

픽셀의 유사정보를 이용한 이미지 압축 모델

박인영*, 위영철

*아주대학교 정보통신공학과

e-mail : parkiy7@ajou.ac.kr

A Model of Image Compression Using Pixel Connection Information

In-Young Park*, Young-Cheoul Wee

*Dept of GSIC, Ajou University

요 약

본 논문에서는 주어진 이미지를 균일하게 블록화 시킨 후 각각의 Range 블록에서 Domain 블록을 찾아나간다. 각 Range 블록마다 Domain 블록이 찾아지게 되면 픽셀별로 포인팅을 하게 한다. 이러한 포인팅을 이용하여 링크드 리스트를 만들어 나가게 되면 사이클을 포함하게 되는 링크드 리스트가 조인된 형태의 모습을 갖게 된다. 그러면 사이클을 포함하는 이러한 형태의 픽셀의 평균값을 계산하여 뿌려주게 되는 것이다.

1. 서 론

본 논문에서는 이미지 압축을 하는 모델을 제시한다. 이미지를 블록으로 분할하여 Range, Domain 블록으로 나누는 것은 Fractal Image Compression[1][2][3]에 기초하고 있다. 일반적으로는 Domain 블록은 Range 블록의 면적이 4배가 되는 것으로 하게 된다. 하지만 여기서는 Range와 Domain의 크기는 같도록 설정한다. 왜냐하면 각각으로 픽셀로서 매핑을 하려 하기 때문이다. Range블록은 Non-overlapping으로 모든 Range의 합은 입력 이미지를 나타낸다. Domain 블록은 Overlapping을 허용하게 된다.

Fractal Image Compression은 이미지 내에 자기유사성을 이용하여 Domain 블록의 위치, 영상 보정을 위한 s, o 값을 이용하여 이미지를 압축하는 방식이다.

같은 크기의 Range, Domain 블록의 이용하여, 먼저 가장 유사성이 높은 블록이 쌍이 찾아지게 되면, 픽셀별로 포인팅을 하게 된다. 그런 후에, 이러한 포인팅을 재귀적으로 적용을 하게 되면 사이클을 포함하게 되는 링크드 리스트가 조인된 형태의 모습을 갖게 된다. 그러면 사이클을 포함하는 이러한 형태

의 픽셀의 평균값을 계산하여 뿌려주게 되는 것이다.

다시 말하면 256×256 원래 이미지를 복원하기 위해서 사이클을 포함하는 링크드 리스트의 연결된 끝의 대표값(평균값)만으로써 이미지를 표현할 수 있게 되는 것이다.

2. 영상 압축의 개요

압축되지 않은 비디오 자료의 저장공간은 초당 23.6 메가바이트(512 화소 × 512 화소 × 화소당 3바이트 요구 × 초당 30프레임)를 필요로 한다. MPEG-1 압축 기법을 사용하면 full-motion 비디오 자료를 영상의 질적인 측면에서 조그만 손실을 감수하고 초당 187 킬로바이트로 압축할 수 있다.

압축되지 않은 비디오 자료는 초당 23.6 메가바이트의 저장 공간을 요구한다. 만약에 영화가 2시간 분량이라면, 저장공간으로 169 기가 바이트가 필요하게 된다. 시청자에게 초당 180 메가바이트를 송신할 수 있는 대용량 대역폭의 전화선이 설치되어 있으면, 비디오 점포에서 시청자 집으로 영화를 송신하는 데에 15.7분이 소요될 것이다. 송신시간은 영화

관람시간의 1/8에 해당하는 것이다.

영화가 MPEG-1 으로 압축되어 있다면, 저장 공간은 1.3 기가바이트로 줄어든 것이다. 높은 대역폭의 링크를 이용한다면, 송신시간은 7.48 초가 소요될 것이다.

이러한 단적인 예를 보더라도 영상압축은 필요하다. 이러한 사실은 영상압축만을 위해서 많은 수의 새로운 하드웨어와 소프트웨어 제품이 제공되는 것으로 증명되었다. 이것은 CompuServe가 그래픽 과일을 압축하기 위해서 GIF 파일 형식을 제안하는 이유를 보면 쉽게 알 수 있다. 고해상도로 구성되는 컴퓨터 그래픽스보다 높은 명암도(화소당 많은 비트)를 요구하는 영상처리 응용에서는, 영상압축의 필요성이 증가할 것이다. 의학적인 영상은 공간적인 해상도와 명암도의 양측면에서 자료가 증가되는 기본적인 예이다. 비록 인간들이 명암 영상을 보고자 할 때는 화소당 8비트 이상을 요구하지 않을지라도, 컴퓨터 비전은 훨씬 높은 명암도의 자료를 분석하여야 한다.

압축률은 자료압축에 관한 논의에서 일반적으로 제기되며, 압축률을 원시 자료의 크기를 압축된 자료의 크기로 나눈 결과값으로 단순하게 표현할 수 있다. 예를 들어, 1 메가바이트 영상을 100킬로바이트로 압축한 경우는 10의 압축비율로 달성된다.

$$\begin{aligned} \text{압축비율} &= \text{원시자료} / \text{압축된자료} \\ &= 1 \text{ 메가바이트} / 100 \text{ 킬로바이트} \\ &= 10.0[4] \end{aligned}$$

주어진 영상에 대한 압축률이 클수록 압축된 최종 영상의 크기는 보다 더 작아질 것이다.

영상압축의 기본적인 두 가지 유형으로는 무손실 압축과 손실 압축을 제시한다.

무손실 방법은 자료를 완벽하게 암호화하고 복원하며, 이 방법에서는 결과 영상이 정확하게 원영상과 일치된다. 처리상에서 자료의 손실이 없으므로 영상의 질이 저하되지 않는다. 손실 압축 방법은 중복되고 필요하지 않은 정보가 손실되는 것을 허용한다. 전형적으로 이 방법은 압축률과 영상 질적인 측면에서 트레이드 오프를 나타낸다. 엄청나게 작은 크기로 영상을 압축할 수 있지만, 압축한 결과 영상의 질이 저하될 것이다. 어느 경우이나 해당되는 것은 아니지만, 이 방법은 전형적으로 보다 더 복잡하고 많은 계산을 요구하게 된다.

손실 영상압축 방법은 인간의 눈으로 감지할 수 없는 자료는 영상에서 제거한다. 이 방법은 인간에게 보여주는 데 목적이 있는 영상을 처리하는 데 적당하다. 영상이 기계에 의해서 분석되어진다면, 이 방법은 부적절하다. 컴퓨터는 인간의 눈으로 감지할 수 없는 정보를 쉽게 간파할 수 있기 때문이다. 이 방법의 목표는 최종적으로 복원된 영상에서 시각적으로 감지할 수 있는 정보가 손실되지 않게 하는 것이다. 다행스럽게, 영상으로부터 제거된 정보는 인간의 눈으로 감지할 수 없다.

많은 사람들은 손실 영상압축을 사용하여 끝없이 영상의 질을 저하시키고 있다. 사람들이 실감할 수 없는 것은 질 저하의 대부분이 인지하기에는 아주 작기 때문이다. 대부분의 영상 연산은 자료 손실을 포함하고, 영상의 스케닝 또는 디지털이징도 손실을 야기한다. 화면상의 영상 재현 또는 하드 카피에 의한 인쇄도 손실을 야기한다. 목표는 눈에 띄지 않을 정도의 손실을 허용하는 것이다.

3. 이미지 압축 과정

3.1 초기 패킹

입력 이미지는 256×256의 Raw 이미지이다. 이미지를 4×4의 균일한 블록으로 나눈다. 이렇게 하게 되면 64×64개의 블록이 만들어지게 된다. 이러한 블록을 Range 블록이라고 한다. 각각의 모든 Range 블록의 합은 원 이미지가 된다.

여기서 자기유사성을 찾기 위한 작업이 필요하게 된다. Domain 블록은 Overlapping이 허용되는 블록으로 역시 4×4의 크기를 갖게 된다.

단, 여기서 Range블록과 Domain블록의 크기가 같기 때문에 Rnage블록과 Domain블록이 1/2까지만 겹치는 조건을 추가한다.

각각의 Range 블록마다 가장 유사성이 높은 Domain 블록을 찾아나가는 데는 여러 가지 거리가 쓰일 수가 있다. 여기서는 RMSE를 사용하기로 한다.

$$\sigma = \sqrt{\frac{1}{N} \sum_{n=1}^N (X_n - Y_n)^2}$$

주어진 식에서 N은 4×4 블록의 픽셀갯수가 된다. 즉 16개가 된다.

X_n 는 Range블록의 각 픽셀 Y_n 은 Domain블록의 각 픽셀이 된다.

각각의 Range블록마다 최대 1/2 까지 겹치지 않는 범위내에서 Domain블록을 찾게 되면, 각각의 픽셀 별로 Range블록에서 Domain블록을 포인팅하게 되는 초기 링크드 리스트를 만들어 낸다.

결국 256×256 개 만큼의 초기 링크드 리스트가 만들어지게 된다.

3.2 사이클을 포함하는 조인된 링크드 리스트 구성

각 픽셀마다 초기매핑을 이루어졌다면 이것을 재귀적으로 사용하여 링크드 리스트를 계속 만들어 나간다. 본 프로그램에서 사이클을 포함하는 조인된 링크드 리스트 구성은 3가지 경우로 나누어 생각을 하였다.

먼저 이러한 구조를 선언하기 위해 다음과 같은 클래스가 필요하게 된다.

```
class Linked_list
{
    private:
    int x;
    int y;
    int visit;
    int cycle;
    int group_id;
    class Linked_list *next;
};
```

표1. 필요한 클래스

x, y는 가리키게 되는 픽셀의 위치가 된다.

그리고 visit와 cycle이란 flag을 두어서 사이클을 포함하는 조인된 링크드 리스트를 구성해 나가도록 한다. visit는 단순한 방문여부를 나타내고 cycle은 현재의 픽셀위치가 사이클을 포함하는 링크드 리스트의 조인된 형태인지 아닌지 판별할 수 있도록 한다.

그리고 이러한 독립적인 사이클을 포함하는 조인된 링크드 리스트를 나타내게 위해 group_id를 두었다. 256×256 개의 픽셀을 운행하면서 사이클을 포함하는 조인된 링크드 리스트를 구성하기 위해서 다음과 같은 경우를 생각할 수가 있다.

Case 1:
이미 픽셀이 사이클을 포함하는 조인된 링크드 리스트에 포함된 경우

Case 2:
운행을 하면서 사이클을 만드는 경우

Case 3:
이미 만들어진 사이클을 포함하는 리스트에 조인되는 경우

표2. 사이클을 포함하는 조인된 링크드 리스트를 구성하는 경우

각각의 사이클을 포함하는 조인된 링크드 리스트는 group_id 로서 구분이 되어지게 되고, 이 값은 결국 총 사이클을 포함하는 조인된 링크드 리스트의 개수를 나타내게 되어 Compression Ratio를 나타내는데 이용되게 된다.

3.3 복원

각 group_id를 구성하는 픽셀마다 픽셀 갯수의 합과 이에 해당하는 그레이 값의 합을 구한다. 그리고 나서 group_id에 속하는 픽셀을 위에서 구한 픽셀 개수의 합과 그레이 값의 합을 이용하여 평균을 구한다. 각각 group_id에 구성하는 픽셀을 평균값으로 나타내게 한다.

4. 실험결과

입력이미지는 256*256 그레이 이미지이다.

사용한 PC는 CPU: P-IV 1.6GHz, Memory: 256M에서 수행하였다.



그림1. Cronk 원이미지(좌) 복원이미지(우)

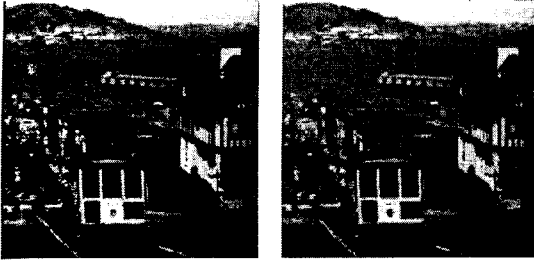


그림2. Cable 원이미지(좌) 복원이미지(우)

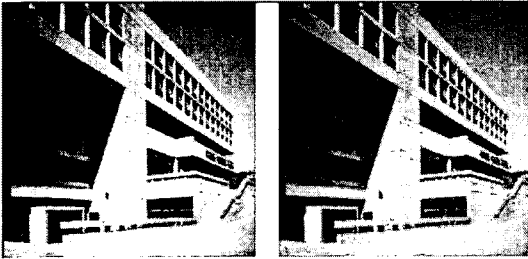


그림3. Bank 원이미지(좌) 복원이미지(우)



그림4. Lenna 원이미지(좌) 복원이미지(우)

이미지	PSNR	Time(sec)	compression ratio
Cronk	33.981070	198.396000	26.947368
Cable	30.349291	198.376000	23.173975
Bank	28.836614	198.696000	17.945235
Lenna	27.716877	198.696000	28.680963

표3. 실험결과

5. 결론 및 과제

블록을 분할 하는 개념과 가장 가까운 즉 유사성이 가장 높은 블록을 찾는다는 프랙탈 이미지 압축에 기초하였다.

각각의 Range 블록에 대응되는 Domain 블록에 대해 픽셀별로 매핑을 하고 난 후, 이것을 재귀적으로 이용하여 링크드 리스트를 만들어 나가게 된다. 이 링크드 리스트는 단방향이고 어디에서 출발을 하던 지 간에 사이클을 직접 이루게 되거나, 사이클이 이미 만들어진 리스트에 조인을 하게 되는 특성을 이용하여 이미지 압축 모델을 논문에서 제시해 보았다.

실제로 사이클을 포함하는 조인된 링크드 리스트의 좌표(위치)를 뽑아보게 되면, 독립된 개개의 것들에 포함되는 픽셀의 수가 천차만별임을 알 수 있다. 몇 천개의 픽셀이 모여서 하나의 독립적인 그룹을 이루는 반면 단지 몇 개의 픽셀이 모여서 이루는 경우도 있다. 적은 수의 픽셀로서 이루어진 독립된 그룹을 다른 그룹에 어떻게 적용시키느냐가 압축률을 높일 수 있으리라 생각이 되고, 과제로서 생각되어진다.

참고문헌

- [1] Khalid Sayood, "Introduction to Data Compression", pp517-526, 1996
- [2] Mark Nelson, Jean-Loup Gailly, "The Data Compression Book", pp.457-458, 1996
- [3] Yuval Fisher, "Fractal Image Compression", Siggraph '92 course notes, p14
- [4] Randy Crane, "Simplified Approach to Image Processing", p246, 1997