

논문 2011-4-27

# 닮은꼴을 이용한 영상구현 임계값설정과 정지영상 복원법

## Establishment Threshold Value of Image Realization & Reconstruction of Stoppage Image using Picture Resemblance

진현수\*

Hyun-Soo Jin

**요 약** 본 논문에서는 JPEG(Joint Photographic Experts Group)영상 디코더에서 영상데이터를 디코딩하는 기술에 관한 것으로, 종래의 정지영상 처리기에서는 허프만 코드화 처리를 하여 제1영상을 부호화한후 분할기반 부호화 방식의 연속영상 처리기법을 사용하였으나 작은화면의 코드화시에도 많은 계산 알고리즘이 필요하고 이미지의 닮음정도를 객관적수치로 표현할 수 없어서 비슷한 이미지를 중복 코드화 및 전송 작업을 하는 결함이 있었다. 본 논문에서는 이를 해결하기 위하여 이미지의 닮음정도를 수치화하고 기준값 이하의 영상은 코드화 작업을 하지않고 계산된 영상 변화값에 의해 구현된 영상으로 대체하여 저장 및 전송하도록 저장용량의 감축 및 시간적으로 전혀 거부감이 없는 정지영상을 구현하였다.

**Abstract** In this paper, JPEG(Joint Photographic Experts Group) image data video decoding technique is presented, it is Huffman decoding method and fractal image method which is very complexive algorithm and the time required much it to implement this method and the first image is decoded to video frame image. This have defect of overlap decoding and transport work because of impossible to represent objective value of resemblance. The proposed method was calculated the mathematical absolute image resemblance and simplify the moving picture process to reducing the step of moving picture codefying. The results show that smoothed moving picture compared recent methods.

**Key Words** : Threshold value, Huffman decoding, Image Resemblance, fractal image method

### 1. 서론

JPEG은 칼라나 그레이 레벨의 정지화상 부호화를 다루고 있으며, 무손실 부호화와 손실허용 부호화 방식 모두에 대하여 작업이 이루어지고 있어 사용자로 하여금 목적에 맞는 방식을 선택할 수 있도록 하고 있다. 특히 화상의 저장, 표시 및 정보 교환을 원하는 경우, 화상정보를 있는 그대로 이용하여서는 엄청난 양의 데이터 저장 공간, 매우 빠른 데이터 처리속도를 갖는 광대역 통신망

이 요구된다. 따라서 정보 이용의 실용화는 거의 불가능하다고 할 수 있다.<sup>[1]</sup>

다행히 이차원 화상신호에는 상당량의 중복성 정보가 내포하고 있는 정확한 양의 정보데이터로 정확한 영상의 표시 및 화상 전송 통신선로의 대역폭 감축을 가져왔다. 그러므로 영상 신호의 중복성을 이용한 정지영상신호의 데이터 압축기술은 화상정보처리시스템 신호에 대한 영상압축 실현을 위한 핵심기술이라 할 수 있겠다. 종래의 방법은 정지 영상을 코드화 하기 위하여 부호화 모드를 통한 허프만 부호화 코드를 사용하나 복잡한 알고리즘 방법과 많은 계산 시간을 소모하게 되고 또한 이를 활용하여 동영상 처리를 하고자 할 때는 객체기반 부호화 방

\*정회원, 백석대학교 정보통신학부

접수일자 2010.10.3, 수정일자 2011.6.29

게재확정일자 2011.8.12

식과 모델기반 부호화방식, 분할기반 부호화 방법을 사용하게되어 이중 명확한 이미지 전달 방법에는 뚜렷한 해답을 주지 못하고 있는 실정이다, 즉 MPEG2 비디오 디코더에서는 정지영상 데이터를 디코딩하는 기술에 있어서 화면을 디코딩하여 1초에 30프레임 이상의 화면을 전송 및 저장하여 정지영상의 움직이는 동영상 효과를 얻는 방법에 위의 방법들을 사용하나 종래의 방법은 영상의 1프레임 코드화시에도 많은 계산 알고리즘이 필요하고 비슷한 이미지를 중복 코드화 및 전송과 저장 작업을 하는 결함이 있었다. 따라서 본 논문은 이를 해결함에 있어 우선 영상의 이미지 영상맑은꼴을 정의하고 이를 절대 이미지 맑은꼴이라 정한후 각 화면마다 맑은꼴의 임계 기준값을 결정하여 기준 맑은꼴값 이하의 영상에 대해서는 동영상시 코드화 작업을 하지않고 계산된 영상 변화값에 의해 구현된 정지영상으로 대체하여 저장 및 전송하도록 하였다. 근래의 영상 압축기술은 허프만 지수를 넘지않는 범위내에서 압축 및 전송이 시행되어지나 허프만지수가 각 개별적인 이미지에서는 구별할 수 없어서 이를 각각의 개별적인 영상을 비교하기 위해서는 영상 맑은꼴을 사용하는 방법이 제시되고 있다.<sup>[8]</sup>

본 영상을 구현하기 위한 정지영상의 맑은꼴 비교값을 기준 프레임 저장공간에 표시하기 위하여 초당 30프레임에서 20프레임으로 줄이는 효과를 보였다.

## II. 화면의 맑은정도

### 1. 영상의이미지 맑은꼴값

그림(Picture)의 맑은정도를 평가하는 작업은 지금까지는 기하적(Geometric)형태로서만 평가 하여서 그림1과 같은 임의의 그림이 기하적으로 같을수 있다.

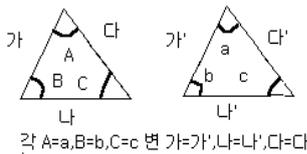


그림 1. 맑은꼴 기하그림  
Fig. 1. Geometry figure of resemblance

그러나 단순한 영상(Image)차원에서 맑음을 계산하고 자 할 때는 이 영상을 기하적(Geometric)적으로 분할하거나 복원하는 차원으로 해결할수가 없으므로 화소단위(Pixel)로 나뉘어 그림 2와 같이 어떤 화면이 본래의 화면과 더욱 많은 맑음정도의 화소를 포함하고 있는가를 나타내면 어떠한 형태의 영상도 서로 유사성이 있음을 발견 할 수가 있게 된다.

영상(image)-1	영상(image)-2
① ② ③ ④	① ② ③ ④
① 0 0 0 0	① 1 1 1 1
② 0 0 0 0	② 1 1 1 1
③ 0 0 0 0	③ 1 1 1 1
④ 0 0 0 0	④ 1 1 1 1

그림 2. 임의의 영상의 맑은꼴  
Fig. 2. Image resemblance

즉, 그림 2에서 이미지의 크기가 4x4화소 단위로 구성된 그림을 사용한다면 영상-1은 흑(Black)의 화소(Pixel)가 하나도 없는 흰색(White)의 화소를 나타내고 있고 영상-2는 반대로 흑의 화소만 나타나는 그림으로서 나타낼 수 있다. 두 그림을 비교한다면 전혀 영상의 내용으로서 는 관련이 없는 그림임을 알 수 있다. 이를 맑은꼴 비교 값이라는 수치로서 나타낸다면 무한대(∞)를 나타낸다.

### 2. 두개의 2진 정지 영상비교

위의 과정을 디지털 이진영상으로 비교하는 과정을 구체적으로 나타내면 다음 그림 3이다.

1화면	2화면	3화면
0 0 0 1 0 0 0	0 0 0 1 0 0 0	1 1 1 1 1 1 ①1
0 0 1 1 1 0 0	0 0 1 1 1 1 0	1 1 1 1 1 ①0 ①
0 1 1 1 1 1 0	0 1 1 1 1 1 0	1 1 1 1 1 ①1
1 1 1 1 1 1 1	1 1 1 1 1 1 1	1 1 1 1 1 1 1
0 0 0 0 0 0 0	0 0 0 0 0 0 0	1 1 1 1 1 1 1
	3화면	1,3화면 비교값
	0 0 0 1 0 0 0	①0 0 1 0 ①1
	1 0 1 1 1 1 0	0 ①1 1 ①0 ①
	0 1 1 1 1 1 0	①1 1 1 1 ①0
	1 1 1 1 1 1 1	1 1 1 1 1 1 1
	0 0 0 0 0 0 0	1 1 1 1 1 1 1

그림 3. 영상의 디지털 비교값  
Fig. 3. Digital compare value of image

첫 번째, 동영상상을 구현하기 위하여 우선 정지영상을 코드화 한다. 기본이 되는 영상(기준화면을)도 1에서 가장 첫 번째 입력되어지는 BS(Beat Stream)영상으로 정하면 그림2와 그림3의 1화면으로 나타내어질수 있는데, 기준 화면을 그림1의 12번 항목인 프레임 메모리에 16x16영상면을 데이터화 한다. 화소(pixel)가 있을 경우 '1'로 데이터화 하고 화소가 없을 경우 '0'로 데이터화 하여 전에 데이터화 한 화면중 기준화면 다음에 들어오는 화면도 기준화면과 비교하기 위하여 16x16크기로 분할하여 기준화면과 비교되어지는데 우선, 그림3에서와 같이 기준화면인 1화면과 다음번 화면인 2화면을 중첩하여 화소가 중첩이 될 경우 '1'로 데이터화하고 중첩이 되지 않을 경우 '0'로 데이터화하여 비교화면 데이터를 그림3의 1,2화면 비교값, 1,3화면 비교값 처럼 생성한다. 그림3에서는 화면을 7x7로 나타내어 실행하였다.

1화면	2화면	3화면
0 0 0 1 0 0 0	0 0 0 1 0 0 0	0 0 0 1 0 0 0
0 0 1 1 1 0 0	0 0 1 1 1 1 0	0 0 1 1 1 1 0
0 1 1 1 1 1 0	0 1 1 1 1 1 0	0 1 1 1 1 1 0
1 1 1 1 1 1 1	1 1 1 1 1 1 1	1 1 1 1 1 1 1
0 0 0 0 0 0 0	0 0 0 0 0 0 0	0 0 0 0 0 0 0

4화면	5화면
0 0 0 1 0 0 0	0 0 0 1 0 0 1
1 0 1 1 1 1 0	1 0 1 1 1 1 0
0 0 1 1 1 1 0	1 0 1 1 1 1 0
1 1 1 1 1 1 1	1 1 1 1 1 1 1
0 0 0 0 0 0 0	0 0 0 0 0 0 0

(a)

1화면	2화면	3화면
0 0 0 1 0 0 0	1 1 1 1 1 1 1	1 1 1 1 1 1 1
0 0 1 1 1 0 0	1 1 1 1 1 1 1	1 1 1 1 1 1 1
0 1 1 1 1 1 0	1 1 1 1 1 1 1	1 1 1 1 1 1 1
1 1 1 1 1 1 1	1 1 1 1 1 1 1	1 1 1 1 1 1 1
0 0 0 0 0 0 0	1 1 1 1 1 1 1	1 1 1 1 1 1 1

3화면	1,3화면 비교값
①0 0 0 1 ① 1	①1 1 1 1 1 ①'
0 ①0 1 ① 1 ①	0 ①1 1 1 ① 0
①1 1 1 1 ① 1	①1 1 1 1 1 ①
1 1 1 1 1 1 1	1 1 1 1 1 1 1
0 0 0 0 0 0 0	1 1 1 1 1 1 1

(b)

그림 4. 영상의 닮음꼴 2진 비교값 구하기  
Fig. 4. Binary compare value of image resemblance

### III. 이진 영상의 닮은 비교값 구하기

실제로 2진영상의 비교값을 구하여 보면 그림 4(a),(b)와 같이 기준 영상 데이터를 1화면이라하고 비교할 데이터 화면을 2,3화면이라 할 때 비교된 1,2비교 데이터 화면과 1,3비교 데이터화면의 각 화소 데이터 가운데 이때 그림4에서는 다음 계산값을 통해 기준 화면과 2,3화면과의 절대 영상 닮음꼴 계산을 다음과 같이 수행한다, 2화면의 영상 닮음꼴계수는 1,2화면 비교값에서 등급(class)2로 인한 화소인 ①이 4곳 있고 각각 떨어져 있는 거리가 1x+1y이므로 총거리로 인한 값은 4 x(1x+1y)=4x+4y이고 1,3화면의 영상 닮음꼴 계수는 등급(class)2의 화소(pixel)군이 2개가 생기고 두 계수값을 비교하면 3화면이 2화면보다 거리계수값이 크므로 1화면과 덜 닮았다는 것을 나타내고 있다. 화면당 좌표값을 해당 화면의 '절대 닮은꼴치'라 설정한다. 지금 까지 과정을 그림1의 영상 비교기가 수행하게 된다. 그림5에서는 위의 절대 닮은꼴 계산 과정을 순서도로 표시하였다.

### IV. 정지화면 연결 과정

위와 같은 처리과정을 거쳐 동영상으로 복원하기 위해서 위의 과정을 연결하여 진행 한다면 세 번째, 이와 같은 정지영상의 영상 닮음꼴값을 계산한 후, 그림6의 13번 항목인 동영상 복원기 실행과정을 그림 5에서 설명하였다, 기준화면인 1화면 외에 4개의 정지화면이 입력이 되어 '0'이고 1,3화면의 영상 닮음꼴 계산값은 4x+4y이고 1,4화면의 영상 닮음꼴 계산값은 15x+2y이어서 화소의 평균 변화값은 0+(4x+4y) +(17x+2y)/3=7x+2y이고 화면 전송의 가부판정 임계값으로 쓰이는 수치는 (20x+20y)/(7개의 화소/16개의 화소)=15x+15y이므로 화면2에서 화면 4까지는 직접 전송하지 않고 평균변화값의 변화분으로 대체된 화면으로 복원하게 되고, 나머지 1,5화면의 영상 닮음꼴 계산은 등급1인 화소 ①이 5곳이 있으므로 거리값 10x+2y이고 등급 2인 화소를 위해서 위의 과정을 연결하여 진행 하면된다. 세 번째, 이와 같은 정지영상의 영상 닮음꼴값을 계산한 후, 그림6의 13번 항목인 동영상 복원기 실행과정을 그림 5에서 설명하였는데, 기준화면인 1화면 외에 4개의 정지화면이 입력이 되어 '0'이고 1,3화면의 영상 닮음꼴 계산값은 4x+4y이고 1,4화면의 영상

닭은꼴 계산값은  $15x+2y$ 이어서 화소의 평균 변화값은  $0+(4x+4y)+(17x+2y)/3=7x+2y$ 이고 화면전송의 가부판정 임계값으로 쓰이는 수치는  $(20x+20y)/(7개의 화소/16개의 화소)=15x+15y$ 이므로 화면2에서 화면 4까지는 직접 전송하지 않고 평균변화값의 변화분으로 대체된 화면으로 복원하게 되고, 나머지 1,5화면의 닭은꼴 계산은 등급 1인 화소 ①가 2곳 있으므로 거리값  $30x(1x+1y)=30x+30y$ 이므로 총 닭은꼴 계산 값은  $40x+32y$ 으로 임계값을 초과하여 직접 전송하여야 하므로 다음 기준화면으로 사용하기 위하여 프레임메모리에 저장되어진다.

1화면	2화면
(0,0)	(7,0)
0 0 0 1 0 0 0	0 0 0 1 0 0 1
0 0 1 1 1 1 0 0	0 0 1 1 1 1 1 0
0 1 1 1 1 1 1 0	0 1 1 1 1 1 1 0
1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1
0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0
(0,7)	(7,7)
3화면	4화면
0 0 0 1 0 0 0	0 0 0 1 0 0 1
0 0 1 1 1 1 1 0	0 0 1 1 1 1 1 0
0 1 1 1 1 1 1 0	0 0 1 1 1 1 1 0
1 1 1 1 1 1 1 1	0 1 1 1 1 1 1 1
0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0

(a)

그림 5. 동영상 처리화면 결정도  
Fig. 5. Conclusion of Dynamic image

1. 정지영상 연속처리과정구성

위의 과정은 C++로서 프로그램을 구성하였는데 정지 화면은  $256 \times 256$  pixel로서 구성하여  $16 \times 16$  pixel로서 나뉘어 정지영상을 비교실험 및 동영상 복원 실험과정을 거쳤다. 그림 6에서 보는바와 같이 동영상 처리기의구성은 입력되어지는 화상신호를 비트 스트림으로 받아들여 기준 영상을 토대로 다른 영상과의 비교를 수행하는 영상 비교기와 비교한 데이터를 저장하는 프레임 메모리로 구성할 수 있는데 프레임 메모리는 주소의 지정열이 들어오는 데이터의 순서대로 저장하였다. 출력시켜주는 열은 램으로 처리하였으며 영상 복원기는 프레임 메모리에 저장되어 있는 영상 데이터중 프로그램은 랭귀지 C++로 구성하였고 처리화면은  $256 \times 256$  픽셀로 처리하고 그에

대한 화면은 그림6에 나타내었다.  $8 \times 8$ 화면은  $32 \times 32$ 개의 화면으로 나뉘어지는데 닭은꼴 비교값은  $8 \times 8$ 화면당 비교하게 된다. 픽셀당 비교한 값을 구분하여 닭은꼴값을 계산하여 내는 과정을 그림 7에 나타내었다. 등급으로 구분된 픽셀들의 모임에서 계산값을 얻어낼 때 처리화면은  $256 \times 256$ 픽셀로 구성하였다. 픽셀당 비교한값을 구분하여 닭은꼴 값을 계산하여 내는 과정은 그림 6에 나타내었다. 알고리즘 진행과정을 행렬로서 표시하였는데 그림 6에서  $a_{n,n}$  주위의 행렬 원소  $a_{n,n-1}$ ,  $a_{n-1,n}$ ,  $a_{n,n+1}$  을 조사하여 '1'의 개수를 파악한다. 가장 높은등급인 클래스 1은 1의 개수가 4개, 클래스2는 3개,클래스3는 2개,클래스4는 1개,클래스4는 0개로 파악하여  $8 \times 8$ 픽셀을 모두 조사한다.조사한 픽셀이 주소가 판정이 되었으면 같은 픽셀의 근접도를 좌표값으로 구한다. 화면의 구성은 왼쪽 상단으로부터 오른쪽 하단으로 이루어 지게 되며  $8 \times 8$ 화면을 채우게 된다.총32개의 가로세로 화면을 채우면 다시 다음프레임의 화면을 조사하면 된다.

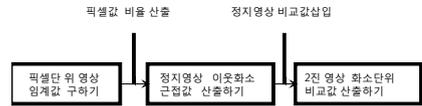


그림 6. 정지영상 처리기  
Fig. 6. Stoppage image processor

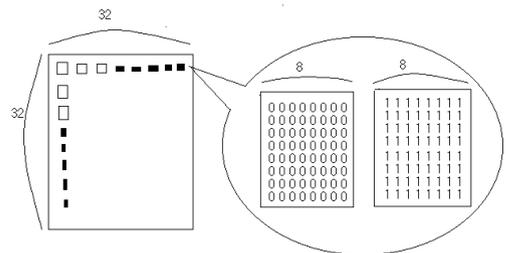


그림 7. 화면구성 픽셀 배치도  
Fig. 7. Fixel display digram of composing image

알고리즘 구성을 위하여 8개의 블록들 가운데 각 각 픽셀들을 중첩하고 중첩된 화면을 조사하여 '1'과 '0'의 배치를 통해 그림 7의 화면원소 구성도를 완성한다.좌표값의 위치원소값을 계산하여 각 클래스별로 좌표의 거리를 계산하여야 하는데 2차원 배열 평면의 분산도를 취하

여 각 클래스별 거리의 분산도를 구하게 된다.

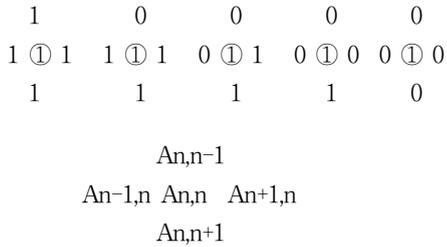


그림 8. 화면의 블록 구성도  
Fig. 8. Image block digram

좌표값의 위치원소값을 계산하여 각 클래스별로 좌표의 거리를 계산 하여야 하는데 2차원 배열평면의 분산도를 취하여 각 클래스별 거리의 분산도를 구하게 된다. 클래스1의 분산은 X좌표축상의 7.2이고 Y좌표축상의 8.3이므로 평면 분산값에 대한 거리계산값은 9.3이라 할 때 그림 8에 대한 식은 그림 8의 하위에 위치하는 모든 식이된다.

층별의 계산은 층간 관계를 설정하여야 한다. 속도가 느린 층이 가장 크다. 즉, 기본 자체값을 갖고 그 이후로 속도의 층별로 비율값을 정한후 기본값을 분할하여 가지게 된다. 속도를 구현하는 방법은 분산값이 크면 속도가 느리고 분산값이 크면 속도가 빠르게 된다.좌표값을 위치 원소값을 계산하여 각 클래스별로 좌표의 거리를 계산하여야 하는데 2차원 배열 평면의 분산을 취하여 각 클래스별 위치 분산을 구하게 된다. 그림 8에서 각 클래스별 거리의 분산을 구해보면 클래스1의 분산은  $\sigma_{x1}$ 으로서 3.2이고 Y좌표축상  $\sigma_{y1}$  은 4.5이므로 평면분산값에 대한 거리계산값은 식 (1),(2),(1)',(2)'와 같다.

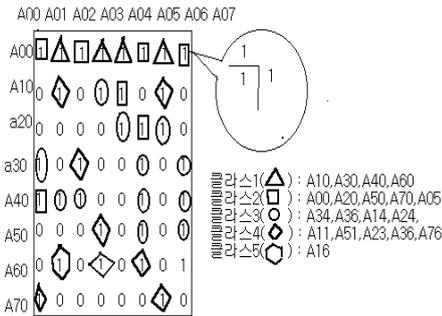


그림 9. 화면의 닮음꼴을 위한 중복도  
Fig. 9. Image resemblance of overlap digram

클래스1의 x좌표축 분산값 =

$$\sum_{i=1}^n (x\text{의좌표}-\text{클래스1의}x\text{좌표})^2 \quad (1)$$

클래스1의 y좌표축 분산값 =

$$\sum_{i=1}^n (y\text{의 좌표}-\text{클래스1의 }y\text{좌표})^2 \quad (2)$$

$$\sigma_{class1x} = \frac{1}{n} \sum_{i=1}^n (x_i - x_1)^2 \quad (1')$$

$$\sigma_{class1y} = \frac{1}{n} \sum_{i=1}^n (Y_i - Y_1)^2 \quad (2')$$

값을 구하면 X좌표값은 3.2이고 직선상의 임의의 점에서 해당 클래스의 첫 번째 원소와의 가장 가까운 거리를 계산 하게 되면 해당 클래스의 일정 화면내의 벡터 속도값이 나오게 된다. 이 벡터 속도값은 일정 화면내의 같은 닮은 화소군간의 변화 속도값을 나타내므로 값이 작을수록 화면내의 픽셀의 변화가 적다는 뜻이 된다. 같은 클래스내의 원소들간의 관계값은 분산값을 통하여 구한다. 다른 계층들간의 관계값은 그림9를 통해 구해볼수 있다.

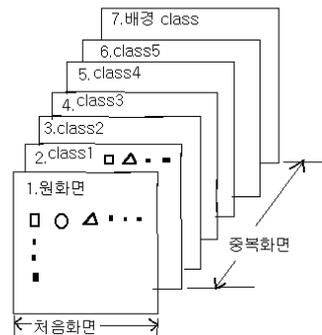


그림 10. 화면의 닮음꼴을 위한 중복  
Fig. 10. Overlap of image resemblance

즉, 화면에서 변화되어지는 원소들간의 변화값을 구해보면 픽셀들의 이전 화면과의 변화 모양군을 픽셀군으로 분리하여 해석한 결과 class1에서부터 class5까지의 5개의 등급과 나머지 배경 등급으로 표시하였으며 등급간의 관계는 각 층을 분리하여 원화면을 화면 처리하여 보고 그때 매우 다른 화면들 간에도 많은 유사한 점이 많음을 발견하게 된다 처음화면은 원본 화면을 나타내고 중복 화면은 다음 화면을 중첩하여 나타낸 화면으로 6개의 중

복 화면으로 구성되어 있다.

## V. 모의 실험 및 결과

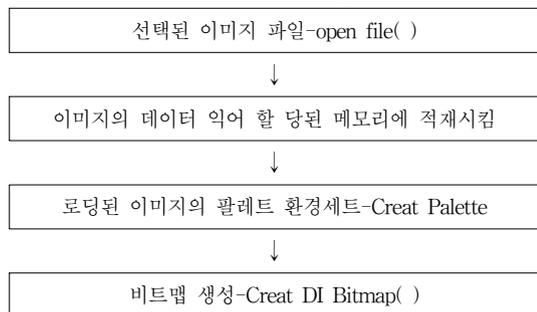
### 1. 제작과정

이미지 출력 프로그램의 설계를 위해 윈도우 프로그램의 IMAGE Explore를 사용하여 256x256픽셀의 화면을 구성하였다.

윈도우즈 프로그램 컴파일에는 같은 회사의 'c'컴파일러인 MSC 6.0이후의 버전을 쓰는데 라이브러리,헤더파일, 프로그래밍 레퍼런스를 사용하여 화면 키트를 제작하였다. 여기서 윈도우즈 프로그램은 파일로서 크게 'c' 리소스 파일들과 모듈정의 파일로 구성이 되고 부가적으로 헤더파일, 아이콘 파일 등이 부가적으로 속해져 있는 256그래픽 모드의 DOS VGA상의 비디오 메모리에 비트맵 데이터가 차례로 적재되면 디스 플레이에는 해당 비트맵 이미지가 출력된다. 이러한 이미지를 디스플레이 상에서 스크롤 및 좌우 이동을 시키기 위해서 비트맵 이미지 데이터가 그만큼의 데이터를 갖는 윈도우즈에서 비트맵 이미지의 디맨전 뿐 아니라 이미지의 정보도 포함되어 있다.

표 1.외부프로그램 순서도

Table 1. Flawchart of out program



### 2. 영상 출력메모리 할당과 사용

크기가 큰 이미지를 데이터를 위하여 전역적 메모리를 할당시키는데 윈도우즈에서 메모리 구분은 로컬 메모리이므로 64kb의 제한을 받는 이미지 데이터로는 사용하지 않는다. 따라서 본 프로그램의 골격은 메인 모듈, 파일 선택모듈, bmp 이미지 로딩 모듈로 나뉘어지고 이모듈로 될 수 있는한 다른 모듈과의 관계에 있어서 상당하게 독립적으로 실행 되게끔 설계되어 추가 코딩이 되어진다.

화면출력결과 다음과 같은 bmp화면으로 구성하였다. 256×256칼라 이미지를 사용하여 이미지를 출력하였는데 다음과 같은 화면으로 구성되어졌다. 히스토그램으로 분리하여 분석하여 본 결과 샘플넘버에서 BMP화면과 새로운 디코딩 화면과의 출력의 차이가 생겼음을 알 수 있다. 화면의 출력의 차이가 생기는 것은 이미지 화면의 컬러디코딩의 차이가 생겨서 결과가 나타난 것이다. 일반DCT 변환알고리즘은 화면당 모두 중복되어지는 픽셀로 수치 변환을 하는 결과가 나타나지만 이미지 맑음꼴 변환은 일단중복이 되는 화면의 픽셀은 다음과 같은 방법으로 중복되어지는 것을 피하고 새로운 화면을 픽셀로 사용하기 때문에 처음 화면을 비교하는 작업만 제외하게 되면 화면의 수가 많아질수록 더욱 유리하다고 판단할 수가 있다.2개의 영상을 변형하는 가운데 첫 번 영상을 두 번째 영상까지 변형하는데 30프레임이 드는것을 20프레임이 드는것으로 image explore를 가지고 실험 하였다.

그림11의 양쪽 두그림의 이미지 중복도를 알아보는 실험은 이미지 플로어(image plore)를 통해 알아 본다.



그림 11. 맑음꼴 구현이미지  
Fig. 11. Resemblance realization

메모리 사이즈는 전송 팩터에서  $1 \times 10^6$ 으로 나누면 각 블록 사이즈별로 메모리 사이즈가 나오게 된다.

표 2. Block Coding 결과의 두영상간의 속도비교 및 메모리 사이즈비교

Table 2. Block Coding result value of compare velocity and memory size

Block size	Transform time factor	
	DCT decoding	resemblance decoding
64x64x4	17x10 <sup>6</sup>	14x10 <sup>6</sup>
32x32x4	1x10 <sup>6</sup>	0.7x10 <sup>6</sup>
16x16x4	65536	57327
8x8x4	4096	4013
4x4x4	256	256

그결과 블록의 수가 많을수록 즉 보내야할 화상의 크기 클수록 메모리 사이즈의 축소되는 면이 작다고 할 수가 있다. 따라서 많은수의 화상을 많은양의 화상크기로 전송할 경우가 가장 많음꼴 전송방법에서 유리하다고 볼수 있다. 다음은 각 방법별로 샘플링 히스토그램을 관찰한 결과가 그림12에 나타내었는데 블록사이즈에 관계없이 일정한 크기변화를 보이는 것이 많음꼴 디코딩 방법으로서 많은 블록 즉, 많은 화면일수록 샘플링수가 줄어듦을 볼수가 있다.

계산한 결과를 토대로 기존의 DCT 결과보다 약 14% 정도의 샘플링수가 줄어듦을 알수가 있고 메모리 사이즈로서는 25%의 줄어듦을 표 3을 통해 볼수가 있다.

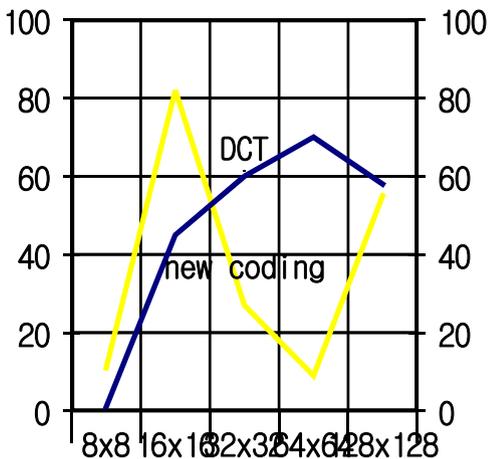


그림 12. 디코딩별 화면당 샘플링 수  
Fig. 12. No. of sampling decoding image

## VI. 결론

본 논문에서는 기존의 화면당 모든 화소들을 인정하여 보내야할 처음의 화면부터 마지막 화면까지 각 픽셀 하나 하나까지 트랜스 폼 하는 방식을 DCT 방식이라하여 정지화상 및 동영상에 이르기까지 사용하여왔으나 이 방법은 알려진 대로 화상의 존재를 인정하는 것이 아니고 즉 화상이 아닌 픽셀을 인정하는 방식이라하여 아무리 같은 화상이라 할지라도 모두 디코딩 하는 방법을 사용하였으나 이방법은 매우 반복되어져서 보내 저장할 화면에서는 비교할수 있는 방법이 없어 무조건 다 전송하는 원시적인 방법이라 할 수가 있다. 그러나 새로운 디코딩 방식은 한 픽셀을 설정하여 기존 화면으로 정한후 임계값으로 결정한 값을 비교하여 모든 화면의 픽셀들을 디코딩하는 방법이다. 화면의 특성상 중복되는 픽셀들을 임계값으로 결정하여 그 값 이하는 유사성 화면으로 대체되는 방법을 통하여 메모리의 절감 효과를 가져왔다. 원래의 DCT 방법보다 한주기내 표본화를 시켜야할 데이터의 14%정도의 삭제를 시도하여 데이터를 복원하여도 무방한 데이터 절감방법이다.

## 참고문헌

- [1] P.H.Westerink, D.E.Boekke, J.BiemondandJ.W.Woods, "Subband Coding of Image UsingVectorQuantization", IEEE Trans.Commun.,vol.36,pp.713-719. June 1988
- [2] P.A.Chau, T.Lookabaugh, and R.M.Gray, "Entropy-Constrained Vector Quantization", IEEE Trans. Acoust., SpeechandSignalProcess., vol.ASSP-37, pp.31-42,Jan.1989
- [3] A.Gersho, "AsymptoticallyOptimal Block Quantization," IEEE trans.Inform. Theory, vol. IT-27, pp.280-291, May 1981.
- [4] J.W.Modestino and Y.H.Kim, "Adaptive Entropy-Coded Predictive Vector Quanti -zation of Images," to appear in IEEE Trans. Acoust., Speech, and Signal Process., June 1992.
- [5] Y.H.Kimand J.W.Modestino, "Adaptive Entropy-CodedPruned ree-Structured Predictive Vector quantization of images," to appear in IEEE

Trans. on Commun.

- [6] J.B.Anderson, T.Aulin, and C.E.Sundber Digital Phase Modulation, Plenum Press, New York,1986
- [7] H.Musmann, "Object Oriented Analysis -Synthesis Coding of Moving Images", Signal Process: Image Communication,1(1),1998
- [8] R.C.Wood, "An optimum quantization," IEEE Trans.Inform.Theory, vol, IT-5, March 1999
- [9] I.F. Akyildiz, W. Su,Y. Sankarasu "A Digital Decoding Threshold Hold Value," IEEE Communication Magazine, vol.40, no.8, pp.102-114, Aug. 2002.
- [10] Y.Tian, K.Xu, and N.Ansari, "Tcp Wireless environment:Problems and Solution," IEEE Communication Magazine, vol. 70,no. 6, pp.12-14, Mar. 2005.

#### 저자 소개

##### 진 현 수(정회원)



- 1986년 2월 : 서울시립대학교 전자공학  
학과졸업
- 1993년 8월 : 서울시립대학교 전자공  
학과석사졸업
- 2000년 2월 : 서울시립대학교 전자공  
학과박사졸업
- 2001년 3월~현재 : 백석대학교 정보  
통신학부 근무

<주 관심분야: 퍼지제어, 웹 콘트롤, 인터넷통신>