

도시고속도로를 위한 실시간 가변 속도 제한 (Real-Time Variable Speed Limits for Urban Freeway)

조 영 태 * 정 인 범 **
(YoungTae Jo) (InBum Jung)

요약 IT 기술은 최근 바이오 기술(BT), 나노 기술(NT) 등의 다른 기술들과 융합하면서 새로운 패러다임을 창출하고 있다. 교통 분야에도 IT 기술과 융·복합하여 효과적인 교통 환경을 조성하는 지능형 교통 시스템(ITS)에 관한 연구가 활발히 진행되고 있다. 가변속도제한(VSL)은 ITS 연구의 한 분야로 현재 도로 상황에 알맞게 속도제한을 조절하여 도로의 안전성과 효율성을 증대하는 시스템이다. 기존에는 대부분 단일 스테이션에서의 VSL 알고리즘 연구에 중점을 두었다. 하지만 도시고속도로와 같이 다수의 스테이션이 일정한 간격으로 설치되어 있는 환경에 적용하기에는 미흡한 점이 있다. 본 논문에서는 다수의 스테이션들 사이의 연계를 통해 VSL로 얻을 수 있는 효과를 높이는 알고리즘을 제안한다. 제안하는 알고리즘은 30초 단위의 VSL 변화로 도로 상황에 빠르게 반응 할 수 있다. 알고리즘은 교통 혼잡을 유발하는 스테이션 검색 루틴과 교통 혼잡의 크기 산출 루틴, VSL 컨트롤 스테이션 개수 계산 루틴, VSL 계산 루틴의 총 4 단계로 구성된다. 제안하는 알고리즘은 도로의 안전성 향상과 차량 이동시간 변화량의 최소화를 목적으로 한다. 이동 시간을 고려하는 이유는 VSL의 도로 적용 시 운전자가 가장 민감하게 반응하는 부분이 이동시간의 변화이기 때문이다. 실험 평가를 위해 마이크로스코픽 시뮬레이터인 PTV사의 VISSIM 시뮬레이터를 사용한다. 실험 결과를 통해 제안하는 알고리즘이 도로의 안전성 향상에 기여하고 이동시간에 최소한의 영향을 미치는 것을 보인다.

키워드 : 지능형교통시스템, 가변속도제한, 도시고속도로

Abstract Recently, the convergence of information technology with bio-technology, nano-technology or other technologies has been creating a new paradigm. In a field of transportation, the intelligent transport systems which is a convergence of intelligent technologies and transportation systems have been studied. The Variable Speed Limit(VSL), is one of ITS technologies, is thought to improve safety and efficiency of transportation while controlling speed limit based on road conditions. Legacy studies have considered only one station for VSL algorithm. However, it is not appropriate for an urban freeway installed with many stations. In this paper, new algorithm is proposed to not only enhance effectiveness of VSL based on cooperation of stations but also reflect road conditions within 30 seconds. The proposed algorithm consists of 4 steps: the first is a "searching bottleneck station" step, the second is a "calculating a size of congestion" step, the third is a "calculating the number of controlled stations" step, the final is a "calculating VSL" step. This algorithm guarantees improved safety and minimum additional travel time. The travel time should be considered because drivers would against the VSL algorithm when the proposed algorithm occurs additional travel time. In our experiments, microscopic traffic simulator VISSIM is selected to perform a modeling work. The results show that proposed algorithm provides the improved safety and minimum increase of travel time.

Key words : Intelligent Transport System, Variable Speed Limit, Urban Freeway

* 이 논문은 2010년도 정부 교육과학기술부의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(No. 2010-0024458)

* 학생회원 : 강원대학교 컴퓨터정보통신공학과
ytjoe@snslab.kangwon.ac.kr

** 종신회원 : 강원대학교 컴퓨터정보통신공학과 교수
ibjung@snslab.kangwon.ac.kr

논문접수 : 2010년 8월 5일

심사완료 : 2010년 9월 7일

Copyright©2010 한국정보과학회 : 개인 목적이나 교육 목적인 경우, 이 저작물의 전체 또는 일부에 대한 복사본 혹은 디지털 사본의 제작을 허가합니다. 이 때, 사본은 상업적 수단으로 사용할 수 없으며 첫 페이지에 본 문구와 출처를 반드시 명시해야 합니다. 이 외의 목적으로 복제, 배포, 출판, 전송 등 모든 유형의 사용행위를 하는 경우에 대하여는 사전에 허가를 얻고 비용을 지불해야 합니다.

정보과학회논문지: 컴퓨팅의 실제 및 레터 제 16 권 제 10 호(2010.10)

1. 서론

산업 혁명 이후 가장 빠르게 발전하고 있는 기술 중 하나는 IT(Information Technology) 기술이다. IT 기술은 컴퓨터·소프트웨어·인터넷·멀티미디어 등 정보화에 필요한 유무형 기술을 처리하는 기술이다. IT 기술에서 가장 기본적인 기술은 정보처리를 위한 컴퓨터와 그 안에서 실질적인 정보처리를 하는 소프트웨어 그리고 통신 기술 일 것이다. 최근 저전력, 저비용 통신기술과 MEMS(Micro-Electro-Mechanical-Systems)기술의 비약적인 발전으로 컴퓨터의 크기는 초소형화 되고 그 처리 능력은 커지게 되었다. 컴퓨터는 사람의 눈에 잘 보이지 않을 만큼 작아서 사람이 인지하지 못하는 곳에서 사람을 위해 동작하는 수준에 까지 이르게 되었다. 이러한 기술발전은 유비쿼터스 컴퓨팅 환경을 창조하게 되었다. 유비쿼터스란 물이나 공기처럼 시공을 초월해 '언제 어디에나 존재한다'는 뜻의 라틴어로, 사용자가 컴퓨터나 네트워크를 의식하지 않고 장소에 상관없이 자유롭게 네트워크에 접속할 수 있는 환경을 말한다. 유비쿼터스화가 이루어지면 가정·자동차는 물론, 심지어 산꼭대기에서도 정보기술을 활용할 수 있다. 이러한 IT 기술은 최근 BT(Bio-Technology), NT(Nano-Technology) 등 다른 기술들과 융합되어 IT 융합 기술로 급속도로 발전하고 있다. 그중 교통 분야와 IT가 융합된 기술을 지능형 교통시스템(ITS: Intelligent Transport Systems)이라고 한다.

기술 및 산업의 급속한 발전으로 인하여 개인의 삶은 윤택해지고 자동차를 소유한 사람들의 수도 늘어나게 되었다. 급속한 자동차의 증가는 도로의 교통 혼잡을 가중시켰고 이를 해결하기 위해 도로 확충과 같은 물리적 해결방법을 사용해 왔다. 하지만 이러한 방법은 그 비용이 크고 최근에는 이러한 해결 방법에 한계가 오고 있는 실정이다. ITS는 이러한 문제를 저비용, 고효율로 해결하기 위해 교통 분야에 IT를 융합한 것으로 기존의 교통 시설물을 좀 더 효율적으로 이용하고 도로위의 안전성을 향상시키는 것을 목적으로 한다.

가변속도제한(VSL: Variable Speed Limit) 기술은 도로위의 속도제한을 도로 상황에 맞게 변경함으로써 기존 도로 사용에 있어 그 효율성을 높이고 운전자의 안전도를 향상시키는 ITS 기술 중 하나이다. VSL은 다양한 정보를 표시하는 장치인 DMS(Dynamic Message Sign)와 디텍터로 구성된 스테이션 단위로 구성되며 날씨나 교통흐름, 그 외 다양한 조건에 의해 컨트롤된다. 예를 들어, 눈이 오거나 비가 내려 도로 상태가 좋지 않다면 속도제한을 낮춰 도로상황에 맞는 안전한 속도를 제시함으로써 운전자의 안전을 도모할 수 있다. 또한 자동차의 진행방향에서 다음 스테이션에 교통사고나 차량의 증가로 인해 교통 혼잡이 발생하였을 때 진입 차량

에게 순차적으로 속도를 감속할 수 있도록 속도제한을 조절한다면 안전도를 증가시킬 수 있을 뿐 아니라 때에 따라 교통 혼잡을 완화시킬 수도 있을 것이다. 기존의 VSL 알고리즘은 대부분 단일 스테이션을 위한 알고리즘 연구에 중점을 두고 있다. 이러한 알고리즘은 도시고속도로와 같이 다수의 스테이션이 가까운 거리를 두고 설치되어 있는 환경에 적용하기엔 무리가 따른다. 따라서, 넓은 범위에서 스테이션 간 연계동작을 하는 VSL 알고리즘의 연구가 필요하다.

본 논문에서는 도시고속도로에서 스테이션 간 연계 동작 기반의 실시간 VSL 알고리즘을 제안한다. 제안하는 알고리즘은 실시간으로 속도, 밀도, 교통량 등 교통 데이터를 수집하여 VSL을 결정한다. 도로상황에 빠르게 대응하기 위해 알고리즘은 30초 단위로 교통데이터를 수집하고 수집된 데이터를 기반으로 VSL을 계산한다. 알고리즘은 크게 4단계로 분류된다. 첫째, 도로위에 교통 혼잡을 유발하는 지점이 있는지 검사한다. 둘째, 교통 혼잡을 유발하는 지점이 있다면 해당 교통 혼잡의 크기가 얼마인지 파악하고 교통 혼잡을 유발하는 지점이 없다면 30초 후 다시 첫 번째 단계로 넘어간다. 셋째, 검사된 교통 혼잡의 크기 및 위치를 기반으로 VSL 컨트롤을 해야 할 구역을 선정한다. 넷째, 선정된 VSL 구역을 기반으로 VSL 값을 계산한다. 제안하는 알고리즘은 도로의 안전도를 높이고 VSL 컨트롤에 따른 이동시간의 증가를 최소화함으로써 도로 사용의 효율성을 높이는 것을 목적으로 한다. 알고리즘의 효율성을 측정하기 위하여 마이크로소프트 시뮬레이터를 통하여 실험을 진행하였다.

본 논문의 구성은 다음과 같다. 2장에서는 ITS와 VSL의 연구 동향을 살펴본다. 3장에서는 실험 도로 및 시뮬레이션 환경에 대해 설명한다. 4장에서는 본 논문에서 제안하는 고속도로를 위한 실시간 VSL 알고리즘에 대해 알아본다. 5장에서는 제안하는 VSL 알고리즘의 성능평가를 기술하고 6장에서는 결론 및 향후 계획을 제시한다.

2. 관련연구

2.1 ITS

인간이 산업혁명을 이룬 후 탈것은 말과 같은 동물에서 자동차로 변화하였다. 자동차의 발달은 사람에게 보다 편리한 이동 수단을 제공해 왔고 산업 분야에 빠른 이송을 가능하게 하였다. 하지만 산업 및 기술의 발전으로 자동차의 보급이 급속도로 증가하여 교통의 혼잡은 갈수록 심화되고 있다. 이러한 문제를 해결하기 위해서 도로확충과 같은 물리적 시설물의 확장을 사용해 왔다. 하지만 물리적 확장은 그 비용이 상당히 크고 지리적

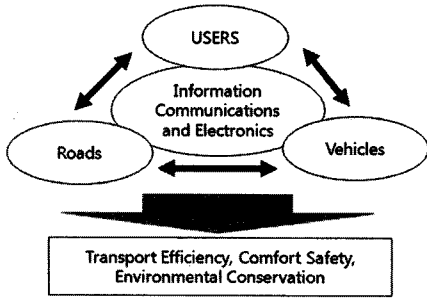


그림 1 ITS 개념도

위치에 따라 물리적 확장이 불가능 할 수도 있다. 이러한 문제를 해결하기 위해 첨단 IT 기술을 이용해 기존 시설물을 좀 더 효율적으로 이용하고자 하는 움직임의 일환으로 ITS가 연구되고 있다.

ITS는 그림 1과 같이 IT와 교통 분야가 융합된 것으로써 도로, 차량, 신호시스템 등 기존 교통체계의 구성 요소에 전자·제어·통신 등의 지능형 첨단기술을 접목시켜 구성요소들이 상호 유기적으로 작동 되도록 하는 차세대 교통체계라고 할 수 있다. ITS는 차량과 교통 시설물 양쪽에 안전성 증가와 쾌적성 증가, 효율성 증가, 환경영향의 감소를 목적으로 현재 여러 국가에서 연구, 발전되고 있다[1,2].

ITS를 구분하는 기준은 다양하지만 가장 크고 간단하게 나누어 보면 도로 시설물측면의 ITS와 자동차측면의 ITS로 구분할 수 있다[3]. 도로 시설물 측면의 ITS는 적절한 교통 통제를 도로 시설물을 통해 운전자에게 제공하는 것으로 이러한 어플리케이션의 예로는 가변 속도제한(Variable Speed Limits), 나들목 진입 통제 시스템(Ramp Metering), 교차로 충돌 방지 시스템(Intersection Collision Warning System) 등이 있다. 자동차 측면의 ITS는 자동차 차체에서 수집되는 정보 또는 도로 시설물로부터 수집된 정보를 이용해 자동차의 장비를 통해 운전자에게 서비스를 제공하는 것을 말한다. 이러한 어플리케이션의 예로 졸음 방지 시스템(Drowsy Driver Warning System), 장애물 감지 시스템(Obstacle Detection System) 등이 있다.

본 논문에서 서술하고 있는 VSL 시스템은 도로시설 기반 ITS로 운전자에게 도로 상황에 맞게 적절한 가변 속도제한을 제공하여 운전자의 안전을 도모하고 도로의 효율적 사용을 목적으로 한다.

2.2 VSL

고속도로에서의 교통사고는 일반 국도에서의 사고에 비해 그 심각도가 상당히 높다. 고속도로에서의 교통사고를 줄이기 위해 다양한 연구가 진행되어 왔으며 그 중 하나가 VSL 시스템이다. VSL 시스템은 기존의 고

정 속도제한과는 달리 도로의 각종 상황 정보를 실시간으로 측정하여 적절한 속도제한으로 변경하는 시스템을 말한다. VSL 시스템은 교통상황을 감지하는 센서(Detector)와 센서로부터 감지된 데이터를 저장 및 처리하는 통제시스템, VSL을 표시하기 위한 DMS(Dynamic Message Sign)으로 구성된다. 그리고 VSL 수행 단위는 DMS와 디텍터로 구성된 스테이션으로 구분할 수 있다. VSL은 여러 연구에서 교통 흐름의 안전성을 향상 시키고 운전자의 운전 스트레스를 감소시키는 것으로 확인되어 다양한 분야에서 VSL을 적용하는 연구가 진행되고 있다.

고속도로의 공사현장에 사용되는 VSL은 공사현장에 진입하기 전에 공사로 인해 지체된 교통상황에 따라 적절하게 변경된 속도제한을 제공함으로써 안전도와 도로 효율성을 향상시키는 것을 목적으로 한다[4,5]. 다른 연구로는 고속도로의 교통 데이터를 실시간으로 측정하여 고속도로의 안전성을 향상 시키려는 연구가 진행되었다 [6,7]. 앞서 제시한 VSL 연구는 속도, 밀도, 교통량과 같은 교통 데이터를 기반으로 동작하는 VSL이었던 반면 [8]은 날씨 데이터를 기반으로 동작하는 VSL 연구를 진행하였다. [9,10]은 VSL 컨트롤과 나들목 교통 컨트롤(Ramp Metering)이 연동되어 동작하는 통합 시스템을 제안하였다.

[6]의 연구는 안전도를 향상시키기 위해 VSL 알고리즘에 CP(Crash Potential)값을 도입하는 것을 제안하였다. 실시간으로 측정된 CP를 기반으로 안전도를 측정하여 VSL을 결정하고 제안된 VSL 알고리즘이 안전도 향상에 기여함을 증명하였다. 하지만 각각의 단일 스테이션의 위치만을 고려하여 VSL을 결정하기 때문에 다수의 스테이션을 위한 VSL을 결정해야 하는 도시고속도로에 적용하기에는 미흡한 점이 있다. 또한 VSL 지속 시간으로 5분 ~ 10분을 이상적인 시간으로 보고 있다. 실시간으로 도로상황을 파악하여 빠르게 대응해야 하는 도시고속도로에 5분 ~ 10분의 시간은 너무 긴 시간이다. 만약 교통사고가 발생하거나 어떤 다른 이유에 의해 교통체증이 시작되고 5분 ~ 10분이면 이미 차량이 막혀 큐가 생성되기에 충분한 시간이다. [7]의 연구 역시 다수의 스테이션 간 연계성을 고려하지 않고 단일 스테이션을 위한 알고리즘을 제안하였다. 따라서, 실시간으로 도로 상황을 파악하여 빠르게 VSL에 반영할 수 있고 스테이션 간 연계성을 고려한 VSL 알고리즘에 대한 연구가 필요하다.

3. 실험 환경

3.1 실험 도로 환경

실험 도로로 미국 미네소타주의 미네아폴리스 도시

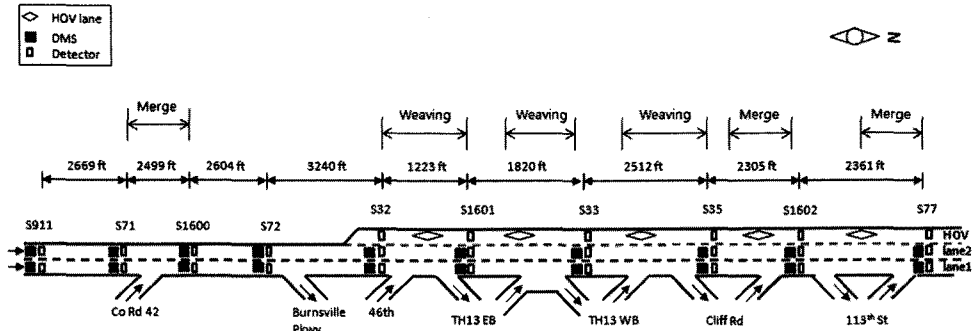


그림 2 실험 도로 (I-35W NB)

안에 위치해 있는 I-35W NB 중 그림 2와 같이 Co Rd 42부터 113th St 구간을 사용하였다. 이 구간은 아침 시간에 많은 출근 차량이 사용하는 도로로 빈번한 교통 체증과 사고가 발생하는 도로이다. 이 구간은 11개의 진출입 지점이 존재하고 2개의 일반 차선과 1개의 HOV (High Occupancy Vehicle) 차선이 있으며 65mile/h의 고정 속도제한을 가진다. 이 구간에는 1200ft ~ 2600ft 마다 디텍터로 이루어진 스테이션이 있고 각 스테이션마다 속도(Speed), 밀도(Density), 통행량(Flow) 데이터를 30초마다 수집한다.

해당 구간은 아침 출근 시간대인 6:00 - 9:00AM에 빈번한 교통 체증이 발생한다. 교통체증은 Cliff Rd와 TH13 WB, TH13 EB에서 진입하는 교통량과 TH13 WB로 빠져나가는 교통량이 많아 병목현상이 일어나기 때문에 빈번히 발생한다. 교통 체증이 발생하면 교통 밀림 현상은 Cliff Rd에서 시작하여 보통 Co Rd 42까지 밀려 내려가고 이 구간의 차량 속도는 30mile/h 이하로 떨어지게 되며 STOP and GO 현상이 발생하게 된다.

3.2 시뮬레이션 환경 및 교정(Calibration)

시뮬레이션 환경 구축을 위하여 PTV사의 VISSIM 시뮬레이터를 사용하였다[11]. VISSIM은 독일의 PTV사에서 개발된 마이크로소프트 시뮬레이터로 다양한 도로 환경을 구현, 실험할 수 있는 시뮬레이터이다. 또한 COM Interface를 제공해 JAVA, C++, Python 등 다양한 언어를 통해 시뮬레이터를 컨트롤 할 수 있다.

실험을 위하여 그림 2와 같이 HOV 차선을 제외한 각 스테이션마다 DMS를 설치하였다. HOV는 이용하는 차량이 많지 않아 교통체증이 발생하지 않는 차선이다.

따라서 해당 차선에는 DMS를 설치하지 않고 VSL에 사용하는 데이터 역시 HOV 차선을 제외한 나머지 차선의 데이터만 사용한다.

[12]의 연구 결과에 의하면 운전자들은 실제 속도 제한보다 8-16km/h(4.9-9.9mile/h) 높은 속도로 주행하는 것으로 밝혀졌다. I-35W역시 측정된 평균 속도를 보면 65-80mile/h의 분포를 그리며 평균 70mile/h로 속도 제한보다 5mile/h 가량 높게 나타난다. 따라서, 시뮬레이션의 운전자 운전속도 패턴을 최대 속도제한보다 10mile/h 크게 설정하고 평균 속도는 5mile/h로 설정하였다. 예를 들어 속도제한이 60mile/h라면 측정되는 평균 속도는 65mile/h 이고 속도 분포는 최소 60mile/h, 최대 70mile/h를 가진다.

시뮬레이션을 통해 교통 통제 시스템을 실험하기 위해서 무엇보다 먼저 수행되어야 하는 것은 교정(Calibration)작업이다. 교정 작업이 제대로 수행되지 않으면 시뮬레이션이 실제 도로의 교통 상황을 제대로 반영하고 있다고 볼 수 없기 때문이다. 교정을 위해 UC Berkely에서 수행한 Vehicle-Following Behavior 값을 변경하는 방법을 참고하였다[13]. 본 논문에서는 그림 2와 같이 Freeway, Weaving, Merge로 분류하여 CC값을 수정하였다. 수정된 CC는 CC0, CC1, CC4, CC5, Waiting Time before Diffusion 값으로 해당 값은 표 1과 같다.

그림 3은 교정된 시뮬레이션의 교통량(5회의 시뮬레이션 결과)와 실제 교통량을 각 스테이션별로 비교한 그래프이고 그림 4는 속도를 비교한 등고선 그래프이다. 그림 4의 (a)는 실제 측정된 속도의 등고선 그래프이고 (b)는 VISSIM의 디폴트 파라미터를 이용한 결과이다.

표 1 교정된 파라미터 값

Link Type	CC0	CC1	CC4/CC5	Waiting Time before Diffusion
Freeway	5.8ft	1.0 s	-2.0/2.0	1 s
Merge	5.8ft	1.0 s	-2.0/2.0	60 s
Weaving	5.8ft	1.3 s	-2.0/2.0	60 s

Defaults: CC0 = 4.92ft, CC1 = 0.9s, CC4/CC5 = -0.35/0.35, waiting time = 60s

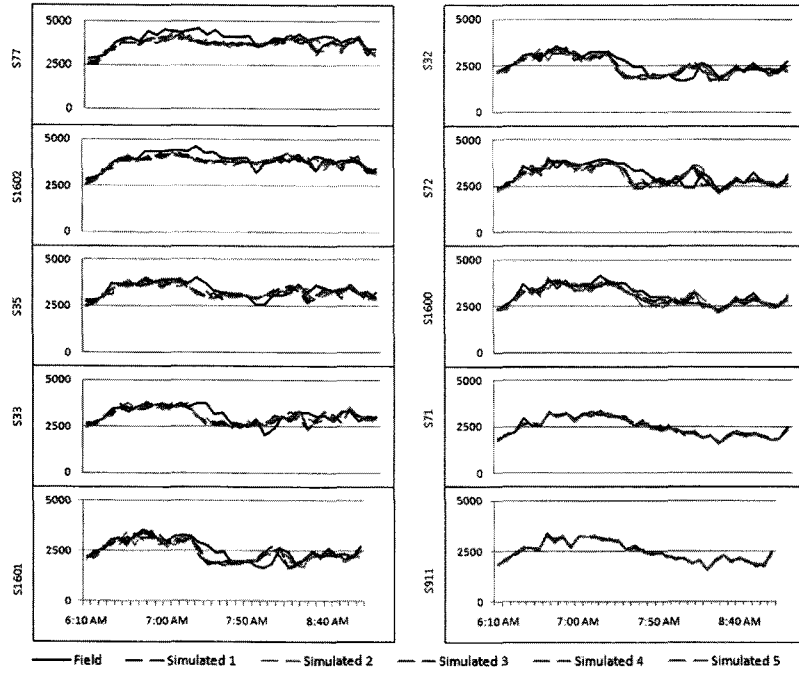
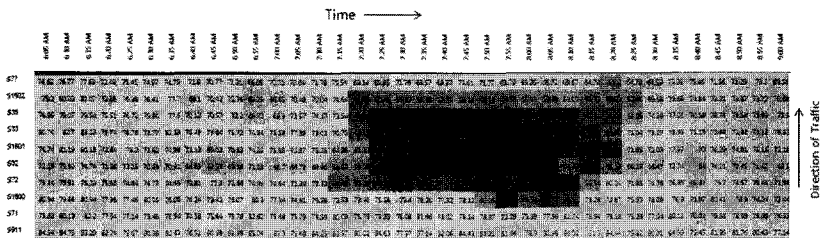
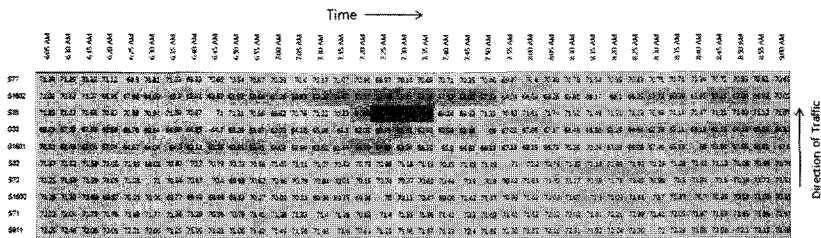


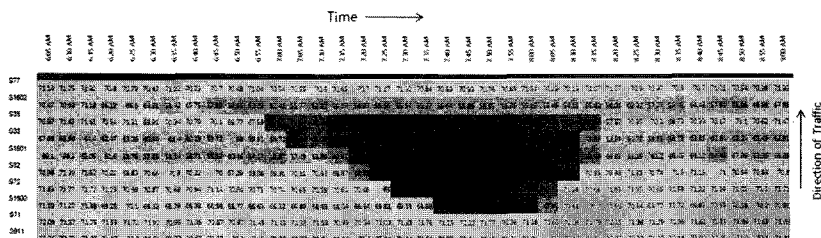
그림 3 실제 교통량과 교정된 시뮬레이션 교통량 비교 그래프(vph)



(a) 실제 속도 등고선 그래프



(b) 디폴트 파라미터를 이용한 시뮬레이션 속도 등고선 그래프



(c) 교정된 파라미터를 이용한 시뮬레이션 속도 등고선 그래프

그림 4 속도 등고선 그래프 비교

그림 4의 (c)는 교정된 파라미터를 통해 얻어진 결과이다. 결과에서 알 수 있듯이 다폴트 파라미터를 사용한 시뮬레이션 결과는 교통체증이 거의 발생하지 않는 것을 알 수 있다. 하지만 교정된 파라미터를 사용한 (c)는 실제 측정된 값과 비교하였을 때 교통 혼잡의 발생시점과 끝나는 시점이 비슷하고 그 지속 시간도 비슷한 것을 알 수 있다.

4. VSL 알고리즘

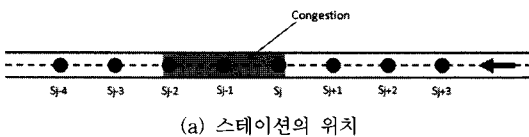
고속도로에서 사고 위험이 가장 높은 상황은 고속의 자동차가 저속의 자동차와 만나는 것과 같이 두 차량의 속도차이가 심하게 나는 상황이다. 그림 5의 (a)와 같이 스테이션 S_{j-2} 에서부터 S_j 까지 교통 혼잡이 발생하게 되면 그림 5의 (b)와 같이 S_j 부분에서 속도차이가 크게 나게 되고 해당 구간은 사고 위험이 높아진다. 따라서, 이러한 부분을 찾아내고 속도차이가 크게 나는 부분의 속도차이를 줄여주는 것이 중요하다. 하지만 속도차이를 줄이기 위해서 너무 많은 스테이션에 기존 고정 속도제한보다 낮은 속도를 설정한다면 이동시간이 늘어나고 운전자의 불만이 증가할 것이다. 또한 속도차이를 줄이기 위한 스테이션의 개수를 너무 많이 사용한다면 문제가 발생할 것이다. 본 논문의 VSL 알고리즘은 이러한 문제점을 고려하여 설계하였다.

교통 혼잡을 검출하기 위해서는 가장 먼저 교통체증을 유발하는 스테이션의 위치를 찾아내야 한다. 본 논문에서는 교통체증을 유발하는 위치의 스테이션을 BS(Bottleneck Station)라 정의하고 BS로부터 발생된 교통체증으로 인하여 생성된 큐의 가장 끝단의 스테이션을 VSS(VSL Start Station)로 정의한다. 예를 들어, 만약 그림 5의 (a)와 같이 S_{j-2} 부터 S_j 까지 교통체증이

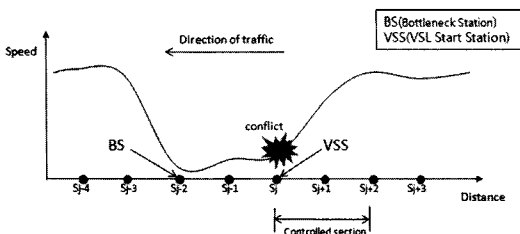
발생하고 있을 때 S_{j-2} 가 교통체증을 유발하고 있기 때문에 S_{j-2} 가 BS가 된다. 그리고 그림 5의 (a)와 같이 BS로부터 발생된 교통체증으로 인하여 생성된 큐가 S_j 까지 생성되어 있을 때 S_j 가 VSS에 해당한다.

VSL 알고리즘은 그림 6과 같이 크게 4부분으로 나뉜다. 먼저 (1) BS 존재 유무 및 위치를 검사하고 (2) BS의 위치를 기반으로 VSS 위치를 검사한다. 그리고 (3) VSS 위치를 기준으로 몇 개의 스테이션을 컨트롤해야 하는지 그 개수를 계산한 후 (4) VSL 값을 계산한다. 이러한 루틴은 30초마다 반복된다. 30초마다 VSL을 재 계산하는 이유는 고속도로의 상황 변화에 빠르게 반응하기 위함이다. 하지만 사용되는 기본 데이터인 속도, 밀도, 교통량은 과거 1분 동안의 평균 데이터를 사용한다. 실제 고속도로에서 수집되는 데이터는 항상 올바른 데이터가 들어오는 것이 아니기 때문이다. 현재 가장 많이 사용되고 있는 디텍터는 루프디텍터로 항상 100% 신뢰성있는 데이터가 수집되지 않는다. 이러한 데이터의 에러에 대해 충격을 완화하기 위해 VSL 계산의 기본이 되는 데이터는 1분 동안의 평균데이터를 사용한다.

VSS의 상향 방향에 위치한 스테이션 중 그림 5의 (b)와 같이 가까운 3개의 스테이션을 controlled section으로 정의한다. 본 VSL 알고리즘의 세 번째 단계에서 스테이션 컨트롤 개수를 계산할 때 계산된 개수에 의해 controlled section의 스테이션들 중 컨트롤되어야 하는 스테이션들이 결정된다.



(a) 스테이션의 위치



(b) 스테이션의 속도 그래프
그림 5 스테이션과 속도 그래프

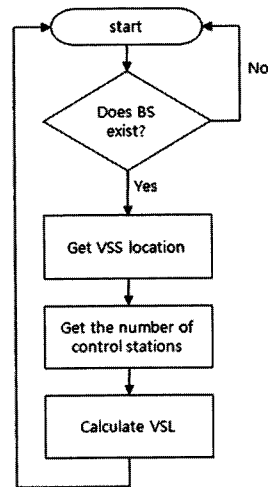


그림 6 VSL 알고리즘

① Bottleneck Station(BS) 위치 검출

BS는 교통 혼잡을 유발하는 스테이션으로 표 2는 BS를 검출하는 알고리즘을 나타낸 것이다.

표 2 BS 검출 알고리즘

If $U_j \leq U_{j+1}$ and $U_j \leq U_{j+2}$ and $U_j < (StaticSpeedLim - 5mile/h)$ for 1m 30s
then $BS = S_j$

표 2에서 U_j 는 j 번째 스테이션의 실제 측정된 속도를 의미하고 $StaticSpeedLimit$ 은 기존의 고정 속도제한을 의미한다. 해당 알고리즘은 그림 7과 같이 가장 하향 스테이션부터 검사를 시작한다. 예를 들어, 그림 7에서 가장 하향 스테이션은 S_{j-4} 이므로 S_{j-4} 부터 검사를 시작한다. S_{j-4} 는 S_{j-3} 보다 속도가 낮긴 하지만 S_{j-2} 보다는 속도가 높아 표 2의 알고리즘에서 두 번째 조건인 $U_j \leq U_{j+2}$ 를 만족하지 않기 때문에 BS가 될 수 없을 것이다. S_{j-3} 역시 같은 이유로 BS가 될 수 없을 것이다. 그림 7의 S_{j-2} 는 S_{j-1} 보다 속도가 낮고 S_j 보다도 속도가 낮다. 즉, 표 2알고리즘의 $U_j \leq U_{j+1}$ 과 $U_j \leq U_{j+2}$ 조건을 만족한다. 만약 S_{j-2} 가 1분 30초 동안 고정속도제한 보다 낮다면 마지막 조건인 $U_j < (StaticSpeedLimit - 5mile/h)$ for 1m30s를 만족하여 S_{j-2} 가 BS가 될 것이다. 마지막 조건에 1분 30초간의 제약사항을 둔 것은 순간순간 가끔씩 발생하는 에러값을 제거하기 위함이다. 디텍터의 오동작이나 가끔 예외적인 상황으로 속도가 감소되었다 해서 VSL을 시작하면 필요치 않은 상황에 VSL이 시작되어 오히려 교통에 혼란을 줄 수 있기 때문이다.

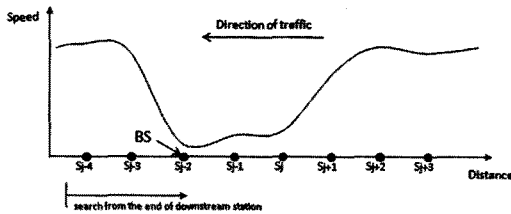


그림 7 BS 위치 검색 예시

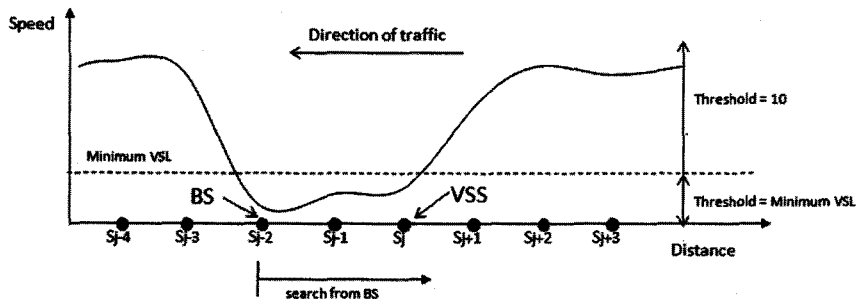


그림 8 VSS 위치 검색 예시

② VSL Start Station(VSS) 위치 검출

VSS는 교통 혼잡으로 속도가 낮은 부분과 속도가 높은 부분이 만나는 지점의 스테이션이다. 이를 검색하기 위한 알고리즘으로 표 3을 사용한다.

표 3 VSS 검출 알고리즘

If $U_{j+1} - U_j > threshold$ then $VSS = S_j$
where
If $U_{j+1} > Minimum VSL$ then $threshold = 10mile/h$
If $U_{j+1} \leq Minimum VSL$ then $threshold = Minimum VSL$

표 3에서 U_j 는 j 번째 스테이션의 실제 측정된 속도이다. 해당 알고리즘은 그림 8과 같이 검출된 BS의 위치를 기준으로 상향 방향으로 검색을 한다. 예를 들어, 그림 8에서 BS는 S_{j-2} 이므로 S_{j-2} 부터 검사를 시작한다. 만약 S_{j-2} 의 속도보다 S_{j-1} 의 속도가 $threshold$ 이상 크다면 S_{j-2} 는 VSS가 될 것이다. 단, 표 3과 같이 j 번째 스테이션을 검사할 시 $j+1$ 스테이션의 속도에 따라 사용되는 $threshold$ 의 값이 변경되는 제약사항이 있다. 제약사항에 의하면 S_{j-2} 는 VSS가 될 수 없다. 왜냐하면 그림 8에서 S_{j-2} 와 S_{j-1} 의 속도가 모두 최소 VSL보다 낮기 때문에 이 두 스테이션의 속도 차이는 최소 VSL값 보다 클 수는 없기 때문이다. S_{j-1} 역시 S_j 의 속도가 최소 VSL값 보다 작으므로 같은 이유로 VSS가 될 수 없다. 표 3의 알고리즘에서 $threshold$ 를 정하는 조건에 두 가지 제약사항을 둔 이유는 이미 교통 혼잡이 발생 되어 있는 구간 안에서 VSS가 선정되는 에러를 막기 위함이다. 예를 들어 만약 $threshold$ 값으로 단순히 10mile/h를 사용한다면 다음과 같은 문제가 발생한다. 그림 8에서 만약 S_{j-2} 의 속도가 9mile/h 이고 S_{j-1} 의 속도가 20mile/h 이라면 속도차이가 11mile/h로 S_{j-2} 가 VSS로 선언되는 문제가 발생한다. 하지만 S_{j-2}

와 S_{j-1} 은 이미 각각 평균 속도가 9mile/h와 20mile/h로 교통 혼잡이 이미 발생한 구간으로 이 두 스테이션 간의 속도차이를 줄여야할 이유가 없다. 결과적으로 표 3의 알고리즘에 의하면 그림 8에서 S_j 가 VSS로 선정될 것이다. S_j 의 속도는 최소 VSL보다 작지만 S_{j-1} 의 속도가 최소 VSL보다 크므로 threshold값으로 10mile/h를 사용한다. 만약 이 두 스테이션의 속도 차이가 10mile/h 이상 난다면 S_j 가 VSS가 될 것이다.

③ 컨트롤 스테이션의 개수 계산

VSS의 위치가 결정되었으면 VSS를 기준으로 몇 개의 스테이션을 사용할 것인지 결정해야 한다. 이를 위해 그림 9의 알고리즘을 사용한다. 너무 많은 스테이션을 이용한 VSL 컨트롤은 이동시간(Travel Time)의 증가를 가져오므로 본 알고리즘은 이를 고려하여 최대 스테이션 수를 3개로 제한한다.

그림 9에서 VSS Speed는 VSS 스테이션의 속도이고 Shock wave는 VSS를 기준으로 계산된 충격파의 크기이다. Speed difference는 VSS와 VSS의 상향 방향에 위치한 스테이션과의 속도차이를 말한다.

본 알고리즘에서 가장 먼저 VSS Speed에 따라 상황을 나누는 이유는 VSS의 속도가 고정속도제한보다 10mile/h 이상 내려가 있지 않은 상황에서는 스테이션 간 속도차이가 크게 나지 않는 상황이기 때문에 이러한 상황에서는 컨트롤을 최소화하기 위해서이다. 그림 9의 알고리즘에서 만약 VSS의 속도가 고정속도제한보다 10mile/h 이상 내려가 있지 않은 상황이라면 컨트롤되는 최대 스테이션 수가 2개이고 반대의 경우는 3개인 것을 알 수 있다. VSS의 속도가 고정속도제한보다 10mile/h 이상 내려가 있을 시에는 VSS와 VSS의 상향에 위치한 스테이션과의 속도차이를 검사한다. 이 속도차이를 검사하는 이유는 속도차이가 크게 나는 경우에는 많은 스테이션을 컨트롤 하여 속도 감소를 단계적으로 뱉을 수 있도록 하기 위해서다. 반면에 속도차이가 작을 경우

에는 최소의 스테이션을 컨트롤 하여 단계적 속도감소를 유도하는 동시에 이동시간의 증가를 막기 위함이다. 마지막으로 검사하는 것은 Shock wave이다. Shock wave의 크기는 식 (1)을 사용한다. 식 (1)에서 SW_j 는 j 번째 스테이션의 Shock wave이고 Q 는 스테이션의 교통량이며 K 는 스테이션의 밀도양이다. Shock wave를 계산하는 이유는 계산된 Shock wave가 양수이냐 음수이냐에 따라 교통 혼잡이 심화되고 있는지 풀려나가는지 알 수 있고 그 크기에 따라 교통 혼잡의 증가 및 감소 속도를 알 수 있기 때문이다.

$$Shockwave(SW_j) = (Q_d - Q_u) / (K_d - K_u) \quad (1)$$

where

$$Q_d = (Q_j + Q_{j+1}) / 2$$

$$Q_u = (Q_{j-2} + Q_{j-1}) / 2$$

$$K_d = (K_j + K_{j+1}) / 2$$

$$K_u = (K_{j-2} + K_{j-1}) / 2$$

④ VSL 값 계산

앞서 결정된 컨트롤 스테이션 개수를 기반으로 하는 VSL 컨트롤 방법은 그림 10과 같다. 컨트롤 스테이션 개수가 1개일 때는 Strategy1을 사용하고 2개일 때는 Strategy2, 3개일 때는 Strategy3을 사용한다.

Strategy1은 VSS로 선정된 스테이션만 컨트롤한다. VSS에 설정되는 VSL 값으로 하향 스테이션에서 측정된 속도값을 사용한다. 예를 들어 그림 10에서 VSS인 S_j 를 위한 VSL 값은 S_{j-1} 에서 측정된 속도가 된다. 이러한 방법을 사용하는 이유는 스테이션 컨트롤 개수가 1개일 때는 그림 9에서 보는 것과 같이 VSS의 속도차이가 고정속도제한에서 10mile/h 이상 내려가 있지 않고 Shock wave가 양의 방향인 상황이거나 VSS의 속도가 고정속도제한에서 10mile/h 이상 내려가 있다고 해도 속도차이가 크지 않고 Shock wave가 음의 방향이 아닌 양의 방향이기 때문이다. 이것은 다시 말해서 교통 혼잡이 발생하지 않아 VSL 컨트롤로 스테이션 간 속도 차

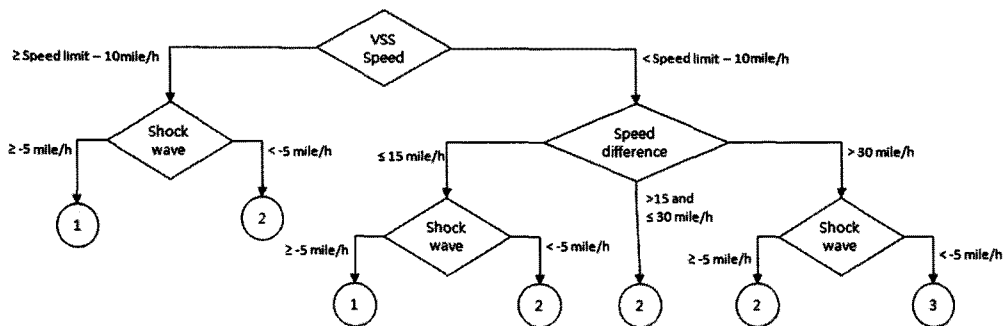


그림 9 컨트롤 스테이션의 수 결정 루틴

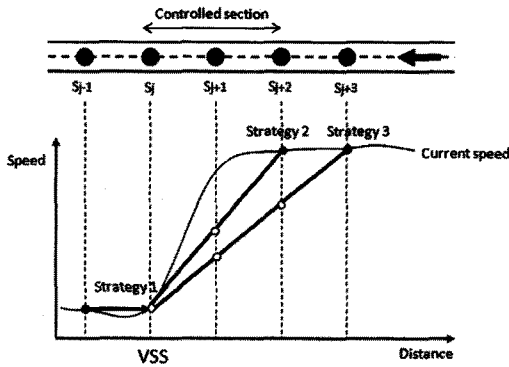


그림 10 VSL 컨트롤 방법

이를 줄일 필요가 없는 상황이거나 교통 혼잡이 풀려 나가고 있는 상황인 것이다. VSL 컨트롤로 스테이션 간 속도차이를 줄일 필요가 없을 때는 하향 스테이션의 속도를 그대로 표시해 줌으로써 스테이션 간 속도를 비슷하게 맞추어 안전화를 꾀할 수 있다. 반면 교통 혼잡이 풀려 나가고 있는 상황에서는 하향 스테이션의 속도를 그대로 반영해 VSL 컨트롤이 교통 혼잡이 풀리는 상황에 방해가 되지 않도록 할 수 있다.

Strategy2와 Strategy3은 그림 10에서 보는 것과 같이 각각 스테이션 간 속도의 중간값을 VSL로 사용한다. 예를 들어 그림 10에서 Strategy2는 VSS를 위한 VSL로 Strategy1과 같이 S_{j-1} 의 속도를 사용하고 S_{j+1} 을 위한 VSL로 S_j 와 S_{j+2} 의 중간 값을 사용한다. Strategy 3은 Strategy2와 같이 VSS를 위한 VSL로 S_{j-1} 을 사용하고 S_{j+1} 과 S_{j+2} 를 위한 VSL로 S_j 와 S_{j+3} 의 중간 값을 사용한다. 이것은 속도 차이를 단계적으로 줄여 스테이션 간 속도 차이를 최소화 하여 안전도를 향상시키기 위함이다.

VSS에서 상향 방향에 있는 스테이션은 Controlled section으로 분류하여 앞서 알아본 것과 같이 VSL을 계산하였다. 그렇다면 VSS 하향 방향에 있는 스테이션들도 컨트롤 방법을 정의해야 할 것이다. VSS 하향 방향의 스테이션들은 다음 세가지 중 하나의 상태일 것이다. 첫째, VSS가 교통 혼잡이 발생 되어 있는 상황이라면 VSS의 하향 스테이션들은 분명 교통 혼잡 상태일 것이다. 둘째, VSS가 교통 혼잡 상태가 아닐 경우에는 그 하향 스테이션들도 교통 혼잡 상태가 아닐 것이다. 셋째, VSS가 교통 혼잡 상태인 경우라도 그 하향 스테이션들은 교통 혼잡 상태가 아닌 경우일 것이다. 세 번째 경우는 BS와 VSS가 같은 스테이션일 경우 이러한 현상이 발생한다. 이 모든 경우에 최선의 방법은 그림 11과 같이 각 스테이션에 해당 스테이션의 하향 스테이

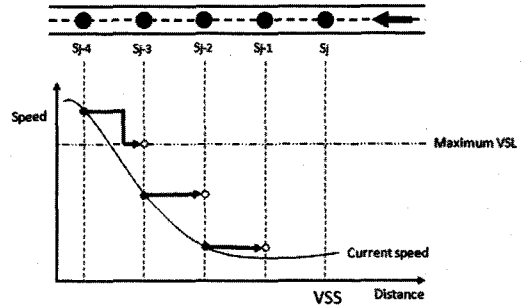


그림 11 VSS 이후 스테이션의 컨트롤 방법

션의 속도를 VSL로 사용하는 것이다. 단 VSL를 설정하는 스테이션은 VSS부터 해당 속도가 최대 VSL 보다 낮은 속도를 가지는 스테이션 까지 한다. 예를 들어, 그림 11에서 S_{j-3} 까지만 VSL을 설정하고 S_{j-4} 는 현재 속도가 최대 VSL을 넘기 때문에 컨트롤을 하지 않는다. 이러한 방법을 사용하면 앞서 말한 세가지 경우에 모두 적절한 VSL을 결정할 수 있다. 첫째와 셋째 경우에는 VSS의 하향 스테이션이 교통 혼잡 상태이거나 VSS 혼자만 교통 혼잡 상태인 경우이다. 이러한 상황에서 본 방법을 사용할 경우 교통 혼잡에 갖혀 있는 교통량을 현재 도로속도 상황에 맞춰 최대한 빨리 혼잡구간에서 뺄 수 있는 효과를 가져 올 수 있다. 두 번째 경우에는 교통 혼잡이 발생하지 않은 경우이기에 이 상황에서 각 운전자는 바로 앞 도로의 속도를 사전에 알 수 있기 때문에 안전도 향상에 기여한다.

표 4 VSL 변화 제약사항

<p>i. $U(j,t) - VSL(j,t) \leq 10\text{mile/h}$</p> <p>ii. If $VSL(j,t) > VSL(j,t-1)$ then $VSL(j,t-1) - VSL(j,t) \leq 10\text{mile/h}$ else $VSL(j,t-1) - VSL(j,t) \leq 5\text{mile/h}$</p>
--

본 논문의 VSL은 30초 단위로 그 값을 재계산 한다. 30초 단위로 VSL를 변경하는 이유는 VSL을 도로 상황에 맞게 빠르게 변경하기 위함이다. 하지만 30초 단위로 VSL이 변경될 때 그 변화 폭이 크다면 오히려 도로에 혼잡을 초래할 것이다. 따라서, 표 4와 같이 VSL이 변경될 때 두 가지 제약사항을 설정한다.

첫째, VSL은 현재 스테이션의 실제 속도에서 10mile/h를 초과하여 속도를 감소시킬 수 없다. 이러한 제약 사항을 두는 이유는 현재 속도에서 10mile/h 이상으로 속도를 갑자기 감소시키면 해당 위치에서 Shock wave가 발생하여 교통 혼잡을 유발할 수도 있기 때문이다.

둘째, VSL을 이전 VSL과 비교하여 증가시킬 경우에는 이전 VSL과 비교하여 10mile/h 까지 증가시킬 수

있고 VSL을 감소시킬 경우에는 최대 5mile/h 까지 감소시킬 수 있다. 이것은 VSL의 변화폭을 제한하여 VSL의 변화로 인하여 발생할 수 있는 운전자의 혼란 및 속도의 빈번한 변화를 방지하기 위함이다.

5. 실험결과

5.1 속도 분산

한 스테이션에서 속도 분산이 높다는 것은 속도의 변화가 크다는 의미로 그만큼 사고의 위험성이 높다는 이야기다. 새롭게 제안된 VSL 알고리즘의 안전도 향상 여부를 측정하기 위하여 그림 12와 같이 각 스테이션 별 속도 분산을 계산하였다. VSL을 적용했을 때와 적용하지 않았을 때의 정확한 상황 비교를 위하여 해당 스테이션에서 VSL이 처음으로 시작된 시점과 마지막으로 끝난 시점까지의 구간만을 속도 분산을 각각 계산하여 비교하였다. 실험결과 S71을 제외한 전체 스테이션에서 속도 분산이 감소한 것을 볼 수 있다. S71에서만 속도 분산이 VSL을 적용하였을 때 높은 이유는 VSL을 적용하지 않았을 때의 S71 속도 분포를 보면 알 수 있다. 앞에서 살펴보았던 그림 4의 (c)는 교정 후 각 스테이션 별 시뮬레이션 속도의 등고선 그래프를 나타낸 것으로 S71의 속도를 보면 시뮬레이션이 시작하여 끝날 때 까지 70mile/h 이상을 유지하는 것을 볼 수 있다. 이것은 BS에서 발생한 교통 혼잡이 S71까지 밀려 내려오지 않았다는 것을 의미한다. 이러한 상황에서 속도 분산은 0에 가깝게 나올 것이다. 하지만 VSL을 적용하였을 때는 S1600과 S71사이의 속도차이를 줄이기 위하여 VSL을 적용하여 S71의 속도를 감소시켰을 것이다. 이러한 이유 때문에 S71에서만 VSL을 적용했을 때 속도

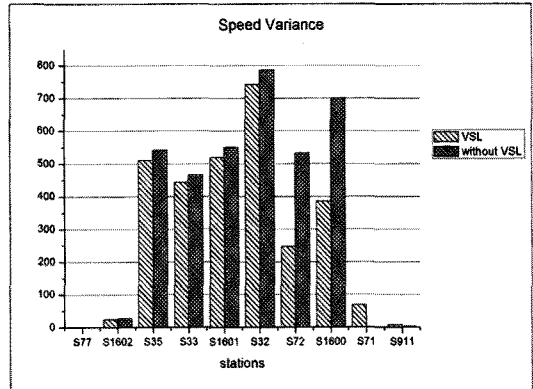


그림 12 각 스테이션 별 속도 분산

분산이 더 좋지 않은 결과가 발생하였다. 하지만 다른 모든 스테이션에서는 VSL을 적용했을 때 속도 분산이 더 낮은 결과를 얻었다. 특히 S72와 S1600에서는 큰 속도 분산의 감소를 나타내는 것을 볼 수 있다. 이것은 교통 혼잡이 발생하고 오랫동안 S72와 S1600지점이 속도차이가 심한 구간으로 유지되기 때문이다. 다시 말해, 교통 혼잡이 발생된 구간과 그렇지 않은 구간의 접경 부분으로 오랫동안 유지되었기 때문이다. 이러한 접경 구간은 교통 혼잡이 조금이라도 심해지면 속도 감소가 분명해지고 교통 혼잡이 잠깐이라도 풀리면 바로 속도가 올라가 속도 변화가 심한 부분이다. 하지만 VSL을 적용하게 되면 도로상황을 고려하여 하향 스테이션의 속도에 맞추어 상향 스테이션의 속도를 적절하게 미리 낮추어주어 급격한 속도 증가나 감소를 막을 수 있다.

5.2 이동 시간

그림 13은 S911에서 S77 구간의 이동시간을 비교한

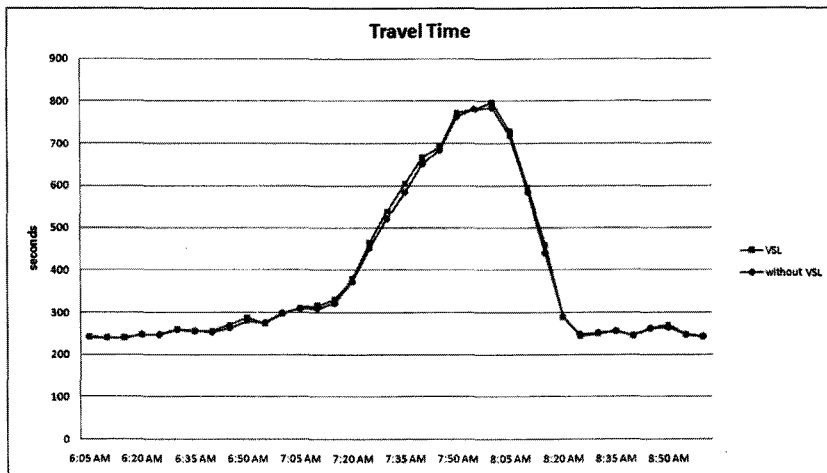


그림 13 이동 시간

그래프이다. VSL은 도로상황에 맞게 미리 속도를 컨트롤 하여 안전도를 향상 시키는데 목적이 있다. 하지만 안전도만을 생각하여 VSL을 너무 낮게 설정하거나 너무 넓은 영역을 컨트롤 하게 되면 실질적인 이동시간의 증가가 발생하게 된다. 안전도를 향상 시키되 이동시간의 증가는 최소화 시킬 수 있어야 한다. 그림 13을 보면 VSL을 적용했을 때와 적용하지 않았을 때 이동시간의 변화가 거의 없음을 확인할 수 있다. 교통 체증이 시작되어 이동시간이 증가하기 시작하는 7시 20분부터 7시 45분 사이에 약간의 이동 시간 증가를 보이지만 해당 증가는 10초 이하로 미미한 수준이다. 해당 그래프를 통해 이동시간의 증가는 거의 발생하지 않았음을 알 수 있다.

5.3 두 스테이션 간 최대 속도차이

본 논문의 VSL은 스테이션 간 속도차이를 줄임으로써 안전도 향상을 하나의 목적으로 하고 있다. 이러한 목적을 평가하기 위해 그림 14와 같이 두 스테이션 간 최대 속도차이를 계산하였다. 해당 그래프에서의 속도차이는 상향 스테이션에서 하향 스테이션의 속도차이를 말한다. 실험결과 VSL을 적용한 경우가 VSL을 적용하지 않은 경우보다 속도차이가 적은 것을 알 수 있다. 특히 교통 혼잡이 심한 7시 20분부터 8시 20분 사이에 VSL을 적용한 경우의 속도차이가 크게 작아지는 것을 알 수 있다. VSL을 적용한 경우는 속도차이가 30mile/h~40mile/h 부분을 그리는 반면 VSL을 적용하지 않은 경우는 50mile/h~60mile/h 부분을 그리고 있다. 이것으로 본 논문의 VSL 알고리즘이 효율적으로 스테이션 간 속도차이를 줄여주고 있음을 알 수 있다.

그림 15는 두 스테이션 간 최대 속도차이를 전체 시뮬레이션 시간동안 평균한 것이다. 전체 평균은 VSL을 적용한 경우 25mile/h이고 그렇지 않은 경우는 35mile/h로 10mile/h의 차이가 있음을 알 수 있다.

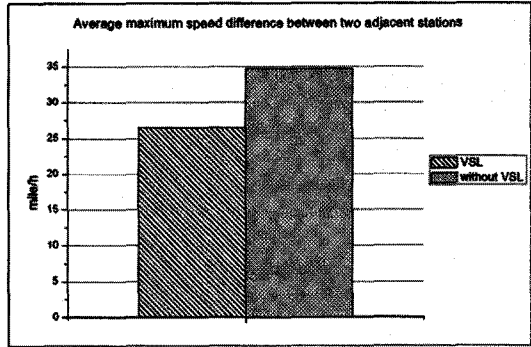


그림 15 두 스테이션 간 최대 속도차이의 평균

5.4 평균 변화량

VSL 알고리즘의 일관성을 측정하기 위해 랜덤 시드를 달리하여 총 5번의 시뮬레이션을 더 수행하였다. VISSIM은 랜덤 시드라는 요소를 두어 해당 랜덤 시드를 변경하면 시뮬레이션을 할 때 마다 조금씩 다른 교통환경을 만들어 낸다. 표 5는 앞서 실험한 3가지 경우의 평가 요소를 각각 5번의 시뮬레이션을 통하여 VSL을 적용한 경우와 적용하지 않은 경우의 변화량을 계산한 값이다. 속도 분산은 S911부터 S77까지 전체 10개 스테이션들의 속도 분산의 평균값을 5번의 시뮬레이션에 대해 평균한 값으로 19%가 감소한 것을 알 수 있다. 두 스테이션 간 최고 속도차이의 평균은 21%가 감소하

표 5 평균 변화량

내용	변화량(%)
Speed Variance	-19
Average Maximum Speed Difference between two adjacent stations	-21
Travel Time	-1

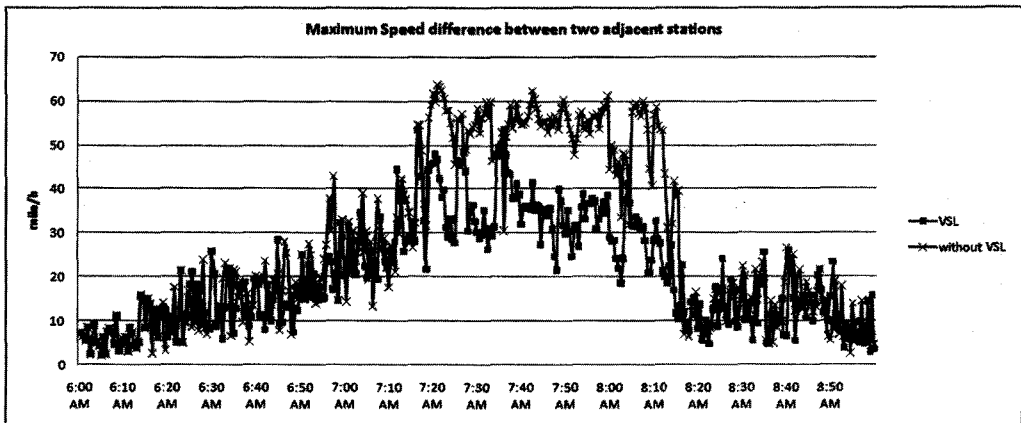


그림 14 두 스테이션 간 최대 속도차이

였고 이동 시간은 1%가 감소하였다. 해당 결과를 통해 제안하는 VSL 알고리즘이 랜덤한 다른 요소에 의해 그 결과가 변경되지 않음을 알 수 있다. 또한 본 VSL 알고리즘이 목적으로 하는 안전성 향상과 이동시간의 최소증가를 이루어낸 것을 알 수 있다.

6. 결론 및 향후계획

유비쿼터스 환경을 맞이하여 IT 기술은 사람이 인식하지 못하는 사이 사람에게 가장 가까이 침투하고 있다. 최근에는 IT 기술과 BT, NT 등의 다른 분야와 융합되면서 새로운 패러다임을 창출하고 있다. 교통 분야에서 이와 같은 융합은 ITS 기술로 이어지게 되어 기존의 도로를 좀 더 안전적이고 효율적으로 사용하려는 노력이 이루어지고 있다. 이러한 노력의 일환으로 고속도로의 속도제한을 도로상황에 맞게 적절히 조절함으로써 좀 더 나은 도로환경을 만들려는 연구가 진행되고 있다. 하지만 기존의 연구들은 독립적인 스테이션을 위한 VSL 알고리즘에 중점을 두어 도시고속도로와 같이 다수의 스테이션이 가까운 거리에 위치에 있을 때 적용하기에는 미흡한 점이 있다.

본 논문은 이러한 도시고속도로를 위한 스테이션 간 연계 기반 VSL 알고리즘을 제안하였다. 제안하는 VSL 알고리즘은 총 4단계로 구성되며 도로의 안전성을 향상시키고 이동 시간의 최소 변화를 이룰 수 있도록 설계되었다. 실험을 통하여 각 스테이션 간 속도 분산이 감소됨을 확인하였고 스테이션 간 속도 차이도 감소됨을 확인하였다. 이러한 데이터는 도로의 안전성을 향상시켰다고 볼 수 있다. 또한 이동시간의 차이가 거의 없음을 확인하였다. VSL 컨트롤 환경에서 운전자에게 있어 이동시간의 증가는 운전자가 가장 민감하게 느낄 수 있는 요소이기 때문에 이동시간의 증가는 운전자의 불만을 늘릴 수 있기 때문이다. 또한 알고리즘의 일관성을 측정하기 위해 5번의 시뮬레이션을 진행하였다. 실험결과 속도분산은 19%가량 감소하였고 스테이션 간 최대 속도 차이는 21% 가량 감소하였다. 이동시간은 변경되지 않는 것이 이상적인 것이었는데 오히려 1%가량 감소함으로써 해당 알고리즘이 교통 혼잡 완화에 조금이나마 영향을 준 것을 확인하였다.

현재 알고리즘은 VSL을 스테이션에 적용할 때 하나의 스테이션에 동일한 값을 적용하고 있다. 이러한 계산은 차선이 2차선이나 3차선과 같이 차선의 개수가 적어 교통 혼잡이 발생하면 모든 차선이 동시에 같이 교통 혼잡에 빠지는 상황에는 문제가 없을 것이다. 하지만 5차선 이상의 도로와 같이 각 차선마다 속도차이가 많이 날 수 있는 곳에서는 본 논문에서 제시하는 알고리즘은 문제가 발생할 수 있다. 그렇다고 각 차선마다 각각 따

로 VSL을 컨트롤 한다면 운전자에게 큰 혼란을 초래할 수가 있다. 또한 현재는 각 스테이션 간 거리를 고려하고 있지 않다. 이러한 이유는 도시고속도로에서는 스테이션 간 거리의 대부분 일정하게 설치되기 때문이다. 하지만 그렇지 않을 수 있는 경우도 있을 수 있다. 따라서, 차후에는 스테이션 간 거리를 고려한 VSL 알고리즘에 대해 연구할 것이며 차선 간 속도차이가 심한 구간에서 적절한 VSL을 계산할 수 있는 알고리즘에 대해 연구할 계획이다.

참고 문헌

- [1] I. T. King-Man, "Intelligent Transport Systems," Better air Quality Motor Vehicle Control & Technology Workshop, 2000.
- [2] L. Figueiredo, I. Jesus, J. A. Tenreiro Machado, J. R. Ferreira, J. L. Martins de Carvalho, "Towards the Development of Intelligent Transportation Systems," 2001 IEEE Intelligent Transportation Systems Conference, 2001.
- [3] ITS AMERICA, "www.itsa.org," Accessed May 19, 2010.
- [4] K. P. Kang, G. L. Chang, N. Zou, "Optimal Dynamic Speed-Limit Control for Highway Work Zone Operations," *Transportation Research Record: Journal of the Transportation Research Board*, no. 1877, pp.77-84, 2004.
- [5] Yadlapati, S. Sivanaga, B. Park, "Development and Testing of Variable Speed Limit Control Logics for Work Zones Using Simulation," *Center for Transportation Studies at the University of Virginia, Research Report No. UVACTS-13-0-43*, 2004.
- [6] C. Lee, B. Hellinga, F. Saccomanno, "Evaluation of variable speed limits to improve traffic safety," *Transportation Research Part C*, vol.14, no.3, pp.213-288, 2006.
- [7] P. Allaby, B. Hellinga, M. Bullock, "Variable Speed Limits: Safety and Operational Impacts of a Candidate Control Strategy for Freeway Applications," 2006 IEEE Intelligent Transportation Systems Conference, 2006.
- [8] P. Rama, "Effects of Weather-Controlled Variable Speed Limits and Warning Signs on Driver Behavior," *Transportation Research Record: Journal of the Transportation Research Board*, no.1689, pp.53-59, 2007.
- [9] I. Papamichail, K. Kampitaki, M. Papageorgiou, A. Messmer, "Integrated Ramp Metering and Variable Speed Limit Control of Motorway Traffic Flow," *Proceedings of the 17th World Congress The International Federation of Automatic Control*, 2008.
- [10] A. Hegyi, B. De Schutter, H. Hellendoorn, "Model Predictive Control for Optimal Coordination of Ramp Metering and Variable Speed Limits," *Trans-*

portation Research Part C, vol.13, pp.185-209, 2005.

- [11] PTV, "VISSIM Version 5.2 Manual," Innovative Transportation Concepts. PTV Planung Transport Verkehr AG, Karlsruhe, Germany, 2009.
- [12] FHWA, "Effects of Raising and Lowering Speed Limits," Report No. FHWA-RD-92-084
- [13] G. Gomes, A. May, R. Horowitz, "Congested Freeway Microsimulation Model Using VISSIM," *Transportation Research Record: Journal of the Transportation Research Board*, no.1876, pp.71-81, 2004.



조 영 태

2007년 강원대학교 정보통신공학과 학사
 2007년~현재 강원대학교 컴퓨터정보통신공학과 석사과정. 2009년~2010년 University of Minnesota Duluth 방문연구원. 관심분야는 멀티미디어 시스템, 센서 네트워크, 지능형 교통 시스템



정 인 범

1985년 고려대학교 전자공학과 학사. 1985년~1995년 (주) 삼성전자 컴퓨터 시스템 사업부 선임 연구원. 1992년~1994년 한국과학기술원 정보통신공학과 석사. 1995년~2000년 8월 한국과학기술원 전산학과 박사. 2001년~현재 강원대학교 컴퓨터정보통신공학 전공 교수. 관심분야는 운영체제, 소프트웨어 공학, 멀티미디어 시스템, 센서네트워크