

## Hager-Belhumeur 알고리즘의 실시간 원근투영 영역 추적기로의 확장

홍현석, 정명진  
한국과학기술원 전기 및 전자공학과

### Realtime Perspective Motion Tracking using Hager-Belhumeur Inverse Additive Algorithm

Hyun Seok Hong, Myung Jin Chung

Department of Electrical Engineering & Computer Science, Korea Advanced Institute of Science and Technology

**Abstract** - 시각센서로부터 가까운 물체들을 다루게 되는 경우 원근효과를 통해 로보트 주변의 물체들을 인지하게 된다. 영역추적기법은 컴퓨터비전에서 가장 널리 사용되는 기술 중 하나이며, 실시간 추적을 위해 많은 효율적 추적기법들이 개발되어 왔다. Hager와 Belhumeur는 Kanade 추적기를 수정한 효율적인 Hager-Belhumeur 알고리즘을 제안하였으며, 어파인(affine) 움직임까지 추적가능함을 보였다. 본 연구에서는 Hager-Belhumeur 알고리즘을 원근 움직임에 적용할 수 있도록 확장하였으며, 기존의 원근 움직임 영역추적기법들과 수렴테스트를 통해 성능비교를 하였으며, 실시간 추적을 수행하였다.

#### 1. 서 론

영상처리 및 컴퓨터비전에서 영상정렬(image alignment)은 가장 널리 사용되는 기법 중 하나이다. 영역추적(region tracking)기법은 이 영상정렬 기법을 초기 영상과 입력영상 관계에 적용함으로써 구현할 수 있다. 영상정렬 기법은 움직임 모델의 파라미터를 통해 픽셀위치가 변형된 입력영상과 템플릿 영상사이 오차를 최소화시키는 움직임 모델 파라미터를 찾는 과정이 된다. Lucas와 Kanade는 광흐름(optical flow)에 기초한 영상정렬 알고리즘을 유도하였다 [3]. 이 영상정렬 알고리즘은 서브픽셀(sub-pixel) 단위의 정교한 추적성능을 보여주지만, 최적화를 수행하는 반복주기마다 매번 현재 입력영상의 변화율영상(gradient image) 및 움직임 모델의 자코비안(Jacobian)을 계산해야 하므로 추적할 영역이 넓을 경우, 실시간 연산이 어려워지는 단점이 있다. Hager와 Belhumeur는 템플릿에 대응되는 입력영상의 밝기가 시간이 흐름에 따라 변하지 않는다는 이미지 항상성을 가정 후, 자코비안의 계산이 파라미터의 변화와 무관한 부분과 그렇지 않은 부분으로 분리될 수 있음을 인지하고, Lucas-Kanade 알고리즘에서 매 최적화 수행 반복 주기마다 계산했던 부분을 영상추적 초기화시 한번만 수행함으로써, 수행속도를 획기적으로 높인 알고리즘을 제안하였다 [2]. 또한 영상 밝기변화나 부분적인 가려짐에 대처할 수 있음을 보였다. 그들은 어파인 움직임(affine motion)까지 추적할 수 있음을 보였으며, 원근 움직임(perspective motion)에 대하여는 언급이 없다. Shum과 Szeliski는 여러 단편적인 영상들을 정렬하여 하나의 파노라마 영상을 얻어내는 연구를 하였으며, 움직임 모델 파라미터를 변화량만큼 더하지 않고, 합성하는 방식으로 영상정렬을 수행하였으며, Lucas와 Kanade의 방식에 비해 알고리즘의 효율성을 증대시켰다 [5]. 최근 Baker와 Matthews는 움직임 모델 파라미터를 합성하는 방식과 함께 효율적인 알고리즘[4]을 제안하였으며, 기존의 알고리즘들을 통합하는 프레임워크를 제안하였다 [1]. 그 프레임워크는 앞서 언급한 네가지 알고리즘의 다음의 이름으로 포함하고 있다: forward additive (Lucas and Kanade's algorithm), forward compositional (Shum and Szeliski's algorithm), inverse additive (Hager and Belhumeur's algorithm), inverse compositional (Baker and Matthews's algorithm). 움직임 모델로는 어파인 움직임과 원근 움직임에 대해서 다루었는데, 아쉽게도 Hager-Belhumeur 알고리즘은 원근움직임에 대해서 적용할 수 없다고 잠정 결론을 내리고 있다. 영상에 투영되는 원근 움직임은 원근 기하학과 관련되어 있으며, 대표적인 예로는 평행한 두 직선이 영상에서는 교차하는 것으로 나타나는 영상왜곡현상이다. 이 왜곡현상은 시각 시스템이 근접한 물체들을 다루는 경우, 흔히 접하게 되는 현상이 된다. 따라서 원근 움직임의 추적은 휴에노이드와 같이 가까운 물체를 만지거나 다루게 되는 시스템에는 필수적이다.

본 논문의 2장에서는 Hager-Belhumeur 알고리즘에 대해 간단히 언급한 후, 원근 움직임 모델에 대한 알고리즘으로 확장하는 수식을 유도하고, 3장에서는 기존의 알고리즘들과 함께 원근 움직임 모델에 대한 영역 추적의 수렴속도 및 수렴율에 관한 시뮬레이션 결과를 보이고 실시간 물체추적의 실험결과를 보인다.

#### 2. 원근 움직임 모델에 대한 Hager-Belhumeur 알고리즘의 유도

##### 2.1 Hager-Belhumeur 알고리즘

Hager-Belhumeur 알고리즘의 목표는 템플릿 영상과 움직임 모델의 파라미터를 통해 픽셀위치가 옮겨진 입력영상 사이의 밝기오차를 최소화하는 파라미터 벡터의 증분값  $\Delta p$ 를 구하는 것이다:

$$\sum_{i=1}^N [I(W(x_i; p + \Delta p)) - T(x_i)]^2 \quad (1)$$

단  $x_i$ 는  $i$ 번째 픽셀위치벡터( $i=1,2,\dots,N$ ),  $p$ 는 움직임 모델 파라미터 벡터,  $W(x_i; p)$ 는 움직임 모델,  $I$ 와  $T$ 는 각각 입력영상과 템플릿 영상이다.

$I(W(x_i; p + \Delta p)) \approx T(x_i)$ 처럼 영상의 항상성을 가정하고, 식(1)에서 1차 테일러 확장을  $I(W(x_i; p + \Delta p))$ 에 수행하여 식(1)을 다시 쓰면 다음과 같다:

$$\sum_{i=1}^N \left[ I(W(x_i; p)) - T(x_i) + \frac{\partial T(x_i)}{\partial x_i} \left( \frac{\partial W(x_i; p)}{\partial x_i} \right)^{-1} \frac{\partial W(x_i; p)}{\partial p} \Delta p \right]^2 \quad (2)$$

식(2)에서 움직임 모델식에 대한 두 자코비안의 곱이 다음 식(3)과 같이 움직임 파라미터의 변화와 무관한 부분과 그렇지 않은 부분으로 분리되는 경우, 효율적인 Hager-Belhumeur 알고리즘의 적용이 가능하게 된다:

$$\left( \frac{\partial W(x_i; p)}{\partial x_i} \right)^{-1} \frac{\partial W(x_i; p)}{\partial p} \equiv \Gamma(x_i) \Sigma(p) \quad (3)$$

만약, 식(3)이 만족되는 움직임 모델이라면, 식(1)에서  $\Delta p$ 의 최적화된 솔루션은 Gauss-Newton 최적화 기법에 의해 다음과 같이 정리된다:

$$\Delta p_* = H_*^{-1} \sum_{i=1}^N \left[ \frac{\partial T(x_i)}{\partial x_i} \Gamma(x_i) \right]^T [I(W(x_i; p)) - T(x_i)] \quad (4)$$

$$H_* = \sum_{i=1}^N \left[ \frac{\partial T(x_i)}{\partial x_i} \Gamma(x_i) \right]^T \left[ \frac{\partial T(x_i)}{\partial x_i} \Gamma(x_i) \right] \quad (5)$$

식(4)을 통해 계산되는 움직임 파라미터 벡터의 증분값을 통한 최종 파라미터의 업데이트는 다음식과 같다:

$$p \leftarrow p - \Sigma^{-1}(p) \Delta p_* \quad (6)$$

계산량을 확인해보면,  $\frac{\partial T(x_i)}{\partial x_i}$ ,  $\Gamma(x_i)$  및  $H_*$ 의 계산과 그들 사이의 곱은 오프라인으로 한번만 계산하면 되므로, 상당한 계산을 매 계산주기마다 하지 않아도 되므로 효율적인 알고리즘임을 확인할 수 있다.

##### 2.2 원근 움직임 모델에 대한 Hager-Belhumeur 알고리즘

원근 움직임 모델은 8개의 원소를 가지는 파라미터 벡터  $p = (p_1, \dots, p_8)^T$ 와 픽셀위치  $x_i = (x_i, y_i)^T$ 에 대해 다음의 비선형 식으로 표현할 수 있다:

$$W(x_i; p) = \frac{1}{p_3 x_i + p_6 y_i + 1} \begin{pmatrix} p_1 x_i + p_4 y_i + p_7 \\ p_2 x_i + p_5 y_i + p_8 \end{pmatrix} \quad (7)$$

앞 2.1절에서 언급했던 것과 같이, Hager-Belhumeur 알고리즘이 적용가능하기 위해서는 식(3)처럼 움직임 모델의 두 자코비안의 곱이 파라미터 벡터  $p$ 에 관한 자코비안과, 픽셀위치  $x_i$ 에 관한 자코비안의 곱으로 분리되어야 한다.

원근 움직임 모델의 미분은 다음과 같다:

$$\frac{\partial W(x_i; p)}{\partial x_i} = \frac{1}{p_3 x_i + p_6 y_i + 1} \begin{pmatrix} p_1 & p_4 \\ p_2 & p_5 \end{pmatrix} \quad (8)$$

$$\frac{\partial W(x_i; p)}{\partial p} = \frac{1}{(p_3 x_i + p_6 y_i + 1)^2} \begin{pmatrix} p_1 x_i + p_4 y_i + p_7 \\ p_2 x_i + p_5 y_i + p_8 \end{pmatrix} \begin{pmatrix} p_3 & p_6 \\ p_3 & p_6 \end{pmatrix} \quad (9)$$

$$\begin{pmatrix} x_i & 0 \\ 0 & x_i \\ 0 & y_i \\ 0 & y_i \end{pmatrix} \begin{pmatrix} p_1 & p_4 \\ p_2 & p_5 \\ p_3 & p_6 \\ p_3 & p_6 \end{pmatrix} \begin{pmatrix} y_i & 0 \\ 0 & y_i \\ p_3 & p_6 \\ p_3 & p_6 \end{pmatrix} \begin{pmatrix} p_1 x_i + p_4 y_i + p_7 \\ p_2 x_i + p_5 y_i + p_8 \\ p_3 & p_6 \\ p_3 & p_6 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

식(8)로 표현되는 행렬의 역행렬과 식(9)의 곱에 해당하는 각 원소들은 비선형식으로 표현되며,  $x_i$ 와  $p$ 가 밀접히 연결되어 있어서 식(5)처럼 두 자코비안으로 분리하는 것이 불가능해 보인다. 하지만 식(8)의 역행렬과 식(9)의 곱의 각 원소들은  $x_i$ 와  $y_i$ 에 대해 2차식의 형태를 갖는다는 것에 착안하면, 식(3)의 형태로 분리시킬 수 있다. 즉, 식(8)의 역행렬과 식(9)의 곱에

해당하는 각 원소는 복잡하긴 하지만  $\sum_{a=0}^2 \sum_{b=0}^{2-a} c_{ab}(p) x_i^a y_i^b$  처럼 표현된

다. 따라서  $x_i^a y_i^b$ 에 해당하는 계수  $c_{ab}(p)$ 를 찾아냄으로써  $\Gamma(x_i)$ 와  $\Sigma(p)$ 를 분리시킬 수 있다. 분리시킨 결과는 다음과 같다:

$$\Gamma(x_i) = \begin{pmatrix} x_i^2 & 0 & -x_i^2 & y_i & 0 & -x_i y_i & 1 & 0 \\ 0 & x_i & -x_i y_i & 0 & y_i & -y_i^2 & 0 & 1 \end{pmatrix}, \quad (10)$$

$$\Sigma(p) = \frac{1}{\det} \begin{pmatrix} a_1 & a_4 & a_7 & 0 & 0 & 0 & -a_3 & -a_6 \\ a_2 & a_5 & a_8 & 0 & 0 & 0 & 0 & 0 \\ a_3 & a_6 & a_9 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & a_1 & a_4 & a_7 & 0 & 0 \\ 0 & 0 & 0 & a_2 & a_5 & a_8 & -a_3 & -a_6 \\ 0 & 0 & 0 & a_3 & a_6 & a_9 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & a_1 & a_4 \\ 0 & 0 & 0 & 0 & 0 & 0 & a_2 & a_5 \end{pmatrix}. \quad (11)$$

단,  $a_1 = p_5 - p_6 p_8$ ,  $a_2 = p_3 p_8 - p_2$ ,  $a_3 = p_2 p_6 - p_3 p_5$ ,  
 $a_4 = p_6 p_7 - p_4$ ,  $a_5 = p_1 - p_3 p_7$ ,  $a_6 = p_3 p_4 - p_1 p_6$ ,  
 $a_7 = p_4 p_8 - p_5 p_7$ ,  $a_8 = p_2 p_7 - p_1 p_8$ ,  $a_9 = p_1 p_5 - p_2 p_4$ ,

$$\det = \begin{vmatrix} p_1 & p_4 & p_7 \\ p_2 & p_5 & p_8 \\ p_3 & p_6 & 1 \end{vmatrix}.$$

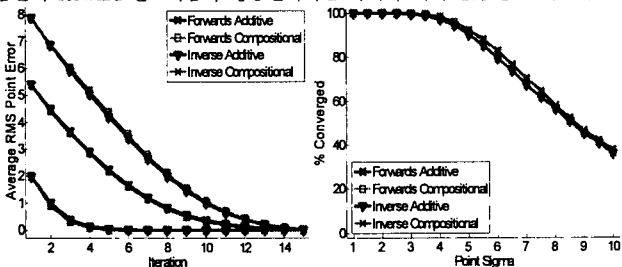
따라서 움직임 모델의 파라미터 업데이트 식(8)에 사용되는  $\Sigma^{-1}(p)$ 는 다음과 같이 간단히 표현된다:

$$\Sigma^{-1}(p) = \begin{pmatrix} p_1 p_4 p_7 & 0 & 0 & 0 & -p_1 p_3 & -p_1 p_6 \\ p_2 p_5 p_8 & 0 & 0 & 0 & -p_2 p_3 & -p_2 p_6 \\ p_3 p_6 & 1 & 0 & 0 & -p_3^2 & -p_3 p_6 \\ 0 & 0 & 0 & p_1 p_4 p_7 & -p_4 p_3 & -p_4 p_6 \\ 0 & 0 & 0 & p_2 p_5 p_8 & -p_5 p_3 & -p_5 p_6 \\ 0 & 0 & 0 & p_3 p_6 p_9 & -p_6 p_3 & -p_6^2 \\ 0 & 0 & 0 & 0 & 0 & p_1 - p_7 p_3 p_4 - p_7 p_6 \\ 0 & 0 & 0 & 0 & 0 & p_2 - p_8 p_3 p_5 - p_8 p_6 \end{pmatrix}. \quad (12)$$

이제 식(4-6), (10), (12)을 통해 원근 움직임을 추적할 수 있다.

### 3. 실험결과

2장에서 유도한 Hager-Belhumeur 알고리즘 (inverse additive)과 기존의 세계의 알고리즘 (forward additive, forward compositional and inverse compositional)과 비교하여, 그 성능을 수렴속도와 수렴을 두가지 측면에서 검증하였다. 검증에 사용된 시뮬레이션 영상 및 초기 움직임 파라미터 데이터는 [1]에서 사용한 것과 동일한 것을 사용하였다. 성능의 척도는 추적영역의 네 꼭지점의 픽셀오차의 RMS(root mean square)값을 사용하였으며, 수렴 여부는 15번의 반복되는 최적화 계산 후에 이 RMS오차가 1픽셀오차보다 작은 경우에 수렴한 것으로 간주하였다. 그림1은 5000개의 동일 데이터에 대해 forward additive, forward compositional, inverse compositional 알고리즘과 원근움직임에 대해 확장된 inverse additive 알고리즘의 평균수렴속도와 평균수렴율을 보여준다. 속도면에서는 훨씬 빠른 inverse 알고리즘들이 forward 알고리즘과 성능면에서는 차이가 거의 없음을 보여준다.



〈그림 1〉 네 알고리즘의 원근움직임에 대한 평균수렴속도 및 평균수렴율



〈그림 2〉 Hager-Belhumeur 알고리즘의 실시간 원근 움직임추적 결과

실시간 물체추적도중의 부분적인 가려짐에 강인하게 하기 위하여, Dutter 과 Huber가 사용하였던 iteratively reweighted least squares (IRLS) 알고리즘[6]을 추가로 구현하였다. 그림 2는 실시간 관심영역 추적결과를 보여준다. 추적결과영역은 흰색 선으로 표현되어 있다. 팔로 박스를 들고 임의로 움직이면서 카메라로는 박스를 추적하고 있으며, 부분적인 가려짐과, 임의의 회전 크기변화, 매끄러운 물체면에 반사되는 부분적인 반사광 등의 환경에서도 주어진 영역을 강인하게 추적하고 있음을 알 수 있다.

〈표 1〉 100×100픽셀 영역을 실시간으로 추적할 때 소요시간 (P4 3.4GHz CPU, C++언어프로그래밍, 단위: msec)

Offline/1회	Online/1회	Online/8회	Online/15회
2.042	0.740	5.920	11.10

표1은 실시간추적시 소요시간을 보여준다. 새로운 영상이 들어왔을 때 움직임을 계산하기 위해 15회의 반복계산을 사용한다고 가정하면, 11.10msec의 시간이 필요하였으며, 카메라를 통한 실시간 영상은 통상 33msec마다 입력되므로, 약 30%의 시스템 자원만을 소모하며, 추적이 목표영역을 아주 빠르게 추적함을 알 수 있다.

### 4. 결 론

휴머노이드 로봇처럼 시각센서와 가까운 물체들을 추적하는 경우, 물체들의 움직임은 시각센서에는 원근움직임 모델을 통해야만 완벽하게 추적할 수 있게 된다. 또한 로봇이 물체를 조작하는 동안 로봇자신의 팔이나 다른 물체들에 의해 가려짐에도 강인한 추적이 필요하다.

본 논문에서는 빠른 추적속도를 가지는 Hager-Belhumeur inverse additive알고리즘을 원근 움직임 모델에 대해 확장 하였으며, 기존의 영역추적의 세 알고리즘(inverse compositional, forward additive, forward compositional)들과 함께 수렴속도 및 수렴을 측면에서 성능을 비교 분석 하였으며, 원근움직임 모델에 대해 유도한 Hager-Belhumeur알고리즘을 사용하여 밝기변화 및 부분적인 가려짐의 환경하에서 실시간 추적을 수행 하였다.

### Acknowledgement

본 연구는 정보통신부 대학 IT연구센터 육성. 지원사업의 연구결과로 수행되었습니다.

### 〈참고 문헌〉

- [1] S. Baker and I. Matthews, "Lucas-Kanade 20 Years On: A Unifying Framework," International Journal of Computer Vision, 56(3):221-255, 2004.
- [2] G. D. Hager and P. N. Belhumeur, "Efficient Region Tracking with Parametric Models of Geometry and Illumination," IEEE Transactions on Pattern Analysis and Machine Intelligence, 20(10):1025-1039, 1998.
- [3] B. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," In Proceedings of the International Joint Conference on Artificial Intelligence, pp.674-679, 1981.
- [4] S. Baker and I. Matthews, "Equivalence and efficiency of image alignment algorithms," In Proceedings of the IEEE Conference of Computer Vision and Pattern Recognition, vol. 1, pp. 1090-1097, 2001.
- [5] H. -Y Shum and R. Szeliski, "Construction of panoramic image mosaics with global and local alignment," International Journal of Computer Vision, 16(1):63-84, 2000.
- [6] R. Dutter and P. Huber, "Numerical Methods for the Nonlinear Robust Regression Problem," J. Statistical Computer Simulation, vol. 13, no. 2, pp. 79-113, 1981.