Proceedings of the 12th
KACC, October 1997

Optical Flow를 이용한 Object Perception System 구성에 대한 연구

The Study on Design of Object Perception System by Optical Flow

°이 형 국*, 정 진 현*

*광운대학교 제어계측공학과(Tel : 02-940-5156; E-mail : dark@ral.kwangwoon.ac.kr)

*광운대학교 제어계측공학과(Tel : 02-940-5156; E-mail : chung@ral.kwangwoon.ac.kr)

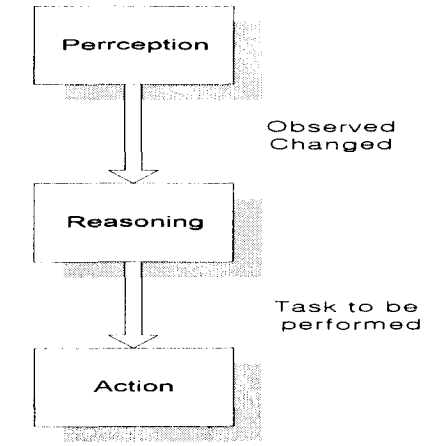
Abstracts Vision system is mainly consist of three parts of perception, reasoning, action. One of these parts, perception system detects visual target in surrounding environment. Block-based motion estimation with compensation is one of the most popular approaches without accuracy. The hierarchical method the optical flow with gradient is used to improve optical flow time delay .

Keyword perception, vision, optical flow

1. 서 론

급속한 공학의 발전은 인간 행동의 많은 부분을 대신하는 대체물을 만들어 냈고 계속적인 연구를 거듭하고 있으며 그 미래는 아무도 예측할 수 없게 되었다. 그 중 Vision은 동물이나 인간의 눈을 대신하고 그 자료를 산업사회의 산물인 기계에게 인식시켜 움직임을 낳게 한다. Vision에서의 주요 논의 대상은 움직이는 물체가 일으키는 시각적 변화를 감지하여 그에 따른 즉각적인 반응을 보이도록 하는데 있다. 즉 CCD나 CAMERA에 의하여 들어온 영상을 받아서 원하는 물체나 대상을 구별해 내어 이것을 쫓아 추적하거나 다음 움직임 을 추정하는 등의 작업을 하게 된다. 이와 같은 일련의 일들을 일반적으로 AI(Artificial Intelligence)에서의 문제 해결 방법인 PRA(Perception-Reasoning-Action) Control Loop라 한다. 그 중 이 논문에서는 지각하여 행동하기 전인 물체의 움직임을 인식하는 부분인 Perception System에 관하여 주로 다루도록 하겠다.

보통 이동하는 물체의 움직임 벡터를 구하기 위해서 블록 매칭 알고리즘을 많이 사용한다. 그러나 블록 매칭 알고리즘은 임의의 한 블록 내 모든 픽셀에서의 움직임이 일정하다는 가정하에 블록 단위로 움직임 벡터를 추정하기 때문에 영상에서의 경계부분에서 부자연스러운 움직임을 보여준다. 반면에 Optical Flow를 이용한 움직임 벡터의 추정은 한 픽셀 단위로 추정하기 때문에 아주 세밀하고 원 움직임에 가까운 움직임 벡터를 추정할 수 있다. Optical flow를 이용한 움직임 벡터의 추정은 gradient를 이용한 방법, region-matching(또는 correlation)을 이용한 방법, energy를 이용한 방법 등이 있다. 하지만 Optical Flow는 전 픽셀의 속도를 구하는 등의 많은 시간을 요구하는 연산을 수행하여야 하므로 이 시간 지연을 해결할 다른 방법이 사용 되어져야 한다. 그 중 이미지를 계층별로 나누어서 optical flow를 구하여 본다.



그 림 1. PRA Communication

2. 본 론

2.1 Perception system

Perception System 의 역할은 주위의 변화를 살피는데 있다. 그런데 주위 변화를 받아들여 계산하면 아무리 강력한 이미지 프로세싱 하드웨어를 사용한다 하더라도 짧은 시간이라도 소요될 것이다. 그럼 결과적으로 이런 이미지들의 연속적인 흐름에서 더 나중에 프로세싱이 이루어지게 된다. Reasoning 프로세싱 계산과 연결된 데이터시간 지연을 고려한다면 Reasoning Process 에서 데이터가 필요할 때 이미지들을 잃지 않고 모든 데이터를 유용하게 사용하려면 특별한 제어 절차가 있어야 한다. 이에 Optical flow 이미지들에 의 하여 제공된 데이터의 트랙을 지키기 위해서 active monitor 와 accumulation process 를 사용한다.

Perception system 은 세 개의 서브 시스템을 구성 되어지고 그림 1 은 다른 프로세서에 메시지를 보내며 병렬로 통신 되어진다.

2.2 Correlation Process

여기의 Perception System 은 믿을만한 위치 정보를 빠르게 얻을 수 있다는데 기인한다. 그것의 방법으로 sum of squared differences(SSD) 나 sum of absolute values of differences(SAD)와 같은 간단한 correlation 기법을 사용한다. Correlation 은 전 픽셀에 행해지고 고정된 윈도우로 첫번째 이미지에서 Correlation 값의 행렬을 계산 하고 두 번째에서는 윈도우를 이동하여 계산한다. 가장 잘 매치된 픽셀은 SSD 를 사용했을 때 최소값이었을 때처럼 optimal correlation 값이다.

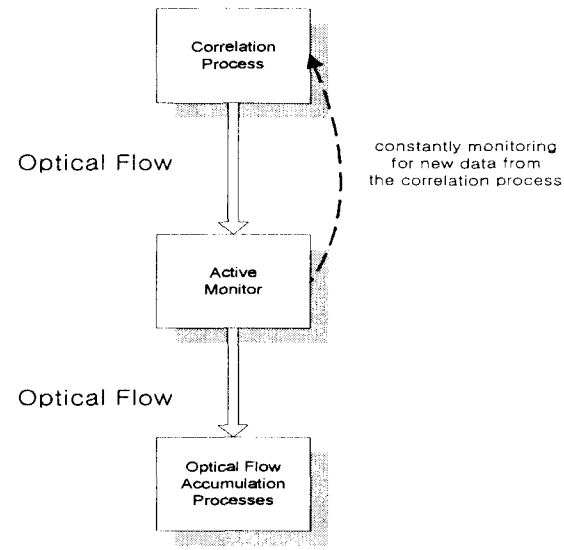


그림 2. **Perception system**

2.3 An Active Monitor

Optical flow 나 stereo disparity , 관리 시스템 등은 특별한 목적의 이미지 프로세서 하드웨어, 다른 프로그램에서 실행될 수 있다. 가능하다면 다른 구성의 플랫폼에서 실행할 것이다. 이를 순환하는 데이터 형식을 균일화하고 시스템 타이밍이나 프로토콜등을 맞추어주는 frame grabber process 를 active monitor 라 한다. 이 특별한 관찰 프로세서는 간단하나 다른 모듈을 묶거나 통신을 동기화 시키는데 매우 중요한 역할을 한다. Optical flow 나 입력영상은 reasoning system 이 프로세스할 준비가 되었는지 안 되었는지에 관계없이 데이터를 쏟아져 들어올 것이다.위치 측정하는 비율보다 관찰 프로세서가 절대 느리지 않다는 것이 중요하다. 만약 데이터를 손실할 경우 뒤의 여러 프레임에서

accumulate displacement 를 연산 할 때 치명적일 수 있다.

2.4 Optical Flow Accumulation

본 논문에서 적용한 optical flow 는 image 에서의 밝기 부분의 변화에 관계된 velocity field 이다. 따라서, optical flow 를 구하기 위해서는 한가지 가정이 필요하게 되는데, ,image 에서 밝기의 변화가 일정해야 한다. 여기서는 두 프레임에서의 correspondence 를 이용하여 optical flow 를 정의하기로 한다. Image sequence $s(X,t)$ 의 변화에 근거하여 image 좌표계 $X(x,y)$ 의 시간 t1 에서 t2 에서의 displacement 를 correspondence vector 라 하며 식 (2-2)로 정의한다.

$$d(X,t)=[d_1(X,t)d_2(X,t)]^T \qquad (2\text{-}2)$$

따라서, 시간 t 에서의 복원될 이미지는 식 (2-3)과 같이 표현된다.

$$s(x+d_1(X,t),y+d_2(X,t),t+\Delta t)=s(x,y,t) \qquad (2\text{-}3)$$

식 (2-3)에서는 나머지에 있어서 조명의 변화가 없이 세기의 변화가 단지 displacement (d_1,d_2)에 만 영향을 받는다는 가정을 두고 있다. Optical flow 벡터는 image 좌표계에서 임의의 픽셀 (X,t)에서의 순간적인 변화율로 정의한다. Optical flow 를 이용한 움직임 추정은 주어진 image 에서의 속도 성분 벡터를 구하는 것이다. 따라서, 식 (2-3)을 (2-4)과 같이 쓸 수 있는데, 2 프레임에서 속도 성분에 변화가 없이 일정하다는 가정에 근거한다. 즉,

$$s(x+u(X,t),y+v(X,t),t+\Delta t)=s(x,y,t) \qquad (2\text{-}4)$$

$u(x,t)$, $v(X,t)$는 각각 $\frac{dx}{dt},\frac{dy}{dt}$ 로써 이미지 좌표계에서의 좌표계 각각의 속도 벡터 성분을 나타낸다. 식 (2-6)을 다시 정리하면, 시간의 변화에 따라 이미지 $S(x,y,t)$의 각각의 픽셀 점은 움직임 궤적에 따라 일정하게 되는 점을 찾아 간다. 즉, $S(x,y,t)$의 시간에 대한 미분 값은 식 (2-5)와 같이 0 이 된다.

$$\frac{ds(X,t)}{dt}=0 \qquad (2\text{-}5)$$

식 (2-5)는 시간의 변화에 따른 image-plane 좌표계의 변화율로서, 움직임 궤적에 따른 세기의 변화율을 나타낸다. 식 (2-4)을 Taylor's expansion 을 이용하여 고차 항은 버리고, 나머지 초기 몇 개의 항만을 선택하여 식 (2-6)을 유도 할 수 있다.

$$\frac{\partial s(X,t)}{\partial x}u(X,t)+\frac{\partial s(X,t)}{\partial y}v(X,t)+\frac{\partial s(X,t)}{\partial t}=0 \qquad (2\text{-}6)$$

식 (2-6)을 optical flow equation 또는 optical flow constraint 라 한다. Image-plane 에서 식 (2-6)을 이용하여 임의의 한 점에서 움직임으로 인한 세기의 변화에 따라 optical flow 를 구해야 한다.

3 . 실험 고찰 및 추후과제

gradient 방법을 이용한 optical flow 를 사용하여 물체의 움직임을 구별해 내는 것을 IBM PC 상에서 시뮬레이션 했다. 동작하는 물체들의 움직임 벡터를 구하여 그것을 인식하도록 하기위한 것이다.

테스트 이미지로 사용된 것은 360x240 의 가든 이미지와 핑퐁 이미지 이다.

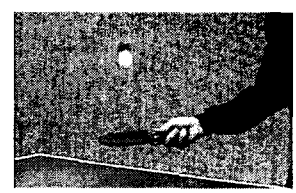


그림 3. Test1 첫번째 image

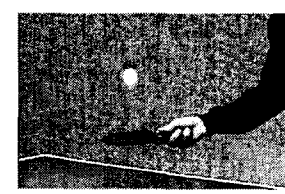


그림 4. Test1 두번째 image

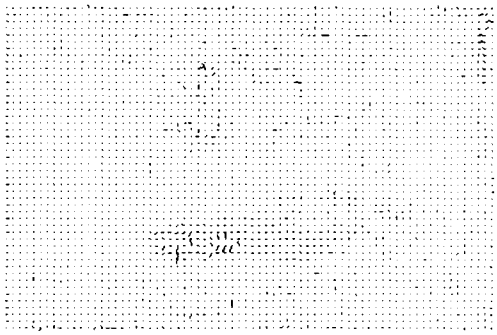


그림 5. 핑퐁 test 이미지의 움직임 벡터

그림 6. Test2 첫번째 image

그림 7. Test2 두번째 image

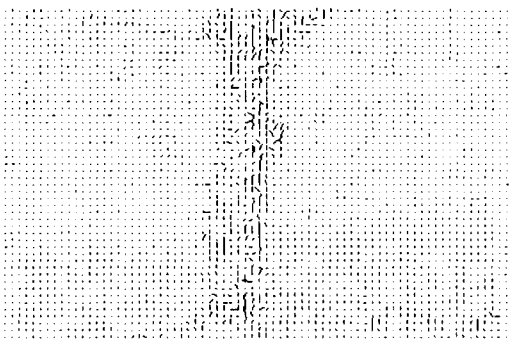


그림 8. 가든 test 이미지의 움직임 벡터

optical flow 가 블록 매칭 알고리즘 보다는 소요 시간이 많이 걸리는 단점을 보완 하기위해서 계층 레벨을 4 단계로 나누어서 각각의 단계별로 optical flow 를 계산하여 최종적으로 가장 높은 단계에서의 optical flow 를 구하였다. 계층 레벨을 4 개로 나누면 하위 레벨의 optical flow 값이 다음 레벨의 초기값이 되고 그 레벨의 optical flow 의 값이 다시 다음 초기값이 된다. 그래서 많은 계산량이 줄어들게 된다. 그리고 그림 5 와 그림 8 의 움직임 벡터를 보면 active target 의 움직임이 눈으로도 보이게 된다. 그럼 이 픽셀들의 위치 값을 다음 reasoning system 에 보내어 다음 동작을 하게 된다.

원래 optical flow 는 비전에서 움직임 추정을 위해 쓰이는 방법으로써, 3-D 움직임추정에 자주 사용되어지지만 이 논문에서는 2-D 의 움직임 벡터로 구현했다. 움직이는 물체를 다른 배경과 정확히 가시적으로 보이기 위해서는 그것을 구별하는 segmentation 을 동시에 가능하게 하여야 할 것이다. 더욱 빠른 속도의 optical flow 를 구현하기 위해서 optical flow accumulation 을 더욱 연구 하여야 할 것이다.

4. 참 고 문 헌

[1] 변재응, 정진현, "Optical flow 의 sequence coding 의 적용에 관한 연구",광운대학교,1996

[2] 변재응, 김재영, 이원희, 김상기, 정진현, "Optical flow 를 이용한 motion estimation 에 관한 연구", KACC, 1996

[3] Johnny Wai Yee Kam, "A Real-Time 3D Motion Tracking System", The University of British Columbia, 1993

[4] A.Murat Tekalp, "Digital video processing", prentice hall, 1995

[5] David John Coombs, "Real-time Gaxe Holding in Binocular Robot Vision", University of Rochester,1992