

휴대폰에 소프트웨어 추가 없이 영상 스트림을 전송하는 방법 연구

A Study on a Transferring Method of Video Stream to Existing Handphones without Downloading Additional Software

박 대 원* 홍 마 리 아** 김 규 정*** 임 영 환****
Dae-Won Park Maria Hong Kyoo-Jung Kim Young-Hwan Lim

요 약

본 논문은 휴대폰에 소프트웨어를 추가하지 않고 카메라에서 실시간으로 캡처한 영상을 휴대폰으로 전송하는 방법을 제안하고, 실험한 결과를 제시하였다.

본 논문에서는 실시간 영상을 휴대폰으로 전송하는 방법으로 WAP 풀 기반의 영상전송 방법과 WAP 푸시 서비스 기반의 영상전송 방법에 대하여 제안하고, 실시간 영상을 WBMP 이미지로 변환시에 문제가 되는 임계값 설정을 영상의 히스토그램을 사용하여 전체 픽셀 값들에 대한 평균값으로 사용한 방법을 제안하였다.

이 방법을 구현하기 위해서 스트림 엔진과 연동한 메커니즘으로 카메라에서 캡처된 영상을 WBMP 포맷으로 변환하는 영상변환 필터(WBMPFilter)를 구현하고, 무선인터넷상에서 영상을 전송할 수 있는 영상전송 미디움(WirelessMedium)을 구현하였다.

Abstract

This paper proposes a real-time transferring method of captured video stream to existing handphoned without downloading additional software.

For real-time transferring of captured video data to handphoned, two methods, the WAP based pull-method and the WAP based push-method, are proposed in the paper. Also for transcoding of captured frame into a WBMP image, a threshold setting method which uses the average value of total values of pixels in a scene is proposed.

The methods we proposed are implemented in stream engines as video transcoding filter(called WBMPFilter) and video transferring medium(called WirelessMedium). And the experimental system shows 1 frame per 3 seconds transfer of video to existing handphoned.

1. 서 론

세계적으로 이동 통신 가입자가 급속히 증가함에 따라 사용자들은 경보상황 발생시 경보상황을 휴대폰으로 통보 받고, 감시현장의 영상을 휴대폰으로 볼 수 있는 감시제어 시스템을 요구하고 있

는 상황이다. 하지만 경보상황 통보는 SMS를 통하여 휴대폰으로 받을 수 있으나 실시간 영상을 휴대폰으로 보기에는 현재의 무선 인터넷 환경에서는 많은 제약들이 있다.

이런 제약들을 해결하기 위해 무선 인터넷에서의 높은 에러 발생률과 오랜 지연, 협소한 대역폭 등을 극복하고 무선 인터넷 서비스를 원활히 지원할 수 있도록 전송되는 데이터의 양을 감소시킬 필요가 있다. 또한 휴대폰의 낮은 CPU 처리속도, 메모리의 부족, 작은 화면 표시 창, 짧은 전지수명 등의 제한된 자원으로 사용자가 만족할 만한 서비스를 하기 위해서는 저장할 데이터의 양

* 숭실대학교 대학원 컴퓨터학과 석사과정 졸업
castleofsky@dreamwiz.com

** 숭실대학교 대학원 컴퓨터학과 박사과정
maria@media.ssu.ac.kr

*** 숭실대학교 미디어학부 교수
kjkim@comp.ssu.ac.kr

**** 숭실대학교 미디어학부 교수
yhlim@computing.ssu.ac.kr

을 최소화하여 메모리 소모 및 휴대폰에서의 처리량을 줄여야 한다.

본 논문은 휴대폰에 별도의 소프트웨어를 추가하지 않고 카메라에서 캡처한 영상을 휴대폰으로 전송하고자 한다. 그 연구방법으로 WAP 폴 기반과 WAP 푸시 서비스의 기반의 영상전송 방법을 찾아보고, 또한 무선 인터넷에서 휴대폰으로 전송하는 영상의 데이터 양을 최소화하고, 휴대폰의 제한된 자원을 사용하여 그래픽 정보를 표현하기에 적합한 WBMP 이미지로 변환하고자 한다.

2. 문제점 및 연구방향

2.1 실시간 영상을 휴대폰 전송시의 문제점

현재 무선 인터넷 환경에서 실시간 영상 및 동영상을 휴대폰으로 전송하고 재생하기에는 다음과 같은 문제점들을 갖고 있다.

첫째, 무선 인터넷 환경은 휴대폰 액정의 크기, 컴퓨팅 능력, 무선망의 낮은 대역폭, 데이터 전송 지연과 불안정한 접속 등 다양한 문제점을 갖고 있다.

둘째, 현재 사용되는 미디어 형식들은 많은 리소스를 요구하고 있고, 특히 MPEG과 같은 고품질 동영상인 경우는 디코딩 과정이 복잡하여 휴대폰에 많은 부하를 준다. 그래서 낮은 CPU 성능과 작은 메모리, 작은 제한된 리소스를 가진 휴대폰에는 통상적인 동영상을 처리하지 못하고 있다.

셋째, C나 Java를 기반으로 한 Virtual Machine 방식을 사용하는 휴대폰들은 다양한 멀티미디어 서비스들을 제공하고 있지만 동영상을 처리하기 위해서는 휴대폰에 맞는 동영상 포맷을 개발하고 소프트웨어를 탑재해야 한다. 또한 휴대폰의 제조사 및 기종에 따라 별도로 소프트웨어를 개발해야 하는 부담이 따르게 된다.

넷째, 현재 무선 인터넷 기술 표준의 미정으로 인해 각각의 이동 통신사마다 서로 다른 방식으로 서비스를 시행하고 있으므로, 무선 인터넷 서

비스를 사용하고자 할 때에는 이동 통신사에 따라 소프트웨어를 별도로 개발해야 한다.

마지막으로 현재 휴대폰에 동영상을 직접 전송할 수 있는 상용화된 기술은 전무한 상황이다. 즉 사용자가 먼저 휴대폰으로 이동 통신사의 무선 인터넷망에 접속한 후 동영상을 다운로드를 받아야 하는 문제가 있다.

2.2 영상을 1bit 이미지로 변환시 임계값 설정에 대한 문제점

무선 인터넷에서 기존의 이미지를 효과적으로 나타내기 위해서는 무선선로와 휴대폰상의 제약 조건이 문제가 된다. 따라서 이를 해결할 수 있는 방법으로 WAP 포럼에서는 무선 인터넷에서의 데이터 양을 최소화하고, 휴대폰에서의 처리과정을 줄이기 위해 이미지를 표현할 때 1bit 이미지인 WBMP 이미지를 사용하도록 정의하고 있다[1,2].

WBMP 이미지는 다음과 같은 제약을 가지고 있다.

첫째, WBMP 이미지의 최대 사이즈는 가로 127 픽셀, 세로 127 픽셀이다.

둘째, WBMP 이미지의 크기가 WML Deck의 사이즈인 1492byte를 넘으면 안 된다.

셋째, WBMP 이미지는 1bit으로 표현하기 때문에 명암대비 문제가 생기게 된다. 즉 카메라에서 캡처한 원본 이미지에서 비슷한 휘도와 색깔로 이루어진 영역을 볼 때 인간은 같은 휘도와 색깔이라고 인식을 한다. 하지만 같은 휘도로 보일지라도 사실은 조금씩 차이가 생겨 명암분포가 하나에 집중되지 않고 넓게 분포가 된다. 명암이 넓게 분포되어 있는 경우에 인간은 사물이 더 선명하다고 느낀다.[13]

2.3 연구 방향

본 논문의 목표는 휴대폰에 별도의 소프트웨어를 추가하지 않고 영상전송 및 정보상황 통보방

법을 제안하고 구현하고자 한다. 본 논문에서는 카메라에서 실시간으로 캡처한 영상을 WBMP 이미지로 변환한 후 WAP 풀 방식과 WAP 푸시 서비스의 기반에서 영상전송 및 경보상황을 통보하는 것으로 제한한다.

3. 실시간 영상을 휴대폰으로 전송하는 방법

3.1 실시간 영상을 휴대폰으로 전송하는 방법에 대한 제약사항

본 논문에서 제안하는 실시간 영상을 휴대폰으로 전송하는 방법에는 다음과 같은 제약사항을 가지고 있다.

첫 번째, 휴대폰으로 영상전송을 받기 위해서는 무선 인터넷을 사용할 수 있어야 하며, WAP 브라우저가 탑재되어 있는 휴대폰이 있어야 한다. 따라서 기존의 모든 휴대폰을 지원할 수는 없다.

두 번째, 휴대폰 내에 별도의 소프트웨어 없이 WAP에서 지원해주는 이미지만을 사용하여 영상을 전송하고자 한다.

세 번째, 현재 실시간 영상 및 동영상을 사용자의 요청 없이 휴대폰으로 직접 전송할 수 있는 방법은 없다. 하지만 감시제어 시스템에서는 경보상황 발생시 휴대폰으로 감시하고 영상을 보내주어야 한다. 이러한 상황을 고려하여 휴대폰에 먼저 SMS와 같은 서비스를 통해 사용자에게 통보하고 감시제어 시스템의 내부 모듈에서는 경보상황이 종료될 때까지 캡처된 영상을 WBMP 이미지로 변환 후 저장하여 휴대폰으로 전송해 줄 수 있는 WML 콘텐츠로 구성한다. 사용자는 통보 받은 웹서버의 URL에 요청하면 준비된 웹서버의 콘텐츠로부터 전송을 받을 수 있다.

네 번째, 본 논문에서는 WAP 풀 기반과 WAP 푸시 서비스 기반에 대해서 기술하고 있지만 현재 국내의 이동통신업체들은 상용화된 WAP 푸시 서비스를 지원하고 있지 않아, 실제 휴대폰에서

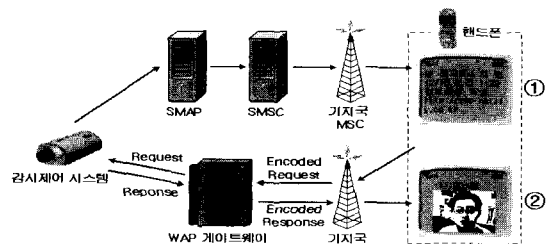
영상을 전송 받을 수 있는 방법은 WAP 풀 기반의 방법뿐이다. 하지만 WAP 푸시 서비스를 지원하기 위해 각 이동통신업체들은 개발 중에 있거나 개발을 완료한 후 내부적으로 사용하고 있는 상태이다. 이에 본 논문에서는 WAP 푸시 서비스를 통한 통지방법 및 영상전송을 테스트할 수 있도록 Openwave System Inc에서 제공해주는 Push Proxy/Gateway를 사용한다.

3.2 WAP 풀 기반 영상전송 방법

본 논문에서의 WAP 풀 기반 영상전송 방법은 그림 1과 같이 SMS(Short Message Service)를 통해 사용자에게 통보하고, 감시제어 시스템의 내부 모듈에서는 경보상황이 종료될 때까지 캡처된 영상을 WBMP 이미지로 변환 후 저장하여 휴대폰으로 전송해 줄 수 있는 WML 콘텐츠로 구성한다. 사용자가 통보 받은 웹서버의 URL에 요청을 하면 준비된 웹서버의 콘텐츠로부터 전송 받게 되는 방식이다. 이 방식은 크게 여섯 가지의 흐름으로 나눌 수 있다.

첫 번째, 감시제어 시스템에서 경보상황 발생시 경보발생과 영상을 볼 수 있는 웹서버 URL 정보를 가진 문자메시지를 SMAP(Short Message Application Platform), SMSC(Short Message Service Center), 기지국 & MSC를 거쳐 사용자의 휴대폰으로 통보한다. 문자메시지는 그림 1의 ①과 같이 경보상황이 발생했다는 통보메시지 및 발생시각, 웹서버 URL를 담고 있다.

두 번째, 사용자가 휴대폰에 들어온 문자메시지



(그림 1) WAP 풀 기반 영상전송 방법의 흐름

를 확인 후 휴대폰 상에서 WAP 브라우저를 사용하여 이동 통신사의 무선 인터넷망에 접속하여 웹서버 URL를 입력하면 WSP 요구 메시지가 WAP 게이트웨이로 전해진다.

세 번째, WAP 게이트웨이는 암호화된 요구 메시지를 HTTP 요구 메시지로 바꾼 후 감시제어 시스템의 웹서버로 전송한다.

네 번째, 감시제어 시스템의 웹서버는 실시간 영상에서 WBMP 이미지로 변환된 정보를 가진 WML 콘텐츠를 동적으로 생성한 후 WAP 게이트웨이에 되돌려 준다.

다섯 번째, 감시제어 시스템의 웹서버로 수신된 응답 메시지를 WAP 게이트웨이는 암호화된 응답 메시지로 변환하여 휴대폰에 전달한다.

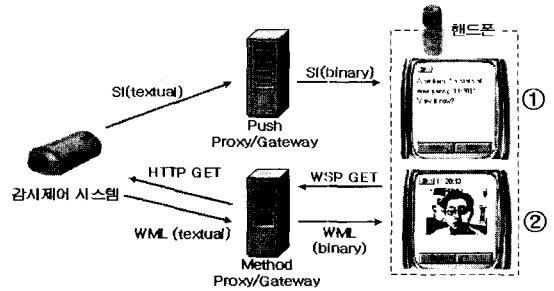
마지막으로 휴대폰에서는 그림 1의 ②와 같이 WBMP 이미지로 변환된 감시현장의 영상이 보여지게 된다. 이후부터는 사용자가 종료요청을 할 때까지 감시제어 시스템의 웹서버로부터 실시간 영상을 전송 받는다.

3.3 WAP 푸시 서비스의 SI 기반 영상전송 방법

WAP 푸시 서비스의 SI(Service Indication) 콘텐츠 타입은 비동기적인 방법으로 사용자의 단말기에 이벤트 통보를 보내기 위한 메커니즘을 제공하며, 짧은 텍스트 확인 메시지가 URI(Uniform Resource Identifier)를 통해 전송된다. 사용자는 푸시 콘텐츠의 텍스트 메시지를 받고, 서비스를 로드 할지의 여부를 선택할 수 있다. 이 콘텐츠 타입은 사용자에게 통보하지만 무시할 만한 메시지가 있을 법한 메커니즘에 주로 사용된다[7,11].

본 논문에서 WAP 푸시 서비스의 SI 기반 영상 전송 방법은 그림 2와 같은 크게 다섯 가지의 흐름으로 나눌 수 있으며, 3.2절에서 제안한 WAP 폴 기반 영상전송 방법과 비슷하다.

첫 번째, 감시제어 시스템에서 경보상황 발생시 경보발생을 사용자의 휴대폰에 통보하기 위해 Push Access Protocol[5]를 사용하여 Push Proxy/Gateway



(그림 2) WAP 푸시 서비스의 SI 기반 영상전송 방법의 흐름

에 통보한다. 이때 SI의 URI는 경보발생 메시지와 감시제어 시스템의 웹서버 URL의 정보를 가진다.

두 번째, Push Proxy/Gateway는 감시제어 시스템에서 통보 받은 SI를 Push OTA Protocol[6]를 사용하여 사용자의 휴대폰에 보낸다.

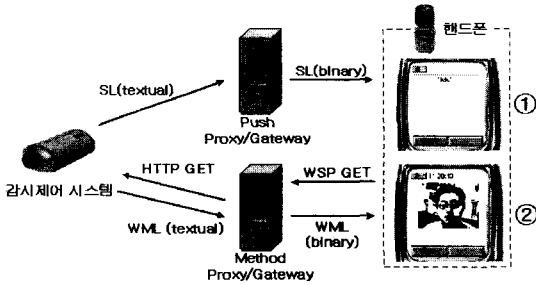
세 번째, 사용자가 휴대폰에 들어온 그림 2의 ①과 같은 경보발생 내역을 확인 후 “View”를 선택하면 SI의 URI를 통해 즉시 Method Proxy/Gateway를 거쳐 감시제어 시스템의 웹서버로부터 실시간 영상을 WBMP 이미지로 변환한 WML 콘텐츠를 가져온다.

네 번째, 감시제어 시스템의 웹서버에서 가져온 텍스트의 WML 콘텐츠를 이진화로 바꾼 후 휴대폰으로 전달한다.

마지막으로 휴대폰에서는 그림 2의 ②와 같이 WBMP 이미지로 변환된 감시현장의 영상이 보여지게 된다. 이후부터는 사용자가 종료요청을 할 때까지 감시제어 시스템의 웹서버로부터 실시간 영상을 전송 받는다.

3.4 WAP 푸시 서비스의 SL 기반 영상전송 방법

WAP 푸시 서비스의 SL(Service Loading) 콘텐츠 타입은 휴대폰의 사용자 에이전트가 서버 로딩을 통해 서비스를 로드하고 실행시킬 수 있는 메커니즘을 제공한다. 이 메커니즘은 사용자의 개입 없이 사용자 에이전트에 의해서 처리되는 방식이다[8,11].



(그림 3) WAP 푸시 서비스의 SL 기반 영상전송 방법의 흐름

본 논문에서 WAP 푸시 서비스의 SL 기반 영상전송 방법은 그림 3과 같이 크게 다섯 가지의 흐름으로 나눌 수 있다.

첫 번째, 감시제어 시스템에서 정보상황 발생시 사용자의 휴대폰에 영상을 전송하기 위해서는 Push Access Protocol[5]를 사용하여 Push Proxy/Gateway에 통보한다. 이때 SL의 URI는 사용자 에이전트에서 실행될 수 있도록 감시제어 시스템의 웹서버 URL의 정보를 WML Deck 형태로 가진다.

두 번째, Push Proxy/Gateway는 감시제어 시스템에서 통보 받은 SL를 Push OTA Protocol[6]를 사용하여 사용자의 휴대폰에 보낸다.

세 번째, 휴대폰은 SL를 전송 받으면 사용자의 개입 없이 즉시 Method Proxy/Gateway를 거쳐 감시제어 시스템의 웹서버로부터 실시간 영상을 WBMP 이미지로 변환한 WML 콘텐츠를 가져온다.

네 번째, 감시제어 시스템의 웹서버에서 가져온 텍스트의 WML 콘텐츠를 이진화로 바꾼 후 휴대폰으로 전달한다.

마지막으로 휴대폰에서는 그림 3의 ②와 같이 WBMP 이미지로 변환된 감시현장의 영상이 보여지게 된다. 이후부터는 사용자가 종료요청을 할 때까지 감시제어 시스템의 웹서버로부터 실시간 영상을 전송 받는다.

4. 실시간 영상을 WBMP 이미지로 변환

WBMP 이미지는 무선환경에서 그래픽 정보표

현을 가능하게 하기 위해 무선환경 특성에 맞게 정의된 새로운 이미지 형식이다. 협소한 대역폭, 무선 인터넷에서의 높은 에러 발생률, 메모리 부족 등의 제약이 있는 무선환경에서 기존의 이미지를 이용한 서비스를 제공하는 데는 많은 어려움이 있으므로 전송되는 데이터의 양, 무선 단말기에 저장할 데이터의 양 등을 최소화할 필요가 있다.

WBMP 이미지는 다른 이미지 파일과 비교할 때 헤더의 구조가 간단하고, 이미지를 나타내는데 사용되는 데이터의 양이 적으므로 이동단말의 제한된 자원을 사용하여 그래픽 정보를 표현하기에 적합하다[2].

본 장에서는 카메라에서 캡처한 영상을 WBMP 이미지로 변환할 때 휴대폰의 스크린 사이즈, 픽셀 당 비트수 등을 고려하여 이미지를 축소하는 방법과 축소된 이미지에서 픽셀의 값을 추출하여 이미지 변환의 기준이 되는 임계값 결정 방법에 대하여 설명한다.

4.1 카메라에서 캡처된 영상에서 명암정보만을 사용

카메라에서 캡처할 수 있는 영상포맷은 다양하다. 예를 들어 JPEG, GIF, BMP 등과 같은 RGB 포맷일 경우 이미지의 비트 깊이, 팔레트 사용 여부, 가비지 값 유무 등을 고려하여 픽셀의 RGB 값을 추출한다. 이렇게 추출한 값을 이용하여 식 (1)으로 명암 단계 값을 계산해야 하며, 캡처된 영상이 클수록 계산해야 할 연산이 많아지게 된다. 또한 캡처된 영상을 변환처리하기 위한 버퍼 사이즈도 커지게 된다.

$$Y=(0.257 \times R)+(0.504 \times G)+(0.098 \times B)+16 \quad \text{식(1)}$$

본 논문에서는 카메라에서 CIF 사이즈인 YUV 4:1:1이나 YUV 4:2:0 포맷으로 영상을 캡처하여 명암정보인 Luminance 값을 사용한다. 그림 4는 RGB 24bit인 영상이고, 그림 5는 명암 값으로 표현한



(그림 4) RGB 24bit 영상



(그림 5) 명암값으로 표현한 영상

영상이다.

이 두 영상을 담을 수 있는 버퍼사이즈를 비교를 하면 식(2)와 식(3)과 같다.

(그림 4)의 버퍼사이즈= $352 \times 288 \times 3 = 304,128 \text{ byte}$ 식(2)

(그림 5)의 버퍼사이즈= $352 \times 288 \times 3/2 = 152,064 \text{ byte}$ 식(3)

위의 식에서 보는 바와 같이 명암 값으로 표현한 영상이 RGB 영상보다 2배가 더 적으며, 실제로 휘도만 사용하므로 RGB 영상보다 3배가 더 적어지게 된다. 또한 RGB 영상에 비해 명암 단계 값을 계산할 필요가 없다.

4.2 Fant의 재추출 알고리즘을 이용한 영상 축소

카메라에서 캡처된 영상은 CIF 사이즈로 휴대폰의 스크린 사이즈 보다 크므로 스크린 사이즈 보다 작은 이미지로 변환해야 하며, 만약 이미지의 크기가 WML Deck의 사이즈인 1492byte를 넘으면 휴대폰에 전송되지 않는 문제가 생긴다. 또한 휴대폰의 스크린 사이즈에 맞게 이미지를 축소할 때는 실수 배로 해야 한다.

본 논문에서는 Fant의 재추출 알고리즘을 사용하여 이미지를 축소한다. Fant의 재추출 알고리즘은 영상을 스케일링(scaling)하는 두 단계 알고리즘으로 출력 화소가 생성될 때 모든 입력 화소 값을 평가함으로써 aliasing 현상을 감소시키며, 한 행

또는 한 열을 입력하고, 새 행이나 새 열을 출력하는 방법이다[13]. 예로 영상을 100×82 로 축소할 경우 WBMP 이미지 사이즈는 1070byte이며, 가로방향, 세로방향에 대해 각각 Fant의 재추출 알고리즘을 사용하여 축소한다. 아래 그림 6은 가로방향으로 축소하는 예이다.

Scale = 0.284 (100 / 352)

92	90	80	99	89	76	84	67	79	88	69
.....										
89	85	77								

(그림 6) Fant의 재추출 알고리즘을 이용한 영상 축소

첫 번째 출력 화소는 다음과 같이 계산된다.

$$[92 \times 1 + 90 \times 1 + 80 \times 1 + 99 \times 0.52] \times 1/3.52 = 89$$

두 번째 출력 화소와 세 번째 출력 화소는 다음과 같다.

$$[99 \times 0.48 + 89 \times 1 + 76 \times 1 + 84 \times 1 + 67 \times 0.04] \times 1/3.52 = 85$$

$$[67 \times 0.96 + 79 \times 1 + 88 \times 1 + 69 \times 0.56] \times 1/3.52 = 77$$

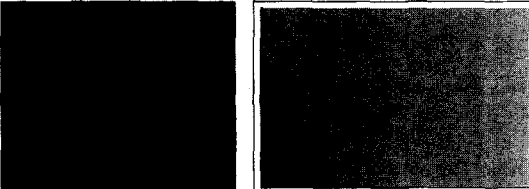
4.3 축소된 영상을 WBMP 이미지로 변환

WBMP 이미지는 검정색과 흰색만으로 표현되는 이미지로 축소된 영상의 픽셀 값을 임계값과 비교하여 임계값이 작으면 흰색으로 표현하고 크면 검은색으로 표현한다. 따라서 WBMP 이미지로 변환시 임계값의 설정은 중요하다.

본 논문에서 사용한 임계값은 식(4)와 같이 영상 히스토그램을 사용하여 전체 픽셀 값들에 대한 평균값으로 사용한다. 그림 7, 8, 9, 10 들은 밝은 영상과 어두운 영상에 대한 히스토그램 및 임계값에 따른 WBMP 이미지로 변환했을 때의 영상을 보여주고 있다.



(그림 7) 밝은 영상과 밝은 영상의 히스토그램



(그림 8) 어두운 영상과 어두운 영상의 히스토그램

$$Threshold = \frac{\sum_{k=1}^{width \times height} Pixel[k]}{width \times height} \quad \text{식(4)}$$

영상 히스토그램은 영상의 모든 픽셀들에 대한 밝기 값을 출현빈도로 나타낸 것으로 일반적으로 어두운 영상은 픽셀 값 분포가 왼쪽으로 편중된 히스토그램을 가지며, 밝은 영상은 픽셀 값 분포가 오른쪽으로 편중된 히스토그램을 갖는다. 또한 명암대비가 낮으면 픽셀이 히스토그램의 왼쪽, 오른쪽 또는 중앙의 오른쪽으로 집중된다. 그리고 히스토그램의 막대가 촘촘하게 밀집되어 픽셀 값의 범위가 일부분에만 분포한다. 반대로 명암대비가 높으면 넓은 범위의 픽셀 값을 포함한다[13].

다음 그림 9와 그림 10은 밝은 영상과 어두운 영상에서 임계값을 중간 값인 128로 했을 경우와 영상의 픽셀 값들의 전체에 대한 평균값으로 했을 때를 비교한 그림이다.



(그림 9) 밝은 영상에서 임계값이 128(좌)인 경우와 183(우)인 경우



(그림 10) 어두운 영상에서 임계값이 128(좌)인 경우와 27(우)인 경우

위의 그림에서 보는바와 같이 밝은 영상에서는 임계값인 128인 경우와 183인 경우와 많은 차이를 내지는 않지만 어두운 영상에서는 임계값을 27로 했을 경우에는 영상을 알아 볼 수 있지만, 128로 했을 경우에는 전혀 알아볼 수가 없다.

5. 구현 및 실험결과

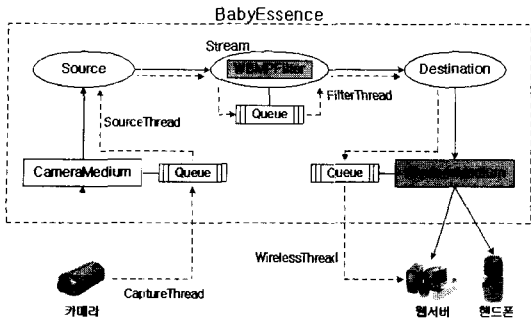
5.1 구현

5.1.1 구현 환경

본 논문의 구현은 Windows 2000 Profession 상에서 윈도우기반의 적응적 스트림 처리기(BabyEssence for Windows)와 Visual C++ 6.0를 사용하여 영상변환 필터(WBMPFilter)와 영상전송 미디움(WirelessMedium)을 구현하였으며, Openwave System에서 제공하는 WAP Push Library와 JDK 1.3.1을 사용하여 WAP 푸시 서비스의 SI(Service Indication)와 SL(Service Loading)를 이용한 영상전송 방법을 구현하였다. 웹서버는 IIS 5.0(Internet Information Server 5.0)을 사용하였다.

5.1.2 스트림 엔진에서 영상변환 필터와 영상전송 미디움의 동작 모델

스트림 엔진에서 영상변환 필터와 영상전송 미디움의 동작 모델은 그림 11과 같이 Source 객체인 CameraMedium과 Destination 객체인 WirelessMedium, Source 객체에서 보내진 데이터를 WBMP 포맷으로 변형하는 WBMPFilter로 구성된다.



(그림 11) 스트림 엔진에서 영상변환 필터와 영상전송 미디어의 동작 모델

CameraMedium 객체는 CaptureThread에서 카메라로부터 영상 데이터를 YUV 4:1:1나 YUV 4:2:0으로 캡처하여 CameraMedium이 가지고 있는 Queue에 저장하고 이벤트를 발생한다.

Source객체의 SourceThread는 CameraMedium에서 이벤트가 발생하면 CameraMedium의 ReadFrame() 함수를 호출하여 Queue에서 데이터를 가져와 Stream 객체로 전송한다.

Stream 객체는 필터 리스트(필터 파이프라인)를 구성하고 있는 첫 번째 필터인 WBMPFilter가 가지고 있는 Queue에 Source 객체로부터 전달받은 데이터를 저장한다.

WBMPFilter 객체는 가지고 있는 Queue에 데이터가 들어오면 이벤트가 발생하고, 발생한 이벤트를 FilterThread에서 받아 WBMPFilter 객체가 가지고 있는 Queue에서 데이터를 가져와 WBMP 포맷으로 데이터를 변환한 후 Destination 객체로 전달한다.

Destination 객체는 전송 받은 데이터를 WirelessMedium 객체가 가지고 있는 Queue에 데이터를 저장한다.

WirelessMedium 객체는 가지고 있는 Queue에 데이터가 들어오면 이벤트가 발생하고, 발생한 이벤트를 WirelessThread에서 받아 Wireless Medium 객체가 가지고 있는 Queue에서 데이터를 가져와 WBMP 파일 및 시각정보를 WML Deck 형태의 서버사이드 스크립트(asp)를 생성하여 WebServer에 저장한다.

5.1.3 영상변환 필터 (WBMPFilter) 구현

영상변환 필터는 스트림 엔진의 스트림 상에 위치하면서 입력되는 영상 데이터를 WBMP 포맷의 데이터로 변환한다. 각 동작에 따른 구현방법을 기술하면 다음과 같다.

1. 영상변환 필터를 스트림에 추가할 때 사용자로부터 QoS 정보(입력되는 이미지의 크기, 변환할 이미지 크기, 임계값)를 받는다.
2. 영상변환 필터가 가지고 있는 Queue에 카메라에서 캡처한 영상 데이터가 들어오면 이벤트를 발생하고, 발생한 이벤트를 Filter Thread에서 받는다.
3. FilterThread에서는 영상변환 필터가 가지고 있는 Queue에서 데이터를 가져오고, Luminance 성분만을 가지고 Fant의 재추출 알고리즘을 사용하여 사용자가 지정한 이미지의 크기로 축소한다.
4. 4.3절에서 제안한 방법으로 축소된 영상에서 임계값을 추출한 후 사용자가 지정한 임계값에 가감을 하여 최종적인 임계값을 결정한다. 결정된 임계값으로 축소된 영상의 픽셀 값과 비교하여 0 또는 1로 변환한다.
5. WBMP 포맷으로 변환된 데이터를 Destination 객체인 WirelessMedium으로 전송한다.

5.1.4 영상전송 미디어(WirelessMedium) 구현

영상전송 미디어는 SMS 및 WAP 푸시 서비스의 SI 및 SL 콘텐츠를 이용하여 사용자의 휴대폰으로 통지한다. 또한 영상변환 필터에서 받은 데이터로 WBMP 이미지를 생성하고 WML Deck 형태의 서버사이드 스크립트를 생성하여 웹서버에 저장한다. 각 동작에 따른 구현방법을 세 가지로 나누어 기술하면 다음과 같다.

첫 번째는, 사용자로부터 다음과 같은 QoS 정보를 입력받는다.

- WAP 풀 기반, 푸시 서비스의 SI 및 SL 기반

전송방법 중 선택

- 통지할 휴대폰 번호 리스트
- 휴대폰에 통지할 메시지
- 웹서버의 URL
- WBMP 이미지와 서버사이드 스크립트를 저장할 위치

두 번째는, 정보상황 발생시 영상전송 미디움의 메서드인 SendNotifyPhone()을 호출하면 사용자가 지정한 전송방법에 따라 3장에서 제안한 방법으로 휴대폰에 통보한다. WAP 푸시 서비스를 이용할 경우에는 WAP Push Library로 구현된 Java 클래스를 호출하여 통지한다.

세 번째는, 영상전송 미디움이 가지고 있는 Queue에 데이터가 들어온 후부터의 구현방법은 다음과 같다.

1. 영상전송 미디움이 가지고 있는 Queue에 영상 변환 필터에서 WBMP 포맷으로 변환한 데이터가 들어오면 이벤트가 발생하고, 발생한 이벤트를 WirelessThread에서 받는다.
2. WirelessThread는 영상전송 미디움이 가지고 있는 Queue에서 데이터를 가져와 WBMP 타입, 고정헤더, 너비, 높이 정보를 가진 WBMP 헤더를 추가하여 사용자가 지정한 디렉토리에 WBMP 이미지파일을 저장한다.
3. 저장된 WBMP 파일 및 시각정보를 표 1, 2, 3과 같이 WML Deck 형태의 서버사이드 스크립트(asp)를 생성하여 사용자가 지정한 디렉토리에 저장한다.

서버사이드 스크립트를 동적으로 생성할 때 표 1의 코드를 추가해야 한다. 그렇지 않으면 IIS는 ASP 파일의 콘텐츠를 HTML를 위한 MINE 타입을 쓰는 브라우저로 전송할 것이며, 이는 사용자의 휴대폰에서는 거부가 된다. 그래서 서버에게 단지 WML을 위한 MINE 타입만을 사용하라고 알려줘야 한다[12].

(표 1) ASP에서 WML을 위한 MINE 타입 셋팅

```
<% response.ContentType = "text/vnd.wap.wml" %>
```

(표 2) WAP 브라우저의 캐시기능을 중지시키는 Cache-Control 명령

```
<%
Response.Expires = -1
Response.AddHeader "Pragma", "no-cache"
Response.AddHeader "Cache-Control",
                        "no-cache,must-revalidate"
%>
```

(표 3) 휴대폰으로 영상을 전송하기 위한 WML 문서

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN" "http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>
<card id="main" title="시각정보" newcontext="true">
  <onevent type="ontimer">
    <go href="웹서버 URL"/>
  </onevent>
  <timer value="10"/>
  <p align="center">
    
  <br/></p>
</card></wml>
```

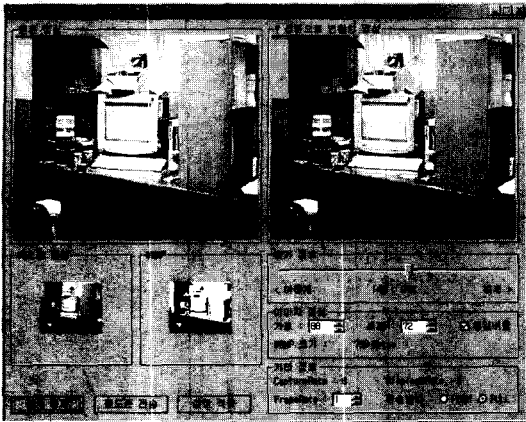
표 2는 WAP 브라우저의 캐시 기능을 중지시키는 Cache-Control 명령을 사용하여 접속될 때마다 갱신하도록 하는 코드이다.

5.2 실험 및 실험결과

5.2.1 실험 환경

실험 환경은 Windows 2000 Profession 상에서 윈도우 기반의 적응적 스트림 처리기에 본 논문에서 제안한 방법으로 구현한 영상변환 필터 및 영상전송 미디움을 추가한 후 스트림 엔진의 API를 사용하여 그림 12에서 보는바와 같이 휴대폰으로의 실시간 전송 프로그램을 작성하였다.

WAP 풀 기반의 영상전송 방법은 011, 017, 019



(그림 12) 실시간 영상을 휴대폰으로의 전송 프로그램

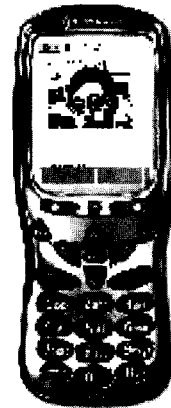


(그림 13) WAP 브라우저가 탑재된 휴대폰

등의 이동 통신업체의 WAP 브라우저가 탑재된 휴대폰으로 테스트하였으며, WAP 푸시 서비스 기반의 영상전송 방법은 국내에는 사용할 수 있는 Push Proxy/Gateway가 없어 Openwave System Inc에서 제공해주는 Push Proxy/Gateway를 사용하고, Openwave SDK WAP Edition 5.0의 Simulator를 WAP 브라우저로 테스트하였다.

5.2.2 실험 결과

WAP 풀 기반의 영상전송 방법으로 테스트한 결과는 그림 13과 같으며, 전송되는 프레임 수는 CDMA2000용 단말기는 1초에 1프레임이 전송되었으며, 다른 단말기들은 4~6초마다 1프레임이 전송되었다.



(그림 14) Openwave SDK, WAP Edition 5.0 Simulator

WAP 푸시 서비스 기반의 영상전송 방법으로 테스트한 결과는 그림 14와 같으며, 전송되는 프레임 수는 3초마다 1프레임이 전송되었다.

6. 결 론

실시간으로 캡처한 영상을 이동 전화에 실시간으로 영상을 전송하기 위한 방법으로 실시간 캡처한 영상을 WBMP 영상으로 변환하고, WAP PUSH-SI와 PUSH-SL기반으로 영상 전송하는 것을 제안했다.

WBMP 영상 변환 시, 변환에 기준이 되는 임계 값이 전체 픽셀의 히스토그램 평균 값을 사용하는 WBMPFilter를 구현하였다. 또 WAP을 이용하여 실시간 영상 전송하기 위한 WirelessMedium을 구현하였다.

참 고 문 헌

- [1] Wireless Application Protocol Architecture, WAP Forum, Apr. 30. 1998.
URL:<http://www.wapforum.org>.
- [2] Wireless Application Environment Specification, WAP Forum, Mar. 29. 1998.
URL:<http://www.wapforum.org>.

- [3] Wireless Application Protocol Push Architectural Overview, WAP Forum, Nov. 8. 1999.
URL:<http://www.wapforum.org>.
- [4] Wireless Application Protocol Push Proxy Gateway Service Specification, WAP Forum, Aug. 16. 1999.
URL:<http://www.wapforum.org>.
- [5] Wireless Application Protocol Push Access Protocol Specification, WAP Forum, Nov. 8. 1999. URL:<http://www.wapforum.org>.
- [6] Wireless Application Protocol Push OTA Protocol Specification, WAP Forum, Feb. 17. 2000.
URL:<http://www.wapforum.org>.
- [7] Wireless Application Protocol Service Indication Specification, WAP Forum, Jul. 31. 2001.
URL:<http://www.wapforum.org>.
- [8] Wireless Application Protocol Service Loading Specification, WAP Forum, Jul. 31. 2001.
URL:<http://www.wapforum.org>.
- [9] Wireless Markup Language Specification, WAP Forum, Feb. 19. 2000.
URL:<http://www.wapforum.org>.
- [10] RFC2068, Hypertext Transfer Protocol-HTTP/1.1, Jan. 1997.
URL:<ftp://ftp.isi.edu/in-notes/rfc2068.txt>.
- [11] WAP Push Library Developer's Guide, Openwave Systems Inc, September 2001.
URL:<http://www.openwave.com>.
- [12] Charles Arehart의 12인 공저, Professional WAP, WROX Press, 2000.
- [13] Randy Crane, A Simplified Approach to Image Processing, Prentice-Hall, 1997.
- [14] 임영환, "ComBiStation : 분산 멀티미디어 컴퓨팅 환경을 위한 컴퓨터 플랫폼," 정보과학 회논문지, 제2권, 제1호, pp. 160~181, 1996.

◎ 저자 소개 ◎



박 대 원

1995년 상지대학교 통계학과 졸업(학사)
2002년 숭실대학교 대학원 컴퓨터학과 졸업(석사)
관심분야 : 스트림엔진, 멀티미디어, 모바일
E-mail : castleofsky@dreamwiz.com



홍 마리아

1997년 서남대학교 영문학과 졸업(학사)
2000년 숭실대학교 대학원 정보미디어 학과 졸업(석사)
2002년~현재 : 숭실대학교 대학원 컴퓨터학과 박사 과정
관심분야 : 스트림 엔진, 영상압축, 모바일, WAP
E-mail : maria@media.ssu.ac.kr



김 규 정

New York Univ. MA
New York Univ. D.A.
숭실대학교 정보과학대학 미디어학부 교수
관심분야 : 영상디자인, 미디어 아트
email : kjkim@comp.ssu.ac.kr



임 영 환

1977년 경북대학교 수학과(이학사)
1979년 한국과학 기술원 전산학과(이학석사)
1979년~1996년 한국 전자 통신 연구소 책임연구원
1985년 Northwestern University(이학박사)
1996년~현재 : 숭실대학교 미디어학부 교수
email : yhlhm@computing.ssu.ac.kr