탐색 영역과 특징의 가중치를 동적으로 조절하는 활동적 블록 정합 알고리듬

(An Active Block Matching Algorithm by Adapts Search Area and Weights of Features Dynamically)

장 석 우 † 최 형 일 ††
(Seok-Woo Jang) (Hyung-Il Choi)

요 약 본 논문에서는 탐색 영역과 특징의 가중치를 동적으로 조절하여 블록 단위의 움직임 벡터를 추출하는 활동적 블록 정합 알고리듬을 제안한다. 본 논문에서 제안하는 알고리듬은 탐색 영역의 중심 위치를 결정하기 위해 시간에 따른 블록의 동작 변화는 작다고 가정한다. 그리고 탐색 영역의 크기는 공간적으로 인접한 블록들의 신뢰도에 따라 조절된다. 또한 본 논문에서 제안하는 알고리듬은 다중 특징을 사용하는 블록 정합 알고리듬으로 블록 정합 시 특징의 기여 정도를 나타내는 가중치를 블록 안에서 각 특징이 가지는 구분력에 따라 자동으로 설정하는 정합 유사 함수를 사용한다. 실험 결과는 본 논문에서 제안한 블록 정합 알고리듬이 기존의 알고리듬 보다 정확하게 움직임 벡터를 추출함을 보여준다.

Abstract We propose an active block matching algorithm for motion estimation. The proposed algorithm dynamically determines the search area and the matching metric. We exploit the constraint of small velocity changes of a block along the time to determine the origin of the search area. The range of the search area is adjusted according to the motion coherency of spatially neighboring blocks. Our matching metric includes multiple features. The degree of overall match is computed as the weighted sum of matches of individual features. We adjust the weights depending on the distinctiveness of features in a block, so that we may discriminate features according to the characteristics of an involved block. The experimental results show that the proposed algorithm can yield very accurate block motion vectors.

1. 서 론

영상 매체 및 멀티미디어 기술의 발전에 따라 동영상의 분석에 관한 연구에 많은 관심이 집중되고 있다. 특히, 연속적으로 입력되는 동영상을 분석하여 물체나 배경의 이동 및 동작에 관한 정보를 추론하는 동작 이해에 관한 연구가 컴퓨터 비전 분야를 중심으로 활발히 진행되고 있다. 동작은 일반적으로 시각적인 정보를 얻기 위해 사용될 수 있는 중요하고 기본적인 원천으로 영상의 동작 패턴은 주위 환경의 3차원적인 구조 및 카메라와 물체 사이의 상대적인 동작을 결정하는데 유용한 정보를 포함하고 있다. 그러나 영상으로부터 정확한 움직임을 분석하는 작업은 매우 어려운 문제로 인식되고 있다[1].

영상으로부터 움직임을 분석하기 위해 전통적으로 많이 사용하는 방법은 크게 화소 단위의 움직임 벡터를 추출하는 광류(optical flow) 측정 방법과 블록 단위의 움직임 벡터를 추출하는 블록 정합 알고리듬(BMA : block matching algorithm)을 이용한 방법이 있다. 화소 단위의 움직임 벡터를 추출하는 방법은 정확도가 높고 종류가 다양하지만 영상의 전체 범위에 걸쳐 화소 단위의 복잡한 연산을 수행하므로 병렬 처리 환경이 아닐 경우 처리 시간이 오래 걸린다. 따라서 처리 시간을 줄인 블록 단위의 움직임 벡터를 추출하는 블록 정합 알고리듬을 많이 사용하고 있다.

· 본 논문은 두뇌한국21 사업 중 핵심분야 사업(E-0075)과 첨단정보기술연구센터를 통하여 과학재단의 일부 지원을 받았음.
† 비 회 원 숭실대학교 생산기술연구소 연구원
swjang@vision.soongsil.ac.kr
†† 종신회원 : 숭실대학교 미디어학부 교수
hic@computing.soongsil.ac.kr
논문접수 : 2000년 1월 5일
심사완료 : 2000년 10월 12일

블록 단위의 움직임 벡터를 추출하기 위해 일반적으로 사용되는 알고리듬은 전역 탐색 블록 정합 알고리듬(FSBMA : full search BMA)이다[2]. 이 알고리듬은 영상을 모양과 크기가 동일한 사각형의 블록으로 분할한 후 블록 단위의 특징을 정의하고 정의된 특징에 의해 구성되는 정합 척도를 일정한 탐색 영역 내의 블록들에 적용하여 가장 높은 정합 척도를 가지는 블록을 찾는다. 그러나 전역 탐색 블록 정합 알고리듬은 탐색 영역의 내부 전체를 탐색하면서 가능한 모든 블록들에 대한 정합을 수행하므로 여전히 처리 시간이 오래 걸린다. 따라서 처리 시간을 줄이기 위해 3단계 탐색에 의한 방법(TSS : three step search), 새로운 3단계 탐색에 의한 방법(NTSS : new three step search), 계층적인 1차원 탐색에 의한 방법, 4단계 탐색에 의한 방법(FSS : four step search)들이 제안되었다[3][4][5][6]. 그러나 이런 방법들의 공통적인 문제점은 대부분 명암이라는 하나의 특징만을 이용하여 블록 정합을 수행하므로 영상 안에 포함된 다른 유력한 특징들을 고려하지 않는다. 따라서 불충분한 특징을 사용하여 정합을 수행하므로 정합의 정확도가 저하되고 부정확한 정합으로 인해 잡음이 포함된 움직임 벡터를 추출할 확률이 높다. 또한, 시간적인 복잡도만을 줄이는 데 치중한 나머지 알고리듬 내부의 문제인 국부적 최소(local minimum)에 빠지는 경향이 있다.

본 논문에서는 이런 문제점을 해결하기 위해 영상 내의 여러 가지 특징들을 동시에 고려하여 블록 정합을 수행하면서 탐색 영역을 동적으로 변화시키는 활동적 블록 정합 알고리듬을 제안한다. 본 논문에서 제안하는 블록 정합 알고리듬은 칼라 특징, 에지 특징, 텍스쳐 특징과 같은 다중 특징을 사용하는 알고리듬으로 블록 정합 시 각 특징의 기여도를 나타내는 가중치(weighting factor)를 자동으로 조절할 수 있도록 정합 유사 함수를 정의함으로써 영상 내의 특징을 보다 정확히 고려할 수 있다. 그리고 이전 시점에서 구한 움직임 벡터의 신뢰도(certainty factor)를 기반으로 현 시점의 탐색 영역을 동적으로 조절한다.

그림 1은 본 논문에서 제안하는 활동적 블록 정합 알고리듬의 전체적인 구조를 보여준다. 그림에서 보는 바와 같이 활동적 블록 정합 알고리듬은 크게 특징 추출 단계, 탐색 영역 설정 단계, 정합 유사 함수 정의 단계의 세 단계로 구성된다. 특징 추출 단계에서는 연속적으로 입력되는 영상을 모양과 크기가 동일한 사각형의 블록으로 분할하고 블록 단위의 특징, 즉 칼라 특징, 에지 특징, 텍스쳐 특징을 추출한다. 이런 특징들은 블록 사이의 유사도를 정의하는 정합 유사 함수 정의 단계에서 사용된다.

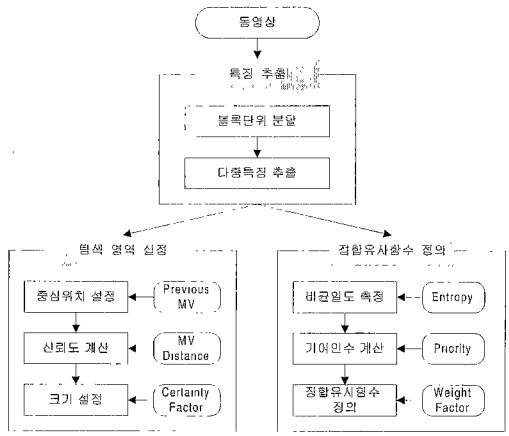


그림 1 활동적 블록 정합 알고리듬의 전체 구조

탐색 영역을 설정하는 단계에서는 블록 정합에 사용할 탐색 영역을 동적으로 설정한다. 탐색 영역의 중심 위치는 블록 정합 초기에 분할된 블록의 중심 위치와 이전 시점의 블록에서 추출한 움직임 벡터를 이용하여 정의되며, 탐색 영역의 크기는 이전 시점에서 추출한 움직임 벡터들과 이들의 신뢰도를 기반으로 설정된다. 그리고 탐색 영역의 크기는 움직임 벡터의 신뢰도에 따라 연속적으로 변화하며, 움직임 벡터의 신뢰도는 탐색 영역의 크기를 동적으로 변화시키는 가중치의 역할을 하도록 정의된다.

정합 유사 함수를 정의하는 단계에서는 각 블록에 대해 설정된 탐색 영역 내의 특징값들의 분포를 고려하여 정합 유사 함수를 정의한다. 정합 유사 함수는 칼라 특징, 에지 특징, 텍스쳐 특징과 같은 다수의 특징으로 정의되며 각 특징의 기여 정도를 나타내는 기여 인수가 가중치의 형태로 적용된다. 정합 유사 함수를 구성하는 특징들은 정합의 기준이 되는 블록과 정합 대상이 되는 탐색 영역에 속한 블록 사이의 특징값들의 차이로 정의된다. 또한, 특징의 기여 정도를 나타내는 가중치는 엔트로피 개념을 이용하여 탐색 영역 내의 특징값들의 비균일도(heterogeneity)를 나타내도록 정의된다. 따라서 본 논문에서는 고정된 가중치를 사용하는 기존의 방법과는 달리 가중치를 동적으로 조절할 수 있으므로 영상의 특징들을 보다 효과적으로 반영한 블록 정합이 가능하다.

1장에서는 연구 동기와 활동적 블록 정합 알고리듬의 전체적인 개요를 설명하였다. 2장에서는 블록 정합에 사

용되는 특징들에 대해 설명하고, 3장에서는 탐색 영역을 동적으로 조절하는 방법에 대해 설명하며, 4장에서는 정합 유사 함수를 정의하는 방법에 대해 설명한다. 그리고 5장에서는 실험 결과 및 결론을 기술한다.

2. 블록 정합에 사용되는 특징

본 장에서는 블록 정합에 사용되는 특징들에 대해 설명한다. 본 논문에서는 블록 단위의 정합을 수행하기 위해 칼라 특징, 에지 특징, 텍스쳐 특징과 같은 3가지의 특징들을 사용한다. 이들 특징은 블록 단위로 계산되며 각 특징의 기여도를 나타내는 가중치와 함께 정합 유사 함수를 정의하게 된다.

2.1 칼라 특징

일반적으로, 칼라 영상을 표현하기 위해 RGB 칼라 공간을 많이 사용하지만 영상 처리를 수행하려 할 때 세 가지의 R, G, B 칼라값을 모두 처리해야 하므로 영상 처리에 비효율적이다. 따라서 본 논문에서는 조명의 변화에 강건한 칼라 특징을 추출하기 위해 YIQ 칼라 공간을 사용한다. YIQ 칼라 공간은 NTSC에서 정한 합성 칼라 비디오 표준 방식으로 밝기값의 변화에 대해 색상이나 채도가 변하지 않는다는 특징을 가지고 있다.

YIQ 칼라 공간은 YUV 칼라 공간에서 유도된 것으로 Y(Luminance) 요소는 밝기값을 나타내고, I (Inphase) 요소와 Q(Quadrature) 요소는 두 요소를 합성하여 색상과 채도를 나타낸다. 식 (1)은 I, Q 요소와 색상, 채도와의 관계를 나타낸다.

$$\text{색상} = \tan^{-1}\left(\frac{Q}{I}\right) \qquad (1)$$
$$\text{채도} = \sqrt{I^2+Q^2}$$

식 (2)는 RGB 칼라 공간을 YIQ 칼라 공간으로 변환하는 식을 나타낸다[7].

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.30 & 0.59 & 0.11 \\ 0.60 & -0.27 & -0.32 \\ 0.21 & -0.52 & 0.31 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \qquad (2)$$

본 논문에서는 YIQ 칼라 공간에서 밝기값을 나타내는 Y 요소를 제외한 I와 Q 요소만을 이용하여 조명의 변화에 강건한 칼라 특징을 추출한다. 본 논문에서 사용하는 칼라 특징은 블록에 포함되는 영상 화소들의 I와 Q 요소의 평균과 표준편차를 이용하여 정의된다. 따라서 본 논문에서 사용하는 칼라 특징은 식 (3)과 같이 정의된다.

$$C = [\mu_I,\ \sigma_I,\ \mu_Q,\ \sigma_Q] \qquad (3)$$

식 (4)와 식 (5)는 영상 좌표 (x,y)를 시작 위치로 하는 블록에 포함되는 영상 화소들의 I와 Q 요소의 평균과 표준 편차를 구하는 식을 나타낸다. 아래의 식에서 N은 블록의 크기, 즉 블록의 너비 또는 높이를 나타낸다.

$$\mu_I = \frac{1}{N^2}\sum_{i=0}^{N-1}\sum_{j=0}^{N-1} I(x+i, y+j) \qquad (4)$$
$$\mu_Q = \frac{1}{N^2}\cdot\sum_{i=0}^{N-1}\sum_{j=0}^{N-1} Q(x+i, y+j)$$

$$\sigma_I = \sqrt{\frac{1}{N^2}\cdot\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}\{I(x+i,y+j)-\mu_I\}^2} \qquad (5)$$
$$\sigma_Q = \sqrt{\frac{1}{N^2}\cdot\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}\{Q(x+i,y+j)-\mu_Q\}^2}$$

이와 같이 계산되는 칼라 특징은 정합 유사 함수를 구성하는 특징으로 사용되고, 탐색 공간 내에서 칼라 특징의 비균일도를 계산하기 위해 사용된다.

2.2 에지 특징

본 논문에서는 블록을 구성하는 특징 중의 하나로 화소 단위가 아닌 블록 단위의 에지 특징을 사용한다. 블록 단위의 에지 특징을 구하는 과정은 각 블록의 평균 밝기값을 계산하고 이 블록을 중심으로 하는 인접 블록들의 평균 밝기값을 계산한 후 라플라시안(Laplacian) 에지 연산자를 이용하여 블록의 에지 특징을 계산한다. 밝기값의 계산은 YIQ 칼라 공간에서 밝기값을 나타내는 Y 요소를 이용한다. 그림 2는 i번째 블록 B_i^{xy} 및 인접 블록들을 나타낸다.

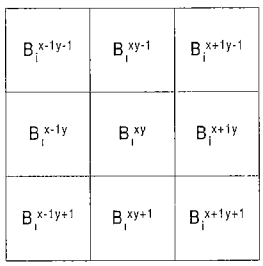


그림 2 에지 특징의 계산

블록 B_i^{xy}에 대한 에지 특징은 식 (6)과 같이 계산된다.

$$E = -\mu_Y(B_i^{x-1y-1}) - \mu_Y(B_i^{xy-1}) - \mu_Y(B_i^{x+1y-1}) \qquad (6)$$
$$-\mu_Y(B_i^{x-1y}) + 8\times\mu_Y(B_i^{xy}) - \mu_Y(B_i^{x+1y})$$
$$-\mu_Y(B_i^{x-1y+1}) - \mu_Y^{xy+1} - \mu_Y(B_i^{x+1y+1})$$
$$\mu_Y(B_i^{xy}) = \frac{1}{N^2}\cdot\sum_{i=0}^{N-1}\sum_{j=0}^{N-1} Y(x+i, y+j)$$

식 (6)에서 $\mu_Y(B_i^{xy})$는 블록 B_i^{xy}의 평균 밝기값을 나타낸다. 이와 같이 계산되는 에지 특징은 칼라 특징과 함께 정합 유사 함수를 구성하는 특징으로 사용되며 에지 특징의 비균일도를 계산하기 위해 사용된다.

2.3 텍스쳐 특징

텍스쳐 특징은 칼라 특징 및 에지 특징과 함께 영상에서 화소값들의 분포를 특징짓는 주요한 요소로 활용되고 있다. 텍스쳐 특징을 추출하기 위해 사용되는 텍스쳐 분석 알고리듬은 랜덤 필드 모델(random field model)에서부터 웨이블릿 변환 기술들에 이르기까지 다양하게 연구되어 왔는데 최근에는 가보 웨이블릿 변환(Gabor wavelet transform)을 이용하는 가보 특징이 많이 사용되고 있다.

본 논문에서도 가보 웨이블릿 변환을 수행하여 얻은 가보 웨이블릿 인수(Gabor wavelet coefficients)를 텍스쳐 특징으로 사용한다. 가보 웨이브릿 인수는 가보 웨이블릿 커널(Gabor wavelet kernel)을 영상에 적용(convolution)하여 구해지는데, 이는 영상의 텍스쳐 정보를 잘 반영하는 특징으로서 기존의 다른 텍스쳐 특징들보다 뛰어난 텍스쳐 구분력을 가진 것으로 알려져 텍스쳐에 기반한 영상 코딩, 영상 인식, 영상 데이타베이스 검색 등에서 널리 사용되고 있다[8].

가보 웨이블릿 커널은 기본적으로 사인 곡선(sinusoidal wave or plane wave)에 가우시안 함수(Gaussian envelop)를 덮어 씌운 형태로서 모함수(mother wavelet)를 적절히 확대(dilation)또는 회전(rotation)시켜 다양한 형태의 커널을 생성한다. 따라서 가보 웨이블릿 커널은 방향과 크기를 나타내는 인수에 의해 식 (7)과 같이 정의된다.

$$\psi_{mn}(\boldsymbol{x}) = \frac{\|\boldsymbol{k}_{m\theta}\|^2}{\sigma^2} \cdot \exp\left\{-\frac{\|\boldsymbol{k}_{m\theta}\|^2 \cdot \|\boldsymbol{x}\|^2}{2\sigma^2}\right\} \cdot \left\{\exp\{i\,\boldsymbol{k}_{m\theta} \cdot \boldsymbol{x}\} - \exp\left\{-\frac{\sigma^2}{2}\right\}\right\} \quad (7)$$

$$\begin{cases} \exp\left\{-\frac{\|\boldsymbol{k}_{m\theta}\|^2 \cdot \|\boldsymbol{x}\|^2}{2\sigma^2}\right\} : \textit{Gaussian envelope function restricting the plane wave} \\ \exp\{i\,\boldsymbol{k}_{m\theta} \cdot \boldsymbol{x}\} : \textit{complex-valued plane wave} \\ \exp\left\{-\frac{\sigma^2}{2}\right\} : \textit{to make kernel DC free} \\ \frac{\|\boldsymbol{k}_{m\theta}\|^2}{\sigma^2} : \textit{to compensate for frequency-dependent decrease of power spectrum} \end{cases}$$

식 (7)에서 $\boldsymbol{x}$는 좌표의 위치 (x,y)를 나타내는 벡터이고, $\boldsymbol{k}_{m\theta}$는 커널의 크기 및 방향을 나타내는 색인 벡터(index vector)이다. 각 커널의 크기 및 방향을 나타내는 색인 벡터는 식 (8)과 같이 정의된다.

$$\boldsymbol{k}_{m\theta} = (k_m \cdot \cos k_\theta,\ k_m \cdot \sin k_\theta) \quad (8)$$

where

$$\begin{cases} k_m = \frac{\pi}{2} \cdot 2^{-\frac{m}{2}} & \textit{for } m = 0, 1, 2, \ldots, M \\ k_\theta = \frac{\theta \cdot \pi}{N} & \textit{for } \theta = 0, 1, 2, \ldots, N-1 \end{cases}$$

크기 색인 m은 사인 곡선의 빈도수(frequency)를 결정하고 방향 색인 θ는 사인 곡선의 방향을 결정한다. 크기 색인은 m이 클수록 낮은 빈도수를 결정한다. 식 (8)의 색인 벡터를 이용하여 식 (7)의 벡터 $\boldsymbol{k}_{m\theta}$와 $\boldsymbol{x}$를 구성 요소들로 풀어서 정리하면 커널 $\psi_{m\theta}(\boldsymbol{x})$는 식 (9)와 같이 표현된다.

$$\begin{aligned} \psi_{m\theta}(x,y) &= \textit{real part} + i \cdot \textit{imaginary part} \quad (9) \\ &= \frac{(k_m \cos k_\theta)^2 + (k_m \sin k_\theta)^2}{\sigma^2} \\ &\quad \cdot \exp\left\{-\frac{((k_m \cos k_\theta)^2 + (k_m \sin k_\theta)^2) \cdot (x^2+y^2)}{2\sigma^2}\right\} \\ &\quad \cdot \left\{\cos(xk_m \cos k_\theta + yk_m \sin k_\theta) + i \cdot \sin(xk_m \cos k_\theta + yk_m \sin k_\theta) - \exp\left\{-\frac{\sigma^2}{2}\right\}\right\} \end{aligned}$$

본 논문에서는 M은 3, 그리고 N은 4로 한다. 즉, 4개의 방향과 4개의 크기를 고려하여 총 16개의 커널을 사용한다. 또한, 가우시안 함수의 형태를 결정하는 σ는 2π의 값으로 하며 블록의 크기인 N×N 크기의 마스크로 커널을 표현한다.

식 (9)와 같이 가보 웨이블릿 커널이 정의되면 본 논문에서 텍스쳐 특징으로 사용하는 가보 웨이블릿 인수는 각 커널을 블록에 적용하여 계산될 수 있다. 식 (10)은 영상 F에서 (x,y)를 시작 위치로 하는 블록에 대한 가보 텍스쳐 인수를 계산하는 식을 보여준다.

$$G_{m\theta} = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} F(x+i, y+j) \cdot \psi_{m\theta}(i,j) \quad (10)$$

본 논문에서 사용하는 가보 텍스쳐 특징은 식 (10)과 같이 4개의 방향과 4개의 크기에 의해 생성되는 16개의 가보 웨이블릿 인수를 가지는 벡터로 식 (11)과 같이 정의된다.

$$\boldsymbol{G} = [G_{00}, G_{01}, \ldots, G_{m0}] \quad (11)$$

3. 탐색 영역의 동적 설정

본 장에서는 블록 정합에서 사용되는 탐색 영역을 동적으로 설정하는 방법에 대해 설명한다. 본 논문에서는 고정적인 탐색 영역을 사용하는 기존의 블록 정합 방법과는 달리 이전 시점에서 구한 움직임 벡터들과 이들의 신뢰도를 측정하여 탐색 영역의 중심 위치와 크기를 동적으로 변화시킨다.

블록 정합은 현 시점 영상의 해당 블록과 가장 유사한 후보 블록을 이전 시점 영상의 탐색 영역 내에서 찾는 과정이다. 일반적으로, 블록 정합에서 사용하는 탐색 영역 SA(Search Area)를 중심 위치와 크기를 나타내는 벡터로 표현하면 식 (12)와 같다.

$$\boldsymbol{SA} = (\boldsymbol{p}(x,y),\ \boldsymbol{s}(x,y)) = ((x,y),(xs,ys)) = (x,y,xs,ys) \quad (12)$$

여기서, $\boldsymbol{p}(x,y)$는 탐색 영역의 중심 위치를 나타내는 벡터이고, $\boldsymbol{s}(x,y)$는 탐색 영역의 크기, 즉 가로와 세로

의 길이를 나타내는 벡터이다. 본 논문에서는 식 (12)와 같은 표현법을 기반으로 t 시점의 i번째 블록 B_i의 블록 정합 시 적용하게 될 탐색 영역 $\boldsymbol{SA}_t(B_i)$를 식 (13)과 같이 정의한다.

$$\boldsymbol{SA}_t(B_i) = (\boldsymbol{p}_t^{B_i}(x,y),\ s_t^{B_i}(x,y)) \tag{13}$$

$$\boldsymbol{p}_t^{B_i}(x,y) = \boldsymbol{c}_{t-1}^{B_i}(x,y) + MV_{t-1}(B_i)$$

$$s_t^{B_i}(x,y) = s_{\min}(x,y) + (1 - CF_{t-1}(B_i)) \cdot (s_{\max}(x,y) - s_{\min}(x,y))$$

식 (13)에서 탐색 영역의 중심 위치 $\boldsymbol{p}_t^{B_i}(x,y)$는 블록 정합 초기에 분할된 블록 B_i의 중심 위치 $\boldsymbol{c}_{t-1}^{B_i}(x,y)$와 $t-1$ 시점에서 추출한 블록 B_i의 움직임 벡터 $MV_{t-1}(B_i)$의 합으로 정의된다. 영상의 움직임은 급격히 변화하기 보다는 완만하게 변화한다는 연속성 제약 사항(motion continuity constraint)[9]에 근거하여 $t-1$ 시점에서 추출한 움직임 벡터를 이용해서 t 시점에서의 움직임 벡터의 위치를 예측할 수 있다.

식 (13)에서 탐색 영역의 크기 $s_t^{B_i}(x,y)$는 본 논문에서 사용하는 탐색 영역의 최소 크기와 최대 크기를 나타내는 $s_{\min}$과 $s_{\max}$ 및 블록 B_i에서 추출되는 움직임 벡터의 신뢰도 $CF_{t-1}(B_i)$를 이용하여 정의된다. 식 (13)에서와 같이 탐색 영역의 크기는 움직임 벡터의 신뢰도에 따라 연속적으로 변화하며, 움직임 벡터의 신뢰도는 탐색 영역의 크기를 증폭(amplify) 또는 위축(deamplify)시키는 가중치의 역할을 하도록 정의된다. 즉, $t-1$ 시점에서 추출된 움직임 벡터가 높은 신뢰도를 가질 경우에는 블록 정합이 정확하고 안정적으로 수행되었다는 것을 나타내므로 t 시점에서 탐색 영역의 크기를 축소하여 시간적인 복잡도를 최소화하고, $t-1$ 시점에서 추출된 움직임 벡터가 낮은 신뢰도를 가질 경우에는 블록 정합이 불안정하게 수행되었다는 것을 나타내므로 t 시점에서 탐색 영역의 크기를 확대하여 블록 정합의 정확도를 최대화할 수 있도록 한다. 본 논문에서는 이러한 개념을 기반으로 움직임 벡터의 신뢰도를 식 (14)와 같이 정의한다. 식 (14)에서와 같이 $t-1$ 시점의 움직임 벡터의 신뢰도는 블록 B_i에서 구한 움직임 벡터와 블록 B_i의 인접 블록들에서 구한 움직임 벡터들을 비교하여 구한다. 일반적으로, 인접한 블록들은 유사한 움직임 벡터를 가질 확률이 높으므로 블록 B_i에서 구한 움직임 벡터와 인접 블록들에서 구한 움직임 벡터들의 차이가 클수록 블록은 큰 움직임을 가질 확률이 높고, 움직임 벡터들의 차이가 작을수록 블록은 작은 움직임을 가질 확률이 높다.

$$CF_{t-1}(B_i) = \frac{K_1}{1 + K_2 \cdot MD_{t-1}(B_i)} \tag{14}$$

$$MD_{t-1}(B_i) = \frac{\|r_{t-1} - \mu_{t-1}\|^2}{\sigma_{t-1}^2}$$

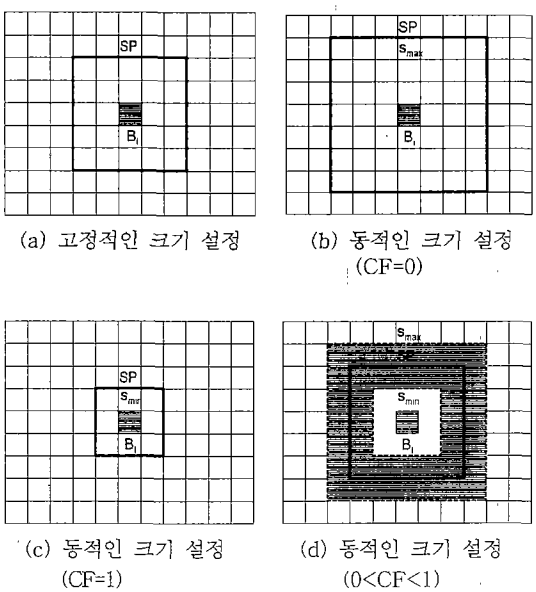
$$r_{t-1} = MV_{t-1}(B_i)$$

$$\mu_{t-1} = mean\ of\ MV_{t-1}(B_j),\ j \in neighborhood\ of\ i$$

$$\sigma_{t-1} = variance\ of\ MV_{t-1}(B_j),\ j \in neighborhood\ of\ i$$

식 (14)의 $MD_{t-1}(B_i)$는 $t-1$ 시점의 i번째 블록 B_i의 움직임 벡터인 r_{t-1}의 마할라노비스 거리(Mahalanobis Distance)를 나타낸다. 즉, 인접 블록에서의 움직임 벡터들과 비교하여 블록 B_i의 움직임 벡터의 차이가 얼마나 상이한 값을 갖는가를 평가한다. 그리고 식 (14)에서 K_1과 K_2는 상수로서 K_1은 신뢰도가 취할 수 있는 최대값을 결정하고, K_2는 신뢰도의 범위를 결정하는 역할을 한다. 본 논문에서는 이들을 0.1과 1로 설정한다.

그림 3은 탐색 영역의 크기를 고정적으로 설정하는 기존의 방법과 동적으로 설정하는 방법을 보여주고 있다. 그림 3의 (a)는 탐색 영역의 크기가 고정적으로 설정되는 기존의 방법이고 그림 3의 (b), (c), (d)는 본 논문에서 제안한 방법으로 움직임 벡터의 신뢰도를 나타내는 CF값이 0, 1, 그리고 0과 1 사이의 값을 가질 때의 탐색 영역의 크기 변화를 보여준다.



(a) 고정적인 크기 설정 (b) 동적인 크기 설정 (CF=0)

(c) 동적인 크기 설정 (CF=1) (d) 동적인 크기 설정 (0<CF<1)

그림 3 탐색 영역의 크기 설정

4. 정합 유사 함수 정의

본 장에서는 블록 사이의 정합 척도로 사용되는 정합 유사 함수를 정의하는 방법에 대해 설명한다. 본 논문에서 정의하는 정합 유사 함수는 칼라 특징, 에지 특징, 텍스쳐 특징과 같은 다수의 특징들을 사용하여 정의되며 각 특징의 기여 정도를 나타내는 가중치를 포함하고 있다. 본 논문에서는 블록 정합 시 고정된 가중치를 사용하지 않고 특징값들의 비균일도를 측정하여 가중치를 자동으로 설정한다.

일반적으로, 블록 단위의 정합에서 발생하는 가장 큰 문제 중의 하나는 특징값이 유사하거나 균일한 탐색 영역 내에서의 정합이다. 즉, 특징이 균일한 탐색 영역에서 정합을 수행할 경우에는 블록을 구분할 수 있는 특징값들의 부족으로 부정확한 정합이 이루어질 확률이 높으며, 반대로 특징이 비균일한 탐색 영역에서 정합을 수행할 경우에는 보다 정확한 정합이 이루어진다. 특히, 대부분 하나의 특징만을 사용하여 정합을 수행하는 기존의 블록 정합 알고리듬들은 특징값이 균일한 탐색 영역에서 많은 정합 오류가 발생한다. 따라서 본 논문에서는 다수의 특징을 사용하여 정합할 수 있도록 정합 유사 함수를 정의한다. 그리고 탐색 영역 내에서 각 특징값들의 비균일도를 측정하고 이 값들을 정합 유사 함수를 구성하는 특징들의 가중치로 사용한다. 각 블록에 대한 정합을 수행할 때 탐색 영역 내의 특징값들의 분포를 반영한 가중치를 사용하므로 보다 정확한 블록 정합을 수행하게 된다.

본 논문에서는 탐색 영역 내의 특징값들의 비균일도를 구하기 위해 엔트로피(entropy)를 사용한다. 엔트로피는 통계학이나 코딩 이론에서 널리 사용되는 개념으로 확률 분포의 불확실성(uncertainty)을 얻기 위해 사용된다[10]. 즉, 균일한 분포를 이루는 모집단일 경우 낮은 값의 엔트로피를 가지며, 비균일한 분포를 이루는 모집단일 경우 높은 값의 엔트로피를 가진다. 일반적으로, 모집단에 대한 엔트로피는 식 (15)와 같이 정의된다.

$$H=-\sum_{i=1}^{n} P_i \log P_i \tag{15}$$

식 (15)에서 P_i는 i번째 샘플이 발생할 확률을 의미한다. 이와 같은 엔트로피 개념을 이용하여 블록 정합을 수행할 탐색 영역에서 특징값들의 비균일도를 구한다. 칼라 특징에 대한 비균일도는 식 (16)과 같이 정의된다.

$$H_{color}=-\sum_{B_i\in W} P_{color}(C_{B_i}) \log P_{color}(C_{B_i}) \tag{16}$$

여기서, W는 탐색 영역을 나타내고 B_i는 탐색 영역 내의 i번째 블록을 나타낸다. 그리고 C_{B_i}는 블록 B_i의 칼라 특징값을 나타내고 $P_{color}(C_{B_i})$는 탐색 영역 내에서 칼라 특징값 C_{B_i}를 가지는 블록이 존재할 확률을 의미한다. 에지 특징에 대한 비균일도는 식 (17)과 같이 정의된다.

$$H_{edgeness}=-\sum_{B_i\in W} P_{edgeness}(E_{B_i}) \log P_{edgeness}(E_{B_i}) \tag{17}$$

여기서, E_{B_i}는 블록 B_i의 에지 특징값을 나타내고 $P_{edgeness}(E_{B_i})$는 탐색 영역 내에서 에지 특징값 E_{B_i}를 가지는 블록이 존재할 확률을 의미한다. 텍스쳐 특징에 대한 비균일도는 식 (18)과 같이 정의된다.

$$H_{texture}=-\sum_{B_i\in W} P_{texture}(G_{B_i}) \log P_{texture}(G_{B_i}) \tag{18}$$

여기서, G_{B_i}는 블록 B_i의 가보 텍스쳐 특징값을 나타내고 $P_{texture}(G_{B_i})$는 탐색 영역 내에서 가보 텍스쳐 특징값 G_{B_i}를 가지는 블록이 존재할 확률을 의미한다.

이러한 특징값들의 비균일도를 고려하여 본 논문에서 정의한 정합 유사 함수는 식 (19)와 같다.

$$\Phi(\alpha,\beta,\gamma)=\max_{i\in W}\left(\alpha\cdot\frac{1}{M_{color}(B_i)}+\beta\cdot\frac{1}{M_{edgeness}(B_i)}+\gamma\cdot\frac{1}{M_{texture}(B_i)}\right) \tag{19}$$

$$where\quad 0\le\alpha,\beta,\gamma\le 1,\quad \alpha+\beta+\gamma=1$$

여기서, α, β, γ는 정합 유사 함수를 구성하는 칼라 특징, 에지 특징, 텍스쳐 특징의 기여 정도를 나타내는 가중치이다. 특징값들의 가중치는 식 (16), 식 (17), 식 (18)에서 구한 특징값들의 비균일도를 식 (20)과 같이 정규화하여 사용한다.

$$\alpha=\frac{H_{color}}{H_{color}+H_{edgeness}+H_{texture}} \tag{20}$$

$$\beta=\frac{H_{edgeness}}{H_{color}+H_{edgeness}+H_{texture}}$$

$$\gamma=\frac{H_{texture}}{H_{color}+H_{edgeness}+H_{texture}}$$

식 (19)에서 $M_{color}(B_i)$, $M_{edgeness}(B_i)$, $M_{texture}(B_i)$는 각각 i번째 블록 B_i에 대한 칼라 특징의 유사도, 에지 특징의 유사도, 텍스쳐 특징의 유사도를 나타낸다. 이들 특징들의 유사도는 블록 B_i와 정합 시 이에 대응되는 블록이 가지는 특징과의 거리로 정의될 수 있다. 칼라 특징의 유사도는 식 (21)과 같이 정의된다.

$$M_{color}(B_i)=\left|\left|\frac{C_{B_R}^{t}}{C_{\max}}-\frac{C_{B_i}^{t-1}}{C_{\max}}\right|\right| \tag{21}$$

여기서, $C_{B_R}^{t}$는 블록 정합에서 참조(reference) 블록으로 사용하는 블록에 대한 칼라 특징 벡터를 의미하고 $C_{B_i}^{t-1}$는 정합을 시도할 탐색 영역 내의 i번째 블록에

서 계산한 칼라 특징 벡터를 의미한다. 그리고 $C_{\max}$는 칼라 특징값의 최대값으로 칼라 특징 벡터를 정규화하기 위해 사용한다. 에지 특징의 유사도는 식 (22)와 같이 정의된다.

$$M_{edgeness}(B_i)=\left|\left|\frac{E^{t}_{B_k}}{E_{\max}}-\frac{E^{t-1}_{B_i}}{E_{\max}}\right|\right| \quad (22)$$

여기서, $E^{t}_{B_k}$는 블록 정합에서 참조 블록으로 사용하는 블록에 대한 에지 특징 벡터를 의미하고 $E^{t-1}_{B_i}$는 정합을 시도할 탐색 영역 내의 i번째 블록에서 계산한 에지 특징 벡터를 의미한다. 그리고 $E_{\max}$는 에지 특징값의 최대값으로 에지 특징 벡터를 정규화하기 위해 사용한다. 텍스쳐 특징의 유사도는 식 (23)과 같이 정의된다.

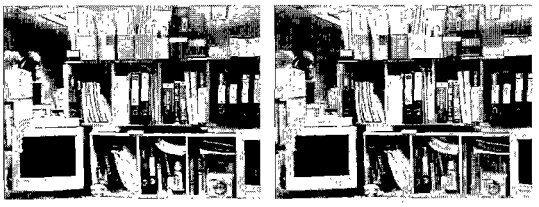
$$M_{texture}(B_i)=\left|\left|\frac{G^{t}_{B_k}}{G_{\max}}-\frac{G^{t-1}_{B_i}}{G_{\max}}\right|\right| \quad (23)$$

여기서, $G^{t}_{B_k}$는 블록 정합에서 참조 블록으로 사용하는 블록에 대한 텍스쳐 특징 벡터를 의미하고 $G^{t-1}_{B_i}$는 정합을 시도할 탐색 영역 내의 i번째 블록에서 계산한 텍스쳐 특징 벡터를 의미한다. 그리고 $G_{\max}$는 텍스쳐 특징값의 최대값으로 텍스쳐 특징 벡터를 정규화하기 위해 사용한다.

본 장에서 정의한 정합 유사 함수는 블록 정합 시 사용되며 각 블록에 대한 정합을 시도할 탐색 영역 내에 존재하는 특징값들의 비균일도에 따라 가중치를 자동으로 설정하므로 보다 정확한 정합 결과를 얻을 수 있다.

5. 실험 결과 및 결론

본 장에서는 제안한 활동적 블록 정합 알고리듬에 대한 실험 결과를 보인다. 실험에서는 IBM-PC에 장착된 Meteor 칼라 영상 처리 보드를 통해 실내 환경에서 카메라를 천천히 움직이면서 연속적으로 촬영한 칼라 영상들을 입력 영상으로 사용하였다. 그리고 마이크로소프트의 비주얼 C++ 컴파일러를 이용하여 윈도우즈 98 운영체제 환경에서 실험하였다.

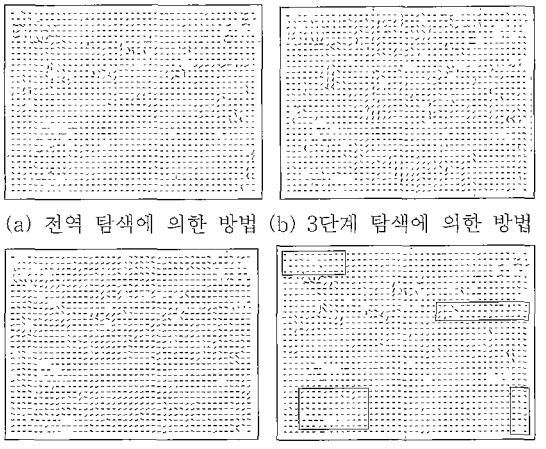


(a) 입력 영상 t (b) 입력 영상 t+△t

그림 4 입력 영상 1

그림 4는 실험에서 사용한 입력 영상을 보여준다. 입력 영상은 실내 공간에서 카메라를 2 pixels/frame의 속도로 왼쪽으로 수평 이동하면서 촬영하였다.

본 논문에서는 기존의 방법들과 본 논문에서 제안한 방법들과의 성능을 평가하기 위해 시각적인 측면과 정확성 측면을 중심으로 결과를 비교 분석한다. 본 논문에서 입력 영상의 크기는 320×240 이고 블록의 크기는 8×8 이다. 그리고 탐색 영역이 변화하지 않는 블록 정합 알고리듬에서 사용한 탐색 영역의 크기는 15×15 이다. 그림 5는 블록의 크기를 8×8로 하여 기존의 블록 정합 알고리듬인 전역 탐색에 의한 방법, 3 단계 탐색에 의한 방법, 4 단계 탐색에 의한 방법과 본 논문에서 제안한 활동적 블록 정합 알고리듬을 이용하여 추출한 움직임 벡터의 예를 시각적으로 보여주고 있다.



(a) 전역 탐색에 의한 방법 (b) 3단계 탐색에 의한 방법

(c) 4단계 탐색에 의한 방법 (d) 본 논문에서 제안한 방법

그림 5 움직임 벡터 추출 결과

그림 5에서 확인할 수 있듯이 기존의 블록 정합 방법들 모두 칼라 특징값이 분명한 영역에서는 움직임 벡터를 어느정도 정확히 추출하였지만 에지가 존재하는 물체의 경계 부분이나 칼라 특징이 균일하게 분포하는 영역에서는 부정확한 블록 정합으로 인해 움직임 벡터를 잘못 추출하였다. 그리고, 3 단계 탐색에 의한 방법은 탐색 횟수의 부족으로 지역적 최소화에 빠지는 경향이 있으며, 4 단계 탐색에 의한 방법 역시 잡음에 민감한 반응을 보였다. 그러나 그림 5의 (d)에서 사각형의 실선으로 표시된 영역에서 확인할 수 있듯이 본 논문에서 제안한 방법은 다수의 특징을 사용하므로 물체의 경계 부분이나 칼라 특징이 균일하게 분포한 영역 뿐만 아니

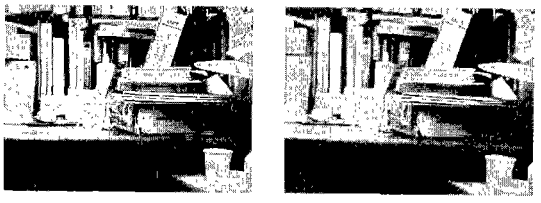
표 1 가중치와 탐색 영역의 변화

	Minumum	Maximum	Average	Standard Deviation
w_{color}	0.216932	0.843844	0.546314	0.204063
$w_{edgeness}$	0.014030	0.458906	0.285734	0.238745
$w_{texture}$	0.152126	0.324162	0.167954	0.129015
Search Area	5×5	15×15	6.33	0.86

라 텍스쳐 특징이 존재하거나 잡음이 존재하는 영역에서도 움직임 벡터를 보다 정확히 추출했다.

표 1은 본 논문에서 사용한 탐색 영역과 특징값의 가중치에 대한 동적인 변화를 수량적으로 보여준다. 특히, 다른 특징에 비해 에지 특징에 대한 가중치의 변화가 크다는 것을 관찰할 수 있다. 이것은 그림 4의 입력 영상이 코너와 라인과 같은 유형의 에지가 많이 포함되어 있기 때문이다. 그리고 표 1에서 탐색 영역의 평균 크기가 탐색 영역의 최소 크기와 근접하다는 것은 본 논문에서 제안한 알고리듬의 효율성을 나타내고 있다. 즉, 본 논문에서 제안한 활동적 블록 정합 알고리듬은 다중 특징을 사용하여 탐색 영역을 축소한다.

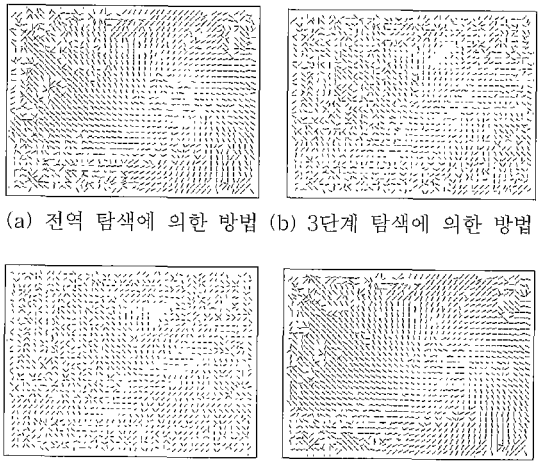
본 논문에서는 모든 종류의 카메라의 동작이 포함된 매우 복잡한 환경에 대해서도 실험을 하였다. 그림 6은 2 pixels/frame 속도의 수평 이동, -2 pixels/frame 속도의 수직 이동, 1.05 배율/frame 속도의 줌인, 2 도/frame 속도의 시계 방향으로의 회전 동작이 혼합되어 생성된 입력 영상을 보여준다.



(a) 입력 영상 t (b) 입력 영상 t+△t

그림 6 입력 영상 2

그림 7은 그림 6으로부터 추출된 움직임 벡터를 보여준다. 이상적으로, 움직임 벡터는 영상의 중심으로부터 남동쪽으로 2 pixels 떨어진 위치를 중심으로 나선형 모양을 형성하여야 한다. 그림 7에서 확인할 수 있듯이 본 논문에서 제안한 방법이 보다 정확하게 움직임 벡터를 추출함을 확인할 수 있다.



(a) 전역 탐색에 의한 방법 (b) 3단계 탐색에 의한 방법

(c) 탐색 영역을 변화하는 방법 (d) 제안한 방법

그림 7 움직임 벡터 추출 결과

본 논문에서는 움직임 벡터의 정확성을 수량적으로 확인하기 위해 식 (24)와 같이 이미 알고 있는 정확한 움직임 벡터와 측정된 움직임 벡터 사이의 정규화된 방향 에러(normalized angular error)를 사용한다[11].

$$E_\theta = \sum_{i=1}^{N} \cos^{-1}\left(\frac{(\mathbf{V}_c^i, 1)\cdot(\mathbf{V}_e^i, 1)}{\sqrt{1+\|\mathbf{V}_c^i\|^2}\sqrt{1+\|\mathbf{V}_e^i\|^2}}\right) \qquad (24)$$

그림 8은 입력 영상 1과 입력 영상 2로부터 추출한 정규화된 방향 에러를 그래프로 보여준다. 그림 8에서 확인할 수 있듯이 본 논문에서 제안한 방법이 가장 우수함을 확인할 수 있다. 기존의 방법 중 3단계 탐색에 의한 방법이 가장 나쁜 결과를 나타내고, 4단계 탐색에 의한 방법도 3단계 탐색에 의한 방법과 유사한 결과를 나타낸다. 전역 탐색에 의한 방법은 기존의 다른 방법에 비해 좋은 결과를 나타내지만 제안된 방법보다는 나쁜 결과를 나타낸다.

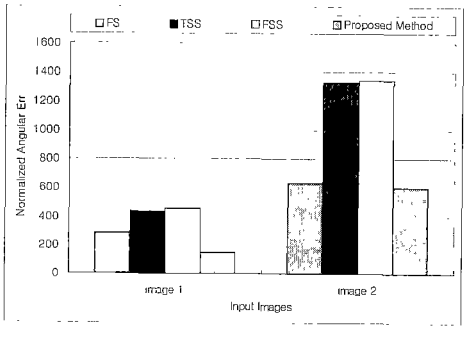


그림 8 정규화된 방향 에러

본 논문에서는 블록 단위의 움직임 벡터를 효율적으로 추출하는 활동적 블록 정합 알고리듬을 제안하였다. 활동적 블록 정합 알고리듬은 칼라 특징, 에지 특징, 텍스쳐 특징과 같은 다중의 특징값을 이용하는 알고리듬으로 각 특징의 기여 정도를 나타내는 가중치를 탐색 영역 내의 특징값들의 비균일도를 측정하여 자동으로 조절할 수 있도록 정합 유사 함수를 정의하였다. 그리고 고정적인 탐색 영역을 사용하는 기본의 블록 정합 방법과는 달리 이전 시점까지 구한 움직임 벡터들의 신뢰도를 기반으로 탐색 영역의 중심 위치 및 크기를 동적으로 설정하였다.

본 논문에서 제안한 활동적 블록 정합 알고리듬을 개선하기 위해서는 다음과 같은 내용들이 보완되어야 한다. 다중의 특징으로 정의된 정합 유사 함수를 이용하여 블록 정합을 수행할 경우 블록 정합의 정확도는 증가하지만 필연적으로 시간적인 복잡도가 높아진다. 따라서 보다 단순화된 기법을 통해 다중의 특징을 고려할 수 있는 방법에 대한 연구가 필요하다. 그리고 일정한 시간 간격을 두고 연속적으로 추출되는 움직임 벡터의 크기를 일정하게 조절할 수 있는 시간적인 동기화에 대한 연구도 보강될 필요가 있다.

참 고 문 헌

[1] Anandan, "Computational Framework and an Algorithm for the Measurement of Visual Motion," International Journal of Computer Vision, Vol. 2, pp. 283-310, 1988.

[2] Kyoung Won Lim, Byung Cheol Song, and Jong Beom Ra, "Fast Hierarchical Block Matching algorithm Utilizing Spatial Motion Vector Correlation," Proceedings of Visual Communications and Image processing, Vol. 3024, pp. 284-291, 1997.

[3] T. Koga, K. Iinuma, A. Hirano, Y. Iijima, and T. Ishiguro, "Motion Compensated Interframe Coding for Video Conferencing," in Proc. NTC81, pp. C9.6.1-9.6.5, New Orleans, LA, 1981.

[4] R. Li and M. L. Liou, "A New Three-step Search Algorithm for Block Motion Estimation," IEEE Trans. On Circuits and Systems for Video Tech., Vol. 4, No. 4, pp. 438-441, 1994

[5] L. G. Chen, E. T. Chen, Y. S. Jehng, and T. D. Chiueh, "An Efficient Parallel Motion Estimation Algorithm for Digital Image Processing," IEEE Trans. On Circuits and Systems for Video Tech., Vol. 1, No. 4, pp. 378-385, 1991.

[6] Lai-Man Po and Wing-Chung Ma, "A Novel Four-Step Algorithm for Fast Block Motion Estimation," IEEE Trans. On Circuits and Systems for Video Tech., Vol. 6, No. 3, pp. 313-317, 1996.

[7] Christopher C. Yang and Jeffrey J. Rodriguez, "Efficient Luminance and Saturation Processing Techniques for Bypassing Color Coordinate Transformations," IEEE Systems, Man and Cybernetics, pp. 667-672, 1995.

[8] W. Y. Ma and B. S. Manjunath, "Texture Features and Learning Similarity," IEEE International Conference on Computer Vision and Pattern Recognition, 1996.

[9] Vincent S. and S. Hwang, "Tracking Feature Points in Time-Varying Images using an Opportunistic Selection Approach," Pattern Recognition, Vol. 22, pp. 247-256, 1989.

[10] Ian Hannah and Roy Davies, "The Use of Variance and Entropic Thresholding Methods for Image Segmentation," Pattern Recognition, Vol. 28, No. 8, pp. 1135-1143, 1995.

[11] Yu-Te Wu, Jeffrey Cohn and Ching-Chung Li, "Optical Flow Estimation using Wavelet Motion Model," International Conference on Computer Vision, pp. 992-998, 1998.

장 석 우

1995년 2월 숭실대학교 전자계산학과 졸업(공학사). 1997년 2월 숭실대학교 대학원 전자계산학과 졸업(공학석사). 2000년 8월 숭실대학교 대학원 컴퓨터학과 졸업(공학박사). 현재 숭실대학교 생산기술연구소 연구원. 관심분야는 컴퓨터 비전, 영상처리, 동작이해, 비디오 검색 등임.

최 형 일

1979년 2월 연세대학교 전자공학과 졸업(공학사). 1982년 6월 미시간대학교 전산공학과 졸업(공학석사). 1987년 6월 미시간대학교 전산공학과 졸업(공학박사). 현재 숭실대학교 미디어학부 교수. 관심분야는 컴퓨터 비전, 패턴 인식, 워터마킹, 인터페이스 에이전트 등임.