

The background of the slide is a dark gray map of Seoul, South Korea, showing the intricate network of streets and the Han River. The map is rendered in a light gray line-art style against the dark background.

Automating Architecture Services Computation workshop

SIDE PROJECT

Soyoon Kwon

Brainstorming

Automating Architecture

건축자동화

save time

시간 확보

Accuracy

정확성

Rapid Simulation

빠른 시뮬레이션

Data-Driven Decision-Making

데이터 기반 의사결정

Rapid Analysis

빠른 분석

Why?

왜 하는데?

Consistency

일관성

Flexibility

유연성

Improved Efficiency

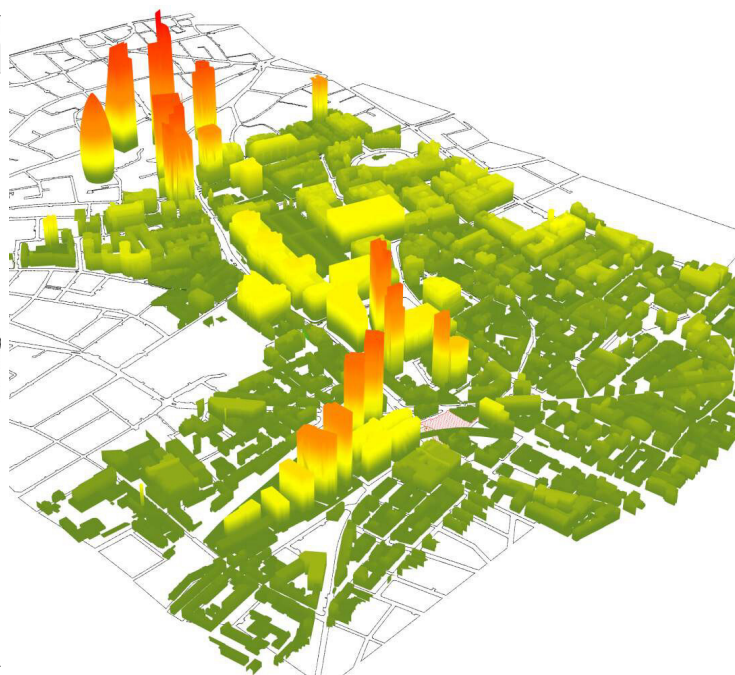
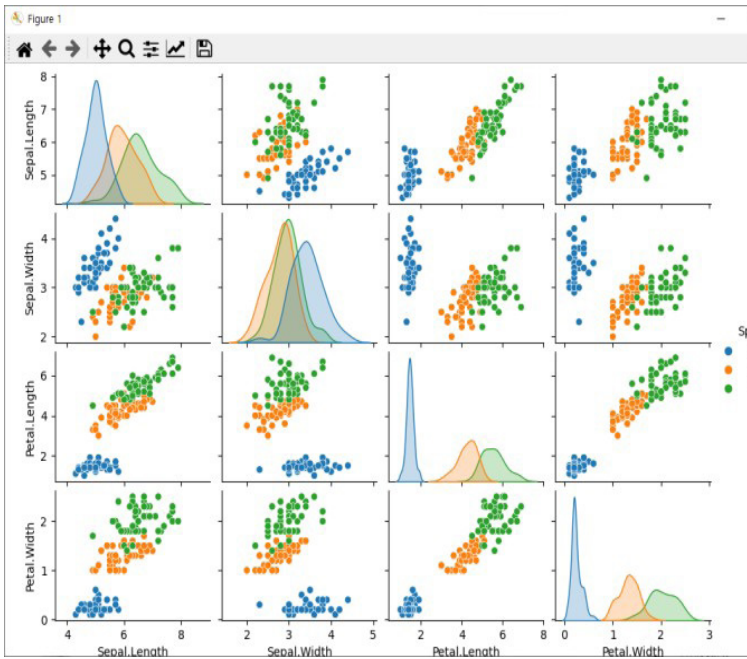
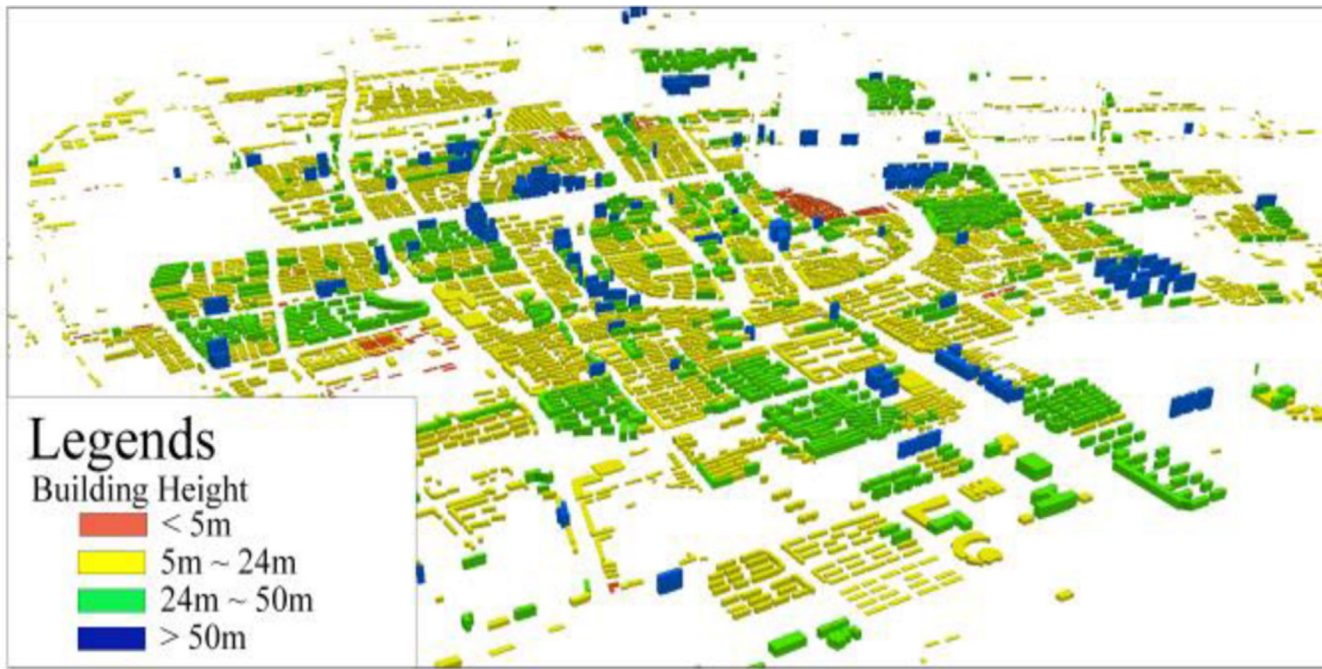
효율성 향상

Data Transformation

데이터 전환

SIDE PROJECT

IDEA



Analysis

Building Height measurement

Node

Site Planning

information

Path

Edge

Base massing

magnitude

Landmark

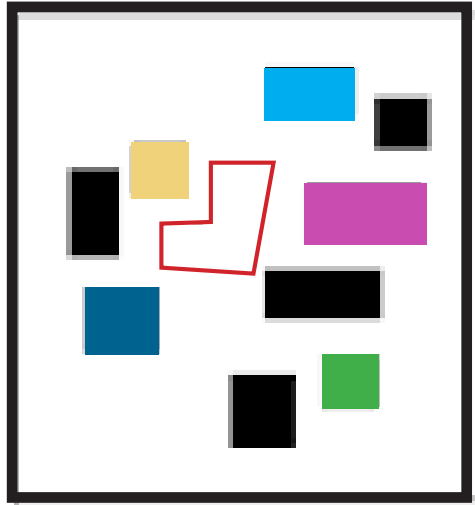
District

Layout

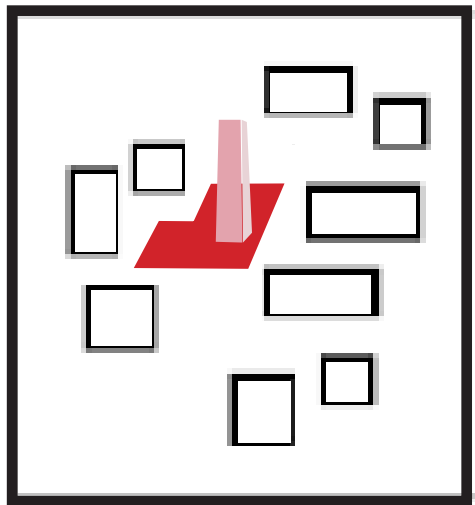
Mapping

SIDE PROJECT

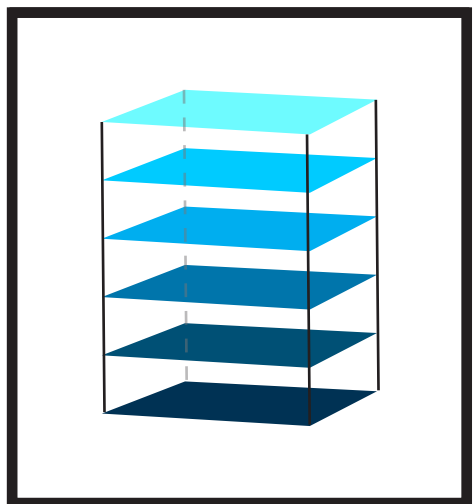
Point of view



OPT1 Automate Site Analysis
주변 데이터를 시각화 한다.
분석 자동화



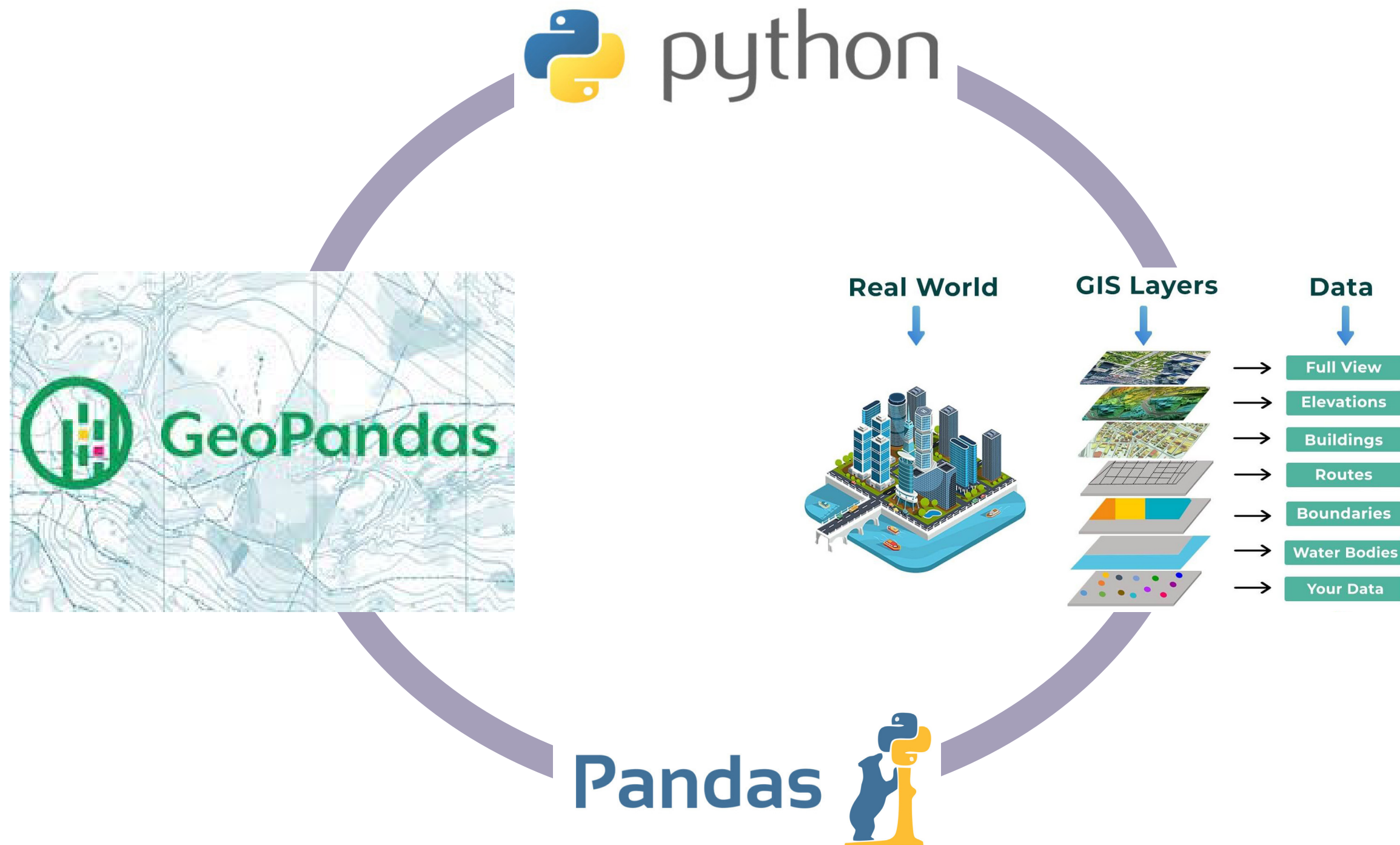
OPT2 Automate building height
내 건물 높이를 산정한다.
높이(배치) 자동화



OPT3 Automate Basemassing build
기준층 평면에 맞춰서 베이스매싱을 만든다.
베이스매싱 자동화

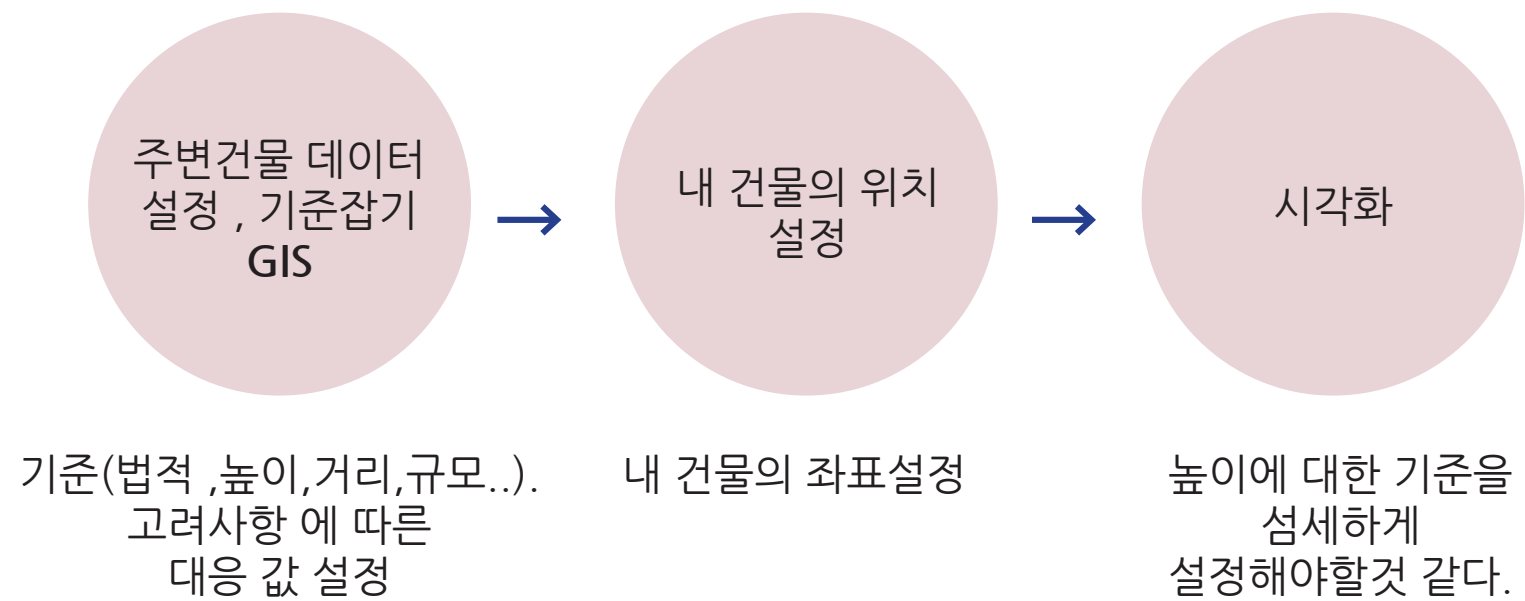
SIDE PROJECT

Approach



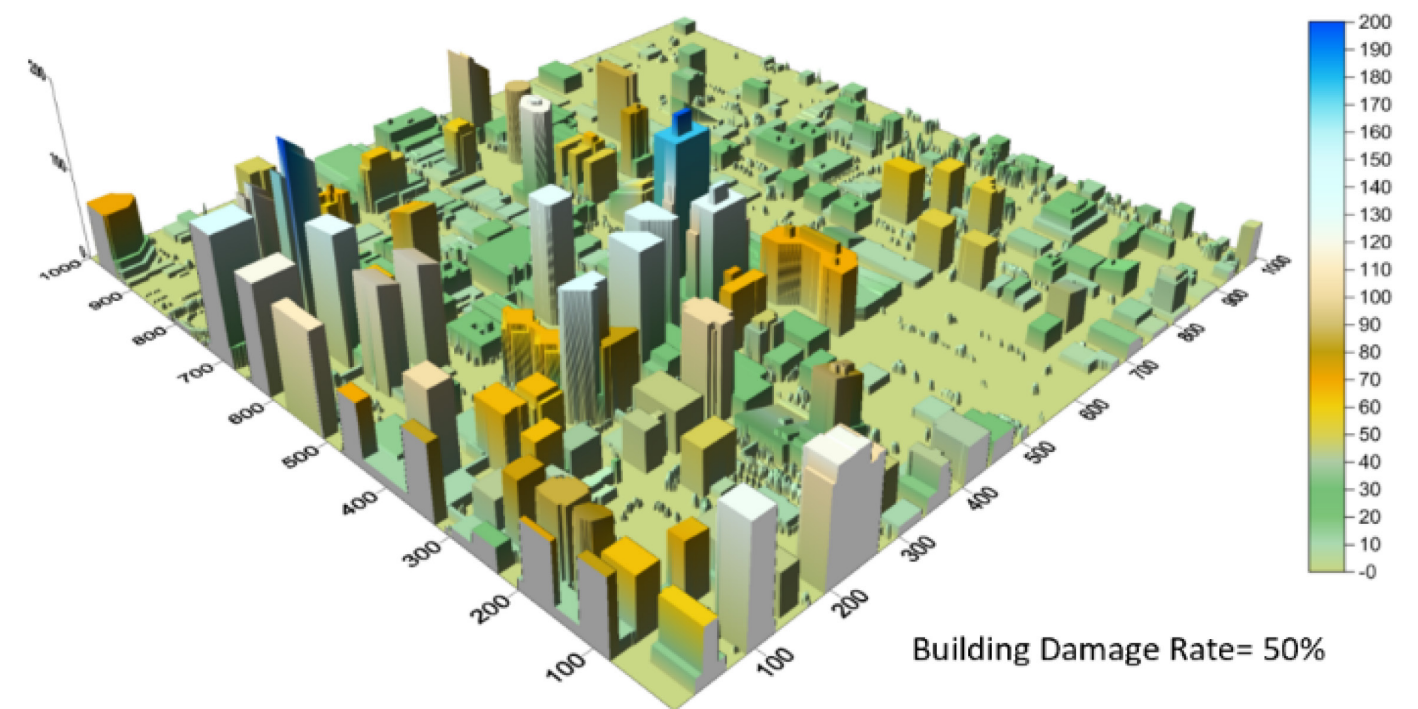
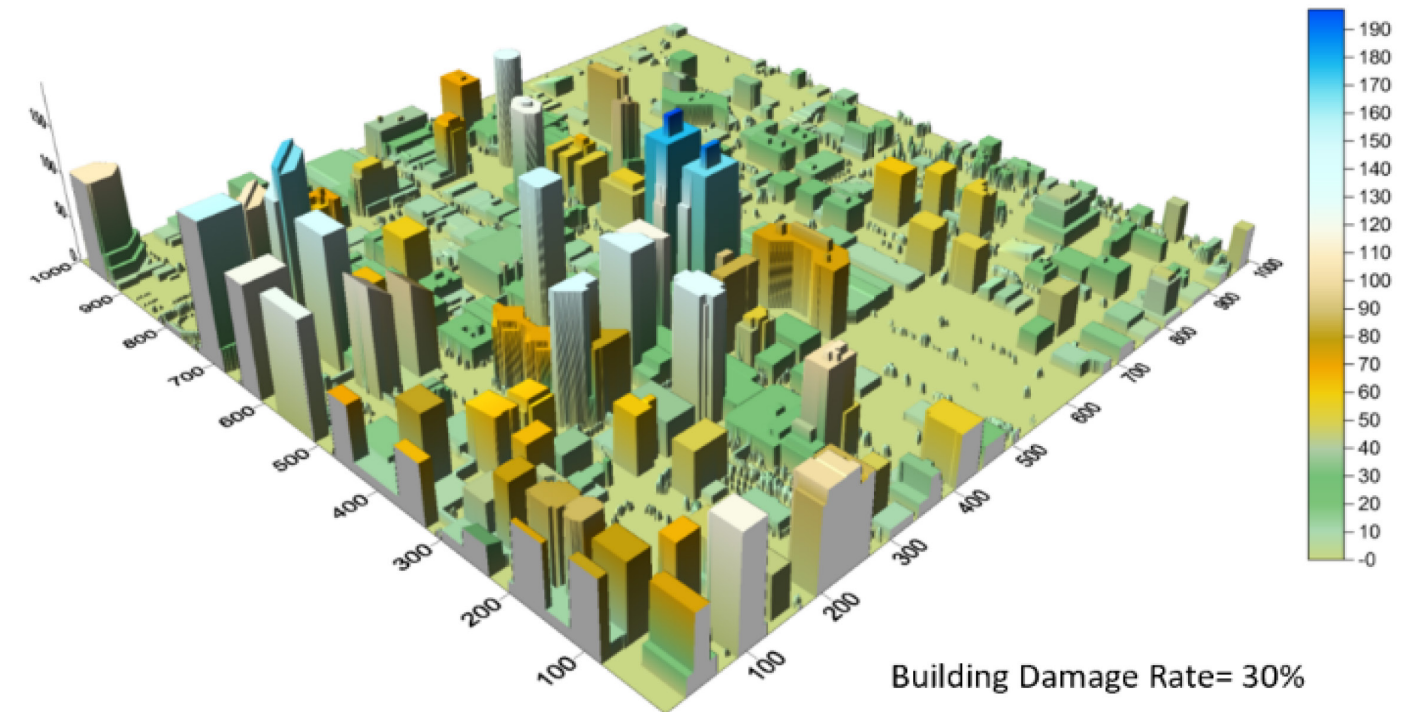
OPT1_Automate building height

Processing IDEA



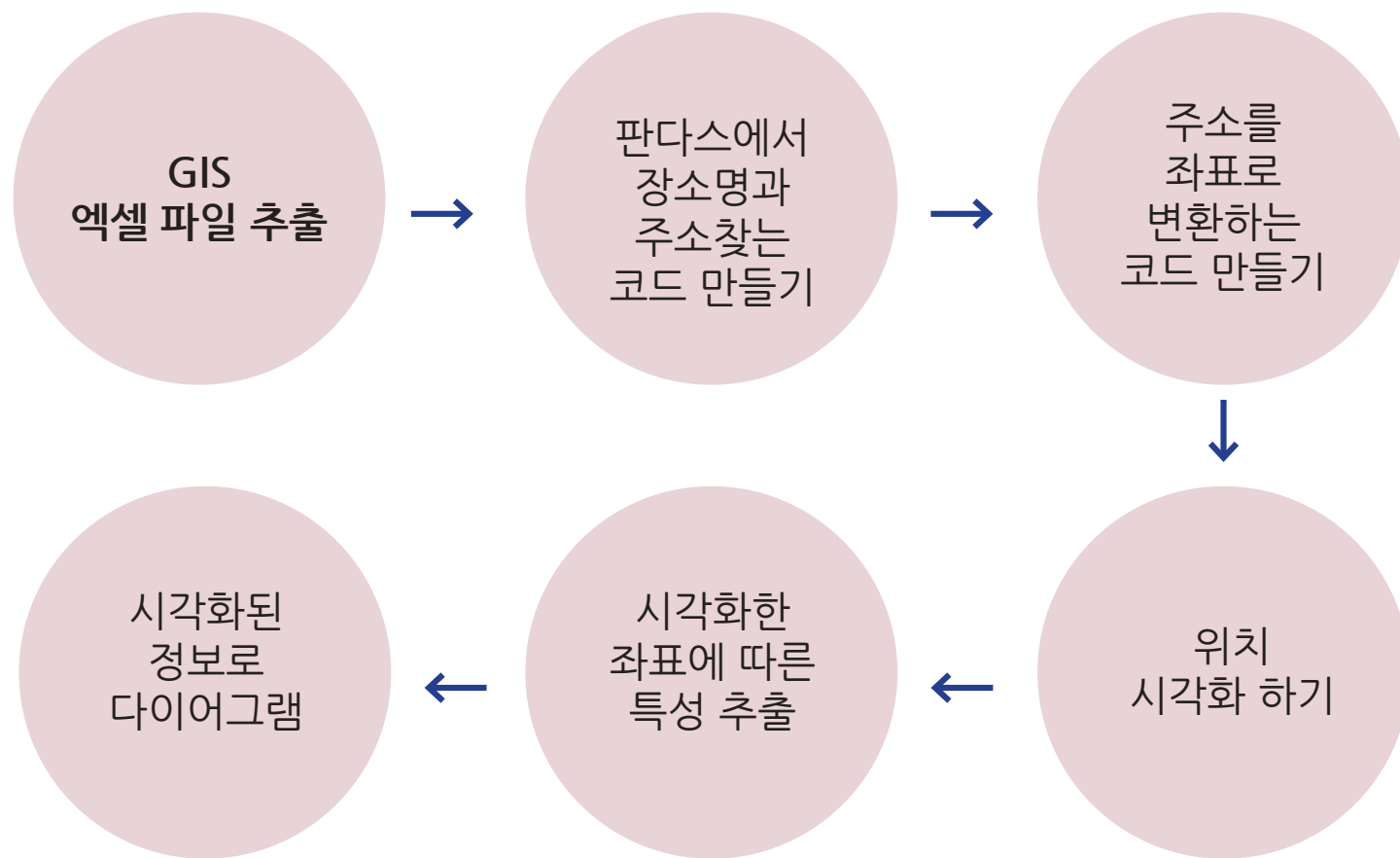
Strength

디자인 전 단계를 빠르고 정확하게
여러가지 고려사항을 한번에 반영
바뀌는 값들에 대한 유연성
디자인하는 시간에 더 투자



OPT2_Automate Site Analysis

Processing IDEA



Strength

GIS에 있는 자료, 높이, 연도, 재료 등 각종 특성을
한번에 시각화 가능
엑셀로된 자료의 형태를 시각화.
분석 다이어그램 순식간.

side.py > ...

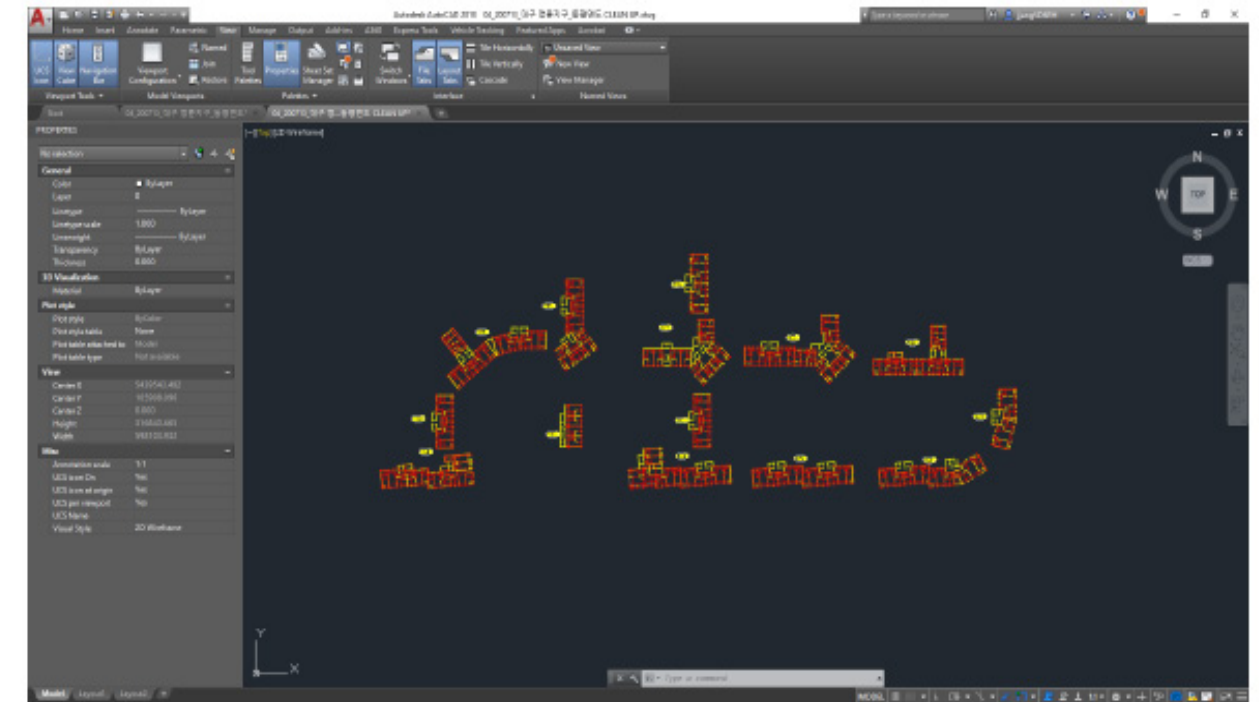
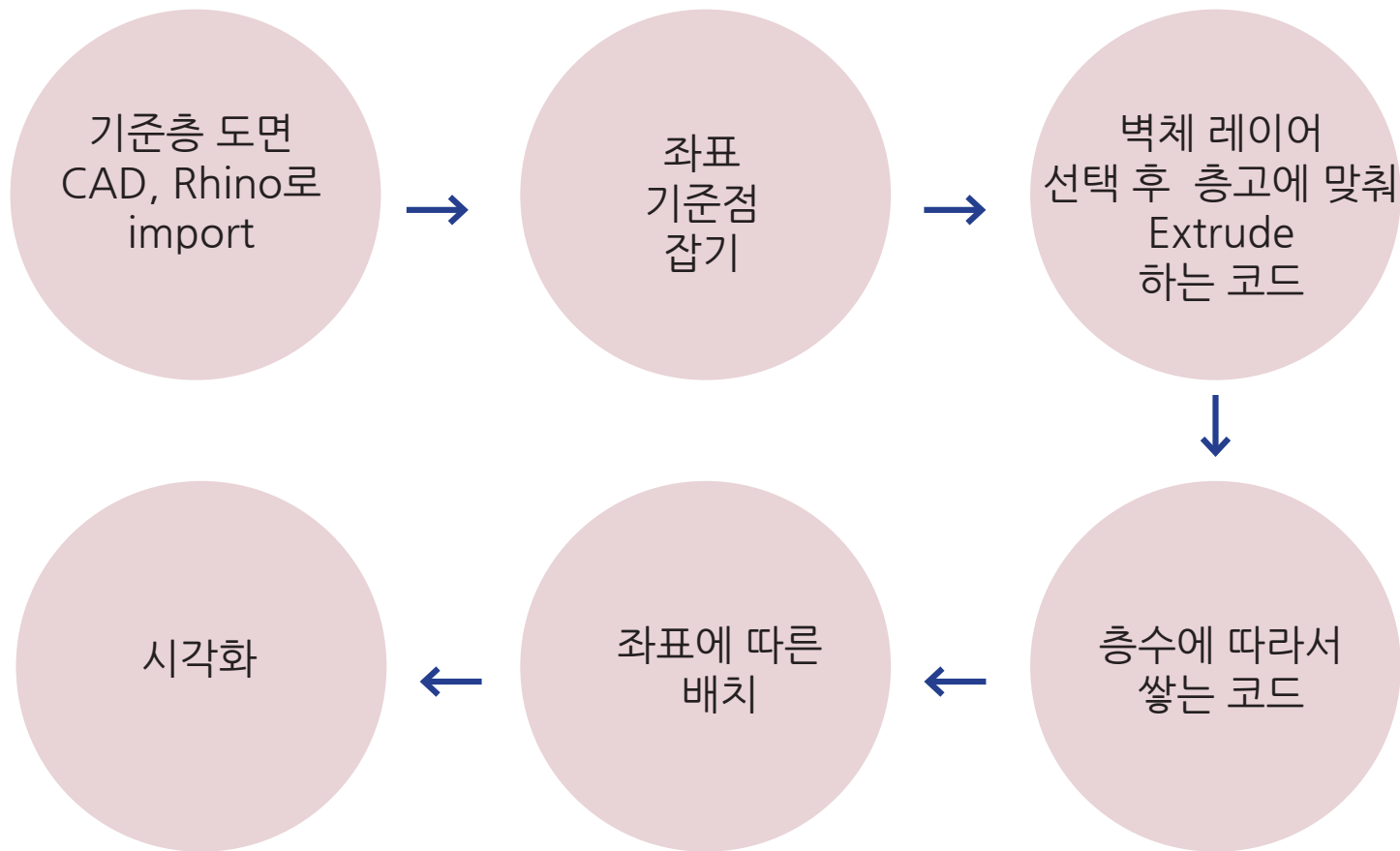
```
1  import geopandas as gpd
2  from shapely.geometry import Point
3  import matplotlib.pyplot as plt
4  import math
5
6  # 주변 건물 데이터 (가상의 데이터)
7  building_data = {'geometry': [Point(0, 0), Point(1, 1), Point(2, 0)],
8                  'height': [20, 30, 25]}
9  buildings = gpd.GeoDataFrame(building_data, crs='EPSG:4326')
10
11 # 나의 건물 위치 (가상의 위치)
12 my_building_location = Point(1.5, 0)
13
14 # 시각화: 주변 건물과 나의 건물 위치
15 ax = buildings.plot(column='height', cmap='OrRd', legend=True)
16 gpd.GeoSeries([my_building_location]).plot(ax=ax, color='blue', markersize=50, label='My Building')
17 plt.legend()
18 plt.show()
19
20 # 나의 건물 높이 계산
21 def calculate_building_height(buildings, my_location):
22     distances = buildings.distance(my_location)
23     angles = buildings.geometry.angle(my_location)
24
25     total_weighted_height = 0
26     total_weight = 0
27
28     for distance, angle, height in zip(distances, angles, buildings['height']):
29         weight = math.cos(angle) / distance
30         total_weighted_height += weight * height
31         total_weight += weight
32
33     if total_weight != 0:
34         my_building_height = total_weighted_height / total_weight
35     else:
36         my_building_height = 0
37
38     return my_building_height
39
40 my_building_height = calculate_building_height(buildings, my_building_location)
41 print(f"나의 건물의 높이는: {my_building_height:.2f} 미터")
42
```


SIDE PROJECT

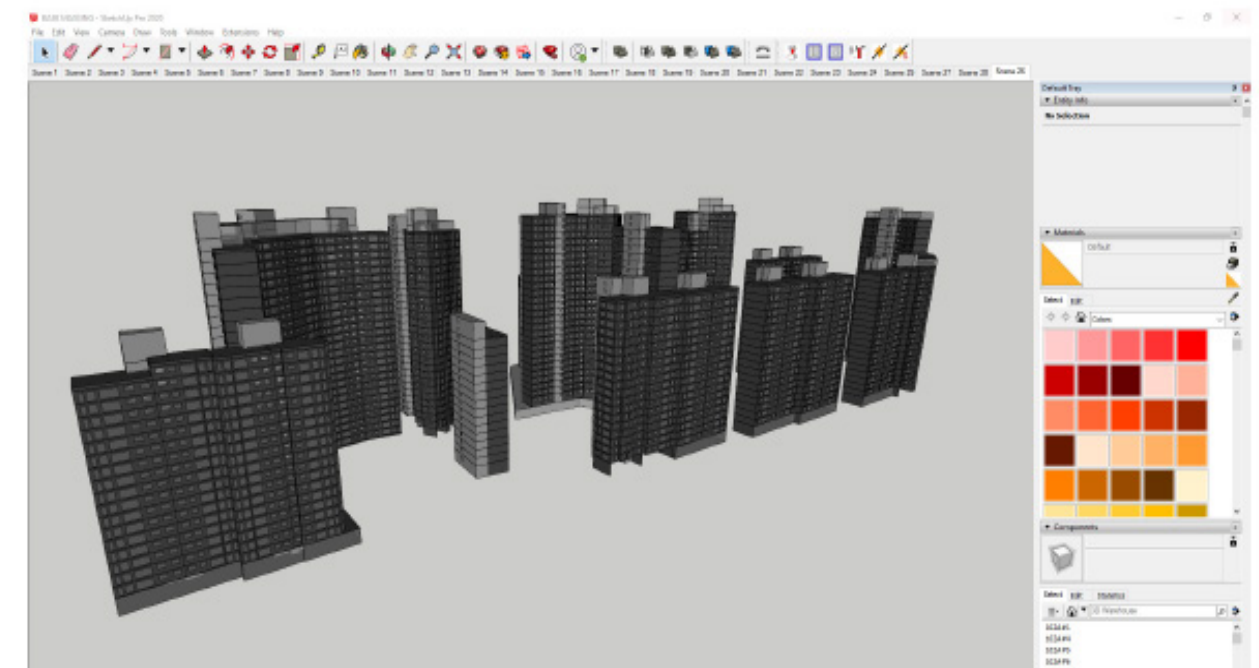
OPT3_Automate Base Massing Build

1. Plan Clean-up

Processing IDEA



2. Build Base Massing



Strength

귀찮은 베이스 매싱만들기를 쉽고 간단하게!
도면이 수정될때마다 새로 만들 필요 없이 도면만 바꾸면 됨!
시간 절약, 디자인 단계에서 시간 투자 가능



