웹 상에서의 가상 메이크업 시뮬레이션 시스템 설계 및 구현

(Design and Implementation of Web-based Virtual Makeup Simulation System)

손 형 도 † 이 재 협 ††
(Hyung-do Shon) (Jae-hyub Lee)

요 약 본 시스템은 웹 상에서 가상으로 메이크업과 헤어스타일 그리고 액세서리 등을 사용자 자신의 얼굴에 적용시켜 볼 수 있는 시뮬레이션 시스템이다. 시뮬레이션의 결과는 데이타베이스에 저장되며 프린터로도 출력이 가능하다. 본 시스템은 자바 애플릿으로 제작되어 서비스가 되므로 웹 브라우저 이외의 클라이언트 프로그램은 필요하지 않으며, 웹 브라우저에 자바 플러그인만 설치되어 있으면 실행이 가능하다. 윈도 2000의 IIS 웹 서버와 MS-SQL7 데이타베이스를 이용하여 구현하였으며, 자바 애플릿과 데이타베이스의 연결은 상용화된 JDBC Driver Type 4를 이용하였다. 구현된 VMSS는 웹 기반의 서비스 제공을 위한 새로운 컨텐츠 개발의 유형을 제시하였으며, 다양한 분야에 대한 응용이 가능할 것으로 생각된다.
키워드 : 자바, 애플릿, 가상, 시뮬레이션

Abstract The VMSS(Virtual Makeup Simulation System) is a virtual simulation system which can apply makeup, hair style, accessory to user's own face image through internet connection. The simulation result can be stored to server side DB and also can be printed out on client side printer. It is implemented using Java so, clients don't need any other client program to run the VMSS except Web browser and Java Plug-in installed on browser. IIS(Internet Information Server) Web server of Window 2000 and MS-SQL 7 are used to develop the overall system. Commercial JDBC Driver Type 4 is used to connect the applet and DB. It suggests a new direction of Web-based contents service development and can be applied to many other areas.
Key words : Java, Applet, Virtual, Simulation

1. 서 론

우리는 21세기 디지털 혁명을 이끌어 가는 이른바 닷컴 기업들이 주도하는 생활을 실감하고 있다. 이러한 시점에서 웹 사이트를 운영하고 관리하는 회사들의 가장 큰 관심사는 무엇보다도 사용자들의 흥미를 유발시킬 수 있는 웹 컨텐츠 제작에 있을 것이다. 그리고 웹 컨텐츠는 사이트에 접속한 방문자의 흥미를 유발시키면서 그들의 요구사항을 충족시켜야 한다.

본 시스템은 이러한 웹 컨텐츠 중의 하나로 메이크업과 헤어스타일 그리고 액세서리 등을 웹 브라우저 상에서 적용 시켜볼 수 있는 시스템으로써, 직접 자신의 얼굴 이미지 위에 화장을 해 보거나 헤어스타일을 바꾸거나 또는 액세서리(안경, 모자 등의 장신구)들을 적용시켜 볼 수 있게 되어있다. 그리고 자바 애플릿을 이용하여 개발하였기 때문에 사용자에게 실시간으로 반응할 수 있어 마치 웹 상에서가 아닌 일반 애플리케이션 프로그램을 사용하는 것처럼 작업할 수 있는 장점이 있다.

본 논문은 2장에서 시스템의 개요에 대하여 설명하고, 3장에서는 본 시스템을 실제로 어떻게 구현하였는지에 대하여 언급하며, 4장에서는 시스템에 대한 고찰과 발전방향에 대하여 논하고, 마지막으로 5장에서 결론을 맺는다.

† 정 회 원 : 아산정보기능대학 정보통신시스템과 교수
shd@kopo.or.kr
†† 정 회 원 : 한국기술교육대학교 정보기술공학부 교수
jae@kut.ac.kr
논문접수 : 2001년 6월 8일
심사완료 : 2001년 12월 27일

2. 시스템 개요

2.1 시스템 구성

본 시스템에는 그림 1과 같이 웹 서버와 데이타베이스 서버가 운영되고 있으며, 각 서버는 독립적으로 운영될 수도 있고 같은 서버에서 운영될 수도 있다. 현재 본

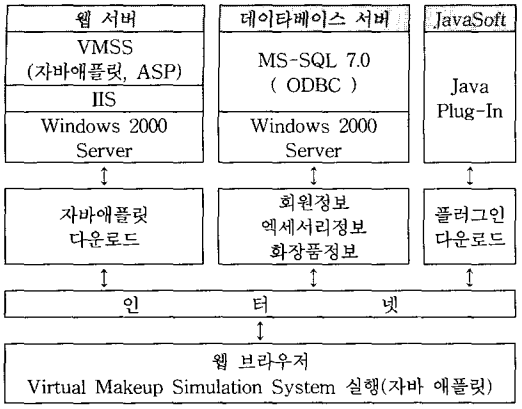


그림 1 시스템 구성도

시스템은 웹 서버와 데이타베이스 서버를 동일한 시스템에서 운영하도록 구성되어 있다.

VMSS는 플랫폼 독립적인 자바로 개발되었기 때문에 Windows NT나 Windows 2000 Server, 리눅스 또는 솔라리스 컴퓨터 모두에서 운영될 수 있지만 현재 본 시스템은 Windows 2000 Server에 데이타베이스 서버로는 MS-SQL 7 그리고 웹 서버로는 IIS(Internet Information Server)[1]를 사용하고 있다.

사용자가 본 시스템에 처음으로 접속할 경우 자동으로 자바 플러그 인, 즉 JRE(Java Runtime Environment)가 다운로드 되어 설치된 후 실행이 되므로 사용자 측면에서 필요한 소프트웨어는 웹 브라우저뿐이며, 넷스케이프나 익스플로러를 모두 지원한다. 본 시스템을 등록 회원제로 운영할 경우 로그인 과정을 거쳐 자신의 이미지를 업로드 하여 사용하게 된다. 회원 로그인 및 이미지 업로드 부분은 ASP로 구현되었으며, 시뮬레이션을 위한 주 프로그램은 자바 애플릿으로 구현되어 있다.

2.2 시스템 흐름도

그림 2는 사용자가 서버에 접속하여 로그인을 한 후, 자신의 얼굴 이미지를 서버에 업로드하고 주 프로그램(자바 애플릿)을 실행하게 되면, 웹 브라우저에서는 자바 플러그인이 설치되어 있는지 확인한 후 설치되어 있지 않으면 자동으로 다운로드 하여 설치한 다음 프로그램을 실행하게 되는 과정을 보여준다.

본 시스템이 실행되면 우선 시뮬레이션을 적용할 이미지를 선택한다. 사용자 자신의 이미지를 업로드 하여 사용할 수도 있으며 시스템에서 제공되는 모델 이미지에 적용할 수도 있다. 본인의 사진(이미지)을 이용하면 시뮬레이션 결과를 4개까지 데이타베이스에 저장할 수 있고, 시스템에서 제공하는 모델을 이용하게 되면 저장

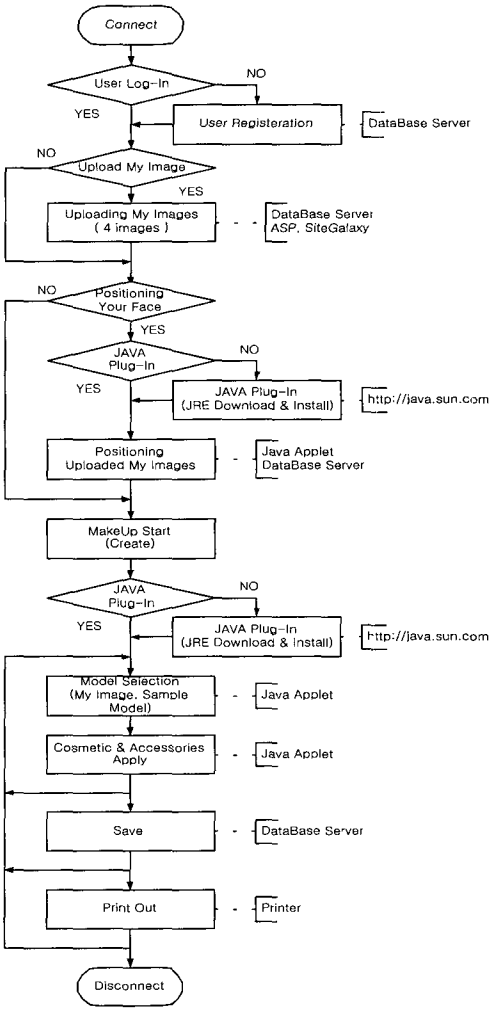


그림 2 시스템 흐름도

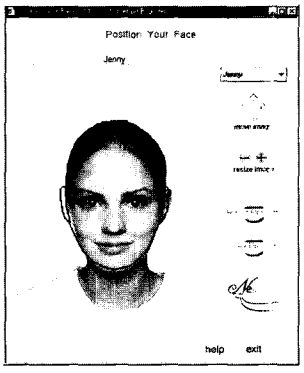


그림 3 이미지 위치 정렬

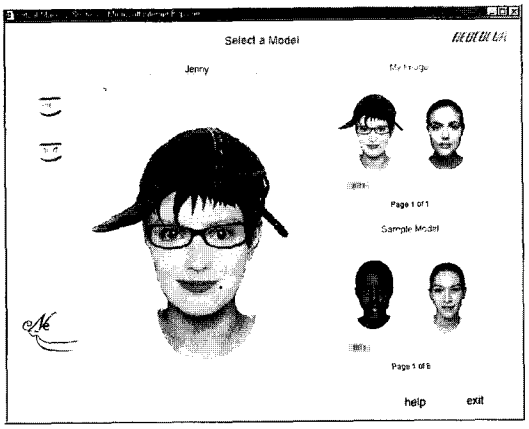


그림 4 모델 선택

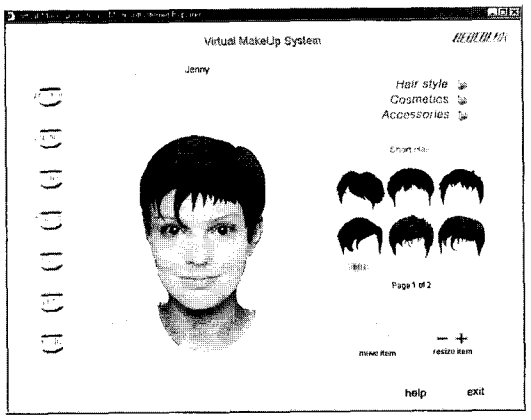


그림 5 메이크업 시뮬레이션

은 되지 않는다. 이미지를 업로드하기 위해서는 그림 3과 같이 이미지를 표준위치에 정렬시킬 필요가 있다. 그리고 그림 4와 같이 모델을 선택한 후에 비로소 그림 5에서와 같이 메이크업, 헤어스타일 선택 그리고 액세서리 적용을 할 수 있게 되며 데이타베이스로의 저장과 프린터 출력도 할 수 있다.

2.3 시스템 기능

본 시스템의 주요 기능은 크게 헤어(Hair)와 화장(Cosmetic) 그리고 액세서리(Accessories)로 구분되어 있으며, 각 기능에 대한 세부 기능이 구현되어 있다. 세부 기능에는 적용할 수 있는 내용이 모두 다르게 되어 있는데, 적용할 수 있는 내용으로는 브러쉬(Brush)의 크기, 아이템(헤어스타일, 모자, 안경)의 크기 및 위치 이동, 헤어스타일의 색상, 지우개 등이 있다. 표 1은 본 시스템의 세부 기능에 대한 요약이다.

표 1 시스템 기능

기 능		아이템		헤어 색상	화장				지우개
		이동	크기 조절		팬슬	브러쉬	립스틱	아이 브러쉬	
Hair	Hairstyle								
	Short	O	O	O					
	Medium	O	O	O					
	Long	O	O	O					
	Highlight					O			O
Cosmetic	Foundation					O			O
	Coverup					O			O
	Eyeshadow lid							O	O
	Eyeshadow crease							O	O
	Eveliner				O				O
	Evebrows				O				O
	Lipstick						O		O
	Lipliner				O				O
	Blush					O			O
	Beautymark				O				O
accessories	Hat	O	O	O					
	Glasses	O	O	O					

메이크업을 할 경우 메이크업 순서에 따라 화면에 겹쳐서 나타나야 할 부분이 있다. 예를 들어 Eyeliner를 적용할 경우 Foundation 부분이 같이 나타나야 하며, 표 2는 기능별 관계를 나타낸다.

표 2 기능별 관계

번호	기 능	관계	번호	기 능	관계
①	Hairstyle	③+④	⑧	Eyebrows	③+④
②	Highlight	③+④+①	⑨	Lipstick	③+④
③	Foundation	④	⑩	Lipliner	③+④+⑨
④	Coverup		⑪	Blush	③+④
⑤	Eye Lid	③+④	⑫	Beautymark	③+④
⑥	Eye Crease	③+④+⑤	⑬	Hat	①+③+④
⑦	Eyeliner	③+④	⑭	Glasses	

2.4 시스템 개발 환경

본 시스템은 JDK(Java Development Kit) 1.3을 이용하여 스윙 애플릿[2]으로 개발하였다. 스윙 애플릿은 HTML파일의 <applet>태그에서 실행이 되지 않고 <embed>태그와 <object>태그에서 실행이 되는데, 이러한 이유에서 htmlconverter[3] 툴을 이용하여 <applet> 태그를 변환하여 사용하였다.

데이타베이스는 Windows 2000 Server용 MS-SQL7 [4]을 이용하였고, 자바와의 연동은 JDBC[5]를 이용하였다. 자바 애플릿에서 데이타베이스에 직접 접근하기

위하여 i-net software사의 i-net SPRINTA(tm) 2000 JDBC Driver Version 3.05 for MS SQL Server[6]을 사용하였다. 따라서 본 시스템을 다른 플랫폼이나 웹 서버에서 운영하더라도 별도의 설정이나 재 컴파일은 필요하지 않다.

3. 시스템 구현

3.1 브러쉬 구현

3.1.1 인덱스 컬러 모델

자바를 이용하여 화면 컬러를 표시하는 방법 중 하나는 각 픽셀에 RGB 컬러 데이타를 저장하는 방법이며, 인덱스 컬러 모델을 사용하면 픽셀 컬러 데이타를 저장하는데 편리한 모델을 제공한다. 즉 픽셀 데이타는 RGB 및 알파 값(투명도)을 포함하며, 컬러를 제한하지 않으면서 픽셀 데이타의 저장 크기를 줄일 수 있다. 인덱스 컬러 모델 객체를 작성하려면, *IndexColorModel* 클래스의 생성자에서 비트수, RGB 및 알파값 데이타를 이용하여 생성한다[7].

3.1.2 인덱스 컬러 모델을 이용한 이미지 만들기

인덱스 컬러 모델을 이용하여 이미지를 만드는 방법 중의 하나는 MemoryImageSource 클래스를 이용하여 정수 픽셀 값으로 작성하는 것이다. 그러나 이 방법은 미리 모든 이미지의 데이타를 지정한 다음 *createI*-mage() 메소드를 사용하여 Image 객체를 만들어야 한다는 단점이 있다. 하지만 BufferedImage 클래스를 이용하면 인덱스 컬러 모델을 작성 한 후 BufferedImage 객체를 생성하여 사용할 수 있다. 이 방법은 각 픽셀 데이타의 추출과 지정을 자유롭게 할 수 있기 때문에 본 시스템은 BufferedImage 클래스를 이용하여 브러쉬 기능을 구현하였다.

3.1.3 브러쉬 페인팅을 위한 이미지 생성

브러쉬 페인팅 기능으로는 Highlight, Foundation, Coverup, Eyeshadow Lid, Eyeshadow Crease, Eyeliner, Eyebrows, Blush, Beautymark 등이 구현되어 있다.

코드 1은 브러쉬 페인팅을 위한 이미지 생성 부분의 소스를 보여준다. 인덱스 컬러 모델은 해당 RGB에 대해 알파 값이 0부터 255까지 총 256가지의 색상을 표현할 수 있다. 즉, 한가지의 RGB에 대해 256단계의 알파 값을 가지는 모델이 생성되며 알파 값을 이용하여 색상의 농담을 조정함으로써 브러쉬가 여러 번 적용될 경우 좀 더 진하게 표시될 수 있다.

newcolor 변수에 색상이 지정되면, 해당 RGB값으로 IndexedColorModel 클래스의 icm 객체를 생성하고, 이것을 이용하여 BufferedImage 클래스의 image 객체를 생성한다.

```
Color newColor = new Color(newcolor);
int r = newColor.getRed();
int g = newColor.getGreen();
int b = newColor.getBlue();
byte R[], G[], B[], A[];

for(int i=0; i<256; i++)
    R[i]=(byte)r; G[i]=(byte)g; B[i]=(byte)b; A[i]=(byte)i;

IndexColorModel icm = new IndexColorModel(8,R.length,R,G,B,A);
BufferedImage image = new BufferedImage(panelWidth, panelHeight,
BufferedImage.TYPE_BYTE_INDEXED, icm);
```

코드 1 브러쉬 페인팅을 위한 이미지 생성

3.1.4 브러쉬 페인팅의 농담 조정

브러쉬가 덧칠이 되었을 경우 이미 해당 픽셀에 지정되어 있는 알파 값을 추출하여 알파 값을 증가시킨 후 재 지정을 함으로써 농담을 조정한다. 여기에서 알파 값은 7E가 최대값이 된다.

```
int pixel = colorImage.getRGB(x,y);
int alpha = ( 0xFF000000 & pixel ) + 0x01000000;

int rgb   = 0x00FFFFFF & pixel;
alpha = Math.min(0x7E000000, alpha);
colorImage[position-1].setRGB(x, y, alpha | rgb );
```

코드 2 브러쉬 페인팅의 농담 조절

```
dy = (float)( y2 - y1 );
dx = (float)( x2 - x1 );
mx = dy / dx;
my = dx / dy;

y  = x1 > x2 ? y2 : y1;
for( x = Math.min(x1,x2); x < Math.max(x1,x2); x++ ) {
    int X = (int)x;
    int Y = Math.round(y);
    for(i=X-Xistart; i<=X+Xiend; i++)
    for(j=Y-Yistart; j<=Y+Yiend; j++)
            drawRGB(i,j,sort);
    for(i=X-Xjstart; i<=X+Xjend; i++)
    for(j=Y-Yjstart; j<=Y+Yjend; j++)
            drawRGB(i,j,sort);
    y = y + mx;
}
x  = y1 > y2 ? x2 : x1;

for( y = Math.min(y1,y2); y < Math.max(y1,y2); y++ ) {
    int X = Math.round(x);
    int Y = (int)y;
    for(i=X-Xistart; i<=X+Xiend; i++)
    for(j=Y-Yistart; j<=Y+Yiend; j++)
            drawRGB(i,j,sort);
    for(i=X-Xjstart; i<=X+Xjend; i++)
    for(j=Y-Yjstart; j<=Y+Yjend; j++)
            drawRGB(i,j,sort);
    x = x + my;
}
```

코드 3 브러쉬 페인팅의 두께 조절 및 라인 생성

3.1.5 브러쉬 페인팅 구현

마우스가 드래그 되면 해당 지점에 라인을 그리게 되는데, 이때 실제 메이크업을 하듯이 두텁게 그려져야 한다. 그러므로 Xistart, Xiend, Yistart, Yiend, Xjstart, Xjend, Yjstart, Yjend 변수는 라인의 두께를 결정하게 되고, x1, y1, x2, y2 변수는 라인을 생성하는 좌표가 된다.[8] 생성된 라인의 두께는 메이크업의 종류와 브러쉬의 두께 선택에 따라 결정이 된다.

라인을 그릴 경우 각 픽셀에 대해 그림 6과 같이 가운데는 약간 진하게 알파 값이 지정되므로 전체적인 라인에 있어서 바깥쪽은 안쪽보다 상대적으로 엷게 지정된다. 처음에는 (Xistart, Yistart)에서 (Xiend, Yiend)까지의 알파 값을 증가시키고, 다음으로 (Xjstart, Yjstart)에서 (Xjend, Yjend)까지의 알파 값을 증가시킨다. 그림 7은 브러쉬 기능을 적용하여 메이크업을 한 경우와, 적용하지 않은 경우에 대한 비교이다.

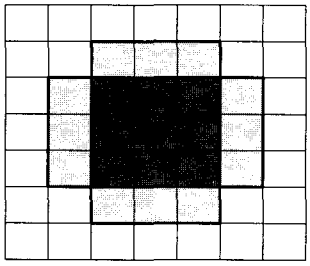
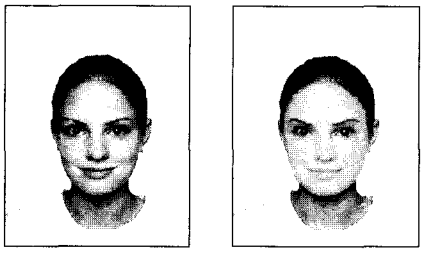


그림 6 브러쉬 페인팅을 위한 알파값 적용



적용전 적용후

그림 7 브러쉬 페인팅 적용(Highlight, Foundation, Coverup, Eyeshadow Lid, Eyeshadow Crease, Eyeliner, Eyebrows, Blush)

3.1.6 지우개 기능 구현

지우개 기능도 브러쉬 페인팅과 동일하게 구현되지만 알파 값에 대한 증감대신 알파 값을 0으로 재 지정함으로써 구현하였다.

3.2 립스틱 및 립라이너의 구현

3.2.1 립스틱 구현

립스틱은 브러쉬 페인팅과는 전혀 다르게 동작한다. 우선 립스틱을 선택하게 되면, 모델의 입술주위가 확대되어 사용자가 쉽게 립스틱을 적용할 수 있도록 한다. 여기에서 마우스로 드래그를 하면 해당 포인트마다 라인을 그린 다음 마우스를 놓으면 해당 다각형 안에 알파 값이 7F인 색상을 채우게 된다. 여기에서 7F는 입술과 립스틱이 가장 잘 어울리는 알파 값으로 농담 조절 값을 의미하며, 이 값은 실험적으로 결정한 값이다.

3.2.2 립라이너 구현

립라이너는 립스틱의 적용상태에 따라 자동으로 구현된다. 립스틱을 그린 다음 립라이너의 색상과 두께를 선택하면 립스틱의 외곽부분에 그려지며 그림 8은 립스틱과 립라이너를 적용시켜본 결과이다.

적용전 적용후

그림 8 립스틱과 립라이너 적용

3.3 JDBC 드라이버를 이용한 데이타베이스로의 접근 구현

데이타베이스에 접근하기 위한 JDBC는 네 가지의 타입으로 구분되는데 본 시스템은 JDBC 드라이버 타입 4인 i-net 사의 sprinta 2000 드라이버를 사용하여 MS-SQL 7.0에 접근한다. 그리고 JDBC를 이용하여 SQL문장으로 데이타베이스의 데이타를 처리하게 된다.

본 시스템은 코드 4와 같이 JDBC를 이용하여 MS-SQL 서버에 접속을 하며 "com.inet.tds. TdsDriver"가 i-net software사의 Type 4 JDBC Driver이다. 드라이버를 메모리로 로드시킨 다음, 주소와 로그인 ID 및 패스워드로 데이타베이스에 접속을 설정하며 SQL문을 이용하여 데이타베이스의 테이블에 추가, 수정, 삭제 작업을 한다.

JDBC Driver Type 1과 2는 JDBC를 이용하여 접근할 경우 클라이언트에서만 가능하기 때문에 CGI나 자바 서블릿(servlet) 그리고 JSP등에 적합하고, Type 3는

```
URL html = applet.getDocumentBase();
String address = "jdbc:inetdae:" + html.getHost() +
                 ":1433?database=MAKEUP&sql7=true";
try {
  Class.forName("com.inet.tds.TdsDriver").newInstance();
  connection=DriverManager.getConnection(address,login,password);
} catch(Exception e) {  }
```

코드 4 JDBC 드라이버를 이용한 데이타베이스 접근

네트워크 상에 분산되어 있는 여러 가지 데이타베이스를 단일 창구로 처리할 경우 적합하지만 VMSS의 용도로는 부적합하다. 따라서 본 시스템에서는 애플릿에서 네트워크를 통해 직접 MS-SQL7에 접속하기 위하여 JDBC Driver Type 4를 사용하였다.[9]

4. 시스템 고찰

현재 온라인상에서 VMSS와 유사한 서비스를 제공하는 사이트는 많지 않으며, 국외의 몇 곳에서 운영되고 있다. 이러한 사이트들의 기능 및 성능을 아래 표 3에 정리하였다.

첫 번째로 www.makeoverstudio.com은 자바스크립트(Javascript)를 이용하여 헤어스타일과 메이크업에 관련된 일부 기능을 제공하고 있다. 이미지는 시스템에서 제공되는 모델만 사용할 수 있고 저장과 프린트 기능이 없으며 단순한 시뮬레이션만 지원한다.

두 번째로 www.ezface.com은 웹 브라우저 이외의 별도 클라이언트 프로그램을 다운받아 사용하게 되어있다. 기능 면에서 보면 본 시스템에서 구현한 대부분의 기능이 있으며, 전자상거래 위주로 구성되어 있다. 부수적인 기능은 많은 편이지만 전체적인 시뮬레이션 수준은 만족할 만 하지 못하다. 즉 시뮬레이션 기능보다는 전자상거래 기능에 더 큰 비중을 두고 있다.

세 번째로 www.compucloz.com은 VMSS와 같이 자바 애플릿으로 구현되어 있지만 기능 면에서 다양하지 못하고 시뮬레이션의 수준이 그다지 정교하지 못한 단점이 있다.

네 번째로 www.ukhairdresser.com은 CGI를 사용하여 헤어스타일과 모델만을 차례로 선택하게 되어 있는 사이트이다.

상기에서 살펴본 바와 같이 대부분의 사이트는 헤어스타일과 메이크업 그리고 액세서리 기능을 제공은 하지만 사용기술에 있어서 실시간으로 속도의 제약 없이 실행되지 않고 있으며, 본 시스템과 같이 사용자가 직접 브러쉬하거나 자신의 이미지를 사용하여 시뮬레이션하여 저장하고 프린터로 출력하는 기능이 없다. 즉, 가상 메이크업을 제공해 주지 못하고 있으며 CGI나 자바스크립트를 이용하여 구현된 사이트는 실행 속도면 에서 매우 느리고 기능 면에도 많은 제약을 갖고 있다.

본 시스템은 이러한 점에서 사용자 자신의 이미지를 이용하여 메이크업과 헤어스타일, 액세서리를 적용시켜 볼 수 있는 장점을 갖고 있으며 사용자 인터페이스면에서 보아도 여타 시스템보다 우수하다고 할 수 있다.

그리고 일반 사용자들을 대상으로 실용성 측면에서 실험을 한 결과 본 시스템의 한글화로 접근성을 증대시키고, 시스템 사용의 이해에 도움이 되는 도움말 기능의 강화 그리고 웹상에서 실행이 되는 시스템이기 때문에 데이타베이스를 초기화하는데 시간이 걸리는 문제 등이 지적되었다.

표 3 시스템 비교

항목 \ 사이트명	make-overstudio	ezface	compucloz	ukhair-dresser	VMSS (본 시스템)
구현기술	Javascript	Client Pro.	Java Applet	CGI	Java Applet
헤어스타일	하	상	하	중	상
액세서리					
메이크업	하	중	하	하	상
시스템제공모델	○	○	○	○	○
사용자 이미지	×	×	×	×	○
DB 저장	×	○	×	×	○
프린터 출력	×	○	×	×	○
전자상거래	×	○	×	×	×
직접 Brush	×	×	×	×	○
GUI 수준	하	상	하	중	상
사용의 편리성	하	하	상	중	상

5. 결 론

본 시스템은 자바 애플릿으로 구현된 가상 메이크업 시뮬레이션 시스템으로 웹 브라우저 상에서 직접 실행되는 장점을 가지고 있을 뿐만 아니라 소프트웨어의 업그레이드 시에도 사용자가 프로그램을 재 설치 할 필요가 없다. 이러한 방식은 기존의 소프트웨어 배포 방법에 비해 신속하고 비용이 저렴할 뿐만 아니라, 단지 소프트웨어만 판매하고 끝나는 것이 아니라 지속적인 회원관리가 가능하며 광고를 할 수 있는 장점이 있다. 인터넷을 이용한 생활이 점점 더 확산되고, 의존적으로 될수록 이러한 방법은 매우 크게 각광을 받게 될 것이다.

또한, 화장품이나 액세서리 제조 회사의 상품 판매용으로도 사용이 가능하다. 새로운 상품이 나올 때마다 제조 회사는 웹에 제품을 소개하고 구매자는 웹에 연결하

여 직접 사용해 보고 바로 구매할 수 있기 때문이다. 또한 메이크업을 교육하는 교육기관의 교육용으로도 사용이 가능할 것이다.

추후 본 시스템은 실시간 전자 상거래 기능 및 3차원 모델링 기능 등의 추가, 메이크업 이미지와 배경이미지의 합성 등의 기능들이 필요하며, 메이크업 뿐만 아니라 패션 분야에도 적용, 발전시켜 나아갈 수 있을 것이다.

참 고 문 헌

[1] Frisch, Eleen, "Essential Windows NT System Adiministration", O'Reilly, 1998
[2] Kathy Walrath & Mary Campione, "The JFC Swing Tutorial", 2000
[3] JavaSoft Homepage, "http://java.sun.com"
[4] Dooley, Sharon, "SQL Server 7 Essential Reference", New Riders, 2000
[5] G. Hamilton, R.Cattell, M. Fisher, "JDBC Database Access with Java", Addson Wesley, 1997
[6] I-net software Homepage, "http://www.inetsoftware.de"
[7] Jonathan Knudsen, "JAVA 2D Graphics", O'Reilly, 1999
[8] Foley, van Dam, Feiner, Hughes, "Computer Graphics : Principles and Practice", Addson Wesley, 1990
[9] M. Hughes, M. Shoffner, D. Hammer, "JAVA Network Programming", Manning, 1999

손 형 도

1996년 한국기술교육대학교 정보통신공학과 졸업(공학사). 1999년 한국기술교육대학교 전기전자공학과 졸업(공학석사). 2001년 ~ 현재 아산정보기능대학 정보통신시스템과 전임강사. 관심분야는 자바, 컴퓨터네트워크

이 재 협

1984년 홍익대학교 화학공학과 졸업(공학사). 1987년 일리노이공대 전산학과 졸업(공학석사). 1992년 일리노이공대 전산학과 졸업(공학박사). 1993년 ~ 현재 한국기술교육대학교 정보기술공학부 부교수. 관심분야는 컴퓨터그래픽스, 멀티미디어