

영상처리 9주차 과제

학번: 201502049

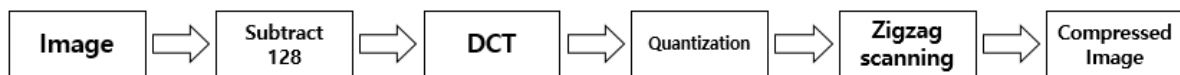
이름: 노효근

1. 과제 내용

:: 구현한 과제에 대한 설명, 어떤 방식으로 접근해야 하는지

Image를 주파수로 처리하고, 주변 값은 주파수가 비슷한 값인 것을 이용한다. 이 때 주파수 차이는 양자화를 통해 한다. 압축된 파일을 Decoding하는 것은 반대의 과정을 거친다.

아래의 그림과 같은 순서로 이미지를 압축하고 반대로 이미지를 압축해제해본다.



2. 구현한 방법에 대한 이유

:: 구현한 방법에 대한 설명 (왜 이렇게 구현했는지 자세히)

1) Encoding

- encoding의 순서와 맞게 이미지를 double형으로 바꾼 후, 이미지에서 128를 뺀다. 과제 요구사항에 맞게 8의 배수에 맞게 이미지를 가져온다 이때 8*8로 이미지를 나누어야 하므로 사이즈가 맞지 않게 된다면, 미리 패딩을 실시하여 이미지의 사이즈를 맞추게 된다. 이후 DCT계산식을 적용하도록 한다. 계산식 이후 주어진, 양자화 행렬을 이용하여, 이미지를 양자화하도록 한다. 양자화를 통한 이미지에서 지그재그를 통해 값을 읽어온다. 이 때, flag변수를 통해 0이 나오는 곳을 저장한다. count변수를 이용해 뒤 0값을 찾도록 한다. Tail 변수를 이용하여, 마지막 값에 "999"를 저장하여 flag가 있는 곳까지 유효한 값을 리스트에 저장하도록 한다. 리스트에 저장하여 decoding에 쓸 수 있도록 한다. 이로써 압축의 과정을 마친다.

2) Decoding

- Encoding의 순서와 반대로 역순으로 decoding을 진행한다. Check변수를 이용하여 0이 저장된 값의 위치를 확인한다. 이후 temp_block으로 지그재그 함수를 실행하며 각 8*8블록의 내용을 저장한다. 다시 양자화를 실행하고 DCT계산식을 적용하여 이전 이미지의 모습으로 복원을 하도록 한다. 이전 -128한 값을 128을 더하여 압축 전 이미지로 만든다.

3) 느낀 점

:: 구현하면서 느낀 점, 어려웠던 점, 혹은 설명이 필요하다고 느낀 부분

이미지 압축과 압축을 푸는 과정과 알고리즘의 이해는 어렵지 않았지만 실질적으로 코딩으로 해당 내용을 구현하려고 하니 많이 어려웠다. 특히 지그재그 함수는 생각하고 고려할 상황이 많았다. 다른 학생들의 코드를 참고하여 이번 과제를 실시했습니다. 너무 어려워요. 학기가 다 지나가는데 아직도 matlab이 익숙하지가 않아요. 죽겠어요...