

공학석사 학위논문

GIS기반에서 댐 저류량 산정시스템 개발

지도교수 김 상 용



2004년 8월

부경대학교 산업대학원

토목공학과

김 선 용

이 논문을 김선용의 공학석사
학위논문으로 인준함

2004년 6월 일

주	심	공 학 박 사	손 인 식	
위	원	공 학 박 사	이 환 우	
위	원	공 학 박 사	김 상 용	

목 차

표 목차	vi
그림 목차	vi
Abstract	viii
제 1 장 서론	1
1.1 연구배경 및 목적	1
1.2 연구동향	3
1.3 연구방법	5
제 2 장 GIS의 기반에서 지형의 수치화	7
2.1 지형의 수치화 및 체적의 산정	7
2.1.1 양단면 평균법에 의한 체적의 산정	7
2.1.2 DTM 보간기법에 의한 체적의 산정	8
2.1.3 TIN 데이터구조에 의한 체적의 산정	8
2.1.4 DEM기법에 의한 체적의 산정	9
2.2 개발 식에 의한 체적의 산정	10
2.2.1 델로니 삼각망(Delaunay triangulation)	10
2.2.2 기존 지형 체적 산정	12
2.2.3 지형체적의 오차	14
2.2.4 지형 체적산정식의 개발	17
제 3 장 개발 시스템의 적용	19
3.1 적용 대상 유역	19
3.2 댐의 수치화	23
3.3 댐 저수량 산정	25

3.4 댐관리 시스템의 개발	29
3.4.1 지형의 형성 및 관리시스템 구성	29
3.4.2 시스템의 구축	31
3.4.3 시스템의 검정	31
제 4 장 결론	32
참고문헌	33
부 록	35

표 목차

표 1 차리지수지의 수위에 따른 면적 및 저수량	27
----------------------------------	----

그림 목차

그림 1 연구방법	6
그림 2 델로니 삼각망(Delaunay triangulation)	10
그림 3 수치지도의 버텍스(Vertex)	11
그림 4 TIN 구축	11
그림 5 TIN에서 면의 생성	11
그림 6 3D의 표현	11
그림 7 다각뿔 공식	12
그림 8 점고법	13
그림 9 TIN에 의한 삼각기둥	14
그림 10 수위(H)가 $Z_0 \leq H \leq Z_1$ 인 경우	15
그림 11 수위(H)가 $Z_1 \leq H \leq Z_2$ 인 경우	15
그림 12 수위(H)가 $Z_2 \leq H \leq Z_3$ 인 경우	16
그림 13 TIN에 의한 삼각기둥	17
그림 14 체적의 계산	17
그림 15 적용대상구역	19
그림 16 수치지도의 평면계획도	20
그림 17 적용 구역의 댐 건설 전	20
그림 18 댐 건설전의 모습(1)	21
그림 19 건설전 바닥 고르기	22

그림 20	댐의 수치화	23
그림 21	적용 구역의 댐 건설 후	24
그림 22	댐 저류량 산정	25
그림 23	저수지의 표고 및 저류량	26
그림 24	수위별 댐 저류량 산정	28
그림 25	개발된 시스템의 적용(표고230m)	29
그림 26	개발된 시스템의 적용(표고240m)	30
그림 27	개발된 시스템의 적용(표고254m)	30

Development of DAM-Reservoir Capacity Estimation System based on GIS

Kim, Sun Yong

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

Abstract

Recently, research through link with water resources field and geographic information system(GIS) is in progress, as the countermeasure about flood damage by abnormal weather. We can calculate result that connected with GIS, which improve processing ability and objectivity more correct in existent water resources field. This research expressed topographic feature using Triangulated irregular network(TIN) of geographic information system base. And presented method that can analyze this very effectively. Also, we emboss advantage that can achieve third dimension topography modelling effectively, and developed methodology to calculate earth-volume, the inundation volume, pondage etc. Especially, we could reduce error than existent volume calculate method at volume computation by water level's change calculating volume about each triangle that is constructed TIN's component. Also, attempted development by internal technology in this research. By using contour line and altitude of digital topographical map, put the purpose of this research that wish to develop undercurrent volume calculating method

and program of event method and present efficiency at undercurrent volume calculation.

제 1 장 서론

1.1 연구배경 및 목적

최근 이상기후로 인한 홍수 등의 피해에 대한 대책으로써 수자원분야와 지리정보시스템(Geographic Information System, GIS)의 연계를 통한 연구가 진행되고 있다. GIS와의 연계는 기존의 수자원분야에서 처리능력 및 객관성 등을 향상시켜 더욱 더 정확한 값을 산정할 수 있게 한다. 이러한 GIS는 다양한 지형자료들을 3차원으로 분석함으로써 수문학 분야에서 많은 적용을 하고 있는 실정이다. 본 연구는 지리정보체계 기반의 불규칙 삼각망(Triangulated irregular network, TIN)을 이용하여 지형을 표현한 후 수위에 따른 댐의 저수량을 산정하였다.

댐은 용수공급이라는 역할에서 매우 중요하며 용수공급은 댐 저류량과 직접적인 인과관계를 가지고 있다. 따라서 댐 저류량을 정확하게 산정하는 것은 무엇보다도 중요하다. 그러나 댐 저류량은 강우에 의한 유입량과 방류에 의해 수위가 변하며 이에 따라 저류량을 산정해야 된다. 따라서 수위에 따른 지형의 체적을 정확하게 계산함으로써 더욱 더 정확한 댐 저류량을 산정할 수 있다.

이와 같이 정확한 저류량을 산정하기 위해서 많은 연구가 수행된 바 있다. 주로 등고선법에 의하여 저류량을 산정하고 관리하고 있는데 수위의 변화량에 따른 제약조건을 두고 있다. 즉 댐 관리 조건에 따라 다르지만 1m 간격의 댐 저류량을 산정하여 관리하고 있다. 이는 댐을 실시간 관리하는데 있어서 저류량 산정시 오차 값을 갖게 한다.

따라서 본 연구에서는 이를 해소하기 위하여 선형보간법에 의하여 실시간 저류량을 산정할 수 있도록 하였으며, 저류량 산정시 오차를 줄일수 있었다. 또한 수치지도를 이용하여 가상의 댐을 건설한 후 댐 저류량을 산정하고 댐 건설시 저류면적에 의한 영향권을 분석할 수 있으므로 댐 입지선정하는

는데 도움을 줄 수 있으며, 무엇보다도 강우에 의한 댐관리를 수행하는데 도움을 줄 수 있다고 사료된다. 특히 국내의 기술에 의하여 시스템을 개발하였기 때문에 향후 국내 조건에 맞는 댐 관리시스템 구축을 하는데 큰 도움을 주며 용수공급관리의 체계화에도 도움을 줄 수 있을 것으로 사료된다.

1.2 연구동향

댐의 저류량은 용수공급의 큰 목적을 두고 있다. 따라서 정확한 용수공급 및 댐 관리를 위해서는 좀더 정확한 댐 저류량을 산정해야 된다. 따라서 정확한 체적을 계산하기 위하여 국내외의 경우 다음과 같은 많은 연구를 수행하고 있다.

신봉호(1989) 등은 수치지형을 이용한 도로의 최적토공량 산정에 관한 연구 및 1992년도에는 수치적분에 의한 토공량 산정에 관한 연구를 발표하였다. 이는 선형식보다는 비선형식이 등간격에 의한 방법보다는 부등간격에 의한 공식이 대체로 정확도가 향상되었음을 밝혔다. 또한 1999년도에 들어 GIS 응용을 위한 지형정보 획득기법에 따른 토공량 산정에 관한 연구를 수행하여 현장측량을 통해 지형정보를 획득하는 방법보다 등고선도로부터 격자기반의 지형변환점의 지형정보를 랜덤하게 획득하고 수치적분을 통해 지형체적을 산정하는 방법이 시간적으로나 경험적으로 매우 효율적임을 제시하였다.

이후 1994년 이석찬 등은 비정규삼각망 데이터구조에 의한 지형의 수치화의 연구를 통하여 데이터 구조는 복잡하지만 지형특성을 잘 살릴 수 있고 여러 응용분야에 있어서 유용한 비정규삼각망 데이터 구조를 연구, 개선함으로써 좀더 효율적인 수치지형모델을 개발하여 비정규삼각망이 지형을 재현하는데 상당히 정확한 수치표고모델임을 나타내었다.

2001년도에는 박운용 등은 수치지형모형(digital terrain model, DTM) 보간기법별 토공량 산정과 지형분석에 관한 연구를 수행하여 역거리보간법이 면적과 표면적 그리고 토공량에서 가장 큰 값을 보였고, 최근린보간기법이 가장 작은 값을 보였으며, 특히 불규칙삼각망(TIN) 방법이 정확도 측면에서 가장 효율적인 구축 방법임을 제시하였다. 또한 김상석(2001) 등은 불규칙 삼각망을 이용한 골프장의 지표면적 산출에 관한 연구를 통하여 격자형방식에 비해 불규칙삼각형방식에 의한 방법이 교차가 적게 발생

함을 알 수 있었다. 또한 정확도 및 활용도에서 불규칙삼각형 방식이 효율적임을 제시한 바 있다.

김준권(2002) 등은 습지의 체적-면적-깊이에 대한 관계식을 추정하여 우리나라 습지의 수문학적 측면에 대한 기초 연구로 향후 습지의 정량적인 연구의 방향을 제시하였다. 그해 김종성은 GPS/Echo Sounder 조합에 의한 저수지 지형 추출기법에 대한 학위논문연구를 통하여 GPS와 Echo Sounder의 조합에 의해 취득된 자료가 댐의 준공 후 관개용수량 및 수위 자료, 용량증대를 위한 개보수 등의 수자원의 관리에 효율적인 이용방향에 대해서 제시한 바 있다.

댐 저류량에 대한 수문학적 연구에서는 고덕구(1992) 등은 홍수시 저수지 실시간 운영 의사결정 지원 시스템을 구축함으로써 저수지를 홍수시에 실시간으로 홍수량 예측과 방류량을 결정할 수 있는 시스템으로 이용하는데 효율성을 제시한 바 있으며, 이근상(2002) 등은 GIS기반에서 TIN 데이터 모델을 이용한 댐 저수량 산정에 관한 연구를 통하여 TIN에 의한 저수량 계산방법이 기존의 방법에 비해 시간과 비용을 줄일 뿐만 아니라 효과적임을 제시하였다.

2003년도에는 정인주가 불규칙삼각망(TIN)을 이용하여 지형의 체적산정식을 개발하여 표고에 대한 평균값에 따른 오차를 줄일 수 있었다. 또한 표고별 지형체적을 산정함에 따라 수위별 내용적을 산정할 수 있는 수식을 개발하였다. 이에 따라 실시간 내용적을 산정할 수 있음에 따라 그 활용도를 제시할 수 있었다. 특히 침수분석시스템을 개발하여 내용적 및 수문해석을 연계함으로써 관리방안을 박사학위논문에 의하여 제시하였으며, 2004년도에는 댐 저류량 산정에 개발된 식을 적용함으로써 그 효율성을 입증하였다.

1.3 연구방법

본 연구는 수치지형도 상의 로드(node)를 포인트(point)로 인식하여 이들 절점들의 상호 기하학적 관계를 구축함으로써 지형의 3차원적인 표현과 더불어 지형의 체적 및 수면적을 산정하였다. 이러한 TIN 데이터 구조는 대부분 델로니 삼각망(Delaunay triangulation)에 그 바탕을 두고 있다. 델로니 삼각망은 세 개의 절점을 연결한 삼각형의 외접원안에 다른 점이 포함되어 있지 않도록 연결된 삼각망을 말한다(Shin et al., 1999; Rebay, S., 1993). 델로니 삼각망의 각 변의 이등분선으로 만들어지는 다각형을 보로노이 다각형(Voronoi diagram) 또는 티센 다각형이라 한다. 이러한 보로노이 다각형의 꼭지점이 델로니 삼각망의 외접원의 중심이다.

본 연구에서는 한 개의 절점과 선분에서 반시계방향으로 각을 측정했을 경우 가장 큰 값을 가지는 선분을 찾고, 새로운 절점에서 같은 방법으로 선분을 찾아 중복되는 선분이 나왔을 때 삼각형을 형성하는 알고리즘에 의하여 TIN을 구축하였다.

위와 같이 구축된 불규칙삼각망은 체적산정알고리즘에 의하여 기준선 아래의 체적을 산정하고 수위에 따른 저류량과 수면적을 산정할 수 있다. 따라서 본 연구에서는 이론식, ArcView 등과 체적산정 결과를 비교하여 오차를 분석한 후 댐 저수지에 적용하여 효율적으로 저수량을 산정하였다.

시스템을 적용하기 위하여 울산광역시 울주군 차리에 위치하고 있는 차리저수지를 선정하였다. 선정된 이유는 댐 건설 전·후의 자료를 획득할 수 있었기 때문이다. 특히 위성자료와 수치지형도의 자료를 수집하여 분석을 수행할 수 있었다. 차리저수지의 유역면적은 398 ha이고, 유효저류량은 835,000 m^3 으로 농업기반공사에서 관리하고 있는 저수지이다.

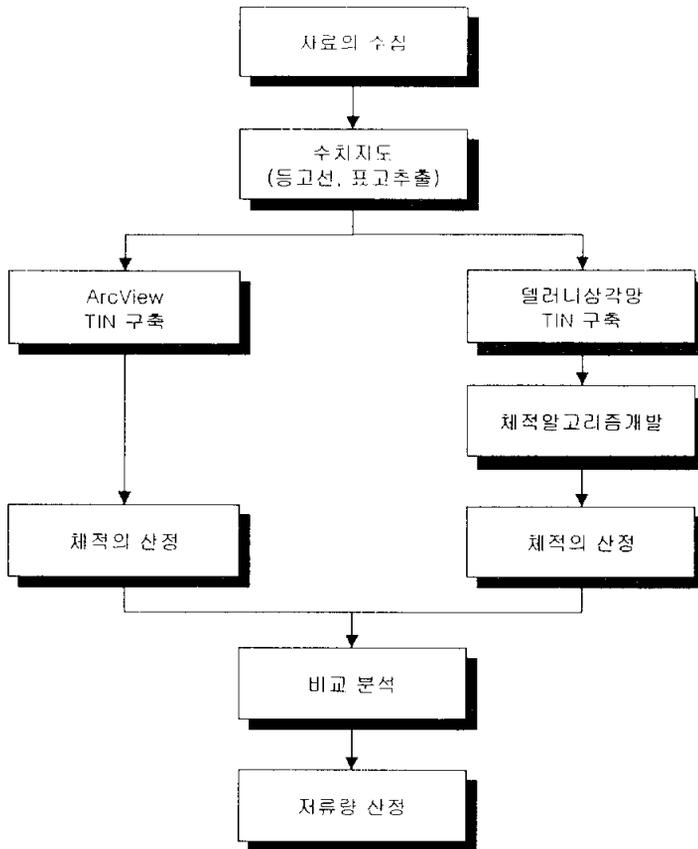


그림 1 연구방법

제 2 장 GIS 기반에서 지형의 수치화

2.1 지형의 수치화 및 체적의 산정

지형의 수치화를 통한 토공량 및 저류량 산정에 관한 연구는 현재까지 꾸준하게 연구가 이루어지고 있다. 따라서 기존의 체적 산정방법들과 본 연구에서의 체적산정 개발식을 비교 검토한 후 수위별 저류량을 산정하였다.

2.1.1 양단면 평균법에 의한 체적의 산정

신(1992)의 3명은 충북 청주시 개신동일대를 대상으로 양단면 평균법에 의하여 체적을 산정하였다.

양단면 평균법은 양끝 면적을 평균하여 이것에 대해 양단면간의 거리를 곱하여 토적을 산출하는 방법이다. 양끝의 단면적을 A_1 , A_2 로 하고 양 끝면간의 거리를 l 이라고 하면 토적 V 는 식(2.1)과 같이 나타낼 수 있다.

$$V = \left(\frac{A_1 + A_2}{2} \right) l \quad (2.1)$$

횡단측량을 일정한 거리로 하여 각 단면간의 거리 l 을 같게하고, 각 단면적을 A_1 , A_2 , \dots , A_n 이라고 하면, 그 총토적 V_t 는 식(2.2)와 같이 표현할 수 있다.

$$V_t = l \left(\frac{A_1 + A_2}{2} + A_2 + A_3 + A_4 + \dots + A_{n-1} \right) \quad (2.2)$$

위와 같은 결과 선형식보다는 비선형식이, 등간격에 의한 방법보다는 부등간격에 의한 공식이 대체로 정확도가 향상되었다. 따라서 향후 토공량

산정시 비선형식으로 대체하는 것이 합리적이라고 제시한 바 있다.

2.1.2 DTM 보간기법에 의한 체적의 산정

수치지형모형(digital terrain model, DTM)은 지형을 이미지 또는 수학 모형으로 표현한 것으로 대상지역 내에서 추출한 임의의 3차원 좌표를 처리하여, 지표기복의 기하학적 관계를 재현함으로써, 지형을 수치적으로 해석하기에 용이하도록 구조화한 것이다.

부산광역시를 대상으로 DTM 구축방안을 제시한 바 DTM은 다른 기법보다 상대적으로 적은 점을 필요로 하며, 점간의 위상적 관계가 명확하게 계산되고 기록되므로 지형분석에 유리하며 저장공간이 절약되는 장점이 있었다. 또한 실험결과 면적과 토공량 산정시 불규칙 삼각망(triangulated irregular networks, TIN) 방법이 정확도 측면에서 가장 효율적인 구축방법으로 확인된바 있다.

2.1.3 TIN 데이터구조에 의한 체적의 산정

수치표고모델에서 비정규삼각망(triangulated irregular networks, TIN)은 불규칙하게 위치해 있는 데이터들의 상호 기하학적 관계를 구축함으로써 지형의 3차원적인 표현을 가능하게 하기 위한 데이터 구조이다.

TIN에 의한 체적은 삼각형이 평면을 이룬다는 가정하에서 식(2.3)과 같이 산정할 수 있었다.

$$V_1 = \frac{1}{3} A_1 \sum_{i=1}^n (T_1 + T_2 + T_3)_i \quad (2.3)$$

여기서, V_1 : 삼각망에 의한 토공량(m^3)

A_1 : 삼각형의 면적(m^2)

T_x : 각각 꼭지점의 높이(m)

또한 정규격자형 수치표고모델로부터 다음과 같이 토공량이 계산되었다.

$$V_2 = \frac{1}{4} A_2 \sum_{i=1}^{n^2} (R_1 + R_2 + R_3 + R_4)_i \quad (2.4)$$

여기서, V_2 : 삼각망에 의한 토공량(m^3)

A_2 : 삼각형의 면적(m^2)

R_x : 각각 사각형의 꼭지점의 높이(m)

식(2.3)과 식(2.4)에서 구한 체적을 비교 조사하여 그 응용성을 검토한 결과 다음과 같이 발표된 바 있다.

TIN으로부터 계산된 체적은 정규격자형 수치표고모델에 의한 것보다 적었으며 그 차이는 22% 이내였다. 이는 지형의 보간을 선형에 의한 결과로 고차식의 경우 좀 더 근접한 결과를 얻을 수 있을 것으로 생각된다. 또한 티센 이웃점에 의한 보간의 정확도는 TIN 수치표고모델에 의한 보간의 정확도보다 낮음을 알 수 있었다. 이는 TIN이 지형을 재현하는데 상당히 정확한 수치표고모델임을 나타낸다.

2.1.4 DEM기법에 의한 체적의 산정

수치표고모델(digital elevation model, DEM)은 공간상에 나타나는 연속적인 기복의 변화를 수치적으로 표현하는 기법이며 규칙적인 간격으로 표본지점이 추출된 래스터 형태의 데이터 모델이다. DEM은 래스터 데이터 구조와 동일하며 블록, 단면, 표고점의 세 가지 요소에 의해 나타난다. DEM기법을 이용한 토량 산정에 있어서 등고선도로부터 지형변환점의 정보를 획득하여 Easaqiqclr과 Simpson3/8법칙을 적용하여 단면적을 결정하고 각주공식을 이용하여 토량을 산정하는 방법이 신속하고 효율적임을 신(1999) 제시하였다.

2.2 개발 식에 의한 체적의 산정

2.2.1 델로니 삼각망(Delaunay triangulation)

수치표고모델에서 비정규삼각망은 불규칙하게 위치하고 있는 노드(node)들의 상호 기하학적 관계를 구축함으로써 지형의 3차원적인 표현을 가능하게 하기 위한 구조이다. 이러한 비정규삼각망 데이터 구조는 대부분 델로니 삼각망(Delaunay triangulation)에 그 바탕을 두고 있다.

델로니 삼각망은 세 개의 노드(node)를 연결한 삼각형의 외접원안에 다른 점이 포함되어 있지 않도록 연결된 삼각망을 말한다(그림 2). 델로니 삼각망의 각 변의 이등분선으로 만들어지는 다각형을 보로노이 다각형 또는 티센 다각형이라 한다. 이러한 보로노이 다각형의 꼭지점이 델로니 삼각망의 외접원의 중심이다.

본 연구에서는 한 개의 노드(node)와 선분에서 반 시계방향으로 각을 측정했을 경우 가장 큰 값을 가지는 선분을 찾고, 새로운 노드에서 같은 방법으로 선분을 찾아 중복되는 선분이 나왔을 때, 삼각형을 형성하는 알고리즘에 의하여 TIN을 구축하였다.

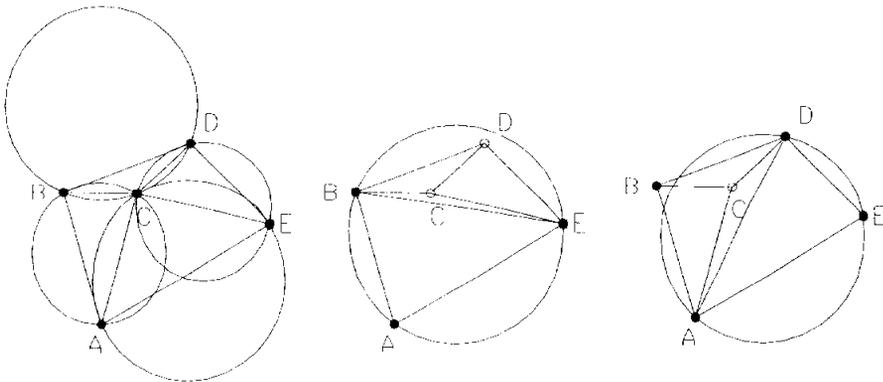


그림 2 델로니 삼각망(Delaunay triangulation)

그림 3은 수치지도의 버텍스(Vertex)를 나타낸 그림으로 버텍스의 정보를 통하여 TIN을 그림 4와 같이 구축할 수 있다. 그림 4에서 보는 바와 같이 등고선의 굴곡부분에는 TIN의 에러를 볼 수 있는데 이를 해결하기 위해서는 많은 TIN 알고리즘을 통한 분석이 요구된다. 이에 따라 본 연구에서는 체적의 오차가 발생한 것으로 분석되었다.

그림 5는 구축된 TIN은 3개의 선으로 구성되어 있으며, 본 연구에서는 이와 같은 3각형을 폐합함으로써 다각형(Polygon)을 형성하여 각각의 TIN에 대하여 면을 생성하고 색깔을 부여한다. 다음 OpenGL를 통하여 빛을 반사시킴으로써 그림 6과 같이 3차원으로 지형을 형성할 수 있다.

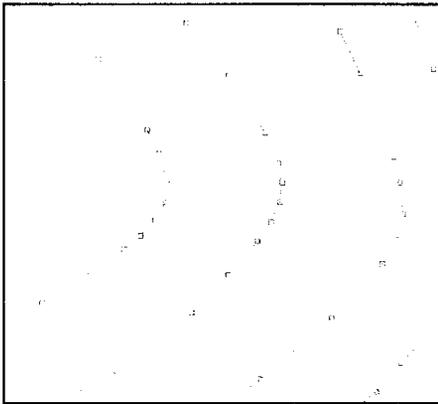


그림 3 수치지도의 버텍스(Vertex)

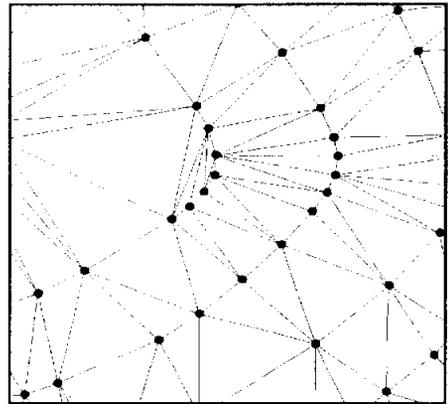


그림 4 TIN 구축

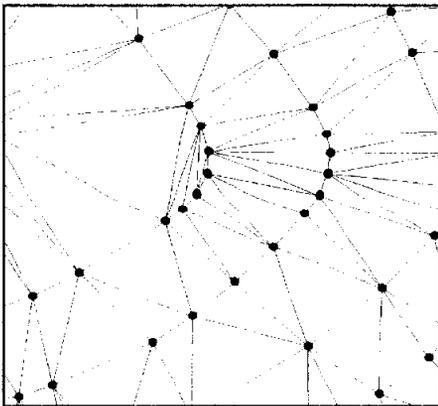


그림 5 TIN에서 면의 생성

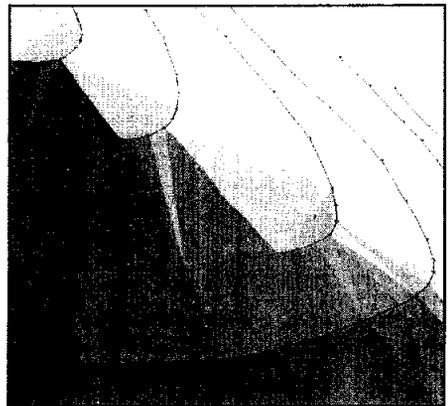


그림 6 3D의 표현

2.2.2 기존 지형 체적 산정

지형체적의 계산을 계산하는 방법으로는 단면법, 등고선법, 점고법 등이 있다.

1) 단면법

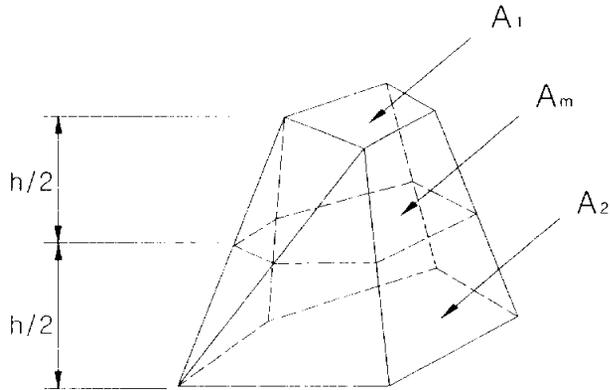


그림 7 다각뿔 공식

단면법에는 양단면평균법, 중앙단면법, 다각뿔공식이 있다. 그 중 다각뿔공식은 그림 7과 같이 위, 아래의 면이 평행이고 측면이 모두 평면형으로 된 입체를 다각뿔(角柱)라고 하며 체적은 심프슨법칙에 의해 구할 수 있다.

$$V = \frac{h}{6} (A_0 + 4A_m + A_1) \quad (2.5)$$

2) 등고선법

등고선을 이용해서 체적을 계산하는 방법으로서 저수지의 내용적, 사방댐의 유사량, 토취장 및 채석장의 굴착량 혹은 역으로 사토장의 허용 사토량 등을 구하고자 할 때 많이 이용되는 방법이다.

3) 점고법

넓은 지역 또는 택지조성공사 등에 필요한 토공량을 계산하는데 응용된다. 일반적으로 단면적이 평면이면, 어떠한 형상이더라도 체적은 단면중심에서의 연직거리에 수평면적을 곱한 것과 같다. 그림 8에서 사각형에 대한 밑면적을 A, 네 점의 높이를 h_1, h_2, h_3, h_4 라고 하면 체적 V는 사각형(a)인 경우에는

$$V = \frac{1}{4} A(h_1 + h_2 + h_3 + h_4) \quad (2.6)$$

이고, 삼각형(b)의 경우에는 밑면적을 A'라 할 때

$$V = \frac{1}{3} A'(h_1 + h_2 + h_3) \quad (2.7)$$

로 구할 수 있다.

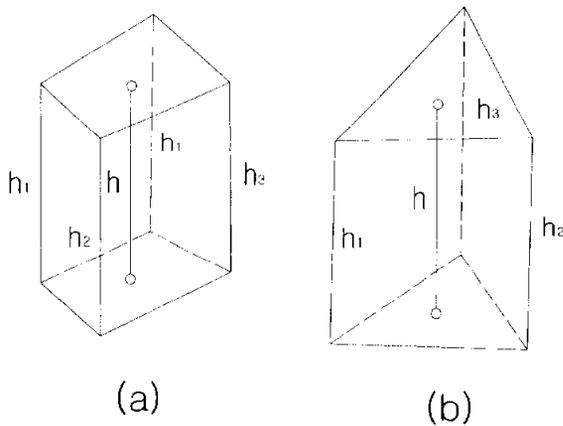


그림 8 점고법

그러나 단면법, 등고선법, 점고법은 각각의 꼭지점들의 평균값을 산정하기 때문에 실시간 수위의 변화에 따른 체적을 구하는데는 그림 7에서 보는 바와 같이 오차가 발생한다. 따라서 본 연구에서는 식(2.8)에 의하여 지형의 체적을 산정하였다.

2.2.3 지형체적의 오차

지형체적의 오차는 기존의 방법인 점고법과 개발된 방법에 의하여 비교하였다. 비료를 위하여 TIN에 의한 삼각기둥은 그림 9에서 보는 바와 같다.

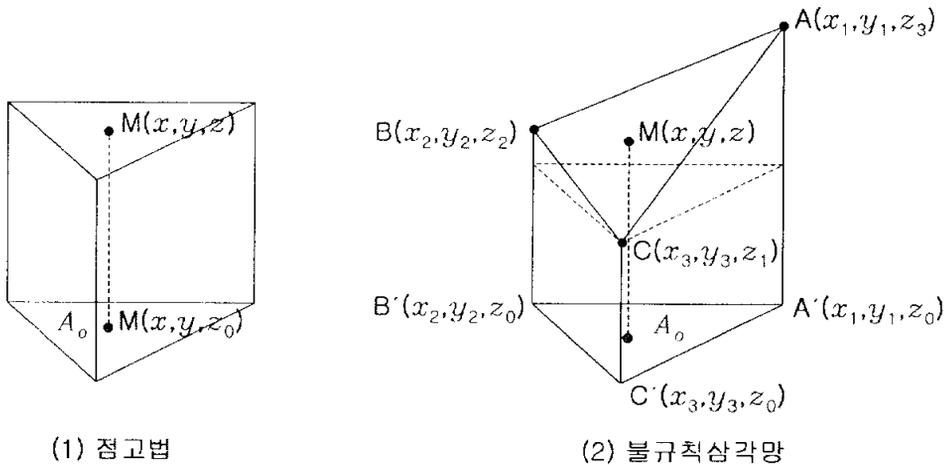


그림 9 TIN에 의한 삼각기둥

그림 9에서 (1)과 (2)의 체적을 수위별 계산을 수행한 결과 다음과 같은 오차를 가지고 있었다. TIN에 의한 삼각형은 3개의 높이 값을 가지고 있다. 그러나 (1)은 3개의 높이 값이 같게 계산을 수행하였고 (2)는 각각 다른 높이 값을 가지며 체적을 계산하였다. 그 결과는 다음과 같다.

먼저 수위가 세 개의 높이 값 중 가장 작은 값(Z_1)보다 적었을 경우에 대하여 단면적을 그림 10에서 비교한 결과 단면적(A_0)로 같게 나타났다 ($Z_0 \leq H \leq Z_1$).

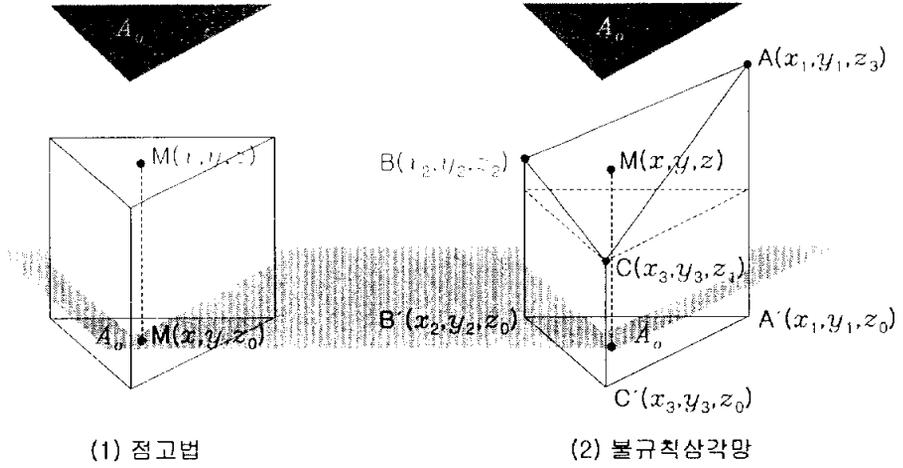


그림 10 수위(H)가 $Z_0 \leq H \leq Z_1$ 인 경우

그러나 수위가 두 번째 높이 값(Z_2)보다 작고 세 번째 높이 값(Z_1)보다 큰 경우에는 (1)의 경우 단면적(A_0)은 변화하지 않지만 (2)경우는 단면적(A_0)이 적어짐을 알 수 있다. 또한 체적계산에 있어서 (1)의 경우에는 A_0H 이지만 (2)의 경우는 그렇지 않다.

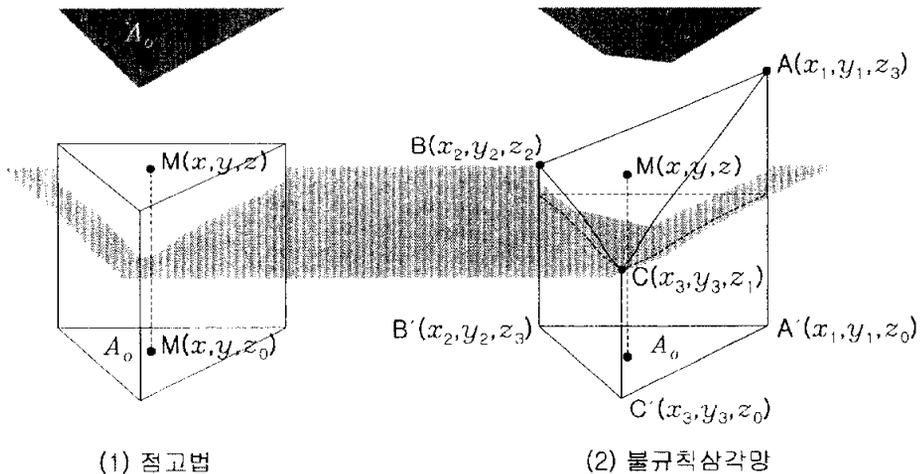


그림 11 수위(H)가 $Z_1 \leq H \leq Z_2$ 인 경우

수위가 가장 높은 높이 값(Z_3)과 두 번째 높이 값(Z_2) 사이에 위치할 경우 (1)의 경우에는 그림 12에서 보는바와 같이 단면적은 0이고 (2)의 경우는 0이 아니다. (1)의 체적은 그림 11의 (1)과 같은 결과 A_0H 와 같지만 (2)는 수위의 변화에 따라 체적이 변화하고 있다.

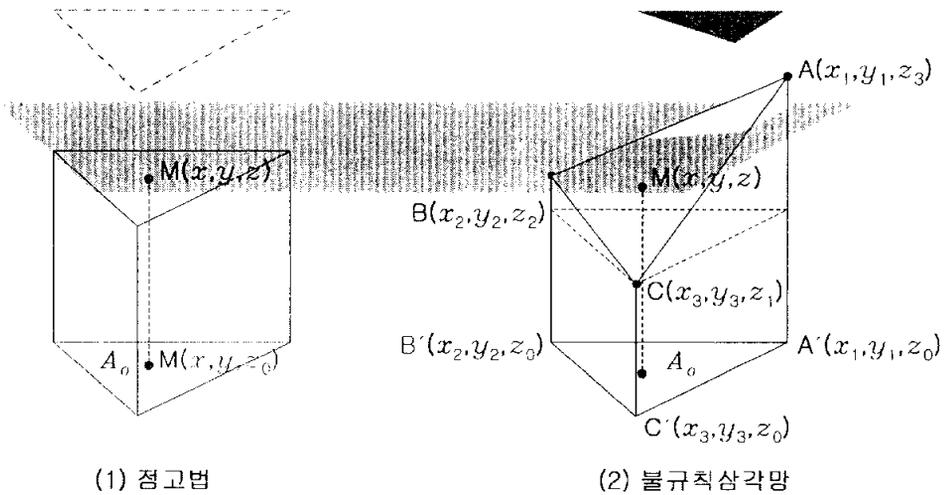


그림 12 수위(H)가 $Z_1 \leq H \leq Z_2$ 인 경우

즉, 그림 12의 (1)에서 보는바와 같이 점 M 를 중심으로 삼각형 ABC 의 투영단면적으로 계산한다. 그렇다면 수위 H 가 점 M 보다 작을 때는 A_0H 로 체적을 산정할 수 있다. 그러나 실제로는 BCM 의 4면체는 A_0H 의 체적에서 제외되어야 된다. 그러나 단면법, 등고선법, 점고법은 이를 고려하지 못하였다.

2.2.4 지형 체적산정식의 개발

지형의 체적은 전체체적, 수위의 변화에 따른 체적으로 구분하여 체적을 산정하였다. 특히 수위에 따른 체적의 변화는 댐 저류량 및 하천의 내용적, 침수의 영향 등을 분석할 수 있는 방법을 제시할 수 있었다. 따라서 본 연구에서는 개발된 식을 이용하여 저류량을 산정하는데 적용을 하였다.

그림 13은 TIN에 의하여 생성된 삼각형을 삼각기둥으로 변환한 그림이다. 즉 전체체적을 산정할 경우에 필요한 그림이고, 수위의 변화에 따른 체적은 그림 14와 같다. 그러나 수위가 변화하는 동안 체적산정은 다음과 같이 3개의 경우로 구분되어 체적을 산정할 수 있다.

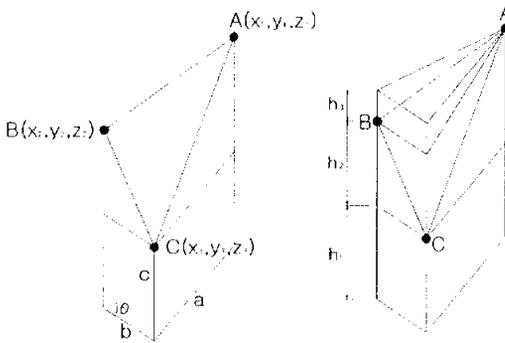


그림 13 TIN에 의한 삼각기둥

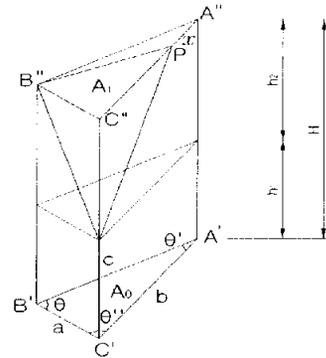


그림 14 체적의 계산

1) 지형의 전체체적

지형의 전체체적은 그림 13에 의하여 식 2.8과 같이 정의 할 수 있다.

$$V_t = \sum_{i=1}^n A_{0i} \left\{ h_{1i} + \frac{2}{3} h_{2i} + \frac{1}{3} h_{3i} \right\} \quad (2.8)$$

여기서, V_t : 지형의 전체 체적(m^3)

A_{0i} : TIN에 의한 삼각형의 투영단면적(m^2)

h_1 : $Z_1 - Z_0$

h_2 : $Z_2 - Z_1$

h_3 : $Z_3 - Z_2$

이때, 수위에 따른 체적은 $H \leq h_1$, $h_1 \leq H \leq h_2$, $h_2 \leq H \leq h_3$ 의 경우로 구분하여 체적을 다음과 같이 계산할 수 있었다.

2) 수위가 $H \leq h_1$ 인 경우,

$$V_{S_1} = \sum_{i=0}^n [A_{0i} \cdot H] \quad (2.9)$$

3) 수위가 $h_1 \leq H \leq h_2$ 인 경우,

$$V_{S_2} = \sum_{i=0}^n \left[A_{0i} \left(H - \frac{(H - h_{1i})^4}{3h_{2i}^2(h_{2i} + h_{3i})} \right) \right] \quad (2.10)$$

4) 수위가 $h_2 \leq H \leq h_3$ 인 경우,

$$V_{S_3} = \sum_{i=0}^n \left[A_{0i} \left(h_{1i} + \frac{2}{3} h_{2i} + \frac{1}{3} h_{3i} \right) - \frac{1}{3} A_{0i} \left(\frac{(h_{1i} + h_{2i} + h_{3i} - H)^3}{h_{3i}(h_{2i} + h_{3i})} \right) \right] \quad (2.11)$$

제 3 장 개발 시스템의 응용

3.1 적용 대상 유역

검정을 거친 개발된 시스템을 대상 저수지에 적용을 하였으며, 대상 저수지는 울산광역시 울주군 차리에 위치한 차리저수지로써 농업기반공사 소재이다. 이 저수지는 최근 신설된 저수지로써 유효저류량은 835,000m³이며, 유역면적은 398ha이다. 차리저수지 하류는 그림 15와 같이 구량천이 위치하며 경부고속도로를 관류하여 대곡천으로 합류된 후 대화강으로 유입된다.

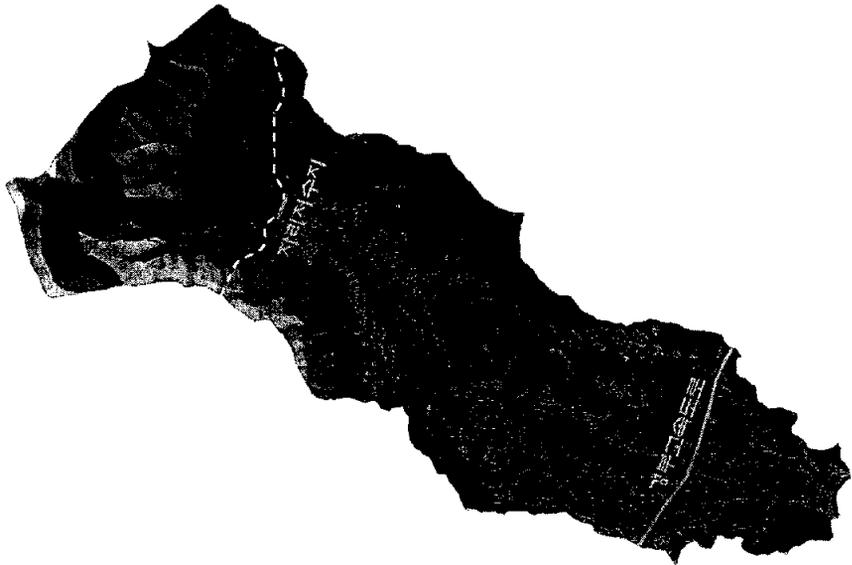


그림 15 적용대상유역

그림 16은 대상구역의 수치지도와 저수지 건설전의 계획 평면도를 나타낸 그림이다. 대상구역의 등고선 자료를 이용하여 그림 17과 같이 불규칙삼각망을 형성하여 OpenGL로 지형을 3차원으로 표현하였다.

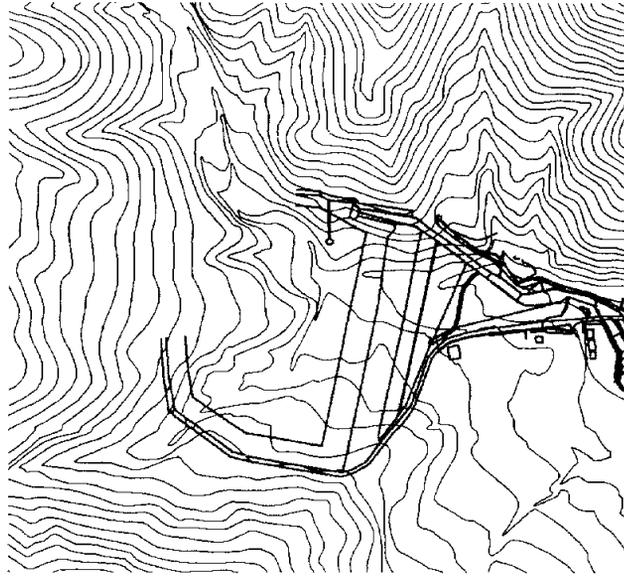


그림 16 수치지도의 평면계획도



그림 17 적용 구역의 댐 건설 전

본 적용대상 유역의 댐 건설전의 위성사진은 그림 18과 같으며, 그림 19는 댐을 건설하기 위하여 바닥 고르기 작업을 수행하고 있는 모습을 나타낸 것이다. 이로 인하여 저류량을 예측하는데 표고 오차가 발생하였으며 본 연구에서는 이를 보정하여 저류량을 산정하였다.

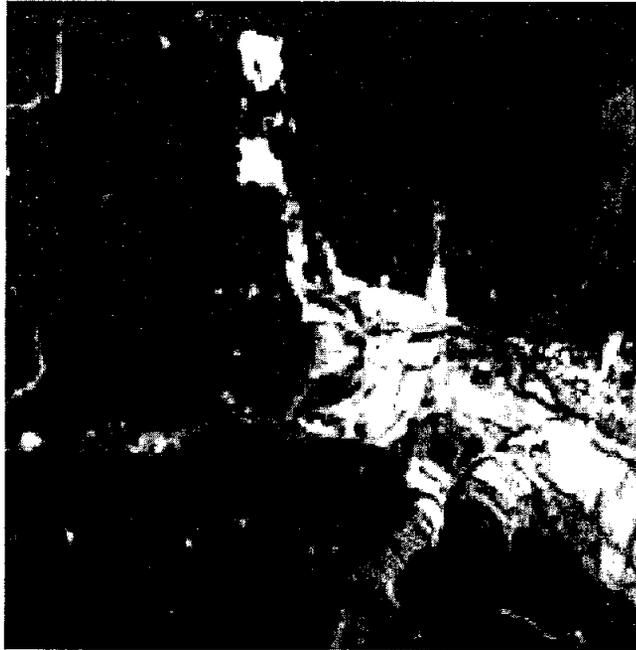


그림 18 댐 건설전의 모습(1)

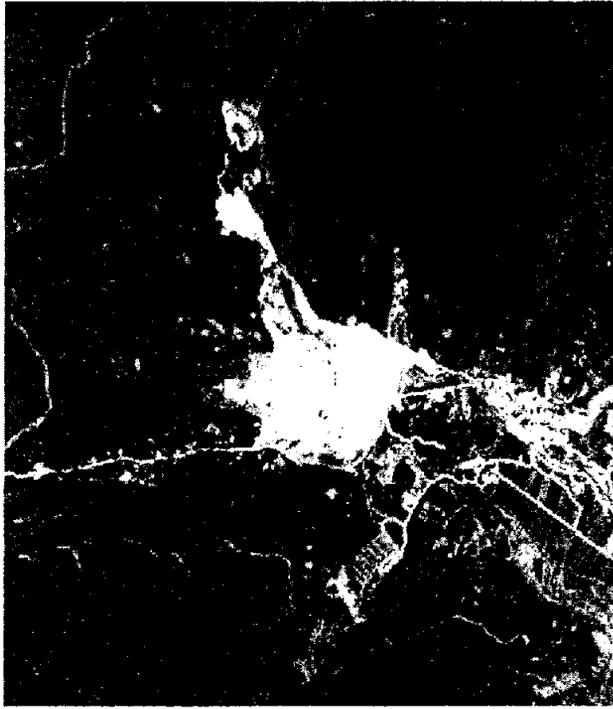


그림 19 건설전 바닥 고르기

3.2 댐의 수치화

그림 20은 대상구역의 등고선 위에 댐을 가상으로 건설하였다. 그러나 댐을 건설하기 위해서는 바닥 고르기를 한 후 댐을 건설하기 때문에 저류량을 산정하는데 오차가 발생하였다. 실제 그림 19에서 보는 바와 같이 댐 바닥을 고르는 공사를 하였기 때문에 기존의 수치지도를 사용하는데 오차의 범위를 증가시켰다.

그림 21에서는 수치지도상에 댐을 건설 후의 모습을 TIN을 구축한 후 3차원으로 나타내었다. 따라서 수치지도를 이용하여 댐 건설시 적지분석 및 침수의 범위를 모델링 함으로써 효율적인 설계를 하고 관리하는데 도움을 줄 수 있다고 사료된다.

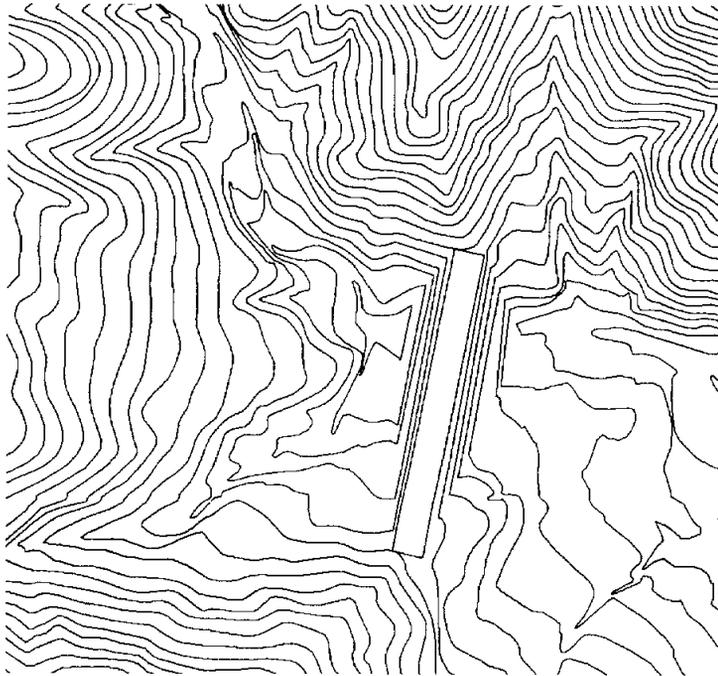


그림 20 댐의 수치화



그림 21 적용 구역의 댐 건설 후

3.3 댐 저류량 산정

댐 저류량은 그림 22에서 보는 바와 같이 수위(H)를 기준으로 하여 단위길이 당 □abcd를 산정한다. 그리고 수위 아래 부분의 체적(V_2)를 산정하여 저류량(S)을 산정하며, 식 3.2에 의하여 계산된다.

$$A_0 \cdot H = V_2 + S \quad (3.1)$$

$$S = A_0 \cdot H - V_2 \quad (3.2)$$

여기서, S : 댐 저류량(m^3)

A_0 : TIN에 의한 삼각형의 투영단면적(m^2)

H : 수위(m)

V_2 : 수위에 따른 침수체적(m^3)으로써 식 2.9, 2.10, 2.11에 의하여 산정한다.

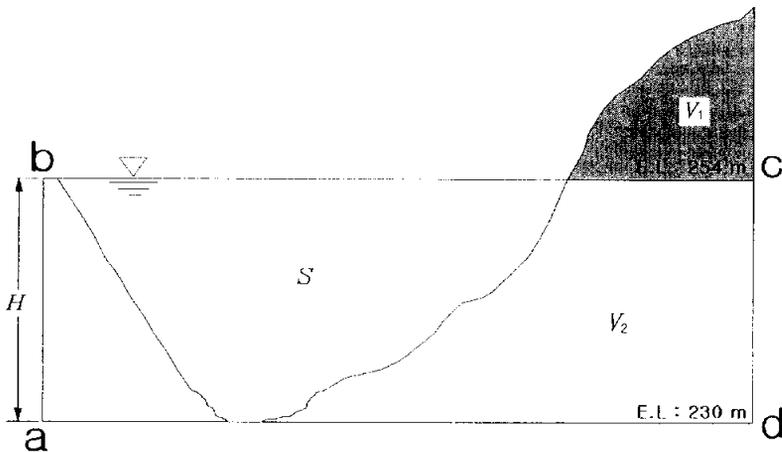


그림 22 댐 저류량 산정

본 연구대상구역의 댐 저류량은 개발된 식에 의한 시스템에 의하여 그림 23과 같이 산정하였다. 지형의 표고·체적 곡선은 주로 지수함수식을 나타내고 있으며, 수위가 상승함에 따라 수면적은 커지기 때문에 체적은 급속하게 증가하는 양상을 보였다.

표 1에서는 농업기반공사에서 관리하고 있는 저류량과 본 연구에서 산정한 저류량을 비교 분석하였다. 그 결과 그림 24에서 그래프로 나타내었으며 2m의 표고를 보정하여 값을 획득할 수 있었다.

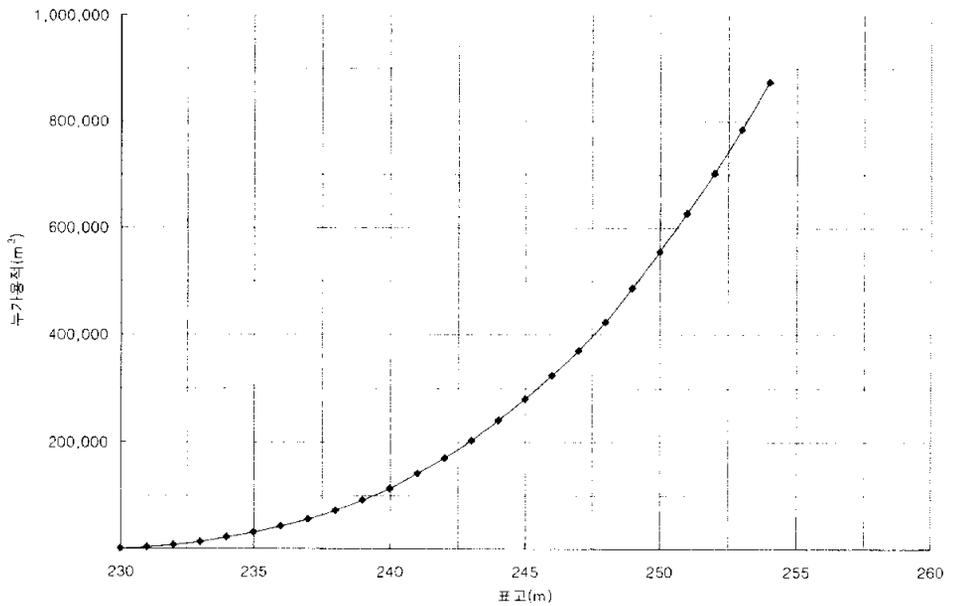


그림 23 저수지의 표고 및 저류량

표 1 차리지수지의 수위에 따른 면적 및 저수량

표고 (m)	농업기반공사		개발된시스템	
	면적 (m ²)	누가용적 (m ³)	면적 (m ²)	누가용적 (m ³)
230	305	405	1,201	461
231	755	1,160	4,837	3,932
232	2,300	3,460	6,560	8,592
233	5,080	8,540	8,023	14,573
234	7,550	16,090	9,227	22,268
235	9,548	25,738	10,173	32,332
236	12,138	37,876	15,654	45,482
237	14,614	52,490	18,827	60,700
238	17,161	69,651	21,591	78,143
239	19,479	89,129	23,947	98,342
240	21,906	111,035	25,895	122,205
241	24,775	135,810	31,754	149,342
242	28,340	164,150	35,435	178,944
243	33,070	197,220	38,646	211,305
244	37,330	234,550	41,388	247,202
245	41,380	275,930	43,660	287,897
246	45,760	321,690	52,088	333,906
247	50,125	371,815	56,971	383,819
248	55,150	426,965	61,132	437,807
249	60,275	487,240	64,572	496,588
250	65,120	552,360	67,290	561,422
251	71,080	623,440	75,520	631,203
252	78,755	702,195	81,354	706,129
253	87,279	789,474	86,268	786,485
254	94,793	884,267	90,263	873,261

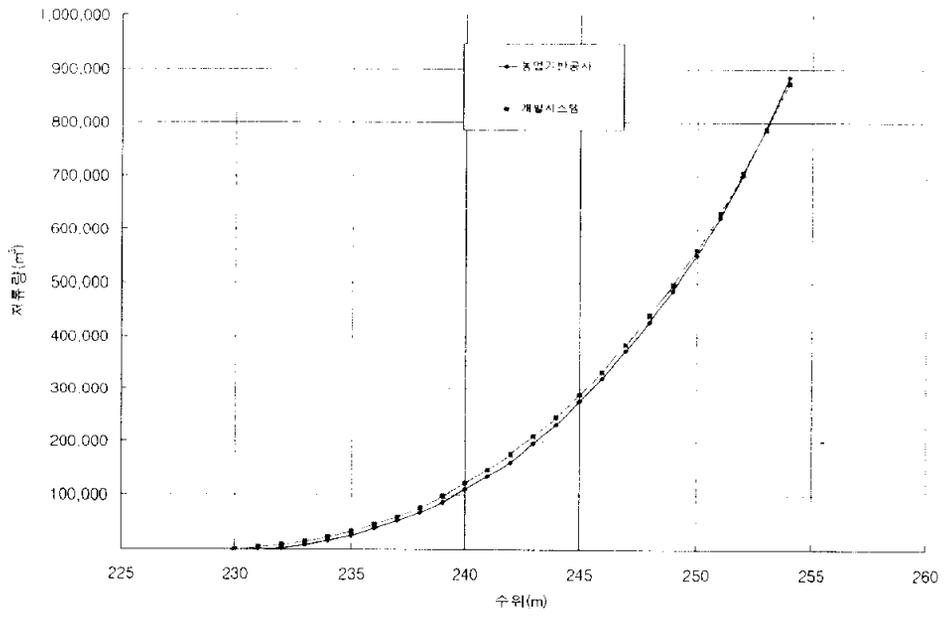


그림 24 수위별 댐 저류량 산정

3.4 댐관리 시스템의 개발

3.4.1 지형의 형성 및 관리시스템 구성

본 연구에서는 MFC기반으로 3Dview.ocx를 이용하여 시스템을 개발하였다. 시스템은 dxf파일을 입력데이터로 하며 입력된 수치지도에 대하여 불규칙삼각망을 구축하여, 각각의 삼각형을 폴리곤으로 형성시킨다. 따라서 각각의 삼각형은 면을 형성하게 되며, 그러한 각각의 면에 색을 부여하여 빛의 각도에 따라 음영처리를 하여 3차원으로 표현하였다.

본 시스템은 3차원적인 표현 및 면적-저류량을 표고별로 실시간 자동으로 산정할 수 있다는 것이 큰 장점이라 할 수 있다. 그림 25, 26, 27에서는 표고 230, 240, 254m에 대한 3차원 해석 및 저류량과 수면적을 산정한 그림이다.

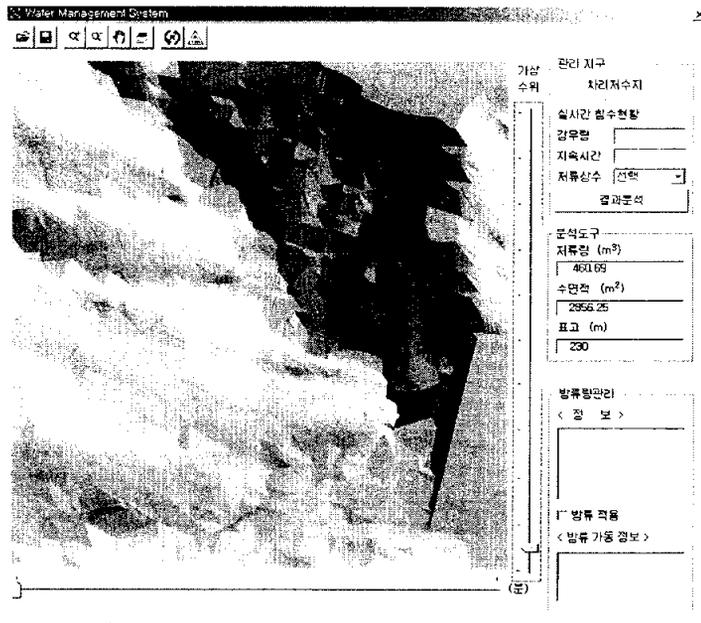


그림 25 개발된 시스템의 적용(표고230m)



그림 26 개발된 시스템의 적용(표고240m)



그림 27 개발된 시스템의 적용(표고254m)

3.4.2 시스템의 구축

3Dviewer.ocx는 불규칙삼각망을 구축하여 각각의 삼각형을 면처리 하여 OpenGL에 의하여 3차원으로 표현한다. 따라서 수위에 따른 저수량을 산정할 뿐만 아니라 강우에 의한 시간별 저수량을 산정할 수 있다. 더구나 방류량을 적용하여 정확한 저수지량을 산정하기 위하여 노력하였다. 불규칙삼각망은 델로니삼각화에 의하여 구축을 하였고, 이들은 모두 MFC에 의하여 시스템이 개발되었다. 따라서 국내에 맞는 프로그램의 응용에 활용성이 아주 크다고 사료된다.

3.4.3 시스템의 검정

시스템의 검정은 체적을 산정하는 방법에 따라 저수량을 산정할 수 있으므로 체적을 산정시 오차를 줄이는 것이 무엇보다도 중요하다.

본 연구에서 체적산정식에 대한 검정은 정인주(2003)에 의하여 검정된 바 있으며, 원뿔모양의 체적을 이론식과 ArcView로 산정한 후 개발된 체적을 하여 비교 검정하였다. 검정 결과 이론식과 0.654~0.855(%)의 오차를 발생하였으며 양호한 값을 구할 수 있었다.

따라서 댐 저수량을 산정하는데 오차를 줄일 수 있었다고 사료된다.

제 4 장 결론

본 연구는 불규칙삼각망을 구축하여 개발된 식에 의하여 저류량을 산정하는 연구를 수행한 결과 다음과 같은 결론을 얻을 수 있었다.

1. 수위의 변화에 따른 수면적 및 저류량을 산정하는데 오차를 줄일 수 있었다.
2. 3차원이 가능한 모델링을 통하여 효율적인 관리시스템을 구축하였으며, 향후 수자원관리에 있어서 큰 도움을 줄 수 있다고 사료된다.
3. 저류량의 산정결과 농업기반공사의 자료와 약 4%의 오차가 발생하였으며, 수치지도의 사용에 대한 방안을 제시할 수 있다.
4. 3차원 GIS를 이용한 물관리시스템(Water Management System)을 개발함으로써, 설계시 입지분석 및 저류량을 예측하는데 도움이 될 것으로 사료된다.

참고문헌

1. 신봉호, 송왕재, 윤철규, 임재구, “수치적분에 의한 토공량 산정에 관한 연구” 건설기술연구소 논문집, 제11권, 제1호, 1992, pp. 153~161.
2. 박운용, 김천영, 이현우, “DTM 보간기법별 토공량 산정과 지형분석에 관한 연구”, 한국지형공간정보학회논문집, 제9권, 제1호, 2001, pp.39~47.
3. 이석찬, 고영호, 이창경, 최병길, “비정규삼각망 데이터구조에 의한 지형의 수치화”, 대한토목학회논문집, 제14권, 제2호, 1994, pp.325~334.
4. 신봉호, “GSIS 응용을 위한 지형정보 획득기법에 따른 토공량 산정에 관한 연구”, 건설기술연구소 논문집, 제18권, 제1호, 1999, pp.1~9.
5. 정인주, 김상용, “불규칙삼각망에 의한 지형체적 산정에 관한 연구”, 한국지형공간정보학회논문집, 제11권, 1호, 2003, pp. 77~82.
6. 정인주, “3차원 Virtual GIS를 이용한 도시하천관리시스템 구축”, 부경대학교 토목공학과, 박사학위논문, 2003, pp. 1~85.
7. 박상주, “지리정보체계를 이용한 홍수재해관리시스템 구축”, 부경대학교 토목공학과, 박사학위논문, 2004, pp. 1~106.
8. 정인주, 김상용, “Virtual GIS 응용을 위한 지형의 체적 산정에 관한 연구”, 대한토목학회논문집, 제23권, 5호, 2003, pp. 389~395.
9. 이석찬, 고영호, 이창경, 최병길, “비정규삼각망 데이터구조에 의한 지형의 수치화”, 대한토목학회논문집, 제14권, 제2호, 1994, pp. 328~329.
10. 김상성, 장용구, 곽재하, 김윤수, “불규칙삼각망을 이용한 골프장의 지표면적 산출에 관한 연구”, 한국지리정보학회지, 제4권, 제4호, 2001, pp. 61~71.
13. 김의명, 이석균, 김태균, “방사삼각망알고리즘을 이용한 불규칙삼각망의 생성과 경사분석”, 대한토목학회논문집, 제20권, 6-D호, 2000, pp. 701~708.
14. 백은기, 육찬상, 김원익, 김감래, 강양진, 이영진, 측량학, 청문각, 제2판,

- 1995, pp. 205~210.
15. Rebay, S., "Efficient Unstructured Mesh Generation by Means of Delaunay Triangulation and Bowyer Watson Algorithm," *Journal of Computational Physics*, Vol. 106, 1993, pp. 125~138.
 16. G. Gambolati, P. T. M. Gonella (2002) GIS Simulations of the Inundation Risk in the Coastal Lowlands of the Northern Adriatic Sea, *Journal of Mathematical and Computer Modelling*, Vol. 35, pp. 963i 972.
 17. Philip A. T., Stephen J. W. (1998) Modeling floodplain inundation using an integrated GIS with radar and optical remote sensing, *Journal of Geomorphology*, Vol. 21, pp. 295i 312.
 18. Clark, C. O. (1943) Storage and unit hydrograph, *Proc, ASCE*, Vol. 9, pp. 1333i 1336.
 19. Snyder, F. F. (1938) *Hydrologic System, Rainfall Runoff Modeling*, Vol. 1, Prentice Hall, Englewood Cliffs, N.J.
 20. Jeong, In Ju, Kim, Sang Yong (2004) A Study on the Application of Flood Disaster Management using GIS, *Korea Water Resources Association intersession*, pp. 15~24.
 21. 정인주, 김상용 (2004) GIS 기반의 유출모형 입력데이터 작성에 관한 연구, *한국수자원학회 학술발표회 논문집*, pp. 61~64.
 22. 육승우, 정인주, 김상용 (2004) GIS를 이용한 댐저류량 산정에 관한 연구, *한국수자원학회 학술발표회 논문집*, pp. 196~199.
 23. 김종성 (2002) GPS/Echo Sounder 조합에 의한 저수지 지형 추출 기법, *부경대학교 산업대학원 토목공학과, 공학석사학위논문*, pp 30~51.

부 록

```

// DlgRealTime.cpp : implementation file
#include "stdafx.h"
#include "RiverSystem.h"
#include "DlgRealTime.h"
#include "DlgRealTimeList.h"
#include "EtcFunction.h"
#include <math.h>
#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif
// CDlgRealTime dialog
CDlgRealTime::CDlgRealTime(CWnd* pParent /*=NULL*/)
    : CDialog(CDlgRealTime::IDD, pParent)
{
    //{{AFX_DATA_INIT(CDlgRealTime)
    m_strFlood = _T("");
    m_strTime = _T("");
    m_strFloodArea = _T("");
    m_strFloodHeight = _T("");
    m_strWaterVolume = _T("");
    m_strFloodVolume = _T("");
    //}}AFX_DATA_INIT
}

void CDlgRealTime::DoDataExchange(CDataExchange* pDX)
{
    CDialog::DoDataExchange(pDX);
    //{{AFX_DATA_MAP(CDlgRealTime)
    DDX_Control(pDX, IDC_EBFloodVolume, m_cbFloodVolume);
    DDX_Control(pDX, IDC_CHKOWl, m_chkOwl);
    DDX_Control(pDX, IDC_CMBAmc, m_embAmc);
    DDX_Control(pDX, IDC_LSTPumpInfo, m_lstPumpInfo);
    DDX_Control(pDX, IDC_LSTPump, m_lstPump);
    DDX_Control(pDX, IDC_CHKump, m_chkPump);
    DDX_Control(pDX, IDC_EBWaterVolume, m_cbWaterVolume);
    DDX_Control(pDX, IDC_EBFloodHeight, m_cbFloodHeight);
    DDX_Control(pDX, IDC_EBFloodArea, m_cbFloodArea);
    DDX_Control(pDX, IDC_SLIDERDivTime, m_SlideDivTime);
    DDX_Control(pDX, IDC_EBTime, m_cbTime);
    DDX_Control(pDX, IDC_EBFlood, m_cbFlood);
    DDX_Control(pDX, IDC_SLIDERRealTime, m_SlideRealTime);
    DDX_Control(pDX, IDC_3DVIEWER, m_3DViewer);
    DDX_Text(pDX, IDC_EBFlood, m_strFlood);
    DDX_Text(pDX, IDC_EBTime, m_strTime);
    DDX_Text(pDX, IDC_EBFloodArea, m_strFloodArea);
    DDX_Text(pDX, IDC_EBFloodHeight, m_strFloodHeight);
    DDX_Text(pDX, IDC_EBWaterVolume, m_strWaterVolume);
    DDX_Text(pDX, IDC_EBFloodVolume, m_strFloodVolume);
    //}}AFX_DATA_MAP
}

BEGIN_MESSAGE_MAP(CDlgRealTime, CDialog)
    //{{AFX_MSG_MAP(CDlgRealTime)
    ON_COMMAND(ID_TBRealTimeFit, OnTBRealTimeFit)
    ON_COMMAND(ID_TBRealTimePan, OnTBRealTimePan)
    ON_COMMAND(ID_TBRealTimeZoomIn, OnTBRealTimeZoomIn)
    ON_COMMAND(ID_TBRealTimeZoomOut, OnTBRealTimeZoomOut)
    ON_COMMAND(ID_TBRealTimeRotate, OnTBRealTimeRotate)
    ON_COMMAND(ID_TBRealTimeFileOpen, OnTBRealTimeFileOpen)
    ON_NOTIFY(NM_RELEASEDCAPTURE, IDC_SLIDERRealTime, OnReleasedcaptureSLIDERRealTime)
    ON_COMMAND(ID_TBRealTimeFileSave, OnTBRealTimeFileSave)
    ON_COMMAND(ID_TBRealTimeMakeTin, OnTBRealTimeMakeTin)
    ON_COMMAND(ID_TBRealTimeList, OnTBRealTimeList)
    ON_BN_CLICKED(IDC_BUTResult, OnBUTResult)
    ON_BN_CLICKED(IDC_RDBMngBanya, OnRDBMngBanya)
    ON_BN_CLICKED(IDC_RDBMngSuYoung, OnRDBMngSuYoung)
    //}}AFX_MSG_MAP
}

```

```

        ON_NOTIFY(NM_RELEASEDCAPTURE, IDC_SLIDERDivTime, OnReleasedcaptureSLIDERDivTime)
        ON_BN_CLICKED(IDC_CHKump, OnCHKump)
        ON_BN_CLICKED(IDC_CHKowl, OnCHKowl)
        //))AFX_MSG_MAP
    END_MESSAGE_MAP()
    BOOL CDlgRealTime::OnInitDialog()
    {
        CDialog::OnInitDialog();
        m_dSWValue = 0.0;
        OnInitControl();
        return TRUE; // return TRUE unless you set the focus to a control
    }
    void CDlgRealTime::OnInitControl()
    {
        OnInitToolBar();
        OnInitEdit();
        OnInitCombo();
        OnInitOption();
        OnInitSlider();
        OnInitCheck();
        OnInitPumpInfo();
        OnInitOwlInfo(); // 수문 관련 정보
        OnInitRepRead();
    }
    void CDlgRealTime::OnInitToolBar()
    {
        if(m_wndToolBar.Create(this)||m_wndToolBar.LoadToolBar(IDR_TBRealTime) ){
            TRACE0("Failed to create toolbar\n");
            return ; // fail to create
        }
        m_wndToolBar.SetBarStyle(m_wndToolBar.GetBarStyle()|CBRS_TOOLTIPS| CBRS_FLYBY);
        RepositionBars(AFX_IDW_CONTROL_BAR_FIRST,AFX_IDW_CONTROL_BAR_LAST,0);
    }
    void CDlgRealTime::OnInitEdit()
    {
        m_strFloodArea.Format(_T("0.0"));
        m_cbFloodArea.SetWindowText(m_strFloodArea);
        m_strFloodHeight.Format(_T("0.0"));
        m_cbFloodHeight.SetWindowText(m_strFloodHeight);
        m_strWaterVolume.Format(_T("0.0"));
        m_cbWaterVolume.SetWindowText(m_strWaterVolume);
        m_strFloodVolume.Format(_T("0.0"));
        m_cbFloodVolume.SetWindowText(m_strFloodVolume);
    }
    void CDlgRealTime::OnInitCombo()
    {
        m_cmbAmc.InsertString(0,_T("선택"));
        m_cmbAmc.InsertString(1,_T("건조"));
        m_cmbAmc.InsertString(2,_T("조금전조"));
        m_cmbAmc.InsertString(3,_T("습윤"));
        m_cmbAmc.InsertString(4,_T("포화"));
        m_cmbAmc.SetCurSel(0);
    }
    void CDlgRealTime::OnInitOption()
    {
        OnInitMng();
    }
    void CDlgRealTime::OnInitMng()
    {
        CButton* pButton;
        pButton = (CButton*)GetDlgItem(IDC_RDBMngBanya);
        pButton->SetCheck(0);
        pButton = (CButton*)GetDlgItem(IDC_RDBMngSuYoung);
        pButton->SetCheck(0);
    }

```

```

void CDlgRealTime::OnInitSlider()
{
    m_SlideRealTime.SetTicFreq(10);
    m_SlideRealTime.SetPageSize(10);
    m_SlideRealTime.SetRangeMin(0);
    m_SlideRealTime.SetRangeMax(100);
    m_SlideRealTime.SetPos(100);
}

void CDlgRealTime::OnInitCheck()
{
    m_chkPump.SetCheck(0);
    m_chkOwl.SetCheck(0);
}

void CDlgRealTime::OnInitPumpInfo()
{
    m_PumpInfo.RemoveAll();
    if(! m_pDB->PumpSelectQ())
        return ;
    int nRecordCount = m_pDB->m_Result.GetSize();
    CPumpInfo PumpInfo;
    CEtcFunction EtcFunc;
    char strBuffer[8][100];
    for(int i = 0 ; i < nRecordCount ; i++){
        sscanf(m_pDB->m_Result[i],"%s %s %s %s %s %s %s %s",strBuffer[0],strBuffer[1],strBuffer[2],strBuffer[3],
        strBuffer[4],strBuffer[5],strBuffer[6],strBuffer[7]);
        PumpInfo.strPumpName.Format("%s",EtcFunc.DeleteNull(strBuffer[0]));
        PumpInfo.strPumpNum.Format("%s",EtcFunc.DeleteNull(strBuffer[1]));
        PumpInfo.strPumpType.Format("%s",EtcFunc.DeleteNull(strBuffer[2]));
        PumpInfo.dvolume = atof(EtcFunc.DeleteNull(strBuffer[4]));
        PumpInfo.dDangerHeight = atof(EtcFunc.DeleteNull(strBuffer[7]));
        m_PumpInfo.Add(PumpInfo);
    }
    m_nPumpIndex = 0;
}

void CDlgRealTime::OnInitOwlInfo()
{
    m_OwlInfo.RemoveAll();
    if(! m_pDB->OrificeSelectQ())
        return ;
    int nRecordCount = m_pDB->m_Result.GetSize();
    COwlInfo OwlInfo;
    CEtcFunction EtcFunc;
    char strBuffer[4][100];
    for(int i = 0 ; i < nRecordCount ; i++){
        sscanf(m_pDB->m_Result[i],"%s %s %s %s",strBuffer[0],strBuffer[1],strBuffer[2],strBuffer[3]);
        OwlInfo.strName.Format("%s",EtcFunc.DeleteNull(strBuffer[0]));
        OwlInfo.dNum = atof(EtcFunc.DeleteNull(strBuffer[1]));
        OwlInfo.dC = atof(EtcFunc.DeleteNull(strBuffer[2]));
        OwlInfo.dA = atof(EtcFunc.DeleteNull(strBuffer[3]));
        m_OwlInfo.Add(OwlInfo);
    }
}

void CDlgRealTime::OnInitRepRead()
{
    CEtcFunction EtcFunc;
    CString strFileName=EtcFunc.GetExeFolder()+_T("\\data\\OWL\\beaver.rep");
    CStdioFile fp;
    if(! fp.Open( strFileName, CFile::modeRead ))
        AfxMessageBox("Beaver.dat File을 찾을 수 없습니다. 매뉴얼을 참조 하십시오.");
    return;
}

int nIndex = 0;
CString strData;
while(fp.GetPosition() < fp.GetLength()){
    fp.ReadString(strData);
}

```

```

        if(strData.Find(_T("W.S. Elev")) != -1){
            for(int i = 0 ; i < 2 ; i++){
                fp.ReadString(strData);
                while(fp.GetPosition() < fp.GetLength()){
                    fp.ReadString(strData);
                    if(strData.Find(_T("Kentwood")) != -1){
                        nIndex = 0;
                        fp.ReadString(strData);
                    }
                    char cTemp[125];
                    double dTemp1,dTemp2,dNum,dWsElev;
                    sscanf(strData,"%s %lf %lf %lf %lf",cTemp,
&dNum,&dTemp1,&dTemp2,&dWsElev);
                    switch(GetLocationNum(dNum)){
                        case 0: // 반여지구
                            if(m_ABan.GetSize() == nIndex)
                                m_ABan.Add(dWsElev);
                            else
                                m_ABan[nIndex] += dWsElev;
                            break;
                        }
                        nIndex++;
                    }
                }
            }
        }
        fp.Close();
    }
}

int CDlgRealTime::GetLocationNum(double dNum)
{
    int nCount = m_OwInfo.GetSize();
    for(int i = 0 ; i < nCount ; i++){
        if(m_OwInfo[i].dNum == dNum)
            return i;
    }
    return -1;
}

void CDlgRealTime::OnTBRealTimeFileOpen()
{
    CFileDialog dlgFileOpen(TRUE,"rms",NULL,OFN_ALLOWMULTI
SELECTION|OFN_FILEMUSTEXIST,
    "RMS Files(*.rms)|*.rms|DXF Files(*.dxf)|*.dxf|ALL Files(*.*)|*.*|",
    NULL);
    if(dlgFileOpen.DoModal() == IDOK){
        m_strFileName = dlgFileOpen.GetPathName();
        m_strFileName.MakeUpper(); // 대문자
        if((m_strFileName.Right(4)!= _T("RMS"))&&(m_strFileName.Right(4)!= _T(".DXF"))){
            AfxMessageBox((*RMS), (*DXF)파일만 가능합니다. 다시 선택해 주세요.);
            return;
        }
        CEtcFunction EtcFunc;
        EtcFunc.DisplayWaitStart();
        InitFileOpen();
        m_3DViewer.GraphicDataRead(m_strFileName);
        TinLayerOn();
        EtcFunc.DisplayWaitEnd();
    }
}

void CDlgRealTime::OnRDBMngBanya()
{
    CButton* pButton;
    pButton = (CButton*)GetDlgItem(IDC_RDBMngBanya);
    if(pButton->GetCheck() == 1){
        CEtcFunction EtcFunc;
        EtcFunc.DisplayWaitStart();
        OnInitEdit();
        InitFileOpen();
    }
}

```

```

        m_strFileName = EtcFunc.GetExcFolder() + _T("\\data\\rms\\차리적수지.rms");
        m_strFileName.MakeUpper(); // 대문자
        m_3DViewer.GraphicDataRead(m_strFileName);
        TinLayerOn();
        DisplayPumpInfo();
        EtcFunc.DisplayWaitEnd();
    }

void CDlgRealTime::DisplayPumpInfo()
{
    int nCount = m_lstPumpInfo.GetCount();
    for(int i = nCount ; i > 0 ; i--)
        m_lstPumpInfo.DeleteString(i);
    if(m_strFileName.Right(4) != _T(".RMS")){
        m_lstPumpInfo.AddString(_T("해당 펌프의 정보가 없습니다.));
        return;
    }
    m_nPumpIndex = 0;
    CEtcFunction EtcFunc;
    CString strFileName = EtcFunc.GetFileName(m_strFileName);
    strFileName.Replace(_T(".RMS"), "");
    nCount = m_PumpInfo.GetSize();
    for(i = 0 ; i < nCount ; i++){
        if(m_PumpInfo[i].strPumpName.Find(strFileName) != -1)
            DisplayPumpInfo(i);
    }
}

void CDlgRealTime::InitFileOpen()
{
    m_3DViewer.DataClear();
    m_3DViewer.SetWSValue(0.0);
    m_SlideRealTime.SetPos(100);
    m_dSWValue = 0.0;
}

void CDlgRealTime::OnTBRealTimeFit()
{
    m_3DViewer.ViewControlFit();
}

void CDlgRealTime::OnTBRealTimePan()
{
    m_3DViewer.ViewControlPan();
}

void CDlgRealTime::OnTBRealTimeZoomIn()
{
    m_3DViewer.ViewControlZoomIn();
}

void CDlgRealTime::OnTBRealTimeZoomOut()
{
    m_3DViewer.ViewControlZoomOut();
}

void CDlgRealTime::OnTBRealTimeRotate()
{
    m_3DViewer.ViewControlRotate();
}

void CDlgRealTime::TinLayerOn()
{
    if(m_strFileName.Right(4) != _T(".RMS"))
        return;
    int nLayerCount = m_3DViewer.GetLayerCount();
    for(int i = 0 ; i < nLayerCount ; i++){
        CString strLayerName = m_3DViewer.GetLayerName(i);
        if(strLayerName.Right(1) != 'P')
            m_3DViewer.SetLayerOff(i);
        if(strLayerName == _T("RIVERL"))
            m_3DViewer.SetLayerOn(i);
    }
}

```

```

    }
}
void CDlgRealTime::OnReleasedcaptureSLIDERRealTime(NMHDR* pNMHDR, LRESULT* pResult)
{
    OnInitEdit();
    int nWSValue = 100 - m_SlideRealTime.GetPos();
    m_3DViewer.SetWSValue(double) nWSValue);
    DisplayHightFloodInfo(nWSValue);
    m_SlideDivTime.SetPos(0);
    m_dSWValue = double(nWSValue);
    *pResult = 0;
}
void CDlgRealTime::DisplayHightFloodInfo(int nValue)
{
    m_strFloodHeight.Format("%.2F",double(nValue));
    m_cbFloodHeight.SetWindowText(m_strFloodHeight);
    m_strFloodArea.Format("%.2F",m_3DViewer.GetFloodArea(nValue));
    m_cbFloodArea.SetWindowText(m_strFloodArea);
    m_strWaterVolume.Format("%.2F",m_3DViewer.GetWaterVolume(nValue));
    m_cbWaterVolume.SetWindowText(m_strWaterVolume);
    m_strFloodVolume.Format("%.2F",m_3DViewer.GetFloodVolume(double(nValue)));
    m_cbFloodVolume.SetWindowText(m_strFloodVolume);
}
void CDlgRealTime::OnTBRealTimeFileSave()
{
    "RMS Files(*.rms)|*.rms|DXF Files(*.dxf)|*.dxf|ALL Files(*.*)|*.*||",
    NULL);
    if(dlgFileSave.DoModal() == IDOK){
        CString strFileName;
        strFileName = dlgFileSave.GetPathName();
        strFileName.MakeUpper(); // 대문자
        if(strFileName.Right(4) != _T(".RMS")){
            AfxMessageBox("(*.RMS) 파일만 가능합니다. 다시 입력해주세요);
            return;
        }
        CFunc EtcFunc;
        EtcFunc.DisplayWaitStart();
        MakeRmsFile(strFileName);
        EtcFunc.DisplayWaitEnd();
    }
}
void CDlgRealTime::MakeRmsFile(CString strFileName)
{
    if((m_strFileName.Right(4) != _T(".DXF"))){
        AfxMessageBox("입력 DATA가 (*.DXF)파일만 가능합니다. 다시 확인 후 저장 해주세요.");
        return;
    }
    int nLayerCount = m_3DViewer.GetLayerCount();
    for(int i = 0 ; i < nLayerCount ; i++){
        CString strLayerName = m_3DViewer.GetLayerName(i);
        if(strLayerName == _T("MADETINP"))
            break;
    }
    if(i == nLayerCount)
        AfxMessageBox("TIN DATA가 구성되지 않았습니디. TIN 구성 후 다시 저장해 주세요.");
    else
        m_3DViewer.MakeRms(strFileName);
}
void CDlgRealTime::OnTBRealTimeMakeTin()
{
    if((m_strFileName.Right(4) != _T(".DXF"))){
        AfxMessageBox("입력 DATA가 (*.DXF)파일만 가능합니다. 다시 확인 후 저장해 주세요.");
        return;
    }
    int nLayerCount = m_3DViewer.GetLayerCount();
}

```

```

for(int i = 0 ; i < nLayerCount ; i++){
    CString strLayerName = m_3DViewer.GetLayerName(i);
    if(strLayerName == _T("MADETINP"))
        break;
}
CEtcFunction EtcFunc;
EtcFunc.DisplayWaitStart();
if( i == nLayerCount)
    m_3DViewer.CreateTIN();
else
    AfxMessageBox("아미 TIN DATA가 구성되어 있습니다.");
EtcFunc.DisplayWaitEnd();
}
void CDlgRealTime::OnTBRealTimeList()
{
    CEtcFunction EtcFunc;
    EtcFunc.DisplayWaitStart();
    m_strResult.RemoveAll();
    m_3DViewer.SetWSVValue(m_dSWVValue);
    int nCount = m_3DViewer.SearchFloodBuild();
    for(int i = 0 ; i < nCount ; i++){
        int nValue = m_3DViewer.GetFloodBuild(i);
        SearchFloodBuild(nValue);
    }
    EtcFunc.DisplayWaitEnd();
    MakeFloodBuildList();
}
void CDlgRealTime::SearchFloodBuild(int nValue)
{
    if(! m_pDB->AddressSelectQ(nValue))
        return ;
    int nRecordCount = m_pDB->m_Result.GetSize();
    for(int i = 0 ; i < nRecordCount ; i++){
        CString strTemp = m_pDB->m_Result[i];
        m_strResult.Add(strTemp);
    }
}
void CDlgRealTime::OnBUTResult()
{
    if(! BUTResultCheck())
        return;
    OnInitEdit();
    CEtcFunction EtcFunc;
    EtcFunc.DisplayWaitStart();
    double dHeight = m_3DViewer.GetFloodDeep(GetFloodExtinction());
    OnInitSlideDivTime();
    int nZValue = int(m_3DViewer.GetMaxElevation());
    int nWSVValue = int(dHeight * 100 / nZValue);
    m_SlideRealTime.SetPos(100 - nWSVValue);
    m_3DViewer.SetWSVValue((double) dHeight);
    m_dSWVValue = dHeight;
    EtcFunc.DisplayWaitEnd();
}
bool CDlgRealTime::BUTResultCheck()
{
    if(m_strFileName.Right(4) != _T(".RMS")){
        AfxMessageBox("(*.RMS)파일만 가능합니다. RMS 파일을 Display한 후 다시 시도해 주세요.");
        return false;
    }
    m_cbFlood.GetWindowText(m_strFlood);
    if(atoi(m_strFlood) == 0.0){
        AfxMessageBox("강우량 DATA가 부정확 합니다. 강우량을 정확히 입력해 주세요.");
        return false;
    }
    m_cbTime.GetWindowText(m_strTime);
}

```

```

    if(atof(m_strTime) -- 00){
        AfxMessageBox("지속시간 DATA가 부정확 합니다. 지속시간을 정확히 입력해 주세요.");
        return false;
    }
    if(m_cmbAmc.GetCurSel() == 0){
        AfxMessageBox("AMC DATA가 부정확 합니다. AMC DATA를 정확히 입력해 주세요.");
        return false;
    }
    return true;
}
void CDlgRealTime::OnReleasedcaptureSLIDERDivTime(NMHDR* pNMHDR, LRESULT* pResult)
{
    int nCount = m_ElevM.GetSize();
    if(nCount == 0)
        return;
    CEtcFunction EtcFunc;
    EtcFunc.DisplayWaitStart();
    int nIndex = int(m_SlideDivTime.GetPos() / 10);
    if(m_chkPump.GetCheck() == 0){
        m_3DViewer.SetWSValue(m_ElevM[nIndex]);
        m_dSWValue = m_ElevM[nIndex];
    }else{
        m_3DViewer.SetWSValue(m_APElevM[nIndex]);
        m_dSWValue = m_APElevM[nIndex];
    }
    DisplayTimeFloodInfo(nIndex);
    m_SlideRealTime.SetPos(100);
    EtcFunc.DisplayWaitEnd();
    *pResult = 0;
}
void CDlgRealTime::DisplayTimeFloodInfo(int nIndex)
{
    if(m_chkPump.GetCheck() == 0){ // 펌프미작용
        if(m_ElevM[nIndex] < 0.0)
            m_strFloodHeight.Format("0.0");
        else
            m_strFloodHeight.Format("%10.2f",m_ElevM[nIndex]);
        m_cbFloodHeight.SetWindowText(m_strFloodHeight);
        if(m_ElevM[nIndex] < 0.0)
            m_strFloodArea.Format("0.0");
        else
            m_strFloodArea.Format("%10.2f",m_3DViewer.GetFloodArea(m_ElevM[nIndex]));
        m_cbFloodArea.SetWindowText(m_strFloodArea);
        if(m_Volume[nIndex] < 0.0)
            m_strWaterVolume.Format("0.0");
        else
            m_strWaterVolume.Format("%10.2f",m_Volume[nIndex]);
        m_cbWaterVolume.SetWindowText(m_strWaterVolume);
        double dFloodVolume=m_3DViewer.GetFloodVolume(m_ElevM
[nIndex]);
        if(dFloodVolume < 0.0)
            m_strFloodVolume.Format("0.0");
        else
            m_strFloodVolume.Format("%10.2f",dFloodVolume);
        m_cbFloodVolume.SetWindowText(m_strFloodVolume);
    } else {
        if(m_APElevM[nIndex] < 0.0)
            m_strFloodHeight.Format("0.0");
        else
            m_strFloodHeight.Format("%10.2f",m_APElevM[nIndex]);
        m_cbFloodHeight.SetWindowText(m_strFloodHeight);
        if(m_APElevM[nIndex] < 0.0)
            m_strFloodArea.Format("0.0");
        else
            m_strFloodArea.Format("%10.2f",m_3DViewer.GetFloodArea(m_APElevM[nIndex]));
    }
}

```

```

        m_cbFloodArea.SetWindowText(m_strFloodArea);
        if(m_APVolume[nIndex] < 0.0)
            m_strWaterVolume.Format("0.0");
        else
            m_strWaterVolume.Format("%.2f",m_APVolume[nIndex]);
        m_cbWaterVolume.SetWindowText(m_strWaterVolume);
        double dFloodVolume = m_3DViewer.GetFloodVolume(m_APElevM[nIndex]);
        if(dFloodVolume < 0.0)
            m_strFloodVolume.Format("0.0");
        else
            m_strFloodVolume.Format("%.2f",dFloodVolume);
        m_cbFloodVolume.SetWindowText(m_strFloodVolume);
    }
}
void CDlgRealTime::OnInitSlideDivTime()
{
    int nCount = m_Volume.GetSize();
    m_SlideDivTime.SetTicFreq(10);
    m_SlideDivTime.SetPageSize(10);
    m_SlideDivTime.SetRangeMin(0);
    m_SlideDivTime.SetRangeMax(int(nCount*10));
    m_SlideDivTime.SetPos(0);
}
double CDlgRealTime::GetFloodExtinction()
{
    OnInitExecuteFlood();
    ExecuteFlood();
    CalWaterElevation();
    return m_Volume[0];
}
void CDlgRealTime::OnInitExecuteFlood()
{
    m_cbTime.GetWindowText(m_strTime);
    double dTotalTime = double(int(atof(m_strTime) * 60. / 10) * 10); // 시 -> 분
    m_cbFlood.GetWindowText(m_strFlood);
    double dTotalWater = atof(m_strFlood);
    m_HuffInfo[0].st = 0.0;
    m_HuffInfo[0].et = dTotalTime * 0.1; // 10 %
    m_HuffInfo[0].Ratio = 4.4 * dTotalWater / 100.;
    m_HuffInfo[1].st = m_HuffInfo[0].et;
    m_HuffInfo[1].et = dTotalTime * 0.2; // 20 %
    m_HuffInfo[1].Ratio = 8.2 * dTotalWater / 100.;
    m_HuffInfo[2].st = m_HuffInfo[1].et;
    m_HuffInfo[2].et = dTotalTime * 0.3; // 30 %
    m_HuffInfo[2].Ratio = 15.3 * dTotalWater / 100.;
    m_HuffInfo[3].st = m_HuffInfo[2].et;
    m_HuffInfo[3].et = dTotalTime * 0.4; // 40 %
    m_HuffInfo[3].Ratio = 20.5 * dTotalWater / 100.;
    m_HuffInfo[4].st = m_HuffInfo[3].et;
    m_HuffInfo[4].et = dTotalTime * 0.5; // 50 %
    m_HuffInfo[4].Ratio = 20.0 * dTotalWater / 100.;
    m_HuffInfo[5].st = m_HuffInfo[4].et;
    m_HuffInfo[5].et = dTotalTime * 0.6; // 60 %
    m_HuffInfo[5].Ratio = 13.6 * dTotalWater / 100.;
    m_HuffInfo[6].st = m_HuffInfo[5].et;
    m_HuffInfo[6].et = dTotalTime * 0.7; // 70 %
    m_HuffInfo[6].Ratio = 7.7 * dTotalWater / 100.;
    m_HuffInfo[7].st = m_HuffInfo[6].et;
    m_HuffInfo[7].et = dTotalTime * 0.8; // 80 %
    m_HuffInfo[7].Ratio = 4.5 * dTotalWater / 100.;
    m_HuffInfo[8].st = m_HuffInfo[7].et;
    m_HuffInfo[8].et = dTotalTime * 0.9; // 90 %
    m_HuffInfo[8].Ratio = 3.6 * dTotalWater / 100.;
    m_HuffInfo[9].st = m_HuffInfo[8].et;
    m_HuffInfo[9].et = dTotalTime * 1.0; // 100 %
}

```

```

        m_HuffInfo[9].Ratio = 2.2 * dTotalWater / 100;
    }
    void CDlgRealTime::ExecuteFlood()
    {
        m_ebTime.GetWindowText(m_strTime);
        int nTotalTime = int(atoi(m_strTime) * 60. / 10); // 시 -> 분
        m_ebFlood.GetWindowText(m_strFlood);
        CEtcFunction EtcFunc;
        CString strInputFile, strInputTemp;
        switch(GetDistrict())
        {
            case 1:
                strInputFile = EtcFunc.GetExcFolder() + _T("\\data\\flood\\floodb.dat");
                strInputTemp = EtcFunc.GetExcFolder() + _T("\\data\\flood\\flood.dat");
                break;
            case 2:
                strInputFile = EtcFunc.GetExcFolder() + _T("\\data\\flood\\floods.dat");
                strInputTemp = EtcFunc.GetExcFolder() + _T("\\data\\flood\\flood.dat");
                break;
        }
        CStdioFile fp1, fp2;
        if(! fp1.Open( strInputFile, CFile::modeRead ))
            AfxMessageBox("Flood Input File을 찾을 수 없습니다. 메뉴얼을 참조하십시오.");
            return;
        }
        if(! fp2.Open( strInputTemp, CFile::modeCreate | CFile::modeWrite ))
            AfxMessageBox("Flood Input File을 생성할 수 없습니다. 메뉴얼을 참조하십시오.");
            return;
        }
        CString strData;
        while(fp1.GetPosition() < fp1.GetLength()){
            fp1.ReadString(strData);
            fp2.WriteString(strData + _T("\n"));
            if(strData.Find(_T("Mononobe time distribution")) != -1){
                for(int i = 0 ; i < 2 ; i++){
                    fp1.ReadString(strData);
                    fp2.WriteString(strData + _T("\n"));
                }
                fp1.ReadString(strData);
                char cTemp[10][10];
                sscanf(strData, "%s %s %s %s %s %s %s %s %s",
                    cTemp[0], cTemp[1], cTemp[2], cTemp[3], cTemp[4],
                    cTemp[5], cTemp[6], cTemp[7]);
                int nCount;
                if((atoi(cTemp[1]) / 10.) == (atoi(cTemp[1]) / 10))
                    nCount = atoi(cTemp[1]) / 10;
                else
                    nCount = atoi(cTemp[1]) / 10 + 1;
                for(i = 0 ; i < nCount ; i++){
                    fp1.ReadString(strData);
                    sprintf(cTemp[1], "%d", nTotalTime + 1); // 강우량 개수
                    sprintf(cTemp[4], "%d", nTotalTime * 10);
                    sprintf(cTemp[7], "%d", m_embAme.GetCurSel());
                    strData.Format("%10s%10s%10s%10s%10s%10s%10s%10s%10s%10s",
                        cTemp[3], cTemp[4], cTemp[5], m_strFlood, cTemp[7]);
                    fp2.WriteString(strData);
                    strData = _T("");
                    CString strTemp;
                    for(i = 0 ; i < nTotalTime + 1 ; i++){
                        double dNewWater = GetNewWaterPerMin(i * 10);
                        strTemp.Format("%8.2lf", dNewWater);
                        strData = strData + strTemp;
                        if((i != 0) && ((i % 9) == 0)){
                            strData = strData + _T("\n");
                            fp2.WriteString(strData);
                        }
                    }
                }
            }
        }
    }
}

```

```

                strData = _T("");
            }
            if(strData.GetLength() != 0){
                strData = strData + _T("\n");
                fp2.WriteString(strData);
                strData = _T("");
            }
        }
        fp1.Close();
        fp2.Close();
        CString strExecName=EtcFunc.GetExecFolder()+_T("%data\\flood\\flood_ exe");
        CString strOldFolder = EtcFunc.GetCurrentFolder();
        CString strCurrentFolder=EtcFunc.GetExecFolder()+_T("%data\\flood");
        EtcFunc.SetCurrentFolder(strCurrentFolder);
        FloodExecute(strExecName);
        CString strDelFile =EtcFunc.GetExecFolder()+_T("%data\\flood\\LLUDA_ S.S");
        DeleteFile(strDelFile);
        EtcFunc.SetCurrentFolder(strOldFolder);
    }
void CDlgRealTime::FloodExecute(CString strExecName)
{
    SHELLEXECUTEINFO sei;
    ZeroMemory(&sei, sizeof(SHELLEXECUTEINFO));
    sei.cbSize = sizeof(SHELLEXECUTEINFO);
    sei.lpFile = strExecName;
    sei.lpVerb = _T("open");
    sei.nShow = SW_HIDE;
    sei.fMask = SEE_MASK_NOCLOSEPROCESS;
    ShellExecuteEx(&sei);
    ::WaitForSingleObject(sei.hProcess, INFINITE);
}
int CDlgRealTime::GetDistrict()
{
    CButton* pButton;
    pButton = (CButton*)GetDlgItem(IDC_RDBMngBanya);
    if(pButton->GetCheck() == 1)
        return 1;
    pButton = (CButton*)GetDlgItem(IDC_RDBMngSuYoung);
    if(pButton->GetCheck() == 1)
        return 2;
    return 0;
}
double CDlgRealTime::GetNewWaterPerMin(int nValue)
{
    if(nValue == 0)
        return 0.0;
    m_cbTime.GetWindowText(m_strTime);
    int nTotalTime = int(atof(m_strTime) * 60. / 10);
    for(int i = 0 ; i < 10 ; i++){
        if((m_HuffInfo[i].st <= nValue) && (m_HuffInfo[i].et >= nValue))
            break;
    }
    double dTotalValue;
    int nResult1 = int(nValue - m_HuffInfo[i].st);
    if(nResult1 > 10)
        return 10 * m_HuffInfo[i].Ratio / nTotalTime;
    dTotalValue = nResult1 * m_HuffInfo[i].Ratio / nTotalTime;
    int nResult2 = 10 - nResult1;
    int nCount = nResult2 / nTotalTime;
    for(int j = 0 ; j < nCount ; j++){
        dTotalValue = dTotalValue + m_HuffInfo[ i].Ratio;
        nResult2 = nResult2 - nTotalTime;
    }
}

```

```

        dTotalValue=dTotalValue+(nResult2*m_HuffInfo[ i].Ratio/nTotalTime);
    return dTotalValue;
}
void CDlgRealTime::CalWaterElevation()
{
    m_Volume.RemoveAll();
    m_OElevM.RemoveAll();
    m_OVVolume.RemoveAll();
    CEtcFunction EtcFunc;
    CString strOutputFile;
    strOutputFile= EtcFunc.GetExeFolder()+ _T("\\data\\flood\\flood.out");
    CStdioFile fp;
    if(! fp.Open( strOutputFile, CFile::modeRead )){
        AfxMessageBox("Flood Output File을 찾을 수 없습니다. 메뉴얼을 참조하십시오.");
        return;
    }
    int nIndex = 0;
    double dVolume[10];
    CString strData;
    while(fp.GetPosition() < fp.GetLength()){
        fp.ReadString(strData);
        if(strData.Find(_T("Outfall Hydrograph in cms")) != -1){
            nIndex = 0;
            fp.ReadString(strData);
        }
        for(int i = 0 ; i < 10 ; i++){
            dVolume[i] = 0.0;
            sscanf(strData,"%lf %lf %lf %lf %lf %lf %lf %lf %lf %lf
%lf",&dVolume[0],&dVolume[1],&dVolume[2],&dVolume
[3],&dVolume[4],&dVolume[5],&dVolume[6],&dVolume[7],
&dVolume[8],&dVolume[9]);
            for(i = 0 ; i < 10 ; i++){
                if(dVolume[i] >= 0.0){
                    if(m_Volume.GetSize() != nIndex)
                        m_Volume[nIndex]= m_Volume[nIndex] + dVolume[i];
                    else
                        m_Volume.Add(dVolume[i]);
                }
                nIndex ++;
            }
        }
        fp.Close();
        m_ElevM.RemoveAll(); // H
        int nVolumeCount = m_Volume.GetSize();
        for(int j = 0 ; j < nVolumeCount ; j++){
            double dEleV = m_3DViewer.GetFloodDeep(m_Volume[j]);
            m_ElevM.Add(dEleV);
            m_OElevM.Add(dEleV);
            m_OVVolume.Add(m_Volume[j]);
        }
    }
}
double CDlgRealTime::GetFloodDeep()
{
    m_ElevM.RemoveAll();
    m_Volume.RemoveAll();
    OnInitExecuteFlood();
    double dTotalArea = m_3DViewer.GetTotalLandArea();
    double dBuildArea = m_3DViewer.GetBuildArea();
    m_cbTime.GetWindowText(m_strTime);
    int nTotalTime = int(atof(m_strTime) * 60. / 10); // 시 > 분
    m_cbFlood.GetWindowText(m_strFlood);
    double dTotalWater = atof(m_strFlood);
    double dNewWater = 0.0;
    for(int i = 0 ; i < nTotalTime-1 ; i++){
        dNewWater = dNewWater + GetNewWaterPerMin(i) * 10; *

```

```

((dNewWater*(i * 10)) * 60) * (dTotalArea / 1000000.);
    m_Volume.Add(dHeight);
    double dVolume = m_3DViewer.GetFloodDeep(dHeight);
    m_ElevM.Add(dVolume);
}
ApplyPump();
return m_ElevM[0];
}
double CDlgRealTime::GetConstantValue()
{
    m_cbFlood.GetWindowText(m_strFlood);
    double dTotalWater = atof(m_strFlood);
    double dT = 0.833 * (2.39 / pow(0.17,0.6)) / 60;
    double dR = (dTotalWater / 24) * pow((24 / dT),2/3);
    return dR;
}
void CDlgRealTime::ApplyPump()
{
    m_APElevM.RemoveAll();
    m_APVolume.RemoveAll();
    int nIndex = 0;
    double dElevM = 0.0;
    int nStart = -1,nEnd = -1;
    int nCount = m_ElevM.GetSize();
    for(int i = 0 ; i < nCount ; i++){
        m_APElevM.Add(m_ElevM[i]);
        m_APVolume.Add(m_Volume[i]);
        double k1 = m_ElevM[i];
        if(m_PumpInfo[m_nPumpIndex].dDangerHeight < m_ElevM[i]){
            if(nStart == -1)
                nStart = i - 1;
        }
        if(dElevM < m_ElevM[i]){
            nIndex = i;
            dElevM = m_ElevM[i];
        }
        if(m_PumpInfo[m_nPumpIndex].dDangerHeight > m_ElevM[i]){
            if((nStart != -1) && (nEnd == -1))
                nEnd = i;
        }
    }
    if(nStart == -1)
        return;
    if(nEnd == -1)
        nEnd = nCount - 1;
    int nPumpNum = atoi(m_PumpInfo[m_nPumpIndex].strPumpNum);
    double dVolume,dHeight;
    for(i = 0 ; i < nPumpNum + 1 ; i++){
        dVolume=m_Volume[nIndex]+(m_PumpInfo[m_nPumpIndex].dVolume*10*(i - 1));
        dHeight = m_3DViewer.GetFloodDeep(dVolume);
        if(m_PumpInfo[m_nPumpIndex].dDangerHeight > dHeight)
            break;
    }
    if(i == (nPumpNum + 1))
        m_nPumpNum = - 1;
    m_nStartTime = nStart * 10;
    m_nEndTime = nEnd * 10;
    m_nPumpNum = i + 1;
    for(i = nStart ; i <= nEnd ; i++){
        double k = m_Volume[i];
        dVolume=m_Volume[i]+(m_PumpInfo[m_nPumpIndex].dVolume*10)*
m_nPumpNum;
        m_APVolume[i] = dVolume;
        dHeight = m_3DViewer.GetFloodDeep(dVolume);
        m_APElevM[i] = dHeight;
    }
}

```

```

    }
}
void CDlgRealTime::ApplyBanOwl()
{
    int nCount = m_OElevM.GetSize();
    int nOwlIndex = GetOwlIndex(_T("한약"));
    double dVolume = 0.0;
    for(int i = 0 ; i < nCount ; i++){
        double k1 = m_OElevM[i];
        double k2 = m_ABan[i];
        if(m_OElevM[i] > m_ABan[i]){
            dVolume = m_OwlInfo[nOwlIndex].dC*m_OwlInfo[nOwlIndex].dA *
            sqrt(2*9.8*(m_OElevM[i] - m_ABan[i]));
            m_Volume[i] = m_OVolume[i] - dVolume;
            m_ElevM[i] = m_3DViewer.GetFloodDeep(m_Volume[i]);
        }
    }
}
}

```

감사의 글

항상 마음 한구석에 가지고 있던 토목에 대한 학업 미련을 떨치지 못하고 배움에 대한 욕심으로 대학원에 진학 한지도 엇그제 같은데 벌써 2년 6개월이란 세월이 지나 논문을 쓰게되니 지나간 학교 생활에 대한 아쉬움과 감회가 새롭습니다.

이번 기회를 통하여 학문연구가 얼마나 힘들고 어려운 길인가 하는 것을 절실히 느끼고 깨달으면서 앞으로의 삶에 새로운 활력을 얻게된 것은 참으로 다행스러운 일이라 여깁니다

먼저 여러 가지로 부족한 저에게 새로운 배움의 길을 열어 학문에 대한 바른 자세와 진리를 일깨워 주시고 본 논문이 완성되기까지 아낌없는 사랑과 정성으로 세심하게 지도해 주신 김상용 지도교수님께 진심으로 머리 숙여 감사 드립니다.

그리고 바쁘신 와중에도 본 논문이 알찬 결실을 맺을 수 있도록 면밀히 검토해 주시고 바로잡아 주신 손인식 교수님, 이환우 교수님께 진심으로 감사 드리며, 또한 깊은 학문적 가르침을 주신 김종수 교수님, 장희석 교수님, 이동욱 교수님, 이종출 교수님, 이영대 교수님, 김명식 교수님, 이종섭 교수님, 정진호 교수님, 김수용 교수님, 정두회 교수님, 국승규 교수님, 이상호 교수님께 존경과 깊은 감사를 드립니다.

본 논문을 위하여 바쁜 시간 중에도 최선을 다해 아낌없이 성심껏 도와준 환경수리연구실 정인주 박사님, 조용재, 채종훈, 육승우, 조희호에게 고마움을 전하며, 그리고 항상 아낌없는 조언과 배움에 길을 열어주신 노수상, 손차범 선배님께 진심으로 감사 드리며, 특히 2년 6개월 동안 동문수학하면서 서로에게 힘이 되어 주었던 김문수, 박봉규, 김재호님께 감사 드리며, 앞날에 영광이 있기를 기원합니다.

또한 학교생활에 지장이 없도록 여러 가지로 배려하여 주시고 격려해주신 직장 상사, 동료 직원을 비롯한 모든 분들께 진심으로 감사 드립니다.

무엇보다도 가정살림과 직장생활 등 여러 가지 어려운 환경에서도 석
민이와 정언이가 바르게 성장하도록 최선을 다하면서 힘들어 할 때마다 용기
를 북돋아준 사랑하는 아내 김분옥과 효도 한번 제대로 못하고 막내로서 응석만
부렸지만 항상 자식 잘 되기만을 기원하시는 어머님께 이 논문을 바치고 싶습
니다.

2004년 7월

김선용 올림