

# 무선 인터넷 서비스를 위한 WBMP 이미지 변환기 및 해석기 구현†

김태주<sup>o</sup>, 최은정, 이해동, 임경식  
경북대학교 컴퓨터과학과  
e-mail : {taejoo, ejchoi, hdlee, kslim}@ccmc.knu.ac.kr

## An Implementation of the WBMP Image Converter and Interpreter for Wireless Internet Services

Taejoo Kim<sup>o</sup>, Eunjeong Choi, Haedong Lee, Kyungshik Lim  
Dept. of Computer Science, Kyungpook National University

### 요 약

본 논문에서는 무선선로상의 제약과 무선 단말기의 제한된 자원을 이용하여 무선 인터넷 서비스를 좀 더 효율적으로 제공하기 위한 WBMP(Wireless BitMaP) 변환기와 해석기의 설계 및 구현에 대하여 기술한다. WBMP 변환기는 기존의 이미지를 단색의 압축되지 않은 형태인 WBMP 이미지로 변환하고, 단말기 내의 WBMP 해석기는 WBMP 이미지를 해석하여 브라우저를 통해 나타낸다. WBMP 이미지는 헤더의 구조가 간단하고, 이미지 표현에 사용되는 데이터의 양이 적으므로 무선 환경에 적합한 형태이다. 그러나 컬러 이미지를 변환하면 이미지 정보가 손실되거나 왜곡될 수 있고, 표현된 컬러의 수가 많을수록 손실률은 높아진다. 따라서 향후 그레이-레벨 이미지 또는 컬러 이미지로의 변환 및 해석 기능을 확대 적용할 필요가 있다.

### 1. 서론

최근 이동통신 가입자가 급속히 증가하고 인터넷이 일반화되면서 휴대성과 이동성으로 대표되는 이동통신을 인터넷과 결합한 새로운 형태의 무선 인터넷 서비스가 보편화되고 있다. 휴대용 전화나 PDA(Personal Digital Assistant)를 이용한 무선 인터넷 서비스는 공간적 제약이 적은 무선 터미널로 언제 어디서나 인터넷에 접속하여 다양한 정보검색, 전자상거래 등을 가능하게 하였으며 특히 텍스트 위주의 정적인 서비스에서 이미지가 추가된 서비스로 변함에 따라 가독성이 증가하였고, 이미지를 다양하게 응용함으로써 서비스의 질적 향상을 가져왔다[1].

그러나 기존의 이미지를 효과적으로 나타내는 데는 무선선로와 단말기상의 제약조건이 문제가 되므로 이를 해결할 수 있는 적절한 방법이 필요하다. 우선, 무선선로에서의 높은 에러 발생률과 지연, 협소한 대역폭 등의 문제점이 있으므로 가능한 한 전송되는 데이터의 양을 감소시킬 필요가 있다. 다음으로 무선 단말

기에서의 낮은 CPU 처리속도, 메모리의 부족, 작은 화면크기, 짧은 전지 수명 등 서비스에 필요한 자원이 제한되어 있으므로 데이터의 양을 최소화하여 메모리 소모를 줄이고, WAP(Wireless Application Protocol) 게이트웨이에서 가능한 한 많은 처리과정을 거치도록 하여 무선단말기에서의 처리과정을 줄여야 한다[2]. 이를 고려하여 WAP 포럼에서는 이미지를 표현할 때 이동단말과 WAP 게이트웨이 사이에서 WBMP 이미지를 사용하도록 정의하고 있다[3].

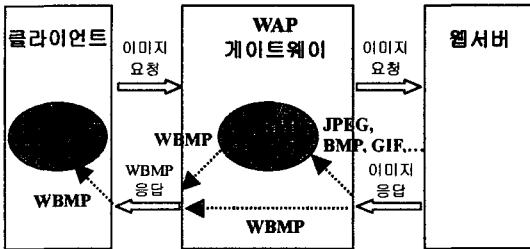
### 2. WAP 동작원리

WAP은 무선 인터넷 서비스를 지원하기 위한 국제적 개방 통신 규약으로 인터넷망과 이동통신망의 접속점에 게이트웨이를 두어 두 프로토콜 체계 사이의 중계역할을 하도록 한다[4, 5].

<그림 1>은 이미지 표현을 위한 WAP 네트워크 동작원리를 보여준다. 클라이언트가 이미지를 요청할 경우 WAP 게이트웨이 내에 그 이미지가 없으면 게이트웨이는 웹서버에 다시 요청하여 해당하는 이미지를 받는다. 클라이언트는 WBMP 이미지와 일반 이미지

† 본 연구는 정보통신부 초고속정보통신 응용기술개발사업 지원으로 수행되었음.

모두 요청이 가능하며, 클라이언트에 의해 요청된 이미지의 형식이 WBMP가 아닌 경우에는 WBMP 변환기에 의해 WBMP 이미지로 변환되어 클라이언트로 전송된다. 전송된 WBMP 이미지는 클라이언트 내의 WBMP 해석기에 의해 브라우저 가능한 형태로 바뀌게 되고, 브라우저를 통해 표현된다.



<그림 1> WAP 네트워크 동작원리

WBMP 변환기는 이미지 파일을 무선통신환경과 이동단말기의 제약환경에 적합한 새로운 이미지 형태로 변환하고, 단말기에 탑재되어 있는 WBMP 해석기는 WBMP 이미지를 브라우저를 통하여 표현할 수 있는 형태로 만든다.

### 3. WBMP

WBMP는 무선환경에서 그래픽 정보 표현을 가능하게 하기 위해 무선환경의 특성에 맞게 정의된 새로운 이미지 형식이다. 협소한 대역폭, 무선선로에서의 높은 에러 발생률, 메모리 부족 등의 제약이 있는 무선환경에서 기존의 이미지를 이용한 서비스를 제공하는 데는 많은 어려움이 있다. 따라서 전송되는 데이터의 양, 무선 단말기에 저장할 데이터의 양 등을 최소화할 필요가 있는데, WBMP 이미지는 다른 이미지 파일과 비교할 때 헤더의 구조가 간단하고, 이미지를 나타내는 데 사용되는 데이터의 양이 적으므로 이동단말의 제한된 메모리 내에서의 그래픽 정보 표현에 적합하다.

#### 3.1 WBMP 자료구조 및 인코딩

WBMP는 헤더와 이미지 데이터로 구성되어 있으며, <표 1>은 WBMP 헤더의 구조를 나타낸다.

<표 1> WBMP 헤더 구조

필드 이름	길이(bit), b,k,m,n:임의의 정수
타입	8...8*h
고정헤더	8
확장헤더	8...8*k
너비	8...8*m
높이	8...8*n

타입 필드는 WBMP의 형식을 지정하는데, 현재는 타입 0 이미지만 지원이 가능하다. 타입 0 이미지는 픽셀 당 한 비트를 사용하여 검은색 또는 흰색으로 표현되며 압축되지 않은 이미지를 나타낸다. 고정헤더

필드는 8비트로 길이가 고정되어 있으며 일반적인 헤더의 정보를 나타낸다. 확장헤더 필드는 타입 0 이미지에서는 사용되지 않고, 부가적인 헤더 정보를 명시하거나 최적화 또는 애니메이션 이미지 등의 특수 목적으로 사용된다. 너비와 높이 필드는 이미지의 크기를 나타내며, 필드의 길이는 이미지의 크기에 따라 동적으로 결정된다.

WBMP 이미지 데이터의 자료구조는 헤더에 정의되어 있는 타입에 따라 달라진다. 현재는 압축되지 않은 단색 이미지를 나타내는 타입 0 이미지의 데이터 구조가 정의되어 있다. 타입 0 이미지의 데이터에서 픽셀의 각 행은 너비 한 행을 나타내며 옥텟의 배열로 이루어진다. 또, 각 픽셀의 명암도를 나타내기 위해 한 비트를 사용하며 1은 흰색, 0은 검은색을 나타낸다. 픽셀의 각 행을 옥텟으로 표현할 때, 행의 길이가 8로 나누어지지 않을 경우에는 옥텟의 나머지 부분에 0을 추가시킨다. 따라서 각 행의 인코딩은 옥텟의 첫 비트에서 시작하게 된다. 데이터를 인코딩 할 때에는 최상위 비트가 먼저 인코딩되는 빅-엔디안 방식을 사용하고, 가장 먼저 인코딩 된 옥텟이 먼저 전달되도록 빅-엔디안 방식을 사용하여 나열한다[3].

### 4. WBMP 변환기 구현

#### 4.1 구현환경

ALZZA Linux release 6.0 상에서 JDK 1.2.2를 사용하여 구현하였는데, JDK 1.2.2는 O/S에 독립적이고, 이미지 처리에 필요한 여러 가지 함수들을 제공하는 장점이 있다. 그러나 함수를 사용할 수 있는 이미지의 타입에 제한이 있으므로 구현 시 이를 고려해 주어야 한다.

#### 4.2 동작과정 및 구현

WAP 게이트웨이가 웹서버로부터 이미지를 받으면 먼저 이미지의 타입을 검사하여 WBMP가 아닌 경우 WBMP 변환기를 이용하여 WBMP 이미지로 변환시키게 된다.

이미지를 입력받은 변환기는 헤더를 분석하여 너비와 높이를 추출하는데, 이는 WBMP 이미지의 너비와 높이를 결정할 뿐만 아니라 이미지 처리를 위한 여러 가지 경우에 이용된다. 다음으로 픽셀의 R(Red), G(Green), B(Blue)를 추출하여 이미지 변환의 기준이 되는 임계값을 결정한다. 임계값이 결정되면 이를 기준으로 각 픽셀 값을 0 또는 1로 변환하여 WBMP 이미지의 데이터를 생성한다. 헤더와 데이터가 모두 만들어지면 클라이언트가 지정하거나 임의로 정한 이름으로 WBMP 이미지 파일을 생성함으로써 변환기의 동작이 끝나게 된다. 변환기의 각 동작에 따른 구현 방법을 기술하면 다음과 같다.

##### • 이미지 입력

먼저 이미지의 타입을 검사하여 이미지를 입력받는 방법을 결정해야 한다. 인터넷 표준 이미지는 JAVA에서 지원하는 getImage() 함수를 이용하여 이미지로 입력받고, BMP 이미지는 이 함수를 사용할 수 없으므로 파일로 입력받는다. BMP 파일은 헤더

의 두 바이트를 읽어 첫 번째 바이트가 'M'이고, 두 번째 바이트가 'B'인가를 검사하여 확인한다[6].

• 너비 및 높이 추출

인터넷 표준 이미지의 경우, JAVA에서 지원하는 getWidth() 함수와 getHeight() 함수를 이용하면 너비와 높이를 쉽게 구할 수 있으나 BMP 이미지의 경우에는 파일의 헤더를 분석하여 너비와 높이를 찾아야 한다[7, 8].

• 픽셀의 RGB 추출

이미지의 RGB 또한 인터넷 표준 이미지의 경우에는 JAVA에서 지원하는 grabPixels() 함수를 이용하여 구하고, BMP 이미지는 파일을 읽어서 데이터 즉, 픽셀의 RGB가 저장되어 있는 곳을 찾아서 구하는데, BMP 이미지의 종류에 따라 RGB를 데이터 부분에 나타낼 수도 있고 파일 내에 팔레트를 두어 저장할 수도 있으므로 이를 고려하여야 한다[6].

파일의 헤더 정보에 따라 해당하는 형태의 이미지로부터 픽셀의 RGB를 구한 후에는 이를 조합하여 하나의 픽셀 값을 계산한 후 저장한다.

• 임계값 결정

임계값은 WBMP 데이터를 만들 때 기준으로 사용되는 값이며, 앞에서 구해놓은 픽셀의 RGB를 0.33, 0.58, 0.11의 비율로 조합하여 0에서 255사이의 그레이-레벨 값으로 바꾼 후, 그 값을 바탕으로 ITS (Iterative Threshold Selection) 알고리즘[9]을 적용하여 계산한다.

다음은 ITS 알고리즘에 대해 설명한 것이다.

- 이미지는 물체와 배경으로 이루어져 있으며, 물체의 정확한 위치정보는 모르는 것으로 가정한다.
- 0에서 255사이의 중간값인 127 또는 128을 임시 임계값으로 정한다.
- 이전 단계에서 결정된 임계값(Ti)을 기준으로 이미지를 물체와 배경으로 구분하여 물체의 평균(tO)과 배경의 평균(tB)을 각각 계산한다.
- tO와 tB의 평균을 구하여 이 값을 새로운 임시 임계값(T(t+1))으로 결정한다.
- Ti와 T(t+1)이 같으면 이를 임계값으로 결정하고, 그렇지 않으면 이 값을 기준으로 이미지를 물체와 배경으로 구분하는 단계부터 이전 임계값과 새로 구한 임계값을 비교하여 임계값을 결정하는 단계까지 반복한다. 두 값이 같아질 때까지 반복 수행한 후의 값이 이미지의 임계값이 된다.

• 픽셀 값 변환

임계값이 결정되면 이 값을 기준으로 각 픽셀값을 흰색 또는 검은색으로 변환한다. 변환된 WBMP 이미지에 대해 배경과 물체의 전환기능, 사용자 입력을 통한 임계값 변화에 의한 이미지 변환기능, 시각적 효과를 높이기 위한 확산기능 등이 이 과정에서 첨가될 수 있다.

• WBMP 헤더와 데이터 생성

이미지의 각 픽셀에 대해 WBMP 데이터로 변환하는 작업이 끝나면 지금까지 계산하여 저장해 둔

모든 정보를 이용하여 WBMP 파일을 생성한다. WBMP 파일 생성은 WAE 스펙 1.2에 정의된 WBMP의 포맷에 따라 구조체를 선언해 두고, 필드에 이미지의 해당하는 값을 넣는 작업이다. <표 2>는 이미지 변환을 위해 WBMP 포맷에 따라 선언한 헤더와 이미지의 구조를 나타낸다.

<표 2> 구현에 사용되는 WBMP 구조

필드 이름	길이(bit), k,l,m:임의의 정수
Type	8
FixHeader	8
Width	8...8*k
Height	8...8*l
Data	8...8*m

현재 타입 0 이미지만 지원되므로 Type 필드에는 0을 넣는다. 또한, 타입 0 이미지는 확장헤더를 사용하지 않으므로 FixHeader 필드에 0을 넣고, 확장헤더를 나타내는 필드는 생략한다. Width 필드는 이미지의 너비에 따라 동적으로 만들어지므로 미리 계산을 한 후 그 결과를 넣는다. 너비가 결정되면 그 다음부터 Height 필드가 되며 이 필드 또한 이미지의 높이에 따라 필드의 길이가 동적으로 결정되므로 미리 계산을 해야 한다.

헤더의 생성이 끝나면 앞에서 구한 픽셀 값을 이용하여 데이터 부분을 생성한다. 데이터의 크기는 이미지의 너비와 높이에 의해 결정되는데, 데이터의 크기가 결정되면 한 비트가 픽셀 하나의 컬러를 나타내도록 앞에서 구한 픽셀값을 넣어준다. 이미지의 너비를 하나의 행으로 나타내므로 너비가 8에 의해 나누어지지 않을 경우 사용하지 않는 비트를 필요한 만큼 행의 끝에 추가시켜야 한다[3].

• WBMP 파일 생성

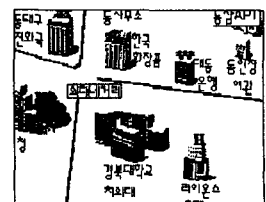
데이터의 생성이 끝나면 앞에서 만들어 놓은 헤더와 합쳐서 WBMP 파일을 생성한다. 이때 파일 이름은 클라이언트에서 지정하여 요청할 수 있고, 파일 이름에 대한 명시가 없다면 입력된 이미지 파일 이름에서 확장자만 '.wbmp'로 변경한다.

4.3 이미지 변환 예

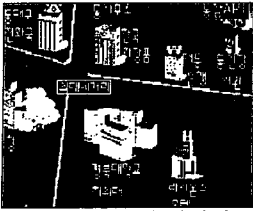
<그림 2>는 JPEG 이미지이고 <그림 3>은 WBMP 변환기를 이용하여 <그림 2>를 WBMP 이미지로 변환한 것이다. <그림 4>는 물체와 배경의 전환기능의 예이고 <그림 5>는 이미지의 시각적 효과를 높이기 위해 확산기능을 적용한 예이다.



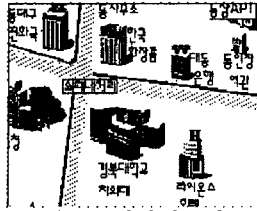
<그림 2> JPEG 이미지



<그림 3> WBMP 이미지



<그림 4> 물체,배경이 전환된 이미지



<그림 5> 확산기능이 적용된 이미지

## 5. WBMP 해석기 구현

클라이언트에서 이미지를 요청하면 WAP 게이트웨이는 WBMP 변환기를 이용하여 요청된 이미지를 WBMP 이미지로 변환한 후 다시 클라이언트로 전송한다. 브라우저는 게이트웨이로부터 받은 WBMP 이미지를 디코딩하여 보여주어야 하는데, 무선 단말기 내의 WBMP 해석기가 WBMP 이미지를 디코딩하는 역할을 수행한다.

### 5.1 구현환경

ALZZA Linux release 6.0 상에서 JDK 1.2.2를 사용하여 구현하였다.

### 5.2 동작과정 및 구현

WBMP 해석기는 게이트웨이로부터 WBMP 이미지를 입력받아 우선 헤더를 분석하여 이미지의 너비, 높이 등 필요한 정보를 추출하고, 데이터를 분석하여 픽셀 값을 추출한다. 그리고 이 값들을 이용하여 이미지를 생성한 후, 브라우저로 반환하면 브라우저가 브라우저에게 된다. 해석기의 각 동작과정에 대한 구현 방법을 기술하면 다음과 같다.

#### • 헤더 분석

파일의 첫 번째 바이트는 타입을 나타내며, 해석기는 파일을 입력받으면 우선 이 값을 검사하여 WBMP 파일이 맞는지 확인하여야 한다. 다음 한 바이트는 고정헤더 필드이며, 현재 지원되는 WBMP 이미지는 확장헤더를 사용하지 않으므로 이 바이트의 5, 6번째 비트의 값이 0인지 확인하여야 한다. 파일의 세 번째 바이트부터 이미지의 너비를 나타내는데 멀티-바이트로 인코딩되어 있으므로 먼저 너비를 나타내는 바이트 수를 알아내어 해당하는 바이트만큼 디코딩하여 너비를 구한다. 다음 바이트부터 이미지의 높이를 나타내며 높이도 너비를 구하는 것과 같은 방법으로 구한다.

#### • WBMP 데이터 추출

이미지의 너비에 따라 데이터가 저장되어 있는 형태가 다르므로 이를 고려하여 픽셀값을 추출해야 한다. 너비를 8로 나눈 나머지를 검사하여 0이면 추가되지 않은 경우이므로 파일의 옥텟을 차례대로 읽어 한 비트씩 저장한다. 추가된 비트가 존재하는 데이터의 경우에는 파일의 옥텟을 차례대로 읽어 한 비트씩 저장하다가 한 행의 마지막 옥텟에서는 미리 계산한 나머지만큼만 읽어 저장하고, 그 다음 부분은 버린다.

## • 이미지 생성 및 반환

입력받은 이미지를 분석하여 너비와 높이, 픽셀값을 구한 다음, 이 값들을 이용하여 이미지를 생성해야 한다. JDK 1.2.2를 사용하여 구현할 경우, JDK1.2.2에서 제공하는 createImage() 함수를 이용하면 이미지는 쉽게 생성되며 생성된 이미지는 브라우저로 반환한다.

WBMP 이미지의 데이터는 검은색 또는 흰색 즉, 한 비트가 0 또는 1의 값을 가지므로 픽셀값을 이용하여 이미지를 생성하는 과정에서 브라우저 할 이미지의 색을 임의로 결정할 수 있다. 일반적으로 이미지의 배경은 브라우저의 배경과 같은 색으로 정하고, 물체는 검은색으로 나타낸다.

## 6. 결론 및 향후과제

본 논문에서 구현한 WBMP 변환기와 해석기는 기존의 이미지를 헤더의 크기가 작고, 1-depth 이미지로 변환하여 표현함으로써 무선선로에서의 데이터 양을 최소화했을 뿐만 아니라 무선 단말기에서의 이미지 처리부담을 경감시켰다.

현재 1-depth 이미지로의 변환만 가능하지만 앞으로 그레이-레벨 또는 컬러-레벨까지 확장된 이미지 변환 및 해석 기능이 추가된다면 원본이미지에 가까운 시각적 효과를 얻을 수 있다. 또한 현재 사용되지 않고 있는 WBMP 헤더의 확장 필드를 이용하는 Animated WBMP 이미지에까지 그 기능이 확대 적용된다면 무선 환경에서도 동적 이미지를 이용한 서비스가 가능해지므로 현재 제공되는 서비스의 질적 향상을 기대할 수 있다.

## 참고문헌

- [1] 이해동, 최윤석, 임경식, "SDL을 이용한 무선 트랜잭션 프로토콜의 설계 및 구현," 제10회 통신 정보 합동 학술대회(JCCI2000) 발표논문집, 2000년 5월.
- [2] 최은정, 임경식, "무선 인터넷 서비스를 위한 WAP 게이트웨이용 바이너리 WML 인코더의 구현," 제10회 통신 정보 합동 학술대회(JCCI2000) 발표논문집, 2000년 5월.
- [3] *Wireless Application Environment Specification*, WAP Forum, Nov. 4. 1999. URL: <http://www.wapforum.org/>
- [4] 김기조, 최윤석, 최은정, 임경식, "무선 응용 프로토콜 기술," 정보처리학회지 제7권 제3호, 2000년 5월.
- [5] *Wireless Application Protocol Architecture*, WAP forum, Apr. 30. 1998. URL: <http://www.wapforum.org/>
- [6] Steve Rimmer, *Windows Bitmapped Graphics*, Windcrest/McGraw-Hill, 1993.
- [7] 안철호, *윈도우 그래픽 입문*, 정보문화사, 1994.
- [8] 손동익, *Windows 이미지 그래픽 프로그래밍*, 에이스출판사, 1993.
- [9] Milan Sonka, Vaclav Hlavac and Roger Boyle, *Image Processing, Analysis and Machine Vision*, Chapman & Hall, 1993.