

차량의 겹침현상에 강인한 영상검지시스템 연구

A Study of Robust Image Vehicle Detection System to Occlusion of Vehicle

한승준

정광훈

김종복

(세인시스템, 책임연구원) (세인시스템, 주임연구원) (세인시스템, CTO)

Key Words : 영상검지기, 차량검침, 차량추적, 교통정보, 높이추정

목 차

- I. 서론
- II. 영상검지기에서의 겹침문제
- III. 제안된 겹침현상에 강인한 알고리즘
- IV. 실험 및 평가
- V. 결론

I. 서론

영상검지시스템은 도로의 훼손 없이 설치하여 다차로의 교통정보를 취득할 수 있고 유지보수가 용이하여 선호되는 차량 검지 시스템이다. 이러한 영상검지시스템은 일출·일몰 같은 천이 시간, 폭우·폭설 등의 기상 이변, 카메라의 설치 위치에 따라 그 성능에 있어 큰 영향을 받는 단점을 지닌다. 특히 카메라의 설치 높이가 높을수록 좋은 성능을 낸다는 것은 잘 알려진 사실이다[1][2]. 그러나 도로구조, 비용, 미관 등의 이유로 많은 제약을 갖게 되며 이로 인해 같은 영상 검지 시스템이라 할지라도 그 위치에 따라 확연한 성능 차이를 보이게 된다.

영상검지기를 위한 영상처리 기술들은 가상루프 또는 Tripwire 방식과 시공간해석(Spatial-Temporal Analysis)방식 그리고 추적(Tracking)방식으로 분류 될 수 있다. 가상루프, Tripwire방식은 개념적으로 큰 차이 없이 영상의 특정 영역만을 관찰하여 그 영역 내를 지나가는 물체를 분석하여 결과는 얻어내는 일종의 매설형 루프 검지기를 모방하는 것이라 할 수 있다. 이들은 간단하지만 그 결과의 신뢰도가 낮고 개념적으로 한계를 가지고 있어 그 발전가능성이 낮다. 시공간 해석 방법은 개별차량의 정보보다는 단위시간당 공간을 차지하는 특성을 분석하여 도로공간을 기술하는 방법으로 우리나라 실정에 적합하지 않다. 현재 상용화된 대부분의 검지 방식은 Tripwire방식이지만, 국내외적으로 가장 활발히 연구되고 있는 방법은 추적방식이다. 추적 방식은 정보처리 양이

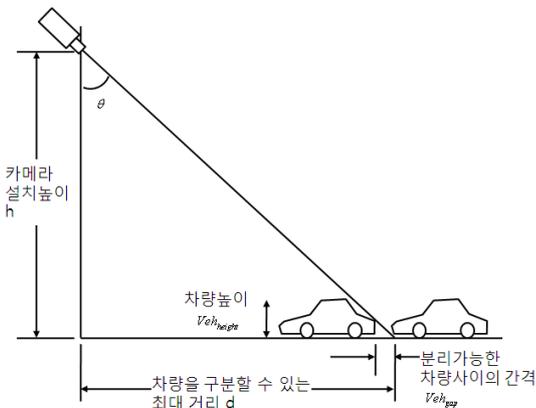
많고 복잡하여 높은 시스템 성능이 요구되지만 앞으로 많은 발전 가능성을 내포하고 있다. 추적 방식 역시 일부 영역 내에서만 추적 하는 방식과 물체의 외곽(contour)을 추출 추적 하는 방식 그리고 특징점을 추출 추적하는 방식 등 다양한 방법이 연구되고 있다. 추적 방식의 핵심은 추적하는 물체를 어떻게 분리하여 어떻게 추적 하는가 인데, 앞서 설명한 영역, 외곽, 특징점 추적 등의 방법 역시 2차원적인 데이터를 바탕으로 하기 때문에 환경변화, 겹침(occlusion)등의 외란에 약하고, 이를 극복하기 위하여 예측이나 추론 등의 방법이 사용되지만 근본적인 해결 방법이 될 수는 없다[3]-[14].

본 논문에서는 외부 환경적 요인이 아닌 카메라 설치 위치에 대해 관심을 가지고 카메라의 높이가 낮아질 경우 필연적으로 발생하는 겹침현상에 대해 효과적인 대응을 할 수 있는 알고리즘에 관한 연구를 수행하였다. 2장에서는 이러한 영상검지 시스템에서 발생하는 겹침현상에 대하여 설명하였고, 3장에서는 모든 움직이는 물체의 특징점 분석 및 특징점의 높이 추정을 이용한 차량의 분할방법과 분할된 개별차량의 주행 경로추적 방법, 수집된 경로정보를 통한 영상검지기의 성능향상 및 기능 개선방안을 제안하였다. 4장에서는 실험을 통해 본 논문에서 제안한 방법의 유효성을 증명하였으며, 마지막으로 5장에 결론과 향후과제에 대해서 논하였다.

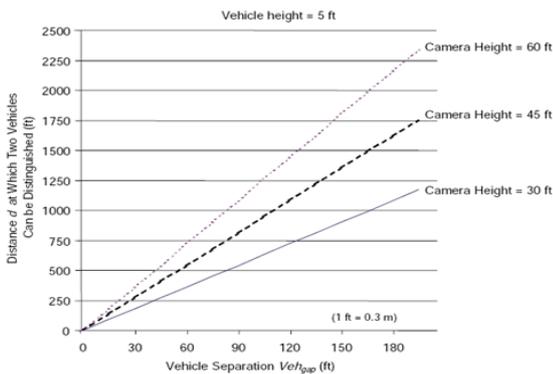
II. 영상검지기에서의 겹침문제

영상검지 시스템에서 카메라 위치와 차량 크기 등의 이유로 겹침현상이 발생하게 된다. 이로 인해 차량을 검지할 수

있는 영역의 제약이 발생하게 되고, 이로 인해 궁극적으로 검지정보의 신뢰도가 낮아지게 된다. 이러한 겹침현상에 대한 문제점은 이미 잘 알려져 있다[1][2]. Klein 등은 이러한 겹침현상과 차량의 거리상이의 상관관계에 대해 연구하였다[15]. 그림 1 에 카메라의 설치 높이와 겹침현상과의 상관관계를 나타냈고, 도로가 평탄하다고 가정할 때, 차량의 구분 가능한 최대 거리 d 는 $d = h \cdot Veh_{gap} / Veh_{height}$ 과 같이 근사화 될 수 있다. 여기서 h 는 카메라의 설치 위치, Veh_{gap} 는 차량사이의 거리, Veh_{height} 는 차량의 높이이다. 위 수식에 의해서 차량의 높이가 5ft(1.5m) 일 차량의 경우에 대한 차량을 구분할 수 있는 최대 거리와 앞, 뒤차의 분리 가능한 거리를 도식화 하면 그림 2 와 같다. 결과적으로 충분한 해상도가 확보되었다고 가정할 경우 카메라의 높이가 높을수록 차량을 검지할 수 있는 거리를 멀리 할 수 있다. 그러나 현실적으로 경제성의 이유로 8m에서 최대 15m 이내에서 카메라 설치 높이를 결정하게 된다. 만일 9m에 설치된 카메라로 90m떨어진 승용차(높이 1.5m)를 검지할 경우, 앞뒤승용차 사이의 최소 거리는 30m 이상이 되어야 한다. 이러한 이유로 대부분의 영상검지기는 검지기가 설치된 근방만을 감시하여, 정보를 취득하도록 설계되어 있다. 본 연구에서는 영상검지기에서 앞뒤차량의 부분 가려짐이 발생하더라도 두 차량을 분리하여 앞뒤차량의 분리가능거리를 좁힘으로써 영상검지기의 검지가능영역을 넓혀 검지기의 근본적인 성능을 향상 시키고자 한다.



<그림 1. 카메라의 설치높이와 겹침현상과의 상관관계>

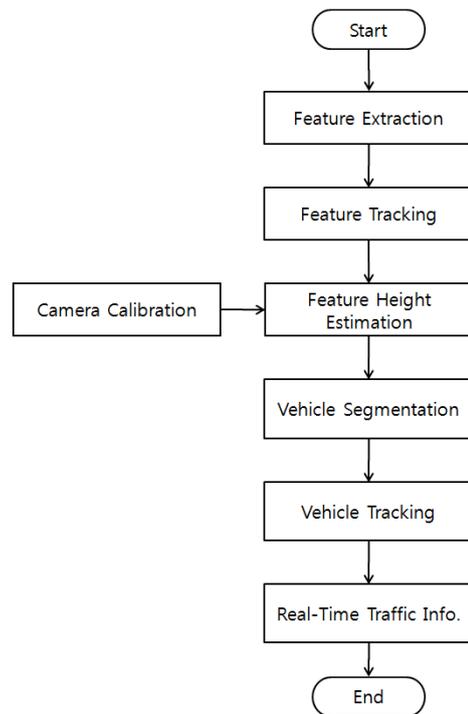


<그림 2. 카메라의 설치높이에 따른 차량의 분리가능 거리>

III. 제안된 겹침현상에 강인한 알고리즘

1. 알고리즘 개요

본 연구에서 제안하는 겹침현상에 강인한 알고리즘의 흐름을 간략하게 정리하면 아래 그림 3 과 같다. 이동체 특징점 궤적의 추적 후, 단일 카메라 보정 정보를 활용하여 특징점의 높이를 추정하고 이로부터 차량을 분류하게 된다. 이후 칼만 필터와 특징점 추적으로 통과 차량을 추적하고 3차원 가상터널을 정하여 관심 구역의 실시간 교통정보를 취득한다. 본 연구가 제안하는 알고리즘을 본 장의 2절부터 7절에 걸쳐 소개한다.



<그림 3. 제안된 알고리즘의 흐름도>

2. 카메라 보정

이동체의 특징점의 궤적을 통해 특징점의 높이를 추정하고, 이로부터 부분적으로 가려진 물체를 분리해 내는 것이 본 연구의 핵심으로 특징점 높이 추정 및 교통정보 연산을 위해 우선적으로 카메라 보정이 우선되어야 한다. 카메라 보정은 실제계에 존재하는 특정 위치가 카메라상에 투영되어 보여질 때 이 둘 사이에 상관관계를 규명하는 것으로, 본 연구에서는 산술적인 방법을 통해 이들의 정방향, 역방향 관계식을 유도하여 사용한다.

그림 4 와 같이, 실제계 좌표계 {R}에 존재하는 한 점 $P = \{X_0, Y_0, Z_0\}$ 카메라를 통해 비추는 카메라 좌표계 {C}

의 한점 $p = \{x_0, y_0, z_0\}$ 표현 될 경우, 이 사이의 상관관계를 규명하는 것이 카메라 보정이며, 이를 행렬식 형태로 표현하면 (식1)과 같다.[16][17]

$$\lambda x = \Pi_0 X = \Pi_0 g X_0 \quad (\text{식 1})$$

여기서 Π_0 는 standard projection, g 는 transformation이다. 결과적으로 3x4 차원을 가지는 Π_0 행렬 및 4x4차원을 가지는 g 행렬의 해를 구하는 것이 카메라 보정 문제이다. 이를 위한 여러 방법들이 연구되어 오고 있으며, 체크 같은 특정 모양의 지그를 사용하는 것이 일반적인 방법이다. 본 연구에서 요구되는 높이 정보에 따른 실제 물리좌표 계산 및 카메라 회전에 따는 정보 보정을 위해서는 앞에서 명시한 보정행렬의 엘리먼트 값뿐만 아니라, 이 엘리먼트의 요소인 실제 카메라 파라미터 정보, 즉 회전각도 $R(\alpha, \beta, \gamma)$ 와 설치위치 $T(x_c, y_c, z_c)$ 및 포커스 거리 f 와 화소크기 $\delta x, \delta y$ (대부분의 경우 $\delta x = \delta y$)를 알 필요가 있다.

따라서 본 연구에서 실세계 좌표계(R)의 바닥면($Z_0 = 0$)에 존재하는 크기를 아는 임의 직사각형의 4개의 꼭지점 P_1, P_2, P_3, P_4 이 영상 안에서 그림 5와 같이 관찰되었을 경우 수리적인 방법으로 해를 구해 사용한다. (식1)에서 $Z_0 = 0$ 라고 가정하면 $\Pi_0 g$ 행렬은 2x2로 간략화 될 수 있으며, 이를 Π_g 라고 한다면, 각 꼭지점의 실세계와 영상사이의 관계는 아래 (식 2)~(식 5)와 같이 표현할 수 있다.

$$p_1 = \Pi_g P_1 \quad (\text{식 2})$$

$$p_2 = \Pi_g P_2 \quad (\text{식 3})$$

$$p_3 = \Pi_g P_3 \quad (\text{식 4})$$

$$p_4 = \Pi_g P_4 \quad (\text{식 5})$$

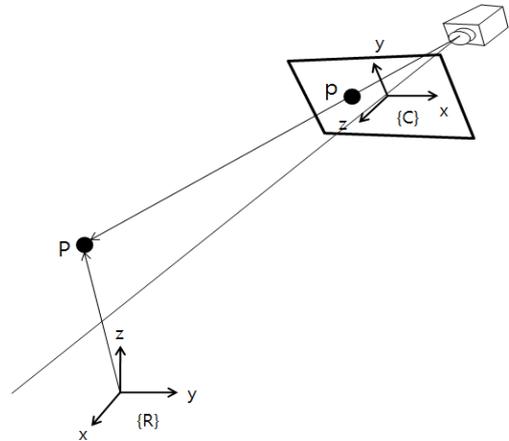
위 (식 2)~(식 5)의 연립방정식의 해를 구해야 하며, 이때 행렬의 엘리먼트를 구하는 선형방정식의 해는 쉽게 구할 수 있겠지만, 우리가 원하는 카메라 파라미터를 얻기 위해 각 파라미터를 대입하여 비선형 방정식을 구해야 하며, 이 경우 산술적인 방법으로는 그 해를 구하기 매우 어렵기 때문에 수리적 방법을 사용한다. 그리고 비선형 다원 고차 방정식의 해를 구하기 위한 일반적이고 강건한 방법은 Newton 방법이 있지만, 이상의 경우와 같은 복잡한 비선형 방정식에는 local-minima에 빠지는 경우가 다수 발생하므로, 전역 수렴이 가능한 Broyden 방법을 사용한다[18].

이상과 같이 카메라 파라미터가 구해지면 임의의 실세계 좌표계(R)에 대한 값에 대해서 영상좌표계(C)에 나타나는 값은 (식 1)을 통하여 구할 수 있게 된다. 반대로 영상에 보여지는 좌표를 실세계에서 어떤 좌표에 있는지 계산하기 위해서는 (식 1)의 역함수를 구하면 가능하다. 그러나 카메라 좌표계의 변수는 2개이고, 실세계 좌표계의 변수는 3개로 단일

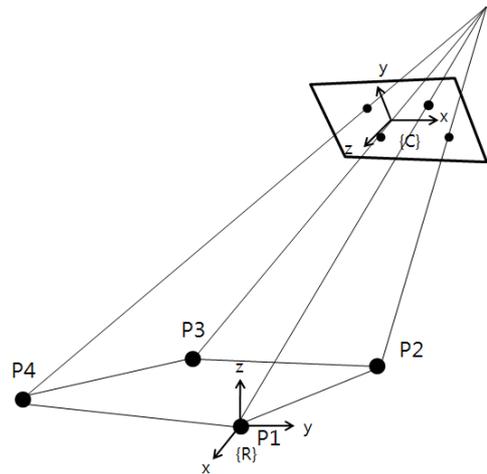
해가 구해지지 않고 무한히 많은 해가 존재한다. 이때 두 개의 다른 카메라로 각각의 카메라 좌표를 구하여 해를 구하는 스테레오 비전같은 방법이 있지만, 본 연구에서는 단일카메라 사용을 위하여 높이를 또 하나의 변수로 입력하고 그 해를 구하면 (식6)과 같다.

$$X_{xy} = \begin{bmatrix} X_0 \\ Y_0 \end{bmatrix} = \Pi(Z_0)^{-1} x_c \quad (\text{식 6})$$

최종적으로 (식 1)로부터 임의의 실세계 좌표로부터 사상(mapping)되는 카메라 좌표와 (식 6)으로부터 카메라에 보이는 한 점이 특정높이일 경우의 실세계좌표를 구해 낼 수 있게 되었으며, 이는 본 연구의 여러 부분에서 핵심적인 역할을 하게 된다.



<그림 4. 실세계 좌표계와 카메라 좌표계의 상관관계 >



<그림 5. 실세계 4점을 이용한 카메라 보정>

3. 특징점 추출

본 연구의 수행을 위한 첫 단계로 포인트 특징점을 찾아 추적한다. 포인트 특징점과 그 추적은 기본적으로 많은 연산이 소요되며, 기본적으로 실시간 처리가 어렵다고 알려져 있

으나, 본 연구에서는 (그림 6)과 같이, 움직이는 물체를 찾는 단계, 움직이는 물체에 대한 특징점 추출단계, 특징점에 대해 피라미드 기법, 정규화 상호 상관도(NCC:Normalized Cross-Correlation) 및 광류(Optical Flow)연산 단계를 거쳐 실시간 처리가 가능하도록 하였다.

움직이는 영상에서 이동체를 검지하기 위해 배경영상을 사용하는 방법은 카메라의 이용이나 광량의 변화가 있을 경우 심한 오차를 발생함으로 환경변화가 급격한 영상검지 시스템에서는 유효하지 않으며, 본 연구에서는 이동체를 찾기 위해서는 연속하는 3개의 예지영상으로부터 각각 이웃하는 영상사이의 절대값과 그 값 사이의 논리적 AND값을 구하면 움직이는 물체만의 에지를 구한다. 이상의 내용을 수식으로 표현하면 (식 7)과 같다.

$$O = \|\hat{I}_{t-2} - \hat{I}_{t-1}\| AND \|\hat{I}_{t-1} - \hat{I}_t\| \quad (식 7)$$

$$M = O \oplus B$$

여기서 O 는 발견된 움직이는 물체가 포함된 예지영상이며, \hat{I}_t 는 시간 t 에서의 예지영상이다. 또한 M 은 마스크 영상으로 물체영상 O 와 서브마스크 $B = [1\ 1\ 1; 1\ 1\ 1; 1\ 1\ 1]$ 와 Dilation 연산을 통해 구하며, 최종적으로 $t-1$ 시간의 이동물체의 영역을 구한다.

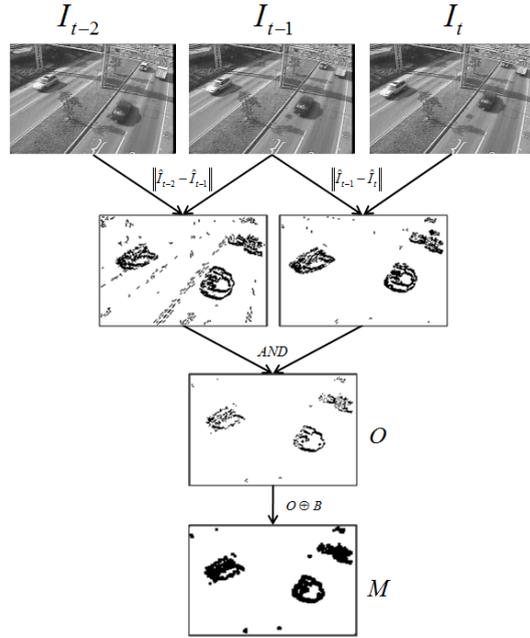
다음으로는 앞에서 구해진 움직이는 물체영역에 대한 포인트 특징점을 추출한다. 포인트 특징점 추출은 모든 점에 대해 (식 8)과 같이 정의된 행렬의 고유값으로 부터 구할 수 있다.[16]

$$G = \begin{bmatrix} \sum i_x^2 & \sum i_x i_y \\ \sum i_x i_y & \sum i_y^2 \end{bmatrix} \quad (식 8)$$

여기서 i_x, i_y 는 특정크기의 마스크 크기를 갖는 X축에 대한 기울기 영상과 Y축에 대한 기울기 영상이다. (식 9)는 Harri's criterion로 알려진 특징점 성질평가 함수이며,

$$C(G) = \det(G) + k \cdot \text{trace}^2(G) \quad (식 9)$$

여기서 k 는 무게변수(weight value)이며, 본 연구에서는 다음 단계에서 특징점의 광류(Optical flow)를 구하기 위해 행렬 G 의 계수(rank)가 2로 역함수를 구할 수 있고, (식 9)에서 구해진 벡터 $C(G)$ 의 엘리먼트 값 중 최소값이 임계값보다 큰 경우를 모두 만족할 때 그 점을 포인트 특징점으로 정의한다.



<그림 6. 실시간 특징점 추출을 위한 물체의 움직임을 추출 과정>

4. 특징점 추적

이상과 같이 선정된 포인트 특징점에 대한 추적을 실시간으로 수행하기 위해 특징점에 대한 정규화 상호 상관도(NCC:Normalized Cross-Correlation) 매칭, 광류(Optical flow), 피라미드 영상 기법을 활용한 추적을 구현하며, 이를 도식화 하면 아래 그림 7과 같다.

먼저 시간 $t, t-1$ 의 영상에 대해 3단계 피라미드 영상을 생성하고, 가장 작은 3번째 단계의 영상(I_{t-1}^3, I_t^3)으로부터 추적을 시작한다. 영상의 변화가 클 경우 광류를 사용한 추적에 실패할 확률이 높아지므로, 첫 번째 수행하는 3번째 단계의 다음과 같이 정의된 정규화 상호 상관도를 구해 매칭점을 찾는다.

$$NCC(h) = \frac{\sum_{w(x)} (I_{t-1}(\tilde{x}) - \bar{I}_{t-1})(I_t(h(\tilde{x})) - \bar{I}_t)}{\sqrt{\sum_{w(x)} (I_{t-1}(\tilde{x}) - \bar{I}_{t-1})^2 (I_t(h(\tilde{x})) - \bar{I}_t)^2}} \quad (식 10)$$

여기서 $I_{t-1}(\tilde{x}), I_t(h(\tilde{x}))$ 는 각각 시간 t 및 $t-1$ 에 대한 비 균일한 영상영역이며, \bar{I}_{t-1}, \bar{I}_t 는 각 영역의 밝기 평균값이다. NCC는 -1에서 1사이의 값을 갖는데 각 포인트 특징점에 대해 임계영역내에서 최대가 되는 값을 찾는다. 여기서 찾아진 포인트 특징점의 매칭 위치의 두배가 되는 점을 2번째 단계의 영상(I_{t-1}^2, I_t^2)의 초기값으로 선정하고 광류 이동 벡터를 구한다.

(식 11)과 같이 정의되는 두 영상의 오차함수가 최소가 되는 값으로 광류는 정의 될 수 있으며, (식 11)의 양변을 편

미분하여 행렬 형태로 정의하면 (식 12)와 같다.

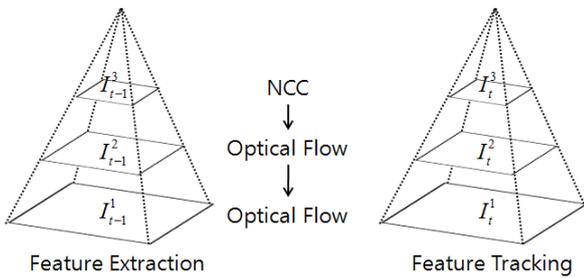
$$E_b(\mathbf{u}) = \sum_{W(x)} \left[\nabla I^T(\tilde{x}, t) \mathbf{u}(x) + I_t(\tilde{x}, t) \right]^2 \quad (\text{식 11})$$

$$\nabla E_b(\mathbf{u}) = \left[\begin{array}{cc} \sum i_x^2 & \sum i_x i_y \\ \sum i_x i_y & \sum i_y^2 \end{array} \right] \mathbf{u} + \left[\begin{array}{c} \sum i_x i_t \\ \sum i_y i_t \end{array} \right] \quad (\text{식 12})$$

여기서 오차가 최소가 되는 점($E_b(\mathbf{u}) = 0$)으로부터 최종적으로 광류 이동벡터 \mathbf{u} 를 구할 수 있다.

$$\mathbf{u} = -G^{-1}\mathbf{b} \quad (\text{식 13})$$

여기서 $\mathbf{b} = [\sum i_x i_t; \sum i_y i_t]$ 이며, i_t 는 I_t, I_{t-1} 사이의 시간축에 대한 기울기 영상이다. 앞서 단계의 NCC 매칭을 통해 구해진 값을 초기값으로 광류 이동벡터 \mathbf{u}_k 를 구하고, 이 이동벡터 만큼 이동한 위치에 대해 다시 이동벡터를 구하는 과정 ($\mathbf{u} = \mathbf{u} + \mathbf{u}_k$)을 이동벡터 값이 '0'이 될 때 까지 반복하며, 첫 번째 단계의 영상(I_{t-1}^1, I_t^1)에도 동일한 방법을 적용하여 최종적인 이동벡터를 구함으로써 추적단계를 종료한다.



<그림 7. 특징점 추적 방법>

5. 특징점의 높이 추정

특징점의 높이를 추정하는 단계는 서로 겹쳐진 물체를 분리해 내기 위한 핵심적인 정보를 제공하며, 바닥면에 투영된 특징점에 대한 실제 위치 계산, 특징점 궤적의 변위벡터 clustering을 통한 기준점 선정, 기준점과 특징점 사이의 물리적 변위의 오차가 최소가 되는 특징점들의 높이 추정 단계로 이루어진다.

아래 그림 8에서 보는 바와 같이 실제에 존재하는 물체의 특징점은 a 나 b처럼 실제로 하나지만 선분 A 또는 B를 따라 무수히 많이 존재하는 점들이 영상 평면상에는 같은 점에 사상된다. 특징점의 높이를 추정하는 문제는 이 무수히 많이 존재하는 점들 중 하나인 벡터 $\mathbf{P}_a(z_a), \mathbf{P}_b(z_b)$ 를 구하기 위해 높이 z_a, z_b 를 선택하는 문제로 다시 귀결되며, 특징점 a의 경우 특징점의 높이가 0이므로 실제 특징점 벡터와 바닥면에 투영된 특징점의 벡터는 같으며($\mathbf{P}_a(z_a) = \mathbf{P}_a(0)$), 강

체인 경우 두 벡터의 차이인 \mathbf{P}_{ab} 는 물체가 이동하더라도 변하지 않으므로 이러한 특성을 이용해 각각의 높이를 추정한다.

우선 각 특징점을 바닥면에 투영된 벡터를 카메라보정의 역변환식인 (식 6)에서 높이 $z=0$ 을 대입하여 구한다. 이전 단계에서 각 특징점들의 이동 궤적을 추적하였으므로 바닥면에 투영된 특징점의 궤적을 알 수 있다. 이 궤적은 이동벡터로 표현될 수 있으며, 영상평면상에는 그 높이에 따라 다르게($\mathbf{P}_b(0, \Delta t_1) \neq \mathbf{P}_b(0, \Delta t_2)$) 나타나는 강체이며, 선형운동을 하는 경우 실제의 이동벡터는 동일하다 ($\mathbf{P}_b(z_b, \Delta t_1) = \mathbf{P}_b(z_b, \Delta t_2)$). 따라서 각 이동벡터를 연립하여 풀면 높이를 구할 수 있으나 실제로 많은 경우 그 해는 이동벡터가 최소가 되는 값 즉 높이가 최대가 되는 값으로 수렴하는 경우가 발생하며, 원하는 해를 구하기 어려워진다.

이 경우 물체의 실제 이동벡터를 구함으로써 문제를 해결할 수 있으나 이것은 사실상 불가능하며 실제 이동벡터를 추정하여야 한다.

이동체 추출단계에서 구한 이동체 마스크영상에서 각 이동체를 레벨링(leveling)하고 각각의 레이블 안에 속하는 특징점들을 같은 물체라 가정한다. 그리고 같은 물체로 가정된 특징점들의 집합에서 특징점의 운동특징을 분류하기 위해 특징점들의 바닥면에 투영된 움직임 벡터 $\mathbf{P}(0, \Delta t)$ 들의 norm과 각도로부터 거리(distance)함수를 (식 14)와 같이 정의한다.

$$d = k_1 \|\mathbf{P}(0, \Delta t)\|^2 + k_2 (\angle \mathbf{P}(0, \Delta t))^2 \quad (\text{식 14})$$

여기서 k_1, k_2 는 각 항에 대한 가중치(weight value)이다. (식 14)의 거리 함수를 퍼지 k-means 알고리즘을 사용하여 clustering 하면 (그림 10)과 같은 형태로 나타날 수 있다. 그러나 (그림 10)과 같이 A,B,C 세 그룹으로 clustering 되었다라도 같은 방향을 갖은 그룹(A,B)는 같은 물체로 볼 수 있으나 다른 방향으로 이동하는 그룹 C는 다른 물체이거나 잡음이 된다.

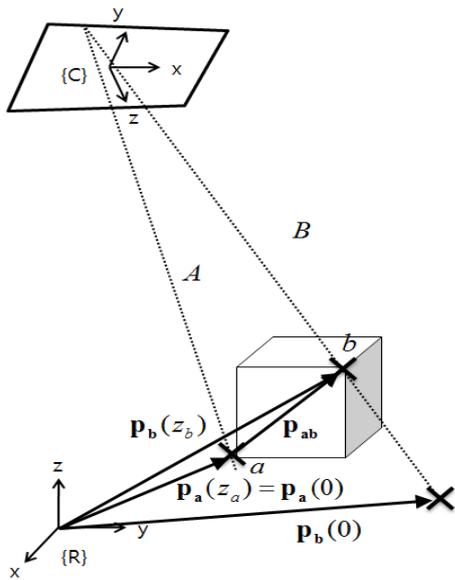
같은 물체로 여겨지는 그룹(A,B) 중 이동량이 적은 그룹 A는 물체의 하부에 존재하는 특징점들로 가정할 수 있으며, 만일 이들이 바닥면에 존재하는 특징점이라면, 이들의 이동거리의 평균인 a 는 물체의 실제 이동거리와 같으므로, a 를 이동체의 실제 이동거리로 둔다. 이상의 정의로부터 각각의 특징점들의 이동벡터를 (식 15)로 정의할 수 있으며, 이를 양변을 제곱하여 정리하면 (식 16)과 같이 정리할 수 있다.

$$\|\mathbf{P}(z_b, \Delta t)\| = a \quad (\text{식 15})$$

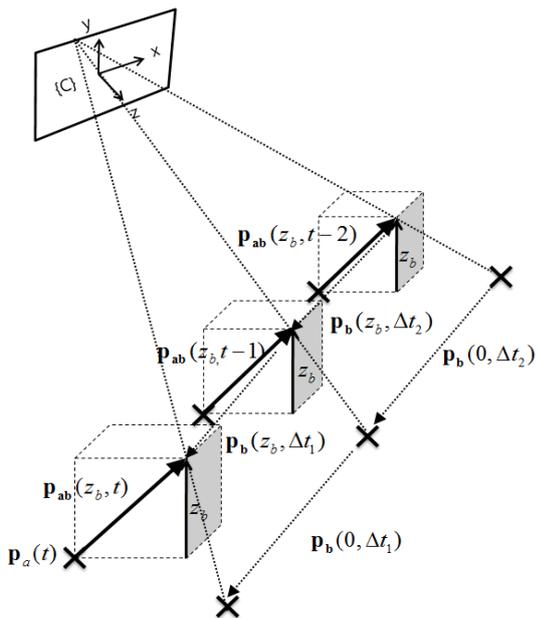
$$\begin{aligned} \sqrt{\mathbf{P}(z_b, \Delta t_1)^2 + \mathbf{P}(z_b, \Delta t_2)^2} &= a \quad (\text{식 16}) \\ \mathbf{P}(z_b, \Delta t_1)^2 + \mathbf{P}(z_b, \Delta t_2)^2 - a^2 &= 0 \end{aligned}$$

(식 16)의 해인 특징점의 높이는 (식 6)을 (식 16)에 대입한

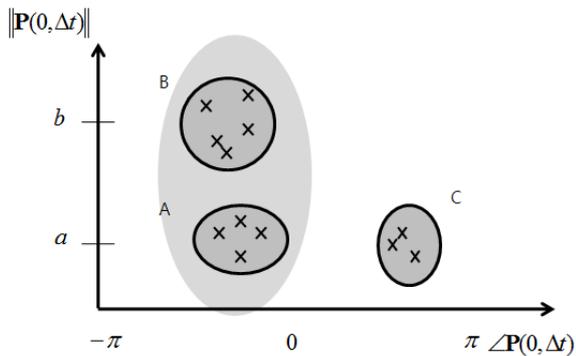
일원 방정식의 해로 쉽게 구할 수 있다.



<그림 8. 실세계 물체에 대한 영상공간으로의 사상 관계>



<그림 9. 이동체의 높이 추정 개념 >



<그림 10. 바닥면에 투영된 움직임 벡터의 클러스터링 >

6. 차량분할

앞서의 단계를 통해 특징점들의 이동궤적, 특징점의 운동 벡터 특성에 의한 그룹과 각 특징점들의 실세계 높이 추정 정보를 획득하였고, 이들 정보와 차량의 형태 정보를 활용하여 특징점들을 분할(segmentation)을 수행한다.

추정된 특징점의 높이를 특징점의 이동궤적에 순차적으로 대입하고, 이로부터 얻어지는 특징점의 실세계의 3차원 벡터들의 상태 거리를 확인한다. 이때 이들 상대 거리가 오차범위 이상으로 변하는 경우 이 특징점은 잡음이거나 다른 물체이므로 제거한다.

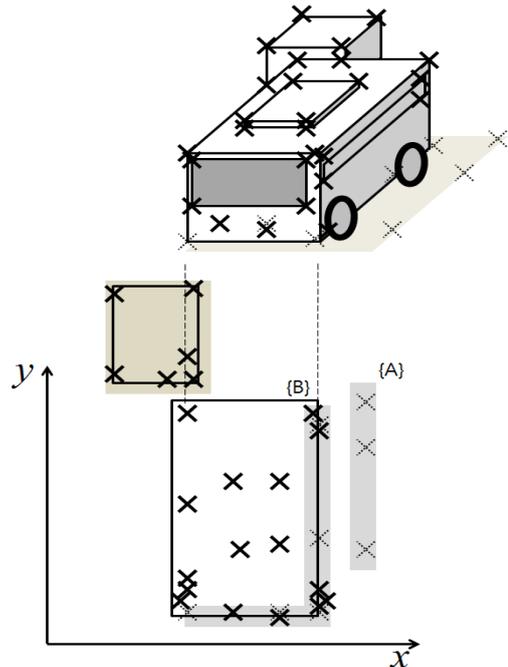
다음으로 (그림 11)의 {A}그룹과 같이 물체의 외곽에 존재하면서 높이가 임계값 이하로 낮은 특징점들은 그림자로 판단하고 제거 한다.

다음으로는 낮은 높이의 특징점과 높은 높이의 특징점이 혼재하는 {B}그룹은 차량의 외곽이며, 차량의 외곽이 발견되면, 임계값 이상의 특징점들의 외곽의 크기로부터 차량의 겹침 상태를 확인한다.

대부분 차량을 그 크기로 구분하면 다음과 같은 범주에 속하게 되며, 앞에서 찾은 차량의 외곽으로부터 특징점이 어떤 범주에 속하는지 찾아서 차량으로 판단된 특징점은 차량으로 구분하고 제거한다. 모든 특징점이 제거될 때 까지 같은 방법을 반복하여 겹쳐진 차량을 구분하여 분리한다.

<표 1. 크기로 분류한 차종>

구분	폭	높이	길이
경차	1.5	1.5	3.5
승용	1.8	1.5	4.5
승합/소형버스/소형트럭	1.8	2.0	5.0
버스/대형트럭	2.5	3.3	10.0
대형트레일러(40pt)	2.5	3.3	15.0



<그림 11. 움직임 벡터 클러스터링에 의한 최종 차량겹침 구분>

7. 차량 추적 및 실시간 교통정보 생성

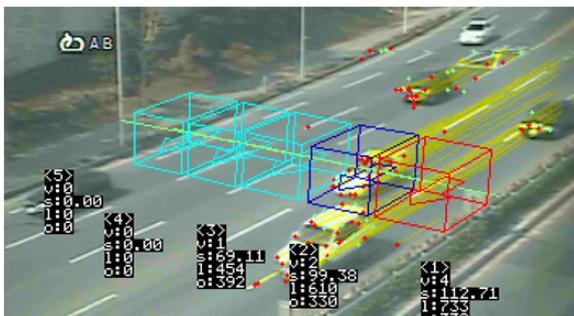
차량의 추적은 선형강인 예측기인 칼만 필터와 특징점 추적을 통해 차량 추적을 수행한다[19]. 차량으로 구분된 특징점들의 이동벡터의 평균값을 차량의 이동벡터로 정의하고, 차량의 외곽은 선택된 범주로 정의하며, 차량의 위치는 외곽의 중심으로 정의한다.

처음 차량으로 구분되면, 특징점들이 차량으로 구분되기까지 여러 장면에 걸쳐 특징점들이 추적되었다. 이 특징점들의 발견 시점부터 차량으로 구분된 현재까지의 궤적을 입력으로 칼만 필터의 계인을 초기화 한다.

필터계인의 초기화가 완료되면, 다음 장면에서의 차량의 위치를 예측한다. 다음 장면에서 특징점을 추적하여 차량에 소속된 특징점이 예측위치의 범위 안에 존재하면 예측위치를 차량의 새로운 위치로 정하고 칼만필터 계인을 갱신한다. 예측범위를 벗어나면, 차량구분을 새로 수행하여 새로운 위치를 얻어 이로부터 필터계인을 갱신한다.

영상검지 시스템이 요구되는 기본적인 교통정보들은 교통량, 통행속도 및 점유율이다. 이상의 정보를 취득하기 위한 기준이 필요한데, 기존시스템의 호환성을 위해 기존 정보를 취득하는 영역을 정의한다. 도로상의 정보 취득을 요하는 위치에 가상의 3차원 터널 영역을 설정하고, 이 터널을 통과하는 차량의 대수를 교통량, 터널을 통과하는 속도를 통행속도 및 터널을 지나는 동안의 시간을 점유시간으로 정의하여 기존 시스템과 호환되는 교통정보를 취득한다. 여기서 통행속도는 차량의 속도가 아니라, 차량의 각 특징점의 속도의 평균으로 정의한다. 이는 속도 정보를 취득하는 샘플수를 증가시켜 확률이론에서 잘 알려진 Central limit theorem에 따라 속도 오차를 줄일 수 있어 더욱 정확한 속도 정보를 취득할 수 있다.

이상의 기본 교통정보 이외 차량의 크기로 구분되는 차량의 차종정보를 취득 할 수 있고, 주행차량의 실시간 현재 위치와 궤적 및 이동속도를 통해 이용차로 이외의 차량 주행(갓길 주행), 불법 주정차, 역주행, 불법 유턴, 교통사고 검지 등의 돌발상황 정보를 취득 할 수 있다.



<그림 12. 기존 차량검지 시스템과 정보호환을 위한 가상 검지 터널>

IV. 실험 및 평가

1. 실험환경 및 실험방법

본 연구의 타당성을 검증을 위한 실험방법은 우선 2장에서 설명한 알고리즘을 그림13에 나타낸 것과 같은 영상검지기 전용 하드웨어에 구현하였으며, 교통량이 많고 차량의 검침이 빈번하게 발생하는 두 가지 실험사이트를 선정 하여, 주간 30분의 영상을 취득하였다. 비교장비로는 교통량과 속도에 대하여 주간의 경우 최상급의 성능을 갖는 당사의 기존 영상검지 장비를 선정하고, 교통량에 한하여 비교 분석을 수행하였다. 속도와 점유율의 경우 그 참값을 구하기 위해서는 별도의 기준검지장비가 요구되며, 본 연구의 핵심 사항인 차량의 검침현상의 개선과 직접적 상관관계 적으므로 제외하였다.

구분	사양
CPU	TMS320DM642-720Mhz (5760MIPS, max)
메모리	64bit - 32Mbyte SDRAM
크기	Half 4U
영상입력	1CH - NTSC or PAL
영상출력	1CH - NTSC or PAL
UART	4CH - RS-232x3, 422x1
Ethernet	10/100 Mbps
기타	RTC, WDT

<그림 13. 실험에 사용된 하드웨어 외형 및 사양>



<그림 14. 시험사이트 1 (과천시,과천대로)>



<그림 15. 시험사이트 2 (안양시,시민로)>

④ Results & Environment

Description	Dawn	Day	Dusk	Night	Average	Class
Traffic(%)	95	95	95	87	93	high
Speed(%)	97	98	98	97	98	highest
Occupancy(%)	91	92	91	85	90	high
Temp. (C)	2~6	2~17	7~17	5~17	-	-
Wind velocity(%)	0~5	0~8	0~6	0~6	-	-
Rainfall(mm)	0	0	0	0	-	-
Relative humidity(%)	72~81	17~79	21~61	36~70	-	-
Vehicle speed(km/h)	49~94	40~82	40~85	40~86	-	-

Refer to the below attachment. "Specific Test Results"

⑤ Calibration Traceability :
 * Baseline data source(laser detector) : Upward 1st-2nd lanes, downward 1st-2nd lanes
 Speed uncertainty : within ± 1% (KTL Cal. No. 04-1552-1461-1~4)

⑥ Tested by : * Name : Jang Jin-hwan * Tel. No. : 031-910-0884

⑦ Term of validity : 11/24/2006 ~ 11/23/2008

<그림 16. 비교장비의 공인성능>

2. 실험결과 및 평가

표 2. 실제 각 장비의 계측된 교통량과 표 3에는 측정 결과의 평가를 위해서 (식 17)에 나타낸 것과 같은 산술 오차식을 사용한 교통량의 오차를 각 차로별로 도시하였다.

$$\left(1 - \frac{T-C}{T}\right) \times 100\% \quad (\text{식 17})$$

여기서 T는 참값, C는 측정값이다. 그리고 각 표에서 A는 기존의 비교 검지기이며, B는 새로운 알고리즘이 적용된 검지기, T실제의 교통량이다. 표 2에서 보여 지는 것과 같이 차량이 검침이 적은 낮은 번호의 검지 차선의 경우 두 검지기 모두 실제 사용하기에 부족함이 없는 결과를 갖지만, 검침이 심한 높은 번호의 차선의 경우 기존 장비에서는 다른 차선을 지나는 차량의 이중 검지로 인해 교통량 측정에 심각한 오차가 발생하지만, 새로운 알고리즘을 적용한 경우 확연한 개선이 이루어 졌음을 확인할 수 있었다.

<표 2. 실험 측정값, 단위 대수>

사이트	차선	1 2 3 4 5 6					
		1	A	285	336	474	502
B	291	322	346	354	401		
T	297	329	312	328	352		
2	A	167	187	198	223	274	274
	B	170	193	200	223	237	243
	T	172	195	204	235	232	221

<표 3. 오차평가 결과, 단위 %>

사이트	차선	1 2 3 4 5 6					
		1	A	96	102	152	153
B	98	98	111	108	114		
2	A	97	96	97	95	118	124
B	99	99	98	95	102	110	

V. 결론

영상검지시스템은 여러 장점이 있지만, 환경에 취약하다는 결정적인 단점을 지니고 있다. 본 연구에서는 이러한 영상검지시스템의 단점을 극복하여, 장점을 극대화 하는 것에 목표를 두고 영상검지시스템의 대표적인 결점인 차량의 검침문제를 개선하는 알고리즘을 제안하고 실험을 통해 본 연구의 타당성을 검증하였다. 이를 통해 궁극적으로는 검지정보의 정확성을 향상시킬 수 있고, 나아가 검지영역의 확장을 통해 카메라 설치위치의 유연성을 제공할 수 있으며, 이로 인해 공사비 등의 경제적 이익도 기할 수 있다. 또한 본 알고리즘은 차량의 특징점 추적을 기본으로 하고 있기에 차량의 분리와 추적을 통해 교통정보 이외의 돌발검지 등의 부가정보를 얻을 수 있다. 본 연구에서는 주간 교통량에 대한 실험 및 평가만 이루어 졌으나, 일몰·일출, 주간·야간, 우천·우설 등의 다양한 환경에서 교통량, 속도, 점유율 등의 중요 교통팩터에 대한 실험 및 평가가 이루어 져야 할 것이며, 아울러 자사의 검지기뿐 아니라 국외 유수의 검지기와의 비교 평가가 이루어 져야 할 것으로 판단된다.

Acknowledgement

본 연구는 국토해양부 교통체계효율화사업의 지원을 받아 수행되었음

참고문헌

- [1] U.S.Department of Transportation Federal Highway Administration, "Traffic Detector Handbook : Third Edition-Volume I", October 2006
- [2] Yuqi Feng, Xiaodong Wang, Peter T. Martin, "Detector Technology Evaluation", Department of Civil and Environmental Engineering University of Utah, Traffic Lab, Nov., 2003
- [3] Weiming Hu, Xuejuan Xiao, Zhouyu Fu, Dan Xie, Tieniu Tan, Steve Maybank, "A System for Learning Statistical Motion Patterns", IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 28, NO. 9, SEPTEMBER 2006.
- [4] Edward Rosten, "High performance rigid body tracking", University of Cambridge, Doctor of Philosophy Thesis, February 2006.
- [5] Atef, S., Arumugam, H., Masoud, O., Janardan, R., Papanikolopoulos, "A vision-based approach to collision prediction at traffic intersections", N.P., ITS(6), No. 4, December 2005, pp. 416-423.
- [6] Zhou, Aggarwal, J.K., "Object tracking in an outdoor environment using fusion of features and cameras", IVC(24), No. 11, 1 November 2006, pp. 1244-1255.

- [7] Sato, K.[Koichi], Aggarwal, J.K., "Temporal spatio-velocity transform and its application to tracking and interaction", CVIU, No. 2, November 2004, pp. 100-128.
- [8] Hotta, K.[Kazuhiro], "A Robust Object Tracking Method under Pose Variation and Partial Occlusion", IEICE, No. 7, July 2006, pp. 2132-2141.
- [9] Imai, J.I.[Jun-Ichi], Kaneko, M.[Masahide], "Visual Tracking in Occlusion Environments by Autonomous Switching of Targets", IEICE, No. 1, January 2008, pp. 86-95.
- [10] Zhu, L.[Lin], Zhou, J.[Jie], Song, J.Y.[Jing-Yan], "Tracking multiple objects through occlusion with online sampling and position estimation", PR, No. 8, August 2008, pp. 2447-2460.
- [11] Pan, J.Y.[Ji-Yan], Hu, B.[Bo], Zhang, J.Q.[Jian Qiu], "Robust and Accurate Object Tracking Under Various Types of Occlusions", CirSysVideo, No. 2, February 2008, pp. 223-236.
- [12] Bennett, B.[Brandon], Magee, D.R.[Derek R.], Cohn, A.G.[Anthony G.], Hogg, D.C.[David C.], "Enhanced tracking and recognition of moving objects by reasoning about spatio-temporal continuity", IVC, No. 1, 1 January 2008, pp. 67-81.
- [13] Greenhill, D.R., Renno, J.P., Orwell, J., Jones, G.A., "Occlusion Analysis: Learning and Utilising Depth Maps in Object Tracking", IVC, No. 3, 3 March 2008, pp. 430-441.
- [14] Loutas, E., Pitas, I., Nikou, C., "Entropy-based metrics for the analysis of partial and total occlusion in video object tracking", VISIP, No. 6, December 2004, pp. 487-497.
- [15] Klein, L.A. "Sensor Technologies and Data Requirements for ITS.Artech House", Norwood, MA. 2001.
- [16] Yi Ma, Stefano Soatto, Jana Kosecka, S. Shankar Sanstry, "An Invitation to 3D Vision", Springer, 2004
- [17] Richard Hartley, Andrew Zisserman, "Multi View Geometry : Second Edition", Cambridge University Press, 2006
- [18] William H. Press, Saul A. Teukolsky, William T. Vetterling, Brian P. Flannery, "Numerical Recipes: The Art of Scientific Computing, Third Edition (2007)", Cambridge University Press, 2007
- [19] M.E Van Valkenburg, M.G. Morgan, Glen Wade, "Kalman Filtering : Theory and Application", IEEE Press, 1985