

## 동적 분할 기법을 이용한 네트워크 계층 모델에 관한 연구

주용진\* · 이용익\*\* · 문경기\*\*\* · 박수홍\*\*\*\*

### A Study on Network Hierarchy Model which uses a Dynamic Segmentation Technique

Yong-Jin Joo\* · Yong-Ik Lee\*\* · Kyung ky Moon\*\*\* · Soo-Hong Park\*\*\*\*

#### 요 약

이동성을 지원하는 모바일 환경에서 위치정보의 활용과 사용자 요구가 증가되고 GIS 공간 DB와 연계된 다양한 서비스가 진행되고 있다. 일반적으로 도로 DB를 구성하는 교통 네트워크의 논리적 관계의 표현은 노드-링크 구조를 사용한다. 이러한 단일 수준에 적합하게 설계된 구조는 다양한 모형 적용에 유연하지 못하고, 데이터베이스 검색과 유지관리 측면에서 비효율적이다. 본 연구에서는 동적 분할(Dynamic Segmentation)을 이용한 네트워크 모델의 설계와 구축을 통해 기존 도로망 모델의 문제점과 구축상의 한계점을 보완하고, 네트워크의 검색과 표현에 효율적인 계층 모델을 구현하고자 하였다. 설계된 모델은 다양한 수준의 단계별 표현과 계층 간 개체 관계성을 지원하며, GIS가 지닌 네트워크 공간 모델링 기능을 대폭 보완할 수 있을 것으로 기대된다.

주요어 : 동적 분할, 선형참조, 네트워크 계층 모델, 멀티 링크

**ABSTRACT** : A practical use of the location information and user requirement are increased in a mobile environment which supports the portability. And Various service which GIS is related with a Spatial DB have been processed. Generally, logical relation of a traffic network which organizes the Road DB uses a basic node-link structure. In this way, Designed structure can not be flexible at various model apply and are not efficient with a

\* 인하대학교 지리정보공학과 박사과정

\*\* 인하대학교 지리정보공학과 석사과정

\*\*\* 인하대학교 지리정보공학과 석사과정

\*\*\*\* 인하대학교 지리정보공학과 부교수

database retrieval in a maintenance management side. In this research, We supplement with the problem of a existing network model and the limitation of the building through the design of a network model which uses dynamic segmentation. And we tried to implement efficient hierarchy model at the retrieval of the network and presentation. Designed model supports a stage presentation of various level and a hierarchy entity relation and We are expected to supplement a network spatial modelling function which the GIS has.

**Keywords** : Dynamic Segmentation, Linear Reference System, Network Hierarchy Model, Multi Link

## 1. 서 론

### 1.1 연구배경과 필요성

이동성을 지원하는 모바일 환경에서 위치정보의 활용과 사용자 요구가 증가되고 GIS 공간 DB와 연계된 다양한 서비스가 진행되고 있다. 특히, 교통과 위치정보 기반의 이동형 정보 활용을 위한 텔레매틱스·ITS·LBS 서비스는 u-IT839 8대 서비스의 하나로 신 성장 동력을 위한 핵심 전략 분야로 주목받고 있다.

텔레매틱스 서비스의 중요한 요소인 GIS DB 기술은 공간데이터 스키마와 연관자, 공간 인덱스, 공간참조를 통한 공간 데이터 관리, 공간개체 검색기술을 포함하는 공간서비스, 차량항법서비스를 위한 위치기반서비스 등이 있다.(건설교통부, ITS Korea 2003) 이 중 위치기반 서비스에 활용되는 차량 탑재형 GIS DB는 배경, 네트워크, POI, 문자로 구성된다. 배경 DB는 선, 면을 중심으로 화면에 표시되는

그래픽 요소의 집합으로, 일반적으로 신속한 출력을 위해 도엽과 레벨별로 중복 저장하는 경우가 많다. 도로 DB는 도로 중심선과 교차로 연결 상태를 추상화하여 나타낸다. 또한 교차로에서의 회전규제, 도로 연결성 등에 대한 정보를 계층적 구조로 저장시켜 경로 서비스를 제공한다. (한국정보통신기술협회 2003)

일반적으로 도로 DB를 구성하는 교통 네트워크는 두 개의 선이 교차할 때마다 하나의 점이 존재하는 기본 노드-링크 구조를 사용하여, 상호간의 논리적 관계나 물리적 연결 표현을 한다. 이러한 구조의 문제점은 모든 링크의 교차점들에서 노드가 존재하여 언더패스나 오버패스와 같은 현실세계의 교통 네트워크 특징을 반영할 수 없다. 또한, 개체간 일대다(one-to-many)의 관계를 나타낼 수 없어 단일 링크를 균질(homogeneous)하게만 표현한다. 즉, 속성이 변하는 부분(제한속도, 차로 수, 포장 재질 등)에 대해 유연한 표현이 어렵다. 이러한 정보들을 표현하기 위해 링크를 분할한다면 도로가 지나치게 세분화되고, 동

일한 정보가 다수의 링크에 대해 중복 저장되어 데이터베이스의 양이 급격히 증가된다. 결국, 동일 개체에 대한 다중 참조(multiple reference)는 데이터 변경의 불일치를 유발시키고 다른 사용자들과 데이터 공유를 어렵게 한다.

이를 해결하기 위해 기존 선형구조는 변화하지 않고 동적으로 연관된 속성의 집합을 구성하는 데이터 모델링 기법이 요구되고, 계층적 도로 DB의 검색과 표현에 효율적인 모델에 대한 고찰과 모델 적용을 위한 효과적인 통합·변환 과정의 정립이 필요하다.

## 1.2 연구목적과 방법

본 연구에서는 동적 분할을 이용한 네트워크 모델의 설계와 구축을 통해 기존 도로망 모델의 문제점과 구축상의 한계점을 보완하고, 네트워크의 검색과 표현에 효율적인 특성을 지원하는 모델을 구현하고자 한다. 이러한 네트워크 모델(멀티 링크)이 도로 DB의 계층 구조를 지원하기 위한 특성은 아래와 같다. 지도 축척에 따라 표현할 대상을 설정하고 저장하기 위한 계층 구조로 구축되어 레벨 간 디스플레이가 가능하도록 한다. 개체 사이의 상·하 관계성을 정의 할 수 있어야 한다. 또한 도로의 위상구조에 따라 네트워크 연결성과 근접성을 고려한 저장구조로 검색의 성능을 향상될 수 있도록 하고(Dimitis Papadias, Jun Zhang 2003), 도로 DB가 차지하는 데이터 저장 용량이 경량

화 되어 단순성을 가져야 한다.

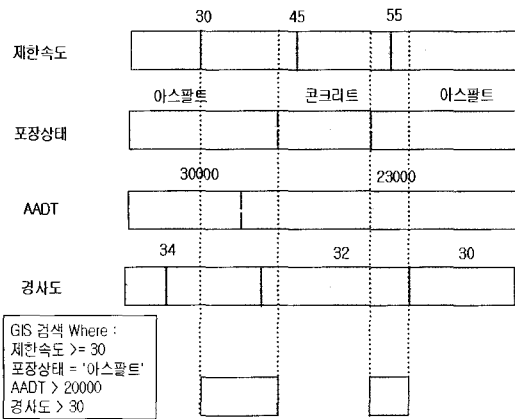
이를 위해 본 연구에서는 첫째, 모델 적용을 위한 노드·링크 체계와 레벨을 정의하고 둘째, 기존 노드-링크 모델의 문제점을 파악하여 동적 분할을 적용한 개선된 모델과 알고리즘을 설계하였으며 셋째, 제시된 알고리즘을 인천·서울권역 도로망에 적용하여 멀티링크로 구성된 계층 도로망을 구축하고, 마지막으로 구축된 데이터를 공간 데이터베이스(SQLite) 저장한 후 분석실험을 통해 모델을 평가하였다.

## 2. 동적 분할 (Dynamic Segmentation) 개요

### 2.1 동적 분할 개념

일반적으로 지리정보 시스템에서 도시의 도로망, 노선, 하천의 흐름 등은 네트워크로 추상화되고, 정적인 단일 특성을 가진 선형 개체로 구축된다. 하지만, 때로는 선형의 상대적인 위치에 따라 변화하는 동적인 특성을 표현하는 모델로 다뤄질 필요가 있다. 이러한 예로 도로의 포장상태, 차선 변화, 교통량 변화, 관리상태, 고속도로 상의 사고지점, 시설물(교량, 지하, 고가), 하천의 유량계, 하수관망에서의 맨홀 등이 있다. 이것은 하나의 선형 개체와 연관된 다른 개체들과의 단순한 중첩을 통해서서는 불가능하다. 선형 참조모델(Linear Referencing System)에서 이러한 개체를 유지관리를 가능하게 하기 위해, 아래 [그림 1]에서와 같이 선형 개

체의 부분을 다중 속성의 집합으로 연관시켜 표현하도록 동적 분할기법(Dynamic Segmentation)을 사용한다. (Bo Huang, 2002)



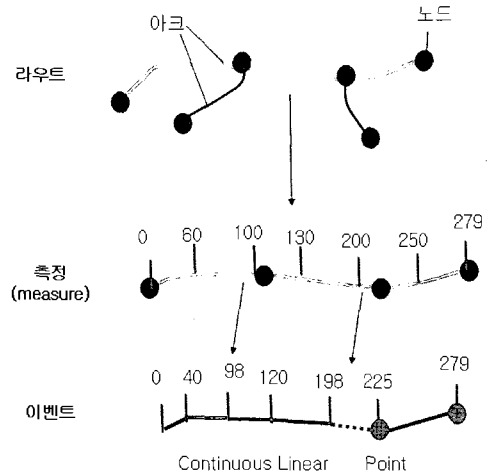
[그림 1] 동적 분할을 이용한 선형참조 모형

동적 분할은 기존 네트워크 자료구조를 변화시키지 않고 선형 참조 데이터(event)를 기반으로 기존의 링크 데이터를 섹션으로 구성하고, 섹션간의 포함관계로 도로 구간의 위상관계 정립과 속성자료를 연결시키는 모델링 방법이다(ESRI,2001). 따라서 이 모델의 적용을 통해 목적에 따라 네트워크의 여러 링크를 하나로 합치거나 또는 하나의 링크를 보다 세분화시켜 분석이 가능하게 된다.

## 2.2 동적 분할의 구성요소

[그림 2]에서와 같이 동적 분할은 라우트(Routes), 섹션(Sections)과 이벤트(Event)로 구성된다. 이벤트는 선형모델과 연관되어 발생한 일련의 사건이고, 섹션은 공

통 속성을 가진 연속된 아크를 의미하며, 라우트는 이 섹션들의 집합을 말한다. 아크와 노드로 선형 요소를 구성하고 그 위에 의미 있는 속성을 가진 아크와 노드를 연결하여 라우트를 만들어 구성한다.



[그림 2] 동적분할 구성

## 3. 동적 분할 기법의 설계와 구현

### 3.1 입력 데이터 설계

#### (1) 노드 데이터

노드는 도로상의 주요 교차점이나 선형의 변화 지점으로 포인트 데이터이다. 노드의 경우 아이디, 레벨, 종별, 인접 링크수로 구성되고, 노드의 레벨은 인접 링크의 최상위 레벨을 부여한다. 모델 적용을 위한 노드의 저장 구조와 종별코드는 아래 <표 1>, <표 2>와 같다.

<표 1> 노드 데이터 스키마

스키마 설계	대상 스키마	도로노드 데이터		
항목설명	항목명	Type	Byte	비 고
레이어 번호	Nod_LY	character	8	Not NULL
X좌표	Nod_X	double	4	Not NULL
Y좌표	Nod_Y	double	4	Not NULL
노드 ID	Nod_ID	character	8	Not NULL, PK
노드 레벨	Nod_LV	character	8	0 ~ 8
노드 종별	Nod_C	Tinyint	1	노드종별코드
인접 링크수	Nod_Adj	Tinyint	1	Not NULL

<표 2> 노드종별코드

코드	코드내역	코드	코드내역
1	교차점	6	U턴점
2	더미노드 종별	7	레벨 분할점
3	속성변환점	8	지도한계점
4	도로 종료점	9	로터리점
5	도엽경계점	10	P턴점

(2) 링크 데이터

링크는 아이디, 시점과 종점 노드 아이디, 레벨, 일방통행 유무, 차선 수, 속도, 도로등급과 노선번호로 구성된다. 노선번호는 지방도 이상의 모든 도로에 부여되고, 국도 중용구간에는 낮은 번호가 우선되어 부여된다. 링크 데이터에 대한 구조는 아래 <표 3>와 같다. 링크의 레벨은 도로 이동성과 접근성 등 기능에 따라 위계를 나누어 구성한 것으로 도로등급에 따른 레벨 부여 기준은 아래 <표 4>와 같다. 도로의 레벨은 0~5 레벨의 값을 입

력하고, 연결로의 경우 레벨은 고립도로를 포함시키지 않기 위해 본선의 레벨이 아닌 연결된 도로의 레벨을 입력하고 도로등급은 본선의 등급을 입력한다.

3.2 동적 분할 적용 : 멀티 링크 생성 알고리즘

(1) 멀티 링크(Multi Link) 개념

2장에서 언급했듯이 선형참조 모델에서는 동적 분할기법을 사용하여 개체의 부분을 다중 속성의 집합으로 연관시켜 표현한다.

<표 3> 링크 데이터 스키마

스키마 설계		대상 스키마	도로링크 데이터		
항목설명		항목명	Type	Byte	비 고
레이어 번호		Lk_LY	character	8	Not NULL
X좌표		Lk_V_XN1..N	double	4	Not NULL
Y좌표		Lk_V_YN1..N	double	4	Not NULL
도로 ID		Lk_ID	character	8	Not NULL, PK
시점 노드 ID		Lk_FN_ID	character	8	Not NULL
종점 노드 ID		Lk_TN_ID	character	8	Not NULL
도로레벨		Lk_LV	Tinyint	1	0 ~ 8
일방동행코드		Lk_OW	Tinyint	1	0 ~ 3
차선 수		Lk_LN	smallint	2	
속도 제한(상한)		Lk_SD_H	smallint	2	
도로 명		Lk_NAME	character	30	
노선번호	도로등급	Lk_N_L	Tinyint	1	0 ~ 15
	노선번호	Lk_N_NO	character	8	ex) 7번국도

<표 4> 도로등급과 레벨

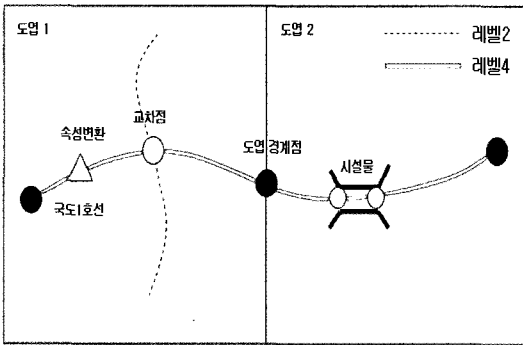
코드	도로등급	레벨	코드	도로등급	레벨
0	그 외 도로 선형	0	8	군도	3
1	국제고속국도	5	9	구도	3
1	국제간선도로	5	10	주요간선도	3
2	도시 고속도로	5	11	간선도	2
3	일반국도	4	12	일반도로	2
4	국가지원지방도	4	13	기타도로	1
5	지방도	4	14	세도로	0
6	특별/광역시도	4	15	페리항로	0
7	시도	3	19	기타도로2	0

이 모델의 적용을 통해 목적에 따라 네트워크의 여러 링크를 하나로 합치거나 또는 하나의 링크를 보다 세분화시켜 분석이 가능하게 된다. 즉, 멀티 링크는 하위

레벨의 여러 링크가 공통 속성을 가진 계층화된 상위 레벨 링크에 대응되도록 재구성한 것이다. (Takahiko Hamada, 2002)

일반적으로 교통망은 하위 레벨을 기준

으로 상위 레벨 도로를 계층적으로 구성한다. 상위 레벨은 하위 기준 레벨의 노드와 링크를 도엽단위로 병합하여 구성하게 되므로 하위 레벨의 복잡한 데이터 구조를 그대로 가지게 된다. 아래 [그림 3]과 같이 상위 레벨 링크는 하위 레벨의 링크를 단순 추출을 통해 구성되므로 속성변환, 시설물(지하, 고가, 교량), 도엽경계 노드 등으로 분할된 구조를 그대로 사용하게 된다.



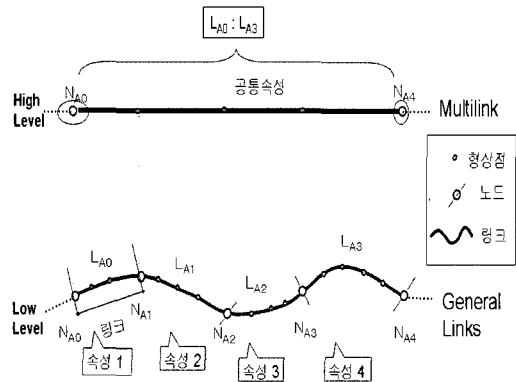
[그림 3] 분할된 상위레벨 링크 예

하위 레벨의 모델을 상위 레벨에 일률적으로 적용하는 것은 상세 분석에 적합하게 설계된 하위레벨 구조를 지역간 분석에 사용하게 되어 모형 적용면에서 적합하지 않다. 또한, 과도하게 분할된 노드와 링크로 인해 데이터베이스 규모가 커지고 유지관리를 어렵게 하여 궁극적으로 상위 레벨에 사용된 비효율적인 구조로 인해 공간 DB에서의 검색 성능을 저하시키게 된다. (Maarten Vermeij 2000) 이러한 문제를 해결하기위해 동적 분할기법을 적용하여 동일 속성을 가지는 도로를 통합한 후 상위레벨에서 사용될 수 있도록 멀티 링크 구조를 설계 할 수 있다.

## (2) 멀티링크 군집화

### (Multi Link Clustering)조건

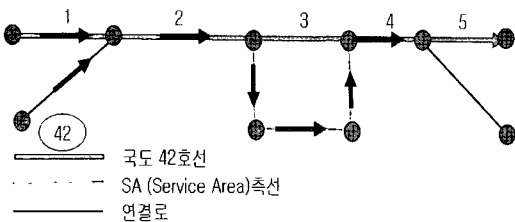
동적 분할을 이용하여 멀티 링크를 구성하기위해 최하위 레벨의 링크는 최소성과 유일성을 가지도록 식별번호(Absolute Link ID)를 부여한다. 이러한 최하위 링크는 도로데이터, 경로계산 데이터, 경로유도 데이터, 안내 데이터간의 상호 공통화에 사용될 수 있다. 공통된 속성을 가지는 기준 레벨 링크에 오름차순으로 링크 아이디(Differential Link ID)를 부여한다. 상위 레벨의 링크는 통합화된 하위 레벨의 시점 링크 ID와 종점 링크 ID로 표현하여 상·하위 레벨 간에 관계성을 정의한다.



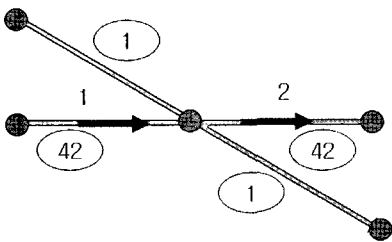
[그림 4] General Link와 Multi Link 구조  
(Teruo Mimori, 2003)

멀티링크 생성 시 링크 연결을 위한 기본 조건은 아래와 같다. 첫째, 아래 [그림 5]와 같이 상위 레벨 구성을 위한 링크의 공통 속성은 도로등급과 도로번호로서 노선이 정의된 연결성이 있는 도로를 대상으로 한다. 따라서 도로망의 주 골격을

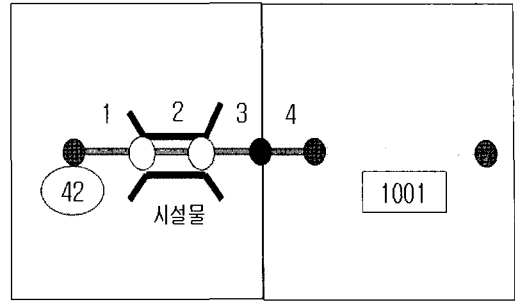
형성하여 지역 상호간 연결하는 주간선도로나 고속도로 또는 주간선도로에 연계되는 보조 간선도로를 대상으로 하고 통행거리가 짧은 집산도로는 제외한다. 둘째, [그림 6]에서와 같이 동일 도로등급의 링크 아이디가 연속된 링크를 연결하고, 동일 도로 등급이 교차 시에는 노선번호가 같은 링크를 연결한다. 또한, 동일한 도로 등급과 노선번호가 같은 링크가 다수일 경우에는 속도가 같은 링크를 연결한다. 셋째, [그림 7]과 같이 시설물 노드(터널, 교량, 고가, 지하차도)와 도엽 경계로 끊어진 링크는 인접 링크와 연결시킨다. 속성 변환점(속도, 차선수 등)은 연결시키지만 도로등급의 변화, 노선의 시·종점, 자동차 전용도로의 기·종점, 유료 도로의 기종점에서는 링크를 연결시키지 않는다. 마지막으로, 기본도로에 오버패스나 언더패스 되는 도로는 절단시키지 않는다.



[그림 5] 도로등급이 같은 링크 ID 연결



[그림 6] 노선번호 같은 링크 연결



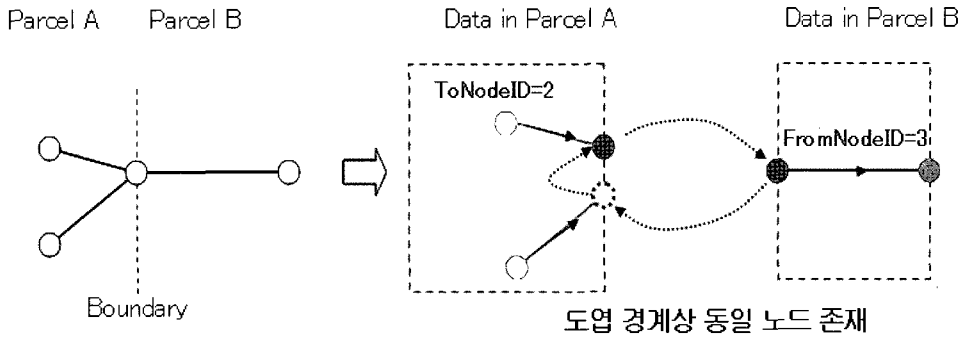
[그림 7] 기종점에서 링크 단절

### (3) 멀티 링크 알고리즘 구현

위의 군집화 조건에 따른 멀티 링크 생성 알고리즘은 전처리, 링크 아이디부여, 멀티 링크 추출 3단계로 구성된다. 전처리 단계에서는 기준 레벨 링크의 공통 속성을 추출하기 위해 단위 도엽들의 모든 링크와 노드를 병합시킨다. 단위 도엽 내에 노드 아이디는 도엽 간 중복이 존재할 수 있으므로 병합된 노드를 대상으로 노드 아이디를 새로 부여한다. 링크와 노드 간에 인접성과 연결성을 보장하기 위해 링크의 시점과 종점 노드 아이디를 부여하여 위상을 정립한다.

그 다음 단계에서는 기준 레벨에 대한 링크 아이디를 부여한다. 이를 위해 최상위 레벨에서부터 노드와 링크를 대상으로 도로등급과 도로번호에 따라 정렬된 링크의 집합을 생성한다. 각 링크마다 교차하는 노드에 대해 이벤트 집합을 생성한 후 인접하는 링크 수를 부여한다. 인접 링크 수가 1이고 이벤트 리스트 아이디와 링크의 시점 노드가 같은 링크에서부터 링크 아이디를 순차적으로 마지막 링크까지 부여한다(Case 1 : General Node).





[그림 8] 도입 경계에서의 멀티링크 연결 (Case 2 : Boundary Node)

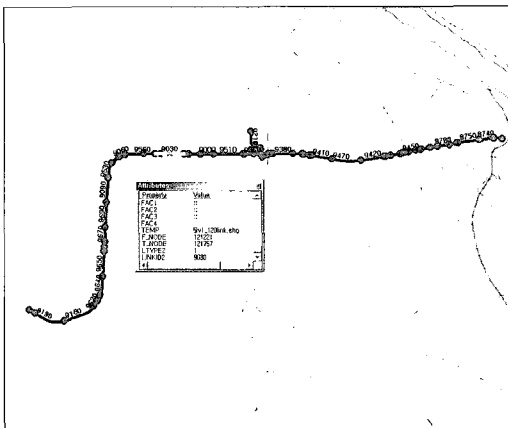
섹션들을 결정하는 동안 연결되는 노드나 링크가 여러 개인 경우가 존재한다. 노드가 2개 이상 중첩되는 경우에는 도입의 경계에서 동일 위치에 있는 2개의 경계 노드가 경합되는 경우로 Link.current.ToNodeID와 Link.Next.FromNodeID가 다르지만 중첩된 노드의 ID를 확인한 후 멀티링크로 연결 되도록 처리한다(Case 2 : Boundary Node).

또한 링크가 3개 이상 중첩되는 경우

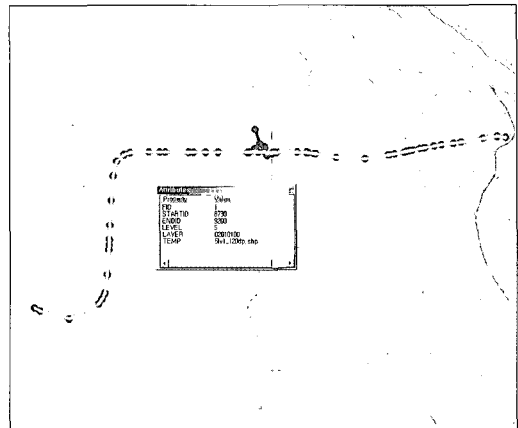
(고속도로, 일반국도 본선과 진출과 진입을 위한 IC나 JC가 교차)는 속도가 높은 링크로 ID를 부여하여 연결 되도록 처리한다(Case 3 : Connection Link).

마지막 단계는 시점과 종점 링크 아이디를 공통속성으로, 이벤트 리스트 좌표를 형상점으로 하는 멀티 링크를 작성한다.

이에 대한 Pseudo 알고리즘과 적용된 예(경인고속도로)는 [그림 9], [그림 10], <표 5>와 같다.



[그림 9] 링크 아이디 부여



[그림 10] 멀티 링크 생성

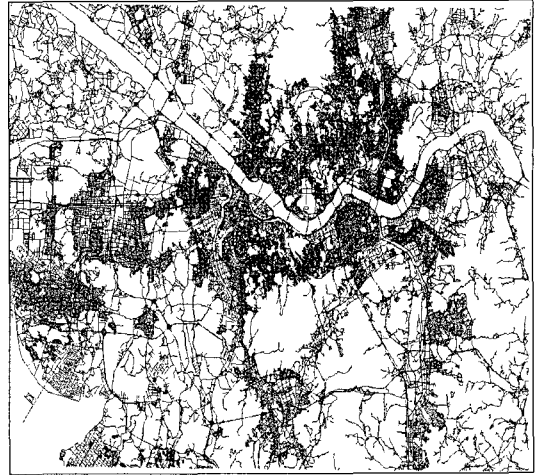
<표 5> 멀티 링크 생성 알고리즘

ALGORITHM	
Pre-Processing	<pre> // Step 1 : Node-Link Topology Define MergeNode = NodeThemeList.Merge For MergeNode.Count iteration     MergeNode.ID += 1 } MergeLink = LinkThemeList.Merge For MergeLink.Count iteration     MergeLink.FromNodeID = MergeLink.GetNodeID(Link.First)     MergeLink.ToNodeID = MergeLink.GetNodeID(Link.Last) }                     </pre>
Multi Link Clustering Process	<pre> // Step 2 : Multi Link ID Define MergeNode, MergeLink Open Theme For i iteration{ // From Upper Level     LinkSortList.Make // Linktype &amp; RoadNum Sort     For j iteration { // LinkSortList         EventList = LinkSortList(j).Intersect(MergeNode)         EventList.adjLink = EventList.count(LinkSortList(j))         cnt = 0         If (EventList.adjLink == 1 &amp;&amp; EventList.ID == Link.FromNodeID){             Link.current.ID = cnt             Do { // Case 1 (General Node) If (Link.current.ToNodeID == Link.Next.FromNodeID)     Link.Next.ID = cnt + 1 // Case 2 (Boundary Node) ElseIf (EventList.Count(Link.current.ToNodeID.GetXY) == 2)     Link.Next.ID = cnt + 1 // Case 3 (Connection Link) ElseIf (EventList.adjLink &gt;= 3)     Link = CompareSpeed.Get(Link.current)     Link.Next.ID = cnt + 1             } while(EventList.adjLink == 1) // Step 3 : Multi Link Generation Routes.Make(EventList.GetXY, StartID, OffsetEndID)         } // if     } // For j } // For i                     </pre>

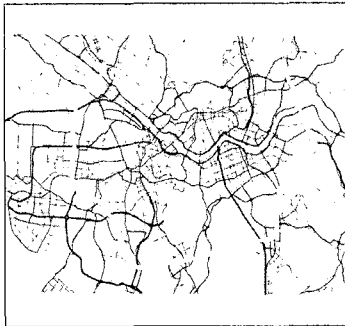
### 3.3 알고리즘 적용

#### (1) 멀티 링크 생성 결과

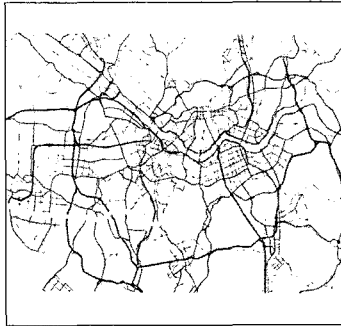
설계된 알고리즘을 아래 [그림 11]과 같이 인천지역을 포함한 서울권역 55,000km × 36000 km에 해당하는 도로망을 대상으로 적용하였다. 아래 [그림 12]에서 (a)~(f)는 제시된 알고리즘을 적용하여 추출된 멀티 링크를 굵은 선으로 표시하였다. 또한 <표 6>은 레벨별로 추출된 멀티 링크 중 도로연장이 10km이상 되는 링크의 목록이다. 레벨별 추출 결과는 아래 [그림 13] 과 같다.



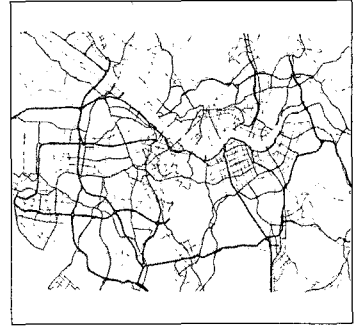
[그림 11] 대상 지역 (인천·서울권역)



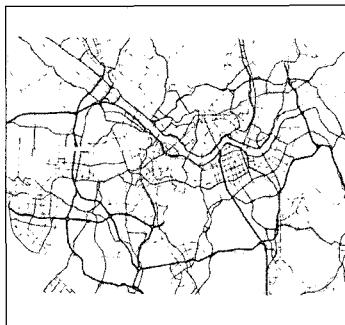
(a)서울외곽순환고속국도



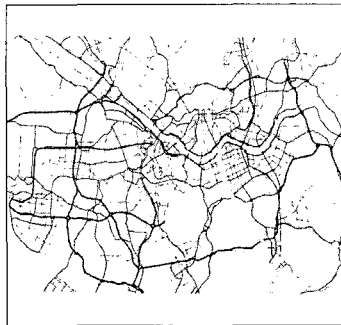
(c)제2경인고속국도



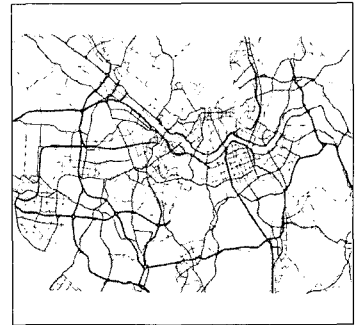
(b)강변북로



(d)경인고속국도



(e)경부고속국도



(f)내부순환로

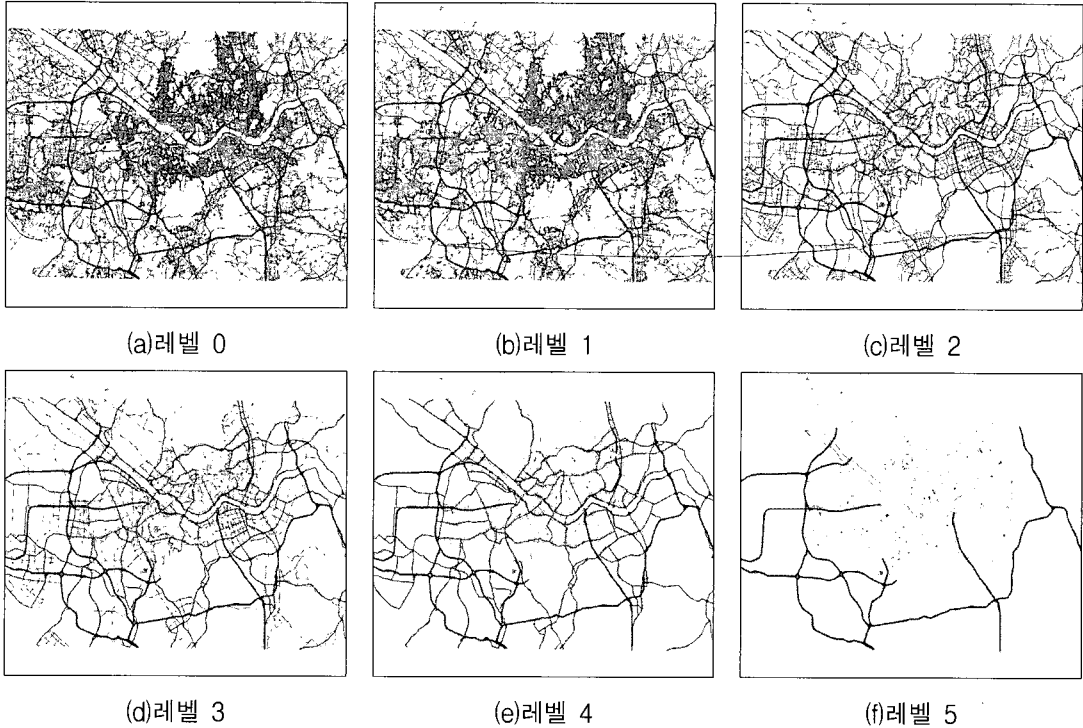
[그림 12] 생성된 멀티 링크

<표 6> 레벨별 멀티 링크 리스트(10km이상)

(단위 : m)

레벨	도로명	등급	번호	연장	레벨	도로명	등급	번호	연장
5	서울외곽순환고속국도	1	100	89497	4	분당장지간도시고속화도로	2	61	14241.1
5	제2경인고속국도	1	110	52895.5	4	북부간선도로	2	20	13823.5
5	영동고속국도	1	50	39612	3	397번지방도	5	397	13400.7
4	강변북로	2	70	34056.1	4	39번국도	3	39	12932
5	경인고속국도	1	120	28556.5	4	동1로	3	3	12798.7
5	경부고속국도	1	1	23031.9	4	분당내곡간도시고속화도로	2	51	12164.9
4	올림픽대로	2	88	21735.6	3	동부간선도로	2	61	12084.7
4	내부순환로	2	30	21498.3	3	23번지방도	4	23	11850.7
4	남부순환로	6	92	21131.6	4	성남광주간산업도로	3	3	11779.1
5	인천국제공항고속국도	1	130	19736.9	4	43번국도	3	43	11667.9
4	399번지방도	5	399	19475.7	4	수인산업도로	3	42	11430.1
4	경수산업도로	3	1	16740.8	4	자유로	2	23	11284.2
4	48번국도	3	48	16615.8	4	행주대로	3	39	11059.3
3	57번지방도	4	57	15938.6	4	과천고색간고속화도로	2	312	10947.3
5	서해안고속국도	1	15	15735.3	4	서부간선도로	2	1	10910.1
3	강동대로	6	60	15561.5	3	389번지방도	5	389	10818.3
4	통일로	3	1	15248.3	3	테헤란로	6	90	10479.6
3	78번지방도	4	78	15193.2	3	시흥대로	6	21	10107.5

(2) 레벨별 추출 결과



[그림 13] 레벨별 도로 DB

3.4 성능 평가

본 연구에서 제시된 멀티링크 생성 알고리즘과 네트워크 계층 모델에 대한 효율성 검증을 위해 실험을 실시하였다. 성능분석에서는 본 연구에서 제시된 멀티링크를 적용한 어플리케이션이 기존 노드-링크 체계를 적용한 일반 모델과의 실행 성능이 어떠한 차이를 보이는지 평가하고자 하였다. 구체적으로 성능평가 실험은 멀티 링크와 일반 링크를 사용하여 레벨별 검색에 대한 반환 개체 수 (레코드 수, 형상점수)와 응답시간(질의처리, 랜더링)에

대한 측정 비교로 이루어졌다. 질의처리 실험을 위해 공간 데이터는 RDBMS (SQLite 3.3.4)에 저장되고 Microsoft Visual C++를 사용하여 그래픽 루틴을 처리하였다. 연산 평가를 위한 공간질의는 각 레벨별로 수행되었으며, 연산의 수행시간은 밀리세컨드(1/1000초) 단위로 측정하였다. 이 실험에서 주어진 위치와 각 레벨에 따른 그리드 영역(공간 인덱스) 조건은 두 모델에 동일하게 처리되었다. 레벨별 성능평가를 위한 공간질의를 <표 7>과 같으며, 알고리즘이 적용된 멀티링크가 저장된 테이블들을 DP로, 알고리즘이 적용되

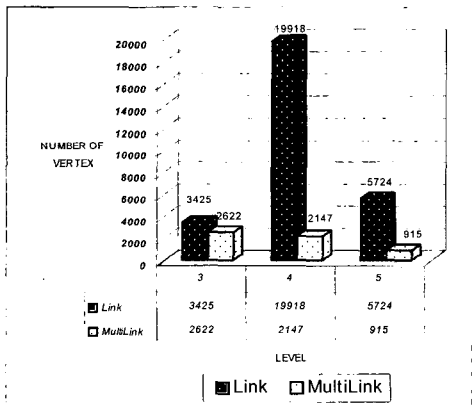
<표 7> 성능평가를 위한 공간질의 (Level 3)

General Link	Multi Link
<pre>// Link select RdFTable.Shape , RdClass.RGBColor, from RdSTable, RdFTable,RdClass where RdSTable.RdLevel &gt;= 3 and ((RdSTable.GMinX&lt;=14 AND RdSTable.RdSTable.GMinY&lt;=26) and (RdSTable.GMaxX &gt;=13 AND RdSTable.GMaxY=25)) and ((RdSTable.MinX &lt;= 292403 and RdSTable.MinY &lt;= 544052 ) and (RdSTable.MaxX &gt;= 274041 and RdSTable.MaxY)= 533384 )) and RdSTable.LinkID = RdFTable.LinkID and RdFTable.ClassCode = RdClass.ClassCode</pre>	<pre>//MultiLink select DpFTable.Shape,RdClass.RGBColor from DpSTable, DpFTable, RdClass where DpSTable.RdLevel &gt;= 3 and ((DpSTable.GMinX&lt;=14 AND DpSTable.DpSTable.GMinY&lt;=26) and (DpSTable.GMaxX &gt;=13 AND DpSTable.GMaxY=25)) and ((DpSTable.MinX &lt;= 292403 AND DpSTable.MinY &lt;= 544052) AND (DpSTable.MaxX &gt;= 274041 AND DpSTable.MaxY = 533384)) AND DpSTable.LinkID = DpFTable.LinkID AND DpFTable.ClassCode = RdClass.ClassCode</pre>

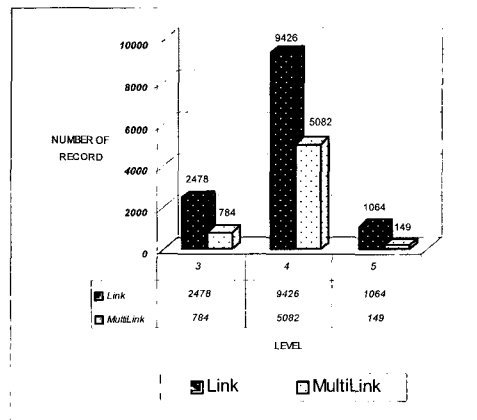
지 않은 일반 링크와 관련된 테이블들을 RD로 정의하였다.

[그림 14]에서 공간질의 결과 개체수와 반환된 개체의 형상점수를 비교해 보면 멀티 링크 형태로 작성된 데이터의 레코드와 형상점 수가 상대적으로 감소되었음을 알 수 있다. 이는 3레벨 이상의 링크들은 연결성을 가진 주간선도로와 고속도로 등이 해당되므로, 동적 분할에 의한 링크병합과 그에 따른 일반화 과정에 영

향을 많음을 알 수 있다. <표 7>에 제시된 질의 처리에 대한 수행시간은 [그림 15]와 같다. 기존 노드·링크 모델은 상위 레벨로 갈수록 다수의 링크와 형상점의 표현으로 질의 처리시간과 렌더링 시간이 현저히 증가되고 있다. 멀티 링크의 경우는 평균 질의처리 시간이 36 밀리세컨드로 기존 모델에 비해 30배정도 단축되었으며, 평균 렌더링 시간은 31 밀리세컨드로 64배 단축되었다. 두 개의 수행시간

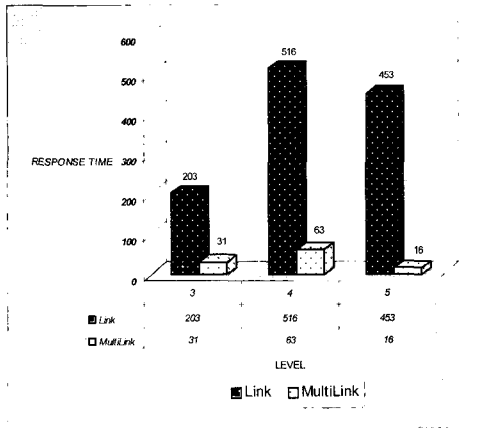


(b) 형상점수

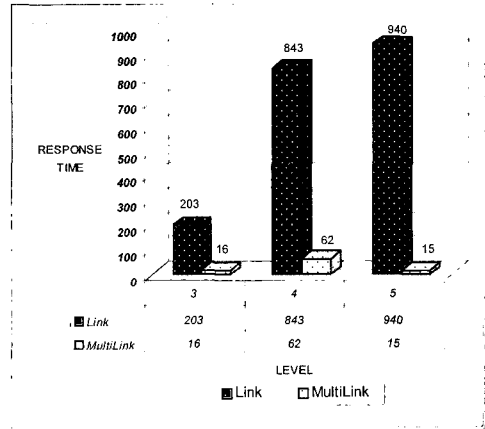


(a) 레코드수

[그림 14] 검색 개체수 결과



(a) 질의처리 시간(millisecond)



(b) 랜더링 시간(millisecond)

[그림 15] 수행시간 결과

측면에서 모두 기존 모델에 비해 수행시간이 단축되어 검색과 디스플레이에 효율적이고 우수한 성능임을 알 수 있다.

#### 4. 결 론

본 연구에서는 동적 분할 기법을 이용한 네트워크 모델의 설계와 구축을 통해 기존 도로망 모델의 문제점과 구축상의 한계점을 보완하고, 네트워크의 검색과 표현에 효율적인 계층 모델을 구현하고자 하였다. 이를 위해 첫째, 모델적용을 위한 노드·링크 체계와 레벨을 정의하고 둘째, 기존 노드·링크 모델의 문제점을 파악하여 동적 분할을 적용한 개선된 모델을 설계하였으며, 셋째, 제시된 알고리즘을 인천·서울권역 도로망에 적용하여 멀티링크로 구성된 계층 도로망을 구축하고, 마지막으로 분석실험과 성능분석을 통해 모델을 평가하였다.

이러한 동적 분할 기법이 적용된 네트워크 모델은 성능평가 결과에서 나타난 바와 같이 기존 모델에 비해 데이터 용량 측면에서 경량화되어 모델의 단순성을 지원하고, 도로 연결성과 근접성을 고려한 저장구조로 정의되어 검색의 효율성을 보장하게 된다. 또한, 모델 생성을 위한 자동화된 변환 과정은 모델의 일관성과 표준화된 절차를 유지시켜준다. 현재 레벨의 계층적인 표현과 개체사이의 관계성을 지원하는 멀티링크는 다중표현(Multiple Representation)과 다중 스케일(Multi-Scale) 측면에서 GIS가 지닌 네트워크 공간 모델링 기능을 대폭 보완할 수 있을 것으로 기대된다.

#### 참고문헌

건설교통부, ITS Korea, 2003, 표준 위치참조 표준/기술보고서 지능형교통시스템

- 건설교통부, 2005 지능형교통체계 표준 노드링 크 구축 · 운영지침
- 한국정보통신기술협회, 2003, 기존 GIS DB를 활용한 모바일 서비스용 GIS DB 구축 지침
- 한국교통연구원 국가교통DB센터, 2004. 국가교통DB구축사업 교통시설물조사 · 교통주제도 및 교통분석용 네트워크 구축
- [http://www.dot.state.ia.us/gis/info\\_page.htm](http://www.dot.state.ia.us/gis/info_page.htm)
- Bo Huang, 2002, An ODMG-based object model for dynamic segmentation, IEEE ITS International Conference
- Dimitis Papadias, Jun Zhang, 2003, Query Processing In Spatial Network Database
- ESRI White Paper, 2001, Linear Referencing and Dynamic Segmentation in ArcGIS
- Koncz and Adams, 2002, A Data Model for Multi-dimensional Transportation Location Referencing Systems
- Keechoo Choi, 2000, Development of a transit network from a street map database with spatial analysis and dynamic segmentation
- Maarten Vermeij, Peter Van Oosterom, 2000, Storing And Using Multi-Scale Topological Data Efficiently In A Client-Server DBMS Environment I
- Sabine Timpf, Thomas Devogele, 1997, New Tools For Multiple Representations
- Shashi Shekhar, 1997, A Connectivity-Clustered Access Method for Network and Network Computations
- Stephen M. Perone, 1997, Integrating Transportation Modeling Networks Using Dynamic Segmentation
- Takahiko Hamada, 2002, Kiwi Format and Telematics
- Teruo Mimori, 2003, How to Access to the Kiwi Data