

論文2001-38SP-11-1

동적 정규화 메쉬를 이용한 움직임 추정

(Motion Estimation Using Dynamic Regular Mesh)

李 東 奎 * , 李 斗 秀 *

(Dong Gyu Lee and Doo Soo Lee)

요 약

기존의 블록 정합 알고리즘에서는 움직임 벡터로 평면적인 움직임만을 기술할 수 있었고 이로 인해 블록 화 잡음이 발생하였다. 이러한 문제점을 해결하기 위해서 삼각형 메쉬를 이용한 움직임 추정방법이 제안되었다. 영상을 일정한 크기의 삼각형으로 분할하고 같은 연결 상태를 가지도록 하는 정규화 메쉬는 메쉬의 구조를 기술하기 위한 추가적인 정보가 필요 없으나 움직임이 많은 영역과 작은 영역을 같은 크기의 메쉬로 분할함으로써 성능을 저하시키게 된다. 본 논문에서는 정규화 메쉬의 형태를 유지하면서 움직임영역에 따라 메쉬의 크기와 연결상태가 가변될 수 있는 동적 정규화 메쉬를 사용한 움직임 추정방법을 제안한다. 실험결과를 통해 제안한 방법이 기존의 블록 정합 방법이나 정규화 메쉬 방법보다 PSNR이 향상된 결과를 얻을 수 있었고 노드의 재정렬시에도 다른 방법보다 초기 수렴속도가 우수함을 알 수 있었다.

Abstract

In Conventional BMA, the motion vector can describe only translational movement and blocking noise is generated. To overcome this defect, motion estimation using triangular mesh has been proposed. The regular mesh is the method of dividing the image area into equal size triangle and having the same node connection. It has no additional information about mesh structure, but do not reflect the real motion because it represents the regions by equal mesh structure regardless of the amount of motion. In this paper, motion estimation using dynamic regular mesh is proposed, In this method, the mesh structure is varied from the amount of motion and maintain the form of regular mesh. By the simulation, proposed method have better performance in PSNR and is superior to the other method in convergence rate.

I. 서 론

전송용량이 작은 통신회선을 통해서 영상신호를 전송하기 위해서는 영상신호가 가지는 높은 용장성(redundancy)을 감소시키기 위한 여러 가지 압축 방법을 사용하는 것이 필수적이다. 일반적으로 영상신호는 인접화소 간에 높은 공간적 상관성을 가지고 있고, 프레임 간에도 화소의 위치와 값은 시간적인 상관성을

가지고 있다. 이러한 시공간적인 용장성을 줄이기 위해 프레임 간 움직임 예측과 DCT 변환을 사용한 하이브리드 코딩 방식을 사용하고 있고, H.261, H.263, MPEG-1, MPEG-2 등의 표준안에서도 이러한 방법을 채택하고 있다.

이러한 표준안에서는 움직임 추정을 위해 일정한 크기의 블록이 현재 프레임과 이전 프레임사이에서의 이동 경로를 찾기 위한 블록 정합 방법(Block Matching Algorithm)을 사용하고 있다. 그러나 블록 정합 방법에서는 움직임 벡터로 평면적인 이동만을 기술할 수 있기 때문에 회전이나 확대, 축소 등 다양한 형태의 움직

* 正會員, 漢陽大學校 電子工學科

(Dept. of Electronic Engineering, Hanyang University)

接受日字:2001年7月24日, 수정완료일:2001年9月29日

임을 기술할 수가 없고 이로 인해 블록화 잡음(blocking noise)이 발생하였다. 그리하여 대상체의 다양한 움직임을 기술할 수 있는 여러 가지 움직임 추정 방법들이 연구되어 있고 그 중에서 6개의 파라미터의 의해 움직임을 표현하는 Affine 변환법^[2]이나 8개의 파라미터를 사용하는 쌍선형 변환(Bilinear Transform)^[3], 또는 원근사영 변환(Perspective Transform)^[4]등을 사용한 방법들이 연구되고 있다. 쌍선형 변환이나 원근사영 변환은 Affine 변환에 비해 추가 자유도를 가지는 장점이 있으나 성능면에서는 거의 유사하고 비유리 함수의 역변환을 구해야 하므로 연산량이 증가하게 된다.^[5]

이러한 움직임 추정 방법들은 먼저 적당한 메쉬의 구조를 정의하고 이 메쉬의 각 꼭지점들이 이전 프레임의 어느 위치로 이동하였는지를 추정한 후 공간 변환에 따라 해당화소의 값들 찾아내는 과정을 거치게 된다. 이때 메쉬의 구성 방법에 따라 일정한 크기의 삼각형 또는 사각형으로 분할하는 정규화 메쉬(Regular Mesh)와 객체의 움직임이나 에지정보를 바탕으로 특징점을 추출한 후 이 점들을 연결하여 메쉬를 구성하는 비정규화 메쉬(Irregular Mesh)로^[3] 구분할 수 있다.^[5] 정규화 메쉬는 각 메쉬의 크기와 위치, 연결상태가 일정하므로 메쉬의 구조를 전송하기 위한 추가적인 정보가 필요없으나 움직임이 많은 영역과 배경처럼 움직임이 적은 영역을 같은 크기의 메쉬로 분할함으로써 하나의 메쉬 영역에 여러 가지 운동성분이 혼재할 때는 각각의 움직임을 추정할 수가 없게 된다. 이러한 문제점을 해결하기 위해 비정규화 메쉬에서는 움직임이 많을수록 많은 특징점을 설정함으로써 실제적인 움직임을 추정할 수 있도록 하였으나 메쉬의 위치와 구조에 대한 정보를 전송해야 하기 때문에 추가 전송되는 정보의 양이 많아지게 된다. 이러한 추가 정보를 줄이기 위해 사각형 형태의 메쉬 구조를 움직임의 양에 따라 계층적으로 분할하고 각 분할 영역을 quadtree 형태의 코딩 방법을 사용하는 계층적 그리드 보간법(HGI)등이 개발되었다.^[6] 그러나 이 방법은 쌍선형 변환을 사용하기 때문에 비유리함수의 역함수를 구하기가 어렵게 된다. 본 논문에서는 Affine 변환을 사용하기 위해 삼각형 구조의 정규화 메쉬를 사용하며 메쉬의 위치와 구조에 대한 추가정보의 전송을 최소화하고 움직임의 양에 따라 크기와 구조를 가변할 수 있는 동적 정규화 메쉬방법을 제안한다.

II. Affine 모델

Affine 변환에서는 평면상에서의 움직임을 식 (1)과 같은 6개의 변수에 의해 표현한다. 이때 각 변수들의 값에 따라서 3차원 공간상에서의 움직임을 평면상에서 표현하는 형태를 결정하게 된다. 즉 a_2, a_3, a_5, a_6 이 모두 0일 때는 블록정합에서와 같이 평면적인 운동을 나타내게 되고 이 변수들의 형태에 따라 회전, 축소, 확대 등 여러 가지 이동형태를 기술할 수 있게 된다.

$$\begin{aligned} u(x, y) &= (x - x') = a_1 + a_2x + a_3y \\ v(x, y) &= (y - y') = a_4 + a_5x + a_6y \end{aligned} \quad (1)$$

$$d(x, y) = \begin{bmatrix} u(x, y) \\ v(x, y) \end{bmatrix} = \begin{bmatrix} a_1 \\ a_2 \\ \cdot \\ \cdot \\ a_5 \\ a_6 \end{bmatrix} = \begin{bmatrix} 1 & x & y & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & x & y \end{bmatrix} \quad (2)$$

여기서 (x, y) 와 (x', y') 은 시간 t 와 t' 에서 각 점의 위치를 나타내고 $u(x, y), v(x, y)$ 는 각각 점 (x, y) 에서 변위 벡터의 x 성분과 y 성분을 나타낸다. 이 식에서 6개의 변수를 결정하기 위해 그림 1과 같이 삼각형을 형성하는 3개의 꼭지점의 좌표 (x_i, y_i) 가 어느 위치로 이동했는지를 결정하면 이동한 점의 좌표 (x'_i, y'_i) 로부터 각 변수들을 결정할 수 있게 된다.

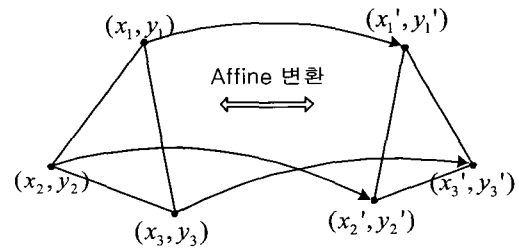


그림 1. 시간 t 와 t' 에서의 노드의 위치
Fig. 1. Node position of time t and t' .

이동 전과 이동 후의 각 점의 좌표 $(x_i, y_i), (x'_i, y'_i)$ 을 식 (1)에 대입하여 다음과 같은 행렬을 구할 수 있다.^[2]

$$\begin{pmatrix} u(x_1, y_1) \\ u(x_2, y_2) \\ u(x_3, y_3) \\ v(x_1, y_1) \\ v(x_2, y_2) \\ v(x_3, y_3) \end{pmatrix} = \begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ v_1 \\ v_2 \\ v_3 \end{pmatrix} = \begin{pmatrix} 1 & x_1 & y_1 & 0 & 0 & 0 \\ 1 & x_2 & y_2 & 0 & 0 & 0 \\ 1 & x_3 & y_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & x_1 & y_1 \\ 0 & 0 & 0 & 1 & x_2 & y_2 \\ 0 & 0 & 0 & 1 & x_3 & y_3 \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \end{pmatrix} = \begin{bmatrix} \phi & 0 \\ 0 & \phi \end{bmatrix} a \quad (3)$$

여기서 $\phi = \begin{bmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{bmatrix}$ 이고 벡터 a 를 ϕ^{-1} 로 표현하면

$$a = \begin{bmatrix} \phi^{-1} & 0 \\ 0 & \phi^{-1} \end{bmatrix} \begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ v_1 \\ v_2 \\ v_3 \end{pmatrix} \quad (4)$$

여기서

$$\phi^{-1} = \frac{1}{\det(\phi)} \times \begin{pmatrix} x_2y_3 - x_3y_2 & x_3y_1 - x_1y_3 & x_1y_2 - x_2y_1 \\ y_2 - y_3 & y_3 - y_1 & y_1 - y_2 \\ x_3 - x_2 & x_1 - x_3 & x_2 - x_1 \end{pmatrix} \quad (5)$$

식 (4)로부터 Affine 변수를 구할 수 있고 식 (1)에 임의의 좌표를 대입하면 변환된 결과의 위치 좌표를 구할 수가 있다. 이때 $\det(\phi)$ 가 0일 때는 이동한 세 점의 좌표가 직선을 이루는 경우에 해당한다. 일반적으로 Affine 변수 a 는 실수 값으로 나타나고 그 결과 변위 벡터나 변환 좌표가 실수형태로 나타나게 된다. 그러므로 변환좌표의 화소값을 구하기 위해서는 쌍선형 보간(Bilinear Interpolation) 방법으로 주변화소와의 상대적인 거리에 의해 화소값을 계산해 주어야 한다.

III. 동적 정규화 메쉬의 생성

Affine 변환을 움직임 추정에 사용하기 위해서는 먼저 입력된 두 영상에 노드(node)의 위치를 정하고 이 노드들을 연결하여 삼각형의 구조를 결정하는 삼각화(triangulation)를 해야한다.

노드의 위치를 결정하는 방법에 따라 일정한 간격으로 배치하는 정규화 메쉬(regular mesh)와 객체의 특성에 따라 노드의 배치를 다르게하는 비정규화 메쉬(irregular mesh)로 나눌수 있다.^[5] 정규화 메쉬에서는

일정한 간격으로 노드를 배치하고 이 노드들을 연결하여 삼각화를 하게 되는데 그림 2는 노드들의 연결상태에 따른 여러 가지의 정규화 메쉬의 형태를 나타내고 있다.

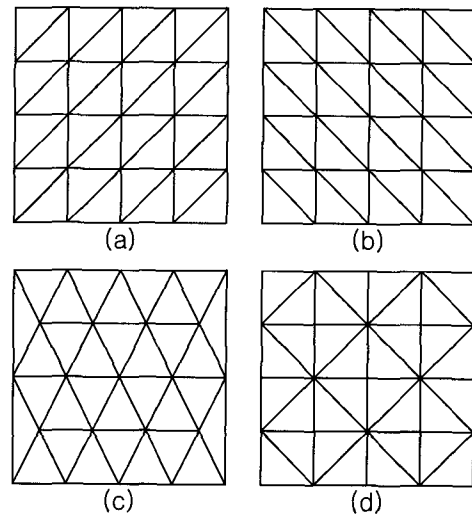


그림 2. 정규화 메쉬 (a)좌향메쉬 (b)우향메쉬 (c)Hexagonal 메쉬 (d)4-8 연결 메쉬
Fig. 2. Regular mesh (a)left-side mesh (b)right-side mesh (c)hexagonal mesh (d) 4-8 connected mesh.

그림 2에서 (a)(b)(c) 형태의 메쉬는 한 개의 노드의 주변에 있는 6개의 노드와 연결되어 있고 (d)의 경우는 4개 또는 8개의 노드와 연결되어 있다. 본 논문에서는 (d)와 같은 형태의 메쉬를 기본으로 하는 동적 정규화 메쉬의 구조를 제안한다.

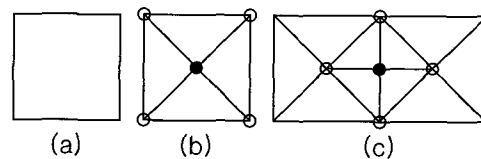


그림 3. 내부 노드와 외부노드의 삽입 (a)기본메쉬 (b)내부노드 (c)외부노드
Fig. 3. The Insertion of internal and external node (a)primary mesh (b)internal node (b)external node

동적 정규화 메쉬는 그림 3과 같이 일정한 간격으로 배치된 4개의 노드가 연결된 사각형 메쉬를 기본 형태로 사용하고 이러한 기본메쉬(Primary Mesh)의 중심과

각 변에 새로운 노드를 추가하여 이 노드들을 연결함으로써 삼각화를 구성해 나간다. 움직임 정보의 양에 따라 노드를 추가함으로써 움직임이 많은 영역은 더욱 세밀한 구조의 삼각화를 형성하고 적은 영역은 거친 삼각화를 형성하게 한다. 메쉬의 구조를 기술하기 위한 정보를 최소화하기 위해 노드의 존재 여부를만 전송함으로써 송신측과 같은 결과를 얻을수 있도록 하기 위해서 다음과 같은 삼각화의 제한 조건을 결정한다.

- 기본메쉬를 구성하는 노드들은 서로 연결되어 사각형의 형태를 구성한다.
- 연결된 삼각형의 각변에는 노드가 존재하지 않는다.
- 연결된 노드의 연장선 상에는 같은 레벨의 노드가 존재한다.

위와 같은 조건을 만족하도록 하기위해 기본메쉬에 내부 노드와 외부노드를 삽입하여 계층적인 구조의 삼각화를 구성한다. 계층적인 삼각화를 위해 기본메쉬의 크기를 이전 단계의 절반으로 축소시키면서 노드의 삽입 과정을 반복하게 된다. 이때, 설정된 가장 큰 기본메쉬와 삽입된 노드들을 레벨 0의 노드들로 정의하고 기본메쉬의 크기를 줄여나가면서 레벨을 증가시키게 된다. 그림 4는 동적 정규화 메쉬의 생성 과정을 나타낸다.

1. 내부 노드의 삽입

레벨 N의 4개의 노드로 구성되는 기본메쉬는 서로 연결 되어 사각형의 형태를 이루어야 한다. 노드를 추가 하면서 세밀한 삼각화를 구성해 나갈 때 사각형의 형태로 연결되지 않은 노드들은 메쉬의 확장에서 제외된다. 즉 주변에 있는 4개의 노드들이 기본 메쉬의 형태를 이루고 있을 때 내부노드를 추가하여 삼각화를 구성한다.

2. 외부노드의 삽입

하나의 기본 메쉬는 주변에 있는 메쉬들과 한 변을 공유하고 있으므로 외부 노드의 삽입은 주변에 변을 공유하는 메쉬들의 구조에 영향을 미치게 된다. 그러므로 외부노드의 삽입여부는 그 노드 주변에 연결된 메쉬들이 내부노드를 가지고 있는지를 검사하여 이 조건을 만족하고 해당 영역의 움직임이 큰 경우에만 외부노드를 추가해 준다.

움직임의 양을 나타내는 기준으로 프레임 간 차이값

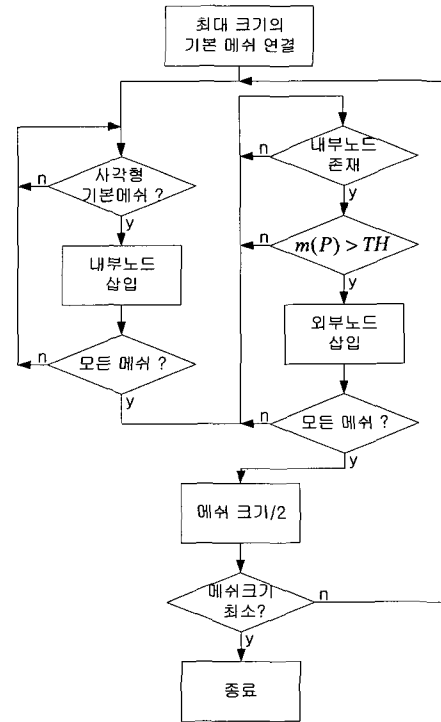


그림 4. 메쉬 생성 흐름도

Fig. 4. The flow chart of the mesh generation.

의 분산을 사용한다.

$$m(P) = \frac{1}{N_P} \sum_{(x,y) \in P} (f_{FD}(x,y) - \overline{f_{FD}})^2 \quad (6)$$

여기서 영역 P는 삽입될 외부노드와 연결될 4개의 노드로 구성되는 영역을 나타내고 N_P 는 이 영역에 포함되는 화소의 개수, $f_{FD}(x,y)$ 는 (x,y) 에서의 두 프레임 간의 화소 차이값, $\overline{f_{FD}}$ 는 해당 영역의 평균 프레임 간 차이값을 나타낸다. $m(P) > TH$ 일 때 노드를 삽입하고 그 외의 경우는 움직임이 작은 영역으로 판단하여 외부노드를 삽입하지 않는다. 이때 TH는 레벨이 올라갈수록 움직임이 큰 영역만 분할되도록 하기 위해 초기 설정값 $InitTH$ 으로부터 다음과 같이 설정한다.

$$TH = InitTH * 2^{level} \quad (level = 0..N) \quad (7)$$

3. 레벨의 확장

계층적인 삼각화를 구성하기 위해 검사할 기본 메쉬의 크기를 절반으로 줄여 다시 내부노드와 외부노드의 추가 여부를 결정해 나간다. 기본 노드의 간격을 절반

으로 줄이고 1,2의 과정을 반복하여 계층구조 삼각화를 구성할 수가 있다.

다음 그림 5는 동적 정규화 메쉬 방법에 의한 메쉬 생성결과를 보인다.

IV. 움직임 벡터의 추정

생성된 메쉬로부터 움직임을 추정하기 위해서는 두 프레임 사이에서 해당 노드의 이동 벡터를 추정하여야 한다. 동적 정규화 메쉬 방법에 의해 구성된 메쉬 구조는 한 개의 노드에 4개에서 8개까지의 주변 노드들이 연결되어 있다. 그러므로 한 개의 노드의 위치를 이동시키면 이 노드를 공유하는 주변 메쉬들에도 영향을 미치게 된다. 움직임 벡터의 추정은 HMA(hexagonal matching algorithm)과^[1] 유사한 방법을 사용한다.

1. BMA를 사용한 초기검색

노드의 대략적인 이동 위치를 검색하기 위해, 해당 노드를 중심으로 하는 BMA(Block Matching Algorithm)를 사용한다. 이때 메쉬의 구조가 변형되는 것을 막기 위하여 탐색 영역은 그림 6와 같이 그 노드와 연결된 주변 노드들로 구성되는 다각형 영역으로 제한한다. 계층적 구조로 형성된 메쉬는 레벨이 올라갈수록 노드간의 간격이 작아지므로 레벨 0에 해당하는 노드에 대해서만 BMA를 적용하고 탐색제한영역도 레벨 0의 노드로 구성되는 영역으로 제한한다. 나머지 노드들은 레벨 0의 노드 위치를 이용해 쌍선형 보간방법에 의해 초기 위치를 결정한다.

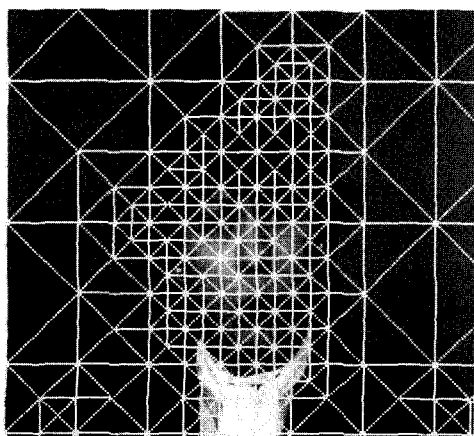


그림 5. 동적 정규화메쉬(16번 프레임- 얼굴부분 확대)
Fig. 5. Dynamic regular mesh (frame No. 16).

2. 노드 재정렬(Node Refinement)

BMA로 결정된 노드의 위치로부터 Affine 변환에 의해 노드의 위치를 재정렬(refinement)한다. 주변에 연결된 노드의 위치는 고정시키고 해당 노드의 위치만을 이동시키면서 Affine 변환에 의해 영상 데이터를 복원한다. 이때 에러가 가장 작은 위치를 노드의 움직임 벡터로 결정하고 에러의 평가함수는 주변 노드로 연결된 탐색제한 영역에 대한 원 영상과 복원 영상 사이의 PSNR을 사용한다.

$$PSNR = 10 \log_{10} \frac{255^2}{MSE} \text{ dB} \quad (8)$$

$$MSE = \frac{1}{N_p} \sum_{(x,y) \in P} (I(x,y) - \widetilde{I(x,y)})^2$$

여기서 P는 주변 노드로 구성되는 탐색 제한 영역을 나타내고 N_p 는 탐색 제한 영역에 포함되는 화소의 수, $I(x,y), \widetilde{I(x,y)}$ 는 (x,y) 에서 원 영상의 화소 값과 복원된 화소값을 나타낸다.

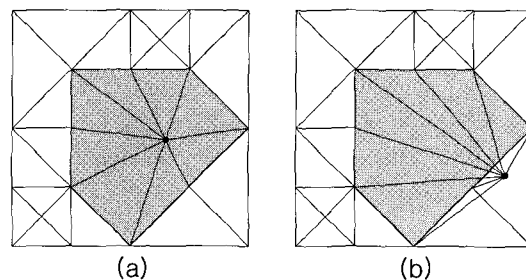


그림 6. 탐색영역 (a)탐색 제한 영역 (b) 제한영역을 벗어난 경우
Fig. 6. Search region (a) restricted search region (b) out of restricted search region

노드의 재정렬은 레벨 0에서 결정된 노드들을 먼저 처리하고 차례로 각 레벨에 해당하는 노드들의 움직임 벡터를 결정한다. 모든 노드의 재정렬이 완료되면 위와 같은 재정렬 과정을 반복하여 모든 노드들이 최적의 위치로 수렴하도록 한다.

수렴여부의 판정은 연결된 주변 노드들이 이전의 재정렬 과정에서 위치가 변하지 않았고 해당 노드도 위치가 변하지 않았을 때 수렴한 것으로 판정한다. 그림 7은 수렴한 이후에 복원 영상과 각 노드를 나타내고 있다.

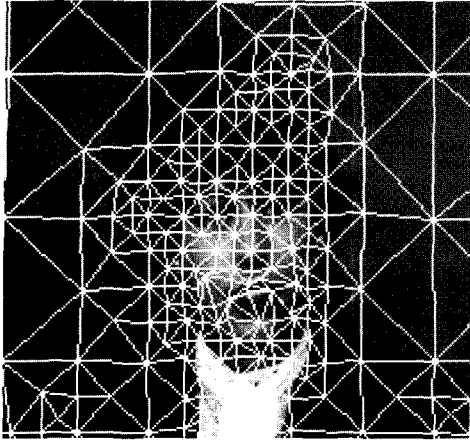


그림 7. 재정렬 후의 노드위치(13번 프레임-얼굴부분 확대)

Fig. 7. Node position after refinement(frame No. 13).

V. 메쉬 구조의 코딩

앞에서 정의된 메쉬의 구조를 전송하기 위해서 노드의 연결상태만을 검사하여 연결선이 있는 노드에는 1을 할당하고 연결선이 없는 노드에는 0을 할당함으로써 메쉬의 구조를 전송할 수가 있다. 즉 가장 높은 레벨의 메쉬(기본 메쉬의 크기가 가장 작을 때)에서 나타날 수 있는 노드의 간격을 기준으로 다른 노드와 연결되어 있는 연결 노드에는 1을 할당하고 연결된 노드가 없는 고립 노드일 때는 0을 할당함으로써 그림 8과 같이 메쉬의 구조를 나타내는 비트열을 생성시킬 수 있다.

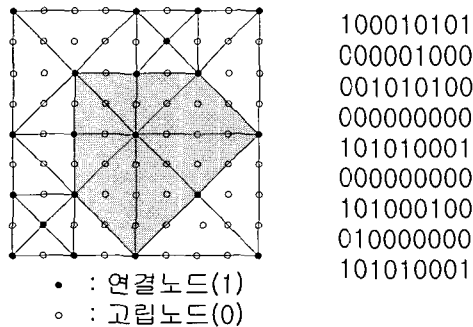


그림 8. 연결노드와 고립노드의 코딩

Fig. 8. The coding of the connected node and the isolated node.

생성된 비트열을 전송하고 각각의 연결노드에 해당하는 움직임 벡터를 전송함으로써 메쉬의 구조와 움직

임 벡터를 수신측으로 전달할 수가 있다. 메쉬의 재 구성은 메쉬 생성의 과정과 유사하다. 가장 낮은 레벨(가장 큰 크기의 기본 메쉬)의 메쉬 위치를 기준으로 정하고 내부노드의 위치에 연결노드가 있는지를 검사하여 내부노드의 연결 구조를 구성한다. 또 네 번에 있는 외부 노드도 위와 마찬가지로 연결노드의 존재 여부를 검사하여 외부노드 위치를 추가한다. 이와 같은 과정을 마지막 레벨까지 반복하면 메쉬의 구조를 재 구성할 수가 있다. 위와같이 생성된 비트열들은 연속적인 0이나 1의 상태들이 많이 나오게 된다. 그러므로 전송 효율을 높이기 위해 run-length coding 방법을 사용하면 추가 전송되는 정보의 양을 더욱 줄일 수 있게 된다.

VI. 실험결과

동적 정규화 메쉬의 성능평가를 위해 352x288 크기의 "Clair" 영상과 "Miss America" 영상에서 휘도 신호만을 사용하였고, 1번 화면에서 100번째 화면까지에 대해서 3 프레임 간격으로 여러 가지 방법들을 적용해 보았다.

실험에서는 움직임 추정 방법을 비교하기 위해 BMA와 정규화 메쉬의 구조에 따라 우향 메쉬(RM), 좌향 메쉬(LM), 4-8연결 메쉬(4-8M), 헥사 메쉬(HM), HGI의 방법을 제안한 방법(DRM)과 비교하였다. 추정된 움직임 벡터로부터 각각의 블록의 위치와 노드의 위치를 구하고 이로부터 움직임 보상에 의해 영상을 합성한다. 화질의 평가를 위하여 입력 영상과 합성 영상 사이의 PSNR을 사용하였다.

$$PSNR = 10 \log_{10} \frac{255^2}{MSE} \text{ dB} \quad (8)$$

여기서 MSE(Mean Square Error)는 원 영상과 합성 영상 사이의 평균제곱 오차이다. BMA에서의 블록의 크기는 16*16, 탐색 영역은 가로, 세로 「-8, +7」의 영역에서 전역탐색(full search) 방법을 사용하였다. 사용한 정규화 메쉬에서는 노드의 배치거리는 16*16을 사용하고 초기에 BMA를 사용한 탐색시에는 노드를 중심으로 하는 16*16 화소를 사용하였고 탐색영역은 BMA와 같이 「-8, +8」을 사용하였다. 노드의 재정렬시 탐색영역은 「-3,+3」로 설정하였다. HGI에서는 발생하는 노드의 수가 다른 방법과 비슷하도록 분할의 임계치는 35로 하였고 탐색영역도 다른 방법과 마찬가지로 「-3,+3」으

로하고 재정렬 방법도 동일하게 적용하였다.

표 1. 평균 PSNR (dB)
Table 1. Average PSNR(dB).

입력영상	알고리즘	완전수렴	3회반복
clair	BMA	36.0098	
	LM	37.9697	37.8855
	RM	37.9667	37.8380
	4-8M	37.9411	37.7845
	HM	38.0738	37.9504
	HGI	38.6513	38.4131
	DRM	39.0942	38.9071
MissAmerica	BMA	36.8027	
	LM	38.6333	38.6138
	RM	38.5805	38.5569
	4-8M	38.6077	38.5905
	HM	38.6194	38.6015
	HGI	38.7697	38.7101
	DRM	39.2468	39.2026

동적 정규화 메쉬(DRM)는 레벨 0에서 노드의 간격은 64*64, 레벨1에서는 32*32, 레벨3에서는 16*16, 으로 설정하여 3단계의 계층구조를 사용하였고, 외부 노드의 삽입을 결정하기 위한 프레임간 차의 분산값은 초기 문턱값 *InitTH*는 10으로 설정하였으며 나머지 초기검색이나 재정렬을 위한 조건들은 다른 정규화 메쉬 방법과 같도록 하였다. 재정렬시 노드의 재정렬 순서는 가장 낮은 레벨에 해당하는 노드로부터 상위 레벨로 정렬되도록 하였다.

정규화 메쉬를 사용한 방법에서는 노드의 재정렬이나 최종의 영상 합성시 Affine 변환된 좌표로부터 이중 선형 보간 방법을 사용하여 화소값을 계산하였다. 또 노드의 재정렬시 보통 3회 정도 반복하면 충분한 성능을 얻을 수 있으므로 각 알고리즘을 3회 반복 했을 때와 완전 수렴시켰을 때의 성능을 비교하였다.

그림 9는 완전 수렴시켰을 때 움직임 보상된 영상의 PSNR을 나타내고 표 1은 각 조건에 따른 평균 PSNR을 나타낸다.

표 2는 각 방법에 따른 노드의 수와 메쉬의 구조를 전송하기 위한 비트수를 비교한 것이다. 각 노드의 움직임 벡터는 x, y 각각 4비트를 할당하고 HGI와 DRM에서의 전송비트 수는 메쉬의 구조를 전송하기 위한 비트수와 각 노드의 움직임 벡터를 전송하기 위한 비트를 합한 값이 된다. HGI의 경우는 quadtree를 사용하

여 메쉬구조를 코딩한 결과이고 DRM은 5장에서 처럼 최소 크기의 노드에 0과 1을 할당하여 처리한 결과를 제시하였다.

표 2. 전송 비트수
Table 2. The number of Transmitted Bits.

입력영상	알고리즘	전송비트수(bits)
clair	BMA	396*8
	LM, RM, 4-8M	437*8
	HM	428*8
	HGI	441*8+254
	DRM	427*8+1584
Miss America	BMA	396*8
	LM, RM, 4-8M	437*8
	HM	428*8
	HGI	458*8+312
	DRM	473*8+1584

일반적으로 정규화메쉬를 사용한 방법들이 BMA보다는 뛰어난 성능을 나타내고 있고 RM, LM, 4-8M는 거의 같은 성능을 나타내며 HM는 구성된 삼각형들이 모두 예각을 이루고 있으므로 다른 정규화 메쉬에 비해 다소 우수한 성능을 나타내고 있다. 본 논문에서 제안한 DRM은 다른 정규화 메쉬뿐만 아니라 비슷한 수의 노드를 사용하는 HGI에 비해서도 높은 PSNR을 얻을 수 있었다. 이 결과는 HGI에서는 재정렬에 참여하지 않는 노드들이 발생하지만 DRM은 모든 노드에 대해서 최적의 위치를 찾도록할 수 있기 때문이다. 전송 비트수에서는 표 2와 같이 다른 방법에 비해 DRM의 구조를 전송하기 위해 어느 정도의 추가 비트가 소요되지만 대부분의 비트들이 1이나 0이 반복되는 형태로 나타나기 때문에 run-length coding을 사용하면 추가 비트를 더욱 줄일 수 있게 된다. 본 논문에서 제안한 방법은 완전 수렴했을 때 다른 정규화 메쉬보다 약 1dB정도 우수한 성능을 나타내고 있고 특히 움직임이 큰 영상에서는 상대적으로 뛰어난 성능을 발휘함을 알 수 있다. 표 1에서와 같이 반복 횟수가 3일 때도 다른 방법보다 초기 수렴 성능에서도 우수함을 알 수 있다.

Ⅶ. 결 론

본 논문에서는 입력영상의 움직임 정보에 따라 메쉬의 모양과 구조가 가변되는 동적 정규화 메쉬를 사용

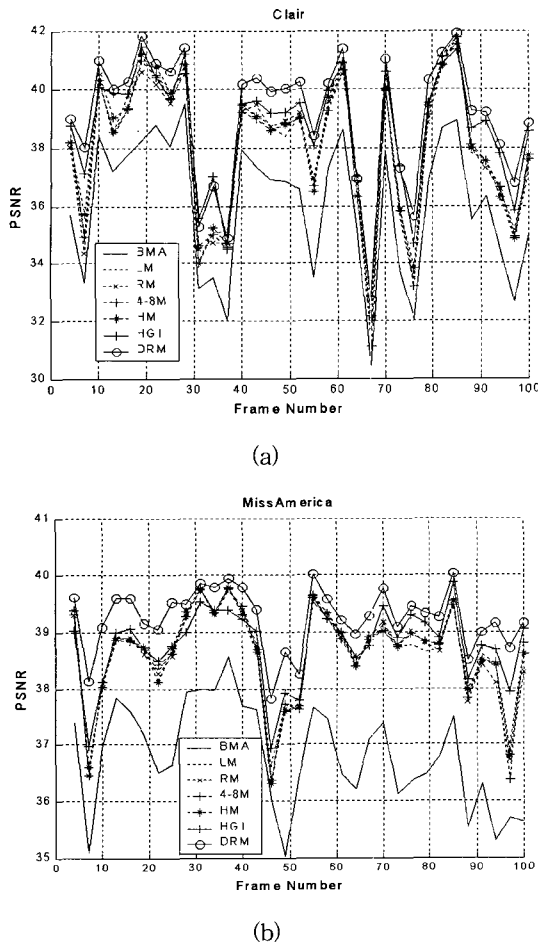


그림 9. 움직임 보상된 영상의 PSNR (a) Clair (b) Miss America
 Fig. 9. PSNR of motion compensated image (a) Clair (b) Miss America.

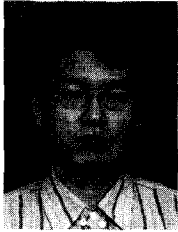
한 움직임 추정방법을 제안하였다. 초기 설정된 노드의 위치로부터 메쉬구조에 적합한 내부 노드를 삽입하고 연속된 두 영상의 차의 분산이 임계값 이상일 때 외부 노드를 삽입함으로써 삼각화를 형성한다. 계층적인 구조에 따라 기본 노드의 간격을 줄여나가며 내부 노드와 외부 노드를 삽입함으로써 메쉬의 구조에 대한 추가 정보를 최소화 한 동적 메쉬를 구성하였다. 구성된 메쉬 구조에 Affine 변환을 적용함으로써 노드의 위치 변화를 이용한 움직임 추정과 움직임 보상 방법을 사

용하였다. 실험결과 동적 메쉬 구조에 의해 휘도값이 변하지 않는 배경 영역보다는 움직임이 많은 영역에 삼각형 패치의 크기를 작게 함으로써 BMA나 다른 정규화 메쉬에 비해서 뛰어난 성능을 나타냈고 저 비트율 영상 전송에 적합한 방법임을 확인하였다.

참 고 문 헌

- [1] Y. Nakaya and H. Harashima, "Motion Compensation Based on Spatial Transformations," *IEEE Trans. on Circuits and Systems for Video Technology*, vol.4, no.3, pp.339-356, June 1994.
- [2] M. Dundon, O. Avaro, C. Roux, "Triangular active mesh for motion estimation," *Signal Processing: Image Communication*, vol.10, ELSEVIER, pp.21-41, 1997.
- [3] Gary J. Sullivan and Richard L. Baker, "Motion compensation for video compression using control grid interpolation," *Proc. of ICASSP 91*, vol.4, pp.2713-2716, Toronto, My. 1991.
- [4] V. Seferdis and M. Ghanbari, "General approach to block-matching motion estimation," *Optical Engineering*, vol.32, no.7, pp.1464-1474, July 1993.
- [5] Y. Wang and O. Lee, "Use of Two-Dimensional Deformable Mesh Structures for Video Coding, Part I-The Synthesis Problem: Mesh-Based Function Approximation and Mapping," *IEEE Trans. on Circuits and Systems for Video Technology*, vol.6, no.6, pp.636-646, Dec. 1996.
- [6] C-L Hung and C-Y Hsu, "A New Motion Compensation Method for Image Sequence Coding Using Hierarchical Grid Interpolation," *IEEE Trans. on Circuits and Systems for Video Technology*, vol.4, no.1, Feb. 1994.

저 자 소 개



李 東 奎(正會員)

1968년 11월 21생. 1991년 2월 한양대학교 전자공학과 학사. 1993년 2월 한양대학교 전자공학과 석사. 1993년 3월~현재 한양대학교 전자공학과 박사과정. 주 관심분야 : 영상처리, 화상통신, DSP 응용



李 斗 秀(正會員)

1946년 7월3일생. 1968 2월 전북대학교 전자공학과 학사. 1970년 2월 전북대학교 전자공학과 석사. 1973년 2월 전북대학교 전자공학과 박사. 1973년 2월~1979년 3월 전북대학교 전자공학과 조교수. 1979년 3월~현재 한양대학교 전자전기컴퓨터 공학부 교수. 주 관심분야 : DSP, 의공학, 화상처리