비트패턴을 이용한 고속 움핑 예측

강봉구* · 안재현**

요 약

본 논문에서는 비트패턴을 이용한 고속 움핑 예측을 제안한다. 움핑 예측은 이동하는 노드들의 움직임 벡터에서 움직임 정보를 가져오는 공간적인 의존성 때문에 최적의 움직임을 찾기 위해서는 반복적인 탐색을 요구한다. 반복적인 탐색에 의해 발생하는 연산량의 증가는 움핑 예측의 성능을 저해하는 큰 이유 중 하나이다. 본 논문에서 제안한 방법은 비트패턴을 이용해서 움직임 예측을 수행하기로 하는 생성적인 방법이다. 움직임이 없는 영역에서의 의미 없는 움직임 예측을 제거하고, 움직임이 있는 영역에서는 움핑 예측을 수행함으로써 빠른 움직임 예측이 가능하게 된다. 제안된 방법으로 실험한 결과 기존의 움핑 예측을 수행했을 때보다 성능은 극심하게도 연산량이 75% 이상 감소하였다.

Fast Warping Prediction using Bit-Pattern for Motion Estimation

Bong-Goo Kang* and Jae-Hyeong Ahn**

ABSTRACT

In this paper, we propose a fast warping prediction using bit-pattern for motion estimation. Because of the spatial dependency between motion vectors of neighboring node points carrying motion information, the optimization of motion search requires an iterative search. The computational load stemming from the iterative search is one of the major obstacles for practical usage of warping prediction. The motion estimation in the proposed algorithm measures whether the motion content of the area is or not, using bit-pattern. Warping prediction using the motion content of the area make the procedure of motion estimation efficient by eliminating an unnecessary searching. Experimental results show that the proposed algorithm can reduce more 75% iterative search while maintaining performance as close as the conventional warping prediction.

1. 서 론

동영상의 불만족 고속의 네트워크 환경이 구축되었고 이에 멀티미디어 데이터 통신이 가능하게 되었다. 현재의 기술은 멀티미디어 데이터의 시간적 차리를 충분히 만족스럽게 제공하기 때문이다. 이것은 해결하기 위해 멀티미디어 데이터의 동영상 압축과 압축 기술에 대한 연구가 활발히 진행되고 있으며, 동영상 압축도 그 중 하나이다. 동영상 압축은 동영상의 높은 압축률, 품질, 기간의 시간적 간격성을 제거하는 움직임 예측은 동영상 압축의 가장 핵심적인 부분이다.

현재 움직임 예측에는 간단하고 효과적인 방식인 블록 매칭 알고리즘(block matching algorithm; BMA) [1]이 주로 사용되고 있다. BMA는 H.261/263, MPEG-1/2에 채택되어있고 알고리즘에 앞서 한 블록 내의 모든 화소들이 동일한 움직임을 갖는다는 가정 하에 한 블록 당 하나의 움직임 벡터(motion vector; MV)를 할당한다. 그러나, 동영상 각각의 프레임을 일정한 규칙의 정방형 블록으로 나누어서 움직임 벡터를 찾고 또 움직임을 보상하는 과정에서 보스킹 잡음(blocking artifacts)[2]을 생성하게 되며,
블록 단위로 움직임을 보상하는 과정에서는 복잡한 계산에서 불연속적인 움직임이 존재할 때 필연적으로 나타나는 현상이다.

위평 예측(warping prediction)은 세어 각각 모간법(control grid interpolation: CGI)[2]이라고 불리기도 하며, BMA는 CGI의 특별한 예로서 보여질 수 있다. 위평 예측은 전체 프레임에서 움직임 벡터를 찾는데 있어서 BMA처럼 규칙화된 블록으로 나누는 것이 아니라 걸쳐져 있는 폴리곤(polygon)의 집합으로 그룹(mesh) 모양의 형태로 프레임을 분할 하게 된다. 위평 예측은 어떤 규칙화된 모양이 아닌 폴리곤의 형태로 프레임을 나누어 움직임 벡터를 찾기 때문에 BMA의 단점인 분석적 참조를 생성하지 않는다. 그러나, 위평 예측은 새로운 폴리곤의 형태로 찾기 위한 무수히 많은 반복적인 연산으로 연산량이 증가하는 단점이 있다.

본 논문에서는 위평 예측의 반복적인 연산량을 줄이기 위해서 비트패턴(bit-pattern)을 이용한 고속 위평 예측 알고리즘을 제안한다. 먼저 비트변환을 통해서 영상을 단순화한 후 프레임 내 움직임이 존재하는 영역을 찾아낸다. 움직임이 없는 영역에서는 단순화된 반복적인 연산량을 제거하고, 움직임이 탐색된 영역에는 위평 예측을 수행함으로써 기존의 위평 예측보다 빠른 움직임 예측을 한다.

2장은 위평 예측을, 3장은 비트패턴의 기본 개념을 설명하고, 4장은 제안 알고리즘을 설명한다. 5장에서는 실험결과를 통해 성능을 평가하고 6장에서 결론을 냈다.

\[ f(x, y) \text{와 } g(x, y) \text{가 변형되는 것은 } \hat{F}_n(x, y) \text{와 } \hat{F}_n(x', y') \text{ 간의 기하학적 관계에 의해 결정된다. 다음의 두 개의 함수 } f(x, y) \text{와 } g(x, y) \text{를 사용하여 각각의 폴리곤에 대한 움직임 벡터를 계산할 수 있으며, 양선형 변환(bilinear transform)을 사용할 경우 식 (2)와 같다.} \]

\[
\begin{align*}
    f(x, y) &= a_{1}x + a_{2}y + a_{3} + a_{4} \\
    g(x, y) &= a_{1}x + a_{2}y + a_{3} + a_{4}
\end{align*}
\]

변환 계수 \( a_{1}, a_{2}, a_{3}, a_{4} \) 는 노드들의 움직임 벡터로부터 계산된다. 폴리곤 형태로 나누는 방법은 양선형 변환이나 아핀 변환(affine transformation), 공간 변환(spatial transform) 등을 사용하며, 폴리곤의 모양은 삼각형이나 사변형 그룹 구조 등을 사용할 수 있다. 보통 위평 예측에서는 폴리곤 모양은 사변형 그룹 구조를 선택하고 변환 방법은 양선형 보간법(bilinear interpolation)을 사용하여 움직임을 보상한다[2].

그림 1은 노드에서 움직임 보상의 과정을 폴리곤 매핑을 보여준다. 노드 \( N \)의 움직임 벡터 \( MV \)는 움직임 벡터에 영향을 준다. 움직임 벡터 \( MV \)의 변화는 움직임 벡터를 포함하는 움직임 패턴을 변형시키고 따라서 움직임 보상을 위한 예측되어진 이미지를 변환시킨다.

2. 위평 예측

CGI라고도 알려진 위평 예측은 전체 프레임을 그 동안 볼리곤의 격자로 분할한 그림 구조(mesh structure)를 이루며 위평 예측에서 움직임 벡터는 노드(node)를 전달하며, 폴리곤 내부의 움직임 패턴(motion field)은 공간 변환(spatial transformation)과 분할된 노드들의 움직임 벡터로부터 구해진다. 위평 예측에서 예측되어진 프레임은 식 (1)과 같다. 현재 프레임 이미지 \( F_e \)에 대한 예측되어진 프레임 이미지 \( \hat{F}_e \)는 이전 프레임 이미지 \( \hat{F}_{e-1} \)에서 찾는다[5].

\[ \hat{F}_e(x, y) = \hat{F}_{e-1}(f(x, y), g(x, y)) \]
8bit 화소 값을 1bit로 표현한 이진 영상을 의미한다. 이는 탐색과정에서의 계산량을 줄여 빠른 탐색이 이루어지도록 하는 수단이다. 탐색범위와 탐색영역의 비트패턴 B는 식 (3)을 통해서 얻어진다.  

\[ B(x,y) = \begin{cases} 1, & \text{if } p(x,y) \geq M \\ 0, & \text{otherwise} \end{cases} \]

여기서 \( p(x,y) \)는 탐색블록 또는 탐색영역의 \((x, y)\)번째 화소값을 의미하고, \( M \)은 탐색블록의 평균 화소값을 의미한다. 비트패턴은 각각의 블록에 대해서 직관적인 블록의 정보를 나타낸다. 즉, 탐색블록의 비트패턴이 탐색영역 내의 어떤 블록의 비트패턴과 매우 유사하다면 원 영상에서의 두 블록도 유사한 이미지의 특징을 갖고 있다고 볼 수 있다.

\[ \text{DBD}(0,0) = \frac{1}{N \times N} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} |F_x(x,y) - F_y(x,y)|, \]

\[ \begin{cases} \text{if DBD}(0,0) \geq TH_{	ext{motion}}, & \text{motion searching} \\ \text{otherwise,} & \text{no motion searching} \end{cases} \]

차, DBD(0,0)가 특정 임계값 보다 작으면 움직임이 없다고 추정할 수 있으므로 그 영역에 대한 움직임 예측 과정을 생략할 수 있다. 이는 통해서 움직임이 없는 영역에서의 불필요한 연산을 제거할 수 있다. 일반 움직임이 있는 영역과 움직임이 없는 영역을 구분한 후 다음 단계로 웅직임이 없다고 판단되는 영역은 불필요한 연산을 할려고 움직임이 있는 영역을 폐기하는 메커니즘을 수행한다. 즉, DBD 값이 특정 임계값 보다 높을 경우 식 (5)을 통해 폐기할 메커니즘을 하는 것이다. 각 변형 함수 (disortion measure function) \( D_r \)는 각각의 노드에서의 후보 움직임 벡터 \( MV_r \)의 연산으로 얻어진다. 

\[ D_r = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} |F_x(x,y) - F_y(x,y)|, \]

본 논문에서 제안한 알고리즘의 과정은 다음과 같고 그림 3은 제안 알고리즘의 처리 과정을 순서대로 나타낸다.

1 단계 : 영상의 2진화 단계

잠재 프레임과 이전 프레임을 2진화한다. 영상의 단순화작업을 통해 다음 단계에서 프레임내 움직임 추적을 쉽게 수행하게 해준다.

2 단계 : 움직임 추적 단계

2진화된 프레임에서 움직임을 검색하여 움직임이 있는 영역과 없는 영역을 판단하고 폐기된 영역이 필요한 영역과 필요하지 않은 영역으로 나눈다.
• 3 단계: 폴리곤 매칭이 필요 없는 노드 결정 단계

응작임이 없다고 판단되는 영역은 폴리곤 매칭을 수행하지 않고 2단계에서 얻어진 불록의 노드를 최종 노드로 결정한다.

• 4 단계: 폴리곤 매칭을 통한 노드 결정 단계

응작임이 검출된 영역은 폴리곤 매칭을 통해 최적의 사변형 폴리곤을 찾아 노드의 응작임 벡터를 예측한다.

• 5 단계: 최종 응작임 벡터 결정 단계

3단계와 4단계 과정을 거쳐 최종적으로 전체 프레임의 그물 구조를 결정하고 각 노드에서의 응작임 벡터를 예측한다.

그림 3. 게임 알고리즘의 순서도

위의 예측과 제한한 알고리즘의 폴리곤 매칭을 한 예를 그림 4에서 보여준다. 응작임이 없는 영역에서의 반복적인 연산의 제거로 전체 연산량을 감소시킬 수 있다.

5. 실험 결과

본 논문에서 제한한 알고리즘을 평가하기 위해서 CIF(common intermediate format: 352x288 color pixels) 형식의 Clare영상과 Salesman영상을 각각 1번부터 106번 프레임까지 사용하였다. 성능을 비교

그림 4. 전체 프레임에서 알고리즘간의 폴리곤 매칭 비교

하기 위해서 전체 탐색 불록 매칭 알고리즘(full search block matching algorithm: FS-BMA)은 불록 크기를 16x16으로 하였고, CGI와 제한한 알고리즘은 사변형 그물 구조와 양선형 보간법(bilinear interpolation)을 사용하여 200개의 노드들이 존재하도록 했다. 응작임 예측은 일반적인 동영상 코딩의 참여 프레임의 주기를 적용하여 새 프레임을 간격으로 생성하였다. 본 논문에서는 알고리즘의 비교 분석을 위해 MSE(mean square error)를 구한 다음 식(6)을 사용해서 PSNR(peak signal to noise ration)을 얻었다.

\[
PSNR = 10 \log_{10} \frac{255^2}{MSE} \text{[dB]}
\]

表 1은 CGI와 제한 알고리즘간의 PSNR과 폴리곤 매칭 횟수의 평균값을 비교하였다. 그림 5는 FS-BMA, CGI 와 제한 알고리즘간의 PSNR을, 그림 6은 CGI 와 제한 알고리즘간의 폴리곤 매칭 횟수를 그래프로 비교하였다. 결과에서 본 논문에서 제한한 알고리즘이 CGI에 근접하는 성능을 가지고 있으며, 사용된 PSNR은 75% 이상 감소시킨 것을 알 수 있다.

表 1. 제한 알고리즘에 대한 성능 평가

<table>
<thead>
<tr>
<th></th>
<th>Clare Proposed</th>
<th>CGI Proposed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average PSNR</td>
<td>38.63882</td>
<td>38.21402</td>
</tr>
<tr>
<td>Average numbers of polygon matching/Frame</td>
<td>1062.34</td>
<td>242.86</td>
</tr>
<tr>
<td></td>
<td>1090.17</td>
<td>272.20</td>
</tr>
</tbody>
</table>

6. 결론

본 논문에서는 비트패턴을 이용한 고속 위성 예측
그림 5. PSNR 비교

그림 6. 프레임 내에서 폴리곤 매칭의 총 인산 횟수 비교

을 제안한다. 기존의 위형 예측은 프레임 전체에서 폴리곤 매칭을 사용하기 때문에 각 노드의 움직임 벡터를 찾기 위해 무수히 많은 반복적인 인산을 한다. 제안한 알고리즘은 반복적인 인산을 줄이고자 비트변환을 통해 영상을 먼저 전환 한다. 이전화하여 단순화된 영상을 가지고 프레임 내의 움직임 정보를 찾아 움직임이 있는 영역과 움직임이 없는 영역을 판단한다. 움직임이 없는 곳에서는 무의미한 폴리곤 매칭을 통한 노드의 움직임 벡터 찾기를 중지하고 분광을 사상형 폴리곤으로 사용한다. 움직임이 있는 영역에 한하여 폴리곤 매칭을 수행하여 노드에서의 움직임 벡터를 추적하여 움직임 예측을 한다. 즉, 움직임이 없는 영역에서는 반복적인 폴리곤 매칭을 제거함으로써 기존의 위형 예측에 비해 높은 인산량 감소를 가져온다.

Clare영상에서 폴리곤 PSNR은 제안한 알고리즘이 FS-BMA보다 1,4827 dB 정도 높고, CGI에 비해 0.44248 dB 정도 낮았다. 반복 인산 횟수에서는 Clare영상에서 프레임마다 폴리곤 매칭 횟수는 CG는 1062.34회이고, 제안한 알고리즘은 242.86회로 77.4% 감소하였으며, Salesman 영상의 경우 CGI는 1000.17회인 반면 제안한 알고리즘은 272.29회로 75.3%의 인산량 감소 효과를 가져왔다. 실험 결과에서 알 수 있듯이 제안 알고리즘은 FS-BMA보다 성능이 우수했고, CGI에 근접한 PSNR을 나타내면서도 폴리곤 매칭 횟수는 평균 75% 이상 감소하며 고속 움직임 예측이 가능하였다. 제안 알고리즘은 영상 내의 움직임의 변화가 작은 화상 화면이나 화상 전환이 같은 서브트럴 진송을 위한 움직임 예측에 효과적으로 적용할 수 있다.

참고 문헌


가봉구

1999년 충북대학교 정보통신공학과(학사)
2001년 충북대학교 정보통신공학과(석사)
2001년~현재 대신중건 전산본부

안재형

1981년 충북대학교 전기공학과 (학사)
1983년 한국과학기술원 전기 및 전자 공학과(석사)
1992년 한국과학기술원 전기 및 전자 공학과(박사)
1987~현재 충북대학교 전기 전자

공학부 교수

관심분야: 영상 통신 및 영상정보처리, 멀티미디어 제작 및 정보제공, 인터넷 통신 및 프로그래밍