



저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)

공학박사 학위 논문

Sub-pixel gradient 를 활용한  
compound 영상 압축

Compound image compression using sub-pixel gradient

2014년 2월

서울대학교 대학원

전기·정보공학부

김 규 동

Sub-pixel gradient를 활용한  
compound 영상 압축

Compound image compression using sub-pixel gradient

지도교수 김수환  
이 논문을 공학박사 학위논문으로 제출함  
2014년 2월

서울대학교 대학원  
전기·정보공학부  
김 규 동

의 공학박사 학위 논문을 인준함  
2014년 2월

위 원 장:           채 수 익           (인)  
부위원장:           김 수 환           (인)  
위      원:           조 남 익           (인)  
위      원:           이 혁 재           (인)  
위      원:           김 진 성           (인)

# 초 록

컴퓨터 성능과 네트워크 속도가 발전함에 따라 컴퓨터 화면에 표시되는 compound image 의 기술은 다양한 전송 환경에서 비디오 및 양방향 서비스가 가능해졌다. 그러나 compound image는 다양한 종류의 영상이 복합적으로 나타나기 때문에 영상의 종류를 명확히 구분하고 각 종류에 맞는 영상 데이터 처리 방식이 필요하게 된다. 영상의 데이터 처리 방식이 복잡해 질수록 서버와 클라이언트의 성능 불균형은 데이터를 원활히 생성/재현 하지 못하는 문제를 가질 수 있다.

Compound image 의 분류는 텍스트로 구성된 부분에 대하여 다른 종류의 영상으로 분류하지 않아야 한다. 이는 블록 단위로 구분하여 분류하는 방법에서 인접한 블록간에 서로 다른 코딩 방법을 적용하게 되면 사람이 느끼는 영상의 화질은 낮아지게 된다. 본 연구에서는 이러한 문제점을 해결하기 위해서 텍스트의 생성과정을 역이용한 sub-pixel gradient 블록 분류 방법을 제시한다. 평판 디스플레이에서는 텍스트의 부드러움을 표현하기 위하여 sub-pixel 단위로 컬러의 변화량을 조절하게 된다. 이를 whole-pixel 의 단위로 영상을 구분하게 되면, 텍스트의 영역을 명확하게 구분하지 못한다. 본 연구에는 sub-pixel gradient 블록 분류 방법을 통하여 텍스트로 구성된 영역과 텍스트가 아닌 영역에 대한 판단이 정확히 이루어짐을 실험을 통하여 확인하였다.

텍스트의 코딩방법 중 손실 압축방법은 텍스트로 구성된 영상이 높은

주파수를 가지는 영상이기 때문에 양자화나 변환과정을 거치게 되면 영상의 손실이 커지게 된다. 하지만 무 손실 압축 방법은 높은 데이터 량을 가지게 되고, 영상 전송 속도가 높아져야 하는 문제가 발생한다. 본 연구에서는 sub-pixel gradient 방법을 이용한 텍스트 영역에 대한 코딩 방법을 제시한다. 텍스트 영상이 가지는 특성을 이용하여 영상에서 발생하는 기울기에 대하여 코딩을 진행한다. 이를 통하여 영상의 손실을 줄이고 텍스트의 가독성을 높일 수 있다. 동일한 압축률에서 다른 압축 알고리즘에 비하여 텍스트의 화질과 가독성이 뛰어난을 확인하였다.

Compound image는 자연 영상과는 다르게 움직임이 단순하고 노이즈가 없다는 특성을 가진다. 이는 기존의 움직임 추정방법에 비하여 복잡도가 낮은 방법을 가능하게 한다. 본 연구에서는 이러한 compound image의 영상 특성을 이용한 그룹 움직임 추정 방법을 제시한다. 픽셀의 움직임을 확인하기 전에 영상의 분류에 따라 분류된 영역의 움직임을 먼저 파악하고 이를 통하여 최종적인 움직임을 추정하게 된다. 그룹 움직임 추정 방법을 사용하면 기존의 탐색영역 방법과 비교하여 탐색 영역을 최소화 할 수 있으며, 복잡도를 낮출 수 있음을 실험을 통하여 확인하였다.

**주요어:** sub-pixel gradient, text coding, sub-pixel 기울기, 그룹 움직임 추정, 텍스트 압축

**학번:** 2009-30179

# 차 례

초 록 .....	i
차 례 .....	iii
그림 목차 .....	vi
표 목 차 .....	ix
제1장 서 론 .....	1
1.1 연구 배경 .....	1
1.2 연구 내용 .....	4
1.3 논문 구성 .....	6
제2장 텍스트 생성과정 및 기존압축방법 .....	7
2.1 텍스트 생성과정 .....	7
2.2 표준 영상 압축 방법 .....	14
2.3 H.264 inter prediction .....	16
2.4 Compound image 의 압축 알고리즘 .....	19

제3장 Sub-pixel gradient 블록 분류 방법.....	23
3.1 Background & Text color extraction .....	28
3.2 Text De-colorization .....	32
3.3 블록 분류 실험 결과 .....	38
제4장 Sub-pixel Gradient text 블록 코딩 방법 .....	46
4.1 Gradient fitting process.....	51
4.2 Text Coding.....	56
4.2.1 Gradient로 구성된 부분의 코딩방법.....	56
4.2.2 Gradient가 없는 부분의 코딩방법.....	57
4.2.3 local min/max 값 예측.....	57
4.2.4 Whole-pixel 코딩.....	59
4.2.5 화질 enhancement.....	60
4.3 텍스트 코딩 동작 .....	64
4.3.1 텍스트 코딩 입력.....	65
4.3.2 Whole-pixel 코딩 1.....	66
4.3.3 역방향 Sub-pixel gradient 코딩 1 .....	67
4.3.4 Local minimum 코딩 1 .....	69
4.3.5 순방향 gradient 코딩 1.....	70

4.3.6	Local maximum 코딩 1.....	71
4.3.7	역방향 gradient 코딩 2.....	72
4.3.8	Local minimum 코딩 2.....	73
4.3.9	순방향 gradient 코딩 2.....	74
4.3.10	Whole-pixel 코딩 2.....	75
4.4	텍스트 블록 코딩 실험 결과 .....	77
제5장 그룹 움직임 추정 방법 .....		88
5.1	Block Grouping .....	944
5.2	Group Matching.....	977
5.3	Group motion vector calculation .....	1011
5.4	그룹 움직임 추정 방법 실험 결과 .....	1044
제6장 결 론 .....		1099
참 고 문 헌 .....		1122
Abstract .....		119



# 그림 목차

그림 2-1 ‘츠’ 텍스트의 sub-pixel 변환과정.....	8
그림 2-2 ‘u’ sub-pixel fringing 현상.....	9
그림 2-3 sub-pixel 에너지 분산방법.....	11
그림 2-4 필터링 전/후 sub-pixel rendering 텍스트 .....	13
그림 2-5 Inter prediction 모드 종류.....	17
그림 3-1 compound image 종류.....	24
그림 3-2 sub-pixel color 분포도.....	25
그림 3-3 whole-pixel color 분포도 .....	26
그림 3-4 Sub-pixel gradient를 이용한 블록 분류 방법 .....	27
그림 3-5 텍스트 문서.....	29
그림 3-6 텍스트 문서에서의 배경 픽셀 비중.....	29
그림 3-7 de-colorization을 수행하기 전과 후의 text sub-pixel 컬러값.....	33
그림 3-8 분류 실험 영상.....	40
그림 3-9 complex rendered text의 text 블록 판단 결과.....	43
그림 3-10 natural 영상에서의 text 블록 판단 결과.....	44
그림 4-1 sub-pixel rendered text.....	47
그림 4-2 gradient를 형성하는 sub-pixel 컬러 수.....	48
그림 4-3 Text 블록 코딩 방법.....	50

그림 4-4 Gradient fitting process .....	52
그림 4-5 gradient fitting process의 동작 .....	55
그림 4-6 $N_m = 1$ 인 경우와 $N_m = 0$ 인 경우 .....	61
그림 4-7 텍스트 코딩 입력 .....	65
그림 4-8 Whole-pixel 코딩 1 .....	66
그림 4-9 역방향 Sub-pixel gradient 코딩 1 .....	67
그림 4-10 Local minimum 코딩 1 .....	69
그림 4-11 순방향 gradient 코딩 1 .....	70
그림 4-12 Local maximum 코딩 1 .....	71
그림 4-13 역방향 gradient 코딩 2 .....	72
그림 4-14 Local minimum 코딩 2 .....	73
그림 4-15 순방향 gradient 코딩 2 .....	74
그림 4-16 Whole pixel 코딩 2 .....	75
그림 4-17 화질 비교를 위한 실험 영상 .....	79
그림 4-18 PSNR 실험결과 .....	81
그림 4-19 가독성 실험 영상 .....	83
그림 4-20 simple rendered text 화질 비교 .....	84
그림 4-21 complex rendered text 화질 비교 .....	85
그림 5-1 탐색영역 별 encoding 시간 (H.264/AVC inter mode) .....	89
그림 5-2 탐색영역별 데이터 변화량 .....	90
그림 5-3 compound image의 움직임 .....	911

그림 5-4 그룹 움직임 추정 방법.....	933
그림 5-5 그룹 생성 방법.....	955
그림 5-6 그룹의 분류.....	966
그림 5-7 이전 영상과 현재 영상 간의 정합성 비교.....	988
그림 5-8 그룹간의 방향성 선택.....	988
그림 5-9 움직임 벡터 생성을 위한 탐색 방법.....	1022
그림 5-10 그룹 움직임 추적 방법 실험영상.....	1055

# 표 목 차

표 3-1 텍스트 블록 구분율(%).....	41
표 4-1 fitting threshold.....	53
표 4-2 $C_s$ 와 $C_f$ 가 배경과 문자 컬러일 확률.....	58
표 4-3 $N_m$ 값의 변화에 따른 방법의 오차.....	62
표 4-4 $N_m$ 에 따른 보정 값.....	62
표 4-5 동작 예시에 사용된 text coder parameter.....	64
표 4-6 simple rendered text 압축율 비교.....	87
표 4-7 complex rendered text 압축율 비교.....	87
표 5-1 실험영상의 움직임 (pixels).....	1066
표 5-2 탐색 알고리즘별 압축율 (kbit/s @ 30frames).....	1088
표 5-3 탐색 알고리즘별 수행시간 (sec @ 60frames).....	1088
표 5-4 탐색알고리즘별 움직임 추정시간 (sec @ 60frames).....	1088

# 제1장 서론

## 1.1 연구 배경

컴퓨터의 성능과 네트워크 속도가 증가함에 따라 컴퓨터 스크린으로 표현되는 compound image 전송을 위한 다양한 접근이 시도되고 있다. 이러한 시도는 스크린 분할과 VDI (Virtual Desktop Infrastructure)와 같은 컴퓨터 가상화 시스템 (computer virtual system)으로 나타난다. [44]

가상화 시스템은 사용자가 공간의 구매를 받지 않고 원하는 데이터를 사용할 수 있는 장점을 가진다. 기존의 1:N 전송방식과는 다르게 서버와 클라이언트 간의 1:1 혹은 1:N의 상호 정보전송 방식을 통하여 전송된다. 이러한 상호 정보 전송 방식은 실시간 영상 전송이 중요한 고려사항이 된다.

가상화 시스템에서 생성되는 영상은 실행 프로그램에 의해 생성되는 다양한 형태의 영상 조합으로 구성된다. 특히 문서 프로그램과 같은 텍스트로 구성된 부분과 이미지 화면을 디스플레이 하는 영상 프로그램 등 서로 다른 특성을 가진 영상의 조합으로 구성되는 특징이 있다.

가상화 시스템은 실시간 전송 및 다양한 영상 조합에 의해서 구성되기 때문에 서버와 클라이언트 사이에서의 동일한 환경을 요구하게 된다. 다양한 영상을 압축 전송하기 위한 다양한 압축방식들이 서버와 클라이언트 사이에 존재하여야 하기 때문에 서버의 성능 뿐만 아니라 클라이언트의

성능도 중요하다. 또한 이를 제어하기 위하여 각 프로그램을 제어할 수 있는 오퍼레이팅 시스템 (OS) 으로 대변하는 상위 프로그램이 필요하다. 하지만 1:N의 상호 정보 전송에 클라이언트의 성능을 서버의 기준에서 맞추는 것은 불가능하다. 클라이언트와 서버간의 성능 불균형은 재현 할 수 있는 영상형태를 제약하게 된다. 반면에 클라이언트의 성능 증가는 전체 시스템의 비용을 증가시키고, 동일한 환경에서의 영상재현을 요구함으로써 가상화 시스템의 확산에 영향을 미치게 된다.

이러한 문제점을 해결하고자 최근의 연구 방향은 성능의 제약을 받지 않는 thin client 구현을 위하여 컴퓨터 스크린의 영상을 상위 프로그램에 정보를 요구하지 않고 디스플레이 되는 영상만을 확인하여 영상 압축 및 전송하는 방법에 많은 관심을 가지고 있다. [23-27][40][45][46] 이러한 영상 캡처(capture) 방식은 생성되는 영상의 프로그램의 의존도를 낮출 수 있을 뿐만 아니라 클라이언트의 성능에 대한 제약 조건을 해결 할 수 있다. 이러한 캡처 방식을 통한 compound image의 압축, 전송 방식이 thin client를 구현하는 방법으로 효과적이다. 하지만 이러한 캡처 영상 전송 방식은 영상의 크기가 증가하고 다양한 영상을 동일한 압축 알고리즘에 따라 생성하기 때문에 영상의 특성에 따른 화질 저하 현상이 발생하게 된다.

네트워크 속도의 발전에 따라 단일 시간 내에 전송하는 데이터의 양은 증가 하였지만 컴퓨터 영상의 사이즈가 커짐으로써 실시간 영상 전송에 많은 어려움을 겪게 된다. 따라서 이러한 대용량의 데이터를 효과적인 방법으로 압축 재현하는 방법이 필요하게 된다. 이에 따라 기존의 단일 압축방식에서 영상의 특성 별로 압축방식을 분리하여 사용하는 방법이 요구

되고 있다. 하지만 손실 및 무 손실 압축이 필요한 compound image 압축 방식에서 이러한 영상의 높은 압축률을 유지하기는 쉽지 않다. 또한 각 영상의 특성에 따른 화질 저하 및 급격한 영상 데이터의 변화를 제어하는 것이 쉽지 않다. 이러한 문제는 compound image 압축방식에 높은 압축률 뿐 만 아니라 화질을 보장할 수 있는 방법을 요구하게 된다.

Compound image 압축에서 고려사항은 실시간 전송의 구현이 요구된다. 따라서 복잡한 압축방식 보다 실시간 구현에 적합한 저 복잡도를 가진 알고리즘이 필요하다. 본 연구에서는 이러한 compound image 압축의 다양한 요구조건을 충족시키고자 저 복잡도를 가진 텍스트 압축 알고리즘을 제안하고, 영상의 특성에 맞는 움직임 추정 알고리즘 및 영상의 분류 방법에 대해서 설명하고자 한다.

## 1.2 연구 내용

Compound image의 실시간 압축 및 전송에 요구가 높아짐에 따라 화질의 저하가 없는 저 복잡도의 압축 알고리즘이 필요하게 되었다. 본 연구에서는 이러한 요구조건에 맞는 compound image 압축 시스템에 대하여 설명한다.

첫째로 영상의 특성에 맞는 분류 방법에 대하여 설명한다. Compound image는 다양한 영상의 조합으로 구성된다. 영상의 구성은 단일 압축 알고리즘을 사용하기 보다는 영상의 특성에 맞는 압축 알고리즘을 선택하는 것이 중요하다. 효율적인 압축 알고리즘의 선택을 위하여서 각 영상의 특징을 이용하여 블록 단위의 영상 분류과정을 본 연구에서는 제안한다. 컴퓨터 영상은 RGB로 구성된 sub-pixel 단위로 컬러가 구성된다. 이러한 특징을 이용하여 텍스트와 이미지 블록으로 구분하게 된다. 특히 텍스트의 생성 방법을 분석하여 캡처 된 영상에서 텍스트로 구성된 블록을 정확히 구분할 수 있다. 따라서 기존의 방법에 비해 텍스트 블록의 분류의 정확도를 높인 새로운 블록 분류 방법에 대하여 제안한다.

둘째로 compound image의 특성에 맞는 움직임 추정 방법에 대하여 설명한다. Compound image는 자연영상과 다르게 영상의 움직임이 단순하다. 이러한 영상의 움직임은 각 픽셀 및 블록의 움직임을 추정하는 방식보다 영역의 움직임을 추정하여 픽셀의 움직임을 확인하는 방법이 복잡도를 줄일 수 있는 방법이 된다. 본 연구에서는 영역 움직임을 확인하여 영상의 움직임을 확인할 수 있는 그룹 움직임 추정 방법에 대하여 설명하기로 한다.



그룹 움직임 추정방법은 전체 시스템의 압축률 및 복잡도를 낮출 수 있는 방법이 된다. 그룹 움직임 추정방법은 움직임을 단순화 하여 움직임 추정방법의 복잡도를 낮출 수 있다.

마지막으로 sub-pixel gradient를 이용한 텍스트 블록 압축방법을 설명한다. 블록 분류 방법과 마찬가지로 컴퓨터 스크린 영상의 특성을 이용한 압축 방법이다. 하나의 whole-pixel을 sub-pixel 단위로 분리하여 sub-pixel gradient 차이를 이용한 압축방법이다. 기존의 whole-pixel 단위의 압축 방식에 비하여 텍스트의 생성과정을 역이용함으로써 텍스트 블록의 화질 유지 및 압축률을 높일 수 있다. 일반적으로 문서파일과 같은 영상에서의 압축방식은 무 손실 압축 및 높은 화질의 손실 압축을 선택하게 된다. 이는 영상의 화질이 단순한 PSNR의 기준과 함께 사람이 문서를 읽는 가독성 (readability)이 중요한 요소이기 때문이다. 문서의 화질은 텍스트가 구성되어 있는 픽셀의 변화량이 많을수록 영상의 가독성은 떨어지게 된다. 본 연구에서는 이러한 문서의 특징을 이용하여 저 복잡도의 텍스트 영상 압축 방식을 설명하고자 한다.

### 1.3 논문 구성

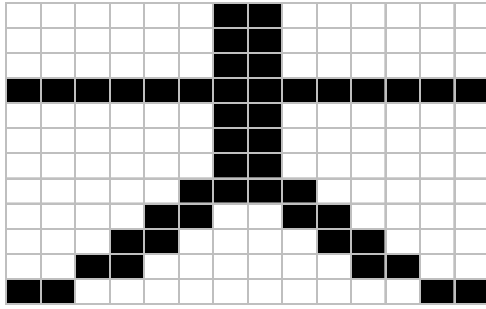
본 연구의 나머지 구성은 다음과 같이 구성된다. 2장에서는 텍스트를 구성하는 sub-pixel rendering 과정과 H.264/AVC inter mode의 움직임 추정에 대하여 확인하고, compound image에 관련한 기존연구들을 소개한다. 3장에서는 sub-pixel gradient를 이용한 블록 구분 방법에 대하여 자세히 설명한다. 또한 블록 구분방법에 대한 실험결과를 확인한다. 4장에서는 sub-pixel gradient를 이용한 텍스트 블록 압축방식에 대하여 설명한다. 이를 통하여 복잡한 컬러를 가진 다양한 텍스트에 대한 압축 성능을 평가한다. 5장은 compound image의 특성을 이용한 그룹 움직임 추정 방법에 대하여 설명한다. 이를 통하여 다양한 이미지 압축 방식에 적용하여 결과를 확인한다. 마지막으로 본 연구에 대한 결론을 맺는다.

## 제2장 텍스트 생성과정 및 기존압축방법

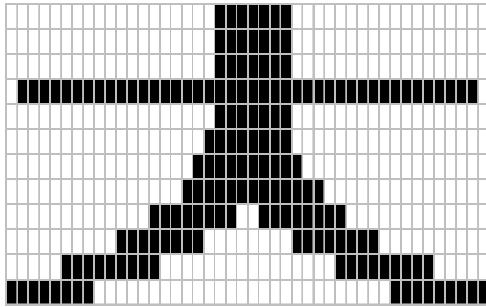
### 2.1 텍스트 생성과정

평판 디스플레이에서의 색을 표현하는 whole-pixel은 실질적으로 Red, Blue, Green의 3개의 sub-pixel channel로 구성되어 있다. 3개의 sub-pixel이 하나의 whole-pixel로 구성되는 것이다. 평판 디스플레이의 가로 해상도는 세 배의 해상도로 이루어진 sub-pixel 배열로 이루어져 있다고 볼 수 있다. Text의 sub-pixel rendering은 이러한 가로 해상도가 세 배인 평판 디스플레이에서 text를 부드럽게 보여주는 방법이다. [31-35]

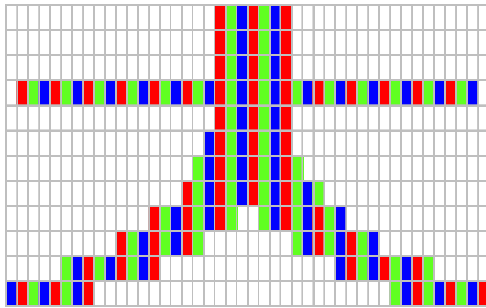
그림 2-1 과 같이 ‘ㅏ’ 의 텍스트의 변환 과정을 살펴보면 이러한 sub-pixel rendering 과정의 단계를 알 수 있다. 텍스트를 whole-pixel 을 기준으로 구성하여 확대해보면 그림 2-1(a) 와 같다. 그림 2-1(a) 는 부드러운 라인이 생성되지 않고 픽셀 별 블록현상이 나타나는 것을 볼 수 있다. 그림 2-1(b)는 이러한 whole-pixel 단위의 글씨를 sub-pixel 단위의 해상도로 확장하여 표현한 것이다. 이렇게 해상도를 확대해보면 실제 글씨와 같이 부드럽게 만들 수 있게 된다. 이것이 평판 디스플레이에서 sub-pixel rendering 을 가능하게 하는 것이다.



(a) 확대된 whole-pixel 텍스트



(b) 확대된 sub-pixel 텍스트



(c) 확대된 sub-pixel 텍스트의 RGB 값

그림 2-1 '츠' 텍스트의 sub-pixel 변환과정

그림 2-1 (c)는 이러한 평판 디스플레이에서 RGB의 sub-pixel 컬러를 sub channel로 표현한 것이다. Whole-pixel 에서 인접한 sub-pixel의 해상도를 사용하여 텍스트를 조정하게 되면 기존 whole-pixel 의 글씨보다 더 높은 부드러운 모양으로 표현 할 수 있다. Sub-pixel 로 표현한 컬러는 보상되는 컬러와 인접하기 때문에 눈에 보이는 컬러는 검은색과 흰색으로만 인식되게 된다.

하지만 이러한 단순한 sub-pixel rendering은 컬러 fringing 현상을 일으킨다. 이러한 fringing 현상은 RGB의 sub-pixel 단위로 rendering을 진행하기 때문에 텍스트의 외곽에 적색과 청색이 도드라지게 나타날 수 있는 문제점을 보여준다. 텍스트를 구성하는 sub-pixel이 whole-pixel 단위로 떨어지지 않기 때문에 그림 2-2와 같이 fringing 현상을 설명한다.

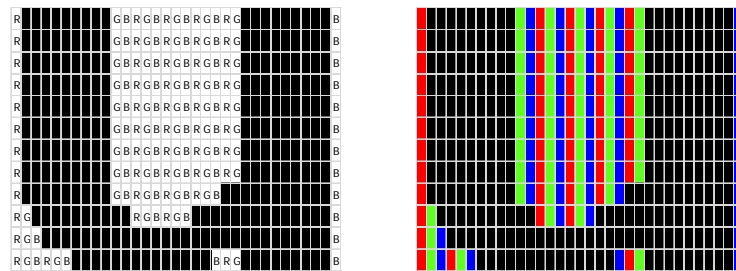


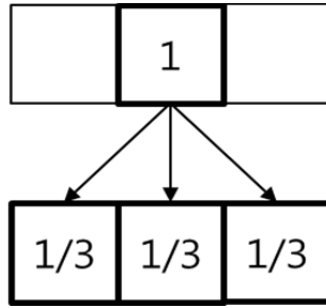
그림 2-2 ‘u’ sub-pixel fringing 현상

그림 2-2 의 ‘U’의 글씨의 왼쪽 경계면은 적색 sub-pixel 로, 오른쪽 경계면은 청색 sub-pixel 로 이루어져 있다. 이러한 현상 때문에 ‘U’의 왼쪽에서 적색 빛이 나타나는 것처럼 보이고 오른쪽에서는 청색 빛이 나타나

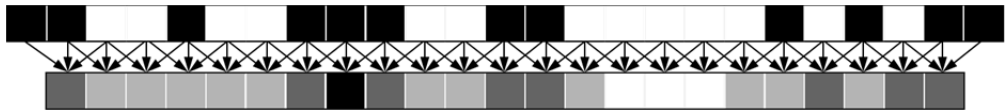
는 것처럼 보인다. 또한 텍스트를 구성하는 sub-pixel이 3의 배수가 아닌 경우, 각 RGB channel 빛 양의 불균형에 따른 컬러 불균형 현상이 발생할 수 있다. 사람의 눈은 이러한 컬러 불균형 현상에 예민하기 때문에 텍스트를 검은색이 아닌 다른 색으로 인지하게 된다. 기존의 whole-pixel rendering 방식은 RGB를 하나의 단위로 보았기 때문에 이러한 문제가 발생하기 않았다. 하지만 sub-pixel로 텍스트를 세분화 하게 되면 컬러 불균형 현상이 나타나게 되는 것이다.

이러한 문제점을 해결하는 간단한 방법은 각 sub-pixel channel의 에너지를 인접한 두 sub-pixel channel 에 분산시키는 것이다. 그림 2-3(a)은 이러한 에너지 분산에 대한 간단한 방법이다.

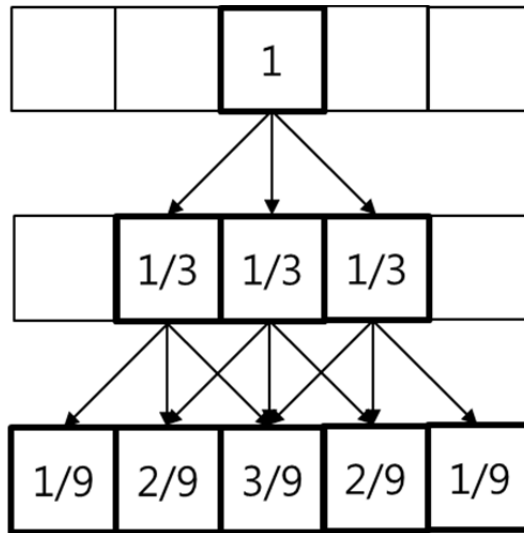
하나의 sub-pixel이 on 되었을 때, 빛의 세기를 1/3정도만 배분하고, 인접한 두 개의 sub-pixel에 1/3씩 분배를 한다. 하나의 sub-pixel이 에너지를 다 갖는 것이 아니라, 인접한 sub-pixel들과 공유하는 방법이다. 각각의 인접한 sub-pixel 은 보색 관계이기 때문에, 이 방법은 컬러의 균형을 맞추게 된다. 항상 같은 양 만큼 RGB sub-pixel의 빛의 양이 되게 때문에 컬러 불균형 현상을 감소 시킬 수 있다. Filtered sub-pixel 의 밀도가 해당 sub-pixel의 밝기와 인접한 두 개의 sub-pixel 밀도가 1/3로 분산된다. 만약 특정 sub-pixel 과 인접한 두 개의 sub-pixel 이 모두 on 이 되어 있다면 filtered sub-pixel은 에너지는 1이 될 것이다. 인접한 sub-pixel 이 하나만 on 되어 있다면 밀도는 2/3 가 되는 것이다.



(a) 1차 에너지 분산 방법



(b) 에너지 분산 결과



(c) 2차 에너지 분산 방법

그림 2-3 sub-pixel 에너지 분산방법

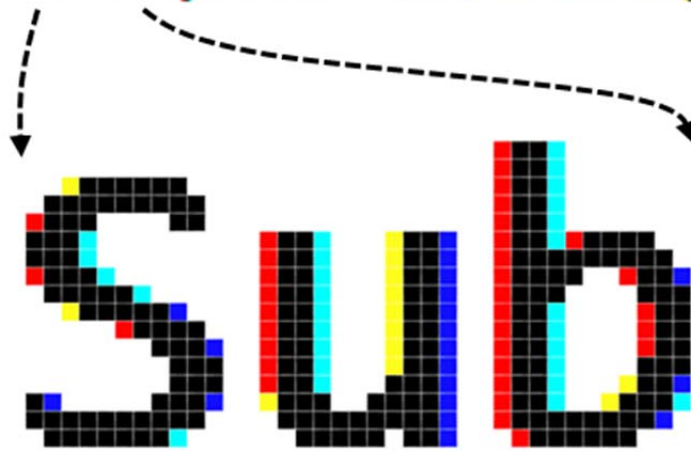
그림 2-3(b)는 이러한 분산 방법을 하나의 텍스트 라인에 적용한 결과이다. 하지만 1차 에너지 분산 방식은 이러한 색 blurring 현상을 완전히 해결한 것은 아니다. 따라서 이러한 blurring 현상을 줄이기 위하여 다양한 기법들이 추가 되고 있다. 그러한 기법 중에 가장 간단한 방법은 필터링 방법을 한번 더 반복하는 것이다. 이러한 방법은 그림 2-3(c)에서 보여주고 있다. 그림 2-3(c)와 같이 에너지 분배 과정을 두 번 반복하게 되면 컬러 불균형 현상을 감소 시킬 수 있다. 동시에 인접한 sub-pixel의 영향력을 인접한 sub-pixel에 재 분배 하게 되면서 중심의 sub-pixel에 인접한 sub-pixel의 영향력을 감소 시킬 수 있다. Sub-pixel의 밀도 분배는  $[1/9, 2/9, 3/9, 2/9, 1/9]$ 의 계수를 가지는 low-pass 필터로 볼 수 있다. 이러한 필터는 계수의 합이 1이 되기 때문에 하나의 sub-pixel 의 에너지가 인접 다섯 개의 sub-pixel로 분리되는 것을 알 수 있다.

그림 2-4(a) 는 한 번의 필터링으로 sub-pixel rendering 된 텍스트의 형태이다. 확대된 그림 2-4(a) 와 같이 텍스트의 주변에 적색과 청색의 컬러가 보이는 것을 알 수 있다. 그림 2-4(b) 는 두 번의 필터링 후에 보여지는 텍스트 이다. 그림 2-4(b) 와 같이 텍스트의 외곽선이 그림 2-4(a) 와 비교하여 부드러워 졌음을 알 수 있다.

이렇게 sub-pixel rendering은 평판 디스플레이의 컬러 특성을 이용하여 텍스트를 부드럽게 표현하는 과정이다. 이러한 과정을 통하여 최종적으로 모니터에 나타나는 텍스트가 표현된다. 이와는 별도로 텍스트 프로그램에 따라 sub-pixel rendering에 추가하여 감마 보정 등 다양한 기법이 사용된다.

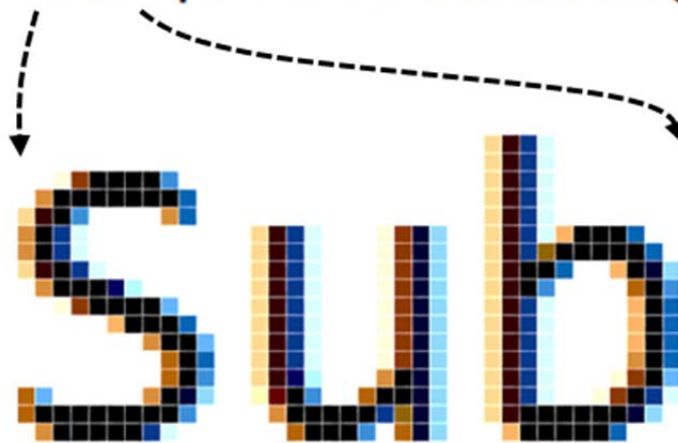


# Sub-pixel Rendering



(a) 필터링 전 Sub-pixel rendering 텍스트

# Sub-pixel Rendering



(b) 필터링 후 sub-pixel rendering 텍스트

그림 2-4 필터링 전/후 sub-pixel rendering 텍스트

## 2.2 표준 영상 압축 방법

본 절에서는 본 연구와 관련한 기존의 영상 압축 방법에 대하여 설명하고자 한다. 영상 압축은 영상의 데이터를 최소화 하여 전송을 목적으로 한다. 압축 과정에서 복구된 영상이 원본 영상과 동일하지 않는다면 손실 압축 (lossy coding)으로 표현하고, 원본 영상의 손실 없이 복구 가능하다면 무손실 압축 (lossless coding) 이라고 할 수 있다. 영상을 무손실 압축으로 코딩을 진행한다면 압축률에 한계가 있다. 반대로 손실 압축의 경우 압축률을 더 높일 수 있지만 데이터 손실에 따른 영상의 왜곡이 발생하게 된다.

영상 압축의 과정은 영상 데이터를 변환 하여 공간적 중복성을 줄이는 것으로 시작한다. 영상 압축에 주로 사용하는 대표적인 변환과정은 DCT(discrete cosine transform) 과 DWT(discrete wavelet transform)이다. DCT 방식의 표준 영상 압축 방법으로 JPEG (Joint Photograph Experts Group)오류! 참조 원본을 찾을 수 없습니다. 이 대표적이다. DWT 방식의 표준 영상 압축 방법은 JPEG2000[3]으로 설명할 수 있다. 하지만 이러한 표준 영상 압축은 자연영상의 압축에 적합하다고 할 수 있다. 위의 방식으로 영상을 압축할 경우에 영상은 양자화 (quantization) 과정을 거친다. 양자화 과정을 통하여 데이터의 일부가 제거 된다. 영상의 손실을 줄이고자 다양한 양자화 방법이 사용되고 있다. 이러한 양자화 과정을 통하여 일어나는 손실로 인하여 원본 데이터와 다른 손실 압축 방식이 되는 것이다.

영상 압축의 가장 널리 사용되는 대표적인 표준방식은 JPEG 이라고 할

수 있다. 인터넷 영상의 압축에 많이 사용된다. JPEG은 위에서 설명한 바와 같이 영상 변환 과정으로 DCT 방식을 사용하고, 양자화 행렬을 이용한 양자화 과정을 거친다. 마지막으로 엔트로피 코딩을 위하여 run-length[5]와 허프만 코딩[4]을 사용한다. 최근에는 웨이블릿 (wavelet) 기반 영상 압축 방식인 JPEG2000도 널리 사용되고 있다. 웨이블릿 기반의 영상 압축방식은 기존의 DCT 대신 DWT 방식이 사용된다. DCT 기반의 압축 방식 보다 높은 압축 효율을 가진다. 또한 엔트로피 코딩으로 기존의 run-length나 허프만 코딩이 아닌 비트 평면 코딩을 사용함으로써 효율을 높였다.

위에 설명한 JPEG 이나 JPEG2000의 경우에는 양자화 과정을 통하여 데이터 손실이 발생하기 때문에 양자화의 결과에 따라 영상의 화질을 떨어뜨릴 수 있다. 이와는 반대로 영상의 손실을 가지지 않는 무손실 압축 방법으로 PNG (Portable Network Graphics)[6] 를 설명 할 수 있다. PNG 는 기존의 LZW 알고리즘[7] 기반의 GIF 방식에 256 컬러로 표현하는 한계를 극복하게 되었고, 패턴이 반복되거나 단순한 형태의 파일을 압축할 경우 높은 압축률을 보인다. 기본 압축 알고리즘은 deflate 알고리즘[8]을 사용한다. Deflate 알고리즘은 zip 포맷에 사용하기 위해 만들어진 압축 알고리즘이다. LZ77 알고리즘[7]으로 1차 압축한 후에 2차로 허프만 압축 알고리즘을 사용한다. 이러한 무 손실 압축 알고리즘의 경우 단순한 컬러를 가진 영상 압축에 적합하다. 또한 원본 데이터를 유지할 수 있기 때문에 최근에 와서 인터넷 영상 압축 방식으로 널리 사용되고 있다.

## 2.3 H.264 inter prediction

기존의 움직임 추정방법은 H.264/AVC 오류! 참조 원본을 찾을 수 없습니다. inter mode와 같은 표준 방법이 사용되고 있다. 특히 과거 영상과 현재 영상의 차이를 이용한 움직임 추정을 많이 사용하게 된다. 본 절에서는 이러한 H.264/AVC의 inter prediction 방법에 대해 간단히 살펴보고자 .

H.264 /AVC inter prediction에서는 한 가지로 고정된 크기의 블록이 아니라 다양한 블록의 움직임을 찾는다. 이러한 다양한 블록의 구성은 H.264/AVC inter prediction에서 정의된 모드 분류로 구성된다. 모드1은 매크로 블록이 16x16 블록 하나로 처리됨을 의미하고 모드2는 매크로 블록이 16x8 블록 두 개로 나누어 처리 되었음을 의미한다. 이 경우에는 블록마다 움직임 벡터가 존재하게 된다. 마찬가지로 8x16 블록 2개로 구성될 때는 모드3으로 정의한다. 이러한 모드 분류는 H.264/AVC inter prediction에서 총 7개의 모드를 정의하고 있다.

그림 2-5은 이러한 H.264 inter prediction에서 사용하고 있는 7개의 모드를 표현한 것이다. 앞선 영상과 현재의 영상의 움직임을 추정하는 inter prediction은 한 개 모드에 한가지의 predicted 이미지가 있는 것이 아니라 탐색 범위내의 픽셀 개수만큼 존재하게 된다. 이러한 predicted 이미지에 대하여 하나의 움직임 벡터를 가지게 되는데 움직임 벡터들은 최종 움직임 벡터의 후보 군이 되는 것이다. 이러한 움직임 벡터의 후보군들 중 최적의 predicted 이미지를 가리키는 움직임 벡터가 최종 움직임 벡터가 된다.

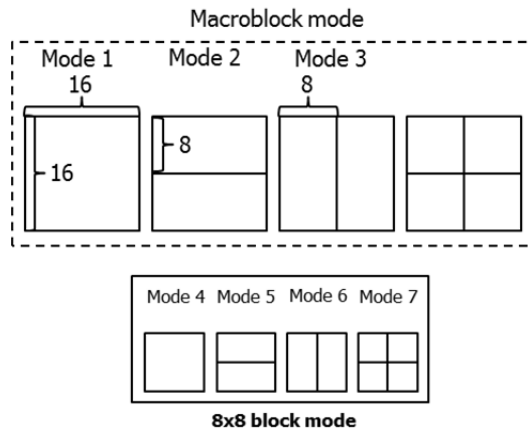


그림 2-5 Inter prediction 모드 종류

최적의 predicted 이미지를 찾기 위하여 SAD 기반의 distortion 방식을 사용한다. Distortion 을 계산할 때에는 SAD(Sum of Absolute Differences) 를 사용할 수 있다.

$$SAD = \sum_x \sum_y |S(x, y) - P(x, y)|, \quad (x, y \in \{0, 1, 2, 3\}) \quad (2-1)$$

식 (2-1)은 4x4 블록 크기를 가지는 블록을 기준으로 SAD 를 계산 하는 방법이다. S는 현재의 4x4 블록의 영상을 의미하고, P는 현재의 4x4 블록과 비교할 과거 영상을 나타낸다. x와 y는 4x4블록 내 픽셀의 x 방향과 y 방향의 위치를 나타낸다. 따라서  $S(x,y) - P(x,y)$  는 가로방향으로 x, 세로 방향으로 y위치에 있는 픽셀 컬러 값의 예측 오차를 의미한다. SAD는 이

러한 4x4 블록 내 모든 픽셀 위치의 예측 오차를 합한 값이 된다. Inter prediction은 계산된 SAD 값들을 기준으로 distortion이 가장 작은 블록을 선택하게 되는 것이다. 가장 작은 SAD 값은 데이터의 양을 줄이기 때문에 최종적으로 압축률이 좋아진다고 할 수 있다.

## 2.4 Compound image 의 압축 알고리즘

Compound image는 natural image와는 다르게 이미지의 구성이 다양한 특성을 가진 부분으로 구성되어 있는 영상이다. 특히 텍스트 부분과 이미지 부분으로 대표되는 서로 상이한 영역으로 구성되어 있기 때문에 각 특성을 해결할 수 있는 단일 압축 알고리즘을 찾기는 쉽지 않다.

이미지 압축 알고리즘인 H.264, JPEG, JPEG2000 과 같은 표준 압축 알고리즘은 자연 영상 압축에 집중되어 있다. 이러한 알고리즘은 DCT 나 wavelet 기반의 압축 알고리즘으로서 이미지의 압축률에 관점을 둔 알고리즘이라고 할 수 있다. 다양한 조건의 이미지 압축에 대한 고려를 포함하고 있지 않기 때문에, 특성이 상이한 이미지의 조합에서는 화질과 압축률의 최적 조건을 만족하기에는 효과적이지 않다. 이러한 상이한 영역을 가진 이미지 압축 방식은 다양한 방법으로 연구되어 왔다.

Compound image 압축에 관한 연구진행은 크게 layer 기반의 접근과 블록 기반의 접근 방식으로 나누어 볼 수 있다. 이러한 압축에 관한 연구진행은 영역을 효과적으로 나누는 방법을 포함하여 영역에 적합한 압축 방식을 선택하는 것으로 생각해 볼 수 있다.

MRC로 대표되는 layer 기반 접근 방식은 각 이미지 내용의 특성에 따라 최소한 3개의 layer로 나누게 된다 [9-15]. Background / foreground / mask layer로 영상을 나누어 각 layer에 맞는 압축 알고리즘을 선택한다. layer 분리 및 압축 방식은 복잡한 이미지를 특성에 맞는 layer 로 나누고 효율적인 압축 알고리즘을 사용함으로써 높은 화질과 높은 압축률을 확보

할 수 있다. 하지만 픽셀 기반으로 분리 및 압축 함으로써 복잡도가 높아지는 문제는 피할 수 없는 약점이다.

블록 기반의 접근방식은 16x16 non-overlapping 블록의 기준에서 이미지의 특성에 따라서 이미지를 블록 별로 분리한다. 이러한 블록 별 분리방법은 각 블록의 적합한 압축방식을 선택하여 압축을 진행한다.

기존 논문에서는 이러한 블록의 분리방법에 대하여 color count, color histogram 등의 다양한 방식을 제시하고 있다[19-26]. 분리된 블록을 기준으로 적합한 표준 압축 알고리즘을 선택하고 압축을 진행한다.

가장 대표적인 compound image 압축방식인 Lin [19]의 SPEC 논문을 살펴보면 블록을 color count 기준으로 text / graphic / picture 블록으로 분리한다. 이렇게 분리된 블록을 기반으로 각 특성에 맞는 알고리즘을 선택한다. Lin [19]논문에서는 각 블록의 압축방법으로 text / graphic은 LZW, picture는 JPEG을 선택하고 있다. 이러한 압축방법은 가독성 및 화질에 민감한 텍스트 블록에서 대체서 무 손실 기반의 LZW를 사용하고, 영상 화질에 대하여 덜 민감한 그래픽 부분은 손실 기반의 압축 알고리즘인 JPEG을 선택함으로써 각 영역의 특성에 대한 손실을 줄이면서 영상의 압축 효율성을 높일 수 있었다. 하지만 텍스트 압축에 사용하는 LZW는 복잡도가 높은 압축알고리즘으로서 계산량이 많아 지기 때문에 압축 효율성을 높일 수는 있었지만 수행시간을 증가시키는 문제점을 가지게 된다.

Lan [20]의 논문을 살펴보면 텍스트 블록의 압축 효율 및 화질을 동시에 고려하고 있다. 텍스트 블록의 압축을 위하여 블록 기반의 BCIM (base color and index map)을 사용한다. 텍스트 영역이 가진 컬러의 조합이 기준



의 흑백 컬러가 아닌 다양한 컬러를 가지기 때문에 최적의 base 컬러를 선택하여 텍스트 블록의 화질 손실을 최소화 하는 방법을 선택하였다. 이러한 접근 방법은 기존의 무손실 기반의 텍스트 블록 압축방식 선택에서 압축률의 효율성을 위한 발전된 방법으로 판단된다. 다만 최적의 base 컬러의 선택에 있어서 복잡한 dynamic 알고리즘을 선택함으로써 여전히 높은 복잡도를 유지하게 된다. 또한 최적의 압축률을 위하여 텍스트 압축 방법을 H.264 intra mode의 하나로 포함하게 함으로써, 전체 텍스트의 압축 방식으로서의 기능보다는 텍스트의 특성을 많이 내포하고 있는 부분의 압축방식으로서 한정된다.

이렇게 텍스트 블록 압축방식의 선택에 있어서 Pan [21]은 문제의 분석을 복잡도를 낮추는 방향으로 접근하였다. Base 컬러를 선택하는 방법에서 dynamic 알고리즘 대신 histogram 방식을 사용하였다. 이러한 방법은 Lan [20]의 방법보다 복잡도를 낮추는 효과를 보여준다.

이러한 base 컬러를 이용한 index map 방식들은 단순한 컬러의 조합으로 구성되는 블록의 경우는 아주 효과적이다. 하지만 PDF 문서와 같은 복잡한 컬러를 가지는 블록에서는 압축률의 저하를 가져온다.

Compound image의 텍스트 영역의 경우는 기존은 단순한 블록 분류 방법으로는 해결하지 못하는 문제가 있다. 다양한 프로그램에서 생성되는 텍스트 영역은 텍스트의 고유한 가독성을 높이기 위하여 gamma 조정, sub-pixel rendering, anti-aliasing 등 다양한 방법을 포함하고 있다. 이러한 방법들은 다양한 특성의 텍스트 영역을 생성해 낸다.

이러한 sub-pixel rendering 에 대한 텍스트 영역의 분석은 [29],[30]에서 처

럼 rendered 방식의 텍스트를 분리해 내고 인식하는 것에 대하여 연구 되어 왔다. 이러한 컴퓨터 스크린 영상은 가독성 및 화질 관점에 의한 다양한 컬러의 rendered 텍스트로 분류 될 수 있다.

움직임이 있는 동영상에서의 압축방식을 살펴보면 H.264/AVC inter mode 압축 방식이 대표적이다. H.264의 움직임 추정 방법은 탐색범위 내에서 최적의 탐색 점을 찾는 방법으로 발전해 왔다. 탐색 중심을 기반으로 탐색 점을 설정하는 방식에 따라 three step search(3SS)[47], diamond search[48]와 같은 integer 움직임 추정 방식이 사용되어 왔다. 이러한 알고리즘은 다양한 비교 블록의 SAD를 구하여 최소의 cost를 이용하여 움직임을 추정한다. 움직임 추정 방법은 탐색 범위가 넓어지게 되면 연산량도 같이 증가하는 문제점을 가지게 된다. Compound image에서의 움직임은 단일 움직임으로 형성되고 natural image의 움직임보다 움직임의 폭이 크다. 따라서 H.264/AVC 의 움직임 추정 방법은 많은 연산량을 가지고 있기 때문에 compound image에서의 움직임 추정 방법으로 사용하기 위하여는 복잡도를 줄여야 할 필요성이 있다. Wang[49]의 논문과 같이 compound image에서의 내용을 분리하여 압축방식을 달리하는 방식을 연구되었다. 하지만 이러한 영상 분리 방법은 영상의 움직임을 추정하는 것이 아니라 움직이는 영상의 영역을 찾아내는 방법으로, 영상 압축을 위한 영상 움직임 추정은 아니다. 본 연구에서는 compound image의 특성을 이용한 영역 움직임을 추정하여 압축효율을 높이고자 한다.

### 제3장 Sub-pixel gradient 블록 분류 방법

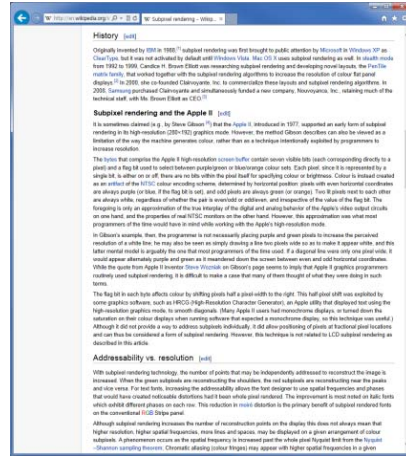
본 장에서는 compound image를 sub-pixel gradient를 이용한 text와 image 블록으로 분류하는 방법에 대하여 설명한다. 분류의 기준은 16x16 non-overlapping 블록을 기반으로 compound image를 text와 image 블록으로 분류한다. Sub-pixel gradient 블록 분류 방법은 다음 장에서 설명하는 그룹 움직임 추정 방법 및 text 블록 압축 방법의 알고리즘의 중요한 입력이 된다.

그림 3-1은 일반적으로 사용되는 compound image를 나타낸다. 그림 3-1(a) 와 같이 배경 컬러와 텍스트의 컬러로만 구성되어 있는 단순한 텍스트 영상과, 그림 3-1(b)과 같이 다양한 폰트를 사용하는 복잡한 텍스트 영상, 그리고 그림 3-1(c) 와 같이 문서 저장용으로 사용되는 PDF등 서로 다른 특성을 가진 텍스트 형태를 가진 문서 화면이 존재하게 된다. 또한 텍스트로 만 구성되어 있는 것이 아니라 그림 3-1(d) 과 같이 이미지로 구성될 수 있다.

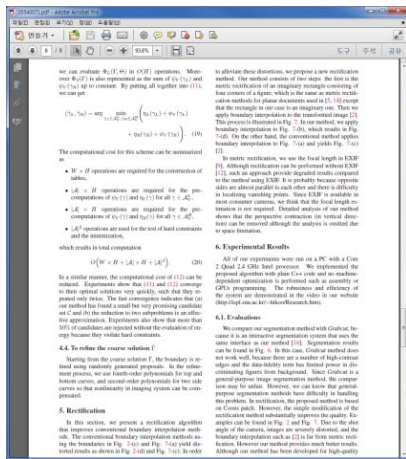
본 연구에서는 이러한 영상의 블록 분류를 bi level text, simple rendered text, complex rendered text, 그리고 image 블록으로 구분한다. 분류 기준은 다양한 문서에서 나타나는 텍스트의 특성을 기반으로 text 블록을 구분하였다. 기존의 블록 분류 방법[19]과 같이 컬러 분포도 분석 방법[21]을 사용하는 경우에는 분류하는 기준 값에 따라서 분류 오차율을 가지게 된다.



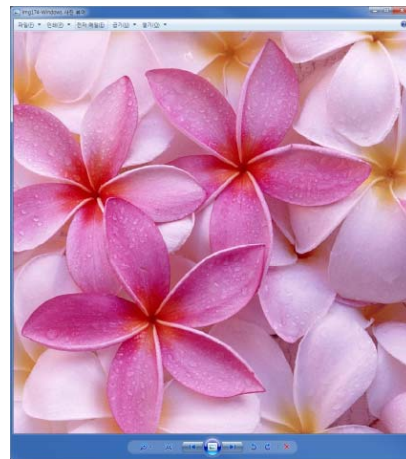
(a) 단순 컬러 텍스트 문서 (2-color)



(b) 복잡한 컬러 텍스트 문서 (N-color)

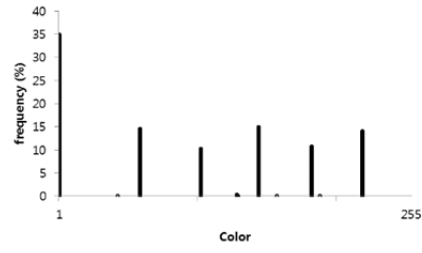
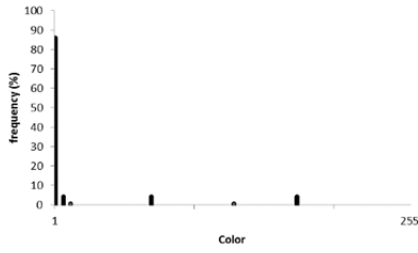


(c) PDF 문서

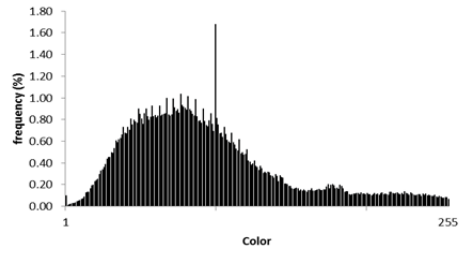
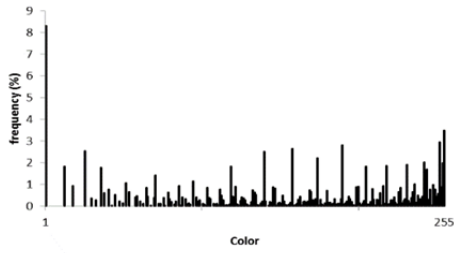


(d) Natural image

그림 3-1 compound image 종류



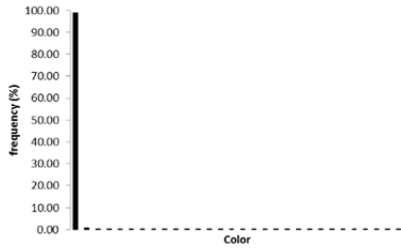
(a) 그림3-1(a) sub-pixel color 분포 (b) 그림3-1(b) word의 sub-pixel color 분포



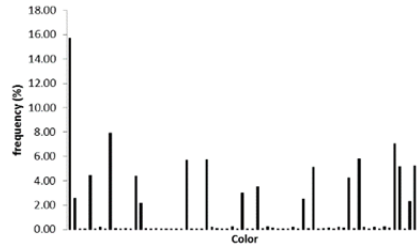
(c) 그림3-1(c) sub-pixel color 분포 (d) 그림3-1(d) sub-pixel color 분포

### 그림 3-2 sub-pixel color 분포도

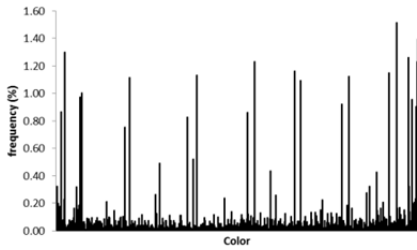
그림 3-2와 그림 3-3과 같이 text 블록내의 sub-pixel 과 whole-pixel 컬러 값을 분석해보면 complex rendered text의 경우 이미지의 컬러분포 비교하였을 때 특정한 컬러 값으로 수렴되는 것이 아니라 다양한 컬러의 조합을 가지게 된다. 따라서 컬러 값 분포도 분석을 기반으로 하는 분류 방법은 text 블록 분류에서 오차를 가지게 된다. 이러한 분류 오차율은 텍스트 영역 내에서 다른 압축 알고리즘을 적용하게 되면 블록의 화질저하를 일으키게 된다. 본 연구는 이러한 오차를 낮출 수 있는 효과적인 sub-pixel gradient 분석을 통한 블록 분류 방법을 설명하고자 한다.



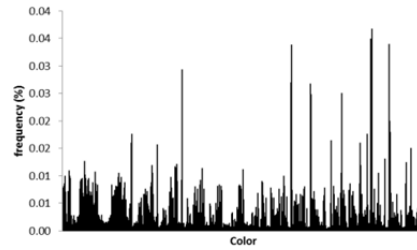
(a) 그림3-1(a) whole-pixel color 분포



(b) 그림3-1(b) whole-pixel color 분포



(c) 그림3-1(c) whole-pixel color 분포



(d) 그림3-1(d) whole-pixel color 분포

### 그림 3-3 whole-pixel color 분포도

Sub-pixel gradient 분석 분류 방법은 평면 영상 디스플레이에서 사용하는 sub-pixel 컬러 단위의 gradient를 분석하여 분류하기 때문에 블록 판단의 오차율을 최소화 할 수 있는 효과적인 방법이라고 할 수 있다. Sub-pixel gradient 분류 방식은 총 3단계의 과정을 거치게 된다. 첫째 단위 블록당 생성된 Sub-pixel 컬러로부터 배경과 텍스트(font) 컬러를 분리하는 배경 & 텍스트 컬러 추출 단계, 둘째로 추출된 텍스트 컬러의 값을 검은색 텍스트 컬러로 변환하는 텍스트 de-colorization 단계, 마지막으로 검은색으로 변환된 텍스트의 sub-pixel gradient를 분석하여 단위 블록의

분류를 진행하는 블록 분류 단계를 수행하게 된다. 그림 3-4은 본 장에서 설명하는 sub-pixel gradient를 이용한 블록 분류 방법의 전체 흐름도 이다.

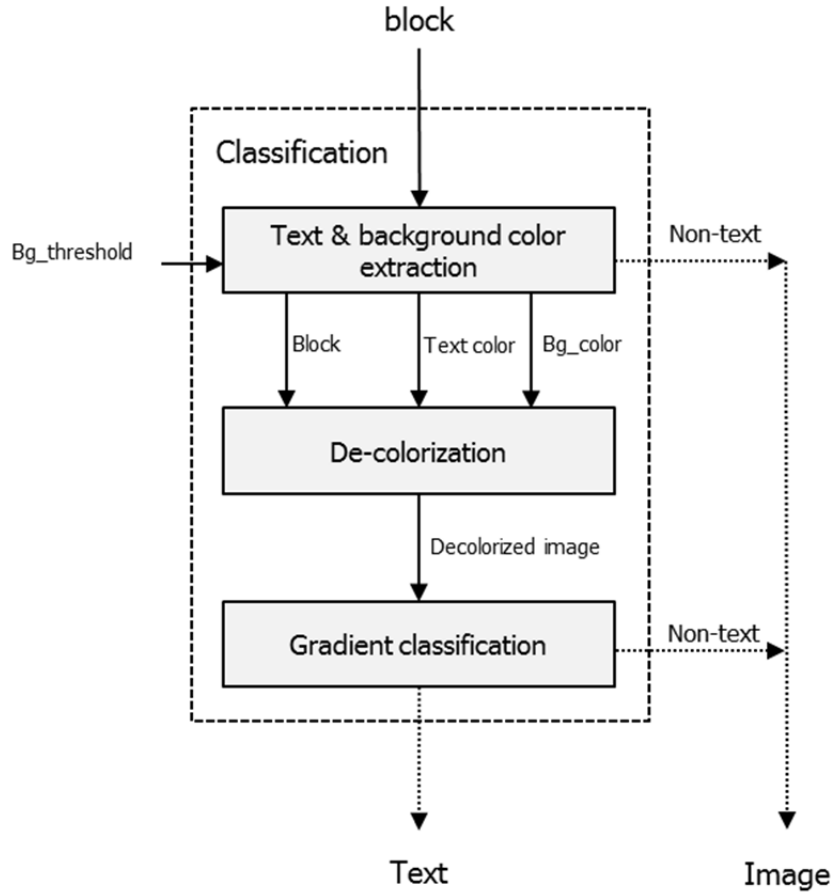


그림 3-4 Sub-pixel gradient를 이용한 블록 분류 방법

### 3.1 Background & Text color extraction

텍스트를 포함하고 있는 블록은 단일 색을 가지는 배경 컬러의 비중이 높다. 이러한 특성은 텍스트 블록의 분류에서 중요한 판단 기준이 된다. 본 연구에서는 배경 컬러를 블록내의 whole-pixel 컬러를 기준으로 가장 많은 개수를 가진 컬러로 결정한다. 이 단계에서 가장 많은 컬러의 수가  $N_{th}$ 를 넘지 않는 경우에는 이미지 블록으로 결정하게 된다. 일반적으로 텍스트 문서는 단색의 배경을 가진다. 그림 3-5와 같이 하얀색 배경에 검은 색 글자를 가진 영역으로 구성된다. 이러한 문서 영상은 배경 비중이 다수를 차지하게 된다. 텍스트 블록에서 확인하면 글씨의 크기에 따라 배경이 차지하는 비중은 전체 영상에서의 배경 비중과는 차이를 보인다. 이러한 이유로 본 연구에서는 실험을 통하여 배경 비중의 값 ( $N_{th}$ )을 결정하였다.

그림 3-5와 같이 하얀색 배경의 텍스트 영상에서 텍스트를 포함하지 않는 영역의 비중은 100개의 텍스트 문서 기준으로 확인한 결과 그림 3-6과 같다. 그림 3-6에서 확인한 바와 같이 텍스트 영상의 배경 비중은 평균적으로 전체 영상에서 약 70% 정도 차지하게 된다. 하지만 블록기준의 배경은 블록 단위로 편차를 가지게 된다. 따라서 본 연구에서는  $N_{th}$  값을 다수의 실험을 통하여 선택하였다.



tory, Massachusetts Institute of Technology, Cambridge. In 1989, he joined the faculty of Purdue University, West Lafayette, IN, where he holds the rank of Professor with a primary appointment in the School of Electrical and Computer Engineering and a secondary appointment in the Department of Biomedical Engineering. His research focuses on the use of statistical image models, multiscale techniques, and fast algorithms in applications, including multiscale image segmentation, tomographic image reconstruction, image printing and rendering, and document segmentation and compression. His research has resulted in new methods for image rendering, halftoning, and display that have been widely used in commercial products.

Dr. Bouman is a Fellow of the American Institute for Medical and Biological Engineering (AIMBE), a Fellow of the society for Imaging Science and Technology (IS&T), a member of the SPIE professional society, a recipient of IS&T's Raymond C. Bowman Award for outstanding contributions to digital imaging education and research, and a University Faculty Scholar of Purdue University. He is currently the general Co-Chair of the SPIE/IS&T Symposium on Electronic Imaging, Secretary of the IEEE Biomedical Image and Signal Processing Technical Committee, and a member of the Steering Committee for the IEEE TRANSACTIONS ON MEDICAL IMAGING. He has been an Associate Editor for the IEEE TRANSACTIONS ON IMAGE PROCESSING and the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE. He has also been the Awards Chair for the ICIP 1998 organizing committee, Co-Chair of the SPIE/IS&T conferences on Visual Communications and Image Processing 2000 (VCIP), and a member of the IEEE Image and Multidimensional Signal Processing Technical Committee, a Vice President of Publications and a member of the Board of Directors for the IS&T Society, and he is the Founder and Co-Chair of the SPIE/IS&T conference on Computational Imaging.

그림 3-5 텍스트 문서



그림 3-6 텍스트 문서에서의 배경 픽셀 비중

$$\text{Color\_hist}_{\max} > N_{\text{th}} * \text{block\_x} * \text{block\_y} \quad (3-1)$$

본 연구에서는 실험 데이터를 기준으로 식(3-1)의  $N_{\text{th}}$ 를 0.3으로 결정하였다.  $N_{\text{th}}$ 가 0.3의 의미는 전체 컬러의 수에서 배경 컬러로 계산된 개수가 30%를 넘었음을 의미한다. 배경컬러의 비중이 0.3을 넘지 않으면 text 블록이 아닌 image 블록으로 판단한다. 배경컬러의 비율이 식(3-1)를 만족하는 블록에 대하여 텍스트 컬러를 추출하는 과정을 수행하게 된다.

Colorized 텍스트는 검은색 sub-pixel rendered text의 alpha blending 과정을 통하여 생성된다. Alpha blending[36] 은 검은색의 sub-pixel rendered text를 기준으로 배경 컬러와 변경하고자 하는 text 컬러의 조합으로 계산된다. 이러한 alpha blending의 기본 식은 식(3-2) 와 같다.

$$C_{\text{blended}} = C_{\text{font}} + C_{\text{base}} * (C_{\text{bg}} - C_{\text{font}}) / 255 \quad (3-2)$$

Alpha blending 은 RGB로 구성된 sub-pixel channel 단위의 독립된 컬러를 기준으로 수행된다. 식(3-2)를 살펴보면  $C_{\text{blended}}$  는 alpha blended 된 텍스트의 sub-pixel의 컬러 값을 의미하고,  $C_{\text{font}}$ 는 변경하고자 하는 컬러의 sub-pixel 컬러 값을 의미한다. 또한,  $C_{\text{base}}$ 는 검은색의 텍스트의 sub-pixel 컬러 값이고,  $C_{\text{bg}}$ 는 배경의 sub-pixel 컬러 값이 된다. 따라서 이러한 식을 기준으로  $C_{\text{base}}$ 는 텍스트 컬러가 위치한 중앙 부분에서 0이 되고, 배경에서는 255 값을 가지게 된다. 식(3-2)로부터  $C_{\text{blended}}$ 는  $C_{\text{font}}$ 부터  $C_{\text{bg}}$ 까지 값을 가

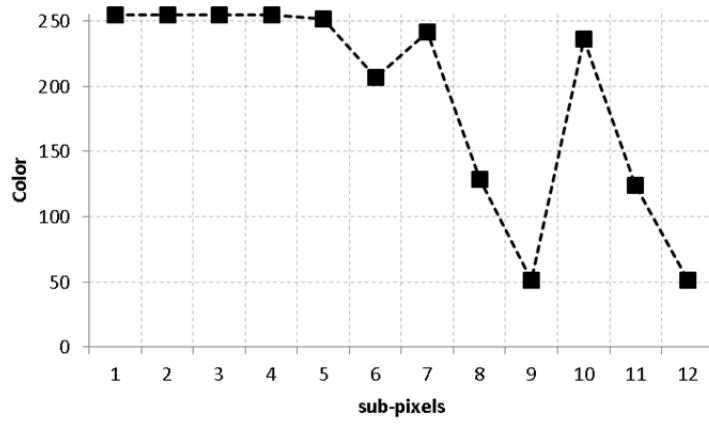
진다는 것을 알 수 있다. 따라서 블록내의  $C_{font}$  값은  $C_{bg}$ 로 부터 가장 차이가 큰 값을 가지는 것을 알 수 있다. 결과적으로 구하고자 하는  $C_{font}$ 는 블록 내에서 배경 컬러와 가장 차이가 많이 나는 sub-pixel 값으로 구한다. 또한  $C_{font}$ 는 0에서 255값을 가지기 때문에, 각 Channel에서  $C_{blended} - C_{bg}$ 는 한가지 부호만 가져야 한다는 것을 알 수 있다. 예를 들어  $C_{bg}$ 가 128일 때, 블록의  $C_{blended}$ 는 0~128의 값을 가지거나, 128~255의 값을 가져야 한다. 이를 만족하지 않는  $C_{blended}$  값이 나오지 않으면 image 블록으로 판단하게 된다.

### 3.2 Text De-colorization

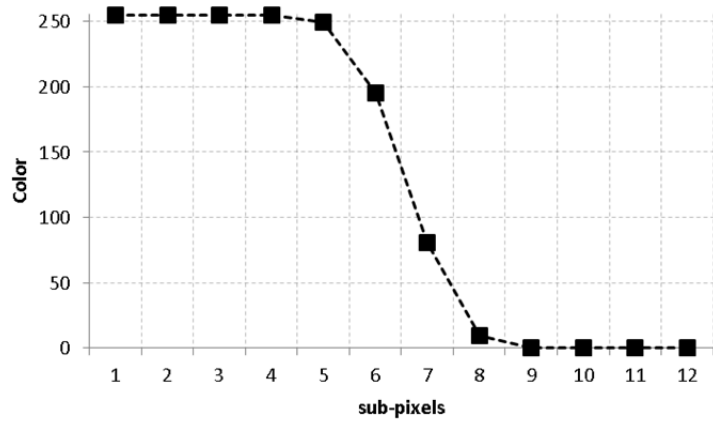
텍스트 de-colorization의 목적은 rendered text에서 나타나는 gradient의 경향성을 복원하기 위함이다. Colorized rendered text는 sub-pixel 단위의 gradient의 경향성을 확인하기 힘들기 때문에 colorized text를 위의 컬러 추출에 의한 텍스트의 컬러를 검은색의 텍스트로 바꾸게 되면 gradient의 경향성을 쉽게 확인할 수 있다. 이러한 gradient의 경향성은 블록을 구분하는 과정에서 사용된다. 텍스트 de-colorization 단계는 gradient의 경향성을 복원하는 과정에서 나타나는 경향성을 기준으로 텍스트와 이미지 블록을 구분하기 위함이다.

Sub-pixel 컬러의 gradient는 본 연구에서 설명하고 있는 sub-pixel gradient 분석 방법의 기본이 되는 입력이 된다. 그림 3-7은 colorized text가 이러한 de-colorization을 통하여 검은색의 text로 변환 되었을 때의 sub-pixel 컬러 값을 보여준다.

그림 3-7 (a) 와 같이 sub-pixel 컬러의 gradient는 일정한 방향성을 가지지 않는다. 실질적으로 sub-pixel 컬러의 gradient는 colorized된 영향에 따라 바뀌게 되는 것이다. 그림 3-7 (b) 와 같이 de-colorized 과정을 거치게 되면 일정한 방향성을 가지는 sub-pixel 컬러의 gradient를 가지게 된다. 이러한 gradient 변환 방식은 alpha blending의 역함수에 의해서 구할 수 있다. 식 (3-1)과 같이 Alpha blending은 검은색 텍스트로부터 colorized 텍스트로 변환되는 과정이다.



(a) Colored text



(a) Decolorized text

그림 3-7 de-colorization을 수행하기 전과 후의 text sub-pixel 컬러값

이러한 변환 과정을 반대로 파악을 하면 colored 된 텍스트를 검은색 텍스트로 변환할 수 있다. 식(3-3)은 gradient가 나타나는 검은색 텍스트를 구하기 위한 alpha blending 의 역함수이다.

$$C_{base} = 255 - 255 * (C_{bg} - C_{blended}) / (C_{bg} - C_{font}) \quad (3-3)$$

텍스트 컬러 추출과정에서 배경의 컬러와 텍스트의 컬러를 분리하는 과정을 진행하게 된다. 이렇게 분리 된 컬러를 기준으로  $C_{bg}$  와  $C_{font}$ 의 컬러 값을 확인 할 수 있다. Colorized 텍스트가 아닌 경우에는  $C_{base}$ 와  $C_{blended}$  는 동일한 값을 가진다. 따라서 현재 colored sub-pixel 값은 식(3-3)에 의해서  $C_{base}$ 으로 대치된다. 식(3-3) 에 의해서 텍스트 부분에 대하여 순차적으로 sub-pixel color 값은 대치된다. 이렇게 변환된 텍스트 블록을 기준으로 최종적으로 encoding 과정을 거치게 된다.

이러한 de-colorization 을 수행하게 되면 텍스트 컬러와 배경 컬러에서 다수의 gradient가 존재하던 텍스트에서 gradient가 단순해 짐을 알 수 있다. 이러한 de-colorization 방식은 흰색 배경색이 아닌 다른 배경색과 검은 색 텍스트가 아닌 텍스트의 조합을 가진 블록에서도 동일하게 적용된다

### 3.3 블록 분류 (Block Classification)

De-colored 된 sub-pixel 블록 영상은 최종적으로 gradient의 반복성을 판단하여 text와 image 블록으로 결정하게 된다. Text 는 그림 3-5 와 같이 sub-pixel 컬러 값이 순방향 gradient와 역방향 gradient의 형태를 반복하게 된다. 이는 단순한 배경 컬러와 텍스트 컬러 사이에 gradient 가 반복되어 나타나는 text 블록의 특성 때문이다. 이러한 경향성은 image 블록에 비해 text 블록에서 두드러지게 나타난다. Text 블록과 image 블록의 분류는 블록 단위로 이루어 지기 때문에 sub-pixel rendered 텍스트의 gradient의 형태는 안정된 구간 없이 local min/max 를 반복하게 됨으로 식 (3-4)을 만족해야 한다.

$$\sum_i^k D_{grad,i} < 2, \quad i = 1, 2 \dots k$$

$$\begin{cases} D_{grad} = -1, & \text{for negative direction} \\ D_{grad} = 0, & \text{for non\_gradient} \\ D_{grad} = 1, & \text{for positive direction} \end{cases} \quad (3-4)$$

$D_{grad}$ 는 각 gradient의 방향값으로 이전 sub-pixel 과 비교하여 순방향을 가지면 1, 역방향을 가지면 -1, gradient가 없으면 0을 나타낸다. gradient의 방향은 이전 sub-pixel color 값과의 크기 비교를 통하여 계산된다. k는 text 에서 나타나는 direction의 개수이다.  $D_{grad}$ 는 direction이 바뀔 때만 계산하게 된다. Text는 sub-pixel 단위의 순방향과 역방향을 반복하기 때문에 방향

이 바뀔 때 마다  $D_{grad}$ 를 더하면, 그 절대값이 1을 넘지 않아야 한다. 하지만 sub-pixel rendered text의 rendering 과정에서 rendering 이 아닌 다른 화질 개선 방법을 통하여 같은 방향의 gradient가 두 번 반복이 가능하기 때문에 본 연구에서 gradient의 절대값의 threshold는 2로 선택하였다. 결과적으로 식(3-4)를 만족하지 않으면 image 블록으로 결정하게 된다.

image 블록이 아닌 text 블록으로 결정되는 블록에서는 좀 더 자세한 분류 과정을 거치게 된다. 이는 다음절에서 설명하는 그룹 움직임 추정방법에서의 중요한 정보이다. 앞 장에서 설명한 바와 같이 텍스트 블록은 bi-level text와 simple rendered text, complex rendered text로 구분할 수 있다. Bi-level text 블록은 gradient를 구성하는 sub-pixel color의 개수가 1이 되면 bi-level 텍스트 블록으로 분류한다. gradient를 구성하는 sub-pixel color 개수가 4 이하가 되고, gradient의 시작점 혹은 끝 지점에서 non-linear 현상이 발생하지 않으면 simple rendered text 블록으로 분류한다. 최종적으로 gradient를 구성하는 sub-pixel color 개수가 4 이상이 되고, gradient의 시작점 혹은 끝 지점에서 non-linear 현상이 발생하는 text 블록을 complex rendered text 블록으로 분류하게 된다. 이러한 블록 분류 방식은 기존의 방법에서 구분하지 못하는 PDF 와 같은 복잡한 sub-pixel 컬러를 가진 텍스트 문서를 빠르고 정확하게 텍스트 블록으로 구분할 수 있다.

복잡한 텍스트 문서의 분류 방법은 문서의 가독성 유지를 위하여 필요하다. 텍스트 문서의 특성상 압축방식이 다른 블록에서는 블록간의 화질 차이로 인하여 가독성이 저하되는 현상을 가진다. Sub-pixel gradient 분석 방식은 이러한 문제점을 해결하고자 원본 컬러 데이터를 sub-pixel 단위로



분석하게 되면 컬러의 생성 단위로 분석이 가능하게 됨으로써 분류를 정확하게 할 수 있게 된다. 다음 절의 실험 결과를 통하여 본 연구에서 설명하는 sub-pixel을 사용한 블록 분류 방법의 결과를 확인 할 수 있다.

### 3.3 블록 분류 실험 결과

블록 분류에서 가장 중요하게 생각하는 부분은 텍스트로 판단되어야 할 블록이 이미지로 판단되는 비율을 최소화 하여야 한다는 것이다. 기본적으로 텍스트 블록의 화질은 이미지 블록의 화질보다 더 중요하다. 특히 텍스트 부분의 화질이 떨어지면 영상의 PSNR 뿐만 아니라 문자를 읽는 가독성 저하의 문제를 가진다. 텍스트 블록과 이미지 블록에 사용되는 압축방식이 다를 경우 이러한 현상은 도드라지게 되는 것이다. 따라서 본 연구에서는 이러한 블록 분류의 정확도에 대하여 실험을 진행 하였다. Bi-level text 영상, simple rendered text 영상, complex rendered text 영상과 natural 영상으로 분류하여 각 100개의 영상들에 대하여 실험을 진행하였다. 3가지 종류의 텍스트 이미지에서 텍스트로 판단되지 않는 블록의 수와, 1가지의 natural 영상에서 텍스트로 판단되는 블록의 수를 확인하였다. 설명하는 블록분류 알고리즘의 성능을 비교하고자 Lin[19]의 논문에서 사용한 컬러 수를 이용한 블록 분류 알고리즘과 Pan[21]의 논문에서 사용한 컬러 분포도 (histogram) 분류 방법을 비교하여 본 연구의 성과에 대한 결과를 확인 하였다.

그림 3-8는 실험에 사용한 4가지 종류의 이미지의 예이다. 그림3-8(a)와 같이 인터넷 영상에서 많이 사용하는 bi-level 이미지의 경우는 배경과 텍스트 컬러로 분리되는 이미지 이다. 그림 3-8(b)는 컬러의 종류가 단순한 simple rendered 텍스트를 사용한 이미지이다. 대부분의 Microsoft 의 프

로그래머에서 사용되고, 최근에 인터넷 영상에서도 사용되는 형태의 텍스트 영상이다. 본 연구에서는 Word 프로그램의 이미지와 인터넷 영상의 이미지를 혼합하여 테스트 영상으로 사용하였다. 그림 3-8(c) 와 같은 complex rendered 텍스트 영상은 PDF 와 같은 문서에서 사용된다. PDF 와 같은 문서는 simple rendered 텍스트 이미지와 다르게 감마 보정과 같은 여러 가지 가독성 향상 기법을 사용한 이미지로 위에서 설명한 바와 같이 sub-pixel 단위로 분석되는 컬러의 수가 자연 이미지의 컬러의 수와 비슷하다.

본 실험의 기준은 텍스트로 구성된 실험영상과 이미지로 구성된 실험영상을 기준으로 텍스트로 구분하는 비율을 통하여 본 연구의 성과를 확인하였다. 영상의 선택 기준은 텍스트 영상의 경우 2-color 로 구성되는 bi-level 영상, sub-pixel 컬러의 수가 비교적 단순한 simple rendered 텍스트, sub-pixel 컬러의 수가 복잡한 complex rendered 텍스트, 그리고 텍스트의 영역이 전혀 분포하지 않는 순수 이미지 영상으로 구성 하였다. 각 영상의 특성에 따라 각 알고리즘이 분석하는 결과가 다르게 나타난다. 영상의 경계 블록 보다 영상 내의 블록이 오차를 보일 경우 전체 영상의 가독성이 떨어지는 결과가 발생 할 수 있다.

264/AVC는 2003년 표준안 발표 이후, 여러번의 개선 및 기술 추가작업을 진행 하여 왔다.

Version 1: 2003년 5월 발표된 최초 출판 버전, Baseline, Main, Extended 의 세 가지 프로파일로 포  
Version 2: 2004년 5월 발표되었으며, 출판시의 몇 가지 오류에 대한 수정이 있었다.  
Version 3: 2005년 5월 작성되었으며, 기존 출판오류 수정 및 첫 번째 개정안으로서 FFE+TFidelity f  
Version 4: 2005년 9월 발표되었으며, 수렴안(compendium) 및 개정안(amenment)이 통합되었다.  
Version 5: FFE+T 프로파일 중 High 4:4:4 프로파일들이 제거되었다.  
Version 6: 2006년 6월 버전 5의 황제 발표되었으며, 역상 공간 지원 기능을 추가하였다.  
Version 7: 2007년 4월 발표되었으며, 개정안에 5개의 프로파일들이 포함되었다.(High 10 Intra, High 4  
VLC 4:4:4 Intra, High 4:4:4 Predictive)  
Version 8: 2007년 11월에 발표되었으며, 세 개의 스케일러블 프로파일(Scalable Baseline, Scalable  
Version 9: 몇 가지 오류에 대한 수정을 하여 2008년 1월 발표하였다.  
Version 10: 다중 카메라를 이용할 수 있는 Multiview Video Coding(MVC) 프로파일(Multiview High)  
Version 11: 버전 10과 황제 Constrained Baseline 추가하여 2009년 3월 발표하였다.  
낮은 (물만 이하의) 비트레이트에서 비슷하게 나 더 좋은 화질을 얻을 수 있도록 개발되었다. 그리고 구  
/한 종류의 네트워크와 시스템에서 작동하도록 하였다. (보기: 방송, DVD 저장, RTP/RTMP 및 인터넷  
결과 FFE+TFidelity Range Extensions)라고 부르는 표준을 기반한 확장률 발표 하였CK- 2007년 비  
같은) 또한 SVC(Scalable Video Coding) 모드를 추가로 지원하였으며(2007년 버전 8), 2009년 다  
대개 H.264/AVC 혹은 AVC/H.264, H.264/MPEG-4 AVC, MPEG-4/H.264 AVC라고 부른다. H.264  
에서 공통 개발된 MPEG-2 영상 코덱의 경우도 ITU-T 내부에서는 이를 H.262라고 부른다.)

(a) bi-level text 영상

The plea comes ahead of a landmark decision by the He  
on the future of major casualty units which are likely to be  
protected by local communities.

Medics say the plans are the only way to protect the safe  
and to put hospitals on a seven-day footing, with enough  
working at weekends.

Next month, Prof Sir Bruce Keogh, NHS England medica  
due to publish national proposals to improve the quality o  
emergency care and improve standards in care seven da

Senior officials say this will mean fewer larger hospitals,  
are sufficient specialists and senior staff in each, althoug  
services should still be available more widely.

(b) simple rendered text 영상

5. System prototyping

The algorithms, the proposed HW-SW partitioning  
interfacing and the system memory hierarchy definit  
have to be exhaustively verified and optimised before  
con implementation. Even if a SW simulation environm  
can be used for the co-verification of mixed HDL-C/C-  
system descriptions, a rapid prototyping approach has b  
preferred in our design flow. Due to the huge amount of c  
and operations, typical of image processing, a SW sim  
tion takes hours or days for few seconds of video sequer  
Simulating all the possible configurations of complex H  
SW architectures needs very long validation patterns, wh  
leads to prohibitive time with respect to typical consu



(c) complex rendered text 영상

(d) natural 영상

그림 3-8 분류 실험 영상

표 3-1 텍스트 블록 구분율(%)

실험영상	Bi-level Text	Simple Rendered Text	Complex Rendered Text	Natural image
Proposed	100	99.69	99.03	0.65
Lin[19]	100	98.32	29.54	5.10
Pan[21]	99.21	97.01	83.10	2.95

표 3-1은 각 알고리즘이 테스트 영상을 기준으로 텍스트 블록을 판단하는 분류율을 표시하였다. 전체 블록의 수를 100으로 하였을 때 텍스트 블록으로 판단하는 비율이 된다.

표 3-1와 같이 컬러의 수가 단순한 bi-level text의 경우 3개의 알고리즘 모두 높은 비율로 text 블록으로 구분하였다. 하지만 컬러의 조합이 작은 simple rendered text의 경우 제안하는 알고리즘은 99.69%의 구분률을 보인다.

Lan[19]의 결과에서처럼 컬러의 개수를 통하여 블록을 구분하는 방법도 bi-level text 와 simple rendered text에서 text블록을 구분하는 빠르고 좋은 방법이기도 하지만 natural 영상에서도 텍스트로 판단되는 영역이 5.10% 정도 발생하기 때문에 텍스트와 이미지 블록을 명확히 구분하기 어려운 점이 있다. 특히 complex rendered text의 경우 Lin[19], Pan[21] 의 알고리즘은 명확히 구분하지 못하는 경향이 있다. 제안하는 알고리즘의 경우 이러한 complex rendered text에서도 99.03% 의 구분률을 보인다. 특히 image 블록을 text블록으로 잘못 판단하는 비율이 실험 결과 0.65% 로 다른 알고리

즘에 비하여 image 블록과 text 블록을 정확히 구분 할 수 있음을 알 수 있다.

그림 3-9는 complex rendered text로 구성된 PDF 문서 영상을 기준으로 각 알고리즘이 분리되는 블록을 분리한 것이다. 회색으로 표현된 블록은 알고리즘 적용에 따라서 image 블록으로 판단된 것이다.

그림 3-9과 같이 complex rendered text의 경우 Pan[21] 알고리즘의 실험 결과는 텍스트 블록을 이미지 블록으로 잘못 판단하는 결과를 가져온다. 이 알고리즘은 컬러의 분포도를 확인하여 분류하기 때문에 PDF 와 같이 컬러의 분포도가 이미지와 차이가 없는 부분에서 차이를 보이게 되는 것이다. Color count를 사용하는 Lan[19]의 실험 결과를 보면 대부분의 블록이 image 블록으로 판단되었음을 알 수 있다. 본 연구에서 제안하는 블록 분류 방법은 PDF와 같은 complex rendered text 에서도 정확한 분류가 가능함을 알 수 있다.

이와는 반대로 image 에서 text 블록으로 판단되는 비중도 중요하다. 일반적으로 image를 압축하는 압축 방법과 text를 압축하는 압축 방법이 다르게 된다. 이럴 경우에 image안에서 서로 다른 압축방법이 사용되면 영상의 왜곡현상을 가져 올 수 있다.

그림 3-10는 이미지 영상을 기준으로 각 알고리즘이 분리되는 블록을 분리한 것이다. 회색으로 표현된 블록은 알고리즘 적용에 따라서 텍스트 블록으로 판단된 것이다. 그림 3-10 과 같이 Lan[19] 과 Pan[21] 의 알고리즘의 경우 이미지 영상에서 텍스트 블록으로 잘못 판단하는 경우가 나타나게 된다.

Image  $I$  is a mapping  $I : \mathcal{D} \subset \mathbb{Z}^2 \rightarrow \mathcal{S}$ . Extremal regions are well defined on images if:

1.  $\mathcal{S}$  is totally ordered, i.e. reflexive, antisymmetric and transitive binary relation  $\leq$  exists. In this paper only  $\mathcal{S} = \{0, 1, \dots, 255\}$  is considered, but extremal regions can be defined on e.g. real-valued images ( $\mathcal{S} = \mathbb{R}$ ).
2. An adjacency (neighbourhood) relation  $A \subset \mathcal{D} \times \mathcal{D}$  is defined. In this paper 4-neighbourhoods are used, i.e.  $p, q \in \mathcal{D}$  are adjacent ( $pAq$ ) iff  $\sum_{i=1}^4 |p_i - q_i| \leq 1$ .

**Region**  $Q$  is a contiguous subset of  $\mathcal{D}$ , i.e. for each  $p, q \in Q$  there is a sequence  $p, a_1, a_2, \dots, a_n, q$  and  $pAa_1, a_1Aa_2, \dots, a_nAq$ .

**(Outer) Region Boundary**  $\partial Q = \{q \in \mathcal{D} \setminus Q : \exists p \in Q : qAp\}$ , i.e. the boundary  $\partial Q$  of  $Q$  is the set of pixels being adjacent to at least one pixel of  $Q$  but not belonging to  $Q$ .

**Extremal Region**  $Q \subset \mathcal{D}$  is a region such that for all  $p \in Q, q \in \partial Q : I(p) > I(q)$  (maximum intensity region) or  $I(p) < I(q)$  (minimum intensity region).

**Maximally Stable Extremal Region (MSER)**. Let  $Q_1, \dots, Q_{i-1}, Q_i, \dots$  be a sequence of nested extremal regions, i.e.  $Q_i \subset Q_{i+1}$ . Extremal region  $Q_i$  is maximally stable iff  $q(i) = |Q_{i+\Delta} \setminus Q_{i-\Delta}| / |Q_i|$  has a local minimum at  $i^*$  ( $|\cdot|$  denotes cardinality).  $\Delta \in \mathcal{S}$  is a parameter of the method.

Table 1: Definitions used in Section 2

(a) 원본



(b) Lin[19]

Image  $I$  is a mapping  $I : \mathcal{D} \subset \mathbb{Z}^2 \rightarrow \mathcal{S}$ . Extremal regions are well defined on images if:

1.  $\mathcal{S}$  is totally ordered, i.e. reflexive, antisymmetric and transitive binary relation  $\leq$  exists. In this paper only  $\mathcal{S} = \{0, 1, \dots, 255\}$  is considered, but extremal regions can be defined on e.g. real-valued images ( $\mathcal{S} = \mathbb{R}$ ).
2. An adjacency (neighbourhood) relation  $A \subset \mathcal{D} \times \mathcal{D}$  is defined. In this paper 4-neighbourhoods are used, i.e.  $p, q \in \mathcal{D}$  are adjacent ( $pAq$ ) iff  $\sum_{i=1}^4 |p_i - q_i| \leq 1$ .

**Region**  $Q$  is a contiguous subset of  $\mathcal{D}$ , i.e. for each  $p, q \in Q$  there is a sequence  $p, a_1, a_2, \dots, a_n, q$  and  $pAa_1, a_1Aa_2, \dots, a_nAq$ .

**(Outer) Region Boundary**  $\partial Q = \{q \in \mathcal{D} \setminus Q : \exists p \in Q : qAp\}$ , i.e. the boundary  $\partial Q$  of  $Q$  is the set of pixels being adjacent to at least one pixel of  $Q$  but not belonging to  $Q$ .

**Extremal Region**  $Q \subset \mathcal{D}$  is a region such that for all  $p \in Q, q \in \partial Q : I(p) > I(q)$  (maximum intensity region) or  $I(p) < I(q)$  (minimum intensity region).

**Maximally Stable Extremal Region (MSER)**. Let  $Q_1, \dots, Q_{i-1}, Q_i, \dots$  be a sequence of nested extremal regions, i.e.  $Q_i \subset Q_{i+1}$ . Extremal region  $Q_i$  is maximally stable iff  $q(i) = |Q_{i+\Delta} \setminus Q_{i-\Delta}| / |Q_i|$  has a local minimum at  $i^*$  ( $|\cdot|$  denotes cardinality).  $\Delta \in \mathcal{S}$  is a parameter of the method.

Table 1: Definitions used in Section 2

(c) Pan[21]

Image  $I$  is a mapping  $I : \mathcal{D} \subset \mathbb{Z}^2 \rightarrow \mathcal{S}$ . Extremal regions are well defined on images if:

1.  $\mathcal{S}$  is totally ordered, i.e. reflexive, antisymmetric and transitive binary relation  $\leq$  exists. In this paper only  $\mathcal{S} = \{0, 1, \dots, 255\}$  is considered, but extremal regions can be defined on e.g. real-valued images ( $\mathcal{S} = \mathbb{R}$ ).
2. An adjacency (neighbourhood) relation  $A \subset \mathcal{D} \times \mathcal{D}$  is defined. In this paper 4-neighbourhoods are used, i.e.  $p, q \in \mathcal{D}$  are adjacent ( $pAq$ ) iff  $\sum_{i=1}^4 |p_i - q_i| \leq 1$ .

**Region**  $Q$  is a contiguous subset of  $\mathcal{D}$ , i.e. for each  $p, q \in Q$  there is a sequence  $p, a_1, a_2, \dots, a_n, q$  and  $pAa_1, a_1Aa_2, \dots, a_nAq$ .

**(Outer) Region Boundary**  $\partial Q = \{q \in \mathcal{D} \setminus Q : \exists p \in Q : qAp\}$ , i.e. the boundary  $\partial Q$  of  $Q$  is the set of pixels being adjacent to at least one pixel of  $Q$  but not belonging to  $Q$ .

**Extremal Region**  $Q \subset \mathcal{D}$  is a region such that for all  $p \in Q, q \in \partial Q : I(p) > I(q)$  (maximum intensity region) or  $I(p) < I(q)$  (minimum intensity region).

**Maximally Stable Extremal Region (MSER)**. Let  $Q_1, \dots, Q_{i-1}, Q_i, \dots$  be a sequence of nested extremal regions, i.e.  $Q_i \subset Q_{i+1}$ . Extremal region  $Q_i$  is maximally stable iff  $q(i) = |Q_{i+\Delta} \setminus Q_{i-\Delta}| / |Q_i|$  has a local minimum at  $i^*$  ( $|\cdot|$  denotes cardinality).  $\Delta \in \mathcal{S}$  is a parameter of the method.

Table 1: Definitions used in Section 2

(d) Proposed

그림 3-9 complex rendered text의 text 블록 판단 결과



(a) 원본



(b) Lan[19]



(c) Pan[21]



(d) Proposed

그림 3-10 natural 영상에서의 text 블록 판단 결과



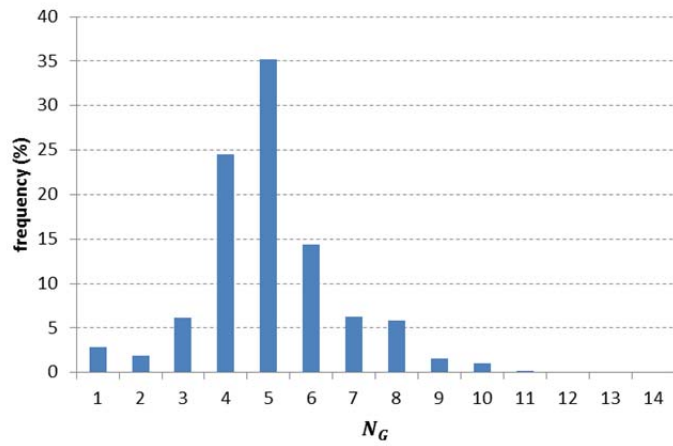
Lan[19] 알고리즘의 경우 분류기준이 컬러의 수를 32이 이하의 경우 텍스트, 컬러의 수가 32를 넘는 경우는 이미지로 분리하게 된다. 이러한 분류 방법의 경우 단순한 텍스트 블록은 분류가 가능하지만, 컬러의 수가 많은 complex rendered 블록에서는 정확한 분류를 할 수 없다. 또한 이미지 영역 또한 평이한 컬러를 가지는 블록을 텍스트 영역으로 잘 못 분류하게 되는 오류를 가진다. 정확한 블록 분류는 이미지와 텍스트영역에서 서로 다른 압축 방법을 사용하는 compound image 압축방식에서는 블록간의 화질을 결정하는 중요한 요소이다. sub-pixel gradient 블록 분류 방법은 text와 image 블록을 정확히 구분할 수 있음을 확인할 수 있다.

## 제4장 Sub-pixel Gradient text 블록 코딩 방법

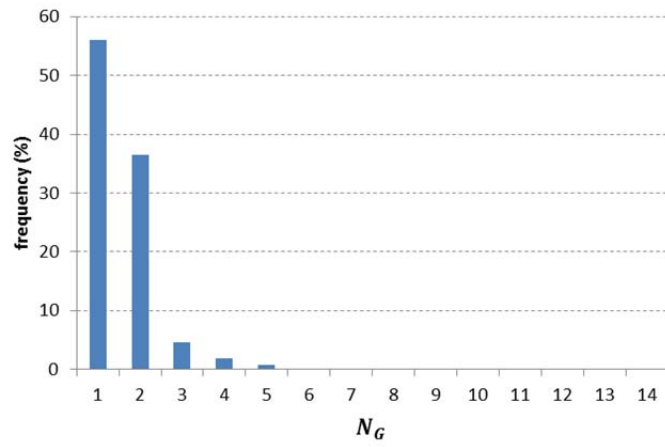
4장에서는 3장에서 분류된 블록을 기준으로 텍스트 블록의 압축 방법에 대해서 설명하고자 한다. Text 블록의 압축은 텍스트 영상의 특성상 가독성과 압축률이 중요한 조건이다. 압축률과 가독성은 trade-off 관계이기 때문에 압축률이 높을수록 가독성이 떨어지게 된다. 특히 H.264/AVC 나 JPEG과 같은 양자화 영상 변환 과정을 거치는 압축 알고리즘은 텍스트 영상에서 자연 영상에 비해 시각적인 화질 저하가 많이 일어나게 되는 것이다. 본 연구에서는 이러한 가독성 저하를 막고, 높은 압축률을 가질 수 있는 Sub-pixel gradient를 사용한 text 블록 압축방식을 설명한다.

Sub-pixel rendered text는 그림 4-1(a)와 같이 다양한 컬러의 조합으로 이루어진다. 그림 4-1(b)는 그림 4-1(a)의 sub-pixel 컬러 값을 나타낸 것이다. 그림 4-1(a)는 배경의 색이 255로 구성된 흰색 바탕이 되고, 텍스트 컬러는 0으로 구성된 검은색의 텍스트로 구성되어 있다. 하지만 sub-pixel rendering을 거치면서 그림 4-1(a)와 같은 다양한 컬러의 조합으로 구성되는 것이다. 그림 4-1(c)는 그림 4-1(b)의 한 pixel 라인을 기준으로 sub-pixel 컬러 값을 그래프로 나타내었다. 그림 4-1(c)와 같이 sub-pixel rendered text는 텍스트와 배경 사이에서 sub-pixel 컬러로 구성되는 gradient를 형성한다. 이러한 gradient는 시작 지점으로부터 끝나는 지점까지 선형 관계를 유지한다.





(a) Complex rendered text image



(b) Natural image

그림 4-2 gradient를 형성하는 sub-pixel 컬러 수

그림 4-2와 같이 gradient를 형성하는 sub-pixel의 수는 rendered text에서 일정하게 나타난다. 이러한 sub-pixel gradient 수는 텍스트의 컬러와 배경색의 컬러 사이에서 rendering 된 영향으로 확인할 수 있다. 하지만 simple rendered text와 complex rendered text의 경우 위에서 설명한 바와 같이 rendering 의 방식의 차이로 인하여 각 텍스트의 특성에 따른 다양한 gradient의 개수를 보여준다. 특히 image 블록에서는 gradient가 일정하게 증가하거나 감소하는 패턴이 반복되지 않고, gradient의 연속성이 떨어지게 된다.

본 연구에서 설명하는 text 블록 코딩 방법은 블록 분류방법에 의해 분류된 text 블록에 대하여 진행된다. 블록 내의 코딩 동작은 RGB 의 픽셀 단위를 구분하지 않으며 sub-pixel 단위의 3개의 channel을 독립적으로 판단한다. 또한 3-channel sub-pixel의 컬러의 순서는 RGB를 기준으로 진행된다.

Text 블록 코딩은 블록 분류 방법에서 수행한 de-colored text를 기반으로 이루어진다. 이는 본 연구가 제안하는 text 블록 코딩 방식이 sub-pixel 기반의 gradient의 선형 특징을 이용하기 때문이다. text 블록 코딩을 하기 전에 블록 분석 단계로 압축효율을 높이고자 gradient 보정과정 (gradient fitting process)을 진행한다. gradient 보정과정을 거친 데이터를 기준으로 gradient 코딩을 진행한다. 그림 4-3은 이러한 Text 블록 코딩 방법 진행과정이다.

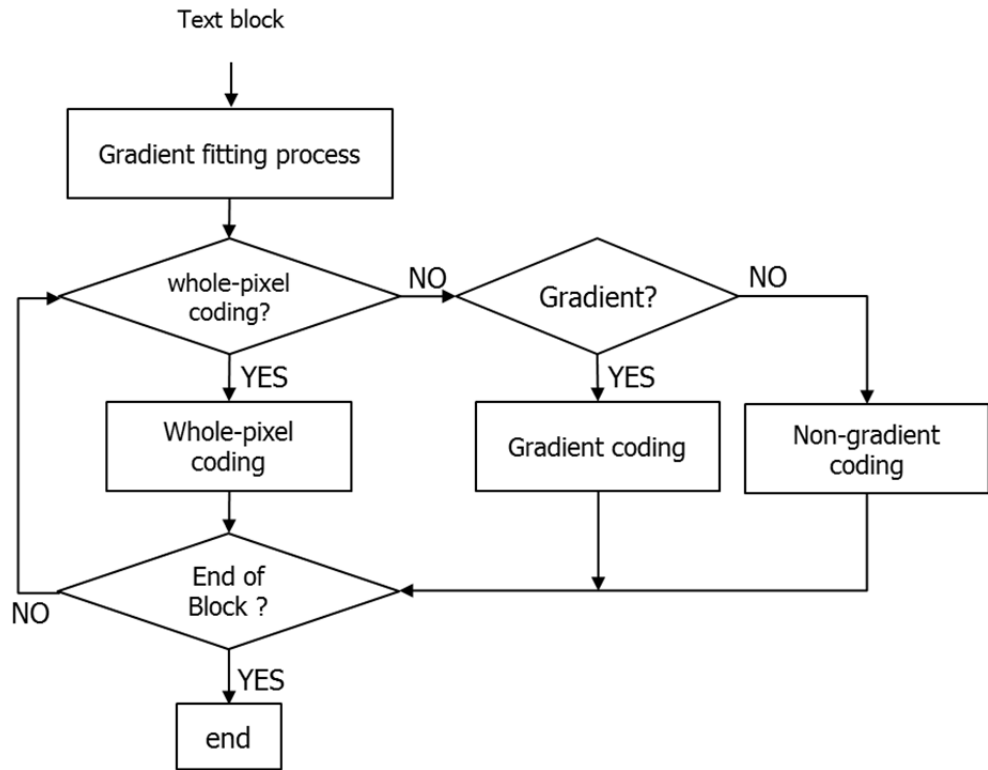


그림 4-3 Text 블록 코딩 방법

## 4.1 Gradient fitting process

Sub-pixel rendered 텍스트의 가독성을 유지하기 위하여 decode되는 데이터는 원본 텍스트와 gradient의 오차율이 최소화 되어야 한다. 그러나 text를 gradient의 개수만으로 encoding 할 경우, decoding 과정에서 원본 텍스트와 오차율이 증가하게 된다. 이는 sub-pixel rendered 텍스트가 생성되는 과정에서 비선형 원인이 사용되었기 때문이다. gradient의 비선형 부분은 주로 gradient의 시작점 혹은 끝부분의 gradient가 이를 구성하는 평균적인 gradient와 많이 다를 경우, 이를 텍스트 압축을 진행하기 전에 미리 제거하여 gradient의 선형성을 유지하도록 한다. 이렇게 함으로써 텍스트의 압축효율을 높일 수 있다. 식(4-1)은 gradient의 시작 위치 또는 끝나는 위치의 sub-pixel 컬러 값을 코딩에서 제거하는 방법을 나타낸다.

$$\begin{cases} P_i = C_s, & \text{if } \text{abs}(P_i - C_s) < FT \\ P_i = C_f, & \text{if } \text{abs}(P_i - C_f) < FT \end{cases}, i = 0, 1, \dots, N_G - 1 \quad (4-1)$$

$$FT = \text{abs}(C_s - C_f) / \text{fitting\_value}$$

$P_i$ 는 gradient를 이루는 sub-pixel 컬러 값이며, FT는 보정 threshold 값을 말한다.  $C_s$ 는 gradient의 시작지점의 sub-pixel 컬러 값이며  $C_f$ 는 gradient의 끝 지점의 sub-pixel 컬러 값이다. gradient를 이루는 각 sub-pixel 컬러 값이  $C_s$  혹은  $C_f$ 와 값 차이가 FT 이하일 경우, 이를  $C_s$  혹은  $C_f$ 의 값으로

로 바꾼다. FT를 결정하는 fitting 값은 영상의 quality에 따라 다르다. 보정 값의 조정은 뒤에서 설명할 텍스트 압축의 영상 화질 enhancement 에서 설명하도록 한다.

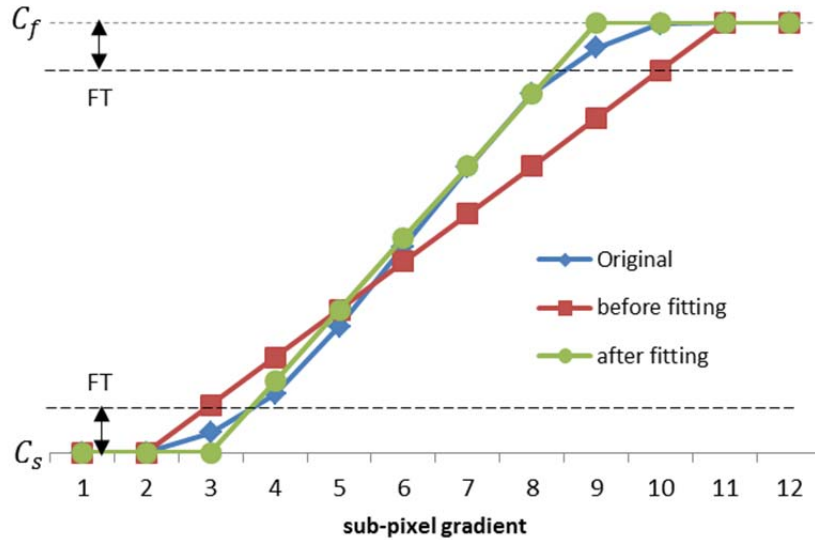


그림 4-4 Gradient fitting process

그림 4-4는 텍스트 압축에서 gradient fitting process의 효율성을 설명하기 위한 그림이다. 그림 4-4의 original 로 표시된 라인은 sub-pixel rendered 텍스트의 gradient 부분으로 시작점과 끝점에서 gradient의 경향성이 다르게 나타나는 것을 확인할 수 있다. 그림 4-4의 before fitting으로 표시된 라인은 gradient fitting process를 적용하지 않았을 때 decoding 된 결과를 표시한다. gradient의 시작부분의 경향성이 중간부분의 gradient 와 다르기 때문에 gradient 를 sub-pixel 단위로 encoding 하였을 때 decoding 된 결과가 원본



데이터와 차이를 가지는 것을 알 수 있다. 그림 4-4에서 after fitting 으로 표시된 라인은 식 (4-1) 를 이용하여 경향성이 다른 sub-pixel gradient 의 시작점과 끝점을 encoding 단계에서 제외시켜 encoding을 진행한 결과를 보여 준다. gradient 의 시작점과 끝점의 sub-pixel 컬러 값이  $C_s$ 와  $C_f$  로 변경된 후에 gradient 가 원본 데이터와 유사하게 decoding 되는 것을 확인할 수 있다.

표 4-1 fitting threshold

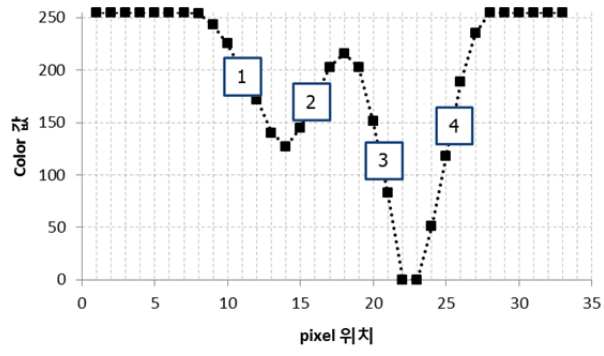
Fitting threshold	Function : $abs(C_s - C_f) / fitting\_value$	Value
FT1	$abs(255-127) / 8$	16
FT2	$abs(127-216) / 8$	11
FT3	$abs(216-0) / 8$	27
FT4	$abs(0-255) / 8$	32

그림 4-5와 같이 gradient 보정과정은 3단계 과정으로 구성된다. 첫 번째 단계로 sub-pixel로 구성된 텍스트에서 각 gradient를 구성하는 sub-pixel을 그룹화하는 단계이다. gradient를 구성하는 sub-pixel은 gradient의 시작지점 부터 gradient의 끝 지점까지를 포함한다. gradient가 시작되는 지점은 sub-pixel 의 컬러 값이 일정하게 유지되다가 값이 바뀌는 지점과 gradient의 방향이 바뀌는 지점이다. gradient가 끝나는 지점은 gradient의 방향이 바뀌

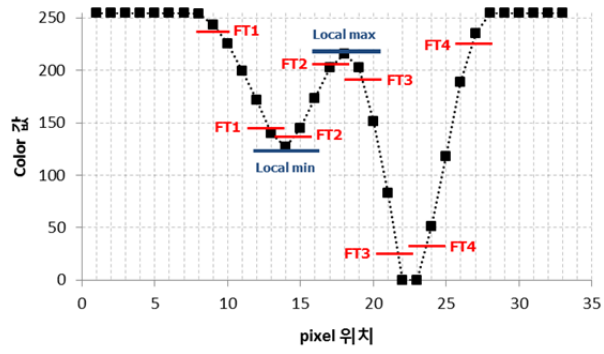
거나 sub-pixel 컬러 값이 일정하게 유지되는 형태로 바뀌는 지점이다. 이렇게 각 gradient를 분리하는 이유는 gradient fitting process가 각 gradient마다 독립적으로 이루어지기 때문이다. 그림 4-5(a)는 sub-pixel rendered 텍스트의 한 라인이 4개의 sub-pixel 그룹으로 분리된 것을 보여준다. 각 그룹은 gradient의  $C_s$ 와  $C_f$  정보를 가지고 있게 된다.

두 번째 단계는 각 sub-pixel 그룹에 대하여 보정 threshold 값을 구하는 단계이다. 첫 번째 단계에서 구한  $C_s$ 와  $C_f$ 를 이용하여 식(4-1)로 FT를 구하게 된다. 보정 값이 8일때 각 그룹에서의 FT 값은 표 4-1와 같다.

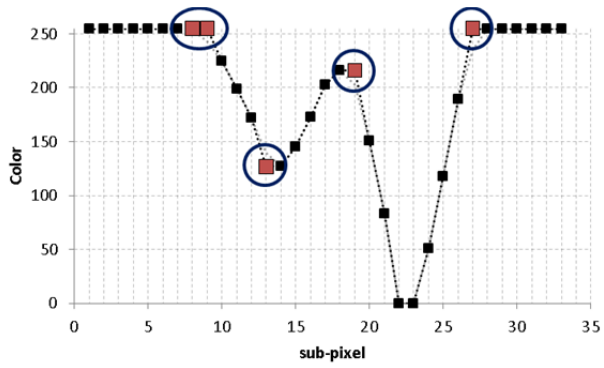
세 번째 단계는 보정 threshold를 기준으로 sub-pixel의 컬러 값을 local min/max값으로 변경하는 단계이다. 보정을 위한 실제 FT의 위치는  $C_s$ 와  $C_f$ 에서 FT 만큼 떨어진 위치이다. 그림 4-5(b)는 각 그룹별 FT의 위치를 나타낸다. 그림 4-5(b)과 같이 FT는 그룹마다 gradient 안으로  $C_s$ 와  $C_f$ 에서 FT 만큼 떨어진 곳에 존재한다. 이를 기준으로 FT와  $C_s$ 와  $C_f$  사이의 sub-pixel 컬러 값은  $C_s$ 와  $C_f$ 의 값으로 바뀌게 된다. 그림 4-5(c)는 최종적으로 보정과정을 거친 결과이다. 보정과정을 적용하기 전 그림 4-5(a)와 비교하여 gradient의 선형성을 유지하는 것을 확인 할 수 있다



(a)



(b)



(c)

그림 4-5 gradient fitting process의 동작

## 4.2 Text Coding

텍스트 코딩은 sub-pixel gradient를 이용하여 코딩하는 과정이다. 이러한 코딩과정을 통하여 텍스트의 가독성 유지와 압축률 향상을 꾀할 수 있다.

### 4.2.1 Gradient로 구성된 부분의 코딩방법

gradient 코딩은 기본적으로 sub-pixel rendered 텍스트에서 sub-pixel gradient의 선형성을 이용하여 gradient를 구성하는 sub-pixel 의 개수로 코딩 하는 방법이다. 텍스트를 이루는 gradient는 gradient 보정과정을 통하여 선형성을 유지하기 때문에 이를 단순히 gradient를 구성하는 sub-pixel 의 개수로 코딩 할 수 있다. 반대로 decoding 과정은 gradient의 픽셀 수를 이용하여 gradient의 선형성을 복원하게 된다. 이러한 gradient는 식(4-2) 로 표현된다.

$$G_N = \begin{cases} (C_s - C_f)/N & \text{for negative gradient} \\ (C_f - C_s)/N & \text{for positive gradient} \end{cases} \quad (4-2)$$

$C_s$  는 encoding을 진행할 때, gradient의 시작지점의 sub-pixel 컬러 값이고,  $C_f$ 는 gradient의 끝 지점의 sub-pixel 컬러 값이다.  $N$ 은 gradient의 시작지

점부터 끝 지점까지의 sub-pixel 컬러의 개수다. N값의 범위는 gradient 보정 과정을 통하여 정해진다. Decoding 과정에서 식(4-2)의  $C_s$ 와  $C_f$ 를 확인하면 텍스트의 sub-pixel gradient를 복원할 수 있다.

#### 4.2.2 Gradient가 없는 부분의 코딩방법

텍스트의 sub-pixel 컬러 값은 텍스트의 중심부나 배경 부분에서 동일한 값을 가지는 확률이 높다. 이러한 gradient가 없는 부분은 gradient가 있는 부분과 특성이 다르고 단순하다. 따라서 효율적인 코딩을 위하여 gradient가 존재하는 부분과 존재하지 않는 부분을 구분하여 코딩 한다. 텍스트 블록 내의 픽셀라인 단위로 진행하면서 gradient가 존재하지 않는 부분은 1bit를 할당하여 코딩 한다. 현재의 sub-pixel의 컬러 값과 이전 sub-pixel의 컬러 값이 다르게 되면 gradient가 발생한 것으로 확인하고 gradient의 값을 코딩 하게 된다.

#### 4.2.3 local min/max 값 예측

Encoding된 데이터를 decoding 단계에서 복원하기 위하여 pixel의 개수뿐만 아니라 식(4-2)와 같이  $C_s$ 와  $C_f$ 의 값을 확인하여야 한다. 그러나 텍스트의 특성상 gradient의 시작점 혹은 끝점은 배경 이거나 문자의 컬러일 확률이 높다.

표 4-2  $C_s$  와  $C_f$  가 배경과 문자 컬러일 확률

Rendered text의 종류	Probability (%)
Simple rendered text	89
Complex rendered text	61

표 4-2와 같이 sub-pixel rendered 텍스트에서 simple rendered 텍스트뿐만 아니라 complex rendered 텍스트에서도  $C_s$ 와  $C_f$ 의 값은 배경이거나 문자의 컬러로 구성될 확률이 높다. 따라서  $C_s$ 와  $C_f$ 의 값은 텍스트 컬러 또는 배경 컬러로 예측하고 배경 컬러나 텍스트 컬러가 아닐 경우에만 실제 컬러 값을 코딩 한다. Decolorized text의 문자 컬러는 0이고, 배경 컬러는 255 이므로 역방향 gradient에 대해서는 0으로 예측하고 순방향 gradient에 대해서는 255로 예측하게 된다. 또한 텍스트의 gradient는 대부분 순방향과 역방향을 반복하기 때문에 gradient의 방향에 대한 정보도 코딩 하지 않는다. 결과적으로  $C_s$ 와  $C_f$ 에 대한 비트 및 방향에 대한 비트를 최소화 함으로써 높은 압축 효율을 가지게 한다.

#### 4.2.4 Whole-pixel 코딩

Sub-pixel 단위의 압축 방법은 sub-pixel rendered 텍스트의 gradient를 압축할 때 가장 효율적이다. 하지만 배경과 같은 단순한 패턴의 반복에서는 비효율적일 수 있다. 텍스트의 배경은 sub-pixel 컬러 값의 변화가 없고 단순하기 때문에 sub-pixel 컬러 단위의 코딩 방식은 불필요한 비트 수를 늘리게 된다. 따라서 본 연구에서는 sub-pixel 코딩방법과 함께 배경이나 문자의 컬러의 반복성을 이용하여 RGB 3개 channel의 데이터를 한꺼번에 코딩 하는 whole-pixel 코딩방법을 병행한다. Whole-pixel 코딩 방법은 whole 픽셀을 구성하는 sub-pixel 의 값이 모두 동일 할 경우 이를 1bit로 할당하여 코딩 하는 방법이다. 텍스트는 배경의 비율이 높으며 de-colored된 텍스트의 경우 문자의 whole 픽셀의 값이 모두 동일 할 경우가 많다. 결과적으로 배경과 문자의 동일한 sub-pixel 의 값을 whole-pixel 코딩 하여 압축률 향상에 기여할 수 있다.

#### 4.2.5 화질 enhancement

gradient 코딩 방법은 sub-pixel rendered 텍스트의 비선형 특징을 모두 재현해 내기에는 한계를 가지게 된다. gradient 보정 과정과 텍스트 gradient를 하나의 gradient로 압축하는 과정에서 비선형 요소들의 손실이 증가하게 된다. 따라서 본 연구에서는 텍스트 gradient를 구성하는 일부 sub-pixel 컬러 값들을 코딩 하여 텍스트의 비 선형성을 보완하는 방법을 설명한다. 먼저 위에서 설명한 gradient를 구성하는 sub-pixel의 개수를 코딩 한다. 이후에 비선형 특징을 보완하기 위하여 gradient를 구성하는 sub-pixel 컬러 값을 일부 추가적으로 코딩 한다. gradient를 구성하는 sub-pixel 중 하나의 sub-pixel 값을 추가적으로 코딩 한다면, 먼저 gradient의 개수를 코딩하고 gradient를 구성하는 첫 번째 sub-pixel 컬러 값을 코딩 한다. Decoding 과정에서는 gradient의 개수와 첫 번째의 sub-pixel 컬러 값을 복원한 후 나머지 부분을 직선으로 이어 gradient를 복원한다.

그림 4-6 은 gradient를 구성하는 sub-pixel 컬러 값을 추가적으로 코딩 하였을 때와 하지 않았을 때 gradient의 복원 결과를 비교한 것이다.  $N_m$ 은 gradient를 구성하는 sub-pixel 컬러 값을 코딩 한 개수이다. gradient 코딩만 수행할 때 보다 gradient 내부의 sub-pixel 을 하나 더 코딩 하였을 때 복원 결과가 원본 데이터에 근접한 것을 알 수 있다.



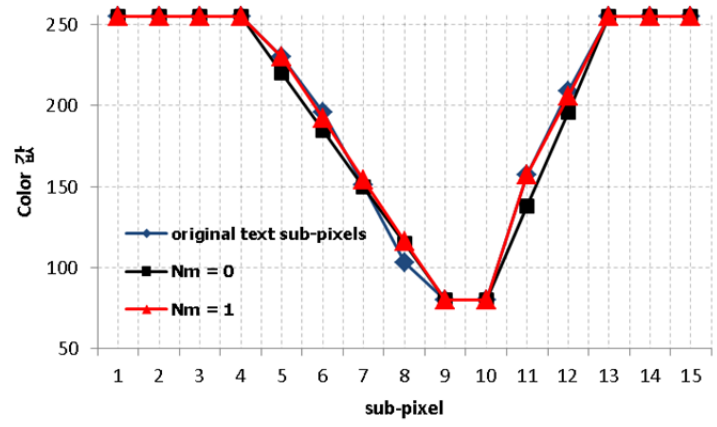


그림 4-6  $N_m = 1$  인 경우와  $N_m = 0$  인 경우

표 4-3 과 같이 위의 방법을 적용하였을 때 오차율이 감소한 것을 알 수 있다. 위와 같은 방법은 sub-pixel rendered 텍스트의 특성을 복원하는 방법이지만, 이전의 gradient 보정 과정에서 많은 부분의 비선형성이 제거 되기 때문에 텍스트 복원에 한계가 존재한다. 따라서 본 연구에서는  $N_m$ 에 따라 gradient 보정과정에서 보정 threshold를 조절한다. 본 연구에서  $N_m$ 에 따른 보정 threshold 값은 표4-4에 나타나 있다.

표 4-3  $N_m$  값의 변화에 따른 방법의 오차

Gradient 내부 sub-pixel coding 개수	Error : $\sqrt{\sum(P_{ori,i} - P_{decoded,i})}$
$N_m = 0$	8.1
$N_m = 1$	4.8

표 4-4  $N_m$ 에 따른 보정 값

$N_m$	fitting value
0	8
1	8
2	16
3	16
else	32

gradient를 구성하는 sub-pixel 컬러 값을 그대로 코딩 하게 되면 많은 량의 비트가 소모 된다. 텍스트의 gradient를 구성하는 sub-pixel은 gradient의 시작지점과 끝 지점의 사이의 값만을 가질 수 있는데, 이러한 성질을 이용하게 되면 sub-pixel 컬러 값 코딩 시 높은 효율을 가질 수 있다. 따라서 본 연구에서는 이러한 성질을 이용하여 적은 비트 량으로 최대한 근사한

sub-pixel 컬러 값을 코딩 하는 방법을 사용한다. gradient를 구성하는 sub-pixel 컬러 코딩은 식 (4-3)과 같이 진행한다.

$$\frac{abs(P_i - P_{i-1})}{abs(C_s - C_f)} * B_G, i = 0, 1, \dots, N_m - 1 \quad (4-3)$$

$N_m$ 은 gradient를 구성하는 sub-pixel 컬러 값을 몇 개까지 코딩을 할 것 인지를 나타낸다.  $B_c$ 는 기본 gradient 값으로 이 값이 커질수록 코딩 되는 gradient의 정확성이 높아진다. Decoder에서의 sub-pixel 복원과정은 식 (4-3)에 의하여 코딩 된 sub-pixel 값을  $B_c$ 의 값으로 나누고  $C_s - C_f$ 의 절대값을 곱하여 gradient를 구성하는 sub-pixel을  $N_m$  개수만큼 생성하고, gradient를 구성하는 나머지 sub-pixel 들은 선형성을 이용하여 복원한다.  $N_m$ 과  $B_c$  값이 커질 수록 gradient를 세밀하게 코딩 하기 때문에, 화질은 높아지게 된다. 하지만 반대로 압축률이 낮아지기 때문에 선택되는 영상의 화질 목표에 따라서 조절할 수 있다.

### 4.3 텍스트 코딩 동작

표 4-5 동작 예시에 사용된 text coder parameter

Parameter	Value
$N_m$	1
Bit size for inner sub-pixel in the gradient	4
Bit size for gradient length	4

표 4-5는 본 절에서 동작 예시를 위하여 사용된 parameter이다.  $N_m$ 은 화질 enhancement를 위하여 사용되는 parameter로  $N_m=1$ 은 gradient를 구성하는 sub-pixel중 하나를 추가적으로 코딩을 진행하는 의미이다. 두 번째 parameter는 gradient를 구성하는 sub-pixel 컬러 값을  $N_m$ 개 코딩 할 때 각각 4bit으로 코딩 한다는 것을 의미한다. 세 번째 parameter는 gradient의 길이를 코딩 할 때 몇 비트로 코딩을 진행 할지를 정하는 parameter로, gradient 길이는 4 bits으로 코딩 한다. 본 절에서는 이러한 코딩 동작을 단계적으로 설명하기로 한다.

### 4.3.1 텍스트 코딩 입력

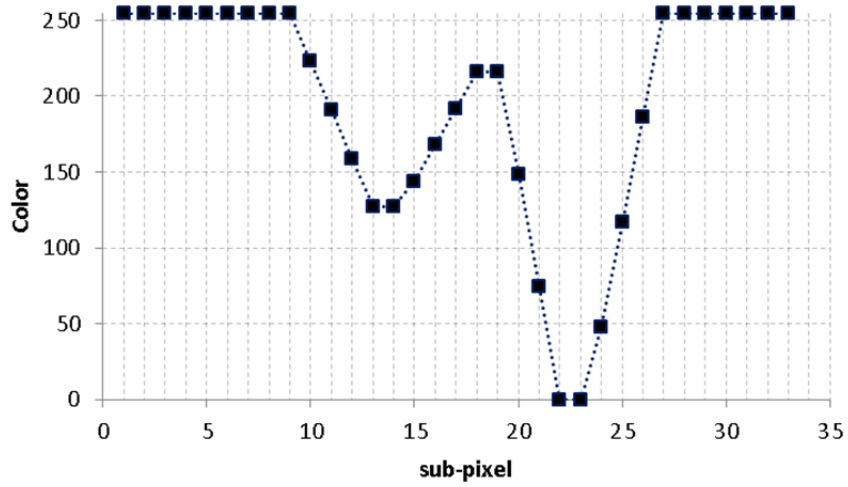


그림 4-7 텍스트 코딩 입력

그림 4-7 은 gradient 보정 과정에 의해 decolorized text의 한 라인이다. Decolorized 된 텍스트는 검은색 컬러의 텍스트와 하얀색 컬러의 배경으로 구성되게 된다. 3장에서 설명한 바와 같이 decolorized text는 텍스트 원본을 기준으로 sub-pixel rendering 되어 있으므로 텍스트의 sub-pixel gradient가 정확하게 표현되어 있다.

### 4.3.2 Whole-pixel 코딩 1

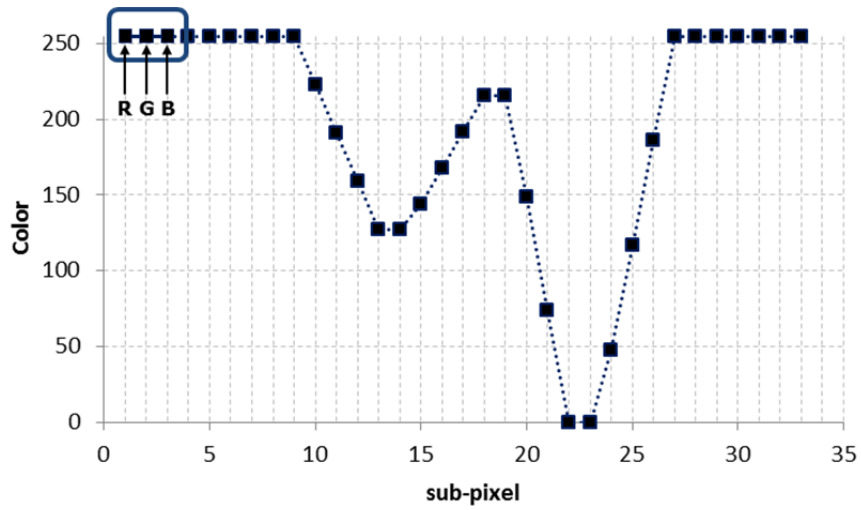


그림 4-8 Whole-pixel 코딩 1

그림 4-8과 같이 첫 번째 sub-pixel은 R channel 시작 지점이고, 연속된 세 개의 sub-pixel (whole-pixel)값이 모두 같으므로, whole pixel 단위로 압축된다. 압축된 비트는 1 (1bit)이다. 그 후 여섯 개의 sub-pixel도 모두 같은 값을 가지므로 whole pixel 코딩 방법으로 압축된다. 생성되는 bit stream 은 1 1 1이다.

### 4.3.3 역방향 Sub-pixel gradient 코딩 1

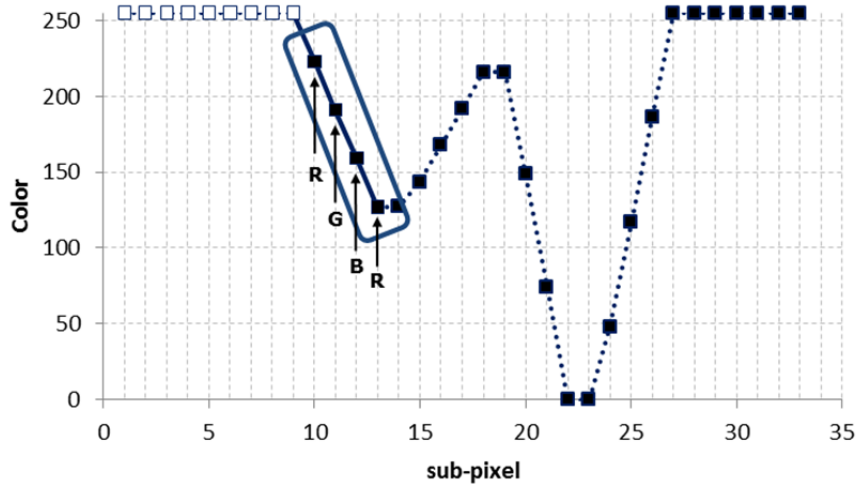


그림 4-9 역방향 Sub-pixel gradient 코딩 1

그림 4-9와 같이 R channel 시작지점에서 코딩 하지만, 연속된 세 개의 sub-pixel 값이 같지 않기 때문에, sub-pixel 단위로 압축하며 0 (1bit)으로 코딩 한다. 첫 번째 sub-pixel의 값이 이전 sub-pixel의 값과 다르므로 gradient 코딩을 수행하게 되며, 이에 따른 index bit는 0 (1bit)이다. 순차적으로 gradient를 구성하는 sub-pixel의 개수를 코딩 한다. 그림 4-9와 같이 gradient를 구성하는 sub-pixel의 개수가 4 이므로, 0100의 4bit으로 coding한다. 화질 enhancement를 위하여 gradient 내부를 코딩 하는 변수  $N_m$  이 1 이므로, gradient를 구성하는 첫 번째 sub-pixel 코딩 한다.  $C_s$ 는 255,  $C_f$ 는 127,  $P_1$ 은 225으로, 식 (4-3)에 따라  $\frac{abs(255-225)}{abs(255-127)} * 16 = 4$ 이고 따라서 코딩

되는 비트는 0100 (4bit)이다. 역방향 gradient 지만  $C_f$  값이 0 (text color)가 아니므로 컬러 비트를 추가적으로 코딩 한다. Index bit으로 0 (1bit)을 코딩 하고, 컬러 값으로 01111111 (8bit)을 코딩 한다. 생성되는 bit stream은 0 0 0100 0100 0 01111111이다. 누적 bit stream은 1110 0010 0010 0001 1111 11 이다.



#### 4.3.4 Local minimum 코딩 1

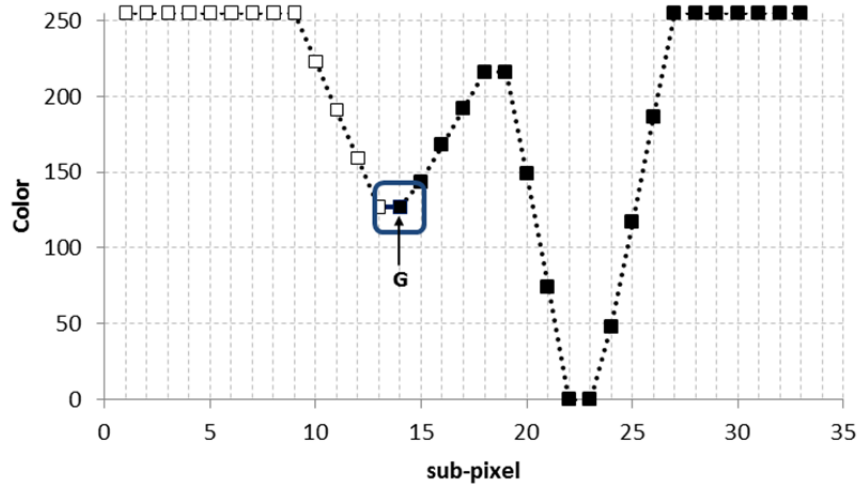


그림 4-10 Local minimum 코딩 1

그림 4-10과 같이 코딩을 시작하는 지점이 G channel의 시작지점이다. R channel이 코딩의 시작지점이 아니기 때문에 sub-pixel 코딩을 진행 한다. 현재의 sub-pixel이 이전 sub-pixel의 값과 같기 때문에, non-gradient 코딩을 한다. 코딩 되는 비트는 1 (1bit)이다. 생성되는 Bit stream은 1이다. 누적 bit stream은 1110 0010 0010 00 01 1111 111 이 된다.

### 4.3.5 순방향 gradient 코딩 1

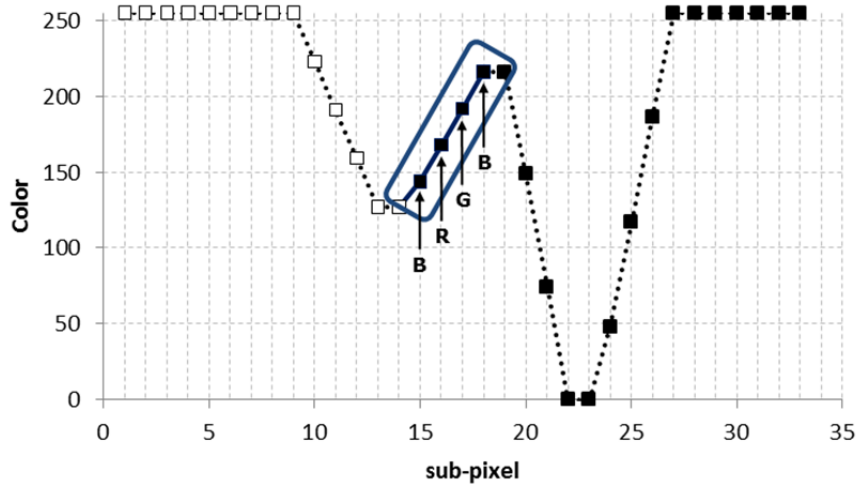


그림 4-11 순방향 gradient 코딩 1

그림 4-11과 같이 코딩을 시작하는 지점이 B channel의 시작지점이다. R channel이 코딩의 시작지점이 아니기 때문에 sub-pixel 코딩을 한다. 첫 번째 sub-pixel의 값이 이전 sub-pixel의 값과 다르므로 gradient 코딩을 수행하게 되며, 이에 따른 index bit는 0 (1bit)이다. gradient를 구성하는 sub-pixel의 개수를 코딩 한다. 그림 5-11과 같이 gradient를 구성하는 sub-pixel의 개수가 4 이므로, 0100의 4bit으로 코딩 한다.  $N_m$  이 1이고,  $C_s$ 는 127,  $C_f$ 는 216,  $P_1$ 은 145으로, 식 (4-3)에 따라  $\frac{abs(127-145)}{abs(127-216)} * 16 = 3$  이고 따라서 코딩 되는 비트는 0011 (4bit)이다. 순방향 gradient지만  $C_f$  값이 255 (background color)가 아니므로 컬러 비트를 추가적으로 코딩 한다. Index bit

으로 0 (1bit)을 코딩하고, 컬러 값으로 11011000(8bit)을 코딩 한다. 생성되는 bit stream은 0 0100 0011 0 11011000이다. 누적 bit stream은 1110 0010 0010 0001 1111 1110 0100 0011 0110 1100 0 이다.

#### 4.3.6 Local maximum 코딩 1

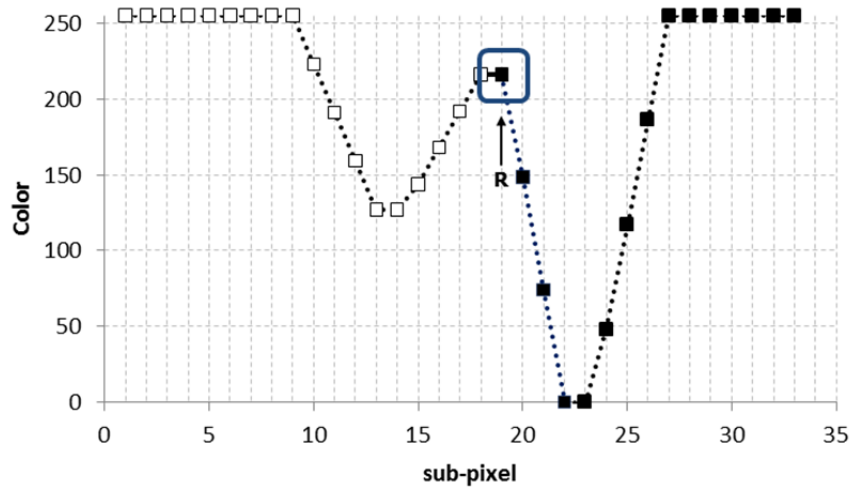


그림 4-12 Local maximum 코딩 1

그림 4-12와 같이 코딩을 시작하는 지점이 R channel의 시작지점이지만 연속된 세 개의 sub-pxiel의 값이 같지 않으므로 sub-pixel 코딩을 하고 index bit으로 0 (1bit)을 코딩한다. Sub-pixel의 값이 이전 sub-pixel의 값과 같으므로 non-gradient 코딩을 하고, 코딩되는 비트는 1 (1bit)이다. 생성되는 Bit stream은 0 1이다. 누적 bit stream은 1110 0010 0010 0001 1111 1110 0100 0011 0110 1100 001이다.

### 4.3.7 역방향 gradient 코딩 2

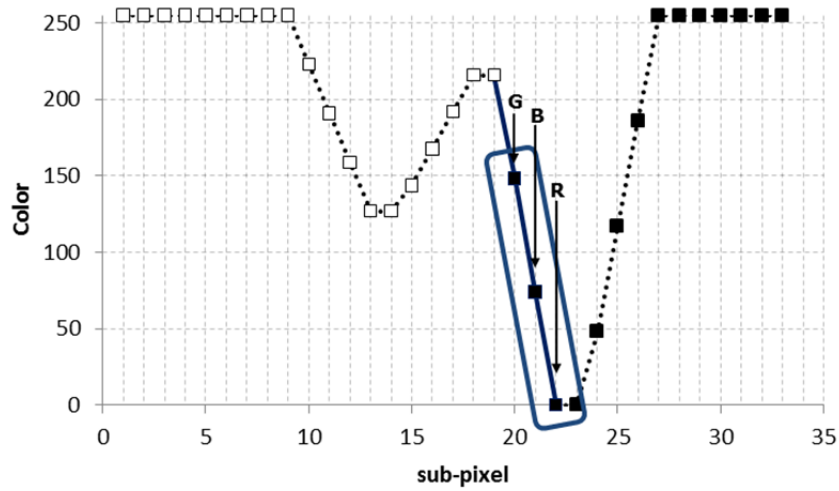


그림 4-13 역방향 gradient 코딩 2

그림 4-13과 같이 코딩을 시작하는 지점이 G channel의 시작지점이다. R channel이 코딩의 시작지점이 아니기 때문에 sub-pixel 코딩을 한다. 첫 번째 sub-pixel의 값이 이전 sub-pixel의 값과 다르므로 gradient 코딩을 수행하게 되며, 이에 따른 index bit는 0 (1bit)이다. gradient를 구성하는 sub-pixel의 개수를 코딩한다. 그림과 같이 gradient를 구성하는 sub-pixel의 개수가 3 이므로, 0011의 4bit으로 coding한다.  $N_m$  이 1이고,  $C_s$ 는 216,  $C_f$ 는 0,  $P_1$ 은 151으로, 식 (4-3)에 따라  $\frac{abs(216-151)}{abs(216-0)} * 16 = 5$  이고 따라서 코딩 되는 비트는 0101 (4bit)이다. 역방향 gradient이고  $C_f$  값이 0 (text color)이므로 color bit을 추가적으로 코딩 할 필요하고 없이 index bit만 추가적으로 1

(1bit)을 coding한다. 이 과정에서 생성되는 bit stream은 0 0011 0101 1이다.  
 누적 bit stream은 1110 0010 0010 0001 1111 1110 0100 0011 0110 1100 0010  
 0011 0101 1이다.

#### 4.3.8 Local minimum 코딩 2

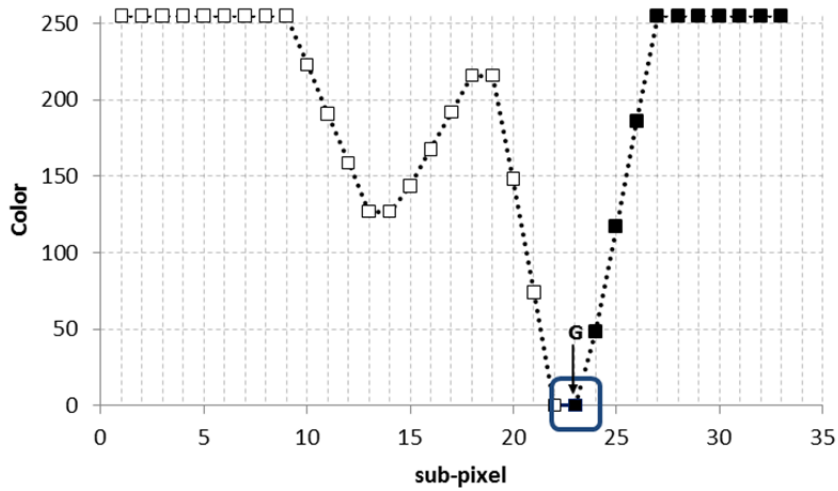


그림 4-14 Local minimum 코딩 2

그림 4-14와 같이 코딩을 시작하는 지점이 G channel의 시작지점이다. R channel이 코딩의 시작지점이 아니기 때문에 sub-pixel 코딩을 한다. Sub-pixel의 값이 이전 sub-pixel과 같으므로 non-gradient 코딩을 하고, 코딩 되는 비트는 1이다. 이 과정에서 생성되는 Bit stream은 1이다. 누적 bit stream은 1110 0010 0010 0001 1111 1110 0100 0011 0110 1100 0010 0011 0101 11이다.

### 4.3.9 순방향 gradient 코딩 2

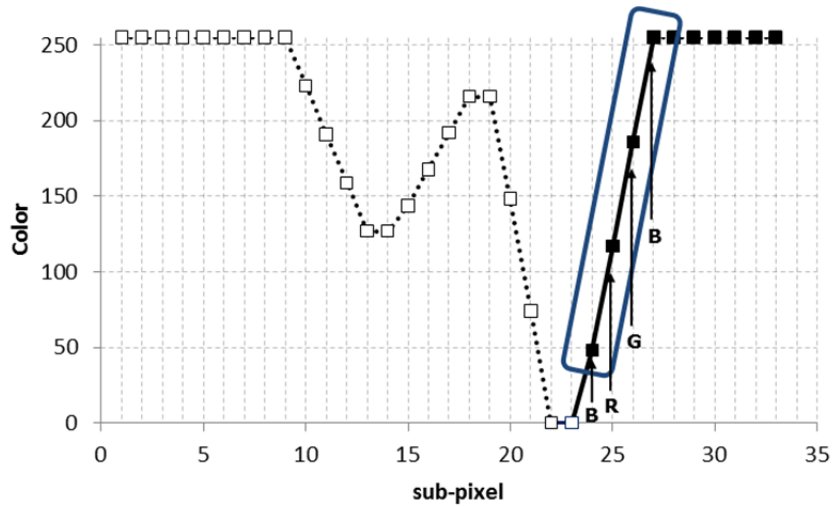


그림 4-15 순방향 gradient 코딩 2

그림 4-15와 같이 코딩을 시작하는 지점이 B channel의 시작지점이다. R channel이 코딩의 시작지점이 아니기 때문에 sub-pixel 코딩을 한다. 첫 번째 sub-pixel의 값이 이전 sub-pixel의 값과 다르므로 gradient 코딩을 수행하게 되며, 이에 따른 index bit는 0 (1bit)이다. gradient를 구성하는 sub-pixel의 개수를 코딩 한다. 그림과 같이 gradient를 구성하는 sub-pixel의 개수가 4 이므로, 0100의 4bit으로 코딩 한다.  $N_m$  이 1이고,  $C_s$ 는 0,  $C_f$ 는 255,  $P_1$ 은 51으로, 식 (4-3)에 따라  $\frac{abs(0-51)}{abs(0-255)} * 16 = 3$  이고 따라서 코딩 되는 비

트는 0011 (4bit)이다 순방향 gradient이고  $C_f$  값이 255 (background color)이므로 color bit을 추가적으로 코딩 할 필요하고 없이 index bit만 추가적으로 1 (1bit)을 코딩 한다. 이 과정에서 생성되는 bit stream은 0 0 0100 0011 1이다. 누적 bit stream은 1110 0010 0010 0001 1111 1110 0100 0011 0110 1100 0010 0011 0101 1100 0100 0011 1이다.

#### 4.3.10 Whole-pixel 코딩 2

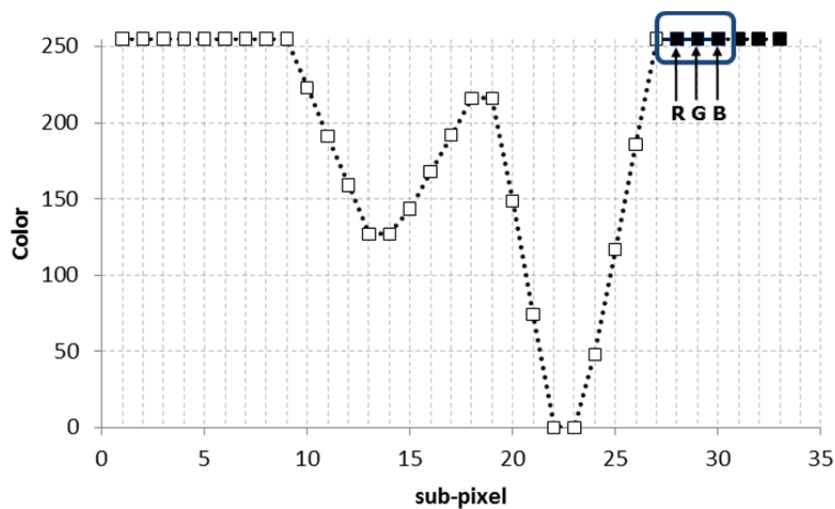


그림 4-16 Whole pixel 코딩 2

그림 4-16과 같이 코딩을 시작하는 지점이 R channel의 시작지점이고, 연속된 세 개의 sub-pixel의 값이 같으므로 whole pixel 코딩을 한다. 나머지 세 개의 sub-pixel도 같은 방법으로 코딩 될 수 있다. 이 과정에서 최종

bit stream은 1 1 이다. 누적 bit stream은 1110 0010 0010 0001 1111 1110 0100 0011 0110 1100 0010 0011 0101 1100 0100 0011 111이다.

이렇게 gradient 코딩 방법을 진행하게 되면 최종적으로 생성되는 bit stream은 1110 0010 0010 0001 1111 1110 0100 0011 0110 1100 0010 0011 0101 1100 0100 0011 111 로 표현된다. 본 과정의 모든 텍스트는 gradient의 반복으로 볼 수 있기 때문에 gradient 코딩으로 진행하면 모든 텍스트 블록에 대하여 코딩을 진행 할 수 있다.



## 4.4 텍스트 블록 코딩 실험 결과

Sub-pixel gradient를 이용한 코딩 압축 과정은 위에서 설명한 바와 같이 진행된다. 본 실험 결과는 이러한 sub-pixel gradient를 이용한 코딩 방법의 효율성을 확인한다. 본 연구에서 설명하는 Sub-pixel gradient 코딩 압축 방법과 표준 알고리즘인 JPEG[2], JPEG2000[3], H.264/AVC[1]와 Pan [10]에서 제안한 compound image 압축 알고리즘과 비교를 진행한다. 다양한 형태의 영역을 가진 영상을 기준으로 제안하는 알고리즘이 기존의 압축 알고리즘에 비하여 높은 압축률 및 화질을 가지는지에 대하여 확인하였다. 실험 영상에 사용된 프로그램은 JPEG의 경우 OpenCV2.0[16]의 JPEG library, JPEG 2000는 jasper[51], H.264 Intra mode는 JM18.4[15]를 사용하였다. 실험 영상은 RGB 포맷의 컴퓨터 스크린 영상을 캡처하여 사용하였다. 실험에 사용한 RGB 영상은 컬러 변환에 의한 영상 손실을 고려하여 사용되었다. 또한 본 연구의 실험결과는 RGB 포맷을 기준으로 작성되었다. 공정한 비교를 위하여 실험비교를 위한 H.264 Intra mode의 실험에 사용한 영상 포맷의 경우 컬러 영상 변환에 의한 화질 손실을 고려하여 YUV 4:4:4의 영상 포맷을 사용하였다. 실험결과는 2가지의 영상을 기준으로 진행하였다. 기존의 알고리즘과 비교를 위한 실험영상으로 1024x768 크기의 영상을 하였으며, 텍스트 부분의 화질 및 압축률 비교를 위하여 640x480 크기의 영상을 사용하였다.

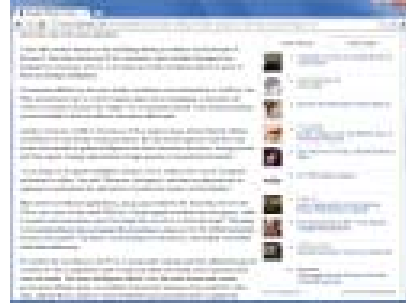
실험결과는 2가지의 기준으로 진행되었다. 첫째 1024x768의 컴퓨터 화면 영상에서 나타나는 4가지의 이미지를 실험 영상으로 사용하였다. 이러

한 실험영상의 경우 텍스트와 이미지로 분리가 되는데, 본 연구에서 이미지 부분의 압축은 JPEG 을 사용하였으며, 비교를 위한 Pan [10]의 논문의 알고리즘은 논문에 제시한 것과 같은 텍스트 이미지 분리 방법을 사용하였으며 텍스트 영역의 경우 Huffman coding 방식을 추가적으로 사용하여 결과를 확인하였다. 본 연구의 압축 알고리즘 역시 기존 알고리즘과의 대등한 비교를 위하여 최종적으로 Huffman coding 방법이 포함되어 있는 결과이다. 총 4개의 이미지와 텍스트 영역을 포함하고 있는 영상을 사용하였는데, 영상의 선택기준은 텍스트 영역의 영상의 종류이다.

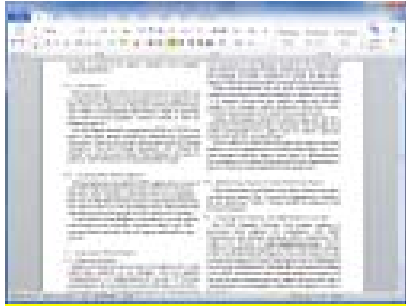
그림 4-17(a) 는 bi-level 텍스트를 포함하는 webpage 이고, 그림 4-17(b) 는 simple rendered 텍스트를 포함한다. 그림 4-17(c)에서는 문서 프로그램에서 나타나는 simple rendered 텍스트 경우를 확인한다. 마지막으로 그림 4-17(d)는 PDF 문서에서 사용되는 complex rendered 텍스트를 포함하고 있다.



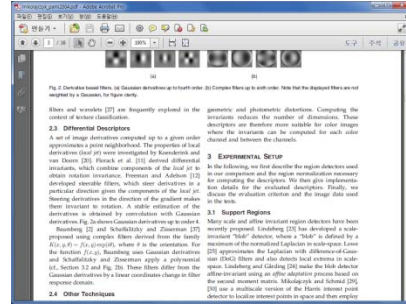
(a) bi-level text (web)



(b) simple rendered text (web)

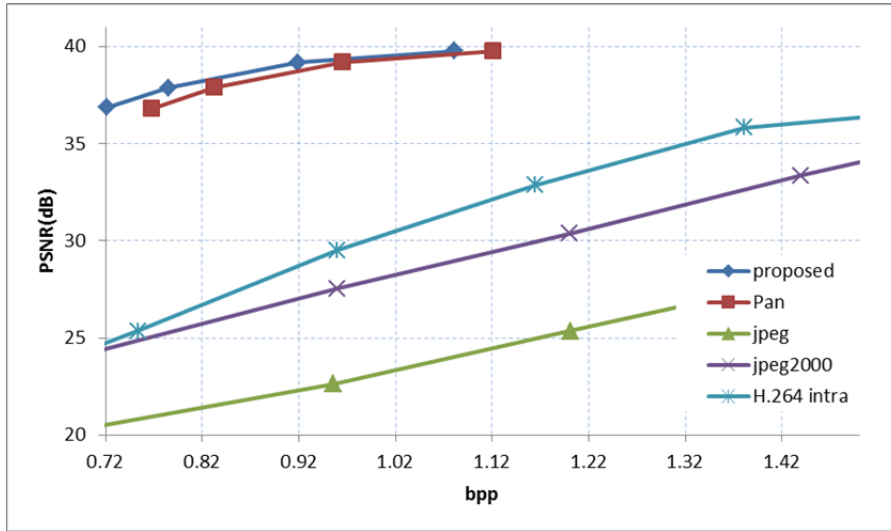


(c) simple rendered text (word)

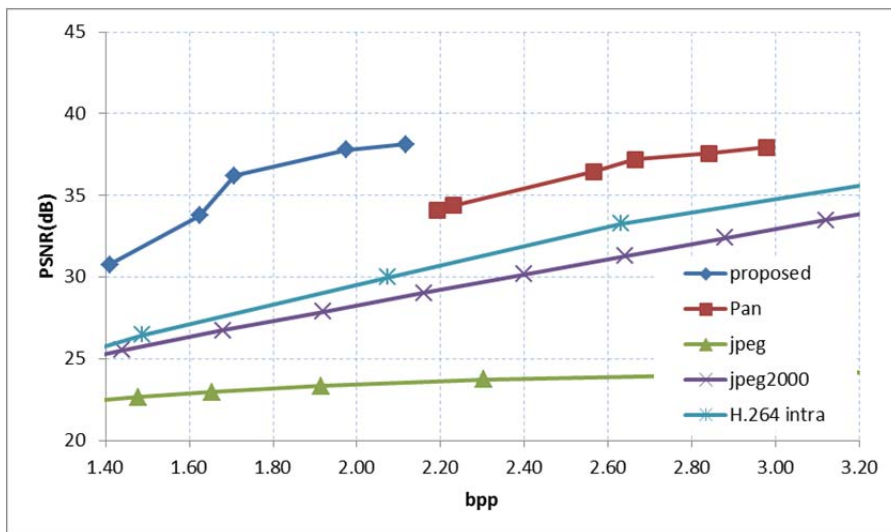


(d) complex rendered text (PDF)

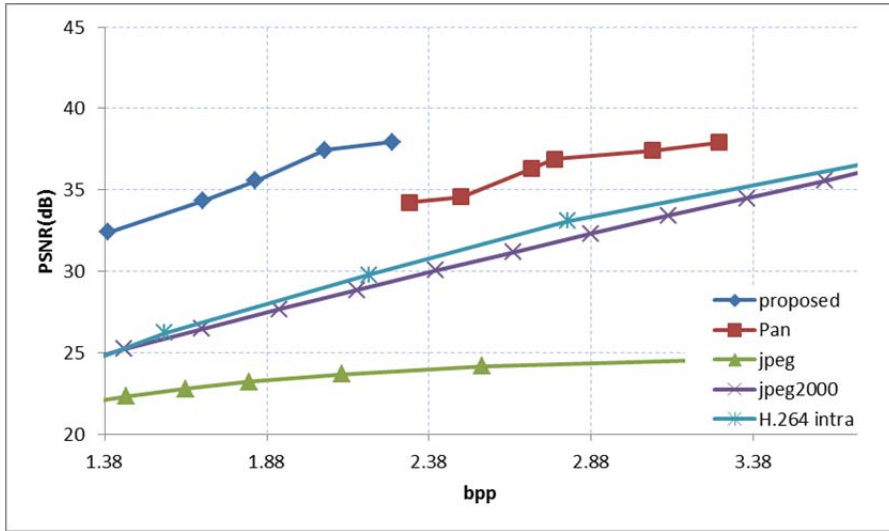
그림 4-17 화질 비교를 위한 실험 영상



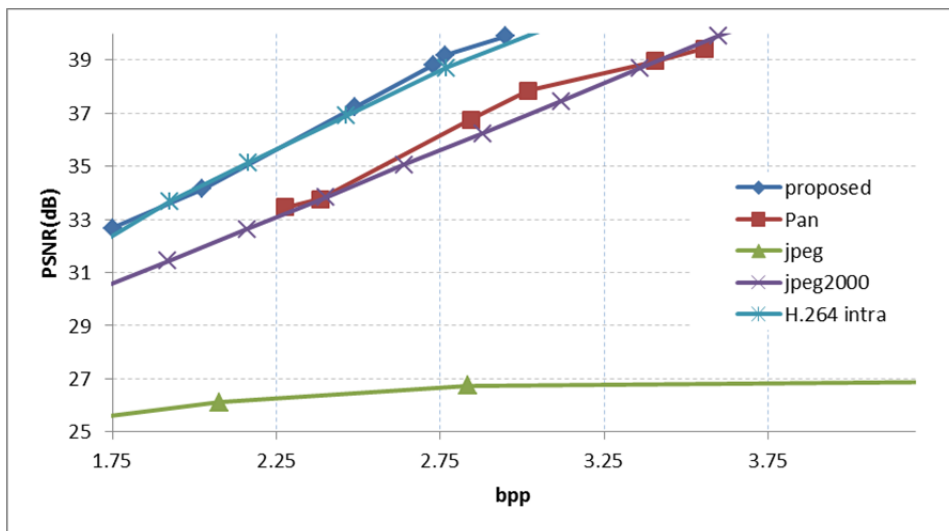
(a) bi-level text



(b) simple rendered text



(c) simple rendered text



(d) complex rendered text

그림 4-18 PSNR 실험결과

sub-pixel gradient 코딩 방식을 적용하고, 이미지 영역은 JPEG을 적용하여 최종 압축 효율과 영상의 화질에 대한 비교는 그림 4-18에 나타내었다. 그림 4-17(a) 와 같이 bi-level 텍스트를 포함하고 있는 영상의 경우 제안하는 알고리즘의 성능이 H.264 Intra mode와 JPEG2000 과 같은 상용 압축 알고리즘에 비해 약 10 ~ 12 dB 성능향상을 보여주고 있다. 또한 index map을 사용하는 Pan[21] 의 접근방식에 비해서도 동일 영상 화질에서 압축효율이 높아짐을 알 수 있다. Simple rendered 텍스트를 포함하고 있는 영상의 경우 그림 4-18(b) 와 그림 4-18(c) 와 같이 H.264 Intra mode 와 JPEG2000 에 비하여 5 ~ 7 dB 의 화질 개선 효과를 확인 할 수 있다. 또한 Pan[21] 의 접근방식에 비하여 약 80% 의 압축효율 개선에 대한 결과도 확인 할 수 있다. 그림 4-18(d)와 같이 complex rendered 텍스트를 포함하는 영상의 경우 Pan[21] 과 JPEG2000에 비하여 3dB 의 화질 개선 효과를 확인할 수 있었으며 H.264 Intra mode의 경우는 동일한 압축 효율을 나타냄을 알 수 있다. 하지만 H.264 의 intra mode의 경우 제안하는 방법과 PSNR은 동일 하지만 가독성의 측면에서는 제안하는 알고리즘의 결과가 나음을 그림 4-19에 보여주는 결과로 설명할 수 있다.

**Related: Earth's 'twin' will be found in 2013, so start trashing this planet**

Besides its color, the planet is unusual because of its distance from its sun, GJ 504 or 59 Virginis — nearly nine times the distance Jupiter orbits the sun, or 43.5 astronomical units. (One astronomical unit is the distance from the Earth to the sun, or about 92,955,807 miles) The sheer distance makes astronomers start to question some of the theories they have around how planets form.

"This is among the hardest planets to explain in a traditional planet-formation framework," said Markus Janson, a Hubble postdoctoral fellow at Princeton and one of the researchers on the project. "Its discovery implies that we need to seriously consider alternative formation theories." At the very least, assessing basic assumptions in the current popular planet formation theory would be important, said Janson in a NASA release.

**Related: Three new habitable planets discovered — and they're not very far away**

Infrared photos of the planet were taken by the Subaru Telescope on Mauna Kea in Hawaii. The approximately 160 million-year-old star system in which the exoplanet resides orbits around the star GJ 504, and is approximately 57 light-years from Earth. Especially exciting is the fact that GJ 504 is about only 1/30th the age of our own sun, says Michael McElwain, a member of the discovery team at NASA's Goddard Space Flight Center. "Studying these systems is a little like seeing our own planetary system in its youth."

(a) simple rendered 텍스트 (web page, 640x480)

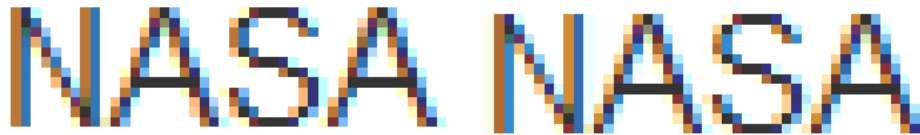
An intrinsic limitation of a standard MRC imaging model is that the binary mask can only represent discontinuous transitions between text, line art, and background colors. In practice, this type of hard transition can introduce substantial artifacts since real documents often contain subtle continuous-tone transitions between regions. These continuous transitions are useful in anti-aliasing, and allows the document to be encoded with lower dpi, which can further reduce bit rate. In principal, it is possible to add edge detail to the foreground or background layers; however, in practice, this detail is lost when these layers are subsampled and encoded using lossy natural image coders at acceptable bit rates.

In this paper, we propose a method called resolution-enhanced rendering (RER) for jointly optimizing the MRC encoder and decoder to achieve low distortion rendering of edge transitions in the MRC binary mask layer [19]. The method works by adaptively dithering the mask layer of a three-layer MRC encoding to produce the intermediate tone

(b) complex rendered 텍스트 (PDF, 640x480)

그림 4-19 가독성 실험 영상

그림 4-19 는 본 연구에서 분류하는 그림 4-19(a) simple rendered 텍스트와 그림 4-19(b) complex rendered 텍스트의 압축효율 및 가독성을 설명하기 위한 실험 결과이다. 그림 4-20과 그림 4-21은 본 연구의 텍스트 블록 코딩과 H.264 Intra mode, JPEG2000, JPEG 코딩의 비교를 위하여 동일한 압축비에 대하여 복원영상을 확대하여 나타낸 것이다.



(a) 원본

(b) proposed



(c) JPEG



(d) JPEG2000



(e) H.264 Intra mode(QP = 35)

그림 4-20 simple rendered text 화질 비교



lossy lossy

(a) 원본

(b) proposed

lossy lossy

(c) JPEG

(d) JPEG2000

lossy

(e) H.264 Intra mode(QP = 35)

그림 4-21 complex rendered text 화질 비교

그림 4-20과 그림 4-21에서 보여지는 바와 같이 H.264 Intra mode 와 JPEG2000, JPEG 과 같은 DCT, Wavelet 기반의 압축방식은 높은 압축률을 고려하면 text 주변에서 영상 왜곡현상이 나타날 수 있다.

표 4-6 은 그림 4-19(a) 의 simple rendered text의 실험 영상에 대하여 각 압축방식에 대한 유사한 압축률에서의 PSNR 및 압축된 크기를 비교한 것이다.

표 4-7는 그림 4-19(b)의 complex rendered text의 실험 영상에 대하여 표 4-7에서와 같이 동일하게 비교 하였다.

본 결과와 같이 sub-pixel gradient를 이용한 text 블록 압축 방식은 기존의 표준 알고리즘인 JPEG, JPEG2000, H.264 Intra mode와 비교 하였을 때 동일 압축 비에서 약 2 ~ 12 dB 의 PNSR 향상을 보인다. 특히 양자화 기반의 압축 알고리즘에 비하여 텍스트의 가독성이 뛰어난을 확인할 수 있다.

표 4-6 simple rendered text 압축율 비교

	File size (Kbytes)	Bpp	PSNR(dB)
Proposed	323	1.08	32.39
JPEG	401	1.34	20.20
JPEG2000	431	1.44	23.78
H.264 Intra mode	393	1.31	23.30

표 4-7 complex rendered text 압축율 비교

	File size (Kbytes)	Bpp	PSNR(dB)
Proposed	493	1.65	30.57
JPEG	498	1.66	22.86
JPEG2000	503	1.68	26.30
H.264 Intra mode	543	1.81	28.40

## 제5장 그룹 움직임 추정 방법

본 장에서는 compound image가 가지는 특성을 이용하여 이에 적합한 움직임 추정방법을 설명한다. Compound image의 움직임은 natural 영상 움직임과는 다른 특성을 가진다. 이러한 compound image의 특성을 이용하면 효율적인 움직임 추정을 할 수 있다.

첫째, compound image의 움직임 추정은 컴퓨터 스크린의 캡처된 영상을 기준으로 움직임을 추정하기 때문에 자연 영상과 같은 외부 요인에 의한 노이즈가 없다. Natural 영상은 카메라와 같은 영상의 입력장치에서의 밝기 변화에 의한 노이즈를 가질 수 있다. 하지만 compound image는 입력장치에 의한 노이즈에 대하여 고려하지 않는다. 따라서 현재 영상과 이전 영상 간의 노이즈에 의한 영상 왜곡은 발생하지 않는다. 둘째, 영상의 움직임은 마우스나 키보드와 같은 입력장치에 의한 단일 움직임을 가진다. 영상을 생성하는 컴퓨터의 특성상 동일한 입력이 동시에 실행 되지 않는다. 또한 다중 입력이라 하더라도 영상의 생성 순서에 의하여 하나의 움직임으로 표현된다. 셋째, 영상의 움직임의 폭이 자연 영상의 움직임에 비해 크고 다양하다. 영상의 압축률은 영상을 비교하는 탐색영역에 비례한다. 따라서 compound image와 같이 움직임의 폭이 다양한 영상에서는 탐색영역의 크기에 따라 압축률의 변화량이 커질 수 있다. 이러한 3가지의 compound image 특성을 고려하여 본 연구에서는 영역 움직임을 기반으로 한 그룹 움직임 추정 방법을 설명한다.

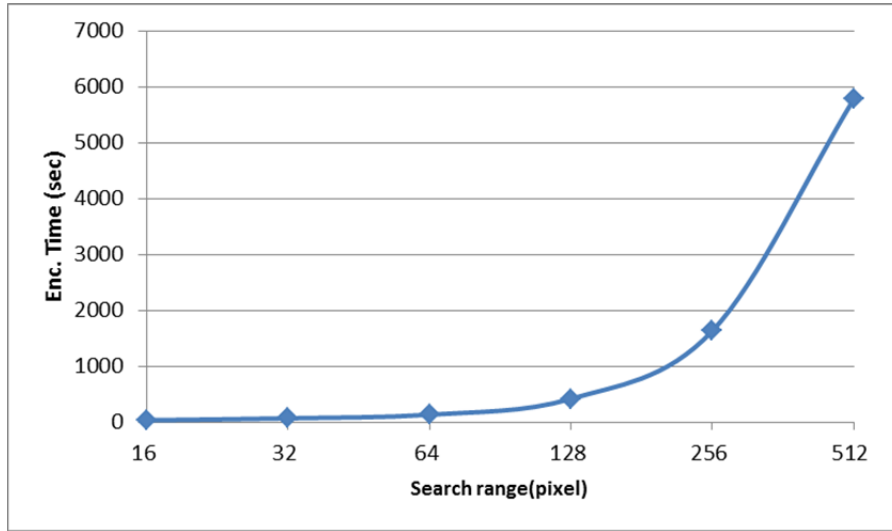
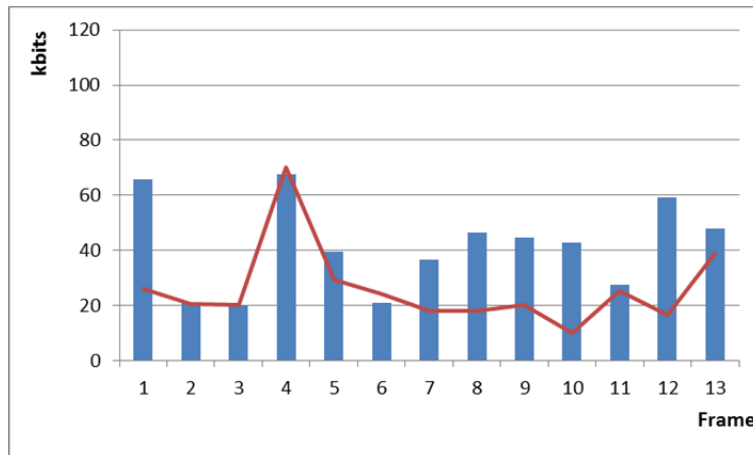


그림 5-1 탐색영역 별 encoding 시간 (H.264/AVC inter mode)

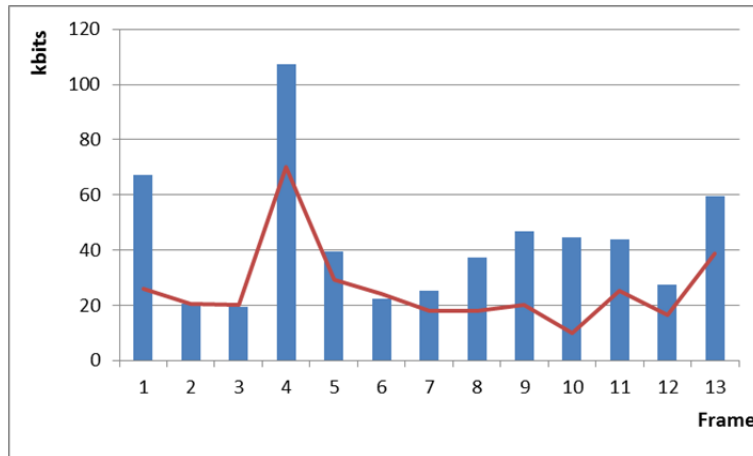
움직임 추정방법으로 가장 많이 사용되는 H.264/AVC의 full search 움직임 탐색 방법의 탐색 영역별 수행시간을 살펴 보면 전체 수행시간이 그림 5-1과 같이 탐색 영역의 크기 변화에 대하여 지수 증가를 보이는 것을 알 수 있다. 따라서 움직임 추정 방법에서 수행시간을 줄이는 가장 큰 요소는 탐색 영역을 줄이는 것이라 할 수 있다.

하지만 이러한 탐색영역을 축소하는 방법은 동일한 조건의 움직임 탐색 방법에서 압축률을 저하시키게 된다. 그림 5-2는 H.264/AVC의 full search 움직임 탐색 방법을 사용하여 64 크기와 512 크기의 탐색영역을 기준으로 10frame 영상에 대하여 확인한 bitrate 결과이다. 4번 frame에서만 영상은 64 크기의 탐색 범위를 벗어난다. 그림 5-2(a)에서 나타난 바와 같이 탐색 영역을 벗어나지 않는 경우 데이터가 균일하게 압축 됨을 알 수 있다. 그림 5-2(b) 와 같이 탐색 영역을 벗어나는 4번 frame은 데이터 량이 급격히

늘어나게 된다.



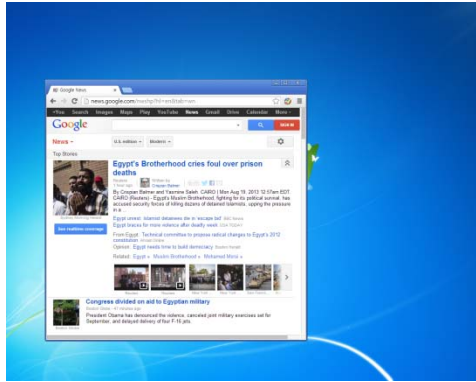
(a) Full search (탐색영역 512)



(b) Full search (탐색영역 64)

그림 5-2 탐색영역별 데이터 변화량

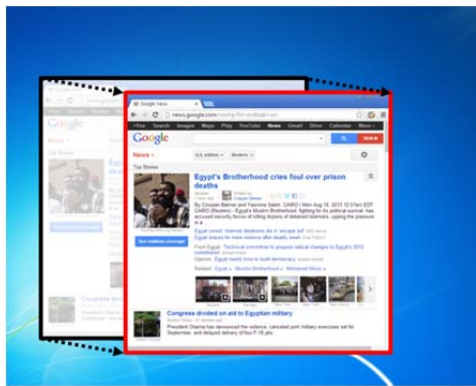
이러한 3가지의 compound image 특성을 이용하여 본 연구에서는 영역 움직임을 기반으로 한 그룹 움직임 추정 방법을 설명한다.



(a) 이전영상



(b) 현재영상



(c) 영상의 움직임



(d) 움직임에 의한 영역분류

그림 5-3 compound image의 움직임

그림 5-3은 compound image의 움직임을 나타내는 그림이다. 그림 5-3(a)의 영상에서 그림 5-3(b)의 영상으로 움직임을 가진다. 이러한 두 개의 영상 사이에서의 움직임은 그림 5-3(c)와 같이 표현될 수 있다. 이렇게 compound image에서의 움직임은 윈도우를 기반으로 영역이 움직이는 것으로 표현된다. 움직임 사이의 차이를 살펴 보면 그림 5-3(d)의 1로 표현된 영역은 이전 영상과 현재영상에서 움직임이 없는 영역이다. 그림 5-3(d)

에서 2로 표현 된 영역은 이전영상에서는 없었으나 현재 영상에서 새로이 나타난 영역이다. 마지막으로 그림 5-3(d)에서 3으로 표현 된 영역은 이전 영상에 존재하는 영역이 현재 영상으로 이동한 영역이 된다. 이러한 결과는 영상의 시간적 연관성을 이용하지 않는 경우에는 2 영역과 3 영역은 새로운 영역으로 판단되어야 한다. 하지만 시간적 연관성을 고려한다면 2 영역만이 새로운 영역이 된다. 따라서 본 연구에서는 이러한 시간적 연관성을 고려하여 3 영역으로 판단 될 수 있는 영역에 대하여 그룹간의 움직임 추정을 통하여 빠르게 움직임을 찾을 수 있는 방법을 설명한다.

그룹 움직임 추정 방법은 그림 5-4와 같이 3 단계의 과정을 거치게 된다. 첫 번째, sub-pixel gradient 블록 분류 방법을 통하여 분류된 각 블록을 기준으로 라인 블록 그룹 동작을 수행하게 된다. 두 번째, 형성된 그룹을 기준으로 움직임의 예측을 위한 이전 영상과 현재 영상의 그룹을 정합성을 확인하고, 이렇게 생성된 데이터를 기준으로 선택된 움직임 벡터가 위치할 영역을 생성하게 된다. 마지막으로, 영역 벡터를 사용하여 16x16 크기의 탐색 영역을 탐색하여 최종 벡터를 생성한다. 이러한 최종 벡터를 기준으로 움직임이 있는 블록(Motion block)과 움직임이 없는 블록(Same block)으로 구분하게 된다.



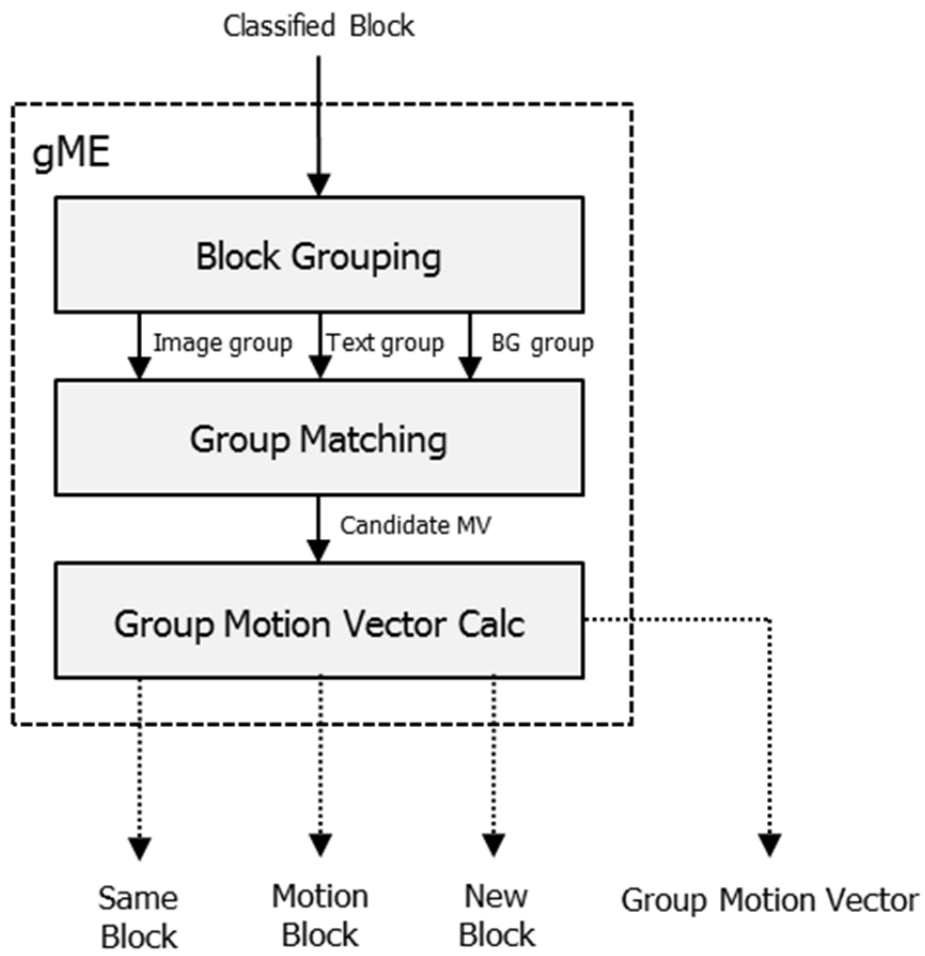


그림 5-4 그룹 움직임 추정 방법

## 5.1 Block Grouping

본 단계는 sub-pixel gradient 블록 분류 방법을 통하여 분류된 블록을 기준으로 그룹을 생성하는 단계이다. 이 단계에서 하나의 색으로 구성된 블록의 경우 배경 블록으로 판단하여 기존의 블록 분류에서 그룹화를 위한 블록의 분류 기준이 추가 된다.

그룹 생성방법은 compound image에서 나타나는 텍스트 영역과 이미지 영역에서의 특성을 고려한 것이다. 텍스트 영역을 블록으로 나누어 보면 블록 내부의 픽셀의 특성이 나타나지 않는다. 하지만 텍스트를 영역에서 살펴 보면 문장의 띄어쓰기나 줄 바꿈과 같이 영역의 특성을 잘 나타내는 것들이 존재하게 된다.

그림 5-5과 같이 인접한 블록을 수평 방향과 수직방향의 그룹으로 생성한다. 현재의 블록 분류가 좌측 블록의 분류와 동일하면 동일 그룹으로 확장되고, 동일하지 않으면 새로운 그룹을 생성하게 된다. 이러한 그룹화 방법은 동일한 그룹의 움직임 추정하기 위함이다. 그림 5-5은 이러한 라인 블록을 기준으로 그룹이 생성되는 과정을 설명한다. 이러한 방법으로 그룹을 한 라인을 기준으로 그룹을 나눈다.

본 연구에서는 텍스트 블록 영역이 가질 수 있는 특성을 고려하여 수평 방향과 수직 방향으로 이루어진 라인 블록 그룹을 생성한다. 라인 블록을 그룹화 하는 과정은 현재 블록이 좌측에 인접한 그룹과 동일한 배경과 블록 분류 기준이 일치하면 하나의 그룹으로 통합을 하고 일치하지 않으면 새로운 그룹을 생성하게 된다. 이렇게 생성되는 그룹은 각 블록의 분류

기준에 따라 다시 분류 된다. 새로운 라인은 새로운 그룹으로 생성된다. 이러한 그룹 방법은 compound image 가 서로 다른 분류의 영역이 다수 생기는 특성을 이용한 것이다.

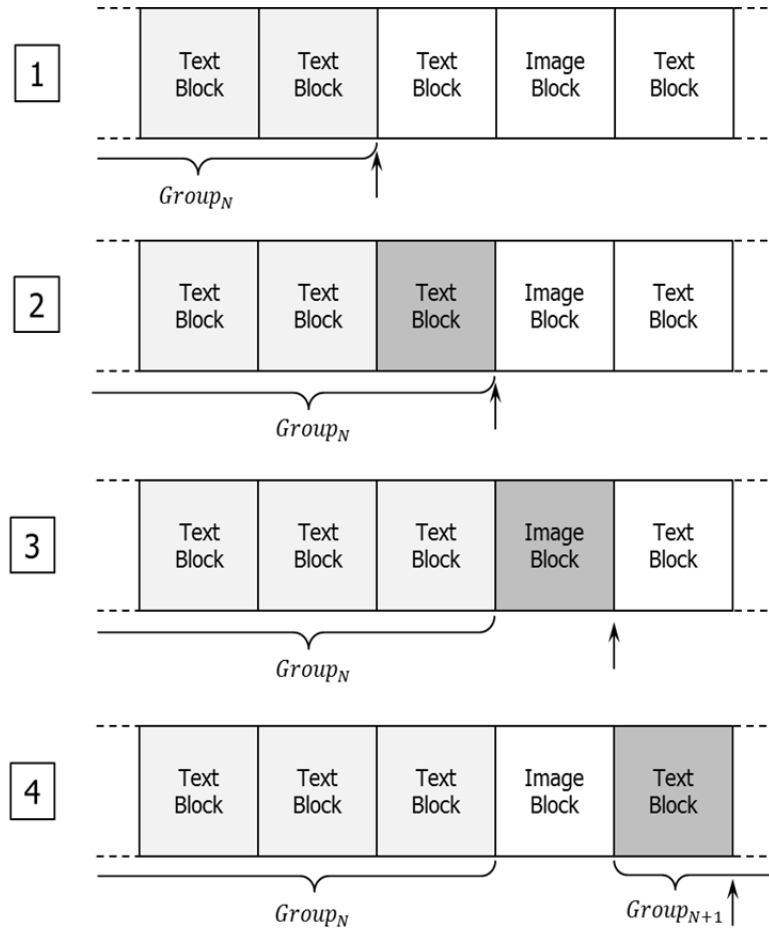


그림 5-5 그룹 생성 방법

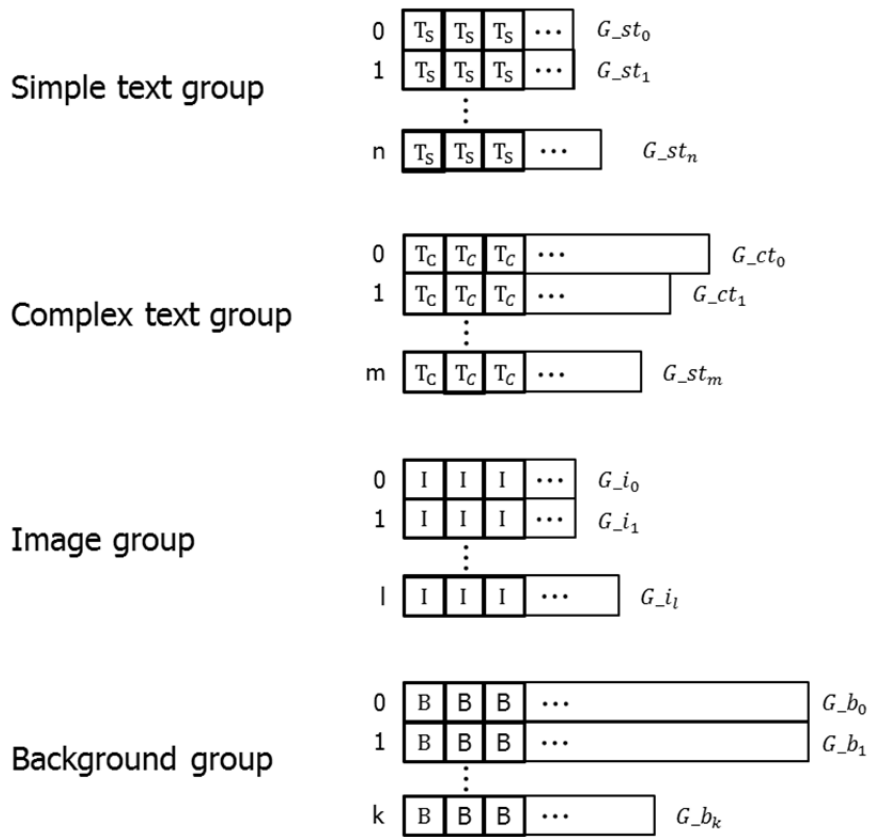


그림 5-6 그룹의 분류

따라서 생성되는 그룹은 simple rendered text, complex rendered text, image, background 그룹으로 그림 5-6와 같이 4개의 그룹으로 분류된다. 분류된 그룹들을 기준으로 다음절에서 설명하는 그룹 정합성 단계를 진행하게 된다.

## 5.2 Group Matching

본 절에서의 그룹의 정합성은 위에서 설명한 바와 같이 동일한 분류의 그룹을 대상으로 진행한다. 이전 영상과 현재 영상의 그룹에서 동일 분류 그룹을 대상으로 그룹의 블록크기, 블록의 대표 컬러를 비교하여 각 그룹간의 방향성과 그룹간의 거리를 측정하게 된다. 그림 5-7는 정합성 비교를 나타낸 그림이다. 같은 분류이면서 동일 크기의 그룹을 연결하게 된다. 크기가 다른 그룹은 연관성이 없는 것으로 판단하여 그룹간의 정합성 판단은 하지 않는다.

이렇게 연결된 그룹간의 움직임을 파악하고자 2 단계의 과정을 거치게 된다. 먼저 연결된 각 그룹간의 방향성을 파악한다.

$$Direction_{Gn} = \frac{Y_{p\_center} - Y_{c\_center}}{X_{p\_center} - X_{c\_center}} \quad (5-1)$$

그룹간의 방향성은 식 (5-1) 와 같이 표현된다.  $Y_{p\_center}$  는 이전 영상에서 선택된 그룹 중앙의 y 좌표를 의미한다.  $Y_{c\_center}$  는 현재 영상에서 선택된 그룹 중앙의 y 좌표를 의미한다. 마찬가지로  $X_{p\_center}$ ,  $X_{c\_center}$  는 이전영상과 현재영상에서 선택된 그룹 중앙의 x 좌표를 의미한다.

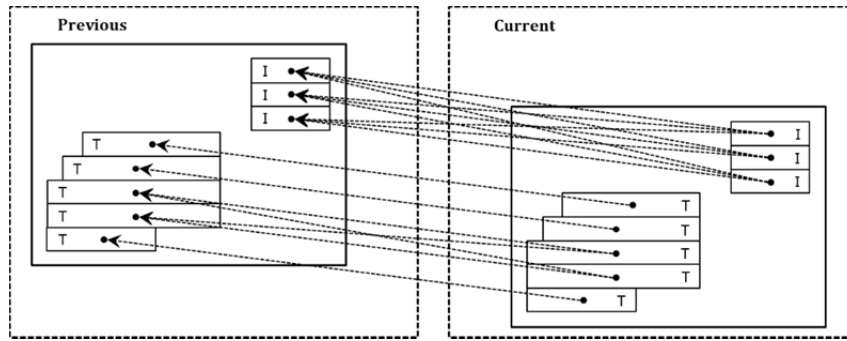


그림 5-7 이전 영상과 현재 영상 간의 정합성 비교

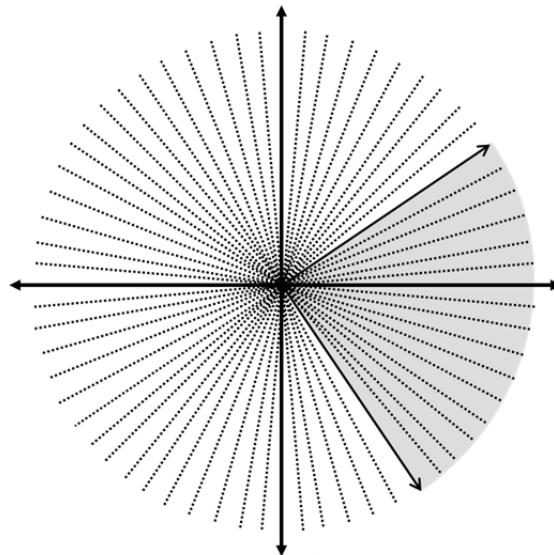


그림 5-8 그룹간의 방향성 선택

$Direction_{Gn}$ 으로 표현된 각 그룹 간의 방향성 결과 값으로 64분위의 방향으로 나누어 분포도를 생성한다. 이러한 분포도를 기준으로 연속된 16분위의 방향의 분포가 가장 많은 방향을 이전영상에서 현재영상으로 이동한 영역 움직임의 방향으로 선정한다. 그림 5-8는 이러한 64분위를 기준으로 선택되는 방향성을 나타낸다. 그림 5-8에서 회색으로 표현된 부분은 연속된 16분위의 분포의 합이 가장 큰 것을 선택하여 선택된 영역 방향성을 나타낸다. 이러한 분류 기준은 그룹간의 연결에서 포함될 수 있는 오차율을 줄이고 대표 방향성을 선택하기 위함이다. 블록을 기준으로 연관성을 판단하기 때문에 각 픽셀의 특성보다는 블록의 특성을 가지기 때문에 오차율은 증가할 수 있다. 다수의 방향으로 방향성이 나타나기 때문에 정확한 그룹 연결 방향성을 찾기 위하여 먼저 방향성을 판단한다.

다음 단계로 연속된 16분위의 분포에 포함되어 있는 각 그룹의 연결을 기준으로 각 그룹간의 이동성을 확인한다. 각 그룹간의 이동을 확인하기 위하여 연결된 그룹을 기준으로 x 와 y 의 이동거리를 측정한다. 거리를 측정하는 기준은 식 5-3, 식 5-4 에서 나타내었다.

$$Dist_x = |X_{p\_center} - X_{c\_center}| \quad (5-2)$$

$$Dist_y = |Y_{p\_center} - Y_{c\_center}| \quad (5-3)$$

일차 방향성이 결정된 상태이므로 식 (5-2) 과 같이 x의 이동성( $Dist_x$ )의 측정은 이전영상의 그룹과 현재 영상의 그룹의 중앙점을 중심으로 이동한 거리를 측정하게 된다. 동일하게 식 (5-3) 와 같이 y의 이동성( $Dist_y$ )도 측

정을 한다. 방향성 계산과 마찬가지로 8 픽셀 단위로 분포도를 생성하여 분포가 가장 많은 2개 거리를 각 그룹별 영역 이동 움직임의 후보 군으로 선택하게 된다. 최종적으로 그룹 움직임 벡터 후보군은 총 16개가 생성된다. 이러한  $x, y$  거리의 후보 군은 다음에서 설명할 최종 그룹 움직임 벡터 계산에 사용되는 것이다. 생성된 이동성 후보 군은 영상의 움직임을 블록 단위로 추정하는 것이기 때문에 최대 블록의  $x, y$ 의 길이의 오차율을 가질 수 있다.



### 5.3 Group motion vector calculation

그룹 정합성에서 계산된 움직임 후보 군을 기준으로 정확한 그룹 움직임 벡터를 생성한다. 전체 영역의 모든 픽셀을 비교하게 되면 이와 같은 단계가 필요 없지만 H.264/AVC와 같은 움직임 영상 압축 알고리즘에서도 탐색영역 및 탐색 회수를 줄이는 것에 대한 연구가 많이 되어 왔다. 본 연구에서는 이러한 탐색 영역을 최소화 하고 복잡도를 개선하기 위한 움직임 추정 방법을 설명한다.

그룹 움직임 벡터를 생성하기 위한 첫 번째 단계는 이전영상과 현재영상의 동일한 위치를 확인하는 것이다. Compound image는 영역의 이동성이 단순하기 때문에 배경과 같이 움직임이 없는 경우가 많이 발생한다. 현재영상의 블록과 이전 영상의 블록이 일치하게 되면, 그림 5-3의 영역 1과 같이 이전영상이 변화가 없는 블록이 된다. 이러한 블록은 이전 영상과 동일 블록(Same Block)으로 판단된다. 블록과 블록간의 일치 여부는 식 (5-4) 와 같이 SAD 값으로 계산된다. 블록 내의 모든 픽셀의 값이 동일하면 식 5-4 의 SAD는 0의 값을 가지게 된다. 따라서 SAD 값이 0인 경우를 동일 블록으로 판단한다. 현재 블록이 SAD 값이 0이 아닌 경우에는 두 번째 단계 그룹 움직임 벡터 생성으로 진행한다.

두 번째 단계는 그룹 움직임 후보 군에 대하여 각 분류 별 그룹 움직임을 선택한다. 각 분류 별 그룹 움직임이 모두 동일하게 되면 가장 먼저 움직임 확인을 위한 벡터로 선정한다. 이렇게 선정된 벡터를 기준으로 그룹 움직임 벡터를 확인한다.

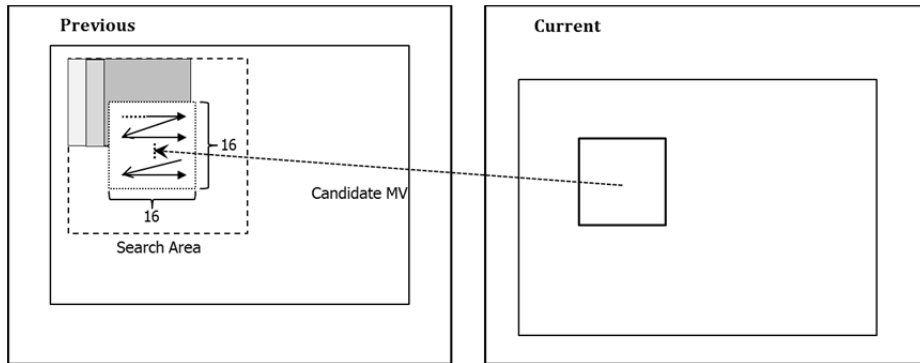


그림 5-9 움직임 벡터 생성을 위한 탐색 방법

$$SAD = \sum_x \sum_y |C(x,y) - P(x,y)|, \quad (x,y \in \{0,1,2,3\}) \quad (5-4)$$

그룹 움직임 벡터를 확인하는 방법은 선정된 그룹 움직임 벡터 후보가 가리키는 위치로 이동하여 16픽셀 크기 주변 영역을 모두 탐색하게 된다. 따라서 본 연구에서의 탐색영역의 크기는 16이 되는 것이다.

텍스트의 첫 번째 그룹을 선택하여 그룹내의 각 블록을 기준으로 이전 영상의 탐색영역에서 탐색을 진행한다. 탐색을 진행하는 방법은 그림 5-9와 같이 탐색영역에 대하여 식 (5-4)의 SAD가 0인 부분을 찾는다. SAD값이 0인 부분이 발생하게 되면 다음 블록의 움직임 벡터 확인에 대해서 SAD값이 0으로 발생한 위치부터 확인하게 된다. 하나의 텍스트 그룹내의 모든 블록의 이동 위치가 동일하면 이를 그룹 움직임 벡터로 선정한다.

그룹 움직임 벡터는 이전 영상에서 각 그룹이 이동한 위치로서 compound image 에서는 그룹 움직임 벡터가 움직임이 확인된 모든 그룹에서 동일하다. 이러한 그룹에 속하여 있는 모든 블록은 움직임이 있는 블록(Motion Block)으로 판단한다. 최종적으로 움직임이 있는 블록과 움직임이 없는 블록으로 판단되지 않는 모든 블록에 대하여 새로운 블록(New Block)으로 판단한다. 따라서 최종적으로 압축은 새로운 블록으로 선택된 블록만 진행한다.

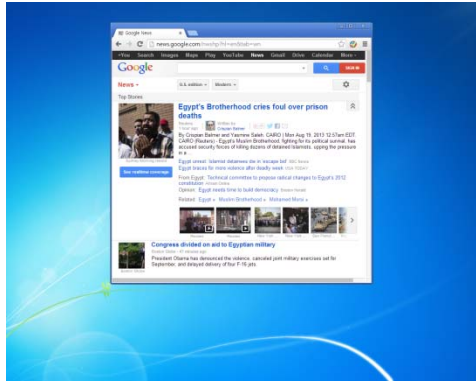
그룹 움직임 벡터는 정지영상 이미지 코딩 방법과 동영상 이미지 코딩 방법을 병행하여 사용할 경우 차이를 가질 수 있다. H.264/AVC inter mode 와 같이 움직임 추정방법이 포함되어 있는 이미지 압축 방식을 사용할 경우 최종적인 그룹 움직임 벡터 계산 단계를 포함 하지 않아도 된다. H264/AVC inter mode는 본 연구가 제시하고 있는 그룹 움직임 벡터 계산과 유사한 방법으로 각 블록의 움직임 벡터를 계산하게 된다. 따라서 본 연구에서는 이미지 코딩 방법에 따라서 선택적으로 단계를 조절하여 진행할 수 있다.

## 5.4 그룹 움직임 추정 방법 실험 결과

그룹 움직임 추정 방법은 시간적 연관성을 사용하는 압축 알고리즘과 마찬가지로 압축률과 수행시간에 대한 비교를 진행한다. JM18.4[50]의 H.264/AVC의 intra mode와 inter mode를 모두 사용하며 기존의 H.264/AVC의 움직임 추정 알고리즘과 비교를 진행한다. 위에서 설명한 바와 같이 본 실험에서는 그룹 움직임 추정 방법에 의해서 확인된 움직임이 없는 블록(Same Block) 과 움직임이 확인된 블록(Motion Block)을 제외한 새로운 블록(New Block) 에 대하여 H.264/AVC의 압축방법을 사용하였다.

새로운 블록을 압축하는 방법은 H.264/AVC의 full search 움직임 추정 방법을 사용하였으며 비교 블록은 16x16 만을 사용하였다. 또한 탐색영역의 크기는 16으로 진행하였다.

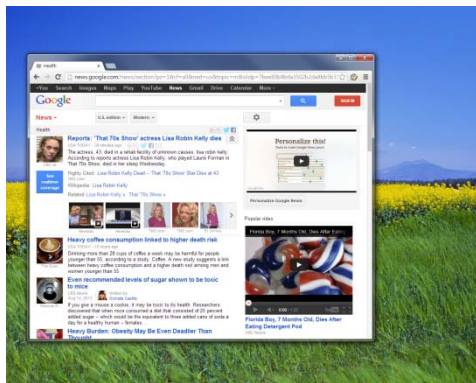
그림 5-10은 본 실험에 사용된 4개의 실험영상이다. 그림 5-10(a)는 윈도우의 움직임이 원형을 그리는 움직임을 나타내며 그림 5-10(b)는 윈도우의 움직임이 대각선을 이룬다. 그림 5-10(c)는 윈도우가 좌우로 빠르게 움직이는 영상이다. 그림 5-10(d)는 여러 개의 윈도우 중에서 하나의 윈도우가 원형을 그리는 움직임을 나타낸다. 각 영상은 빠르게 움직인 움직임과 느리게 움직이는 움직임이 혼합되어 있다. 각 영상의 평균 움직임과 최대 최소 움직임은 표 5-1에 나타내었다.



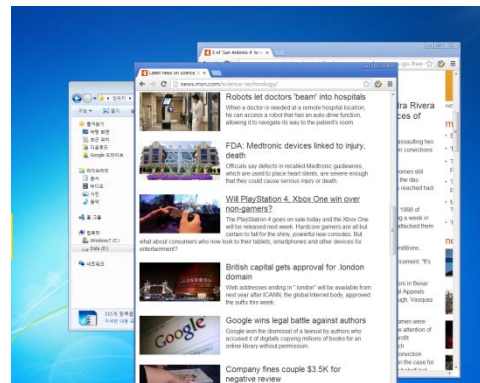
(a) 실험영상1



(b) 실험영상2



(c) 실험영상3



(d) 실험영상4

그림 5-10 그룹 움직임 추적 방법 실험영상

4개의 실험영상은 일반적으로 사용할 수 있는 compound image에서의 사용환경을 기준으로 선정하였으며 각 영상은 텍스트 블록이 15% ~ 40% 정도 포함되어 있고, 나머지 부분은 이미지 블록으로 구성되었다. 각 영상은 QP 값이 16으로 선택하여 압축되었고, 영상당 60frame 기준으로 실험을 진행하였다. QP값이 16으로 선정된 이유는 텍스트 영역의 경우 가독성을 유지하기 위하여 화질의 PSNR이 40dB 이상을 유지해야 하기 때문이다.

표 5-1 실험영상의 움직임 (pixels)

	최대	최소	평균
실험영상1	166	0	44
실험영상2	300	0	60
실험영상3	333	0	106
실험영상4	116	0	36

표 5-1 은 실험영상의 최대, 최소, 평균 움직임이다. Compound image에서의 움직임은 평균 움직임 보다 최대 움직임이 압축률에 영향을 미친다. 이러한 움직임의 폭은 H.264/AVC에서 사용하는 탐색 영역을 벗어나게 된다. 탐색 영역이 벗어나는 경우 압축률이 떨어지게 된다. 본 실험에서는 결과에 대한 비교를 위하여 H.264/AVC의 inter, intra mode를 모두 사용한다. 하지만 그룹 움직임 추정 방법에 대해서는 H.264/AVC inter mode의 계산량을 줄이고자 움직임 추정 블록을 16x16에 대해서만 진행한다. 그룹 움직임 추정 방법과의 비교를 위하여 H.264/AVC의 움직임 추정 방법 중 가장 속도가 빠른 EPZS 방법[39]과 가장 정확한 Full search 방법을 가지고 비교를 진행한다. EPZS방법과 Full search 방법은 inter mode의 비교 블록을 모두 포함하고 있다. 표 5-2는 실험영상에 대하여 30frame 기준의 평균 데이터 량이다. 표 5-3을 통하여 각 움직임 추정방법에 대한 전체 수행시간을 확인하고 표 5-4를 통하여 움직임 추정 시간을 나타내었다.

표 5-2와 같이 그룹 움직임 추정방법은 데이터의 발생량이 Full search 방법에 비교하여 거의 차이가 없음을 알 수 있다. 이는 H.264의 압축 방

식상 움직임이 있는 블록과 새로운 영역 사이에서 발생하는 움직임 예측 비트를 제외하고 full search 방법과 동일하게 데이터를 발생시킴을 알 수 있다. 또한 전체 수행시간 측면에서 보면 가장 빠른 EPZS 알고리즘보다 최소 10% 정도 수행시간이 빠르며, full search 알고리즘에 비해 비교 회수가 적기 때문에 수행시간이 빠르다. 움직임 추정시간도 픽셀과 블록 기반의 비교 방법에 비하여 그룹의 움직임을 추정하고 픽셀의 비교가 적은 그룹 움직임 추정방법이 가장 빠른 것을 확인할 수 있다.

이러한 이유는 compound image의 경우 움직이는 영역이 단순하고 움직임이 크기 때문에 전체 영상에서 움직임이 차지하는 비중이 약 70% 이상이 된다. 따라서 탐색 영역이 크지 않는 이상 기존 알고리즘은 탐색 영역 밖의 움직임에 대해서 압축률이 저하된다. 그룹 움직임 추정 방법의 경우는 전체 움직임이 이동하는 방향을 예측하여 탐색영역을 최소화 함으로써 전체 압축률 및 속도향상에 기여할 수 있다.

표 5-2 탐색 알고리즘별 압축율 (kbit/s @ 30frames)

	Proposed	Full search	EPZS	EPZS
Search range	16	64	64	512
실험영상1	1662.34	1731.67	2424.70	1917.40
실험영상2	16589.53	18489.23	23188.79	18202.51
실험영상3	11614.93	21419.36	32925.88	18867.79
실험영상4	6176.47	5908.26	6661.71	6254.59

표 5-3 탐색 알고리즘별 수행시간 (sec @ 60frames)

	Proposed	Full search	EPZS	EPZS
Search range	16	64	64	512
실험영상1	134.03	2604.21	227.51	234.27
실험영상2	164.84	3538.97	297.26	301.83
실험영상3	157.65	3590.95	310.16	314.03
실험영상4	222.68	2429.82	254.35	258.06

표 5-4 탐색알고리즘별 움직임 추정시간 (sec @ 60frames)

	Proposed	Full search	EPZS	EPZS
Search range	16	64	64	512
실험영상1	14.80	2360.47	20.39	20.88
실험영상2	25.44	3250.44	33.33	36.72
실험영상3	21.33	3309.59	32.10	36.43
실험영상4	27.33	2248.58	20.43	19.52



## 제6장 결 론

본 연구에서는 sub-pixel gradient 방법을 사용한 compound image 시스템에서 compound image 압축에 사용되는 각 단계의 방법을 제안하였다. 기존의 compound image 압축 알고리즘 연구는 각 단계의 압축 알고리즘의 개선을 우선시하여 전체 시스템으로서의 구현이 쉽지 않았다. 본 연구에서는 compound image 전체 시스템에서의 inter, intra 방식의 압축을 설명하고 실험 결과로 확인하였다.

첫째로 sub-pixel gradient 방법을 이용한 블록 분류 방법을 통하여 기존 알고리즘보다 정확한 텍스트와 이미지 블록을 분류할 수 있는 방법을 제시하였다. 기존 알고리즘이 약 80% 수준의 분류 확률을 가지는 반면에 본 연구에서 제시하고 있는 알고리즘의 경우 복잡한 텍스트 영상에서도 99%의 정확도를 가지는 것으로 실험을 통하여 확인하였다. 이러한 분류의 정확도는 compound image 안에서 생길 수 있는 블록간의 화질 저하 문제를 해결할 수 있다. 텍스트 내부에 다른 코딩 방법을 통하여 압축하게 되면 블록간의 압축방법의 차이로 인한 블록 경계에서의 화질 왜곡 현상이 일어난다. 이러한 왜곡 현상은 텍스트의 가독성 저하로 나타날 수 있다.

둘째로 sub-pixel gradient 방법을 이용한 텍스트 블록 압축 방식을 통하여 기존의 whole-pixel 을 사용한 압축 방식 보다 평판 디스플레이의 특성에 적합한 코딩 방법으로 접근을 시도하였다. 특히 제안하는 텍스트 블록

의 압축은 동일 PSNR에서도 양자화를 통한 방법과 비교하여 화질의 차이를 보이게 된다. 이러한 특성은 텍스트가 가지고 있는 가독성의 관점에서 확인을 할 수 있다. 영상의 픽셀 변화량이 큰 텍스트 image는 자연 영상과 다르게 조그마한 변화에도 큰 차이를 느낄 수 있다. 본 연구에서는 이러한 문제점을 화질 측면뿐만 아니라 가독성 측면에서도 화질의 저하가 적은 텍스트 알고리즘을 설명하였다. 제안하는 텍스트 블록 압축 알고리즘의 경우 JPEG2000, H.264/AVC intra mode 와 같은 표준 알고리즘보다 약 2 ~ 12 dB 의 PNSR의 향상을 보였고, 기존의 whole-pixel 텍스트 블록 압축 방식 보다 약 80% 압축 효율 개선을 실험을 통하여 확인하였다. 특히 동일 PSNR에서도 영상의 가독성이 나아짐을 확인하였다. 이러한 이유는 영상의 생성방법을 기준으로 압축을 진행하였기 때문에 픽셀과 픽셀 사이에서 나타날 수 있는 영상의 차이를 최소화 할 수 있다. 특히 문서에서의 가독성은 기존 알고리즘에 비하여 압축률 변화에 의한 왜곡도의 변화가 적게 된다. 이는 텍스트가 가지는 Edge 변화에 의한 시각효과를 최소화 함으로서 높은 압축률에서 가질 수 있는 시각적 변화량을 최소화 한 것이다.

셋째로 compound image에서의 시간적 연관성을 이용한 그룹 움직임 추정을 확인하였다. 기존의 H.264 와 같은 알고리즘의 경우 자연 영상의 움직임 추정을 목적으로 하였다면 본 연구에서의 그룹 움직임 추정 방식은 compound image의 특성을 이용하여 기존 방법의 복잡도 낮출 수 있는 장점을 가진다. Compound image는 사람의 눈에 의해 확인되는 영상이기 때문에 영상의 연속성을 확보해야 한다. 이는 Compound image 의 전송 환경

상 데이터의 평균 압축률도 중요하지만 최대 압축률도 중요하게 된다. 특히 컴퓨터 화면의 재생 빈도수는 최소 초당 60장이상을 확보하게 되어 있다. 이를 네트워크 환경과 연결하여 생각하면 장당 전송속도는 16ms 이하로 전송되어야 한다. 데이터의 최대 압축시간과 압축 데이터는 이러한 전송시간의 제약을 가지게 된다. 기존 움직임 추정 방법에 비해 최소 10% 속도 향상을 확인하였으며, 압축률도 영상에 따라 최소 30% 이상의 향상 효과를 확인하였다.

## 참 고 문 헌

- [1] “Draft ITU-T recommendation and final draft international standard of joint video specification (ITU-T Rec. H.264/ISO/IEC 14 496-10 AVC,” in Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, JVT-G050, 2003.
- [2] W. P. Pennebaker and J. L. Mitchell, *JPEG: Still Image Compression Standard*, Van Nostrand Reinhold, 1993.
- [3] D. Taubman and M. Marcellin, *JPEG2000: Image Compression Fundamentals, Standards, and Practice*, Kluwer Academic Publishers, 2001.
- [4] D. E. Knuth, "Dynamic huffman coding," *Journal of algorithms*, vol. 6, no. 2, pp. 163-180, Jun. 1985.
- [5] D. Pountain, "Run-length encoding," *Byte*, vol. 12, no. 6, pp. 317-319, Jun. 1987.
- [6] T. Boutell, "PNG (Portable Network Graphics) Specification Version 1.0," RFC 2083, Mar. 1997.
- [7] M. R. Nelson, "LZW data compression," *Dr. Dobb's Journal*, vol. 14, no. 10, pp. 29-36, Oct. 1989.
- [8] L. P. Deutsch, "DEFLATE compressed data format specification version 1.3," RFC 1951, May. 1996.
- [9] “Mixed raster content (MRC),” ITU-T Study Group 8, Draft Recommendation T.44, May. 1997.
- [10] R. de Queiroz, R. Buckley and M. Xu, “Mixed raster content (MRC) model for compound image compression,” in *Proc. of Visual Communications and Image Processing '99*, vol. 3653, pp. 1106–1117, Dec. 1999.

- [11] G. Feng and C. A. Bouman. "High-Quality MRC Document Coding," IEEE Trans. on Image Processing, vol. 15, no. 10, pp. 3152-3169, Oct. 2006.
- [12] R.L. de Queiroz, "Pre-Processing for MRC Layers of Scanned Images," 2006 IEEE International Conference on Image Processing, pp. 3093–3096, Oct. 2006.
- [13] R.L. de Queiroz, "On data filling algorithms for MRC layers," in Proc. of 2000 International Conference on Image Processing, Vol. 2. pp. 586-589, Sep. 2000.
- [14] D. Mukherjee, C. Chrysafis and A. Said, "JPEG2000-matched MRC compression of compound documents," in Proc. of 2002 International Conference on Image Processing, vol. 3, pp. 73-76, 2002.
- [15] L. Bottou, P. Haffner, P. G. Howard, P. Simard, Y. Bengio and Y. LeCun, "High quality document image compression with "DjVu"," Journal of Electronic Imaging, vol. 7, no. 3, pp. 410-425 , Jul. 1998.
- [16] T. Wiegand, G. J. Sullivan, G. Bjontegaard and A. Luthra, "Overview of the H. 264/AVC video coding standard," IEEE Trans. on Circuits and Systems for Video Technology, vol. 13, no. 7, pp. 560-576, Jul. 2003.
- [17] B. Zeng and J. Fu. "Directional Discrete Cosine Transforms-A New Framework for Image Coding," IEEE Trans. on Circuits and Systems for Video Technology, vol. 18, no. 3, pp. 305-313, Mar. 2008.
- [18] Y. L. Lee, K. H. Han and G. J. Sullivan, "Improved lossless intra coding for H. 264/MPEG-4 AVC," IEEE Trans. on Image Processing, vol. 15, no. 9, pp. 2610–2615, Sep. 2006.
- [19] T. Lin and P. Hao, "Compound image compression for real-time computer screen image transmission," IEEE Trans. on Image Processing, vol. 14, no. 8, pp. 993–1005, Aug. 2005.
- [20] C. Lan, G. Shi and F. Wu, "Compress Compound Images in H. 264/MPGE-4 AVC by Exploiting Spatial Correlation," IEEE Trans. on Image Processing, vol. 19, no. 4, pp. 946–957, Apr. 2010.

- [21] Z. Pan, H. Shen, Y. Lu, S. Li and N. Yu, "A Low-Complexity Screen Compression Scheme for Interactive Screen Sharing," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 23, no. 6, pp. 949–960, Jun. 2013.
- [22] W. Ding, D. Liu, Y. He and F. Wu, "Block-based Fast Compression for Compound Images," 2006 IEEE International Conference on Multimedia and Expo, pp. 809–812, Jul. 2006.
- [23] H. Shen, Y. Lu, F. Wu and S. Li, "A high-performanance remote computing platform," IEEE International Conference on Pervasive Computing and Communications 2009, pp. 1-6, Mar. 2009.
- [24] Z. Pan, H. Shen, Y. Lu and S. Li, "Browser-friendly hybrid codec for compound image compression," 2011 IEEE International Symposium on Circuits and Systems, pp. 101–104, May. 2011.
- [25] A. Said, "Compression of compound images and video for enabling rich media in embedded systems," *Electronic Imaging 2004*, International Society for Optics and Photonics, vol. 5308, pp. 69-82, Jan. 2004.
- [26] C. Lan, J. Xu and F. Wu, "Compression of compound images by combining several strategies," 2011 IEEE 13th International Workshop on Multimedia Signal Processing, pp. 1-6, Oct. 2011.
- [27] T. Lin and S. Wang, "Cloudlet-screen computing: A multi-core-based, cloud-computing-oriented, traditional-computing-compatible parallel computing paradigm for the masses," IEEE International Conference on Multimedia and Expo 2009, pp. 1805–1808, Jun. 2009.
- [28] W. Ding, Y. Lu and F. Wu, "Enable Efficient Compound Image Compression in H. 264/AVC Intra Coding," IEEE International Conference on Image Processing 2007, vol. 2, pp. 337-340, Sep. 2007.
- [29] S. Gleichman, B. Ophir, A. Geva, M. Marder, E. Barkan and E. Packer, "Detection and Segmentation of Antialiased Text in Screen Images," 2011 International Conference on Document Analysis and Recognition, pp. 424–428, Sep. 2011.

- [30] S. Wachenfeld, H.-U. Klein and X. Jiang, "Recognition of Screen-Rendered Text," 18th International Conference on Pattern Recognition 2006, vol. 2, pp. 1086-1089, Aug. 2006.
- [31] L. Fang, O. C. Au, K. Tang and A. K. Katsaggelos "Antialiasing Filter Design for Subpixel Downsampling via Frequency-Domain Analysis," IEEE Trans. on Image Processing, vol. 21, no. 3, pp. 1391-1405, Mar. 2012.
- [32] J. Sheedy, Y. Tai, M. Subbaram, S. Gowrisankaran and J. Hayes, "ClearType sub-pixel text rendering: Preference, legibility and reading performance," Displays, vol. 29, no. 2, pp. 138-151, Mar. 2008.
- [33] C. Betrisey, J. F. Blinn, B. Dresevic, B. Hill, G. Hitchcock, B. Keely, D. P. Mitchell, J. C. Platt and T. Whitted, "20.4: Displaced Filtering for Patterned Displays," SID Symposium Digest of Technical Papers, vol. 31, no. 1, pp. 296-299, Blackwell Publishing Ltd, May. 2000.
- [34] J. Farrell, J. Xu, K. Larson and B. Wandell, "47.2: Visual Preference for ClearType Technology," SID Symposium Digest of Technical Papers, vol. 40, no. 1, pp. 702-705, Blackwell Publishing Ltd, Jun. 2009.
- [35] T. W. Phinney, "TrueType, PostScript Type 1, & OpenType: What's the difference," 2001.
- [36] T. Porter and T. Duff, "Compositing digital images," ACM Siggraph Computer Graphics, vol. 18, no. 3, pp. 253-259, Jul. 1984.
- [37] B. Han, D. Wu and H. Zhang. "Block-based method for real-time compound video compression," SPIE Defense, Security, and Sensing, International Society for Optics and Photonics, vol. 7708, pp. 77080S-77080S-8, Apr. 2010.
- [38] H. S. Oh and H. K. Lee, "Block-matching algorithm based on an adaptive reduction of the search area for motion estimation," Real-Time Imaging, vol. 6, no. 5, pp. 407-414, Oct. 2000.
- [39] A. M. Tourapis, "Enhanced predictive zonal search for single and multiple frame motion estimation," Electronic Imaging 2002, International Society for Optics and Photonics, Jan. 2002.

- [40] H. Shen, Y. Lu, F. Wu and S. Li, "Low-cost realtime screen sharing to multiple clients," 2010 IEEE International Conference on Multimedia and Expo, pp. 980-985, Jul. 2010.
- [41] X. Li and S. Lei. "Block-based segmentation and adaptive coding for visually lossless compression of scanned documents," in Proc of. 2001 International Conference on Image Processing, vol. 3, pp. 450-453, Oct. 2001.
- [42] A. Said and A. Drukarev, "Simplified segmentation for compound image compression," in Proc. of 1999 International Conference on Image Processing, vol. 1, pp. 229-233, Oct. 1999.
- [43] A. Zaghetto and R. L. de Queiroz, "Segmentation-driven compound document coding based on H. 264/AVC-INTRA," IEEE Trans. on Image Processing, vol. 16, no. 7, pp. 1755-1760, Jul. 2007.
- [44] Y. Lu, S. Li and H. Shen, "Virtualized screen: A third element for cloud-mobile convergence," IEEE MultiMedia, vol. 18, no. 2, pp. 4-11, Feb. 2011.
- [45] C. Yang, Y. Niu, Y. Xia and X. Cheng, "A Fast and Efficient Codec for Multimedia Applications in Wireless Thin-Client Computing," IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks 2007, pp. 1-12, Jun. 2007.
- [46] M. Nakao, M. Kita and T. Matsui, "50.3: Low Delay and Real-time Image Transmission Hardware for Remote Desktop," SID Symposium Digest of Technical Papers. vol. 38, no. 1, pp. 1547-1550, Blackwell Publishing Ltd, May. 2007.
- [47] R. Li, B. Zeng and M. L. Liou, "A new three-step search algorithm for block motion estimation," IEEE Trans. on Circuits and Systems for Video Technology, vol. 4, no. 4, pp. 438-442, Aug. 1994.
- [48] J. Y. Tham, S. Ranganath, M. Ranganath and A. A. Kassim, "A novel unrestricted center-biased diamond search algorithm for block motion estimation," IEEE Trans. on Circuits and Systems for Video Technology, vol. 8, no. 4, pp. 369-377, Aug. 1998.



- [49] S. Wang, J. Fu, Y. Lu, S. Li and W. Gao, "Content-aware layered compound video compression," 2012 IEEE International Symposium on Circuits and Systems, pp. 145-148, May. 2012.
- [50] Reference Software of H.264/AVC 2013 [Online]. Available: [http://iphome.hhi.de/suehring/tml/download/old\\_jm/](http://iphome.hhi.de/suehring/tml/download/old_jm/), version jm18.4
- [51] M. D. Adams. The JasPer Project Home Page. [Online] Available: [www.ece.uvic.ca/~mdadams/jasper/](http://www.ece.uvic.ca/~mdadams/jasper/)



## **Abstract**

As computing performance and network speed increase, compound image compression can be used in many applications including video and multi communication. However, a compound image consists of many different types of image contents, which needs classification to identify the types of image contents and to apply different processing methods to process each type of image contents. Even if images of non-text type are handled by image coders such as JPEG or H.264/AVC, classification of different types of images and proper compression suitable to text images are needed to encode compound image.

Compound image classification is used to identify a text area from different types of images. Wrong text area classification causes degraded image quality due to inappropriate coding method in neighboring blocks. Because flat display uses sub-pixel color differentiation to maintain smoothness in text, this paper proposes sub-pixel gradient block classification using the inverse operation of the text creating method. Previous whole pixel-based image classifications cannot classify accurately between text areas and image areas. This paper uses block classification based on the sub-pixel gradient characteristics to achieve correct classification between text areas and image areas. Experimental results show that the proposed classification achieves the better block classification performance than previous

approaches.

Text encoding based on transformation and quantization loses a significant amount of text characteristics because of the loss of high frequency information in text images. On the other hand, lossless text encoding generates a large amount of data, which require very high bandwidth for transmission. The proposed sub-pixel gradient coding (SPGC) uses the characteristics of text on a flat panel display to encode gradients in texts. This method improves text image quality and readability. In the same compression ratio, the proposed method is tested to prove that image quality and readability is better than those of the previous compound image encoding method.

Compound image has characteristics which are simple movement and less noise than natural image. These characteristics make it possible to use less complicated motion estimation than natural image motion estimation. This paper uses these characteristics to propose block based motion estimation which classifies the type of image first and merges same type of blocks to simplify motion estimation. Group-based motion estimation is proven to require less search areas and computation time than previous methods.

**Key word :** sub-pixel gradient, text coding, group motion estimation, text compression

**Student number :** 2009-30179