

동영상에서 물체의 추출과 배경영역의 상대적인 깊이 추정

박 영 민[†] · 장 주 석^{**}

요 약

컴퓨터 비전에 관한 고전적인 연구 주제들 중의 하나는 두 개 이상의 이미지로부터 3차원 형상을 재구성하는 3차원 변환에 관한 것이다. 본 논문은 단안 카메라로 촬영한 일반적인 2차원 영상물에서 능동적으로 움직이는 3차원 영상의 깊이 정보를 추출하는 문제를 다룬다. 연속하는 프레임들간의 영상 블록의 움직임을 평가하여 카메라의 회전과 배울효과를 보상하고 다음과 같은 두 개의 단계에 걸쳐 블록의 움직임을 추출한다. (i) 블록의 위치와 움직임을 이용하여 카메라의 이동과 초점거리에 대한 전역 파라메타를 계산한다. (ii) 전역 파라메타, 블록의 위치와 움직임을 이용하여 평균 영상 깊이에 대한 상대적인 블록의 깊이를 계산한다. 다양한 동영상을 대상으로 특이적인 경우와 그렇지 않은 경우를 실험하였다. 결과로 얻어지는 상대적인 깊이 정보와 객체는 인간이 판단하는 경우와 동일함을 보였다.

키워드 : 깊이 추정, 영역 분할, 물체 추출

Moving Object Extraction and Relative Depth Estimation of Background regions in Video Sequences

Young-Min Park[†] · Chu-Seok Chang^{**}

ABSTRACT

One of the classic research problems in computer vision is that of stereo, i.e., the reconstruction of three dimensional shape from two or more images. This paper deals with the problem of extracting depth information of non-rigid dynamic 3D scenes from general 2D video sequences taken by monocular camera, such as movies, documentaries, and dramas. Depth of the blocks are extracted from the resultant block motions throughout following two steps: (i) calculation of global parameters concerned with camera translations and focal length using the locations of blocks and their motions, (ii) calculation of each block depth relative to average image depth using the global parameters and the location of the block and its motion. Both singular and non-singular cases are experimented with various video sequences. The resultant relative depths and ego-motion object shapes are virtually identical to human vision.

Key Words : Depth Estimation, Region Segmentation, Object Extraction

1. 서 론

정보화 사회로 발전하면서 2차원으로 표현하던 동영상도 점차 3차원으로 표현 방법이 전환되고 있으며, 효과적인 정보전달을 위한 수단으로 음성과 시각을 동시에 이용하는 동영상의 위치는 더욱 더 중요하게 부각되고 있다[1]. 이러한 3차원 영상으로 구현하는 기법으로는 두 대 이상의 카메라를 사용하여 동시에 여러 방향에서의 이미지를 획득하여 각 영상에서의 연관성을 가진 두 점의 상관관계를 이용하여 깊이 정보를 추출하는 스테레오(stereo)영상 모델과 레이저 장비를 사용하여 실세계의 3차원 거리 정보를 직접 획득하는 3차원 레이저 모델이 있으며 마지막으로, 단안(monocular)카

메라를 사용하여 만들어진 대부분의 TV, 영화와 같은 영상물에서 프레임사이에 존재 하는 기하학적 구조부터 깊이를 추정하는 단안 영상 모델이 있다[2].

본 논문의 목적은 한 개의 카메라로 촬영한 단안 영상에 대하여 카메라의 움직임을 추출하고 깊이정보에 영향을 받지 않은 회전과 배울성분을 제거한 후 카메라 이동성분만을 이용하여 영상 프레임의 각 영역에 대한 상대적인 깊이 정보를 추출하는 것이다.

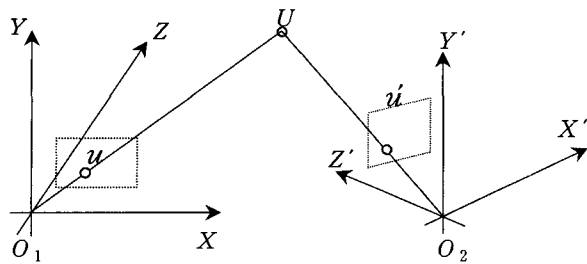
2. 단안 영상 모델

단안영상 시퀀스에서 연속한 프레임 간의 기하학적 관계는 비평행축 스테레오 기하와 동일하다. (그림 1)에서 보면, 실세계의 한 점 $U(x, y, z)$ 는 영상면에서 한 점 $u(x, y)$ 로 투영되고, 카메라가 움직여서 회전과 이동이 일어난 후 점

[†] 정 회 원 : 경운대학교 멀티미디어공학부 전임강사

^{**} 정 회 원 : 경운대학교 멀티미디어공학부 교수

논문접수 : 2004년 11월 2일, 심사완료 : 2005년 1월 5일



(그림 1) 비평행축 스테레오 기하

U 는 $u'(x', y')$ 에 투영된다. 이때 두 점 u, u' 의 기하학적 구조로부터 깊이를 추정할 수 있다.

2.1 외극기하 방법의 문제점

비평행축 기하로부터 깊이 정보를 추출하기 위한 방법으로서 외극기하를 가장 많이 사용한다. 외극기하는 임의의 점 P 가 공간상에 고정되어 있고 순수하게 카메라가 움직인 경우에만 성립된다. 따라서 카메라움직임과 관계없이 자의적으로 움직이는 물체의 경우에는 국외점(outlier)에 의하여 결과가 부정확하다[3]. 이 경우, 로렌츠 함수(lorentzian function) [4] 혹은 절삭 2차 평가기준(truncated quadratic estimation criteria)을 사용하여 어느 정도의 국외점을 제거할 수는 있지만, 움직이는 물체의 영역이 움직이지 않고 고정되어 있는 영역에 비해 상대적으로 더 클 경우에는 적용이 곤란하다. 또한 외극기하 방법에서는 전형적으로 모서리나 직선을 특징점으로 사용한다. 그러나 실제의 영상에서는 그러한 인위적인 도형이 존재하지 않는 경우가 많기 때문에 적용대상이 건축물과 같은 인공물의 영상으로 제한되고 에러에 민감하다. 이러한 에러민감성 해결하기 위해서 과평가 기법을 사용하지만 추정된 대응점 쌍의 오차분포가 영평균 오차가 아닐 경우 추정에 실패한다[5]. 그리고 영상에 따라 외극기하가 성립되지 않는 경우도 발생할 수 있다[6][7]. 특징점과 대응점 쌍들이 독립적이지 않은 경우에는 E-행렬의 퇴화차수(nullity)가 증가하거나, E-행렬의 풀이에 사용되는 SVD는 계산비용이 과다하며 E-행렬이 R과 T로 분해되지 않는 경우도 발생한다[8].

2.2 비디오 영상의 입체영상화

영상에는 전형적으로 근경과 원경이 존재한다. 논문[9][10]에서 영상블록의 움직임 벡터에 근경과 원경을 판별하는 판별식을 적용하여 원경을 추출하고 원경에 관한 정보를 이용하여 카메라 회전과 배율 파라미터를 추출한다. 논문[11][12]에서는 블록의 움직임 벡터에 카메라 회전과 배율 파라미터를 보상하여 카메라 이동 성분만으로 블록의 깊이를 추정한다. 블록의 위치와 움직임 벡터로써 식 (1)과 같이 카메라 이동 파라미터들의 관계를 추출하고 이를 이용하여 영상의 평균 깊이에 대한 블록의 상대적 깊이를 산출한다.

$$f \frac{T_x}{T_z} = f_1(x, y, v_x, v_y)$$

$$f \frac{T_y}{T_z} = f_2(x, y, v_x, v_y) \tag{1}$$

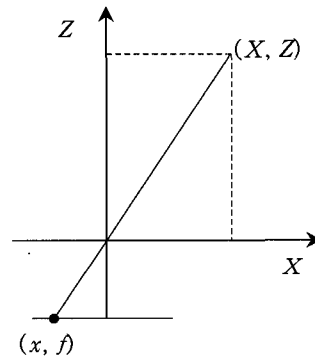
(식 1)에는 객체의 움직임도 포함되어 있으므로 객체가 영상의 넓은 부분을 차지할 경우, 산출된 값이 부정확한 문제점이 있다. 이에 대하여 본 논문에서는 반복계산법으로써 객체에 포함된 블록은 (식 1)에 적용하지 않게 함으로써 더욱 정확한 깊이 정보를 추출할 수 있고, 부산물로서 객체 영역도 추출할 수 있다. 본 논문에서는 화소의 색상으로써 영역을 분할하고 블록의 깊이를 영역의 깊이로 사상(mapping)시킨다.

2.3 움직임 벡터와 전역 파라미터와의 관계

실세계의 좌표 X, Z 축과 영상에서의 x 사이에는 (그림 2)와 같은 관계가 성립된다. 삼각형의 닮음비를 이용하면 식 (3)과 같다. (O 는 카메라 초점, f 는 초점거리)

$$U - T = U' [X Y Z]^T - [T_x T_y T_z]^T = [X' Y' Z']^T \tag{2}$$

$$\frac{X}{Z} = \frac{x}{f}, \quad \frac{Y}{Z} = \frac{y}{f} \tag{3}$$



(그림 2) 실세계 좌표와 영상 좌표계의 관계

식 (2)에 식 (3)를 적용하면 식 (4),(5)와 같이 된다.

$$\frac{Z}{f}x - T_x = \frac{Z'}{f}x' \tag{4}$$

$$\frac{Z}{f}y - T_y = \frac{Z'}{f}y' \tag{5}$$

이 식에서 $Z' = Z - T_z, v_x = x' - x, v_y = y' - y$ 라 두면 식 (6)으로 나타낼 수 있다.

$$\frac{1}{v_x} \left\{ \frac{T_z}{f} (x + v_x) - T_x \right\} = \frac{1}{v_y} \left\{ \frac{T_z}{f} (y + v_y) - T_y \right\} \tag{6}$$

식 (6)에서 양변에 공통으로 있는 $\frac{T_z}{f}$ 로 나누어 정리하면 식 (7)과 식(8)로 정의된다.

$$f \frac{T_x}{T_z} = \frac{(y_q - y_p)v_x(p)v_x(q) + x_p v_x(q)v_y(p) - x_q v_x(p)v_y(q)}{v_x(q)v_y(p) - v_x(p)v_y(q)} \quad (7)$$

$$f \frac{T_y}{T_z} = \frac{(x_q - x_p)v_y(p)v_y(q) + y_p v_y(q)v_x(p) - y_q v_y(p)v_x(q)}{v_y(q)v_x(p) - v_y(p)v_x(q)} \quad (8)$$

2.4 특이점 해석

식 (7)과 식 (8)의 분모는 같은 식으로 구성되는데, 분모가 영이 되는 특이점이 존재한다. 만일 $T_x \neq 0 \wedge T_z = 0$ 이면, 식 (7)은 무한대(∞)로 접근하고, 마찬가지로 $T_y \neq 0 \wedge T_z = 0$ 이면, 식 (8)이 무한대로 접근한다. 식 (7)과 (8)의 분자는 무한대로 접근할 수 없지만, 분모는 무한대가 될 수 있다. 따라서 $T_z = 0$ 일 때 특이점이 되며 식 (9)과 같이 전후이동의 존재 유무에 관한 판별식이 구해진다.

$$v_x(p)v_y(q) = v_x(q)v_y(p) \rightarrow T_z = 0 \quad (9)$$

$$\frac{Z(p)}{f} = -\frac{T_x}{v_x(p)} \quad (10)$$

$$\frac{Z(p)}{f} = -\frac{T_y}{v_y(p)} \quad (11)$$

식 (10)과 (11)를 이용하여 프레임의 평균깊이 \overline{Z} 를 구하면 식 (12)와 (13)과 같다.

$$\overline{Z} = \frac{1}{N} \sum_{p=1}^N Z(p) = -f T_x \frac{1}{N} \sum_{p=1}^N \frac{1}{v_x(p)} = -f T_x \overline{v_x^{-1}} \quad (12)$$

$$\overline{Z} = \frac{1}{N} \sum_{p=1}^N Z(p) = -f T_y \frac{1}{N} \sum_{p=1}^N \frac{1}{v_y(p)} = -f T_y \overline{v_y^{-1}} \quad (13)$$

여기에서 $\overline{v_x^{-1}}$ 과 $\overline{v_y^{-1}}$ 는 각각 프레임 내의 움직임 벡터의 x 방향 성분의 역수의 평균과 y 방향 성분의 역수의 평균을 뜻한다. 프레임의 평균깊이 \overline{Z} 에 대한 점 p 의 상대적인 깊이를 $z(p)$ 라 하면, 식 (10), (11), (12), (13)로부터 아래의 식 (14)와 식 (15)이 구해진다.

$$z(p) = \frac{Z(p)}{\overline{Z}} = \frac{1}{v_x(p)v_x^{-1}} \quad (14)$$

$$z(p) = \frac{Z(p)}{\overline{Z}} = \frac{1}{v_y(p)v_y^{-1}} \quad (15)$$

따라서 점 p 가 식 (16)의 판별식을 만족하면 점 p 는 배경에 속한 화소이며 그 상대적인 깊이 $z(p)$ 는 식 (14)에서 구할 수 있다.

$$\text{판별식} : v_x(p)\overline{v_x^{-1}} = v_y(p)\overline{v_y^{-1}} \quad (16)$$

위의 판별식을 만족하지 않으면 점 p 는 움직이는 영역에 속한 화소로서 판단한다.

2.5 일반적인 경우

전후이동이 있는 일반적인 경우에는 판별식 $v_x(p)v_y(q) \neq v_x(q)v_y(p)$ 의 관계가 성립된다. 이 경우 파라메타 α, β 를 아래와 같이 정의한다.

$$\alpha \equiv f \frac{T_x}{T_z}, \quad \beta \equiv f \frac{T_y}{T_z}$$

새로 정의된 파라메타를 이용하여 $\frac{Z(p)}{T_z}$ 를 구하면 식 (17), (18)이 얻어진다.

$$\frac{Z(p)}{T_z} = \frac{1}{v_x(p)}(x_p + v_x(p) - \alpha) \quad (17)$$

$$\frac{Z(p)}{T_z} = \frac{1}{v_y(p)}(y_p + v_y(p) - \beta) \quad (18)$$

점 p 가 아래의 식 (19)의 판별식을 만족하면 배경에 속한 화소로서 판단하고 그렇지 않으면 움직이는 영역에 속한 화소로서 판단한다.

판별식 :

$$v_y(p)(x_p + v_x(p) - \alpha) = v_x(p)(y_p + v_y(p) - \beta) \quad (19)$$

전역변수인 T_z 를 식에서 제거하기 위하여 프레임의 평균깊이 \overline{Z} 에 대한 점 p 의 상대적인 깊이를 구한다. 파라메타 γ 를 식 (20)과 같이 정의하면 평균 깊이 $\overline{Z} = \gamma T_z$ 가 된다.

$$\gamma \equiv \frac{1}{N} \sum_{p=1}^N \frac{Z(p)}{T_z} = \frac{\frac{1}{N} \sum_{p=1}^N Z(p)}{T_z} = \frac{\overline{Z}}{T_z} \quad (20)$$

식 (17), (18)와 \overline{Z} 를 이용하여 프레임의 평균깊이 \overline{Z} 에 대한 임의의 점 p 의 상대적인 깊이인 $z(p)$ 를 구하면 식 (21)가 얻어진다.

$$z(p) = \frac{Z(p)}{\overline{Z}} = \frac{x_p + v_x(p) - \alpha}{\gamma v_x(p)} \quad (22)$$

$$z(p) = \frac{Z(p)}{\overline{Z}} = \frac{y_p + v_y(p) - \beta}{\gamma v_y(p)} \quad (23)$$

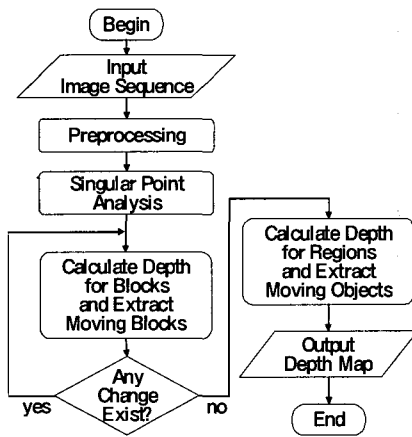
3. 동영상의 깊이 추정 알고리즘

실제의 동영상들은 매우 다양하고 불균일하며 많은 잡영이 포함되어 있다. 따라서 2장에서 개발한 수식 체계를 동

영상에 적용하는 알고리즘을 구축한다.

3.1 알고리즘 개요

(그림 3)은 동영상으로부터 깊이정보를 추출하기 위한 알고리즘이다. 영상 시퀀스를 입력하고 전처리를 행한 다음 그 움직임 벡터로써 특이점 분석을 하여 카메라의 이동이 있는지 여부를 판별하고 카메라의 이동이 있는 경우에는 전후이동이 있는지를 판별한다. 영상에 카메라의 이동이 없다고 판단되면 깊이 정보를 구할 수 없으므로 다음 프레임으로 진행한다. 카메라의 이동이 있는 경우, 전후이동 유무에 따라 해당되는 수식 체계를 적용한다.

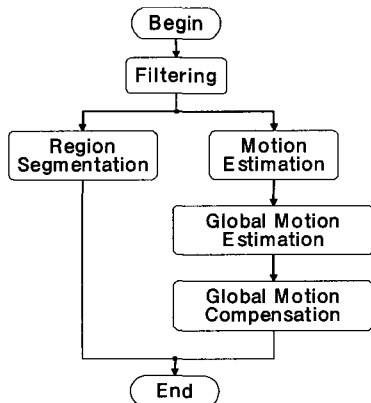


(그림 3) 전체 알고리즘 플로차트

이렇게 계산한 깊이와 객체 정보는 블록에 관한 것이므로 전처리에서 구한 영역 정보에 사상한다. 그리하여 각 영역이 배경에 속하는지 움직이는 객체에 속하는지를 결정하고 배경에 속할 경우에는 그 평균 깊이를 계산한다.

3.2 전처리

전처리 과정은 (그림 4)와 같이 평활화(filtering), 영역분할(region segmentation), 움직임 추정(motion estimation), 전역움직임 추정(global motion estimation) 및 보상(compensation)으로 구성된다.



(그림 4) 전처리 과정의 플로차트

3.2.1 평활화

거의 모든 영상에 존재하는 잡영은 잘못된 영역분할이나 움직임 벡터 추정을 초래하는 원인이 된다. 영상을 처리하기 전에 영역 경계선의 선명도를 보존하면서 잡영을 제거할 수 있는 중간값 평활화(median filter)를 이용하여 잡영을 제거한다[13].

중간값 평활화는 2차원 영상의 한 화소를 중심으로 윈도우의 크기를 3×3 이라고 할 때, 9개의 화소의 밝기값을 조사한 후 작은 순서에서 큰 순서로 나열하여 중간값을 새로운 화소값으로 바꾸는 방법이다. 3차원 RGB 공간에서 화소값 X_i 와 점 p 에 인접한 화소의 평균값 \bar{X}_p 를 식 (23)과 같이 정의한다.

$$X_i = [R_i, G_i, B_i]^T, \bar{X}_p = [\bar{R}_p, \bar{G}_p, \bar{B}_p]^T \quad (23)$$

여기서, $\bar{R}_p, \bar{G}_p, \bar{B}_p$ 는 각각 점 p 에 인접한 Red, Green, Blue 화소의 평균값을 나타낸다. 점 p 에 인접한 화소와 평균값과의 거리를 d_i 로 나타내면 식 (24)과 같이 된다.

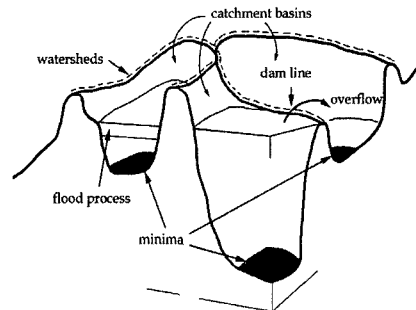
$$d_i = \sqrt{(R_i - \bar{R}_p)^2 + (G_i - \bar{G}_p)^2 + (B_i - \bar{B}_p)^2} \quad (24)$$

점 p 에 인접한 n 개 화소의 평균값에 가장 가까운 화소값은 RGB의 중간값에 근사하다고 할 수 있으므로 식 (25)와 같이 중간값을 구할 수 있다.

$$m_p = \min \{d_i\} \quad (25)$$

3.2.2 영역 분할

영역 분할은 영상을 겹쳐지지 않는 동질 영역들의 집합으로 나누는 기법이라 할 수 있으며 닫혀진 경계를 얻을 수 있으면서 빠른 속도로 전체 영상을 분할할 수 있는 워터셰드를 이용한다.



(그림 5) 워터셰드

3.2.3 움직임 추정

평활화하여 잡영을 완화한 영상에 RGB 색상정보를 이용하여 블록의 움직임을 추정한다. 움직임 벡터 추정을 위한 방법 중 블록정합 방법은 한 블록내의 모든 화소들이 같은

방향으로 평행 이동한다는 가정 아래 후보 블록을 선택하여 블록내 화소값의 정합 에러를 비교하여 가장 차가 적은 블록을 선택하는 기법으로 움직임 추정 방법이 간단하고 하드웨어 구현이 용이하며 예측 효율과 추정의 정확도 등을 고려했을 때 전체적으로 좋은 특성을 보인다.

3.2.4 전역 움직임 추정

움직임 벡터를 분석하여 전역 움직임 중 카메라 회전과 배율을 추정한다. 여러 가지 전역 움직임 추정 기법이 제안되어 있으나 계산시간이 적게 걸리고 계산과정에서 움직이는 물체를 배제할 수 있는 3점 기하분석법을 채택한다[9]. 카메라가 x 방향으로 이동할 때 화소의 움직임 벡터 v_x 는 식 (26)과 같다.

$$v_x = \frac{fT_x}{Z} \tag{26}$$

카메라의 움직임, 원경의 세 점 u_1, u_2, u_3 와 그 움직임 v_1, v_2, v_3 의 관계는 식 (27)과 같다.

$$(sR - I)[u_1 u_2 u_3] = [v_1 v_2 v_3] \tag{27}$$

s 는 배율, R 은 회전 행렬, I 는 단위행렬을 뜻한다. $u_i = [x_i \ y_i \ f]^T$ 로 나타내면 위 식은 다음의 식 (28)로 나타낼 수 있다.

$$\begin{bmatrix} s-1 & -s\Phi & s\Psi \\ s\Phi & s-1 & -s\Theta \\ -s\Psi & s\Theta & s-1 \end{bmatrix} \begin{bmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \\ f & f & f \end{bmatrix} = \begin{bmatrix} v_{x1} & v_{x2} & v_{x3} \\ v_{y1} & v_{y2} & v_{y3} \\ 0 & 0 & 0 \end{bmatrix} \tag{28}$$

$[u_1 u_2 u_3]$ 와 $[v_1 v_2 v_3]$ 의 i 열에 어떤 정수 k 를 곱하여 j 열에 더하는 과정을 수행하여 점집합을 대각행렬로 변환하면 $[u_1 u_2 u_3]$ 의 점들이 서로 정수관계이므로 식 (29)가 구해진다.

$$\begin{bmatrix} s-1 & -s\Phi & s\Psi \\ s\Phi & s-1 & -s\Theta \\ -s\Psi & s\Theta & s-1 \end{bmatrix} \begin{bmatrix} x & 0 & 0 \\ 0 & y & 0 \\ 0 & 0 & f \end{bmatrix} = \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ 0 & 0 & 0 \end{bmatrix} \tag{29}$$

식 (29)로부터 주어진 블록이 원경인지를 판별하는 2개의 판별식 식 (30)이 산출된다.

$$\begin{aligned} \text{판별식 1 : } & yb_{11} = xb_{22} \\ \text{판별식 2 : } & yb_{21} = -xb_{12} \end{aligned} \tag{30}$$

판별식을 통과한 점들의 집합은 다음 식 (31)에 의해 각

각의 전역 파라미터가 구해진다.

$$s = 1 + \frac{b_{11}}{x}, \Psi = \frac{b_{13}}{sf}, \Theta = \frac{-b_{23}}{sf}, \Phi = \frac{b_{21}}{sx} \tag{31}$$

3.2.5 전역 움직임 보상

카메라의 움직임에 의해 영상에 전역적인 변화가 있을 때 카메라 움직임을 반대 방향으로 적용하여 카메라가 움직이지 않았을 경우의 상태로 복원하는 과정을 움직임 보상이라 한다. 움직임 벡터에 카메라 회전과 배율 파라미터들을 반대 방향으로 적용하여 역방향 변환함으로써 카메라 움직임을 보상을 한다. 역방향으로 사상한 움직임 벡터 ($v'_x \ v'_y$)는 식 (32)와 같다. 이 식에서 W 는 영상의 폭(pel 단위)를 뜻한다.

$$\begin{aligned} v'_x &= v_x - sx/W - \psi + \phi y/W \\ v'_y &= v_y - sy/W - \theta - \phi x/W \end{aligned} \tag{32}$$

3.3 객체 블록의 추출과 배경 블록의 깊이 추정

영상을 움직이는 객체에 포함되는 블록과 배경에 포함되는 블록으로 구분하고 배경에 포함되는 블록에 대한 깊이를 추정한다. 먼저 특이점 분석을 하여 특이점인 경우(전후이동이 없는 경우)와 일반적인 경우(전후이동이 있는 경우)를 나누어서 수행한다.

3.3.1 특이점 분석

(그림 6)은 특이점을 분석하여 영상의 모드(mode)를 결정하는 알고리즘이다. 모드 값의 의미는 <표 1>과 같다.

<표 1> 모드 값과 그 의미

모드	의미
0	움직임이 없음
1	전후이동이 없고, 좌우이동만 있음
2	전후이동이 없고, 상하이동만 있음
3	전후이동이 없고, 좌우이동과 상하이동이 모두 있음
4	전후이동이 있음

전역변수 N 은 영상을 구성하는 블록의 개수를 나타내고 지역변수 N_x, N_y 는 각각 X, Y 방향의 움직임 벡터의 값이 0 인 블록의 개수를 나타낸다. N_x 는 전후이동이 없음을 나타내는 블록 쌍의 개수를 뜻하고, n 은 전후이동이 있음을 나타내는 블록 쌍의 개수를 나타낸다. 총 블록의 개수 N 에 대하여 N_x 와 N_y 가 모두 과반수이면 카메라 이동은 없는 것으로 판단하여 모드를 0으로 한다. 대부분의 영상은 이 조건을 만족한다. N_x 는 반수 이하이고 N_y 가 과반수이면 좌우이동만 있으므로 모드를 1로 한다. N_x 는 과반수이고 N_y 가 반수 이하이면 모드를 2로 한다. N_x 와 N_y 가 모두 반수 이하이면 식 (9)의 판별식을 모든 블록 쌍에 적용하여 검사한다. 판별식을 통과한 블록 쌍의 개수 N_2 가 통과하지

못한 블록 쌍보다 많으면 전후이동이 없는 것으로 판단하여 모드를 3으로 하고, 그렇지 않으면 모드를 4로 한다.

```

global N=number of blocks in an image;
mode Singular Point Analysis(vector array V)
{
    Nx=0, Ny=0, Nz=0, n=0;
    for all blocks, do {
        if vx=0 then Nx=Nx+1;
        if vy=0 then Ny=Ny+1;
    }
    if Nx > N/2 and Ny > N/2 then Mode = 0;
    else if Nx ≤ N/2 and Ny > N/2 then Mode = 1;
    else if Nx > N/2 and Ny ≤ N/2 then Mode = 2;
    else {
        for all block pairs (p, q), do {
            if vx(p)vy(q) = vx(q)vy(p)
            then Nz=Nz+1;
            else n = n + 1;
        }
        if Nz > n then Mode = 3; else Mode = 4;
    }
    return Mode;
}
    
```

(그림 6) 특이점 분석 알고리즘

3.3.2 특이점인 경우(전후이동이 없는 경우)

특이점 분석과정에서 결정한 영상의 모드에 따라 수행하는 내용이 결정된다. 특이점인 경우, 모드는 0, 1, 2, 또는 3이 된다. (그림 7)은 이 경우의 블록 깊이 추정 알고리즘을 나타내고 있다. 모드가 0인 경우, 카메라 이동이 없으므로 깊이정보가 없어 모든 블록의 깊이를 0으로 한다.

```

global T1 = threshold value;
depth array Block_Depth_Singular(mode Mode, vector array V)
{
    case Mode {
        0: for all block i, z(i) = 0;
        1: for all block i, calculate z(i) = 1 / (vxvx-1);
        2: for all block i, calculate z(i) = 1 / (vyvy-1);
        3: clear boolean array M;
           repeat {
               for all block i such that M(i)=false,
                   calculate average vx-1 and vy-1;
               for all block i such that M(i)=false,
                   if |vxvx-1 - vyvy-1| > T1
                   then M(i) = true;
           } until not any change in M;
           for all block i such that M(i)=false,
               calculate z(i) = 1 / (vxvx-1);
    }
    return z;
}
    
```

(그림 7) 특이점인 경우의 블록 깊이 추정 알고리즘

모드가 1 또는 2인 경우, 좌우이동이나 상하이동 둘 중 하나만 있으므로 움직이는 객체에 관한 정보가 없어 식 (14)와 (15)을 각각 적용하여 깊이를 산출한다. 모드가 3인 경우, 좌우이동과 상하이동이 모두 있으므로 객체를 추출할 수 있다. M은 블록이 움직이는 객체인지 여부를 나타내는 boolean 배열로서 초기에는 거짓(false)으로 초기화(clear)된다. 먼저 M(i) = false인 모든 블록 i에 관해서 x, y 각 방향의 움직임 벡터의 역수의 평균 $\overline{v_x^{-1}}$ 와 $\overline{v_y^{-1}}$ 를 구하고, 임계값 T₁보다 큰 오차를 갖는 블록은 객체에 속한 블록으로 간주하여 M(i)를 참(true)으로 한다. 임계값 T₁은 실험적으로 결정된다. 끝으로 M(i)가 거짓인 모든 블록에 대하여 식 (14) 또는 (15)을 적용하여 블록의 깊이를 계산한다.

3.3.3 일반적인 경우(전후이동이 있는 경우)

(그림 8)은 전후이동이 없는 경우의 블록 깊이 추정 알고리즘을 나타내고 있다. M은 블록이 움직이는 객체인지 여부를 나타내는 boolean 배열에 관해서 α, β를 구하고, 임계값 T₂보다 큰 오차를 갖는 블록은 객체에 속한 블록으로 간주하여 M(x, y)를 참으로 한다. 임계값 T₂는 실험적으로 결정된다. M의 변화가 없을 때까지 이러한 과정을 반복한다. 다음에는 M(x, y)가 거짓인 모든 블록에 대하여 식 (17) 또는 식 (18)와 식 (20)을 적용하여 γ를 계산한다. 끝으로 M(x, y)가 거짓인 모든 블록에 대하여 식 (21) 또는 식 (22)을 적용하여 z를 계산한다.

```

global T2 = threshold value;
depth array Block_Depth_General(vector array V)
{
    clear boolean array M;
    repeat {
        for all block (x, y) such that M(x, y)=false,
            calculate average α and β;
        for all block (x, y) such that M(x, y)=false,
            if |vy(x+vx-α) - vx(y+vy-β)| > T2 then
                M(x, y) = true;
    } until not any change in M;
    for all block (x, y) such that M(x, y)=false, calculate γ;
    for all block (x, y) such that M(x, y)=false,
        calculate z = (x+vx-α) / γvx;
    return z;
}
    
```

(그림 8) 일반적인 경우의 블록 깊이 추정 알고리즘

3.4 객체 영역의 추출과 배경 영역의 깊이 추정

객체 블록에 관한 정보와 배경 블록의 깊이 정보를 전처리 과정에서 얻어진 영역에 적용하여 영역이 객체에 포함되는지 여부를 판정하고 배경에 포함될 경우 영역의 평균 깊이를 산출한다. 식 (33)은 영역 i에서 움직이는 객체에 속한 화소의 비율을 나타낸다. a_{ij}는 영역 i에 포함된 블록 j의

화소의 개수를 뜻하고, m_j 는 블록 j 가 객체에 포함되는지를 나타낸다. 블록 j 가 객체에 포함되면 $m_j=1$ 이고 아니면 $m_j=0$ 의 값을 가지며, A_i 는 영역 i 의 면적을 나타내는데 그 값은 식 (34)와 같다.

$$M_i = \frac{1}{A_i} \sum_j a_{ij} m_j \quad (33)$$

$$A_i = \sum_j a_{ij} \quad (34)$$

M_i 가 영역 면적의 과반수를 점유하면 영역 i 를 객체에 포함된 영역으로 인정할 수 있으므로 판별식은 식 (35)와 같다. 영역이 배경에 속한 것으로 판정되면, 그 평균 깊이를 계산한다. 식 (36)은 영역 i 의 평균 깊이를 나타낸다. z_j 는 영역 i 에 속한 블록 j 의 깊이를 나타낸다.

$$M_i > \frac{A_i}{2} \quad (35)$$

$$D_i = \frac{1}{A_i} \sum_j a_{ij} z_j \quad (36)$$

```

depth array Region_Depth(region R, depth D)
{
  for all region i, do {
    calculate  $A_i = \sum_j a_{ij}$ 
    calculate  $M_i = \frac{1}{A_i} \sum_j a_{ij} m_j$ ;
    if  $M_i > \frac{A_i}{2}$  then
      region i belongs to a moving object;
    else calculate  $D_i = \frac{1}{A_i} \sum_j a_{ij} z_j$ ;
  }
  return D;
}
    
```

(그림 9) 객체 영역의 추출과 배경 영역의 깊이 추정 알고리즘

(그림 9)는 영역이 객체에 포함되는지 여부와 배경에 포함될 경우 영역의 평균 깊이를 산출하는 알고리즘이다. 진 처리로 얻어진 영역 정보 R 과 3.4절에서 설명한 알고리즘에 의해 얻어진 블록의 깊이 정보 d 를 인수로서 받아 온다. 깊이는 모두 양수의 값을 가지므로 객체 블록은 음수로서 표현함으로써 깊이 정보 d 에는 객체 정보도 포함될 수 있다. 영상의 모든 영역에 관해서 면적을 계산하고 식 (33)의 M_i 를 계산한다. 식 (35)의 판별식을 만족하면 객체 영역으로 판단하고, 그렇지 않으면 배경 영역으로 판단한다. 배경 영역으로 판단되면 식 (36)을 적용하여 영역의 평균 깊이를

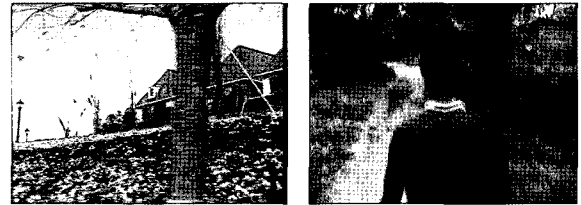
산출한다.

4. 실험 및 결과 분석

3장에서 수립한 알고리즘을 Visual C++ 6.0을 사용하여 펜티엄 컴퓨터에서 실행한다.

4.1 시험 영상 시퀀스

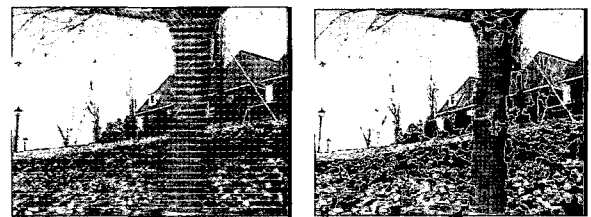
연속적인 1개의 영상 시퀀스를 샷(shot)이라 한다. 본 논문에서는 10개의 프레임은 1개의 샷으로 만들어서 실험을 수행한다. (그림 10)의 (a)와 (그림 10)의 (b)는 10개의 프레임 중 첫 번째 프레임으로서 24비트 트루칼라의 비트맵 영상이며, 720×480 화소의 해상도를 가진다.



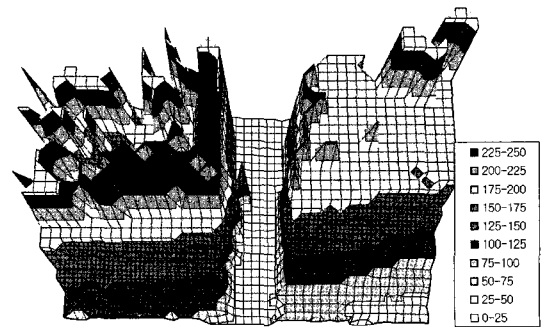
(a) 플라워가든 (b) 아프리카 마을
(그림 10) 테스트 영상

4.2 특이점인 경우의 실험 결과 및 분석

(그림 11)의 (a)는 전후이동이 없는 특이점인 경우에 해당하는 (그림 10)의 (a) 플라워 가든 영상에 대한 움직임 벡터를 화살표로써 표시한 영상이다. (그림 10)의 (b)는 플라워 가든 영상을 인접 화소간의 컬러 값의 유사도에 따라 영역 분할한 영상이다.



(a) 움직임 벡터 (b) 영역분할 결과
(그림 11) 테스트 영상의 처리 결과



(그림 12) 플라워 가든 영상의 깊이 그래프

(그림 12)는 플라워 가든 영상에 대하여 16×16 픽셀의 블록 단위로 상대적인 깊이를 나타낸 그래프로서 상대적인 깊이를 0부터 255까지의 10단계로 구분하여 깊이 정보를 표현하였다. 가로축은 영상 가로축의 블록 번호이고 세로축은 영상 세로축의 블록 번호이다. 그래프의 내용은 각 블록의 깊이 정보를 컬러로 표현하였다. (그림 13)은 (그림 11)에 나타난 블록의 깊이 그래프를 적용하여 영상을 깊이 정보에 따라 그레이 레벨로 표현하였다. 카메라 초점이 가까울수록 어두운 색으로 나타나고 멀어질수록 밝은 색으로 나타난다.



(a) 원영상 (b) 깊이 영상
(그림 13) 플라워 가든 영상의 깊이 영상

(그림 11)과 (그림 12)에서, 원영상의 깊이정보가 대부분 정확하게 표현된 것을 알 수 있다. 왼쪽 위 나뭇가지 부분의 일부 오류는 분할된 영역의 크기가 너무 커서 하늘과 잔가지의 깊이 평균이 영역의 평균으로 계산된 것으로 추정되며, 꽃밭 뒤의 작은 나무들과 가로등은 영역분할은 되었으나 넓은 영역을 차지하는 하늘과 같은 흰색으로 나타난 것으로 추측된다. 하지만 전체적으로 카메라가 좌우이동을 하면서 획득된 영상에 대해 인간이 느끼는 깊이와 아주 근사함을 알 수 있다.

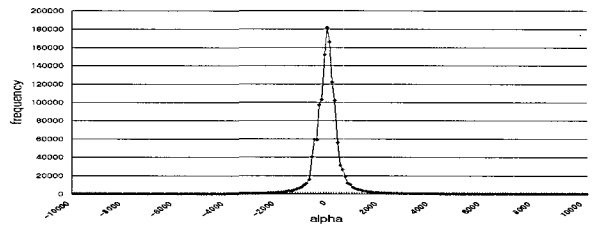
4.3 일반적인 경우의 실험 결과 및 분석

(그림 14)는 전후이동이 있는 일반적인 경우에 해당하는 (그림 10)의 (b) 아프리카 마을 영상에 대한 움직임 벡터를 화살표로써 표시한 영상이다.

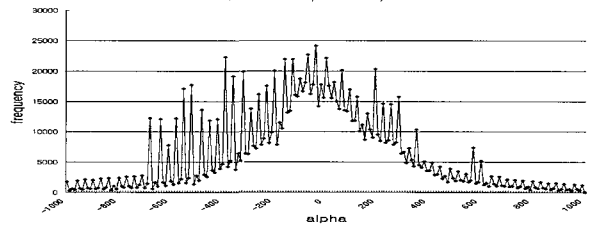


(a) 움직임 벡터 (b) 영역분할 결과
(그림 14) 움직임 벡터와 영역분할 결과

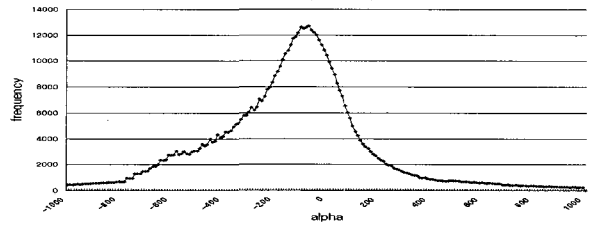
(그림 15)는 아프리카 마을영상에 대한 α 값의 분포도를 나타낸다. (그림 14)의 (a)는 범위를 -10,000에서 +10,000까지로 설정할 경우를 나타내며 그림에서 보는바와 같이 중간 영역에 밀집되어 있어 (b)에서 범위를 -1,000에서 +1,000으로 조정하여 나타내었다. 그림 (c)에서 카메라의 수평이동 성분은 거의 없는 것으로 나타났다.



(a) 범위 : -10,000~10,000

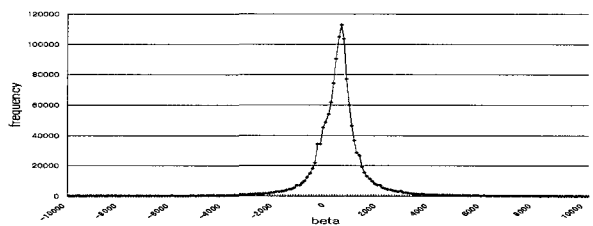


(b) 범위 : -1,000~1,000

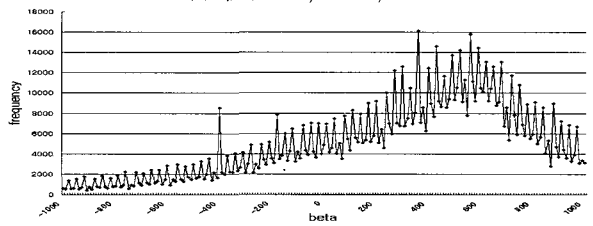


(c) (b)의 평활화

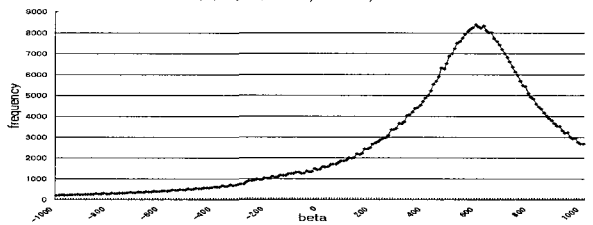
(그림 15) 아프리카 마을 영상의 α 값 분포도



(a) 범위 : -10,000~10,000



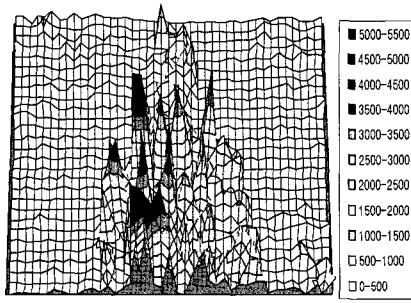
(b) 범위 : -1,000~1,000



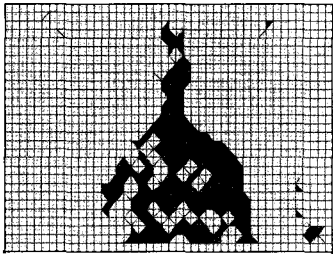
(c) (b)의 평활화

(그림 16) 아프리카 마을 영상의 β 값 분포도

(그림 16)는 아프리카 마을영상에 대하여 β 값의 분포도를 나타낸다. (그림 16)의 (c)는 그림(b)의 분포도를 필터를 이용하여 평활화시킨 결과로 β 값이 700인 지점에서 최대빈



(그림 17) 에러 e의 분포도

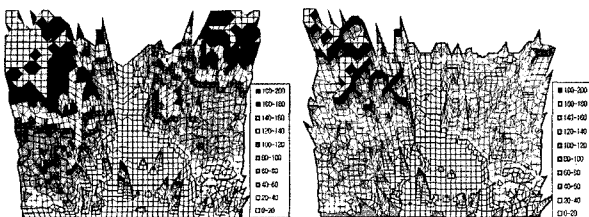


(그림 18) 배경과 물체의 영역

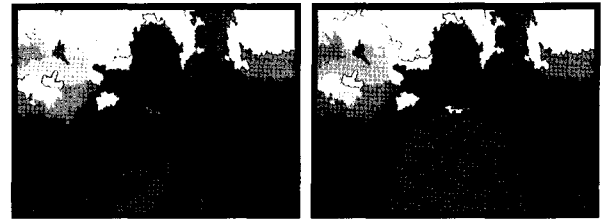
도를 나타내었다. 따라서 (그림 15)와 (그림 16)의 분석 결과 수평이동은 없고 상하이동과 전후이동이 있는 것으로 나타났다.

(그림 17)은 앞에서 기술한 식 (19)의 판별식을 평가하여 평균 오차와 블록에 속한 화소의 움직임 벡터의 오차를 비교하여 블록별로 움직이는 물체를 찾기 위한 그림이다. 각 블록의 오차성분 e 를 3차원 그림으로 나타낸 것이다.

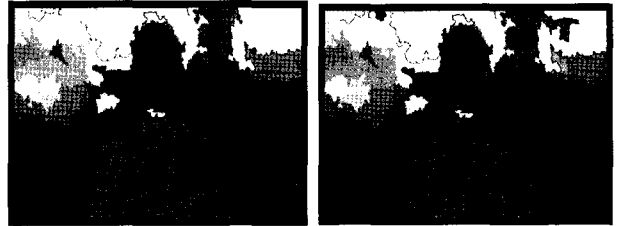
(그림 18)은 (그림 17)에서 오차가 임계값 T_2 보다 큰 블록은 객체블록으로, 작은 경우에는 배경블록으로 구분한 것이다. T_2 는 500으로 설정한다. 평균 오차보다 화소의 움직임 벡터의 오차 값이 크면 움직이는 물체에 속하며 평균 오차보다 화소의 움직임 벡터의 오차 값이 작으면 배경으로 간주한다. 움직이는 사람이 있는 부분은 물체의 영역으로 나타나고 그 외의 영역은 배경 영역으로 나타난 것을 볼 수 있다. 실제 동영상에서 사람이 인식하는 것도 거의 동일한 결과를 나타내고 있음을 알 수 있다. (그림 19)는 x 벡터와 y 벡터를 사용하여 프레임의 평균 깊이에 대한 상대적인 블록의 깊이를 추정한 것으로 실제 영상에서의 깊이와 거의 흡사한 결과가 나왔다. 다만 사람의 몸통부분에서 일부가 길 양쪽의 풀밭과 같은 깊이로 인식된 부분은 움직임 추정단계에서 오류가 발생한 것으로 추정된다.



(a) x 벡터 사용 (b) y 벡터 사용
(그림 19) 블록의 상대적인 깊이



(a) 제 1 단계 (b) 제 2 단계



(c) 제 3 단계 (d) 제 4 단계



(e) 제 5 단계 (f) 마지막 단계
(그림 20) 깊이 추정의 단계별 과정

(그림 20)은 전후이동이 있는 일반적인 영상에 대한 깊이 추정 알고리즘의 단계별 결과를 보여준다.

그림 (f)의 위쪽 부분은 프레임의 이동과정에서 나뭇가지와 배경의 관계가 바뀌어 움직이는 객체로 추출된 것으로 추정되며, 오른쪽의 가운데와 아래 부분은 촬영시 실제 바람에 의해 나뭇가지와 풀의 움직임이 있었던 것으로 추정된다. 또한 사람 왼쪽의 길의 경우 아래에서 위로 올라갈수록 검은색에서 회색으로 나타나야 하는데 길 전체가 하나의 영역에 포함되어 길 전체의 평균으로 계산되어 나타났다. 하지만 전반적으로 원영상에서의 움직임이 있는 물체 즉, 사람 전체를 완전하게 추출하였고, 상대적인 깊이도 우리 인간이 느끼는 깊이와 아주 근사하게 나타났다.

5. 결론

본 논문에서는 단안 영상에 대하여 여러 가지 움직임 추정 기법들 중에서 가장 정확한 움직임 벡터를 구하는 전역 탐색 기법을 도입하여 16×16 픽셀의 블록을 기준으로 하여 프레임간의 움직임 벡터를 구한다. 이러한 움직임벡터의 상관관계를 이용하여 카메라의 움직임을 추출하여 카메라 움직임 성분 중에서 카메라 회전과 배울 성분을 구하여 보상하면 카메라의 이동성분만 남게 된다. 카메라의 이동성분과 깊이와의 기하학적인 관계를 이용하여 카메라의 전후이동이 있는 경우와 없는 경우로 나누어 깊이 정보를 추출한다. 이때 카메라의 이동성분의 비율이 영상 내에서 고르게 분포함을 이용하여 프레임내의 정지영역인 배경영역과 카메라의

움직임에 상관없이 자유로운 움직임에 가지는 물체 영역으로 분할하여 배경영역에 대한 깊이 정보를 반복 기법을 이용하여 추출한다.

본 논문은 단안 카메라로 촬영한 2차원 동영상에 대하여 인간의 시각에 입체감을 느끼게 하는 입체영상화가 최종 목표이며, 2차원 영상의 3차원 영상으로의 변환에 있어 가장 중요한 정보가 되는 깊이 정보를 추출하는 알고리즘을 제안한다. 본 논문에서 사용한 실험영상은 카메라의 움직임을 잘 표현하고 있는 MPEG 표준영상과 DVD로 제작된 일반 영상을 사용하였으며 영상전체를 인간이 느끼는 깊이 감각과 알고리즘의 적용 결과가 얼마나 근사하느냐 하는 문제를 주관적 판단에 따라 평가하였다.

본 논문에서 제안하는 알고리즘은 깊이 정보와 관련된 어떠한 정보도 사전에 필요하지 않으며, 간단한 형태의 수식으로 나타낼 수 있어 처리속도를 향상시킬 수 있으며, 하드웨어 구현에 적합하다는 특징을 가진다.

참 고 문 헌

[1] Tony Jebara, Ali Azarbayejani, Alex Pentland, 3D Structure from 2D Motion, MIT Media Lab, Cambridge MA 02139, 1999.
 [2] Jae-Woong. Yi, Jun-Ho Oh, "Estimation Of Depth And 3D Motion Parameter Of Moving Object With Multiple Stereo Images," Image and Vision Computing 14, pp.501-516, Oct., 1996.
 [3] ITU-T DRAFT H.263, "Video Coding for Low Bitrate Communication," ITU-T, Dec., 1993.
 [4] Borko Furht, Joshua Greenberg, Raymond Wet-water, *Motion Estimation Algorithms For Video Compression*, Kluwer Academic Publishers, 1997.
 [5] A. K. Jain, "Image data compression: A review," *Proc. of IEEE*, Vol.69, No.3, pp.349-389, Mar., 1981.
 [6] T. Koga, K. Iinuma, A. Hirano, Y. Iijima, and T. Ishiguro, "Motion-Compensated Interframe Coding for Video Conferencing," *Proc. of NTC81*, pp.G5.3.1-5.3.5, Nov., 1981.
 [7] W. Lee, Y. Kim, R. J. Gove, and C. J. Read, "Media station 5000: Integrating video and audio," *IEEE Multimedia*, Vol. 1, No.2, pp.1011-1014, Sep., 1985.

[8] Borko Furht, Joshua Greenberg, and Raymond Wetwater, *"Motion Estimation Algorithms for Video Compression,"* KAP, 1997.
 [9] 박준호, "카메라 이동이 있는 단안 영상열에서 움직임 추정 기반의 깊이 정보 추출," 영남대학교 박사학위논문, 2001. 12.
 [10] 편성원, "카메라 이동이 있는 단안 영상에서의 깊이 추정," 영남대학교 석사학위 논문, 2002. 12
 [11] Dae-Seong Jeoune, "Global Motion Estimation for Panning, Tilting, and Zooming Parameters using the Geometry in Digital Video Sequence," Yeungnam University Ph.D. Dissertation, Dec., 2001.
 [12] 문성현, "동영상에서 배경의 세 대칭점을 이용한 카메라 움직임 추정," 영남대학교 박사학위 논문, 2002. 12.
 [13] 유영준, "담수중 경사정보와 색차분산을 이용하여 병합하는 워터셰드 방식의 영상분할," 영남대학교 석사학위논문, 2002. 12.

박 영 민



e-mail : ympark@ikw.ac.kr
 1993년 영남대학교 전산공학과(학사)
 1995년 영남대학교 전산공학과(공학석사)
 2005년 영남대학교 컴퓨터공학과(공학박사)
 1995년~1999년 삼성전자 정보통신연구소
 근무

2000년~현재 경운대학교 멀티미디어공학부 전임강사
 관심분야: 컴퓨터비전, ASIC, 임베디드시스템 등

장 주 석



e-mail : cschang@ikw.ac.kr
 1982년 영남대학교 전자공학과(학사)
 1984년 영남대학교 전자공학과(공학석사)
 1997년 영남대학교 컴퓨터공학과(공학박사)
 1984년~1997년 대구전문대학 전자계산과
 교수

1997년~현재 경운대학교 멀티미디어공학부 교수
 관심분야: 영상처리, 패턴인식 등