



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2022-0107963
(43) 공개일자 2022년08월02일

- | | |
|---|--|
| <p>(51) 국제특허분류(Int. Cl.)
 <i>G06F 21/32</i> (2013.01) <i>G06F 21/45</i> (2013.01)
 <i>G06V 40/18</i> (2022.01) <i>H04L 9/06</i> (2006.01)</p> <p>(52) CPC특허분류
 <i>G06F 21/32</i> (2013.01)
 <i>G06F 21/45</i> (2013.01)</p> <p>(21) 출원번호 10-2022-0008458
 (22) 출원일자 2022년01월20일
 심사청구일자 2022년01월20일</p> <p>(30) 우선권주장
 1020210010156 2021년01월25일 대한민국(KR)</p> | <p>(71) 출원인
 고려대학교 산학협력단
 서울특별시 성북구 안암로 145, 고려대학교 (안암동5가)</p> <p>(72) 발명자
 정익래
 서울특별시 광진구 아차산로 69길 19, 904동 2205호</p> <p>신영아
 경기도 고양시 덕양구 화신로 76, 104-601</p> <p>정재열
 서울특별시 성북구 안암로9가길 55, 304호</p> <p>(74) 대리인
 김등용</p> |
|---|--|

전체 청구항 수 : 총 8 항

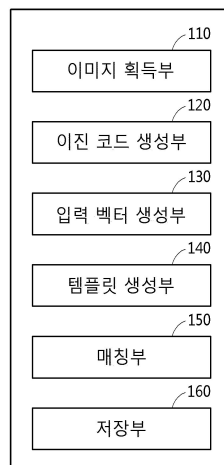
(54) 발명의 명칭 열 기반 슬라이싱을 이용한 폐기 가능한 홍채 템플릿 생성 장치 및 방법

(57) 요약

열 기반 슬라이싱을 이용한 폐기 가능한 홍채 템플릿 생성 장치 및 방법이 개시된다. 상기 폐기 가능한 홍채 템플릿 생성 장치는 홍채 이미지로부터 홍채 이진 코드를 생성하는 이진 코드 생성부, 상기 홍채 이진 코드로부터 IoM 해싱(Index-of-Max Hashing) 기법을 위한 입력 벡터를 생성하는 입력 벡터 생성부, 및 IoM 해싱을 수행하여 상기 입력 벡터에 대응하는 해시 코드를 생성하는 템플릿 생성부를 포함한다.

대표도 - 도5

100



(52) CPC특허분류

G06V 40/18 (2022.01)

H04L 9/0643 (2013.01)

이 발명을 지원한 국가연구개발사업

과제고유번호	1711104076
과제번호	2019R1F1A1060637
부처명	과학기술정보통신부
과제관리(전문)기관명	한국연구재단
연구사업명	개인기초연구(과기정통부)(R&D)
연구과제명	생체 정보에 대한 프라이버시 보장 연산 연구
기 여 율	1/1
과제수행기관명	고려대학교 산학협력단
연구기간	2020.03.01 ~ 2021.02.28

명세서

청구범위

청구항 1

홍채 이미지로부터 홍채 이진 코드를 생성하는 이진 코드 생성부;

상기 홍채 이진 코드로부터 IoM 해싱(Index-of-Max Hashing) 기법을 위한 입력 벡터를 생성하는 입력 벡터 생성부; 및

IoM 해싱을 수행하여 상기 입력 벡터에 대응하는 해시 코드를 생성하는 템플릿 생성부를 포함하는 폐기 가능한 홍채 템플릿 생성 장치.

청구항 2

제1항에 있어서,

상기 폐기 가능한 홍채 템플릿 생성 장치는 상기 해시 코드와 미리 저장된 복수의 해시 코드들 중 적어도 하나의 매칭 여부를 결정하는 매칭부를 더 포함하는,

폐기 가능한 홍채 템플릿 생성 장치.

청구항 3

제1항에 있어서,

상기 입력 벡터 생성부는 상기 홍채 이진 코드에 열(column) 기반의 슬라이싱을 적용하여 상기 입력 벡터를 생성하는,

폐기 가능한 홍채 템플릿 생성 장치.

청구항 4

제1항에 있어서,

상기 입력 벡터 생성부는,

b (b 는 임의의 자연수)행 a (a 는 임의의 자연수)열의 상기 홍채 이진 코드의 각 열(column)을 분리하여 a 개의 b 행 1열 데이터를 추출하고,

추출된 b 행 1열 데이터들 각각을 1행 b 열의 데이터로 변환하고,

1행 b 열의 데이터들을 결합하여 1행 ($a \times b$)열의 테이블을 생성하고,

1행 ($a \times b$)열의 테이블을 c (c 는 임의의 자연수) 단위로 분할한 후 십진수로 변환하고,

십진수로 변환된 값들 각각을 1,024로 나눈 후 0.5를 감산함으로써 상기 입력 벡터를 생성하는,

폐기 가능한 홍채 템플릿 생성 장치.

청구항 5

적어도 프로세서를 포함하는 컴퓨팅 장치에 의해 수행되는, 폐기 가능한 홍채 템플릿 생성 방법에 있어서,

홍채 이미지로부터 홍채 이진 코드를 생성하는 단계;

상기 홍채 이진 코드로부터 IoM 해싱(Index-of-Max Hashing) 기법을 위한 입력 벡터를 생성하는 단계; 및

IoM 해싱을 수행하여 상기 입력 벡터에 대응하는 해시 코드를 생성하는 단계를 포함하는 폐기 가능한 홍채 템플릿 생성 방법.

청구항 6

제5항에 있어서,

상기 폐기 가능한 홍채 템플릿 생성 방법은 상기 해시 코드와 미리 저장된 복수의 해시 코드들 중 적어도 하나의 매칭 여부를 결정하는 단계를 더 포함하는,

폐기 가능한 홍채 템플릿 생성 방법.

청구항 7

제5항에 있어서,

상기 입력 벡터를 생성하는 단계는, 상기 홍채 이진 코드에 열(column) 기반의 슬라이싱을 적용하여 상기 입력 벡터를 생성하는,

폐기 가능한 홍채 템플릿 생성 방법.

청구항 8

제5항에 있어서,

상기 입력 벡터를 생성하는 단계는,

b(b는 임의의 자연수)행 a(a는 임의의 자연수)열의 상기 홍채 이진 코드의 각 열(column)을 분리하여 a개의 b행 1열 데이터를 추출하고,

추출된 b행 1열 데이터들 각각을 1행 b열의 데이터로 변환하고,

1행 b열의 데이터들을 결합하여 1행 (a×b)열의 테이블을 생성하고,

1행 (a×b)열의 테이블을 c(c는 임의의 자연수) 단위로 분할한 후 십진수로 변환하고,

십진수로 변환된 값들 각각을 1,024로 나눈 후 0.5를 감산함으로써 상기 입력 벡터를 생성하는,

폐기 가능한 홍채 템플릿 생성 방법.

발명의 설명

기술 분야

[0001] 본 발명은 홍채 인식 환경에서 인증의 정확성과 보안성을 높일 수 있는 폐기 가능한 홍채 템플릿의 생성 방법과 관련한 것으로, 특히 열 기반의 슬라이싱 기법을 활용하여 보안성과 효율성을 향상시킬 수 있는 폐기 가능한 홍채 템플릿을 생성하는 장치 및 방법에 관한 것이다.

배경 기술

[0002] 최근 코로나 19로 인한 언택트 지향의 문화가 형성되고, 2020년 5월 전자서명법 개정안이 국회 본회의에 통과되면서 사용자인증기술에 대한 관심이 증가하였다. 특히, 홍채를 이용한 생체 인증은 높은 정확성을 가지며 비접촉 방식을 취하고 있어 더욱 주목을 받고 있다. 그런데 홍채의 불변성은 인증 시 장점이지만, 유출될 경우 해당 홍채 정보로 다른 사용자도 시스템에 접속할 수 있게 된다. 따라서, 이러한 위협을 사전에 방지하기 위해서는 홍채 정보를 통해 생성한 템플릿을 폐기 가능하도록 설계하여 저장해야 한다. 폐기 가능한 생체 템플릿은 같은 홍채로부터 다수의 구별불가능한 새로운 템플릿을 다시 생성한다.

[0003] Kim et al.은 Jin et al.의 IoM(Index-of-Max) 해싱(Z. Jin, J.Y. Hwang, Y.L. Lai, S. Kim and A.B.J. Teoh, "locality sensitive hashing-enabled cancelable biometrics: Index-of-max hashing," IEEE Transactions on Information Forensics and Security, vol. 13, no. 2, pp. 393-407, 2018.)을 지문이 아닌 홍채 템플릿에 적용하여 유출 등의 공격에 강력함을 입증하였다(김진아, 정재열, 김기성, 정익래. "해싱을 이용한 폐기가능한 홍채 템플릿," 정보보호학회논문지, vol. 29, no. 3, 565-577, 2019.). 그러나 Kim et al.의 홍채 템플릿 생성 과정은 사전정렬을 위해 시프트 연산을 진행하게 되면서 추가적인 연산이 발생한다.

[0004] 본 발명에서는 이러한 점에 주목하여 연산 처리속도를 향상하고자 홍채 이진 코드를 행 기반이 아닌 열 기반으로 분할한다. 홍채 이진 코드에서 가로 10bit씩이 아닌 세로로 슬라이싱하여 IoM 해싱의 입력벡터를 구성한다면

기존 연산의 효율성을 개선할 수 있다.

발명의 내용

해결하려는 과제

- [0005] 본 발명이 이루고자 하는 기술적인 과제는 연산 처리속도가 향상된, 폐기 가능한 홍채 템플릿을 생성하는 방법 및 장치를 제공하는 것이다.

과제의 해결 수단

- [0006] 본 발명의 일 실시예에 따른 폐기 가능한 홍채 템플릿 생성 장치는 홍채 이미지로부터 홍채 이진 코드를 생성하는 이진 코드 생성부, 상기 홍채 이진 코드로부터 IoM 해싱(Index-of-Max Hashing) 기법을 위한 입력 벡터를 생성하는 입력 벡터 생성부, 및 IoM 해싱을 수행하여 상기 입력 벡터에 대응하는 해시 코드를 생성하는 템플릿 생성부를 포함한다.
- [0007] 또한, 본 발명의 일 실시예에 따른 폐기 가능한 홍채 템플릿 생성 방법은 적어도 프로세서를 포함하는 컴퓨팅 장치에 의해 수행되고, 홍채 이미지로부터 홍채 이진 코드를 생성하는 단계, 상기 홍채 이진 코드로부터 IoM 해싱(Index-of-Max Hashing) 기법을 위한 입력 벡터를 생성하는 단계, 및 IoM 해싱을 수행하여 상기 입력 벡터에 대응하는 해시 코드를 생성하는 단계를 포함한다.

발명의 효과

- [0008] 본 발명의 실시예에 따른 폐기 가능한 홍채 템플릿 생성 장치 및 방법에 의한 경우, 열 기반 슬라이싱 기법을 통해 IoM 해싱의 매칭 수행 시간을 향상시킬 수 있는 효과가 있다.

도면의 간단한 설명

- [0009] 본 발명의 상세한 설명에서 인용되는 도면을 보다 충분히 이해하기 위하여 각 도면의 상세한 설명이 제공된다.
- 도 1은 홍채 이미지의 일 예를 도시한다.
- 도 2는 홍채 이진 코드의 일 예를 도시한다.
- 도 3은 본 발명의 일 실시예에 의한 IoM 해싱의 입력값 변환 단계를 설명하기 위한 도면이다.
- 도 4는 본 발명과 종래 기법의 실험 결과를 도시하는 그래프이다.
- 도 5는 본 발명의 일 실시예에 따른 홍채 템플릿 생성 장치의 기능 블록도이다.
- 도 6은 본 발명의 일 실시예에 따른 홍채 템플릿 생성 방법을 설명하기 위한 흐름도이다.

발명을 실시하기 위한 구체적인 내용

- [0010] 본 명세서에 개시되어 있는 본 발명의 개념에 따른 실시예들에 대해서 특정한 구조적 또는 기능적 설명들은 단지 본 발명의 개념에 따른 실시예들을 설명하기 위한 목적으로 예시된 것으로서, 본 발명의 개념에 따른 실시예들은 다양한 형태들로 실시될 수 있으며 본 명세서에 설명된 실시예들에 한정되지 않는다.
- [0011] 본 발명의 개념에 따른 실시예들은 다양한 변경들을 가할 수 있고 여러 가지 형태들을 가질 수 있으므로 실시예들을 도면에 예시하고 본 명세서에서 상세하게 설명하고자 한다. 그러나, 이는 본 발명의 개념에 따른 실시예들을 특정한 개시 형태들에 대해 한정하려는 것이 아니며, 본 발명의 사상 및 기술 범위에 포함되는 모든 변경, 균등물, 또는 대체물을 포함한다.
- [0012] 제1 또는 제2 등의 용어는 다양한 구성 요소들을 설명하는데 사용될 수 있지만, 상기 구성 요소들은 상기 용어들에 의해 한정되어서는 안 된다. 상기 용어들은 하나의 구성 요소를 다른 구성 요소로부터 구별하는 목적으로만, 예컨대 본 발명의 개념에 따른 권리 범위로부터 벗어나지 않은 채, 제1 구성 요소는 제2 구성 요소로 명명될 수 있고 유사하게 제2 구성 요소는 제1 구성 요소로도 명명될 수 있다.
- [0013] 어떤 구성 요소가 다른 구성 요소에 "연결되어" 있다거나 "접속되어" 있다고 언급된 때에는, 그 다른 구성 요소에 직접적으로 연결되어 있거나 또는 접속되어 있을 수도 있지만, 중간에 다른 구성 요소가 존재할 수도 있다고 이해되어야 할 것이다. 반면에, 어떤 구성 요소가 다른 구성 요소에 "직접 연결되어" 있다거나 "직접 접속되어"

있다고 언급된 때에는 중간에 다른 구성 요소가 존재하지 않는 것으로 이해되어야 할 것이다. 구성 요소들 간의 관계를 설명하는 다른 표현들, 즉 "~사이에"와 "바로 ~사이에" 또는 "~에 이웃하는"과 "~에 직접 이웃하는" 등도 마찬가지로 해석되어야 한다.

[0014] 본 명세서에서 사용한 용어는 단지 특정한 실시예를 설명하기 위해 사용된 것으로서, 본 발명을 한정하려는 의도가 아니다. 단수의 표현은 문맥상 명백하게 다르게 뜻하지 않는 한, 복수의 표현을 포함한다. 본 명세서에서, "포함하다" 또는 "가지다" 등의 용어는 본 명세서에 기재된 특징, 숫자, 단계, 동작, 구성 요소, 부분품 또는 이들을 조합한 것이 존재함을 지정하려는 것이지, 하나 또는 그 이상의 다른 특징들이나 숫자, 단계, 동작, 구성 요소, 부분품 또는 이들을 조합한 것들의 존재 또는 부가 가능성을 미리 배제하지 않는 것으로 이해되어야 한다.

[0015] 다르게 정의되지 않는 한, 기술적이거나 과학적인 용어를 포함해서 여기서 사용되는 모든 용어들은 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자에 의해 일반적으로 이해되는 것과 동일한 의미를 가진다. 일반적으로 사용되는 사전에 정의되어 있는 것과 같은 용어들은 관련 기술의 문맥상 가지는 의미와 일치하는 의미를 갖는 것으로 해석되어야 하며, 본 명세서에서 명백하게 정의하지 않는 한, 이상적이거나 과도하게 형식적인 의미로 해석되지 않는다.

[0016] 이하, 본 명세서에 첨부된 도면들을 참조하여 본 발명의 실시예들을 상세히 설명한다. 그러나, 특허출원의 범위가 이러한 실시예들에 의해 제한되거나 한정되는 것은 아니다. 각 도면에 제시된 동일한 참조 부호는 동일한 부재를 나타낸다.

[0017] 본 발명에서 제안하는 폐기 가능한 홍채 템플릿을 생성하는 방법은 크게 다음과 같이 구성된다.

[0018] 1) 홍채 이진 코드를 생성한다.

[0019] 2) 홍채 이진 코드로부터 추출한 이진값을 실수화하여 IoM 해싱(Index-of-Max Hashing)의 입력 벡터값으로 변환한다.

[0020] 3) GRP(Gaussian Random Projection) 혹은 URP(Uniform Random Permutation) 방식으로 IoM 해싱을 수행한다.

[0021] 4) LSH(Locality Sensitive Hashing) 기반으로 IoM 해싱의 매칭을 수행한다.

[0022] 본 발명에서 제안하는 열 기반 슬라이싱은 두번째 단계인 IoM 해싱의 입력 벡터 변환단계에서 활용된다. 따라서, 본 명세서에서는 해당 단계에서 효율성을 개선하는 방법을 자세히 기술한 후 나머지 단계는 간략히 서술하고자 한다.

[0023] 홍채 이미지의 이진 코드 생성 단계

[0024] 홍채 이미지(도 1 참조)는 해싱 및 매칭을 위해 이진 코드로 변환하는 과정을 거친다. 이때 사용되는 도그만(Daugman) 알고리즘은 홍채 이미지에서 노이즈를 제거하고, 정규화 작업을 진행한 뒤 홍채 특징을 가버 필터(Gabor Filter)로 검출한다. 이렇게 생성된 홍채 이진 코드는 가로 a비트(a는 임의의 자연수로써 예시적인 값은 480임), 세로 b비트(b는 임의의 자연수로써 예시적인 값은 20임)의 크기를 가진다(도 2 참조).

[0025] IoM 해싱의 입력값 변환 단계

[0026] IoM 해싱의 입력값(입력 벡터 또는 특징 벡터로 명명될 수도 있음)은 실숫값의 특징 벡터이므로, 앞선 단계에서 도출한 홍채 이진 코드는 실수화 변환과정이 필요하다. 본 발명에서는 열(column) 기반 슬라이싱을 통한 실수화를 진행한다(도 3 참조).

[0027] 열 기반 슬라이싱을 통한 실숫값 변환과정은 정렬 과정이 불필요(alignment-free)하다는 점에서 명확한 강점을 가진다. 일반적으로 생체정보는 사전에 정확히 정렬되어야 비교가 가능하다. 홍채의 경우 코드를 시프트함으로써 완전한 정렬을 찾아간다. 그러나 세로(열)를 기반으로 코드를 분할하여 정렬할 경우 별도의 시프트 과정이 필요로 하지 않아 과정이 간편해 진다. 따라서 처리 속도 측면에서 효율성이 높아지게 된다.

[0028] 홍채코드의 실수화 변환 순서는 다음과 같다.

[0029] 먼저, 홍채 이진 코드를 $b \times a$ 비트(예컨대, 20×480 비트)의 테이블로 생성한다. 다음으로 열별로 슬라이싱하여 b (예컨대, 20)행 1열의 데이터를 추출한 후 1행 b (예컨대, 20)열의 행렬로 해당 행렬을 전치한다. 이와 같은 방식으로 a (예컨대, 480) 번을 진행하면 총 $1 \times (a \times b)$ 비트(예컨대, 1×9600 비트)의 테이블이 형성된다. 원활한 실수화와 보안상의 강점을 가지기 위해서는 벡터의 값이 음과 양의 값으로 생성해야 한다. 따라서 해당 연산을 진

행하기 위해 $1 \times (a \times b)$ 비트(예컨대, 1×9600 비트)의 테이블을 c 비트(c 는 임의의 자연수로써, 예시적인 값은 10임)씩 분할하여 $1 \times (a \times b)/c$ (예컨대, 960)개의 벡터로 생성한다. 그리고 각 c (예컨대, 10)비트씩의 이진수를 0~1023 범위의 십진수로 변환한 후, 1,024로 나누고 0.5를 빼 -0.5~0.4990 사이의 값을 가지도록 계산한다. 실시예에 따라, 512를 뺀 후 512로 나누는 것도 가능하다.

[0030] IoM 해싱 수행 단계

[0031] IoM 해싱은 실숫값의 특징 벡터를 입력으로 하여 최대 값을 갖는 인덱스를 가지는 순위 기반의 코드로 변환함으로써 다음과 같은 장점을 갖는다.

[0032] 1) 해싱된 코드를 가지고 원본 특징 벡터를 복원할 수 없기 때문에 비가역성을 만족한다.

[0033] 2) 상대적인 순서에 의존적인 방법이기 때문에 특징 벡터의 크기에 영향을 받지 않는다. 따라서, 노이즈나 변형에 강력하다.

[0034] 3) IoM 해싱의 크기 독립성으로 인해 해시 코드를 크기 불변(scale-invariant)하게 한다.

[0035] IoM 해싱은 두 가지 방법으로 실현될 수 있고, IoM 해싱과 이하에서 기술될 매칭과 관련하여서는, Z. Jin 등의 논문(Z. Jin, J.Y. Hwang, Y.L. Lai, S. Kim and A.B.J. Teoh, "locality sensitive hashing-enabled cancelable biometrics: Index-of-max hashing," IEEE Transactions on Information Forensics and Security, vol. 13, no. 2, pp. 393-407, 2018.)이 참조될 수 있다.

[0036] 첫 번째, GRP 기반 IoM 해싱은 d (d 는 임의의 자연수)차원의 특징 벡터 x 를 입력받아 가우시안(Gaussian) 랜덤 투영 벡터 q 를 생성하여 m (m 은 임의의 자연수)개의 가우시안(Gaussian) 랜덤 행렬에서 최댓값의 인덱스를 출력한다. GRP 기반의 IoM 해싱을 수행하기 위한 알고리즘 1은 아래와 같다.

[0037] [알고리즘 1]

Algorithm 1 Gaussian Random Projection Based IoM Hashing

Input Feature vector x with d dimensions, number of Gaussian random matrices m and the number of Gaussian random projection vector q

Step 1: Generate m Gaussian random matrices $\mathbf{W}^i = (\mathbf{w}_1^i, \dots, \mathbf{w}_q^i) \quad i = 1, \dots, m.$

Step 2: Initialize i^{th} hashed code $t_i = 0.$

Step 3: Perform *random projection* and record the *maximum index* in the projected feature vector.

For $k = 1: m$

$$\bar{x}^k = \mathbf{W}^k x$$

Find $\mathbf{x}_j^k = \max(\bar{x}^k), j = 1, \dots, q$

Then $t_i = j$ (j refers the index of $\bar{x}^k$)

End for

Output Hashed code $t_{GRP} = \{t_i | i = 1, \dots, m\}$ and $t_{GRP} \in [1, q].$

[0038]

[0039] 두 번째, URP 기반 IoM 해싱은 특징 벡터 x 를 입력받아 균일 랜덤 치환을 수행하여 p (p 는 임의의 자연수)개의 벡터를 생성한 후 아다마르 곱(Hadamard Product) 연산으로 각각의 집합을 만들어낸 뒤 최댓값의 인덱스를 출력한다. URP 기반의 IoM 해싱을 수행하기 위한 알고리즘 2는 아래와 같다.

[0040] [알고리즘 2]

Algorithm 2 Uniformly Random Permutation Based IoM Hashing

Input Feature vector $\mathbf{x}$, p order of Hadamard product, number of random permutation m , window size k
 For each permutation set $\theta_{(l,i)} | l = 1, \dots, q, i = 1, \dots, m$
Step 1: Permute elements in $\mathbf{x}$ based on $\theta_{(i,l)}, \hat{\mathbf{x}} = \text{perm}(\mathbf{x})$, $\text{perm}(\cdot)$ is the random permutation function
Step 2: Initialize i^{th} hashed code $t_i = 0$
Step 3: Hadamard product vector generation and output hashed codes
 For $j = 1: k$
 Set $\bar{\mathbf{x}}(j) = \prod_{l=1}^p (\hat{\mathbf{x}}_l(j))$
 if $\bar{\mathbf{x}}(j) > \bar{\mathbf{x}}(t_i)$ then $t_i = j$
 End for
Output Hashed code $t_{URP} = \{t_i \in [1, k] | i = 1, \dots, m\}$

[0041]

[0042] IoM 해싱 결과값간의 매칭 단계

[0043] IoM 해싱의 매칭은 유사도가 높은 t^e 와 t^q 간에 충돌 확률이 높은 순위 기반 LSH(Locality Sensitive Hashing) 알고리즘을 따른다. 두 개의 홍채 벡터가 유사도가 높으면 해시코드의 충돌 확률이 높고, 유사도가 낮으면 해시코드(IoM 해싱의 결과값을 의미할 수 있음)의 충돌 확률이 낮다.

[0044] 등록된 해시코드 $t^e = \{t_i^e | i = 1, \dots, m\}$ 와 쿼리 해시코드 $t^q = \{t_j^q | j = 1, \dots, m\}$ 사이의 충돌 가능성은 유사도 $S(t^e, t^q)$ 로 표현된다. 즉, $\mathbb{P}[t_i^e = t_j^q] = S(t^e, t^q)$ for $1, \dots, m$ 이 된다.

[0045] 1) GRP 기반의 IoM 매칭

[0046] h 를 $h: \mathbb{R}^d \rightarrow \{1, \dots, q\}^m$ 로 정의되는 LSH 함수라 하자. $\mathbf{u}, \mathbf{v} \in \mathbb{R}^d$ 이고 $h(\mathbf{u}) = \{\arg \max_{j=1 \dots q} \langle \mathbf{w}_j^i, \mathbf{u} \rangle | i = 1, \dots, m\}$ 이다. 여기서 $\{\mathbf{w}_j \in \mathbb{R}^d | j = 1, \dots, q\} \sim \mathcal{N}(0, \mathbf{I}_d)$ 이다.

[0047] GRP 기반 IoM 해싱에서의 유사도 $S_{GRP}(t^e, t^q) = \mathbb{P}\left\{\arg \max_{j=1 \dots q} \langle \mathbf{w}_j^i, \mathbf{u} \rangle = \arg \max_{j=1 \dots q} \langle \mathbf{w}_j^i, \mathbf{v} \rangle\right\}$ 이다.

[0048] 구현에서의 매칭 점수는 해시코드의 총 엔트리(m)에 대한 t^e 와 t^q 간의 요소별 뱀셈에 의해 '0'(충돌)이 되는 횟수이다.

[0049] 2) URP 기반 IoM 매칭

[0050] URP 기반 IoM 해싱에서의 유사도는 일정한 범위 내에서의 상대적인 순서 측정이며, 수학적 식 1과 같다.

[0051] [수학적 식 1]

$$S_{URP}(t^e, t^q) = \frac{\sum_{i=0}^{n-1} \binom{R_i(t^e, t^q)}{k-1}}{\binom{d}{k}}$$

[0052]

[0053] 구현에서의 매칭 점수는 해시코드의 총 엔트리(m)에 대한 t^e 와 t^q 간의 요소별 뱀셈에 의해 '0'(충돌)이 되는 횟수이다.

[0054] 결국, GRP 기반의 IoM 매칭과 URP 기반의 IoM 매칭에서, 매칭 점수가 미리 정해진 기준치를 초과(또는 기준치와 같거나 기준치를 초과)하는 경우 두 해시코드가 매칭된다고 결정될 수 있고, 그렇지 않은 경우 두 해시코드가 매칭되지 않는다고 결정될 수 있다. 두 해시코드가 매칭된다는 것은 비교대상인 두 홍채가 동일인의 홍채임을 의미할 수 있다.

[0055] 이하에서는 실험 및 평가 결과를 설명한다.

[0056] 제안하는 방법론의 효율성을 평가하기 위한 실험 환경은 아래 표 1과 같다.

[0057] [표 1]

Item	Configuration
OS	Windows 10
CPU	Intel(R) Core(TM) i7-4790 CPU @3.60GHz
Memory	8GB
Software	MATLAB Ver.R2020a
Data Set	CASIA v3

[0058]

[0059] 실험에서는 본 방법론의 성능을 평가하기 위해 IoM 해싱의 매칭 시간에서 비교적 큰 차이를 보이는 GRP 기반의 IoM 해싱을 기준으로 비교·분석하였다. Kim et al.에 따르면, 기존 가로 기반 슬라이싱 기법 중 GRP 기반의 IoM 해싱은 0.065초, URP 기반의 IoM 해싱은 0.058초로 측정되었다.

[0060] 실험 데이터는 Kim et al.과의 동등한 비교를 위해 같은 방식으로 CASIA v3를 사용하였다. 249개의 subject의 이미지를 추출하되, 한 subject의 좌우 홍채는 각기 다른 subject로 취급하였고, 그중 7개 이상의 이미지만 가진 subject를 선택하였다.

[0061] 효율성의 척도는 매개변수인 Gaussian 랜덤 투영 벡터인 q , Gaussian 랜덤 행렬 m 의 값에 따라 1개의 홍채 쿼리 템플릿이 등록된 해시 이진 코드와 매칭하는데 소요되는 시간으로 설정하였다.

[0062] 기존 가로 기반의 슬라이싱 기법은 실험 결과 q 와 m 의 값이 증가할수록 소요시간도 비례하여 증가하였다. 특히 m 이 100 이상으로 커질수록 증가폭도 커졌다. 그러나 본 방법론은 m 의 값이 증가하여도 소요되는 시간은 0.02초 이내로 측정되었다.

[0063] 도 4는 그중 차이가 확연히 드러난 지점인 q 가 5일 때 소요시간의 추이를 나타낸다. m 이 작은 순간인 10일 때를 제외하고, 나머지 값에서는 모두 빠른 속도를 보였다

[0064] 표 2는 그중 결과값이 확연히 드러나는 지점인 q 가 5, m 이 300일 때의 소요시간을 나타낸다. 본 방법론은 시프트 연산을 수행하여 17 개의 템플릿을 생성하는 열 기반 기법에 비해 약 21배 정도의 높은 처리율을 보였다. 한편, 정확도를 평가하는 척도인 EER은 Kim et al.보다 약 10배 정도 높아졌음을 확인하였다

[0065] [표 2]

	Row-Based	Column-Based
Time (sec)	0.373717	0.017553
EER	0.08	0.79

[0066]

[0067] 도 5는 본 발명의 일 실시예에 따른 홍채 템플릿 생성 장치의 기능 블록도이다.

[0068] 도 5를 참조하면, 홍채 템플릿 생성 장치(100)는 적어도 프로세서 및/또는 메모리를 포함하는 컴퓨팅 장치로 구현될 수 있다. 컴퓨팅 장치는 데스크톱 PC, 태블릿 PC, 서버, 스마트폰, 네비게이션 등을 포함할 수 있다. 홍채 템플릿 생성 장치(100)는 이미지 획득부(110), 이진 코드 생성부(120), 입력 벡터 생성부(130), 템플릿 생성부(140), 매칭부(150), 및 저장부(160) 중 적어도 하나 이상을 포함할 수 있다. 홍채 템플릿 생성 장치(100)를 설명함에 있어 앞선 기재와 중복되는 내용에 관하여는 그 상세한 기재를 생략하기로 한다.

[0069] 이미지 획득부(110)는 폐기 가능한 홍채 템플릿을 생성할 대상 홍채 이미지를 획득할 수 있다. 실시예에 따라, 홍채 이미지는 미리 저장부(160)에 저장되어 있을 수 있고, 이 경우 홍채 템플릿 생성 장치(100)는 이미지 획득

부(110)를 포함하지 않을 수도 있다.

- [0070] 구체적으로, 이미지 획득부(110)는 홍채를 스캔할 수 있는 스캐너 등으로부터 유무선 통신망을 통하여 사용자의 홍채 이미지를 수신할 수 있다. 실시예에 따라, USB 저장 장치와 같은 휴대용 저장 장치로부터 소정의 입력 인터페이스를 통해 홍채 이미지를 수신하는 것도 가능하다. 또 다른 예로, 홍채 템플릿 생성 장치(100)에 구비된 카메라 등과 같은 촬상 장치를 이용하여 사용자의 홍채 이미지를 촬영함으로써 사용자의 홍채 이미지를 획득할 수도 있다.
- [0071] 이진 코드 생성부(120)는 이미지 획득부(110)에 의해 획득된 홍채 이미지 또는 저장부(160)에 저장되어 있는 홍채 이미지로부터 이에 대응하는 이진 코드(또는 홍채 이진 코드)를 생성할 수 있다. 이를 위해, 이진 코드 생성부(120)는 홍채 이미지로부터 홍채 영역 검출, 노이즈 제거, 정규화 작업 중 적어도 하나를 수행할 수 있다.
- [0072] 입력 벡터 생성부(130)는 홍채 이진 코드로부터 IoM 해싱을 위한 입력 벡터를 생성할 수 있다. 입력 벡터 생성부(130)는 앞선 기재와 같이, 열 단위의 슬라이싱을 적용하여 IoM 해싱을 위한 입력 벡터를 생성할 수 있다.
- [0073] 템플릿 생성부(140)는 입력 벡터 생성부(130)에 의해 생성된 입력 벡터로부터 폐기 가능한 홍채 템플릿을 생성할 수 있다. 여기서, 홍채 템플릿은 입력 벡터를 입력으로 하는 IoM 해싱의 결과물으로써, 해싱 코드로 명명될 수도 있다.
- [0074] 홍채 템플릿 생성 장치(100)는 매칭부(150)를 더 포함할 수 있다. 이 경우, 홍채 템플릿 생성 장치(100)는 홍채 인식 장치, 홍채 매칭 장치 등으로 명명될 수도 있다.
- [0075] 매칭부(150)는 생성된 홍채 템플릿과 미리 저장되어 있는 복수의 홍채 템플릿들 중 적어도 하나를 비교하여 유사도를 계산함으로써 두 홍채 템플릿의 매칭 여부를 결정할 수 있다.
- [0076] 저장부(160)에는 홍채 템플릿 생성 장치(100)의 구동에 필요한 프로그램, 소프트웨어 등이 저장되어 있을 수 있다. 또한, 저장부(160)에는 폐기 가능한 홍채 템플릿을 생성하는 과정 중에 생성되는 데이터, 예컨대, 홍채 이미지, 홍채 이진 코드, 입력 벡터, 홍채 템플릿, 유사도, 매칭의 결과 등이 일시적으로 또는 비일시적으로 저장될 수 있다.
- [0077] 도 6은 본 발명의 일 실시예에 따른 홍채 템플릿 생성 방법을 설명하기 위한 흐름도이다.
- [0078] 홍채 템플릿 생성 방법은 도 5에 도시된 홍채 템플릿 생성 장치(100), 즉 컴퓨팅 장치에 의해 수행될 수 있으며, 홍채 인식 방법, 홍채 매칭 방법 등으로 명명될 수도 있다.
- [0079] 우선, 홍채 템플릿 생성 장치(100)의 이미지 획득부(110)에 의해 홍채 이미지가 획득된다(S110). 획득된 홍채 이미지는 저장부(160)에 저장될 수 있다.
- [0080] 다음으로, 홍채 템플릿 생성 장치(100)의 이진 코드 생성부(120)는 획득된 홍채 이미지에 대응하는 홍채 이진 코드를 생성할 수 있다(S120). 홍채 이진 코드는 알려진 기법을 이용하여 생성될 수 있으므로, 이에 대한 상세한 설명은 생략하기로 한다.
- [0081] 홍채 템플릿 생성 장치(100)의 입력 벡터 생성부(130)는 홍채 이진 코드로부터 IoM 해싱의 입력을 위한 입력 벡터를 생성할 수 있다(S130). IoM 해싱 기법을 통해 해시 코드, 즉 홍채 템플릿을 생성하기 위해서는, 미리 정해진 규격의 입력이 필요하다. 따라서, 입력 벡터 생성부(130)는 미리 정해진 규격에 맞도록 홍채 이진 코드를 입력 벡터로 변환할 수 있다.
- [0082] 홍채 템플릿 생성 장치(100)의 템플릿 생성부(140)는 IoM 해싱 기법을 이용하여 입력 벡터에 대응하는 해시 코드를 생성할 수 있다(S140). 해시 코드는 입력된 홍채 이미지에 대응하는 홍채 템플릿을 의미할 수 있다. 이에 따라, 사용자의 홍채 이미지에 대응하고 폐기가 가능한 홍채 템플릿을 생성할 수 있다.
- [0083] 홍채 템플릿 생성 장치(100)의 매칭부(150)는 쿼리 해시 코드(쿼리 홍채 템플릿으로써, 템플릿 생성부(140)에 의해 생성된 해시 코드를 의미할 수 있음)와 미리 저장된 해시 코드들 중 적어도 하나와의 매칭 여부를 결정할 수 있다(S150). 이를 통해, 홍채를 이용한 인증을 수행할 수 있다. 즉, 쿼리 해시 코드와 매칭되는 해시 코드가 존재하는 경우 사용자에게 대한 인증이 완료되고, 쿼리 해시 코드와 매칭되는 해시 코드가 존재하지 않는 경우 사용자에게 대한 인증이 실패할 수 있다.
- [0084] 이상에서 설명된 장치는 하드웨어 구성 요소, 소프트웨어 구성 요소, 및/또는 하드웨어 구성 요소 및 소프트웨어 구성 요소의 집합으로 구현될 수 있다. 예를 들어, 실시예들에서 설명된 장치 및 구성 요소는, 예를 들어,

프로세서, 컨트롤러, ALU(Arithmetic Logic Unit), 디지털 신호 프로세서(Digital Signal Processor), 마이크로컴퓨터, FPA(Field Programmable array), PLU(Programmable Logic Unit), 마이크로프로세서, 또는 명령(instruction)을 실행하고 응답할 수 있는 다른 어떠한 장치와 같이, 하나 이상의 범용 컴퓨터 또는 특수 목적 컴퓨터를 이용하여 구현될 수 있다. 처리 장치는 운영 체제(Operation System, OS) 및 상기 운영 체제 상에서 수행되는 하나 이상의 소프트웨어 애플리케이션을 수행할 수 있다. 또한, 처리 장치는 소프트웨어의 실행에 응답하여, 데이터를 접근, 저장, 조작, 처리 및 생성할 수도 있다. 이해의 편의를 위하여, 처리 장치는 하나가 사용되는 것으로 설명된 경우도 있지만, 해당 기술 분야에서 통상의 지식을 가진 자는, 처리 장치가 복수 개의 처리 요소(Processing Element) 및/또는 복수 유형의 처리 요소를 포함할 수 있음을 알 수 있다. 예를 들어, 처리 장치는 복수 개의 프로세서 또는 하나의 프로세서 및 하나의 컨트롤러를 포함할 수 있다. 또한, 병렬 프로세서(Parallel Processor)와 같은, 다른 처리 구성(Processing Configuration)도 가능하다.

[0085] 소프트웨어는 컴퓨터 프로그램(Computer Program), 코드(Code), 명령(Instruction), 또는 이들 중 하나 이상의 조합을 포함할 수 있으며, 원하는 대로 동작하도록 처리 장치를 구성하거나 독립적으로 또는 결합적으로(Collectively) 처리 장치를 명령할 수 있다. 소프트웨어 및/또는 데이터는, 처리 장치에 의하여 해석되거나 처리 장치에 명령 또는 데이터를 제공하기 위하여, 어떤 유형의 기계, 구성 요소(Component), 물리적 장치, 가상 장치(Virtual Equipment), 컴퓨터 저장 매체 또는 장치, 또는 전송되는 신호 파(Signal Wave)에 영구적으로, 또는 일시적으로 구체화(Embody)될 수 있다. 소프트웨어는 네트워크로 연결된 컴퓨터 시스템 상에 분산되어서, 분산된 방법으로 저장되거나 실행될 수도 있다. 소프트웨어 및 데이터는 하나 이상의 컴퓨터 판독 가능 기록 매체에 저장될 수 있다.

[0086] 실시예에 따른 방법은 다양한 컴퓨터 수단을 통하여 수행될 수 있는 프로그램 명령 형태로 구현되어 컴퓨터 판독 가능 매체에 기록될 수 있다. 상기 컴퓨터 판독 가능 매체는 프로그램 명령, 데이터 파일, 데이터 구조 등을 단독으로 또는 조합하여 포함할 수 있다. 상기 매체에 기록되는 프로그램 명령은 실시예를 위하여 특별히 설계되고 구성된 것들이거나 컴퓨터 소프트웨어 당업자에게 공지되어 사용 가능한 것일 수도 있다. 컴퓨터 판독 가능 기록 매체의 예에는 하드 디스크, 플로피 디스크 및 자기 테이프와 같은 자기 매체(Magnetic Media), CD-ROM, DVD와 같은 광기록 매체(Optical Media), 플롭티컬 디스크(Floptical Disk)와 같은 자기-광 매체(Magneto-optical Media), 롬(ROM), 램(RAM), 플래시 메모리 등과 같은 프로그램 명령을 저장하고 수행하도록 특별히 구성된 하드웨어 장치가 포함된다. 프로그램 명령의 예에는 컴파일러에 의해 만들어지는 것과 같은 기계어 코드뿐만 아니라 인터프리터 등을 사용해서 컴퓨터에 의해서 실행될 수 있는 고급 언어 코드를 포함한다. 상기된 하드웨어 장치는 실시예의 동작을 수행하기 위해 하나 이상의 소프트웨어 모듈로서 작동하도록 구성될 수 있으며, 그 역도 마찬가지이다.

[0087] 본 발명은 도면에 도시된 실시예를 참고로 설명되었으나 이는 예시적인 것에 불과하며, 본 기술 분야의 통상의 지식을 가진 자라면 이로부터 다양한 변형 및 균등한 타 실시예가 가능하다는 점을 이해할 것이다. 예를 들어, 설명된 기술들이 설명된 방법과 다른 순서로 수행되거나, 및/또는 설명된 시스템, 구조, 장치, 회로 등의 구성 요소들이 설명된 방법과 다른 형태로 결합 또는 조합되거나, 다른 구성 요소 또는 균등물에 의하여 대치되거나 치환되더라도 적절한 결과가 달성될 수 있다. 따라서, 본 발명의 진정한 기술적 보호 범위는 첨부된 등록청구범위의 기술적 사상에 의해 정해져야 할 것이다.

부호의 설명

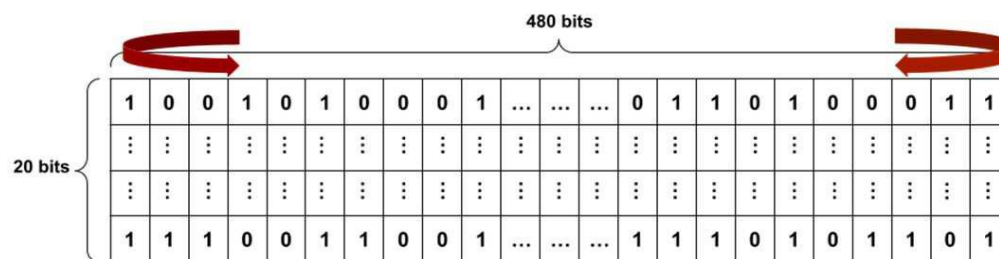
[0088] 100 : 홍채 템플릿 생성 장치
110 : 이미지 획득부
120 : 이진 코드 생성부
130 : 입력 벡터 생성부
140 : 템플릿 생성부
150 : 매칭부
160 : 저장부

도면

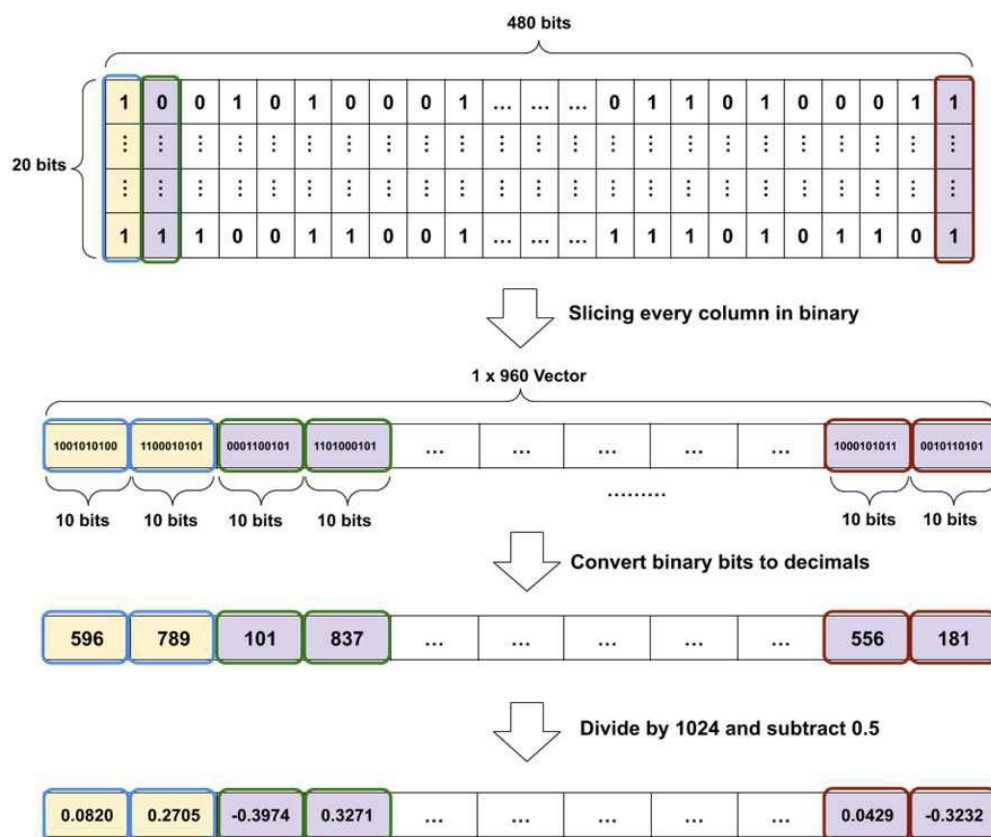
도면1



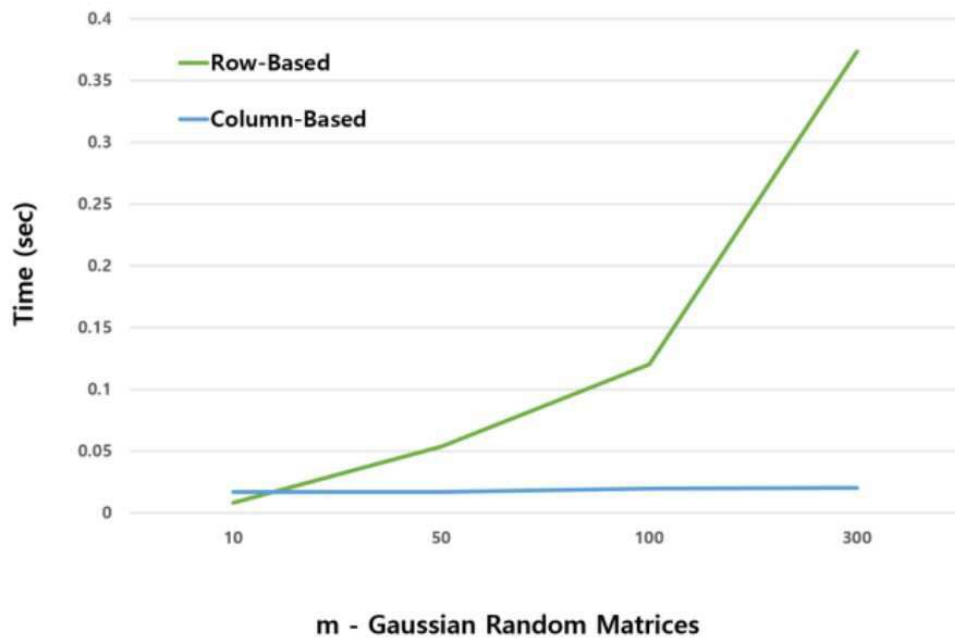
도면2



도면3

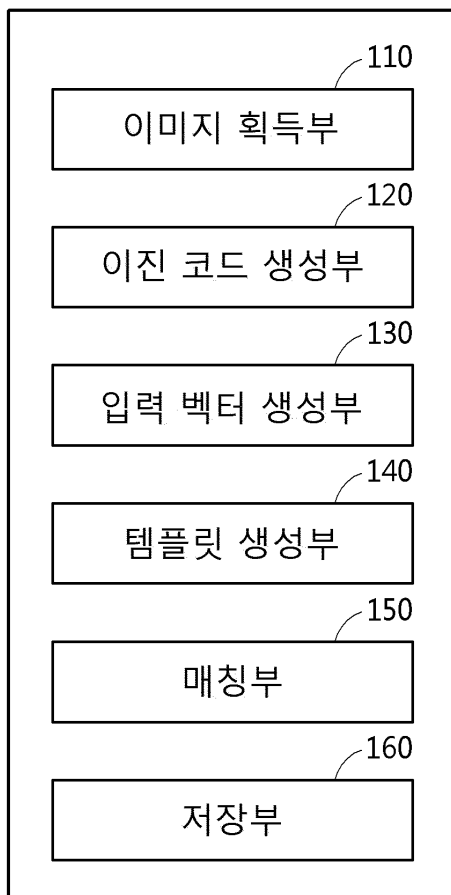


도면4



도면5

100



도면6

