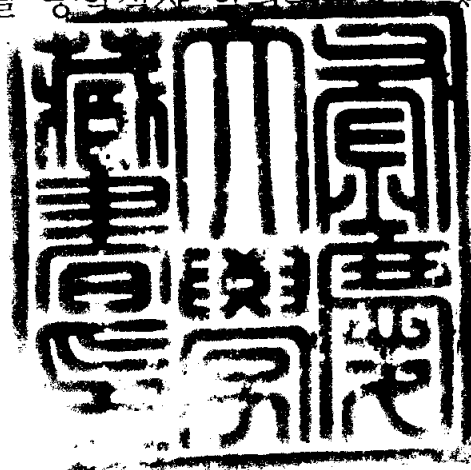


공학석사 학위논문

GIS를 이용한 하천관리 시스템 개발

지도교수 김 상 용

이 논문을 공학석사 학위논문으로 제출함.



2004년 8월

부경대학교 산업대학원

토 목 공 학 과

김 재 호

이 논문을 김재호의 공학석사
학위논문으로 인준함

2004년 6월

주 심 농공학박사 이 영 대



위 원 공학박사 이 종 출



위 원 공학박사 김 상 용



목 차

표차례	vi
그림차례	vii
Abstract	ix
제 1 장 서론	1
1.1 연구배경 및 목적	1
1.2 연구동향	2
1.3 연구방법	4
제 2 장 이론적고찰	6
2.1 수면곡선의 기본이론(부등류 해석기법)	6
2.1.1 기본방정식	6
2.1.2 단면분할	7
2.1.3 유속계수	8
2.1.4 마찰손실계산	8
2.1.5 한계수심	9
2.2 GIS와 자료처리기법	10
2.2.1 지리정보시스템의 자료	10
2.2.2 GIS의 발달 과정과 현황	11
2.2.3 GIS의 자료처리기법	14
제 3 장 적용대상유역의 수문해석	16
3.1 적용대상유역의 자료분석	16
3.1.1 하천 단면자료의 분석	18
3.2 유출모형의 적용 및 선정	20

3.2.1 유출모형의 적용	20
3.2.2 유출모형의 선정	21
3.3 수리학적 모형의 분석 및 선정	24
3.3.1 하도 홍수위 분석	24
3.3.2 기점수위의 결정	24
제 4 장 하천관리 시스템 구축	26
4.1 단면자료 구축	26
4.2 HEC-RAS 연계 프로그램 구축	29
4.2.1 유출모형선정	29
4.2.2 HEC-RAS Input Data 변환 Program	30
4.2.3 HEC-RAS 실행	30
4.3 GIS와 연계한 하천관리시스템 구축	31
4.3.1 하천단면현황 설명	34
4.3.2 래스터형 데이터와 벡터형 데이터	36
4.3.3 하천시설물	37
제 5 장 결과 및 고찰	43
5.1 하천관리시스템 구축	43
5.2 하천시설물관리시스템을 통한 수공구조물 관리	43
5.3 시스템구축을 통한 자연재해예방	43
제 6 장 결론	44
참고문헌	45
부록	48

표 차례

표 1	지점별 빈도별 홍수량 산정	23
표 2	하천단면현황에 대한 기능설명	34
표 3	하천 시설물 장애에 대한 기능설명	38

그림차례

그림 1 연구방법	5
그림 2 대상유역의 토지이용도	16
그림 3 대상유역의 토양도	17
그림 4 수영강 유역도	18
그림 5 회동수원지 상류의 종 · 횡단면도	19
그림 6 회동수원지 하류의 종 · 횡단면도	19
그림 7 대상유역의 산정 지점	22
그림 8 대상유역의 등시간도	22
그림 9 수영강 회동댐 상류 홍수위	25
그림 10 임의 지점(River Station)에 대한 데이터 분석창	26
그림 11 임의 지점(River Station)의 횡단면도	27
그림 12 해당구간의 종단면도	27
그림 13 HEC-RAS 연계 프로그램 개발	29
그림 14 HEC-RAS 연계 프로그램 흐름도	30
그림 15 기능구성도	31
그림 16 하천관리시스템 중 ‘단면현황’실행시 화면	32
그림 17 지리정보시스템(GIS)과 단면도 현황	32
그림 18 하천유역 내 임의 측점에서의 하천단면도	33
그림 19 하천종단도 상에 나타나는 측점의 위치	33
그림 20 지정된 임의의 측점에 대한 각종 데이터	33
그림 21 래스터 및 벡터자료의 하천평면도	37
그림 22 하천시설물현황 창외 실행시 초기화면	39
그림 23 시설물명 선택시 시설물현황 화면	39
그림 24 View Index에서 구역‘A’의 확대화면 실행	40
그림 25 시설물 선택시 나타나는 시설물 제원	40

그림 26 시설물 선택시 나타나는 단면도	41
(5년빈도 홍수위 적용(EL. 37.74m))	
그림 27 시설물 선택시 나타나는 단면도	41
(100년빈도 홍수위 적용(EL. 38.78m))	

Development of River Management System Using GIS

Kim, Jae-Ho

*Department of Civil Engineering, Graduate School of Industry
Pukyong National University*

Abstract

As a result of development of modern civilization and effect of air pollution, it is difficult to predict flood disaster. And many specialists are looking for countermeasure of this problem, connect with geographic information system which applied hydrological theory.

In this way, geographic information system(GIS) is risen by new possibility in water resources field. And is giving many helps to do inundation analysis as a tool that can analyze space data. Also, relation of GIS and hydrology is helping to work of design and examine of hydrological model.

In this research, we connected result of runoff and water level by rainfall, with geographic information system. For that reason, we achieved research to take river management more efficiently. By using specificated runoff volume of target area, we calculated water-level of inundation after considering various topography factors, through link with Hec-Ras.

제 1 장 서론

1.1 연구배경 및 목적

현대 문명의 발달로 인한 지구온난화 및 대기오염의 영향으로 예측하기 어려운 홍수피해 등에 대하여 현재 많은 전문가들은 수리·수문학적 개념을 도입한 지리정보시스템과 연계하여 대책을 찾고 있다. 이처럼 지리정보시스템(geographic information system; GIS)은 수자원 분야에서 새로운 가능성으로 대두되고 있으며, 공간자료를 분석할 수 있는 도구로서 침수분석을 하는데 많은 도움을 주고 있다. 또한 GIS와 수문학의 연관성은 수문 모형의 설계 및 검정 등의 작업에도 도움을 주고 있다.

지리정보시스템이 수문학에 응용 및 적용된 것은 1990년도라 할 수 있으며, 2000년 이후에 더욱 활발하게 연구되고 있다. 최근 2002년도에 들어 침수피해현상을 실시간 3차원적으로 해석하고자 노력하고 있다. 이와 같이 지리정보시스템과 수문학의 연계성은 이상기후 등으로 발생하는 집중호우에 의한 홍수피해를 미리 예측할 수 있으며, 인명피해 및 재산피해를 줄이는데 큰 역할을 담당할 것으로 본다.

본 연구에서는 강우에 의한 유출과 하천의 수위에 대한 결과 값을 지리정보시스템과 연계함으로써 보다 효율적인 하천관리를 하기 위해 연구를 수행하였다. 대상지역의 특정 유출량을 이용하여 각종 지형인자를 고려한 후 Hec-Ras와의 연계를 통해 홍수위를 산정하고, 산정된 홍수위를 한눈에 이해할 수 있도록 Visual C++ 및 MapObject를 이용한 하천관리 시스템을 구축하였다. 앞으로 계속 연구될 것으로 예상되는 GIS와 수문학의 연계는 실시간 제내지 및 제외지의 홍수위를 예측하고 보여줌으로써 피해를 예방할 수 있다고 본다.

1.2 연구동향

급격한 도시화로 인하여 불투수층이 크게 발생하고 지하배설물 등에 의하여 도시유역에서의 수문현상은 대단히 복잡하게 되어가고 있다. 도시유역에서의 강우유출에 따른 일련의 문제를 공학적인 접근으로 문제의 제거와 해결을 시도하고자 하는대는 다음과 같이 크게 4단계로 나눌 수 있다. 첫째는 1850년대에 시작되었는데 주로 관경결정이 문제가 되었고, 둘째 단계는 얼마나 많은 물을 어떻게 배제하느냐에 관한 문제였고, 셋째 단계는 배수의 양적인 문제뿐만 아니라 수질에 관한 문제도 고려하는 방법으로써, 이는 특히 1960년도 후반부터 연구가 시작되었다. 넷째 단계로는 배수로의 최적설계, 유지관리 및 침수위험 등에 관심을 두어 활발한 연구결과가 발표되었다.

19C 후반에 Emil Kückling(1889)는 우수배수조직의 해석 및 계획과 관련하여 합리식을 발표하였다. Gregory와 Arnold(1932)는 유역의 형상과 경사, 배수망의 형태, 개수로의 흐름요소 등을 고려하여 합리식을 개정발표 하였으며, Izzard(1946)는 경사면에서 부정류상태의 지표흐름을 모의시키는 방법을 개발하였다.

Tholin과 Keifer(1960)에 의해 연구 발표된 Chicago hydrograph method에서는 설계유량으로부터 손실량을 고려하였고, 포장, 초지, 지붕위에서의 지표흐름을 Izzard의 방법에 따라 계산하였으며, 배수망을 통해 유역 출구까지 흐르는 유량은 지체시간법을 이용하여 추적하였다.

ILLUDAS(Illinois Urban Drainage Area Simulator)는 RRL법을 도시유역에서 설계 및 유출량을 모의할 수 있도록 Stall et al(1974)이 개발한 모형이며, 시간면적곡선으로 구분유역 유입수문곡선을 모의하고, 관저에서는 유입수문곡선을 단지 수평이동하는 지체시간법과 음함수 및 양함수법을 이용하여 유출수문곡선을 구하였다. 김상용(1991)외 4명은 도시소유역 개발에 따른 유출현상의 연구를 통하여 합리식이나 단위도법은 침투유출

량을 결정하기 쉬운 점이 있으나, 수문순환과정에서 서로 복잡하게 상호작용하는 중요인자들의 역할을 제대로 표현하기는 어려운 점을 보완하여 유출모형을 수정보완하여 개발한 바 있다. 강태호(1999)와 2명은 도시하천에서의 강우, 유출 및 수질예측(I),(II)의 연구에서 시강우제열 모의발생모형(ARF-ARMA)에 의해 발생된 강우자료를 도시유출 및 수질모형인 SWMM에 적용하여 도시하천에서의 유출량과 오염부하량을 예측하는 ARCSWMM기법을 개발하였다. 백승권(2000)은 도시유출모형의 매개변수에 따른 침투유출량의 변화에 관한 연구를 하였으며, 한영만(2001)은 도시유출모형을 이용한 도시화 구역의 유출해석을 수행하였다. 김충수(2002)는 도시유출 모의모형의 매개변수를 추정하는 연구논문을 작성한 바 있다. 함창학(1996)은 GIS를 이용한 수문지형정보 추출에 관한 기초적 연구를 수행함으로써, GIS와 수문학의 연계성을 보여주었다. 차성렬(1994)은 지형공간정보체계를 이용한 도시기반시설 관리에 관한 연구를 통하여 상수도 등의 시설물 관리 및 유지할 수 있는 지형공간정보체계 방안을 제시하였다. 김계현(2001)와 1명은 GIS 기반의 하수도 관망해석기법 개발에 관한 연구를 통하여 하수관망시설에 대한 관망해석을 실시하여 배수지역내 적정 하수배출량을 산정화하고, GIS의 그래픽기능을 이용한 시각화 효과를 나타내는 시스템을 개발하였다. 유환희(2002)와 2명은 GIS를 이용하여 도시지형과 하수관망도를 데이터베이스로 구축하고 홍수의 지표 유출특성과 관로 유출을 분석함으로써 예상강우량 및 지속시간에 따라 피해지역을 예측할 수 있는 홍수관리시스템을 구축하였다. 김철(2003)와 1명은 홍수기록이 부족한 대규모의 넓은 지역에 대해 홍수에 의한 침수가능성이 있는 취약지역의 범위를 개략적으로 예측할 목적으로 수치고도모형(DEM), 토지피복도 등의 지형학적 자료와 빈도별 홍수위 자료에 대해 GIS를 사용하여 몇 가지의 분석방법을 개발하였다.

1.3 연구방법

최근에 들어서 지리정보시스템은 수문학분야와 많은 연관성을 가지며 활용되고 있다. 특히, 실제 지형을 표현하고, 수치표고모형(DEM)을 접목하여 수문분석을 하는데 큰 역할을 담당해주고 있다. 특히, 강우-유출관계의 해석을 위한 공간자료를 더욱 명확하고 체계적으로 획득할 수 있는 방법을 제시해 주고 있다.

과거에는 하천유역의 수문학적 공간자료 및 정보를 추출해 내기 위해서 방대한 자료들의 획득과정에 많은 시간과 인력의 소요가 필요했다. 그러나 최근에는 하천유역과 같은 넓은 지역에 대해서 보다 효율적인 자료를 획득하기 위한 방법으로서 컴퓨터 하드웨어의 발달과 더불어 GIS 기법들이 제시되었다.

지리정보체계를 이용한 종합적인 공간처리 기술을 도입하여 이미 공간 분석까지 수행할 수 있게 되었다. 본 연구에서는 자료를 해석 및 분석하고 처리하는 과정과 함께 지형인자를 통하여 강우-유출 모형을 수행하였다. 이때 얻어진 유출량과 기점수위를 HEC-RAS 모형과 연계하여 최종 홍수위를 계산하였다. HEC-RAS 모형은 수문학분야의 실무에서 이미 검증된 모형으로써, 본 시스템에서는 HEC-RAS 모형의 입력데이터를 강우-유출 모형에서 계산한 값을 자동으로 작성하게 하였으며, 시스템 사용자가 HEC-RAS 모형을 실행함으로써 최종 홍수위가 결정되는 방법으로 모형을 연계시켰다. 또한 HEC-RAS 모형에서 계산된 결과 값(sy.rcp)을 받아들여 본 시스템에서 GIS와 연계하여 화면 출력되는 방식으로 결과 값들을 나타내었다.

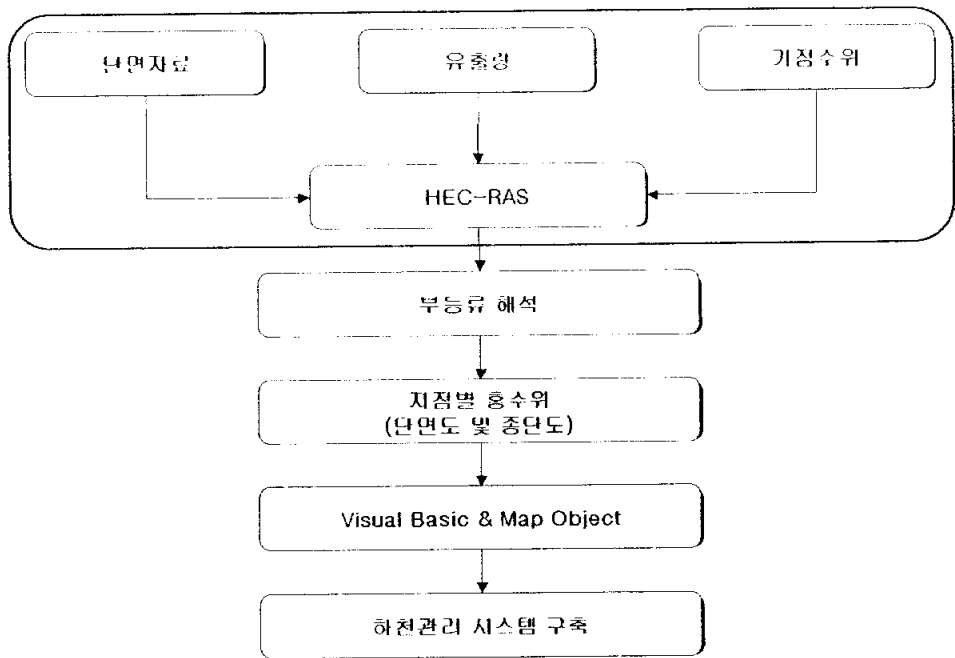


그림 1 연구방법

제 2 장 이론적고찰

2.1 수면곡선의 기본이론(부등류 해석기법)

수면곡선 계산을 위한 기본방정식, 통수능과 유속분포의 결정하기 위한 횡단면 분할, 마찰손실결정, 기초 방정식 해석을 위한 반복절차 그리고 한 계수심 결정등에 대하여 기술한다.

2.1.1 기본방정식

하천에서의 부등류 해석시 각 단면에서의 미지의 수위를 계산하기 위해서 다음의 두 방정식을 표준 축차법(standard step method)에 의해서 반복적으로 계산한다.

$$WS_2 + \frac{\alpha_2 V_2^2}{2g} = WS_1 + \frac{\alpha_1 V_1^2}{2g} + he \quad (2.1)$$

$$he = LS_f + C \left| \frac{\alpha_2 V_2^2}{2g} - \frac{\alpha_1 V_1^2}{2g} \right| \quad (2.2)$$

여기서, WS_1 , WS_2 는 구간 양단에서의 수위, V_1 , V_2 는 구간 양단에서의 평균유속, α_1 , α_2 는 구간 양단에서의 흐름의 유속계수, g 는 중력가속도, he 는 에너지손실계수, L 은 유량 가중 구간길이, S_f 는 구간에서의 대표 마찰경사 값, C 는 단면확대 및 축소에 의한 손실계수, 첨자 1, 2는 각각 하류단과 상류단을 나타낸다.

$$L = \frac{L_{lob}Q_{lob} + L_{ch}Q_{ch} + L_{rob}Q_{rob}}{Q_{lob} + Q_{ch} + Q_{rob}} \quad (2.3)$$

여기서, L_{lob} , L_{ch} , L_{rob} 는 각각 좌측 제방, 본수로, 우측 제방에서의 흐름에 대한 구간길이, Q_{lob} , Q_{ch} , Q_{rob} 는 각각 좌측 제방, 본수로, 우측 제방에 대한 구간 양단에서의 산술평균 유량을 나타낸다.

2.1.2 단면분할

임의 단면에서의 총통수능과 유속계수를 결정하기 위해서는 흐름을 유속분포가 균일한 단면요소로 분할할 필요가 있다. 각각의 분할단면에서의 통수능은 아래 식을 이용하여 계산한다,

$$k = \frac{1.486}{n} \alpha r^{2/3} \quad (2.4)$$

여기서, k 는 개별분할영역의 통수능, n 은 개별분할영역의 Manning의 계수, α 는 개별분할영역의 유속단면적, r 은 개별분할영역의 동수반경(면적/윤변)을 나타낸다.

수로에서는 수로단면내에서 조도계수가 변화할때를 제외하고는 흐름을 분할하지 않는다. HEC-2는 횡단면의 수로부분내에서 분할에 대한 적용성이 검정 및 수정되어 있으며, 분할이 부적합한 경우에는 전체수로에 대한 합성조도계수를 사용한다. 이때 합성조도계수를 사용할 수 있는지에 대한 기준은 수로의 측면경사가 5H:1V보다 가파르고 단면은 분할되었는가 하는 것이며, 합성조도계수는 합성조도계수 산정공식에 의해서 계산한다.

$$n_c = \left[\sum_{i=1}^N \frac{(P_i n_i^{1.5})}{P} \right]^{2/3} \quad (2.5)$$

여기서, n_c 는 합성 또는 등가조도계수, P 는 단면의 윤변, P_i 는 가상분할단면 i 의 윤변, n_i 는 가상분할단면 i 의 조도계수를 나타낸다.

2.1.3 유속계수

유속계수 α 는 3개의 흐름요소 (좌측, 우측제방, 수로)에 대한 각각의 통수능을 이용하여 다음 식으로 계산할 수 있다. 이것은 속도수두를 전체 적으로 사용하기 위해서 설정한다.

$$\alpha = \frac{(A_t)^2 \left[\frac{(K_{lob})^3}{(A_{lob})^2} + \frac{(K_{ch})^3}{(A_{ch})^2} + \frac{(K_{rob})^3}{(A_{rob})^2} \right]}{(K_t)^3} \quad (2.6)$$

여기서, A_t 는 총 유수단면적, A_{lob} , A_{ch} , A_{rob} 는 각각 좌측제방, 수로, 우측제방의 유수단면적, K_t 는 총통수능, K_{lob} , K_{ch} , K_{rob} 는 각각 좌측 제방, 수로, 우측제방의 통수능을 말한다.

2.1.4 마찰손실계산

마찰손실은 $\overline{S_f}$ 와 L 곱으로 계산되며, 여기서 $\overline{S_f}$ 는 구간에 대한 대표 마찰경사이고, L 은 아래 식들에서 정의한 것으로 다른 공식을 설정하지 않으면 일반적으로 식 (2.7)을 사용한다.

$$\overline{S_f} = \left[\frac{Q_1 + Q_2}{K_1 + K_2} \right]^2 \quad (2.7)$$

$$\overline{S_f} = \frac{S_{\Delta} + S_{\rho}}{2} \quad (2.8)$$

$$\overline{S_f} = \sqrt{S_{\Delta} * S_{\rho}} \quad (2.9)$$

$$\overline{S_f} = \frac{2S_{\Delta} * S_{\rho}}{S_{\Delta} + S_{\rho}} \quad (2.10)$$

2.1.5 한계수심

한계수심은 비에너지가 최소인 수심을 말하는 것으로, 하나의 단면에 대한 한계수심은 다음 4가지 경우에 결정된다. 첫째로 사류 흐름 영역이 규정되었을 때 두번째, 한계수심의 계산이 요구될 때, 세번째, 처음 단면에서 시작조건으로 한계수심이 규정되었을 때, 네번째, 상류 수면 곡선에 대해서 한계수심을 점검한다는 것은 계산된 수위와 관련된 흐름영역을 검증할 것인지를 결정하는데 한계수심이 필요함을 의미한다.

임의의 한 단면에서의 비에너지는 다음과 같이 정의한다.

$$H = WS + \frac{\alpha V^2}{2g} \quad (2.11)$$

여기서, H 는 총 에너지 수두, WS 는 수위, α 는 에너지 보정계수, $\frac{V^2}{2g}$ 는 속도수두를 나타낸다.

한계수위는 총 에너지 수두가 최소일 때의 수위이다. 한계수위의 결정 방법은 식 (2.11)에서 수위(WS)를 가정하고 이에 상응하는 H 값이 최소값이 될 때 까지 반복계산하여 결정한다. 반복 계산과정을 빠르게 하기 위해서는 포물선 보간기법을 이용한다.

2.2 GIS와 자료처리기법

GIS는 컴퓨터를 기반으로 하는 지리적 자료를 수집, 저장, 분석, 출력할 수 있는 시스템을 의미한다. 지형공간에 관한 모든 정보를 컴퓨터에 저장, 이를 바탕으로 인간이 사는 공간과 관련된 의사 결정을 효율적으로 하려는 시도의 산물이며 캐나다, 미국, 호주와 같은 광활한 대륙을 개발하고 통제하기 위한 수단으로 나온 환경의 산물이기도 하다.

지리정보시스템을 한마디로 정의하려고 하는 용어들을 크게 대별하면 기술적인 면과 문제해결의 면으로 나뉘는데, 기술적인 면의 정의는 컴퓨터와 관련된 점을 강조해서 사용자의 의도대로 지형에 관련된 공간 데이터에 대한 수집, 저장, 갱신이 용이하며, 변환과 관리를 하는 도구 일체라 하였으며, 문제해결(problem solving)은 시스템의 해석적인 면을 부각시켜 문제해결 환경에 있어서 지형관련 데이터의 통합을 포함한 의사결정지원 시스템이라 하고 있다. 이와 같은 여러 가지 학술적 개념을 종합하면 "컴퓨터를 이용하여 지형, 지세, 토지이용, 도시계획, 지역계획 등 공간을 대상으로 하는 정보를 수집·관리하고 그러한 정보들을 사용목적에 따라 다양한 분석을 통하여 필요한 결과물을 생산하는 공간분석에 관한 종합정보관리시스템" 이라고 정의를 내릴 수 있다.

2.2.1 지리정보시스템의 자료

지리정보시스템에서 처리하는 대상은 공간적 현상의 지리자료이다. 분포의 특성을 가진 것이라면 모두 지리자료라고 할 수 있으며 지도상에 이러한 각종 지리자료를 기입하여 왔다. 흔히 사용되는 주제도(thematic map)로는 지형도, 지질도, 토양도, 지적도 그리고 지하시설물(상·하수도, 지하전화배선, 지하가스관 등)을 표시한 지도 등이 있다. 지리자료는 부호로 표시되는데 그 밖에 주소나 발생건수 등은 글과 숫자로 기재되기도 한다. 완벽한 지리자료가 되기 위해서는 공간자료(spatial data)와 속성자료

(attribute data)가 함께 정의되어야 한다. 공간자료는 각종 지리현상의 위치와 형상 및 사상간의 공간상 상대적 위치관계를 말하는데 지도상에서는 점, 선, 면을 사용하여 표시된다. 속성자료는 점, 선, 면으로 표시된 각종 좌표상의 지리적 사상의 특성을 말한다. 예를 들어 하나의 점으로 표시된 어떤 지표공간을 수질관측소라고 하면, 한 쌍의(X, Y)좌표로 그 위치가 공간 자료로 변환한다. 속성자료에는 관측소의 명칭, 기호, 특정한 시각에 측정된 생화학적 산소요구량(BOD)이나 용존산소량(DO) 등이 포함된다. 도로와 같은 선(線)자료는 일련의 점을 상호 연관시켜 나타난다. 속성자료인 도로의 명칭, 노폭, 노면재료, 교통량 등은 문자와 숫자로 데이터베이스에 기재된다. 다각형 또는 면(面)자료인 행정구획도와 토양도 등은 하나 또는 여러 개의 선분으로 구성되는 폐곡선으로 공간자료가 만들어지고, 속성자료로는 폐곡선으로 구분되어진 지역의 인구특성이나 토양의 성질 등이 기록된다.

2.2.2 GIS의 발달 과정과 현황

지리정보시스템은 1960년에 캐나다 토지 및 환경정보시스템(CGIS)사업의 일환으로 캐나다의 광대한 토지자원조사와 분석관리를 위하여 활용되었다. 그 후 1960년대 후반에는 센서스 자료를 취급하는 캐나다 통계국의 GIS가 개발되었다. 이 시기에는 grid, shell(mesh)에 의한 시스템이 주로 개발되었으며, 그 중 미국의 하버드 대학의 SYMAP이 주로 이용되었다. 1960년대에는 영국, 호주에서도 GIS의 연구개발이 이루어졌으며, 1970년대에는 프랑스, 스페인, 독일 등의 유럽 각국에서도 연구개발이 이루어지게 되었다.

1960년대 후반의 미국의 대도시 계획의 수행과정에서 도시정보시스템의 개발이 본격적으로 시작되었으며, California, Maryland, New York 등 각 주에 있어서도 정보시스템의 개발이 시도되었다. 특히 통계국(bureau of census)이 1970년 국세조사에 geocoding system(주소로부터 위치좌표

를 검색할 수 있는 시스템)을 도입하여, 도로와 위치좌표를 중첩시킴으로써 각 지방공공단체가 쉽게 GIS를 도입할 수 있게 되었다. 이와 같이 표준화가 이루어진 결과 1980년대 초 Syneroom, Intergraph, ESRI와 같은 업체가 GIS Turnkey System을 판매하게 되어 보급에 박차를 가하게 되었다.

초창기 지리정보처리는 간편성, 시스템 구축의 용이성, 하버드 대학의 SYMAP 등과 같은 소프트웨어를 이용하여 출력할 수 있을 것, 데이터를 간단하게 취급할 수 있을 것 등의 요구사항 때문에 격자(mesh)에 의한 자료구조를 갖는 것이 많았다. 이 격자형 시스템은 일반적으로 래스터(raster)라 불리는 자료구조를 갖는다. 이 자료구조는 점이 하나의 격자면으로 표현되고, 선의 경우 지정방향으로 향한 주변 면의 갯수로 표현되며, 다각형의 경우 주변 면들의 집합으로 표현된다. 특히, 줄과 열의 배열을 저장하고, 조작하기가 쉬운 포트란 등과 같은 프로그래밍 언어에서 조작하기 쉽고, 비용이 적게 들며, 각 공간단위의 일관성으로 인한 단순성으로 인해 대학과 같은 기관에 연구용으로 격자형 시스템이 많이 보급되었다. 또한 그 전형적인 시스템이 하버드 대학에서 개발된 GRID/IMGRID system이었다. 그러나 격자형 시스템은 초기의 인기와는 달리 화소크기에 따른 형상학적 인식구조의 변질문제, 이에 따른 속성정보의 손실, 당시의 기술수준에 비하여 엄청나게 요구되는 기억용량 등으로 인해 점차 그 사용의 한계가 노출되었다.

1970년대의 중반 이후 다각형 기반의 GIS는 기존의 격자형 자료구조의 시스템에 비하여 벡터(vector)형식의 자료구조를 갖는다. 이 자료구조는 형상학적 데이터 구조의 제공과 압축된 데이터 구조, 위상의 상호 연결성, 도형과 비도형 데이터의 수정 및 일반화가 가능하다는 등의 장점으로 인해 지방 공공 단체에서 각광받기 시작하였고, 이후의 GIS개발에 있어서 주된 흐름을 차지하였다. 1990년 현재, 북미 대륙 내에서 이미 4,000개 이상의 GIS자동화 시스템이 가동되고 있는 것으로 추정되고 있다. 또한 영

국을 비롯한 유럽지역은 물론 남미 지역에서도 정부 및 민간 연구기관에서 GIS의 활발한 개발과 이용이 계속되고 있다. 일본, 대만, 홍콩 등에서도 활발히 진행되고 있으며, 특히 대만의 경우에는 국립대만대학에서 180만불의 예산을 바탕으로 NGIS(national geographic information system)의 구축을 위한 기초연구가 시행되고 있다.

한편 일본의 경우는 1970년대에 들어와서 미국, 캐나다 등지에서 개발된 GIS가 도입되기 시작하여 1975년에는 국토청의 주도로 국토정보사업이 착수되었으며, 80년대부터는 건설성에서 도시정보 시스템(UIS)의 구축이 시작되었다. 즉, 국토지리원 내에 지도정보실 및 사진측량기술 개발실을 신설하여 컴퓨터를 이용한 지형도, 주제도 등의 자동도화기술 개발에 착수하였다. 이후 일본은 중앙정부와 각급 지방단체에서 지리정보의 활발한 구축이 진행되었고 민간부문에 있어서도 적극적으로 활용되고 있다.

우리 나라의 경우는 해외 연구동향에 비추어 볼 때 GIS의 역사가 매우 짧으나 국립지리원, 국토개발연구원, 한국전력기술, 통계청, 도로공사, 토지개발공사, 농업기반공사, 지적공사, 주택공사, 서울시청, 광주시청, 인천시청, 건설부 등의 각 기관과 서울대학교, 한양대학교, 강원대학교, 수원대학교, 성균관대학교, 경북대학교 등을 비롯한 각 대학과 과학기술 연구원, 해양연구소, 국토개발연구원 등의 연구소에서 활발한 연구가 진행되고 있다.

최근에 들어서 기술상으로 점차 성숙되어 이전의 많은 문제점이 해결되고 실용적인 시스템으로 발전됨에 따라 GIS는 각종 지리정보를 정리, 취급하기 위한 유용한 도구로 자리잡아 가고 있다. 연구보고에 따르면 미국 정부 각 부문의 정책 결정과정에서 80% 이상이 GIS 분석자료의 이용을 필요로 하고 있으며, 미국정부는 매년 많은 예산을 투자하여 필요한 지형과 지리자료를 수집하고 있다. 또한 GIS에 대한 수요가 급증할 것이라는 전망은 하드웨어와 소프트웨어의 발전과 새로운 자료출처의 개발에 있어서의 낙관적인 추세에서 그 근거를 쉽게 찾을 수가 있다. 현재에 있어서

의 GIS 연구방향은 현실세계의 변화를 설명하는 방법과 인간의 인식능력 한계를 보완하는 새로운 수단을 찾는 데 초점을 맞추고 있다.

2.2.3 GIS의 자료처리방법

최근 DEM 자료를 수문학에 이용함에 있어 공간적 해상도가 미치는 영향에 관한 연구가 활발히 이루어지고 있다. DEM 자료가 지형의 표현과 수문학적 과정의 모의에서 편리성을 제공하지만 분산적인 자료의 특성 때문에 약간의 제약을 가지고 있다(Moore 등, 1993 ; Holmgren, 1994). 서로 다른 격자 크기의 DEM 자료는 유역면적이나 유출생성 등의 수문학적 모의에서 상이한 결과를 초래할 수 있으므로(Bruneau 등, 1995) 격자의 크기는 수문모형의 수행에 있어서 중요한 요소가 된다. DEM 자료는 지형고도의 공간적 분포를 표현할 수 있으며 유역의 수문학적 과정을 모의하는데 사용되고 있다. 경사, 방위, 유역면적, 유로연장 등과 같은 많은 수문모형의 매개변수들이 DEM 자료로부터 추출될 수 있다(Vieux, 1996). 지형의 공간적 분산도는 수문모형의 모의에 많은 영향을 미치며, 수문모형에 고도자료를 적용할 때는 지표면 표현을 위해 어떠한 방법이 사용되는지 고려하는 것이 중요하다. 지표면을 나타내기 위해 DEM을 구성하는 방법에는 대표적으로 다음 세 가지 자료를 활용할 수 있다.

- ① 격자기반자료
- ② 불규칙삼각망자료
- ③ 등고기반자료

격자기반자료를 수문모델링에서 최초로 적용한 사례는 1971년 지형의 좌표정보를 래스터형으로 저장한 것이었다. 격자는 규칙적인 간격의 선으로 구성되어 있고, 각 직사각형 내의 면적은 그것의 중심부 좌표에 의해서 기술된다. 지형에 대한 래스터 이용은 대규모의 DEM 데이터 베이스를 이

용하는 경우 매우 합리적이라 할 수 있다. 널리 쓰이는 래스터형 GIS로서는 미국육군공병단의 GRASS(geographic resources analysis support system)가 있다.

DEM을 형성하는 또 다른 기법은 불규칙적으로 분포된 점을 여러 개의 선으로 연결하여 중요한 산봉우리와 계곡 등을 결정할 수 있는 불규칙 삼각망을 이용하는 것이다. 이와 같이 형성된 불규칙삼각망을 TIN이라고 한다. 전형적으로 대부분의 삼각형이 2차원 평면으로 처리되지만 부드러운 선형보간이 가능하다.

지형을 나타내는 세 번째 형태는 등고선이다. 등고선은 동일표고를 가진 일련의 point-to-point 경로에 의해서 수치적으로 표현될 수 있다. 전체 지도가 이와 같은 수치형태로 저장될 때, 이것을 DLG(digital line graph)라 부른다. 상업적으로 이용가능한 대부분의 GIS 소프트웨어들은 DLG, 격자기반 DEM 및 TIN 간에 서로 변환할 수 있는 기능을 가지고 있으나, 등고기반의 기법이 보다 많은 자료용량의 크기를 가지므로 DLG로부터 주로 다른 형태로의 변환이 이루어진다. 이 기법의 가장 큰 장점 중의 하나는 중요한 수문속성이 그 결과로 나타나는 자료구조에 그대로 포함되어 있다는 것이다.

제 3 장 적용대상유역의 수문해석

3.1 적용대상유역의 자료분석

본 연구대상유역은 수영강 유역으로서 부산광역시의 대표 하천으로서 경상남도 양산시와 접하고 있다. 유역면적은 199.57km^2 이고, 유로연장은 28.0km, 하상경사는 13.84%에 달하는 지방 2급 하천이다. 또한 유역은 상류와 하류로 크게 구분할 수 있으며, 상류유역에는 상수도 전용댐인 회동댐이 있다. 회동댐 상류유역은 수영강 전체유역의 약 50%에 달하고 있으며 하류유역에는 크고 작은 교량이 9개소 이상이 있다. 하류유역은 도시하천으로 바다를 접하고 있는 감조하천이 존재하고 있다.

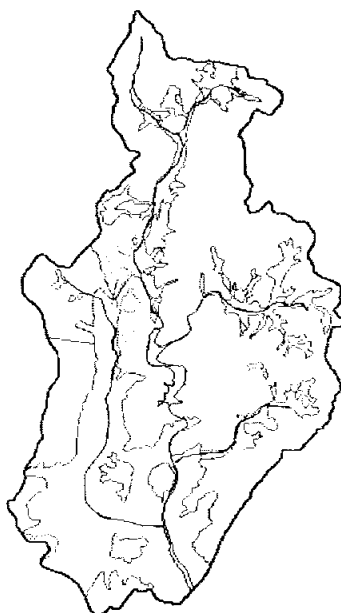


그림 2 대상유역의 토지이용도

본 유역의 토지이용현황은 그림 2와 같이 임야가 59.1%, 농경지가 12.9%, 수체가 1.9%, 초지가 1.0%, 도시지역이 25.1%로 이루어져 있으며, 그림 3과 같은 토양도를 구성한다.

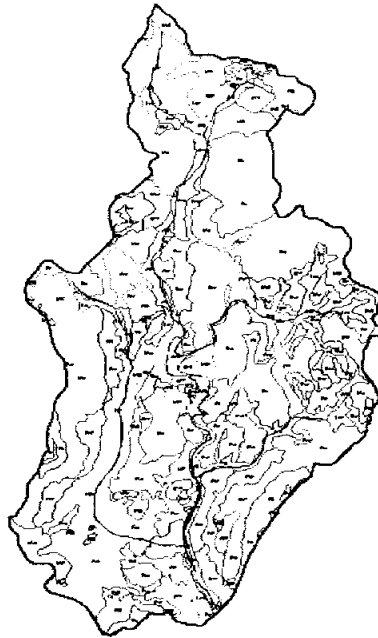
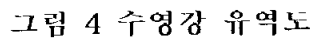


그림 3 대상유역의 토양도

해당유역의 횡단 및 종단면자료 중 대표적인 단면으로 회동수원지의 상류 및 하류 단면을 비교하여 나타내면 횡단면의 경우, 그림 5와 그림 6과 같이 하류가 상류에 비해 대체적으로 하천의 폭이 크다는 것을 알 수 있으며, 종단면의 경우 하류에서 상류쪽으로 올라갈수록 경사가 급해지는 등의 특성을 알 수 있다.



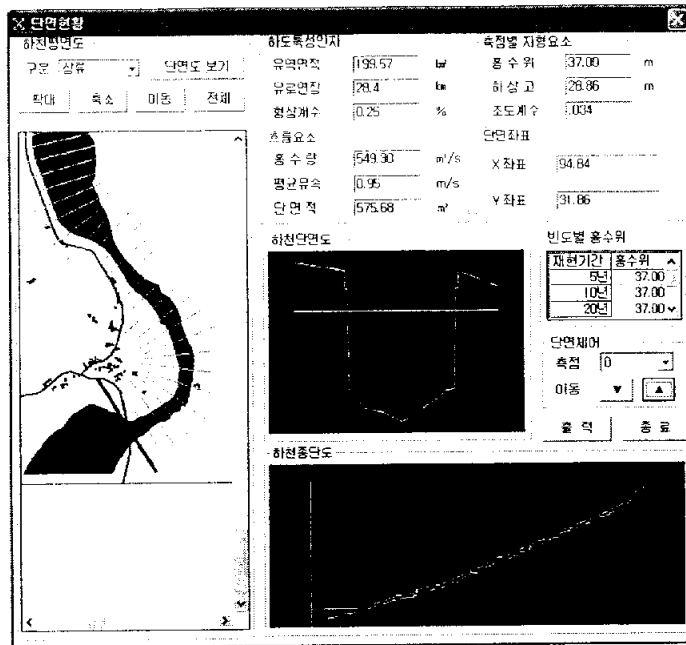


그림 5 회동수원지 상류의 종·횡단면도

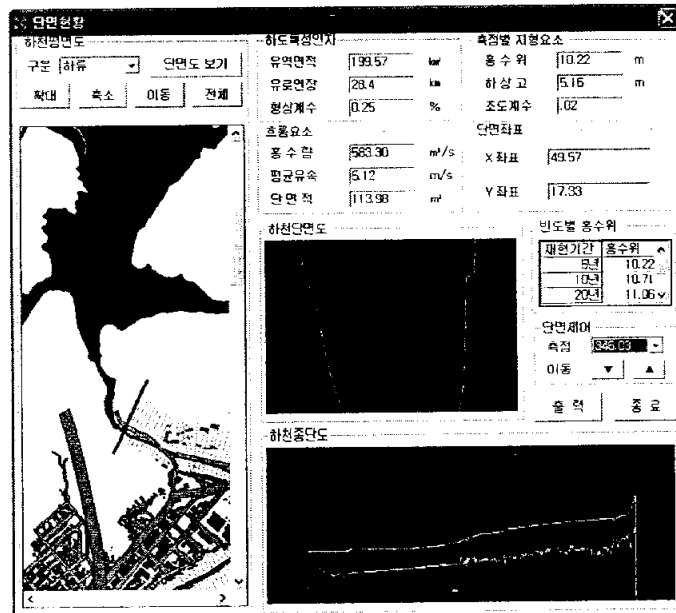


그림 6 회동수원지 하류의 종·횡단면도

3.2 유출모형의 적용 및 선정

3.2.1 유출모형의 적용

본 연구에서의 홍수량의 산정은 평시유출량을 산정할 수 있는 장기유출해석과 첨두홍수량 및 수문곡선을 산정할 수 있는 단기유출해석이 있다.

단기유출해석의 경우 합리식이나 Kajiyama 공식과 같은 첨두홍수량만을 산정하는 방법과 합성단위유량도법이 있다. 본 연구에서는 대상유역내의 하도내의 저류효과를 고려하고, 상류와 하류를 연결하고 있는 회동저수지의 저수지 추적을 통하여 보다 합리적인 홍수량 산정을 위해 단순히 첨두홍수량만을 산정하는 합리식이나 Kajiyama 공식과 같은 방법은 피하고, 본 대상유역내에 수위-유량관계의 실측자료가 전무한 점을 감안하여 유출수문곡선을 산정할 수 있는 합성단위유량도인 Clark유역추적법, Nakayasu단위도법, Nash모형, SCS단위도법 등을 이용하여 산정하였다.

1) 유출모형

① Clark유역추적법 : 유역을 대표하는 시간-면적주상도의 추적에 의하여 지속기간이 영인 단위유효강우량을 유출수문곡선으로 변환하는 것을 기본개념으로 하며 순간단위도 개념을 이용, 한 유역에 대하여 단 하나의 단위도를 유도한다.

② Nakayasu단위도법 : 일본의 유역특성을 이용한 방법으로 SCS무차원단위도법과 유사하며 첨두시간과 첨두유량을 산정한 후 상승부 곡선에 대한 식과 하강부 곡선에 대한 식으로 나누어 종거를 산정한다.

③ Nash모형 : 대표적인 개념적 순간단위도는 매개변수가 2개인 Nash모형으로 이는 n 개의 선형저수지로 구성되고 한 저수지의 유출량은 다음 저수지의 유입량이 된다 수학적으로 Nash모형은 상류 유입량을 하류 선형저수지에 n 번 선회함으로써 얻어진다.

④ SCS단위도법 : 첨두유량과 유역지체가 주어지면 합성 무차원수문곡

선을 이용하여 단위도가 유도될 수 있으며, 유량을 첨두유량과의 비로, 시간을 첨두까지의 시간인 첨두시간(Time to peak)과의 비로 나타낸 합성단위도이다.

3.2.2 유출모형의 선정

빈도별 홍수량은 그림 7과 같이 7개의 산정지점별로 계산하여 표 1과 같이 산정할 수 있었다. 그리고 빈도별 홍수량은 모두 상기에서 서술한 배수구역별 홍수량 산정방법 및 저수지 홍수추적, 하도추적법을 모두 적용한 결과이며, 본 연구에서는 유역의 지형학적 특성과 강우의 시간분포를 고려할 수 있는 유역추적법인 Clark 단위도법을 선정하였다. 그리고, 유출모형을 산정하기 위한 등시간도는 그림 8과 같다.

또한 표 1에서 볼 수 있듯이 해당유역의 유출구(산정지점 SY5)에서의 빈도별 홍수량을 산정방법에 따라 살펴보면 100년빈도에서 Clark유역추적법이 2090.53m³/s, Nakayasu단위도법이 1835.26m³/s, Nash모형이 1741.42m³/s, SCS단위도법이 2166.15m³/s와 같이 산정되었으며, 표 1에서 산정된 유출량은 HEC-RAS 입력데이터로 자동 생성되어 HEC-RAS가 실행된다.

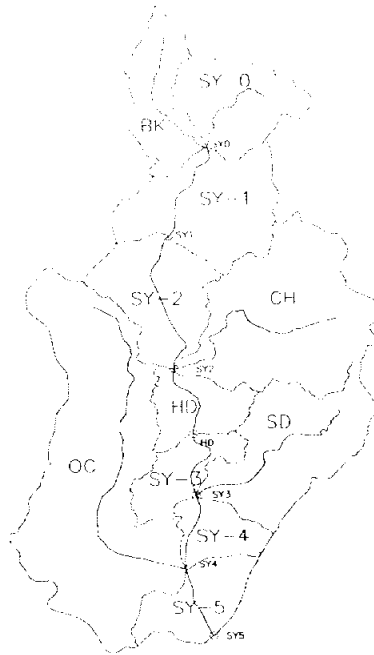


그림 7 대상유역의 산정 지점

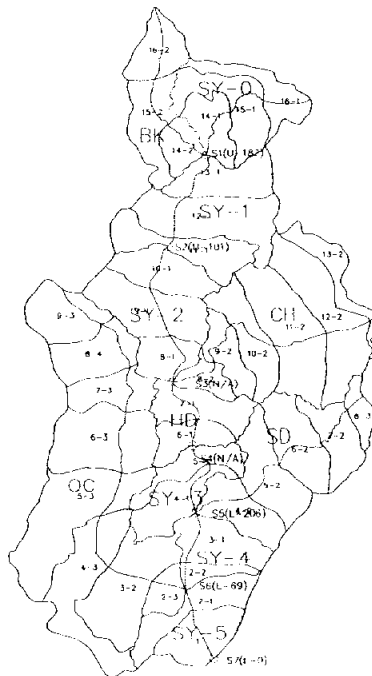


그림 8 대상유역의 동시간도

표 1 지점별 빈도별 홍수량 산정

산정 지점	유역 면적 km^2	산정 방법	빈도별 홍수량(m^3/s)							
			5년	10년	20년	30년	50년	80년	100년	200년
SY0	24.6	Clark	109.36	136.19	162.16	177.14	195.83	212.93	221.00	246.16
		Nakayasu	122.68	160.96	200.13	223.45	252.62	279.71	292.67	332.76
		Nash	87.08	108.85	130.50	143.02	158.69	173.05	179.83	201.01
		SCS	129.72	162.14	195.90	217.55	244.59	269.64	281.62	318.62
SY1	44.1	Clark	268.08	330.33	390.64	425.39	468.80	508.52	527.27	585.73
		Nakayasu	168.02	218.13	267.08	295.66	331.26	364.18	379.89	428.37
		Nash	250.42	308.84	365.19	397.66	438.23	475.35	492.87	547.51
		SCS	386.93	478.40	566.51	617.24	680.55	738.42	765.73	850.80
SY2	92.5	Clark	293.14	360.62	425.25	462.38	508.87	551.55	571.64	638.89
		Nakayasu	387.60	496.81	602.71	664.32	740.87	811.50	845.16	948.87
		Nash	324.44	399.27	471.41	512.98	564.90	612.40	634.82	704.73
		SCS	446.00	548.30	646.80	703.50	774.27	838.98	869.51	964.66
HD	100.4	Clark	549.32	673.34	793.33	862.45	948.85	1028.15	1065.50	1190.53
		Nakayasu	460.30	585.25	705.99	776.10	863.11	943.30	981.49	1099.09
		Nash	506.26	620.63	730.80	794.25	873.48	945.95	980.16	1086.80
		SCS	694.34	850.01	999.78	1085.98	1193.55	1291.90	1338.31	1482.94
SY3	129.9	Clark	583.29	714.88	841.12	913.82	1004.67	1088.08	1127.36	1258.86
		Nakayasu	811.21	1026.55	1234.16	1354.67	1503.94	1641.49	1707.07	1908.13
		Nash	539.56	660.92	777.81	845.13	929.18	1006.06	1042.34	1155.46
		SCS	680.25	832.31	978.66	1062.91	1168.06	1265.75	1312.13	1456.71
SY4	189.5	Clark	677.83	829.84	975.91	1078.43	1174.21	1283.54	1370.83	1458.78
		Nakayasu	702.66	886.19	1062.89	1165.40	1292.33	1409.25	1464.99	1635.83
		Nash	591.37	722.89	849.48	922.37	1013.36	1096.58	1135.85	1258.27
		SCS	707.90	863.67	1013.59	1100.47	1208.95	1308.46	1355.30	1512.02
SY5	199.5	Clark	987.64	1207.02	1446.46	1650.91	1795.76	1960.27	2090.53	2107.57
		Nakayasu	901.51	1127.22	1343.83	1469.27	1624.42	1767.22	1835.26	2043.70
		Nash	918.67	1117.67	1309.00	1419.11	1556.51	1682.14	1741.42	1926.18
		SCS	1137.38	1387.18	1626.28	1763.54	1934.92	2092.14	2166.15	2413.80

3.3 수리학적 모형의 분석 및 선정

3.3.1 하도 홍수위 분석

하도의 홍수위의 선정 기법은 정류 및 부정류 모형, 그리고 등류 및 부정류 모형으로 크게 대별할 수 있다. 본 연구에서는 미 공병단에서 개발된 정상 부정류 및 부정류 모형인 HEC-RAS를 통하여 수영강 본류의 홍수위 분석을 수행하였다. 이 모형은 교량 및 하천 구조물 등의 홍수위 영향을 고려할 수 있으며, 또한 부족한 하도 구간의 내삽 기능 및 임의의 구조물의 삽입, 세굴의 영향 검토 등의 기능이 하천 수리 검토를 위하여 제공되고 있다.

3.3.2 기점수위의 결정

수위는 하천의 흐름이 상류(subcritical flow)일 때는 하류(downstream)의 영향을 받고 하류(supercritical flow)일 때는 상류(upstream)의 영향을 받는다. 따라서 이러한 흐름에 따라 하류 또는 상류측 전단면의 수위를 초기조건으로 갖게 되며, 이때의 수위를 기점수위라 한다.

본 연구대상유역인 수영강은 회동수원지를 중심으로 상류와 하류로 구분되고 상류하천인 경우에는 부산광역시 상수도본부에서 제공한 수위 및 저류자료를 통하여(그림 9) 기점수위를 결정할 수 있었으며, 기점수위는 E.L. 35.3m로 선정하였다. 하류의 경우에는 바다와 인접하고 있는 감조하천으로써 부산항 대조평균만조위와 비교하여 큰 값인 대조평균만조위를 기점홍수위(0.475E.L.m)로 선정하였다.

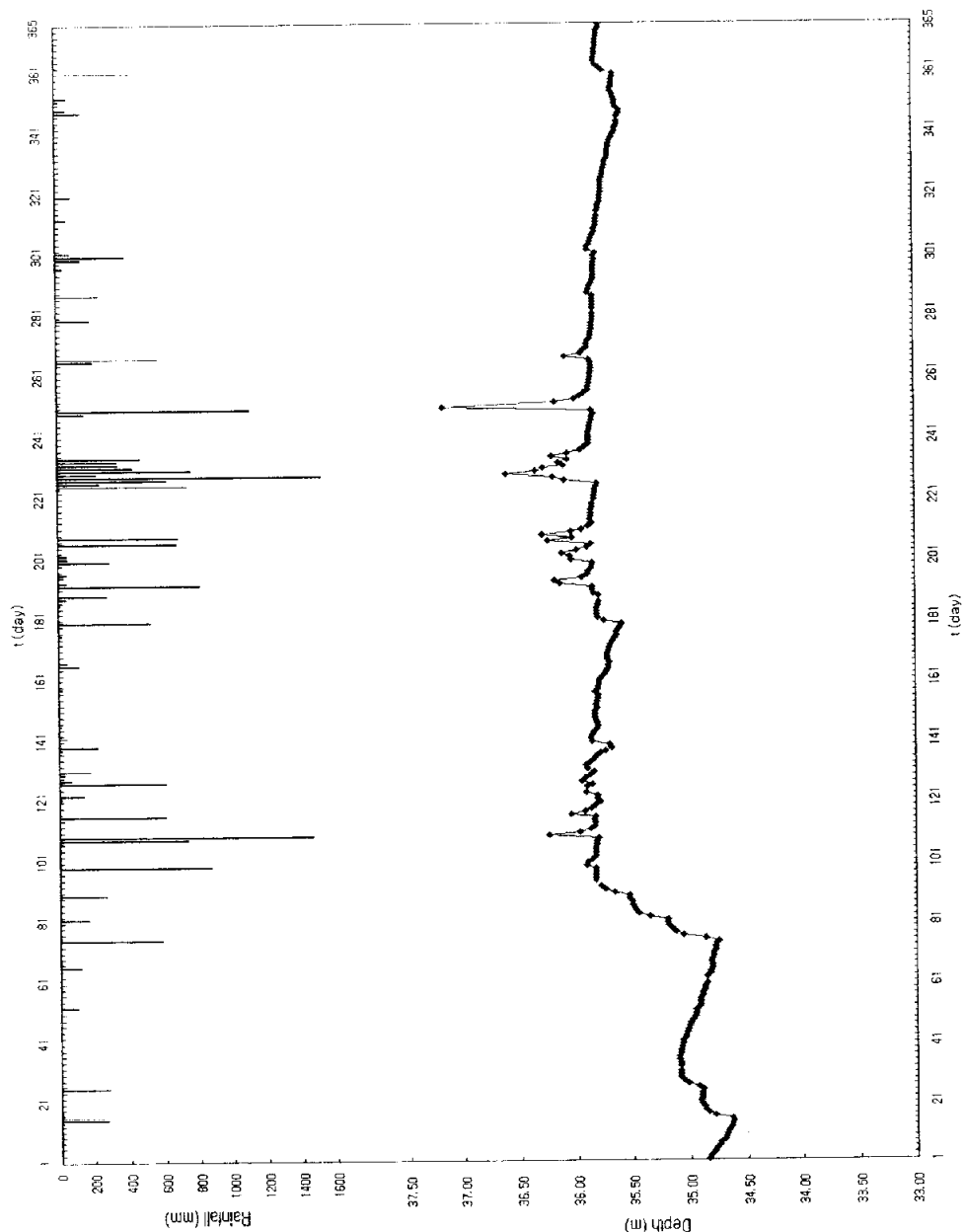


그림 9 수영강 회동댐 상류 홍수위

제 4 장 하천관리 시스템 구축

4.1 단면자료 구축

산정된 유출모형을 HEC-RAS로 실행하면, 다음과 같은 분석창 및 종단면도, 횡단면도를 각각 얻을 수 있으나 아래의 그림과 같이 각각의 결과 화면을 동시에 조회할 수 없고 별도의 창에 나타나게 된다.

Cross Section Data - su young geometry final

Exit Edit Options Plot Help

River: SuYoung River Apply Data

Reach: Upper Reach River Sta: 21

Description

Del Row	Ins Row	Cross Section X-Y Coordinates	
		Station	Elevation
1		-112.75	42.47
2		-112.75	42.47
3		-102.75	40.47
4		-86.85	36.97
5		-82.15	35.97
6		-78.15	35.97
7		-77.65	35.67
8		-70.85	35.67
9		-60.65	33.77
10		-38.95	33.57

Downstream Reach Lengths		
LOB	Channel	ROB
44	50	57

Manning's n Values		
LOB	Channel	ROB
0.04	0.03	0.04

Main Channel Bank Stations	
Left Bank	Right Bank
-112.75	112.75

Contr/Exp Coefficients	
Contraction	Expansion
0.1	0.3

Normal Ineffective Flow Areas
Levee on left and right sides

Enter to move to next downstream river station location

그림 10 임의 지점(River Station)에 대한
데이터 분석창

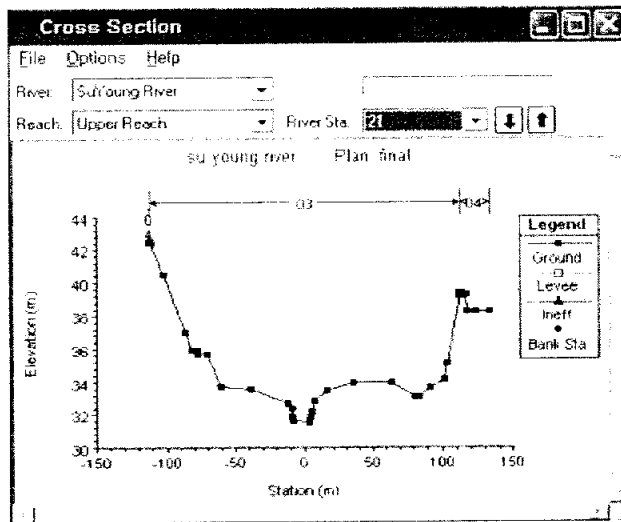


그림 11 임의 지점(River Station)의
형단면도

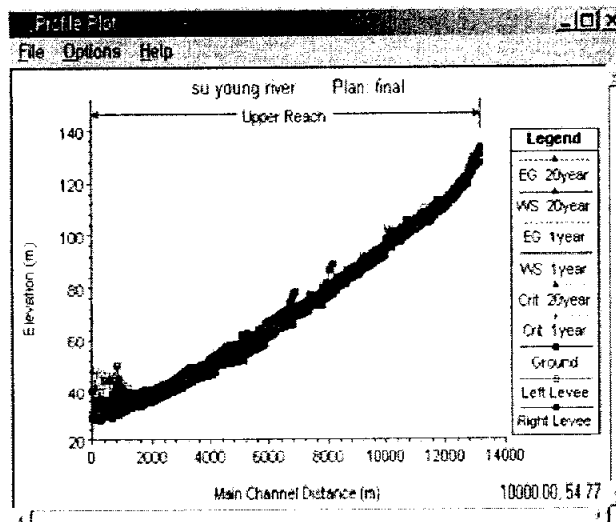


그림 12 해당구간의 종단면도

이상과 같이 HEC-RAS를 이용하여 입력된 유출모형을 실행하면 데이터분석창, 횡단면도 및 종단면도가 각각 모니터에 나타나는데 이로 인한 사용자의 불편함을 다음설과 같이 HEC-RAS 연계 프로그램의 구축을 통해 한 화면에 이 모두를 나타낼 수 있도록 하였다.

4.2 HEC-RAS 연계 프로그램 구축

본 연구에서 구축된 HEC-RAS 연계 프로그램은 하천관리시스템과 연계시키기 위하여 Visual C++로 변환하여 강우-유출모형과 수리학적 모형을 통합시켰다.

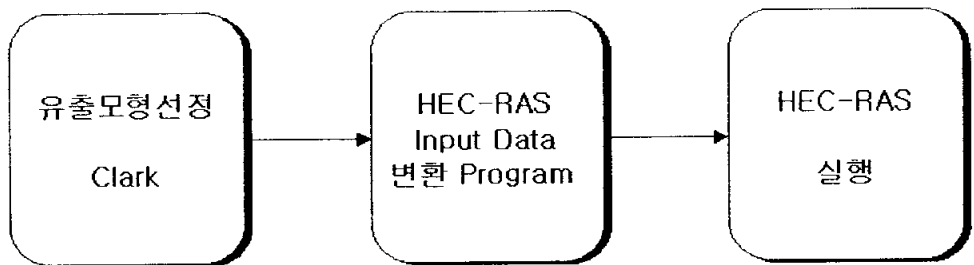


그림 13 HEC-RAS 연계 프로그램 개발

또한, 본 시스템에서는 HEC-RAS 모형의 입력데이터를 강우-유출 모형에서 계산한 값을 자동으로 작성하게 하였으며, 시스템 사용자가 HEC-RAS 모형을 실행함으로써 최종 홍수위가 결정되는 방법으로 모형을 연계시켰다.

4.2.1 유출모형선정

전(前) 절에서 언급한 Clark유역추적법, Nakayasu단위도법, Nash모형, SCS단위도법 등의 유출모형을 이용하여 수영강유역에 대한 홍수량을 산정하여 비교한 후, 유역의 지형학적 특성과 강우의 시간분포를 고려할 수 있는 Clark유역추적법으로 선정하였다.

4.2.2 HEC-RAS Input Data 변환 Program

위와 같이 선정된 Clark유역추적법에 대한 전산 프로그램을 Fortran으로 구축한 후 시스템과 연계하기 위하여 Visual C++로 HEC-RAS 입력데이터를 자동으로 생성할 수 있도록 하였다.

4.2.3 HEC-RAS 실행

HEC RAS 연계 프로그램은 먼저 Clark모형의 입력데이터(Clark_sy.dat)를 읽어 유출모형을 실행시킨 후 빈도별 지속시간별 홍수량을 산출한다. 산출된 홍수량은 HEC-RAS 입력데이터 생성 프로그램(HEC_SYS.exe)에 의하여 입력데이터(sy.f02)를 만든 다음 HEC-RAS를 실행하게 되면 입력된 강우량에 대한 홍수위를 산정할 수 있다.

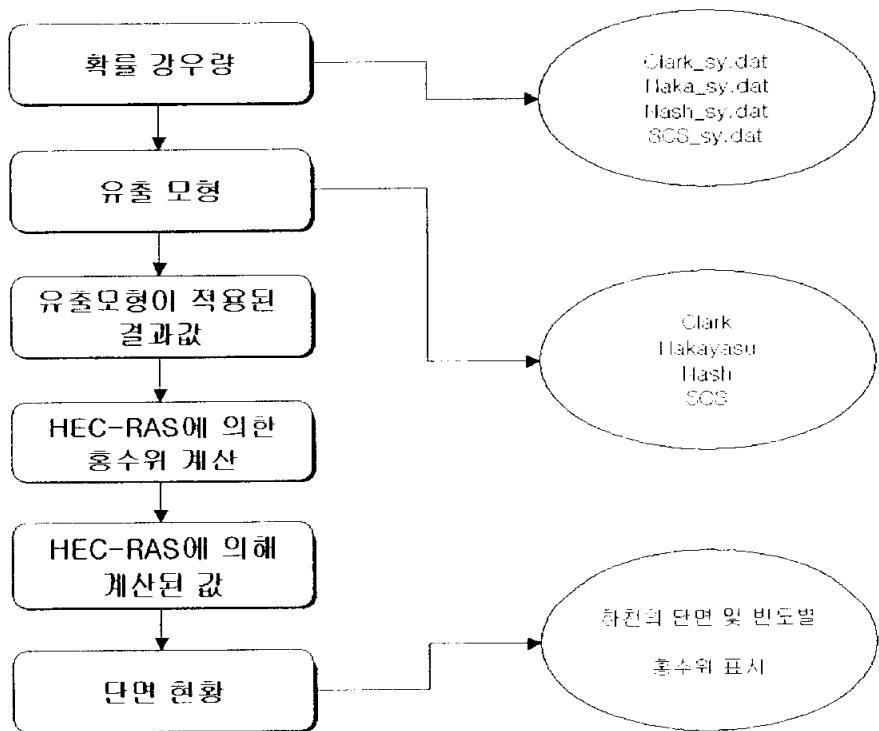


그림 14 HEC-RAS 연계 프로그램 흐름도

4.3 GIS와 연계한 하천관리시스템 구축

하천관리시스템은 크게 지형자료, 하천단면, 하천시설물로 구분하였다. 지형자료는 위치정보와 하천 수위의 중요한 건물 및 기타 시설물 현황에 대한 정보를 제공하여 하천의 수위의 변화에 민감한 지역을 조회하기 쉽게 하였다. 이처럼 GIS와 연계함으로써 하천의 효율적인 관리에 대한 장점을 제공할 수 있다고 본다.

하천관리시스템은 ESRI社의 MapObject와 Map&You社의 DViewer.ocx를 이용하여 지형자료와 단면자료에 대한 수위의 변화 및 정보를 보여준다. 특히 평면도, 단면도, 종단도를 동시에 나타냄으로써 하천관리를 하는데 편리함을 제공하고 있다. 또한, 하천 시설물에 대한 정보는 시설물관리시스템을 구축하여 별도로 관리가 가능하도록 하였다. 시설물관리시스템에서는 교량(bridge) 및 웨어(weir) 부근에 수위의 변화를 확인할 수 있으며 지리정보를 통하여 하천시설물 주위의 피해를 예상할 수 있다.

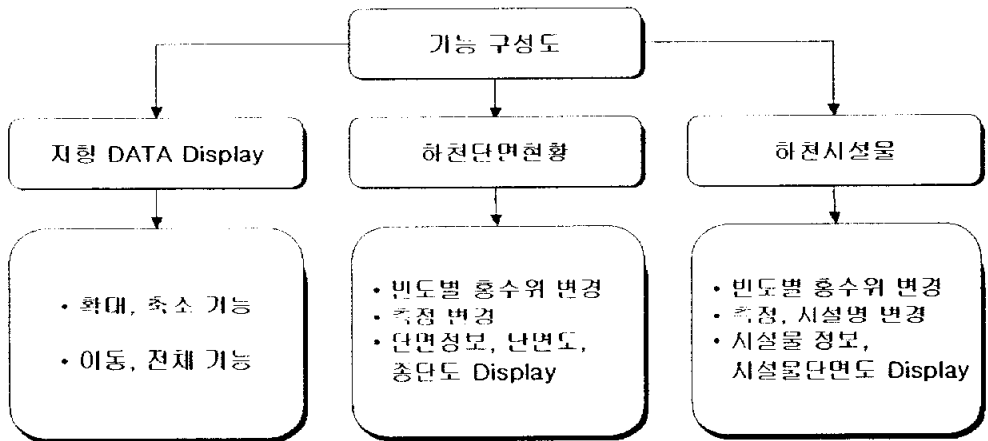


그림 15 기능구성도

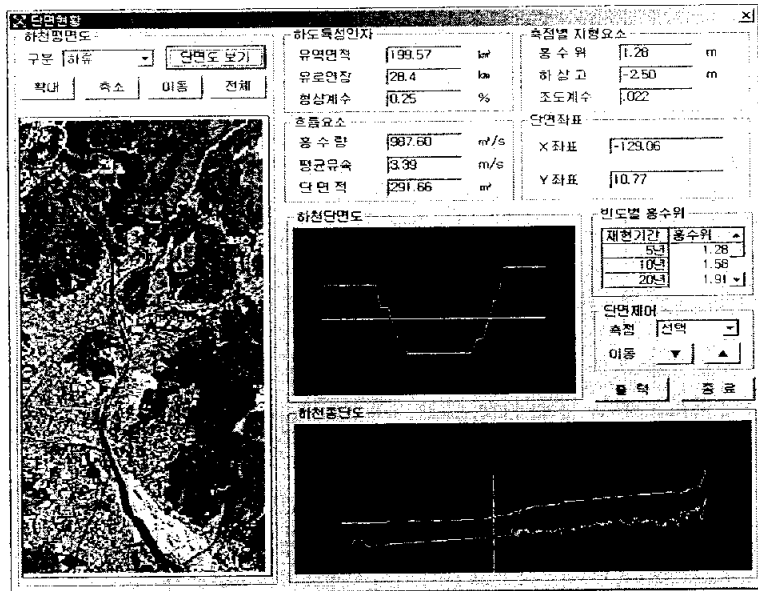


그림 16 하천관리시스템 중 '단면현황'실행시 화면

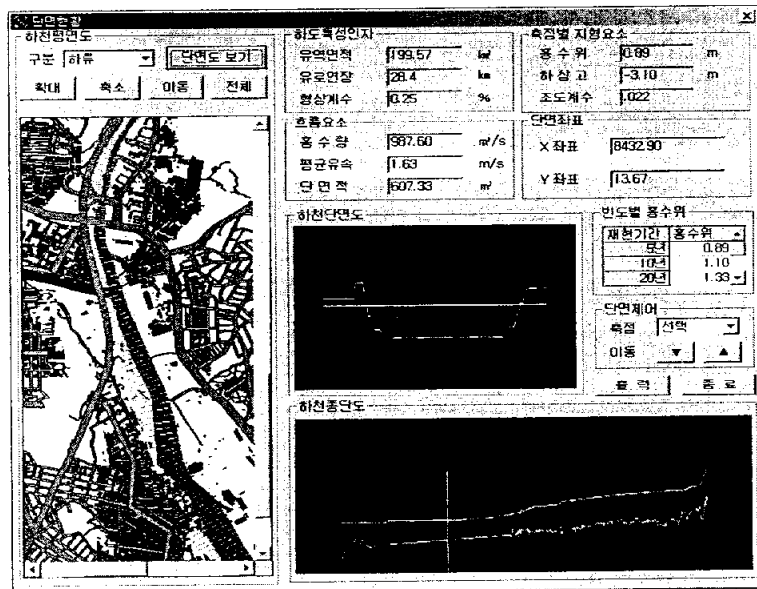


그림 17 지리정보시스템(GIS)과 단면도 현황

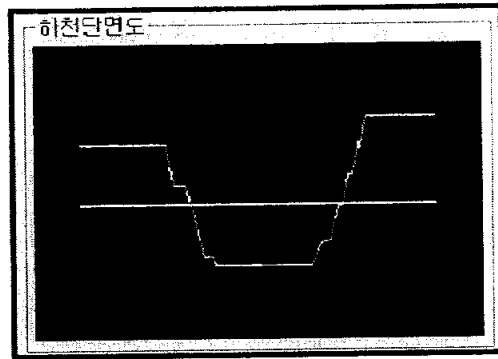


그림 18 하천유역 내 임의
측점에서의 하천단면도

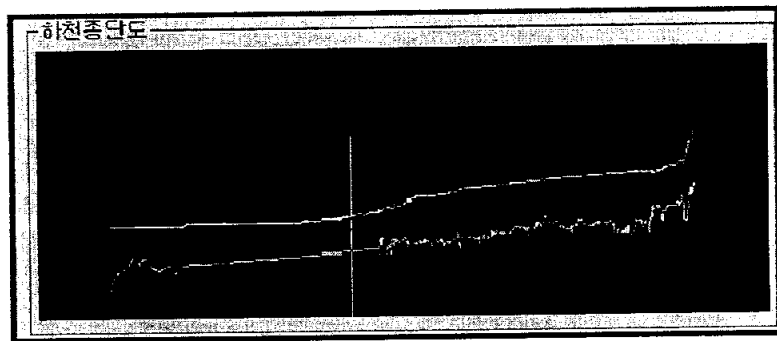


그림 19 하천종단도 상에 나타나는 측점의 위치

측점별 지형요소		
홍 수 위	1.28	m
하 상 고	-2.50	m
조도계수	0.22	

(가) 측점별 지형요소

단면좌표	
X 좌표	-129.06
Y 좌표	10.77

(나) 단면좌표

하도특성인자		
유역면적	199.57	km ²
유로연장	28.4	km
형상계수	0.25	%

(다) 하도특성인자

흐름요소		
홍 수 량	987.60	m ³ /s
평균유속	3.39	m/s
단 면 적	291.66	m ²

(라) 흐름요소

그림 20 지정된 임의의 측점에 대한 각종 데이터

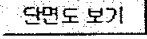
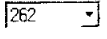
4.3.1 하천단면현황 설명

위 절에서 설명한 바와 같이 하천관리시스템의 초기화면에서 단면현황을 실행하게 되면 유역면적, 유로연장, 형상계수와 같은 하도특성인자와 홍수위, 하상고, 조도계수와 같은 측정별 지형요소 그리고 하천평면도상에서 원하는 부위를 지정할 경우 그 위치의 단면도 및 종단면상의 측정치를 볼 수 있을 뿐만 아니라 측정의 빈도별 홍수위까지 동시에 나타나도록 되어있다. 그리고 각각의 기능을 살펴보면 다음과 같다.

표 2 하천단면현황에 대한 기능설명

구 분	설 명
구분기능	화면 상단의 구분  아이콘을 이용 해당 평면도를 분할하여 상류 또는 하류와 같이 원하는 구간의 선택이 가능하다.
단면도 보기	특정 측정의 단면도를 관찰할 때 화면의 상단에서  아이콘을 마우스로 클릭한 후 평면도 상에서 마우스를 별도의 클릭없이 해당측점에 위치하기만하면 하천관리시스템 화면의 우측에 단면이 나타나므로 사용자의 편리한 검토가 가능하다.
확대기능	화면의 상단에서  아이콘을 마우스로 클릭한 후 평면도 상에서 확대를 원하는 위치에 드래그를 하면 다음의 그림 과 같이 해당위치의 확대화면을 볼 수 있다.
축소기능	화면의 상단에서  아이콘을 마우스로 클릭한 후 확대된 평면도 상에서 축소를 원하는 위치에 클릭하면 축소된 화면을 볼 수 있다.
이동기능	확대된 평면도 상에서 다른 지점의 확대평면을 관찰하고자 하는 경우 화면상단의  아이콘을 클릭한 후 원하는 위치로 마우스를 클릭한 채로 이동하여 화면의 이동이 가능하다.
전체기능	확대 또는 축소된 상태의 평면도를 최초의 전체화면으로 복귀시키고자 하는 경우  버튼을 클릭하면 전체 화면으로의 복귀가 가능하다.

표 2 하천단면현황에 대한 기능설명(계속)

구 분	설 명
하도특성인자	아래 그림과 같이 해당구역의 구역면적, 유로연장, 형상계수를 나타낸다.
흐름 요소	단면도 보기를 클릭한 후 임의 지점에 마우스를 위치하면 측정에서의 홍수량(m^3/s), 평균유속(m/s), 단면적(m^2)을 알 수 있다.
측점별 지형요소	단면도 보기를 클릭한 후 임의 지점에 마우스를 위치하면 그 측정의 홍수위(m), 하상고(m), 조도계수를 나타낸다.
하천단면도	본 시스템의 가장 큰 장점 중 하나로써 관리자가 임의의 지점을 일일이 검색·지정할 필요없이 평면상에서 원하는 측점에 마우스를 위치하기만 하면 그 위치의 단면도를 볼 수 있으며, 우측의 빈도별 홍수위 창을 이용 재현기간에 따른 홍수위변화를 확인할 수 있다.
하천종단도	해당구역의 종단면도를 한눈에 확인할 수 있으며 지정된 측정의 종단도 상의 위치를 나타내 주며 전 구역의 각각의 재현기간에 따른 홍수위를 빈도별 홍수위 구분창을 이용 쉽게 알아볼 수 있다.
단면 제어	 버튼을 이용 마우스의 이동으로 해당구역의 단면도 및 종단도를 쉽게 확인할 수 있는 반면, 측정  또는 이동   버튼을 이용 확인하고자 하는 특정 측정의 단면도 및 종단도를 정확하고 신속하게 검색 가능하다.

4.3.2 래스터형 데이터와 벡터형 데이터

래스터(Raster)형 데이터란 점이 하나의 격자면으로 표현되고 선의 경우 지정방향으로 향한 주변 면의 개수로 표현되며, 다각형의 경우 주변 면들의 집합으로 표현되는 형식으로 줄과 열의 배열을 저장하고 조작하기가 쉬운 포트란 등과 같은 프로그래밍 언어에서 조작하기 쉽고, 비용이 적게 드는 장점이 있다.

그리고, 벡터(Vector)형 데이터는 형상학적 데이터 구조의 제공과 압축된 데이터 구조, 위상의 상호 연결성, 도형과 비도형 데이터의 수정 및 일변화가 가능하다는 등의 장점이 있다.

본 시스템에서는 그림 21과 같이 하천평면도의 초기화면에서는 래스터형 자료구조로 평면도를 표현하여 실제 지형을 보는 것과 같이 지형분석을 쉽게 할 수 있도록 하였으며, 평면의 확대 시에는 벡터형 자료구조로써 각종 정보를 자세하게 조회할 수 있도록 하였다.

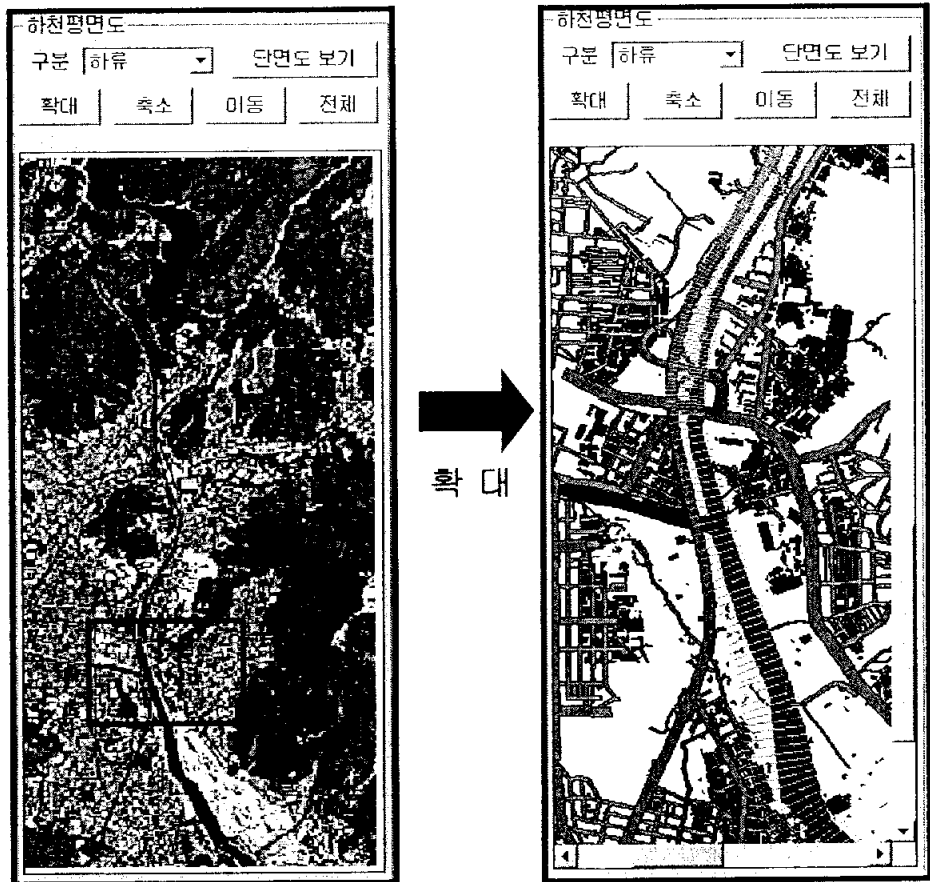


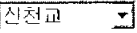
그림 21 래스터 및 벡터자료의 하천평면도

4.3.3 하천시설물

4.3.1절에서 설명한 하천단면 창과 달리 하천시설물 창은 해당유역에 존재하는 교량 등 각종 대상 시설물을 관측할 수 있도록 구축되어 있으며, 전 유역의 View Index(평면)와 빈도별 홍수위가 나타나있는 시설물의 단면도 및 각 시설물의 제원을 쉽게 알아볼 수 있도록 되어 있다.

각각의 기능을 살펴보면 다음의 표 3과 같다.

표 3 하천 시설물 창에 대한 기능설명

구 분	설 명
단면 제어	각 시설물의 단면을 검색할 수 있도록 시설명  검색창을 이용할 수 있으며 지정된 시설물의 축점이 표기된다. 또한  아이콘을 사용하여 View Index 평면 상에서 원하는 시설물에 마우스를 위치함으로써 시설물의 제원, 단면도 및 빈도별 홍수위를 쉽게 확인할 수 있다.
View Index	하천 단면 창의 확대, 축소, 이동, 전체 기능과 동일한 기능을 가지고 있으며 해당구역 상의 시설물을 알아보기 쉽게 표기하였다.
시설물 제원	단면제어 및 View Index창을 통해 지정된 시설물의 연장, 경간장, 하폭, 하단고, Pier형식과 폭, 단면계수, 빈도별 홍수위를 동시에 나타내어 관리가 용이하도록 하였다.
빈도별 홍수위	해당구역내 존재하는 시설물의 빈도별 홍수위를 시스템 사용자가 직접 선택하여 시설물단면도 및 시설물제원 창에 나타나게 할 수 있다.
시설물 단면도	지정된 시설물의 단면도 및 빈도별 홍수위를 도식화하여 재현기간에 따른 홍수위의 변화를 직접 확인할 수 있다.

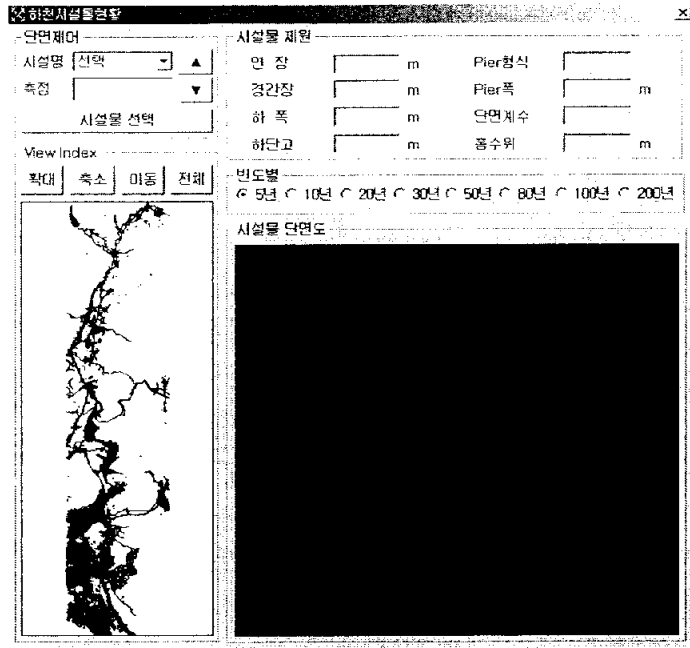


그림 22 하천시설물현황 창의 실행시 초기화면

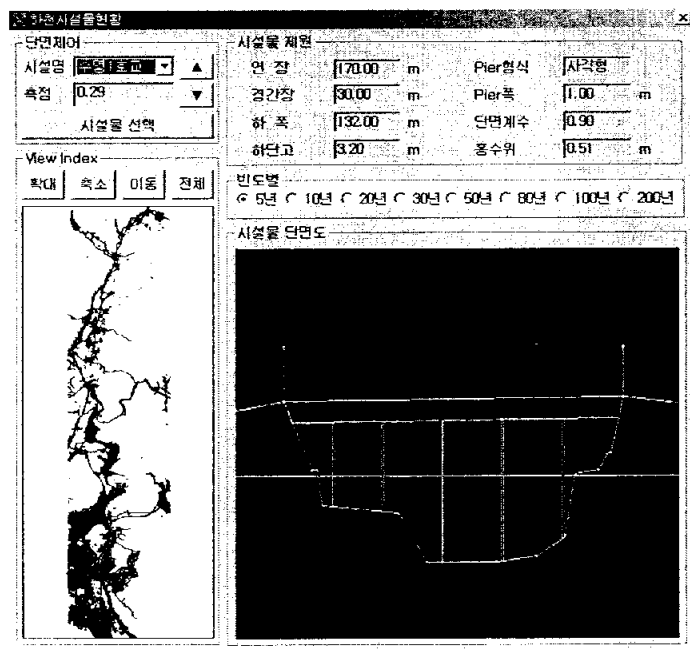


그림 23 시설물명 선택시 시설물현황 화면

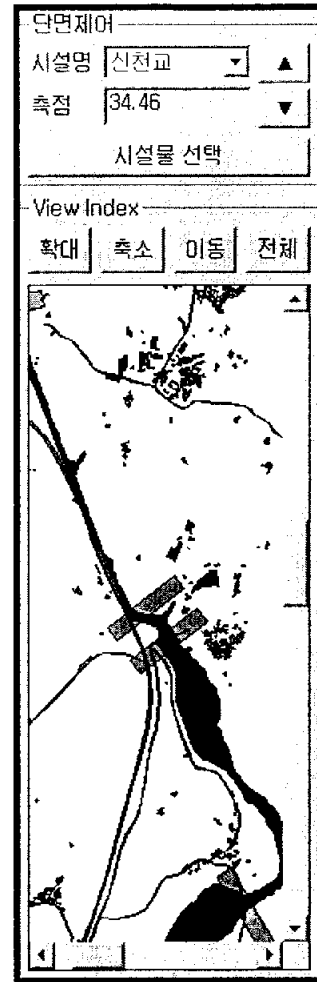
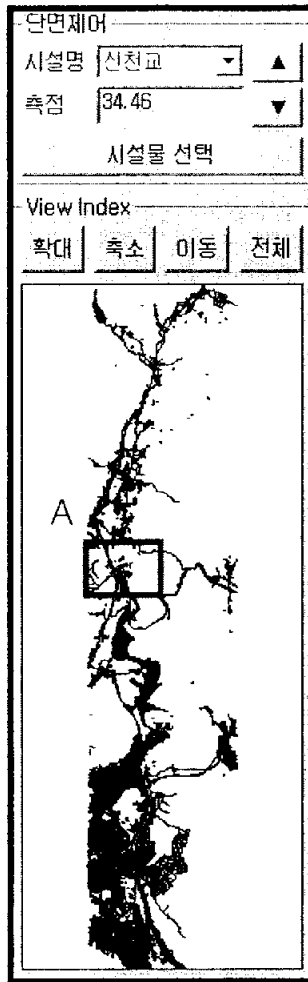


그림 24 View Index에서 구역'A'의 확대화면 실행

시설물 제원					
연 장	94.10	m	Pier형식	사각형	
경간장	16.70	m	Pier폭	1.50	m
하 폭	92.00	m	단면계수	1.25	
하단고	39.28	m	홍수위	37.74	m

그림 25 시설물 선택시 나타나는 시설물 제원

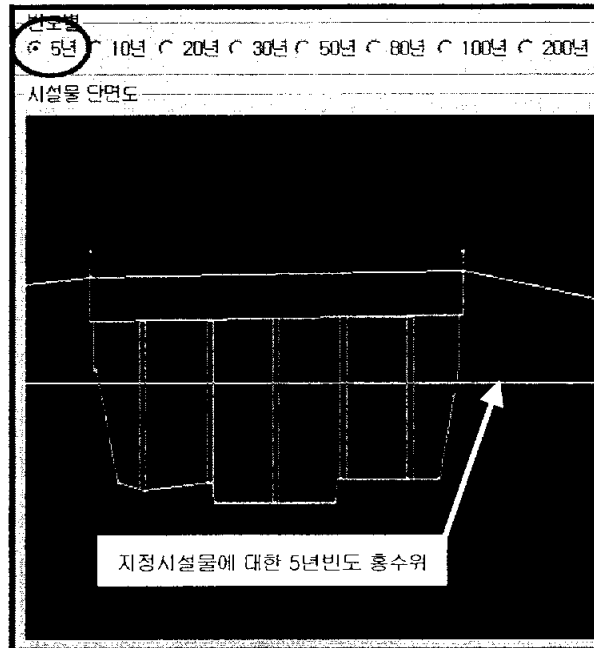


그림 26 시설물 선택시 나타나는 단면도
(5년빈도 홍수위 적용(EL. 37.74m))

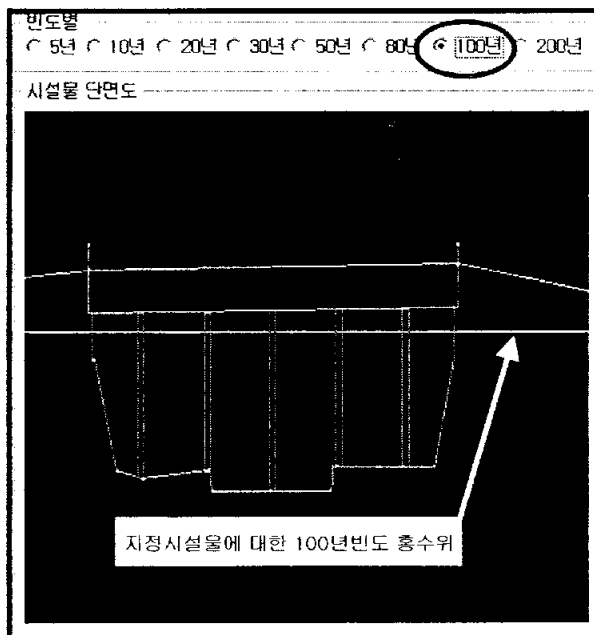


그림 27 시설물 선택시 나타나는 단면도
(100년빈도 홍수위 적용(EL. 38.78m))

이상과 같이 GIS를 이용하여 개발된 하천관리시스템의 특징은 하천을 따라서 마우스를 움직이면 화면상의 마우스 위치와 공간자료가 겹하게 됨으로써 공간정보가 가지고 있는 단면번호에 대한 정보를 DViewer에 의해서 나타내므로, 단면의 자료검토가 용이하고, 단면의 축점을 찾아서 이동할 필요 없이 지도상의 유역을 따라 마우스를 움직이기만 하면 된다는 장점이 있다.

즉, 수위에 대한 정보를 보다 더 빨리 획득할 수 있었으며 동시에 도식화 및 수치화된 빈도별 홍수위 변화와 각종 지형요소, 특성인자, 그리고 사용자가 원하는 위치의 정보를 쉽게 얻을 수 있어서 효율성이 높은 시스템이라 할 수 있다.

제 5 장 결과 및 고찰

5.1 하천관리시스템 구축

하천관리시스템을 GIS와 연계, 구축하여 해당유역의 평면도, 단면도, 위치검색 등을 편리하게 하였으며, 유역면적, 유로연장 등의 하도특성인자와 홍수위, 하상고 등의 측점별 지형요소를 동시에 표시할 수 있도록 하였으므로 사용자로 하여금 기존의 HEC RAS만을 사용할 때 보다 하천관리를 더욱 효율적으로 할 수 있도록 도와줄 것으로 예상된다.

5.2 하천시설물관리시스템을 통한 수공구조물 관리

하천시설물관리시스템을 기존의 HEC-RAS시스템과는 달리 GIS를 이용하여 평면도와 종단도, 단면도를 동시에 관리할 수 있도록 연계하였으므로 수공구조물의 위치를 신속하고 정확하게 파악할 수 있도록 하였다.

또한 이상강우시 해당 구조물에 대한 수위의 영향을 쉽게 알아볼 수 있도록 하였으므로 수공구조물 관리를 보다 안정적이고 효율적으로 할 수 있도록 시스템을 구축하였다.

5.3 시스템구축을 통한 자연재해예방

강우-유출모형을 이용한 HEC-RAS와 GIS를 연계하여 하천을 관리함으로써 홍수에 대한 효율적인 수위의 변동 및 범람예측이 가능하다.

제 6 장 결론

본 연구는 부산광역시 수영강 유역을 대상으로 GIS를 이용한 하천관리 시스템 개발 및 구축에 관한 연구로서 다음과 같은 결론을 얻을 수 있었다.

첫째, 하천관리시스템 구축으로 인하여 강우량, 유출량에 따른 대상유역 내 각 측정별, 빈도별 홍수위변화를 쉽게 알 수 있었다.

둘째, GIS를 이용하여 하천 및 시설물관리시스템을 구축함으로써 하천단면도, 평면도, 종단도를 연계시켜 HEC-RAS의 결과를 도식화함으로써 보다 효율적으로 분석할 수 있었다.

셋째, 하천의 특성에 맞게 수위의 변화 및 수리·수문인자를 쉽게 비교 분석할 수 있으므로 홍수나 집중호우로 인한 침수피해 예방 등의 하천관리에 큰 도움이 될 수 있다고 사료된다.

참고문헌

1. 건설부(1991). “수자원관리기법개발연구조사 보고서,” pp.228~244
2. 경기개발연구원(1998). “HEC-RAS를 이용한 수면곡선계산 Manual”
3. 국립방재연구소 (1998). “강우분석 프로그램의 개발.” 국립방재연구소 연구보고서, pp. 11~60
4. 김계현 (2000). “GIS 개론.” 대영출판사, pp. 2~49
5. 김동근(2001). “미계측 소유역의 유출량 산정에 관한 연구.” 부경대학교 토목공학과 석사학위논문, pp.10~17
6. 김상용(1971). “단위유량도의 특성에 관하여” 동아대학교 대학원 토목공학과, 공학석사학위논문.
7. 김하나(1997). “GIS를 이용한 대학캠퍼스 공간분석.” 경북대학교 대학원 지역정보학과 지리정보전공, pp.5~7
8. 김남원, 김창완, 우효섭 (2002). “태풍 루사(Rusa)의 강우 특성과 홍수피해 특성.” 대한토목학회 학술발표회 논문집, pp. 15~22
9. 우효섭 (2002). “홍수와 더불어 사는 사회의 구현.” 대한토목학회지 논문, pp. 4~5
10. 유복모 (2000). “지형공간정보론.” 동명출판사, pp. 3~4, 6~56
11. 윤용남, 전병호, 이종태 (1992). 도시홍수 재해방지를 위한 내수처리 시스템 분석 및 설계기법의 개발, 한국과학재단.
12. 윤태훈 (2001). “응용수문학.” 청문각, pp.249~255, 379~477, 809~882
13. 이원환 (1980). “도시하천 및 하수도 개수계획상의 계획 강우량 설정에 관한 추계학적 해석.” 대한토목학회지 논문집, 제28권, 제4호.
14. 이종태, 윤세의, 이재준, 윤용남 (1991). “도시화 영향을 고려한 유수지 계획모형.” 한국수문학회지, 제24권, 제4호, pp. 73~83.
15. 이범희 (1998). “지리정보체계 및 전문가시스템을 이용한 도시유출 및 수질보형의 개발.” 박사학위논문, 서울대학교.

16. 이상만(1999). “부산지방 확률강우강도식의 유도에 관한 연구.” 석사학위논문, 부경대학교, pp. 3~21
17. 이운태, 이도훈, 이주현 (1999). “홍수유출계산법.” 구미서관, pp. 18~105
18. 정인주, 이정민, 김상용, 이종출 (2002). “DEM에 의한 지형인자 추출에 따른 격자크기의 영향에 관한 연구.” 한국지형공간정보학회 논문집, 제10권 제2호
19. 정인주 (2003). “3차원 Virtual GIS를 이용한 도시하천관리시스템 구축.” 박사학위 논문, 부경대학교.
20. 최양범 (2001). “수자원공학.” 창원출판사, pp.221~238, 717~743
21. 한동대학교(2002). “GIS의 기초과정.” 한동대학교 GIS 연구소, pp.12~13
22. 한국건설기술연구원 (1995). HEC-2 수면곡선 계산프로그램 (사용자 설명서).
23. 한국수자원공사 (1993). HEC-2 모델에 의한 하천수면형 계산.
24. 한국수문학회(1988). 도시수문학, 1988년도 도시수문학 심포지움 논문집.
25. 한국토지개발공사(1989). 도시 소규모 단지의 우수유출량 산정기법에 관한 연구, 한국건설기술연구원.
26. 한국토지공사 (1996). 도시수문 및 우수관리 기술 세미나 논문집. 한국수자원학회.
27. http://krsc.kari.re.kr/sub/education/edu_02_5_3_3.htm
28. Bedient, P.B. and Huber, W.C. (1998). *Hydrology and Floodplain Analysis*. Addison-Wesley Publishing Company, pp. 225-230.
29. Bras, R.L. and Rodriguez-Iturbe, I. (1985). *Random Functions and Hydrology*. Addison-Wesley, pp. 92-47.
30. Chow, V.T., Maidment, D.R. and Mays, L.W. (1988). *Applied*

Hydrology. McGraw Hill.

31. Chow, V.T. (1959). *Open Channel Hydraulics*. McGraw-Hill.
32. Debo, T.N. and Reese, A.J. (1995). *Municipal Storm Water Management*. pp. 209-220.
33. Draper, N.R. and Smith H. (1981). *Applied Regression Analysis*. John Wiley & Sons, New York.
34. Hall, M.J., Hockin, D.L. and Ellis, J.B. (1993). *Design of Flood Storage Reservoirs*. CIRIA and Butterworth-Heinemann Ltd., p. 73.
35. Helsel, D. R. and Hirsch, R.M. (1992). *Statistical Methods in Water Resources*. Elsevier Science Publishers.
36. Hudson, N. W. (1971). *Soil Conservation*. Badsford, London.
37. Huff, F. A. (1967). "Time Distribution of Rainfall in Heavy Storms." *Water Resour. Res.*, Vol. 3, No. 4, pp. 1007-1019.
38. Kottegota, N.T. (1980). *Stochastic Water Resources Technology*. Macmillan Press.
39. Roesmer, L.A. and Yevjevich, V. (1996). *Mathematical Models for Time Series of Monthly Precipitation and Monthly Runoff*. Hydrology Paper NO. 15, Colorado State University, Fort Collins, Colorado.
40. Singh, Vihay P. (1992). *Elementary Hydrology*. Prentice Hall.

부 록

```

// DlgSection.cpp : implementation file
//
#include "stdafx.h"
#include "RiverSystem.h"
#include "DlgSection.h"
#include "EtcFunction.h"
#include "modataconnection.h"
#include "molayers.h"
#include "moimagelayer.h"
#include "momaplayer.h"
#include "mogeodataset.h"
#include "mosymbol.h"
#include "morectangle.h"
#include "momaplayer.h"
#include "MapObjects2.h"
#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif
//////////////////////////////////////
// CDlgSection dialog
CDlgSection::CDlgSection(CWnd* pParent /*=NULL*/)
    : CDialog(CDlgSection::IDD, pParent)
{
    //{{AFX_DATA_INIT(CDlgSection)
    //}}AFX_DATA_INIT
}

void CDlgSection::DoDataExchange(CDataExchange* pDX)
{
    CDialog::DoDataExchange(pDX);
    //{{AFX_DATA_MAP(CDlgSection)
    DDX_Control(pDX, IDC_EBArea1, m_ebArea);
    DDX_Control(pDX, IDC_EBLength1, m_ebLength);
    DDX_Control(pDX, IDC_EBY, m_ebY);
    DDX_Control(pDX, IDC_EBX, m_ebX);
    DDX_Control(pDX, IDC_EBVelo, m_ebVelo);
    DDX_Control(pDX, IDC_EBSlope, m_ebSlope);
    DDX_Control(pDX, IDC_EBRougNo, m_ebRougNo);
    DDX_Control(pDX, IDC_EBQuan, m_ebQuan);
    DDX_Control(pDX, IDC_EBEle, m_ebEle);
    DDX_Control(pDX, IDC_EBBasinArea, m_ebBasinArea);
    DDX_Control(pDX, IDC_EBBottom, m_ebBottom);
    DDX_Control(pDX, IDC_CMBPoint, m_cmbPoint);
    DDX_Control(pDX, IDC_CMBDist, m_cmbDist);
    DDX_Control(pDX, IDC_MAPIndex, m_cMapI);
    DDX_Control(pDX, IDC_FGRIDFlood, m_fgriDFlood);
    DDX_Control(pDX, IDC_DVIEWSection, m_DViewSection);
    DDX_Control(pDX, IDC_DVIEWDivision, m_DViewDivision);
    //}}AFX_DATA_MAP
}

BEGIN_MESSAGE_MAP(CDlgSection, CDialog)
    //{{AFX_MSG_MAP(CDlgSection)
    ON_BN_CLICKED(IDC_BUTTONZoomin, OnBUTZoomin)
    ON_BN_CLICKED(IDC_BUTTONZoomout, OnBUTZoomout)
    ON_BN_CLICKED(IDC_BUTTONPan, OnBUTPan)

```

```

        ON_BN_CLICKED(IDC_BUTAll, OnBUTAll)
        ON_BN_CLICKED(IDC_BUTSecview, OnBUTSecview)
        ON_WM_CLOSE()
        ON_CBN_SELCHANGE(IDC_CMBPoint, OnSelchangeCMBPoint)
        ON_BN_CLICKED(IDC_BUTUp, OnBUTUp)
        ON_BN_CLICKED(IDC_BUTDown, OnBUTDown)
        ON_BN_CLICKED(IDC_BUTEnd, OnBUTEnd)
        ON_CBN_SELCHANGE(IDC_CMBDist, OnSelchangeCMBDist)
        //}}AFX_MSG_MAP
    END_MESSAGE_MAP()

    //////////////////////////////////////
    // CDlgSection message handlers

void CDlgSection::OnClose()
{
    m_DViewSection.LayerClear();

    m_DViewDivision.LayerClear();

    CDialog::OnClose();
}

BOOL CDlgSection::OnInitDialog()
{
    CDialog::OnInitDialog();
    SetIcon( AfxGetApp()->LoadIcon(IDR_TITLELOGO), FALSE );
    ReadREFPFile();
    InitControl();
    //MapObject와 control
    InitMapObjects();

    return TRUE; // return TRUE unless you set the focus to a control
               // EXCEPTION: OCX Property Pages should return FALSE
}

void CDlgSection::InitControl()
{
    //combo box 초기화
    InitCombo();
    //FlexGrid 초기화
    InitFGrid();
}

void CDlgSection::InitCombo()
{
    m_cmbDist.InsertString(0,"상류");
    m_cmbDist.InsertString(1,"하류");
    m_cmbDist.SetCurSel(1);
    //측점 combo box 초기 지정
    CString strTemp;
    m_cmbDist.GetWindowText(strTemp);
    if(strTemp == _T("상류"))
        strTemp = _T("Upper Reach ");
    else
        strTemp = _T("Lower Reach ");

    m_cmbPoint.AddString("선택");
}

```

```

CString strLayerName;
int nIndex = 1;
int nLayerCount = m_DViewSection.GetLayerCount();
for(int i = 0 ; i < nLayerCount ; i++){

    strLayerName = m_DViewSection.GetIndexToLayerName(i);
    strLayerName = strLayerName.Left(strLayerName.GetLength() - 1);
    if(strLayerName.Find(strTemp,0) != -1){
        strLayerName.Replace(strTemp,"");
        m_cmbPoint.InsertString(nIndex,strLayerName);
        nIndex++;
    }
}

m_cmbPoint.SetCurSel(0);//setcursel은 초기에 콤보박스의 값을 나타낼 셀을 지정해준다.
}

//Combo Box에서 상류, 하류를 선택할 때 띄우는 레이어의 변환
void CDlgSection::OnSelchangeCMBDist()
{
    InitMapObjects();

    //축점 combo box 초기 지정
    CString strTemp;
    m_cmbDist.GetWindowText(strTemp);
    if(strTemp == _T("상류")){
        strTemp = _T("Upper Reach ");
        m_DViewDivision.SetDYScale(50);
    }else{
        strTemp = _T("Lower Reach ");
        m_DViewDivision.SetDYScale(150);
    }

    m_cmbPoint.AddString("선택");

    CString strLayerName;
    int nIndex = 1;
    int nLayerCount = m_DViewSection.GetLayerCount();
    for(int i = 0 ; i < nLayerCount ; i++){

        strLayerName = m_DViewSection.GetIndexToLayerName(i);
        strLayerName = strLayerName.Left(strLayerName.GetLength() - 1);
        if(strLayerName.Find(strTemp,0) != -1){
            strLayerName.Replace(strTemp,"");
            m_cmbPoint.InsertString(nIndex,strLayerName);
            nIndex++;
        }
    }

    m_cmbPoint.SetCurSel(0);//setcursel은 초기에 콤보박스의 값을 나타낼 셀을 지정해준다.
}

void CDlgSection::InitFGrid()
{
    m_fgridFlood.Clear();

    m_fgridFlood.SetCols(2);
    m_fgridFlood.SetRows(9);
}

```

```

        m_fgridFlood.SetTextMatrix(0,0,"재현기간");
        m_fgridFlood.SetTextMatrix(1,0,"5년");
        m_fgridFlood.SetTextMatrix(2,0,"10년");
        m_fgridFlood.SetTextMatrix(3,0,"20년");
        m_fgridFlood.SetTextMatrix(4,0,"30년");
        m_fgridFlood.SetTextMatrix(5,0,"50년");
        m_fgridFlood.SetTextMatrix(6,0,"80년");
        m_fgridFlood.SetTextMatrix(7,0,"100년");
        m_fgridFlood.SetTextMatrix(8,0,"200년");
        m_fgridFlood.SetTextMatrix(0,1,"홍수위");
        m_fgridFlood.SetColWidth(1,900); // 반납 일자
    }

//layer 띄우는 코딩.
void CDlgSection::InitMapObjects()
{
    // 모든 Layer를 화면에서 메모리에서 해제 시킨다.
    LayerRemoveAll();

    CEtcFunction EtcFunc;
    CString strExeFolder = EtcFunc.GetExeFolder();
    CString strFileName;

    int i = m_cmbDist.GetCurSel();

    switch (i){
        case 1:{
            strFileName = strExeFolder + _T("\\data\\shape\\section\\TM_busan_low.tif");
            AddImageLayer(m_cMap1, strFileName);

            strFileName = strExeFolder + _T("\\data\\shape\\section\\river.shp");
            AddLayer(m_cMap1, strFileName, moBlue);

            strFileName = strExeFolder + _T("\\data\\shape\\section\\build.shp");
            AddLayer(m_cMap1, strFileName, moPurple);

            strFileName = strExeFolder + _T("\\data\\shape\\section\\buf_road.shp");

            AddLayer(m_cMap1, strFileName, moGray);

            strFileName = strExeFolder + _T("\\data\\shape\\section\\l_section.shp");

            AddLayer(m_cMap1, strFileName, moGreen);

            strFileName = strExeFolder + _T("\\data\\shape\\section\\low_poly.shp");

            AddLayer(m_cMap1, strFileName, moBlack);

            ThemeLayerOff();

        }
        break;

        case 0:{
            strFileName = strExeFolder + _T("\\data\\shape\\section\\TM_busan_up.tif");
            AddImageLayer(m_cMap1, strFileName);
        }
    }
}

```

```

        strFileName = strExeFolder + _T("\\data\\shape\\section\\u_river.shp");
        AddLayer(m_cMap1, strFileName, moBlue);

        strFileName = strExeFolder + _T("\\data\\shape\\section\\build_u.shp");
        AddLayer(m_cMap1, strFileName, moPurple);

        strFileName = strExeFolder + _T("\\data\\shape\\section\\buf_roadu.shp");
        AddLayer(m_cMap1, strFileName, moGray);

        strFileName = strExeFolder + _T("\\data\\shape\\section\\u_section.shp");
        AddLayer(m_cMap1, strFileName, moGreen);

        strFileName = strExeFolder + _T("\\data\\shape\\section\\up_poly.shp");
        AddLayer(m_cMap1, strFileName, moBlack);

        ThemeLayerOff();
    }
    break;
}
CMoRectangle extent(m_cMap1.GetFullExtent());
m_cMap1.SetExtent(extent);
}

void CDlgSection::LayerRemoveAll()
{
    CMoLayers layers(m_cMap1.GetLayers());
    layers.Clear();
}

void CDlgSection::AddLayer(CMap1& map, const CString& path, COLORREF color)
{
    // Establish connection to data
    CMoDataConnection conn;
    if (!conn.CreateDispatch(TEXT("MapObjects2.DataConnection")))
        throw "unable to create MapObjects2.DataConnection";
    conn.SetDatabase(GetFileDirectory(path));
    if (!conn.Connect())
        throw "unable to connect to database";

    // Add layer specified by path
    CMoLayers layers(map.GetLayers());
    CMoMapLayer layer;
    if (!layer.CreateDispatch(TEXT("MapObjects2.MapLayer")))
        throw "unable to create MapObjects2.MapLayer";

    CMoGeoDataset geoDataset(conn.FindGeoDataset(GetFileTitle(path)));
    layer.SetGeoDataset(geoDataset);
    if (color != -1) // Set color if specified
    {
        CMoSymbol layerSymbol(layer.GetSymbol());
        layerSymbol.SetColor(color);
    }
}

```

```

        layers.Add(layer);
    }

void CDlgSection::AddImageLayer(CMap1& map, const CString& path)
{
    // Establish connection to data
    CMoDataConnection conn;
    if (!conn.CreateDispatch(TEXT("MapObjects2.DataConnection")))
        throw "unable to create MapObjects2.DataConnection";

    conn.SetDatabase(GetFileDirectory(path));
    if (!conn.Connect())
        throw "unable to connect to database";

    // Add image layer
    CMoLayers layers(map.GetLayers());
    CMoImageLayer image;
    if (!image.CreateDispatch(TEXT("MapObjects2.ImageLayer")))
        throw "unable to create MapObjects2.ImageLayer";
    image.SetFile(path);
    layers.Add(image);
}

CString CDlgSection::GetFileDirectory(const CString& path)
{
    ASSERT(path.GetLength());
    int pos = path.ReverseFind('\\');
    if (pos >= 0)
        return path.Left(pos);
    return "";
}

CString CDlgSection::GetFileName(const CString& path)
{
    ASSERT(path.GetLength());
    CString strResult = GetFileName(path);
    int pos = strResult.ReverseFind('.');
    if (pos >= 0)
        return strResult.Left(pos);

    return strResult;
}

CString CDlgSection::GetFileName(const CString& path)
{
    ASSERT(path.GetLength());
    int pos = path.ReverseFind('\\');
    if (pos >= 0)
        return path.Right(path.GetLength() - pos - 1);
    return "";
}

void CDlgSection::ReadREPFile()
{
    m_DViewSection.SetBRepFlag(false);
    m_DViewSection.SetDYScale(10);
}

```

```

CEtcFunction EtcFunc;
CString strExeFolder = EtcFunc.GetExeFolder();
CString strFileName;

strFileName = strExeFolder + _T("\\data\\sy.rep");

m_DViewSection.SetDgnFileName(_T(strFileName));
m_DViewSection.DGNFileRead();

AllLayerOffSection();

m_DViewDivision.SetBRepFlag(true);
m_DViewDivision.SetDYScale(150);
m_DViewDivision.SetDgnFileName(_T(strFileName));
m_DViewDivision.DGNFileRead();

AllLayerOffDivision();
}

/*****
void CDlgSection::OnBUTZoomin()
{
    //확대
    m_curTool=moZoomIn;
    m_cMap1.SetMousePointer(moZoomIn);
}

void CDlgSection::OnBUTZoomout()
{
    //축소
    m_curTool=moZoomOut;
    m_cMap1.SetMousePointer(moZoomOut);
}

void CDlgSection::OnBUTPan()
{
    //이동
    m_curTool=moPan;
    m_cMap1.SetMousePointer(moPan);
}

void CDlgSection::OnBUTAll()
{
    //전체보기
    CMoRectangle rect(m_cMap1.GetFullExtent());
    m_cMap1.SetExtent(rect.m_lpDispatch);
    m_cMap1.SetMousePointer(moDefault);
}

void CDlgSection::OnBUTSecview()
{
    //단면보기
    m_curTool=moIdentify;
    m_cMap1.SetMousePointer(moIdentify);
}

BEGIN_EVENTSINK_MAP(CDlgSection, CDialog)

```

```

//{{AFX_EVENTSINK_MAP(CDlgSection)
    ON_EVENT(CDlgSection, IDC_MAPIndex, -605 /* MouseDown */,
OnMouseDownMAPIndex, VTS_I2 VTS_I2 VTS_I4 VTS_I4)
    ON_EVENT(CDlgSection, IDC_MAPIndex, 2 /* BeforeLayerDraw */,
OnBeforeLayerDrawMAPIndex, VTS_I2 VTS_I4)
    ON_EVENT(CDlgSection, IDC_MAPIndex, -606 /* MouseMove */,
OnMouseMoveMAPIndex, VTS_I2 VTS_I2 VTS_I4 VTS_I4)
    ON_EVENT(CDlgSection, IDC_DVIEWSection, -606 /* MouseMove */,
OnMouseMoveDVIEWSection, VTS_I2 VTS_I2 VTS_I4 VTS_I4)
    ON_EVENT(CDlgSection, IDC_FGRIDFlood, -600 /* Click */, OnClickFGRIDFlood,
VTS_NONE)
    ON_EVENT(CDlgSection, IDC_DVIEWSection, 5 /* DrawAfter */,
OnDrawAfterDVIEWSection, VTS_NONE)
    ON_EVENT(CDlgSection, IDC_DVIEWDivision, 5 /* DrawAfter */,
OnDrawAfterDVIEWDivision, VTS_NONE)
    ON_EVENT(CDlgSection, IDC_DVIEWDivision, -606 /* MouseMove */,
OnMouseMoveDVIEWDivision, VTS_I2 VTS_I2 VTS_I4 VTS_I4)
//}}AFX_EVENTSINK_MAP
END_EVENTSINK_MAP()

```

```

void CDlgSection::OnMouseDownMAPIndex(short Button, short Shift, long X, long Y)
{

```

```

    switch (m_curTool)
    {
        case moZoomIn:
        {
            CMoRectangle r(m_cMap1.TrackRectangle());
            m_cMap1.SetExtent(r);
            break;
        }
        case moZoomOut:
        {
            CMoRectangle r(m_cMap1.GetExtent());
            r.ScaleRectangle(1.5);
            m_cMap1.SetExtent(r);
            break;
        }
        case moPan:
        {
            m_cMap1.Pan();
            break;
        }
        case moIdentify: // 단면도 정보 보기
            GetSectionInfo(X,Y);
            break;
    }
}

```

//다음 코딩은 화면제어를 맵 상에서 수행할 수 있도록 한다. Mouse Move이 단면도를 자동으로 찾는다.

```

void CDlgSection::OnMouseMoveMAPIndex(short Button, short Shift, long X, long Y)
{
    switch (m_curTool)
    {
        case moIdentify: // 단면도 정보 보기
            GetSectionInfo(X,Y);

```

```

        break;
    }
}

/*****
*****/
void CDlgSection::ThemeDisplay()
{
    ThemeLayerOff();
    m_cMap1.Refresh();
}
//레이어를 오프시킨다.초기에 모두를 오프시킨다.
void CDlgSection::ThemeLayerOff()
{
    CMoLayers layers(m_cMap1.GetLayers());
    CMoMapLayer layer;
    int nLayerCount = layers.GetCount();

    for(short i = 0 ; i < nLayerCount ; i++)
    {
        layer = layers.Item(COLEVariant(i));
        layer.SetVisible(false);
    }
    ThemeLayerOn(5);
}
void CDlgSection::ThemeLayerOn(short nIndex)
{
    CMoLayers layers(m_cMap1.GetLayers());
    CMoMapLayer layer;
    layer = layers.Item(COLEVariant(nIndex));
    layer.SetVisible(true);
}

void CDlgSection::OnBeforeLayerDrawMAPIndex(short index, long hDC)
{
    CMoRectangle extent(m_cMap1.GetExtent());
    CMoRectangle fullExtent(m_cMap1.GetFullExtent());

    if (extent.GetWidth() <= (fullExtent.GetWidth() / 3.0))
    {
        ThemeLayerOn(4),ThemeLayerOn(3),ThemeLayerOn(2),ThemeLayerOn(1),ThemeLayerOn(0);

        CMoLayers layers(m_cMap1.GetLayers());
        CMoImageLayer ilayer;
        ilayer = layers.Item(COLEVariant(index));
        ilayer.SetVisible(false);
    }

    else
    {
        CMoLayers layers(m_cMap1.GetLayers());
        CMoMapLayer layer;
        CMoImageLayer ilayer;
        ilayer = layers.Item(COLEVariant(index));
        ilayer.SetVisible(true);
    }
}

```

```

        ThemeLayerOff();
        ThemeLayerOn(7),ThemeLayerOn(8);
        ThemeDisplay();

    }

}

void CDlgSection::GetSectionInfo(long X, long Y)
{
    CMoTableDesc tdesc;
    CMoFields flds;
    CMoField fld;
    CMoRecordset recs;
    CMoMapLayer l;
    CMoPoint p;
    CMoLayers layers;

    layers.CreateDispatch(TEXT("MapObject2.Layers"));
    layers = m_cMap1.GetLayers();

    l.CreateDispatch(TEXT("MapObject2.Layer"));
    l = layers.Item(COleVariant(short (1)));

    long nLayerType = l.GetShapeType();
    CString strLayerName = l.GetName();

    p.CreateDispatch(TEXT("MapObject2.Point"));
    p = m_cMap1.ToMapPoint((float)X, (float)Y);

    recs.CreateDispatch(TEXT("MapObject2.Recordset"));
    recs = l.SearchByDistance((IDispatch *)p, 10, "");

    if ( !recs.GetEof())
    {
        tdesc.CreateDispatch(TEXT("MapObject2.TableDesc"));
        tdesc = recs.GetTableDesc();
        flds.CreateDispatch(TEXT("MapObject2.Fields"));
        flds = recs.GetFields();

        fld.CreateDispatch(TEXT("MapObject2.Field"));

        CString strFieldName, strValue;
        for(short i = 0 ; i < tdesc.GetFieldCount() ; i++){

            fld = flds.Item(COleVariant(TEXT(tdesc.GetFieldName(i))));

            strFieldName = fld.GetName();
            strValue = fld.GetValueAsString();

            if(strFieldName.Find(_T("Section"),0) != -1)
                break;

        }

        // 단면번호 찾았다.
        DisplaySection(strValue);
    }
}

```

```

        DisplayDivision(strValue,1);

    } else {

    }

}

// 단면도 보기
void CDlgSection::DisplaySection(CString strSectionNum)
{
    m_DViewSection.SetJijFunc(true);
    CString strLayerName = MakeLayerNameSection(strSectionNum);

    AllLayerOffSection();

    int nIndex = m_DViewSection.GetLayerNameToIndex(strLayerName);
    if(nIndex == -1)
        return;

    m_DViewSection.SetLayerOn(nIndex);

    // 년도별 단면정보를 화면에 Display 한다.
    DisplaySectionInfo(strLayerName,1);

    // 년도별 정보를 Grid에 나타낸다.
    DisplayGridData(strLayerName);
}

CString CDlgSection::MakeLayerNameSection(CString strSectionNum)
{
    CString strULFlag;
    m_cmbDist.GetWindowText(strULFlag);

    if(strULFlag == _T("상류"))
        strULFlag = _T("Upper");
    else
        strULFlag = _T("Lower");

    CString strLayerName;
    strLayerName.Format("%s Reach %sL",strULFlag,strSectionNum);

    return strLayerName;
}

void CDlgSection::AllLayerOffSection()
{
    int nLayerCount = m_DViewSection.GetLayerCount();

    for(int i = 0 ; i < nLayerCount ; i++)
        m_DViewSection.SetLayerOff(i);
}

void CDlgSection::AllLayerOffDivision()
{
    int nLayerCount = m_DViewDivision.GetLayerCount();

    for(int i = 0 ; i < nLayerCount ; i++)

```

```

        m_DViewDivision.SetLayerOff(i);
    }

    // 년도별 단면정보를 화면에 Display 한다.
    void CDlgSection::DisplaySectionInfo(CString strSectionNum,int nIndex)
    {
        if(strSectionNum.Right(1) == _T("L"))
            strSectionNum = strSectionNum.Left(strSectionNum.GetLength() - 1);

        // 하도특성인자 Display;
        DisplayCSFInfo();

        //홍수위
        m_ebEle.SetWindowText(m_DViewSection.GetJijFuncEle(strSectionNum,nIndex));
        //하상고
        m_ebBottom.SetWindowText(m_DViewSection.GetJijFuncBottom(strSectionNum,nIndex));
        //조도계수

m_ebRougNo.SetWindowText(m_DViewSection.GetJijFuncRougNo(strSectionNum,nIndex));
        //홍수량
        m_ebQuan.SetWindowText(m_DViewSection.GetJijFuncQuan(strSectionNum,nIndex));
        //평균유속
        m_ebVelo.SetWindowText(m_DViewSection.GetJijFuncVelo(strSectionNum,nIndex));
        //단면적
        m_ebArea.SetWindowText(m_DViewSection.GetJijFuncArea(strSectionNum,nIndex));

        m_strLayerNameSection = strSectionNum;
        m_nSectionIndex = nIndex;

        // 모든 데이터를 나타낸 후에 다시 그린다.
        m_DViewSection.ViewControlReDraw();
    }

    // 화면이 이상하여 다 그리고 난 후에 그린다.
    void CDlgSection::OnDrawAfterDVIEWSection()
    {
        m_DViewSection.JijFuncDrawEle(m_strLayerNameSection,m_nSectionIndex);
    }

    // 하도특성인자 Display;
    void CDlgSection::DisplayCSFInfo()
    {
        m_ebBasinArea.SetWindowText(_T("199.57"));
        m_ebLength.SetWindowText(_T("28.4"));
        m_ebSlope.SetWindowText(_T("0.25"));
    }

    // 마우스 mouse시 적용된다.
    void CDlgSection::OnMouseMoveDVIEWSection(short Button, short Shift, long X, long Y)
    {
        CString strX,strY;

        strX.Format("%.2f",m_DViewSection.ScreenToMapX(X));
        strY.Format("%.2f",m_DViewSection.ScreenToMapY(Y));

        m_ebX.SetWindowText(strX);
        m_ebY.SetWindowText(strY);
    }

```

```

}

void CDlgSection::DisplayGridData(CString strLayerName)
{
    if(strLayerName.Right(1) == _T("L"))
        strLayerName = strLayerName.Left(strLayerName.GetLength() - 1);

    // 그리드에 반영된 년도별 데이터를 적용한다.
    for(int i = 0 ; i < 8 ; i++)

m_fgridFlood.SetTextMatrix(i+1,1,m_DViewSection.GetJijFuncEle(strLayerName,i+1));
}

// 그리드 클릭 했을 때....
void CDlgSection::OnClickFGRIDFlood()
{
    if(! DataGridSelectCheck())
        return;

    int nRowIndex = m_fgridFlood.GetRowSel();

    DisplaySectionInfo(m_strLayerNameSection,nRowIndex);

    // 종단도의 홍수위를 조절한다.
    DisplayDivision(_T(""),nRowIndex);
}

bool CDlgSection::DataGridSelectCheck()
{
    if((m_fgridFlood.GetRowSel() < 1) || (m_fgridFlood.GetRows() < 2))
        return false;

    if(m_fgridFlood.GetRow() != m_fgridFlood.GetRowSel()){

        AfxMessageBox("단일 행을 선택해 주십시오.");
        return false;
    }

    if(m_fgridFlood.GetRowSel() < 1){

        AfxMessageBox("데이터를 선택해 주십시오.");
        return false;
    }

    return true;
}

// 종단도 보기
void CDlgSection::DisplayDivision(CString strSectionNum,int nIndex)
{
    m_DViewDivision.SetJijFunc(true);
    CString strLayerName = MakeLayerNameDivision(strSectionNum);

    AllLayerOffDivision();

    int Index = m_DViewDivision.GetLayerNameToIndex(strLayerName);
    if(Index == -1)

```

```

        return;

        m_DViewDivision.SetLayerOn(Index);

        // 홍수위 표현
        DisplayDivisionYear(nIndex);
    }

CString CDlgSection::MakeLayerNameDivision(CString strSectionNum)
{
    CString strULFlag;
    m_cmbDist.GetWindowText(strULFlag);

    if(strULFlag == _T("상류"))
        strULFlag = _T("Upper");
    else
        strULFlag = _T("Lower");

    CString strLayerName;
    strLayerName.Format("%s BottomL",strULFlag);

    return strLayerName;
}

CString CDlgSection::MakeLayerNameDivision(int nIndex)
{
    CString strULFlag;
    m_cmbDist.GetWindowText(strULFlag);

    if(strULFlag == _T("상류"))
        strULFlag = _T("Upper");
    else
        strULFlag = _T("Lower");

    CString strLayerName;
    strLayerName.Format("%s%dL",strULFlag,nIndex);

    return strLayerName;
}

void CDlgSection::DisplayDivisionYear(int nIndex)
{
    CString strLayerName = MakeLayerNameDivision(nIndex);

    int Index = m_DViewDivision.GetLayerNameToIndex(strLayerName);
    if(Index == -1)
        return;

    m_DViewDivision.SetLayerOn(Index);

    m_DViewDivision.ViewControlReDraw();
}

void CDlgSection::OnDrawAfterDVIEWDivision()
{
    m_DViewDivision.JijFuncDrawDivision(m_strLayerNameSection,m_nSectionIndex);
}

```

```

void CDlgSection::OnMouseMoveDVIEWDivision(short Button, short Shift, long X, long Y)
{
    CString strX,strY;

    strX.Format("%.2f",m_DViewDivision.ScreenToMapX(X));
    strY.Format("%.2f",m_DViewDivision.ScreenToMapY(Y));

    m_cbX.SetWindowText(strX);
    m_cbY.SetWindowText(strY);

}

void CDlgSection::OnSelchangeCMBPoint()
{
    if(m_cmbPoint.GetCurSel() == 0)
        return;

    CString strValue;
    m_cmbPoint.GetWindowText(strValue);

    DisplaySection(strValue);

    DisplayDivision(strValue,1);
}

void CDlgSection::OnBUTUp()
{
    int nCount = m_cmbPoint.GetCount();
    int nIndex = m_cmbPoint.GetCurSel();

    if(nCount == nIndex)
        return;

    nIndex = nIndex + 1;

    m_cmbPoint.SetCurSel(nIndex);

    OnSelchangeCMBPoint();
}

void CDlgSection::OnBUTDown()
{
    if(m_cmbPoint.GetCurSel() == 0)
        return;

    int nIndex = m_cmbPoint.GetCurSel() - 1;

    m_cmbPoint.SetCurSel(nIndex);

    OnSelchangeCMBPoint();
}

void CDlgSection::OnBUTEnd()
{
    m_DViewSection.LayerClear();
}

```

```
m_DViewDivision.LayerClear();  
  
// 윈도우 종료  
CDialog::OnCancel();  
}
```

감사의 글

먼저 이 논문을 완성하기까지 많은 지도와 격려를 아끼지 않으시며 학부시절부터 열정으로 학문의 길로 이끌어 주신 김상용 지도교수님께 고개 숙여 깊은 감사의 뜻을 올립니다.

또한 바쁘신 와중에서도 본 논문이 완성되기까지 많은 지도와 심사를 하여 주시고 세심한 배려를 아끼지 않으신 이영대 교수님, 이종출 교수님, 그리고 이상호 교수님께 진심으로 감사드립니다.

정두희 교수님, 장희석 교수님, 국승규 교수님, 이종섭 교수님, 이동욱 교수님, 김명식 교수님, 정진호 교수님, 이환우 교수님, 김종수 교수님, 손인식 교수님께도 감사드립니다.

짧지 않은 시간동안 본 논문을 위하여 불철주야 성심껏 도와주신 환경수리연구실 직장생활의 바쁘신 중에서도 성심으로 도와주신 정인주 박사님, 이정민씨, 채종훈, 육승우, 조희호 군께도 진심으로 감사의 말씀을 전합니다. 그리고 항상 바쁜 건설현장의 업무에도 부하직원의 어려움을 이해해주시고 격려를 아끼지 않으시며, 학업을 위해 많은 배려를 해주신 현대건설(주) 남향대교현장의 김현일 소장님, 고정훈 공무부장님, 도영수 공사부장님, 김문기 관리부장님, 김현필 공무과장님외 맘 흘리며 같이 동고동락하는 동료직원 여러분께도 깊은 감사를 드립니다.

무엇보다도 기나긴 세월동안 저를 위해 모든 것을 희생하며 깊은 사랑으로 힘과 용기를 주셨던 어머니께 이 논문을 바치며, 사위로서 제대로 모시지도 못한 장인어른과 장모님께도 감사의 마음을 전하며 내가 어떠한 일을 하든 항상 곁에서 말없이 믿으며 편안함과 따뜻함을 주었던 사랑하

는 아내에게 고마움과 미안함을 같이 전하며, 누나, 동생 모든 가족과 이 기쁨을 나누고 싶습니다.

그리고, 논문을 완성하기까지 바쁜 친구의 저녁시간을 틈틈이 술잔으로 격려해준 태형, 재민, 신조외 여러 친구들에게도 고맙다는 말을 전합니다.

끝으로 저에게 사랑과 격려를 아끼지 않으시고 도움을 주신 모든 분들에게 다시 한번 고개 숙여 감사를 드립니다.

2004년 6월

김재호 올림