임계치(Thresholding)부; 상기 임계치부로부터 수신된 상기 이진화된 관심영역의 문자열에서 각 문자를 분리하는 문자열 분리부와, 상기 임계치부로부터 수신된 이진화된 관심영역에서 사물을 배경으로부터 분리하는 사물 배경 분리부를 구비하

는 분리(Segmentation)부; 상기 분리부에서 분리된 문자열 데이터를 수신하여 해당문자를 인식하는 문자인식부와, 상기 분리부에서 분리된 사물 데이터를 수신하여 해당사물을 인식하는 사물인식부를 구비하는 인식부;를 구비하는 것을 특징으로 한다.

대표도

도 2

색인어

자동, 에이전트, 영상, 인식, 조정에이전트, 응용에이전트, 입력관리부, WDM 디바이스 드라이버

명세서

도면의 간단한 설명

도 1은 본 발명의 바람직한 일 실시예에 의한 영상인식을 이용한 자동 에이전트 시스템의 구성도이다.

도 2는 일반적인 OAA(Open Agent Architecture) 시스템를 설명하기위한 설명도이다.

도 3은 본 발명의 ICL의 구조를 설명하기위한 구조도이다.

도 4는 타겟 프로그램이 온라인게임인 경우 상황관리자의 상황 분류의 일예이다.

도 5는 이벤트 에디터 인터페이스의 일 예이다.

도 6은 도 1의 입력 관리자부와 WDM 디바이스 드라이버의 구성을 설명하기위한 구성도이다.

도 7은 다층 히스토그램 임계치(Multi-Layer Histogram Thresholding)를 이용한 사물 배경 분리의 일예이다.

도 8은 본 발명의 일실시예에 의한 이미지 트리 생성 순서도이다.

도 9a는 본 발명의 루트 노드의 일실시예를 설명하기위한 '가'의 이미지이다.

도 9b는 도 9a의 '가'의 이미지로부터 폰트 이미지 트리생성을 설명하기위한 폰트 이미지 트리구조이다.

도 10은 도 1의 문자인식부에서 문자 인식 순서도의 일예이다.

도 11은 도 1의 입력관리자부에서 문자 입력 순서도의 일예이다.

발명의 상세한 설명

발명의 목적

발명이 속하는 기술 및 그 분야의 종래기술

본 발명은 영상인식을 이용한 자동 에이전트시스템에 관한 것으로, 보다 상세히는 영상인식에 대한 지식을 바탕으로 모니터로 출력되는 영상의 패턴을 분석하여 그에 따른 적절한 반응을 자동적으로 일으키는 영상인식을 이용한 자동 에이전트 시스템에 관한 것이다.

에이전트 시스템은 자발성, 자율성, 사회성, 반응성을 갖는 독립된 프로그램의 조합으로 사용자에게 편리하고 자연스러운 메타포(Metaphor)를 제공한다. 또한 에이전트 시스템은 사용자가 반드시 처리해야 할 일 중 대기 시간이 많이 걸리는 일,

정보가 많이 흩어져 있는 곳에서 필요한 정보를 수집하는 일, 호텔 예약이나 교통편 예약과 같이 한정된 정보를 이용하여 정해진 작업을 수행하는 일과 같이, 부가가치가 적으면서 많은 노력(cost)을 요하는 작업을 대행해 준다. 따라서 사용자는 부가가치가 높은 일에 좀 더 많은 시간을 보낼 수 있게 된다.

에이전트 시스템은 사용자와 같은 수준의 판단을 하여 사용자가 직접 하지 않아도 미리 부여된 설정에 따른 작업을 수행할 수 있도록 만들어져야 한다.

최근 반도체 기술의 발달, 고성능의 개인용 컴퓨터 보급과 인터넷의 급속한 확산으로 검색 엔진이 등장 하였고 이러한 서비스를 사용자들이 쉽게 접근할 수 있도록 하는 에이전트 시스템이 기능과 역할에 따라 발전하게 되었다.

에이전트 시스템에는 사용자가 수행하는 행동을 관찰하고 어떤 내용에 관심을 가지고 있는지를 판단하여 사용자에게 알맞은 내용을 전달하도록 하는 학습 에이전트 시스템, 사용자의 원하는 작업을 찾아내서 이들을 네트워크나 응용 프로그램 안 어디에서든지 실행할 수 있도록 이동시켜주는 인터페이스 에이전트 시스템, PC나 워크스테이션의 운영체제에 상주하면서 국부적으로 실행되는 데스크탑 에이전트 시스템, 서버에 상주하면서 사용자와 직접적인 상호작용 없이 사용자를 대신해서 작업을 수행하도록 인터넷상에서 분산된 온라인 정보를 접근하는 인터넷 에이전트 시스템, 클라이언트 컴퓨터로부터 원격 실행을 위해 다양한 서버들로 자기 자신을 이동시킬 수 있는 모빌 에이전트 시스템, 사용자를 대신해서 쇼핑을 가서 제품 사양을 얻어 오며 가격을 비교해주는 전자상거래 에이전트 시스템으로 분류할 수 있다.

기존 에이전트 시스템의 한계는 첫째, 사용 환경이 특정 애플리케이션에 종속적(dependent)이므로 호환성이 낮고, 둘째, 에이전트 시스템이 클라이언트 시스템에 있지 않고 서버에 상주하는 경우가 대부분이므로 보안과 정보의 신뢰성 문제를 내포하고 있으며, 셋째, 에이전트 시스템을 사용자가 만드는 것이 아니라 업체가 제공하므로 그 컨텐츠와 기능이 한정적이다.

본 발명의 대상은 특정 애플리케이션에 종속된 에이전트 시스템이 아니라 사용자의 컴퓨터를 통해 들어오는 가상환경의 모든 영상을 인식하는 영상인식 에이전트 시스템에 관한 것이다.

본 발명의 영상인식 에이전트 시스템은 영상인식에 대한 지식을 바탕으로 모니터로 출력되는 영상의 패턴을 분석하여 그에 따른 적절한 반응을 일으키도록 구성되며, 또한 본 발명의 영상인식 에이전트 시스템은 영상인식을 통해 상황에 따라 설정된 전략을 수행한다. 즉, 본 발명은 화면상에 디스플레이되는 중요한 정보를 판단해 사용자와 비슷한 수준으로 끌어올리며 신속 정확하게 작업을 수행하는 영상인식 에이전트 시스템을 제공한다.

본 발명은 사용자가 원하는 영상 또는 텍스트를 키워드로 설정하고, 이에 대한 일종의 행동을 부여하면 영상에 따른 패턴을 비교 분석하여 부여된 행동(Action)에 따라 매크로 반응을 일으키는 영상인식 에이전트 시스템을 제공한다.

발명이 이루고자 하는 기술적 과제

본 발명이 이루고자 하는 기술적 과제는 사용자의 컴퓨터를 통해 들어오는 가상환경의 모든 영상을 인식하는 영상인식 에이전트 시스템을 제공하는 데 있다.

본 발명이 이루고자 하는 또 다른 기술적 과제는 영상인식에 대한 지식을 바탕으로 모니터로 출력되는 영상의 패턴을 분석하여 그에 따른 적절한 반응을 일으키도록 구성되는 영상인식 에이전트 시스템을 제공하는 데 있다.

본 발명이 이루고자 하는 또 다른 기술적 과제는 영상인식을 통해 상황에 따라 설정된 전략을 수행하는 영상인식 에이전트 시스템을 제공하는 데 있다.

본 발명이 이루고자 하는 또 다른 기술적 과제는 디스플레이되는 중요한 정보를 판단해 사용자와 비슷한 수준으로 끌어올리며 신속 정확하게 작업을 수행하는 영상인식 에이전트 시스템을 제공하는데 있다.

발명의 구성 및 작용

이하 본 발명의 일 실시예에 의한 영상인식 에이전트 시스템의 구성 및 동작을 첨부한 도면을 참조하여 상세히 설명한다.

도 1은 본 발명의 바람직한 일 실시예에 의한 영상인식을 이용한 자동 에이전트 시스템의 구성도로, 타겟부(100), 에이전트(Agent)부(200), 인식기(Classifier)부(300), 테스트부(400), 데이터베이스(500)를 구비한다.

타겟부(100)는 에이전트부(200)에 입력되는 타겟 프로그램, 즉 타겟 어플리케이션을 구비하는 시스템이다.

에이전트부(200)는 타겟부(100)로부터 수신된 타겟 어플리케이션을 분석하여 현재의 이벤트 상태를 인식하고, 그 인식된 결과와 도메인지식을 가지고 어떠한 작업을 처리할 것인지를 결정하며 이에맞는 인식기부(300)를 호출한다. 에이전트부(200)는 에이전트간 교류언어(Interagent Communication Language)(이하 ICL이라 한다)부(210), 이벤트 관리자(Event Manager)부(220), 캡쳐어(Capture)부(230), 입력 관리자(Input Manager)부(240)로 이루어진다. 본 발명에서 에이전트 기반구조로는 도 2와 같은 개방 에이전트 구조(Open Agent Architecture)(이하 OAA라 한다)를 채택하였다. 일반적인 OAA 시스템에서 에이전트부(200)는 조정에이전트와, 다수의 응용에이전트를 구비하며, 조정에이전트는 응용에이전트들의 상위에 존재하여 응용에이전트를 총괄한다. 응용에이전트는 특정한 영역의 작업을 전문적으로 처리하는 역할을 하며, 사용자가 지시한 스크립트를 가지고 해당 도메인지식의 학습을 위한 독립적인 프로세스이다.

ICL부(210)는 타겟부(100)로 부터 입력되는 타겟 프로그램의 이벤트 상태를 에이전트간 통신언어(ICL)로 해석한다. 본 발명의 ICL은 도 3과 같이, 콘텐트(content)층, 프로토콜(protocol)층, 랩퍼(wrapper)층으로 구성된 3개의 층으로 이루어진다.

이벤트 관리자부(220)는 ICL부(210)로 부터 타겟 프로그램의 이벤트 상태를 ICL로 해석한 결과를 수신하여 인식하고자 하는 타겟 프로그램의 이벤트를 사용자에 의해 설정된 키워드에 해당하는 영상 및 문자를 인식하여 매치된 트리거의 액션을 출력한다. 이벤트 관리자부(220)는 상황관리자(223)와 사건관리자(227)를 구비한다. 본 발명에서는 에이전트 기반구조로 OAA를 채택하였으며, OAA 시스템에 연결된 모든 에이전트는 ICL로 표현된 기능사양들을 정의하여 조정 에이전트(상황관리자)에게 공지해야만 한다. 이것은 조정 에이전트(상황관리자)에 의해 에이전트와 통신할 때 사용되고, 지시된 작업을 수행하기 위해 어느 에이전트(사건관리자)에게 어떤 작업을 맡길 것인가를 판단하고 적절한 응용 에이전트(사건관리자)를 선택하여 작업을 요구할 때 사용한다.

상황관리자(223)는 조정에이전트에 해당한다. 상기 ICL부(210)에서 타겟부(100)로 부터 입력되는 타겟 프로그램의 이벤트 상태를 ICL로 해석한 결과를 수신하여, 지시된 작업을 수행하기 위해 어느 사건관리자(227)에게 어떤 작업을 맡길 것인가를 판단하고 적절한 사건관리자(227)를 선택하여 작업을 요구한다. 즉, 상황관리자(223)는 타겟 프로그램으로부터 현재 상황을 인식하고 분석된 결과를 토대로 가장 효율적인 일 처리를 위한 판단을 하는 상위의 총괄 관리를 담당한다. 예를 들면, 도 4에서와 같이, 타겟 프로그램이 온라인게임인 경우에, 쉬는중, 적을 탐색 중, 전투 중, 전리품 획득 중, 시야 회전 중, 길 이동 중, 치료 중의 상황으로 분류하여 인식대상을 가공할 수 있다.

사건 관리자(227)는 도메인지식의 학습을 위한 응용에이전트에 해당한다. 사건관리자(227)는 상황관리자(223)로 부터 요구된 작업인 사용자에 의해 설정된 키워드에 해당하는 영상 및 문자, 즉, 인식해야할 내용을 정한다. 사건관리자(227)는 인식하고자 하는 타겟 프로그램의 이벤트를 사용자에 의해 설정된 키워드에 해당하는 영상 및 문자를 인식하여 매치된 트리거의 액션을 보낸다. 예를들어 도 5에서와 같이, 이벤트 에디터 인터페이스처럼 사건관리자(227)의 각종 설정은 사용자가 직접 입력할 수 있는 에디터 툴로서 제공할 수 있다.

캡쳐어부(230)는 이벤트 관리자부(220)에 의해 인식해야 할 내용이 정해지면 필요한 영역만큼만 캡쳐하여 인식기부(300)로 전송한다. 따라서 캡쳐어부(230)는 빠르고 효율적이며, 불필요한 캡쳐(capture)로 인한 오버헤드를 줄일 수 있다. 캡쳐 방식으로는 전체화면 디바이스 콘텍스트(Device Context)(이하 DC라 한다) 또는 응용프로그램의 핸들 값에 해당하는 DC를 얻어와 메모리 DC에서 DIB(Device Independent Bitmap) 형식으로 저장하는 방법과 그래픽 출력 하드웨어의 메모리를 후킹(hooking)하여 직접적(direct)으로 RGB칼라를 가져오는 방법을 사용할 수 있다.

입력 관리자부(240)는, 상황관리자(223)에서 인식하고자 하는 대상을 캡쳐어부(230)에서 캡쳐(Capture)하여 인식기부(300)로 보낸 것에 대한 응답을 수신하고, 이를 사건관리자(227)의 판단에 따라 타겟부(100)로 출력한다. 본 발명에서 입력방식은 윈도우즈 NT의 계층적 디바이스 구조를 가진 WDM(Windows Driver Model) 디바이스 드라이버를 사용한다. 일반적으로 입력 방식은 가상환경 플랫폼에서 제공하는 응용 프로그래밍 인터페이스 (Application Programming Interface)(이하 API라 한다)를 통해 타겟 윈도우에 입력 메시지를 보내거나 키보드와 마우스의 인터럽트 함수를 호출하는 방식이 있으나, 이 방법은 후킹의 대상이 될 수 있으므로, 본 발명에서는 디바이스 드라이버를 통해 입력 하드웨어에 직접 접근하는 방식을 사용한다. 즉, 본 발명은 윈도우즈 NT의 계층적 디바이스 구조를 가진 WDM 디바이스 드라이버를 제작하여 윈도우 드라이버 형태로 입력 관리자부(240)를 구성한다. 본 발명의 윈도우 디바이스 드라이버(WDM)의 구조는 도 6과 같다. 도 6에서 입력 관리자부(240)는, 인식부(340)으로 부터 문자와 사물에 대한 인식 데이터를 수신하고, 이를 프로토콜에 따른 사건관리자(227)의 판단에 따라 응답, 즉 I/O 요청 패킷(IRP)를, WDM 디바이스 드라이버로 전송한다.

WDM 디바이스 드라이버(243)는 입력관리자부(240)에서 출력된 IRP를 수신하여 타겟부(100)에 맞는 데이터형태로 바꾸고 이를 하드웨어 추상화 계층(HAL)를 통해 타겟부(100)로 전송한다. 하드웨어 추상화 계층(HAL)은 WDM 디바이스 드라이버(243)으로 부터 수신된 IRP를 타겟부(100)의 마우스, 키보드, HID(Human Interface Device) 등에 전달한다.

인식기부(300)는 주기적으로 관심영역을 중요도에 따라 순위를 매기며, 관심영역의 임계치를 결정하고 이를 이용하여 이진화를 하며, 이진화된 관심영역의 문자열에서 각 문자를 분리하며 또한 사물을 배경으로부터 분리하고, 분리된 문자열의 해당문자를 인식하며 분리된 사물의 해당사물을 인식하고 그 결과를 에이전트부(200)의 입력관리부로 전송한다. 인식기부(300)는 관심영역(Region Of Interest)(이하 ROI라 한다)부(310), 임계치(Thresholding)부(320), 분리(Segmentation)부(330), 인식부(340)를 구비한다.

ROI부(310)는 관심영역을 결정하며, 결정된 관심영역을 중요도에 따라 순위를 매긴다. ROI부(310)는 이를 특정시간간격으로 주기적으로 행한다. ROI부(310)에서 관심 영역은 사용자에게 정보를 제공해주기 위한 윈도우 또는 다이얼로그 인터페이스에 의해 결정된다. ROI부(310)에서 관심영역은 에이전트에 의해 결정되는 중요도에 따라 우선순위를 매겨지며, ROI부(310)는 특정 시간 간격으로 변화를 감시한다. 수행속도를 높이기 위한 방법으로 관심영역 특정 수직선(vertical line)을 정하여 3 픽셀(pixel) 간격으로 이전 프레임과 비교한다. 이 때 RGB 각각의 값을 한 바이트에 저장함으로써 대입과 비교 연산을 빠르게 수행할 수 있다. 이러한 FRS(Fast ROI Search) 관심영역 필터링을 통해 인식 대상이 되는 사물 또는 문자를 인식기의 임계치부(320)로 보낸다.

임계치(Thresholding)부(320)는 임계치(Thresholding value)를 이용하여 이진화한다. 일반적으로, 이진화 방법은 두가지가 있으며, 그중 제1방법은 해당 윈도우 또는 다이얼로그 인터페이스에 사용자가 미리 설정한 임계치를 비교하는 방법이며, 제2방법은 반복적인 임계치선택(Iterative Threshold Selection)(이하 ITS라 한다) 알고리즘, 2피크 임계치(Two Peak Thresholding) 알고리즘, 이중 임계치(Double Thresholding)알고리즘을 사용하여 실시간으로 임계치를 정하는 방법이다. 본 발명에서는 설정된 임계치가 있으면 제1방법을 취하고, 설정된 임계치가 없을 경우 제2방법을 취한다.

미리 설정한 임계치 비교방법은 영상의 R+ G+ B를 더한 회색 값(gray value)을 임계값으로 하거나 R, G, B를 3차원 벡터로 임계값으로 이진화하는 방법이다.

ITS 알고리즘은 먼저 이미지의 명도(intensity)의 평균값을 초기 임계값 T로 정한 후, T를 기준으로 R1과 R2 그룹으로 나눈다. R1과 R2의 평균 회색 값(gray value) μ1, μ2을 계산하여 T를 (μ1+ μ2)/2로 재설정한다. 지금까지의 과정을 μ1과 μ2의 값이 변하지 않을 때까지 반복한다.

2피크 임계치(Two Peak Thresholding) 알고리즘은 값이 가장 큰 첫 번째 피크인 제1피크(FP)를 찾는다. 나머지 피크(Peak)에서 제1피크(FP)와의 거리와 그 피크(Peak)의 크기 값이 가장 큰 피크를 두 번째 피크인 제2피크(SP)로 설정한다. 제1피크(FP)와 제2피크(SP) 사이의 밸리(valley) 중에서 가장 작은 밸리(valley) 값을 임계값 T로 정하여 이진화한다.

이중 임계치 알고리즘은 먼저 두 개의 임계값(thresholds) T1과 T2를 정한다. T1 이하의 회색 값(gray value)을 가지는 모든 픽셀들(R1)과 T1, T2 사이의 값을 가지는 모든 픽셀들(R2)과 T2이상의 값을 가지는 모든 픽셀들(R3)에 대한 세 영역으로 나눈다. R2에 할당된 각 픽셀을 방문하면서 R1에 근접해 있는 픽셀을 R1 영역으로 재할당한다. 지금까지의 과정을 재할당할 수 있는 픽셀이 없을 때까지 반복한다. R2 영역에 남아있는 픽셀은 R3 영역으로 이동시킨다. R1 영역의 픽셀들과 R3 영역의 픽셀들로 이진화된다.

분리(Segmentation)부(330)는 임계치부(320)으로부터 이진화된 관심영역을 수신하여 문자열에서 각 문자를 분리하며, 또한 이진화된 관심영역에서 사물을 배경으로부터 분리한다. 분리부(330)는 문자열 분리부(333)와 사물 배경 분리부(337)를 구비한다.

문자열 분리부(333)는 문자열을 분리한다. 즉 문자열 분리부(333)는 문자열의 시작 위치를 알면 현재 폰트에서 한 문자의 최대 넓이만큼 자르고, 자른 곳을 기준으로 좌측 수직선 검색(left vertical line search)를 통해 폰트의 자간이 생기는 지점에서 한 문자의 끝을 알아 낸다. 그리고 다음 문자는 현재 문자의 끝나는 위치를 기준으로 현재 폰트의 한 문자의 최대 넓이를 더하고 다시 좌측 수직선 검색(left vertical line search)를 통해 한 문자의 끝을 구한다.

사물 배경 분리부(337)는 사물과 배경을 분리한다. 본 발명에서는 사물과 배경을 분리하는 방법으로, 다층 히스토그램 임계치(Multi-Layer Histogram Thresholding) 방법을 사용한다.

다층 히스토그램 임계치방법은, 투시관측(Perspective view)에서는 거리에 따른 시야(Field Of Vision, fov)가 증가하므로, 실제 면적에 비례하게 화면 위쪽으로 갈수록 작은 비율로 등분하여 층(Layer)을 만들고, 각 층(Layer)의 히스토그램(Histogram)에서 명도((R+ G+ B)/3)의 분포가 낮은 픽셀을 저장한다. 각각의 층(Layer)을 원래의 위치대로 하나의 층(Layer)으로 만들면, 사물의 위치는 키 색(key color)으로 나타내진다. 예를들어 도 7에서, 각각의 층(Layer)을 원래의 위치대로 하나의 층(Layer)으로 만들면, 분홍색(key color) 부분이 사물의 위치임을 알 수 있다. 해당 위치와 키 값으로 연결성분 표식(Connected Component Labeling)을 통해 배경과 사물을 분리시킬 수 있다.

인식부(340)는 분리부(330)에서 분리된 문자열 데이터를 수신하여 해당문자를 인식하며, 또한 분리부(330)에서 분리된 사물 데이터를 수신하여 해당사물을 인식하고 그 결과를 에이전트부(200)의 입력관리부(240)로 전송한다. 인식부(340)는 문자인식부(343)와 사물인식부(347)를 구비한다.

문자인식부(343)는 분리부(330)의 문자열 분리부(334)로부터 수신된 문자정보로 부터 해당문자를 인식한다. 문자인식부(343)에서 문자 인식은 원형 정합 방법(template matching method)중에서 폰트 정합 방법(Font Matching Method)을 사용한다. 원형 정합 방법은 문자열 분리부(333)의 출력신호를 기저장된 표준모델 즉 원형과 비교하여 문자대상을 인식하는 방법이다. 기존의 문자 인식으로 사용되던 구조적 특징을 이용한 자소 분리의 경우에는 세선화 과정에서 속도가 저하되며, 신경회로망을 이용한 서체인식을 하여 문자의 속성과 특징을 추출하는 방법은 학습 과정에서 마비 현상이 일어날 수 있는데, 본 발명의 문자인식부(343)은 이러한 문제점을 폰트를 대상으로 하는 문자단위의 원형 정합 방법을 사용하여 해결하였다.

문자인식부(343)는 폰트 로딩부, 폰트 이미지트리 생성부, 검색부로 이루어진다.

폰트 로딩부는 분리부(330)로 부터 폰트를 결정하고 결정된 폰트파일을 운영체제의 시스템 폰트 테이블에 추가하여 폰트를 로딩한다. 즉, 폰트 로딩부는 사용빈도가 높은 문자를 시드(seed)로 하여 각 폰트에 해당 하는 문자의 이미지를 타겟 영상에서 스캔(scan)하여 정합율(matching score)이 높은 폰트로 결정하거나, 사용자가 미리 지정한 폰트로 결정한다. 결정된 폰트파일을 운영체제의 시스템 폰트 테이블에 추가하여, 폰트를 로딩한다.

폰트 이미지트리 생성부는 폰트로딩부로부터 폰트가 로딩되면, 문자의 코드를 출력하여 이미지트리에 순차적으로 삽입하여, 이진 트리의 단말노드에 각 문자의 코드를 저장시킨다. 한글코드의 경우, 이미지 트리 생성 순서는 도 8에서와 같이 '가'에 해당하는 유니코드 완성형 문자인 '0xAC00'를 화면DC에 출력하면 현재 로딩된 폰트의 '가'에 해당하는 이진화된 문자 이미지를 얻을 수 있다. 얻어진 '가'의 이미지를 시드(seed)로하여 다음과 같은 이미지 트리 생성과정을 거쳐 이미지트리의 단말노드에 '가'유니코드 '0xAC00'를 저장시킨다.

검색부에서 검색시간은 픽셀수에 의해 결정되며, 픽셀수는 폭(width)×높이(height)에 의해 결정된다(도 9a 참조). 키 값은 모두 단말노드에 저장되어 있으므로 한 문자에 해당하는 검색 시간은 모두 동일하다.

검색부는 화면에 출력되는 문자열 이미지를 캐릭터 별로 각각 분리하여, 문자를 폰트 이미지 트리에서 검색한다. 검색 결과는 해당 문자의 완성형코드가 되므로 영상을 문자로 추출하게 된다.

사물인식부(347)는 사물-배경 분리(Segmentation)를 통해 인식된 객체의 이름을 키 값으로 하여 화면으로부터 출력되는 해당 키 값과 매치되는 사물의 특징인색상, 크기 등을 데이터베이스(DB)에 저장하고 학습한다. 인식하고자 하는 사물은 사용자의 설정(Object Recognition Setting)에 의해 정해진다. 설정은 사물 이름(object name), 키 색상(key color), 사물 영상(object image) 등의 항목으로 정해지며, 사물 영상(object image)의 경우 타겟 프로그램으로부터 설정 에디터로 드래그 앤 드롭(drag and drop), 카피 앤 페이스트(copy and paste) 방식을 취한다.

테스트부(400)는 사물인식부(347)로부터 인식된 객체의 이름을 키 값으로 하여 화면으로부터 출력되는 해당 키 값과 매치되는 사물의 특징인 색상, 크기 등을 데이터베이스(DB)에 저장하고 학습한다.

데이터베이스(500)은 테스트부(400)로부터 인식된 객체의 이름을 키 값으로 하여 화면으로부터 출력되는 해당 키 값과 매치되는 사물의 특징인 색상, 크기 등을 저장한다.

도 2는 일반적인 OAA(Open Agent Architecture) 시스템를 설명하기위한 설명도로, 에이전트부(200)는 조정에이전트(본 발명에서는 상황관리자)와, 다수의 응용에이전트(본 발명에서는 사건관리자)를 구비한다.

조정에이전트는 응용에이전트들의 상위에 존재하여 응용에이전트를 총괄한다. 입력되는 타겟 프로그램의 이벤트 상태를 ICL로 해석한 결과를 수신하여, 지시된 작업을 수행하기 위해 어느 응용에이전트에게 어떤 작업을 맡길 것인가를 판단하고 적절한 응용에이전트를 선택하여 작업을 요구한다.

응용에이전트는 조정에이전트로부터 요구가 들어오면 타겟 어플리케이션의 이벤트 상태에 따라 어떠한 작업을 처리할 것인지를 도메인지식을 가지고 결정하여 그에 맞는 인식기부(300)를 호출한다. 즉, 응용에이전트는 특정한 영역의 작업을 전문적으로 처리하는 역할을 하며, 사용자가 지시한 스크립트를 가지고 해당 도메인지식의 학습을 위한 독립적인 프로세스라 할 수 있다.

본 발명의 에이전트부는 범용적인 영상인식 어플리케이션 시스템을 구축하기 위한 필수요건인 확장성, 이식성, 공유성, 관리성, 조작성을 갖추고 있다.

도 3은 본 발명의 ICL의 구조를 설명하기위한 구조도이다. ICL부(210)는 타겟부(100)로 부터 입력되는 타겟 프로그램의 이벤트 상태를 에이전트간 통신언어(ICL)로 해석한다. 일반적으로 분산 네트워크 환경 하에서의 에이전트간의 협동 작업을 수행하기 위해서 에이전트 간의 통신언어를 사용하며, 본 발명의 ICL은 콘텐트(content)층, 프로토콜(protocol)층, 랩퍼(wrapper)층으로 구성된 3개의 층으로 이루어진다.

콘텐트층은 에이전트 간에 교류되는 메시지 내용을 나타낸다.

프로토콜층은 에이전트들 간의 약속된 규칙을 의미한다. 예를 들면, solve라는 프로토콜이 담긴 메시지를 받은 에이전트는 solved라는 프로토콜이 담긴 메시지를 되돌려 주어야 한다는 규칙을 내포할 수 있다.

랩퍼층은 에이전트가 그 내용을 보고 그 메시지가 ICL로 해석되어 질 수 있는 메시지인지 아닌지 판단하며, 어느 에이전트가 보낸 것인가를 알아낼 수 있도록 한다.

도 4는 타겟 프로그램이 온라인게임인 경우 상황관리자의 상황 분류의 일예이다. 상황관리자(223)는 타겟 프로그램으로부터 현재 상황을 인식하고 분석된 결과를 토대로 가장 효율적인 일 처리를 위한 판단을 하는 상위의 총괄 관리를 담당한다. 타겟 프로그램이 온라인게임인 경우에, 상황관리자(223)는 쉬는중, 적을 탐색 중, 전투 중, 전리품 획득 중, 시야 회전 중, 길 이동 중, 치료 중의 상황으로 분류하여 인식대상을 가공할 수 있다.

도 5는 이벤트 에디터 인터페이스의 일 예이다. 사건관리자(227)는 인식하고자 하는 타겟 프로그램의 이벤트를 사용자에 의해 설정된 키워드에 해당하는 영상 및 문자를 인식하여 매치된 트리거의 액션을 보내는데, 사건관리자(227)의 각종 설정은 사용자가 직접 입력할 수 있는 이벤트에디터 툴을 통해 제공된다. 이벤트 에디터 인터페이스에서, 설정은 클래스(Class), 아리아스(Aliases), 트리거(Trigger), 맥크로(Macro) 등으로 분류되며 문자와 영상 모두 패턴에 따른 행동을 부여할 수 있다.

도 6은 도 1의 입력 관리자부와 WDM 디바이스 드라이버의 구성을 설명하기위한 구성도이다. 즉, 도 6에서 본 발명의 윈도우 디바이스 드라이버(WDM)의 구조를 설명한다.

입력 관리자부(240)는, 인식부(340)으로 부터 문자와 사물에 대한 인식 데이터를 수신하고, 이를 프로토콜에 따른 사건관리자(227)의 판단에 따라 응답, 즉 I/O 요청 패킷(IRP)을, WDM 디바이스 드라이버(243)로 출력한다.

WDM 디바이스 드라이버(243)는 입력관리자부(240)의 출력을 수신하여 타겟부(100)에 맞는 데이터형태로 바꾸고, 하드웨어 추상화 계층(HAL)을 통해 타겟부(100)로 전송한다. WDM 디바이스 드라이버(243)는 필터 디바이스 오브젝트(Filter Device Object)(245), 기능 디바이스 오브젝트(Function Device Object)(250), 물리적 디바이스 오브젝트(Physical Device Object)(255), 상위 필터 드라이버(Upper filter driver)(247), 기능 드라이버(Function driver)(252), 버스 드라이버(Bus driver)(257)를 구비하며, 계층적 디바이스 구조를 이룬다.

필터 디바이스 오브젝트(Filter Device Object)(245), 기능 디바이스 오브젝트(Function Device Object)(250), 물리적 디바이스 오브젝트(Physical Device Object)(255)로 연결되는 세로 행은 상위 방향으로 연결된 커널 DEVICE_OBJECT 구조체의 스택인데, 이 구조는 이 입력 시스템이 하나의 하드웨어를 관리하게 하기위한 것이다.

상위 필터 드라이버(Upper filter driver)(247), 기능 드라이버(Function driver)(252), 버스 드라이버(Bus driver)(257)로 연결되는 세로 행은 관리자 역할을 하는 디바이스 드라이버의 집합이다. 그리고 드라이버 통해 지나는 I/O(입출력) 요청 패킷(I/O Request Packet)(이하 IRP라 한다)의 흐름을 알 수 있다.

윈도우 NT에서는 유저모드와 커널모드로 구분되어 소프트웨어가 실행된다. 즉, 유저모드 프로그램은 Win32 시스템에서 실행되며, 커널모드 프로그램은 하드웨어 관련 소프트웨어를 다룬다. 윈도우 NT는 다양한 CPU를 지원하고 여러개를 사용 할 수 있으므로 직접 디바이스 제어하는 드라이버 모델인 VxD(가상 장치 드라이버) 드라이버를 작성할 수 없다. 따라서 본 발명의 WDM 디바이스 드라이버는 하드웨어 추상화 계층(Hardware Abstraction Layer)(이하 HAL이라 한다)이라는 최종 인터페이스를 통해 실제 디바이스인 마우스 키보드, HID등과 통신한다.

상위 필터 드라이버(247)는 기능 드라이버(function driver)(252)의 상위에 위치하며 이 드라이버가 입출력 요구를 처리하도록 만든 구조다. 즉, 필터 드라이버(247)는 자신의 계층을 통과하는 IRP를 필터링하는 WDM 디바이스 드라이버이다. 일반적으로 필터 드라이버는 계층형 드라이버 구조에서 기능 드라이버(function driver)(252)의 상위나 하위에 놓여질 수 있다.

기능 드라이버(252)는 WDM 디바이스 드라이버를 위한 메인 드라이버로, I/O를 위한 기초연산관리, 디바이스 지정 전원 요구량 관리등을 한다.

버스 드라이버(257)은 버스(Bus) 계산, PnP 및 전력 관리(Power Management) IRP에 반응하고, 필요시 버스의 다중 액세스에 관여한다.

OS는 I/O 요청 패킷(IRP)이라는 데이터 구조를 사용해서 커널 모드 디바이스 드라이버와 통신한다. 윈도우 NT에서는 디바이스에 대한 입출력 처리를 패킷 단위로 처리하며 드라이버 개발자는 이 패킷을 데이터 구조체 IRP(I/O Request Packets)를 사용해 생성, 전달, 처리할 수 있다. IRP는 애플리케이션이나 다른 드라이버에서 전달받아 현재 드라이버에서 패킷 처리를 완료하거나 다른 하위 드라이버에 전달해 처리를 요청할 수도 있다.

하드웨어 추상화 계층(HAL)은 WDM 디바이스 드라이버(243)으로 부터 수신된 IRP를 타겟부(100)의 마우스, 키보드, HID(Human Interface Device) 등으로 전달한다. 하드웨어 추상화 계층(HAL)은 입출력 인터페이스, 인터럽트 제어, 다중 프로세서 통신을 구현하는 데 필요한 하드웨어 종속적인 세부 사항을 처리하는 루틴들의 집합체로, 마우스, 키보드, HID(Human Interface Device) 등에 해당하는 입력 하드웨어와 WDM 드라이버와의 인터페이스 호환성을 제공해준다.

도 7은 다층 히스토그램 임계치(Multi-Layer Histogram Thresholding)를 이용한 사물 배경 분리의 일예이다. 도 7은 실제 면적에 비례하게 화면 위쪽으로 갈수록 작은 비율로 등분하여 층(Layer)을 만들고, 각 층(Layer)의 히스토그램에서 명도((R+ G+ B)/3)의 분포가 낮은 픽셀을 저장한후, 각각의 층(Layer)을 원래의 위치대로 하나의 층(Layer)으로 만들어서, 사물의 위치는 키 색(key color)으로 나타내게 한 것이다. 도 7에서 분홍색(key color) 부분이 사물의 위치임을 알 수 있다. 해당 위치와 키 값으로 연결 성분 표식(Connected Component Labeling)을 통해 배경과 사물을 분리시킬 수 있다.

도 8은 본 발명의 일실시예에 의한 이미지 트리 생성 순서도이다. 문자인식부(343)는 폰트 로딩부, 폰트 이미지트리 생성부, 검색부으로 이루어진다. 폰트 로딩부는 사용빈도가 높은 문자를 시드(seed)로 하여 각 폰트에 해당 하는 문자의 이미지를 타겟 영상에서 스캔(scan)하여 정합율(matching score)이 높은 폰트로 결정하며, 결정된 폰트파일을 운영체제의 시스템 폰트 테이블에 추가하여, 폰트를 로딩한다. 폰트 이미지트리 생성부는 폰트로딩부로부터 폰트가 로딩되면, 문자의 코드를 출력하여 이미지트리에 순차적으로 삽입하여, 이진 트리의 단말노드에 각 문자의 코드를 저장시킨다. 한글코드의 경우 '가'에 해당하는 유니코드 완성형 문자인 '0xAC00'를 화면DC에 출력하면 현재 로딩된 폰트의 '가'에 해당하는 이진화된 문자 이미지를 얻을 수 있다. 얻어진 각 문자의 이미지를 시드(seed)로 하여 다음과 같은 이미지 트리 생성과정을 거쳐 이미지트리의 단말노드에 '가'유니코드 '0xAC00'부터 ' R' 유니코드 '0xD7A3'까지 저장시킨다.

이미지트리를 초기화한다(S1100). 즉 루트노드를 생성한다. 여기서 생성은 값을 채우는 것이 아니라 단순히 메모리를 할당한다는 의미이다.

폰트를 결정하고 결정된 폰트파일을 운영체제의 시스템 폰트 테이블에 추가하여 폰트를 로딩한다(S1110).

한글코드의 경우 '가'에 해당하는 유니코드 완성형 문자인 '0xAC00'를 화면DC에 출력한다(S1120).

현재 로딩된 폰트에 해당하는 이진화된 문자 이미지를 출력한다(S1140). 즉, 현재 로딩된 폰트의 '가'에 해당하는 이진화된 문자 이미지를 얻는다

출력된 문자 이미지를 메모리에 저장한다(S1150).

이미지트리의 단말노드에 해당 유니코드를 저장한다(S1160). 여기서는 이미지트리의 단말노드에 '가'유니코드 '0xAC00'를 저장시킨다.

이미지의 지정주소 픽셀의 값이 0이 아닌가, 즉 이미지의 지정주소 픽셀이 배경색(픽셀의 값이 0)인지 아닌지를 검사한다(S1170). 여기서 루트 노드의 진행 방향은'가' 이미지의 첫 번째 지정주소 픽셀 값에 의해 결정된다.

이미지의 지정주소 픽셀의 값이 0이면, 즉 배경색이면, 왼쪽 차일드로 간다(S1180).

이미지의 지정주소 픽셀의 값이 0이 아니면, 즉 배경색이 아니면, 오른쪽 차일드로 간다(S1190).

각 차일드의 노드(Node)가 비어(NULL) 있는가를 검사하여(S1200), 비어(NULL) 있다면 노드를 생성한다(1210).

다음번 픽셀을 지정하도록 픽셀 지정 주소를 인크리먼트하고(S1220), 모든 픽셀을 전부 검사하였는가를 확인한다(S1230). 모든 픽셀수는 폭(width)×높이(height)에 의해 결정된다.

아직 모든 픽셀을 전부 검사하지 않았다면, 다음의 픽셀의 값이 0이 아닌가, 즉, 다음 지정주소 픽셀이 배경색(픽셀의 값이 0)인지 아닌지를 검사하고(S1170), 같은 방법으로 차일드 노드를 생성해 나간다. 이런 식으로 마지막(169번째) 픽셀까지 비교하여 단말노드까지 진행한다.

모든 픽셀을 전부 검사하였다면, 단말노드에 첫 번째 문자 '가'의 유니코드 완성형 '0xAC00'을 저장한다(S1240).여기까지 '가'에 해당하는 이미지트리에 삽입과정이다.

유니코드 완성형 한글코드 11172자인 '가'~' R'까지 이미지트리를 전부 삽입하였는가를 검사한다. 즉 ' R'유니코드 완성형 '0xD7A3'까지 저장되었는가를 검사하고(S1250), ' R'유니코드 완성형 '0xD7A3'까지 저장되었다면 이미지 트리생성을 종료한다(S1270).

이미지 트리가 아직 전부 만들어지지 않았다면, 즉' R'유니코드 완성형 '0xD7A3'까지 저장되어 있지 않다면, 다음 문자로 가 가서(S1260), 현재 로딩된 폰트에 해당하는 이진화된 문자 이미지를 출력하여(S1140), 반복하여 수행한다.

두 번째 문자부터 이미지트리에 삽입할 때 이미 생성된 노드에 대해서는 노드를 생성하지 않고 이동할 수 있다. 즉, 단말노드의 경로에 해당하는 부모노드만 생성된다.

여기서 단말노드를 제외한 모든 노드는 값(데이터)을 가질 필요가 없고, 이진화된 픽셀 값인 0, 1에 의해 진행 방향을 결정할 뿐이므로 노드당 한 비트만 사용하여 트리의 물리적 크기를 줄일 수 있다. 이렇게 만들어진 이미지 트리는 파일로 저장한다.

이러한 과정을 유니코드 완성형 한글코드 11172자인 '가'~' R'까지 반복하여 현재 로딩된 폰트에 해당하는 이미지트리를 만들 수 있다. 에이전트 구동 시에는 해당 폰트의 이미지트리가 저장되어 있는 파일이 로드된다.

도 9a는 본 발명의 루트 노드의 일실시예를 설명하기위한 '가'의 이미지이고, 도 9b는 도 9a의 '가'의 이미지로부터 폰트 이미지 트리생성을 설명하기위한 폰트 이미지 트리구조이다.

도 9b에서 한글의 완성형코드를 출력하여 이미지트리에 순차적으로 삽입하면 이진 트리의 단말노드에 각 문자의 완성형 코드가 저장된다. 이 때 루트 노드는 도 9a에서와 같이 문자 이미지의 첫 번째 픽셀이 된다. 차일드 노드의 진행 방향은 이진화된 첫 번째 픽셀 값에 의해 결정되는데, 배경색(픽셀값이 0)이면 왼쪽 차일드 노드를 생성하고 배경색이 아니면(픽셀값이 1) 오른쪽 차일드 노드를 생성한다. 생성된 노드의 진행방향은 두 번째 픽셀의 값에 의해 결정되며 같은 방법으로 차일드 노드를 생성해 나간다. 단말노드에 첫 번째 문자의 완성형코드가 저장되면 두 번째 문자부터 이미지트리에 삽입할 때 이미 생성된 노드에 대해서는 노드를 생성하지 않고 이동할 수 있다. 즉, 단말노드의 경로에 해당하는 부모노드만 생성되

므로 공간을 낭비하지 않고, 단말노드를 제외한 모든 노드는 이진화된 픽셀 값인 0, 1에 의해 결정되므로 노드당 한 비트만 사용해 트리의 물리적 크기를 최소화할 수 있다. 여기까지 각 폰트에 해당하는 이미지트리를 생성하는 작업이다. 에이전트 구동 시에는 해당폰트의 이미지트리가 저장되어 있는 파일이 로드 된다.

검색부에서 검색시간은 픽셀수에 의해 결정되며, 픽셀수는 폭(width)×높이(height)에 의해 결정된다. 키 값은 모두 단말노드에 저장되어 있으므로 한 문자에 해당하는 검색 시간은 모두 동일하다. 검색부는 화면에 출력되는 문자열 이미지를 캐릭터 별로 각각 분리하여, 문자를 폰트 이미지 트리에서 검색한다. 검색 결과는 해당 문자의 완성형코드가 되므로 영상을 문자로 추출하게 된다.

도 10은 도 1의 문자인식부에서 문자 인식 순서도의 일예이다. 즉 문자인식부(345)의 검색부에서 문자인식에 대한 흐름도이다.

인식하고자 하는 대상(관심영역)을 이진화 한다(S2100). 즉, 화면으로 출력되는 문자열 이미지를 임계치부(320)에서 이진화 한다.

이진화된 관심영역에서 문자열을 분리한다(S2110). 즉, 임계치부(320)에서 이진화된 데이타를 분리부(330)에서 캐릭터(한무자) 단위로 추출한다.

문자 단위로 이미지 트리를 검색한다(S2130). 즉 분리부(330)에서 추출한 문자를 생성된 이미지 트리에서 검색한다.

이미지의 픽셀의 값이 0이 아닌가, 즉 이미지의 픽셀이 배경색(픽셀의 값이 0)인지 아닌지를 검사한다(S2140). 즉 도 8의 이미지트리 생성과정과 마찬가지로 노드의 진행방향은 문자 이미지의 픽셀값에 의해 결정된다.

이미지의 픽셀의 값이 0이면, 즉 배경색이면, 왼쪽 차일드로 간다(S2150).

이미지의 픽셀의 값이 0이 아니면, 즉 배경색이 아니면, 오른쪽 차일드로 간다(S2160).

다음번 픽셀을 지정하도록 픽셀 지정 주소를 인크리먼트하고(S1220), 모든 픽셀을 전부 검사하였는가를 확인한다(S1230). 모든 픽셀수는 폭(width)×높이(height)에 의해 결정된다.

아직 모든 픽셀을 전부 검사하지 않았다면, 다음의 픽셀의 값이 0이 아닌가, 즉, 다음 픽셀이 배경색(픽셀의 값이 0)인지 아닌지를 검사하고(S2140), 같은 방법으로 차일드 노드를 검색해간다. 이런 식으로 마지막 픽셀까지 비교하여 단말노드까지 진행한다.

모든 픽셀을 전부 검사하였다면, 단말노드의 데이터 즉, 유니코드를 출력하며(S2190), 이는 해당 문자의 완성형코드가 되므로 영상을 문자로 추출하게 된다(S2200). 즉, 마지막(169번째) 픽셀 까지 비교하여 단말노드까지 진행하면 문자 이미지에 해당하는 유니코드 완성형 값(예로, 0xAC00)을 얻을 수 있다. 검색결과는 해당문자의 완성형코드가 되므로 영상을 문자로 추출하게 된다.

도 11은 도 1의 입력관리자부에서 문자 입력 순서도의 일예이다.

문자수를 나타내는 문자수 카운터를 0으로 한다(S3100).

문자수 카운터의 값이 글자수보다 작은가를 검사한다(S3110). 즉 글자수에 따른 모든 글자를 입력하였는지를 확인한다.

문자수 카운터의 값이 글자수보다 크거나 같다면, 즉 글자수에 따른 모든 글자를 입력하였다면, 문자입력을 완료한다(S3190).

문자수 카운터의 값이 글자수보다 작다면, 즉 아직 글자수에 따른 모든 글자를 입력하지 못하였다면, 현재의 글자수 카운터가 나타내는 글자가 한글인가를 검사한다(S3120).

현재의 글자수 카운터가 나타내는 글자가 한글이라면, 유니코드로 변환하고(S3130), 초성, 중성, 종성으로 나눈다(S3140).

WDM(Windows Devices Model) 드라이버를 통하여 입력할 것인가를 확인한다(S3150).

WDM 드라이버를 통한 입력이라면, 디바이스로 패킷을 전송한다(S3160).

WDM 드라이버를 통한 입력이 아니라면, 가상 키코드로 입력한다(S3170).

문자수 카운터를 인크리먼트 하고(S3180), 문자수 카운터의 값이 글자수보다 작은가를 검사하여(S3110), 반복한다.

상술한 바와 같이, 본 발명에서는 화면에 출력되는 타겟부(100)의 타겟 프로그램의 영상을 미리 부여된 설정에 따라 인식하여 데이터베이스(500)에 저장하여, 학습에 의한 수행이 이루어진다. 즉 본 발명은 가상현실(가상환경) 시스템(Virtual Environment System)을 이용하여 이루어진다. 가상현실 시스템은 모의실험(simulate)함으로써 그것을 사용하는 사람이 마치 실제 주변 상황 및 환경과 상호작용을 하고 있는 것처럼 만들어 주는 인간-컴퓨터간 인터페이스이다. 작업공간이 하드웨어로 상호 연결된다.

본 발명은 이상에서 설명되고 도면에서 예시된 것에 의해 한정되는 것은 아니며, 당업자라면 다음에 기재되는 청구범위 내에서 더 많은 변형 및 변용예가 가능한 것임은 물론이다.

발명의 효과

이상에서 설명한 바와 같이, 본 발명의 영상인식 에이전트 시스템은 사용자의 컴퓨터를 통해 들어오는 가상환경의 모든 영상을 인식하며, 영상인식에 대한 지식을 바탕으로 모니터로 출력되는 영상의 패턴을 분석하여 그에 따른 적절한 반응을 일으킨다. 또한 본 발명의 영상인식 에이전트 시스템은 영상인식을 통해 상황에 따라 설정된 전략을 수행할 수 있으며, 디스플레이되는 중요한 정보를 판단해 사용자와 비슷한 수준으로 끌어올리며 신속 정확하게 작업을 수행할 수 있다.

본 발명의 영상인식 에이전트 시스템은, 사용자가 반드시 처리해야 할 일 중 대기시간이 많이 걸리는 일, 정보가 많이 흩어져 있는 곳에서 필요한 정보를 수집하는 일, 반복된 신호를 보내야 하는 일, 예를 들어 호텔 예약이나 교통편 예약, 온라인 수강신청, 온라인 경매 시스템과 같이 한정된 정보를 이용하여 특정 시간에 정해진 작업을 수행하는 일과 같이 시간과 노력(cost)을 요하는 작업을 대행해 줄 수 있다. 따라서 사용자는 부가가치가 높은 일에 좀 더 많은 시간을 보낼 수 있고, 반복된 행동을 통해 보수를 얻을 수 있는 온라인게임 시스템에서 사용자가 본 발명의 도구를 통해 전략을 설정하면 반복된 입력과 상황에 따른 작업을 대행해 준다.

본 발명의 영상인식 에이전트 시스템은, 원하는 정보만 히스토리 DB에 저장하고, 객체별 분석, 출력이 가능하다. 사람은 이 데이터를 분석, 평가하여 특정 상황에 따른 행동을 재구성(refactory)할 수 있다. 따라서 시간의 흐름에 따른 효율성이 증가한다. 중요한 사건 또는 예외상황이 발생하면 pc스피커, 이메일, 핸드폰 문자전송 등의 도구를 통해 보고를 해준다.

(57) 청구의 범위

청구항 1.

타겟 프로그램을 구비하는 타겟부; 상기 타겟부로부터 수신된 타겟 어플리케이션을 분석하여 현재의 이벤트 상태를 인식하고, 그 인식된 결과와 도메인지식을 가지고 어떠한 작업을 처리할 것인지를 결정하는 에이전트부; 상기 에이전트부로부터 호출되어, 상기 에이전트부로 부터 수신된 데이터를 이진화를 하며, 이진화된 관심영역의 문자열에서 각 문자를 분리하며 또한 사물을 배경으로부터 분리하고, 분리된 문자열의 해당문자를 인식하며 분리된 사물의 해당사물을 인식하는 인식기부;를 구비하는 영상인식을 이용한 에이전트 시스템에 있어서,

상기 에이전트부는

상기 타겟부로 부터 입력되는 타겟 프로그램의 이벤트 상태를 에이전트간 통신언어(ICL)로 해석하는 ICL부;

상기 ICL부로 부터 타겟 프로그램의 이벤트 상태를 ICL로 해석한 결과를 수신하여 인식하고자 하는 타겟 프로그램의 이벤트를 사용자에 의해 설정된 키워드에 해당하는 영상 및 문자를 인식하여 매치된 트리거의 액션을 보내는 이벤트 관리자부;

상기 이벤트 관리자부에 의해 인식해야 할 내용이 정해지면 필요한 영역만큼만 캡쳐(Capture)하여 상기 인식기부로 전송하는 캡츄어부;

상기 캡츄어부에서 캡쳐하여 상기 인식기부로 보낸 것에 대한 응답을 상기 인식기부로부터 수신하고, 상기 이벤트 관리자부의 판단에 따라 타겟부로 출력하는 입력 관리자부;를 구비하는 것을 특징으로 하는 영상인식을 이용한 에이전트 시스템.

청구항 2.

제1항에 있어서,

상기 ICL부의 ICL은,

에이전트 간에 교류되는 메시지 내용을 나타내는 콘텐트(content)층;

프로토콜층은 에이전트들 간의 약속된 규칙을 나타내는 프로토콜(protocol)층;

에이전트가 그 내용을 보고 그 메시지가 ICL로 해석되어 질 수 있는 메시지인지 아닌지 판단하며, 어느 에이전트가 보낸 것인가를 알아낼 수 있도록 하는 랩퍼(wrapper)층;으로 이루어지는 것을 특징으로 하는 영상인식을 이용한 에이전트 시스템.

청구항 3.

제1항에 있어서,

상기 이벤트 관리자부는 상황관리자와 사건 관리자로 이루어지며,

상기 상황관리자는 상기 ICL부에서 타겟부로 부터 입력되는 타겟 프로그램의 이벤트 상태를 ICL로 해석한 결과를 수신하여, 지시된 작업을 수행하기위한 사건관리자를 선택하여 작업을 요구하며,

상기 사건 관리자는 상기 상황관리자로 부터 수신된 타겟 프로그램의 이벤트에서 사용자에 의해 설정된 키워드에 해당하는 영상 및 문자, 즉 인식해야할 내용을 설정하는 것을 특징으로 하는 영상인식을 이용한 에이전트 시스템.

청구항 4.

제1항에 있어서,

상기 캡츄어부는 전체화면 디바이스 콘텍스트(DC) 또는 응용프로그램의 핸들 값에 해당하는 DC를 얻어와 메모리 DC에서 DIB(Device Independent Bitmap) 형식으로 저장하는 것을 특징으로 하는 영상인식을 이용한 에이전트 시스템.

청구항 5.

제1항에 있어서,

상기 캡츄어부는 상기 캡츄어부는 그래픽 출력 하드웨어의 메모리를 후킹(hooking)하여 직접적(direct)으로 RGB칼라를 가져오는 것을 특징으로 하는 영상인식을 이용한 에이전트 시스템.

청구항 6.

제1항에 있어서,

상기 에이전트 시스템은 WDM(Windows Driver Model) 디바이스 드라이버와 하드웨어 추상화 계층(Hardware Abstraction Layer)(HAL)를 더 구비하며,

상기 입력 관리자부는, 상기 인식부로 부터 문자와 사물에 대한 인식 데이터를 수신하고, 이를 프로토콜에 따른 사건관리자의 판단에 따라 I/O 요청 패킷(IRP)을 상기 WDM 디바이스 드라이버로 전송하며,

상기 WDM 디바이스 드라이버는 상기 입력관리자부에서 출력된 IRP를 수신하여 타겟부에 맞는 데이터형태로 바꾸고 이를 하드웨어 추상화 계층(HAL)로 전송하며,

상기 하드웨어 추상화 계층(HAL)은 상기 WDM 디바이스 드라이버로 부터 수신된 IRP를 타겟부의 마우스, 키보드, HID (Human Interface Device)에 전달하는 것을 특징으로 하는 영상인식을 이용한 에이전트 시스템.

청구항 7.

타겟 프로그램을 구비하는 타겟부; 상기 타겟부로부터 수신된 타겟 어플리케이션을 분석하여 현재의 이벤트 상태를 인식하고, 그 인식된 결과와 도메인지식을 가지고 어떠한 작업을 처리할 것인지를 결정하는 에이전트부; 상기 에이전트부로부터 호출되어, 상기 에이전트부로 부터 수신된 데이터를 이진화를 하며, 이진화된 관심영역의 문자열에서 각 문자를 분리하며 또한 사물을 배경으로부터 분리하고, 분리된 문자열의 해당문자를 인식하며 분리된 사물의 해당사물을 인식하는 인식기부;를 구비하는 영상인식을 이용한 에이전트 시스템에 있어서,

상기 인식기부는

상기 에이전트로부터 수신된 데이터로부터 관심영역을 결정하며, 결정된 관심영역을 중요도에 따라 순위를 매기는 ROI부;

상기 ROI부로 부터 관심영역내의 데이터를 임계치(Thresholding value)를 이용하여 이진화하는 임계치(Thresholding)부;

상기 임계치부로부터 수신된 상기 이진화된 관심영역의 문자열에서 각 문자를 분리하는 문자열 분리부와, 상기 임계치부로부터 수신된 이진화된 관심영역에서 사물을 배경으로부터 분리하는 사물 배경 분리부를 구비하는 분리(Segmentation)부;

상기 분리부에서 분리된 문자열 데이터를 수신하여 해당문자를 인식하는 문자인식부와, 상기 분리부에서 분리된 사물 데이터를 수신하여 해당사물을 인식하는 사물인식부를 구비하는 인식부;를 구비하는 것을 특징으로 하는 영상인식을 이용한 에이전트 시스템.

청구항 8.

제7항에 있어서,

상기 ROI부는 관심영역 특정 수직선(vertical line)을 정하여 3 픽셀(pixel) 간격으로 이전 프레임과 비교하며, 이 때 RGB 각각의 값을 한 바이트에 저장하는 것을 특징으로 하는 영상인식을 이용한 에이전트 시스템.

청구항 9.

제7항에 있어서,

상기 임계치부는

기 설정된 임계치가 있으면, 영상의 R+ G+ B를 더한 회색 값(gray value)을 임계값으로 하여 이진화하거나, 영상의 R, G, B를 3차원 벡터로 하여 이진화하며,

기 설정된 임계치가 없으면, 반복적인 임계치선택(Iterative Threshold Selection)(ITS) 알고리즘, 2피크 임계치(Two Peak Thresholding) 알고리즘, 이중 임계치(Double Thresholding)알고리즘 중 하나를 사용하여 실시간으로 임계치를 정하여 이진화하는 것을 특징으로 하는 영상인식을 이용한 에이전트 시스템.

청구항 10.

제7항에 있어서,

상기 문자열 분리부는, 문자열의 시작 위치로부터, 현재 폰트에서 한 문자의 최대 넓이만큼 자르고, 자른 곳을 기준으로 좌측 수직선 검색(left vertical line search)를 통해 폰트의 자간이 생기는 지점에서 한 문자의 끝을 알아 내는 것을 특징으로 하는 영상인식을 이용한 에이전트 시스템.

청구항 11.

제7항에 있어서,

상기 사물 배경 분리부는 사물과 배경을 분리하기위해 다층 히스토그램 임계치(Multi-Layer Histogram Thresholding) 방법을 사용하는 것을 특징으로 하는 영상인식을 이용한 에이전트 시스템.

청구항 12.

제7항에 있어서,

상기 문자인식부는 폰트를 대상으로 하는 문자단위의 원형 정합 방법을 사용하는 것을 특징으로 하는 영상인식을 이용한 에이전트 시스템.

청구항 13.

제7항에 있어서,

상기 문자인식부는,

사용빈도가 높은 문자를 시드(seed)로 하여 각 폰트에 해당 하는 문자의 이미지를 타겟 영상에서 스캔(scan)하여 정합율(matching score)이 높은 폰트로 결정하며, 결정된 폰트파일을 운영체제의 시스템 폰트 테이블에 추가하여, 폰트를 로딩하는 폰트 로딩부;

상기 폰트로딩부로부터 폰트가 로딩되면, 문자의 코드를 출력하여 이미지트리에 순차적으로 삽입하여, 이진 트리의 단말 노드에 각 문자의 코드를 저장시키는 폰트 이미지트리 생성부;

화면에 출력되는 문자열 이미지를 캐릭터 별로 각각 분리하여, 문자를 폰트 이미지 트리에서 검색하고, 검색 결과로 해당 문자의 완성형코드가 되어 영상을 문자로 추출하게되는 검색부;를 구비하는 것을 특징으로 하는 영상인식을 이용한 에이전트 시스템.

청구항 14.

제7항에 있어서,

상기 사물인식부는 사물-배경 분리(Segmentation)를 통해 인식된 객체의 이름을 키 값으로 하여 화면으로부터 출력되는 해당 키 값과 매치되는 사물의 특징인색상, 크기를 데이터베이스(DB)에 저장하고 학습하는 것을 특징으로 하는 영상인식을 이용한 에이전트 시스템.

청구항 15.

제14항에 있어서,

인식하고자 하는 사물은 사용자의 설정(Object Recognition Setting)에 의해 정해지는 것을 특징으로 하는 영상인식을 이용한 에이전트 시스템.

청구항 16.

제7항에 있어서,

상기 에이전트 시스템은,

사물인식부로부터 인식된 객체의 이름을 키 값으로 하여 화면으로부터 출력되는 해당 키 값과 매치되는 사물의 특징인 색상, 크기 또는 사물의 이미지를 데이터베이스(DB)에 저장하고 학습하는 테스트부;

상기 테스트부로부터 인식된 객체의 이름을 키 값으로 하여 화면으로부터 출력되는 해당 키 값과 매치되는 사물의 특징인 색상, 크기 등을 저장하는 데이터베이스;를 더 구비하는 것을 특징으로 하는 영상인식을 이용한 에이전트 시스템.

청구항 17.

영상인식을 이용한 에이전트 시스템의 동작방법에 있어서,

이미지 트리 생성 방법은

루트노드를 메모리에 할당하는 이미지트리 초기화 단계;

폰트를 결정하고 결정된 폰트파일을 운영체제의 시스템 폰트 테이블에 추가하여 폰트를 로딩하는 폰트로딩단계;

현재 문자에 해당하는 유니코드 문자를 화면DC에 출력하는 유니코드문자 화면DC출력단계;

현재 로딩된 폰트에 해당하는 이진화된 문자 이미지를 출력하고 출력된 문자이미지를 메모리에 저장하는 이진화된 문자 이미지출력 단계;

이미지의 지정주소 픽셀이 배경색인지 아닌지를 검사하는 배경색여부 검사단계;

상기 배경색여부 검사단계에서, 이미지의 픽셀이 배경색이면 왼쪽 차일드 노드를 생성하고, 이미지의 픽셀이 배경색이 아니면 오른쪽 차일드노드를 생성하는 차일드 노드생성단계;

픽셀 지정주소를 인크리먼트하는 단계;

모든 픽셀(폭(width)×높이(height))을 전부 검사하였다면, 단말노드에 유니코드를 저장하는 유니코드 저장단계;를 적어도 포함하는 것을 특징으로 하는 영상인식을 이용한 에이전트 시스템의 동작방법.

청구항 18.

영상인식을 이용한 에이전트 시스템의 동작방법에 있어서,

문자인식 방법은,

인식하고자 하는 대상(관심영역)을 이진화하는 이진화단계;

이진화된 관심영역에서 문자열을 분리하는 문자열분리단계;

문자 단위로 이미지 트리를 검색하는 이미지트리 검색단계;

이미지의 픽셀이 배경색인지 아닌지를 검사하는 배경색여부 검사단계;

상기 배경색여부 검사단계에서, 이미지의 픽셀이 배경색이면 왼쪽 차일드 노드로 가고, 이미지의 픽셀이 배경색이 아니면 오른쪽 차일드노드로 가서 검색하는 차일드노드 검색단계;

모든 픽셀을 전부 검사하였다면, 단말노드의 데이터 즉, 유니코드를 출력하며, 이는 해당 문자의 유니코드가 되므로 영상을 문자로 추출하게 되는 문자추출단계;를 적어도 포함하는 것을 특징으로 하는 영상인식을 이용한 에이전트 시스템의 동작방법.

청구항 19.

영상인식을 이용한 에이전트 시스템의 동작방법에 있어서,

입력관리자부에서 문자 입력방법은

현재의 입력되는 글자가 한글인가를 검사하는 한글여부 검사단계;

현재의 입력되는 글자가 한글이라면, 유니코드로 변환하고, 초성, 중성, 종성으로 나누는 유니코드변환단계;

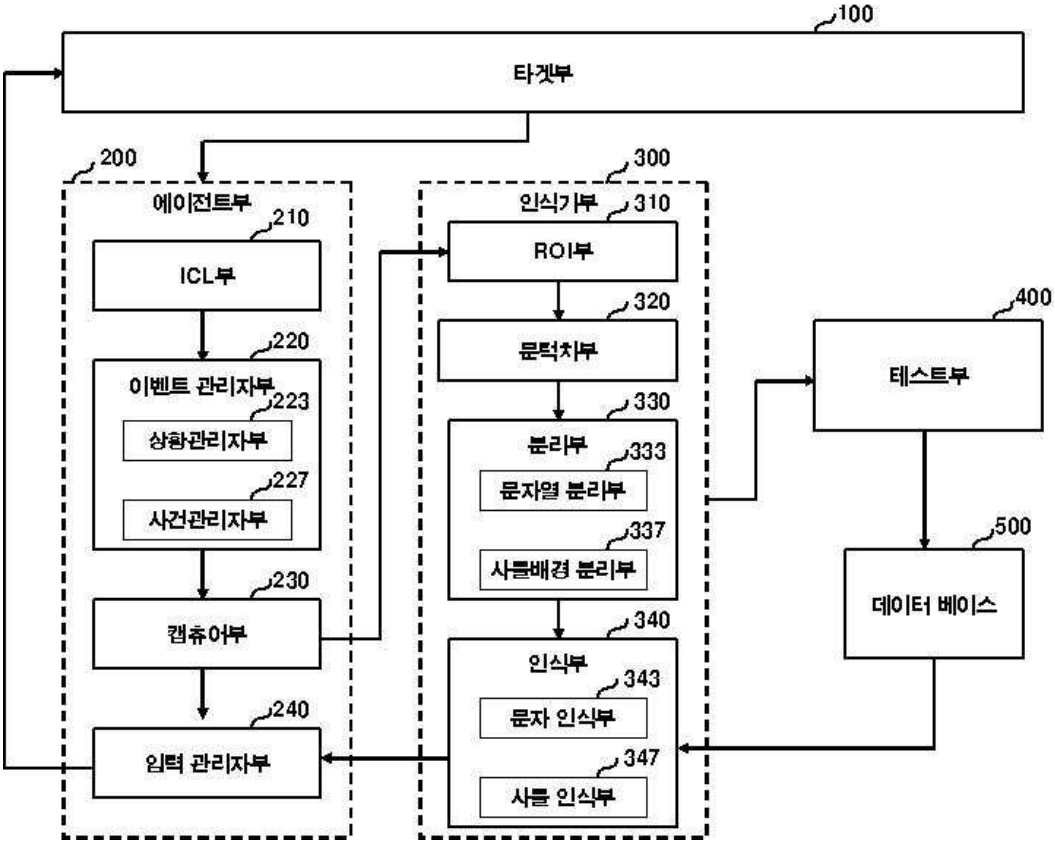
WDM(Windows Devices Model) 디바이스 드라이버를 통하여 입력할 것인가를 확인하는 WDM 드라이버 입력여부 확인단계;

WDM 디바이스 드라이버를 통한 입력이라면, 디바이스로 패킷을 전송하는 패킷전송단계;

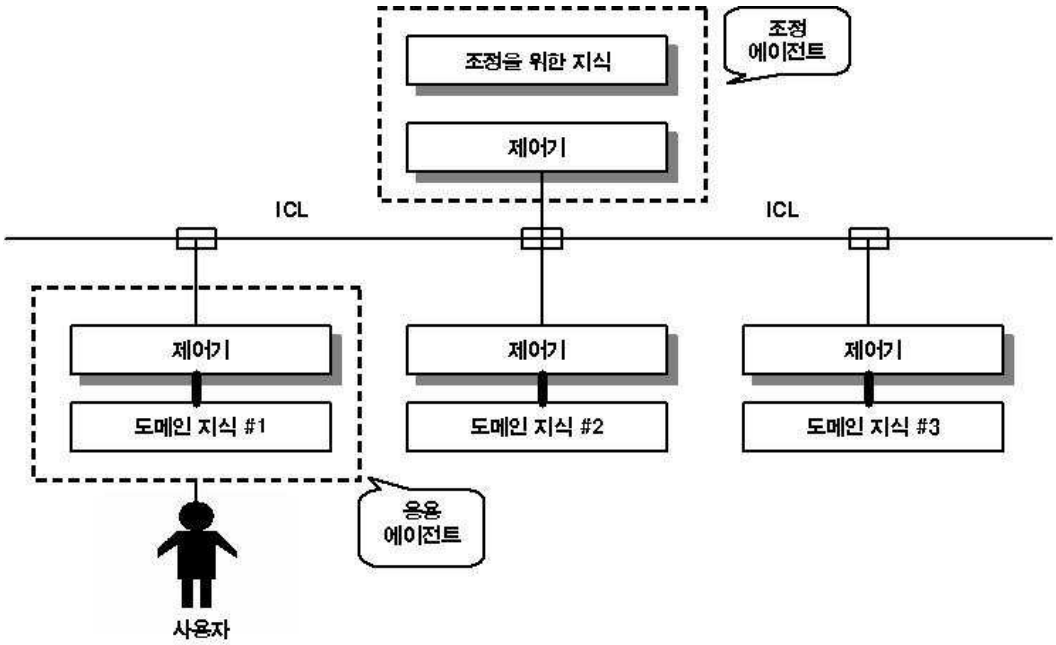
WDM 디바이스 드라이버를 통한 입력이 아니라면, 가상 키코드로 입력하는 가상 키코드 입력단계;를 적어도 포함하는 것을 특징으로 하는 영상인식을 이용한 에이전트 시스템의 동작방법.

도면

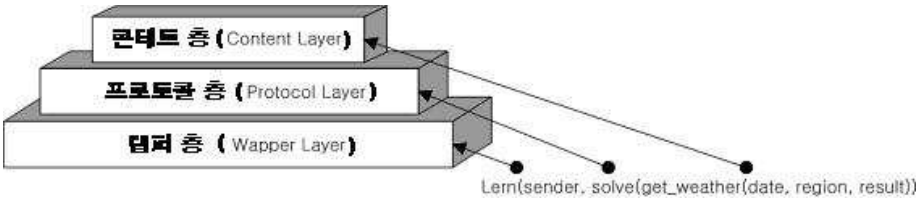
도면1


100
타겟부
200
에이전트부
210
ICL부
220
이벤트 관리자부
223
상황관리자부
227
사건관리자부
230
캡슈어부
240
입력 관리자부
300
인식기부
310
ROI부
320
문턱치부
330
분리부
333
문자열 분리부
337
사물배경 분리부
340
인식부
343
문자 인식부
347
사물 인식부
400
테스트부
500
데이터 베이스

도면2


조정을 위한 지식
제어기
조정
에이전트
ICL
ICL
제어기
도메인 지식 #1
제어기
도메인 지식 #2
제어기
도메인 지식 #3
응용
에이전트
사용자

도면3

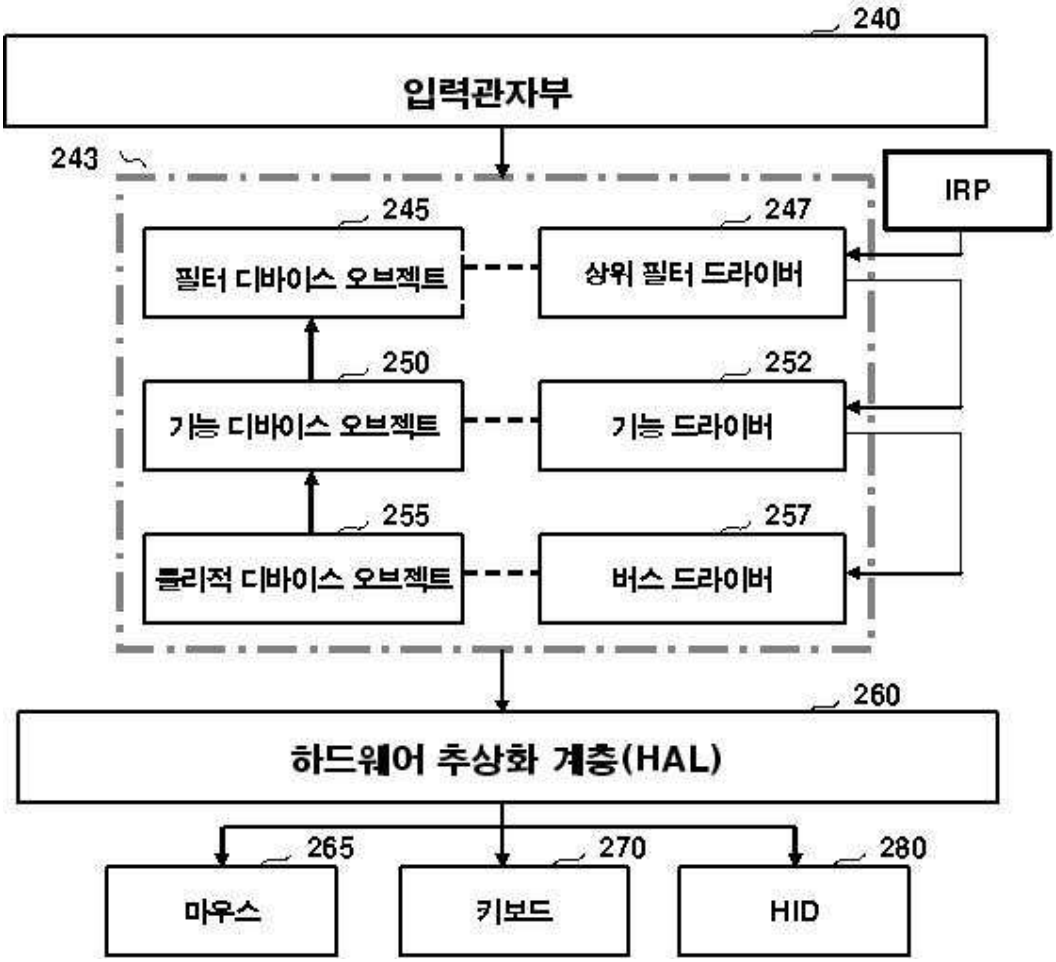


도면4

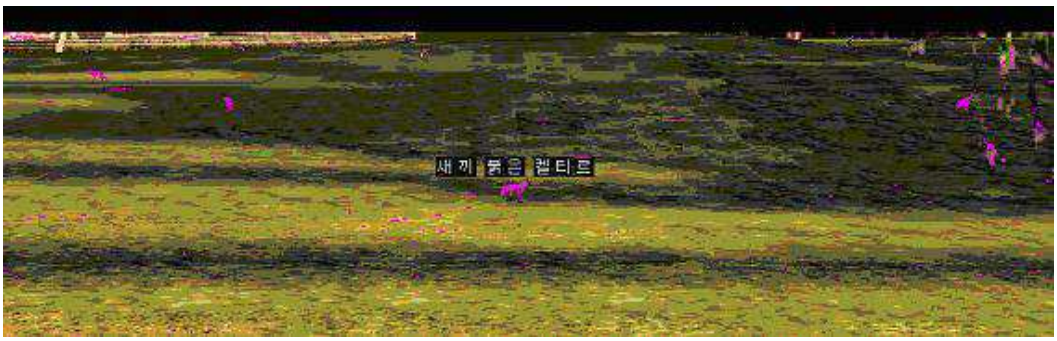
```
enum EventStatus {
        MS_LOGIN,           // 로그인 화면
        MS_IDLE = 1000,     // 쉬는 중
        MS_SEARCH,          // 적을 탐색 중
        MS_COMBAT,          // 전투 중
        MS_TOGGLE,          // 전리품 획득 중
        MS_TURNVISION,      // 시야 회전 중
        MS_FINDPATH,        // 길 이동 중
        MS_HEAL = 2000,     // 치료 중
};
```

도면5

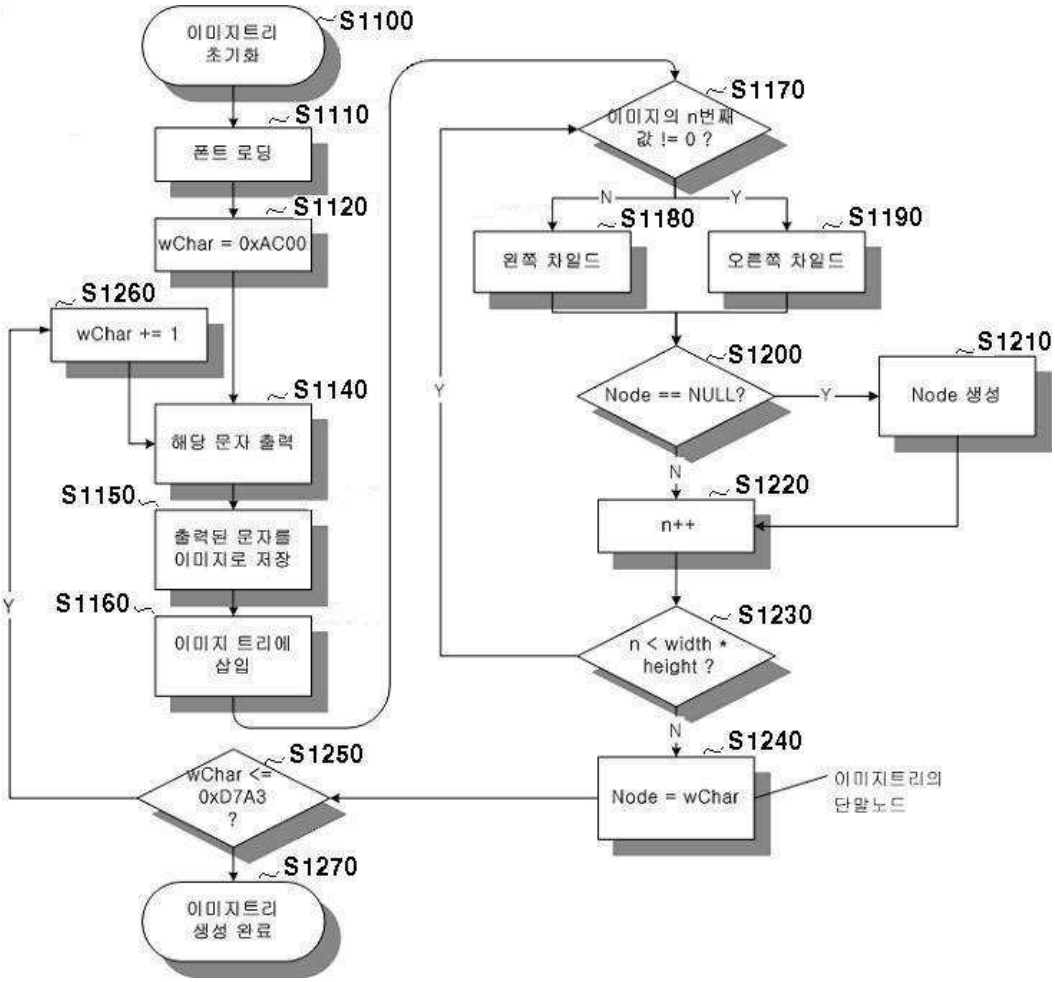
도면6


240
입력관자부
243
IRP
245
필터 디바이스 오브젝트
247
상위 필터 드라이버
250
기능 디바이스 오브젝트
252
기능 드라이버
255
물리적 디바이스 오브젝트
257
버스 드라이버
260
하드웨어 추상화 계층(HAL)
265
마우스
270
키보드
280
HID

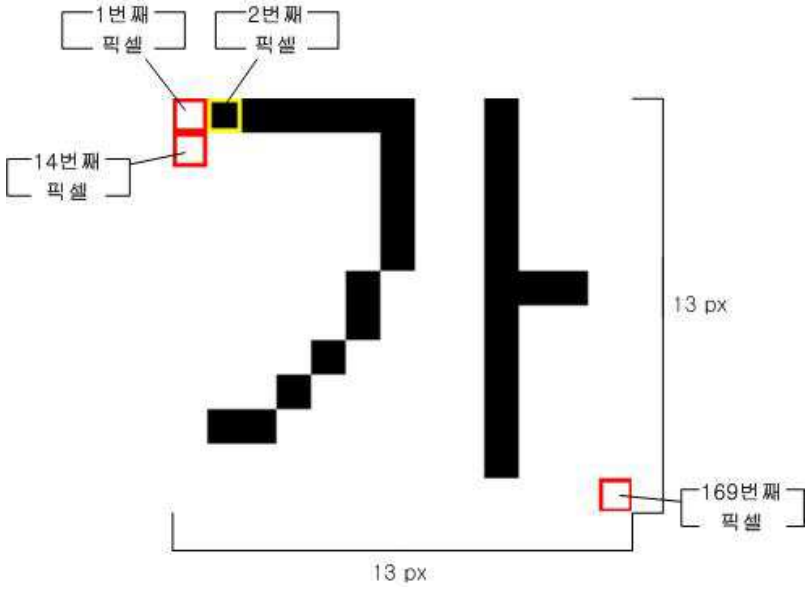
도면7



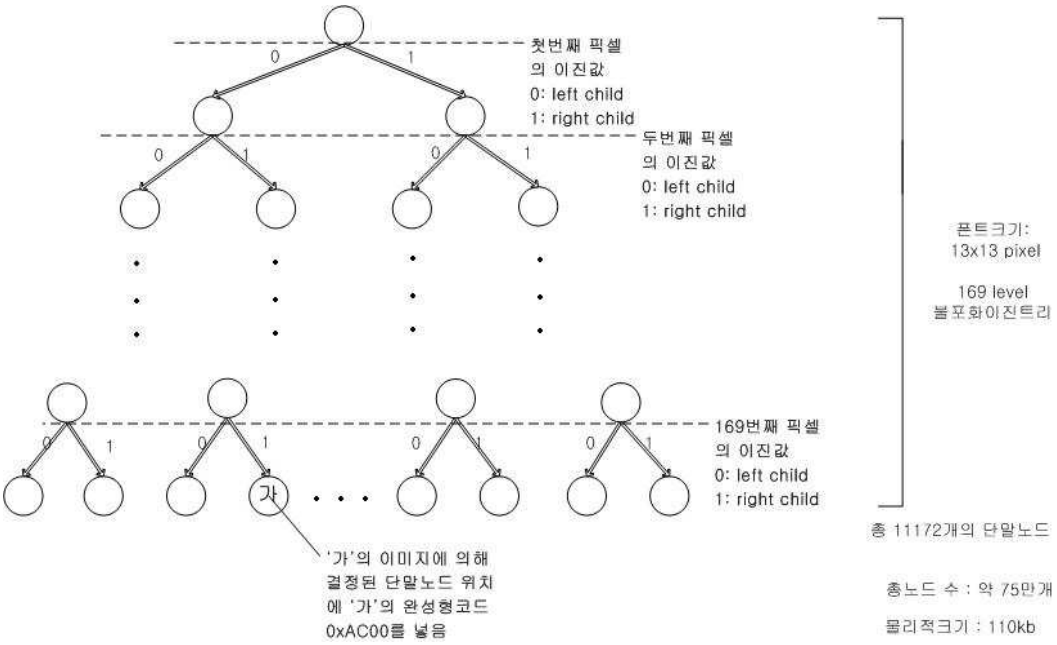
도면8


이미지트리 초기화
S1100
폰트 로딩
S1110
wChar = 0xAC00
S1120
wChar += 1
S1260
해당 문자 출력
S1140
출력된 문자를 이미지로 저장
S1150
이미지 트리에 삽입
S1160
wChar <= 0xD7A3 ?
S1250
이미지트리 생성 완료
S1270
이미지의 n번째 값 != 0 ?
S1170
N
Y
왼쪽 차일드
S1180
오른쪽 차일드
S1190
Node == NULL?
S1200
Node 생성
S1210
n++
S1220
n < width * height ?
S1230
Node = wChar
S1240
이미지트리의 단말노드

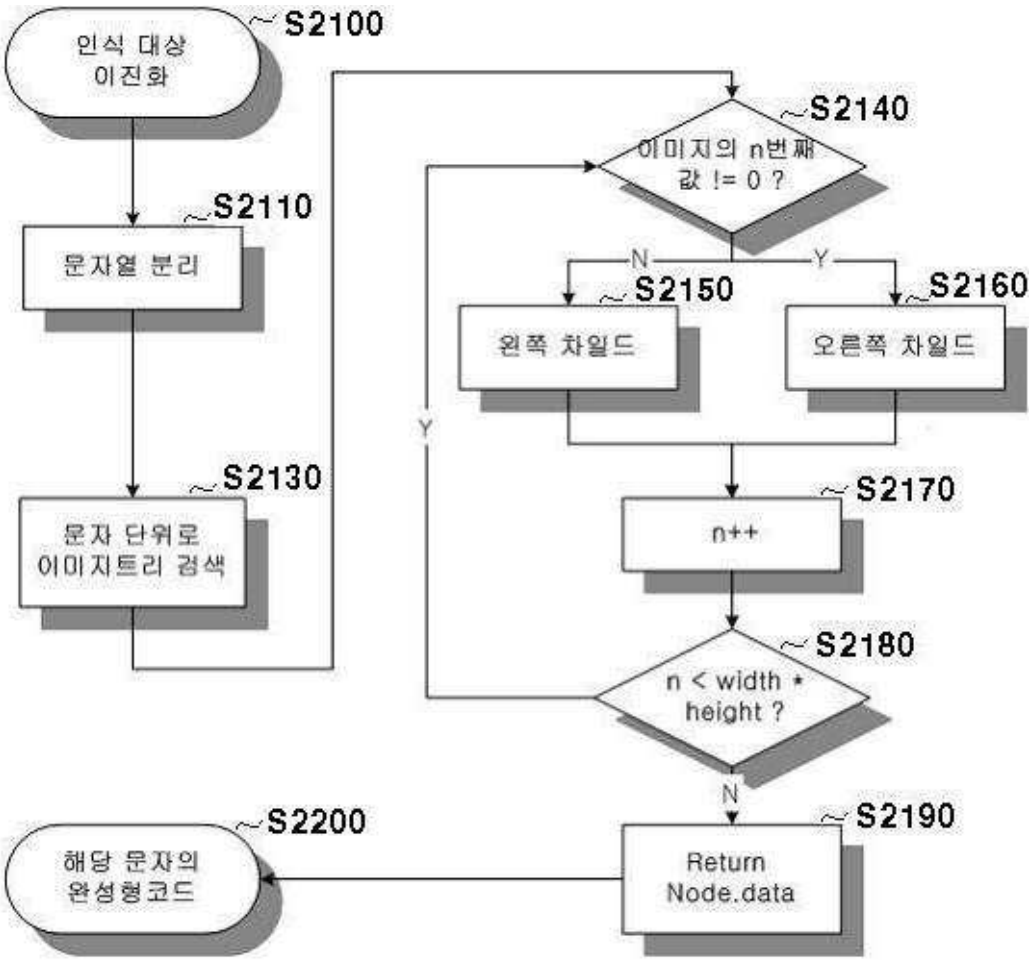
도면9a


1번째 픽셀
2번째 픽셀
14번째 픽셀
13 px
169번째 픽셀
13 px

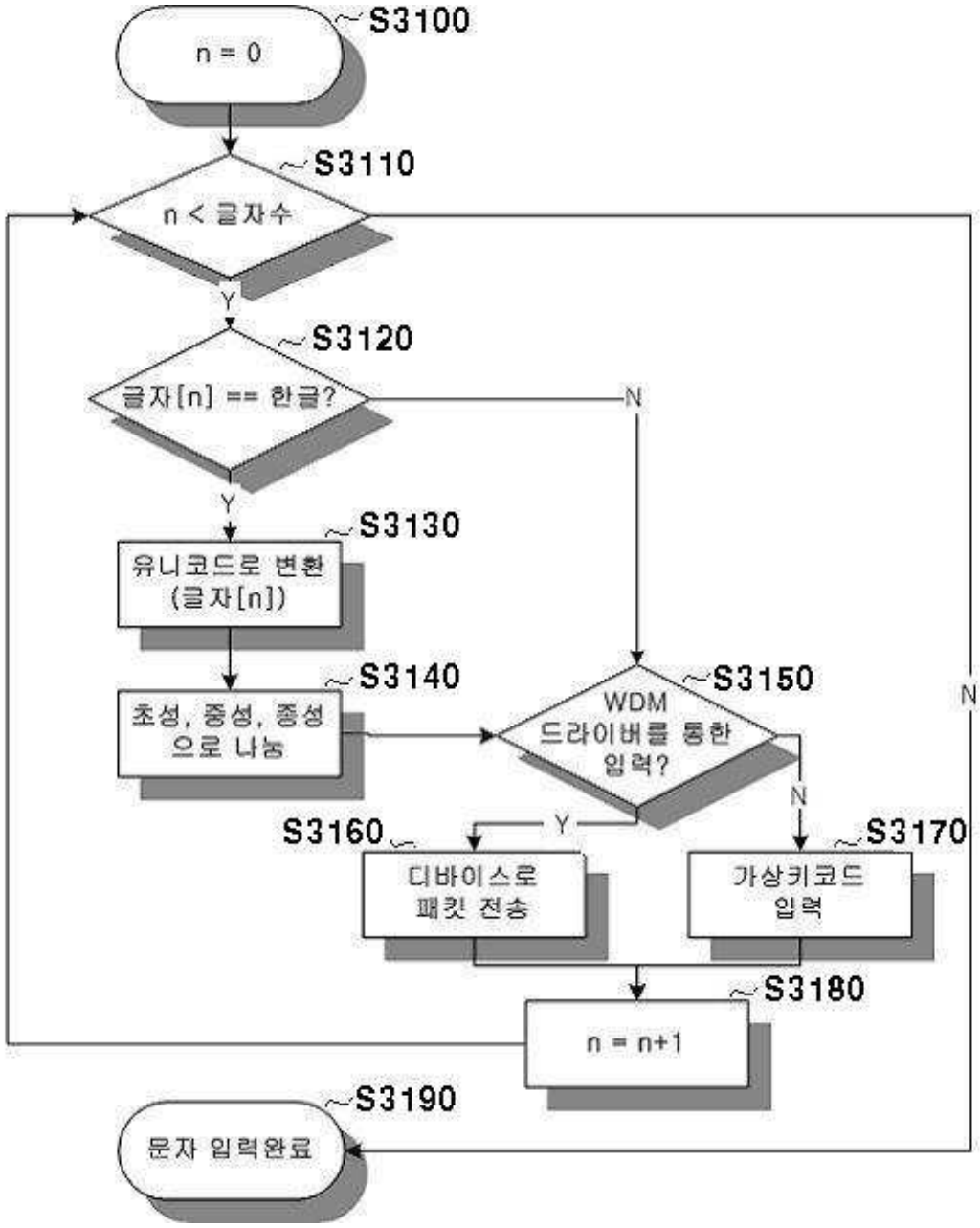
도면9b


첫번째 픽셀
의 이진값
0: left child
1: right child
두번째 픽셀
의 이진값
0: left child
1: right child
169번째 픽셀
의 이진값
0: left child
1: right child
가
'가'의 이미지에 의해
결정된 단말노드 위치
에 '가'의 완성형코드
0xAC00를 넣음
폰트크기:
13x13 pixel
169 level
불포화이진트리
총 11172개의 단말노드
총노드 수 : 약 75만개
물리적크기 : 110kb

도면10


인식 대상
이진화
S2100
문자열 분리
S2110
문자 단위로
이미지트리 검색
S2130
이미지의 n번째
값 != 0 ?
S2140
N
Y
왼쪽 차일드
S2150
오른쪽 차일드
S2160
n++
S2170
n < width *
height ?
S2180
Y
N
Return
Node.data
S2190
해당 문자의
완성형코드
S2200

도면11


S3100
n = 0
S3110
n < 글자수
Y
N
S3120
글자[n] == 한글?
Y
N
S3130
유니코드로 변환
(글자[n])
S3140
초성, 중성, 종성
으로 나눔
S3150
WDM
드라이버를 통한
입력?
Y
N
S3160
디바이스로
패킷 전송
S3170
가상키코드
입력
S3180
n = n+1
N
S3190
문자 입력완료