

# 이산여현변환을 이용한 이미지 압축 알고리즘 원리에 관한 연구

남수태\* · 김도관\* · 진찬용\* · 신성윤\*\*

\*원광대학교(융복합창의연구소), \*\*군산대학교

## Understanding on the Principle of Image Compression Algorithm Using on the DCT (discrete cosine transform)

Soo-tai Nam\* · Do-goan Kim\* · Chan-yong Jin\* · Seong-yoon Shin\*\*

\*Wonkwang University (Institute of Convergence and Creativity), \*\*Kunsan National University

E-mail : stnam@wku.ac.kr

### 요 약

이미지 압축은 디지털 이미지를 데이터 압축을 적용하는 과정이다. 이산여현변환은 시간 도메인에서 주파수 도메인으로 변환하는 기술이며, 이미지 압축에 널리 사용된다. 먼저, 이미지는 8x8 픽셀 블록으로 분리한다. 위에서 아래로 왼쪽에서 오른쪽으로 처리하면서 DCT 각 블록에 적용한다. 각각의 블록은 양자화를 통해 압축된다. 이미지를 구성하는 압축된 블록 배열의 공간은 크게 줄어든다. IDCT를 통해 이미지를 재구성 한다. 본 연구에서는 이산여현변환 기법을 이용하여 이미지 압축/복원의 전 처리 과정을 조망하는 것이 목적이다.

### ABSTRACT

Image compression is the application of Data compression on digital images. The (DCT) discrete cosine transform is a technique for converting a time domain to a frequency domain. It is widely used in image compression. First, the image is divided into 8x8 pixel blocks. Apply the DCT to each block while processing from top to bottom from left to right. Each block is compressed through quantization. The space of the compressed block array constituting the image is greatly reduced. Reconstruct the image through the IDCT. The purpose of this research is to understand compression/decompression of images using the DCT method.

### 키워드

데이터압축, 알고리즘, 이미지, 이산여현변환, 코사인

### 1. 소 개

데이터 압축은 데이터 저장 요구사항 및 그에 따른 통신비용을 줄이기 위해 데이터 표현의 중복을 줄이는 기술이다. 저장소 요구사항을 줄이는 것은 저장소 미디어의 용량을 늘리고 따라서 통신 대역폭을 늘리는 것과 같다. 따라서 효율적인 압축 기술의 개발은 미래의 통신 시스템 및 첨단 멀티미디어 어플리케이션을 위한 설계과제가 될 것이다. 데이터는 정보와 중복성의 조합으로 표현된다. 정보는 데이터의 의미 또는 목적을 정확하게 해석하기 위해 원래 형식으로 영구 보존해야 하는 부분이다. 예를 들어 수십 또는 수백 개의 이미지를 사용하는 웹 페이지나 온라인 카탈로그를 가진 사람은 이미지를 저장하기 위해 이미지

압축을 사용해야 할 가능성이 매우 높다고 볼 수 있다. 이는 순수한 이미지를 유지하는데 필요한 공간이 비용 면에서 엄청나게 클 수 있기 때문이다. 다행히도 현재 사용할 수 있는 여러 가지 이미지 압축 방법이 있다. 이러한 방법으로 크게 두 가지 범주로 나누고 있으며 무손실 및 손실 이미지 압축이다. JPEG 프로세스는 DCT 중심으로 다소 손실이 많이 발생하는 이미지 압축 방식으로 널리 사용되고 있다. 또한 양자화 단계에서 압축으로 인해 픽셀 일부 손실이 발생하지만 크게 중요하지 않은 주파수가 삭제되므로 “손실”이라는 용어를 사용한다. DCT 프로세스는 이미지의 차별되는 주파수 성분만 분리하여 수행한다. 그런데 남아있는 가장 중요한 주파수 성분만 압축 해제 과정에서 이미지 재구성하는데 사용된다. 결과적

으로 재구성된 이미지는 약간의 왜곡이 포함된다. 그러나 곧 알게 되겠지만 압축 단계에서 이러한 레벨의 왜곡을 조정할 수 있다[1]. JPEG 방식은 컬러 및 흑백 이미지 모두에 사용되지만 연구의 초점은 후자의 압축에 관한 것이다.

## II. 연구방법

다음은 JPEG 프로세스의 일반적인 개요이다. JPEG 처리 방법을 자세히 살펴봄으로써 프로세스에 대한 보다 포괄적인 이해를 얻을 수 있다. 본 연구의 서술 방법은 다음과 같다.

1. 이미지는 8x8 픽셀 블록으로 분리한다.
2. 위에서 아래로 왼쪽에서 오른쪽으로 처리하면서 DCT 각 블록에 적용한다.
3. 각각의 블록은 양자화를 통해 압축된다.
4. 이미지를 구성하는 압축된 블록 배열의 공간은 크게 줄어든다.
5. IDCT를 통해 이미지를 재구성한다.

아래의 식 (1)과 식 (2)는 이산여현변환을 위한 이미지 (I, j) 번째 픽셀을 계산한다[2].

$$D(i, j) = \frac{1}{\sqrt{2N}} C(i)C(j) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} p(x, y) \cos\left[\frac{(2x+1)i\pi}{2N}\right] \cos\left[\frac{(2y+1)j\pi}{2N}\right] \quad 1$$

$$C(u) = \begin{cases} \frac{1}{\sqrt{2}} & \text{if } u = 0 \\ 1 & \text{if } u > 0 \end{cases} \quad 2$$

$p(x, y)$ 는 행렬  $p$ 가 나타내는 이미지의  $x, y$  번째 요소이다.  $N$ 은 DCT가 수행되는 블록의 크기이고 방정식은 원본 이미지 행렬의 픽셀 값에서 변환 이미지의 한 픽셀( $i, j$ )을 계산한다. JPEG 압축이 사용하는 표준 8x8 블록의 경우,  $N$ 은 8이고  $x$ 와  $y$ 의 범위는 0에서 7까지이다. 따라서  $D(i, j)$ 는 식 (3)과 같다.

$$D(i, j) = \frac{1}{4} C(i)C(j) \sum_{x=0}^7 \sum_{y=0}^7 p(x, y) \cos\left[\frac{(2x+1)i\pi}{16}\right] \cos\left[\frac{(2y+1)j\pi}{16}\right] \quad 3$$

DCT는 코사인 함수를 사용하기 때문에 결과 행렬은 수평, 대각선 및 수직 주파수에 의존한다. 따라서 주파수의 변화가 많은 검은색 이미지는 매우 랜덤한 결과의 행렬을 가진다. 하나의 색의 이미지 행렬은 첫 번째 픽셀에 대해 큰 값의 결과 행렬을 가지며 다른 픽셀에 대해서는 0이다. 다음은 식 (1)을 통해 아래의 행렬 형태를 얻으려면 다음 식 (4)를 사용한다. 그리고 8x8 블록의 행렬은 아래 Matrix [T]와 같다.

$$T_{ij} = \begin{cases} \frac{1}{\sqrt{8}} & \text{if } i = 0 \\ \sqrt{\frac{2}{8}} \cos\left[\frac{(2i+1)j\pi}{16}\right] & \text{if } i > 0 \end{cases} \quad 4$$

행렬의 첫 번째 행( $i = 1$ )은 식 (4)에서 예상 한

대로 모든 행이  $1/\sqrt{8}$ 의 값을 가지게 된다. 다음으로 [T]의 열은 직교 정규 집합을 형성하므로 [T]는 직교 행렬이다. 또한 역 DCT를 할 때는 [T]의 역함수는 계산하기 쉬운 [T]이므로 [T]의 직교성이 중요하다. 먼저 8x8 블록을 DCT 수행하기 전에 흑백 이미지 픽셀의 값은 0에서 255까지 1 단계로 표시되어야 한다. 여기서 순수 검정은 0으로 표시되고 순수 흰색은 255의 값을 가지게 된다. 따라서 이러한 256가지 음영으로 사진, 일러스트레이션 등을 정확하게 표현할 수 있다. 이미지가 수백 또는 수천 개의 8x8 픽셀 블록으로 구성되기 때문에 하나의 8x8 블록에 대한 설명은 JPEG 프로세스의 축소판이라고 말할 수 있다. 이러한 각각의 블록은 이미지의 왼쪽 코너 상단으로 부터 선택된다.

표 1. Matrix [T].

0.3536	0.3536	0.3536	0.3536	0.3536	0.3536	0.3536	0.3536
0.4904	0.4157	0.2778	0.0975	-0.0975	-0.2778	-0.4157	-0.4904
0.4619	0.1913	-0.1913	-0.4619	-0.4619	-0.1913	0.1913	0.4619
0.4157	-0.0975	-0.4904	-0.2778	0.2778	0.4904	0.0975	-0.4157
0.3536	-0.3536	-0.3536	0.3536	0.3536	-0.3536	-0.3536	0.3536
0.2778	-0.4904	0.0975	0.4157	-0.4157	-0.0975	0.4904	-0.2778
0.1913	-0.4619	0.4619	-0.1913	-0.1913	0.4619	-0.4619	0.1913
0.0975	-0.2778	0.4157	-0.4904	0.4904	-0.4157	0.2778	-0.0975

표 2. Original Image [O].

154	123	123	123	123	123	123	136
192	180	136	154	154	154	136	110
254	198	154	154	180	154	123	123
239	180	136	180	180	166	123	123
180	154	136	167	166	149	136	136
128	136	123	136	154	180	198	154
123	105	110	149	136	136	180	166
110	136	123	123	123	136	154	136

DCT는 -128에서 127까지의 픽셀 값에서 연동되도록 설계되었으므로 원본 이미지 블록은 각 항목에서 128을 빼서 "평평하게"된 결과를 볼 수 있고 결과는 표 3, Matrix [M]과 같다.

표 3. Matrix [M].

26	-5	-5	-5	-5	-5	-5	8
64	52	8	26	26	26	8	-18
126	70	26	26	52	26	-5	-5
111	52	8	52	52	38	-5	-5
52	26	8	39	38	21	8	8
0	8	-5	8	26	52	70	26
-5	-23	-18	21	8	8	52	38
-18	8	-5	-5	-5	8	26	8

### III. 결 론

이제 행렬 곱셈에 의해 수행되는 이산여현변환을 수행할 준비가 모두 되었다.

$$D = TMT' 5$$

다음으로 식 (5)에서 행렬  $[M]$ 은 먼저 이전 섹션의 DCT 행렬  $[T]$ 에 의해 좌측에서 승산하고 이것은 행을 변환한다. 그런 다음 DCT 행렬의 전치 (transpose)에 의해 오른쪽으로 곱해 열을 변환한다. 그러면 다음 행렬이 산출된다. 이러한 방법으로 전체 행렬을 값을 산출한다.

표 4. Matrix  $[D]$ .

162.3	40.6	20.0	72.3	30.3	12.5	-19.7	-11.5
30.5	108.4	10.5	32.3	27.7	-15.5	18.4	-2.0
-94.1	-60.1	12.3	-43.4	-31.3	6.1	-3.3	7.1
-38.6	-83.4	-5.4	-22.2	-13.5	15.5	-1.3	3.5
-31.3	17.9	-5.5	-12.4	14.3	-6.0	11.5	-6.0
-0.9	-11.8	12.8	0.2	28.1	12.6	8.4	2.9
4.6	-2.4	12.2	6.6	-18.7	-12.8	7.7	12.0
-10.0	11.2	7.8	-16.3	21.5	0.0	5.9	10.7

이 블록 행렬은 64 DCT 계수  $C_{ij}$ 로 구성되며 여기서  $i$ 와  $j$ 의 범위는 0에서 7까지이다. 왼쪽 상단 계수  $C_{00}$ 에 위치하는 성분의 값은 원본 이미지 블록의 낮은 주파수와 관련이 있다. 이외 모든 방향에서  $C_{00}$ 에서 멀어지면 DCT 계수는 이미지 블록에서 더 높은 주파수와 관련이 있고 여기서  $C_{77}$ 에 위치하는 성분의 값은 가장 높은 주파수에 해당한다. 인간의 눈은 저주파에 가장 민감하며, 양자화 단계의 결과는 이 사실을 반영한다는 점에 유의해야 한다. 다음으로는 양자화 프로세스이다.  $8 \times 8$  DCT 계수 블록은 이제 양자화 처리를 위한 압축을 준비하게 된다. JPEG 프로세스의 주목할 만한 매우 유용한 기능은 이 단계에서 특정 양자화 행렬을 선택하여 다양한 수준의 이미지 압축 및 품질을 얻을 수 있다는 것이다. 이를 통해 사용자는 1에서 100 사이의 품질 수준을 결정할 수 있다. 여기서 1은 가장 낮은 화질과 가장 높은 압축률을 제공하고 100은 최고 품질과 최저 압축률을 제공한다. 결과적으로 품질/압축 비율은 다양한 요구에 맞게 조정할 수 있다. 인간의 시각 특성과 관련된 주관적인 실험 결과 JPEG 표준 양자화 매트릭스를 제시하고 있다. 양자화 수준이 50인 매트릭스는 압축률이 높고 압축 해제된 이미지 품질이 뛰어난 것을 알 수 있다. 만약 다른 수준의 품질이나 압축이 요구된다면, JPEG 표준 양자화 매트릭스의 스칼라 배수가 사용될 수 있다. 50수준 보다 높은 품질 수준의 경우 표준 양자화 행렬에/50을 곱하고 50수준 보다 낮은 품질 수준의 경우 표준 양자화 행렬에 50/품질 수

준이 곱해진다. 다음으로 스케일한 양자화 행렬은 1에서 255 사이의 양의 정수 값을 가지도록 반올림한다. 예를 들어 다음의 양자화 행렬은 10수준 및 90의 품질 수준을 산출하고자 한다. 양자화는 변환된 이미지 행렬  $[D]$ 의 각 요소를 양자화 행렬의 해당 요소로 나눈 다음 가장 가까운 정수 값으로 반올림하여 수행된다. 다음 단계에서 양자화 행렬  $Q_{50}$ 이 사용된다. 왼쪽 상단 모서리 근처에 있는 계수는 이미지 블록은 인간의 눈에 가장 민감한 낮은 주파수에 해당한다. 또한 0의 값은 중요성을 작게 나타내며, 더 높은 주파수는 버려지고 압축의 손실 부분을 발생시킨다. 앞서 언급한 것처럼 나머지 0이 아닌 계수만 이미지를 재구성하는데 사용된다. 뿐만 아니라 다른 양자화 행렬의 효과에 주목하는 것도 흥미로울 수 있다.

표 5. Matrix  $[Q_{50}]$ .

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

표 6. Matrix  $[Q_{10}]$ .

80	60	50	80	120	200	255	255
55	60	70	95	130	255	255	255
70	65	80	120	200	255	255	255
70	85	110	145	255	255	255	255
90	110	185	255	255	255	255	255
120	175	255	255	255	255	255	255
245	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255

표 7. Matrix  $[Q_{90}]$ .

3	2	2	3	5	8	10	12
2	2	3	4	5	12	12	11
3	3	3	5	8	11	14	11
3	3	4	6	10	17	16	12
4	4	7	11	14	22	21	15
5	7	11	13	16	12	23	18
10	13	16	17	21	24	24	21
14	18	10	20	22	20	20	20

양자화 매트릭스  $Q_{10}$ 을 사용하면 0의 값을 더욱더 많이 나타나게 되고 양자화 매트릭스  $Q_{90}$ 을 사용하면 0의 값은 거의 나타나지 않은 결과를 얻을 수 있다.

표 8. Matrix [C].

10	4	2	5	1	0	0	0
3	9	1	2	1	0	0	0
-7	-5	1	-2	-1	0	0	0
-3	-5	0	-1	0	0	0	0
-2	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

$$C_{i,j} = \text{round}\left(\frac{D_{i,j}}{Q_{i,j}}\right) \quad 6$$

그러면 양자화를 마친 매트릭스 [C]는 압축의 최종 단계를 위해 준비가 되었다. 저장하기 전에 [C] 매트릭스의 모든 계수는 인코더를 하기 위해 이진 데이터 스트림 (01101011 ...) 형태로 변환된다. 코딩 프로세스에 대한 자세한 내용은 본 논문의 범위를 벗어남을 알 수 있다. 그러나 저자는 독자가 알아야 할 중요한 측면을 지적 할 수 있다. 양자화 프로세스 작업 이후에는 대부분의 계수가 0이 되는 것이 일반적이다. JPEG 알고리즘은 그림 1에 표시된 Zig-zag 시퀀스 양자화 완료된 계수를 인코딩하여 이를 활용한다.

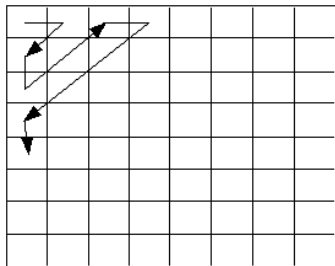


그림 1. 지그재그 스캐닝.

$$R_{i,j} = Q_{i,j} \times C_{i,j} \quad 7$$

표 9. Matrix [R].

160	44	20	80	24	0	0	0
36	108	14	38	26	0	0	0
-98	-65	16	-48	-40	0	0	0
-42	-85	0	-29	0	0	0	0
-36	22	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

다음으로 압축된 이미지 파일을 복원하는 프로세스이다. 이미지의 재구성은 양자화 처리된 매트릭스 [C]를 나타내는 비트 스트림을 디코딩 함으로써 시작한다. 매트릭스 [C]의 각각의 픽셀 값은

원래 사용된 양자화 매트릭스의 대응하는 픽셀 값과 곱하여 행해진다. 다음으로 IDCT는 가장 가까운 정수로 반올림되는 매트릭스 [R]에 적용된다. 마지막으로 변환된 각 픽셀 값에 128을 더하게 된다. 아래의 식 (8)을 수행하면 압축이 해제된 픽셀 값을 얻을 수 있다.

$$N = \text{round}(T' R T) + 128 \quad 8$$

결과적으로 DCT 계수는 거의 70%가 이미지 블록 압축/복원하기 전에 소실된다는 점을 고려하면 놀라운 결과임에는 틀림없다. 이미지 전체를 구성하는 나머지 블록에서도 비슷한 결과가 발생할 것이므로 JPEG 이미지는 원래 이미지와 거의 구별할 수 없다는 것을 알 수 있다. 따라서 흑백 이미지는 256가지 음영을 표현할 수 있다는 것을 기억하십시오. 아래 표 10과 표 11은 압축 전의 원래 이미지 픽셀 값과 복원된 이미지 픽셀 값을 비교할 수 있다.

표 10. Original.

154	123	123	123	123	123	123	136
192	180	136	154	154	154	136	110
254	198	154	154	180	154	123	123
239	180	136	180	180	166	123	123
180	154	136	167	166	149	136	136
128	136	123	136	154	180	198	154
123	105	110	149	136	136	180	166
110	136	123	123	123	136	154	136

표 11. Decompressed.

149	134	119	116	121	126	127	128
204	168	140	144	155	150	135	125
253	195	155	166	183	165	131	111
245	185	148	166	184	160	124	107
188	149	132	155	172	159	141	136
132	123	125	143	160	166	168	171
109	119	126	128	139	158	168	166
111	127	127	114	118	141	147	135

## 참고문헌

- [1] A. J. Maan, "An Introduction to JPEG Image Compression Algorithm," International Journal of Electrical, Electronics and Data Communication, vol. 1, no. 10, pp. 44-46, Dec 2013.
- [2] Y. Devi, "JPEG Image Compression Using Various Algorithms: A Review," International Journal of Computer Science Trends and Technology, vol. 4, no. 3 pp. 89-92, Jun 2016.