(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(51) Int. Cl.[6]
G06F 15/70

(45) 공고일자 2000년02월01일
(11) 등록번호 10-0242785
(24) 등록일자 1999년11월12일

(21) 출원번호 10-1996-0013349
(22) 출원일자 1996년04월27일
(65) 공개번호 특1996-0038650
(43) 공개일자 1996년11월21일
(30) 우선권주장 95-106194 1995년04월28일 일본(JP)

(73) 특허권자 캐논 가부시키가이샤 미다라이 후지오
일본 도꾜도 오오따꾸 시모마루꼬 3쪼메 30방 2고
(72) 발명자 사까구찌 가쯔히꼬
일본국 도꾜도 오오따꾸 시모마루꼬 3쪼메 30방 2고 캐논 가부시키가이샤 내
아라이 쯔네까즈
일본국 도꾜도 오오따꾸 시모마루꼬 3쪼메 30방 2고 캐논 가부시키가이샤 내
다까스 에이지
일본국 도꾜도 오오따꾸 시모마루꼬 3쪼메 30방 2고 캐논 가부시키가이샤 내
요시이 히로또
일본국 도꾜도 오오따꾸 시모마루꼬 3쪼메 30방 2고 캐논 가부시키가이샤 내
(74) 대리인 구영창, 장수길, 주성민

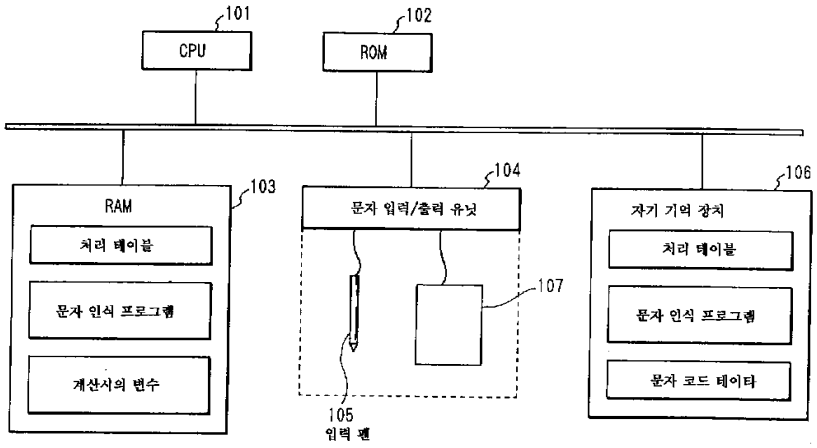
심사관 : 이정숙

(54) 정보 처리 방법 및 장치

요약

본 발명은 배치 처리에서 처리될 문자 또는 스트로크들로 이뤄진 심볼 및 문자 또는 심볼에 대해 실행될 처리를 선택하게 하는 제스쳐 스트로크를 입력시킬 때 입력 스트로크들의 그룹으로부터 제스쳐 스트로크를 구분하고 추출하여, 추출해낸 제스쳐 스트로크를 제외한 모든 스트로크들에 대해 제스쳐 스트로크에 의해 선택된 처리를 실행함으로써 개별 시간 간격으로 문자 또는 심볼 및 제스쳐를 입력시킬 필요가 없게 된다. 이는 동작성을 향상시킨다.

대표도



명세서

[발명의 명칭]

정보 처리 방법 및 장치

[도면의 간단한 설명]

제1도는 본 발명의 제1 실시예에 따른 데이타 처리 장치의 구조를 나타내는 블럭도.

제2도는 본 발명의 제1 실시예에 따른 일반적 데이타 처리를 설명하는 블럭도.

제3도는 특정한 처리 실행부에서 실행되는 처리를 설명하는 다이어그램.

제4도는 수기(handwriting) 변환 처리를 설명하는 다이어그램.

제5(a)도 및 제5(b)는, 특정 스트로크 및 스트로크된 문자가 함께 입력되고 입력된 문자에 대하여 특정 스트로크에 대응하는 처리를 실행한 결과로서 얻어진 샘플을 각각 예시한 도면.

제6도는 문자 인식 처리의 처리 단계를 나타낸 플로우 챠트.

제7(a)도 및 제7(b)도는 본 발명의 제2 실시예에 사용된 특정 스트로크를 각각 나타내는 도면.

제8도는 본 발명의 제2 실시예에 따른 문자 인식 처리의 처리 단계를 부분적으로 나타내는 플로우 챠트.

제9도는 본 발명의 제1 실시예에 사용된 내부 처리 데이타의 예를 나타내는 다이어그램.

제10도는 본 발명의 제2 실시예에 사용된 내부 처리 데이타의 예를 나타내는 다이어그램.

* 도면의 주요부분에 대한 부호의 설명

101 : CPU	102 : ROM
103 : RAM	104 : 문자 입력/출력 유닛
105 : 입력 펜	106 : 자기 기억 장치
107 : 데이타 입력/출력 패널	201 : 특정 스트로크 판별부
202 : 특정 스트로크 추출부	203 : 특정 처리 실행부
204 : 처리 테이블	205 : 특정 처리부

[발명의 상세한 설명]

본 발명은 루커스(locus) 정보로서 입력된 데이타에 포함된 각종의 데이타를 분리하여 각각 분리된 데이타에 알맞는 각각의 처리를 실행할 수 있는 정보 처리 방법 및 장치에 관한 것이다.

또한, 본 발명은 데이타 처리 방법 및 장치에 관한 것으로, 특히 좌표 지시 수단으로서 펜 및 디지타이저를 사용하여 입력 루커스를 조작 코멘트로서 인식하거나, 혹은 입력 루커스를 인식하여 문자 인식을 실행하는 수기 문자 인식 방법 및 장치에 관한 것이다.

수기 문자 인식을 실행하는 종래의 장치에서는, 스트로크 데이타를 인식 처리시에 입력될 데이타로서 사용하여 하나의 문자에 대한 문자 인식을 실행하기 위해 스트로크 데이타의 하나의 문자를 추출한다.

이러한 수기 문자 인식은 장치에 입력될 스트로크 데이타가 인식될 문자로만 구성되어 있다는 가정 하에서 실행될 수 있다. 다시 말하면, 문자 인식 처리는 문자 인식만을 대상으로 한 처리이다.

펜 입력용의 대다수의 전자 장치는 입력 모드로 문자 또는 루커스 정보 뿐만 아니라, 펜을 사용하는 조작 코맨드를 입력할 수 있는 제스쳐 코맨드 입력 수단(gesture command input means)을 포함한다.

종래, 이러한 제스쳐 코맨드 각각은 각 코맨드에 대하여 미리 정해진 입력 루커스를 갖고 있기 때문에, 미리 정해진 입력 루커스를 인식할 때 대응하는 코맨드를 실행한다.

간단히 인식될 루커스 또는 문자 입력으로부터 제스쳐 코맨드를 명확히 판별해내기 위해서는, 제스쳐 코맨드를 다른 모드로 실행하거나 혹은 제스쳐 입력 및 문자 입력의 포맷을 개별적으로 한정할 수 있다. 상술한 바와 같이, 종래의 장치는 문자 입력 처리시에 문자만이 입력되고, 문자 인식의 최종 작업후에 다양한 형태의 코맨드가 제스쳐 코맨드로서 입력되도록 다른 방식에 의해 문자 입력 및 코맨드 입력을 실행하도록 설계되어 있다.

그러나, 이러한 장치에서는, 문자 입력 동안 입력 문자의 형태를 특정화하거나 혹은 인식 패턴을 판독하기 위해 문자 입력 동안 문자 인식 처리의 제어 혹은 다른 코맨드의 실행을 실행할 필요가 있다.

그럼에도 불구하고, 종래의 장치에서는 조작자 혹은 사용자가 분리해서 문자인식을 위한 문자 입력 및 코맨드 실행을 위한 제스쳐 입력을 행하도록 하고 있으므로 사용자에게 번거로움을 제공한다.

또한, 이러한 제어 수단은 문자 인식용의 문자 인식 엔진의 형태에 의존하며, 이러한 인식 처리 엔진이 일본어-한자 변환용의 종래의 프론트-앤드 프로세서(front-end processor)와 같이 시스템마다 재배치가능하면, 모든 인식 제어 수단을 시스템 속에 미리 내장하는 것은 불가능하다.

본 발명은 상술한 환경 하에서 이루어진 것이며, 본 발명의 목적은 데이타에 대해 소정의 코맨드에 대응하는 처리를 실행하도록 데이타 및 코맨드를 함께 입력하는 경우라도, 데이타 및 데이타에 대한 소정의 코맨드를 개별적으로 인식할 수 있는 데이타 처리 바업 및 장치를 제공하는 것이다.

상기한 목적을 달성하기 위해, 본 발명에 따른 화상 입력 방법은,

제1 스트로크를 입력하는 제1 스트로크 입력 단계,

제2 스트로크를 입력하는 제2 스트로크 입력 단계, 및

제1 스트로크 및 소정의 스트로크가 서로 일치할 때, 제2 스트로크에 대한 소정의 처리를 실행하는 제2 스트로크 처리 단계를 포함한다.

본 발명의 다른 특징은 화상 입력 장치를 제공하는 것인데, 이 화상 입력 장치는,

제1 스트로크를 입력하기 위한 제1 스트로크 입력 수단,

제2 스트로크를 입력하기 위한 제2 스트로크 입력 수단, 및

제1 스트로크 및 소정의 스트로크가 서로 일치할 때, 제2 스트로크에 대한 소정의 처리를 실행하기 위한 제2 스트로크 처리 수단을 구비하며,

상기 제1 및 제2 스트로크는 각각 상기 제1 스트로크 입력 수단 및 제2 스트로크 입력 수단에 의해 동일한 입력 위치에 입력된다.

본 발명의 또 다른 특징은 정보 처리 장치를 제공하는 것인데, 이 정보 처리 장치는,

스트로크를 입력하기 위한 입력 수단,

모든 입력 스트로크로부터 소정의 스트로크를 추출하기 위한 추출 수단,

상기 추출 수단에서 추출된 스트로크를 제외한 모든 스트로크에 기초하여 대응하는 심볼을 선택하기 위한 선택 수단, 및

상기 심볼에 대하여 소정의 스트로크에 대응하는 소정의 처리를 실행하기 위하여 심볼 처리 수단을 구비한다.

이제부터 도면을 참조로 하여 상술한 목적, 및 다른 목적 그리고 장점 및 보충 설명을 개시하고자 한다.

[제1 실시예]

본 발명의 제1 실시예에 따른 데이타 처리 장치는 특정 스트로크가 입력 스트로크에 포함되어 있는지를 판단하기 위한 특정-스트로크 판별부, 입력 스트로크로부터 특정 스트로크를 추출하기 위한 추출부, 및 특정 스트로크의 형태에 기초한 처리를 실행하기 위한 특정-처리 실행부를 구비하여, 입력 문자에 특정 스트로크를 기입하여 문자 인식을 실행함과 동시에 입력된 문자에 대한 다양한 처리를 실행할 수 있는 구조를 갖춘 것을 특징으로 한다.

제1도는 본 발명의 실시예에 따른 수기 문자 인식 장치의 기본 구조를 나타낸다.

제1도에서, CPU(101)는 시스템 제어 및 동작을 실행하기 위한 중앙 처리 장치이고, ROM(102)에는 시스템 작동시에 필요한 초기 프로그램이 격납되어 있다. RAM(103)은 입력 및 출력이 가능한 기억 장치이며, 자기 기억 장치에 기억되어 있는 처리 테이블 및 문자 인식 프로그램, 각 처리 시간 등에서 발생된 데이타를 일시적으로 기억한다.

문자 입력/출력 유닛(104)은 데이타 입력/출력 패널(107) 상의 데이타 입력 및 출력을 제어한다. 데이타 입력은 입력 펜(105)을 사용하여 데이타 입력/출력 패널(107) 상에 손으로 문자를 기록함으로써 실행된다. 문자 입력용으로서 디지타이저 혹은 터치 펜 등의 좌표 입력 수단이 사용될 수 있으며, 문자 출력용으로서는 액정 디스플레이와 같이 화상 데이타 및 패턴화된 문자 데이타가 표시될 수 있는 디스플레이가 사용될 수 있다.

자기 기억 장치(106)는 특정 스트로크에 대응하는 특정한 처리를 실행하기 위한 처리 테이블, 문자 인식 프로그램, 후술할 처리 단계에서 사용될 제어 프로그램, 및 문자-코드 테이블의 데이타를 기억한다. 이들 데이타 혹은 프로그램은 RAM(103)으로 전달되고, 이 RAM(103)으로부터 CPU(101)가 이들 데이타 또는 프로그램을 판독하여 실행한다.

자기 기억 장치(106)는 장치 본체에 착탈가능하거나, 자기 기억 장치와는 다른 형태의 기억 장치로 대체될 수 있다.

또한, RAM(103)에 기억될 모든 종류의 제어 프로그램은 공중망 혹은 LAN과 같은 통신 수단을 통해 다른 장치로부터 판독될 수 있다.

제2도는 본 발명의 특징을 가장 잘 표현하고 있는 본 발명의 실시예에 따른 처리 구조를 나타내고 있다. 제2도에 도시된 처리는 펜 컴퓨터와 같은 펜 입력 장치를 사용하는 인식 처리에 관한 것인데, 여기서는 문자 입력 프레임에 기록된 1 문자분의 스트로크를 입력 데이타로서 사용하여 실행한다. 1 문자분의 스트로크 데이타가 전부 입력되었는지를 판단하는 데에 있어서는, 예를 들면, 펜이 프레임으로부터 제거된 이래로 소정의 기간(예를 들면, 1초) 동안 펜이 문자 입력 프레임에 놓이지 않은 시간, 혹은 펜 입력이 다음 문자 입력 프레임에서 개시된 시간에 1 문자분의 스트로크 데이타가 전부 입력되었는지를 판단하는 것과 같은 여러가지 방법이 사용될 수 있다.

이하, 제2도와 관련한 실시예에 따른 인식 처리에 대한 설명을 한다. 제2도의 각 부는 문자 인식 프로그램에서 RAM(103) 내에 기억되어 있고 처리 실행을 위해 CPU(101)에 판독되어 제어 프로그램으로서 자기 기억 장치(106)로부터 각각 로드된 소프트웨어 성분이다.

제2도에서, 특정-스트로크 판별부(201)는 하나 또는 복수의 특정-스트로크가 1 문자분의 스트로크 데이타에 존재하는지를 판별한다. 판별 결과, 특정 스트로크가 없으면, 정상적인 1 문자 인식 처리와 동일하게 인식 프로그램(206)을 실행하여 인식 처리를 종료한다.

특정 스트로크의 판별은 스트로크 형태 및 크기를 고려하면서, 미리 결정된 특정-스트로크와 입력 스트로크의 매칭 거리를 통해 실행된다. 매칭 처리는 온-라인 문자 인식용으로서 일반적으로 사용되는 벡터화 매칭 기술 혹은 특성점 매칭 기술과 같은 공지 기술에 의해 실행된다. 입력 스트로크 데이타가 입력된 순서로 배열되는 시계열 데이타이므로, 판별부는 판별 정보와 함께 특정 스트로크 전에 어느 정도의 스트로크가 있었는지에 관한 정보를 다음 처리부로 출력한다. 따라서, 특정 스트로크는 처리 테이블(204)에 기억된 특정 스트로크의 형태에 따라 판별된다.

특정-스트로크 추출부(202)는 모든 입력 스트로크로부터 특정 스트로크 판별부(201)에서 판별된 특정 스

트로크를 추출한다. 특정 스트로크에 대응하는 스트로크 순번이 특정-스트로크 판별부(201)의 출력으로부터 밝혀지므로, 특정 스트로크를 입력 스트로크의 스트링으로부터 추출하여, 입력 스트로크를 특정 스트로크를 배제한 문자만의 입력 스트로크 데이타의 스트링으로 재구성할 수 있다.

특정-처리 실행부(203)는 처리 테이블(204)에 기억된 정보에 기초하여 특정-스트로크 판별부(201)에서 판별된 특정 스트로크에 대응하는 특정 처리를 실행한다. 처리 테이블(204)은 미리 결정된 특정-스트로크 형태와 각각의 특정 스트로크에 대응하여 실행될 특정 처리를 기억한다.

특정 처리부(205)는 특정-처리 실행부(203)에 의해 실행되고, 이 처리의 내용은 판별된 특정-스트로크 형태에 의존함으로, 각각의 특정 스트로크에 따라 여러가지 처리가 실행된다. 예를 들면, 인식 프로그램(206)을 호출하여 인식 결과를 얻은 후에 적정 프로그램을 실행하는 것, 인식 처리(206)의 동작을 제어하는 인식 처리 플래그 등을 설정하여 인식 처리의 호출 또는 종료를 제어하는 것, 및 등록 인식 등의 관련 어플리케이션의 호출 또는 종료를 제어하는 것 등 여러가지 호출 처리가 가능하다.

동작예로서, 특정 처리부(205)는 입력 문자를 수기 문자로 변환시키기 위한 수기-변환 처리부(210), 입력 문자를 소문자 패턴으로 변환시키기 위한 소문자 변환 처리부(211), 및 입력 스트로크를 그대로 등록하기 위한 문자-등록 처리부(212)를 포함한다.

이어서, 문자-인식 처리부(206)는 입력 스트로크로부터 문자를 인식한다.

다음으로, 제3도는 처리 테이블(204)의 내용을 쉽게 이해할 수 있게 도시한 도면이다. 제3도에서, 좌변은 특정 스트로크의 형태이고, 우변은 좌변에 따라 실행될 특정 처리의 내용 및 각 처리 프로그램의 호출 어드레스를 포함한다.

실제로는, 제2도의 특정-스트로크 판별부에 매칭 딕셔너리와 같이, 특정 스트로크의 형태에 따른 스트로크를 매칭하기 위한 수단이 기억된다. 또한, 실행될 특정 처리부(205)의 내용은 처리 실행을 위한 호출 어드레스를 포함한다. 후술할 제9도의 도면 참조 번호(903)에는 각 특정 처리를 위한 호출 어드레스의 설정이 도시되어 있다.

이러한 처리 테이블을 메모리에 기입하므로, 테이블의 내용을 재기입하면 미리 결정된 특정-스트로크 및 각 처리의 내용을 확장하는 일도 가능하다.

제3도를 참조하여 더 상세하게 설명한다.

“0”와 같은 특정 스트로크가 1 문자분의 스트로크 중에 존재하는 경우, 추출된 특정 스트로크를 제외한 모든 문자 스트로크로부터 문자를 인식하여 수기 패턴으로서 출력되도록 코드로 변환한다. 입력 문자를 변환할 수 없으면 그대로 출력한다. 문자를 수기 패턴으로 변환시키기 위한 수기-변환 처리부(210)는 제2도에서 설명한 특정 처리부(205)에 포함된다. 수기-변환 처리부(210)는 특정-처리 실행부(203)에 의해 선택적으로 실행되며, 여기서는 인식 결과로부터 얻어진 가능한 문자 코드를 수신하도록 인식 처리부(206)를 호출한다. 그 다음, 수신된 문자 코드는 수기 패턴의 코드로 변환된다.

“△”와 같은 심볼이 입력되면, 수기 변환 처리에서 언급한 바와 동일하게, 소문자 변환 처리가 실행된다. 다시 말해서, 인식 결과로 얻어진 문자 코드가 소문자 패턴의 코드로 변환된다. 대체적으로, 문자 스트로크를 구성하는 좌표점이 스케일 다운(scale down)되도록 인식 처리부(206)에서 인식되기 전에 모든 스트로크에 대해 좌표 변환이 실행되면, 인식 처리부에 대한 입력으로서 최종 데이타를 사용함에 의해 코드 변환없이 인식 처리를 실행할 수 있다.

최후로 “¥”와 같은 심볼이 입력되면, 상술한 바와 동일한 방법으로 문자-등록 처리가 실행된다. 이 경우는, 입력 문자 스트로크가 문자 등록을 위한 입력으로서 사용하므로, 특성점 데이타와 같은 데이타가 키보드에 의해 문자 코드 입력에 따라 발생되어 매칭 딕셔너리 속에 등록된다.

상술한 바에 의해, 특정 처리부(205)에서 여러가지 처리를 실행하는 것이 가능하다. 다시 말해서, 특정 처리부(205)는 종래의 프로그래밍 기술로 실현가능한 처리의 어떤 처리 내용도 실현할 수 있고, 그 외의 다른 처리 내용도 포함할 수 있다. 예를 들면, 특정 처리부(205)는 일본어의 히라가나로서 입력된 문자를 한자로 변환하기 위한 변화 처리를 포함할 수 있다.

제5(a)도 및 제5(b)도는 조작자가 문자 입력 처리시에 기록한 특정 스트로크가 있는 문자 입력 패턴 및 문자 인식 결과를 각각 나타낸 도면이다. 각 도면에서, 좌측은 문자 프레임 내에 기록된 특정 스트로크가 있는 문자 패턴의 패스(path)를 나타내며, 우측은 특정 스트로크를 지시한 특정 처리를 통해 인식된 디스플레이될 최종 패턴의 예를 도시하고 있다.

제5(a)도에서, “0”와 같은 스트로크가 문자와 함께 기입되고, 수기-변환 처리가 실행된다. 이 경우에는, 제2도의 특정-스트로크 판별부에 대한 설명에서와 같이, 스트로크 형태 및 크기를 고려하여 특정 스트로크의 판별이 행해지므로, 스트로크 “0” 및, 예를 들면 “Å”용으로 사용된 “°”와 같은 문자 엘리먼트가 혼합되는 것이 방지될 수 있다. 또한, 특정 스트로크 판별 처리에서는 모든 스트로크가 판별되므로 스트로크 “0”는 입력 처리시에는 언제나 기입될 수 있다.

이어서, 상술한 바와 같이 판별된 특정 스트로크 “0”가 추출되므로, 문자 “T”의 스트로크에 대한 수기 변환 처리가 실행된다.

수기 변환 처리에서는 먼저 입력 스트로크의 인식이 실행된다. 인식 결과, 문자 “T”에 대응하는 코드가 문자-코드 변환을 통해 수기 문자 “T”를 출력하도록 가능한 코드로서 선택된다.

제4도는 상술한 수기 변환 처리의 이해를 돕기 위해 수기-변환 처리부(210) 및 인식 처리부(206)를 부분 도시한 플로우 챠트이다.

한편, 제5(b)도는 소문자 변환의 경우로서, 제5(a)도의 수기 변환 처리에서 설명한 것과 동일한 방식으로 특정 스트로크가 추출되어 인식된 후에, 가능한 문자 “C”에 대한 소문자 코드 변환이 실행되어 소문자 “C”가 출력되는 것을 나타내고 있다.

이 경우에, 문자 스트로크데이타의 좌표 자체가 문자 프레임에 비하여 스케일 다운될 수 있도록 스트로크 "△"를 추출한 후에 모든 스트로크에 대한 좌표 변환을 행하면 최종의 소문자가 정상적인 인식 처리를 통해 얻어질 수 있다.

제6도는 상술한 실시예에 따른 문자-인식 처리를 실행하는 처리 단계를 나타낸다. 이하 도면을 참조하여 처리 동작의 절차를 설명한다.

먼저, 스텝(S1)에서는, 1 문자분의 n개의 스트로크를 입력순으로 스트로크 배열 stk(n) : (n = 1,2, 3,…)로 격납한다. 제9도의 도면 참조 번호(901)에 n = 3일 때의 스트로크 배열로 격납된 스트로크의 일례가 도시되어 있다.

이어서, 스텝(S2)에서는, 스트로크의 순번을 나타내는 제어 변수 sct와, 처리테이블(이하, 이 테이블을 "ss"로 간단히 약칭한다)에 격납된 특정 스트로크 배열 "stk"의 위치를 지시하는 인덱스 변수인 "ssct"를 "1"로 초기화한다.

스텝(S3)에서는, 하나의 입력 스트로크 stk(sct)와 처리 테이블(204) 내의 특정 스트로크 ss(ssct), stk와의 매칭을 행하여 동일하다고 판단된 경우에는 스텝(S4)로 이행하고, 다른 경우에는 스텝(S7)로 이행한다. 제9도의 도면 참조 번호(902)에 특정 스트로크 ss(ssct).stk에 격납되어 있는 스트로크 일례가 도시되어 있다.

이어서 스텝(S7) 및 그 이후의 스텝에 대하여 설명한다.

스텝(S7)에서는, 처리 테이블(204)의 인덱스 변수 "ssct"를 1만큼 시프트 한다(ssct = ssct+1). 그 다음, 스텝(S8)에서는 처리 테이블의 최후까지 인덱스가 시프트 되는지에 대해 판단한다. 최후를 초과하지 않은 경우, 새로운 인덱스, 즉 ssct = ssct+1에 의해 스텝(S3)의 스트로크 매칭 처리로 복귀한다. 처리 테이블(204)의 최후까지 매칭을 종료한 경우는 스텝(S9)로 이행한다.

스텝(S9)에서는, 처리 테이블(204)으 인덱스를 최초로 되돌려(ssct = 1) 매칭을 행할 입력 스트로크를 다음으로 시프트하기 위해 입력 스트로크의 순번을 표시하는 변수 sct를 1만큼 카운트 업시킨다.

스텝S(10)에서는, 이 "sct"가 입력 스트로크의 갯수 N을 초과했는지를 체크하여 초과한 경우, 전체 스트로크를 검사한 것으로서 간주하고 스텝(S11)로 이행한다. 이것은 모든 입력 스트로크 중에 특정 스트로크가 없었던 것을 나타낸다. 반대로, 갯수 n을 초과하지 않은 경우 매칭 목적으로 새로운 스트로크를 사용하여 스텝(S3)로부터 스텝S(10)까지를 반복하도록 스텝(S3)으로 진행한다.

이러한 반복 동작에 의해, 전체의 입력 스트로크와 처리 테이블 내의 특정 스트로크의 매칭을 행한다. 따라서, 특정 스트로크는, 입력 스트로크 내의 어떤 순서에 있어서도 판별가능하다. 제9도는, 매칭 후에 특정 스트로크가 제거된 문자 인식 장치 대상이 되는 각 스트로크가 스트로크 배열 stk에 격납되어 있는 상태를 나타낸다.

마지막으로, 스텝(S11)에서는 1 문자 단위의 통상적인 문자 인식을 실행하고 문자 인식의 전 과정을 완료한다.

특정 스트로크가 최초 혹은 최후 등과 같이 소정의 순번으로 특정하게 입력되는 경우에는 본 실시예에 따른 처리 대신에 더 간단한 처리로 특정 스트로크의 판별을 실현할 수 있음을 이해할 수 있다.

스텝(S3)의 스트로크 매칭 처리를 참조하면, 특정 스트로크가 존재하는 경우 스텝(S4)로 이행한다.

스텝(S4)에서는, 특정 스트로크 "stk(sct)"가 입력 스트로크 배열 "stk(n)"으로 부터 추출되어 데이타 stk(1) 내지 stk(n-1)이 발생된다. 따라서, 스텝(S41)에서는 데이타 stk(1) 내지 stk(n-1)에 대한 문자 인식이 실행된다.

이어서, 스텝(S5)에서는 특정 스트로크에 대응하는 특정 처리를 실행하기 위해 처리 테이블 "ss(ssct)"의 처리 어드레스 "ss(ssct).adr"를 추출한다. 제9도는 특정 처리부(205)에서 각 처리를 호출하기 위한 처리 어드레스 "ss(ssct).adr" 세트의 예를 나타내는데, 여기서, xxxx, yyyy 및 zzzz는 각 처리부에 대한 어드레스를 나타낸다.

스텝(S6)에서는, 어드레스에 의해 지시된 특정 처리부가 호출된다. 이상으로 인식 처리를 종료한다. 마지막으로, 스텝(S12)에서 인식의 결과로서 얻어진 문자 코드를 출력하여 모든 처리를 종료한다.

상술한 동작시, 특정 스트로크가 문자 입력 처리 중에 문자와 함께 기입되는 경우, 특정 스트로크의 형태에 따라 특정 처리가 실행될 수 있다.

또한, 새로운 형태의 스트로크를 데이타 stk(1)부터 stk(n-1)까지 특정점을 발생시켜 매칭 딕셔너리에 등록할 수 있다.

[제2 실시예]

본 발명의 제1 실시예에 따른 문자 인식 방법이 특정 스트로크들의 예로서 "O" 또는 "△"와 같이 1 스트로크로 쓰기가능하고 모양과 크기가 문자 엘리먼트로부터 구별가능한 심볼을 사용하지만, 제2 실시예는 그런 제한이 불필요한 예시적 처리과정을 보여준다.

제7(a)도 및 제7(b)도는 제2 실시예에 사용된 예시적 특정 스트로크들을 보여준다. 제7(a)도의 스트로크들은 그 각각이 단일 원에 상응하는 두개의 스트로크로 씌여진 이중 원이다.

제7(b)도의 스트로크는 우측에서 좌측으로 그려져서 특정 스트로크로서 간주되는 크로스 바(cross bar)이다. 환언하면, 문자 엘리먼트로서의 하이픈(hyphen) "-"는 보통 좌측에서 우측으로 그려지기 때문에 이 크로스바는 하이픈과 같은 문자 엘리먼트로부터 구별될 수 있다.

상기와 같이 형태 구별에 어려움을 갖는 특정 스트로크들은 스트로크의 입력방향과 같은 정보를 사용하여

구별될 수 있다. 이는 제2도의 특정-스트로크 판별부(201)에서 사용된 매칭 방법에 의존한다. 상기 설명한 대로 온-라인 문자 인식에 대한 공지된 매칭 방법은 본 발명에서 사용될 수 있고 특성점 매칭 방법만이 사용된다면 매칭은 스트로크 방향에 관계없이 수행될 수 있다. 벡터화 매칭 방법이 사용된다면, 스트로크 방향을 매칭 오브젝트(object)로 사용하여 제7(b)도에 제시된 형태를 갖는 스트로크들도 입력 방향에 따라서 문자 엘리먼트들로부터 구별될 수 있다.

스트로크들이 문자 엘리먼트들로부터 구별될 수만 있다면 임의 유형의 특정 스트로크들이 채택될 수 있음을 이해할 것이다.

제7(a)도에 도시된 다수의 스트로크들로 구성된 심볼을 구별할 때, 스트로크 매칭 방법 자체는 이미 언급한 것과 동일하나 스트로크 특정화 및 스트로크 추출 방법이 다르다. 이후 제8도를 참조하여 제7(a)도의 경우에 대한 처리 단계에 대해 설명할 것이다.

제8도는 제2 실시예에 따른 문자 인식 과정으 처리 단계를 부분 도시한 플로우챠트인데, 제6도에 설명된 스텝(S3 및 S4)에 본 처리 단계가 상응한다. 나머지 처리 스텝은 제6도의 것과 동일하므로 그에 대한 설명을 생략한다.

제8도의 스텝(S61)은 제6도의 스텝(S3)을 대체하여 삽입되었는데, 여기서는 처리 테이블(204)의 내용이 제1 실시예의 것과 다르다. 환언하면, 처리 테이블(204)이 그 각각이 제7(a)도에 도시된 다수의 스트로크들로 구성된 특정 스트로크들을 지정한다. 그래서 처리 테이블에 저장된 각각의 스트로크 데이타가 어레이로 배열된다. 예를 들어 이중 원은 두개의 스트로크로 기입되었기 때문에 이 데이타는 다음과 같이 설명된다;

ss(ssct).stk(1), ss(ssct),stk(2)

결과적으로, 입력 스트로크를 설명하는 스트로크 데이타 "stk(sct)"는 스텝(S61)에서 처리 테이블(204)에 저장된 제1 스트로크 데이타 "ss(ssct)·stk(1)"과 비교된다. 이 스트로크 비교 수단은 제1 실시예의 특정-스트로크 판별부(201)에 사용된 것과 동일하다. 비교 결과로서 어떤 매칭도 일어나지 않았다면, 과정은 제6도의 스텝(S7)로 진행한다. 매칭이 일어나면, 과정은 제8도의 스텝(S62)로 진행하나 스트로크 카운터 "ct"가 "1"로 초기화된다.

그러면 특정 스트로크 데이타에 뒤이은 스트로크가 처리 테이블에 있는지의 여부, 즉 "ss(ssct).stk(ct+1)"이 존재하는지의 여부에 대한 결정이 스텝(S63)에서 존재한다면 처리 과정은 스텝(S64)로 진행하고 임의의 다른 입력 스트로크가 존재하는지의 여부 즉 "sct+ct"가 이 숫자가 n개의 입력 스트로크 이상인지의 여부에 대한 검사가 이뤄진다.

n보다 크다면, 매칭이 없는 것으로 간주되고 과정은 스텝(S7)로 진행한다. n보다 크지 않다면, 과정은 스텝(S65)로 진행하여 다음 입력 스스트로크 "stk(sct+ct)"가 처리 테이블의 처리된 특정 스트로크 데이타에 뒤이은 스트로크 데이타 "ss(ssct).stk(ct+1)"과 비교된다. 비교 결과로서 어떤 매칭도 없다면, 이 과정은 제6도의 스텝(S7)로 진행한다. 매칭이 이뤄지면 다음 스텝(S66)으로 진행한다.

스텝(S66)에서, 스트로크 카운터 "ct"는 증분에 의해 계수 상승(count up)되고 과정은 스텝(S63)으로 진행한다.

상기와 같이 처리 스텝(S63)에서 (S66)까지는 처리 테이블(204)에 저장된 한 처리를 지정하는 특정 스트로크를 구성하는 모든 스트로크들이 완전히 비교되기까지 반복적으로 수행된다. 모든 스트로크들을 비교하는 작용이 끝난 후에 과정은 스텝(S67)로 진행한다. 모든 입력 스트로크들이 비교되었을 때, 어떤 매칭도 스텝(S63)에서 스텝(S66)까지의 비교 중에서 일어나지 않았다면 어떤 특정 스트로크도 미리 결정된 특정 스트로크 데이타와 매칭되지 않은 것으로 간주되고 이 과정은 상기 언급한 대로 제6도의 스텝(S7)로 진행한다.

매칭이 특정 스트로크에 대해 행해지면, 특정 스트로크를 구성하는 다수의 입력 스트로크, "stk(sct) 내지 stk(n-ct-1)"가 스텝(S67)에서 입력 스트로크 어레이 "stk(n)"으로부터 추출되어 이 과정은 제6도의 스텝(S5)로 진행한다.

제7(a)도의 특정 스트로크 샘플에 대해 이중 원의 두 스트로크가 처리 테이블의 특정 스트로크 데이타와 각각 매칭되어 이를 위해 실행될 프로세스를 지시해 주기 위해 처리 스텝(S63)에서 (S66)까지가 실행된다.

상기와 같이 판별될 심볼이 다수의 스트로크로 구성되었다 하더라도 판별 과정은 상기 처리 스텝을 통해 수행될 수 있다.

그 각각이 특정 스트로크들을 갖는 다수의 심볼이 입력 스트로크득에 존재한다 하더라도, 제6도의 스텝(S6)을 실행하기 전에 다수의 처리 과정이 각 심볼에 대해 수행될 수 있어서 이 처리가 스텝(S1)부터 반복적으로 수행되도록 한다는 것을 알 수 있다.

따라서, 판별될 특정 스트로크의 개수는 특정-스트로크 판별부와 처리 테이블에 약간 변화를 가함으로써 자유롭게 설정될 수 있다.

또한, 처리 테이블이 재기입가능하도록 디자인되었다면 특정 스트로크 유형 및 과정들이 조작자 또는 사용자에 의해 쉽게 변경될 수 있다.

제10도는 제2 실시예에 따른 문자 인식 처리의 내부 데이타를 도시하는데, 여기서 참조 번호(1001, 1002, 1003 및 1004)는 제1 실시예에 개별 도시된 입력 스트로크 어레이(901), 템플리트(template)를 구성하는 특정 스트로크 데이타 어레이(902), 개별 프로세스의 어드레스를 포인트하도록 동작하는 어드레스 포인터(903) 및 특정 스트로크 또는 스트로크들을 제외한 모든 스트로크들을 배치하는 스트로크 어레이(904)에 상응하는 데이타 어레이를 나타낸다.

상기 실시예 각각이 입력 문자의 수기 변환예를 도시하였지만, 일본어-한자 변환이 수기 변환 대신에 실

행될 수 있어서 입력 일본어 문자를 상응하는 한자 패턴으로 변환하게 된다.

본 발명은 다수의 디바이스로 구성된 시스템 또는 단일 디바이스를 갖는 장치에 응용될 수 있다. 또한, 본 발명은 시스템 또는 장치가 소망의 프로그램을 로드함으로써 구현되는 경우에 적용될 수 있다.

상기 설명한 대로, 본 발명에 따른 문자 인식 장치는, 입력 과정에 부가적으로 최소한 하나의 특정 스트로크를 입력할 때 입력-문자 인식 시점에서 처리가 특정 스트로크에 대해 수행되어, 사용자가 커맨드 및 문자가 개별 입력되는 입력 상황을 고려할 필요없이 문자 입력 과정과 동시에 입력 문자용 커맨드를 입력시킬 수 있도록 한다. 이는 문자 인식의 마지막 동작 후에 제스쳐 커맨드가 입력되는 것이 필요한 종래의 장치 또는 시스템과 다른 점이다. 따라서 효율적이고 안정적인 자동 환경이 획득될 수 있다.

또한, 본 발명은 펜 입력 시스템 자체에 이미 포함된 제스쳐 커맨드와 섞이지 않으면서 임의의 커맨드를 펜 입력 시스템이 설정하도록 해준다. 이는 본 발명의 수기 문자 인식 장치가 일본어-한자 변환 장치와 동일 방식으로 프론트-엔드 프로세서로 사용될 때 어느 컴퓨터가 이 장치를 포함하는지에 관계없이 문자 입력 처리에 있어서 동일한 작동 환경을 제공하는 것을 가능하게 해준다. 따라서, 사용자는 서로 다른 컴퓨터 시스템 사이의 작동 혼란에 관계없이 이를 작동시키고 컴퓨터 시스템 사이의 차이 또는 사용자에게 독특한 인식 커맨드로 인해 지원되지 않았던 커맨드를 자유롭게 수용할 수 있어서 고성능 인식 장치를 획득할 수 있다.

(57) 청구의 범위

청구항 1

다수의 스트로크를 입력하는 단계, 문자 인식에 의해 상기 입력 스트로크들이 사전내의 복수의 참조 패턴 중 하나와 일치하는 적어도 하나의 스트로크를 포함하는지의 여부를 판단하는 단계, 상기 판단 단계에서 상기 입력 스트로크들이 상기 참조 패턴들 중 하나와 일치하는 적어도 하나의 스트로크를 포함하는 것으로 판단될 때, 상기 복수의 입력 스트로크로부터 상기 적어도 하나의 스트로크를 추출하는 단계, 상기 추출 단계에서 추출된 상기 적어도 하나의 스트로크와 일치하는 참조 패턴에 대응하는 처리를 선택하는 단계, 및 상기 추출 단계에서 상기 복수의 입력 스트로크로부터 상기 적어도 하나의 입력 스트로크를 추출한 후에 남아있는 스트로크에 대해 선택된 처리를 실행하는 단계를 포함하는 것을 특징으로 하는 정보 처리 방법.

청구항 2

제1항에 있어서, 상기 추출 단계 후에 상기 남아있는 스트로크들을 인식하여, 상기 남아있는 스트로크들을 코드로 변환하는 단계를 더 포함하는 것을 특징으로 하는 정보 처리 방법.

청구항 3

제1항에 있어서, 상기 입력 스트로크들 사이의 브레이크(break) 위치를 판단하는 단계를 더 포함하며, 상기 브레이크 위치를 판단하는 단계는 두개의 브레이크 위치로 경계 지워지는 상기 다수의 스트로크의 각 단위로 실행되는 것을 특징으로 하는 정보 처리 방법.

청구항 4

제3항에 있어서, 상기 브레이크 위치를 판단하는 단계는, 스트로크 입력이 한 입력 프레임에서 다른 입력 프레임으로 시프트된 것을 표시하는 스트로크 입력 프레임 정보를 검출함으로써 실행되는 것을 특징으로 하는 정보 처리 방법.

청구항 5

제3항에 있어서, 상기 브레이크 위치를 파단하는 단계를 미리 결정된 기간 보다 긴 스트로크 입력 간격을 검출함으로써 실행되는 것을 특징으로 하는 정보 처리 방법.

청구항 6

제1항에 있어서, 상기 추출된 스트로크를 제외한 상기 모든 스트로크들은 심볼을 구성하는 것을 특징으로 하는 정보 처리 방법.

청구항 7

제6항에 있어서, 상기 추출된 스트로크를 제외한 상기 모든 스트로크들은 문자를 구성하는 것을 특징으로 하는 정보 처리 방법.

청구항 8

제6항에 있어서, 상기 추출된 스트로크는 상기 심볼에 포함되지 않은 특성을 갖는 것을 특징으로 하는 정보 처리 방법.

청구항 9

제8항에 있어서, 상기 심볼로부터 제외되고 상기 추출된 스트로크에 포함된 상기 특성이 스트로크 입력 방향인 것을 특징으로 하는 정보 처리 방법.

청구항 10

제1항에 있어서, 상기 선택된 처리는 대문자를 소문자로 변환하는 변환 처리인 것을 특징으로 하는 정보 처리 방법.

청구항 11

제1항에 있어서, 상기 선택된 처리는 문자를 수기 패턴으로 변환하는 변환 처리인 것을 특징으로 하는 정보 처리 방법.

청구항 12

제1항에 있어서, 상기 선택된 처리는 상기 추출된 스트로크를 제외한 상기 모든 스트로크들의 데이타를 등록하는 등록 처리인 것을 특징으로 하는 정보 처리 방법.

청구항 13

제1항에 있어서, 상기 선택된 처리는 문자를 한자 패턴으로 변환하는 변환 처리인 것을 특징으로 하는 정보 처리 방법.

청구항 14

제1항에 있어서, 상기 스트로크들은 디지타이저(digitizer)에 의해 입력 되는 것을 특징으로 하는 정보 처리 방법.

청구항 15

제1항에 있어서, 상기 추출된 스트로크를 제외한 상기 모든 스트로크들에 대해 상기 선택된 처리를 실행한 결과가 액정 디스플레이 상에 디스플레이되는 것을 특징으로 하는 정보 처리 방법.

청구항 16

다수의 스트로크를 입력하기 위한 입력 수단, 문자 인식에 의해 상기 입력 스트로크들이 복수의 참조 패턴들 중 하나와 일치하는 적어도 하나의 스트로크를 포함하는지의 여부를 판단하기 위한 판단 수단, 상기 판단 수단이, 상기 입력 스트로크들이 상기 참조 패턴들 중 적어도 하나의 입력 스트로크를 포함한다고 판단할 때, 상기 복수의 입력 스트로크들로부터 적어도 하나의 스트로크를 추출하기 위한 수단, 상기 추출 수단에 의해 추출된 상기 적어도 하나의 스트로크와 일치하는 참조 패턴에 대응하는 처리를 선택하기 위한 처리 선택 수단, 및 상기 추출 수단이 상기 복수의 스트로크들로부터 상기 적어도 하나의 입력 스트로크를 추출한 후에 남아있는 스트로크에 대해 선택된 처리를 실행하기 위한 처리 실행 수단을 포함하는 것을 특징으로 하는 정보 처리 장치.

청구항 17

제16항에 있어서, 상기 추출 수단에 의한 추출 후에 상기 남아있는 스트로크들을 인식하여, 상기 남아있는 스트로크들을 코드로 변환하기 위한 인식 수단을 더 포함하며, 상기 처리 실행 수단은 상기 코드에 대응하는 패턴에 대해 상기 선택된 처리를 실행하는 것을 특징으로 하는 정보 처리 장치.

청구항 18

제16항에 있어서, 상기 입력 스트로크들 간의 브레이크 위치를 판단하기 위한 브레이크-위치 판단 수단을 더 포함하며, 상기 입력 수단은 두개의 브레이크 위치로 경계 지워지는 상기 다수의 스트로크들의 각 단위로 스트로크를 입력하는 것을 특징으로 하는 정보 처리 장치.

청구항 19

제18항에 있어서, 상기 브레이크-위치 판단 수단은 스트로크 입력이 한 입력 프레임에서 다른 입력 프레임으로 시프트된 것을 표시하는 스트로크 입력 프레임 정보를 검출함으로써 상기 브레이크 위치를 판단하는 것을 특징으로 하는 정보 처리 장치.

청구항 20

제18항에 있어서, 상기 브레이크-위치 판단 수단은 미리 결정된 기간 보다 긴 스트로크 입력 간격을 검출함으로써 상기 브레이크 위치를 판단하는 것을 특징으로 하는 정보 처리 장치.

청구항 21

제16항에 있어서, 상기 추출된 스트로크를 제외한 상기 모든 스트로크들은 심볼을 구성하는 것을 특징으로 하는 정보 처리 장치.

청구항 22

제21항에 있어서, 상기 추출된 스트로크를 제외한 상기 모든 스트로크들은 문자를 구성하는 것을 특징으로 하는 정보 처리 장치.

청구항 23

제21항에 있어서, 상기 추출된 스트로크들 중 하나는 상기 심볼에 포함되지 않은 특성을 갖는 것을 특징으로 하는 정보 처리장치.

청구항 24

제23항에 있어서, 상기 심볼로부터 제외되고 상기 추출된 스트로크 중 하나에 포함된 상기 특성이 스트로크 입력 방향인 것을 특징으로 하는 정보 처리 장치.

청구항 25

제16항에 있어서, 상기 선택된 처리는 대문자를 소문자로 변환하는 변환 처리인 것을 특징으로 하는 정보 처리 장치.

청구항 26

제16항에 있어서, 상기 선택된 처리는 문자를 수기 패턴으로 변환하는 변환 처리인 것을 특징으로 하는 정보 처리 장치.

청구항 27

제16항에 있어서, 상기 선택된 처리는 상기 추출된 스트로크를 제외한 상기 모든 스트로크들의 데이타를 등록하는 등록 처리인 것을 특징으로 하는 정보 처리 장치.

청구항 28

제16항에 있어서, 상기 선택된 처리는 문자를 한자 패턴으로 변환하는 변환 처리인 것을 특징으로 하는 정보 처리 장치.

청구항 29

제16항에 잇어서, 상기 입력 수단은 디지타이저인 것을 특징으로 하는 정보 처리 장치.

청구항 30

제16항에 있어서, 상기 추출된 스트로크를 제외한 상기 모든 스트로크들에 대해 상기 선택된 처리를 실행한 결과가 표시되는 액정 디스플레이를 더 포함하는 것을 특징으로 하는 정보 처리 장치.

청구항 31

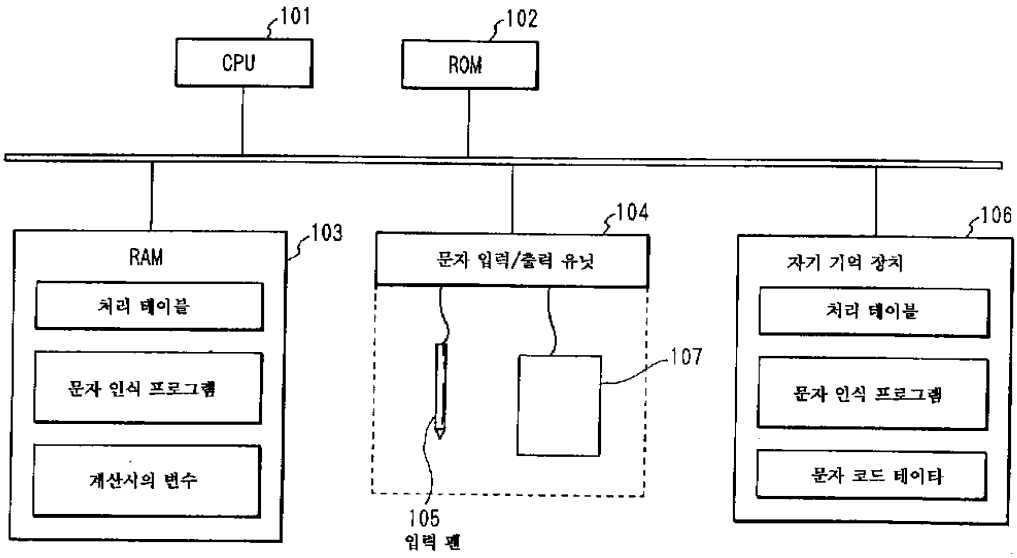
제1항에 있어서, 상기 참조 패턴은 하나의 스트로크를 포함하는 것을 특징으로 하는 정보 처리 방법.

청구항 32

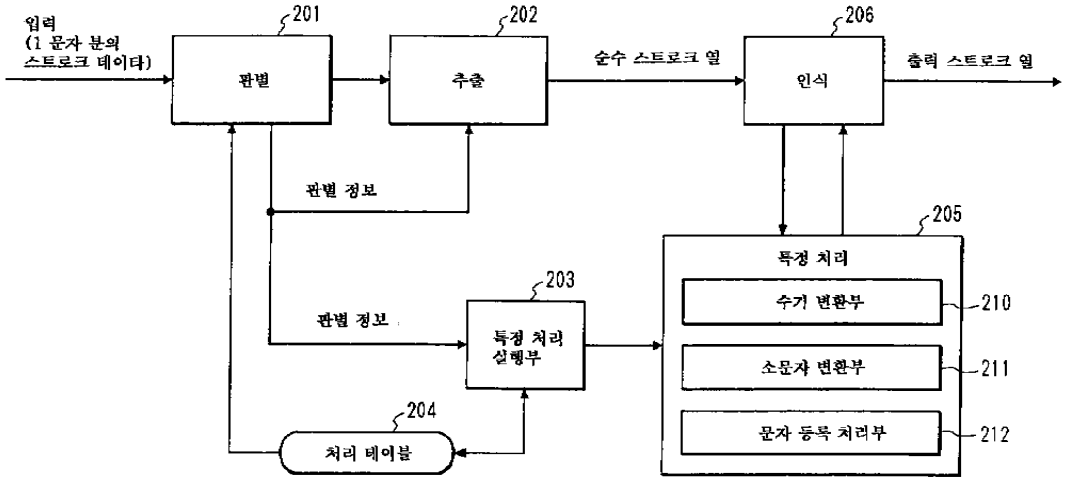
제16항에 있어서, 상기 참조 패턴은 하나의 스트로크를 포함하는 것을 특징으로 하는 정보 처리 장치.

도면

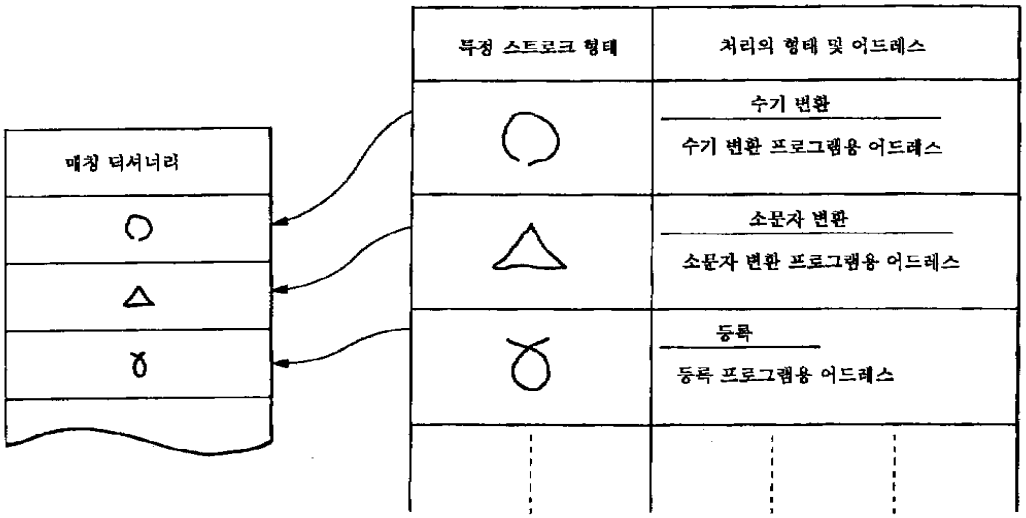
도면1



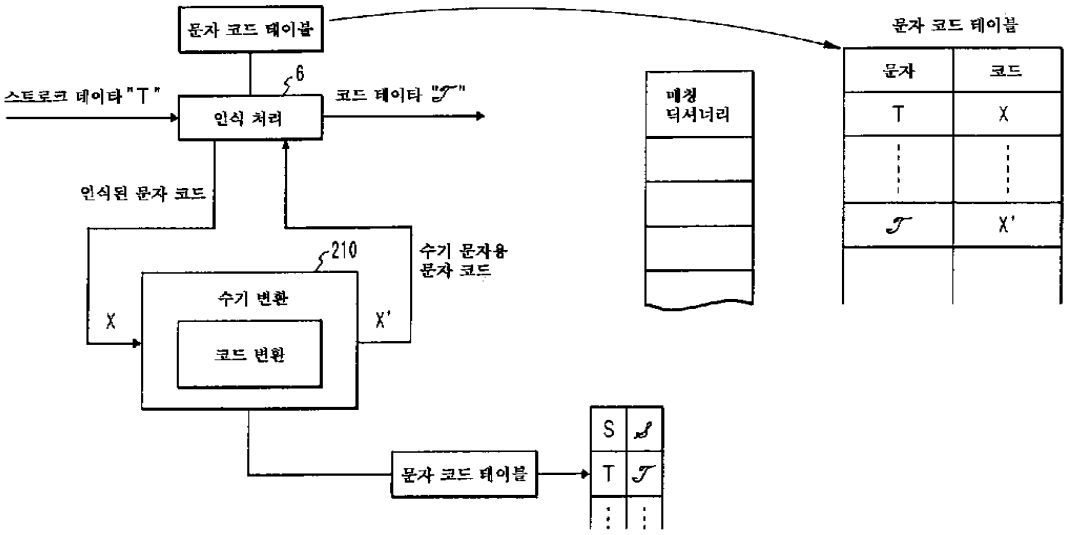
도면2


입력
(1 문자 분의
스트로크 데이타)
201
판별
202
추출
순수 스트로크 열
206
인식
출력 스트로크 열
판별 정보
205
특정 처리
수기 변환부
210
소문자 변환부
211
문자 등록 처리부
212
판별 정보
203
특정 처리
실행부
204
처리 테이블

도면3


특정 스트로크 형태
처리의 형태 및 어드레스
매칭 디셔너리
수기 변환
수기 변환 프로그램용 어드레스
소문자 변환
소문자 변환 프로그램용 어드레스
등록
등록 프로그램용 어드레스

도면4


문자 코드 테이블
문자 코드 테이블
문자
코드
T
X
매칭
디셔너리
스트로크 데이타 "T"
6
인식 처리
코드 데이타 "T"
인식된 문자 코드
210
수기 변환
코드 변환
X
X'
수기 문자용
문자 코드
문자 코드 테이블
S
T

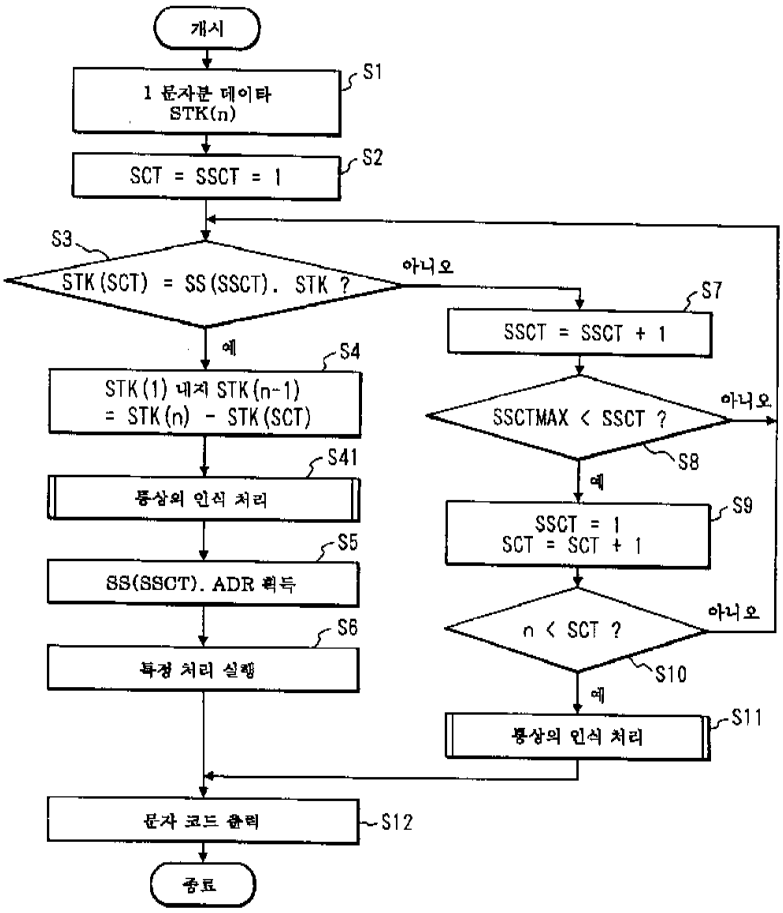
도면5a



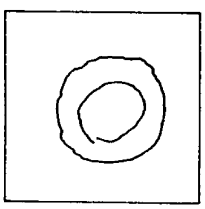
도면5b



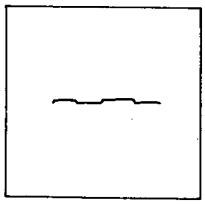
도면6



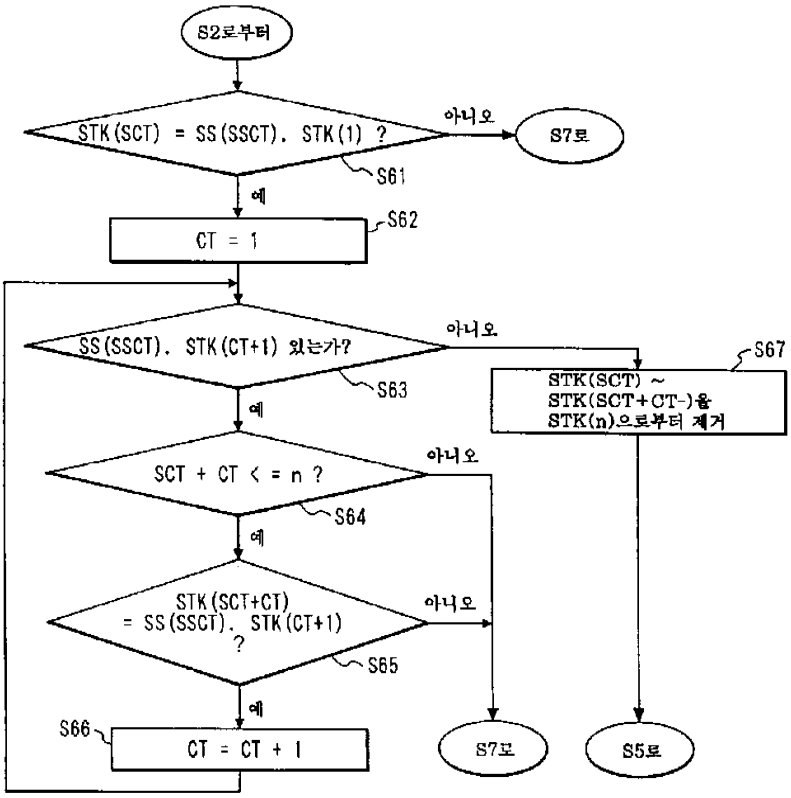
도면7a



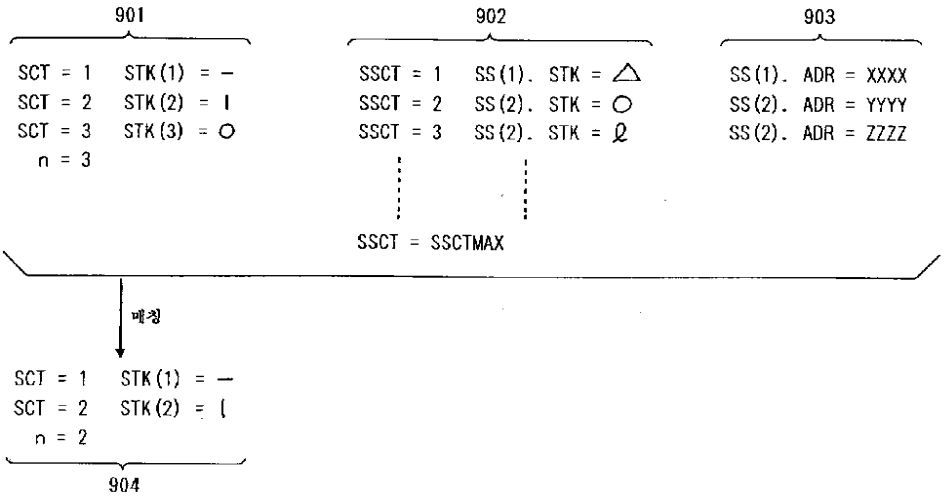
도면7b



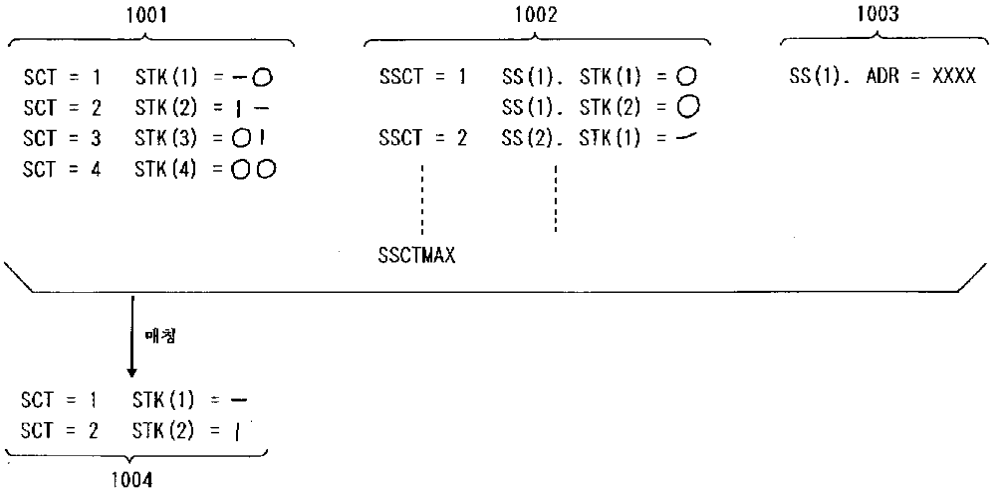
도면8


S2로부터
STK(SCT) = SS(SSCT). STK(1) ?
아니오
S7로
S61
예
CT = 1
S62
SS(SSCT). STK(CT+1) 있는가?
아니오
S67
STK(SCT) ~
STK(SCT+CT-)을
STK(n)으로부터 제거
S63
예
SCT + CT < = n ?
아니오
S64
예
STK(SCT+CT)
= SS(SSCT). STK(CT+1)
?
아니오
S65
예
S66
CT = CT + 1
S7로
S5로

도면9


901
SCT = 1 STK(1) = —
SCT = 2 STK(2) = |
SCT = 3 STK(3) = ○
n = 3
902
SSCT = 1 SS(1). STK = △
SSCT = 2 SS(2). STK = ○
SSCT = 3 SS(2). STK = ℓ
SSCT = SSCTMAX
903
SS(1). ADR = XXXX
SS(2). ADR = YYYY
SS(2). ADR = ZZZZ
매칭
SCT = 1 STK(1) = —
SCT = 2 STK(2) = |
n = 2
904

도면10


1001
1002
1003
SCT = 1 STK(1) = —○
SCT = 2 STK(2) = | —
SCT = 3 STK(3) = ○ |
SCT = 4 STK(4) = ○○
SSCT = 1 SS(1). STK(1) = ○
SS(1). STK(2) = ○
SSCT = 2 SS(2). STK(1) = ⌒
SSCTMAX
SS(1). ADR = XXXX
매칭
SCT = 1 STK(1) = —
SCT = 2 STK(2) = |
1004