

대용량 지문 데이터베이스에 대한 자동 지문 인식 시스템 개발에 관한 연구

설승진*, 노경현**

*대민전자(주) 연구소장

**대민전자(주) 연구원

e-mail : ssj@freetouch.com

A Study on Automated Fingerprint Identification System for Large Fingerprint Database

Seung-Jin Sul*, Kyung-Hyun Roh**

R&D Center, Daemin Electronics Co., Ltd.

요 약

자동 지문 인식시스템은 대부분의 사용자 인증 과정을 처리하기 위해 사용하는 아이디/비밀번호 방법의 보안상 결점인 비밀번호 유출로 인한 보안 위협을 근본적으로 제거할 수 있다. 그러나 아직까지 수십만에서 수천만 건에 이르는 대용량 지문 검색 서버에 대한 성능이 입증된 것이 없으며, 본 연구에서는 지문분류, 대용량 지문 검색, 다중서버, 다중쓰레드, 인증서버 기술을 갖는 대용량 자동 지문 인식 시스템을 연구하였다. 그리고 10만개의 지문 데이터를 기준으로 지문 검색에 대한 성능을 실험하였다.

1. 서론

자동 지문인식 시스템은 전자 상거래 등의 가장 큰 이슈 중의 하나인 본인 인증 문제를 비밀번호가 아닌 생체 정보를 이용하여 해결하게 함으로써 보안성을 높일 뿐만 아니라 비밀번호를 기억하지 않아도 되는 편리성까지 제공할 수 있다. 또한 네트워크 환경에서 이용할 수 있도록 만든 인증 솔루션은 인터넷 뱅킹, 전자 상거래 등의 e-비즈니스의 확산에 따라 그 응용분야가 더 넓어질 것이다.

그러나 아직 수십만에서 수천만 건에 이르는 대용량 지문 데이터베이스에 대한 검색 시스템은 아직까지 그 성능이 입증된 적용 사례가 극히 드문 실정이다[1][2][4]. 본 논문에서는 지문분류, 대용량 지문인식, 다중서버, 다중쓰레드(multi-thread), 인증서버 기술을 가지는 대용량 자동 지문인식 시스템을 제안하고, 10만개의 지문 데이터를 기준으로 지문 검색에 대한 성능을 실험하였으며, 연구 결과물을 학원 수강생 관리 시스템에 적용하여 시범 운영하였다.

2. 연구 내용

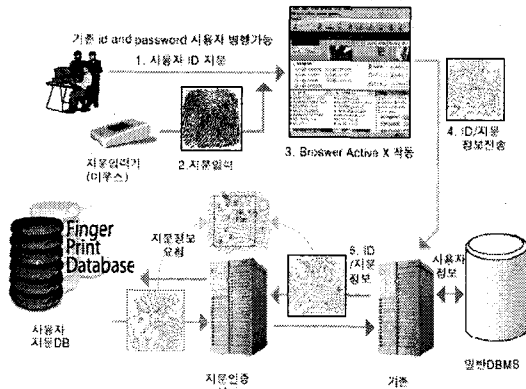
지문인식 시스템에 관한 연구는 입력된 지문 영상에 대한 전처리 과정, 이를 통해 나온 결과 영상에서 지문이 가지는 특징점들을 찾아내는 방법, 찾아낸 특징점들을 바탕으로 참조 지문을 구성하여 저장하는 방법 그리고 매칭 방법에 이르기까지 다양하게 진행되고 있다[2][4]. 본 개발 과제에서는 이미 개발된 자체 지문인식 알고리즘을 바탕으로 대용량 지문 데이터베이스를 위한 자동 지문 인식 시스템을 개발하였다. 기존의 지문인식 알고리즘은 대용량 지문인식 시스템에 보다 적합하도록 적절히 보완하였다. 그리고 지문 분류 알고리즘을 이용하여 지문 데이터베이스의 검색 시간을 최소화하기 위한 색인 기법에 응용하는데, 기존의 분류 알고리즘과는 달리 원본 지문 이미지를 여러 블록으로 나누고 블록에 대한 처리 과정을 거쳐 지문이미지를 분류하여 분류속도 및 정확성을 향상시켰다.

대용량 자동 지문인식 시스템의 개발을 위해 대용량 환경에 적절한 지문인식 알고리즘과 대용량의 지

문을 효율적으로 검색할 수 있는 서버 구조가 필수적이다. 또한 대용량 지문 데이터에 대한 검색의 효율성을 높이기 위해 지역적으로 분산된 데이터베이스에 저장하고, 일차적으로 특징에 따라 분류된 지문의 특성을 반영하여 분산 저장한다. 이 때 미러링이나 자동 백업기능을 통해 지문 데이터의 유실을 방지할 수도 있다.

각각의 검증 서버에서는 다중쓰레드 기법을 이용하여 동시에 여러 부분의 지문 데이터를 동시에 검색하여 성능을 증진시켰다. 또한 입력된 지문을 검증하기 위해 다중 서버에 인식 서버를 가동시키고 실험하였다.

3. 지문인식 처리



[그림 1] 지문인증 시스템 구성도

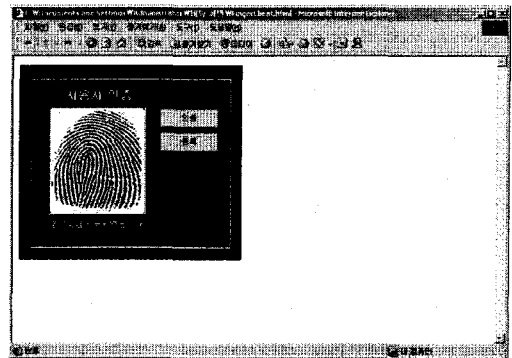
그림 1은 제안 시스템의 지문 인증 과정을 보여준다. 먼저 사용자가 지문 입력기를 사용해 지문을 등록한다. 지문 입력기는 Infineon사의 반도체식 지문 입력 센서를 장착한 지문인증 마우스를 이용하였다. 지문 마우스에서 지문을 받아들이는 것은 인터넷 브라우저의 ActiveX에서 담당한다. ActiveX는 사용자의 지문 정보를 웹서버에 전달하고, 웹서버는 지문 인증 서버에 사용자의 본인 확인을 의뢰한다. 지문 인증 서버는 요청된 지문 정보를 이미 등록된 사용자 지문 DB에서 검색하여 인증 결과를 웹서버에 보내고, 웹서버는 인증이 된 사용자 일 경우 관련된 사용자 정보로 사용자의 요구 처리절차를 수행하게 된다.

지문 인증 서버에서는 사용자 지문 DB를 검색하는 과정은 검색 시간을 단축하기 위해서 다중쓰레드와 다중 서버를 사용한다. 각각의 DB 서버는 등록

요청된 지문의 분류에 따라 저장하게 된다[5][6][7]. 인증시간의 대부분이 DB의 탐색보다 사용자 요청 지문과 저장되어 있는 각각의 참조 지문과의 매칭 시간에 의해 결정이 된다. 검색 시간을 단축하려면 저장된 참조 지문을 일정한 부분만큼 나누어 동시에 매칭 검색을 하는 것이 효과적이다. 지문 분류가 적용되어 분산 저장된 지문 DB는 인증 센터에서 중앙 관리하게 된다. 그리고 각 서버에서 지문을 검색할 때는 다중쓰레드를 사용하여 저장된 지문들을 다시 여러 부분으로 나누어 동시에 검색하여 검색 시간을 단축시킨다. 인증 서버의 사양이 다중 CPU 시스템인 경우에는 각각의 다중 서버를 두는 것보다 하나의 인증 센터에서 다중쓰레드를 사용해 처리하는 것을 비용측면에서 검토해 볼 수 있다.

4. 지문인식 시스템 클라이언트

인증 서버에서 지문 정보를 저장하려면 클라이언트에서 지문의 입력 영상을 받아서 서버로 보내 주어야 한다. 인증 클라이언트는 지문 센서가 달린 지문 입력 마우스와 이 장치를 통해 지문을 읽어 들여 표시하는 클라이언트 프로그램이 필요하다. 이 프로그램은 인터넷 환경에서 사용할 수 있도록 ActiveX로 구현하였다. 개발 도구는 Visual C++를 사용하였다. 그림 2는 사용자의 지문을 인증하는 클라이언트 프로그램을 보여준다.



[그림 2] 지문인식 클라이언트

5. 지문인식 시스템 구현 및 실험

지문 인증 서버는 윈도우 2000 운영체제에서 JDK 1.4를 사용하여 개발하였으며, 개발 도구는 JBuilder 6.0을 사용하였다. 인증 알고리즘은 DLL로 만들어서 Java 인증서버와 JNI(Java Native Interface)를 통해 연결하였다. Java 언어를 사용했기 때문에

JVM(Java Virtual Machine)이 설치될 수 있는 운영 체제이면 어떤 곳에서도 사용할 수 있다. 그림 3은 JNI를 통해 호출되는 알고리즘 메소드(method)를 보여준다.

```
public native int JDFPGetFeature
(int[] img, // 지문이미지 영상
 int[] FeatureArray ); // 지문 특징점

public native int JDFPMatchFingerEx
(int[] FeatureArray1, //기준 특징점
 int[] FeatureArray2, //참조 특징점
 int Level); // 지문 매칭 레벨
```

[그림 3] 지문인증 JNI 메소드

JDFPGetFeature 메소드는 지문 입력 장치에서 읽은 지문 이미지 영상을 받아서 특징점을 추출한다. 여기서 추출된 특징점은 DB에 저장되어 사용자 지문 정보로 사용된다. 현재 하나의 지문당 약 200 바이트 정도의 특징점 데이터가 추출된 특징점 정보를 이용하여 구성된다. JDFPMatchFingerEx 메소드는 기준 특징점과 참조 특징점을 입력받아서 지문 매칭 레벨에 따라서 두 지문을 매칭하고 그 결과 점수에 따라 본인 여부를 가리게 된다. 이 때 인증 레벨에 따라서 오거부율(FRR)과 오인식률(FAR), 인증 시간이 달라지기 때문에 적절하게 구성해야 최적의 성능을 낼 수 있다.

JDFPGetFeature에서 추출된 지문의 특징점 정보는 자체 암호화 알고리즘으로 암호화하여 지문 정보의 유출을 방지하였다. 이 특징점들은 사용자 지문 정보로서 DB에 저장되는데 DBMS는 마이크로소프트사의 SQL Server 2000 Enterprise를 사용하였다. 인증서버와 지문 DB의 인터페이스를 위해 JDBC를 사용했다. JDBC는 Java에서 동일한 방식으로 다양한 DBMS를 사용할 수 있게 하는 연결 방식이므로 SQL Server 뿐만 아니라 인증 서버 사용환경에 적절하게 다른 DBMS도 적용할 수 있다.

인증 실험은 윈도우 2000 서버 운영체제를 사용하는 2대의 Pentium 4(CPU 1.7GHz, RAM 256MB) PC에서 수행되었다. 실험에 사용된 지문 이미지는 직접 무작위로 수집한 성인 지문 이미지 1,000개와 NIST 4 지문 이미지 4,000개를 사용하였다. NIST 이미지는 사용된 지문 입력기에서 읽어들이는 이미지와 동일하도록 적절한 스케일링(scaling)과 크기 조정 단계를 거쳐 반도체식 지문입력 센서에서 획득되

는 지문이미지의 크기인 224x288 픽셀로 수정하여 사용하였다. 이들 5,000개의 지문 이미지를 전체 DB가 10만 레코드가 되도록 임의 중복하여 각 레코드에 배열하였다. 전체 10만개를 4 분류로 지문 분류했을 때의 분포는 표 1과 같다.

분류	개수	백분율 (%)
공상문	27,280	27
와상문	19,960	20
좌제상문	19,660	20
우제상문	33,100	33
전체	100,000	100

[표 1] 지문 분류 분포

실험 방법은 10만개의 지문 이미지를 1 대의 인증 서버와 2 대의 인증서버를 사용했을 때를 나누어서 실행하였다. 각 인증 실험에는 10회와 100회를 실행하여 평균치를 구하였다. 또한 다중서버와 다중스레드를 사용한 경우와 단일서버, 단일스레드를 사용한 경우의 성능을 비교하였다.

인증 서버에서 인증을 시도할 때 각 인증 서버는 다중스레드를 이용하여 지문을 검색하는데 한 스레드가 인증이 되었으면 다른 스레드는 도중에 작업을 멈추어야 한다. 이런 동기화 작업을 위해 synchronized 키워드를 사용하였고, 모든 스레드가 안전하게 공유 변수를 접근할 수 있도록 구성하였다.

```
class MyShareVar {
    private boolean isFind = false;
    private String threadName;
    public boolean getIsFind() {
        return isFind;
    }
    public synchronized void
        setIsFind(boolean value) {
        isFind = value;
    }
    public String getThreadName() {
        return threadName;
    }
    public void setThreadName(String value) {
        threadName = value;
    }
}
```

[그림 4] 스레드 공유 클래스

그림 4는 사용된 클래스(class)를 보여준다. 모든 스레드는 각 지문 레코드에 대해 인증 작업을 시작하기 전에 isFind 값을 체크하게 된다. isFind가

true이면 자신의 모든 작업을 종료하게 된다. 먼저 인증을 성공하여 결과가 true이면 isFind를 true로 설정한다.

실험 결과는 다음과 같다. 표 2는 다중 쓰레드를 사용하는 단일 서버와 다중 서버의 실험 결과를 보여준다. 단중 서버를 사용하는 것이 인증 시간이 더 단축된 것을 보여준다. 2개의 인증서버를 사용했을 때 결과가 2배 이상 차이가 생기지 않는 것은 전체 레코드에서 찾고자 하는 지문의 위치가 매번 난수로 발생되어 분포되어 있기 때문이다.

	10회 평균 (초)	100회 평균 (초)
단일 서버	7.56	8.9
다중 서버	5.2	5.36

[표 2] 인증실험 결과

대용량 지문인식 검색 시스템은 다중서버로 구성하였다. 이들은 통합 관리하는 인증 센터가 중앙서버로써 위치하고, 인증 센터는 클라이언트로부터의 인증 요청을 분류에 따라 해당 인증 서버로 전송하고 각각의 인증 서버는 다중 DB 서버를 이용하여 인증을 처리한다. 클라이언트에서 지문을 등록할 때 지문을 4가지로 분류하여 각각의 인증 서버에 배치한다.

6. 결론 및 향후 연구과제

본 시스템은 다중 서버와 다중쓰레드를 사용하여 대용량의 지문 데이터베이스를 대상으로 빠른 검색 시간을 낼 수 있도록 하였다. 인증서버를 다양한 운영체제에서 사용할 수 있도록 Java 언어를 사용하였고, 지문영상에서 추출된 특징점 정보는 유출에 대비하여 암호화하였다. 클라이언트는 가장 많은 클라이언트 시스템을 차지하는 윈도우 운영체제에서 사용할 수 있도록 ActiveX 기술을 사용하여 개발하였고, 웹 환경에서 동작하도록 구현하였다. 또한 인증 센터로부터 지문 데이터베이스까지를 지문 분류를 적용한 다단계로 구성하여 더욱 빠른 검색이 가능하였다. 결과적으로 지문 검색 시간이 10,000 DB 대상에서 6초 이내 구현되었고, 지문분류에서도 95% 이상의 정확도를 얻을 수 있었다. 이 시스템은 인터넷 전자 상거래와 인터넷 뱅킹 시스템과 같은 환경에서 비밀번호를 대치하여 본인 인증으로 지문을 사용하는데 있어서 빠르고 편리한 방법을 제공해 줄 수 있다. 또한 사용자 ID를 제공할 수 없는 범죄수사 시

시스템과 같은 인식 전용 시스템에 효과적으로 사용할 수 있다. 기타 사용자 인증을 필요로 하는 다양한 분야에 적용 가능할 것이다.

향후 연구 과제는 지문 분류의 정확도를 99%까지 올려서 지문의 오분류를 통해 발생하는 인증의 실패 요인을 최소화시키는 것이다. 그리고 10만 DB를 넘어서는 더 많은 지문 DB에 적용하여 검색 시스템의 효율성을 검증하는 것이 필요하다. 그러나 이를 위해 수많은 사람의 지문이 필요한데 결국 많은 수의 다양한 지문 데이터베이스의 확보가 필요하다. 분포된 사용자 지문의 특성에 따라서 검색 시간의 차이가 현격히 날 수 있기 때문에 이에 대한 보완 연구가 필요할 것이다. 여기에 맞추어 적절한 인증 레벨을 부여하기 위한 최적의 실험 조건을 찾아내야 하겠다. 또한 윈도우 뿐만 아니라 다양한 클라이언트 환경에서 인증을 시도할 수 있도록 클라이언트의 개발이 필요하다.

[참고문헌]

- [1] R. Cappelli, D. Maio, and D. Maltoni, "Indexing fingerprint databases for efficient 1:n matching," in Proc. Sixth Int. Conf. on Control, Automation, Robotics and Vision (ICARCV2000), Singapore, Dec. 2000.
- [2] B. Bhanu and X. Tan, "A triplet based approach for indexing of fingerprint database for identification," in Proc. 3rd Int. Conf. Audio- and Video-Based Biometric Person Authentication (AVBPA2001), pp.205-210, June 2001.
- [3] A. K. Jain, S. Prabhakar, L. Hong, and S. Pankanti, "Filterbank-based fingerprint matching," IEEE Trans. Image Processing, vol.9, no.5, pp.846-859, May 2000.
- [4] Nalini K. Ratha, Kalle Karu, Shaoyun Chen, Anil K. Jain, "A Real-Time Matching System for Large Fingerprint Databases," IEEE Trans. Pattern Analysis and Machine Intelligence, vol.18, no.8, Aug 1996.
- [5] A. K. Jain, S. Prabhakar, and L. Hong, "A multichannel approach to fingerprint classification," IEEE Trans. PAMI, vol.21, no.4, pp.348-359, Apr. 1999.
- [6] Cappelli, R, Lumini, A., Maio, D., Maltoni, D., "Fingerprint Classification by Directional Image Partitioning," IEEE Trans. Pattern Analysis and Machine Intelligence, vol.21, no.5, 1999.
- [7] Kalle Karu and A. K. Jain, "Fingerprint Classification," Pattern Recognition, vol.29, no.3, pp.389-404, 1996.