

지반정보로부터 3차원 가시화 프로그램 개발에 관한 연구*

이봉준^{1*} · 민홍² · 고훈준³

A Study on the Development of a 3D Visualization Program from Geotechnical Information*

Bong-Jun LEE^{1*} · Hong MIN² · Hoon-Joon KOUH³

요 약

시추데이터는 작업자가 작업현장에서 안전하게 공사를 수행할 수 있도록 제공하는 지반정보로, 현재 시추데이터를 3차원 데이터로 만들어서 3차원 형태의 이미지로 볼 수 있도록 지원하고 있다. 지반정보의 3차원 가시화를 이용하여 다양한 프로그램을 개발하는 국내의 기업들은 지반 정보의 3차원 가시화를 위해 C Tech Development Corporation에서 개발한 MVS 프로그램을 사용하고 있다. 그러나 MVS 프로그램은 유료 프로그램이고 국내에서 개발하는 3차원 관련 프로그램에서 사용하기에는 어려움이 있다. 본 논문에서는 파이썬의 Gempy 오픈 소스를 이용하여 군집화된 시추정보로부터 3차원 지층모델을 생성할 수 있도록 MVS를 대체할 수 있는 라이브러리를 개발하고자 한다. 3차원 지층모델 프로그램은 각 지층별 포인트 데이터를 생성하고 보간을 통해서 지층별 표면을 생성한다. 그리고 각 지층별 표면을 합하여 3차원 지층모델 프로그램을 완성한다. 고양 지역의 시추데이터로부터 MVS 프로그램과 제안하는 프로그램으로 3차원 모형을 생성하여 비교하였을 때 큰 차이가 없음을 확인하였다.

주요어 : 지반정보, 시추데이터, 3차원 가시화, 보간, 데이터 크리깅

ABSTRACT

Borehole Data is geotechnical information provided so that workers can safely perform construction at the field. It creates 3D data and supports viewing as a 3D image. Currently, all Korean companies that develop programs using 3D visualization use the MVS program developed by C Tech Development Corporation. However, the MVS program is a commercial program, and it is difficult to use MVS in 3D related programs

2022년 11월 01일 접수 Received on November 01, 2022 / 2022년 11월 09일 수정 Revised on November 09, 2022 / 2022년 11월 17일 심사완료 Accepted on November 17, 2022

* 본 연구는 국토교통부/국토교통과학기술진흥원의 지원으로 수행되었음(과제번호 : 21DCRU-B158159-02)

1 (주)씨엠월드 대표 / CEO, CmWorld Corporation

2 (주)씨엠월드 연구원 / Research Engineer, CmWorld Corporation

3 경인여자대학교 교수 / Professor, Kyung-In Women's University

※ Corresponding Author E-mail: bjlee@cmworld.co.kr

developed by Korean Companies. In this paper, we propose to develop a program that can replace MVS to generate a 3D stratum model from clustered borehole information using Python's Gempy open-source. The 3D stratum model program can create point data for each stratum and can create a surface for each stratum through interpolation. Then, the 3D stratum model program is completed by merging the surfaces of each stratum. It was confirmed that there was no difference when a 3D model was created and compared with the MVS program and the proposed program from the borehole data of a Goyang area.

KEYWORDS : *Geotechnical Information, Borehole Data, 3D Visualization, Interpolation, Data Kriging*

서론

지반정보는 건설공사 및 유지보수에 수반되는 지반의 공학적인 특성을 규명하여 계획입안자, 시공자, 설계자, 감리 및 감독자 등에게 제공됨으로써 안전하고 경제적인 공사를 수행할 수 있도록 지하 시설물 관리에 필수적인 요소이다. 과거에는 지반정보를 문서로 보관하다가 컴퓨터 기술의 발달로 2차원 데이터로 관리되어왔으나 2차원 데이터는 지층별 깊이나 변화를 확인하기에는 어려움이 많았다. 또한 공간 데이터를 관리하는 시스템은 현장에서 변경되는 공간 데이터를 수집하고 서비스하고 관리하는 것이 쉽지 않다(Lee, 2021). 지하공간 데이터는 빅데이터이다(Li *et al.*, 2016). 따라서 이러한 빅데이터의 효율적인 관리와 활용을 위한 연구들이 진행되고 있으며 몇 년 전부터는 지반정보를 3차원 데이터로 만들어 관리하기 시작하였다. 이러한 연구는 기존 문서로 제공되던 지반정보를 3차원 데이터로 제공되었을 때 지하공간을 공사하는 작업자는 지반정보를 3차원 모형으로 쉽게 확인할 수 있기 때문에 지하시설물의 관리를 효율적으로 할 수 있다(Kim and Kouh, 2022; Park *et al.*, 2018).

2015년부터 지하정보 활용지원센터(Underground Information Utilization Support Center)에서는 전국에 있는 지하시설물을 효율적으로 관리하기 위해서 지하공간에 있는 모든 지하시설물

을 통합 관리하는 시스템(Kim and Han, 2019; MOLIT, 2020; Park. *et al.*, 2018)을 개발하기 시작하였다. 이 시스템은 모든 지하시설물을 3차원 데이터로 만들어 관리하고 있으며 지하시설물의 3차원 데이터를 활용하기 위해서 지반정보 또한 3차원 데이터로 만들어 관리하고 있다. 따라서 현장에서 작업자는 지하시설물의 위치를 쉽게 파악할 수 있고 지반정보를 쉽게 이해하여 작업을 안전하고 빠르게 진행할 수 있다(Lee and Kouh, 2022; Seok *et al.*, 2012).

지반정보는 시추데이터로 구성되고 시추데이터를 3차원 이미지로 만들기 위해서는 지금까지 MVS(Mining Visualization System) 외산 프로그램을 가장 많이 사용하고 있었으나 유료 프로그램이고 2015년 12월 이후 업데이트가 중단되어 국내에서 지반정보를 3차원으로 보여주는 관련 프로그램을 개발할 때 개발 환경에 따라 MVS의 라이브러리를 사용하는데 어려움이 있었다. 따라서 국내에서 MVS 프로그램 라이브러리를 대신할 수 있는 프로그램을 개발하는 것이 필요하다.

본 논문에서는 파이썬(Python)의 오픈소스 라이브러리 GemPy를 기반으로 MVS의 기능을 대신할 수 있는 3차원 가시화 프로그램 라이브러리를 개발하고자 한다.

이를 위하여 시추군집데이터분석에서는 3차원 가시화 라이브러리 개발을 위해 본 논문에서 사용할 시추데이터의 구조와 특징을 설명하고 3차원 가시화 모델을 위한 프로그램 설계에서는 시추데이터를 3차원 이미지로 가시화하기 위해

본 논문에서 제안하는 방법을 서술한다. 표면 말림 현장에서는 본 논문에서 제안하는 프로그램 라이브러리 개발에서 발생하는 표면 말림현상을 해결하는 방법을 제안하고 MVS와 Borehole2Model의 비교에서는 MVS와 본 논문에서 개발한 프로그램으로 가시화한 3D 모형 이미지를 비교 설명한다. 마지막으로 결론을 서술한다.

시추군집데이터 분석

시추데이터는 시추공의 위치를 기반으로 만들어지며 일반적으로 지역적으로 균등하게 분포되어 있지 않다(Lee *et al.*, 2017). 따라서 3차원 모델링으로 만들기 위해서 사용하는 기초 데이터는 단일 시추데이터로 구성할 수 없고 여러 개 시추데이터를 군집으로 만들어야만 한다. 따라서 그림 1과 같이 주변의 시추데이터와 결합하여 군집데이터 형태로 구성한 정보 즉, 시추

군집데이터를 사용한다.

시추군집데이터는 각 시추의 위치를 나타내는 시추공 X좌표, 시추공 Y좌표, 지층의 시작깊이, 지층의 끝 깊이, 시추공 총 깊이, 시추공 시작 고도, 지층명, 시추구경 등을 포함하고 있으며, 지층의 종류는 매립토층, 퇴적토층, 봉적토층, 층적토층, 잔적토층, 풍화토층, 풍화암층, 연암층(보통암), 경암층, 극강암으로 총 10개의 지층으로 구성되어있다.

표 1은 고양시 일부 지역의 군집데이터 구조로, 군집 내에 시추공 코드로 구분된 지층별 높이 정보를 기반으로 지표로부터 아래쪽으로 지층을 쌓아 가는 구조로 되어 있다. 표 1에서 START는 Borehole height - Section starting depth를 의미하고 END는 Borehole height - Section end depth를 의미하며 실제 지층의 해발고도 높이를 나타낸다. 따라서 시추데이터에서 END 값이 다음 구간의 START 값과 동일

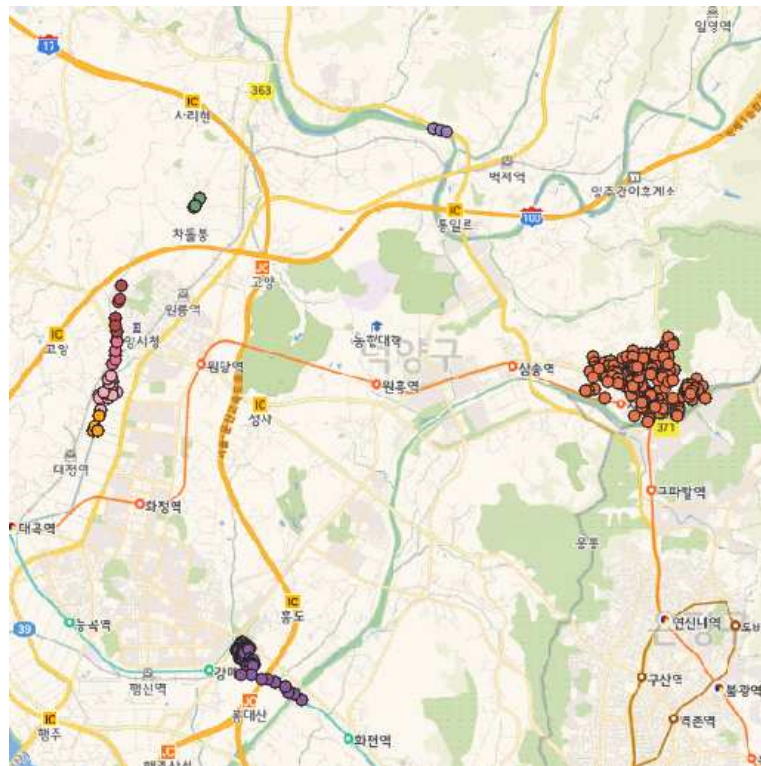


FIGURE 1. Borehole Cluster Data of Goyang City Area

TABLE 1. Cluster Data Structure

Borehole X coordinate (EPSG:5186)	Borehole Y coordinate (EPSG:5186)	Section start depth (m)	Section end depth (m)	Borehole total depth (EL.m)	Borehole height (EL.m)	Korea stratum name1)	START (EL.m)	END (EL.m)
186123.7087	564006.6486	0	1.9	15.8	36.7	매립토층	36.7	34.8
186123.7087	564006.6486	1.9	3.7	15.8	36.7	퇴적토층	34.8	33
186123.7087	564006.6486	3.7	5.2	15.8	36.7	퇴적토층	33	31.5
186123.7087	564006.6486	5.2	6.7	15.8	36.7	퇴적토층	31.5	30
186123.7087	564006.6486	6.7	11	15.8	36.7	풍화토층	30	25.7
186123.7087	564006.6486	11	12.8	15.8	36.7	풍화암층	25.7	23.9
186123.7087	564006.6486	12.8	15.8	15.8	36.7	연암층	23.9	20.9
186056.575	563903.9166	0	0.6	16	36.63	매립토층	36.63	36.03
186056.575	563903.9166	0.6	1	16	36.63	퇴적토층	36.03	35.63
186056.575	563903.9166	1	4.5	16	36.63	퇴적토층	35.63	32.13
186056.575	563903.9166	4.5	10.5	16	36.63	풍화토층	32.13	26.13
186056.575	563903.9166	10.5	13	16	36.63	풍화암층	26.13	23.63
186056.575	563903.9166	13	16	16	36.63	연암층	23.63	20.63

1) Since Korea stratum name is input data, it is written in Korean

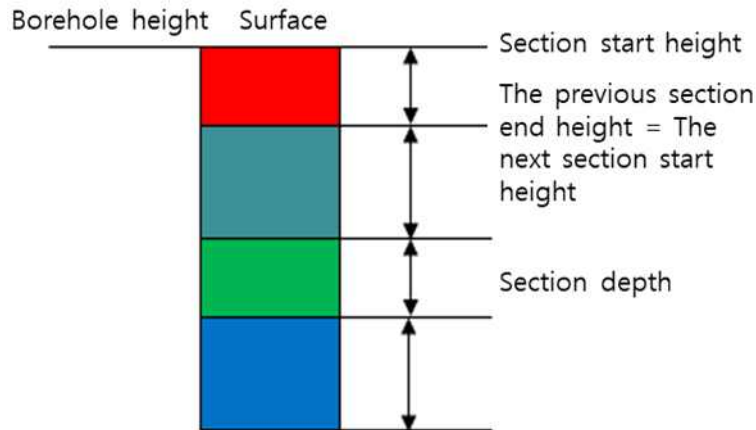


FIGURE 2. Borehole data structure

한 값을 가지게 되며 지층은 그림 2와 같은 구조를 가진다.

3차원 가시화 모델을 위한 프로그램 설계

본 장에서는 시추군집데이터로부터 지층 구조를 3차원 이미지로 생성할 수 있는 Borehole2Model 을 제안하고자 한다. 본 논문에서 제안하는

Borehole2Model은 3차원 가시화 모델을 생성하기 위해 파이썬 라이브러리를 사용하며 지층 모델 생성 과정은 그림 3과 같다.

그림 3과 같이 시추군집데이터를 입력하면 시추데이터로부터 확정된 지점의 정보를 이용하여 각각의 지층별 포인트 데이터를 생성하고 기초 포인트를 이용하여 군집데이터를 포함하는 영역의 추정 값을 추출하는 보간(Interpolation)을 진행한다. 그리고 보간이 된 데이터를 이용

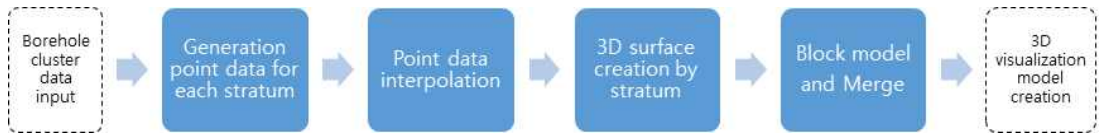


FIGURE 3. 3D visualization model creation process in Borehole2Model

하여 동일 지층을 가지는 표면 데이터를 생성하고 각 지층별 표면 데이터를 블록(Block)으로 만들고 병합하여 하나의 지층인 3차원 가시화 모델을 완성한다.

1. 각 지층별 포인트 데이터 생성

본 논문에서 제안하는 Borehole2Model에서 사용하는 포인트 데이터를 생성하기 위해서 시추군집데이터 중 시추공 X좌표 값과 시추공 Y 좌표 값을 사용한다. 포인트 데이터는 시추 데이터의 각 포인트를 X, Y 좌표로 설정하고, Z 좌표는 시추공 고도 컬럼을 기준으로 구간 시작 깊이를 설정한다. 이때 각 지층의 끝과 바로 아래 지층의 시작은 동일 지점이므로 그림 4와 같이 모든 지층은 시작점의 값을 포인트로 추가하고, 각 지층의 끝점은 포인트로 추가하지 않는다. 그러나 마지막 지층은 아래 지층이 없기 때문에 끝 지점을 추가하여 포인트 데이터를 생성한다.

이와 같이 포인트 데이터 생성은 그림 5에서 insert_point 함수로 정의하고 파이썬의 Pandas

라이브러리를 사용하여 DataFrame() 함수를 사용하여 X, Y, Z 좌표를 설정하여 포인트 데이터 생성을 구현한다.

2. 포인트 데이터 보간

시추 군집 데이터로부터 생성한 포인트 데이터만으로 지층 표면을 생성할 경우에는 실제 지층 형태와는 전혀 다른 모델이 생성되는 문제가 있기 때문에 데이터 보간(Data Interpolation)이 필요하다.

데이터 보간 방법에는 IDW, Kriging, Natural Neighbor, Spline 등 다양한 방법이 있으나 지반정보로부터 3차원 가시화 모델을 만들 때 포인트 데이터에 대한 데이터 보간은 최대한 자연스러운 보간을 진행하기 위해서 일반적으로 많이 사용하는 크리깅(Kriging)을 사용한다(Isaaks and Srivastava, 1989; Kim *et al.*, 2008; Lee. *et al.*, 2016). 보간의 정밀도는 3차원 지층모델의 군집데이터 생성 평가에서 각 레벨별로 정밀도 분석을 수행하였을 때 Lv6정도의 군집거리가 정밀도 면에서 가장 합리적으로 판단된다고 평

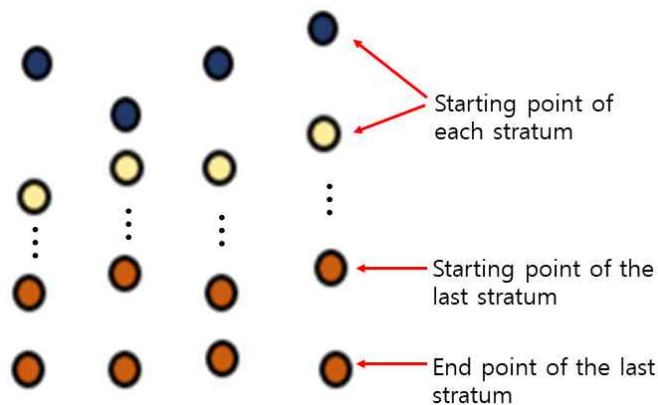


FIGURE 4. Point data generation

```

def insert_point(h_f, df, f):
    df2 = pd.DataFrame()
    if h_f == '보통암층' or h_f == '연암층':
        df2['X']=LOCATION_X['stratumname'==h_f]
        df2['Y']=LOCATION_Y['stratumname'==h_f]
        df2['Z']=borehole_height['stratumname'==h_f]-sec_s_depth['stratumname' == h_f]
        df2['f'] = "SOFT(MODERATE)"
    else:
        df2['X']=LOCATION_X['stratumname'==h_f]
        df2['Y']=LOCATION_Y['stratumname'==h_f]
        df2['Z']=borehole_height['stratumname'==h_f]-sec_s_depth['stratumname'==h_f]
        df2['f']=f
    if len(df2)>1:
        df=pd.concat([df,df2],ignore_index = True)
        logger.info(h_f+' point generation')
    return df

```

FIGURE 5. Core codes for point data generation

```

gp.set_interpolator(geo_model, compile_theano=True, theano_optimizer='fast_compile')
gp.get_data(geo_model, 'kriging')
sol = gp.compute_model(geo_model)

```

FIGURE 6. Core codes for Kriging Interpolation

가된다(Lee. *et al.*, 2017). 따라서 정밀도 분석에 따라 지점 간 거리는 최대 300m로 적용하며 본 논문에서는 최대한 자연스러운 보간을 하기 위해서 결정된 베리오그램(Variogram)을 이용한 베이시안 크리깅(Bayesian Kriging)을 사용하여 적용하며 구현은 그림 6과 같이 파이썬 Gempy 라이브러리를 사용한다.

Gempy는 3D 구조 지질 모델링 라이브러리로 크리깅을 라이브러리로 지원하고 있으며 보간 결과에 대한 정확도는 높다. 그러나 3차원 크리깅의 경우 데이터의 양이 많아지면 그리드의 증가로 속도 지연 뿐 아니라, 메모리 사용량이 초과되어 전체 데이터를 생성하는데 오류가 발생할 수 있다. 따라서 본 연구에서는 생성될 면의 모든 점에 대한 크리깅을 진행하지 않고 입력 데이터의 MBR(minimum bounding rectangle)을 기준으로 가로, 세로, 높이의 그리드를 생성하고 각 그리드 점들만 보간을 진행한다.

3. 지층별 3차원 표면 생성

각 지층의 3차원 표면 생성은 그림 7(a)와 같이 포인트 데이터와 보간된 데이터로부터 그림 7(b)와 같이 삼각망 구조의 메쉬(Mesh)를 구성하고, 그림 7(c)와 같이 메쉬 구조의 3차원 표면을 생성한다.

이에 대한 구현 코드는 그림 8과 같이 create_model('geo_model') 함수로 모델을 생성하고 get_surfaces(geo_data) 함수를 사용하여 표면을 생성한다.

그림 8의 프로그램으로부터 삼각망으로 구성이 된 한 지층에 대한 3차원 표면 이미지는 그림 9와 같이 생성되며 기본적으로 옆면과 밑면이 없이 표면만 생성된다.

4. 지층 Block 모델과 병합

일반적으로 생성된 각 지층은 그림 9와 같이 표면만 존재하고 옆면과 밑면은 존재하지 않는

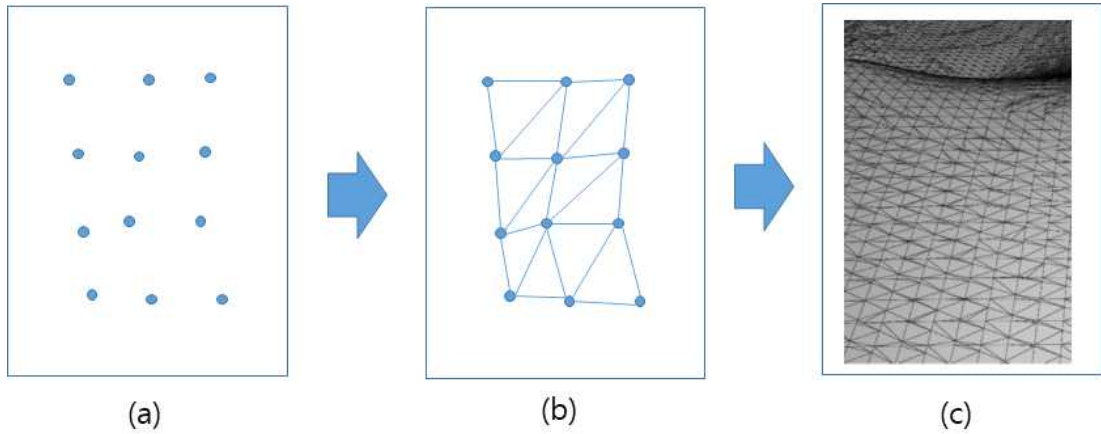


FIGURE 7. Triangulation to generation 3D surface

```
def model_create(ori, poi):
    geo_model = gp.create_model('geo_model')
    gp.init_data(geo_model, [start_x, end_x, start_y, end_y, start_z, end_z],
    [grid],path_iddefault_values=True)
    return geo_model

def export_to_vtk(geo_data, path=None, name=None, voxels=False, block=None, surfaces=True):
    if surfaces is True:
        geo_data.solutions.compute_all_surfaces()
        ver, sim = gp.get_surfaces(geo_data)
        GemPyvtkInteract.export_vtk_surfaces(geo_data, ver, sim, path=path, name=name)
    return True
```

FIGURE 8. Core codes for 3D surface generation



FIGURE 9. 3D surface of a stratum

다. 따라서 여러 지층을 하나의 3차원 이미지로 병합하기 위해서는 생성된 표면의 모서리 점들을 연결하여 네 면의 옆면을 생성하고, 아랫면

은 아래 지층 표면 이미지를 사용하여 생성함으로써 윗면, 옆면, 아랫면 6개의 면을 그림 10과 같이 하나의 구조로 Block 모델을 만든다.



FIGURE 10. A stratum block model that combines the top, side, and bottom sides of FIGURE 9

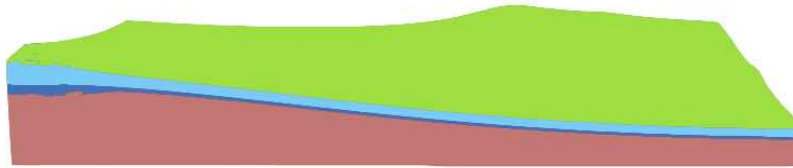


FIGURE 11. Merge of 3D block models for each stratum



FIGURE 12. 3D stratum generated by borehole locations

그림 10과 같이 생성된 각 지층의 Block 모델들을 순차적으로 병합하면 그림 11과 같이 하나의 3차원 모형의 이미지를 생성할 수 있다.

그림 11과 같이 만들어진 3차원 이미지는 각 지층별로 크리깅을 할 때 설정한 MBR을 기준으로 생성하기 때문에 위에서 보았을 때 시추공 위치에 따른 모양이 아닌 직사각형 모양의 모델이 생성된다. 따라서 직사각형 모델로 생성된 3차원 이미지를 시추공 위치 중심으로 둥근 형태의 이미지로 가공할 필요가 있다. 시추공 위치 중심으로 둥근 형태의 이미지를 생성하기 위해서는 시추공 위치 중심으로 버퍼(Buffer)를 두고 버퍼로부터 생성된 점들을 convex_hull 알고리즘을 통해 하나의 다각형(polygon)으로 만

들어준다. 그리고 다각형의 모양대로 시추 군집 모형으로 자르면 그림 12와 같이 둥근 형태의 3D 지층 이미지를 생성할 수 있다.

그림 13은 사각 모양의 3차원 이미지를 시추공의 위치에 따라 둥근 모양의 3차원 지층 이미지로 가공하기 위해 파이썬의 geoPandas 라이브러리를 사용하여 buffer와 convex_hull 알고리즘을 구현한 주요 코드이다.

표면 말림현상

실제 지층은 용기와 침하로 역전된 지층이 존재할 수 있다. 이러한 지층의 정보가 입력된 포인트 데이터를 앞에서 제안한 방법으로 보간을


```

def makeShell(a_face):
    point = pd.read_csv('./points.csv')
    point = gpd.GeoDataFrame(point, geometry = gpd.points_from_xy(point['X'], point['Y']))
    point = point.unary_union.convex_hull.buffer(15)
    poiList = list(point.exterior.coords)
    originShellDf =
        pd.DataFrame(poiList,columns = ['X','Y'])
    shellDf_1 = originShellDf.copy()
    shellDf_1['Z'] = 1000
    shellDf_2 = originShellDf.copy()
    shellDf_2['Z'] = -1000
    shellDf_all=
    pd.concat([shellDf_1,shellDf_2],ignore_index = True)
    np_array_shell = shellDf_all.to_numpy()
    shellSurf = pv.PolyData(np_array_shell)
    shell = shellSurf.delaunay_3차원(alpha=0)

    if a_face:
        plot = pv.Plotter()
        plot.add_mesh(shell, color = color[face_name.index(a_face)])
    plot.export_obj(resource_path(savepath+"./result/obj/"+a_face+"shell"))
    return shell

```

FIGURE 13. Core codes to generate 3D stratum based on borehole location

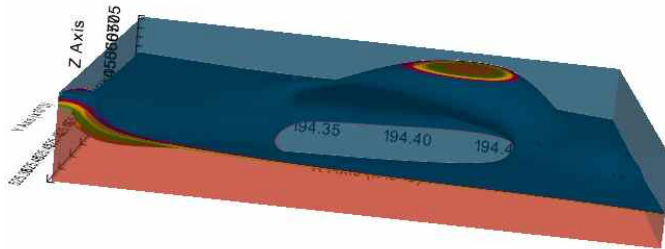


FIGURE 14. Roll-up of the stratum surface

통해 삼각망을 구성하여 각 지층의 표면을 생성하면 그림 14와 같이 표면의 말림현상(roll-up)이 나타난다. 표면의 말림현상은 역전된 지층 데이터가 전달되거나 입력된 포인트 데이터의 거리가 일정 이상 멀어지는 경우 발생한다.

본 연구에서는 그림 14와 같은 말림현상을 해결하기 위해 각 지층 표면의 방향을 나타내는 방향 벡터(Orientation)를 추가하였다. 방향 벡터는 지층면의 한 평면으로부터 이루는 각도가

90도인 직선의 진행 방향을 나타내는 벡터를 말하며 방향 벡터의 X, Y, Z 좌표와 바라보는 각도 설정을 위해 방위각(Azimuth)과 극성(Polarity)을 설정할 수 있으며, 극성을 기본 값으로 방위각은 하늘을 향하게 설정해서 강제로 모든 지층 표면이 하늘을 향하도록 하였다. 그 결과 그림 15와 같이 말림현상이 없는 지층 표면의 형태를 만들 수 있다. 이에 대한 구현 코드는 그림 16과 같다.

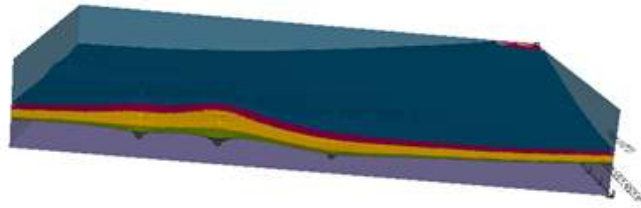


FIGURE 15. Stratum surface with no roll-up effect by adding orientation

```
def insert_orientation(h_f, df, f):
    df2 = pd.DataFrame()
    if h_f=='보통암층' or h_f=='연암층':
        df2['X']=LOCATION_X['stratumname'==h_f]
        df2['Y']=LOCATION_Y['stratumname'==h_f]
        df2['Z']=borehole_height['stratumname'==h_f]-sec_s_depth['stratumname'==h_f]
        df2['azimuth'] = 90
        df2['dip'] = 0
        df2['polarity'] = 1
        df2['f'] = "SOFT(MODERATE)"
    else:
        df2['X']=LOCATION_X['stratumname'==h_f]
        df2['Y']=LOCATION_Y['stratumname'==h_f]
        df2['Z']=borehole_height['stratumname'== h_f]-sec_s_depth['stratumname'==h_f]
        df2['azimuth'] = 90
        df2['dip'] = 0
        df2['polarity'] = 1
        df2['f'] = f
    if len(df2)>1:
        df=pd.concat([df,df2],ignore_index=True)
        logger.info(h_f+' orientation generation')
    return df
```

FIGURE 16. Core codes to eliminate surface roll-up

표면 방향 벡터의 입력이 많으면 많을수록 그림 17과 동일하게 접힌 지층(Fold stratum)과 같은 특수한 형태의 지층표현은 물론 지하 동굴 까지도 표현이 가능하다.

하지만 시추공을 이용하여 수집하는 시추데이터의 경우 한 지점에 관한 정보만 확인이 가능하기 때문에 표면의 방향 정보는 알 수 없다. 따라서 표면 방향 벡터를 추가하더라도 간혹 부자연스러운 지층 표면이 생성될 수 있다. 이는

단층의 존재 때문이며 단층의 위치와 각도를 입력한다면 단층 또한, 표현이 가능하다. 그림 18은 단층이 있다는 가정 하에 단층의 위치와 각도를 임의로 입력하여 생성한 결과로 단층이 잘 표현되고 있다.

MVS와 Borehole2Model의 비교

본 논문에서 제안한 Borehole2Model의 완성

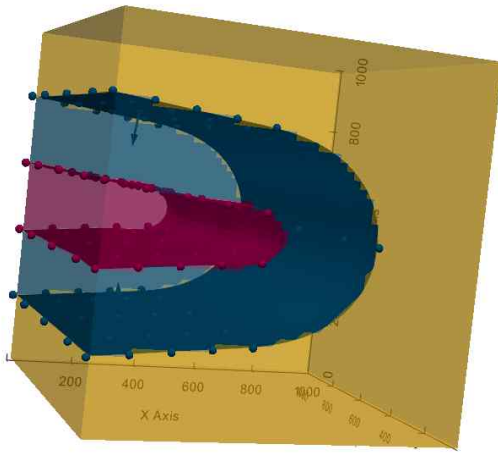


FIGURE 17. Fold stratum represented by addition of orientation

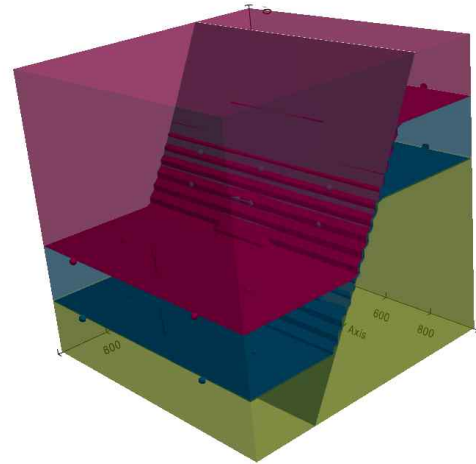


FIGURE 18. Stratum model with faults

도를 알아보기 위해서 같은 지역에 대한 시추 정보로부터 MVS 모델과 Borehole2Model로 3차원 모형 이미지를 생성하여 비교하였다. 일반적으로 시추는 균등한 거리에서 이루어지지 않고, 지역에 따라 시추정보가 존재하지 않는 지

역도 존재한다. 시추정보가 일정 거리 이상 존재하지 않는다면 생성되는 지층모델의 정확도를 확인할 수 없기 때문에 실험에 사용한 시추데이터는 그림 19와 같이 최대한 시추정보가 밀집되어 있는 지역인 고양시 일대를 선정하였다.



FIGURE 19. Borehole in Goyang City used for comparison of the two models

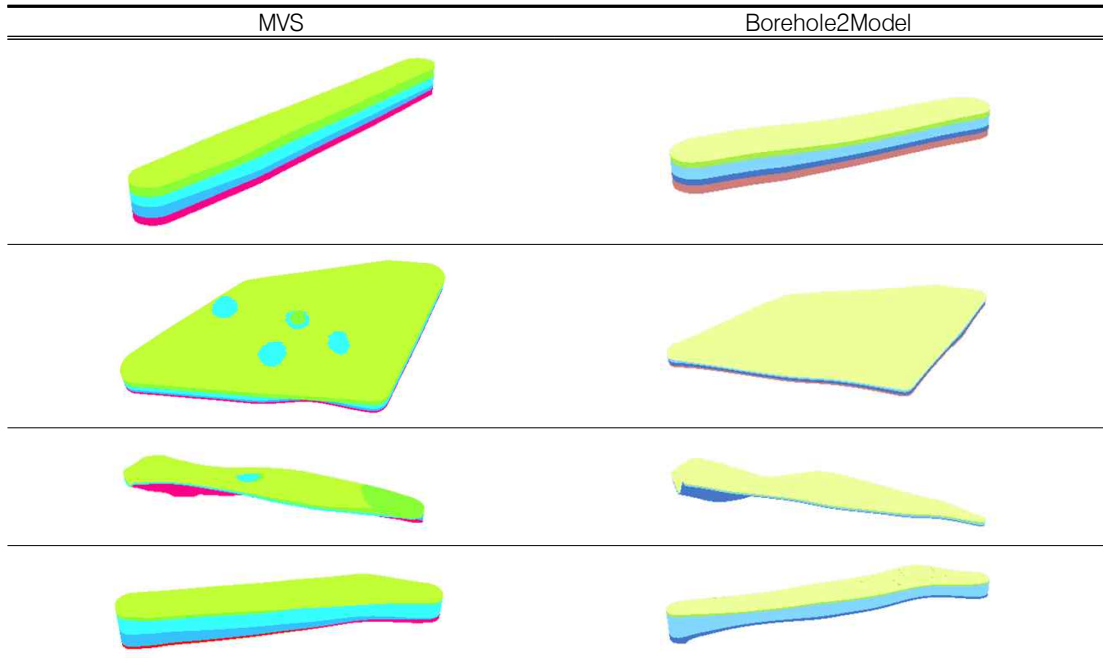


FIGURE 20. Comparison of 3D model of MVS and the proposed method, Borehole2Model

고양시 일대의 시추정보를 대상으로 각 위치별로 군집화하여 지층모델 생성 원시 데이터로 사용하여 두 모델로부터 생성된 지층 이미지를 비교하였다.

그림 20은 그림 19의 고양 지역에 대한 시추공 데이터를 입력으로 MVS 모델과 Borehole2Model로부터 생성된 3차원 모형 이미지 결과를 보여주고 있다. MVS와 Borehole2Model을 비교하였을 때 지층의 깊이와 지층의 굴곡이 일부분에서 약간의 차이는 있었으나 문제라고 인식할 정도로 다른 부분은 보이지 않았으며 전체적으로 거의 동일한 3차원 모형의 이미지로 구현되었다. 두 모델로부터 생성된 3차원 이미지가 100% 일치하지 않는 것은 실제 시추공 데이터가 조밀하지 않기 때문에 두 모델 모두 실제 지층과 비교했을 때 오차가 발생할 것이며 실제로 땅을 파서 지층을 확인하기 전에는 어느 3차원 모형의 결과 이미지가 정확하다고 단정할 수는 없다.

결론

컴퓨터 기술의 발달로 지반정보를 활용하여 3차원 가시화 프로그램을 개발하는 경우가 많고 있다. 현재 지반정보를 활용하여 3차원 가시화 프로그램을 개발할 경우에는 MVS 외산 프로그램을 사용하고 있으나 이 프로그램은 유료 프로그램이고 국내에서 관련 프로그램을 개발할 때 MVS의 라이브러리를 이해하고 사용하는데 어려움이 있었다. 본 논문에서는 이런 불편함을 해결하기 위해서 파이썬의 오픈 라이브러리 Gempy를 사용하여 3차원 지층 이미지를 생성할 수 있는 3차원 가시화 프로그램 Borehole2Model을 제안하였다. Borehole2Model은 시추데이터로부터 시추군집데이터를 구성하고 이를 입력 값으로 각 지층별 포인트 데이터를 생성하고 정확한 지층 표면을 생성하기 위해서 생성한 포인트 데이터를 크리깅 보간 기법으로 보간을 진행하여 지층별 표면을 생성하였다. 그리고 생성한 각각의 지층별 모델에 옆면과 밑면을 추가하고

각 지층을 병합하여 3차원 지층 이미지를 생성하였다. 또한 일부 구간에서 표면 말뚝현상이 발생하여 방향 벡터를 추가하여 이를 해결하였다. 본 논문에서는 Borehole2Model의 완성도를 평가하기 위해서 같은 지역에 대해서 MVS와 Borehole2Model로 부터 3차원 이미지를 생성하여 비교한 결과 지층의 깊이와 지층의 굴곡이 일부분에서 약간의 차이는 있었지만 전체적으로 큰 차이가 없이 유사한 모형을 유지하였다.

향후 다양한 시추공 데이터를 가지고 정확도에 대한 분석과 비교가 필요하다. 또한 단층 정보는 단층의 위치정보만 존재하기 때문에 단층이 이루는 각도는 알 수 없기 때문에 해당 위치 정보만으로는 지층과 수직인 단층만을 생성하여 부자연스러운 지층 모델이 생성될 수 있기 때문에 단층에 대한 추가적인 연구가 필요하다. **KAGIS**

REFERENCES

- GemPy. 2021. <https://docs.gempy.org/>. (Accessed September 23, 2022)
- Kim, J.M. and S.H. Han. 2019. A Proposal for the Development and Application of Underground Space Integration Map Using 3D Spatial Information Data. Proceedings of Korean Society for Geospatial Information Science. pp.89-92 (김재명, 한승희. 2019. 3차원 공간정보데이터를 이용한 지하공간통합지도 구축 및 활용 방안. 대한공간정보학회 학술대회 초록집. 89-92쪽).
- Kim, J.S., H.W. Rim, M.J. Um, W.I. Kim and W.S. Ahn. 2008. Spatial Adjustment of Rainfall using Kriging Method and Application of Distributed Model. Proceedings of the Korea Water Resources Association Conference. pp.130-134 (김진성, 임해욱, 엄명진, 김원일, 안원식. 2008. 크리깅 기법을 이용한 강우의 공간보정과 분포형 모형의 적용. 한국수자원학회 2008년도 학술발표회 초록집. 130-134쪽).
- Kim, Y.T. and H.J. Kouh. 2022. Design of Caching Scheme for Mobile Underground Geospatial Information Map System. Journal of Convergence for Information Technology 12(1):7-14 (김용태, 고훈준. 2022. 모바일용 지하공간정보지도 관리 시스템에서 응답속도 향상을 위한 캐싱 기법. 융합정보논문지 12(1):7-14).
- Isaaks, E.H. and R.M. Srivastava. 1989. An introductions to applied geostatistics. Oxford University Press. Oxford. pp.225-235.
- Lee, B.J. 2021. Operation Technique of Spatial Data Change Recognition Data per File. Journal of the Korean Association of Geographic Information Studies 24(4): 184-193 (이봉준. 2021. 파일 단위 공간데이터 변경 인식 데이터 운영 기법. 한국지리정보학회지 24(4):184-193).
- Lee, B.J. and H.J. Kouh. 2022. Spatial Data Update Method for Efficient Operation of the Integrated Underground Geospatial Map Management System. Journal of Industrial Convergence 20(6):57-64 (이봉준, 고훈준. 2022. 지하공간통합지도 관리 시스템의 효율적인 운영을 위한 공간 데이터 갱신 방법. 산업융합연구 20(6):57-64).
- Lee, B.Y., B.S. Hwang, H.S. Kim and W.J. Cho. 2016. Comparative Evaluation on Geotechnical Information 3차원 Visualization Program for Dredging Quantity Estimation. Journal of the Korean Geo-Environmental Society 17(7):35-42 (이보영, 황범식, 김한샘, 조완제. 2016. 준설 물량 산출을 위한 지반정보 3차원 가시화 프로그램 비교 평가. 한국지반환경공학회 17(7):35-42).
- Lee, B.Y., B.S. Hwang, H.S. Kim and W.J. Cho. 2017. Precision Analysis of Geotechnical

- Information for the 3차원 Integrated Underground Space Map. Journal of the Korean Geoenvironmental Society 18(4): 5-12 (이보영, 황범식, 김한샘, 조완제. 2017. 3차원 지하공간통합지도 구축을 위한 지반정보 정밀도 분석. 한국지반환경공학회는 논문집 18(4):5-12).
- Li, S., S. Dragicevic, F.A. Castroc, M. Sester, S. Wintere, A. Coltekin, C. Pettit, B. Jiang, J. Haworth, A. Stein, and T. Cheng. 2016. Geospatial big data handling theory and methods: A review and research challenges. Journal of Photogrammetry and Remote Sensing 115:119-133.
- MOLIT. 2020. Development of technology for renewal automation and field utilization on the integrated underground geospatial information map. R&D plan. pp.1-30 (국토교통부. 2020. 지하공간통합정보지도의 갱신 자동화 및 현장 활용 기술 개발. R&D plan. 1-30쪽).
- The Mining Visualization System(MVS). 2015. <https://www.ctech.com/products/mvsevs-product-suite/mvs/> (Accessed August 09, 2022)
- Park, D.H., Y.G. Jang, J.Y. Woo and W.K. Lee. 2015. Creation plan of 3D geotechnical information for constructing underground spatial information map. Korean Society of Civil Engineers. pp.103-104 (박동현, 장용구, 우계윤, 이원국. 2015. 지하공간통합지도 구축을 위한 3차원 지반정보 생성 방안 수립. 대한토목학회 학술대회 초록집. 103-104쪽).
- Park, D.H., Y.G. Jang and H.S. Choi. 2018. A Study on the Construction Plan of 3차원 Geotechnical Information for the Support of Underground Space Safety. Journal of the Korean Association of Geographic Information Studies 21(1):23-34 (박동현, 장용구, 최현상. 2018. 지하안전관리 지원을 위한 3차원 지반정보 구축 방안 연구. 한국지리정보학회지 21(1):23-34).
- Seok, C.H., Y.R. Kang, Y.G. Jang and I.J. Kang. 2012. A Study of Standardization for Establishing Integrated Infrastructure of Underground Space Information and its Availability. Journal of the Korean Society for Geospatial Information System 20(4):105-115 (석철호, 강유라, 장용구, 강인준. 2012. 지하공간정보통합 인프라 구축을 위한 표준화방안 및 활용성에 관한 연구. 한국지형공간정보학회지 20(4):105-115).