

# Radial-Basis Function Networks를 이용한 영상 압축 방법

## (An Image Compression Method using Radial-Basis Function Networks)

이재영<sup>†</sup> 김황수<sup>\*\*</sup>

(Jaeyoung Lee)(Hwang-Soo Kim)

**요약** 본 논문에서는 인간 시지각을 고려한 새로운 영상 압축 방법을 제시한다. 영상의 화소의 값들이  $x$ - $y$  평면상에서 정의된 3차원 곡면 위에 있는 점들로 가정하여, 영상을 곡면의 복잡도에 따라 나누고, 나누어진 각각의 곡면(영역)은 Radial-Basis Function (RBF)를 사용하여 근사화하는 방법으로 영상을 압축한다. 본 방법은 JPEG 압축 방법과 비슷한 압축율과 영상의 질을 얻을 수 있었다.

**Abstract** In this paper, a novel image compression technique is proposed, which takes human visual perception into account. Pixel values (intensity) are considered to be points on a three-dimensional surface defined on the  $x$ - $y$  plane. The image is partitioned according to the complexity of the surface, and then each partitioned surface is approximated using Radial-Basis Functions(RBF). This method is very efficient and has comparable performance and quality of JPEG image compression technique.

### 1. 서론

현재, 통신망 기술의 발전과 더불어 인터넷과 같은 실제적 응용이 보편화되고 있다. 인터넷상의 자료는 대부분 영상과 같은 멀티미디어 자료를 포함하고 있다. 이들은 사고를 통해 인식되는 문자와 같은 정보의 표현 방법이 아니라, 직관적으로 획득, 해석할 수 있으므로 사용자가 쉽게 정보를 얻을 수 있게 해 준다. 그러나, 이와 같은 정보의 표현법은 일반적 문자 표현법에 비해 많은 양의 메모리가 필요하고, 통신망 상에서의 자료 교환에 많은 시간이 소요되므로 정보의 표현용량을 줄이는 압축방법이 개발되고 있다. 본 논문에서는 곡면근사법에 근거한 새로운 압축방법을 제시한다.

현재 효율적인 압축방법들이 많이 나와있다. 이 방법

들은 DCT 변환을 이용한 방법, Wavelet 변환을 이용한 방법, 자기 유사성을 이용하여 영상을 함수들의 집합으로 표현한 Fractal 압축 방법 등이다. 이 방법들은 자체로 뛰어난 성능을 가지고 있으며, 관련된 연구도 상당히 활발하다. 그러나, 이 방법들은 인간 시지각과 무관하게 수치적 처리를 통하여 영상을 압축해 왔다. 그러므로, 좀더 인간 시지각과 밀접하게 관계된 영상 내에서의 정보를 활용하여 영상을 압축하고자 하는 시도들이 계속되고 있다. 이와 같은 방법들로서는 영상 분할을 통한 방법, 에지 추출을 통한 방법들이 있다. 이와 같은 방법들은 인간의 시각이 에지에 민감하고, 일정 영역 내에서는 급작스런 밝기의 변화가 없다는 이론에 근거를 두고 제안된 방법들이다. 본 논문은 이와 같은 맥락에서 grey level (8 bits) 영상을 계산적이지 아닌, 좀더 직관적이고 효율적으로 압축하는 방법을 제시한다.

인간이 실제세계에서 보고, 듣고, 즉 지각하는 공간은 3차원의 유클리드 공간이다. 따라서, 영상 내부의 밝기 값들도 서로 조밀하게 연관된 하나의 물체로 간주할 수 있으며, 이것은 결국 표면을 구성한다고 볼 수 있다. 영상의  $(x,y)$  위치에서의 밝기 값을  $z = f(x,y)$  라고 표현

· 본 논문은 경북대학교 공과대학 연구비 지원에 의하여 수행되었음

<sup>†</sup> 비 회 원 : (주)미래산업 소프트웨어 연구원

intvis@softforum.com

<sup>\*\*</sup> 종신회원 : 경북대학교 컴퓨터학과 교수

hsk@knu.ac.kr

논문접수 : 1999년 12월 27일

심사완료 : 2000년 7월 10일

하면,  $z$ 를 높이로 볼 때  $z$ 가 곡면을 이룬다는 것이다. 여기서 제시한 영상 압축 방법은 이렇게 구성된 곡면을 몇 가지 함수들의 조합으로 재구성하여 영상을 표현하는 것이다. 어떤 영상이라도 하나의 함수로 전체를 나타내기에는 영상이 만들어낸 곡면이 너무 불규칙하기 때문에 불가능하다. 따라서, 영역을 주어진 함수에 가장 잘 맞는 영역으로 분할하여 각 분할된 영역을 함수들의 조합으로 나타내도록 한다. 여기서 사용되는 분할 방법은 Quadtree이고, 함수는 Radial-Basis Function (RBF) 들이고, 이 Radial-Basis Function들의 Network (RBFN) 를 구성하여 곡면을 근사화한다.

**2. Radial-Basis Function Network (RBFN)**

RBFN은 신경망의 일종으로 곡면 근사화 문제에 잘 적용되는 신경망이다. 본래 신경망은 사람의 신경세포의 정보처리 방식을 모형화 한 것인데 일반적인 컴퓨터의 직렬 정보처리 방식과 대조되는 고도의 병렬처리 특징이 있다. 신경망은 여러 가지 응용 분야 중에서 함수 근사 문제에도 상당한 효율성을 보여주고 있다. 그 중 RBFN이 고차원의 곡면 근사화 문제에 잘 동작하도록 설계되었다. 일반적으로, RBFN은 임의의 곡면을 정규 분포와 형태가 유사한 봉우리 형태의 곡면들의 선형 조합으로 만들어서 근사화하는 것이다

**2.1 Radial-Basis Function (RBF)**

RBF들은 특정 함수들의 클래스이다. RBF들의 특징은 중심으로부터 거리가 멀어질수록 함수 값들이 단조증가 또는 감소한다는 것이다. 이런 특징으로 인해 RBF들은 대체로 지역성의 특징이 있다[1].

대표적인 RBF는 Gaussian 함수이다. 1차원 일 경우에 다음과 같이 표현된다.

$$G(x) = \exp\left(-\frac{(x-c)^2}{\sigma^2}\right)$$

$c$ 는 중심,  $\sigma$ 는 그 함수가 지원하는 반경

**2.2 RBFN의 구성**

간단히 말하면, RBFN은 RBF의 선형 조합으로[1,2,3], 수식으로 표현할 때,

$$F(X) = \sum_{i=1}^m w_i G(\|X - c_i\|)$$

$X$ 는 입력 벡터

$c_i$ 는 RBF의 중심을 나타내는 벡터

$\{G(\|X - c_i\|) | i=1, 2, \dots, m\}$ 는 RBF의 집합

와 같이 표현되며, 다음과 같은 구조 (그림 1) 를 가지는 네트워크가 된다. (RBF의 개수는 일반적으로 근사화할 대상의 점들의 수보다 월등히 작게 한다.)

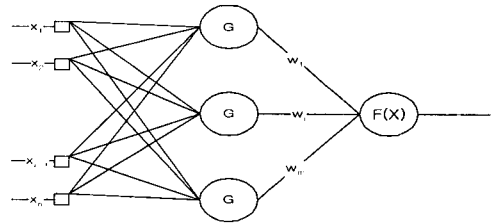


그림 1 RBFN의 형태

**2.3 RBFN의 파라미터 결정**

RBFN의 파라미터들은 RBF들의 선형조합으로 주어진 곡면을 근사화 할 때 사용되는  $w_i$  들을 말하는 것이다. RBFN에서는 이 파라미터를 신경망에서 쓰는 용어로 weight 라고 한다. 따라서, 이 파라미터는 최선의 결과를 가져올 수 있도록 결정해야 한다. 이것은 에러 역전파와 같은 반복적인 에러 수정을 통하여 구할 수 있지만, 이것은 시간이 무척 많이 걸리고 그 결과가 안정적이지 않다. 따라서 현재 알고리즘에서는 최소 자승법을 이용해서 이 파라미터 (weight) 들을 구한다.

원래의 영상의 한 부분을  $F(X)$  ( $X = (x,y)$ ) 라는 함수로 볼 때, 우리가 원하는 것은  $F(X)$ 라는 함수와 가장 유사한 (에러가 적은) 함수  $F^*(X)$ 를 구하는 것이다. 따라서, 수식으로 나타내면,  $F^*(X)$ 를 RBFN으로 구성하므로 다음과 같은 비용함수의 값을 최소화하는  $w_j$  들을 구하면 된다.

$$\xi(F) = \sum_{i=1}^N (d_i - \sum_{j=1}^m w_j G(\|X_i - c_j\|))^2$$

여기서  $N$ 은 점들의 개수,

$d_i$ 는  $F(X)$ 상의  $i$ 번째 점에 대한 함수값을 나타낸다.

이것을 행렬의 형태로 쓰면,

$$\mathbf{d} = [d_1, d_2, \dots, d_N]^T$$

$$\mathbf{G} = \begin{bmatrix} G(\|X_1 - c_1\|) & G(\|X_1 - c_2\|) & \dots & G(\|X_1 - c_m\|) \\ G(\|X_2 - c_1\|) & G(\|X_2 - c_2\|) & \dots & G(\|X_2 - c_m\|) \\ \vdots & \vdots & \ddots & \vdots \\ G(\|X_N - c_1\|) & G(\|X_N - c_2\|) & \dots & G(\|X_N - c_m\|) \end{bmatrix}$$

$$\mathbf{w} = [w_1, w_2, \dots, w_m]$$

이고,  $\|\mathbf{d} - \mathbf{G}\mathbf{w}\|^2$ 의 형태가 된다.

따라서, 이것을 최소화하는  $\mathbf{w}$ 는 Pseudo-Inverse 방법을 이용하여 구한다. (점들의 개수와 RBF의 개수가 다르기 때문에  $\mathbf{G}$ 는 정방 행렬이 아니다.)

$\mathbf{G}^T \mathbf{G} \mathbf{w} = \mathbf{G}^T \mathbf{d}$  이므로 Pseudo-Inverse 행렬은  $\mathbf{G}^+ = (\mathbf{G}^T \mathbf{G})^{-1} \mathbf{G}^T$  가 되고,  $\mathbf{w}$ 는  $\mathbf{w} = \mathbf{G}^+ \mathbf{d}$  로 구할 수 있다. 이렇게 구해진  $\mathbf{w}$ 를 가지고 RBF들을 조합하면 원하는 영역의 근사화 된 곡면(영상의 밝기)을 재구성 할 수 있

다.

그림 2는 RBF들의 개수와 위치를 미리 정한 후에, 임의로 입력된 점들에 대해 RBFN으로 근사화한 곡선을 보여준다. 그림에서 알 수 있듯이, 각 RBF들이 입력된 점들에 맞게 변화되어 있다.

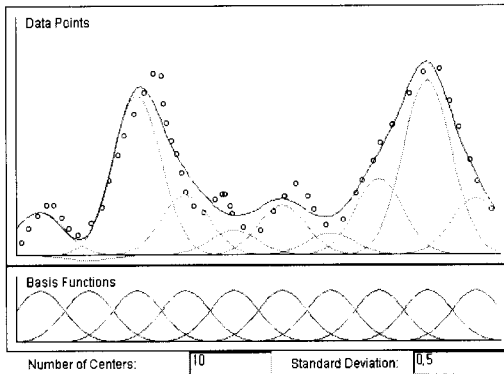


그림 2 주어진 점들에 대해 RBFN으로 근사화 한 결과  
(출처: <http://exodus.mit.edu/~jesse/java/rbfunc/>)

### 3. Quadtree

Quadtree는 영상을 영역분할을 통해 표현하거나, 평면상의 자료들을 분할하는데 이용되는 자료구조이다[4, 5]. 이 자료구조는 평면상의 공간을 주위와의 관련성에 따라, 유사성이 많은 것은 같은 영역으로 묶고 다른 것은 분할하여 다른 영역으로 분리해 표현하는 것이다. 영상 압축에의 응용은 임의로 영상 내의 화소를 선택한다면, 그 주위의 화소들도 비슷한 밝기를 가진다는 사실에 근거를 둔다. 영상을 영역별로 유사한 밝기값을 가지는 지를 찾아서 유사한 밝기이면 더 이상 분할하지 않고, 다른 밝기라고 판단되는 밝기 값이 존재하면 더 작은 영역으로 분할한다. 따라서, Quadtree의 크기는 영상의 복잡도에 비례한다.

Quadtree를 만드는 방법은 하나의 부모로부터 출발하여, 그것을 4개의 사각영역으로 나누고 나누어진 각각은 부모의 자식이 된다. 이 때, 유사한 밝기 값의 자식은 더 이상 분할되지 않고, 유사하지 않은 부분이 존재하면 더 분할될 수 있는 내부 노드가 된다. 따라서, 영상의 복잡도가 크다면, Quadtree의 크기가 커지게 된다. 이 Quadtree는 표현하는데 드는 용량이 적고, 또한 분할된 영역의 형태가 일정하므로 여러 장점이 있다. 그 중 형태가 일정하다는 점이 논문에서 제안한 압축 알고리즘에서 RBF의 개수와 중심 위치 설정에 많은 이점을 제

공한다.

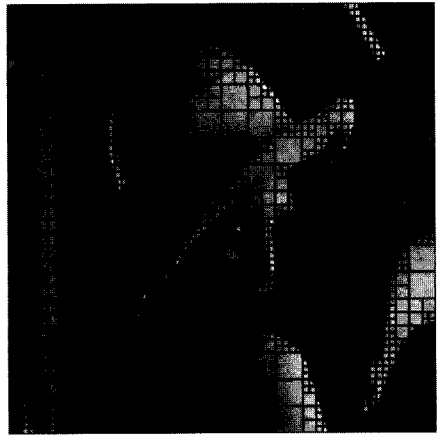


그림 3 Lena 영상에 Quadtree 분할을 적용한 결과

## 4. 압축 알고리즘

### 4.1 영역 분할

영상 전체를 하나 또는 몇 개의 함수로 표현하기는 어렵다. 실제 영상은 곡면의 형태로 볼 때, 매우 불규칙하고 불연속적이므로, 이것을 여러 개의 작은 영역으로 분할하여, 분할된 영역을 표현할 수 있는 최선의 함수를 찾아야 한다. 이 분할을 수행하는 방법이 Quadtree이다. Quadtree는 단순하고, 구현이 쉬우면서 효율적 분할을 수행할 수 있다. 본 논문에서 제안한 알고리즘의 성능을 가장 크게 좌우하는 것이 무엇을 기준으로 분할, 즉 Quadtree를 만드느냐이다. 따라서, RBFN으로, 분할된 영역을 곡면으로 근사화 할 때 근사화 에러를 이용하여 분할기준 검사함수를 다음과 같이 정할 수 있다.

Function Split\_Test( 현재 Partion의 크기 )

```
{
  for (RBF의 모든 패턴에 대하여)
    RBFN으로 현재 영역을 근사화;
    if (현재 패턴에 대한 RBFN이
        주어진 에러 한계치를 만족한다면 )
      return TRUE; // 더 이상 분할하지 않음.
if (현재 Partion의 크기가 허락된 Partition의 최소 크기에 도달했다면)
  return TRUE; // 더 이상 분할하지 않음.
return FALSE; // 더 작게 분할함.
}
```

위 함수가 앞에서 설명한 Quadtree의 분할조건 함수이다.

이 함수에서 여러 형태의 RBFN들에 대해 검사하고 영역의 꼭면을 근사화 한다. 여기서 사용되어지는 에리함계치는 사용자에게 의해 결정된다.

**4.2 RBF의 개수 및 중심의 결정**

Quadtree를 이용한 영역 분할이 전체 압축비율을 결정한다면, RBF의 개수 및 중심의 결정은 각 RBFN의 성능, 즉 근사화 능력을 결정한다고 볼 수 있고, 이것이 압축된 영상의 질을 좌우한다고 볼 수 있다. 실제로 RBFN의 구성에 가장 어려운 부분이 RBF의 개수 및 중심의 결정이다. 이것은 사전에 미리 입력에 대한 모든 정보를 알고 있어야 가능하다. 그러므로, 미리 알고 있지 못한 형태의 꼭면을 근사화 할 때 상당한 결함으로 작용한다. 이 전의 방법들로는 중심의 결정은 랜덤하게 정하는 방법, Self-Organizing Map에 의한 방법, Supervised Learning Network에 의한 방법[3] 등으로 결정할 수 있으나, RBF의 개수의 결정은 휴리스틱에 의존할 수밖에 없다. (미리 입력에 대한 정보를 다 알아야 바르게 정할 수 있다.) 하지만, 위의 방법들 모두 만족할 만한 결과를 얻을 수 없다. Support Vector Machine[3,6,7]을 이용하여 적합한 중심의 위치와 RBF의 개수를 결정하는 데 도움을 줄 수 있으나, 본 논문에서는 사용하지 않는다. 왜냐하면, 압축은 영상의 표현용량을 줄이는데 목적이 있는데 반해, 자동으로 결정하면 RBF의 개수 및 중심이 입력 데이터에 의존적으로 임의로 결정되므로 표현 용량이 커지기 때문이다. 따라서, 본 논문에서는 RBF의 개수 및 중심을 각 분할된 영역에서 미리 개수와 상대적 좌표에 의한 위치를 패턴으로 결정해 놓은 후에, 이 패턴 중에서 근사화를 해 본후 원본과 가장 차이가 작은 패턴을 선택하여 RBFN으로 만든다. 실험에서 사용한 RBF의 개수 및 중심들의 패턴들은 다음과 같다. (그림 4 - 원은 RBF를 나타낸다.)

이렇게 패턴으로 만들어 놓음으로 인해, 많은 량의 부가적 표현 용량을 줄일 수 있다. 물론 이것들 중에 선택된 하나가 가장 좋은 결과 (꼭면 근사화 결과) 를 가져온다고는 보장할 수 없다. 하지만, 압축은 자료의 표현용량을 줄이는데 우선적인 목적이 있으므로, 결국 압축율과 압축된 결과의 질과의 절충이라고 볼 수 있다. 이 패턴들의 구성은, 우선 적은 수의 RBF를 사용하는 것을 기본으로 하여, 최대한의 꼭면 근사화 결과를 얻을 수 있는 RBF들의 개수와 위치를 결정하는 원칙으로 이루어 졌다. 아주 불규칙한 형태를 가진 영상의 표현에서 어떤 원칙에 따라 정형화된 패턴을 설계하는 것은 불가능하다. 따라서, 위의 패턴들은 여러 영상들에 대해 실험 해 본 결과, 압축율과 질을 동시에 만족할 수 있는

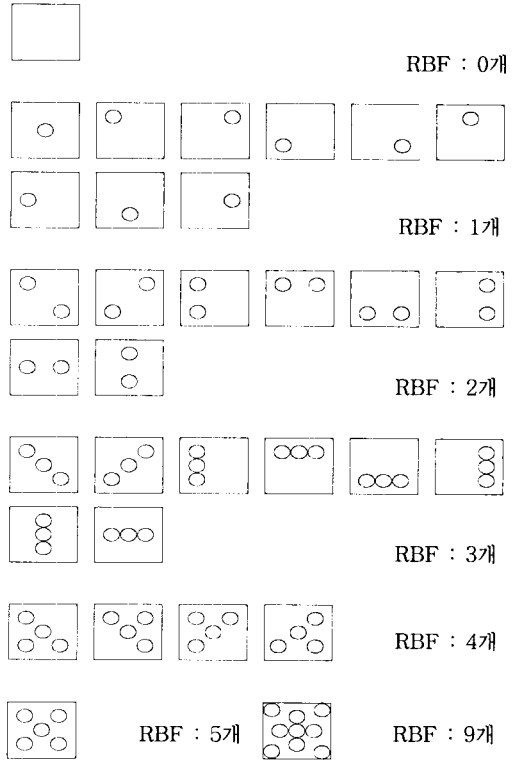


그림 4 RBF의 개수 및 중심의 위치에 대한 패턴들

형태들로 결정된 것이다. 또한 실험에서 만족할만한 결과를 보여주고 있다.

**4.3 하나의 분할 영역 표현**

Quadtree 분할 영역에 근사화 하는 RBFN는 다음과 같이 나타낸다. 1) 현재의 분할된 영역의 Quadtree 내에서의 깊이, 2) 현재 분할된 영역에서의 RBF의 개수 및 중심의 좌표, 3) 근사화 된 RBFN의 weight들이다.(그림 5)

깊이 (3 bits)	패턴 (5 bits)	양자화 된 인자들 (각 8 bits씩)
----------------	----------------	--------------------------

깊이 : 현재 분할된 영역의 Quadtree에서의 깊이  
 패턴 : 사용된 RBF의 패턴  
 인자들 : RBFN의 weight들

그림 5 분할된 한 영역에 대한 RBFN으로의 표현법

깊이는 현재 분할된 영역의 크기와 디코딩 할 때에 위치 정보를 제공해주며, 여기서는 3 bit 로 표현한다.

RBFN의 개수 및 중심의 위치는 사용한 패턴들이 32 가지이므로 5 bit 로 표현된다. 근사화 된 RBFN의 weight들은 8 bits 로 균일하게 양자화되어 표현되어진다. 양자화 과정은  $x$ 번째 weight값이  $f(x)$ 라 할 때, 다음과 같은  $f'(x)$ 로 표현된다.

$$f'(x) = \lfloor \frac{f(x)}{\Delta s} \rfloor,$$

$\Delta s$ 는 양자화 할 간격을 나타냄.(에러율을 좌우함.) 위와 같이 균일하게 양자화 한 이유는, 결정된 RBFN의 weight들이 아주 불규칙하므로 예측기를 만들 수가 없기 때문이다. DPCM, DM[4]과 같은 방식을 사용할 수가 없기 때문이다. 그리고 실험에서도 균일한 양자화가 괜찮은 성능을 보였다.

따라서, 이 분할된 영역의 정보의 배열들이 최종적으로 원래 영상의 다른 표현법이 된다.

4.4 출력 파일의 구조

파일의 구조는 헤더와 몸체로 나뉘어지며 헤더는 변환시의 옵션들과 DC 성분을 저장하며, 몸체는 분할된 영역의 정보의 배열들이 된다.(그림 6)

헤더는 최대 깊이, 양자화 간격, DC와 같은, 압축 시의 조건들을 가지는 데, 각각이 나타내는 것은 다음과 같다. 최대 깊이는 분할될 때 최소 영역의 크기를 나타낸다. 즉 최대 깊이 이상의 영역분할은 하지 않게 된다.

최대 깊이	양자화 간격	DC	영역 자료	...	영역 자료
-------	--------	----	-------	-----	-------

그림 6 출력 파일의 구조

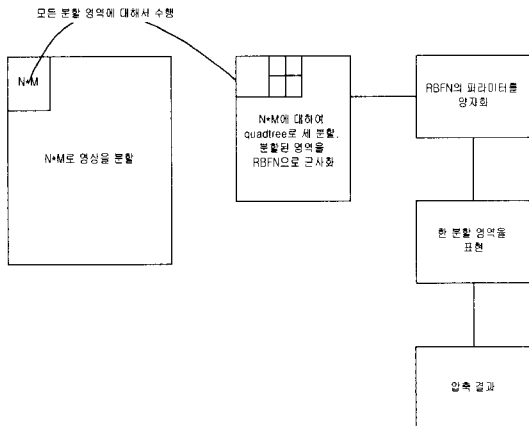


그림 7 전체 압축 과정

양자화 간격은 균일하게 양자화 할 때에 사용한 간격을, DC는 영상 전체 밝기의 평균값이다. 압축 알고리즘을 적용 시, DC를 영상에서 뺀 후에 적용하는 데 이것은, 표면은 0의 밝기 값 근처로 끌어내리기 위함이다. 이렇게 함으로서 근사화의 성능을 향상시킬 수 있다.

물론, 이것은 RBFN으로 영상을 표현할 때에 표현 용량을 줄이기 위해 Bias를 사용하지 않기 때문에 일어나는 문제이다. 만약, 모든 Network 마다 Bias를 사용하면, 표현 용량이 너무 커지므로 Bias를 사용하지 않는 대신, 전체 DC를 제거하여 따로 저장하는 것이다.

그림 7은 제안한 압축 방법에 따른 전체적인 흐름을 블록 다이어그램으로 표시한 것이다.

5. 결 과

압축 알고리즘은 물론 원래의 영상 자료의 용량을 얼마나 줄일수 있는나 압축율과 이렇게 압축된 영상이 얼마나 원래의 영상과 유사하냐는 압축의 질로 압축 알고리즘의 성능이 평가된다.

압축율은 압축된 후의 파일의 크기로 원래 영상 파일의 크기로 나누어 구할 수 있으며, 압축된 영상의 질을 측정하는 척도로 여러 가지가 있을 수 있으나, 여기서는 PSNR을 이용하였다.

$$PSNR = 20 \log \frac{255}{(\frac{1}{N} \sum_{x=0}^N (f(x_i) - f'(x_i))^2)^{1/2}}$$

- $N$  : 영상의 전체 화소 개수
  - $f(x)$  : 원래 영상의  $x_i$ 에서의 밝기 값
  - $f'(x)$  : 재구성된 영상의  $x_i$ 에서의 밝기 값
- 와 같이 계산된다.

그림 8은 512×512 영상으로 테스트한 결과이고, 그림 9은 Lena 영상에서 JPEG과의 결과를 비교한 것이다. 그림 10는 고 압축율에서 제안한 알고리즘의 결과와 JPEG[4,5,8]과의 영상을 보여준다.



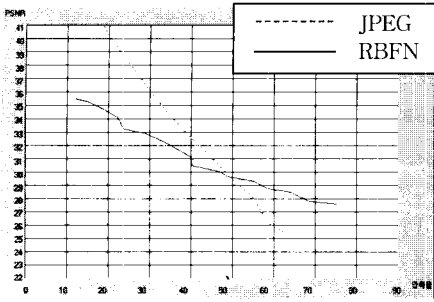
압축율 : 23.65  
PSNR : 33.35 dB

압축율 : 27.36  
PSNR : 32.01 dB

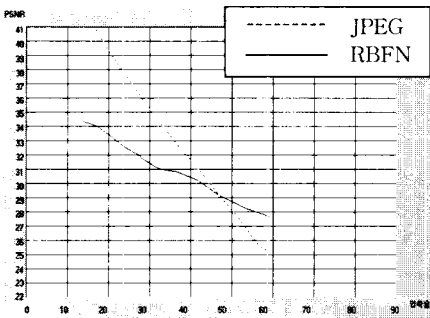
(a) Lena

(b) Peppers

그림 8 제안한 알고리즘의 적용 결과



(a) Lena



(b) Peppers

그림 9 JPEG과의 결과 비교



압축율 : 79.12  
PSNR : 28.55 dB

(a) 제안한 방법



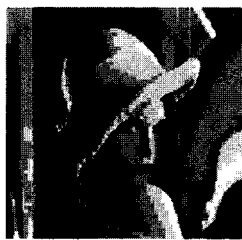
압축율 : 65.67  
PSNR : 25.31 dB

(b) JPEG



압축율 : 55.14  
PSNR : 29.32 dB

(a) 제안한 방법



압축율 : 62.39  
PSNR : 25.47 dB

(b) JPEG



압축율 : 53.39  
PSNR : 28.28 dB

(a) 제안한 방법



압축율 : 60.13  
PSNR : 25.06 dB

(a) 제안한 방법

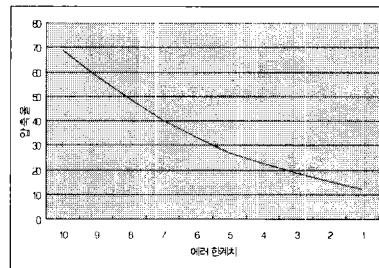
그림 10 JPEG과의 결과 비교 (Lena, Girl과 Peppers영상 : 512x512)

표 1 영역크기 제한에 따른 인코딩 소요 시간 (Pentium II 400MHz 에서 측정)

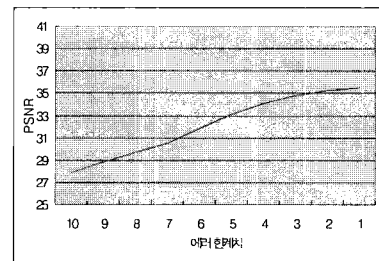
최소 영역의 크기	64×64	32×32	16×16	8×8	4×4
소요 시간 (sec)	120	218	285	319	340

표 2 사용한 RBFN의 갯수에 따른 인코딩 소요 시간 (Pentium II 400MHz 에서 측정)

영역당 사용한 RBFN의 갯수	9개	5개	4개	3개	2개	1개	0개
소요 시간 (sec)	319	288	275	218	124	57	5



(a) 압축율과의 관계



(b) PSNR과의 관계

그림 11 에러 한계치에 따른 압축율과 PSNR의 관계

표 1은 최소영역 크기에 따른 인코딩 소요 시간을 표로 나타낸 것이다.

그림에서 볼 수 있듯이 높은 압축율에서 JPEG보다 더 좋은 결과를 보여주고 있다. 또한 영역 내부에서의 미묘한 밝기의 변화를 JPEG에 비해 잘 표현하고 있다. 이것은 RBF의 부드러운 특성에 기인한다고 볼 수 있다.

그림 11은 Quadtree 분할시 에러 한계치의 변화에 따른 압축율과 PSNR의 변화를 보여준다. 에러 한계치가 작을수록 더 압축율은 떨어지고, 영상의 질은 좋아진다.

## 6. 결론 및 향후 과제

본 논문에서 제안한 방법은 일반적으로 연구, 사용되는 영상 압축 알고리즘과는 달리 영상의 밝기 값들을 서로 연관성을 가진 면들로 보고, 곡면 근사화 방법을 사용하여 영상을 압축하는 새로운 시도이다. 기존의 압축방법들은 신호처리에서 사용되는 여러 변환을 이용하여 시각적으로는 의미없는 자료로 만들어 영상의 전체적인 특징(시각 효과)은 고려하지 않고 결과의 어려움에만 의존해서 영상을 압축하는 방식(수식으로 처리하기 편하게 가공을 하는 방식)인데 반해, 제안한 방법은 먼저 인간의 시지각적인 측면(즉 인간이 어떻게 영상에서 물체를 인식하는가)에 대해서 고려한 방법이다. 인간이 영상을 보면 그것을 3차원으로 인식하듯이 영상의 각 부분을 3차원의 굴곡있는 곡면으로 고려하여 좀 더 자연스러운 압축영상을 얻을 수 있다(원래 영상을 가공하지 않는다). 따라서, 원래의 영상이 가지는 공간적 특성을 잘 나타낼 수 있으며, 한 물체의 내부에서의 밝기 값의 자연스러운 변화를 더 잘 표현 할 수 있다.

JPEG 방법과 비교할 때, 하나의 영역 내부에서의 변화를 잘 표현 할 수 있으나 영상의 에지(윤곽선) 부분에서는 어느 정도의 질의 저하를 가져온다. 하지만, 압축율에 따라 이 문제를 해결할 수 있으며, 압축율이 높아질수록 JPEG보다 전체적인 성능이 우수하다. 하지만, 블럭 코딩 방식을 사용하는 다른 압축 방법들이 가지는 문제점과 마찬가지로, 블럭 형태의 에지가 나타나는 문제점이 있다. 이것은 예전부터 많은 연구가 행해지고 있지만, 손실 압축에서는 아직 해결하지 못한 문제이다.

본 알고리즘에서는 RBF의 개수와 중심의 선택에 상당한 영향을 받는다. 이 문제를 패턴들의 형태로 미리 정해놓고 사용하는데, 이 패턴이 가장 적합하다는 근거는 없다. 다만, 많은 시행착오를 거쳐 압축률과 질의 균형을 얻을 수 있는 패턴을 찾아낸 것이다. 만약 이 패턴을 입력에 적합하게 정할 수 있다면 훨씬 나은 결과를 얻을 수 있을 것이다. 마지막으로, 현재 방법은 근사화

과정에서 많은 시간을 소비함으로써 실시간 응용에는 많은 제한이 있다. 이것을 좀 더 빠르게 개선하여 실시간 응용에도 이용할 수 있도록 해야할 것이다.

## 참 고 문 헌

- [1] Mark J. L. Orr, "Introduction to Radial Basis Function Networks," Centre for Cognitive Science, University of Edinburgh, April 1996.
- [2] Suresh K. Lodha, Richard Franke, "Scattered Data Techniques for Surface," University of California, Santa Cruz, UCSC-CRL-97-19.
- [3] Simon Haykin, Neural Network - A Comprehensive Foundation, 2nd Edition, Prentice-Hall, 1999.
- [4] Rafael C. Gonzalez, Richard E. Woods, Digital Image Processing, Addison-Wesley, 1992.
- [5] David Salomon, Data Compression - The Complete Reference, Springer-Verlag, 1998.
- [6] B. Scholkopf, K. Sung, C. Burges, F. Girosi, P. Niyogi, T. Poggio, V. Vapnik, "Comparing Support Vector Machines with Gaussian Kernels to Radial Basis Function Classifiers," MIT AI Laboratory, A. I. Memo No. 1599, December 1996.
- [7] T.-T. Friess, N. Cristianini, C. Campbell. "The kernel adatron algorithm: a fast and simple learning procedure for support vector machines," 15th Intl. Conf. Machine Learning, Morgan Kaufman Publishers. 1998.
- [8] Gregory K. Wallace, "JPEG Still Picture Compression Standard," Communications of ACM, vol. 34, pp.30-44, April 1991.



이 재 영

1998년 경북대학교 컴퓨터과학과 학사.  
2000년 경북대학교 컴퓨터과학과 석사.  
2000년 ~ 현재 (주)미래산업 소프트웨어  
연구원. 관심분야는 신경망, 컴퓨터 비전,  
신호 처리 등임.



김 황 수

1975년 서울대학교 전기공학부 졸업(학사). 1982년 Univ. of Michigan 석사학위 취득. 1988년 Univ. of Michigan 박사학위 취득. 1989년 ~ 현재 경북대학교 컴퓨터과학과 부교수로 근무. 관심분야는 computer vision, neural networks, AI임.