

교통사고분석을 위한 삼차원 재현장치 구현

권준용⁰ 김용득
 아주대학교 전자공학과
 dragon77@madang.ajou.ac.kr ydkim@madang.ajou.ac.kr

Implementation of Three Dimensional Simulator for Vehicle Accident Analysis

Jun-Yong Kwon⁰ Yong-Deuk Kim
 Dept. of Electronic Engineering, Ajou University

요 약

본 논문에서는 가속도 센서에 의해 사고를 검지하고 사고 전후의 영상정보를 저장하는 차량용 사고분석장치를 설계하였으며 이를 위한 사고 분석 시뮬레이터를 윈도우 기반에서 OpenGL 3차원 그래픽 라이브러리를 사용하여 구현하였다. 이는 알고리즘부와 디스플레이부로 구성되며, 알고리즘부에서는 도로 영상에 대한 영상처리를 수행한다. 여기서 개선된 역원근법에 의해 전처리된 영상을 필터링하여 차선을 검지하고, 검지된 차선을 이용하여 차선 파라미터들을 추출하며, 디스플레이부에서 추출된 파라미터들을 입력받아서 OpenGL 라이브러리 함수를 사용하여 사고를 3차원으로 재현한다.

1. 서론

자동차의 필요성이 증대하면서 안전운전과 자동운전을 구현하고자 많은 연구가 자동차 제조업체를 중심으로 진행되고 있으며, 또한 사용자 측면에서는 공정한 사고 상황에 관한 객관적인 자료를 바탕으로 한 사고상황 재현을 필요로하고 있다.

현재 차량용 사고분석장치의 개발이 진행 중이지만, 가격이 높고, 사용중인 자동차에 설치하는 것이 어려우므로 현재 차량을 사용하고 있는 사용자들에게 그리 많은 호응을 얻지 못하고 있다. 따라서, 본 연구에서는 사용자들에게 쉽게 파괴될 수 있는 저렴한 가격의 차량용 사고분석장치를 개발하여 이를 통한 사고원인의 정확한 분석을 통하여 사고 분쟁을 방지하고 사용자에게 차량의 주행 상태를 알려주고 교통사고를 미연에 방지하기 위하여 적절한 시스템을 구현하였다. 구현된 시스템은 가속도 센서에 의해 사고를 검지하고 사고가 일어나기 전 20초, 그리고, 사고 후 10초 간의 영상을 저장하는 사고분석장치 시스템이며 시뮬레이터에서 저장된 영상을 분석하여 3차원으로 시뮬레이션 하여 사고를 재현함으로써 사고에 대한 객관적 정보와 안전운전을 유도할 수 있는 자료를 제공하고자 한다.

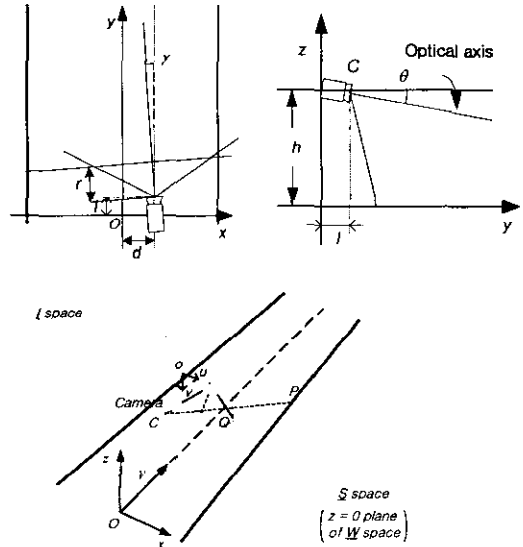


그림 1 카메라 시스템의 기하학적 모델

- (a) W 공간에서 xy 평면 (S space), (b) W 공간에서 yz 평면,
- (c) 두 좌표계 사이의 관계

2. 입체 평면을 복구하기 위한 역원근법

그림 1 은 테스트 차량에 설치된 카메라 모델을 보여준다. 주어진 그림에는 3D 실제 공간 \mathcal{W} ($Oxyz$)와 2D 영상 평면 $\mathcal{I}(Ouv)$ 의 두 가지 공간이 존재한다. $C=(d,h,l)$ 는 \mathcal{W} 에서 카메라의 위치를, γ 는 카메라 수평각을, θ 는 카메라 수직각을 나타낸다. 도로 평면이 평평하다고 가정하면, \mathcal{I} 와 \mathcal{S} 의 관계는 다음의 M_1 과 M_2 두 매핑으로 주어진다.

$$M_1: (u, v) \in \mathcal{I} \rightarrow (x, y, 0) \in \mathcal{S} \quad (1a)$$

$$M_2: (x, y, 0) \in \mathcal{S} \rightarrow (u, v) \in \mathcal{I} \quad (1b)$$

두 매핑은 카메라 외적 파라미터 (카메라 위치와 회전)와 내적 파라미터 (초점 거리, 카메라 해상도)에 의존한다. 변수 f 를 카메라 초점거리라 하면 직관적으로, 점 $P(x,y,0)$ 에 대한 \mathcal{S} 로부터 \mathcal{I} 의 매핑 M_2 는 다음과 같이 표현될 수 있다. ^[1]

$$u = \frac{fx}{y \cos \theta - h \sin \theta} \quad (2a)$$

$$v = \frac{f(h \cos \theta + y \sin \theta)}{y \cos \theta - h \sin \theta} \quad (2b)$$

그리고, M_1 은 다음과 같이 주어진다.

$$x = \frac{ub}{v \cos \theta - f \sin \theta} \quad (3a)$$

$$y = \frac{h(f \cos \theta + v \sin \theta)}{v \cos \theta - f \sin \theta} \quad (3b)$$

$$z = 0 \quad (3c)$$

컴퓨터 그래픽스에서 원근투영 또는 매핑에 대한 반 무한 피라미드 뷰 볼륨에 따라 M_1 과 M_2 의 다른 표현법이 주어질 수 있다. ^[2, 3] 카메라 해상도를 $N \times N$, 카메라 각 개도를 2α 라 하면, $\psi(u)$ 와 $\phi(u)$ 는

$$\psi(v) = (\theta - \alpha) + v \frac{2\alpha}{N-1} \quad (4a)$$

$$\phi(u) = (\gamma - \alpha) + u \frac{2\alpha}{N-1} \quad (4b)$$

도로 평면 \mathcal{S} 에서 주어진 점 P 에 대해, x 와 y 축에 대한 투영은

$$x(u, v) = \frac{h \sin(\phi)}{\tan(\psi)} + a \quad (5a)$$

$$y(u, v) = \frac{h \cos(\phi)}{\tan(\psi)} + b \quad (5b)$$

$$z(u, v) = 0 \quad (5c)$$

만약, $\psi(x,y,0) \in [\theta - \alpha, \theta + \alpha]$ 그리고, $\phi(x,y,0) \in [\gamma - \alpha, \gamma + \alpha]$ 라면, 점 P 는 뷰잉 피라미드에 속하며, \mathcal{I} 에서 좌표 (u,v) 를 γ, θ, d 를 고려하여 표현하면 다음과 같다.

$$u(x, y, 0) = k_u [\phi_{xy0} - (\gamma_0 - \alpha)] - k_u \Delta \gamma \quad (6a)$$

$$v(x, y, 0) = k_v [\psi_{xy0} - (\theta_0 - \alpha)] - k_v \Delta \theta \quad (6b)$$

여기서, $k_v = \frac{N-1}{2\alpha}$ 이며, $k_u = \frac{M-1}{2\alpha}$ 이다.

변수 d 는 단지 수평 이동을 나타내며 식 (6)에서 ϕ_{xy0} 의 계산과 관련이 있으며, $d_0 + \Delta d$ 로 되는 것이 아니라 d_0 로 대체된다. 따라서 ϕ_{xy0} 는 주어진 x 와 y 에 대해 고정된다. 따라서 식 (6a)와 (6b)의 좌변에서 첫 번째 부분은 모두 상수가 된다. 그러므로, 식 (6a)와 (6b)는

$$u(x, y, 0) = u_0(x, y, 0) - k_u \Delta \gamma \quad (7a)$$

$$v(x, y, 0) = v_0(x, y, 0) - k_v \Delta \theta \quad (7b)$$

와 같다. 여기서, $u_0(x,y,0)$ 와 $v_0(x,y,0)$ 는 x 와 y 가 정해지면 상수가 된다.

식 (7)의 장점은 시스템의 신뢰성과 유연성을 떨어뜨리지 않고 하드웨어 시스템의 설계를 단순화할 수 있다는 것이다. $u_0(x,y,0)$ 와 $v_0(x,y,0)$ 는 미리 하드웨어 시스템의 룬에 저장되고, 주어진 도로 영상에 대해, $\Delta \theta$ 와 $\Delta \gamma$ 는 고정된다. 그러므로, $u(x,y,0)$ 와 $v(x,y,0)$ 는 적은 계산 비용과 단순한 하드웨어 구조로 식 (7)에 의해서 쉽고 빠르게 얻어질 수 있다.

3. 제안된 입체 평면으로부터의 차선 감지 방법

복구된 3D 도로 평면으로부터 새로운 차선 감지 방법을 블록도에서 보이고 있다.

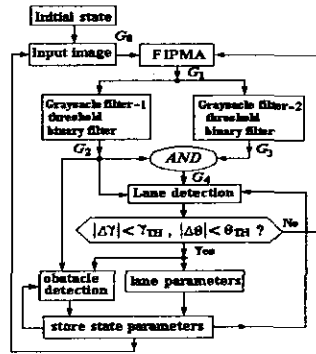


그림 2 차선 감지 방법에 대한 블록도

프레임 n 번째에서의 차선 파라미터는 θ_n, γ_n, d_n , 차선 폭 w_n , 차선 시작 위치 (x_{L0} 와 x_{R0}), 차선 곡률 $1/R_n$ 이며, 차선을 찾기 시작할 때, 파라미터 θ, γ, d 는 초기값인 θ_0, γ_0, d_0 로 세팅된다. 차선 감지는 그림 3에서, 도로 영상이 입력되면, 먼저 역원근법 (IPM) 알고리즘에 의해 3D 도로 평면으로 매핑된다. 그 결과가 맥스칸 윈도우를 단순화한 가중 필터와 평균 필터에 의해 필터링되어 차선을 추출할 수 있게 된다.

필터링된 결과는 각각 이진화한 후, 사이즈 필터와 형태론적 필터에 의해 필터링된다. 여기서 나온 결과들을 AND하여 차선을 찾는데 사용한다. 그리고, $\Delta \gamma$ 와 $\Delta \theta$ 를 계산하여 γ_{TH} 와 θ_{TH} 보다 크게 되면 γ 와 θ 를 갱신하게 된다. 결과적으로 갱신된 카메라 파라미터를 사용하게 되는 역원근법 (IPM) 알고리즘은 정확한 3D 도로 평면을 얻기 위해 재 수행된다.

차선의 위치를 결정하기 위해, 로컬 히스토그램을 사용한다. 로컬 히스토그램을 사용하여 현재 차선을 찾는데 이전 차선의 위치에 대한 정보를 사용하여 차선 구조를 국부화하게 되어 보다 효율적으로 차선을 찾을 수 있게 된다. 그에 따른 피팅 (fitting) 라인 L_L 와 L_R 는 다음과 같이 평가된다.

$$L_L : x = l_L y + x_{L0} \quad (11a)$$

$$L_R : x = l_R y + x_{R0} \quad (11b)$$

여기서 x_{LD} 와 x_{RD} 는 두 개의 라인 L_L 그리고, L_R 과 수평 축 사이의 교점이다. t_L 와 t_R 는 L_L 와 L_R 의 기울기이며, 차선이 새로 측에 평행하다면, t_L 와 t_R 는 0에 가깝게 된다.

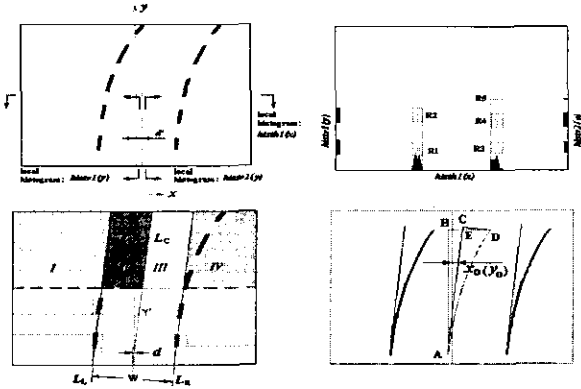


그림 3 차선 검지

- (a) 히스토그램을 위한 영역, (b) 차선의 히스토그램
- (c) 차선 검지 및 검지 평가, (d) 차선의 검지 평가

V. 시뮬레이션 실험

다음은 일련의 영상들을 처리한 결과이다. 그림 4는 원 영상과 그 영상에서 차선을 검지한 결과 그리고, 역원근법(IPM)에 의해 처리된 결과 영상을 각각 보이고 있으며, 그림 7(a)는 도로의 방향을 의미하는 카메라 수평각도(Camera yaw angle)를 추출한 결과이며, 그림 7(b)는 차량의 움직임을 의미하는 차량 앞 28m 앞에서의 측면 오프셋값을 구한 결과를 보이고 있다.

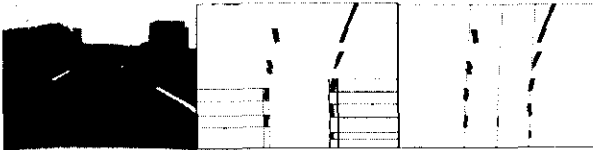


그림 4 차선 파라미터 추출 과정

- a) 도로 영상, b) 하위 영역 마킹, c) 역원근법(IPM) 결과 영상

역원근법(IPM)에 의해 차선을 검지하고 차선 파라미터를 추출하여, 그 추출된 데이터를 이용해서 3차원 시뮬레이션을 할 수 있다. 3차원 시뮬레이션을 위해 일단, 도로의 지리적 정보가 필요하다. 도로의 지리적 정보는 영상 데이터에서 추출하는 것이 어렵기 때문에, 지리정보시스템(GIS) 데이터를 이용하여 시뮬레이션을 위한 도로 환경을 잡는다. 그리고, 알고리즘에서 추출된 차선 파라미터들을 이용해서 차량의 움직임을 시뮬레이션한다. 차선과 차량이 이루는 각도와 차량이 차선으로부터 떨어진 거리를 이용해서 차량의 실제적인 움직임을 표현하고 있다.

다음은 실제로 3차원 시뮬레이션한 결과를 보이고 있다. 원도우 환경에서 개발하였으며, 컴파일러는 비주얼 C++ 를 이용

하였다. 그리고, 3차원 그래픽 라이브러리는 OpenGL^{17, 8} 의을 이용하였다. 차량의 세부적인 모델링은 하지 않았으며, 직육면체를 이용해서 차량을 표현하였다. 구현된 예를 보면, 여러 각도에서 볼 수 있는 팬(Pan) 기능이 있으며, 줌(Zoom) 기능이 있어서 크기를 자유 자재로 변경할 수 있다.

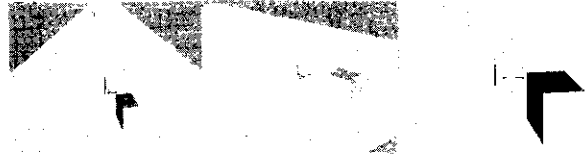


그림 5. 3차원 시뮬레이션

- a) 시뮬레이션, b) 팬(Pan) 효과, c) 줌(Zoom) 효과.

VI. 결론

차량의 주행 중 주행 상태를 알 수 있는 요인으로는 차량의 속도, 브레이크 작동상태, 방향 지시등의 점등 유무, 핸들의 비틀림각, 엔진의 상태 등을 종합하여 판단되어질 수 있다. 이러한 많은 데이터를 모두 획득하기 위해서는 시스템의 구성이 복잡해지며 더욱 정교하게 만들어야 하므로 많은 어려움을 수반하게 된다. 그러므로, 많은 데이터 중 차량의 상태를 직접적으로 파악할 수 있는 데이터에 대한 선정과 획득이 중요하다.

본 연구에서는 이러한 단점을 극복하여 차량의 주행 상태를 잘 표현해주는 차량의 가속도와 상황분석을 위한 주행 영상만을 획득하는 시스템을 설계하여, 이로부터 획득된 영상을 분석하여 차선 파라미터들을 추출하는 기법을 연구하였다. 시스템은 구성이 간단하여, 가격이 저렴하고, 차량을 개조하거나 설치하는데 어려움이 없으므로 시장성 확보에 장점을 가질 수 있다. 또한 추출된 차선 파라미터를 이용하여 사고 상황을 3차원으로 재현해보고 영상을 분석함으로써 사고에 대한 객관적인 판단과 자료를 제공함으로써 교통사고로 인하여 발생하는 분쟁을 효과적으로 해결할 수 있도록 도와 줄 것이며, 안전운전을 유도하는 시스템에서도 이용될 수 있을 것이다.

VII. 참고문헌

- [1] M. Bertozzi and A. Broggi, "Vision-based vehicle guidance", *IEEE Computer*, vol.30, no.7, p.49-55, 1997.
- [2] M. Bertozzi, A. Broggi, "Tools for code optimization and system evaluation of the image processing system PAPRICA-3", *Journal of Systems Architecture*, vol.45, no.2, p.519-542, 1999.
- [3] F. Burger and D. Gillies, "Interactive computer graphics: function, procedural and device-level methods", Addison-Wesley Publishing Company, 1989.
- [4] Ron Fosner, "OpenGL Programming for Windows 95 and Windows NT", Addison Wesley Developers Press Massachusetts, 1996