

무선 인터넷 환경에서 단말에 최적화된 WBMP 이미지 변환기 구현

김태주°, 채경호, 임경식
경북대학교 컴퓨터학과
e-mail : {taejoo, chaekh, kslim}@ccmc.knu.ac.kr

An Implementation of the WBMP Image Converter suitable for a Terminal in Wireless Internet Environment

Taejoo Kim°, Kyungho Chae, Kyungshik Lim
Dept. of Computer Science, Kyungpook National University

요 약

본 논문에서는 이동단말에 최적화된 서비스를 제공하기 위한 이동단말의 정보 이용방법과 무선선로상의 제약 및 무선 단말기의 제한된 자원으로 좀더 효율적인 무선 인터넷 서비스를 제공하기 위한 WBMP(Wireless Bitmap) 변환기의 설계 및 구현에 대하여 기술한다. 이동단말의 정보는 CC/PP (Composite Capabilities/Preferences Profiles) 또는 WSP(Wireless Session Protocol) 헤더로부터 알아낼 수 있으며 이 정보를 이용하여 WBMP 변환기는 기존의 이미지를 이동단말에 가장 적절한 WBMP 이미지로 변환한다. 그러나 컬러 이미지를 변환하는 경우 이동단말이 가지는 제약에 의해 이미지 정보가 손실되거나 왜곡될 수 있으며, 현재 무선환경에서 지원되는 0/1 형식의 WBMP 이미지로는 각 단말의 자원을 최대한 활용할 수 없다. 따라서 향후 그레이-레벨 또는 컬러 이미지로의 변환 기능이 추가 구현되어야 한다.

1. 서론

최근 이동통신 가입자가 급속히 증가하고 인터넷이 일반화되면서 휴대성과 이동성으로 대표되는 이동통신을 인터넷과 결합한 새로운 형태의 무선 인터넷 서비스가 보편화되고 있다. 특히 텍스트 위주의 정적인 서비스에서 이미지가 추가된 서비스로 변함에 따라 가독성이 증가하였고, 이미지를 다양한 응용 서비스에 사용함으로써 서비스의 질적 향상을 가져왔다.

그러나 기존의 이미지를 효과적으로 나타내는 데는 무선선로와 단말기상의 제약조건이 문제가 되므로 이를 해결할 수 있는 적절한 방법이 필요하다. 따라서 WAP(Wireless Application Protocol) 포럼에서는 무선선로에서의 데이터의 양을 최소화하고, 무선 단말기에서의 처리과정을 가능한 한 줄이기 위해 이미지를 표현할 때 WBMP 이미지를 사용하도록 정의하고 있다[1, 2].

그러나 동일한 WBMP 이미지를 휴대용 전화,

PDA(Personal Digital Assistant), 핸드헬드 PC 등 여러 종류의 이동단말에 제공하는 것은 효율적이지 않다. 그러므로 각각의 단말에 대해 자원을 고려한 최적화된 WBMP 이미지로 변환할 필요가 있다.

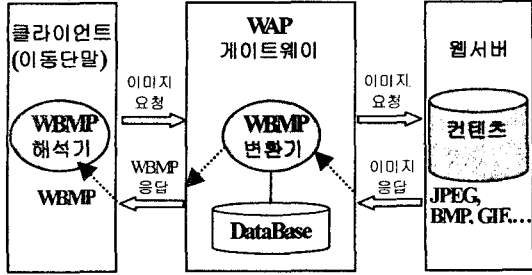
2. WAP 네트워크 구성도

WAP은 무선 인터넷 서비스를 지원하기 위한 국제적 개방 통신 규약으로 인터넷망과 이동통신망의 접속지점에 게이트웨이를 두어 두 프로토콜 체계 사이의 중계역할을 하도록 한다[3].

<그림 1>은 무선 인터넷 서비스를 지원하기 위한 WAP 네트워크 구성도이며, 이동단말에 최적화된 이미지 제공을 위한 WAP 네트워크 동작원리를 나타낸다.

클라이언트가 이미지를 요청할 경우 게이트웨이는 웹서버로부터 해당하는 이미지를 받아온다. 클라이언트는 WBMP 이미지와 기존의 이미지 모두 요청이 가

능하며, 웹서버로부터 받은 이미지의 형식이 WBMP



<그림 1> WAP 네트워크 구성도

가 아닌 경우에는 WBMP 변환기에 의해 WBMP 이미지로 변환된다. 이미지 요청시 클라이언트의 지원 가능한 자원에 대한 정보를 함께 보내면 클라이언트는 브라우저에 가장 적절한 이미지를 받을 수 있다. 전송된 WBMP 이미지는 클라이언트 내의 WBMP 해석기에 의해 브라우저 가능한 형태로 바뀌게 되고, 브라우저를 통해 표현된다.

3. 단말기 정보

이동단말에 적절한 이미지로 변환하기 위해서는 먼저 단말기의 하드웨어, 소프트웨어, 네트워크 등의 정보를 알아야 한다.

이는 CC/PP를 사용하거나 WSP 헤더를 분석함으로써 알아낼 수 있으며 다음은 두 방법에 대한 간단한 설명이다.

3.1 CC/PP 사용

CC/PP는 디바이스에 적절한 콘텐츠를 제공하기 위해 클라이언트와 서버 사이에 주고받는 정보를 정의하기 위한 프레임워크로서 CC/PP를 사용하여 단말의 특성 및 자원, 네트워크에 대한 정보 등을 명시할 수 있다. 그러므로 WAP 클라이언트와 게이트웨이, 웹서버간 콘텐츠 요청시 CC/PP를 사용한다면 이동 단말에 최적화된 무선 인터넷 서비스를 지원할 수 있다[4].

다음은 CC/PP를 사용하여 이동단말의 정보를 게이트웨이 또는 웹서버로 전달하는 과정을 나타낸다[4, 5].

1. 클라이언트는 하드웨어 플랫폼과 소프트웨어 플랫폼, 네트워크 특성 등의 단말기 프로파일을 RDF(Resource Description Framework)를 사용하여 정의한다.
2. 콘텐츠 요청시 프로파일에 대한 URI (Uniform Resource Identifier)가 WSP 헤더의 Profile 필드에 저장되어 게이트웨이로 전송된다.
3. 게이트웨이는 Profile 필드에 있는 URI로부터 단말의 프로파일을 가져와 저장하고 필요할 경우 저장된 정보를 이용하여 단말에 적절한 콘텐츠로 변환한다.
4. 웹서버 또한 게이트웨이로부터 받은 HTTP (Hypertext Transfer Protocol)헤더의 Profile 필드를 분석하여 단말의 프로파일을 가져와 정의되어

는 특성에 맞게 요청한 콘텐츠를 변환하여 전송한다.

5. 변경된 프로파일은 WSP Resume request를 보낼 때 다시 수정할 수 있다.

3.2 WSP 헤더 정보 이용

클라이언트는 이미지 요청 메시지를 보낼 때 'User-Agent', 'Accept' 등의 단말기에 대한 정보를 WSP 헤더에 세팅한다. 정의되어 있는 헤더 필드 외에도 임의로 정의 가능하므로 스크린 크기, 픽셀 당 비트수 등 필요한 정보를 넣을 수 있다. 그리고 WSP 헤더 인코더에 의해 인코딩 된 후 게이트웨이로 전송된다[6, 7].

게이트웨이는 클라이언트로부터 받은 메시지를 디코딩하여 WSP 헤더 정보 중 필요한 필드를 데이터베이스에 저장해 두고 이미지 변환이 필요할 때 저장된 값을 적용하여 WBMP 이미지로 변환한다.

그리고 이 정보 또한 HTTP 헤더를 통해 웹서버로 전달되므로 필요한 경우 웹서버에서 이 정보를 이용할 수 있다.

4. WBMP 이미지 변환기 구현

4.1 WBMP

WBMP는 무선환경에서 그래픽 정보 표현을 가능하게 하기 위해 무선환경의 특성에 맞게 정의된 새로운 이미지 형식이다. 협소한 대역폭, 무선선로에서의 높은 에러발생율, 메모리 부족 등의 제약이 있는 무선환경에서 기존의 이미지를 이용한 서비스를 제공하는데는 많은 어려움이 있으므로 전송되는 데이터의 양, 무선 단말기에 저장한 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].

4.2 구현 환경

ALZZA Linux release 6.0 상에서 JDK 1.2.2를 사용하여 구현하였는데 JDK 1.2.2는 OS, 플랫폼에 독립적이고, 이미지 처리에 필요한 여러 가지 함수들을 쉽게 사용할 수 있는 장점이 있다. 그러나 이미지의 형식에 따라 지원 가능한 함수에 제한이 있으므로 구현시 이를 고려해 주어야 한다.

4.3 WBMP 변환기 동작과정

WAP 게이트웨이가 웹서버로부터 이미지를 받으면 먼저 이미지의 타입을 검사하여 WBMP가 아닌 경우 WBMP 변환기를 이용하여 WBMP 이미지로 변환시킨다. 이때 창의 크기, 스크롤 기능 유무, 픽셀 당 비트수 등 단말기의 정보를 고려하여 각 단말기에 최적화된 이미지로 변환시킬 수 있다.

이미지를 입력받은 변환기는 헤더를 분석하여 너비와 높이를 알아내는데, 이는 WBMP 이미지의 너비와 높이를 결정할 뿐만 아니라 이미지 처리를 위한 여러 가지 경우에 이용된다. 다음으로 픽셀의 R(Red),

G(Green), B(Blue) 값을 추출하여 이미지 변환의 기준이 되는 임계값을 결정하는데, 이때 단말기의 자원에 대한 정보가 있으면 이를 고려하여 계산해야 한다. 임계값이 결정되면 이를 기준으로 각 픽셀의 값을 0 또는 1로 변환하여 WBMP 이미지의 데이터를 생성하고, 헤더와 데이터가 모두 만들어지면 클라이언트가 지정하거나 입력된 이미지 파일을 참고하여 임의로 정한 이름의 WBMP 이미지 파일을 생성함으로써 변환기의 동작이 끝나게 된다.

4.4 WBMP 변환기 구현

변환기의 각 동작에 따른 구현방법을 기술하면 다음과 같다.

● 이미지 입력

이미지의 타입을 검사하여 이미지를 입력받고, 변환할 방법을 결정해야 한다. JPEG, GIF 등 인터넷 표준 이미지는 이미지 타입으로 입력받아 JDK 1.2.2에서 지원하는 함수를 사용하여 변환하고, BMP 이미지는 파일의 헤더를 분석하여 필요한 정보를 알아내어 변환해야 한다. BMP 이미지는 JDK 1.2.2에서 지원하는 이미지 관련 함수들을 사용할 수 없는 반면 헤더의 구조가 비교적 간단하여 필요한 정보를 쉽게 알아낼 수 있다[8].

● 너비 및 높이 정보

입력받은 이미지의 너비와 높이는 WBMP 이미지의 크기를 결정하고, 각 픽셀에 대한 데이터를 저장하는 바디 부분을 만들 때에도 사용되므로 함수를 이용하거나 헤더를 분석하여 값을 구하고, 저장해 둔다[9].

● 이미지의 RGB 데이터

이미지의 각 픽셀에 대한 RGB 값은 이미지 변환을 위한 가장 중요한 정보이다.

인터넷 표준 이미지는 지원하는 함수를 사용하면 픽셀의 RGB 값을 쉽게 구할 수 있고, BMP 이미지는 파일의 헤더정보에 따라 해당하는 형태의 이미지로부터 R, G, B를 구한 후 이를 조합하여 저장한다.

● 단말기 정보 적용

이동단말이 보내는 정보 중 일부는 요청 이미지를 브라우징 할 수 있는 자원에 관계되므로 이미지 변환에 필요한 모든 정보를 구한 후 단말기의 정보를 적용시켜 가장 적절한 형태로 변환해야 한다.

이미지 변환에 관련되는 단말기 정보를 크게 네 가지로 구분하면 스크린 사이즈, 스크롤 기능 지원 여부, 전송 가능한 데이터의 최대 크기, 픽셀당 사용되는 비트수이다. 앞의 네가지 정보를 조합하여 해당하는 이미지로 변환하게 되는데 각각에 대한 내용을 살펴보면 다음과 같다.

- 스크린 사이즈 : 원본 이미지의 너비가 스크린 사이즈보다 크면 스크린 사이즈보다 작은 이미지로 변환해야 한다. 이미지 축소는 Backward Mapping 방법을 사용한다. 일반적으로 원본 이미지에서 결과 이미지로 변환이 이루어 지는데, Backward Mapping은 결과 이미지의 픽셀을 원본 이미지의 픽셀에 매핑시켜 픽셀값을 구하는 방법이다. 변환 과정을 요약해 보면 다음과 같다[10].

1. 스크린과 같은 너비이고 원본의 그레이-레벨 이미지(Img1)와 가로, 세로 비율이 일정하면서 픽셀 값이 없는 가상의 이미지(Img2)를 만든다.
2. Img2의 픽셀에 대해 그레이-레벨 값을 구하기 위해 Img1과 Img2의 비율을 계산하여 Img2의 각 픽셀을 Img1의 각 픽셀에 매핑시킨다. Img2의 각 픽셀 값은 Img1의 최대 4개의 픽셀에 의해 결정됨을 확인할 수 있다
3. 각 픽셀에 대해 그레이-레벨 Interpolation의 한 방법인 Bilinear Interpolation 방법을 사용하여 2의 결과로 나온 4개의 픽셀값을 조합하여 Img2의 그레이-레벨 값을 구한다. Bilinear Interpolation은 Img1의 4개의 픽셀값에 대해 Img2의 한 픽셀이 실제 매핑된 위치에서 4개의 픽셀까지의 거리와 같은 비율을 적용시켜 하나의 값을 구하는 방법이다.

- 전송가능한 데이터의 최대 크기 : WAP 클라이언트는 WAP 게이트웨이와 연결을 설정하면서 전송가능한 데이터 크기를 협상한다. 따라서 이미지의 너비가 스크린 크기보다 작더라도 변환기에 의해 변환된 이미지가 협상된 크기에 비해 크면 이미지는 클라이언트에 전송되지 않는다. 그러므로 WBMP 변환기는 WBMP 이미지를 생성하기 직전 단계에서 헤더와 바디의 크기를 검사하여야 한다. 만약 WBMP 이미지의 데이터가 더 크면 변환기는 이미지의 너비를 2픽셀 단위로 줄여가면서 데이터를 검사하다가 최대 전송크기보다 작게 되면 WBMP 이미지 파일로 생성한다.

- 스크롤 기능 지원 여부 : 스크롤 기능이 지원되는 단말일 경우, 스크린 사이즈가 원본 이미지보다 작더라도 이미지 변환에 영향을 미치지 않는다. 그러나 스크롤 기능이 지원되지 않는다면 스크린 사이즈와 최대 전송량을 고려하여 WBMP 이미지로의 변환이 이루어진다.

- 픽셀 당 비트수 : 현재 WAP은 픽셀을 0/1의 한 비트를 사용하여 나타낸다. 그러나 이동 단말이 컬러 또는 그레이-레벨 이미지를 지원한다면 WBMP 변환기는 단말의 자원을 최대한 사용하고, 시각적 효과를 높이기 위해 그레이-레벨 이미지로의 변환 기능이 추가되어야 한다[2].

● 임계값 결정

임계값은 입력받은 이미지의 픽셀값을 0 또는 1로 변환할 때 기준으로 사용되는 값이므로 이 값에 따라 WBMP 이미지의 픽셀값이 결정된다.

임계값은 미리 구해놓은 픽셀의 RGB 값을 0.33, 0.58, 0.11의 비율로 조합하여 0에서 255사이의 그레이-레벨 값으로 바꾼 후, 여기에 ITS(Iterative Threshold Selection) 알고리즘을 적용하여 계산한다. ITS 알고리즘을 간단히 설명하면, 이미지의 데이터가 물체와 배경으로 이루어져 있을 때 전체 이미지의 그레이-레벨 값을 바탕으로 물체와 배경의 경계가 되는 값을 구하는 것이다[11].

● 픽셀값 변환

임계값을 기준으로 각 픽셀 값을 그레이-레벨에서 흰색 또는 검은색으로 변환한다. 사용자에 의한 임계값 변화가 이 과정에서 첨가될 수 있는데 이 기능은 실시간으로 처리되는 응용에 유용하게 사용될 수 있다.

● WBMP 이미지의 데이터 크기 검사

최대 전송 크기에 대한 정보를 단말기로부터 받을 경우 생성될 WBMP 이미지의 크기를 미리 계산하여 최대 전송 크기보다 작은 이미지를 생성하기 위한 단계이다. 앞에서도 설명했듯이, WBMP 이미지가 최대 전송 크기보다 크면 이미지의 너비를 2픽셀씩 줄여가며 다시 WBMP 이미지로 변환한다.

● WBMP 헤더와 바디 생성

앞에서 설명한 모든 과정에서 저장해 둔 정보를 이용하여 WBMP의 헤더와 바디를 생성한다. 이 과정은 WAE(Wireless Application Environment) 스펙 1.2에 정의된 WBMP의 포맷에 따라 구조체를 선언해 두고, 필드에 이미지의 해당하는 값을 넣는 작업이다.

● WBMP 파일 생성

헤더와 바디 생성이 끝나면 이 데이터를 파일에 쓰고, 확장자가 '.wbmp'인 WBMP 파일을 생성한다.

5. WBMP 이미지 변환 결과

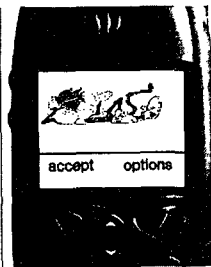
다음은 여러 단말기에 대해 동일한 이미지를 요청하였을 때 WBMP 변환기에 의해 변환된 후 WAP 게이트웨이로부터 전송된 WBMP 이미지의 예를 보여준다.



<그림 2> JPEG 파일



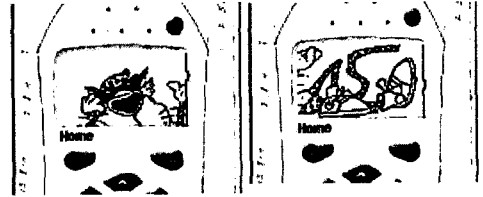
<그림 3> WBMP 파일 <그림 4> KNU WAP Browser



<그림 2>는 원본 이미지 파일이고 <그림 3>은 <그림 2>를 WBMP 이미지로 변환한 것이다. <그림 4>는 본 연구실에서 개발한 'KNU WAP Browser'를 사용하여 WBMP 이미지를 나타낸 것이다. 이미지 요청시 브라우저의 스크린 크기를 함께 보냈으며 그 결과로 <그림 3>과 비교해 볼 때 원본 이미지가 축소된 WBMP 이미지로 변환되어 전송되었음을 확인할 수

있다.

<그림 5>는 Nokia WAP Toolkit 2.0을 사용하여 변환 결과를 보인 것이다. 스크롤이 가능하며 게이트웨이로 스크린의 크기에 대한 정보를 전송하지 않기 때문에 축소되지 않은 WBMP 이미지가 전송되었다.



<그림 5> Nokia WAP Toolkit 2.0

6. 결론 및 향후과제

본 논문에서 구현한 WBMP 변환기는 기존의 이미지를 헤더의 크기가 작고, 픽셀을 표현하는 데이터의 수가 적은 WBMP 이미지로 변환함으로써 무선선로에서의 데이터 양을 최소화할 뿐만 아니라 이동 단말의 정보를 적용하여 단말에 보다 적절한 이미지를 제공한다.

현재 1-depth 이미지로의 변환만 가능하지만 그레이-레벨 또는 컬러-레벨까지 확장된 이미지 변환 기능이 추가된다면 원본 이미지에 가까운 시각적 효과를 얻을 수 있고, 이동 단말의 자원도 보다 효율적으로 사용할 수 있으므로 현재 제공되는 서비스의 질적 향상을 기대할 수 있다.

참고문헌

- [1] *Wireless Application Protocol Architecture*, WAP Forum, Apr. 30. 1998. URL: <http://www.wapforum.org>
- [2] *Wireless Application Environment Specification*, WAP Forum, Mar. 29. 2000. URL: <http://www.wapforum.org>
- [3] 김기조, 최윤석, 최은정, 임경식, "무선 응용 프로토콜 기술," 정보처리학회지 제 7권 제 3호, 2000년 5월.
- [4] *Composite Capabilities / Preference Profiles: Requirements and Architecture*, W3C, Jul. 21. 2000. URL: <http://www.w3.org>
- [5] *Wireless Application Group User Agent Profile Specification*, WAP Forum, Nov. 10. 1999. URL: <http://www.wapforum.org>
- [6] *Wireless Session Protocol Specification*, WAP Forum, May. 4. 2000. URL: <http://www.wapforum.org>
- [7] RFC2068, "Hypertext Transfer Protocol-HTTP/1.1," Jan. 1997. URL: <ftp://ftp.isi.edu/in-notes/rfc2068.txt>
- [8] Steve Rimmer, *Windows Bitmapped Graphics*, Windcrest/McGraw-Hill, 1993.
- [9] 손동익, *Windows 이미지 그래픽 프로그래밍*, 에이스 출판사, 1993.
- [10] Scott E Umbaugh, *Computer Vision and Image Processing*, Prentice-Hall International, 1998.
- [11] Milan Sonka, Vaclav Hlavac and Roger Boyle, *Image Processing Analysis and Machine Vision*, Chapman & Hall, 1993.