

# 계층적 삼각화를 이용한 움직임 추정과 보상

정회원 이 동 규\*, 이 두 수\*\*

## Motion Estimation and Compensation using hierarchical triangulation

Dong-Gyu Lee\*, Doo-Soo Lee\*\* Regular Members

### 요 약

본 논문에서는 움직임의 양에 따라 삼각형 메쉬의 크기와 위치를 가변할 수 있는 계층적 삼각화 방법을 이용한 움직임 추정 방법을 제안한다. 이미지의 차의 분산을 이용하여 움직임의 양을 측정하고 삼각형의 세분화를 행함으로써 움직임이 많은 영역에 노드가 집중되도록 하고 계층적 삼각형 메쉬의 생성이 가능한 세분화 방법을 제시한다. 또한 계층화된 메쉬의 구조를 전송하기 위한 추가정보의 양을 줄일 수 있는 부호화 방법에 대해서 논한다. 실험결과 제안한 방법이 기존의 BMA이나 다른 메쉬 구조를 사용한 방법보다 PSNR이 평균 0.5dB 우수한 성능을 나타냄을 보인다.

### ABSTRACT

In this paper, we propose a motion estimation method using hierarchical triangulation that changes the triangular mesh structure according to its motion activity. The subdivision of triangular mesh is performed from the amount of motion that is calculated from the variance of image difference. As a result, node distribution is concentrated on the region of high activity. The subdivision method that makes it possible to yield hierarchical triangular mesh is proposed as well as the additional information reduction coding method for hierarchical mesh structure is described. By the simulation, proposed method have better performance than the conventional BMA(Block Match Algorithm) and the other mesh based method.

### I. 서 론

H.261, H.263, MPEG1, MPEG2와 같이 최근에 사용되는 대부분의 영상압축 방법들은 영상의 시간적 용장성(redundancy)을 줄이기 위한 움직임 예측과 보상방법을 사용하고 있다. 이러한 표준안에서는 연속된 프레임간의 시간적 상관성을 이용한 블록정합방법(BMA)을 사용하고 있으며 현재의 프레임을 일정한 크기의 블록으로 나누고 이전 프레임에서 가장 유사한 블록의 위치를 탐색하여 움직임 벡터를 구하는 방법이다. BMA은 구조가 간단하고 블록의 크기와 구조에 대한 추가정보가 필요 없어

구현하기가 용이하다는 장점이 있으나 블록내의 모든 화소가 동일한 병진운동만을 한다는 가정 하에 움직임 벡터를 구하기 때문에 한 블록 내에 서로 다른 운동을 하는 두개 이상의 객체가 존재하거나 회전, 확대, 축소 등의 복잡한 운동이 존재할 경우 이를 기술할 수가 없고 이로 인해 블록의 경계부분에 서로 다른 운동의 결과로 나타나는 블록화 효과(blocking effect)가 발생하게 된다.

이러한 문제점을 해결하기 위해서 BMA 보다 움직임 벡터의 표현에 자유도를 가지는 운동 모델들이 제안되었고 객체의 3차원 운동을 2차원 평면상에서 표현하는 방법에 따라 Affine 변환 (Affin

\* 한양대학교 대학원 전자통신전파공학과 신호처리 및 디지털 시스템 연구실(dglee@ihanyang.ac.kr),

\*\* 한양대학교 전자전기컴퓨터공학부(dsplee@hanyang.ac.kr)

논문번호 : 020491-1111, 접수일자 : 2002년 11월 6일

e Transform)<sup>[1][2][5]</sup>, 쌍 선형 변환 (Bilinear Transform)<sup>[3][6]</sup>, 원근 사영 변환(Perspective Transform)<sup>[4]</sup> 등을 사용한 방법들이 연구되고 있다. 이러한 방법들은 입력영상에 적당한 형태의 메쉬를 배치하고 연속적인 프레임에서 메쉬의 연결점들에 이동위치로부터 변환 파라미터를 구하고 해당 메쉬 안의 화소들의 이동위치를 구하여 움직임을 추정, 보상하는 방법이다. 이때 입력영상에 메쉬를 배치하는 형태에 따라서 사람의 얼굴과 같이 객체의 형태에 따라서 이미 정해진 메쉬 모델을 사용하는 모델 기반 부호화(Model Based Coding)나 입력된 영상을 일정한 크기와 구조를 가지는 메쉬로 구성하는 정규화 메쉬(Regular mesh), 또는 객체의 움직임이나 에지 정보 등을 이용해 특징점을 구한 후 이 점들을 연결해 메쉬를 구성하는 비정규화 메쉬(Irregular mesh) 등으로 나눌 수 있다. 모델기반부호화는 화상전화와 같이 입력영상이 제한되어 있는 경우에만 사용할 수 있으며 정규화 메쉬의 경우 모든 메쉬의 위치와 연결 상태를 일정하게 함으로써 추가적인 정보의 전송을 줄일 수 있으나 메쉬의 구조에 입력영상의 특징을 반영하지 못하고 하나의 메쉬 영역에서 두 개이상의 움직임이 존재할 경우 움직임의 추정과 보상을 제대로 할 수 없게 된다. 이러한 문제점을 해결하기 위해서 움직임의 양에 따라 특징점을 추출하고 이 특징점들을 Delaunay 삼각화 방법들을 사용하여 메쉬 구조를 생성하는 비정규화 메쉬를 사용한다<sup>[2]</sup>. 그러나 삼각화의 결과로 발생하는 노드들의 위치를 전송해야하기 때문에 추가정보의 양이 많아지게 된다. 이러한 추가정보의 양을 줄이고 움직임의 양에 따라 메쉬의 형태를 분할해 나감으로써 계층적 메쉬 구조를 형성하고 여기에 쌍선형 변환을 사용하는 HGI(Hierarchical Grid Interpolation) 과 같은 방법들이 연구되고 있다<sup>[6]</sup>.

본 논문에서는 계층적 삼각화를 수행함으로써 메쉬의 생성 시 입력영상의 특징을 반영하고 메쉬의 구조와 위치에 대한 추가정보의 양을 최소화 할 수 있는 삼각화 방법과 Affine 변환을 이용한 움직임 추정과 보상방법을 제안한다.

## II. 계층적 삼각화

움직임의 추정과 보상에 Affine 변환을 이용하기 위해서 먼저 입력영상에 적당한 형태의 삼각형 메쉬를 형성해야 한다. 이렇게 형성된 메쉬의 연결점(노드)의 움직임을 추정하여 노드의 이동위치를 구

하고 삼각형을 구성하는 3개의 노드의 움직임 벡터로부터 Affine 변환의 6개의 파라미터를 구할 수 있게 된다. 하나의 메쉬 안에 있는 화소에 대해서 Affine 변환을 통해 이전 프레임과 현재 프레임 사이에 화소의 위치변화를 구함으로써 움직임 보상을 하게 된다.

계층적 삼각형 메쉬의 구성을 위해서 가장 하위 레벨(레벨 0)에서는 그림3의 (a)와 같은 정규화 메쉬를 사용한다. 이 경우 모든 메쉬의 모양이 직각이등변삼각형을 이루고 각 노드마다 4개 또는 8개의 다른 노드들이 연결되어 있다(4-8M). 이와 같은 기본 메쉬에서 해당 영역의 움직임 양을 측정하여 움직임이 많은 영역은 각 변에 새로운 노드를 추가하여 세밀한 구조의 삼각화를 형성하고 움직임이 적은 영역은 거친 삼각화를 형성하게 한다.

움직임의 양을 판단하는 기준으로 프레임 간 화소 차의 분산을 사용한다.

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

여기서 영역 P는 메쉬를 구성하는 삼각형 영역을 나타내고  $N_P$ 는 이 영역에 포함되는 화소의 갯수,  $f_{FD}(x,y)$ 는  $(x,y)$ 에서의 두 프레임 간의 화소 차이값,  $\overline{f_{FD}}$ 는 해당 영역의 프레임 간 화소 차이값의 평균을 나타낸다.  $m(P) > TH$  일 때 삼각형의 각 변에 3개의 노드를 삽입하고 그 외의 경우는 움직임이 작은 영역으로 판단하여 노드를 삽입하지 않는다.

인접한 삼각형 메쉬 들은 각 변을 다른 메쉬 들과 공유하고 있으므로 이와 같은 노드의 추가는 주변에 연결되어 있는 메쉬의 구조에도 영향을 미치게 된다. 노드의 추가로 인해 삼각형 메쉬의 각 변에 노드가 존재할 수 있는 경우는 그림과 같이 세 변에 노드가 없는 경우, 1개인 경우, 2개인 경우, 3개인 경우로 나눌 수 있다. 변에 노드가 존재 하지 않는 경우는 해당 삼각형 메쉬가 세분화가 필요 없고 또한 주변 메쉬의 영향을 받지 않은 경우이며, 3개의 경우는 해당 메쉬의 움직임이 많아 노드가 추가된 경우이거나 주변에 변을 공유하는 3개의 모든 메쉬의 세분화에 영향을 받은 경우이다. 변의 노드가 1개 또는 2개인 경우는 주변 메쉬의 영향에 의해 노드가 추가된 경우이다. 3변에 추가된 노드의 개수와 그 노드가 직각이등변삼각형의 등변에 존재하느냐 또는 직각의 대변에 존재하느냐에 따라 가

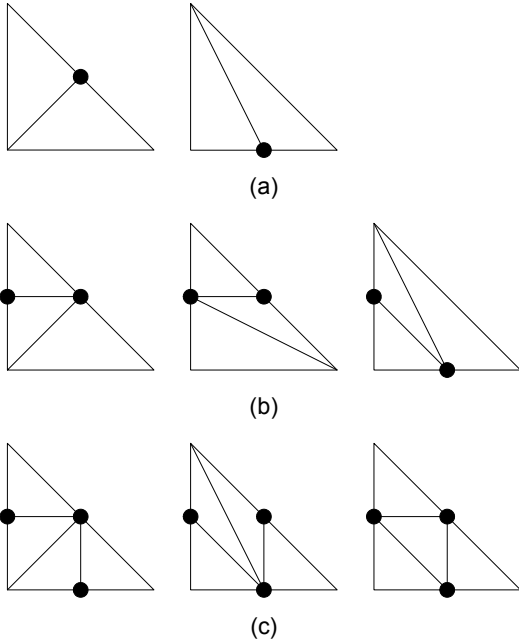


그림 1. 노드의 개수와 삼각화 (a) 노드가 1개인 경우 (b) 노드가 2개인 경우 (c) 노드가 3개인 경우  
 Fig. 1. the number of nodes and triangulation (a) case 1 (b) case 2 (c) case 3

능한 삼각화는 그림1과 같다.

계층적 삼각화를 위해서는 추가된 노드에 의해서 연결된 메쉬 들이 삼각형을 이루도록 해야 하며 하위 레벨에서의 연결 상태가 계층화가 이루어짐에 따라 형성된 새로운 삼각형 구조에서도 유지되어야 상위 레벨의 삼각화가 이루어 질 수 있다. 또한 노드의 배치로부터 형성되는 메쉬 구조가 송수신 측에서 유일한 형태로 나타나도록 하면 메쉬의 구조에 대한 정보를 전송할 필요가 없게 되고 추가된 노드의 위치에 대한 정보만을 전송함으로써 압축 효율을 높일 수가 있게 된다.

초기 상태의 삼각형들은 직각 이등변삼각형을 이루고 있으므로 위와 같은 조건을 만족하도록 하기 위해 세분화가 이루어진 삼각형의 경우에도 직각 이등변삼각형을 이루도록 변에 존재하는 노드의 갯수에 따라 삼각형을 구성하는 방법을 제한한다. 그림1에서 (a),(b),(c)의 첫 번째 삼각형 구조와 (c)의 세 번째 구조는 생성된 삼각형들이 모두 직각 이등변삼각형을 이루고 있으나 나머지 경우는 일반적인 삼각형 구조를 형성하게 된다. 세분화된 삼각형들이 모두 직각이등변삼각형을 이루도록 하기위해 노드의 위치와 개수에 따라 추가노드를 삽입한다. 그림1의

(a)와 같이 노드의 개수가 1개이고 등변에 존재할 경우, 또는 (b)와같이 노드의 개수가 2개이고 모두 등변에 존재할 경우에는 직각의 대변에 추가노드를 삽입하여 삼각화를 이루도록 한다. 이와 같이 직각 이등변삼각형을 이루도록 하기 위해서는 그림2의 (a)와 같이 변에 있는 노드와 그 대각을 연결한 벡터  $\vec{a}$ 와 노드를 포함하는 한 변으로 이루어진 벡터  $\vec{b}$ 가 수직인 위치에 노드가 존재해야 하며 그렇지 않을 경우 직각의 대변에 노드를 추가해 주어야한다.

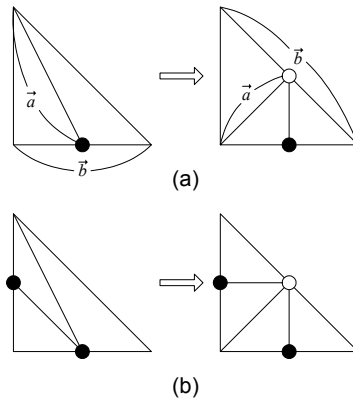


그림 2. 추가노드에 의한 직각이등변삼각화  
 Fig. 2. right-angled isosceles triangulation by node addition.

삼각화를 위한 전제 과정은 다음과 같다.

1. 레벨0일 때 초기메쉬의 구조가 직각이등변삼각형이 되도록 정규화 메쉬를 생성한다.
2. 현재의 레벨에 해당하는 삼각형의 간격을 검사하고 움직임의 양을 측정하여 임계치 이상일 때 삼각형의 각변에 노드를 생성한다.

$m(P) > TH$  일때 노드생성

3. 메쉬의 형태가 직각이등변삼각형을 이루도록 하기위해 모든 메쉬에 대해서 각 변과 그 변에 존재하는 노드와 대각으로 이루어진 벡터의 내적이 0 인지를 검사하여 0인 노드가 없을 경우 직각의 대변에 노드를 추가한다. 전체 메쉬에 대해서 더 이상 추가되는 노드가 없을 때까지 이 과정을 반복한다.

$\vec{a} \cdot \vec{b} = 0$  일때 노드 추가

4. 삼각형의 각 변에 존재하는 노드의 개수에 따라 삼각화를 수행한다.

5. 레벨을 1단계 증가시키고 검색하는 삼각형의 크기를 절반으로 줄인 후 2의 과정을 반복한다.

위와 같은 삼각화 과정의 예를 그림3에서 보이며 claire 영상에 대한 삼각화 결과를 그림4에 나타낸다.

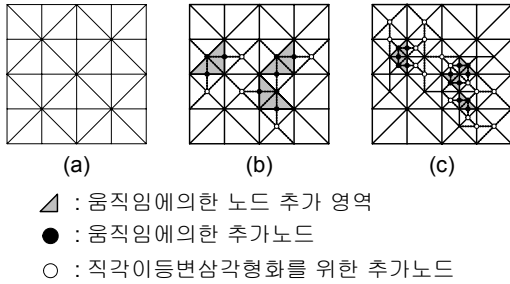


그림 3. 계층적 삼각화의 예  
(a)레벨 0 (b)레벨 1 (c)레벨 2  
Fig. 3. The example of hierarchical triangulation  
(a) level 0 (b) level 1 (c) level 2



그림 4. 계층적 삼각형 메쉬(claire-16번 프레임)  
Fig. 4. hierarchical triangular mesh(claire-frame NO. 16)

### III. 움직임 벡터의 추정

생성된 메쉬로부터 해당 영역의 움직임을 추정하기위해서 메쉬를 형성하는 노드의 움직임 벡터를 구해야 한다. 여기서 구해진 3개의 노드의 움직임 벡터로부터 Affine 변환의 6개의 파라미터를 구할 수 있고 이전 프레임과 현재 프레임의 위치 변화에

대한 관계식을 구할 수가 있다. 움직임 벡터의 추정 은 HMA(hexagonal mesh algorithm)<sup>[1]</sup>과 같이 각 노드의 움직임을 BMA를 사용하여 대략적인 위치를 추정한 후 메쉬를 구성하는 다른 노드의 위치를 고정시키고 하나의 노드만을 이동시키면서 Affine 변환을 하여 복원된 영상과 원영상의 오차가 가장 적어지는 위치를 구하는 방법을 사용한다.

#### 1. BMA를 사용한 초기 검색

노드의 대략적인 이동위치를 검색하기위해 해당 노드를 중심으로 하는 BMA를 사용한다. 계층적 구조로 형성된 메쉬는 레벨이 올라갈수록 메쉬 사이의 간격이 좁아지고 노드의 재 정렬 과정에서 각 노드의 적합한 위치를 추정할 수 있으므로 가장 하위 레벨에 해당하는 레벨 0의 노드에 대해서만 BMA를 적용한다. 나머지 노드들은 레벨0에 해당하는 노드의 움직임 벡터로부터 쌍 선형 보간법을 사용하여 초기 위치를 추정한다.

#### 2. Affine 변환을 이용한 움직임 보상

BMA를 이용해 구해진 노드의 움직임 벡터나 노드의 재 정렬 과정에서 메쉬를 구성하는 삼각형의 세 꼭지점의 이동 위치를 구하고 이로부터 Affine 변환을 통해 움직임 보상된 영상을 구해낸다.

Affine 변환은 식과 같이 삼각형의 세점의 이동 전의 좌표( $x, y$ )와 이동 후 좌표( $x', y'$ )의 위치변화를 6개의 파라미터로 나타낸다.

$$\begin{aligned} u(x, y) &= (x - x') = a_1 + a_2x + a_3y \\ v(x, y) &= (y - y') = a_4 + a_5x + a_6y \end{aligned} \quad (2)$$

삼각형의 세 꼭지점의 이동 전과 이동 후의 좌표 ( $x_i, y_i$ ), ( $x'_i, y'_i$ ) 로부터 다음과 같이 Affine 변환의 파라미터  $a_1, \dots, a_6$ 을 구할 수 있다.

$$\begin{aligned} \begin{bmatrix} u(x_1, y_1) \\ u(x_2, y_2) \\ u(x_3, y_3) \\ v(x_1, y_1) \\ v(x_2, y_2) \\ v(x_3, y_3) \end{bmatrix} &= \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \end{bmatrix} = \begin{bmatrix} 1 & x_1 & y_1 & 0 & 0 & 0 \\ 1 & x_2 & y_2 & 0 & 0 & 0 \\ 1 & x_3 & y_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & x_1 & y_1 \\ 0 & 0 & 0 & 1 & x_2 & y_2 \\ 0 & 0 & 0 & 1 & x_3 & y_3 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \end{bmatrix} \\ &= \begin{bmatrix} \Phi & 0 \\ 0 & \Phi \end{bmatrix} a \end{aligned} \quad (3)$$

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  $\Phi^{-1}$ 로 표현하면

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

여기서

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

구해진 Affine 변수를 식(2)에 대입하면 변환 전과 후의 좌표사이에 관계식을 구할 수가 있다. 일반적으로 Affine 변수  $a$ 는 실수 값으로 나타나고 변환 좌표도 실수형태의 값으로 구해지므로 해당 위치의 화소 값을 구하기 위해서는 쌍 선형 보간법으로 주변에 있는 네개의 화소 값으로부터 해당위치의 화소 값을 계산해 주어야 한다.

### 3. 노드의 재 정렬(Node Refinement)

노드의 이동 위치를 추정하기 위해서 해당 노드를 이동시키며 Affine 변환에 의한 복원 영상과 원 영상사이의 차이가 가장 적은 위치로 재 정렬(refinement)한다. 한 노드의 이동은 주변에 연결된 메쉬에 영향을 미치므로 주변 노드의 위치는 고정시키고 해당 노드의 위치만을 이동시키면서 Affine 변환에 의해 영상 데이터를 복원한다. 이때 에러가 가장 작은 위치를 노드의 움직임 벡터로 결정하고 에러의 평가함수는 그 노드와 연결된 주변 노드들로 구성되는 다각형 영역에 대한 원 영상과 복원 영상사이의 PSNR을 사용한다.

$$PSNR = 10 \log_{10} \frac{255^2}{MSE} \text{ dB} \quad (6)$$

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

여기서  $P$ 는 주변 노드로 구성되는 제한 영역을 나타내고  $N_P$ 는 이 영역에 포함되는 화소의 수,  $I(x,y), I(\widetilde{x,y})$ 는  $(x,y)$ 에서 원 영상의 화소 값과 복원된 화소 값을 나타낸다.

노드의 재 정렬은 레벨 0에서 결정된 노드들을 먼저 처리하고 차례로 각 레벨에 해당하는 노드들의 움직임 벡터를 결정한다. 모든 노드의 재 정렬이

완료되면 위와 같은 재 정렬 과정을 반복하여 모든 노드들이 최적의 위치로 수렴하도록 한다.

수렴여부의 판정은 연결된 주변 노드들이 이전의 재 정렬 과정에서 위치가 변하지 않았고 해당 노드도 위치가 변하지 않았을 때 수렴한 것으로 판정한다. 그림 5은 수렴한 이후에 복원 영상과 각 노드를 나타내고 있다.

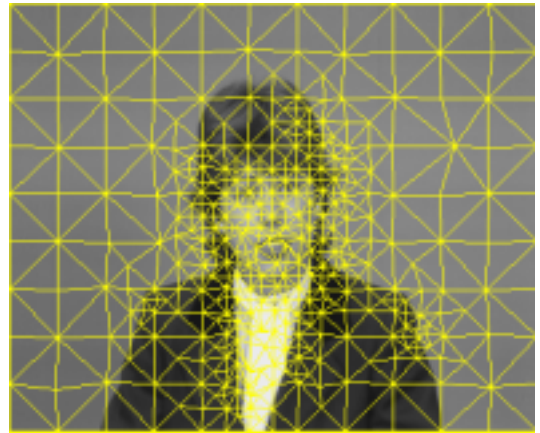


그림 5. 재 정렬후의 노드위치(claire-13번 프레임)  
Fig. 5. the node position after refinement (claire-frame NO. 13)

### IV. 메쉬구조의 부호화와 복원

BAM 이나 정규화 메쉬와 달리 메쉬의 구조가 고정되어 있지 않으므로 메쉬의 구조를 기술하는 추가 정보를 생성해야 한다. 본 논문에서는 움직임의 양에 따라 삼각형을 세분화하고 있으므로 세분화의 결과로 발생된 추가노드의 존재 유무만을 전송하고 이로부터 메쉬 생성의 과정에서와 같은 방법으로 삼각화를 수행함으로써 메쉬 구조를 복원할 수가 있다.

메쉬 구조의 코딩은 그림 6과 같이 각 레벨에 따라 세분화가 이루어진 후 추가된 노드의 개수에 따라 추가노드가 없거나 1개인 경우 0, 2개인 경우 10, 3개인 경우 11를 할당한다. 즉, 1이 나타난 경우 다음 레벨에서 추가적인 세분화가 이루어질 수 있는 메쉬이고 그 다음 비트가 1인지 또는 0인지에 따라 추가노드의 개수가 2개인지 3개인지를 결정할 수가 있다. 레벨이 올라감에 따라 두비트 단위로 나누어 할당된 비트가 00이 아닐 경우 두개의 삼각형으로 이루어진 사각형 영역을 8개의 세분화 삼각형

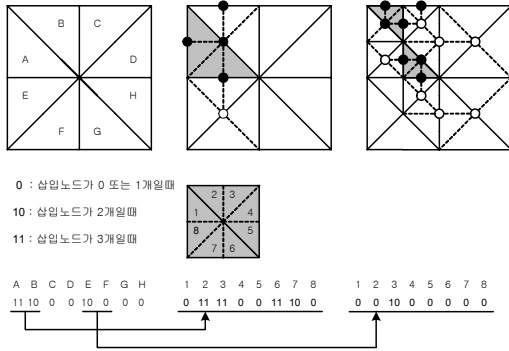


그림 6. 계층적 메쉬 구조의 부호화  
Fig. 6. the coding of hierarchical mesh structure

으로 나누어 추가 노드의 개수를 부호화 한다.  
메쉬 구조의 복원은 수신된 비트 열에서 11이 나타나는 삼각형에 3개의 노드를 추가하고 메쉬 생성 때와 마찬가지로 각각의 대변에 노드를 추가하여 메쉬의 구조를 복원해 나간다.

V. 실험결과

계층적 삼각화 메쉬의 성능평가를 위해 352×288 크기의 “Clair” 영상과 ”Miss America” 영상에서 휘도 신호만을 사용하였고, 1번 화면에서 100번째 화면까지에 대해서 3 프레임 간격으로 여러 가지 방법들을 적용해 보았다.

실험에서는 메쉬의 구조에 따른 움직임 추정 방법의 성능을 비교하기 위해 BMA과 4-8연결 메쉬(4-8M), 헥사 메쉬(HM)<sup>[1]</sup>등의 정규화 메쉬와 HGI<sup>[6]</sup> 방법을 제안한 방법(HTM)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.

$$PSNR = 10 \log_{10} \frac{255^2}{MSE} \text{ dB} \tag{7}$$

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. 정규화 메쉬에서는 노드의 배치거리는 16×16을 사용하고 초기에 BMA을 사용한 탐색 시에는 노드를 중심으로 하는 16×16 영역을 사용하였고 탐

색영역은 BMA과 같이 「-8, +7」을 사용하였다. 노드의 재 정렬시 탐색영역은 「-3,+3」로 설정하였다. HGI에서는 발생하는 노드의 수가 다른 방법과 비슷하도록 분할의 임계치는 35로 하였고 탐색영역도 다른 방법과 마찬가지로 「-3,+3」으로하고 재 정렬 방법도 동일하게 적용하였다.

계층적 삼각화 메쉬(HTM)는 레벨 0에서 노드의 간격은 32×32, 레벨1에서는 16×16, 레벨2에서는 8×8로 설정하여 3단계의 계층구조를 사용하였고, 노드의 삽입을 결정하기 위한 프레임 간 차의 분산값은 초기 문턱값 *InitTH*는 10으로 설정하였으며 나머지 초기검색이나 재 정렬을 위한 조건들은 다른 방법과 같도록 하였다. 재 정렬시 노드의 재 정렬 순서는 가장 낮은 레벨에 해당하는 노드로부터 상위 레벨로 정렬되도록 하였다. 다른 방법과의 비교를 위해 전체 노드의 수가 4-8M이나 HM과 비슷하도록 하기위해 분산의 임계치를 다음과 같이 갱신하여 메쉬의 생성을 반복하면서 전체 노드의 수가 437±5%이내에서 나타나도록 하였다.

$$Th_{var}(n+1) = Th_{var}(n) \times \left\{ 1 + \frac{(TotalNode - 437)}{437} \right\} \tag{8}$$

여기서  $Th_{var}(n)$ ,  $Th_{var}(n+1)$ 은 이전에 사용한 분산의 임계치와 다음에 사용할 임계치를 나타내고 *TotalNode*는 전체 노드의 개수를 의미한다.

정규화 메쉬를 사용한 방법에서는 노드의 재 정렬이나 최종 영상 합성 시 Affine 변환된 좌표로부터 이중 선형 보간 방법을 사용하여 화소 값을 계

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

입력영상	알고리즘	완전수렴	3회반복
clair	BMA	36.0098	
	4-8M	37.9411	37.7845
	HM	38.0738	37.9504
	HGI	38.6513	38.4131
	HTM	39.1313	38.9704
MissAmerica	BMA	36.8027	
	4-8M	38.6077	38.5905
	HM	38.6194	38.6015
	HGI	38.7697	38.7101
	HTM	39.0359	38.9686

산하였다. 또 노드의 재 정렬시 보통 3회 정도 반복하면 충분한 성능을 얻을 수 있으므로 각 알고리즘을 3회 반복 했을 때와 완전 수렴시켰을 때의 성능을 비교하였다. 그림 7은 완전 수렴시켰을 때 움직임 보상된 영상의 PSNR을 나타내고 표 1은 각 조건에 따른 평균 PSNR을 나타낸다.

표2는 각 방법에 따른 노드의 수와 메쉬의 구조를 전송하기 위한 비트수를 비교한 것이다. 각 노드의 움직임 벡터는 x, y 각각 4비트를 할당하고 HGI와 HTM에서의 전송비트 수는 메쉬의 구조를 전송하기 위한 비트수와 각 노드의 움직임 벡터를 전송하기 위한 비트를 합한 값이 되며 이 경우 노드의 갯수가 입력영상에 따라 가변되므로 평균 비트수를 제시하였다.

일반적으로 정규화 메쉬를 사용한 방법들이 BM A보다는 뛰어난 성능을 나타내고 있고 4-8M과

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

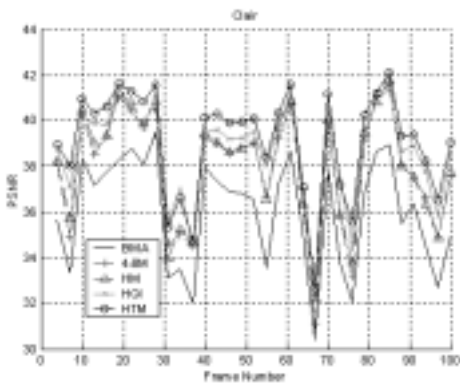
입력영상	알고리즘	전송비트수(bits)
clair	BMA	396*8
	4-8M	437*8
	HM	428*8
	HGI	430.6*8+452.21
	HTM	434.6*8+666.27
Miss America	BMA	396*8
	4-8M	437*8
	HM	428*8
	HGI	483.9*8+474.76
	HTM	436.0*8+681.18

HM은 거의 같은 성능을 나타내고 있다. 본 논문에서 제안한 HTM는 다른 정규화 메쉬 뿐만 아니라 비슷한 수의 노드를 사용하는 HGI에 비해서도 높은 PSNR을 얻을 수 있었다. 정규화 메쉬에 비해서 메쉬의 구조를 전송하기위한 추가정보를 필요로 하지만 5장에서 제시한 방법에 의해 생성되는 비트열을 줄일 수 있어 복원 시 발생하는 잔류오차의 양이 줄어들게 됨으로 전체적인 정보의 양이 비슷해지게 된다.

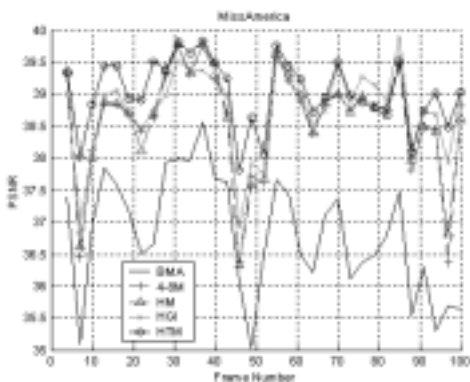
본 논문에서 제안한 방법은 완전 수렴했을 때 다른 정규화 메쉬보다 평균 0.5dB에서 1dB정도 우수한 성능을 나타내고 있고 특히 움직임이 큰 영상에서는 상대적으로 뛰어난 성능을 발휘함을 알 수 있다. 표 1에서와 같이 반복 횟수가 3일 때도 다른 방법보다 초기 수렴 성능에서도 우수함을 알 수 있다.

## VI. 결론

본 논문에서는 입력영상의 움직임에 따라 메쉬의 구조가 가변되는 계층적 삼각화 방법을 제안하였다. 연속된 입력영상의 차의 분산을 이용하여 움직임의 양을 측정하고 이 값의 분포로부터 추가 분할할 영역을 선택함으로써 움직임이 많은 영역에 노드가 집중되도록 하였다. 추가 분할시 계층적 구조를 이룰 수 있도록 생성된 메쉬의 구조도 초기상태의 메쉬와 같이 직각이등변삼각형을 구성하는 방법을 사용하였다. 계층적 구조의 삼각형을 형성함으로써 추가분할이 가능한 형태의 메쉬와 불가능한 형태의 메쉬를 구분함으로써 메쉬의 구조를 기술하기위한 추가정보의 양을 최소화할 수 있는 부호화 방법을



(a)



(b)

그림 7. 움직임 보상된 영상의 PSNR  
(a) Claire (b) Miss America  
Fig. 7. The PSNR of motion compensated image  
(a) Claire (b) Miss America

제한하였다. 여러 가지 동영상에 대한 실험결과 움직임이 많은 영역에 노드의 할당을 집중함으로써 P SNR이 비슷한 수의 노드를 가지는 다른 방법 보다 평균 0.5dB에서 1dB정도 뛰어난 성능을 나타냈고 저 비트율의 영상전송에 적합한 방법임을 확인하였다. 제안한 방법을 객체기반 움직임 추정에 적용하여 임의의 형태를 가지는 객체에 대해서 적용할 수 있는 방법에 대한 연구가 필요할 것으로 생각된다.

참고문헌

[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] C.S. choi and T. Takebe, "Analysis and synthesis of Facial Image Sequences in Model-Based Image Coding," IEEE Trans. on Circuits and Systems for Video Technology, vol. 4, no. 3, pp. 257-275, June 1994.

[6] C-L Hung and C-Y Hsu, "A New Motion Compensation Method for Image Sequence Coding Using Hierarchical Grid Interpolation," IEE

E Trans. on Circuits and Systems for Video Technology, vol. 4, no.1, Feb. 1994.

이 동 규(Dong-Gyu Lee)

정회원



1991년 2월: 한양대학교 전자공학과 학사  
1993년 2월: 한양대학교 전자공학과 석사  
1993년 3월~현재: 한양대학교 전자공학과 박사과정

<관심분야> 영상처리, 화상통신, DSP 응용

이 두 수(Doo-Soo Lee)

정회원



1968년 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.

<관심분야> DSP, 화상처리, 의공학