**Swisstopo 3D Building Volume and Surface Analysis Tools**

**Overview**

This toolset processes Swisstopo 3D building data (multipatch geometries) to calculate building volumes and analyze surface areas. It's designed to handle large datasets efficiently using parallel processing, providing detailed metrics for each building including volume, roof area, footprint, and wall areas.

**What It Does**

1. **Reads** Swisstopo 3D building data from GDB (geodatabase) files
2. **Repairs** mesh geometries to ensure they are watertight for accurate volume calculation
3. **Calculates** building volumes using advanced mesh repair techniques
4. **Analyzes** surface areas, classifying them as roof, footprint, or walls
5. **Outputs** comprehensive results in CSV and Excel formats

**Requirements**

* Python 3.8 or higher
* Required Python packages:
* fiona
* pandas
* numpy
* trimesh
* openpyxl

Install with: python -m pip install fiona pandas numpy trimesh openpyxl

**Files**

* main.py - Main orchestrator script
* mesh\_repair\_volume.py - Mesh repair and volume calculation module
* surface\_analysis.py - Surface area analysis module
* test\_imports.py - Utility to verify installation

**Usage**

**Basic Command Structure**

bash

python main.py <input\_gdb\_path> <output\_directory> [options]

**Parameters**

* <input\_gdb\_path> - Path to Swisstopo GDB file (required)
* <output\_directory> - Where to save results (required)
* --layer - GDB layer name (default: "Building\_solid")
* --limit - Process only first N buildings (optional, for testing)
* --workers - Number of parallel workers (default: CPU count - 1, max 8)

**Example Usage**

1. **Test run with 100 buildings:**

bash

cd "C:\ DEV\Python\SWT 3D Buildings"

python main.py "C:\Users\DavidRasner\Downloads\SWISSBUILDINGS3D\_3\_0.gdb" "C:\BBL DEV\Python\SWT 3D Buildings\output" --limit 100 --workers 8 --layer Building\_solid

1. **Process entire dataset:**

bash

python main.py "C:\DEV\Inputs\SWISSBUILDINGS3D\_3\_0.gdb" "C:\ DEV\Output" --layer Building\_solid

1. **Process with 4 workers:**

bash

python main.py "path\to\buildings.gdb" "output\folder" --workers 4

**Output Files**

**Generated Files**

* building\_analysis\_YYYYMMDD\_HHMMSS.csv - Complete results in CSV format
* building\_analysis\_YYYYMMDD\_HHMMSS.xlsx - Complete results in Excel format
* processing.log - Detailed processing log

**Output Variables**

**Input Fields (preserved from GDB)**

* OBJECTID - Original Swisstopo building ID
* UUID - Unique identifier
* OBJEKTART - Object type
* NAME\_KOMPLETT - Complete building name
* GEBAEUDE\_NUTZUNG - Building usage
* DACH\_MAX/DACH\_MIN - Roof height values
* EGID - Federal building ID
* (and all other original fields)

**Mesh Processing Fields (prefix: mesh\_)**

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| mesh\_volume | float | Building volume in cubic meters |
| mesh\_is\_watertight | bool | Whether mesh is watertight |
| mesh\_vertex\_count | int | Number of mesh vertices |
| mesh\_face\_count | int | Number of mesh faces |
| mesh\_repair\_applied | bool | Whether repair was needed |
| mesh\_repair\_steps | string | Description of repair process |
| mesh\_process\_error | string | Error message if processing failed |

**Surface Analysis Fields (prefix: surf\_)**

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| surf\_roof\_area | float | Roof surface area (m²) |
| surf\_footprint\_area | float | Building footprint area (m²) |
| surf\_wall\_area | float | Total wall area (m²) |
| surf\_sloped\_area | float | Sloped surface area (m²) |
| surf\_total\_area | float | Total surface area (m²) |
| surf\_building\_height | float | Calculated building height (m) |
| surf\_wall\_perimeter | float | Estimated wall perimeter (m) |
| surf\_roof\_complexity | float | Roof complexity ratio (0-1) |
| surf\_min\_elevation | float | Minimum Z coordinate |
| surf\_max\_elevation | float | Maximum Z coordinate |
| surf\_horizontal\_faces | int | Count of horizontal faces |
| surf\_vertical\_faces | int | Count of vertical faces |
| surf\_sloped\_faces | int | Count of sloped faces |
| surf\_analysis\_error | string | Error message if analysis failed |

**Processing Status Fields**

* processing\_status - "success" or "failed"
* processing\_error - Overall error message if failed

**Performance Tips**

1. **Test First**: Always run with --limit 100 to verify everything works
2. **Workers**: Use --workers equal to your CPU cores minus 1
3. **Memory**: For large datasets (>500k buildings), consider processing in chunks
4. **Storage**: Ensure sufficient disk space for output files

**Troubleshooting**

1. **Import Errors**: Run python test\_imports.py to verify installation
2. **Memory Issues**: Reduce --workers or process in smaller chunks using --limit
3. **GDB Access**: Ensure the GDB file path has no special characters
4. **Missing Libraries**: Install with python -m pip install [library\_name]

**Processing Time Estimates**

* 100 buildings: ~10-30 seconds
* 10,000 buildings: ~10-20 minutes
* 100,000 buildings: ~1-2 hours
* 1,700,000 buildings: ~10-20 hours (depending on CPU and workers)

**Notes**

* Surface classification uses 10° tolerance for horizontal/vertical determination
* Footprint is defined as horizontal surfaces in the lowest 10% of building height
* Wall perimeter is estimated from wall area divided by building height
* All area measurements are in square meters (m²)
* All volume measurements are in cubic meters (m³)
* Coordinates are preserved in the original Swiss coordinate system

**Authors**

Developed by the Federal Office for Buildings and Logistics BBL for processing Swisstopo 3D building data (swissBUILDINGS3D 3.0).