

칼만 필터를 이용한 다중 차량 추적 알고리즘

김형태*, 설성욱*, 이철현**, 강창순***, 남기곤*

*부산대학교 전자공학과, **양산대학, ***동의공업대학

htkim@vision.ce.pusan.ac.kr

Multiple Vehicle Tracking Algorithm Using Kalman Filter

Hyung-Tae Kim*, Sung-Wook Seol*, Chul-Hun Lee**, Chang-Sun Kang***, Ki-Gon Nam*

*Dept. of Electronics Eng. Pusan National University, **Yang San College,

***Dong Eui College

Abstract

This paper describes the algorithm which extracts moving vehicles from sequential images and tracks those vehicles using Kalman filter. This work is composed of a motion segmentation stage which extracts moving objects from sequential images and gets features of objects, and a motion estimation stage which estimates the position and the motion of moving objects using Kalman filter. In the motion estimation stage, applying to affine motion model we divided the Kalman filter into position filter and velocity filter to employ linear Kalman filter.

Multi-target tracking requires a data association component that decides which measurement to use for updating the state of which object. We use pattern recognition method to solve this problem.

I. 서론

컴퓨터 비전 기술은 차량 통제 시스템을 구현하는 데 응용되어질 수 있다. 즉, 연속 영상으로부터 동작정보를 가지는 차량을 추출하고, 추출되어진 차량을 추적하여, 그 차량의 궤적, 속도, 가속도 등의 정보를 구함으로써 도로의 혼잡도, 사고의 발생 등의 유용한 정보를 얻을 수 있다.

이러한 비전 시스템을 구현하기 위해서 지금까지 연구된 가장 대표적인 방법은 3차원 모델기반 추적(3D-Model-Based Tracking) 알고리즘이다. 3차원 모델기반 추적 알고리즘은 정확한 추적이 이루어진다는 장점을 가지지만, 계산량이 많고 모든 모델에 대한 사

전 정보를 가져야한다는 단점 때문에 실시간으로 처리하는 추적 시스템을 만들기에 부적합하다. 그러므로 본 논문에서는 차량 모델에 대한 아무런 사전 정보도 필요 없고, 적은 계산량으로 비교적 정확한 추적을 할 수 있는 추적 알고리즘을 제안하고자 한다.

II. 동작 분할

동작 분할은 입력 영상으로부터 물체들을 구분하는 영상 분할(Image Segmentation) 과정을 거친 후, 분할된 물체들의 특징들을 얻어내는 과정을 말한다.

연속 영상에서 고정된 배경으로부터 움직이는 물체만을 뽑아내는 가장 손쉬운 방법은 현재의 입력 영상과 고정된 배경사이의 차 영상을 이용하는 방법이다. 즉, 차 영상을 영상의 히스토그램을 이용하여, 적절한 임계값으로 이진화시켜서 움직임을 가지는 물체부분만 추출하는 것이다. 이 과정을 수식으로 나타내면 다음과 같다.

$$M_t(x) = \begin{cases} 1 & |D_t(x)| > T_t \\ 0 & \text{else} \end{cases} \quad (1)$$

$D_t(x)$ 는 t번째 프레임(Frame)에서의 고정된 배경과 입력 영상의 차영상이고, T_t 는 동작분할을 위한 적절한 임계값, 그리고 $M_t(x)$ 는 동작 분할된 이진 영상을 나타낸다.

이렇게 얻어진 이진 영상으로 영상 분할 과정을 거쳐서 움직이는 물체들을 구분하고, 필요한 특징 값들을 구한다.

III. 칼만 필터

칼만 필터란 잡음에 의해 간섭받는 선형 동적 시스템 (Linear Dynamic System)에서의 상태 벡터(state vector) x_k 의 최적의 추정치 \hat{x}_k 를 구하기 위한 순환적인(recursive) 알고리즘이다. 여기서 k 는 시간을 나타낸다.

칼만 필터링은 크게 예측(prediction), 측정(measurement), 수정(update)의 세 가지 단계로 나뉘어진다.

예측 단계에서는 다음 상태의 상태 벡터 $x_{k+1(-)}$ 과 상태 추정 오차의 공분산 행렬 $P_{k+1(-)}$ 을 예측하는 단계이다. (-)는 아직 측정을 통한 수정 단계가 이루어지지 않았음을 나타낸다. 상태 추정 오차의 공분산 행렬을 수식으로 표현하면 아래와 같다.

$$P_{k+1(-)} = E[(x_{k+1(-)} - \hat{x}_{k+1(-)})(x_{k+1(-)} - \hat{x}_{k+1(-)})^T] \quad (2)$$

예측 단계에서 이루어지는 상태 벡터와 상태 추정 오차의 공분산 행렬의 예측을 나타내는 두 식은 다음과 같이 주어진다.

$$\hat{x}_{k+1(-)} = f(\hat{x}_k) + w_k \quad (3)$$

$$P_{k+1(-)} = P_{k(+)} + Q_k \quad (4)$$

윗 식에서 w_k 는 시스템의 모델 잡음(Model Noise)이고, Q_k 는 w_k 의 공분산 행렬이다.

측정 단계는 아래와 같은 수식(Observation Equation)으로 표현되어 질 수 있다.

$$z_k = H \cdot \hat{x}_k + v_k \quad (5)$$

z_k 는 측정 벡터, H 는 상태 벡터와 측정 벡터사이의 관계를 나타내는 행렬(Observation Matrix), v_k 는 측정 오차를 나타내며, 모델 잡음 w_k 와 측정 오차 v_k 는 비상관(uncorrelated)되었다고 가정한다.

마지막으로 수정 단계에서는 측정 단계에서 구해진 측정값을 바탕으로 상태 벡터와 상태 추정 오차를 새로운 값으로 수정한다. 수식으로 표현하면 아래와 같다.

$$K_k = P_{k(-)} H^T [H_k P_{k(-)} H^T + R_k]^{-1} \quad (6)$$

$$P_{k(+)} = [I - K_k H_k] P_{k(-)} \quad (7)$$

$$\hat{x}_{k(+)} = \hat{x}_{k(-)} + K_k [z_k - (H_k \hat{x}_{k(-)})] \quad (8)$$

위 식에서 R_k 는 측정 오차 v_k 의 공분산 행렬을 나타내고, K_k 는 칼만 이득(Kalman Gain)을 나타낸다. 칼만 이득의 역할은 상태 벡터 x_k 를 수정할 때, 측정값과 추정값사이의 차이(Innovation)에 적절한 가중치(Weighting Factor)를 주는 역할을 한다.

IV. 위치 필터와 속도 필터

동작 분할(Motion Segmentation)과정에서는 움직이는 3차원 물체가 영상에 투영되어져서 나타난 영상 조각(Image Patch)을 찾아내었다.

이 때, 아핀 동작 모델을 나타내는 식은 다음과 같다.

$$m(x) = A(x - x_m) + m_0 \quad (9)$$

$m(x)$ 는 어떤 영상 조각(Image Patch)내의 위치 x 에서의 동작(Motion)을 나타내고, A 는 회전(Rotation)과 크기(Scale)를 나타내는 행렬이며, x_m 은 영상 조각의 중심점의 위치, m_0 는 x_m 에서의 모션을 나타낸다. 즉 아핀 동작 모델이란 같은 영상 조각 내에서의 모션은 선형적으로 변환되어 질 수 있다는 것이다.

동작 분할 과정에서 얻어진 한 물체에 대한 특징점의 위치와 속도를 나타내는 상태 벡터는 각각 아래와 같이 표현된다.

$$x_k = (x_k^1, y_k^1, x_k^2, y_k^2, \dots, x_k^N, y_k^N)^T \quad (10)$$

$$m_k = (u_k, v_k, s_k)^T \quad (11)$$

N 은 특징점의 수를 나타내고, u_k, v_k, s_k 는 각각 물체의 x 축 방향의 속도, y 축 방향의 속도, 카메라로부터 접근하는 지 혹은 멀어지는 지를 결정하는 크기 계수(Scale Factor)를 나타낸다. 실제 추적 시에는 이들의 초기 값이 필요하게 되는 데, 물체가 처음으로 발견된 프레임이 $k-1$ 번 째 프레임이고, k 번 째 프레임에서 물체의 중심 좌표를 $x_{ck} = (x_{ck}, y_{ck})^T$ 이라고 할 때, 속도 필터의 초기 값은

$$m_k = (x_{ck} - x_{ck-1}, y_{ck} - y_{ck-1}, 0) \quad (12)$$

으로 주어진다.

본 실험에서 측정값 z_k 로 사용되어지는 값은 위치 좌표의 값이다. 그러므로, 위치 필터에서 상태 벡터와 측정 벡터의 관계는 아래와 같이 주어진다.

$$z_k = \hat{x}_k + v_{1k} \quad (13)$$

v_{1k} 는 위치 필터에서의 측정 오차를 나타낸다.

속도 필터의 관측 행렬 H_{2k} 가 아래와 같을 때,

$$H_{2k} = \begin{bmatrix} 1 & 0 & \hat{x}_k(-) - x_{ck} \\ 0 & 1 & \hat{y}_k(-) - y_{ck} \end{bmatrix} \quad (14)$$

속도 필터에서의 상태 벡터와 측정 벡터의 관계는 (14)식으로 주어진다.

$$z_k = H_{2k} \cdot \hat{m}_{k(+)} + \hat{x}_{k(+)} + v_{2k} \quad (15)$$

아핀 동작 모델을 이용해서 위치 필터의 예측 단계를

수식으로 나타내면,

$$\hat{x}_{k+1(-)} = \hat{x}_{k(+)} + s_k(\hat{x}_{k(+)} - x_{ck}) + (u_k, v_k)^T + w_{1k} \quad (16)$$

위 식과 같아지고, 위치 필터와 속도 필터가 서로 상호적인 관계를 가지고 있음을 알 수 있다.

속도 필터의 예측 단계는 아래와 같다.

$$\hat{u}_{k+1(-)} = \alpha \cdot \hat{u}_{k(+)} + w_{2k} \quad (17)$$

α 는 감속 계수(deceleration)이며, 물체가 영상의 위쪽으로 이동해나가면서 크기가 작아지기 때문에 생기는 1보다 작은 상수 값이다.

마지막으로 수정 단계는 위치 필터에 대해서는 (6), (7), (8) 식에 H대신 단위 행렬을 대입하면 되고, 속도 필터에 대해서는 약간의 수정이 필요한 데, 그 수식들은 다음과 같다.

$$K_k = P_{2k(-)} H_{2k}^T [H_{2k} P_{k(-)} H_{2k}^T + R_k]^{-1} \quad (18)$$

$$P_{2k(+)} = [I - K_k H_{2k}] P_{2k(-)} \quad (19)$$

$$\hat{u}_{k(+)} = \hat{u}_{(-)} + K_k [z_k - ((H_k \hat{u}_{k(-)}) + \hat{x}_{k(-)})] \quad (20)$$

V. 제한된 탐색 창 의 사용

영상에서 추적하고 있는 물체를 찾아내기 위해서 전 영역에 대해서 탐색하는 것이 아니라 물체의 위치의 예측 값 주변에서만 탐색을 하면 보다 효율적인 탐색이 이루어지게 된다. 본 실험에서는 여유 값으로 예측 위치에서 5-10픽셀정도 더한 부분을 탐색 창(Searching window)으로 만들어서 사용하였다.

탐색 창을 만들어서 사용할 때의 이점은 크게 두 가지가 있다. 먼저, 탐색 창을 사용함으로써, 동작 분할에 걸리는 시간을 단축시킬 수 있다. 두 번째 이점은 Data Association 문제를 어느 정도 해결할 수 있다는 점이다. 탐색 창을 사용하게 되면, 물체가 겹쳐지는 경우를 제외하고는 추적하고자 하는 물체가 그 물체를 추적하기 위한 탐색 창 안에서 바로 측정되어 지기 때문에 Data Association 문제를 해결할 수 있게 된다. 다만, 탐색 창 내에 두 개 이상의 물체가 탐지되었을 경우에는 내가 추적하고자 하는 물체가 어느 물체인지를 결정해야 하는 데 이 때 패턴 인식의 기법이 필요하게 된다.

VI. 패턴 인식 기법의 사용

본 실험에서 추적하고자 하는 물체를 인식하기 위해

서 사용된 특징들은 물체의 픽셀 수 n , 물체의 밝기 정보 i , 물체의 가로 세로 비 r 이다. 이 세 가지 특징 값이 라벨(label) l 을 가지는 물체에 대한 특징 벡터를 이루게 되는 데, 식으로 표시하면,

$$f_l = (n_l, i_l, r_l) \text{ 이 된다.} \quad (21)$$

탐색 창 내에 2개 이상의 물체가 있을 때, 이 전 프레임에서 기억된 물체의 특징과 탐색 창 내에서 발견된 물체들의 특징을 비교해서 가장 짧은 유클리디안 거리(Euclidean Distance)를 가지는 물체를 추적하고자 하는 물체로 인식하게 된다. 이 전 프레임에서 저장된 추적하고 있는 물체의 특징을 $f = (n, i, r)$ 라 하고, 현재 프레임의 탐색 창에서 발견된 물체들 중 추적하고 있는 가장 적합한 물체로 인식되는 물체의 라벨을 l_m 이라고 하면, 아래와 같이 구해진다.

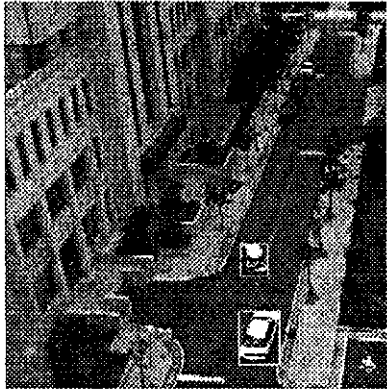
$$l_m = \arg \min \sqrt{(n_l - n)^2 + (i_l - i)^2 + (r_l - r)^2} \quad (22)$$

VII. 실험 결과

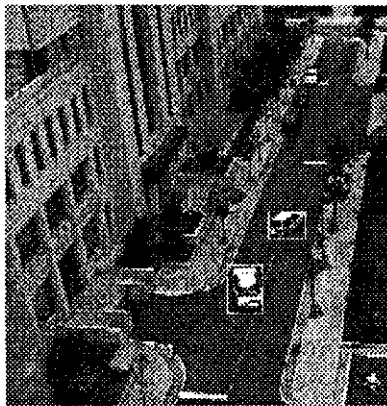
실험에 사용된 영상은 256×256의 8bit gray level 영상이고, 위치 필터에서 특징 점으로 사용된 값은 물체에 외접하는 사각형의 좌측 상단과 우측 하단의 꼭지점이다. 그림 1에서는 여러 없이 두 대의 차량을 추적하고 있는 과정을 보여 준다. 수행 시간은 150MHZ 펜티엄 컴퓨터에서 프레임 당 0.4초 정도의 시간이 걸렸다. 3차원 모델 기반 추적알고리즘과 비교해서 수행 속도가 훨씬 개선 되었음을 알 수 있다. 그림 2에서는 탐색 창 내에서 두 물체가 탐지 되었을 때, 패턴 인식에 의해서 정상적인 추적과정이 계속 진행됨을 보여 준다.



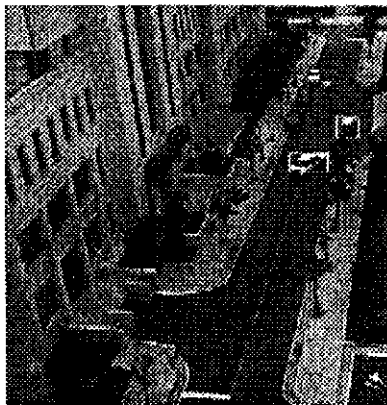
(a)



(b)



(c)

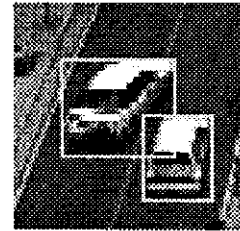


(d)

그림 1. 추적 알고리즘의 수행 결과
(전체 과정을 보이기 위해 샘플링 된 결과)



(a)



(b)

그림 2. 패턴 인식의 사용

- (a) 탐색 창에 두 물체가 동작 분할 된 경우.
- (b) 정상적인 추적 과정이 계속 진행되고 있음.

VI. 결론 및 향후 과제

제안된 추적 알고리즘을 사용하여 실제 연속 영상에 적용하였을 때 빠른 수행 속도와, 비교적 정확한 추적이 가능함을 보였다. 실제로 이러한 알고리즘을 고속도로 등의 환경에 적용할 때는 조명이나 날씨, 물체가 가려지는 경우(Occlusion Case)등을 고려하여야 할 것이다.

참고 문헌

- [1] D. Koller, K. Daniilidis, H.-H. Nagel, "Model-Based Object Tracking in Monocular Image Sequences of Road Traffic Scenes", International Journal of Computer Vision 10:3, 257-281 (1993)
- [2] Dieter Koller, Joseph Weber, and Jitendra Malik, "Robust Multiple Car Tracking with Occlusion Reasoning", Proceedings of the Third European conference on Computer Vision, Vol. 1, pp. 189-199, Stockholm, Sweden, 1994
- [3] Sylvia Gil, Ruggero Milanese, Thierry Pun, "Feature selection for object tracking in the traffic scenes", SPIE International Symposium on Smart Highways, Boston, Massachusetts, Oct. 31 - Nov. 4, 1994