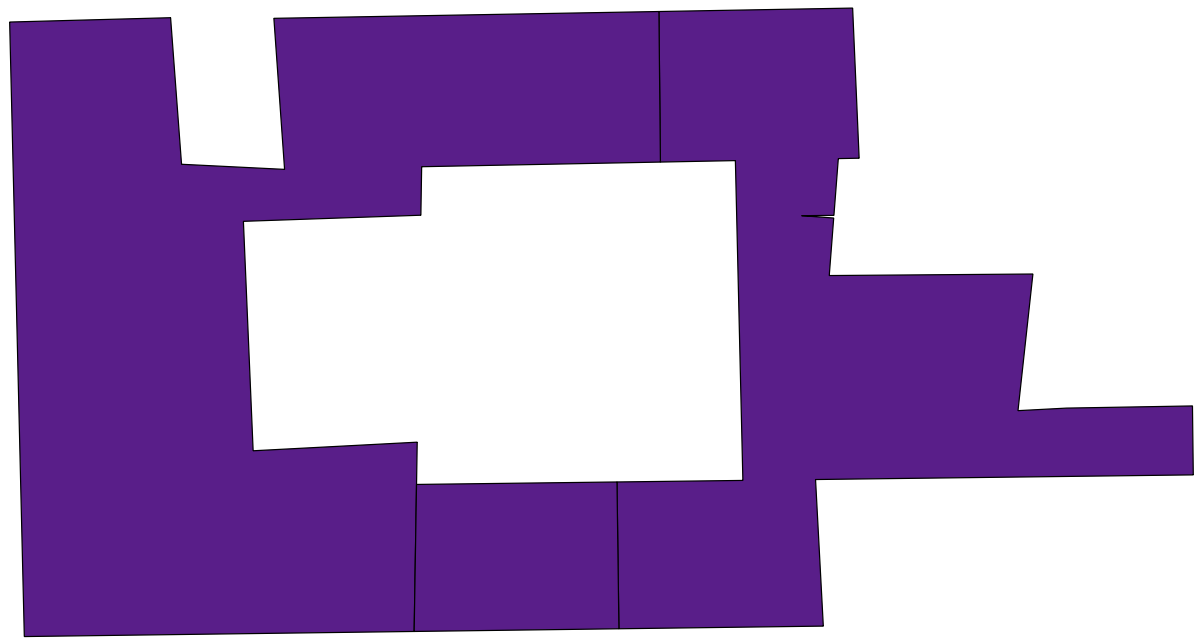
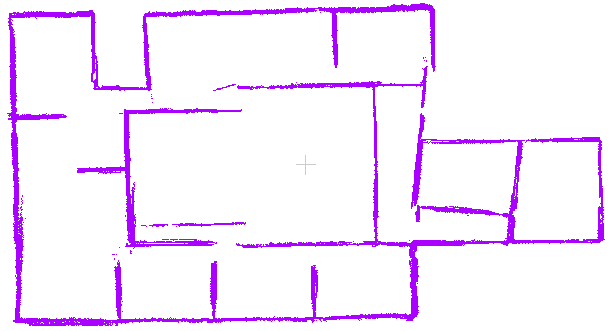
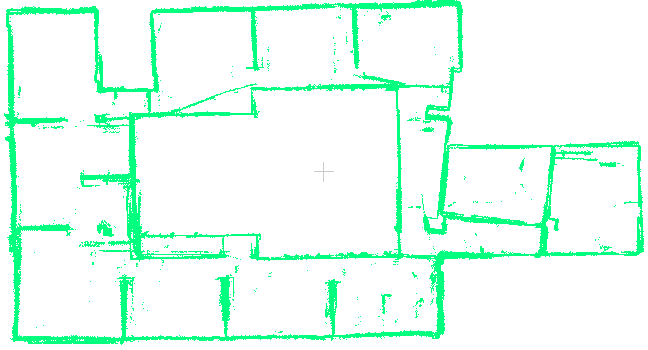
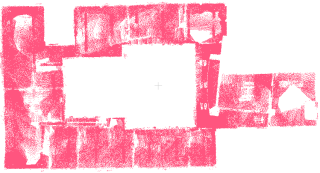
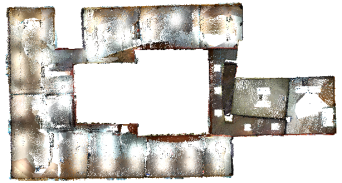
# Indoor Floor-map Reconstruction

## Working flow



Raw point cloud

1. classification (vertical points + horizontal points)

2. plane detection

3. Wall refinement

4. Floormap reconstruction

Input:

* Point cloud captured by LiDAR, RGBD camera or UAV

Output:

* Building boundary
* Indoor polygons
* Indoor polylines

Intermediate results:

* Classified point cloud (horizontal/vertical)
* Segmented plane point cloud and plane parameters
* Polygon and polyline labels (inner/outter; one-side/double-side wall; border/inner wall)

## Detailed Analysis

### Classification

#### Method

Classify the point cloud into horizontal/vertical groups by its normal.

#### Usage

:/utest\_iplclassifier [input point cloud] [output directory]

Input:

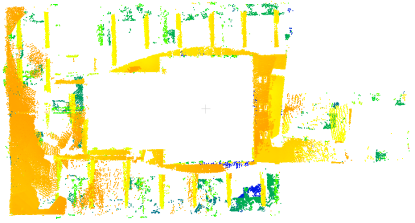
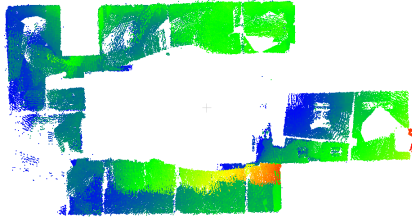
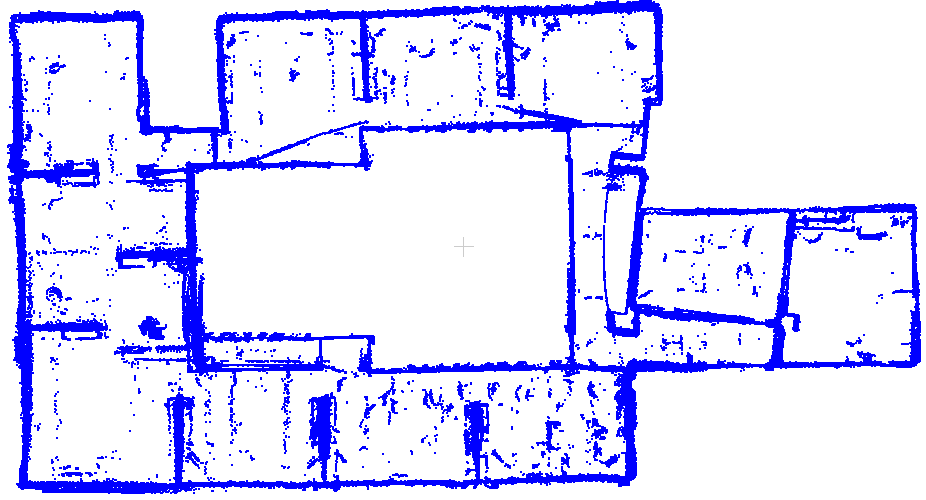
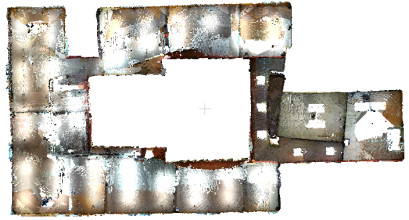
* Raw point cloud file

Output:

This application will output three files in the given directory:

* wall\_pts.pcd
* ceiling\_pts.pcd
* outlier\_pts.pcd

#### Result analysis



vertical points

ceiling points

outliers

1. The expected outliers only contain points on furnitures, but the result contains part of ceiling.
2. The ceiling points is colored by height ramp. It shows obviously unflat. The SLAM results need to be improved.
3. The fixed ceiling threshold is unable to handle this result. A block based statistic is feasible solution. Later I will add it to the system.

### Plane detection

#### Method

Effcient RANSAC for plane fitting [1]

#### Usage

:/ utest\_model\_fitting [vertical\_pts.pcd] [eRANSAC directory] [--doRANSAC]

Input:

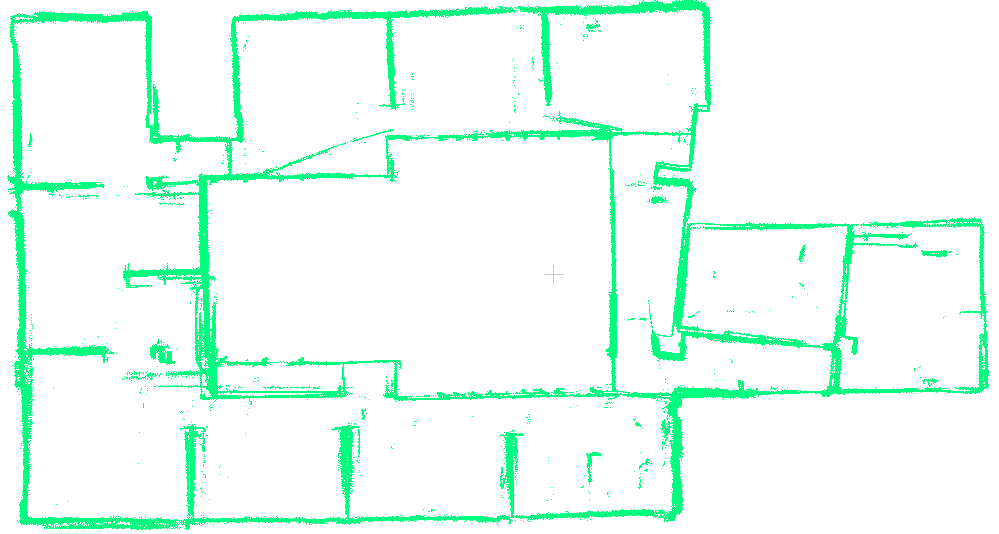
* The vertical point cloud file
* --doRANSAC: using eRANSAC algorithm

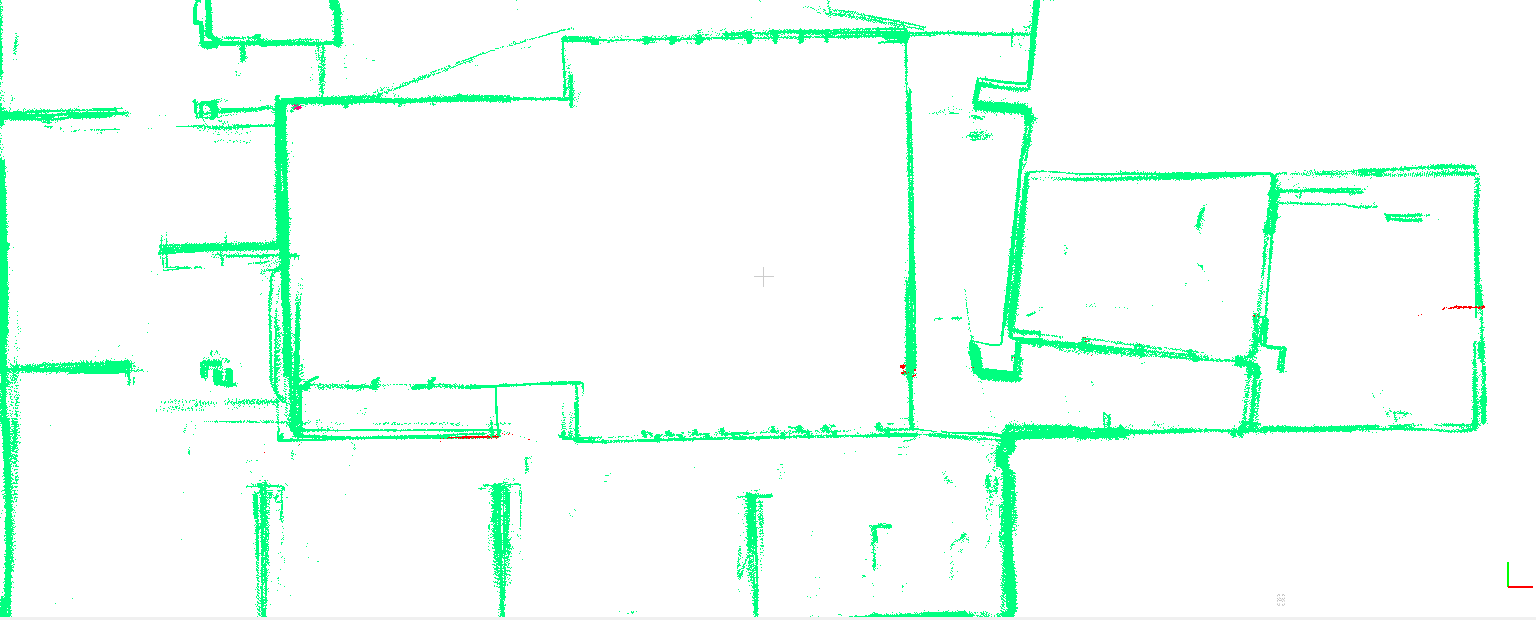
Output:

output seperated point cloud files and plane parameters files in the given directory.

* wall\_model\_0000.pcd the point cloud belongs to plane 0000
* wall\_model\_0000.param the parameters of plane 0000

#### Result analysis





This step tries to detect planes as more as possible. Many fake planes are involved (the red points).

It is a processing strategy as we can filter this fake planes later.

### Wall refinement

#### Method

* A voxel-based method is developed to analyze the distribution of points in a plane cluster and the relationship among plane clusters.
* an axis-aligned transformation is developed to accelerate the distribution analysis.
* Using a cascade classifier to filter fake planes by the sparse degree, occupied area, height scope and plane fitting errors.
* Merge adjacent planes into onside plane or double-side planes by their normal discrepancy.

#### Usage

:/ utest\_wall\_refinement [points directory] [point file extension] [output directory]

Input:

* points directory
* point file extension: “.pcd”

Output:

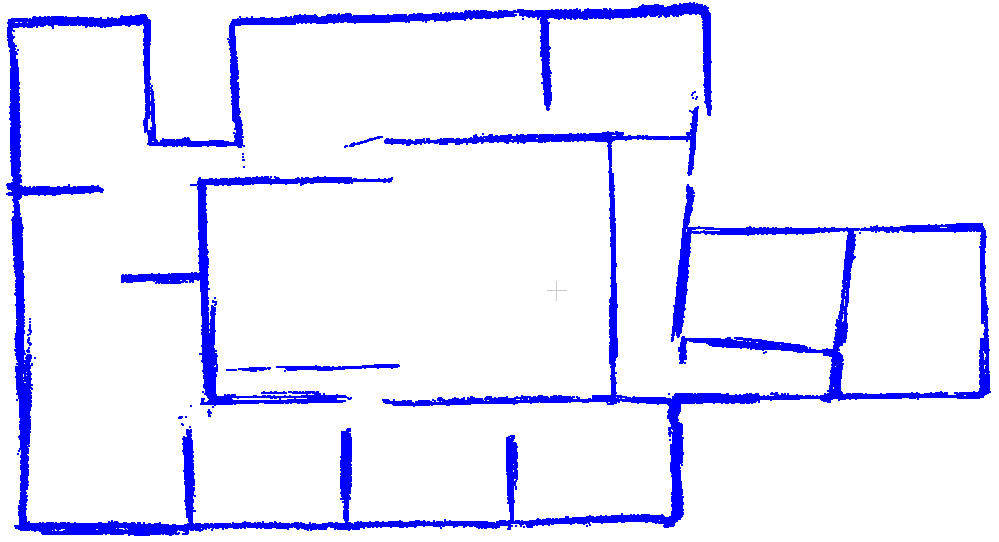
output refined point cloud files and plane parameters files in the given directory.

* wall\_model\_0000.pcd the point cloud belongs to plane 0000
* wall\_model\_0000.param the parameters of plane 0000

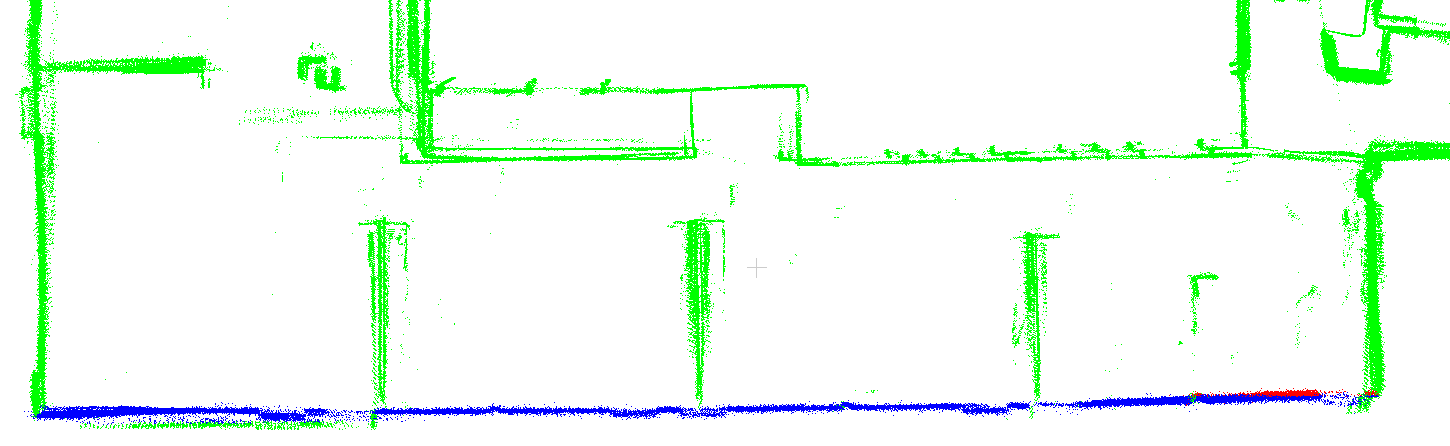
#### Result analysis

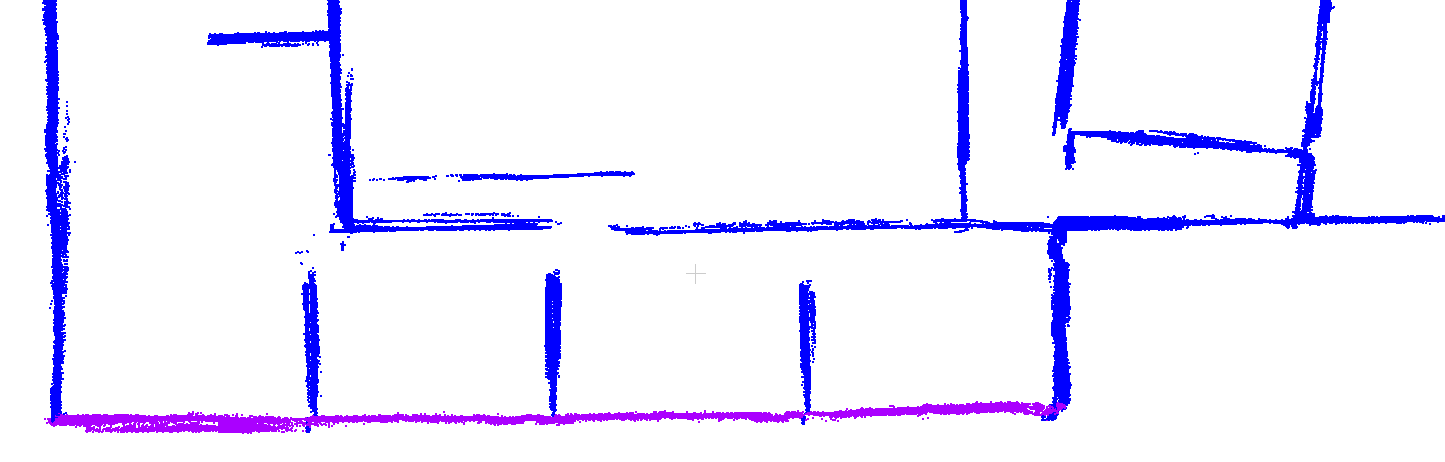
1. Unreliable wall filtered





1. oneside wall merged





1. double side wall merged



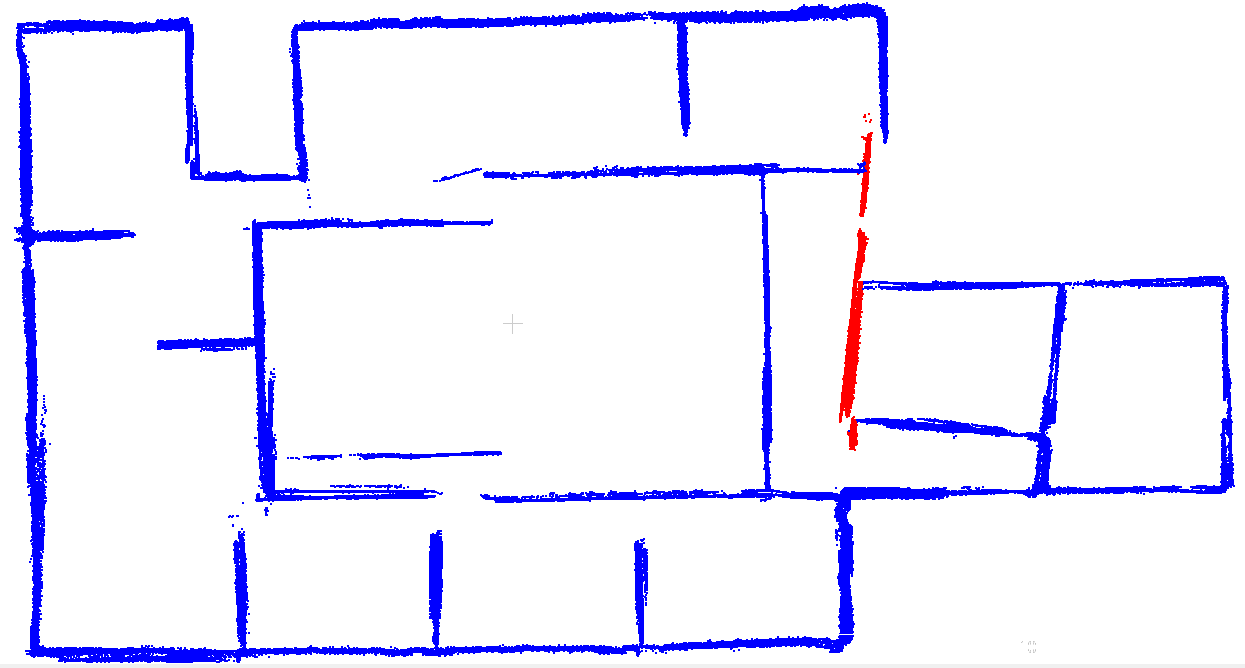
Wall\_model\_0004.pcd



Wall\_model\_0017.pcd



Wall\_model\_0079.pcd



### Floormap reconstruction

#### Method

The developed method is similar to [2, 3].

* Space partition by line arrangement [4]
* Formulize the reconstruction as a binary labeling problem and designe a energy function as follows:

 where, 

Data term:



 which is an indicate of the sparse degree of the space

Smooth term:



Two binary cost are given, experiments show the second one is better.





 is a ratio to balance the data term and smooth term.

The solution is a batch of labels for the cells.

A global optimized method named Graphcut is used to solve this function.



* Postprocessing, like small polygon removal, polyline simplification. Topology are preserved in the whole process.

#### Usage

:/ utest\_floormapRec [points directory] [point file extension] [output directory]

Input:

* points directory
* point file extension: “.pcd”

Output:

output vector files in the given directory.

* boundary.shp the boundary of the building
* all\_polygons.shp the inner polygons
* Polylines.shp the polylines

:/ utest\_polygon\_simplification [original polygon file] [simplified polygon file] [parameter]

Input:

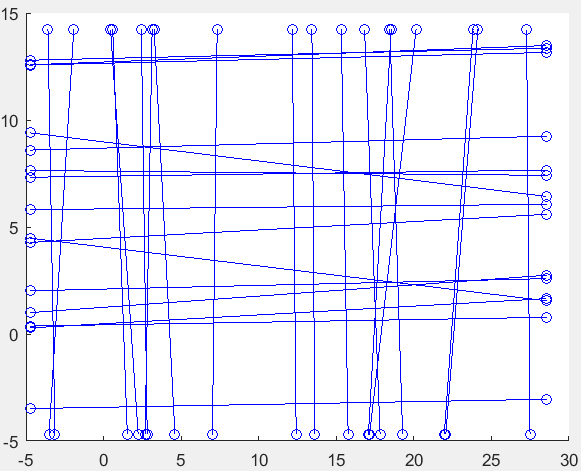
* original polygon file
* parameter: “0.5”

Output:

* simplified polygon file

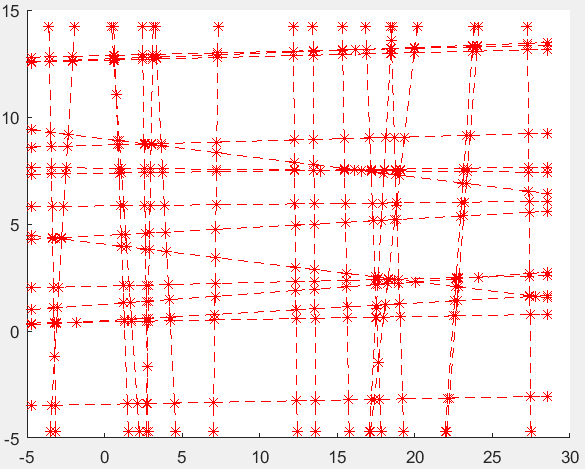
#### Result analysis

1. space partition by line arrangement



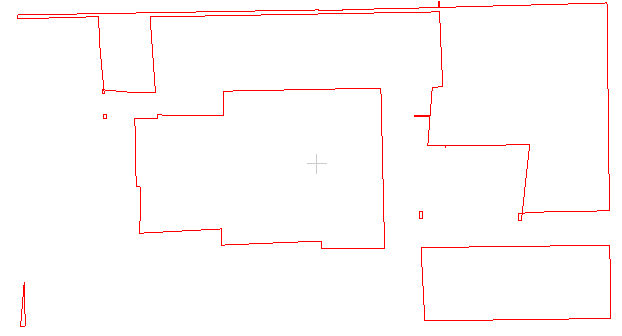
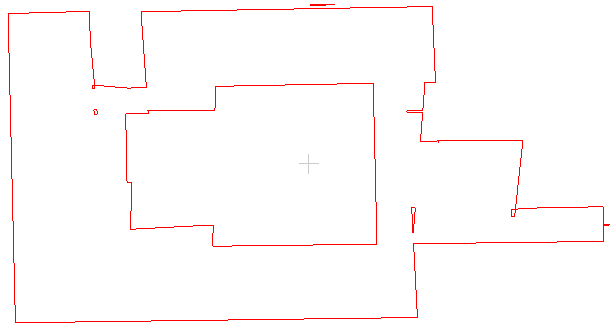
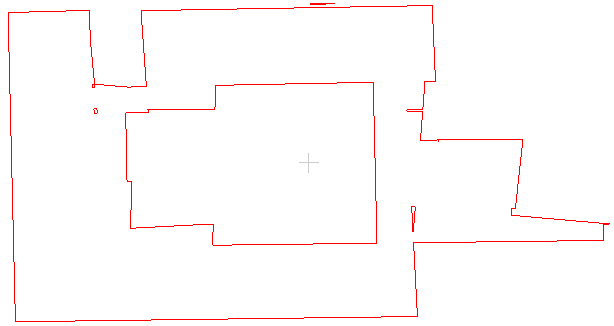
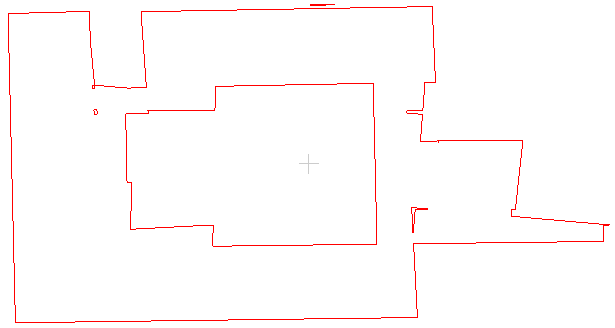
Project the detected planes onto XOY

plane and extend them



Find all intersection points and construct an arrangement containing vertices, edges and faces and the topology.

1. The influence of beta (GraphCut)



0.0

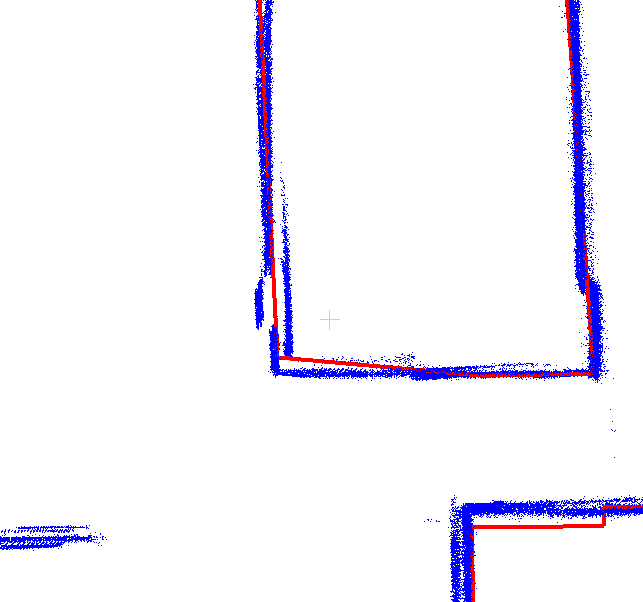
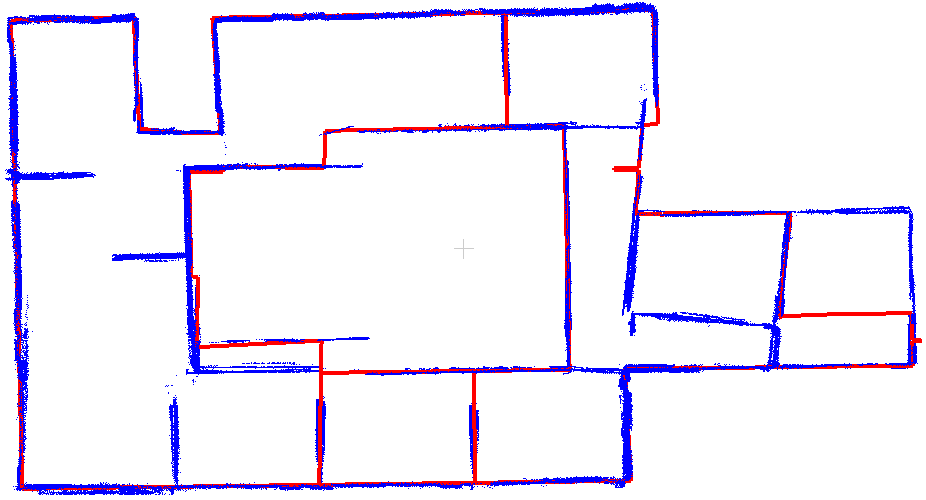
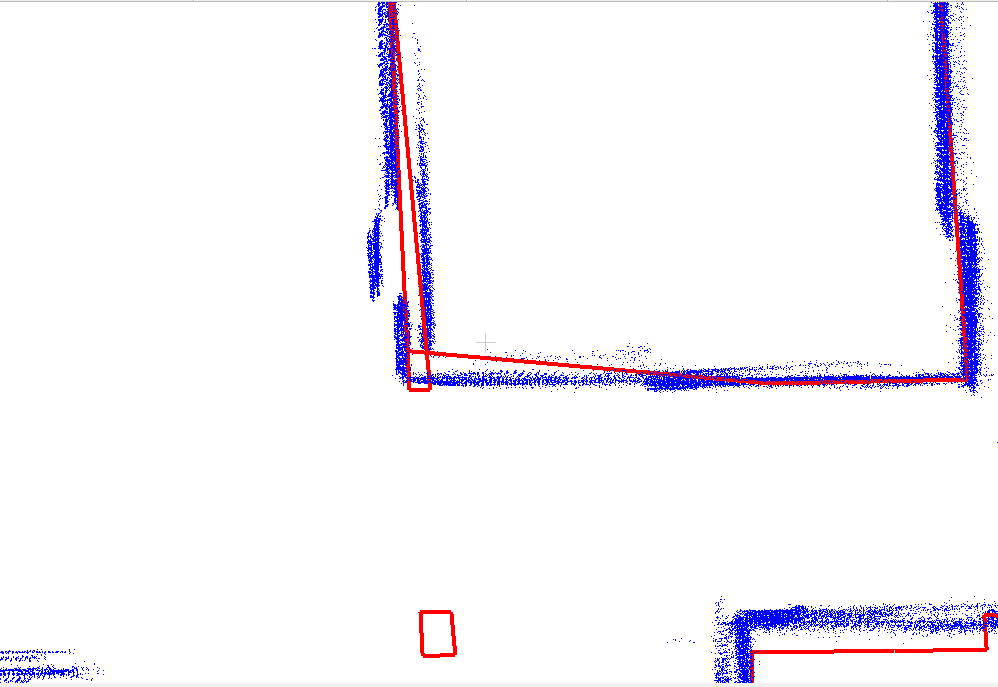
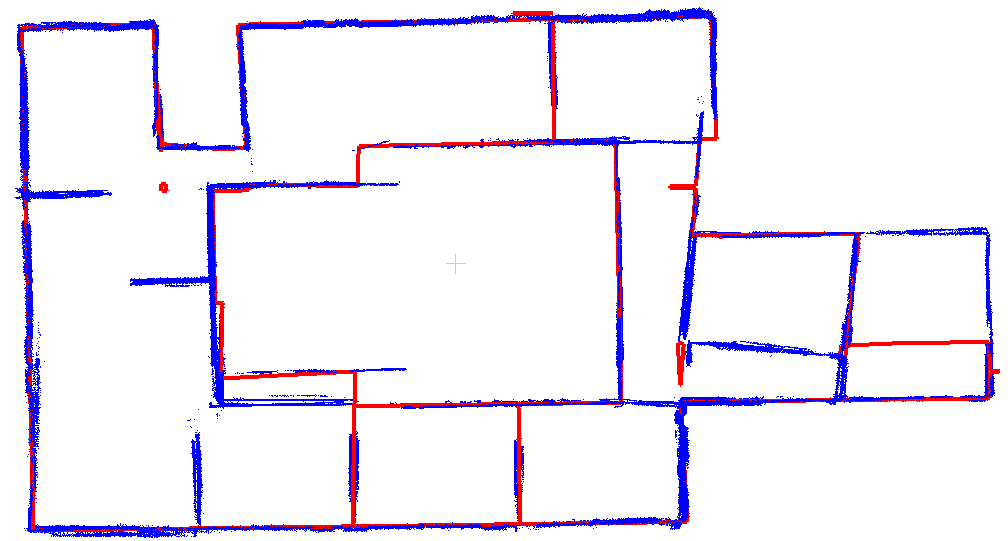
0.05

0.08

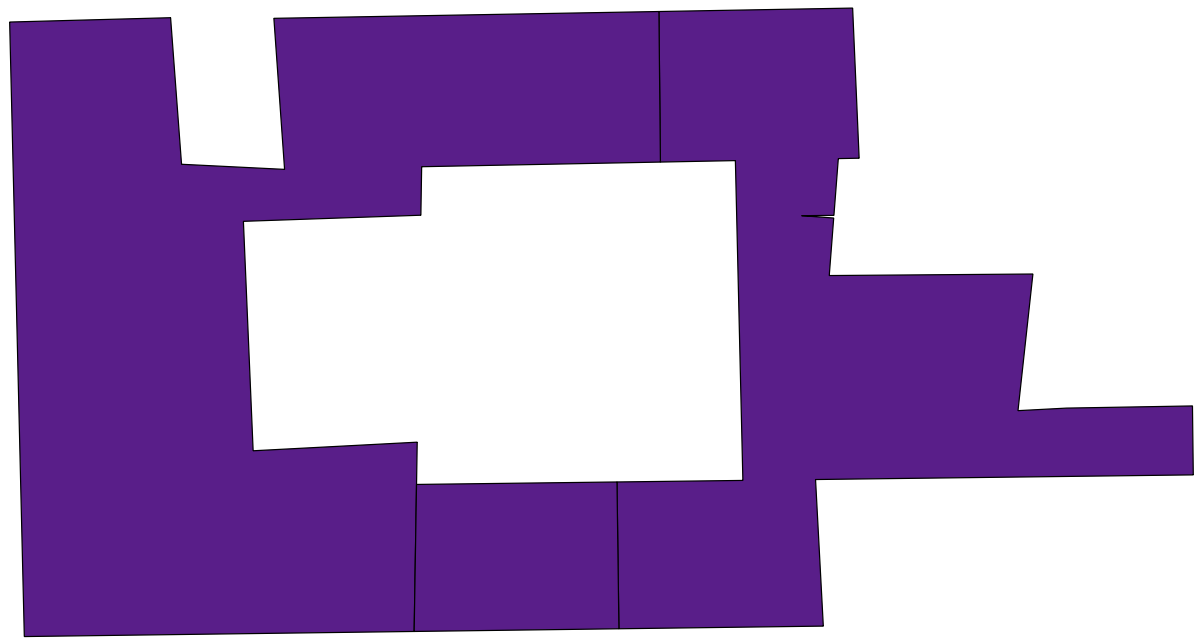
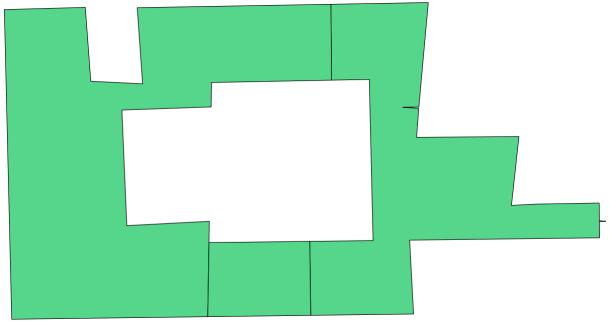
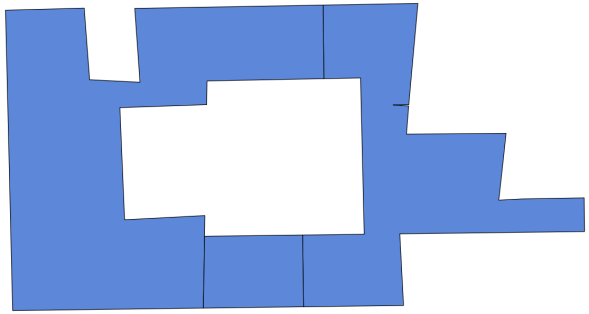
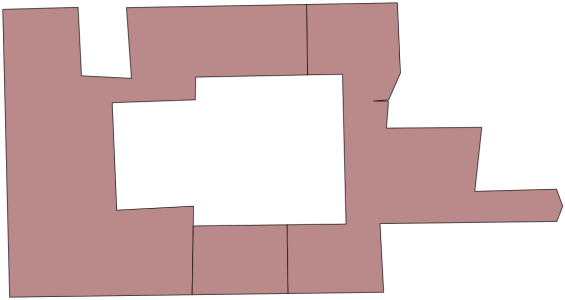
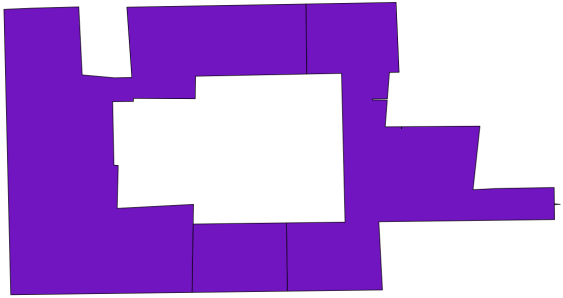
0.1

the smooth term can restrict the boundary alone the wall, but it is sensitive to ’beta’. The data term (ceiling points distribution) plays the most important role in the results.

1. Small polygon removing



1. Polyline simplification



Original polygon

Douglas-Peucker

Squared distance cost

Scaled squared distance cost

Hybrid squared distance cost

# Outdoor Modeling

Note: The boundary is derived from the indoor point cloud. This experiment is just validating the working flow. Later I will test the UAV point cloud for building boundary extraction.

## Experiment Plan

Input:

* Building boundary

Output:

## Working flow

## Results

## Comments

# Indoor Modeling

## Experiment Plan

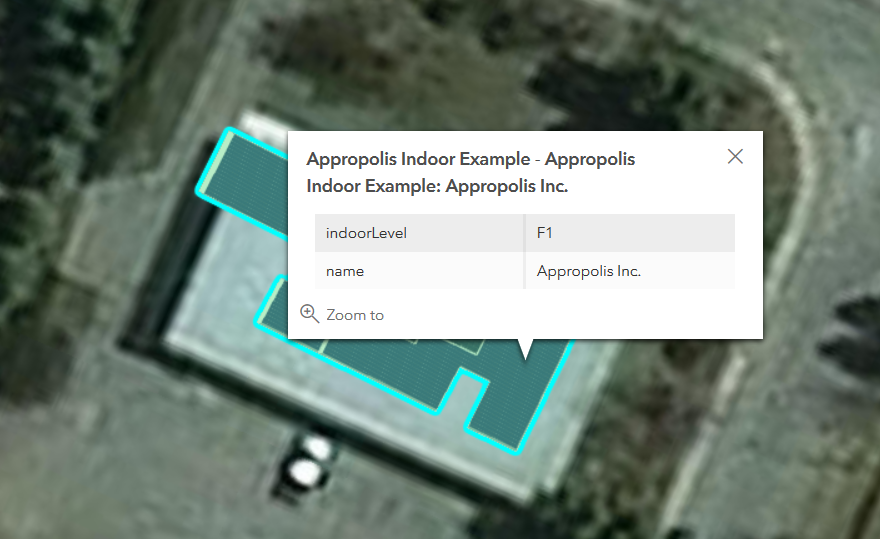
When it comes to modeling, the same challenge is often faced by both outdoor and indoor modeling: the preparation of an accurate footprint. An accurate footprint in both shape and spatial location can greatly increase the quality of the final model output.

Input:

* inner polygons
* Polylines

Output:

An indoor 3D model was generated based on the indoor footprint model. The footprint model provides useful information regarding room segmentation and wall configuration, which can be helpful in simplifying indoor modeling procedure. However, since the footprint does not contain spatial information, the location of the model has to be adjusted manually.

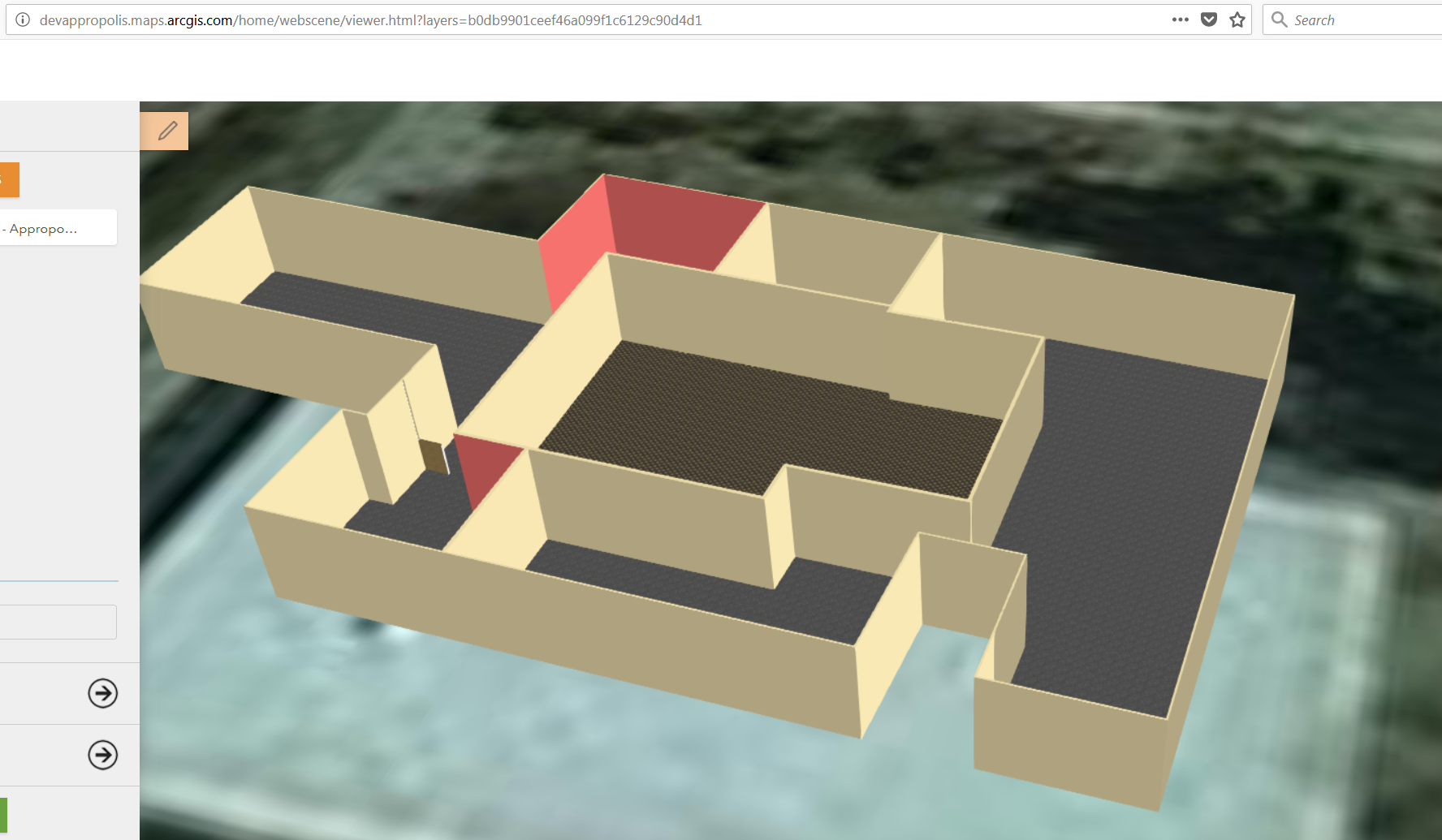


## Working flow

1. Convert the indoor polygon to Sketchup accepted file format: dxf.
2. Import the .dxf file into sketchup. Because .dxf only stores spatial data in polyline formats, simply draw along the boundaries to generate polygons out of the polylines.
3. Resume the routine of indoor/outdoor modeling

## Results

Below is an example of a simplified indoor model published in ArcGIS portal. It also supports user-defined pop-ups.



## Comments

The 3D modeling procedure can be greatly simplified provided with auto-generated 2D footprints. The results have been proven to provide direct service to front-end. However, more information could have been provided in the future to increase the level of automation in the process (e.g. spatial information and indoor detail detection).

Reference

[1] Schnabel, R., et al. (2007). Efficient RANSAC for point‐cloud shape detection. Computer graphics forum, Wiley Online Library

[2] Ochmann, S., et al. (2016). "Automatic reconstruction of parametric building models from indoor point clouds." Computers & Graphics **54**(Supplement C): 94-103.

[3] Mura, C., et al. (2014). "Automatic room detection and reconstruction in cluttered indoor environments with complex room layouts." Computers & Graphics **44**: 20-32.

[4] Fogel, E., et al. (2012). CGAL arrangements and their applications: A step-by-step guide, Springer Science & Business Media.