

# 다차원 영상 시스템을 위한 변형계층 모션벡터 추정알고리즘

김 정 웅<sup>†</sup> · 양 해 술<sup>††</sup>

## 요 약

다양한 종류의 컴퓨터가 사람, 사물, 환경 속에 내재되어 있고, 이들이 서로 연결되어, 필요한 곳에서 활용할 수 있는 유비쿼터스 환경에서는 홈 네트워크를 통해 이 기종 기기간 다양한 데이터 교환을 요구한다. 더욱이 원활한 영상 데이터의 처리, 전송, 모니터링 기술은 핵심적 요소가 아닐 수 없다. 공간 및 시간적인 해상도, 컬러의 표현 그리고 화질의 측정방법 등 고전적 영상 처리 연구 분야뿐만 아니라 국한된 대역폭을 갖는 홈 네트워크의 전송 체계에서 전송률 문제에 대한 심도 있는 연구가 필요하다. 본 논문에서는 홈 네트워크 상황에서 콘텐츠의 중심이 되는 영상 데이터의 전송과 처리 그리고 제어를 위하여 새로운 움직임 추정 알고리즘을 제안하고 이를 이용하여 다중카메라에서 전송된 다차원 영상의 실시간 모니터링 시스템을 구현한다. 각도, 거리등 다양한 환경에서 전송되어지는 스테레오 카메라의 영상 데이터들은 축소, 확대, 이동, 보정 등 전처리 후 제안된 움직임 보상을 위한 변형계층 모션벡터 추정 알고리즘을 이용하여 압축 처리, 전송 된다. 기존 모션벡터 추정 알고리즘인 전역 탐색, 3단계 탐색, 계층적 탐색이 갖는 장점을 계승하고 단점을 보완한 변형계층 알고리즘은 비정형, 소형 매크로 블록을 이용하여 휘도의 편차가 큰 영상의 효율적 움직임 추정에 이용된다. 본 논문에서 제안한 변형계층 움직임 추정 알고리즘과 이를 이용해 구현된 영상 시스템은 유비쿼터스 환경에서 다양하게 활용될 수 있다.

키워드 : 유비쿼터스, 홈네트워크, 영상전송, 계층적 모션 벡터 추정

## Metamorphosis Hierarchical Motion Vector Estimation Algorithm for Multidimensional Image System

Jeong-Woong Kim<sup>†</sup> · Hae-Sool Yang<sup>††</sup>

## ABSTRACT

In ubiquitous environment where various kinds of computers are embedded in persons, objects and environment and they are interconnected and can be used in any place as necessary, different types of data need to be exchanged between heterogeneous machines through home network. In the environment, the efficient processing, transmission and monitoring of image data are essential technologies. We need to make research not only on traditional image processing such as spatial and visual resolution, color expression and methods of measuring image quality but also on transmission rate on home network that has a limited bandwidth. The present study proposes a new motion vector estimation algorithm for transmitting, processing and controlling image data, which is the core part of contents in home network situation and, using algorithm, implements a real time monitoring system of multi dimensional images transmitted from multiple cameras. Image data of stereo cameras to be transmitted in different environment in angle, distance, etc. are preprocessed through reduction, magnification, shift or correction, and compressed and sent using the proposed metamorphosis hierarchical motion vector estimation algorithm for the correction of motion. The proposed algorithm adopts advantages and complements disadvantages of existing motion vector estimation algorithms such as whole range search, three stage search and hierarchical search, and estimates efficiently the motion of images with high variation of brightness using an atypical small size macro block. The proposed metamorphosis hierarchical motion vector estimation algorithm and implemented image systems can be utilized in various ways in ubiquitous environment.

Key Words : Ubiquitous, HomeNetwork, Image Transmission, Hierarchical Motion Vector Estimation

## 1. 서 론

다양한 종류의 컴퓨터가 사람, 사물, 환경 속에 내재되어

있고, 이들이 서로 연결되어, 필요한 곳에서 활용할 수 있는 유비쿼터스(Ubiquitous) 환경에서는 서로 다른 기기들에서 데이터를 교환하게 된다. 특히, 가정이나 건물 내에서 네트워크 콘텐츠는 저품질에서 고품질로, 단방향에서 양방향으로, 그리고 영상 중심으로서의 전송이 많은 기기들에 의해 진행되어지고 있다.

다수의 카메라에서 전송된 다양한 영상을 단안 영상으로

※ 본 연구는 정보통신부 지원 ITRC 프로그램의 지원을 받아 수행되었음.

† 정 회 원 : 서울벤처정보대학원대학교 겸임교수

†† 종신회원 : 호서대학교 벤처전문대학원 교수

논문접수: 2005년 12월 8일, 심사완료: 2006년 2월 22일

표현 처리하거나 이들을 모니터링 또는 제어하는 것은 홈오트메이션이나 유비쿼터스 환경에서는 꼭 필요한 요구 기술이다. 그러나 국한된 전송 대역폭을 갖는 홈 네트워크 전송 체계에서 다중카메라로 부터 전송된 영상을 축소, 확대, 이동, 회전, 보정 등 여러 전처리 과정을 거쳐 압축, 해독, 전송하여 실시간으로 모니터링, 제어하는 것은 쉬운 문제가 아니다.

본 연구는 한정된 자원을 사용해야 하는 유비쿼터스 환경에서의 핵심 정보인 영상 데이터 처리 방안과 다중 카메라에서 전송되어지는 여러 화면이 각각 독립적으로 압축될 경우 단안 화상을 전송하는 것과 비교해 볼 때, 스테레오 화상 전송에 필요한 대역폭은 두 배 이상으로 증가하는 문제를 해결하려는데 연구의 목적이 있다. 본 연구에서는 유비쿼터스 환경에 적합한 영상 모니터링, 제어 시스템을 구현하고자 한다.

먼저 영상의 전처리 과정과 기존 영상의 압축 시 요구되는 움직임 보상을 위한 모션 벡터 추정 알고리즘에 대한 선행 연구를 통하여 본 논문에서는 다중 카메라에서 입력된 영상을 기존의 H.264 방식에서 사용하는 3단계 알고리즘과 계층적인 움직임 벡터 추정 알고리즘을 개선한 변형 알고리즘을 제안한다[1, 2].

이를 통해서 우월한 지위를 확보하고 있는 계층적 방법의 한계를 극복한다. 또한 제안된 효율적 방법으로 영상을 압축하고 전송하게 된다. 전송된 영상은 실시간 모니터링 할 수 있으며, 이러한 시스템은 다차원 영상 전송과 제어에 활용될 수 있다.

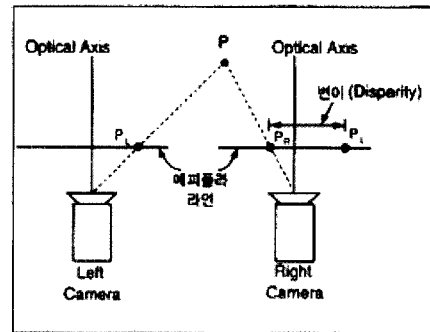
본 논문의 구성은 2장에서 관련 연구로 스테레오 영상 압축의 전처리 과정을 통해 유비쿼터스와 그 환경에서 활용될 수 있는 멀티미디어적 요소인 영상의 기본 처리 및 전송 패턴에 대하여 알아본다. 3장에서는 기존 움직임 벡터 추정 알고리즘에 대해서 살펴보고 4장에서 본 논문에서 제안하는 변형 계층 움직임 벡터 추정 방법에 대해서 기술한다. 5장에서는 이를 증명하기 위하여 실험 및 결과를 갖고 제안된 방법을 통하여 구현한 영상 전송 및 모니터링 시스템을 소개한다. 마지막으로 6장에서는 본 논문의 결론을 맺는다.

## 2. 관련 연구

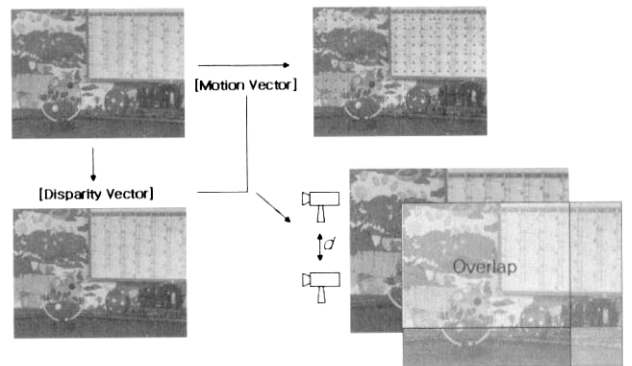
### 2.1. 스테레오 영상 압축의 전처리

스테레오 영상의 기술은 2차원 영상으로부터 3차원 정보를 추출하는 과정으로 2대의 카메라를 이용하는 방법이다. 3차원 정보는 사물에 대한 원근감, 위치, 속도, 깊이 등의 정보를 지니고 있으며, 입체 영상을 만들기 위해서는 최소 좌, 우로 구성된 2장 이상의 평면영상이 필요하며, 일반적으로 2대 이상의 카메라를 이용하여 동시에 영상을 얻는다[3-5].

주어진 한 쌍의 2차원 좌표로부터 3차원 좌표를 만들기 위해서는 각 영상간의 블록 중 대응에 관한 일치 문제(correspond problem)가 해결되어야 한다. 이러한 기본적인 제약사항은 (그림 1)과 같은 에피폴라 기하학(epipolar geometry)[10]으로 해석한다. 에피폴라 평면(epipolar plane)은 공간상에서 점  $P$ 와 초점거리  $F$ 를 가진 두 렌즈의 중심을 연



(그림 1) 스테레오 카메라의 에피폴라 기하학



(그림 2) 스테레오 영상의 중첩 영역

결하는 선에 의해 정의된다. 또한 이평면과 각 투영 영상이 이루는 교선을 에피폴라 선(epipolar line)이라 한다. 에피폴라 기하학이란, 특정한 에피폴라 평면의 모든 점들을 영상 평면의 에피폴라 선상의 해당하는 점에 대응된다는 기하학적 원리이다. 이때, 점  $P$ 에 대한  $P_L$ 과  $P_R$ 의 위치 차를 변위(disparity)라하며, 이를 벡터로 표시한 것이 변위 벡터이다.

스테레오 카메라에서 입력된 두 영상 사이에는 회전, 이동 및 크기의 변화가 있기 때문에 (그림 2)와 같이 움직임 벡터와 변위 벡터를 구하게 되면 큰 오차를 갖게 된다. 따라서 스테레오 영상 쌍 중 기준이 되는 영상을 독립적으로 압축하고, 다른 영상은 기준이 되는 영상에서 움직임 벡터와 변위 벡터를 추출 후 이를 보상한다. 또한, 각각의 카메라에서 취득한 영상의 컬러와 노이즈(noise)가 다르므로 에러가 발생되기 때문에 이에 대한 해결책도 필요하다.

본 논문에서는 FFT(Fast Fourier Transform)를 이용하여 두 카메라의 회전, 이동 및 크기의 상대적인 변화를 추정하고, 이를 통해 계산된 중첩 영역에서 정보를 이용하여, 두 입력 영상의 컬러 값을 보정한다. 중첩되지 않은 영역의 정보는 표준 압축 방식인 H.264를 이용하여 압축하고, 중첩된 영역은 H.264 방식에서 사용하는 3-단계 알고리즘을 개선한 계층적인 움직임 벡터를 변형한 추정 방법을 통하여 영상을 압축하고 전송한다. 이렇게 함으로써 스테레오 영상의 높은 공간적 중복성을 제거하고, 변위 추정을 거쳐 나오는 변위 벡터를 수신 단에 전송하여 추정된 변위 벡터를 통해 예측된 값과 실제 영상의 차이를 변환 부호화해서 높은 압축 효율을 얻을 수 있다[11, 12].

2.2 스테레오 카메라의 상대적인 변화 추정

FFT 알고리즘을 이용하여 주파수 도메인에서 스테레오 카메라가 취득한 영상의 이동, 회전 및 크기의 상대적인 변화를 추정하고, 이를 통해 중첩 영역을 계산해 낸다[11, 12]. 첫 번째, 영상의 이동 변환에 있어서는 수식 1과 같이 푸리에 변환의 시간천이성질(Time Shift Property)을 이용하며,

$$f_2(x, y) = f_1(x - x_0, y - y_0) \xrightarrow{F.T} F_2(u, v) = e^{-j2\pi(ux_0 + vy_0)} F_1(u, v) \quad \text{수식(1)}$$

$$f_2(x, y) = af_1(x - x_0, y - y_0) + b \xrightarrow{F.T} F_2(u, v) = e^{-j2\pi(ux_0 + vy_0)} a \times F_1(u, v) + b \times \delta(u, v)$$

수식 2와 같이 Cross-power Spectrum 상에서  $(x_0, y_0)$ 에서 임펄스가 발생 했을 때 두 신호가 정합되는 이동 위치를 계산한다.

$$\frac{F_1(u, v)F_2^*(u, v)}{|F_1(u, v)F_2^*(u, v)|} = e^{j2\pi(ux_0 - vy_0)} \xrightarrow{I.F.T} \delta(x - x_0, y - y_0) \quad \text{수식(2)}$$

두 번째, 영상의 회전 변환에서는 극 좌표계를 이용하며, 회전이 있는 두 신호의 관계는 수식 3과 같이 표현할 수 있다.

$$f_2(x, y) = f_1(x \cos \theta_0 + y \sin \theta_0, -x \sin \theta_0 + y \cos \theta_0) \xrightarrow{F.T} F_2(u, v) = F_1(u \cos \theta_0 + v \sin \theta_0, -u \sin \theta_0 + v \cos \theta_0) \quad \text{수식(3)}$$

$$\times e^{-j2\pi(ux_0 - vy_0)} \times a + b \times \delta(u, v)$$

그리고 magnitude  $M_1, M_2$ 가 있을 때, 위의 성질을 이용하여, 수식 4와 같이 나타낼 수 있다.

$$M_2(u, v) = |a| M_1(u \cos \theta_0 + v \sin \theta_0, -u \sin \theta_0 + v \cos \theta_0) \quad \text{수식(4)}$$

여기서, 극 좌표계에서 폴라(polar) 좌표계로 변환하게 되면,  $M_2$ 는 수식 5와 같이 나타낼 수 있다.

$$M_2(\rho, \varphi) = |a| \times M_1(\rho, \varphi - \theta) \quad \text{수식(5)}$$

$$(\rho = \sqrt{u^2 + v^2}, \varphi = \tan^{-1}(v/u), u = \rho \cos \varphi, v = \rho \sin \varphi)$$

따라서,  $M_1, M_2$ 를  $H_1, H_2$ 로 FT를 수행하면 수식 6을 얻을 수 있으며, 이를 이용하여 Cross-power spectrum 상에서  $(x_0, y_0)$ 에서 임펄스가 발생 했을 때 두 신호가 정합되는 회전 위치를 계산한다.

$$\frac{H_1(v, \beta)H_2^*(v, \beta)}{|H_1(v, \beta)H_2^*(v, \beta)|} = e^{j2\pi(\beta \times \theta)} \xrightarrow{I.F.T} \delta(\theta, \theta), \quad \text{수식(6)}$$

$$H_2(v, \beta) = |a| \times e^{-j2\pi(\beta \times \theta)} H_1(v, \beta)$$

마지막으로, 영상의 크기 변환은 수식 7과 같은 푸리에 변환 스케일 성질(scale property)을 이용하며,

$$f_2(x, y) = f_1(x/a, y/b) \xrightarrow{F.T} F_2(u, v) = \frac{1}{|ab|} F_1(u/a, v/b) \quad \text{수식(7)}$$

다음과 같이 수식 8을 이용하여 스케일 변환을 시키게 된다.

$$M_2(\log \rho, \varphi) = M_1(\log \rho - \log a, \varphi - \theta) \quad \text{수식(8)}$$

$$\rho = \sqrt{u^2 + v^2}, \varphi = \tan^{-1}(v/u), u = \rho \cos \varphi, v = \rho \sin \varphi$$

2.3 중첩 영역의 정보를 이용한 컬러 값 보정

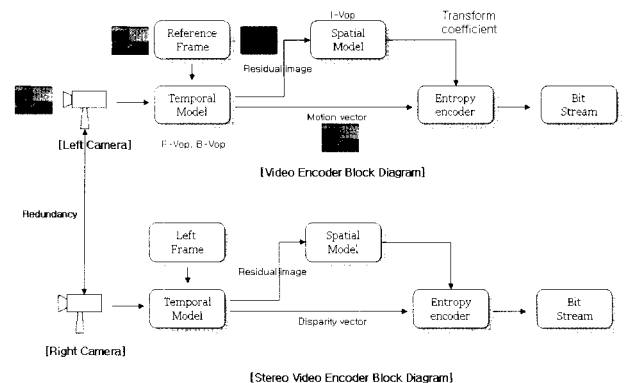
스테레오 영상을 획득하는 과정에서 카메라의 불일치와 외부 환경에 의해 불균형 문제가 발생하게 된다. 좌우 카메라의 변수들이 정확히 일치하지 않기 때문에 획득한 영상의 밝기나 기타 문제들이 발생할 수 있고, 광원의 위치에 의해 물체와 카메라간의 거리등에 의해 획득한 영상의 밝기 차이가 날 수 있다. 이러한 불일치는 변이 추정을 하는 과정에서 오류를 발생 시킬 수 있기 때문에 전처리 과정에서 컬러 값 보정(Balancing)을 수행하여야 한다. 일반적으로 컬러 값 보정에는 수식 9와 같이 영상 전체에 대한 밝기의 평균과 표준편차를 서로 보정해 주는 방법이 많이 사용된다.

$$I_R(i, j) = \frac{\sigma_L}{\sigma_R} (I_L(i, j) - \mu_R) + \mu_L, \forall (i, j) \in I_R \quad \text{수식(9)}$$

$\mu$ 는 각각의 영상의 밝기 평균값이고,  $\sigma$ 는 표준편차를 나타낸다. 이 수식은 좌 영상을 기준 영상으로 하여 우 영상의 전체적인 밝기를 좌 영상에 맞게 변화 시키는 일을 수행한다.

2.4 중첩 현상에 따른 영역의 상이한 압축 방법

중첩 되지 않은 영역의 정보는 (그림 3)의 과정을 통해 표준압축 방식인 H.264를 이용하여 압축을 한다. 스테레오 카메라에서 왼쪽 카메라를 참조영상으로 하고 오른쪽 카메라에서



(그림 3) 스테레오 Encoder

획득한 영상은 차이벡터를 기준으로 인코딩하여 비트열을 구할 수 있다.

중첩된 영역은 기존의 H.264 방식에서 사용하는 3단계 알고리즘을 개선한, 계층적인 움직임 벡터 추정과 변위 추정을 변형한 새로운 형식의 알고리즘을 통하여 영상을 압축하고 전송한다. 방법은 본 논문 4장에서 제안한다.

### 3. 기존 움직임벡터 추정알고리즘

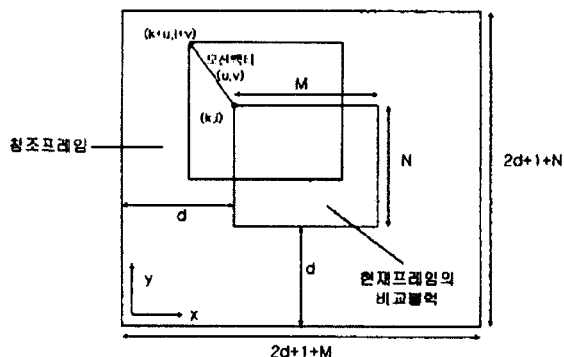
움직임 추정 방식 중 블록 정합 방식은 블록의 픽셀들이 프레임간의 서로 같은 방향으로 움직인다는 특성을 이용한 움직임 계층 방법이다. 블록 정합 알고리즘은 사물의 이동이나 움직임을 추적하기 위하여 카메라나 비디오 카메라로 촬영 후, 생성되는 연속된 이미지 영상들에 대하여 프레임 안에 일정한 크기의 블록을 지정하고 블록안의 픽셀들에 대한 밝기 값을 계산하여 움직임에 대한 움직임 벡터를 구함으로써 움직임 예측을 하게 된다. (그림 4)는 블록 정합 방법의 기본원리를 설명하고 있다[14].

(그림 4)에서 현재 프레임의 블록의 크기는  $M \times N$ 이고 움직임 벡터의 최대 변위는  $x$ 축과  $y$ 축 모두  $\pm d$ 라고 가정한다면 이미지 간의 사물의 움직임을 나타내는 움직임 벡터  $(u, v)$  는  $(2d+1+M) \times (2d+1+N)$  크기의 참조 프레임 안에서 대응되는 블록의 위치를 찾음으로서 구할 수 있게 된다.

대응되는 블록의 위치를 찾기 위해서는 수식 10과 같이 정의되는 블록간의 불일치도를 이용한다. 여기에서  $L'(i, j)$  는 현재 프레임 블록안의 픽셀들에 대한 밝기 값을 나타낸다.  $D(u, v)$ 를  $u, v$ 가  $-d \leq u \leq d, -d \leq v \leq d$ 인 영역에 대하여 구한 다음,  $D(u, v)$  값이 최소가 되는 점을 대응점으로 선정하게 된다.

$$D(u, v) = \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} |L''(i+u, j+v) - L'(i, j)| \quad \text{수식(10)}$$

본 논문에서는 블록 정합 방식에서 기존의 H.264의 3단계 알고리즘을 개선한 계층적 움직임 벡터 추정 알고리즘을 변형 적용 시켰다. 입력되는 현재 프레임에 일정한 크기의 블록을 지정하고 다음 프레임 안에서 현재 프레임과 대응되는



(그림 4) 블록 정합의 기본 원리

블록의 좌표를 찾음으로서 두 프레임 사이의 움직임 벡터를 구할 수 있다.

#### 3.1 전역 탐색 알고리즘(Full Search Algorithm)

전역탐색 알고리즘은 현재 프레임의 블록에 대응되는 블록을 참조 프레임에서 찾는 방법 중에서 가장 기본이 되는 방법이다. 이 알고리즘은 입력으로 들어오는 프레임들에서 현재 프레임에 대응되는 블록을 참조 프레임에서 찾기 위하여 대응블록이 존재 할 수 있는 탐색공간의 모든 위치에서 두 프레임 블록 간 픽셀에 대한 불일치도  $D(u, v)$ 를 계산하여 그 중에서  $D(u, v)$  값을 비교하기 때문에 정확도면에서는 우수하다 그러나 수식 11과 같은 최수의 블록비교를 하기 때문에 처리 속도가 떨어지는 단점을 가지고 있다[15, 19, 23].

$$N = (2d+1) \times (2d+1) \quad \text{수식(11)}$$

#### 3.2 K-단계 탐색 알고리즘(K-Step Search Algorithm)

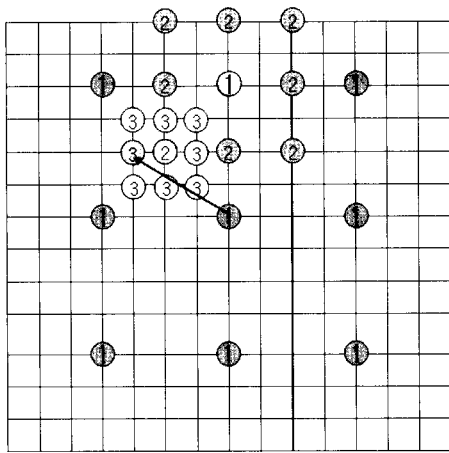
K-단계 탐색 알고리즘은 전역탐색 알고리즘의 처리 속도가 떨어지는 단점을 개선한 방법이다. K-단계 탐색 알고리즘은 블록 간의 대응점을 찾기 위하여 첫 번째 단계에서는 움직임 벡터의 이동범위인  $d$ 의 절반의 값으로 거리를 유지하는 점을 탐색 공간 안에 위치시킨 후, 블록 안에 지정된 각 점들의 위치에서 두 프레임 사이에 존재하는 블록 안의 픽셀들에 대한 불일치도  $D(u, v)$  값들을 계산하여  $D(u, v)$  값이 최소인 점의 위치를 초기 선택위치로 선정하게 된다. 그다음 단계로서, 앞 단계에서 선택된 점들 간의 거리가 절반인 거리를 유지하는 점들을 위치시킨다. 여기에서 초기 변위의 절반 값이 소수점일 경우에는 반올림 값을 적용하여 점의 거리를 유지하게 된다. 그리고 다시 한 번  $D(u, v)$  값을 계산한다. 이어서  $D(u, v)$  값이 최소가 되는 위치, 즉 픽셀에 대한 밝기 값의 차가 최소인 점을 다시 선택하게 된다. 위와 같은 단계를 반복하여 블록 안에 지정된 점들의 거리가 1이 될 때까지 반복한다. 점들의 거리가 1이 되었을 때 블록 안의 픽셀 값차가 최소인 점을 찾게 되고 이 점을 최종 대응점으로 선택하게 된다. K-단계 탐색 알고리즘의 블록 비교회수는 수식 12와 같다.

$$N = \lceil \log_2(d+1) \rceil \quad \text{수식(12)}$$

K-단계 탐색 알고리즘은 탐색 공간 내에 일정한 위치를 지정하고 그 위치에서만 K번의 블록 비교를 하기 때문에 전역탐색 알고리즘에 비해 블록 비교 횟수가 현저하게 줄어들어 처리속도가 향상되는 장점을 가지고 있다[20, 22].

#### 3.3 3-단계 탐색 알고리즘(3-Step Search Algorithm)

3-단계 탐색 알고리즘은 기존의 탐색 반복 단계를 블록 사이즈와는 상관없이 3단계로 제한하는 대신 블록 안의 대응점을 찾기 위한 탐색 점의 간격을 좁혀서 블록 내에서 정밀 탐색하는 방법이다. 3-단계 탐색 알고리즘은 기존의 탐색 방



(그림 5) 3-단계 탐색 방법

법에서 발생할 수 있는 대응점의 오차를 줄이고자 하는 방법이다. 3-단계 탐색 알고리즘은 입력되는 두 프레임에 지정된 블록 안의 점간의 거리가 단계별로 4→2→1이나 3→2→1이 되도록 지정하여 3단계 탐색을 진행하게 됨으로서, 단 3번의 단계를 거치면서 두 블록간의 픽셀에 대한 최소 밝기값의 차이를 갖는 대응점을 찾는 방법이다. (그림 5)와 같이 첫 번째 단계에서 입력되는 두 프레임에 존재하는 블록 안의 거리가 4를 유지하는 점들을 위치시키고 이점들에 대하여 두 블록의 픽셀에 대한 불일치도  $D(u,v)$  값을 계산하여  $D(u,v)$  값이 최소가 되는 점을 선정하게 된다. 두 번째 단계로서 앞 단계에서 선정된 점들을 중심으로 점들의 거리가 앞 단계의 절반인 2가 되는 점들의 위치에서 다시하면  $D(u,v)$  값을 계산하여  $D(u,v)$  값이 최소가 되는 점을 선정하게 된다. 마지막 단계로서 앞 단계에서 선정된 점들을 중심으로 거리가 앞 단계의 절반인 1을 유지하는 점들에 대한 블록간의 픽셀에 대한 불일치도  $D(u,v)$  값을 계산하여  $D(u,v)$  값이 최소가 되는 점의 위치를 찾아 이것을 최종 대응 블록으로 선정하게 된다[17, 18].

이 3단계 알고리즘은 기존의 탐색 알고리즘의 대응점을 찾기 위한 탐색 반복 회수를 블록의 크기와는 상관없이 3회로 제한함으로써 두 프레임간의 대응점을 탐색하는데 필요한 계산량을 줄임으로서 처리시간을 더욱 단축시킬 수 있는 장점을 가지고 있다.

기존의 탐색방법은 블록 크기에 따라 대응점을 찾기 위한 초기 탐색 점들 간의 거리가 유동적으로 변할 수 있어서 블록의 크기가 커지면 정확한 초기 대응점을 찾는데 오차가 발생할 가능성이 높다. 하지만 3단계 탐색 알고리즘은 첫 번째 단계에서 블록 안에 지정된 점들의 거리가 블록의 크기와는 상관없이 4 또는 3으로 지정되기 때문에 블록 크기의 변화와는 상관없이 대응점을 찾기 위하여 초기에 더욱 조밀하게 정밀 탐색하게 되고 이에 따라 첫 번째 단계에서 상당히 정확한 대응점을 찾을 수 있게 된다. 또한 앞 단계의 정확성에 기반을 두어 이어지는 2단계와 3단계의 탐색 단계를 거침으로서 더욱 정확한 대응점을 찾을 수 있기 때문에 기존의 탐색 방법의 대응점에 대한 오차를 줄일 수 있게 된다[20].

### 3.4 계층적 탐색 알고리즘(Hierarchical Algorithm)

움직임 탐색 점의 개수와 SAD의 계산에 고려되는 화소의 개수를 함께 감소시켜 계산량을 감소시키는 대표적인 움직임 추정 방법이 움직임 벡터 필드의 계층적인 탐색이다. 이 방법은 낮은 복잡도와 높은 효율을 제공하며 이 두 가지 기능 사이에서 균형을 유지하는데 있어서 큰 유연성을 보장한다.

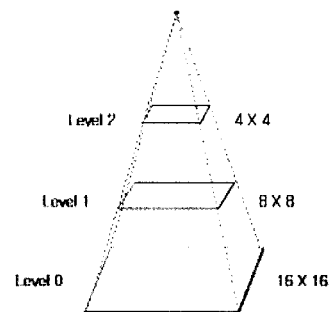
전역 탐색 기법은 계산 비용을 고려하지 않은 대신에 높은 효율을 제공하고 3단계, 4단계, 마름모꼴 탐색 등의 휴리스틱 기법은 선택된 블록에서만 연산이 이루어지기 때문에 계산량은 적지만 비교적 효율성이 낮다. 이에 비해서 계층적인 기법은 빠르고 효율적인 접근 방법으로써 현재와 이전, 두 이미지는 몇 가지 단계의 크기와 해상도로 줄어드는데 각 단계의 이미지는 동일한 이미지에 대하여 특정의 인수(일반적으로 2)에 의해서 줄어든 해상도를 갖는다. 결과적으로 (그림 6)와 같은 피라미드 구조를 이루게 되는데 가장 낮은 단계가 원본 이미지이고 그 위 단계에서는 원본 이미지의 1/4 해상도를 갖는 이미지가 생성된다.

초기의 이미지로부터 더 낮은 해상도의 이미지를 생성하기 위해 두 개의 접근법이 사용될 수 있다. 하나는 휘도의 평균값(mean intensity)을 이용하는 것이고 다른 하나는 부표본을 만드는 것(subsampling)이다.

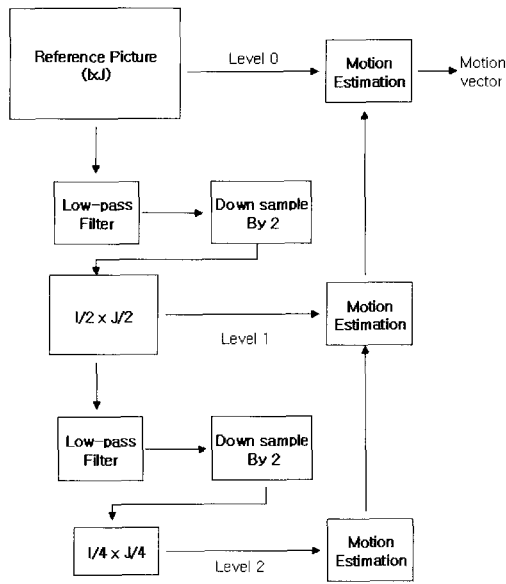
계층적인 움직임 추정은 매우 낮은 계산 비용과 더불어 높은 효율을 제공하고 그 다양성뿐만 아니라 블록 크기, 단계의 수, 크기 조정 인수 등의 특정화된 많은 매개변수들 때문에 영상 부호화, 스테레오 비전을 포함하여 움직임 추정 등의 응용에 널리 사용되는 기법이다.

현재 프레임에 대한 피라미드 구조는 0단계의 이미지로부터 더 높은 단계(저해상도)의 이미지를 생성함으로써 이루어진다. (그림 7)에서는 참조 영상에 대해 Level 1과 Level 2에서 두 장의 저해상도 영상이 필요함을 보이고 있다.

다음 이전 이미지에서도 피라미드 구조상의 같은 단계에서 생성된 이미지에 대하여 탐색을 실행한다. 현재 단계에서 최적의 정합 영역을 찾은 후에 다음단계로 그 결과를 전파하는 것이다. 움직임 벡터는 크기 조정 인수(scaling factor : 일반적으로 2)를 x, y 좌표에 곱하는 방법으로 다음 단계에 전파된다. 움직임 벡터의 정제를 위한 또 다른 방법은 단지 4개의 (L-1)단계 블록들을 고려해서 1개의 (L)단계 블록으로부터 결과를 얻는 것이다. 이 방법은 확실히 처리 속도가 빠르나



(그림 6) 영상의 해상도에 따른 피라미드 구조



(그림 7) 계층적 움직임 추정 방법

정확도가 떨어진다는 단점이 있다. 또한 초기 추정이 이루어진 부근의 한정된 탐색 영역 내에서 각 단계에 대하여 전역 탐색을 다시 실행하는 것도 가능하다[19].

### 3.5 적용한 기존 알고리즘 성능 비교

계층적인 움직임 추정은 최상위 단계에서 유사한 블록 거리 측정값(Block Distance Measure Value)을 갖는 블록의 수가 많으면 그 블록들은 모두 낮은 단계에서 정제될 수 있고 최적의 블록이 선택될 수 있다. 이 과정을 통해 계산 비용의 증가는 있지만 그것에 비해 만족할만한 결과를 얻을 수 있다. 계층적인 움직임 추정은 매우 낮은 계산 비용과 더불어 높은 효율을 제공하고 그 다양성뿐만 아니라 블록 크기, 단계의 수, 크기 조정 인수 등의 특징화된 많은 매개변수들 때문에 영상 부호화, 스테레오 비전을 포함하여 움직임 추정 등의 응용에 널리 사용되는 기법이다.

또한 움직임 추정 오차의 측면에서는, 전역 탐색법, 계층적 탐색법, 3-단계 방법인 대수적 탐색법의 순서대로 우수하다고 알려져 있다. 대수적 탐색법이 계층적 탐색법에 비해 성능이 떨어지는 것은 탐색하는 동안 지역 최소값에 빠질 확률이 크기 때문인 것으로 추정된다. 계층적 탐색법은 저 대역 통과 필터링의 효과로 인하여 지역 최소값에 빠질 가능성이 줄어든다. 이러한 결과는 16x16 매크로블록 또는 8x8 블록에 대한 움직임 추정에 대해 대체적으로 성립한다. 그러나 이 보다 작은 8x4, 4x4 블록의 경우에는 계층적 탐색법이 대수적 탐색법에 비해 성능이 떨어지는 것으로 알려져 있다. 이는 아마도 계층적 탐색법의 경우 SAD를 계산하는데 사용되는 화소 개수가 충분하지 않기 때문인 것 같다.

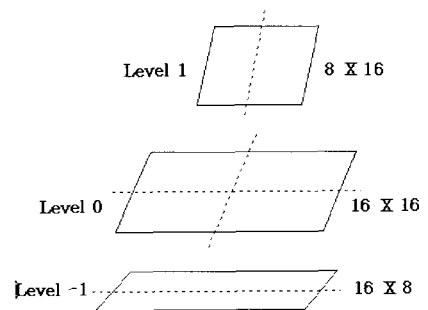
MPEG-4 visual core profile의 경우, 8x8 블록이 움직임 추정의 최소 단위이므로 전역 탐색법을 사용하지 않는 경우, 계산 량의 측면이나 움직임 추정오차의 측면에서 계층적 탐색법이 유용하다.

## 4. 제안된 계층적 움직임 벡터 추정 알고리즘의 변형(Metamorphosis Hierarchical Algorithm)

### 4.1 계층적 움직임 벡터 추정 알고리즘의 변형

변형 계층적 움직임 추정 방법은 기존 방법과 동일하게 우선, 현재 영상과 참조 영상의 저해상도 영상을 얻는다. 그러나 기존의 계층적 방법은 두 가지, 즉 휘도의 평균(mean intensity)을 통한 방법과 부표본을 추출(subsampling)하는 방법으로 구성하나 변형 방법은 상위 휘도 값과 하위 휘도 값(여기서 상위, 하위의 개념은 평균을 기준으로 한다.)을 이분 적으로 분리하여 매크로 블록을 구성한다. (그림 8)에서는 참조 영상에 대해 Level 1과 Level -1에서 이분 적으로 분리된 두 장의 저해상도 영상을 보이고 있다. 분명한 비교를 위하여 현재 영상에 대해서도 참조 영상과 유사하게 저해상도 영상들을 얻어야 한다.

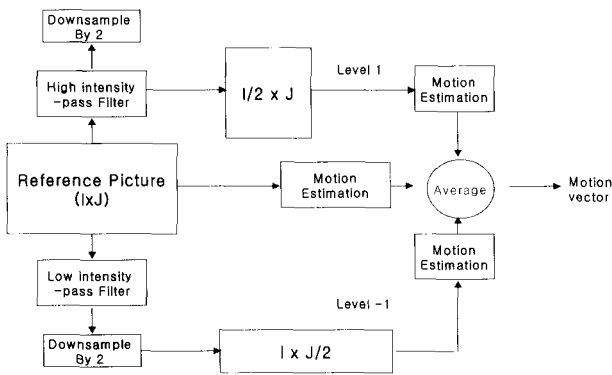
그런 다음 이전 이미지에서도 계층 구조상의 같은 단계에서 생성된 이미지에 대하여 일반적으로 전역 탐색을 실행한다. 전역 탐색 기법은 많은 계산 량을 필요로 하지만 계층적 탐색 기법에서는 해상도를 줄임으로써 많은 수의 비교 과정을 생략하여 최적의 정합을 위해 전역 탐색이 가능하게 한다. 최상위 단계에서는 전역 탐색을 실행할지라도 세밀한 움직임 추정이 이루어지지 못하기 때문에 최종 단계에서 적절한 움직임 벡터를 얻기 위해서는 상, 하 각 단계에서 초기의 움직임 추정이 세밀하게 이루어져야 한다. 이를 위한 다양한 접근법이 있는데 가장 일반적인 방법은 한정된 영역 내에서 앞서 기술된 휴리스틱탐색 기법을 적용해서 상, 하 단계에서의 최적의 정합 영역을 찾은 후에 원본 단계로 그 결과를 전파하는 것이다. 움직임 벡터는 상위 휘도들이 갖는 벡터 값과 하위 휘도들이 갖는 벡터 값의 평균을 더하는 방법으로 다음 단계에 전파된다.



(그림 8) 영상의 휘도에 따른 계층적 구조

### 4.2 변형 알고리즘(Metamorphosis Algorithm)

구체적 변형 계층적 움직임 벡터 추정 알고리즘의 내용은 (그림 9)에 잘 표현되어 있다. 먼저, 원본 영상 내에 (x, y)에 위치한 매크로 블록을 가정해보자. 이 매크로 블록의 Level 1과 Level -1에서 해당 매크로 블록은 각각 (x/2, y)과 (x, y/2)에 위치해 있다. 그리고 기존과 동일하게 Level 0에 해당 하는 매크로 블록을 16x16 탐색 영역을 [-m, +m]라고 하자.



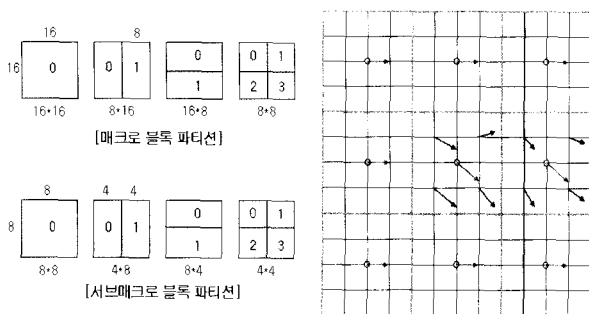
(그림 9) 변형 계층적 움직임 추정 방법

움직임 벡터의 탐색은 Level 1과 Level -1에서 동시에 시작한다. Level 1 단계에서는 매크로 블록이 8×16 크기를 Level -1 단계에서는 매크로 블록이 16×8 크기를 가지고 있다. 또한, 탐색 영역에 있어서도 변화가 있다. 즉 각각  $[-m/2, +m/2]$  탐색 1/2영역으로 감소하게 된다. 그래서 Level 1에서는  $[-m/2, +m/2]$ 의 탐색 영역과 8×16 매크로 블록에 대해 전역 탐색, 3-Step 탐색법등을 사용할 수 있다. 여기서 8×16 매크로 블록의 원점은  $(x/2, y)$ 이고 탐색 영역은 그 원점을 중심으로  $[-1, +1]$  화소의 범위이다. 현재 영상과 참조 영상을 비교하여 얻어진 움직임 벡터를  $(i_1, j_1)$ 이라고 하자.

Level -1에서는 16×8 매크로 블록에 대해 움직임 추정을 수행하게 되는데, 이때 움직임 추정의 원점은  $(x, y/2)$  이고 탐색 영역은 그 원점을 중심으로  $[-1, +1]$  화소의 범위이다. 앞 단계에서 설명한 것처럼, 본 단계에서도 전역 탐색법 뿐만 아니라 3-Step 탐색법등을 사용할 수 있다. 본 단계에서 얻어진 움직임 벡터를  $(i_{-1}, j_{-1})$ 라고 하자.

Level 0에서는 16×16 매크로 블록에 대해 움직임 추정을 수행하게 되는데, 이때 움직임 추정의 원점은  $(x+(i_1+i_{-1})/2, y+(j_1+j_{-1})/2)$ 이고 탐색 영역은 바로 전 단계에서처럼 그 원점을 중심으로  $[-1, +1]$  화소의 범위이다. 본 단계에서도 전역 탐색법 뿐만 아니라 다른 탐색 법을 사용할 수 있다. 본 단계에서 최소의 SAD를 가지는 움직임 벡터를 최종적으로 결정하게 된다.

본 논문에서 제안한 매크로 블록과 서브 매크로 블록 파티션 구성은 (그림 10)과 같다.



(그림 10) 계층적 움직임 추정 방법에 사용되는 블록 파티션 구조

### 4.3 변형 알고리즘 성능 분석

기존의 계층적 방법은 조작된 값, 즉 휘도의 평균값으로 비교 블록을 생성한다. 이는 원본 이미지의 휘도 특성을 왜곡 시킬 수 있다. 또한 휘도의 평균값이 동일한 영상 일지라도 각 휘도 간 편차는 다를 수 있기에(단일 영상에서 명암의 차이가 현저한 경우) 이를 고려하지 않고 있다. 변형 방법은 해상도를 낮춘 부표본 블록을 생성 하면서도 원본 영상의 데이터를 손상 시키지 않는 방법을 제시한다.

기존 계층적 방법에서 부표본(subsampling)을 만드는 것은 이미지의 크기를 줄이는 또 다른 접근법이다. 부표본을 만드는 동안에 여러 개의 화소들로 이루어진 블록은 그 화소들 중의 하나로 대체된다. 이 방법은 휘도의 평균값을 이용하는 방법보다 화질은 떨어지지만 처리 속도는 더 빠르다. 변형 계층적 방법은 화질이 떨어지지 않으면서 속도를 빠르게 할 수 있는 부표본 방법이다.

또한 기존 계층적 방법은 작은 물체를 포함하고 있는 영역에 대한 움직임 벡터 추정에 있어서 부정확성을 보이는 단점을 가지고 있다. 즉 이 탐색 범이 저해상도의 영상에서부터 출발하여 고해상도로 진행되는 구조를 가지고 있으므로 작은 물체의 경우 저해상도 영상에 반영되는 것이 힘들어 정확한 움직임 추정에 실패할 가능성이 높다. 반면, 이 탐색 범은 저해상도 영상에서부터 움직임 추정을 수행하므로 영상 내에 포함된 세밀한 잡음에 둔감한 장점을 가지고 있다. 그러나 계층적 탐색에서 발생하는 일반적인 문제는 높은 단계에서 오류가 생기기 쉬운 정합은 그 오류가 거의 항상 낮은 단계로 전파된다는 것이다. 최상위 단계의 낮은 해상도가 종종 모호한 결과를 유도하기 때문에 이것은 결코 드문 현상이 아니다. 즉 많은 후보 블록들이 유사한 블록 거리 측정값을 가질 것이고 실제적으로 정합된 블록은 다른 블록들에 비해 상위 단계에서 좀 더 적합할 것이다. 그러나 0단계에서 반드시 더 적합한 것은 아니다.

반면 변형 방법은 상하 단계에서 동시에 탐색이 진행되므로 상위 단계가 최종 단계에 주는 악 영향을 어느 정도 차단할 수 있고 단계의 깊이가 실질적으로는 2단계에 해당하므로 작은 물체의 경우에도 저해상도 영상에 반영할 수 있다. 즉 4×4 블록을 원본 영상으로 하여 탐색이 가능하다. 또한 16×8, 8×16, 8×4, 4×8 블록을 이용한 직사각형 매크로 블록 단위로도 계층적 움직임 추정이 가능하다.

다만 기존 계층적 탐색 범의 단점인 다른 해상도의 영상을 저장하기 위한 메모리가 요구되는 점을 배가 시킬 수 있다.

## 5. 성능평가 실험 및 응용

### 5.1 실험 방법

본 논문에서 제안한 변형 계층적 움직임 벡터 알고리즘의 성능을 평가하기 위해 실험에 사용한 영상은 (그림 11)이다. 이 영상은 비교적 질감의 정도가 중간 정도이며 밝기의 변화가 없는 폐쇄 영역을 부분적으로 많이 포함하고 있는 영상이다. (그림 11)의 영상의 크기는 320×240픽셀이며, 각각 좌우



(그림 11) 실험을 위한 스테레오 영상과 기준 시차지도 (a) Left Image (b) Right Original Image (c) Disparity Map

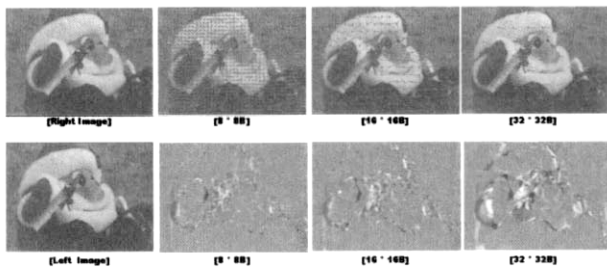
스테레오 영상과 계측된 시차지도를 나타내고 있다. 실험은 2.0 GHz Pentium4, 1024Mbyte RAM 시스템에서 하였다.

5.2 실험 결과

(그림 12)는 변형 계층적 움직임 추정에 사용된 블록 사이즈에 따른 오차 프레임을 보여주고 있다. 기존의 전역 탐색과 3단계 알고리즘 그리고 기존 계층적 방법과 변형에 대한 움직임 추정의 블록 사이즈에 따른 각각의 수행시간은 <표 1>에 나타냈다.

전역 탐색기법은 계산 비용을 고려하지 않아 많은 수행 시간을 필요로 한다. 또한 전역 탐색법이 계층적 탐색 법에 비해 성능이 떨어지는 것은 탐색하는 동안 지역 최소값에 빠질 확률이 크기 때문이다. 3단계 탐색기법은 선택된 블록에서만 영상이 이루어지기 때문에 계산 량은 적지만 비교적 효율이 낮다. 이에 비해서 계층적인 기법은 빠르고 효율적인 접근 방법으로서 현재와 이전, 두 이미지는 몇 가지 단계의 크기와 해상도로 줄어드는데 각 단계의 이미지는 동일한 이미지에 대하여 특정의 인수(일반적으로 2)에 의해서 줄어든 해상도를 갖는다.

또한, 계층적 탐색 기법에서는 해상도를 줄임으로서 많은 수의 비교 과정을 생략하여 최적의 정합을 위해 전역 탐색을 가능하게 한다. 움직임 벡터의 필드의 계층적인 탐색은 움직임 추정에 널리 적용되는데 움직임 탐색 점의 개수와 SAD의 계산에 고려되는 화소의 개수를 함께 감소시켜 계산 량을 감소시킨다. 또한 낮은 복잡도와 높은 효율을 제공하며 이 두 가지 기능 사이에서 균형을 유지하는데 있어서 큰 유연성을 보장한다.



(그림 12) 블록 크기에 따른 오차 프레임

<표 1> 탐색 방법에 따른 움직임 벡터 수행시간

| 탐색 방법 | 전역 탐색      | 3단계 탐색     | 계층 (8x8)   | 계층 (16x16) | 변형계층 (16x16) | 계층 (32x32) |
|-------|------------|------------|------------|------------|--------------|------------|
| 수행 시간 | 1.5000 sec | 0.5000 sec | 0.2350 sec | 0.1250 sec | 0.1220 sec   | 0.1090 sec |

계산량 측면에서 계층적 탐색 법은 매우 효과적이며, 저해상도 영상에서부터 움직임 추정을 수행하므로 영상 내에 포함된 세밀한 잡음에 둔감한 장점이 있다. 계층적인 움직임 추정은 매우 낮은 계산 비용과 더불어 높은 효율을 제공하고, 그 다양성뿐만 아니라 블록크기, 단계의 수, 크기 조정 인수 등의 특정한 많은 매개변수들 때문에 스테레오 비전의 움직임 추정에 적합한 기법이다.

계층적 탐색 법은 저 대역 통과 필터링의 효과로 인하여 지역 최소값에 빠질 가능성이 줄어든다. 이러한 결과는 16x16 매크로 블록 또는 8x8 블록에 대한 움직임 추정에 대해 대체적으로 성립한다.

변형 계층적 탐색 법은 기존 방법의 한 접근법인 부표본의 일종이기에 속도에서 약간의 우월성을 보였다. 휘도의 평균을 이용하는 계층적 방법보다는 많은 시간 단축을, 부표본을 이용하는 계층적 방법 보다는 정확도가 높아졌다. 그러나 변형 방법의 주된 목적은 속도를 유지하면서도 화질을 높이고자 하는 알고리즘이기에 앞으로 다양한 영상에 대한 추정 속도 분석이 필요하고, 최악의 상황 등 여러 여건에서 화질의 측정이 요구된다.

5.3 다중영상 전송 및 모니터링 시스템의 구축

스테레오 카메라에서 취득된 영상을, 기존의 H.264 방식에서 사용하는 3단계 알고리즘과 이를 개선한 계층적인 움직임 추정 알고리즘을 서브 단계에서 실험적으로 각각 행하고 전체적으로는 제안한 변형 알고리즘을 통하여 영상을 압축하고 전송할 수 있는 시스템을 구현하였다. 영상의 전송을 위한 압축과 해독 시스템 개요도는 (그림 13)과 같다.

전체적으로 구현된 시스템의 영상압축에 관해서는 H.264 방식을 취하였다. 영상압축의 표준안을 따른다는 것은 인코더의 경우에 인코더가 압축 비트율과 같은 제한 사항을 만족하는 동시에, 생성하는 비트스트림이 표준안에 정의된 체계(syntax)의 요구 조건을 만족시키고 표준안에 설명된 디코딩 과정을 사용하여 디코딩될 수 있다는 것을 의미한다. 디코더의 경우에 표준안을 따른다는 것은, 정의된 제한 사항 및 정확도를 만족시키면서 표준안을 따르는 비트스트림을 디코딩

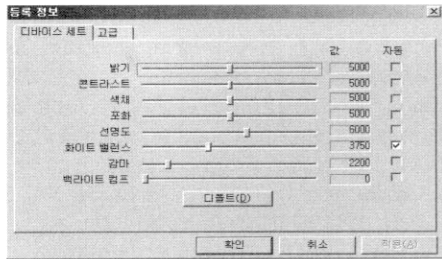


(그림 13) 영상의 Encoder/Decoder

<표 2> 시스템의 주요 기능

| 시스템의 주요 기능   | 비고        |
|--------------|-----------|
| 카메라 컨트롤러     |           |
| 다중카메라의 영상 입력 | 송신측       |
| 영상 Encoding  | H.264를 이용 |
| 영상 전송 시스템    | 홈 네트워크    |
| 영상 Decoding  | H.264를 이용 |
| 모니터링         | 수신측       |
| 베신저          |           |





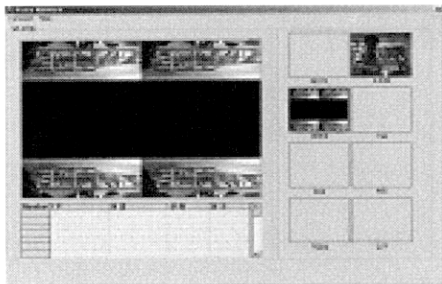
(그림 14) Camera Control Parameter



(그림 15) 다중 영상의 전송

<표 3> 모니터링 시스템의 검색 기능 항목 예

| Number | IP          | 이름 | 상태 | 참고    |
|--------|-------------|----|----|-------|
| 1      | 59.5.202.95 | 현관 | 정상 | 11:50 |



(그림 16) 다중 영상의 모니터링

할 수 있다는 것을 의미한다.

다중영상 전송 및 모니터링 시스템의 주요 기능은 <표 2>와 같다.

(그림 14)는 카메라 컨트롤러를 통한 메타 설정 화면이며 (그림 15)는 송신측의 입력 영상을 보여주고 있다.

모니터링 시스템 명은 Home Network이며 컨트롤 화면의 메뉴는 Connect, Help로 구성 되어 있다. 최고 8개의 화면 모니터링이 가능하다. 모니터링 시 확인 가능 항목은 <표 3>과 같다.

구현된 시스템의 결과는 예제 화면에서 볼 수 있다. (그림 16)은 모니터링 되는 영상을 보여주고 있다. 실시간 영상 감시와 여러 발생 시 알림기능, 보고서 제공, 메신저 연결 기능이 있다.

## 6. 결 론

다양한 종류의 컴퓨터, 상이한 성격의 데이터, 이를 활용하는 이기종 장치들 그리고 기기와의 친숙도 차이를 보이는 많

은 이용자들이 한정된 홈 네트워크 자원을 효율적으로 분배하고 이것들이 다시 유무선 인터넷 환경과 접목되어 포괄적 커뮤니티를 형성하는 환경이 유비쿼터스(Ubiquitous) 환경이다.

유비쿼터스를 위한 하드웨어적, 소프트웨어적 기술에 대한 연구는 다방면에서 진행되고 있으며 제한된 자원을 효율적으로 이용하고자 하는 연구도 활발하다.

따라서, 본 논문에서는 홈 네트워크 상황에서 콘텐츠의 중심이 되는 영상 데이터의 전송과 처리 그리고 제어에 관한 탐색을 하였다.

각도, 거리등 다양한 환경에서 전송되어지는 스테레오 카메라의 영상 데이터들은 전송되어지기 전에 축소, 확대, 이동, 보정 등 효율성 제고를 위한 전처리 과정이 요구 되었다. 또한 두 영상의 중첩된 영역과 중첩되지 않는 부분에 대한 압축 시 다른 방법도 필요하였다.

중첩된 영역에 대한 압축을 위하여 기존 움직임 보상을 위한 모션 벡터 추정 알고리즘인 전역 탐색, 3단계 탐색, 그리고 계층적 움직임 벡터 추정 방법을 이용하여 실험 한 결과, 계층적 움직임 벡터 추정 알고리즘이 전역 탐색 방법과 3단계 탐색 방법에 비하여 정확한 대응점을 찾는데 있어서 불필요한 픽셀 값들의 계산을 피할 수 있었고 전체적인 처리 시간을 단축시킬 수 있었다.

그러나 기존 계층적 추정 알고리즘은 작은 물체를 포함하고 있는 영역에 대하여 부정확성을 보였고 저 해상도에서 얻는 이점 못지않게 모호한 결과를 유도하여 최종단계에 전파되는 모순을 도출 하였다.

그래서 본 논문에서는 기존 계층 방법의 한 접근 방법인 부표본 방법을 변형 제안하여 부표본 방법이 갖는 속도의 이점을 살리면서도 화질이 떨어지지 않도록 하였다. 기존 계층 방법이 휘도의 평균을 이용한 두 단계의 상위 단계로 구성되는 반면 변형된 방법은 상위 휘도와 하위 휘도로 구성된 같은 수준의 두 단계를 통해 적은 해상도의 비교 블록을 통한 움직임을 추정하였다. 이는 휘도의 평균값이라는 조작된 데이터에 의한 비교가 아닌 원본에 속한 데이터를 비교함으로써 정확도를 높이고 소형, 직사각형 블록에서도 추정이 가능함을 보였다. 결과적으로, 휘도의 평균값을 이용하는 계층적 방법보다는 많은 시간 단축을, 부표본을 이용하는 계층적 방법 보다는 정확도가 높아졌다. 다만 기존 방법이 갖는 부표본 저장 공간 문제를 배가 시키는 문제점도 도출 되었다.

향후 휘도의 편차가 큰 경우와 같은 다양한 악조건 영상에 대한 변형 알고리즘의 실험, 검증을 시행할 예정이며, 도출된 저장 공간 문제를 개선할 수 있는 방법과 휘도의 성분을 더욱 특성화할 수 있는 방안에 대하여 연구를 계속할 생각이다. 또한 처리속도를 더욱 단축시키는 움직임 추정 모듈의 최적화 방법에 대한 연구를 계속 진행할 계획이다.

최종적으로, 본 논문에서는 다차원 영상의 효율적인 압축을 위하여 스테레오 영상 압축의 전처리를 설계하고 기존의 H.264 방식에서 사용하는 움직임 보상을 위한 모션 벡터 추정 알고리즘을 개선한, 계층적인 움직임 벡터 추정 방법을 변형 적용하여 영상을 압축하고 전송하는 다중 영상의 전송 및 모니터링 시스템을 구현했다. 본 논문에서 제시한 변형

계층적 움직임 벡터 추정 방법은 유비쿼터스 환경에서 다양하게 적용이 가능할 것으로 생각된다.

**참 고 문 헌**

[1] <http://www.vqeg.org/> (Video Quality Experts Group).  
 [2] Iain E.G. Richardson, H.264 and MPEG-4 [조중휘·손요환, 차세대 영상압축기술(홍릉과학출판사, 2004)].  
 [3] D. Geiger, B. Ladendorf, and A. Yuille, "Occlusions and binocular stereo", Intl. Journal of Computer Vision, Vol.14, No.3, pp.211-226, 1995.  
 [4] ISO/IEC 14496 - 10 and ITU - T Rec. H.264, Advanced Video Coding, 2003.  
 [5] F.Pereira and T.Ebrahimi(eds), The MPEG -4 Book, IMSC Press, 2002.  
 [6] K.T. Tan and M. Ghanbari, A multi-metric objective picture quality measurement model for MPEG video, IEEE Trans. Circuits and Systems for Video Technology, 10(7), October, 2000.  
 [7] ISO/IEC 14495-1:2000 information technology - lossless and near-loss compression of continuous-tone still image : Baseline, (JPEG-LS).  
 [8] ITU-T Recommendation, Information technology - coded representation of picture and audio information - progressive bi-level image compression, T.82(JBIG).  
 [9] D. W. Kim, J. S. Choi and J. T. Kim, "Adaptive motion estimation based on spatiotemporal correlation," Signal Processing: Image Commun., Vol.13, pp.161-170, 1998.  
 [10] D.Tzovaras, N.Grammalidis and M.G.Strintzis, "Object-Based Coding of Stereo Image Sequences using Joint 3-D Motion/Disparity Compensation," IEEE Trans. on Video Technology, Vol.7, No.2, pp.312-328, April, 1997.  
 [11] MPEG-4 Industry Forum, <http://www.m4if.org>  
 [12] ITU-T Recommendation H.263, Video coding for low bit rate communication, Version2, 1998.  
 [13] J. C. Tsai, C. H. Hseigh, S. K. Weng, and M. F. Lai, "Block-matching motion estimation using correlation search algorithm," Signal Processing: Image Commun., Vol.13, pp.119-133, 1998.  
 [14] L. K. Liu and E. Feig, "A block-based gradient descent search algorithm for block motion estimation in video coding", IEEE Trans. on Circuits Syst. Video Technol., Vol.6, pp.419-423, 1996.  
 [15] M. Brunig, W. Niehsen, "Fast full-search block matching," IEEE Trans on Circuits and Systems for video Technology, Vol.11, No.2, pp.241-247, Feb., 2001.  
 [16] A. Ahmad, N. Khan, S. Masud, and M. A. Maud, "Selection of variable block sizes in H.264," IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '04), Vol.3, pp.173-176, May, 2004.  
 [17] W.Li and E. Salari, "Successive elimination algorithm for motion estimation," IEEE Trans. on Image Processing, Vol.4, No.1, pp.105-107, 1995  
 [18] X. Q. Gao, C. J. Duanmu, and C. R. Zou, "A multilevel successive elimination algorithm for block matching motion estimation," IEEE Trans. on Image Processing, Vol.9, No.3, pp.501-504, 2000.  
 [19] J. Y. Lu, K. S. Wu and J. C. Lin, "Fast full search in motion estimation by hierarchical use of Minkowski's inequality," Pattern Recognition, Vol.31, No.7, pp.945-952, 1998.  
 [20] L.M. Po and W. C. Ma, "A novel four-step search algorithm for fast block motion estimation," IEEE Trans. on Circuits

Syst. Video Technol., Vol.6, pp.313-317, 1996.  
 [21] M. Ghanbari, The cross-search algorithm for motion estimation, IEEE Trans. Commun., 38, July, 1990.  
 [22] P. Kuhn, G. Diebel, S. Hermann, A. Keil, H. Mosshofer, A. Kup, R. Mayer and W.Stechele, Complexity and PSNR-Comparison of Several Fast Motion Estimation Algorithms for MPEG-4, Proc. Applications of Digital Image Processing XXI, San Diego, 21-24, July, 1998; SPIE, 3460, pp.486-499.  
 [23] J. N. Kim and T. S. Choi, "A fast full-search motion- estimation algorithm using representative pixels and adaptive matching scan," IEEE Trans. on CSVT, Vol.10, No.7, pp.1040-1048, 2000.  
 [24] Z. Zhou, M. T. Sun, and Y. F. Hsu, "Fast variable block-size motion estimation algorithm based on merge and slit procedures for H.264 / MPEG-4 AVC," International Symposium on Circuits and Systems, Vol.3, pp.725-728, May, 2004.  
 [25] Fitzek FHP, Reisslein M, "MPEG-4 and H.263 video traces for network performance evaluation," IEEE Network, Vol.15 No.6, pp.40-54, 2001.  
 [26] Thomas Wiegand, Gray J Sullivan, and Ajay Luthra, "Overview of the H.264/AVC Video coding Standard," IEEE Trans. Circuits Syst. Video Technol, July, 2003.



**김 정 응**

1998년 국민대학교 교육대학원(전자계산 교육학석사)  
 2005년 호서대학교 벤처전문대학원 컴퓨터응용기술학과(공학박사)  
 2001년~2005년 국민대, 세종대, 강남대, 중부대, 서일대, 한림성심대 강사

2001년~2003년 경북대학 겸임교수  
 2006년~현재 서울벤처정보대학원대학교 겸임교수  
 관심분야 : 소프트웨어공학(특히, 게임 S/W 품질 보증과 평가, 감리, 객체지향 프로그래밍 분석 및 설계, 컴포넌트 기반 S/W), 유비쿼터스(영상처리, RFID, IPv6), 웹 어플리케이션



**양 해 술**

1975년 홍익대학교 전기공학과(학사)  
 1878년 성균관대학교 정보처리학과(석사)  
 1991년 日本 오사카대학 기초공학과 정보공학과 소프트웨어공학 (공학박사)  
 1975년~1979년 육군중앙경리단 전자계산실 시스템분석장교

1980년~1995년 강원대학교 전자계산학과 교수  
 1986년~1987년 日本 오사카대학교 객원연구원  
 1994년~1995년 한국정보처리학회 논문편집위원장  
 1995년~2002년 한국S/W품질연구소 소장  
 2001년~현재 한국정보처리학회 부회장  
 1999년~현재 호서대학교 벤처전문대학원 교수  
 2005년~현재 호서학원 서울벤처정보대학원대학교 교무처장  
 관심분야 : 소프트웨어공학(특히, S/W 품질보증과 품질평가, 품질 감리, 품질컨설팅, OOA/ OOD/OOP, CASE, SD, S/W 프로젝트관리, 컴포넌트 기반 개발방법론과 품질평가