

스테레오 매칭을 통한 시차맵 생성 및 홀 메우기

이범중⁰ 윤중현 박종승

인천대학교 컴퓨터 공학과

{leeyanga⁰, jhyoon, jong}@incheon.ac.kr

Computation of Dense Disparity Map and Hole Filling

Bum-Jong Lee⁰ Jong-Hyun Yoon Jong-Seung Park

Department of Computer Science and Engineering, University of Incheon

요 약

스테레오 영상으로부터 3차원 구조를 복원하기 위해서는 깊이맵에 해당하는 시차맵을 생성해야 한다. 시차맵 생성을 위해서는 정합비용을 계산하고, 집성한 후에 시차를 계산하는 절차로 이루어진다. 본 논문에서는 스테레오 영상으로부터 빠르고 안정된 시차맵을 생성하기 위해서 후처리 과정으로 각 스캔라인에 대해서 분산을 이용하여 세그멘테이션을 한 후에 세그멘테이션 별로 평균을 내어 객체간의 구분을 명확히 한다. 조밀 시차맵을 생성하기 위해서는 시차 계산에 실패한 화소들에 대해서도 시차를 계산해야 하는데 본 논문에서는 간단하게 인접 화소의 값을 복사하는 방법으로 홀을 메우는 방법을 제안한다. 실제 환경에서의 다양한 스테레오 영상에 대한 실험 결과들은 제안된 시차맵 생성과 홀을 메우는 방법이 기존의 시차맵 생성 기법만큼 빠르고 기존의 방법보다 좀더 안정적이고 다양한 컴퓨터 비전 시스템 응용에 적용될 수 있음을 보여준다.

1. 서 론

최근 실감나는 영상을 표현하여 실제처럼 느낄 수 있도록 해주는 3차원 그래픽 기술과 사람이 상호작용을 할 수 있는 가상현실, 증강현실, 스테레오 비전 기술에 대한 연구가 활발히 진행되고 있다. 3차원 영상기술이 발전함에 따라 3차원 응용분야에 대한 관심이 고조되고 있는데 3차원 영상 응용분야를 구현하기 위해서는 스테레오 비전에 대한 기술을 사용해야 한다. 스테레오 비전은 컴퓨터 비전에서 가장 광범위하게 연구되고 있는 분야 중에 하나이며, 같은 장면의 두 영상에서 정합되는 점들을 찾아 이 점들의 차이를 표현하는 시차맵을 생성하는 것을 목표로 하고 있다.

스테레오 비전에서 같은 장면의 두 영상에서 정합되는 점을 찾는 대응관계 문제는 스테레오 계산에서 가장 중요한 단계에 속한다. 한 장면에 대해서 두 개 이상의 스테레오 영상이 주어진다면, 대응관계는 그 점이 각 영상 평면의 점들 중에서 어디에 투영되었는지를 찾아내는 것이다. 다시 말해서, 참조 영상의 점이 다른 영상들에서 어느 부분에 나타나는지를 찾아내는 것을 말한다. 대응관계의 문제를 풀어내기 위한 방법으로는 영역기반의 방법이 있다.

영역 기반의 스테레오 기법은 한 영상에서 한 화소의 이웃하는 화소들의 밝기 패턴과 다른 영상에서 대응되는 화소의 이웃하는 화소들의 밝기 패턴 사이의 연관성을 사용한다. 이 과정을 상세히 설명하면 먼저, 한 영상에서 한 점을 선택한 후에 다른 영상에서 정합되는 이웃 화소들로 한 점을 찾아내는 것이다. 영역 기반의 기법은 각 화소의 명암값을 바로 사용하기 때문에 한 장면에 대한 여러 카메라의 위치에 따라 명암값이 변하는 것에서 나타나는 왜곡 등에 매우 취약하다는 단점이 있다. 가려짐(occlusion)으로 인한 깊이 정보의 오류가 나타난다는 단점도 있다[1]. Stefano 등은 좀 더 신뢰할 수 있는 정합을 찾으려면 이전의 정합을 버리는 방식을 제안하였다[2].

본 논문에서 제안하는 방법은 두 장의 스테레오 영상을

입력으로 하여 조밀 시차맵을 생성하게 된다. 본 논문에서 제안한 방법은 두 입력 스테레오 영상에 대해서 SSD 방법을 사용하여 정합비용을 계산하고 집성하여 시차맵을 계산하고, 분산 기법을 이용하여 스캔라인별로 세그멘테이션한 후에 세그멘테이션별로 평균을 적용하여 객체 간의 좀더 명확한 구분을 짓도록 한다. 좀 더 부드럽고 조밀 시차맵을 계산하기 위해서 간단하게 인접 화소의 값을 복사하여 홀을 메운다.

2. 관련 연구

그림 1은 스테레오 비전에 대한 카메라 설정을 보여주는 그림이다. 두 장의 스테레오 입력 영상에 대해서 왼쪽의 영상을 I^l 이라 하고, 오른쪽의 영상을 I^r 이라 하고, 두 입력 영상에 대한 카메라의 광학의 중심은 각각 O_l 과 O_r 이라 한다. P 는 장면의 3D점을 의미하며, 그 점이 두 스테레오 영상에 각각 p_l 과 p_r 로 투영된다. O 는 카르테시안(Cartesian) 좌표계로서 3차원의 기준이 되는 좌표계를 의미하고, f 는 초점거리, B 는 두 카메라 중심 사이의 거리인 베이스 라인의 길이를 의미한다.

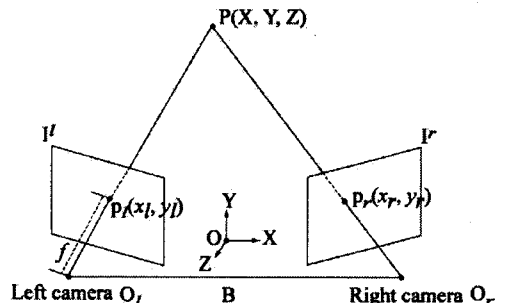


그림 1. 스테레오의 기하 구조 및 카메라 설정 상태

스테레오 비전 기술은 스테레오 영상에 대해서 3차원 정보인 시차값을 계산하는데 매우 유용하며 다음과 같은 3가지 절차에 의해 수행된다[3].

1. 정합 비용 계산(matching cost computation)
2. 비용 집계(cost aggregation)
3. 시차 계산 및 최적화(disparity computation optimization)

정합 비용 계산단계는 좌, 우의 영상에서 대응되는 부분을 찾기 위해 화소간의 유사도를 계산하는 것이다. 화소 기반의 정합 비용을 계산하는 가장 대표적인 방법으로는 식 1과 같이 좌, 우 영상의 명암값의 차이의 제곱으로 비용을 계산하는 명암 차의 제곱(SSD: Squared Intensity Differences)[4]과 식 2와 같이 좌, 우 영상의 명암값의 차이의 절대값으로 비용을 계산하는 명암 차의 절대값(AD: Absolute Intensity Differences)[5]이 있다.

$$(I^l(x, y) - I^r(x + d_x, y + d_y))^2 \quad (1)$$

$$|I^l(x, y) - I^r(x + d_x, y + d_y)| \quad (2)$$

블록 기반의 정합 비용을 계산하는 대표적인 방법으로는 식 3과 같이 계산되는 정규화 교차 상관(NCC: Normalized Cross Correlation)[6]과 식 4와 같이 계산되는 차의 제곱 합(SSD: Sum of Squared Differences)과 식 5에 의해 계산되는 차의 절대값 합(SAD: Sum of Absolute Differences)이 있다. 정합 비용 단계에서는 위의 방법들을 사용하여 모든 화소와 일정 범위의 시차에 걸쳐서 시차공간영상(DSI: Disparity Space Image)을 만들게 된다.

$$\frac{\sum_{x,y} (I^l(x, y) - \bar{I}^l) \cdot (I^r(x + d_x, y + d_y) - \bar{I}^r)}{\sqrt{\sum_{x,y} (I^l(x, y) - \bar{I}^l)^2 \cdot \sum_{x,y} (I^r(x + d_x, y + d_y) - \bar{I}^r)^2}} \quad (3)$$

$$\sum_{x,y} (I^l(x, y) - I^r(x + d_x, y + d_y))^2 \quad (4)$$

$$\sum_{x,y} |I^l(x, y) - I^r(x + d_x, y + d_y)| \quad (5)$$

비용 집계 단계에서 공간과 윈도우 기반의 방법은 시차공간영상에서 올바른 영역에 걸쳐 합계나 평균의 방법으로 정합 비용을 집계한다. 올바른 영역은 고정된 시차에서 2D이거나 x, y, z 로 이루어진 공간에서 3차원이 될 수 있다. 2D는 정사각 윈도우나 가우시안 회선이나 이동할 수 있는 윈도우, 적응적인 크기의 윈도우 등과 같은 복합 윈도우를 사용하여 집계한다. 3D는 제한된 시차의 차이를 이용하여 집계한다.

고정된 혹은 영역으로 집계하는 것은 직사각형의 윈도우의 경우에는 2D나 3D 회선을 사용하거나, 효율적인 박스필터를 사용하여 수행될 수 있다. 이동할 수 있는 윈도우의 경우에는 최소필터를 사용하여 효율적으로 구현될 수 있다. 반복 흐려짐(iterative diffusion)과 같은 집계 방법은 각 화소의 비용에 이웃하는 화소의 비용들의 가중치 값을 반복적으로 더함으로써 구현된다.

시차를 계산하고 최적화하는 단계는 공간방법이 있다. 공간 방법에서는 정합 비용계산 단계와 집계 단계를 강조하며, 시차계산은 간단하게 각 화소에서 최소의 비용을 갖는 값과 관련한 시차를 선택함으로써 이루어진다. 공간 방법은 각 화소에서 WTA(winner take all)를 수행한다. 공간 방법의

단점은 실제 참조 영상에 대해 단지 하나의 정합만을 허용하지만, 다른 영상에서 점은 여러 점과 정합할 수 있다는 것이다.

3. 제안된 방법

스테레오 영상으로부터 시차값을 생성하기 위해서 본 논문에서는 영역 기반의 방법을 사용한다. 영역기반의 방법으로는 색상의 유사도에 기반하여 주어진 윈도우에서 화소의 가중치를 조정하는 가변 가중치를 사용하여 대응관계를 찾는 방법[7]이 있고, 영역에 기반하여 두 영상 사이의 공통의 정보를 최대화하여 유사도를 측정하는 방법[8]이 있다. Veksler[9][10]는 유용한 윈도우의 크기와 형태를 발견하였지만, 너무 많은 파라미터를 정의해야 하는 단점이 있다.

3.1 세그멘테이션 및 평균을 사용한 시차값 생성

본 논문에서는 여러가지 유사도 측정 계산 방법 중에서 SSD 방법을 사용한다. SSD 방법은 식 4와 같이 계산되며, 그 원리는 한 영상의 한 점에 대해서 대응영상에서 대응 후보 점들에 대해 화소 값의 차이의 제곱한 값을 사용하여 두 화소간의 차이를 양의 실수로 표현하게 된다.

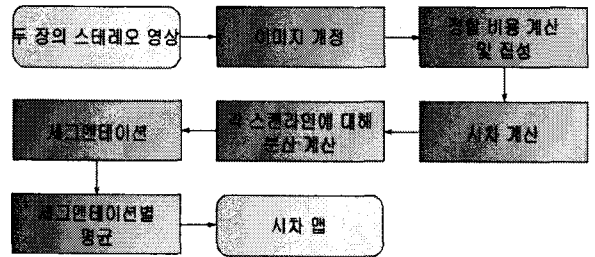


그림 2. 제안된 방법의 전체 과정

본 논문에서 제안하는 스테레오 영상으로부터 시차값을 생성하는 전체 과정이 그림 2에 나타나 있다. 가장 먼저 두 영상의 스캔라인을 맞춰주기 위해 개정(rectification)을 한다. 두 영상의 정합은 두 영상의 에피폴라 기하학을 사용하여 두 영상의 변환 관계를 구한 후에 참조 영상에 정합 영상을 맞추게 된다.

그 다음에는 식 4의 SSD 방법을 사용하여 두 영상에 대한 유사도를 측정한다. 이 단계에서는 이전의 이미지 개정 단계에서 에피폴라 라인을 정렬시켰기 때문에 두 영상에 대해 같은 행에 있는 두 라인을 대응되는 스캔라인으로 하여 정합비용을 계산하고, 집계한다. 이 단계에서는 시차의 단계별로 식 4의 결과인 시차공간영상을 생성하게 된다.

그 다음으로는 시차를 계산하게 되는데, 시차 별로 생성된 시차공간 영상에 대해서 식 6을 사용하여 각 화소에서 비용이 가장 적은 화소를 선택하는 WTA(winner-take-all) 방법을 사용하여 시차를 생성한다. C_d 는 d_i 시차에 대한 x, y 에서의 시차공간영상의 화소값으로서 정합비용을 의미하며, WTA는 각 시차 중에 이 정합 비용이 가장 적은 시차를 선택하는 것이다.

$$\min_d C_d(x, x) \quad (6)$$

다음으로는 세그멘테이션을 하기 위해 각 스캔라인에 대해서 식 7과 같이 분산을 계산한다. 여기서 I^d 는 시차맵에 해당하는 영상을 의미한다.

$$\sigma(x, y) = (I^d(x, y) - \bar{I}^d)^2 \quad (7)$$

다음으로는 식 7로 계산된 분산 값을 이용하여 각 스캔라인별로 세그멘테이션을 하고, 각 세그멘테이션 별로 평균을 내어 각 화소의 값을 결정한다.

3.2 인접 화소값의 복사를 통한 홀 메우기

시차의 계산에 실패한 화소인 홀에 대해서는 홀을 채우는 것이 3차원 복원시 매우 효율적이다. 본 논문에서는 이러한 홀을 채우기 위해 그림 3과 같은 알고리즘을 통해 각 홀에서 가장 가까운 홀이 아닌 화소의 값을 복사하여 홀을 채운다.

```

입력 :  $\sigma, I^d$ 
출력 :  $I^d$ 
파라미터 : width, height(사차맵의 폭과 높이),

win_half_size = 1
for y:=1 to height do
  for x:=1 to width do
    if  $I^d(x, y)$ 가 홀일 경우 then
      for win:=1 to win_half_size do
        min_dist:=MAXINT
        for yy:=y-win to y+ win do
          for xx:=x-win to x+ win do
            if  $I^d(xx, yy)$ 가 홀이 아닌 경우 then
              dist:=sqrt((xx-x)^2+(yy-y)^2)
              if (dist < min_dist) then
                min_dist = dist
                 $I^d(x, y) = I^d(xx, yy)$ 
              end if
            end if
          end for xx
        end for yy
      end for win
    end if
  end for x
end for y
    
```

그림 3. 홀에서 가장 가까운 홀이 아닌 화소의 값을 복사하여 홀을 채우는 알고리즘

4. 실험 결과

제안된 방법의 유효성을 증명하기 위해 본 논문에서는 윈도우 플랫폼에서 middlebury의 코드를 인용하여 시차맵 생성 및 홀을 메우는 시스템을 구현하였다. PC는 2.4GHz 코어2 프로세서와 1GB DDR RAM을 보유하고 512MB DRAM의 GeForce 6800 GPU의 그래픽 카드를 사용하였다. 유효성을 증명하기 위한 스테레오 영상으로는 그림 4의 영상을 사용한다.



그림 4. 384 x 288 크기의 tsukuba의 좌우 스테레오 영상

그림 5는 각각 그림 4의 영상을 입력으로 시차맵을 생성한 결과로서 윗줄의 왼쪽은 실제값, 가운데는 SSD, 오른쪽은 DP(Dynamic Programming), 아랫줄의 왼쪽은 SO(Scanline Optimization), 가운데는 GC(Graph Cuts), 오른쪽은 제안한 방법(PM: Proposed Method)으로 시차맵을 생성한 것이다.

표 1은 그림 5의 시차맵 생성 결과에 따른 성능을 분석한 결과를 표로 나타낸 것이다. RMS(Root Mean Squared) 에러는 식 8과 같이 계산되며 실제 시차맵과 얼마나 많은 차이를 가지는지를 의미한다. 여기서 N 은 화소의 수를 의미하며, I^d 와 I' 는 각각 시차맵과 실제 화소값을 의미한다. 안 좋은 정합(Bad Matching Pixels)은 식 9와 같이 계산되며 정합이 안 좋게 이루어진 화소들의 비율을 나타낸다. 여기서 δ 는 안 좋은 화소로 판별하는 임계치를 의미하며, 본 논문에서는 1.0으로 설정하여 실험하였다.

$$R_{err} = \left(\frac{1}{N} \sum_{(x,y)} |I^d(x, y) - I'(x, y)|^2 \right)^{\frac{1}{2}} \quad (8)$$

$$B_{err} = \frac{1}{N} \sum_{(x,y)} \left(|I^d(x, y) - I'(x, y)| > \delta \right) \quad (9)$$

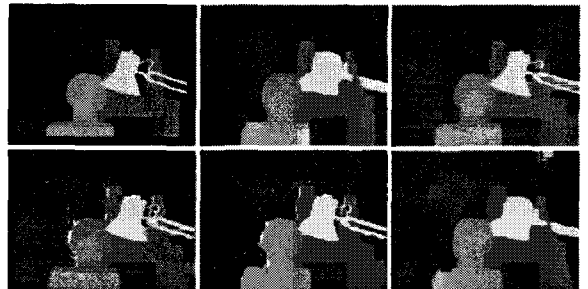


그림 5. tsukuba의 시차맵의 생성 결과

표 1. tsukuba의 시차맵 생성 결과에 따른 성능 분석

	SSD	DP	SO	GC	PM
RMS 에러	1.92046	1.97323	3.11331	2.92175	1.89033
안 좋은 정합	0.11219	0.13135	0.13773	0.11733	0.11055
시간	8.03200	8.23500	10.0780	1253.40	7.87500

표 1을 분석해 보면 제안한 방법이 기존의 방법들 중에서 가장 빠른 방법인 SSD 방법과 수행시간이 거의 비슷하게 나타나면서, RMS 에러 및 안 좋은 정합이 다른 방법에 비해 좀 더 좋은 성능을 나타내는 것을 확인할 수 있다.

그림 6은 생성된 시차맵에 가려짐(occlusion)을 표시한 결과로서, 왼쪽 영상을 기준으로 오른쪽 영상에서 정합되는 화소를 찾을 수 없는 경우에는 녹색, 반대의 경우에는 파란색으로 표시하였으며, 본 논문에서는 왼쪽 영상을 참조 영상으로 하였기 때문에 파란색의 경우는 영역으로 나타나지 않고 선과 같은 경계로 나타나는 것을 확인할 수 있다.

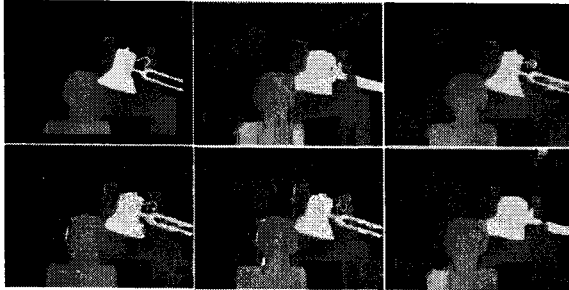


그림 6. tsukuba의 시차맵에 가려짐을 표시한 결과

그림 7은 생성된 시차맵에서 홀에 해당하는 부분들을 인접한 홀이 아닌 화소의 값을 복사하여 홀을 메운 결과들을 나타낸다.

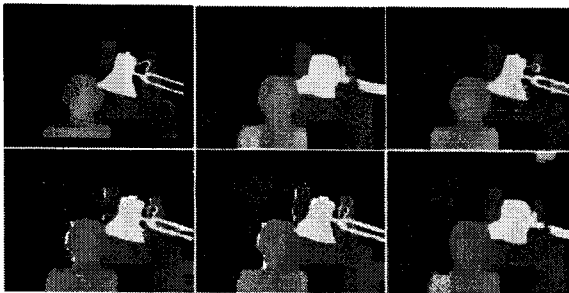


그림 7. tsukuba의 시차맵에서 홀을 메운 결과

5. 결론

본 논문은 스테레오 영상으로부터 3차원 구조를 복원하기 위한 시차맵을 생성하는 방법과 홀을 메우는 방법에 대해서 설명하였다. 본 논문에서는 스테레오 영상으로부터 빠르고 안정된 시차맵을 생성하기 위해서 각 스캔라인에 대해서 분산을 이용하여 세그멘테이션 한 후에 각 세그멘테이션 별로 평균을 내어 객체간의 구분을 좀 더 명확하게 해 주는 후처리 과정을 제안하였다. 3차원 구조를 복원하기 위해서는 시차맵의 모든 화소에 대해서 시차를 계산해야 하는데 시차를 계산하지 못한 화소인 홀에 대해서 홀을 홀이 아닌 가장 인접한 화소의 값을 복사하여 메우게 된다. 제안된 방법은 3가지의 스테레오 영상에 대해서 적용하고, 다른 방법들과 비교하였다. 실험결과들은 제안된 방법이 다른 방법들에 비해

시간 비용을 많이 들이지 않고도 좋은 결과를 얻을 수 있다는 것을 보여준다.

[참고 문헌]

- [1] Han-Suh Koo, Chang-Sung Jeong, "An Area-Based Stereo Matching Using Adaptive Search Range and Window Size," Proceedings of the International Conference on Computational Science-Part II, pp. 44-56 2001.
- [2] L. Di Stefano, M. Marchionni, S. Mattoccia, "A Fast Area-Based Stereo Matching Algorithm," Image and Vision Computing, Vol. 22, No. 12, pp. 983-1005, 2004.
- [3] D. Scharstein, R. Szeliski, "A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms," International Journal of Computer Vision, vol. 47, No. 1-3, pp. 7-42, 2002.
- [4] P. Anandan, "A Computational Framework and An Algorithm for The Measurement of Visual Motion," International Journal of Computer vision, Vol. 2, No. 3 pp. 283-310, 1989.
- [5] T. Kanade, "Development of a Video-Rate Stereo Machine," IUW'94, pp. 549-558, 1994.
- [6] R. C. Bolles, H. H. Baker, M. J. Hannah, "The JISC Stereo Evaluation," DARPA93, pp. 263-274, 1993.
- [7] Kuk-Jin Yoon, In So Kweon, "Adaptive Support-Weight Approach for Correspondence Search," IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol. 28, No. 4, pp. 650-656, 2006.
- [8] Z. Zivkovic, B. Krose, "On Matching Interest Regions Using Local Descriptors: Can an Information Theoretic Approach Help?," British Machine Vision Conference05 2005.
- [9] O. Veksler, "Stereo Correspondence with Compact Windows via Minimum Ratio Cycle," IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol. 24, No. 12, pp. 1654-1660, 2002.
- [10] O. Veksler, "Fast Variable Window for Stereo Correspondence using Integral Images," IEEE Compute Vision and Pattern Recognition, Vol. 1, pp. 556-561 2003.