

### WebCAM을 이용한 지역관리 시스템

하근희\*, 김도년, 조동섭  
이화여자대학교 컴퓨터 학과

### Local Area Management System Using WebCAM

Ha Geun-Hee, Kim Do-Nyun, Cho Dong-Sub  
Dept. of Computer Science and Engineering, Ewha Womans University

**Abstract** - As there comes out better and better solutions for delivering multimedia data on the internet, the number of WebCAM sites which transmits pictures of any spot to the user's system is getting increased. WebCAM service over the internet can be compared with the CATV in that it is beyond the spatial constraints.

In this project, we construct a WebCAM site which performs not only the fundamental WebCAM functions of capturing images and of transmitting them periodically but also various kinds of modules carrying out tasks such as image processing, remote control of the camera, processing multiple user requests, making panorama views and so on using Java which is the most popular language for the internet or CGI. The past images are stored in a database implemented for the WebCAM system, so the system can be extended to get over the constraints in time also.

### 1. 서 론

WebCAM 기술은 디지털 카메라를 이용해 캡처한 사진을 인터넷을 통해 일정 시간 간격으로 전송 받을 수 있도록 하는 서비스를 구현한다. WebCAM 사이트를 구축함으로써 웹을 통한 라이브 이미지 획득으로 원격지 방문이 가능하며, 이를 현장 작업의 관리 및 지역 관리, 웹 캐스팅 또는 개인적인 모니터링 등의 다양한 용도로 이용할 수 있다.[1]

대개의 WebCAM 사이트에서는 이미지 재적재가 자동으로 이루어지도록 하고 있는데, 이러한 이미지 재적재는 HTML 헤더에서 META language나 CGI를 통해서, 또는 Java자체의 코드를 이용하여 구현될 수 있다 [1,2,3]. 좀 더 많은 기능이 제공되는 사이트에서는 카메라를 사용자의 요청에 따라 조절할 수 있는 패널을 제공하기도 한다. 본 연구에서 구축하는 WebCAM 시스템은 이러한 기본적인 서비스들을 포함해서 이미지 프로세싱, 다중 사용자 요구의 처리, 파노라마 뷰 등을 지원한다. 그리고, 데이터베이스를 구축하여 효율적으로 데이터들을 저장함으로써 지나간 이미지에 대한 접근이 쉽게 이루어지도록 하고 있다. 제어 명령을 처리할 때 고려해야 할 문제로 다중 요청에 대해 어떤 순서로 처리를 할 것인가와 어떻게 사용자로 하여금 자신의 요청이 수행되었음을 알리는가 등이 있다. 우선 일반적인 WebCAM의 시스템 구성도와 응용사례들에 대하여 간단히 언급하고 본 연구에서 구현하는 WebCAM 시스템의 설계와 구성 그리고 구현방식 등을 비교 분석한다.

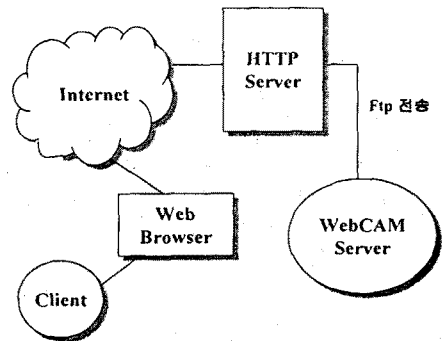
### 2. 본 론

#### 2.1 WebCAM의 현황

국외의 WebCAM 사이트는 그 수가 수천에 이를 정도로 일반화 되었지만 국내에서는 아직도 흔하게 볼 수 있는 장치는 아니다.

초기에는 자동 재적재나 간단한 원격 조절 등의 기술만이 제공되더니 최근에는 사이트의 특성에 따라 여러 가지 기능들이 추가되어 캡처한 사진으로 다양한 이미지 변환을 시도한다거나 사용자 요청에 따른 카메라의 조절이 훨씬 고급화된 다양한 기능들이 제공되고 있다. 가장 기본적인 WebCAM 시스템은 이미지를 캡처하여 이를 사용자에게 전달되도록 하는 원격 또는 로컬 저장 매체로 전송하는 WebCAM 서버와 웹 브라우저를 통해 이를 요청할 클라이언트, 그리고 이 둘 사이를 연결하는 인터넷으로 구성된다.

[그림 1]은 이를 도식화한 것이다.



[그림 1] 일반적인 WebCAM 시스템 구성도

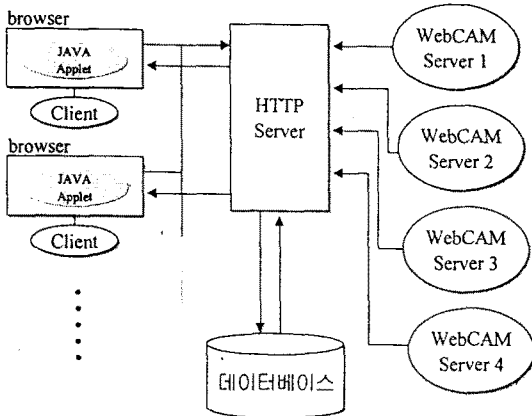
WebCAM 서비스는 인터넷을 통해 이루어지기 때문에 클라이언트들의 요구를 실시간으로 수용할 수 있는 시스템의 구성과 부가적인 서비스들에 대한 효율적인 수행 등을 고려해야 한다.

#### 2.2 시스템 구성

WebCAM 서버 측에서 해 주어야 할 일은 우선 이미지를 캡처할 수 있는 장치를 마련하는 것이다. 카메라를 부착한 캡처 장치를 통해 이미지를 적절한 포맷으로 저장한 후 인터넷을 통해 이를 서비스 할 수 있는 웹 서버로 전송해야 한다. 인터넷에 적합한 포맷은 JPG나 JIF로서 이들 이외의 포맷을 다룰 때는 플러그-인이 설치되어 있어야 한다. 본 연구에서는 압축률이 뛰어나면서 웹 상에서의 기본적인 이미지 포맷인 JPG를 채택하고 있다. 실질적으로 이미지를 서비스할 HTTP 서버는 카메라가 부착된 WebCAM 서버와 분리된 리모트 시스템일 수도 있고 WebCAM 서버가 바로 HTTP 서버로서의 역할을 할 수도 있다.

본 연구는 정보통신부 초고속정보통신 응용기술개발 사업 [과제번호 : AB-97-E-0097]연구비 지원으로 수행된 것임. 과제명: 초고속 통신망 을 이용한 인터넷티브 WebCAM/MIC 시스템의 개발

[그림 2]는 본 연구에서 구현하는 시스템의 도식도를 나타낸다.



[그림 2] 제안한 WebCAM 시스템 구성도

사용자의 요청이 있게 되면 JAVA Applet이 사용자의 웹브라우저에 로드되어 실행되고 이 Applet을 통해 사용자와 서버간의 요청과 수행이 인터랙티브하게 이루어진다. Java언어의 socket networking 기능을 이용해 서버와 클라이언트간의 통신 채널을 확립하고 나면 이 채널을 통해 클라이언트의 요구가 서버로 전달되게 된다.

본 연구에서는 지역적으로 분리된 이미지들을 사용자가 선택할 수 있게 되어 있으며 지나간 이미지들은 데이터베이스로 저장되어 이후에 사용자에게 의해 다시 요청될 수 있다.

데이터베이스는 카메라 식별번호, 이미지가 캡처된 날짜와 시각, 장소의 세 가지 필드로 이루어진다. 한 칼럼, 또는 칼럼의 조합으로 테이블에 대한 유일한 식별자를 제공하는 것이 프라이머리 키인데, 카메라 식별번호와 이미지가 캡처된 날짜와 시각의 조합으로 각 이미지에 대한 프라이머리 키를 구성할 수 있다[4].

[그림 3]은 데이터베이스의 필드 구조를 나타낸다.

CAM_ID	Date_Time	Location
1	1998-05-02 14:20:10	Loc1
2	1998-05-19 03:20:30	Loc2
3	1998-05-02 15:50:00	Loc1
...		

Primary Key

[그림 3] Database field 구조

### 2.3. 기능별 모듈 구성

본 연구에서는 캡처 이미지의 전송 및 Server Push 등의 기본적인 WebCAM 기능 외에도, 이미지 프로세싱, 애니메이션 기능, 카메라의 원격 제어 및 파노라마 뷰등을 구현한다.

#### 2.3.1. 이미지 프로세싱 모듈

사용자는 이미지 프로세싱을 위한 Applet을 통해 선택한 이미지에 대해 다양한 영상 처리 모듈들을 두고 있다. 이 모듈은 미디언 필터를 통한 영상의 잡음 제거, 밝기 조절 및 선명도 조절, 평활화 등의 영상 강조 작업과 회전, 확대, 축소 등의 기하학적 처리들을 위한 클래스들로 구현된다. 미디언 필터는 화소의 농도를 오름차순으로 정렬하여 그 중앙 값으로 화소를 구하며, 선명도는 중폭이득을 변화시켜 대조를 강조함으로써 조절한다. 회전이나 확대, 축소 등의 기하학적 처리들은 픽셀의 좌표 매핑으로 구현할 수 있다[5]. 각각의 클래스들은 이미지 프로세싱 툴킷으로서 이용할 수 있도록 되어 있어서 추가적으로 이미지 처리 클래스들을 구현하여 이 Applet에 포함하는 작업이 쉽게 이루어질 수 있다.

#### 2.3.2. Server Push

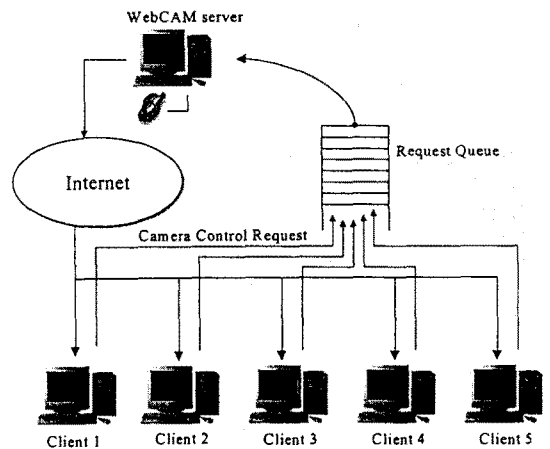
Server Push 기능은 CGI를 이용하는 방법과 자바 코드 내에 포함하는 방법을 모두 구현하여 제공한다. CGI는 Perl로 구현하여 지정된 interval만큼 sleep이 되도록 하고 있고, 자바에서는 Interval을 Applet의 파라미터로 넘겨 주게 되면 flush()와 repaint() method를 이용하여 매 지정된 interval 마다 새로운 이미지가 적재되도록 하고 있다[1,2,3]. 이러한 기능은 클라이언트가 매 번 수동으로 재적재를 해야 하는 부담을 덜어주고 있으며 더욱이 사용자가 여러 연속된 이미지를 데이터베이스로부터 지정하여 이들을 애니메이션화하여 디스플레이하는 기능도 제공된다.

#### 2.3.3. 카메라 원격제어

사용자들은 카메라 자체에 신호를 보냄으로써 그 방향을 조절하거나 Zoom in/out을 조절할 수 있다. 이 작업은 클라이언트 시스템에 적재된 Applet과 WebCAM 서버간의 통신 채널을 형성하여 사용자가 요청한 신호를 병렬포트에 연결된 카메라로 전달함으로써 이루어진다. 각 병렬포트의 핀으로 전달되는 신호에 따른 카메라의 동작은 카메라 의존적이다. 원격제어 시 문제가 되는 점은 여러 사용자가 동시에 카메라 원격제어를 요청했을 경우 이를 어떻게 처리하는 가인데, 이를 위해 큐를 형성하여 허용되는 최대 사용자 수를 제한하고 큐에 적재되는 요청을 순차적으로 처리하는 방식을 택한다.

원격제어 설계 시 사용자 인터페이스와 이를 처리하는 내부 구조를 고려해야 한다.

[그림 3]은 큐를 이용한 다중 사용자의 원격 조정이 이루어지기 위한 서버와 클라이언트 간의 작업을 나타낸다.



[그림 4] 카메라 원격 제어를 위한 사용자 요구 큐

### 2.3.4. 다중 사용자 요구 처리

여러 사용자가 동시에 카메라를 제어하려 할 때 큐를 이용함으로써 요청이 받아들여진 순서대로 작업을 처리한다. 카메라 원격 조정을 제공하는 사이트는 전체 WebCAM 사이트 수에 비하면 아직 미미하다. 대부분은 다중 사용자 요구에 대해서 각 사용자에게 카메라 원격 제어를 위한 일정한 시간을 할당하고 그 시간 동안의 사용자 요청을 받아들여 수행한다. 그러나, 이러한 방식은 할당된 시간을 모두 이용할 필요가 없는 사용자 수의 비율에 따라 효율의 저하가 있게 된다. 따라서, 각 사용자의 요청을 그대로 큐에 적재해 가면서 최대 수용 가능한 요청까지만 저장하면서 이들을 순차적으로 처리해 나간다. 대부분의 원격제어를 위한 사용자 인터페이스는 제어 패널의 조절 버튼을 클릭함으로써 현재 카메라의 위치로부터 상대적인 움직임을 제어하도록 제공되고 있다. 그러나, 사용자가 목표하는 카메라 상태가 현재 카메라 방향과 현저한 차이가 있다면 여러 번의 요청을 거친 후에야 목표한 바를 얻을 수 있다. 이러한 문제를 절대적 위치를 지정할 수 있도록 함으로써 해결하고 있으며, 사용자의 요청이 큐의 어느 부분에 위치하고 있는지와 현재 화면에서 보여지는 이미지가 자신의 요청에 의한 것인지 표시한다.

### 2.3.5. 애니메이션 뷰

데이터베이스를 구축함으로써 이전의 이미지들을 효율적으로 관리 저장할 수 있으며 사용자에게 언제라도 지난 이미지들을 제공할 수 있다. 이렇게 저장된 이미지들을 통해 이미지 업데이트에 이용한 Server Push 기술을 연속된 이미지 애니메이션화에 적용할 수도 있다. 애니메이션 뷰는 사용자가 애니메이션화 될 이미지들의 범위를 지정하고 이들을 로드하여 연속적으로 디스플레이 하는 작업으로 이루어진다. Server Push와 마찬가지로 애니메이션 뷰도 역시 CGI를 이용한 버전과 자바 코드로 구현한 버전을 모두 제공한다.

### 2.3.6. 파노라마 뷰

파노라마 뷰는 위치적으로 이어지는 여러 장의 이미지들을 연결하여 하나의 이미지처럼 보이게 하여 한정된 크기의 패널 위에서 사용자가 스크롤을 통해 이미지를 둘러 볼 수 있도록 하는 것이다. 실제로 360°의 wide angle을 제공하는 카메라로 캡처한 사진으로 wide surround view를 지원할 수도 있는데, 이 때는 하나의 이미지를 스크롤 하는 개념이 된다.

### 2.3.7 데이터베이스의 구축

현장 사진 뿐 아니라 지난 이미지 제공과 애니메이션 뷰 등을 구현하기 위한 이미지의 저장/관리소로서 데이터베이스를 이용한다.

본 연구에서는 DBMS로 그래픽 혹은 사운드 등의 멀티미디어 데이터를 다루기에 좋은 객체 지향 데이터 베이스 엔진인 postgres를 선택한다. postgres는 사용자가 시스템을 쉽게 확장시킬 수 있도록 하는 클래스, 상속성, 데이터형, 함수의 기본적인 네 가지 구성을 제공하며 부가적으로 강력한 rule 시스템을 제공한다. Postgres에서 사용하는 질의 언어는 PostgreSQL로서 다중 사용자 환경의 클라이언트/서버 데이터베이스에 적합하고 다중 인덱스형 또는 단일 인덱스와 다중 칼럼 인덱스들, 사용자 정의 함수와 연산자, 데이터형이 지원되는 등 다양한 기능이 제공된다.

### 2.4 구현 환경

본 연구에서 WebCAM 서버로 사용한 4대의 PC는 모두 Windows95환경의 Pentium 200Mhz CPU에 Memory 64M이며, HTTP 서버로는 Solaris 2.5.1 운영체제에 UltraSPARC-II, 128M의 환경에서 NCSA가 동작하도록 하여 원활한 서비스가 이루어지도록 하고 있다. 4 대의

WebCAM 서버에는 Snappy, AVI Capture Board, Snazzy Capture Board 등의 다양한 캡처 장치를 장착하여 각각을 카메라와 연결하고 있다. WebCAM 서비스를 위한 소프트웨어로는 이미지 캡처와 ftp 전송을 지원하는 Webcam32를 이용하고 있는데, 이미지 캡처 시에는 비디오 캡처 드라이버에 따라서 여러 가지 형식이 지원되며 캡처 시간 간격도 다양하게 지정해 줄 수 있다. 서버간의 이미지 전송은 T1급 LAN상에서 이루어지며 클라이언트는 브라우저만 있다면 인터넷에 연결될 수 있는 어떤 환경에서도 서비스를 받을 수 있다.

DBMS으로는 Postgress를 사용하고, Java Applet을 구현하기 위한 툴로 JDK 1.1.5와 Symantec Visual Cafe를 사용하였으며, 이미지 프로세싱을 위한 모듈들은 Image/J를 참조하였다.

### 3. 결 론

WebCAM 시스템은 공간적 제약을 받지 않는다는 점에서 기존의 CATV와는 본질적으로 다르다. 본 연구에서는 정지 영상만을 다루고 있지만, 네트워크 환경이 충분히 갖추어져 있다면 WebCAM을 통해서 동영상도 제공할 수 있다. 점점 더 많은 멀티미디어 데이터 형식들이 웹을 통해 제공되고 있는 가운데, WebCAM은 영상을 제공하기 위한 하나의 솔루션으로 간주할 수 있다. WebCAM은 현재 많은 곳에서 개인적 취미나 지역의 소개 등 다양하게 응용되고 있으며, 원격리 감시나 건설 현장의 관찰, 관광지 소개 등 보다 실질적인 분야에 적용될 수도 있다.

고속통신망 상에서 구현되는 경우에는 훨씬 효율적으로 동작하는 시스템이 제공될 수 있고, 무엇보다도 Web Camera와 같은 형태의 통신이 제공하는 가능성을 살펴 볼 수 있다.

향후 과제로는 캡처한 이미지에서 문자를 추출하여 인식하는 문자 인식 모듈을 추가하는 작업이 있을 수 있겠다. 이러한 모듈은 교통감시 시스템에서 번호판 인식 등에 효과적으로 이용될 수 있다.

### [참 고 문 헌]

- [1] Jeffry Dwight and Michael Erwin 공저, Using CGI, 정보문화사, special ed., p567-596, 1996.
- [2] Laura Lemay, Web Publishing with HTML 3.2, Sams.net, 3rd ed. p262, 1996.
- [3] Patrick Naughton and Herbert Schildt, The Complete Reference JAVA, 정보문화사, p649-694, 1997.
- [4] C.J.Date, an introduction to Database Systems, Addison Wesley, p79, 6th ed., 1995.
- [5] 이문호, 염재훈 공저, 영상신호처리, 대영사, p115-192, 1996.