

---

# 스트리킹 현상을 감소시키기 위한 다이내믹 프로그래밍 기반의 스테레오 정합 방법

박장호\* · 최현준\*\* · 서영호\* · 김동욱\*

## A Stereo Matching Method Based on the Dynamic Programming to Reduce the Streaking Phenomena

Jang-Ho Park\* · Hyun-Jun Choi\*\* · Young-Ho Seo\* · Dong-Wook Kim\*

---

본 연구는 지식경제부 및 한국산업기술평가관리원의 IT산업원천기술개발사업의 일환으로 수행하였음.  
[K1002058, 대화형 디지털 홀로그램 통합서비스 시스템의 구현을 위한 신호처리 요소 기술 및 SoC 개발]

---

### 요 약

다이내믹 프로그래밍 기반의 스테레오 정합 기법은 전체 영상 또는 한 열의 영상정보를 특정 화소의 정합에 모두 사용하는 전역-대상 기법으로 정합 연산시 폐색영역을 찾을 수 있다는 장점을 가지고 있다. 본 논문에서는 영상의 특징점, 즉 경계정보를 추가로 사용하여 스트리킹 현상을 감소시키고 변이지도의 오차율을 줄이는 방법을 제안한다. 이 방법은 기본적으로 경로선택에 있어서의 페널티를 대상화소의 주변 화소들에 따라 조정한다. 또한 경계정보를 사용하여 특정 화소에 대한 신뢰성을 재검사하는데, 이 신뢰성 재검사는 역추적과정에서 실시한다. Middlebury에서 제공하는 네 쌍의 영상으로 실험한 결과 제안한 기법의 에러율을 6.33% 29위에 랭크됐다. 이 결과는 이전에 제안된 다이내믹 프로그래밍 기반의 정합 기법 중 가장 좋은 결과이다.

### ABSTRACT

The dynamic programming based methods, a kind of globally optimizing stereo matching methods, has the inherent advantage that the occlusion regions can be found during the process. But it also has a serious drawback of streaking phenomena. This paper focuses on reducing the streaking phenomena by adjusting the penalties in calculating the cost matrix and re-establishing the optimal path in the back-tracing process using the boundary information of the image. Especially we use a pixel expansion method in re-establishing the path, which is the results from expanding the pixel information of the ones just left the boundaries. Experiments with the four image pairs provided by the Middlebury site showed the results that the proposed method has the disparity error ratio of 6.33% and the rank is 29, which is competitive to the best method among the previously published dynamic programming based methods.

### 키워드

다이내믹 프로그래밍, 스테레오 정합, 시차, 스테레오 영상

### Key words

Dynamic Programming, Stereo Matching, Disparity, Stereo Image

---

\* 광운대학교 실감미디어 연구소 (jhpark@kw.ac.kr)

접수일자 : 2010. 01. 14

\*\* 안양대학교 정보통신공학과

심사완료일자 : 2010. 02. 20

## I. 서 론

스테레오 정합(sterео matching)은 컴퓨터 비전 등의 분야에서 아주 오래전부터 연구되어온 분야이며, 최근 HD영상 이후의 차세대 영상으로 3차원 영상이 대두되면서 이 분야에 대한 연구가 더욱 활발히 진행되고 있다 [1]. 스테레오 정합은 두 대의 평행식 또는 수렴식 카메라로 동시에 획득된 좌·우 두 개의 영상을 대상으로 한 영상의 각 화소에 대응되는 점을 다른 영상에서 찾는 것을 뜻하며, 그 결과로 좌·우 영상간의 변이(disparity, 이것을 시차라고 하기도 함)를 얻을 수 있다. 이 변이는 카메라 파라미터를 포함하는 간단한 수식으로 카메라로부터 특정 점까지의 거리(깊이(depth))를 구할 수 있다. 이 깊이정보는 2차원 또는 3차원 자유시점(free view point 또는 autostereoscopy) 영상을 실현하기 위한 중간 시점영상이나 디지털 홀로그래프 생성 등에 사용되며, 최근 MPEG에서는 3차원 영상정보를 송신하는 방법으로 2차원 영상과 깊이정보를 부호화하는 방법들이 연구되고 있다[2].

2차원 영상의 모호성, 두 영상의 변이로 인해 특정 영상에는 존재하나 다른 영상에는 존재하지 않는 영역(폐색 영역, occluded region), 변이정보의 불연속 현상 등으로 정확한 변이정보를 구하는 것이 매우 어려운 과정으로 알려져 있다<sup>[1]</sup>. 따라서 지금까지 이 스테레오 정합 문제를 해결하기 위한 많은 연구들이 진행되어오고 있다. 이 연구들은 크게 지역(local)-대상 방법<sup>[3][17]</sup>과 전역(global)-대상 방법<sup>[8][20]</sup>으로 나눌 수 있다. 지역-대상 방법은 특정 화소의 정합점을 찾을 때 그 화소 주변의 일정한 지역의 정보를 사용하는 방법으로, 일반적으로는 정합창(matching window)을 설정한다. 이 방법에는 특정 화소의 위치에 따라 이 정합창의 크기를 적응적으로 가변시키는 방법<sup>[3]</sup>, 영상의 특징을 사용하는 방법<sup>[4]</sup>, 영상을 그 특징에 따라 분할하는 방법<sup>[5]</sup>, 화소의 해상도를 확장하여 정확도를 높이는 방법<sup>[6]</sup> 등이 연구되고 있다. 전역-대상 방법은 기본적으로 전체 영상 또는 한 열의 영상 정보를 특정 화소의 정합에 모두 사용하는 방법이며, 여기에는 신뢰도 함수를 정의하고 이를 기반으로 하는 전역 비용함수를 정의하여 이 함수를 최적화시키는 해를 찾는 belief propagation 방법<sup>[9]</sup>, graph 이론의 graph-cut을 이용하는 방법<sup>[9]</sup>, 비용함수를 경로에 따라 누적하여 최

적의 경로를 찾는 dynamic programming(동적계획법, DP)<sup>[10][20]</sup> 등이 있다. 또한 전역방법과 지역방법의 장점들을 모두 수용하고자 하는 연구<sup>[21]</sup>도 진행되고 있다. 이 중 현재까지는 belief propagation 방법이 가장 우수한 성능을 보이는 것으로 보고되고 있다<sup>[22]</sup>. 그러나 이 방법과 graph-cut을 이용하는 방법은 상대적으로 연산량이 많은 단점을 가지고 있다.

동적계획법은 한 열의 정보를 모두 사용하는 전역방법으로, 상대적으로 일찍 연구가 시작된 분야이다. 이 방법은 변이정보를 획득하는 과정에서 폐색영역을 포함한 변이의 불연속 영역을 파악할 수 있다는 장점이 있는 반면, 변이의 연속성이 강조되는 이 방법의 특성 때문에 변이가 급변하는 지점에서 그 전의 변이값을 연속키는 스트리킹(streaking) 현상이 발생하는 단점을 갖고 있다. 이러한 단점을 극복하기 위해서 영상의 경계화소들이나 폐색영역의 화소들을 구해서 사용하는 방법<sup>[10][11][12]</sup>, 스캔라인(scanline) 간의 정보를 획득하기 위해 영상전체 또는 중요도를 감안한 화소 tree를 형성하는 방법<sup>[13][14]</sup>, 영상을 분할하여 각 분할영상 내에서의 비용함수를 주변의 분할영상으로 확대하는 region-tree 방법<sup>[15][16][17]</sup>, 필요한 조건을 부여하여 특정 화소에 특정 변이값을 가지는 경로와 그 외의 경로에 대한 신뢰도를 분석하여 정해진 신뢰도 이상을 가지는 경우만 변이를 부여하는 신뢰도-기반 방법<sup>[18][19][20]</sup> 등이 연구되어 왔다. 이들 방법은 중요도에 대한 정의(tree-기반 DP), 영상분할 방법(region-tree DP), 여러 경로를 찾기 위해 부여하는 조건(신뢰도-기반 DP) 등에 동작속도가 매우 의존적이며, 이 속도를 높이기 위해 GPU(graphic processing unit)를 사용하기도 하였다<sup>[19]</sup>.

본 논문에서는 원래의 DP 방법을 그대로 유지하면서 영상의 특징점, 즉 경계정보를 추가로 사용하여 스트리킹 현상을 감소시키고 변이지도(disparity map)의 오차율을 줄이는 방법을 제안한다. 이 방법은 기본적으로 경로선택에 있어서의 페널티(penalty)를 대상화소의 주변 화소들에 따라 조정한다. 또한 경계정보를 사용하여 특정 화소에 대한 신뢰성을 재검사하는데, 이 신뢰성 재검사는 역추적과정에서 실시한다. 이 때 경계부분의 좌측 화소들의 정보를 확장하여 생성되는 확장영상을 사용하여 신뢰성 검사를 다시 실시한다. 제안한 방법은 Middlebury site<sup>[22]</sup>에서 제공하는 테스트 영상들에 적용하여 Middlebury site에서 기존 방법들과 오차율을

비교한다.

## II. 동적계획법과 스트리킹 문제

기본적으로 동적계획법(DP)은 기준영상의 한 스캔라인을 대상으로 수행하며, 그 기본적인 방법과 순서는 개략적으로 그림 1과 같다. 즉, 좌·우 영상을 대상으로 특정 비용함수를 사용하여 변이공간영상(disparity space image, DSI)를 구하고 이 비용함수 값을 누적 연산하는 비용행렬(cost matrix)을 계산한다. 이 비용행렬의 값을 기반으로 역추적(back-tracing)을 수행하여 한 스캔라인에 대한 변이값들을 결정하며, 동일한 방법을 모든 스캔라인에 대해서 수행하여 한 영상에 대한 변이지도를 완성한다.

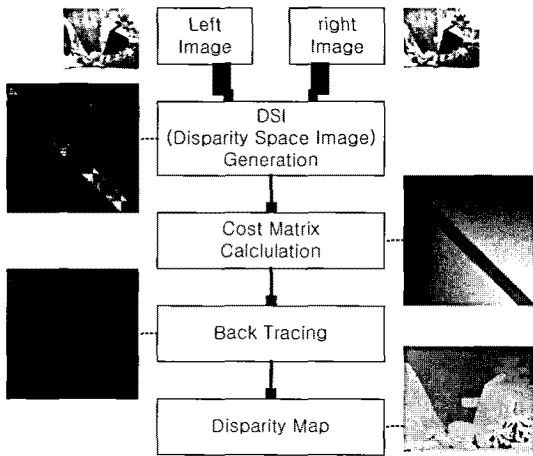


그림 1. 동적계획법에 의한 스테레오 정합 과정  
Fig. 1. Stereo matching procedure by dynamic programming

### 2.1. 변이공간영상

스테레오 정합은 기준영상의 특정 화소와 1:1 대응되는 참조영상의 화소를 찾는 것이다. 그러나 화소 1:1의 대응은 일반적인 영상에서 동일한 값을 가진 화소가 다수 개 존재하기 때문에 정확한 정합이 이루어지기 어렵다. 따라서 일반적으로 특정 화소에 대한 정보량을 늘리기 위해서 주변 화소까지 사용하는데, 이 때 일정

한 모양과 크기의 주변 화소들을 사용하며, 이 크기와 모양을 정합창(matching window)이라고 한다. 기준영상의 특정화소를 기준으로 참조영상의 그 지점으로부터 일정한 범위 내의 화소까지 정합창 내의 화소들을 정합하게 되는데, 이 범위를 정합구간(matching range)라 한다. 정합방법은 특정 비용함수(cost function)를 사용하는데, 화소간의 차이값을 사용하는 방법들과 상관도를 사용하는 방법들 중 하나를 사용한다<sup>[1]</sup>. 본 논문에서는 SAD(sum of absolute differences)를 사용하며, 이 함수는 식 (1)과 같다.

$$SAD(u, v, d) = \sum_{(i,j) \in MW} |I_L(u+i, v+j) - I_R(u+i+d, v+j)| \quad (1)$$

여기서  $SAD(u, v, d)$ 는  $(u, v)$ 의 화소가 변이  $d$ 를 가질 때의 SAD값을 나타내며,  $I_L(i, j)$ 과  $I_R(i, j)$ 는 각각 좌영상(기준영상)과 우영상(참조영상)의  $(i, j)$  화소의 값(intensity)이고  $i$ 와  $j$ 는 정해진 정합창(MW) 내의 모든 화소를 나타낸다. 따라서 이 함수값이 작을수록 정합될 확률이 높다.

DP에서 변이공간영상을 생성할 때에도 같은 방법을 사용한다. 정합구간의 변이 중 이 함수값이 가장 작은 변이를 그 화소의 변이로 결정하는 WTA(winner takes all) 방법과는 달리 DP에서는 정합구간의 모든 변이에 대한 비용함수 값들을 모두 저장하는데, 이것을 변이공간영상(DSI)라고 한다. 그림 1에 보인 DSI는 종축을 좌영상, 횡축을 우영상 화소들로 배열한 DSI를 나타내고 있다. 그림에서 보는 바와 같이 특정 좌영상 화소는 정합구간에 해당되는 비용함수 값만 가지는 것을 확인할 수 있다.

### 2.2. 비용행렬

비용행렬(cost matrix)은 앞에서 구한 DSI의 비용함수 값들을 대상으로 경계성분을 고려한 비용값과 변이의 연속성을 고려한 비용값을 사용하여 비용값들의 누적 계산을 수행한 결과이다. 그림 2는 비용행렬을 계산하는 방법을 도식화한 것으로,  $p$ 점, 즉  $(i, j)$  위치의 비용행렬 값을 계산하는 방법을 나타내고 있다. 좌상측에서부터 우하측으로의 방향이 정해진 경우  $p$ 점에 영향을 줄 수 있는 값들은 그림에서 a, b, c의 세 값이다. 이 세 값으로

부터 누적계산에 의해 p점의 비용행렬 값을 결정하는 기본적인 방법은 식 (2)와 (3)과 같다.

$$\begin{aligned}
 C_b &= Cost(i, j-1) + Occ_{penalty} & (2) \\
 C_c &= Cost(i-1, j-1) + DSI(p) \\
 C_a &= Cost(i-1, j) + Occ_{penalty}
 \end{aligned}$$

$$Cost(p) = \min(C_a, C_b, C_c) \quad (3)$$

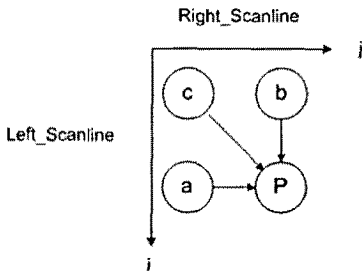


그림 2. 비용행렬 값 계산 방법  
Fig. 2. The method to calculate cost matrix value

식 (2)에서  $C_a, C_b, C_c$  는 각각 수평, 수직, 대각선 방향의 누적연산을 나타내며, 대각선 방향의 경로는 DSI 상의 값이 누적되나 수평 또는 수직방향의 경로에 대해서는 DSI 상의 값 대신 특별한 페널티 값( $Occ_{penalty}$ )을 부여한다. 이와 같이 계산된 비용행렬의 값 중 최소값을 최적화된 disparity 값으로 선택한다(식 (3)). 이 페널티 값의 크기는 대각선 방향의 경로와 수평/수직방향의 경로 중 어떤 것을 택할 것인가에 대한 가중치로 볼 수 있다.

### 2.3. 역추적

비용행렬을 계산하는 식 (2)와 (3)을 살펴보면 이미 식 (3)에서 최적의 경로가 결정된다. 만약 비용행렬의 각 위치에서 그 전으로부터의 진행방향을 저장하였다면 최종의 변이값은 역방향으로 경로를 추적하여 쉽게 얻을 수 있으며, 이 과정을 역추적(back-tracing) 과정이라고 한다. 역추적 과정에서 변이값은 식 (4)의 간단한 방법으로 결정된다.

$$D(i) = \arg_{i-j} \min(C_a, C_b, C_c) \quad (4)$$

그림 3에서 역추적 과정의 예를 보이고 있다. 이 그림에서 비용행렬의 각 위치에 표시한 값을 비용행렬 값이 최소인 경로의 방향을 가리키는데, '1'은 수평방향, 2는 대각선 방향, 3은 수직방향을 나타낸다.

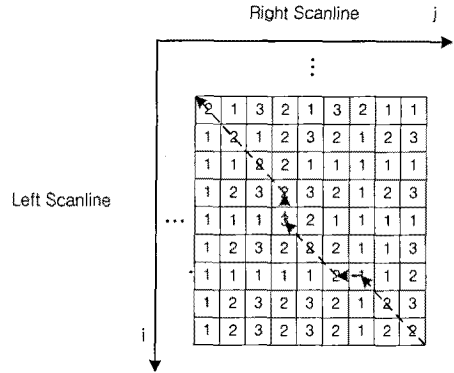


그림 3. 역추적 과정  
Fig. 3. Back-tracing process

역추적 과정에서 수평(수직)방향으로 한 화소 이동하는 것은 변이값이 1 증가(감소)하는 것을 의미한다. 수평(수직)방향의 경로는 그 특성 상 폐색영역일 가능성이 높다. 이론적으로는 수평방향 이동은 좌영상이 기준영상일 때의 폐색영역, 수직방향 이동은 우영상이 기준영상일 때의 폐색영역(이 영역을 좌영상을 기준으로 비폐색(disocclusion)영역이라고 표기하기도 함)을 포함하게 된다. 따라서 DP는 그 특성 상 역추적 과정에서 폐색영역의 가능성이 있는 영역을 모두 검출할 수 있으며, 이것이 DP의 장점이기도 한다.

### 2.4. 스트리킹 현상

식 (2)에서 부여하는 페널티 값의 크기는 역추적 과정에서의 경로선택에 큰 영향을 미친다. 이 값이 상대적으로 작으면 수평 또는 수직방향의 경로가 선택될 가능성이 높은 반면, 이 값이 큰 경우에는 대각선 방향의 경로가 선택될 가능성이 더 높아진다. 일반적인 영상은 동일한 변이(또는 깊이)를 갖는 화소들이 훨씬 많기 때문에 이 페널티 값은 충분히 큰 값을 택하는 경우가 많다. 이 경우 물체의 경계부분이나 변이값이 변하는 부분에서 페널티 값이 과다하면 수평 또는 수직으로 진행되어야

하는 영역에서 대각선 방향으로 경로가 결정되는 오류가 발생된다. 이것은 역추적 방향으로 물체의 경계부분에서 그 전의 변이값이 연속되는 현상을 보이게 되는데, 이 현상을 스트리킹(streaking)이라 하며, 그림 4에 예를 보이고 있다.

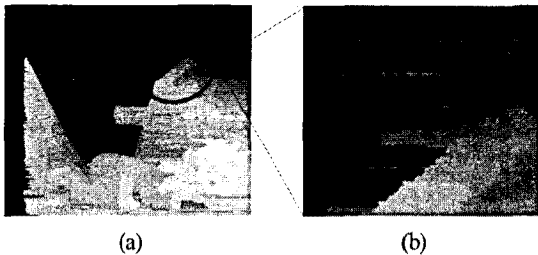


그림 4. 스트리킹; (a) 전체 변이영상  
(b) 확대한 변이영상.

Fig. 4. Streaking; (a) an entire disparity image, (b) enlarged disparity image

스트리킹 현상은 이전 경로의 영향을 받아 현재 경로를 결정짓기 때문에 화소 변화가 적은 부분이나 폐색 영역에서 현재 경로를 정확히 판단하지 못하고 이전의 경로를 그대로 유지하는 성질 때문에 발생한다. 이 스트리킹 현상 또한 일반적인 영상의 특징을 반영한 DP의 고유한 특성이라 할 수 있는데, 최근의 연구는 이 문제를 해결하는데 많이 치중하고 있다.

### III. 경계정보를 이용한 다이내믹 프로그래밍

앞장에서 밝힌 스트리킹 문제 등을 해결하기 위한 최근의 연구는 한 개의 스캔라인의 대상영역을 더 확장하여 전역범위를 확대하는 방향으로 진행되고 있다. 그러나 본 논문에서는 한 스캔라인을 범위로 하는 DP 원래의 방법을 그대로 사용한다. 단, 추가 정보를 사용하기 위해서 본 논문에서는 영상의 특징점, 즉 경계정보를 사용하는데, 이것은 이 장의 중반에서 설명한다.

제안하는 방법의 개략도를 그림 5에 나타내었다. 그림에서 보는 바와 같이 제안하는 방법은 DSI 생성, 비용행렬 연산, 역추적의 기본적인 DP 구성요소를 그대로 사

용하고 있으며, 후처리 과정이 추가된다. 각 요소에 대한 설명은 다음과 같다. 단, 향후의 설명은 좌측영상을 기준 영상으로 한 경우로 가정한다.

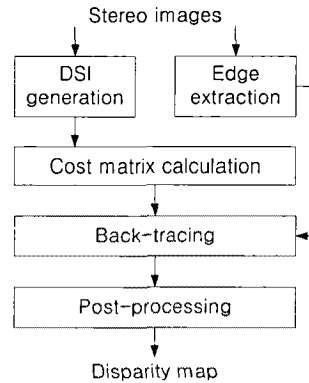


그림 5. 제안한 방법의 블록도

Fig. 5. Block diagram of the proposed method

#### 3.1. 변이공간영상 생성

DP의 시작은 DSI를 구하는 것이다. 이것은 정해진 정합구간 내의 모든 변이에 대해 비용함수의 값을 구하는 것이다. 이 때 일반적으로는 정해진 크기의 정합창을 사용하며, 각 화소의 값은 R, G, B 성분 중 한 개 또는 모두를 사용하거나 Y, Cb, Cr로 변환한 후 Y성분을 사용하기도 한다. 본 논문에서는 정합창의 크기를 3×3으로 하고 Y성분을 화소값으로 사용한다. 정합구간은 영상에 따라 다른데, 4장에서 자세히 설명하도록 한다. 앞에서 언급한 바와 같이 정합에 사용되는 함수는 SAD함수이다.

#### 3.2. 비용행렬 연산을 위한 페널티 조정

앞장에서 설명한 스트리킹 현상은 비용행렬을 계산할 때 변이값이 연속되는 경로(변이연속경로, disparity-consecutive path)에 대한 누적비용값 계산이 올바르게 못하기 때문에 발생한다고 볼로(변이다. 즉, 비용행렬 연산에 사용되는  $\hat{Y}$ 번(2)에서 수평 또는 수직방향의  $Occ_{penalty}$ 에 비해 대각선 방향의  $DSI(p)$ 가 너무 작기 때문으로 볼 수 있다.

DSI와 식 (2)에 의한 비용행렬 연산 결과 데이터를 분석한 결과 식 (2)에서  $Occ_{penalty}$ 의 값을 낮게 책정하면 평

활한 지역에서의 오류가 많이 발생하고,  $\alpha_{c_{penalty}}$  를 높게 책정하면 경계부분에서 오류가 많이 발생하였다. 즉,  $\alpha_{c_{penalty}}$  또는  $DSI(p)$  값을 일반적으로 조정하는 것으로 스트리킹 현상을 감소시키는 것에는 한계가 있다는 것이다. 따라서 기본적으로는 수평/수직방향과 대각선 방향의 페널티를 연동적으로 조정할 필요가 있다. 이에 본 논문에서는 식 (5)와 같이  $\alpha_{c_{penalty}}$  값을 고정시키고 대각선 방향의 페널티를 스케일링 요소  $\lambda$ 를 사용하여 연동시킨다. 앞에서와 같이 여기서는 수직경로 페널티 ( $\alpha_{c_{penalty,ver}}$ )와 수평경로 페널티 ( $\alpha_{c_{penalty,hor}}$ )는 동일한 값을 적용한다.

$$\begin{aligned} C_b &= Cost(i-1, j) + \alpha_{c_{penalty,ver}} & (5) \\ C_c &= Cost(i-1, j-1) + \lambda \cdot DSI(p) \\ C_a &= Cost(i-1, j) + \alpha_{c_{penalty,hor}} \end{aligned}$$

이 식에서  $\lambda$ 는 식 (6)과 같이 현재 화소와 좌측 화소들의 차이값으로 결정된다.

$$\lambda = \begin{cases} q & : DP(i, j) = 0 \\ 1 & : 0 < DP(i, j) < T_{DP} \\ r & : DP(i, j) \geq T_{DP} \end{cases} \quad (6)$$

$$DP(i, j) = \sum_{k=1}^p |I(i, j) - I(i-k, j)| \quad (7)$$

여기서  $I(i, j)$ 는  $(i, j)$ 의 원영상 화소값이며,  $0 < q < 1, 1 < r < 2$ 이다. 즉, 현재 화소의 좌측 p개의 화소값과 현재 화소의 차이값의 크기에 따라 스케일링 요소값을 변화시킨다. 이 때 사용되는 문턱치  $T_{DP}$ 는 실험에 의해 구한 값으로, 본 논문에서는 30을 사용하였다. 식 (5)에서 q와 r 또한 실험에 의해 결정하는 값이다. 본 논문에서는 실험적으로 결정한  $p=2, q=0.4, r=1.8$ 을 사용하였다.

### 3.3. 경계정보를 이용한 역추적 과정

일반적인 DP 방법은 앞 절에서 구한 비용행렬을 사용하여 맨 우측 화소부터 역방향으로 최적경로를 추적하여 식 (4)에 따라 변이값을 결정한다. 그러나 본 논문에서는 역추적을 수행할 때 비용행렬에 의한 경로의 신뢰도를 다시 한 번 평가한다. 이 때 원영상의 경

계정보를 사용하는데, 경계정보는 다음과 같이 추출한다.

#### 3.3.1. 경계정보 추출

영상의 경계정보, 즉 윤곽선 정보를 추출하는 것은 매우 오래된 문제이고, 이 문제에 대한 여러 가지 방법 또한 발표되어 왔다[23]. 본 논문에서도 이 중 하나의 방법을 사용한다. 단, 본 논문에서는 영상을 네 가지의 부영상(sub-image)로 분할하여 각 부영상에 대해 경계정보를 추출한다. 주어진 영상을 네 개의 부영상으로 분리하는 방법은 식 (8)과 같다.

$$\begin{aligned} S_R &= \{p(I_R, I_G, I_B) | (I_R \geq I_G + T_S) \cup (I_R \geq I_B + T_S)\} & (8) \\ S_G &= \{p(I_R, I_G, I_B) | (I_G \geq I_B + T_S) \cup (I_G \geq I_R + T_S)\} \\ S_B &= \{p(I_R, I_G, I_B) | (I_B \geq I_R + T_S) \cup (I_B \geq I_G + T_S)\} \\ S_{II} &= \left\{ p(I_R, I_G, I_B) | (|I_R - I_G| < T_S) \cap (|I_G - I_B| < T_S) \right. \\ &\quad \left. \cap (|I_B - I_R| < T_S) \right\} \end{aligned}$$

이 식에서  $p(I_R, I_G, I_B)$ 는 특정 화소의 R, G, B 성분의 크기값( $I_R, I_G, I_B$ )이 주어진 조건을 만족하는 화소를 의미한다. 즉, 식 (8)의  $S_R, S_G, S_B$ 의 세 부영상은 상대적으로 R, G, B 성분이 강한 화소들의 집합이고,  $S_{II}$ 는 R, G, B의 우세정도가 미약한 화소들의 집합이다. 여기서  $T_S$ 는 상대적인 우세성을 판단하기 간한 문턱치로, 본 논문에서는 7로 설정하였는 이렇게 분리된 부영상들을 합치면 원래의 영상을 얻을 수 있다.

이 부영상들 각각에 경계정보를 추출하는 방법을 적용하여 각 부영상에서의 경계정보( $E_R, E_G, E_B, E_{II}$ )를 획득한다. 그림 6에 Teddy 영상에 대한 네 개의 부영상과 각 부영상에 대해 추출한 경계정보의 예를 보이고 있다. 참고로, 네 부영상에 대한 경계정보를 모두 합하면 원영상에 대해 경계정보를 추출한 결과와 같다.

#### 3.3.2. 확장영상

일반적인 DP의 역추적 과정은 비용행렬 연산 때 이미 최소값에 의해 확정된 경로를 따라 변이값을 계산하는 것이다. 그러나 본 논문에서는, 앞에서 페널티 값을 위치에 따라서 조정하기는 하였으나 역추적 과정

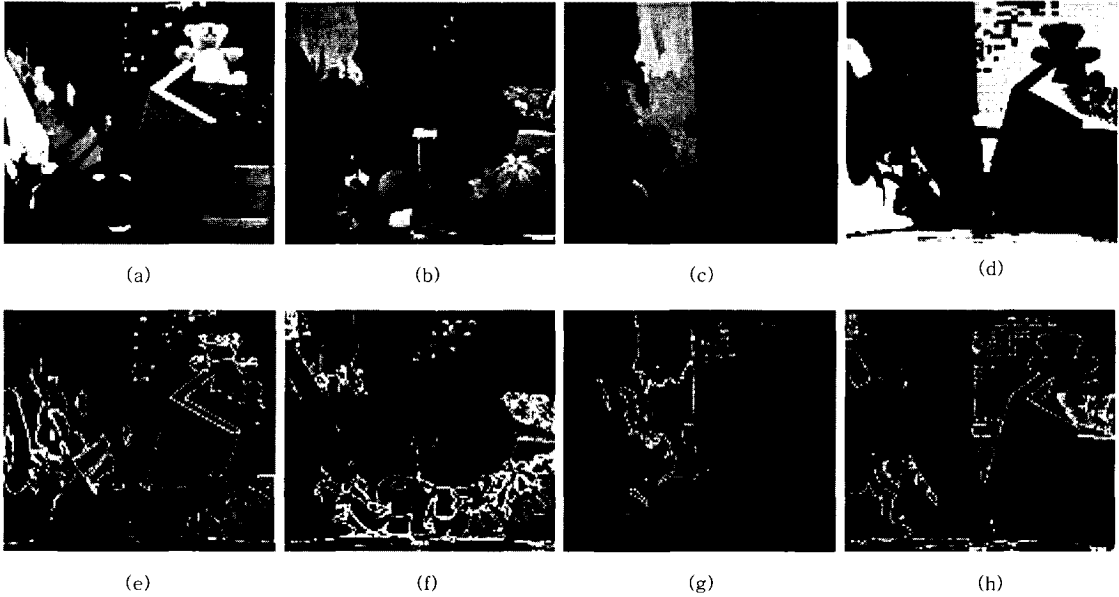


그림 6. 부영상 생성 및 경계정보 획득; (a)  $S_R$ , (b)  $S_G$ , (c)  $S_B$ , (d)  $S_{II}$ , (e)  $S_R$ 의 경계정보( $E_R$ ), (f)  $S_G$ 의 경계정보( $E_G$ ), (g)  $S_B$ 의 경계정보( $E_B$ ), (h)  $S_{II}$ 의 경계정보( $E_{II}$ ).

Fig. 6. Sub-image generation and boundary information extraction; (a)  $S_R$ , (b)  $S_G$ , (c)  $S_B$ , (d)  $S_{II}$ , boundary information of (e)  $S_R(E_R)$ , (f)  $S_G(E_G)$ , (g)  $S_B(E_B)$ , (h)  $S_{II}(E_{II})$ .

에서 그 신뢰도를 다시 점검하고 필요하면 경로를 재 탐색한다.

이 재탐색과정에서, 그림 7에 나타낸 것과 같이, 영상을 확장하여 사용하는데, 이것은 물체의 좌측 경계에서만 적용한다. 그림 7(a)와 (b)에서 보는 바와 같이 우측영상에는 폐색영역에 대한 정보가 없을 뿐만 아니라 폐색영역 때문에 그 영역 좌측의 영상정보들이 우측영상의 정보에 비해 그 위치가 우측으로 이동된다. 이것이 변이이고 이 변위를 찾는 것이 정합의 목적이기는 하지만, 좌영상과 우영상은 2차원 영상들이기 때문에 동일한 정합창을 적용하였을 때 폐색영역 뿐만 아니라 변이 그 자체가 정합을 어렵게 하는 원인이 된다. 따라서 정합을 수행할 때 폐색영역에 해당하는 우측 영상 영역을 찾기는 어렵지만 폐색영역을 제외한 나머지 영상영역은 좌영상과 같은 위치로 이동시켜 정합하는 것이 정확한 정합이 이루어질 가능성이 높다.

그래서 본 논문에서는 우영상에서 물체의 좌측 경계

면 근처의 정합을 수행할 때 그림 7(c)와 같이 물체의 좌측 경계 바로 좌측 화소들을 수평방향으로 확장함으로써 폐색영역 좌측의 영상정보를 정확한 위치로 이동시킨다. 영상정보를 확장하는 방법은 그림 8에 나타내었다. 확장영역은 정해진 정합창의 범위이며, 정합창이 물체의 좌측 경계를 넘어가면 수평방향으로 한 화소씩 영상을 확장한다.

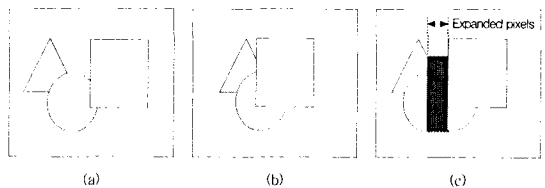


그림 7. 확장영상; (a) 좌영상, (b) 우영상, (c) 확장된 영상

Fig. 7. Image expansion: (a) left image, (b) right image, (c) expanded image

```

Process{Horizontal expansion}{
  define{
    matching window at (i, j):=([i-n, i+n], [j-m, j+m]);
    left boundary of an object at y=j:= (bj, j);
  }
  for pixel at (i-n, j+k)(k=-m to +m, k++){
    if (i-n < bk+i) then
      expand I(bk+i, j+k)
      horizontally for one pixel;
    }
  }
}
    
```

그림 8. 영상정보 확장  
Fig. 8. Expansion of image information

3.3.2. 확장영상을 이용한 경로 재탐색

DP-기반 스테레오 정합에 있어서 문제가 될 수 있는 경우는 대각선방향 경로(변이연속 경로)와 수평경로이다. 앞에서 언급한 바와 같이 변이연속경로는 그 경로가 잘못된 경우 스트리킹을 일으키고, 수평경로는 폐색영역의 가능성이 있다. 따라서 본 논문에서는 역추적 과정에서 이 두 경우의 신뢰성을 점검하여 신뢰도가 낮은 경우 확장영상을 이용하여 경로를 재탐색한다.

(1) 변이연속 경로

변이연속 경로, 즉 대각선 방향의 경로에 대해서는 스트리킹의 가능성을 재점검하고 가능성이 높은 경우 경로를 재설정하는데, 그 방법을 그림 9에 나타내었다. 역방향 추적을 진행하다가 변이연속경로를 만나면 이 경로에 대한 신뢰성 검사를 실시한다. 일반적으로는 그 경로에 대해 비용함수 등을 이용하여 그 경로의 신뢰도 정도를 검사하나, 본 논문에서는 이미 경계정보를 추출하였기 때문에 이 정보를 이용한다. 이 때 특정 화소에 대한 경계정보는 그 화소가 속한 부영상의 경계정보를 사용한다.

즉, 변이연속경로에 대해 그 부영상의 경계정보 중 물체의 좌측 경계인지를 확인하여 경계를 지나치지 않은 경우는 그대로 진행하고, 경계를 지나친 경우 그 화소의 정합범위까지 경계 좌측 화소들을 확장하여 SAD연산을 수행한다.

```

Process{Re-establishment of Diagonal Path} {
  define {
    number of horizontal expansion:= ε;
    matching window at (i, j, d):= ([i+d-n, i+d+n], [j-m, j+m]);
    search range:= r;
    left boundary of an object at y=j in Es(S=R, G, B, W):=(bj, j);
    SADmin = ∞;
  }
  if path[(i+d-1, j)→(i+d, j)] is diagonal then {
  /*
    if i=bj then {
      for (ε=1 to r-d, ε++) {
        Horizontal Expansion;
        SADmin=min[SAD(i, j, d+ε), SADmin];
      }
      if SADmin < TSAD,back then {
        q=arg.(SADmin);
        put all Occpenalty,hor=0 from ε=0 to q;
        re-calculate the cost matrix values for the paths;
        re-perform the back-tracing from ε=0 to q;
      }
      else
        define (i, j) pixel as mismatched;
      */
    }
  }
}
    
```

그림 9. 변이연속경로의 재설정  
Fig. 9. Re-establishment of the disparity-consecutive path

그 중 최소값을 찾아 그 값이 주어진 문턱치 ( $T_{SAD,back}$ )보다 작으면 그 때까지 경로의  $Occ_{penalty,hor}$ 을 모두 0으로 하여 비용행렬을 다시 계산하고, 역추적 과정을 다시 수행한다. 만약 SAD 최소값이 주어진 문턱치보다 크면 이 화소는 정합되지 않는 것(오정합)으로 간주한다. 오정합된 화소에 대해서는 다음 절의 후처리 과정에서 다시 처리한다.



(2) 수평 경로

수평 경로는 그 경로의 길이만큼 변이가 연속 또는 불연속적으로 감소하는 것을 의미한다. 따라서 이 경로는 특정 물체 내에서 변이가 감소하는 경우와 불연속, 즉 경계영역과 폐색영역의 경우를 모두 포함하고 있다. 따라서 이 두 경우를 분리할 필요가 있는데, 본 논문에서는 수평경로의 길이가 1인 경우는 물체 내에서의 변이값이 변하는 경우로 간주하고, 수평경로가 2 이상인 경우에만 폐색영역의 가능성을 검사한다.

폐색영역에 대한 검사와 재탐색은 그림 9의 ‘/’에서 ‘\*’까지를 수행하여 진행된다. 2 이상의 수평경로를 만나면 먼저 그 경로에 경계정보가 포함되어 있는지를 확인하고, 경계정보가 포함되어 있는 경우 영상확장을 통해서 폐색영역의 크기를 재탐색하고 그 때까지의 경로를 수평경로로 설정하게 된다.

3.5. 후처리

기본적인 DP 과정인 그림 1을 참조하면 역추적 과정을 거치면서 한 스캔라인의 모든 화소들에 대한 변이값이 모두 결정되므로 그 다음의 과정은 필요 없다. 그러나 본 논문에서 제안한 앞 장의 방법에서 오정합으로 판정된 화소들이 발생하기 때문에 이 화소들의 변이값을 결정하여야 하며(그림 5 참조), 이 과정을 후처리(post-processing) 과정이라고 한다.

먼저 오정합된 화소들을 분석하면 두 가지의 경우가 많이 나타나는데, 그림 10에 이들을 나타내었다. 첫 번째는 (a)와 같이 정합 화소의 상, 하, 좌측 화소(1, 2, 3)의 변이값이 같은 경우이고 두 번째는 (b)에서 좌, 우 화소(a, b)의 변이가 같은 경우이다. (b)에서 c, d 화소와 그 사이 화소의 변이값이 다르기 때문에 사이 화소의 변이값이 오류일 가능성이 높으나 이 경우는 따로 처리하지 않고 그대로 두는 것으로 한다.

이 오정합 화소들에 대한 변이값은 다음과 같이 할당한다. 아래의 할당방법은 특정 스캔라인의 좌측부터 적용하고, 모든 오정합 화소에 대해 ①의 방법을 먼저 적용한 다음 ②를 적용하며, 모든 오정합 화소의 변이값이 할당될 때까지 반복해서 수행한다.

① 인접한 네 화소 중 상·하·좌측의 세 화소의 변이가 같으면 그 변이값을 오정합 화소에 할당한다(그림 10. (a)).

② 인접한 화소 중 좌·우 화소의 변이값이 같으면 그 변이값을 오정합 변이로 할당한다(그림 10. (b)).

③ 이 외의 경우는 인접한 좌측 화소의 변이값을 그 화소의 변이값으로 할당한다.

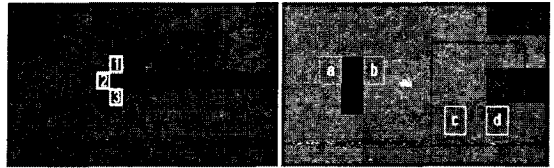


그림 10. 오정합 화소의 분류; (a) 상, 하, 좌측 인접화소의 변이가 같은 경우, (b) 좌, 우 인접화소의 변이가 같은 경우.

Fig. 10. Classification of mismatched pixels of (a) upper, lower, and left, (b) left and right, are the same

IV. 실험 및 논의

4.1. 실험영상 및 파라미터

앞 장에서 제안한 방법의 성능을 객관적으로 실험하기 위해서 Middlebury 사이트[22]에서 제공하는 4개의 실험영상을 사용하여 실험하였다. 이 실험영상들의 파라미터는 영상의 크기와 탐색범위를 제외하고는 모두 동일하며, 각 영상의 크기와 탐색범위는 표 1과 같다. 또한 앞 장에서 제안한 본 논문의 방법에서 사용한 파라미터들과 그 값들은 표 2에 나타내었는데, 참고로 표 2에서 화소값과 정합창의 크기를 제외한 나머지는 모두 실험적으로 구한 값이다.

표 1. 실험영상들의 특성  
Table 1. Characteristics of the test images

Test Image	Image size	Search range
Tsukuba	384 × 288	0 ~ 15
Venus	434 × 384	0 ~ 19
Teddy	450 × 375	0 ~ 59
Cones	450 × 375	0 ~ 59

표 2. 사용한 파라미터들의 값  
Table 2. The values of the parameters used

Symbol	Meaning	Value
I	pixel intensity	Y
MW	size of matching window	3×3
$Occ_{penalty,ver}$	penalty for vertical path	45
$Occ_{penalty,hor}$	penalty for horizontal path	45
$T_{DP}$	threshold for diagonal path exam.	31
p	size of window for diagonal path exam.	2
q	$\lambda$ for $DP(i,j)=0$	0.4
r	$\lambda$ for $DP(i,j) \geq T_{DP}$	1.8
$T_S$	threshold for image separation	7
(2n+1, 2m+1)	window size for image expansion	3×3
$T_{SAD,back}$	SAD threshold for back-tracing	52

4.2. 실험결과

Middlebury에서 제공하는 네 개의 테스트 영상에 대해 본 논문에서 제안한 방법을 적용한 결과(d)를 그림 11에 좌·우 원영상((a), (b)), ground truth (c)와 함께 나타내었다. 또한 이 네 개의 테스트 영상에 대해 제안한 방법을 적용한 결과의 ground truth 영상에 대비한 변이값의 오차율을 표 3에 나타내었으며, 실제 Middlebury site에서 얻은 오차율과 등위를 그림 12에 보였다. 표 3과 그림 12에서는 지금까지 발표된 DP-기반 방법들의 오차율과 등위를 함께 나타내어 제안한 방법과 비교할 수 있도록 하였다.

먼저 그림 11에 나타난 것과 같이, 제안한 방법이 후처리 과정을 포함하고 있기 때문에 ground truth에서 보이고 있는 정합되지 않은 화소들(검은 색)까지 제안한 방법에 의한 결과는 모두 주위 화소의 변이값으로 채워진 것을 확인할 수 있다. 스트리킹 현상을 많이 감소시켜 대체로 정확한 정합이 이루어졌으나, Tsukuba나 Cones와 같이 경계부분이 복잡한 영상에서는 경계부분에서 아직 약간의 스트리킹 현상이 남아 있는 것을 볼 수 있다. 그러나 역추적 과정에서의 경로 재탐색과 후처리 과정에서 오정합 화소에 대한 변이값 결정으로 스트리킹 현상의 길이가 상당히 감소하였으며, 특히 경계부분

이 단순한 영역에서는 스트리킹 현상이 거의 일어나지 않는 것을 볼 수 있다.

이런 결과는 수치적으로 표 3과 그림 12에서 볼 수 있는데, 여기에서는 오차율을 폐색영역이 아닌 부분(nonocc), 경계(불연속) 영역(disc), 전체(all)로 구분하여 나타내고 있다. 제안한 방법은 특히 Teddy와 Cones 영상에서 우수한 특성을 보였는데, Teddy 영상은 영상에 저주파 성분과 고주파 성분이 고루 분포한 영상이고 Cones는 고주파 성분이 특히 많은 영상이다. 따라서 제안한 방법이 고주파 성분이 많은 영상에서 더욱 우수한 성능을 보이는 것으로 판단된다.

본 논문에서 제안한 방법이 스트리킹 현상의 감소에 초점을 맞추었으나, 지금까지 발표된 방법 중 가장 우수한 Region-tree DP[16]과 비교하면, Teddy를 제외한 영상의 경계영역에서 상대적으로 오차율이 높은 것으로 나타났다. 네 영상 전체에 대한 오차율은 제안한 방법이 낮았다. 그럼에도 불구하고 전체적인 등위는 제안한 방법이 약간 낮은 결과를 보였다. 정확히 Middlebury site가 전체 오차율과 등위를 어떻게 매기는지는 모르겠으나, 제안한 방법이 Region-tree 방법에 필적하는 정도의 성능을 보이는 것으로 결론지을 수 있다.

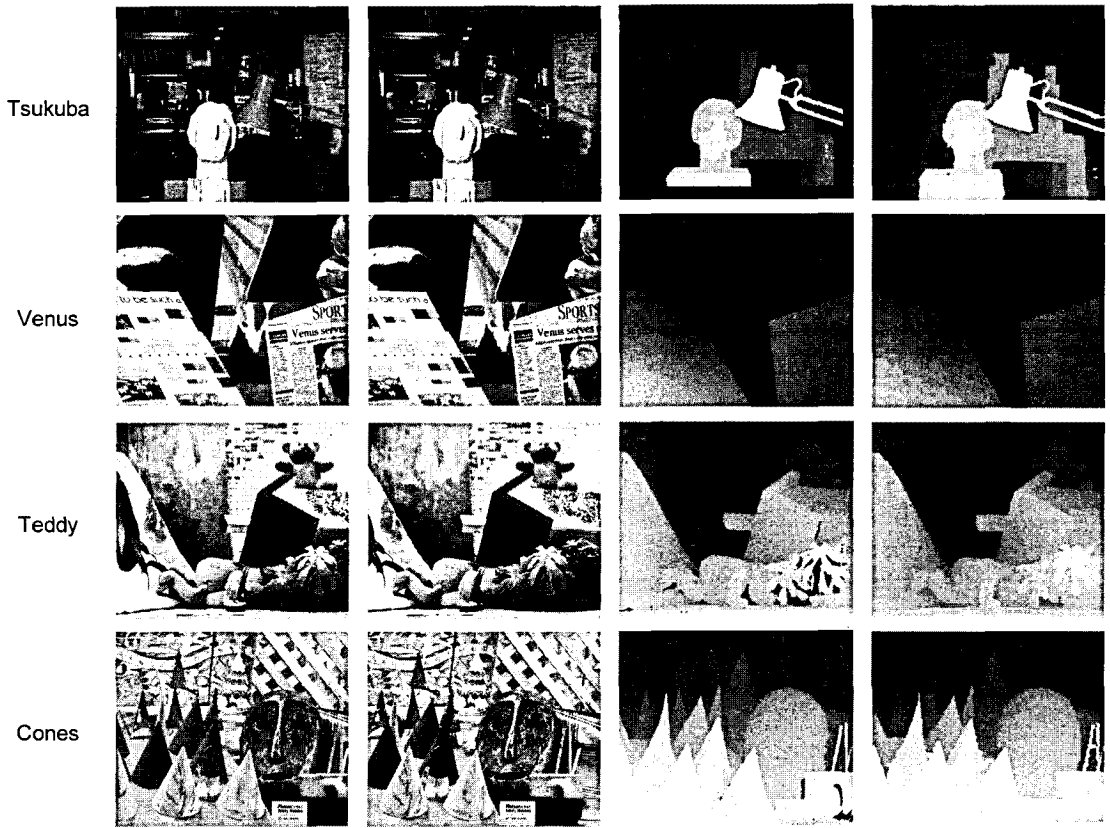


그림 11. 실험영상들에 대한 실험결과; (a)좌측영상, (b)우측영상, (c) ground truth 변이지도, (d) 제안한 방법에 의한 변이지도.

Fig. 11. Experimental results for the test image; (a) left image, (b) right image, (c) disparity map of ground truth, (d) disparity map by the proposed method

표 3. 실험영상들의 변이 오차율

Table 3. The disparity error ratios for the test images

Algorithm	Avg. Rank	Tsukuba			Venus			Teddy			Cones			Average % of bad pixels
		non occ	all	disc	non occ	all	disc	non occ	all	disc	non occ	all	disc	
Proposed	29.9	1.62	2.41	8.39	0.63	0.92	6.85	6.02	7.68	14.8	5.14	9.23	12.3	6.33
Region Tree DP[16]	27.1	1.39	1.64	6.85	0.22	0.57	1.93	7.42	11.9	16.8	3.31	11.9	11.8	6.56
Reliability DP[19]	53.8	1.36	3.39	7.25	2.35	3.48	12.2	9.82	16.9	19.5	12.9	19.9	19.7	10.7
Tree DP[13]	57.0	1.99	2.84	9.96	1.41	2.1	7.74	15.9	23.9	27.1	10	18.3	18.9	11.7
DP[2]	65.0	4.12	5.04	12	10.1	11	21	14	21.6	20.6	10.5	19.1	21.1	14.2

Error Threshold = 1		Sort by nonocc			Sort by all			Sort by disc			Average Percent Bad Pixels			
Algorithm	Avg.	Tsukuba ground truth			Venus ground truth			Teddy ground truth				Cones ground truth		
	Rank	nonocc	all	disc	nonocc	all	disc	nonocc	all	disc		nonocc	all	disc
CostAggr+occ [39]	25.8	1.38 <sub>23</sub>	1.96 <sub>24</sub>	7.14 <sub>28</sub>	0.44 <sub>28</sub>	1.13 <sub>39</sub>	4.87 <sub>36</sub>	6.80 <sub>23</sub>	11.9 <sub>20</sub>	17.3 <sub>33</sub>	3.60 <sub>19</sub>	8.57 <sub>13</sub>	9.36 <sub>23</sub>	6.20
LocallyConsist [69]	25.8	1.70 <sub>34</sub>	2.21 <sub>31</sub>	5.67 <sub>10</sub>	0.16 <sub>9</sub>	0.32 <sub>9</sub>	1.63 <sub>5</sub>	8.68 <sub>48</sub>	13.9 <sub>37</sub>	17.0 <sub>28</sub>	4.19 <sub>35</sub>	10.8 <sub>39</sub>	9.72 <sub>25</sub>	6.33
SegmentSupport [28]	25.9	1.25 <sub>17</sub>	1.62 <sub>10</sub>	6.68 <sub>20</sub>	0.25 <sub>18</sub>	0.64 <sub>21</sub>	2.59 <sub>19</sub>	8.43 <sub>45</sub>	14.2 <sub>42</sub>	18.2 <sub>37</sub>	3.77 <sub>24</sub>	9.87 <sub>31</sub>	9.77 <sub>27</sub>	6.44
RegionTreeDP [18]	27.1	1.39 <sub>25</sub>	1.64 <sub>13</sub>	6.85 <sub>25</sub>	0.22 <sub>15</sub>	0.57 <sub>17</sub>	1.93 <sub>10</sub>	7.42 <sub>32</sub>	11.9 <sub>21</sub>	16.8 <sub>25</sub>	6.31 <sub>57</sub>	11.9 <sub>48</sub>	11.8 <sub>39</sub>	6.56
EnhancedBP [24]	28.7	0.94 <sub>5</sub>	1.74 <sub>13</sub>	5.05 <sub>5</sub>	0.35 <sub>24</sub>	0.86 <sub>28</sub>	4.34 <sub>31</sub>	8.11 <sub>42</sub>	13.3 <sub>31</sub>	18.5 <sub>41</sub>	5.09 <sub>47</sub>	11.1 <sub>42</sub>	11.0 <sub>36</sub>	6.69
PUTv3 [63]	29.7	1.77 <sub>37</sub>	3.86 <sub>50</sub>	9.42 <sub>44</sub>	0.42 <sub>26</sub>	0.95 <sub>33</sub>	5.72 <sub>38</sub>	7.02 <sub>28</sub>	14.2 <sub>41</sub>	18.3 <sub>39</sub>	2.40 <sub>1</sub>	9.11 <sub>20</sub>	6.56 <sub>1</sub>	6.64
YOUR METHOD	29.9	1.62 <sub>31</sub>	2.41 <sub>34</sub>	8.39 <sub>37</sub>	0.63 <sub>39</sub>	0.92 <sub>31</sub>	6.85 <sub>45</sub>	6.02 <sub>13</sub>	7.68 <sub>8</sub>	14.8 <sub>12</sub>	5.14 <sub>48</sub>	9.23 <sub>21</sub>	12.3 <sub>45</sub>	6.33
GradAdaptWat [60]	30.5	2.25 <sub>45</sub>	2.63 <sub>35</sub>	8.99 <sub>41</sub>	0.99 <sub>43</sub>	1.39 <sub>43</sub>	4.92 <sub>37</sub>	8.00 <sub>39</sub>	13.1 <sub>28</sub>	18.6 <sub>43</sub>	2.61 <sub>3</sub>	7.67 <sub>5</sub>	7.43 <sub>4</sub>	6.55
AdaptWeight [12]	31.0	1.38 <sub>23</sub>	1.85 <sub>22</sub>	6.90 <sub>26</sub>	0.71 <sub>38</sub>	1.19 <sub>41</sub>	6.13 <sub>39</sub>	7.88 <sub>36</sub>	13.3 <sub>32</sub>	18.6 <sub>44</sub>	3.97 <sub>31</sub>	9.79 <sub>28</sub>	8.26 <sub>12</sub>	6.67
MultiCue [51]	31.6	1.20 <sub>14</sub>	1.81 <sub>17</sub>	6.31 <sub>17</sub>	0.43 <sub>27</sub>	0.69 <sub>23</sub>	3.36 <sub>26</sub>	7.09 <sub>29</sub>	14.0 <sub>40</sub>	17.2 <sub>32</sub>	5.42 <sub>54</sub>	12.6 <sub>51</sub>	12.5 <sub>49</sub>	6.89
InteriorPILP [34]	32.1	1.27 <sub>18</sub>	1.62 <sub>9</sub>	6.82 <sub>23</sub>	1.15 <sub>47</sub>	1.67 <sub>48</sub>	12.7 <sub>58</sub>	8.07 <sub>40</sub>	11.9 <sub>19</sub>	18.7 <sub>45</sub>	3.92 <sub>30</sub>	9.68 <sub>26</sub>	9.62 <sub>24</sub>	7.26
ImproveSubPix [25]	33.5	3.00 <sub>54</sub>	3.61 <sub>47</sub>	10.9 <sub>52</sub>	0.88 <sub>42</sub>	1.47 <sub>44</sub>	7.10 <sub>47</sub>	7.12 <sub>30</sub>	12.4 <sub>28</sub>	16.6 <sub>24</sub>	2.96 <sub>12</sub>	8.22 <sub>8</sub>	8.55 <sub>16</sub>	6.90
BP+DirectedDiff [61]	34.8	2.90 <sub>51</sub>	4.47 <sub>55</sub>	15.1 <sub>61</sub>	0.65 <sub>37</sub>	1.20 <sub>42</sub>	4.52 <sub>33</sub>	5.07 <sub>8</sub>	14.7 <sub>47</sub>	15.7 <sub>14</sub>	2.94 <sub>11</sub>	12.6 <sub>53</sub>	7.50 <sub>5</sub>	7.29

그림 12. Middlebury 사이트에서 구한 변이 오차율 및 평균 순위  
 Fig. 12. The error ratios and the average ranks by the Middlebury site

### V. 결론

본 논문에서는 동적계획법(dynamic programming)-기반의 스테레오 정합에 있어서 스트리킹 현상을 감소시키고 정합오차를 줄이는 방법을 제안하였다. 이 방법에서는 기존의 비용행렬 계산에 사용하였던 패들티 값들을 주변화소 값에 따라 적응적으로 조정하고, 영상의 특징 값들 즉 경계정보를 이용하여 최적 변이값 추적 경로를 변경하는 방법을 사용하였으며, 후처리 과정을 두어 오정합으로 판단된 화소들의 변이값을 주변 화소들의 변이값에 따라 결정하였다. 특히 추적경로를 변경할 때와 폐색영역에 해당하는 영역의 변이값을 처리할 때에는 경계 좌측 영상을 확장하는 영상확장 방법을 사용하였으며, 이 방법을 역추적 과정에서 적용함으로써 정합 오차율을 줄였다.

제안한 방법은 객관적인 비교를 위해 Middlebury 사이트에서 제공하는 네 가지 영상에 적용하여 성능을 실험하였다. 그 결과 제안한 방법은 상대적으로 고주파성

분이 높은 영상에서 우수한 결과를 보였으며, 물체 내부에 많은 무늬를 갖고 있는 영상에서는 다소 떨어지는 결과를 보였다. 전체적인 변이 오차율을 계산한 결과 기존의 동적계획법을 기반으로 하는 방법 중 가장 우수한 방법보다 약간 낮은 오차율을 보였으나 Middlebury 사이트에서의 순위는 약간 낮아, 이 방법과 필적하는 성능을 보였다.

본 논문에서 제안한 방법 중 영상확장 방법은 향후 더 연구가 필요할 것으로 사료된다. 본 논문에서는 경계영역의 좌측 영상을 단순 확장하는 방법을 사용하였으나, 스테레오 영상을 사용하는 특성 상 폐색영역은 기준영상의 정보를 사용할 수도 있을 것으로 보여, 이에 대한 연구를 진행한다면 동적계획법을 기반으로 하는 방법 뿐만 아니라 다른 스테레오 정합방법에도 유용하게 사용될 수 있으리라 생각된다.

## 참고문헌

- [1] D. SchSarstein and R. Szeliski, "A Taxonomy and Evaluation of Dense Two-frame Stereo Correspondence Algorithms," *International Journal of Computer Vision*, Vol. 47, Issue 1-3, pp. 7-42, April 2002.
- [2] ISO/IEC MPEG & ITU-T VCEG, "Multiview Video plus Depth (MVD) Format for Advanced 3D Video Systems," JVT-W100, April 2007.
- [3] Federico Tombari, Stefano Mattoccia, and Luigi Di Stefano, "Segmentation-Based Adaptive Support for Accurate Stereo Correspondence," *LNCS 4872*, pp.427-438, 2007.
- [4] Tae-june Kim, Ji-sang Yoo, "Hierarchical Stereo Matching with Color Information", 한국통신학회논문지, 제34권, 제3호, pp.279-287, 2009.
- [5] Z. F. Wang and Z. G. Zheng, "A Region Based Stereo Matching Algorithm Using Cooperative Optimization," *IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 1-8, June 2008.
- [6] Q. Yang, et al., "Spatial-Depth Super Resolution for Range Images," *IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 17-22, June 2007.
- [7] Y. Taguchi, B. Wilburn, and C. L. Zitnick, "Stereo Reconstruction with Mixed Pixels using Adaptive Over-Segmentation," *IEEE Conf. on Computer Vision and Pattern Recognition*, June 2008.
- [8] Q. Yang, et al., "Stereo Matching with Color-Weighted Correlation, Hierarchical Belief Propagation, and Occlusion Handling", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 31, No. 3, pp. 495-504, March 2009.
- [9] V. Kolmogorov and R. Zabih, "Computing Visual Correspondence with Occlusions via Graph Cuts," *ICCV 2001*, Vol. 2, pp. 508-515, 2001.
- [10] H. Chen, "Stereo Matching by Dynamic Programming Based on Occlusion Detection," *IEEE Intl. Conf. on Mechatronics and Automation*, pp. 2445-2449, 2007.
- [11] 주재흡, 오종규, 설성욱, 이철훈, 남기곤, "에지 정보를 강조한 동적계획법에 의한 스테레오 정합," 대한전자공학회논문지, 제36권, 제10호, pp. 123-131, 1999.
- [12] J. Zhas and J. Katupitiya, "A Dynamic Programming Approach Based Stereo Vision Algorithm Improving Object Boarder Performance," *IEEE/ERJ Intl. Conf. on Intelligent Robotics and Systems*, pp. 5315-5320, Oct. 2006.
- [13] O. Veksler, "Stereo Correspondence by Dynamic Programming on a Tree," *IEEE Conf. on CVRP*, pp. 20-25, June 2005.
- [14] M. Bleyer and M. Gelautz, "Simple but Effective Tree Structure for Dynamic Programming-Based Stereo Matching," *Intl. Conf. on Computer Vision Theory and Applications*, pp. 415-422, 2008.
- [15] C. Leung, B. Appleton, and C. Sun, "Iterated Dynamic Programming and Quadtree Subregioning for Fast Stereo Matching," *J. of Image and Vision Computing*, Vol. 26, pp. 1371-1383, 2008.
- [16] C. Lei, S. elzer, J. and Y. H. Yang, "Region-Tree Based Stereo Using Dynamic Programming Optimization," *IEEE Conf. on Computer Vision and Pattern Recognition*, Vol. 2, pp.2378-2385, 2006.
- [17] Y. Wei, L. Quan, "Region-based Progressive Stereo Matching," *IEEE Proceeding*, Vol. 1, No. 27, pp. 106-113, July 2004.
- [18] M. Gong, Y. H. Yang, "Fast Unambiguous Stereo Matching Using Reliability-Based Dynamic Programming," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 27, No 6, pp. 998-1003, June 2005.
- [19] M. Gong, Y. H. Yang, "Near Real-time Reliable Stereo Matching using Programmable Graphics Huisiwuse," *IEEE Conf. on Computer Vision and Pattern Recognition*, Vol. 1, pp.20-25, June, 2005.
- [20] M. Gong, Y. H. Yang, "Real-Time Stereo Matching Using Orthogonal Reliability-Based Dynamic Programming," *IEEE Transactions on Image processing*, Vol. 16, No 3, pp. 879-884, Mar 2007.

- [21] H. Hirschmuller, "Stereo Processing by Semiglobal Matching and Mutual Information," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 30, No. 2, pp. 328-341, Feb. 2008.
- [22] <http://vision.middlebury.edu/stereo/>
- [23] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, Pearson Prentice Hall, NJ, 2008.

### 저자소개



박장호(Jang-Ho Park)

2008년 2월 : 청주대학교  
전자공학과 졸업(공학사)  
2010년 2월 : 광운대학교  
일반대학원 졸업(공학석사)

※ 관심분야 : H.264/AVC, Stereo Matching, FPGA/ASIC  
e-mail : jhpark@kw.ac.kr

최현준(Hyun-Jun Choi)

한국해양정보통신학회논문지  
제13권 제8호 참고

서영호(Young-Ho S대)

한국해양정보통신학회논문지  
제13권 제8호 참고

김동욱(Dong-Wook Kim)

한국해양정보통신학회논문지  
제13권 제8호 참고