

손실된 매크로 블록의 움직임 벡터 복원에 관한 연구

김용우, 서덕원, 곽훈성
전북대학교 컴퓨터공학과
e-mail : {kyw, duck0419, hskwak}@chonbuk.ac.kr

A Study on Motion Vector Recovery of Loss Macroblock

Yong Woo Kim, Duck Won Seo, Hoon Sung Kwak
*Dept. of Computer Engineering, Chonbuk University

요 약

제안한 방법은 H.264 옵티컬 플로워 방법을 이용하여 손실된 매크로블록의 움직임 벡터를 4x4 단위로 복원하는 알고리즘을 제안했다. 제안하는 알고리즘은 옵티컬 플로워 기반의 움직임벡터 복원 기법에서의 문제점인 높은 복잡도를 해결하기 위해서 최소 4x4 단위에서의 움직임 벡터 정보를 포함하는 H.264 부호화 표준의 특성을 이용하여 정확한 초기값에서 출발함으로써 계산량을 줄임과 동시에 Optical Flow Region을 4개의 16x16 크기로 제한함으로써 기존보다 복잡도를 크게 줄일 수 있었다.

1. 서론

H.264/AVC 비디오 부호화 방식은 이전의 비디오 부호화 표준들에는 없었던 새로운 부호화 도구들이 추가되었으며, 이러한 도구들은 H.264 부호화 표준이 이전의 부호화 표준과 비교하여 보다 높은 압축효율을 갖게 하였다. 이렇게 추가된 여러 새로운 부호화 도구들 중 이전의 부호화 표준들에 비해 보다 작은 블록단위에서의 움직임 추정은 인접한 블록들의 움직임벡터간의 상관도를 높이는 결과를 가져오는데, 이러한 H.264의 특징은 손실된 매크로블록의 움직임 벡터 복원 시에 유용하게 사용될 수 있다.

MPEG-2가 제정된 이후로 손실된 매크로블록 단위의 움직임 벡터 복원에 관한 연구들이 진행되어 왔다. 가장 대표적인 움직임 벡터 복원에 관한 방법에는 손실된 블록의 주변 화소값을 이용한 BMA(Boundary Matching Algorithm) 기법이 있으며, 이를 수정 보완한 여러 가지 방법들이 제안되었

다. 현재 ITU-T에서 권고하고 JM10.1[1]에 구현되어 있는 움직임벡터 복원기법 또한 이러한 BMA 기반의 방법을 사용한다. 하지만 이와 같은 BMA에 기반을 하고 있는 움직임벡터 복원 방법들은 올바른 움직임벡터가 선정된 경우에는 오류를 포함한 매크로 블록과 주변 매크로블록들 사이의 블록화 현상이 발생하여 화질을 현저히 떨어뜨릴 수 있다는 단점을 가진다. 이러한 단점을 극복하기 위한 방법으로서 블록단위가 아닌 화소단위의 움직임 벡터의 복원을 통해 오류를 줄이는 방법이 Zheng에 의해 제안되었으며 이는 주변 매크로블록의 움직임벡터 정보를 이용하여 Lagrange 보간 방법을 통한 화소 단위의 움직임벡터를 복원해 내는 방법이다[5]. 하지만 이러한 움직임 벡터 복원 기법은 작은 영역에서의 움직임이 발생하였을 경우 높은 성능을 보이지 못한다는 단점을 가진다.

제안한 방법은 옵티컬 플로워를 이용하여 손실된 매크로블록의 움직임 벡터를 4x4 단위로 복원하여 오류를 줄이는 알고리즘을 제안했다[3]. 옵티컬 플

* 이 연구에 참여한 연구자는 2단계 BK21사업의 지원비를 받았음.

로워 기반의 움직임벡터 복원 기법에서의 문제점인 높은 복잡도를 해결하기 위해서 제안하는 알고리즘은 최소 4×4 단위에서의 움직임 벡터정보를 포함하는 H.264 부호화 표준의 특성을 이용하여 비교적 정확한 초기값에서 출발함으로써 계산량을 줄임과 동시에 Optical Flow Region을 4개의 16×16 크기로 제한함으로써 기존보다 복잡도를 크게 경감시킨다[6].

논문의 구성은 2장에서 제안한 움직임 벡터 복원 기법에서 사용한 옵티컬 플로워의 기본적인 개념을 소개하고 3장에서는 제안하는 알고리즘을 설명하고 4장에서는 제안하는 알고리즘을 기존의 움직임 벡터 복원 기법과의 비교를 통해서 제안하는 알고리즘의 효율성을 보이고 5장에서는 결론을 논한다.

2. 옵티컬 플로워의 개요

(1) H.264/AVC 표준안 개요

H.264/AVC 표준안을 개발하는 과정에서 상대적으로 새로운 기술들이 많이 채택 되었는데, 작은 블록 사이즈를 사용하는 가변 블록 사이즈 움직임 보상, 1/4-샘플(quarter-sample) 정확도의 움직임 보상, 픽처의 경계를 벗어나는 움직임 벡터, 여러 개의 참조 픽처를 사용하는 움직임 보상, 참조 순서와 디스플레이 순서를 구분, 픽처의 표현 방법과 참조를 위해 사용하는 능력을 구분, 가중 예측(weighted prediction), 움직임 추정의 생략과 직접(direct) 움직임 추정의 향상, 인트라 부호화를 위한 방향성 공간 예측(directional spatial prediction) 압축 방법과 같이 압축효율을 증가시키기 위해서 향상된 예측 방법들이 채택되었다[5].

작은 블록 사이즈 변환(small block-size transform), 계층적 블록 변환(hierarchical block transform), 짧은 워드 길이 변환(short word-length transform), 정확한 일치 변환(exact-match transform), 산술 엔트로피 부호화(arithmetic entropy coding), 컨텍스트 적응형 엔트로피 부호화(context-adaptive entropy coding)은 향상된 예측 방법 외에도 압축 효율을 향상시키기 위해 포함되었다[5].

(2) 옵티컬 플로워의 개요

2-1 화소값 불변의 제약 조건

옵티컬 플로워란 프레임내의 화소들의 위치변화에

대한 속도를 2차원으로 표현한 것을 의미한다[2]. 이는 시간이 지남에 따라 변화하는 화소들은 화소의 위치만 변화될 뿐 그 값은 동일하게 유지된다는 화소값 불변의 제약조건(brightness invariance constraint)에서 출발한다. 그래서 두 프레임간의 시간차이를 안다면 옵티컬 플로워 속도 벡터를 구함으로써 다음 프레임에서의 각 화소의 위치를 알 수 있다.

2-2 평탄 제약 조건

화소값의 불변에 대한 제약 조건만으로는 옵티컬 플로워 속도벡터가 존재할 수 있는 직선의 방정식을 구할 수 있을 뿐 화소의 위치변화에 대한 정확한 속도벡터를 구할 수 없으며, 정확한 속도벡터를 구하기 위해서는 추가적인 제약조건이 필요하다. 추가적인 제약조건으로 주로 사용되는 것이 벡터의 평탄 제약 조건(smoothness constraint)이다. 이는 옵티컬 플로워 속도벡터의 성분에 대한 라플라시안 값들에 대한 제곱의 합을 최소로 하는 제약 조건을 의미한다.

2-3 1차 편미분 및 속도벡터의 라플라시안값의 근사화

1차 편미분의 근사값을 구하기 위한 대표적인 방법으로서 인접한 여덟 개의 화소값을 이용하여 입방체의 중심점에서의 1차 편미분의 근사값을 구함에 있어 단일 화소위치에서의 기울기가 아닌 $2 \times 2 \times 2$ 단위의 입방체에서의 기울기의 평균으로 근사화 하는 이유는 잡음의 영향을 최소화하기 위함이다.

2-4 속도 벡터의 계산

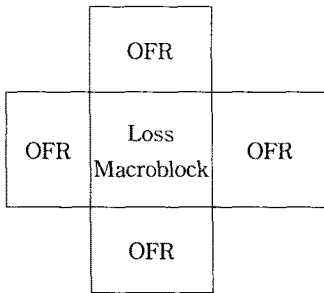
속도 벡터는 두개의 제약조건에 대한 오차를 최소화하는 값을 구함으로써 계산할 수 있다. 먼저 첫 번째 제약 조건인 화소값 불변 제약조건에서의 오차와 두 번째 제약 조건인 평탄제약조건에 대한 오차를 최소화하는 값을 선택함으로써 속도벡터를 구할 수 있다.

3. 제안한 알고리즘

연속되는 비디오 프레임 사이의 각 픽셀의 궤적(trajjectory)을 예측하여, 광의 흐름(optical flow)이라고 알려진 픽셀 궤적의 그림을 만들어 내는 것이 가능하다. 광의 흐름도를 정확하게 알 수 있다면, 참조 프레임의 각 픽셀을 광 흐름 벡터를 따라 이동시킴으로써 현재 프레임의 픽셀을 정확하게 예측하는

것이 가능하다. 그러나 여러 가지 이유로 인해 이것은 일반적인 움직임 보상의 방법이 아니다. 광 흐름을 정확하게 계산하기 위해서는 매우 많은 연산을 필요로 하며 보다 정확한 방법일수록 모든 픽셀에 대해 반복적인 과정을 거친다. 디코더가 예측 프레임의 다시 생성할 수 있도록 하기 위해서는 모든 픽셀에 대한 광 흐름 벡터를 디코더로 전송해 주어야 한다. 결과적으로 많은 양의 데이터를 전송하게 되므로, 오차가 적어짐으로 인해 얻을 수 있는 장점을 상실하게 된다.

H.264 옵티컬 플로워를 이용한 움직임 벡터 복원을 위해서 제안하는 알고리즘은 먼저 OFR(Optical Flow Region)을 정의한다. OFR의 크기는 속도벡터의 정확성 및 연산량과 직접 관련 되어 있으며, OFR의 크기가 크면 보다 정확한 속도벡터를 구할 수 있으나 연산량이 많아지고, 반대로 작은 크기의 OFR은 상대적으로 적은 연산량을 가지지만 속도벡터의 정확성이 떨어진다. 하지만 만일 속도벡터의 초기값이 비교적 정확하다면 작은 크기의 OFR에서도 충분히 정확한 속도벡터를 구해낼 수 있다.

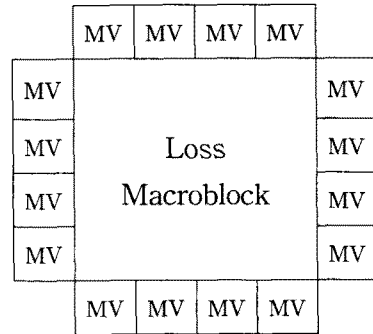


(그림 1) 제안한 알고리즘에서 사용한 OFR(Optical Flow Region)

H.264의 여러 부호화 특성 중 최소 4x4단위의 움직임 추정으로 인하여 각 매크로블록은 이전보다 세밀한 움직임 정보를 포함하며, 이는 속도벡터의 초기값을 이전의 여러 부호화 표준들과 비교하여 보다 정확히 유추할 수 있음을 의미한다. 따라서 제안한 알고리즘은 이러한 H.264의 부호화 특성을 이용하여 움직임 벡터 복원에 사용되는 Optical Flow Region을 손실된 매크로블록의 상하좌우에 위치한 16x16 크기로 한정하였다.

제안한 알고리즘은 손실된 매크로블록의 움직임 벡터를 4x4 크기의 블록 단위로 복원하며, 네 개의

독립된 Optical Flow Region로부터 계산된 속도 벡터 중 손실된 매크로블록에 바로 인접한 화소 위치,



(그림 2) 속도벡터의 초기값 설정에 사용되는 주변 움직임 벡터

중 64개의 화소 위치에서의 속도벡터가 움직임 벡터 복원을 위해 사용되어진다. 또한 손실된 매크로블록의 움직임 벡터는 바깥쪽의 블록에서부터 안쪽 블록순서로 복원된다.

4. 실험 및 결과 분석

제안한 알고리즘의 효율성을 보이기 위한 실험은 움직임의 정도가 큰 QCIF 크기의 테스트 시퀀스를 대상으로 하였으며, 이는 움직임이 적은 시퀀스의 경우 이전 프레임에서의 동일 위치 블록으로 예측할 수 있는 시방향 복사방법(ZM) 또한 기존의 참조 소프트웨어(JM) 만으로도 좋은 성능을 보이기 때문이다. 실험에 사용된 시퀀스들은 Coastguard, Foreman, Mobile, Table Tennis 등이다. 제안한 알고리즘은 H.264 참조 소프트웨어(JM 10.1)에 적용하여 구현하였으며, 비교를 위해 시방향 복사 기법(ZM)과 기존의 H.264 참조 소프트웨어(JM)에 구현된 방법을 사용한 결과와 비교하였다.

표1은 실험에 사용된 시퀀스에 대한 오류율에 따른 PSNR 값을 비교하고 있다. 표1의 결과를 통해서 알 수 있는 바와 같이 제안하는 알고리즘은 다양한 비트율에서 이전의 움직임벡터 복원 기법들보다 우수한 성능을 보여 주고 있음 볼 수 있다.

<표1> 테스트 시퀀스에 대한 매크로블록 오류율에 따른 PSNR

(1) Coastguard

| PSNR | Error Ratio | | | |
|------|-------------|-------|-------|-------|
| | 5% | 10% | 15% | 20% |
| ZM | 35.47 | 34.32 | 33.56 | 32.01 |
| JM | 37.58 | 36.55 | 35.99 | 34.66 |
| OF | 39.85 | 38.89 | 37.12 | 36.68 |

(2) Foreman

| PSNR | Error Ratio | | | |
|------|-------------|-------|-------|-------|
| | 5% | 10% | 15% | 20% |
| ZM | 38.45 | 37.02 | 35.02 | 34.22 |
| JM | 39.76 | 38.11 | 37.45 | 36.09 |
| OF | 41.02 | 39.87 | 38.26 | 37.24 |

(3) Mobile

| PSNR | Error Ratio | | | |
|------|-------------|-------|-------|-------|
| | 5% | 10% | 15% | 20% |
| ZM | 36.45 | 34.97 | 33.26 | 31.98 |
| JM | 36.57 | 35.02 | 33.98 | 32.45 |
| OF | 37.89 | 36.23 | 35.09 | 34.03 |

(4) Table Tennis

| PSNR | Error Ratio | | | |
|------|-------------|-------|-------|-------|
| | 5% | 10% | 15% | 20% |
| ZM | 34.08 | 33.56 | 31.62 | 28.95 |
| JM | 37.29 | 36.25 | 34.75 | 33.05 |
| OF | 38.59 | 37.12 | 36.02 | 35.62 |

5. 결론

H.264/AVC 비디오 부호화 방식은 이전의 비디오 부호화 표준들에는 없던 새로운 부호화 도구들이 추가되었으며, 추가된 여러 새로운 부호화 도구들 중 이전의 부호화 표준들에 비해 보다 작은 블록단위에서의 움직임 추정은 인접한 블록들의 움직임벡터간의 상관도를 높이는 결과를 가져오는데, 이러한 H.264의 특징은 손실된 매크로블록의 움직임벡터 복

원 시에 유용하게 사용될 수 있다.

제안하는 알고리즘은 옵티컬 플로워 벡터를 구해서 손실된 매크로블록에서의 움직임 벡터를 복원하는데 사용된다.

실험 결과 ITU-T에서 권고하고 현재 H.264 참조 소프트웨어에 구현되어 있는 움직임 벡터 복원 방법과 비교하여 제안하는 알고리즘이 다양한 비트율에서 높은 성능을 보이는 것으로 나타났다.

참고문헌

- [1] JVT Model; <http://bs.hhi.de/~suehring/tml/download/jm1001.zip>
- [2] B. Horn and B. G. Schunk, "Determining optical flow", *Artificial Intelligence*, 17, 185-203, 1981.
- [3] Thomas Wiegand, Gary J. Sullivan, "Overview of the H.264/AVC Video Coding Standard", *IEEE Trans Circuits and Systems for Video Technology*, Vol.13, pp.560-576, July 2003.
- [4] Jinghong Zheng and Lap-Pui Chau, "A motion vector recovery algorithm for digital video vector recovery algorithm for digital video using lagrange interpolation", *IEEE Trans on Broadcasting*, Vol.49, pp.383-389, December, 2003
- [5] Lain E.G Richardson, "H.264 and MPEG-4", John Wiley & Sons Ltd., 2003
- [6] Gary Sullivan, " H.264/AVC Textbook, Sakae Kibbo, 2004
- [7] 호요성, "H.264 표준의 이해", 두양사, 2006