DOI: 10.3745/KIPSTD.2010.17D.2.087

교통이력 데이터의 품질 개선과 What-If 분석을 위한 자료처리 기법의 구현

이 민 수[†] · 정 수 정[††] · 최 옥 주[†††] · 맹 보 연[†††]

요 약

현재 우리나라에서는 매일 막대한 양의 교통 데이터가 측정장치들로부터 수집되고 있으나 오류 데이터와 누락된 데이터들이 상당히 많은 실정이다. 더구나 이러한 데이터는 중요한 분석의 대상이 될 수 있음에도 불구하고 일정 시간이 지나면 삭제되고 있다. 그리하여 본 논문에서는 이러한 교통 데이터를 지속적으로 누적하여 다차원 모델로 저장하면서 데이터의 품질을 결정하는 유효성과 완전성을 높이면서 what-if 분석 기능을 지원하는 일련의 자료처리 과정을 제공하는 통합 교통이력 데이터베이스 시스템의 구현을 설명한다. 구현된 시스템에서는 다양한 오류 및 누락 데이터 패턴들을 보정하는 기법들을 제공하며, what-if 분석 기능은 다양한 데이터 정제 및 가공 과정들에 관련된 환경변수와 일련의 처리 과정들의 조합을 융통성 있게 정의하도록 함으로써 다양한 상황들을 가정하고 실험하여 결과를 분석할 수 있게 해준다. 이러한 what-if 분석 기능은 교통 데이터의 활용도를 획기적으로 높여주며 외국의 교통데이터 시스템들에서도 제공하지 못하고 있다. 교통이력데이터를 정제한 실험결과 매우 우수한 유효성 및 완전성을 가진 교통 데이터를 생성함을 확인하였다.

키워드 : 교통이력데이터, 유효성, 완전성, What-If 분석, 통합 교통이력 데이터베이스 시스템

Implementation of a Data Processing Method to Enhance the Quality and Support the What-If Analysis for Traffic History Data

Minsoo Lee[†] · Sujeong Cheong[††] · Okju Choi[†††] · Boyeon Meang[†††]

ABSTRACT

A vast amount of traffic data is produced every day from detection devices but this data includes a considerable amount of errors and missing values. Moreover, this information is periodically deleted before it could be used as important analysis information. Therefore, this paper discusses the implementation of an integrated traffic history database system that continuously stores the traffic data as a multidimensional model and increases the validity and completeness of the data via a flow of processing steps, and provides a what-if analysis function. The implemented system provides various techniques to correct errors and missing data patterns, and a what-if analysis function that enables the analysis of results under various conditions by allowing the flexible definition of various process related environment variables and combinations of the processing flows. Such what-if analysis functions dramatically increase the usability of traffic data but are not provided by other traffic data systems. Experimantal results for cleaning the traffic history data showed that it provides superior performance in terms of validity and completeness.

Keywords : Traffic History Data, Validity, Completeness, What-If Analysis, Integrated Traffic History Database System

1. 서 론

현재 우리나라의 도로 등에 설치된 루프와 영상을 포함한 차량검지기 시스템들은 방대한 양의 교통 데이터를 생성하고 있으며 일정기간동안 데이터베이스에 저장된다. 이러한 교통 데이터는 매우 많은 활용에 대한 잠재적인 중요성을 갖고 있다. 교통 데이터의 분석을 통하여 도로의 설계, 지정체 구간의 분석, 교통 전략이나 관련 정책들이 수립될 수 있다. 그러나 실제로 수집된 자료의 활용도는 매우 낮은 실정이다. 그 이유로서 크게 세 가지의 문제점들을 들 수 있다. 첫째, 가장 큰 이유로는 수집된 그대로의 교통 데이터가 분석하기에는 부적절한 품질 문제가 있기 때문이다. 차량검지기들의 고장이나 오류로 인하여 수집된 교통 데이터에는

† 종신회원 : 이화여자대학교 컴퓨터공학과 부교수(교신저자)
†† 준 회 원 : 이화여자대학교 컴퓨터공학과 석사
††† 준 회 원 : 이화여자대학교 컴퓨터공학과 석사과정
논문접수 : 2008년 2월 4일
수 정 일 : 1차 2008년 12월 31일, 2차 2009년 2월 25일, 3차 2009년 10월 20일, 4차 2010년 3월 24일
심사완료 : 2010년 3월 29일

오류 데이터나 누락된 데이터가 상당히 많이 존재하고 있다. 그래서 높은 수준의 품질을 성취할 수 있는 교통 데이터의 정제 및 가공 기법이 필요하다. 둘째로, 다양한 목적으로 교통 데이터를 활용하기 위해서는 정제 및 가공 과정들을 쉽게 변경하고 조합하여 융통성 있게 정의할 수 있어야 한다. 현재 제공되는 정제 및 가공 과정들은 매우 단순하고 고정적이어서 자유로이 변경하기도 어렵고 활용도가 상당히 제한적이다. 셋째로, 교통 데이터를 지속적으로 저장하여 이력 데이터를 만들어야 더욱 의미 있고 정확한 교통 데이터 처리 기법들을 제공할 수 있다. 이는 오랜 시간동안의 데이터를 누적함으로써 유사한 시간대나 유사한 장소의 데이터를 참조하여 오류나 누락 데이터를 보정하는 기법들이 가능하며 더욱 신뢰할만한 경향들을 발견하는 것이 가능하기 때문이다. 현재의 교통 데이터 시스템들에서는 일정 기간동안의 데이터만을 유지하다가 삭제되고 있다.

그리하여 본 논문에서는 위의 문제점들에서 설명한 바와 같이 교통 데이터의 활용도를 높일 수 있는 우수하고 융통성있는 정제 및 가공에 대한 접근 방식이 필요함에 따라 교통 데이터를 지속적으로 저장하여 이력데이터로 구성하고, 품질평가가 연동된 새로운 정제 및 가공 처리 기법을 제안하여 교통 데이터의 품질 측정 기준인 유효성과 완전성을 높이며, 다양한 환경변수와 처리방법들의 조합을 정제 및 가공 처리 과정에 융통성있게 반영하여 결과를 분석할 수 있는 what-if 분석이 가능한 통합 교통이력 데이터베이스 시스템의 구현에 대하여 설명한다.

본 논문의 기여점은 다음과 같이 요약할 수 있다. 첫째, 대용량 교통 데이터 처리에 있어서 IT기술을 적용하여 새로운 IT-교통 융합 응용을 구현하는 시도라는 것이다. 둘째, 교통 데이터의 품질을 높여서 실용성을 제공하여 그동안 충분히 활용되지 못하던 중요한 정보를 활용하게 하는 것이다. 셋째, 다양한 목적으로 교통 데이터를 활용하고자 하는 다양한 분야의 사람들이 자유로이 교통 데이터를 가지고 작업하며 실험하고 분석할 수 있는 편리한 what-if 기반의 교통 데이터 처리 환경을 제공한다. 넷째, 대용량 교통 이력 정보를 누적함으로써 이력 기반의 신뢰성이 높은 데이터 처리 알고리즘들을 제안하였고 차후 알고리즘 실험 및 개발을 위한 연구기반을 다진다. 다섯째, 우리나라 교통 정책 및 전략 수립, 이력 정보에 근거한 명절이나 공휴일 교통 분석, 지정체 구간 및 우회 도로 종합 분석, 도로 설계 및 확충 방안 등의 기반 데이터 제공을 위한 플랫폼으로 사용 가능하다.

본 논문의 구성은 다음과 같다. 먼저 2절에서는 관련연구로서 교통정보 시스템의 기본적인 개념들을 설명하고 국내외 차량 검지기 자료 처리 현황에 관련된 연구들에 대해 알아본다. 3절에서는 교통이력 데이터 자료처리 예시를 통하여 문제를 분석하고4절에서는 통합 교통 이력 데이터베이스 시스템의 개요와 제공되는 교통 데이터의 자료처리 알고리즘들과 what-if 분석기능에 대해 다룬다. 5절은 시스템의 구현에 대하여 자료처리 과정의 알고리즘들과 what-if 분석을 지원하는 웹 기반의 사용자 인터페이스를 설명한다. 6절에서는 교통 데이터를 활용한 자료처리의 검증과 실험을 통하여 새로운 자료처리 기법의 우수성을 확인하며 7절에서 결론을 제공한다.

2. 관련 연구

2.1 교통정보 시스템의 기본적인 개념과 용어

IT 와 교통 분야의 융합 응용을 구현하기 위해서는 교통 분야에 대한 이해가 필요하므로 우선 기본적인 몇 가지 교통 분야의 개념과 용어에 대한 설명을 하고자 한다.

교통 정보를 제공하는 수집장치를 차량 검지기(vds: vehicle detection system)라고 하며 대표적인 것들은 도로 밑에 매설된 루프(loop)라는 케이블과 유사한 장치도 있고 지상에 설치된 영상 장치도 있는데 주로 루프 데이터가 비교적 정확하여 이를 많이 활용한다. 도로 상의 하나의 지점에는 보통 한 개 또는 두 개의 루프가 설치되고 차로별로 번호가 매겨져서 데이터 수집시 식별자로 사용된다. 차량 검지기에서 나오는 데이터의 종류로는 교통량, 속도, 점유율 자료가 있다. 교통량(volume)이란 시간당 지나가는 차량의 대수를 의미하며 속도(speed)는 차량들의 시간당 가는 거리이며 점유율(occupancy)은 단위시간당 차량이 차로를 점유하는 시간을 의미한다.

교통 데이터는 수집된 그대로의 자료를 원시(source) 자료라고 하며 누락되거나 오류 데이터를 포함하고 있어 추가적인 자료처리가 필요하다. 몇 가지 방법들로는 오류판단, 결측보정, 평활화가 있으며 오류판단의 과정은 정상 자료를 선별하여 주며 결측보정의 과정은 오류 자료를 정상치로 보정하고 평활화 과정에서는 전체 데이터의 극소값과 극대값을 제거하여 정상치화 하여 줌으로써 이후 자료의 다양한 활용이 가능토록 해준다. 이들 처리 과정과 관련된 기본적인 알고리즘들에 대한 이론적인 연구들은 진행되고 있으나 실제 구현에 적용하여 시스템을 구축하는데에는 극히 일부만 제한적으로 적용하여 사용하고 있고 이들의 다양한 조합들에 대해서는 연구나 구현이 제대로 진행되지 못하고 있다.

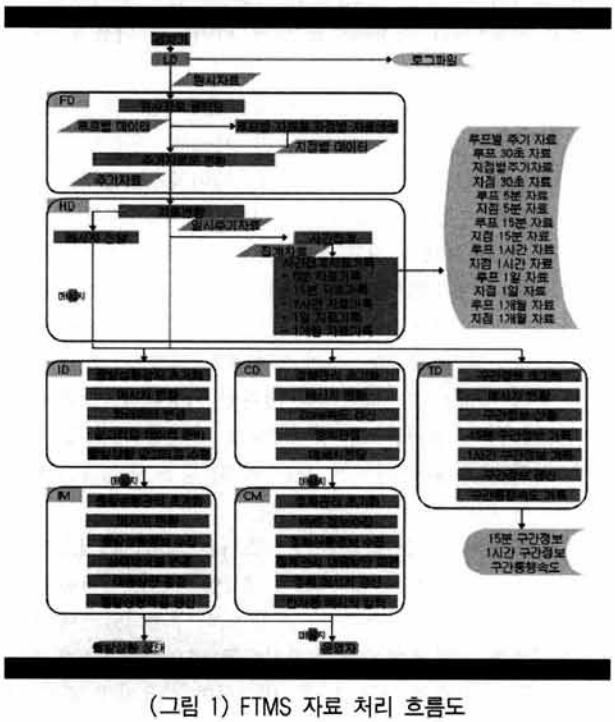
품질평가는 교통 데이터가 얼마나 누락되었는지, 오류가 없는지 등을 정량화하여 판단하도록 하는 것으로서 매우 중요한 정보인데 이러한 정보를 체계적으로 제공하는 시스템은 없으며 본 논문에서는 두 가지 기준인 완전성(completeness)과 유효성(validity)을 사용한다.

2.2 국내외 교통정보 시스템의 자료처리 관련 연구 및 장단점 분석

국내에서 교통정보 데이터와 관련하여 품질 향상과 분석을 지원하기 위한 시스템으로서 한국도로공사에서 운영하는 고속도로 교통 관리 시스템인 FTMS(Freeway Traffic Management System)가 있다[1]. FTMS에서는 검지기로부터 수집된 교통량, 속도, 점유율 자료를 저장하고 관리한다. 검지기로부터 수집된 교통량, 속도, 점유율의 원시 자료는

오류자료를 포함할 수 있다. 오류자료(invalid data)는 교통량, 속도, 점유율 등의 속성이 논리적으로 정상적인 상태의 교통류 특성을 나타내는 범위를 벗어난 교통자료를 의미한다. 이러한 오류자료는 자료처리 과정을 거쳐서 정제된다. 이렇게 정제된 검지기의 교통량, 속도, 점유율은 루프별 데이터와 지점별 데이터로 집계된다. 이렇게 1차 가공된 루프별 데이터와 지점별 데이터는 주기자료로 다시 변환되어 시간 집계 과정을 거친다. 시간 집계는 연구자의 필요 시간 단위인 5분, 15분, 1시간, 1일, 1개월이 되며 루프별, 지점별로 집계되므로 루프 30초 자료, 지점 30초 자료, 루프 5분 자료, 지점 5분 자료, 그리고 점차로 단위 시간간격을 넓히며 루프 1개월 자료, 지점 1개월 자료까지 생성하게 된다. 루프별, 지점별 자료는 다양한 교통 정보 제공을 위한 처리 시스템에 사용되는데 돌발상황 감지, 정체관리, 구간정보 산출 등을 위한 자료로 쓰인다. (그림 1)에서는 이러한 FTMS의 자료처리 과정을 보여주고 있다.

FTMS의 장단점과 본 연구의 장단점을 분석해보면 다음과 같다. FTMS의 장점은 빠른 데이터 저장과 적재가 가능하며 실시간 데이터 제공을 위에 매우 적합한 구조로 데이터베이스가 구성되어 있는 반면 단점으로는FTMS에서는 자료처리를 위한 고정적인 알고리즘이 적용되어 수정하기가 매우 어렵다는 것이며 간단한 오류판단과 결측보정 알고리즘 정도만 적용하고 있어 품질이 낮다는 것이다. 본 논문의 구현된 시스템은 장점으로서 체계적인 품질평가를 자료처리에서 지원하며 다양한 종류의 자료 처리 알고리즘들을 제공하고 있고 이를 다양하게 설정하여 적용할 수 있는 차별점이 있어 데이터의 품질을 높일 수 있다. 그러나 FTMS에 비해 추가적인 자료처리를 수행해야 하므로 실시간 데이터 제공에는 제한이 있다.



(그림 1) FTMS 자료 처리 흐름도

국외에서 수행하는 대표적인 연구로서는PeMS(Performance Measurement System)가 있다. PeMS는 캘리포니아의 교통부에서 운영하는 교통 관리 시스템으로 캘리포니아 전역에서 수집된 루프 검지기의 자료를 수집, 가공, 저장하여 관리하는 시스템이다[2]. UC버클리 대학과 CalTrans가 공동으로 연구개발하였으며, 교통 관리 시스템의 운영 효과를 어떻게 측정할 것인가에 관한 쟁점으로부터 시작하여, 1997년부터 기초연구가 시작되었다. 2003년에 PeMS 4.0 버전을 발표하면서 지금과 같은 형태의 모습을 갖추게 되었으며, 현재 8.0 버전이 발표된 상태이다. 웹 서비스로 제공되는 이 서비스는 검지기의 상태 및 자료의 질에 대한 정보를 제공하고, 이용자가 원하는 자료를 인터넷을 통해 제공한다. 다양한 형태로 시각화되어 이용자의 편의성을 높였고 자료 해석 수준을 향상시켰다. 캘리포니아 주 고속도로는 단일루프 검지기와 이중루프 검지기가 혼재해 있으며 단일루프검지기를 통해 수집된 자료로 속도를 추정하기 위해 g-factor라는 수치를 개발하여 자료처리 과정에 적용하고 있다. 5분 단위 자료로 집계하여 속도를 계산하고, 이상치(outlier)를 필터링하고 결측 데이터(missing data)를 과거자료를 이용하여 생성하여 채워넣는다. 이러한 절차로 처리된 자료는 1시간 단위, 1일 단위로 다시 집계된다.

본 연구에서 개발한 시스템과 비교하여 PeMS의 장단점을 분석해보면 다음과 같다. PeMS의 장점으로는 웹 기반으로서 사용자 인터페이스가 매우 쉽고 분석 결과를 보여주는 그래픽 기능이 탁월하다는 것이다. 단점으로는 자료처리 알고리즘이 데이터베이스내에 통합되어 있지 않아서 대용량 데이터의 처리에 대해서는 매우 많은 처리 시간이 걸리며 처리 과정에 대한 피드백이 제대로 주어지지 않는다. 또한 다양한 처리 과정들의 조합이 제공되지 않는다. 이에 반하여 본 논문의 구현된 시스템의 장점은 다양한 종류의 자료처리 알고리즘들을 다양하게 설정하고 조합하며 품질평가 과정이 자료처리 과정에 통합될 수 있고 데이터베이스 내부에 알고리즘들을 구현하여 처리의 효율성을 높였으며 웹 기반의 인터페이스에서 처리 상황에 대한 피드백을 주어 사용자 편의성을 제공한다는 차별점이 있다. 본 연구에서 개발한 시스템에서 그래픽 사용자 인터페이스는 기본적인 그래프 형식들을 위주로 제공한다는 단점이 있다.

수많은 교통 데이터 수집 장치로부터 얻는 데이터에 대하여 각 시스템으로 보내어지기 전에 품질 문제를 해결하기 위하여 확률적인 데이터 기반의 방식을 제안한 Ishak[3]의 연구가 최근에 있으나 이는 주로 실시간 용이어서 본 논문의 접근 방식과 비교하여 이력 데이터를 충분히 활용하지 못하는 방식이라는 단점을 갖고 있다. Smith[4]의 연구에서는 데이터의 품질과 관련하여 두 단계의 과정을 거쳐서 품질평가를 진행하는데 먼저 데이터가 이용가능한가를 판단하고 이어서 데이터를 수집하는 장치의 상태를 확인하는 두 단계의 방식을 제안하며 각 데이터 저장 시스템의 필요에 따라 이 단계들을 맞춤형으로 변경할 수 있도록 하고 있다. 본 시스템과는 품질평가를 진행하고 맞춤형 개념의 융통성을 제공한다는 측면에서 유사성이 있으나 본 논문의 구현한

시스템에서는 품질평가를 일련의 데이터 처리 과정에도 자유롭게 통합할 수 있다는 강점이 있다.

2.3 시공간 데이터베이스 관련 연구

교통이력 데이터의 저장과 자료처리에 대한 연구는 기본적으로 시간과 공간에 대한 정보를 다루는 것이어서 데이터베이스 연구 분야에서는 시공간 데이터베이스의 연구와 매우 관련성이 많다. 데이터베이스 내에 시간과 공간 정보를 함께 저장하고 시공간 특성을 반영하는 저장 및 질의 처리 기법들을 다루는 시공간 데이터베이스 연구들이 그동안 많이 진행되었는데 특히 최근의 이동통신의 발달에 따른 이동객체들의 시간에 따른 위치 정보의 저장이 가능해지면서 이러한 이동객체의 이동 패턴에 대한 연구들이 많이 진행되어 왔다[5]. 고속도로의 시공간 데이터와 관련해서는 통행료 수납과 연계하여 차량의 시공간 정보를 저장하고 이동 패턴을 분석하는 연구도 있다[6]. 그러나 통행료 수납 관련 데이터는 대체로 잘 정제된 경우가 많아 추가적인 자료처리의 필요성이 적은 데이터로서 본 연구에서 다루고자 하는 데이터 특성과는 차이가 있다. 시공간 데이터베이스 분야에서는 시계열 데이터가 많이 다루어지는데 이러한 시계열 데이터에서 다양한 데이터 변환기법을 기반으로 하여 패턴을 찾아내는 방법들이 연구되고 있다[7]. 또한 GPS 등의 위치 기반 정보가 일반화되면서 이에 대한 정보처리에 중점을 두는 연구들도 진행되고 있다[8].

3. 교통이력 데이터의 자료처리 예시를 통한 문제 분석

교통이력 데이터에 대한 자료처리 예시를 구성하여 살펴보면 본 연구에서 해결하고자 하는 문제를 좀더 정확하게 분석할 수 있다. 하나의 예시로서 우리나라의 도로 등에 매설된 교통 데이터 수집기로서 속도 등을 수집하는 루프라는 장치가 있다. 이를 통하여 매 30초마다 데이터를 받는다고 하자. 그리고 이 데이터를 누적해서 시간에 따른 특정 도로 구간의 속도의 변화를 분석해보고 싶다고 하자. 그렇다면 첫번째 문제는 이러한 데이터를 어떻게 데이터베이스에 저장할 것인가 하는 문제가 생긴다. 단순하게 이 값들을 데이터가 발생한 시간과 장소 및 속도값만을 테이블에 넣어두게 되면 나중에 대량의 데이터 분석을 하고자 할 때 다양한 평균값이나 최대값 및 최소값 연산 등을 할 때 매우 시간이 많이 걸릴 수 있다. 그리하여 해당 데이터를 분석하기에 좋게, 분석 성능이 빠르도록 데이터베이스의 저장구조와 추가적인 저장 내용들을 구성해야 한다. 두 번째 문제는 교통 데이터 수집 장치들은 물리적인 오류들이 많아서 수집되는 데이터 내에도 많은 오류들이 있다. 검지기가 고장나서 모두 0으로 데이터가 오기도 하고 실제로는 불가능한 속도값인 300 km/h를 생성하기도 한다. 그리하여 이러한 오류가 포함된 데이터로 분석을 하는 것은 아무런 의미가 없으므로 이들 수집된 데이터에 대하여 적절한 처리를 하여 오류를 제거하거나 데이터의 특성을 복원시키는 일을 해주어야 한다. 그리하여 품질을 높이기 위한 자료처리 알고리즘들을 제공해야 한다. 세번째 문제는 이러한 교통 데이터 분석은 다양한 사용자의 관점에서 진행된다는 것이다. 즉, 오류 데이터를 일부 보정하기 원하는 경우가 있을 수 있고, 보정없이 오류는 무시하기를 원할 수도 있고, 또는 데이터가 지나치게 큰 폭으로 변동성을 갖지 않기를 원할 수 있다. 이러한 다양한 교통 데이터 분석의 요구들을 지원하기 위해서는 다양한 데이터 처리 기법들을 사용자가 자유로이 조합하여 원하는 데이터 분석이 쉽게 이루어질 수 있도록 해주어야 한다. 물론 그래픽 사용자 인터페이스에서도 매우 편하게 이러한 다양한 분석을 지원해야 한다.

그러므로 이후의 내용에서는 먼저 교통이력 데이터베이스 시스템의 데이터 저장 구조에 대하여 논한 뒤 데이터 품질을 높이기 위한 자료처리 알고리즘들을 살펴보고 마지막으로 사용자의 다양한 분석 요구들을 편리한 그래픽 사용자 인터페이스를 통하여 지원이 될 수 있도록 하기 위한 시스템 설계를 다룬다.

4. 통합 교통이력 데이터베이스 시스템 구조 및 자료처리 설계

4.1 교통이력 데이터의 다차원 저장 모델 및 시스템 구조

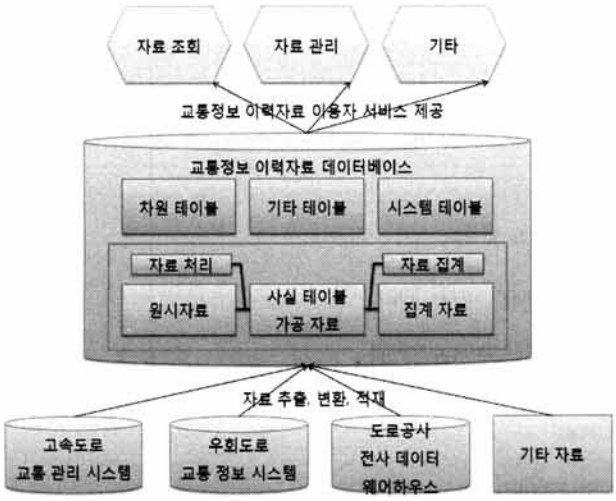
통합 교통이력 데이터베이스 시스템은 다양한 교통 정보를 제공하는 시스템에서 데이터를 가져오며 이를 시간 및 공간 속성을 기반으로 한 다차원 모델로 누적하여 이력정보를 처리 및 분석이 용이하도록 저장한다. 다른 시스템에서는 이력 정보의 저장 및 분석을 위한 다차원 모델을 활용하고 있지 못하고 있으나 본 논문의 구현된 시스템은 이를 이용하여 이력 데이터의 효율적인 저장 및 분석이 가능하도록 하였고 자료 처리 과정에서도 이력 데이터가 이용될 수 있는 기반을 제공한다.

통합 교통이력 데이터베이스 시스템에서는 먼저 기존의 운영시스템인 고속도로 교통 관리 시스템, 우회도로 교통정보 시스템, 도로공사 전사 데이터 웨어하우스로부터 교통 관련 자료를 추출 변환하여 데이터베이스에 적재한다. 적재된 교통 관련 자료는 자료 처리 과정과 자료 집계과정을 거쳐 이용자 서비스를 통해 제공된다. 통합 교통이력 데이터베이스는 교통 정보를 사실 테이블, 차원 테이블, 기타 테이블로 구성하여 저장한다. 사실 테이블은 교통량, 속도, 점유율의 원시 자료와 그 자료처리 결과인 가공자료, 그리고 가공자료에 대한 집계를 한 집계자료로 구성된다. 차원 테이블은 시간 속성을 나타내는 시간 차원, 공간 속성을 나타내는 검지기 관련 차원들로 구성된다.

통합 교통이력 데이터베이스 시스템은 데이터베이스에 적재된 데이터와 함께 이를 처리할 수 있는 프로시저들을 데이터베이스 내부에서 실행할 수 있도록 구현되었으며 이용자 서비스를 미들웨어에서 제공하도록 하였다. 이용자 서비스로 교통 데이터의 조회, 교통 데이터의 사용자별 관리, 관

리자 권한에서의 자료 처리 알고리즘 등록 및 프로세스 관리 등과 같은 기타 기능들을 제공한다. 추가적으로 웹 기반의 사용자 인터페이스를 제공하여 사용자가 편리하게 시각적으로 교통 데이터들을 분석할 수 있고 관리하도록 한다.

(그림 2)에서 통합 교통이력 데이터베이스 시스템의 저장 및 시스템 구조를 보여준다.



(그림 2) 통합 교통이력 데이터베이스 시스템의 구조

4.2 교통이력 데이터의 고품질 자료처리 알고리즘 체계 설계

통합 교통이력 데이터베이스 시스템에서는 교통 데이터를 처리함에 있어서 기본적인 일련의 체계적인 과정을 제공하며 이 과정에서 이력 데이터의 특성을 이용하여 더욱 품질을 향상한 기법들을 제공하고 있다. 이 기본적인 과정은 교통 데이터의 처리에 관한 모든 필요한 단계들을 순차적으로 거칠 수 있도록 설계되었고 처리 전후에 품질의 비교를 할 수 있도록 해준다. 물론 이를 변경하여 다른 처리 과정을 새로이 만들 수도 있다. 다른 시스템들에서는 전체 처리 과정중 일부만을 제공하고 처리 전후의 품질 향상에 대한 정량화 기준을 제공하지 않아 처리 결과가 만족스러운지를 확인해볼 수 없다.

통합 교통이력 데이터베이스 시스템에서 기본적으로 제공되는 자료 처리 과정은 원시자료 선처리(preprocessing), 품질평가(quality evaluation), 오류판단(filtering), 결측보정(imputation), 평활화(smoothing), 품질평가의 일련의 체계적인 과정으로 이루어진다. 이러한 각각의 단계에서 처리하는 알고리즘들이 구현되어 제공되며 알고리즘들은 앞의 순서대로 실행된다. 교통정보수집장치인 검지기 시스템에서 수집된 교통 데이터를 원시자료 선처리를 하여 이후의 자료처리에 적합한 테이블과 데이터 값의 형태로 변환하고, 일련의 자료처리 작업을 수행하기 전에 원시자료에 대한 품질평가를 수행하여 자료처리 전 데이터의 품질을 평가한다. 원시자료 선처리를 통하여 적합한 형태로 변환된 자료는 오류판단의 과정을 거쳐 오류 데이터를 필터링하게 되고 결측보정의 다양한 방법으로 오류 데이터와 누락 데이터를 보정하게 된다. 결측보정 과정의 수행으로 보정된 데이터를 다음 단계인 평활화를 통해 이상치를 제거하고 이러한 일련의 자료처리 과정을 거친 데이터는 최종적으로 품질평가를 통하여 데이터의 질적 평가를 하게 된다. 각 알고리즘에 대한 설명은 다음과 같으며 (그림 3)에서 기본 자료처리 체계를 보여준다. (그림 3)에서는 각 단계의 알고리즘별 수행내용과 함께 세부적인 다양한 기법들도 있음을 보여주며 품질평가 단계는 원하면 모든 단계의 전후에도 수행 가능함을 보여준다. 세부적인 기법들은 5절 구현에서 구체적으로 논한다.

· 원시 자료 선 처리

원시 자료 선 처리 과정이란 자료처리 과정 전 단계에서 순수 원시 데이터를 변경하는 작업이다. 이 과정은 이후 이루어지는 품질평가, 오류판단, 결측보정, 평활화의 자료처리 과정이 효율적으로 수행될 수 있도록 하며 데이터의 실제 환경을 고려하여 현실성 있는 데이터 처리 작업을 의도한다. 차량 검지기로부터 수집된 원시 데이터 테이블을 입력 테이블로 하여 데이터 입력 시간 단위를 30초 단위로 보정 및 변경한다. 또한 우리나라 교통 상황을 반영하여 24개의 루프로 변환하며 루프가 설치되지 않은 차로에서 수집된 교통량, 속도, 점유율 값을 NULL로 처리한다. 또한 결측 자료는 -111로 입력하여 이후의 자료처리가 정확하고 효율적으로 수행되도록 한다.

· 품질평가

품질평가란 데이터의 완전성과 유효성을 검토하는 자료처리 과정으로서 이용 가능한 자료의 비율 및 유효한 자료의 비율을 결과값으로 제공한다. 품질평가는 순수 원시 자료에 대한 자료처리 전 품질평가와, 모든 자료처리 과정을 거친 자료처리 후 품질평가로 나뉘어서 수행된다. 즉, 자료처리를 함으로써 데이터의 품질이 향상된 정도를 자료처리 전후의 값을 비교하여 확인할 수 있다.

완전성이란, 루프/지점별 수집된 전체 원시 자료(교통량(vol), 점유율(occ), 속도(speed)) 중에서 결측되지(누락되지) 않은 자료의 비율을 말하며, 이때의 자료를 이용 가능한 자료라고 한다.

$$\text{완전성}(\%) = \frac{x}{X} \times 100$$

위의 식에서 x는 루프/지점별 관측된 원시 자료 중 결측되지 않은 이용 가능한 자료이며, X는 루프/지점별 전체 원시 자료이다.

유효성이란, 루프/지점별 수집된 이용 가능한 자료 (교통량(vol), 점유율(occ), 속도(speed)) 중에서 오류판단 기준에 의거하여 오류가 없는 자료의 비율을 말하며, 이때의 자료를 유효한 자료라고 한다. 따라서 정량화 방법은 다음과 같다.

$$\text{유효성}(\%) = \frac{x}{X} \times 100$$

```
PROCEDURE High_Quality_Traffic_Data_Processing  /* 고품질 자료처리 알고리즘 체계 */
INPUT: Table raw_table;              /* raw data table with volume, occupancy, speed columns */
       Time_Interval  time_spec;                  /* time interval for data to process */
       Location_Interval loc_spec;               /* location interval for data to process */
       Int imputation_mode;                      /* mode used for imputation */
Boolean user_input_smoothing_constant;    /* True if user sets smoothing value */
       Float input_value, t;                     /* smoothing value given by user, period value*/
OUTPUT: Table processed_table;                   /* processed data table */
Float before_completeness[], before_validity[], after_completeness[], after_validity[];
BEGIN
Table tmp1_table, tmp2_table, tmp3_table;
/* 원시자료 선처리 */
Preprocessing ( IN: raw_table, time_spec, loc_spec; OUT: tmp1_table );
/* 자료처리 전 품질평가 */
Quality_Evaluation ( IN: tmp1_table, time_spec, loc_spec;
OUT: before_completeness[], before_validity[] );
  /* 오류판단 */
Filtering ( IN: tmp1_table, time_spec, loc_spec; OUT: tmp2_table );
/* 결측보정 */
Imputation ( IN: tmp2_table, time_spec, loc_spec, imputation_mode; OUT: tmp3_table );
/* 평활화 */
Smoothing ( IN: tmp3_table, time_spec, loc_spec, user_input_smoothing_constant,
input_value, t; OUT: processed_table );
/* 자료처리 후 품질평가 */
Quality_Evaluation( IN: processed_table, time_spec, loc_spec;
OUT: after_completeness[], after_validity[] );
RETURN ( processed_table, before_completeness[], before_validity[],
after_completeness[], after_validity[] );
END
```

(그림 3) 교통이력 데이터의 고품질 자료처리 알고리즘 체계

위의 식에서 x는 루프/지점별 관측된 이용 가능한 원시 자료 중에서 오류가 없는 유효한 원시 자료이며, X는 루프/지점별 관측된 이용 가능한 원시 자료이다.

· 오류판단

오류판단이란 한국도로공사의 오류판단 기준인 임계값 검사(Data Threshold)와 관계 검사(Data Relation)에 의거하여 오류 데이터를 판별하는 과정이다. 오류판단은 이후 오류 데이터를 보정하는 결측보정 과정을 위해 필요하며 오류 데이터와 정상 데이터를 구별할 수 있는 정보를 제공한다. 오류판단을 거친 오류 데이터는 특정 수치로 변경된다.

한국도로공사에서 제공하는 교통 데이터 오류판단 기준인 임계값 검사와 관계 검사는 좀더 상세히 설명하면 다음과 같다. 임계값 검사는 어떠한 하나의 교통 데이터에 대하여 수치의 범위를 정하여 그 범위에 속하는 데이터를 오류로 분류하는 것이고, 관계 검사는 둘 이상의 교통 데이터에 대하여 관계 조건에 해당하는 데이터는 오류로 분류하는 것이다. 임계값 검사의 예로는 교통량이 0 이하의 값이거나 30 이상의 값에 해당하는 경우, 인접 검지기에서 수집된 교통량 간의 차가 2보다 클 경우, 점유율이 0보다 작거나 100보다 클 경우의 값은 현실적으로 존재하기에 불가능한 값이므로 오류값으로 분류한다. 관계 검사로는 동일 검지기에서 수집된 교통량이 0이고 속도가 0이 아닌 경우와 교통량, 점유율, 속도 간에 다양한 관계들이 현실적으로 존재하기에 불가능한 경우를 관계식으로 설정하여 이들을 오류 값으로 분류한다.

· 결측보정

결측보정이란 오류 데이터를 여러 가지 기법에 의해 보정하는 과정이다. 다음 4절에서 소개되는 인접 지점 참조 기법과 이력 자료 활용 기법으로 오류 데이터를 정상 데이터로 변경한다. 이 단계에서의 결측은 품질평가에서 사용한 누락된 데이터라는 의미로의 결측과는 다르다. 결측보정에서의 결측은 누락된 데이터뿐 아니라 오류인 데이터도 포함한다. 결측 데이터를 보정함으로써 데이터의 품질을 향상시킬 수 있게 된다. 이력 데이터의 특성을 활용하여 4가지의 결측보정 기법들을 제공한다. 이력데이터에서 유사한 공간적 특성 및 시간적 특성을 활용하여 전후지점 동일주기 적용 기법, 이전지점 동일주기 적용 기법, 지점간 이동소요주기 적용 기법, 이력자료 동일주기 적용 기법을 구현하였다.

· 평활화

평활화란 이상치를 제거하는 자료처리의 과정이다. 본 논문에서는 원시 자료선처리, 오류판단, 결측보정의 모든 자료처리 과정을 거친 데이터에 대하여 평활화함으로써 극소값과 극대값을 제거한다. 평활화 과정은 데이터를 정상치로 조절하여 연구자의 데이터 활용이 좀더 의미있게 이루어질 수 있도록 하는 중요한 단계이다. 데이터를 평활화하기 위해 평활화 계수를 사용함으로써 이상치 수준을 변경할 수 있다.

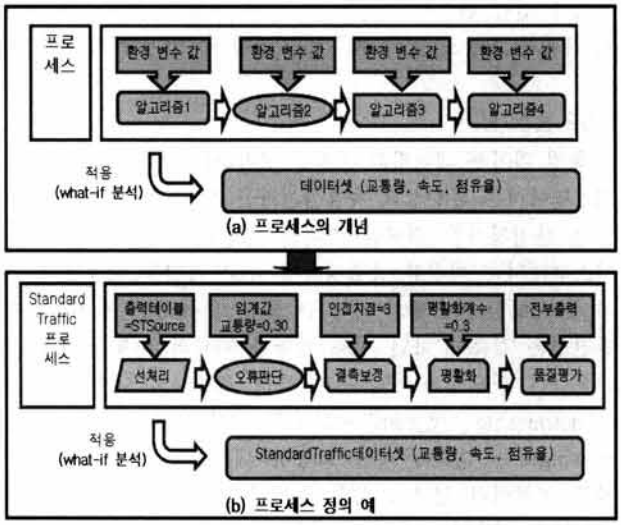
4.3 교통이력 데이터의 what-if 분석 기능을 위한 자료처리 설계

통합 교통이력 데이터베이스 시스템에서는 교통이력 데이

터를 처리함에 있어서 다양한 활용 분야를 위해서 다양한 자료처리 과정을 융통성 있게 생성할 수 있어야 한다. 예를 들면 특정 응용에서는 결측보정 단계를 다른 방식으로 수행하여 더욱 많은 이력 데이터들을 참조하여 보정을 하고 싶을 수도 있고, 다른 응용에서는 평활화 계수를 바꾸어 자료처리를 수행할 필요가 있을 수 있다. 이러한 예들은 특정 단계에 사용된 알고리즘 자체를 변경하는 것이라고 볼 수 있다. 또 다른 응용들에서는 자료처리 단계들의 순서를 바꾸거나 특정 단계를 삭제하거나 중복하는 경우가 요구될 수 있다. 예를 들면 결측보정 후에 바로 품질평가를 하고 싶은 경우가 있을 수 있고, 또는 급격한 데이터의 변동에 관심을 두어 평활화 단계를 원하지 않고 그 외의 단계들만을 수행하기를 원할 수도 있고, 결측보정이 실패하는 경우를 대비해서 몇 가지 다른 방식으로 여러 번 결측보정을 수행하고 싶을 수도 있다.

이와 같은 유연한 자료처리 과정을 지원하기 위하여 본 시스템에서는 what-if 분석 기능을 교통이력 데이터의 자료처리 과정에 제공되도록 설계하였다. What-if 분석이란 특정 계산이나 절차에서 다양한 입력 변수들이 정의되어 있어 이들에 다양한 실험값을 주고 결과들을 비교 분석할 수 있도록 해주는 기능을 말한다. What-if 분석 기능의 도입을 위하여 각 알고리즘별로 알고리즘 설계시에 입력 환경변수들을 정의할 수 있도록 하는 기능을 설계하였다. 그리고 추가로 자료처리 프로세스(process)라는 개념을 도입하였다. 프로세스란 알고리즘들의 실행 순서를 정의하며 각 알고리즘들의 환경변수에 입력될 구체적인 값들을 가지고 있다. 즉 하나의 프로세스는 특정 응용에 적합한 하나의 교통이력 데이터에 대한 자료처리 과정을 정의한다고 볼 수 있다. 프로세스 정의의 예를 들면, 자료처리 과정에서 알고리즘의 실행 순서를 원시자료 선처리, 품질평가, 오류판단, 결측보정 1, 결측보정2, 평활화, 품질평가로 정하고, 입력 환경변수로는 결측보정1은 이력데이터 중 유사 시간대의 데이터 3개씩을 참조하고 결측보정 2는 유사 시간대의 데이터 1개를 참조하고 평활화 계수는 0.3으로 설정하여 하나의 프로세스를 정의할 수 있다. 프로세스 정의시에는 고유한 이름이 주어진다. (그림 4)에서 what-if 분석을 위하여 프로세스의 정의에 필요한 요소들과 함께 프로세스의 개념을 시각적으로 보여주고 프로세스를 정의한 예를 보여준다.

이처럼 자료처리 과정을 정의하는 여러 개의 프로세스가 동시에 정의되고 수행되며 관리될 수 있어야 하므로 프로세스에 수반되는 몇 가지 복잡한 문제들을 해결하기 위한 기능도 설계하였다. 대용량 데이터 저장공간의 필요와 프로세스 처리 시간에 따른 시스템 자원의 사용을 효율적으로 하기 위하여 각 사용자별로 데이터셋(data set)을 개별적으로 관리하도록 하였고 프로세스의 관리 기능도 설계하였다. 프로세스의 진행현황에 대한 실시간 피드백 기능도 시각적으로 제공되어 프로세스의 중간에 어느 알고리즘을 처리 중인지 확인이 가능하도록 하였다.



(그림 4) 교통이력 데이터의 what-if 분석을 위한 프로세스 개념과 프로세스 정의 예

5. 자료처리 알고리즘과 what-if 분석 사용자 인터페이스 구현

교통이력 데이터의 자료처리에 사용되는 다양한 알고리즘들의 구현과 what-if 분석을 위해 제공되는 사용자 인터페이스의 구현 결과를 설명한다. 본 시스템에서는 자료처리 과정을 데이터베이스 내에 PL/SQL로 구현하여 처리의 효율성을 높였다. 다른 시스템들에서는 처리 과정이 데이터베이스 외부의 코드로 구현되어 데이터 접근 오버헤드가 많다.

5.1 원시 자료 선 처리

원시 자료의 선 처리(preprocessing)는 차량 검지기로부터 수집된 데이터의 수집 시간(sample_time), 교통량(LOOP_V), 속도(LOOP_S), 점유율(LOOP_O) 값을 자료처리가 가능한 상태로 변환하여 새로운 테이블에 삽입하여 준비해주는 단계이다. 원시 자료의 선 처리는 두 가지의 처리로 나뉘어지는데 데이터 변환 선처리와 결측 데이터 선처리이다. 데이터 변환 선처리는 데이터의 수집 시간인 sample_time 컬럼의 데이터를 00초, 30초 시간 단위로 보정하고 변환하여 일정한 주기의 단위 수집시간이 되도록 한다. 또한 루프 수에 대해서는 현 도로 상황에 적합하도록 24개의 데이터값들이 존재하도록 재구성한다. 즉, 각 루프 별 수집되는 교통량, 점유율, 속도 값이 24개가 되므로 컬럼을 교통량0, 점유율0, 속도0부터 교통량23, 점유율23, 속도23까지 만든다. 결측 데이터 선처리에서는 차량 검지기의 고장이나 차량의 부재로 인하여 발생한 결측 데이터를 -111의 값으로 삽입하여 다음에 이루어지는 자료처리에 있어서 결측 데이터를 쉽게 다루도록 해준다.

원시 자료의 선 처리는 두개의 프로시저로 나뉘어 구현하였으며 데이터 변환 선처리는 insert_1 프로시저로, 결측 데이터 선처리는 insert_2 프로시저로 구현하였다. 이들 프로시저들에서는 처리하고자 하는 교통 데이터의 시간 및 공간

범위를 입력 변수로 지정할 수 있게 하였고 결과로 만들어질 테이블 이름도 입력 변수로 지정할 수 있게 하였다.

5.2 품질 평가

품질 평가는 데이터의 품질을 평가하는 것이 목적이며 본 시스템에서는 완전성과 유효성이라는 두 개의 기준을 이용하여 품질평가를 수행한 결과를 제공한다. 다른 교통정보 시스템에서는 이러한 품질평가에 대한 개념을 제공하고 있지 못하여 본 시스템의 가장 큰 특징 중 하나로 볼 수 있다. 완전성은 결측이 아닌 이용 가능한 자료의 비율을 말한다. 교통량의 완전성을 구한다고 하면 프로시저 내에서 공식 (v_notmissing / v_sum) * 100 로 계산되는데 변수 v_notmissing에는 결측이 아닌 교통량 데이터의 개수, 즉 결측 자료 선처리의 결과로 삽입된 -111 값들을 제외한 교통량 데이터의 개수가 저장되고, 변수 v_sum에는 GetTotalCount 함수를 이용하여 얻은 차선 id에 해당하는 차로의 모든 교통량의 개수를 저장한다. 이를 퍼센트화 하면 완전성 값을 얻게 된다. 이는 점유율과 속도에도 적용되어 비율 값이 출력되도록 한다. 유효성은 결측이 아닌 이용 가능한 자료의 개수 중 유효한 자료의 개수의 비율이다. 교통량의 유효성을 구한다고 하면 프로시저 내에서 공식 100 - (v_invalid / v_notmissing * 100)으로 계산되는데 변수 v_invalid 에는 오류판단 기준에 의거한 오류에 해당하는 자료의 개수를 세어서 저장하고 변수 v_notmissing 에는 완전성 계산에서 저장되었던 이용 가능한 교통량 자료의 개수가 반환되어 사용되고 전체 100에서 감산하여 퍼센트화한다. 이는 점유율과 속도에도 적용되어 비율 값을 출력한다. 자료처리 전과 후 품질평가의 수행으로 출력된 완전성, 유효성의 결과 값은 'SYS_DP_RSLT'라는 테이블에 저장된다. 자료처리 전과 후에 수행되는 품질평가 알고리즘을 (그림 5)에서 보여준다.

5.3 오류판단

오류 데이터를 판별하기 위해 사용되는 오류 데이터 판별법으로 임계값 검사(Data Threshold)와 관계 검사(Data Relation)가 있다. 본 연구에서 오류판단 조건식으로 적용한 한국도로공사의 오류판단 기준안은 <표 1>과 같으며 이에 대한 근거는 다음과 같다.

오류판단에서 임계값 검사는 특정 값이 오류에 해당하는 수치 범위에 존재하는지를 검사한다. 예를 들어 교통량이 0

〈표 1〉 오류판단 기준

구분	오류판단 기준
임계값 검사	교통량 < 0 또는 교통량 > 30
	인접 검지기와의 교통량 차 > 2
	점유율 < 0 또는 점유율 > 100
관계 검사	교통량 = 0 그리고 속도 ≠ 0
	(S x O)/12 ≤ 2.7V 또는 (S x O)/12 ≥ 18V (V 교통량, O 점유율, S 속도)

```
PROCEDURE Quality_Evaluation          /* 품질평가 프로세스 */
INPUT: Table input_table;        /* input data table with volume, occupancy, speed columns */
       Time_Interval  time_spec;                  /* time interval for data to process */
       Location_Interval loc_spec;               /* location interval for data to process */
OUTPUT: Float completeness[], validity[];
BEGIN
 Table tmp_table; String curr_column;
 /*사용자 시공간 검색 조건에 따라 입력 자료 테이블로부터 검색*/
 ReadSourceData ( IN: input_table, time_spec, loc_spec;
                  OUT: tmp_table );
 FOR i = 1 TO 3 DO
BEGIN
 IF i=1 THEN curr_column = "volume"
 ELSE IF i=2 THEN curr_column = "occupancy"
 ELSE IF i=3 THEN curr_column = "speed"
 /*대상데이터의 교통량, 점유율, 속도의 개수를 세어 합산*/
 GetTotalCount (IN: tmp_table, curr_column; OUT: v_sum;)
 /*대상데이터의 교통량, 점유율, 속도의 결측 데이터 개수*/
 GetMissingCount (IN: tmp_table, curr_column; OUT: v_missing );
 v_notmissing = v_sum - v_missing;
 IF error_in_data= True THEN       /*대상데이터의 교통량, 점유율, 속도에 오류가 있음*/
   /*오류 포함한 교통량, 점유율, 속도의 개수를 세어 합산*/
    GetCountError (IN: tmp_table, curr_column; OUT: v_invalid );
    /*대상데이터의 완전성 계산 = 결측이 아닌 데이터의 개수/대상 데이터 개수 * 100*/
    completeness[i] = v_notmissing / v_sum * 100;
    /*대상데이터의 유효성 계산 = 100 - (오류 데이터의 개수/결측이 아닌 데이터 개수 * 100)*/
    validity[i] = 100 - (v_invalid / v_notmissing * 100);
 END IF;
END FOR;
RETURN (completeness[], validity[]);
END;
```

(그림 5) 품질평가 알고리즘

미만 또는 30 초과이면 오류이다. 이들 오류판단의 근거는 실제 상황에서 존재하기 어려운 값들을 다음과 같이 오류로 판단한다. 즉 30초 동안 지나가는 차량의 대수인 교통량이 0보다 적을 수는 없으며 현실적으로 30대보다 많을 수는 없다. 또한 인접 검지기들 간에 교통량의 차이가 2 보다 많으면 오류로 판단하는데 인접 검지기 간에 차량 대수의 변화가 2대 이상 차이가 나는 것은 현실적으로 어렵다. 즉 한번에 두 대의 차량이 인접 검지기 사이에서 차선 변화 등을 하기는 어렵다. 점유율 값도 0 미만 또는 100 초과이면 오류이다. 이것은 검지기를 차량이 점유하는 시간의 퍼센트 비율의 최소값과 최대값이 각각 0과 100이라는 근거로 정한 것이다.

오류판단에서 관계 검사는 둘 이상의 교통 관련 데이터들 사이에서 이들이 실제로 존재 가능한 관계성을 이루고 있는지를 검사하여 오류를 판단한다. 예를 들면 교통량은 0인데 속도가 0이 아니라면 이것은 실제로 존재가 불가능한 관계성이다. 즉 지나가는 차량이 없는데 속도가 0이 아닌 값이 나올 수 없다는 판단 근거에서 오류가 된다. 교통량과 점유율, 속도 간에도 <표 1>의 수식에 해당하는 관계성이 만족되면 오류로 판단된다. 속도와 점유율을 곱한 값을 이용하여 총 통행차량의 길이의 합을 구할 수 있는데 이 값이 교통량과 최소 길이의 차량의 곱보다 작거나 교통량과 최대 길이의 차량의 곱보다 크면 오류로 판단된다.

구현된 오류판단 프로시저는 각 검지기에서 나오는 데이터에 대하여 차례로 오류판단을 진행하며 오류인 데이터는 -999로 변경하는 역할을 한다. 오류판단 프로시저는 filtering_1 프로시저라 명하며 오류판단을 수행할 입력 테이블명을 사

```
  PROCEDURE Filtering       /* 오류판단 프로세스 */
INPUT: Table input_table; /* input data table with volume, occupancy, speed columns */      Time_Interval  time_spec;
                    /* time interval for data to process */
        Location_Interval loc_spec;                 /* location interval for data to process */
  OUTPUT:  Table processed_table;                        /* processed data tables */
  BEGIN
 Table tmp_table; Cursor tmp_cursor; Record curr_record;  /* cursor to iterate among table
                                                                    records */
   Float volume, occupancy, speed,
       min_v = 0, max_v = 30, min_o = 0, max_o = 100, min_s = 0, max_s = 200;
/*사용자 시공간 검색 조건에 따라 입력 자료 테이블로부터 검색*/
ReadSourceData ( IN: input_table, time_spec, loc_spec;
                 OUT: tmp_table );
GetCursor (IN: tmp_table; OUT : tmp_cursor );
   curr_record = tmp_cursor.getNext();
WHILE curr_record != NULL
volume = curr_record.volume; occupancy = curr_record.occupancy; speed = curr_record.speed;
IF Exist(volume) = True THEN                                    /* 교통량 데이터가 존재 */
 IF (volume < min_v OR volume > max_v) = False THEN              /* 교통량 데이터가 타당 */
   IF Exist(occupancy) = True THEN                              /* 점유율 데이터가 존재 */
      IF (occupancy < min_o OR occupancy > max_o) = False THEN  /* 점유율 데이터가 타당 */
        IF Exist(speed) = True THEN
          IF (speed < min_s OR speed > max_s) = False THEN       /* 속도 데이터가 타당 */
            IF (volume = 0 AND speed !=0) = False THEN      /* 교통량과 속도의 관계가 타당 */
             IF ( (speed * occupancy /12 <= 2.7* volume) AND
                  (speed * occupancy /12 >= 18* volume) ) = False THEN  Do nothing
                                                          /* 교통량,속도,점유율 관계가 타당 */
               ELSE BEGIN  /* 교통량, 속도, 점유율 오류 -999 로 채움 */
                           Error(IN: curr_record,"volume");
                         Error(IN: curr_record, "speed");
                         Error(IN: curr_record, "occupancy");
                    END;
             ELSE BEGIN  /* 교통량, 속도 오류 -999 로 채움 */
                           Error(IN: curr_record,"volume");
                         Error(IN: curr_record, "speed");
                    END;
          ELSE Error(IN: curr_record, "speed");       /* 속도 오류 -999 로 채움 */
        ELSE Missing(IN: curr_record, "speed");       /* 속도 누락데이터처리 */
      ELSE Error(IN: curr_record, "occupancy");       /* 점유율 오류 -999 로 채움 */
    ELSE Missing(IN: curr_record, "occupancy");       /* 점유율 누락데이터처리 */
  ELSE Error(IN: curr_record, "volume");             /* 교통량 오류 -999 로 채움 */
ELSE Missing(IN: curr_record, "volume");             /* 교통량 누락데이터처리 */
  curr_record = tmp_cursor.getNext();                /* 테이블에서 다음 레코드를 접근 */
  END WHILE;
  processed_table = tmp_table;
  RETURN (processed_table);
  END;
```

(그림 6) 오류판단 알고리즘

용자 입력 환경변수로 제공하고 있다. 또한 오류판단 조건들에서 사용하는 수치들을 입력 환경변수로 제공하여 조건식의 교통량 최대값과 점유율 최대값, 인접 검지기간 교통량 차 값을 원하는대로 변경할 수 있다.

오류판단 구현시에 주의했던 점은 오류판단 조건식의 계산시 결측데이터 -111은 제외하였으며, 반복적인 오류판단 자료처리의 수행으로 인하여 채워지는 오류 데이터 표시인 -999가 임계값 검사에서는 인접 검지기와의 교통량 차 > 2를 검사하는 조건식에 영향을 줄 수 있으므로, -111뿐 아니라 -999도 제외한 상태에서 판별이 이루어지게 된다. (그림 6)에서 오류판단 수행 알고리즘을 보여주고 있다.

5.4 결측보정

결측보정은 오류판단의 결과 발견한 오류데이터 및 누락 데이터를 결측 데이터로 보고 이러한 결측 데이터에 보정한 값을 채운다. 이력 데이터를 활용하면 결측 데이터에 대하여 유사한 지점 혹은 유사한 시간대의 정보를 참조하여 다양한 방법으로 결측 데이터를 채울 수 있다. 본 시스템에서 결측보정 기법은 크게 두 가지 유형으로 분류할 수 있다. 동일한 시간 정보와 인접한 공간 정보를 가지는 교통 자료를 참조하는 인접 지점 참조 기법과, 인접한 시간 정보와 동일한 공간 정보를 가지는 교통 자료를 참조하는 이력 자료 활용 기법이다. <표 2>에서 결측보정 기법들을 요약하였다.

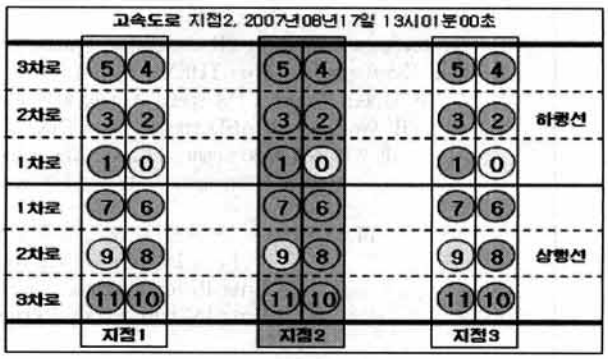
인접 지점 참조 기법은 차량의 유입과 출입이 없는 폐쇄된 구간 내의 교통 자료는 서로 유사성을 가지는 특성을 이용하여 결측 지점과 인접한 지점의 유효 자료를 참조하는 기법으로 전후 지점 동일 주기 적용, 이전 지점 동일 주기 적용, 지점간 이동 소요 주기 적용 기법이 있다. 결측 지점을 통과한 차량은 반드시 참조할 인접 지점을 통과했다는 가정이 포함된 기법으로, 결측 지점과 참조할 인접 지점 사이에 차량의 유입이나 출입이 있으면 적용할 수 없는 기법이다.

이력 자료 활용 기법은 주로 인접 지점 참조 기법으로 결측보정을 할 수 없는 경우 사용하는 기법으로, 동일 시간대의 교통 자료는 서로 유사성을 가지는 특성을 이용하여 이력 자료(즉 결측 시간 이전의 교통 자료)의 유효 자료를 참조하는 기법으로 이력 자료 동일 주기 적용 기법이 있다. 이력 자료 동일 주기 적용 기법은 결측 지점의 과거 교통 자료 중 결측 시점과 동일 요일, 동일 시간의 유효 자료를 참조하여 결측 자료를 보정하는 기법이다.

(그림 7)의 예를 이용하여 지점 참조 기법들을 설명한다. 전후 지점 동일 주기 적용 기법은 결측 지점과 인접한 전후 지점 내 동일 번호의 검지기 교통 자료 중 결측 주기와 동일한 주기의 유효 자료를 참조하여 보정하는 기법이다. (그림 7)은 2007년 08년 17일 13시 01분 00초에 고속도로 지점 2의 0번과 9번 검지기에서 결측 자료가 발생한 경우, 각 검지기는 어떤 검지기의 유효 자료를 참조하는지 보여준다. 지점2의 0번 검지기 결측 자료는 지점3의 0번 검지기(이전 지점)와 지점1의 0번 검지기(이후 지점)의 유효 자료 평균값으로 보정된다. 지점2의 9번 검지기 결측 자료는 지점1의 9번 검지기(이전 지점)와 지점3의 9번 검지기(이후 지점)의 유효 자료 평균값으로 보정된다.

위의 예에서 이전 지점 동일 주기 적용 기법은 결측 지점과 인접한 이전 지점 내 동일 번호의 검지기 교통 자료 중 결측 주기와 동일한 주기의 유효 자료를 참조하여 보정하는 기법이다. 지점간 이동 주기 적용 기법은 결측 지점과 인접한 이전 지점의 거리와 결측 지점의 결측 주기보다 한 주기 전 유효 자료 중 속도를 이용하여 지점간 이동 소요 주기(이전 지점을 지나 결측 지점으로 차량이 이동하는데 소요되는 주기)를 계산하여 결측 지점과 인접한 이전 지점 내 동일 번호의 검지기 교통 자료를 참조할 주기로 사용하는 기법이다. 이전 지점 동일 주기 적용 기법과 비교하면 참조 지점은 동일하나 참조 주기가 달라진다.

본 시스템에서 결측보정은 네 개의 프로시저로 구현되었으며 imputation_1(전후 지점 동일 주기 적용법)과 impu-



(그림 7) 전후 지점 동일 주기 적용 예시

〈표 2〉 결측보정 기법

분류	결측보정 기법	참조 공간	참조 시간
인접 지점 참조	전후 지점 동일 주기 적용	이전 지점, 이후 지점	결측 시점 (동일 날짜, 동일 시간)
	이전 지점 동일 주기 적용	이전 지점	결측 시점
	지점간 이동 소요 주기 적용	이전 지점	결측 시점 - 지점간 이동 소요 주기
이력 자료 활용	이력 자료 동일 주기 적용	결측 지점	동일 요일, 동일 시간

〈표 3〉 결측보정 기법의 입력 변수

분류	결측보정 기법	입력 변수
인접 지점 참조	전후 지점 동일 주기 적용	최대 x 번째 전후 지점 참조
	이전 지점 동일 주기 적용	최대 x 번째 이전 지점 참조
	지점간 이동 소요 주기 적용	최대 x 번째 이전 지점 참조
이력 자료 활용	이력 자료 동일 주기 적용	결측 시점 기준, 과거 x주 동안 이력 참조

tation_2(이전 지점 동일 주기 적용법), imputation_3(지점간 이동 소요 주기 적용법), imputation_4(이력 자료 동일 주기 적용법)라 명하였다. <표 3>은 각 결측보정 기법에 새롭게 추가한 입력 환경변수를 보여준다. (그림 8)은 결측보정 알고리즘을 보여준다.

5.5 평활화

평활화는 이상치를 제거하기 위해서 평활화 계수 k를 두어 가중치에 따라 이상치 평가 정도를 조절하게 된다. 평활화 공식은'현재 주기 평활화 = (1-k) * 이전 주기 평활화 + k * 현재 주기 측정 값'과 같다. 평활화 계수 k는 보통 교통

```
  PROCEDURE Imputation          /* 결측보정 프로세스 */
INPUT: Table input_table; /* input data table with volume, occupancy, speed columns */      Time_Interval  time_spec;
                    /* time interval for data to process */
       Location_Interval loc_spec;                  /* location interval for data to process */
       Int mode;                          /* mode = 1: 전후지점평균법, 2: 이전데이터이용법,
                                             3: 지점간이동주기법, 4: 이력데이터 활용법 */
 OUTPUT:  Table processed_table;                    /* processed data tables */
 BEGIN
Table tmp_table; Cursor tmp_cursor; Record curr_record;  /* cursor to iterate among table
                                                                  records */
Int k,i,j,l; Float imputation; String field;
ReadSourceData ( IN: input_table, time_spec, loc_spec;
                 OUT: tmp_table );
GetCursor (IN: tmp_table; OUT: tmp_cursor );
  curr_record = tmp_cursor.getNext();
WHILE curr_record != NULL
    FOR k=1 TO 3 DO                                 /* for all fields in record */
    BEGIN
      IF k=1 THEN field = "volume"
      ELSE IF k=2 THEN field = "occupancy"
      ELSE field = "speed";
      IF ErrorOrMissing(curr_record, field) = True THEN  /* 모든 오류보정 대상데이터에 대하여 */
      BEGIN
      IF mode = 1 THEN              /* 전후지점평균법 사용 */
      BEGIN
         i = Before(IN: curr_record, field);
          j = After(IN: curr_record, field);
          imputation = i+j/2;                  /* 보정값 = (이전지점값 + 이후지점값)/2 */
          Update(IN: curr_record, field, imputation);        /* 보정값 기록 */
      END IF;
      ELSE IF mode = 2 THEN         /* 이전데이터이용법 사용 */
      BEGIN
          imputation = Before(IN: curr_record, field);   /* 보정값 = 이전지점값 */
          Update(IN: curr_record, field, imputation);
      END IF;
      ELSE IF mode = 3 THEN         /* 지점간이동주기법 사용 */
      BEGIN
         imputation = BeforePeriod(IN: -T, curr_record, field);
                                                    /* 보정값 = 이전지점값(T주기 이전 시간) */
          Update(IN: curr_record, field, imputation);
      END IF;
      ELSE IF mode = 4 THEN         /* 이력데이터 활용법 사용 */
       BEGIN
         FOR l=1 TO 4 DO
         IF ErrorLane(curr_record, i) = True THEN imputation = History(i);
         Update(IN: curr_record, field, imputation);  /*1차로결측은 1차로 이력데이터로 대체*/
                                              /*2,3,4차로결측도 각기 2,3,4차로 이력데이터로 대체*/
         END FOR
      END IF;
     END IF;
 END FOR;
 curr_record = tmp_cursor.getNext();
 END WHILE;
 processed_table = tmp_table;
 RETURN (processed_table);
 END;
```

(그림 8) 결측보정 알고리즘

분야에서 적용하는 바와 동일하게 0.3을 기본 값으로 하며 본 시스템에서는 사용자의 의도에 따라 값을 변경할 수 있도록 입력 환경변수로 제공한다. 평활화 공식 수행의 예를 들면 '107 := 107.9 = 0.7 * 126 + 0.3 * 99'이 있다.

평활화 프로시저는 smoothing_1 이라 명한다. 입력 환경변수로 입력, 출력 테이블명의 값을 지정할 수 있으며 입력 테이블은 보통 결측보정을 거친 결과 테이블을 입력 받게 된다.

5.6 What-if 분석을 위한 사용자 인터페이스

교통이력 데이터의 자료처리를 위한 여러 개의 알고리즘들은 조합되어 일련의 순서에 따라 수행되는 프로세스를 형성한다. 이러한 자료가공 프로세스 실행 요청 화면을 그림 10에 보여준다. 자료가공 프로세스를 실행하기 위해서 사용자는 먼저 하단의 이미 정의된 프로세스 리스트로부터 사용할 프로세스를 선택한다. 프로세스를 선택하고 나면, 가공을 원하는 데이터 셋을 선택한다. 프로세스와 데이터셋을 선택하고 나면, 프로세스를 구성하는 알고리즘들이 요구하는 입력 환경변수 값들을 설정하여야 한다. 입력 환경변수 값을 모두 설정하고 나서, 실행요청 버튼을 클릭하면 자료처리 프로세스 실행요청이 완료된다.

프로세스 실행요청이 이루어지면, 프로세스를 관리하는 SYS_DP_RSLT라는 테이블에 새 레코드가 추가된다. 자료처리 서비스는 SYS_DP_RSLT 테이블을 주기적으로 조회하여, 요청된 프로세스가 있다면, 해당 프로세스를 자동적으로 실행한다. 자료처리 프로세스가 실행되면, 프로세스의 진행상황을 사용자가 모니터링할 수 있도록 SYS_DP_RSLT 테이블과 SYS_DP_STATUS 테이블에 계속해서 기록하게 된다. 프로세스 실행요청 시에 설정되는 입력 환경변수 값들은 SYS_DP_PARAM 테이블에 저장된다.

자료처리 실행요청 후, 자료처리 결과조회 메뉴를 통해서 사용자는 자료처리 과정을 모니터링할 수 있다. (그림 11)은 자료처리 과정을 모니터링 하는 화면이다. 화면에서는 현재 '서울-대전 하루' 데이터셋을 처리 중이며, 알고리즘 insert_1, insert_2, quality_1, filtering_1이 실행되었고, 현재는 imputation_1이 실행중임을 알 수 있다.

사용자는 새로운 알고리즘을 작성하여 입력 환경변수들과 함께 시스템에 등록할 수도 있으며 새로운 프로세스를 정의할 수도 있다. 새로운 프로세스를 정의할 때에는 원하는 알

```
  PROCEDURE Smoothing                    /*지수 평활화 프로세스 */
INPUT: Table input_table; /* input data table with volume, occupancy, speed columns */      Time_Interval  time_spec;
                    /* time interval for data to process */
       Location_Interval loc_spec;              /* location interval for data to process */
       Boolean user_input_smoothing_constant;    /* True if user sets smoothing value */
       Float input_value, t;                 /* smoothing value given by user, period value*/
 OUTPUT:  Table processed_table;                    /* processed data tables */
 BEGIN
Table tmp_table; Cursor tmp_cursor; Record curr_record;  /* cursor to iterate among table
                                                              records */
Int i; Float K, smoothing_result; String field; Int t;
IF user_input_smoothing_constant=True THEN K=input_value;      /* 사용자 평활화계수 입력 */
ELSE  K=0.3;
ReadSourceData ( IN: input_table, time_spec, loc_spec;
                 OUT: tmp_table );
GetCursor (IN: tmp_table; OUT: tmp_cursor );
  curr_record = tmp_cursor.getNext();
WHILE curr_record != NULL
    FOR i=1 TO 3 DO                              /* for all fields in record */
    BEGIN
      IF i=1 THEN field = "volume"
      ELSE IF i=2 THEN field = "occupancy"
      ELSE field = "speed";
    IF NeedSmoothing(curr_record, field) = True THEN    /* 지수평활화 대상데이터 */
    BEGIN
      smoothing_result = (1-K)*Smoothing_result(IN: t-1, curr_record, field) +
                          K*Smoothing_data(IN: t, curr_record, field);
                    /* 평활화출력데이터 = (1-K)*평활화데이터(t-1) + K*입력데이터(t); */
       Update(IN: curr_record, field, smoothing_result);
      END IF;
    END FOR;
curr_record = tmp_cursor.getNext();
END WHILE;
 processed_table = tmp_table;
 RETURN (processed_table);
 END;
```

(그림 9) 지수 평활화 알고리즘

(그림 10) 프로세스 실행 요청 화면

(그림 11) 자료처리 과정 모니터링 화면

고리즘들을 선택하여 실행순서를 정해주면 된다. 다양한 형태의 프로세스를 정의하고 자료처리를 수행하여 결과를 확인해봄으로써 what-if 분석을 위한 기능을 제공한다.

6. 교통이력 데이터의 자료처리 실험 및 검증

본 절에서는 먼저 이론적으로 설계된 알고리즘과 프로세스가 실제 다양한 예외적인 상황들을 갖고 있는 교통이력 데이터에 적용하였을 때 의도한대로 정확하게 결과를 제공하는지를 실험하고 검증한 결과를 설명한다. 그리고 교통이력 데이터에 대하여 자료처리가 진행된 후의 실제 데이터들의 모습을 일부 검토하고, 실험에서 자료처리 전후의 교통이력 데이터의 품질이 향상된 것을 보여준다.

· 자료처리를 실제 교통데이터에 적용시 알고리즘의 구현 정확성 검증

교통이력 데이터에는 다양한 상황들이 반영되어 많은 오류 데이터들도 존재하며 다양한 도로 차선 형태나 구조, 다양한 검지기 설치 상황, 예측 불가능한 교통 돌발 상황 등으로 인하여 이론적으로 설계된 알고리즘들이 실제 교통 데이터에 적용하였을 때 원하는 결과를 얻지 못하는 경우가 많다. 그리하여 본 시스템에서 구현된 자료처리 알고리즘들이 다양한 실제 상황을 반영한 실제 교통 데이터에 대해서도 제대로 정확하게 동작하는지를 검증하기 위하여 30초 집계된 원시데이터를 구현된 시스템을 이용하여 자료처리를 수행한 결과와 30초 집계된 원시데이터를 직접 수작업으로 자료처리한 결과를 비교하였다.

자료처리 검증 사용자료 시간범위는 2006년 10월 10일 00시 00분 00초부터 2006년 10월 10일 23시 59분 30초까지 24시간 데이터를 사용하였다. 자료처리 검증 사용자료 공간범위는 경부선 6개 검지기 지점(001LD3343 ~001LD3387)을 사용하였다. 동일 CONG_ZONE내에서 결측보정이 이루어지고 있는가를 검증하기 위하여 두 개의 CONG_ZONE이 만나는 지점을 선택하였고, 차로가 변경되는 구간에서의 결측보정 수행 여부를 검증하기 위하여 검증구간에 3차로에서 4차로로 변하는 구간을 선택하였다. 자료처리 검증 방법은 각 단계에서 완전성과 유효성을(-999값과 -111값 개수 비교) 비교하여 검증하였다. 검증의 경우들은 오류판단-전후지점동일주기(참조검지기 parameter=2)-평활화(평활화계수=0.3), 오류판단-이전지점동일주기(참조검지기 parameter=3)-평활화(평활화r 계수=0.3), 오류판단-지점간 이동소요주기(참조검지기 parameter=2)-평활화(평활화 계수=0.3)으로 총 3가지 경우에 대하여 검증하였다. 검증 방법을 정리하면 <표 4>와 같다.

자료처리 검증과정 수행 결과, 각 단계별 완전성과 유효성을 비교하여 자료처리 프로그램을 수행한 결과와 직접 검증한 결과를 비교한 값이 오차 없이 정확히 일치한 결과를 나타내었다. 각 단계별 완전성과 유효성 결과는 <표 5>와 같다.

· 자료처리 전후의 데이터 내용 검토

(그림 12)와 (그림 13)은 이전 지점 동일 주기 적용 기법을 사용한 경우의 전과 후의 데이터 모습을 보여준다. sample_time은 시간 정보, lds_id는 공간 정보를 나타낸다. loop는 검지기를 의미하며 (그림 12)에서 표시한 부분이 결측 자료로 -999로 채워져 있다.

〈표 4〉 자료처리 검증 방법

자료처리	자료처리 방법	사용자 입력 환경변수	검증항목
완전성 유효성	-	-	· 자료의 결측정도와 유효데이타 비율
오류판단	· data threshold	· data threshold 값	· v_max_v = 30 · v_max_o = 100 · v_difference_v = 2
결측보정	· 전후지점동일주기 이동평균	· 참조할 검지기 개수	· parameter = 2
	· 이전지점 동일주기		· parameter = 3
	· 지점간 이동 소요주기		· parameter = 2
	· 이력자료 동일주기	-	-
평활화	· 지수평활화	· 평활화 계수(K)	· k=0.3
완전성 유효성	-	-	· 자료의 결측정도와 유효데이타 비율

〈표 5〉 자료처리 검증 결과

자료처리 (품질평가 기준)	자료처리 결과	수작업 검증 결과	Error(%)
완전성 / 유효성	81.9% / 96.2%	81.9% / 96.2%	0
오류판단 (완전성/유효성)	81.9% / 96.2%	81.9% / 96.2%	0
결측보정_이전지점 (완전성/유효성)	82.0% / 98.1%	82.0% / 98.1%	0
결측보정_전후지점 (완전성/유효성)	82.0% / 99.2%	82.0% / 99.2%	0
결측보정_이동소요주기(완전성/유효성)	82.1% / 97.4%	82.1% / 97.4%	0
이전지점_평활화 (완전성/유효성)	82.0% / 91.3%	82.0% / 91.3%	0
전후지점_평활화 (완전성/유효성)	82.0% / 92.4%	82.0% / 92.4%	0
이동소요주기_평활화 (완전성/유효성)	82.1% / 90.6%	82.1% / 90.6%	0

SAMPLE_TIME	LDS_ID	LOOP_V2	LOOP_O2	LOOP_S2	LOOP_V3	LOOP_O3	LOOP_S3
20061010000000	0010LD0115	-999	-999	-999	-999	-999	-999
20061010000000	0010LD0124	1	1	100	-999	-999	-999
20061010000000	0010LD0132	2	1	103	2	1	103
20061010000000	0010LD0140	2	1	112	2	1	112
20061010000000	0010LD0151	2	2	116	2	2	116
20061010000030	0010LD0115	3	2	93	3	2	93
20061010000030	0010LD0124	2	1	92	2	1	92
20061010000030	0010LD0132	3	2	102	3	2	102
20061010000030	0010LD0140	1	1	109	1	1	109
20061010000030	0010LD0151	-999	-999	-999	-999	-999	-999
20061010000100	0010LD0115	-999	-999	-999	-999	-999	-999
20061010000100	0010LD0124	3	1	111	3	1	111
20061010000100	0010LD0132	-999	-999	-999	-999	-999	-999
20061010000100	0010LD0140	-999	-999	-999	-999	-999	-999
20061010000100	0010LD0151	2	1	80	2	1	80

(그림 12) 원시 자료 (이전 지점 동일 주기 적용 기법 수행 전)

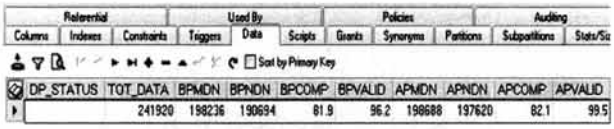
SAMPLE_TIME	LDS_ID	LOOP_V2	LOOP_O2	LOOP_S2	LOOP_V3	LOOP_O3	LOOP_S3
20061010000000	0010LD0115	-999	-999	-999	-999	-999	-999
20061010000000	0010LD0124	1	1	100	-999	-999	-999
20061010000000	0010LD0132	2	1	103	2	1	103
20061010000000	0010LD0140	2	1	112	2	1	112
20061010000000	0010LD0151	2	2	116	2	2	116
20061010000030	0010LD0115	3	2	93	3	2	93
20061010000030	0010LD0124	2	1	92	2	1	92
20061010000030	0010LD0132	3	2	102	3	2	102
20061010000030	0010LD0140	1	1	109	1	1	109
20061010000030	0010LD0151	2	2	116	2	2	116
20061010000100	0010LD0115	3	2	93	3	2	93
20061010000100	0010LD0124	3	1	111	3	1	111
20061010000100	0010LD0132	3	2	103	3	2	103
20061010000100	0010LD0140	2	1	111	2	1	111
20061010000100	0010LD0151	2	1	80	2	1	80

(그림 13) 자료처리 후 (이전 지점 동일 주기 적용 기법 수행 후)

<그림 13>은 이전 지점 동일 주기 적용 기법에서 이전 지점의 개수를 의미하는 입력 환경변수를 2로 주었을때의 보정한 결과이다. 여전히 보정되지 않은 부분은 참조 조건을 만족하는 유효 자료가 없어 보정하지 못한 교통 자료이다.

· 자료처리 전후의 품질향상 검토

(그림 14)는 위의 원시자료의 오류판단과 결측보정 전후의 데이터 품질을 저장한 테이블의 데이터를 보여준다. BPCOMP 컬럼과 BPVALID 컬럼은 자료처리 전 품질평가 프로시저인 quality_1을 수행한 결과가 저장되는데 각각 완전성과 유효성으로 81.9 퍼센트와 96.2 퍼센트를 나타내었다.



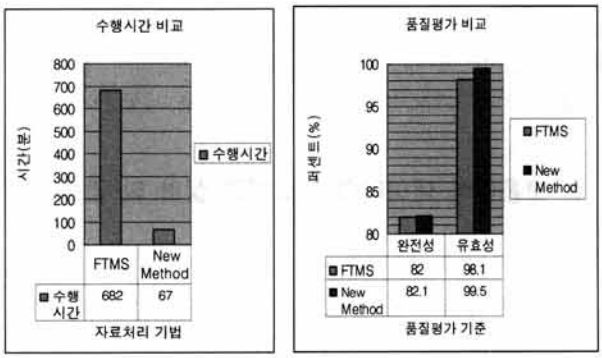
DP_STATUS	TOT_DATA	BPMDN	BPNDN	BPCOMP	BPVALID	APMDN	APNDN	APCOMP	APVALID
	241920	198236	190694	81.9	96.2	198688	197620	82.1	99.5

(그림 14) 품질평가 프로시저 quality_1과 quality_2를 수행한 결과 테이블

APCOMP 컬럼과 APVALID 컬럼은 자료처리 후 품질평가 프로시저인 quality_2를 수행한 결과가 저장되는데 각각 완전성과 유효성으로 82.1 퍼센트와 99.5 퍼센트를 나타내었다. 결과적으로 오류판단과 결측보정으로 인하여 이용 가능한 자료의 비율과 유효한 자료의 비율이 높아진 것을 알 수 있다.

· 기존의 방법과의 수행시간 및 품질향상 비교

기존의 FTMS에서 사용하는 자료처리 기법과 본 연구에서 제안한 자료처리 기법의 수행시간 및 품질향상 정도를 비교하였다. 실험 환경 및 데이터는 위의 예를 사용하였으며 본 연구의 오류판단과 결측보정을 포함한 자료처리 기법을 적용하였다. 이들 성능을 비교한 결과를 (그림 15)에서 보여준다. 수행시간 측면에서는 본 연구의 제안한 기법이 FTMS 기법보다 약 10배 이상의 우수한 성능을 보이며 품질향상 측면에서는 FTMS의 기법보다 완전성이 0.1, 유효성이 1.4 증가하는 수치를 보여 완전성과 유효성을 모두 높여주었음을 보인다.



(그림 15) 기존의 FTMS 기법과 제안한 기법의 자료처리 성능 비교

7. 결 론

본 논문에서 구현한 통합 교통이력 데이터베이스 시스템은 교통 데이터를 지속적으로 누적하며 품질을 높여 체계적이고 융통성있는 what-if 분석을 지원하는 자료처리 기능을 제공함으로써 교통 데이터의 활용도를 획기적으로 높여준다. 그동안 교통 데이터는 매우 많은 활용 가치가 있음에도 불구하고 품질 문제 및 처리 알고리즘과 처리 시스템의 지원 부족으로 충분히 활용되지 못했던 문제를 가지고 있었다. 통합 교통이력 데이터베이스 시스템은 교통 데이터를 다차원 모델로 지속적으로 저장하고 이용자 서비스를 제공하는 미들웨어도 포함한다. 또한 교통 데이터의 품질을 향

상시키기 위해 필요한 오류판단, 결측보정, 평활화 알고리즘들과 함께 교통 데이터의 완전성과 유효성을 측정할 수 있는 품질평가 방법을 제공한다. 그리고 교통이력 데이터에 대해 다양한 상황을 반영하고 실험하여 결과를 분석할 수 있는 what-if 분석 기능을 제공한다. 이를 위해 각 알고리즘별 입력 환경변수를 정의할 수 있도록 해주고 프로세스라는 개념을 도입하여 다양한 형태의 자료처리 과정들을 정의하게 해준다. 자료처리 과정은 데이터베이스 내에 PL/SQL로 구현하여 성능을 높였고 웹 기반의 사용자 인터페이스를 통하여 사용자가 교통이력 데이터에 대한 자료처리를 수행하고 진행현황을 모니터링하고 결과를 조회할 수 있게 하였다.

본 시스템은 교통이력 데이터의 품질을 높이고 다양한 what-if 분석을 할 수 있게 함으로써 교통 정책 및 전략의 수립, 도로 설계, 공휴일 및 명절 수송 대책, 지정체 구간의 분석 등의 많은 분야에 활용이 가능하다. 향후 연구로는 PL/SQL로 구현된 코드를 추가적으로 최적화하여 수행 시간을 단축시키는 것과 더욱 다양한 외부 데이터와 연계한 분석 기능을 제공하는 것도 고려할 수 있다. 예를 들면, 기상정보 등과 연계하면 날씨에 따른 교통 추이를 분석하는 기능을 제공할 수도 있다. 다만 기상정보는 교통 정보의 공간적 개념과 일치하지 않아 좀더 공간 차원의 데이터에 대한 다른 접근 방식이 연구되어야 한다.

참 고 문 헌

[1] 한국도로공사, '고속도로 차량검지기자료 조사・분석 및 활용기법 개발', 한국ITS학회, 2006.

[2] PeMS, https://pems.eecs.berkeley.edu/

[3] Ishak, S., S. Kondagari, C. Alecsandru, "Probabilistic Data-Driven Approach for Real-Time Screening of Freeway Traffic Data," In Transportation Research Record 2012, TRB, National Research Council, pp.94-104, Washington D.C., 2008.

[4] Smith, B. L., R. Venkatanarayana, "New Methodology for Customizing Quality Assessment Techniques for Traffic Data Archives," In Transportation Research Record 1993, TRB, National Research Council, Washington D.C., 2007. pp. 165-174.

[5] 차창일, 원정임, 김상욱, "대용량 궤적 데이터를 위한 효과적인 인덱싱 기법," 한국정보처리학회 춘계학술 발표대회 논문집 제15권 제1호, pp. 227-230, 2008.

[6] 박원식, 김동근, 양영규, "Fuzzy c-means 알고리즘을 이용한 TCS 데이터 주행특성 분류 방법 연구," 제31회 한국정보처리학회 춘계학술발표대회 논문집 제16권 제1호, pp.1021-1024, 2009.

[7] 김형준, 윤태진, 조환규, "시계열 데이터의 양자화된 문자열 변환을 통한 새로운 패턴 분석 기법," 제31회 한국정보처리학회 춘계학술발표대회 논문집 제16권 제1호, pp. 523-526, 2009.

[8] 김용욱, 이준우, 나연묵, "범용 위치 기반 웹 서비스 시스템," 제31회 한국정보처리학회 춘계학술발표대회 논문집 제16권 제1호, pp. 543-546, 2009.

[9] 김정연, 이영인, 백승걸, 남궁성, "차량 검지자료 결측보정처리에 관한 연구(이력자료 활용방안을 중심으로)," 대한교통학회지, 제24권 제7호, pp. 27-40, 2006.

[10] ITS사업실, 도로교통기술원 교통연구그룹, "ITS 구축・운영 편람," 한국도로공사, 2005.

[11] 한국도로공사 도로교통기술원, "차량검지기 자료의 효율적 수집저장 및 관리체계 연구," 한국도로공사, 2006.12

[12] Smith, B., and S. Babiceanu, "Investigation of Extraction, Transformation, and Loading Techniques for Traffic Data Warehouses," In Transportation Research Record 1879, TRB, National Research Council, pp.9-16, Washington D.C., 2004.

[13] Smith, B. D. Lewis, R. Hammond, "Design of Archival Traffic Databases: Quantitative Investigation into Application of Advanced Data Modeling Concepts," In Transportation Research Record 1836, TRB, National Research Council, pp. 126-131, Washington D.C., 2003.

[14] Al-Deek, H. M., C. Chandra, "New Algorithms for Filtering and Imputation of Real-Time and Archived Dual-Loop Detector Data in I-4 Data Warehouse," In Transportation Research Record 1867, TRB, National Research Council, pp. 116-126, Washington D.C., 2004.

이 민 수

e-mail : mlee@ewha.ac.kr

1992년 서울대학교 컴퓨터공학과 졸업(학사)

1995년 서울대학교 컴퓨터공학과(공학석사)

2000년 University of Florida 컴퓨터공학과 (공학박사)

1995년~1996년 LG전자 미디어통신연구소 연구원

2000년~2002년 미국 Oracle Corporation, Senior Member of Technical Staff

2002년~현 재 이화여자대학교 컴퓨터학과 부교수

관심분야 : 데이터웨어하우스, XML, 지식기반 시스템, 웹 데이터베이스

정 수 정

e-mail : bloom01@ewhain.net

2005년 목원대학교 컴퓨터교육학과 졸업 (학사)

2008년 이화여자대학교 컴퓨터공학과 졸업 (공학석사)

2008년~현 재 한국공간정보통신

관심분야 : 데이터웨어하우스, 교통 데이터 처리

최 옥 주

e-mail : pensica@naver.com
2008년 이화여자대학교 컴퓨터공학과 졸업 (학사)
2008년~현 재 이화여자대학교 컴퓨터공학과 석사과정
관심분야 : 데이터웨어하우스, 데이터마이닝, 스트림 데이터 처리

맹 보 연

e-mail : mngby@ewhain.net
2007년 성결대학교 컴퓨터공학과 졸업(학사)
2008년~현 재 이화여자대학교 컴퓨터공학과 석사과정
관심분야 : 데이터웨어하우스, 데이터마이닝, 임베디드 DBMS