

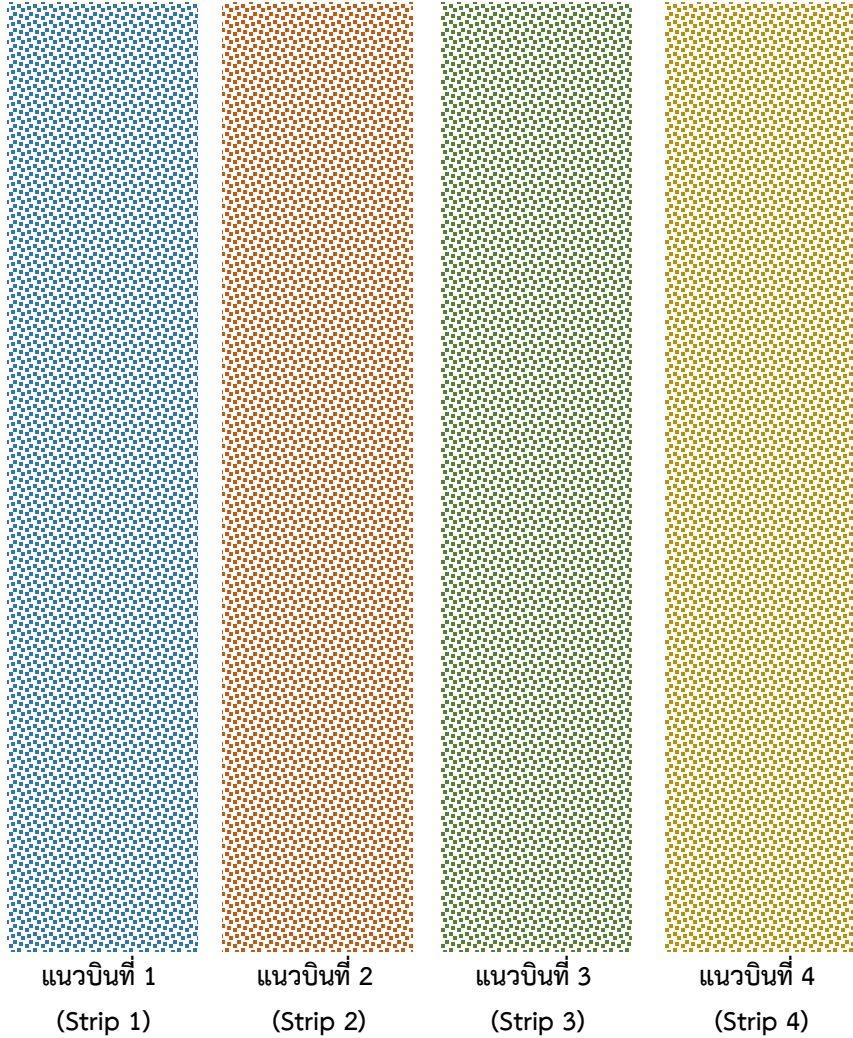
Software : ChkStrip_LCP.py and EstimLCP.py



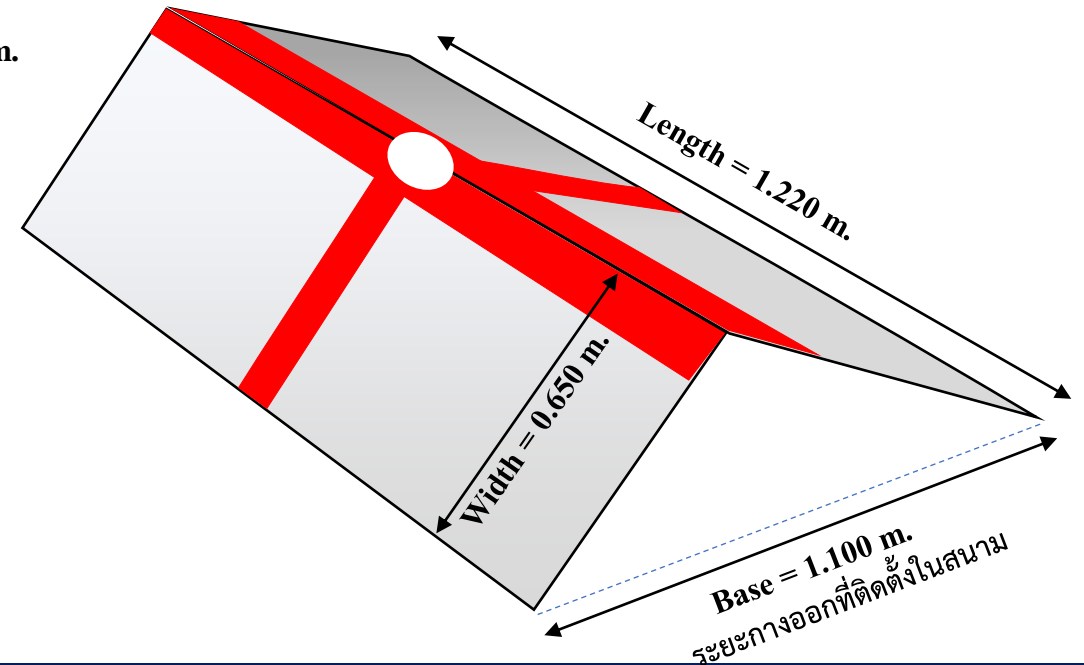
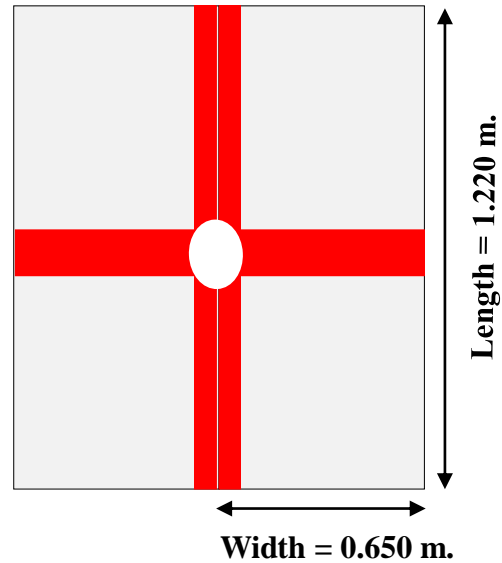
Thirawat Bannakulpiphat & Phisan Santitamnont

Department of Survey Engineering, Chulalongkorn University

E-mail: thirawat.bannakulpiphat@gmail.com and phisan.chula@gmail.com

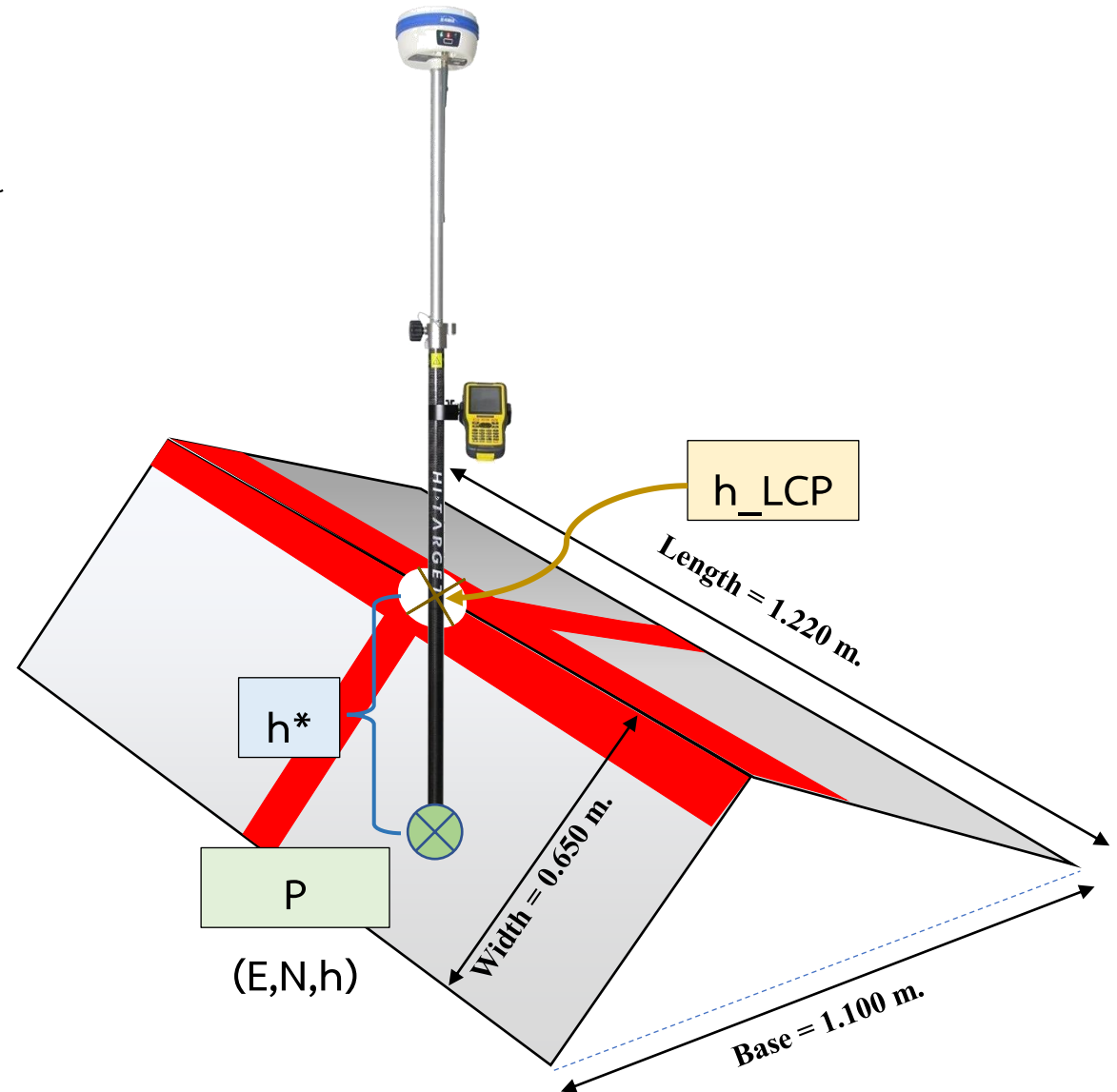


ข้อมูลพอยต์คลาวด์เป็นแนวจีน (Strip)

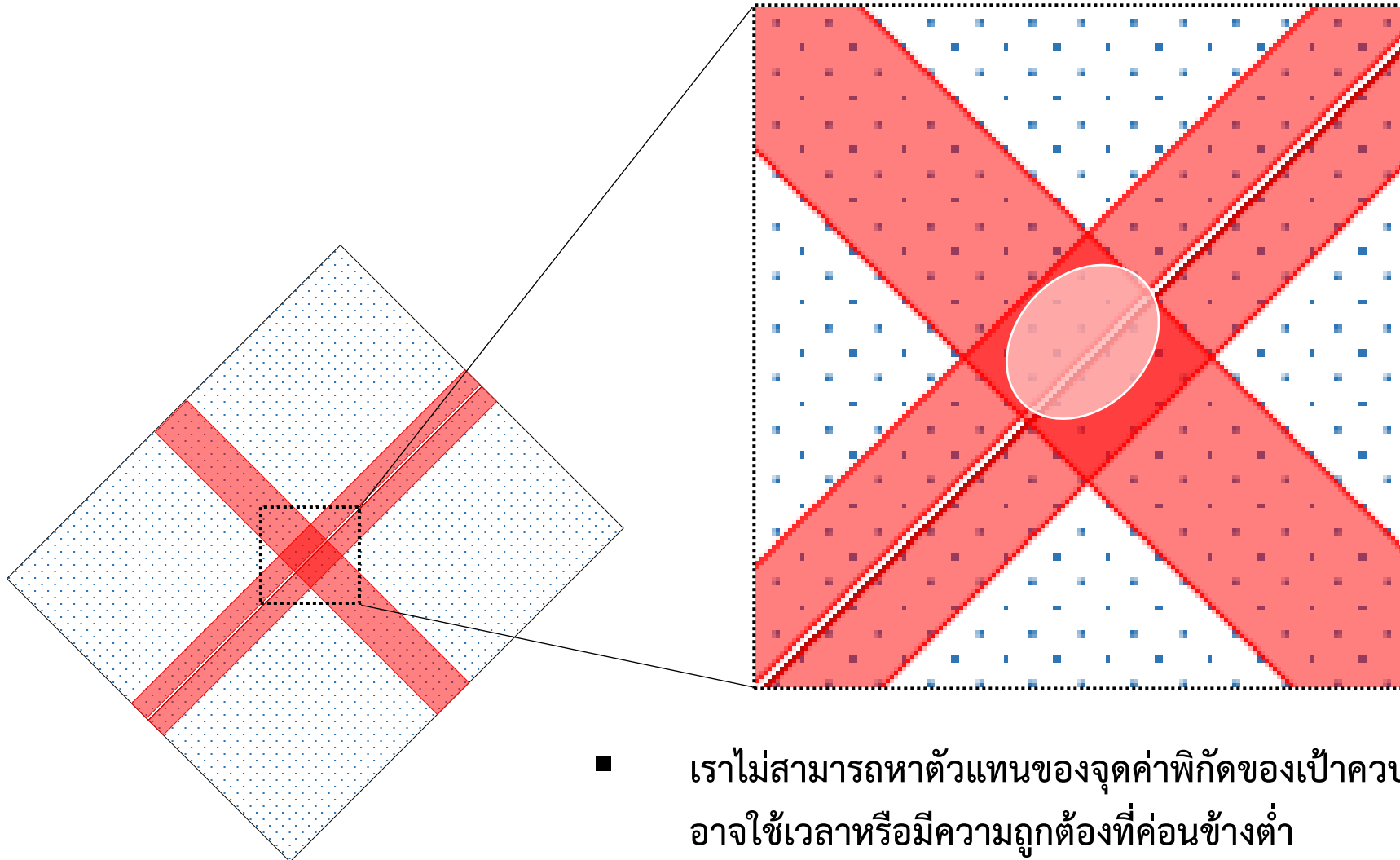


- จากรูปภาพด้านบนพบว่า ค่าพิกัดได้จากการรังวัด GNSS ด้วยเทคนิควิธีการรังวัดแบบจนวนลิในทันที (RTK) ให้ค่าพิกัดทางดิ่งของจุด P ดังนั้นหากต้องการความสูงของเป้าบังคับทรงระนาบหลังคาของเลเซอร์สแกนจะต้องทำการบวกระยะจากหัวหมุด P ถึงบริเวณสันหลังจากของเป้าบังคับทรงระนาบของเลเซอร์สแกนได้ ดังสมการต่อไปนี้

$$h_{LCP} = h_P + h^*$$



- เพื่อช่วยงานรังวัดจุดค่าพิกัด 3D ณ ตำแหน่งกึ่งกลางหลังคา LCP



- เราไม่สามารถหาตัวแทนของจุดค่าพิกัดของเป้าควบคุมได้ หากทำได้ อาจใช้เวลาหรือมีความถูกต้องที่ค่อนข้างต่ำ

```
python ChkLCP_strip.py -h
```

```
usage: ChkLCP_strip.py [-h] [-r REDUCE] [-s SHRINK] [-y] [-l LIMIT] LAS_PATTERN
```

Read lidar by flight-strip and analyze if any LCP falls on any flight-strip, then write the result in YAML for processing in further step with EstimLCP.py.
(P.Santitamnont, Chula.Unive Feb, 2023)

positional arguments:

LAS_PATTERN input LAS_PATTERN `"/Data/AA450/LasFile/AA450-*.las"`

options:

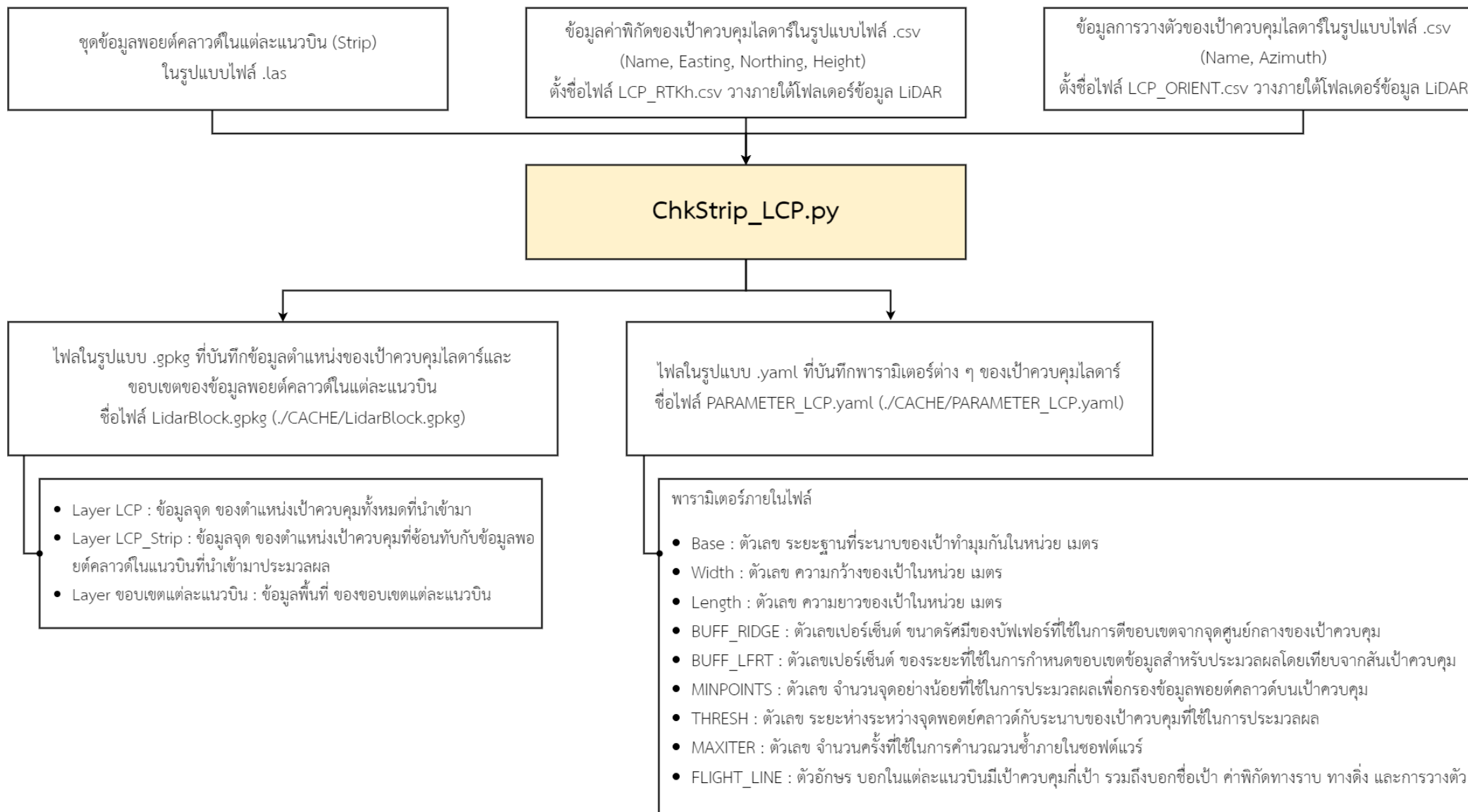
-h, --help show this help message and exit

-r REDUCE, --reduce REDUCE
 reduce number of point-cloud, default 1000-times, suggest 10/100/1000/10000

-s SHRINK, --shrink SHRINK
 shrink polygon hulling point-cloud by default -5 meter

-y, --yaml generate YAML file for later used by EstimLCP.py...

-l LIMIT, --limit LIMIT
 limit only first n-files !!! FOR DEBUG !!!

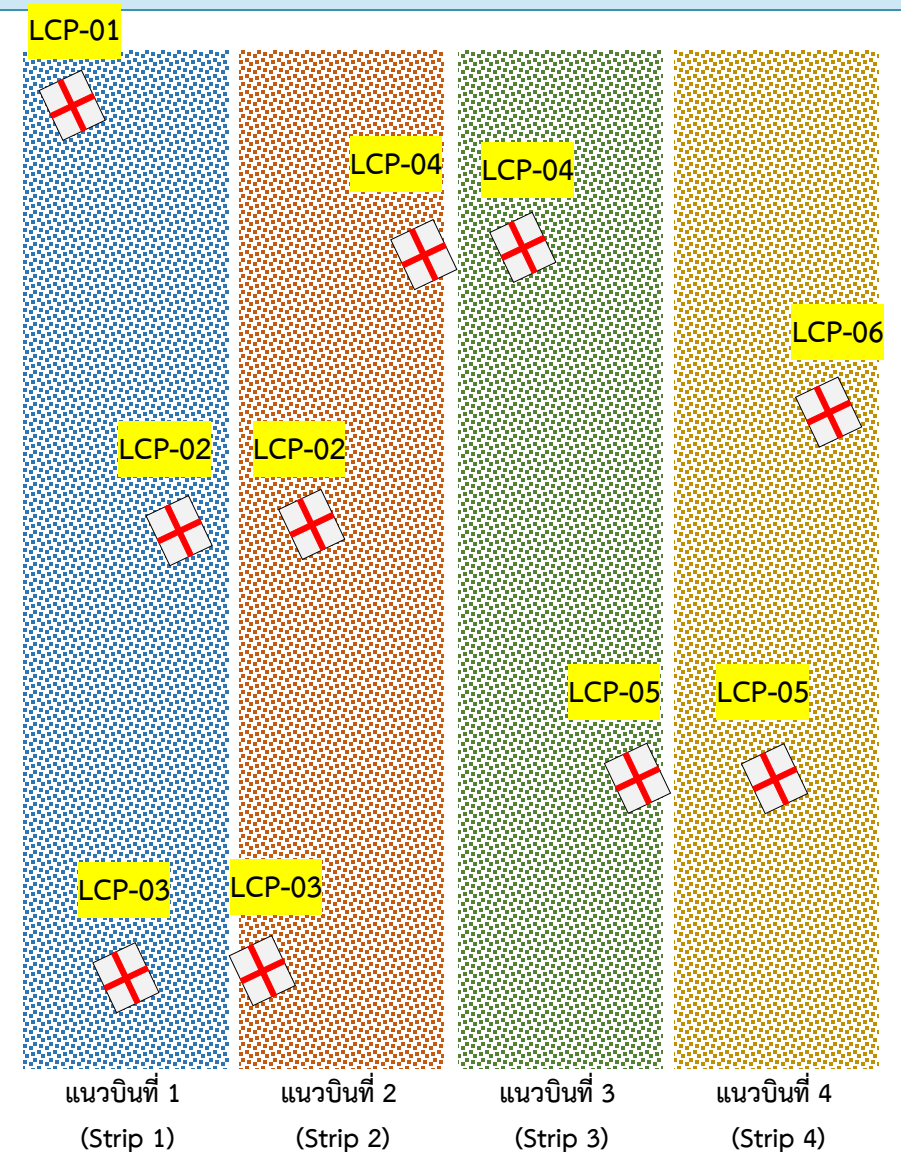


ข้อมูลที่ต้องนำเข้า

- ชุดข้อมูลพอยต์คลาวด์ในแต่ละแนวบิน (Strip) ในรูปแบบไฟล์ .las
- ข้อมูลค่าพิกัดของเป้าควบคุมไลดาร์ในรูปแบบไฟล์ .csv (Name, Easting, Northing, Height) ตั้งชื่อไฟล์ LCP_RTKh.csv วางภายใต้โฟลเดอร์ข้อมูล LiDAR
- ข้อมูลการวางตัวของเป้าควบคุมไลดาร์ในรูปแบบไฟล์ .csv (Name, Azimuth) ตั้งชื่อไฟล์ LCP_ORIENT.csv วางภายใต้โฟลเดอร์ข้อมูล LiDAR

ข้อมูลที่ได้หลังจากประมวลผล

- ไฟล์ในรูปแบบ .gpkg ที่บันทึกข้อมูลตำแหน่งของเป้าควบคุมไลดาร์และขอบเขตของข้อมูลพอยต์คลาวด์ในแต่ละแนวบินชื่อไฟล์ LidarBlock.gpkg (./CACHE/LidarBlock.gpkg)
- ไฟล์ในรูปแบบ .yaml ที่บันทึกพารามิเตอร์ต่าง ๆ ของเป้าควบคุมไลดาร์ชื่อไฟล์ PARAMETER_LCP.yaml (./CACHE/PARAMETER_LCP.yaml)



ข้อมูลที่ต้องนำเข้า

- ชุดข้อมูลพอยต์คลาวด์ในแต่ละแนวกบิน (Strip) ในรูปแบบไฟล์ .las
- ข้อมูลค่าพิกัดของเป้าควบคุมไลดาร์ในรูปแบบไฟล์ .csv (Name, Easting, Northing, Height) ตั้งชื่อไฟล์ LCP_RTKh.csv วางภายใต้โฟลเดอร์ข้อมูล LiDAR
- ข้อมูลการวางตัวของเป้าควบคุมไลดาร์ในรูปแบบไฟล์ .csv (Name, Azimuth) ตั้งชื่อไฟล์ LCP_ORIENT.csv วางภายใต้โฟลเดอร์ข้อมูล LiDAR

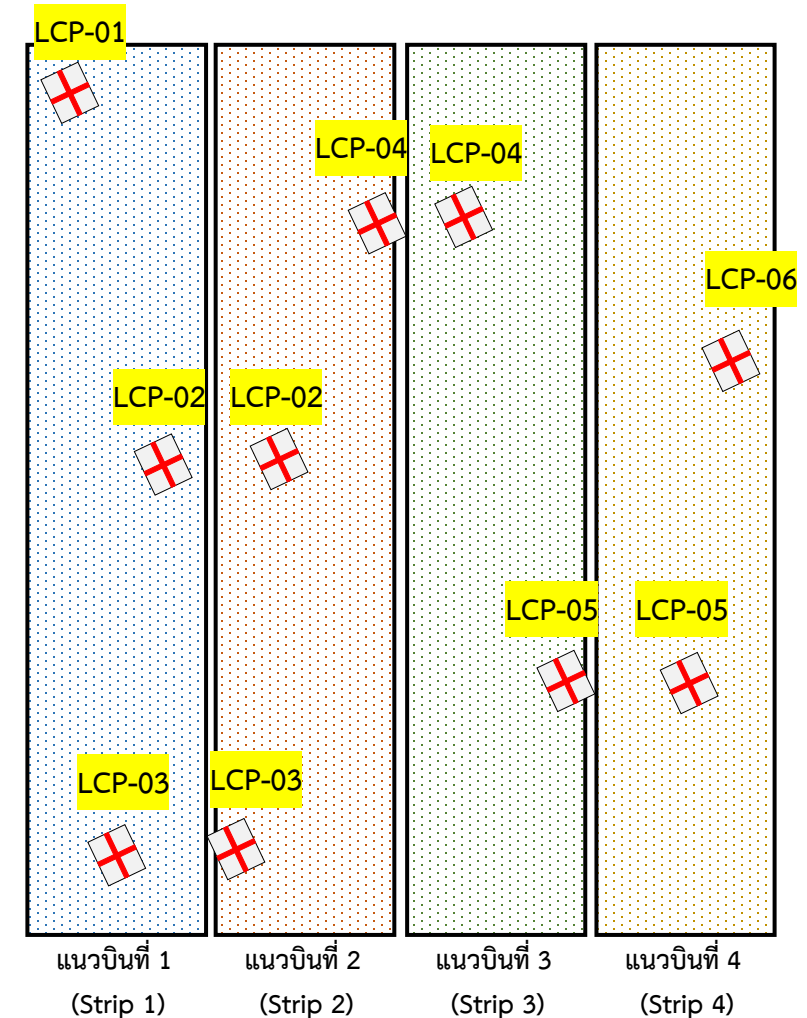
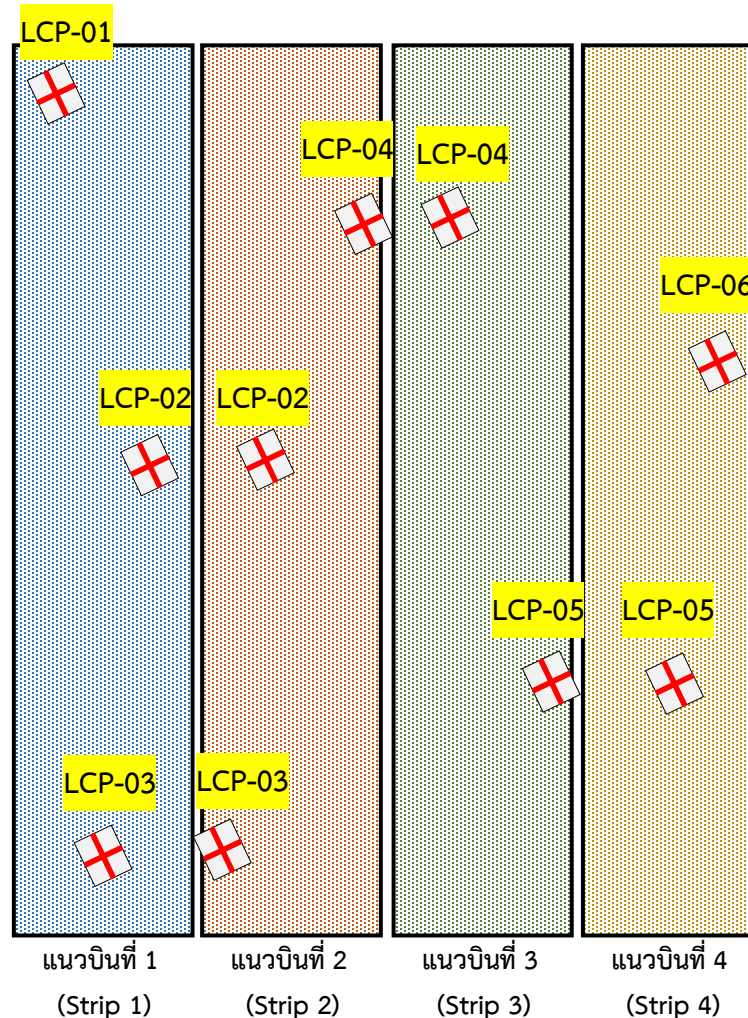
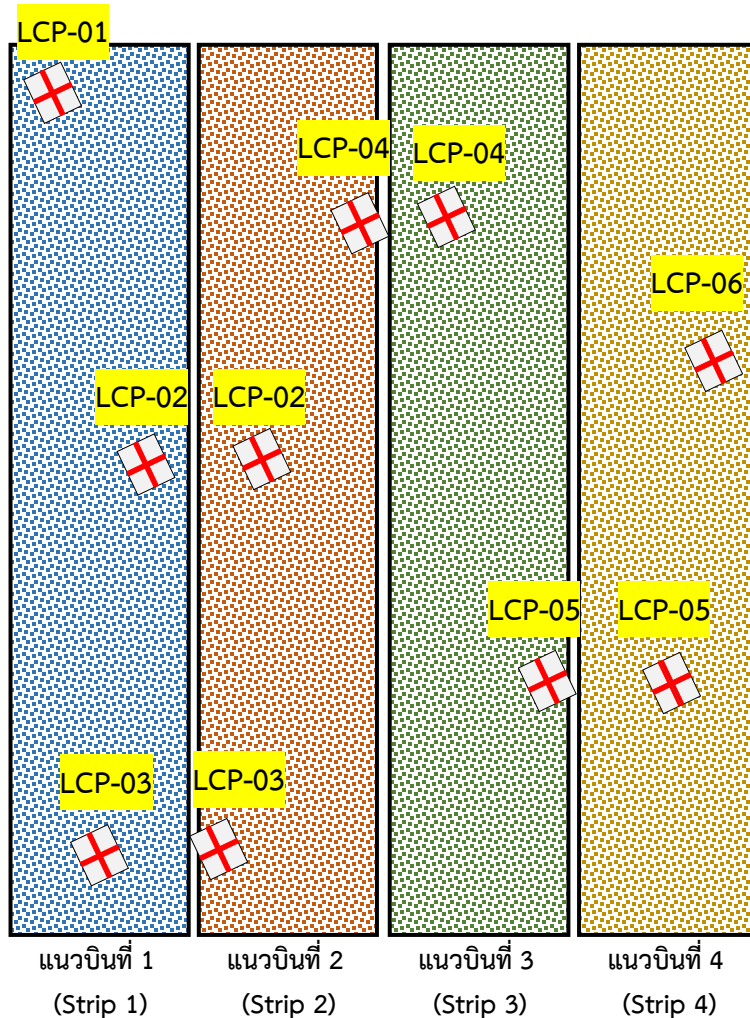
ข้อมูลค่าพิกัดของเป้าควบคุมไลดาร์ ได้มาด้วยวิธีการรังวัดแบบจนวนในทันที (RTK)

ข้อมูลการวางตัวของเป้าควบคุมไลดาร์ได้มาจากการรังวัดบน QGIS

NAME	Northing	Easting	HAE
LCP-01	1605528.377	717418.553	1.151
LCP-02	1605620.326	717552.563	2.307
LCP-03	1605683.495	717697.043	1.490
LCP-04	1605768.160	717830.917	3.438
LCP-05	1605863.974	717960.246	4.093
LCP-06	1605655.491	717365.073	0.883
LCP-07	1605722.976	717476.276	2.762
LCP-08	1605811.713	717599.831	3.646
LCP-09	1605878.421	717757.886	6.726
LCP-10	1605953.416	717899.997	8.637
LCP-11	1605813.784	717283.176	2.202
LCP-12	1605865.322	717389.870	3.341
LCP-13	1605963.981	717514.390	4.580
LCP-14	1606027.335	717677.676	8.789
LCP-15	1606124.053	717803.261	11.184
LCP-16	1605930.336	717190.757	0.853
LCP-17	1606035.136	717335.724	1.051
LCP-18	1606085.675	717458.675	1.986
LCP-19	1606212.081	717551.850	5.141
LCP-20	1606241.924	717708.133	7.392
LCP-21	1606079.114	717071.043	0.097
LCP-22	1606163.019	717231.352	-0.663
LCP-23	1605901.416	717531.824	4.501
LCP-24	1606299.168	717477.921	4.177
LCP-25	1606381.581	717604.707	3.717
LCP-26	1606209.208	716985.734	-0.289
LCP-27	1606298.496	717136.305	0.421
LCP-28	1606355.043	717298.736	2.171
LCP-29	1606458.052	717400.039	4.290
LCP-30	1606525.877	717523.855	5.902
LCP-32	1606245.198	717382.116	0.638

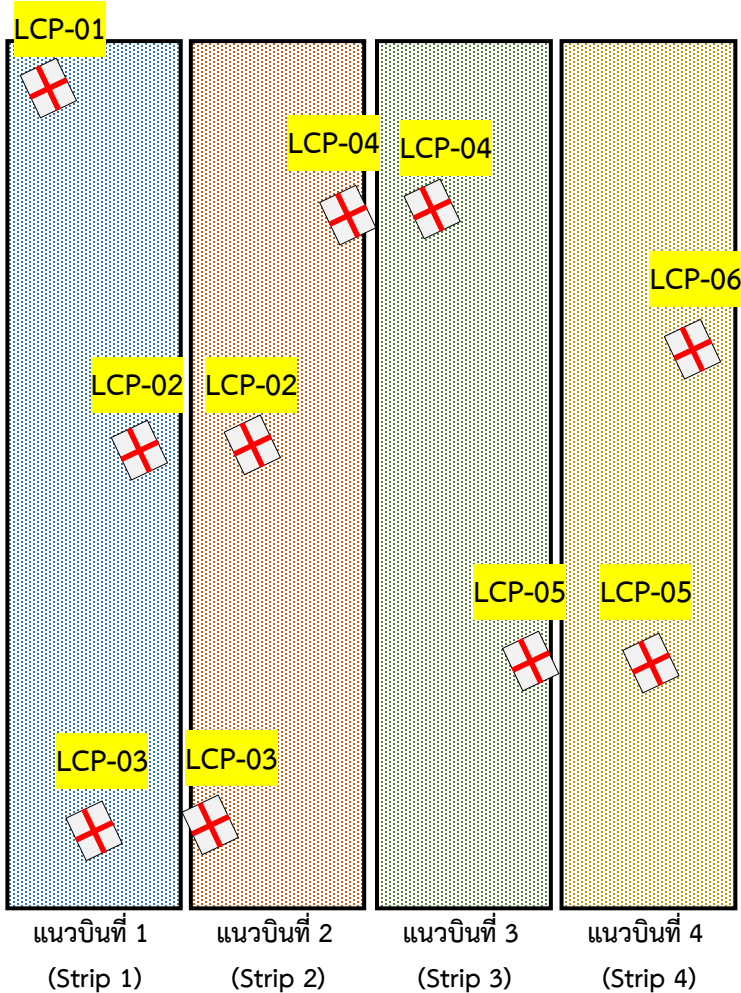
NAME	AZ
LCP-07	12.1
LCP-08	14.1
LCP-09	15.6
LCP-10	15.9
LCP-12	22
LCP-13	17.1
LCP-14	17.1
LCP-15	14.6
LCP-17	10.6
LCP-18	12.8
LCP-19	12
LCP-20	14.9
LCP-22	16.6
LCP-23	16.3
LCP-24	10
LCP-25	0.5
LCP-27	18.3
LCP-28	13.6
LCP-29	15.9
LCP-30	14.9
LCP-32	6.6


```
options:
-h, --help            show this help message and exit
-r REDUCE, --reduce REDUCE
                        reduce number of point-cloud, default 1000-times, suggest 10/100/1000/10000
-s SHRINK, --shrink SHRINK
                        shrink polygon hulling point-cloud by default -5 meter
-y, --yaml            generate YAML file for later used by EstimLCP.py...
-l LIMIT, --limit LIMIT
                        limit only first n-files !!! FOR DEBUG !!!
```



ลดจำนวนข้อมูลพอยต์คลาวด์เพื่อให้สะดวกต่อการประมวลผลตรวจสอบ

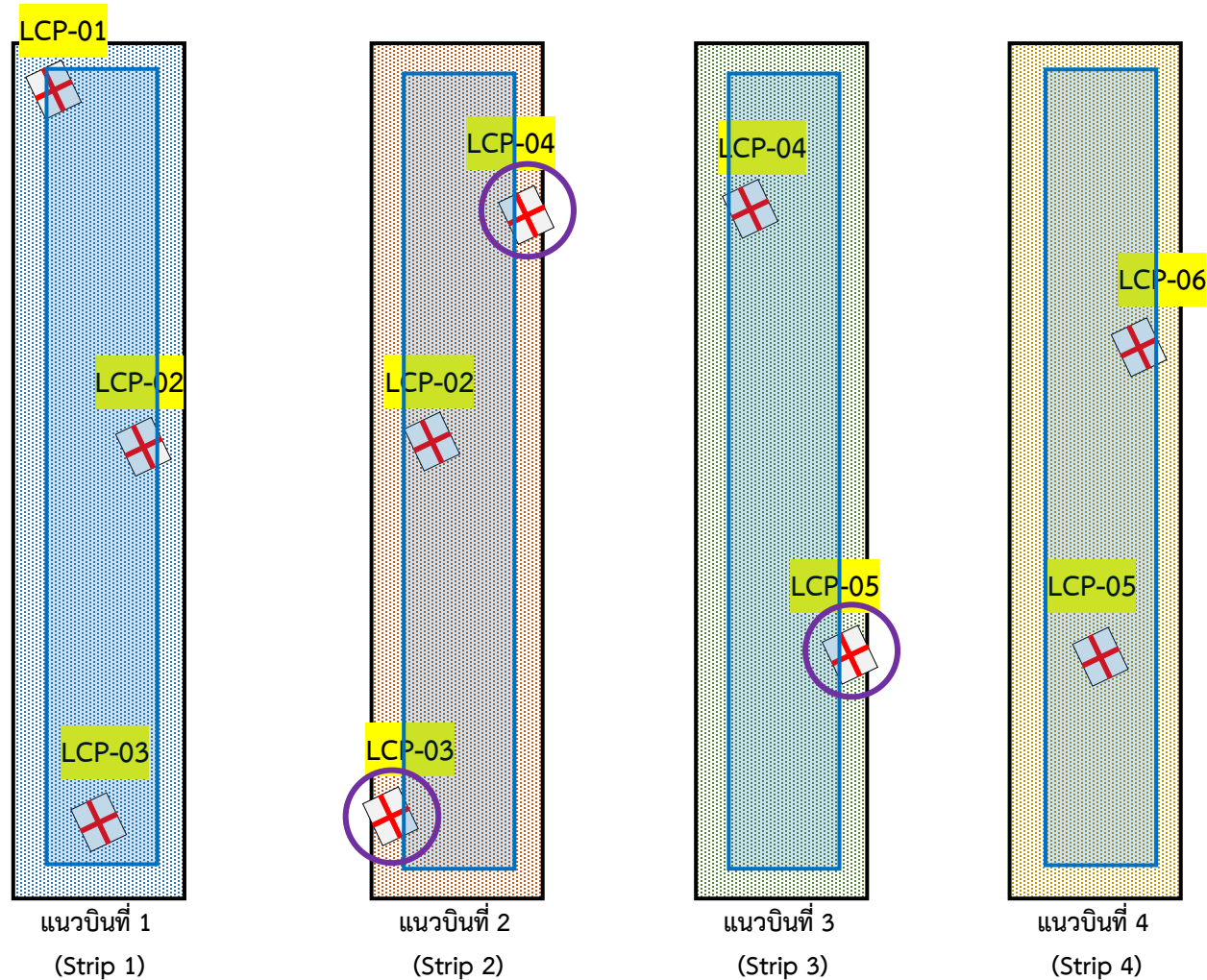
```
options:
-h, --help            show this help message and exit
-r REDUCE, --reduce REDUCE
                        reduce number of point-cloud, default 1000-times, suggest 10/100/1000/10000
-s SHRINK, --shrink SHRINK
                        shrink polygon hulling point-cloud by default -5 meter
-y, --yaml            generate YAML file for later used by EstimLCP.py...
-l LIMIT, --limit LIMIT
                        limit only first n-files !!! FOR DEBUG !!!
```



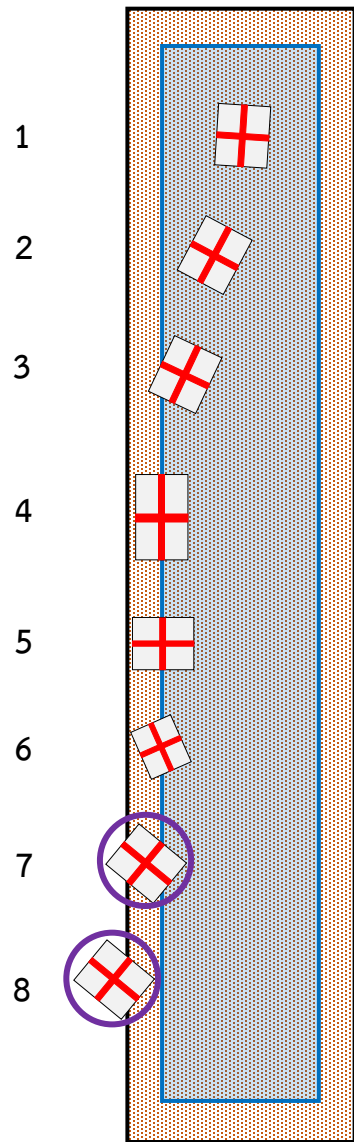
ระยะ shrink



ถือว่าไม่ปรากฏในแนวนั้น



ตรวจสอบเป้าหมายในแต่ละแนวนั้น เพื่อเตรียมความพร้อมในการประมวลผลขั้นต่อไป



หลักการพิจารณาเป้าควบคุมไคดาร์อยู่ในแนวนั้นหรือไม่

- หลังจากซอฟต์แวร์ประมวลผลสร้างบัฟเฟอร์ตามค่าระยะ Shrink แล้ว ซอฟต์แวร์จะพิจารณาโดยการตรวจสอบค่าพิกัดของเป้าควบคุมไคดาร์ว่าอยู่ในส่วนของพื้นที่ Shrink หรือไม่ หากค่าพิกัดอยู่ในพื้นที่จะถือว่าเป้าควบคุมหมายเลขนั้นอยู่บนแนวนั้น
- เหตุผลที่ต้องทำเช่นนี้ เพื่อเป็นการตรวจสอบข้อมูลเพื่อให้มั่นใจว่าข้อมูลพอยต์คลาวด์ตกลงบนเป้าควบคุมทั้งสองระนาบ เนื่องจากข้อมูลพอยต์คลาวด์ในส่วนนี้จะถูกนำไปคำนวณเพื่อหาเรขาคณิตของเป้าควบคุมและไปสู่การประมาณค่าจุดพิกัดตรงกลางของเป้าควบคุมในการคำนวณปรับแก้ Strip Adjustment
- จากตัวอย่างเมื่อพิจารณาส่วนซ้อนของพื้นที่ Shrink และจุดกึ่งกลางของเป้าควบคุม สรุปได้ว่าเป้าหมายเลข 7 และ 8 ถือว่าไม่ปรากฏบนแนวนั้นของข้อมูลไคดาร์แนวนั้น



ระยะ shrink



ถือว่าไม่ปรากฏในแนวนั้น

```
python EstimLCP.py -h
```

```
usage: EstimLCP.py [-h] [-c] [-p] [-s STRIP] [-l LCP] YAML
```

Detect and estimate 3D coordinate of a Gable-roof Lidar Control Plane(LCP)

positional arguments:

YAML input YAML configuration file for LCP detection and positioning

options:

-h, --help show this help message and exit

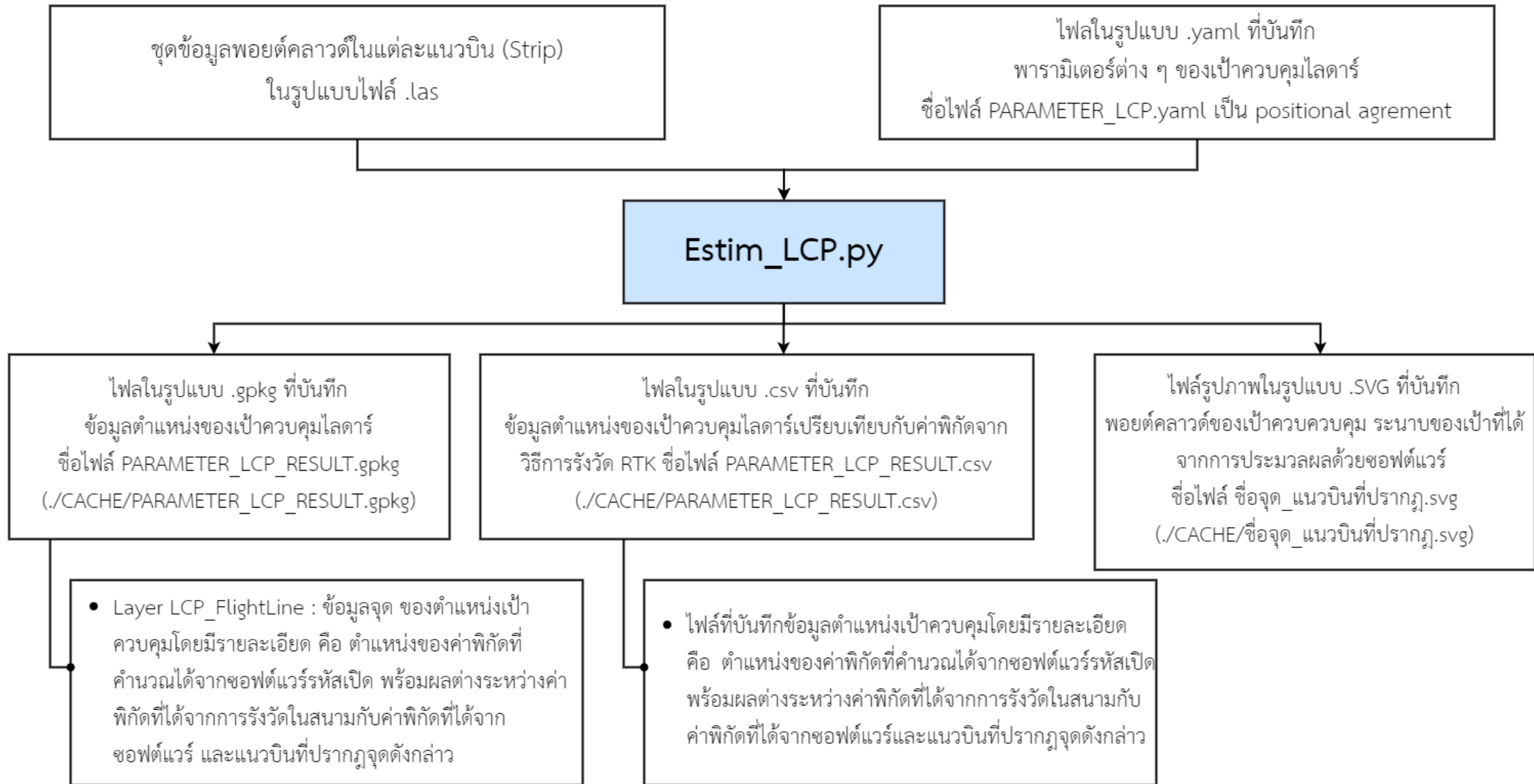
-c, --cache caching the circled target, for DEBUG only !!!

-p, --plot plot 3d LCP target and point cloud

-s STRIP, --strip STRIP

 limit processing by specifying strip name

-l LCP, --lcp LCP limit processing only LCP ,otherwise process all LCPs

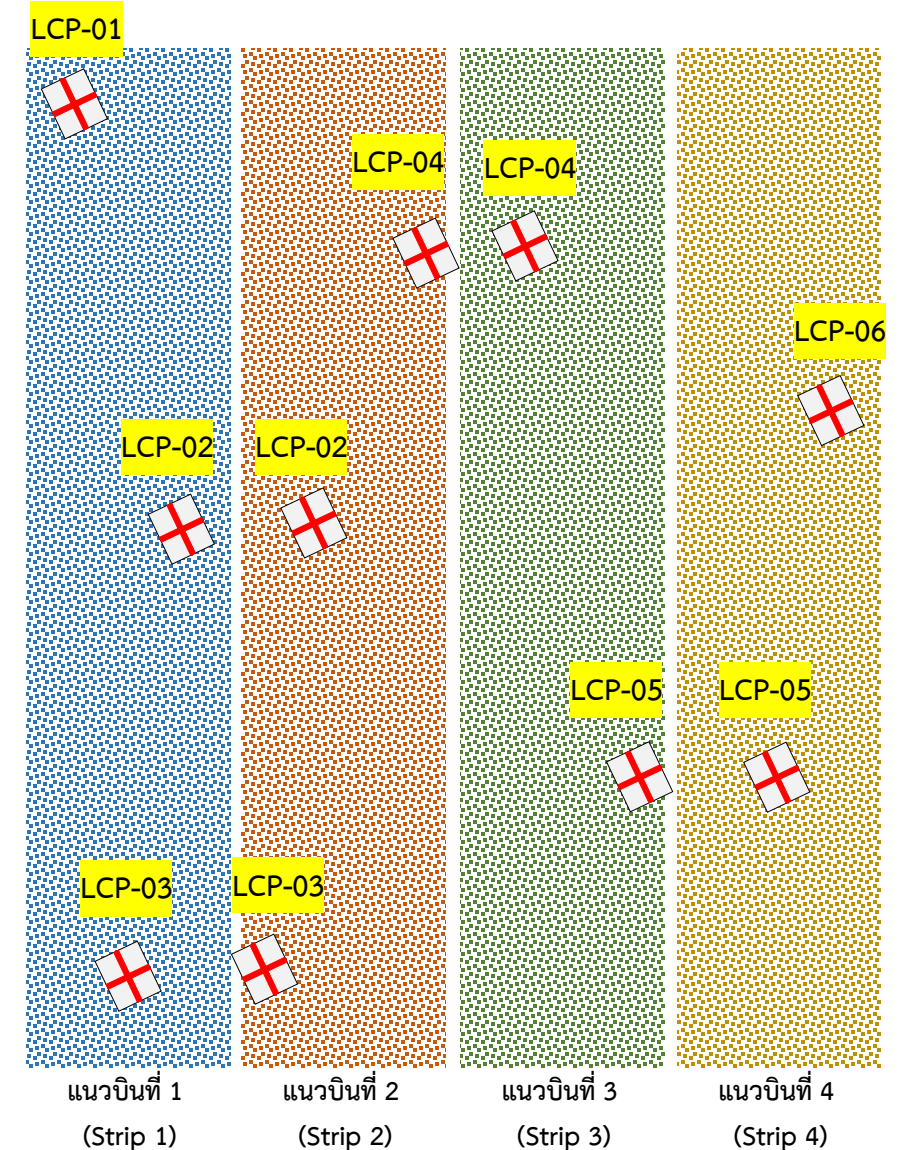


ข้อมูลที่ต้องนำเข้า

- ชุดข้อมูลพอยต์คลาวด์ในแต่ละแนวบิน (Strip) ในรูปแบบไฟล์ .las
- ไฟล์ในรูปแบบ .yaml ที่บันทึกพารามิเตอร์ต่าง ๆ ของเป้าควบคุมไลดาร์ ชื่อไฟล์ PARAMETER_LCP.yaml (./CACHE/PARAMETER_LCP.yaml)

ข้อมูลที่ได้หลังจากประมวลผล

- ไฟล์ในรูปแบบ .gpkg ที่บันทึกข้อมูลตำแหน่งของเป้าควบคุมไลดาร์ ชื่อไฟล์ PARAMETER_LCP_RESULT.gpkg (./CACHE/PARAMETER_LCP_RESULT.gpkg)
- ไฟล์ในรูปแบบ .csv ที่บันทึกข้อมูลตำแหน่งของเป้าควบคุมไลดาร์เปรียบเทียบกับค่าพิกัดจากวิธีการรังวัด RTK ชื่อไฟล์ PARAMETER_LCP_RESULT.csv (./CACHE/PARAMETER_LCP_RESULT.csv)
- ไฟล์รูปภาพในรูปแบบ .SVG ที่บันทึกพอยต์คลาวด์ของเป้าควบคุม ระบายของเป้าที่ได้จากการประมวลผลด้วยซอฟต์แวร์ ชื่อไฟล์ ชื่อจุด_แนวบินที่ปรากฏ.svg (./CACHE/ชื่อจุด_แนวบินที่ปรากฏ.svg)



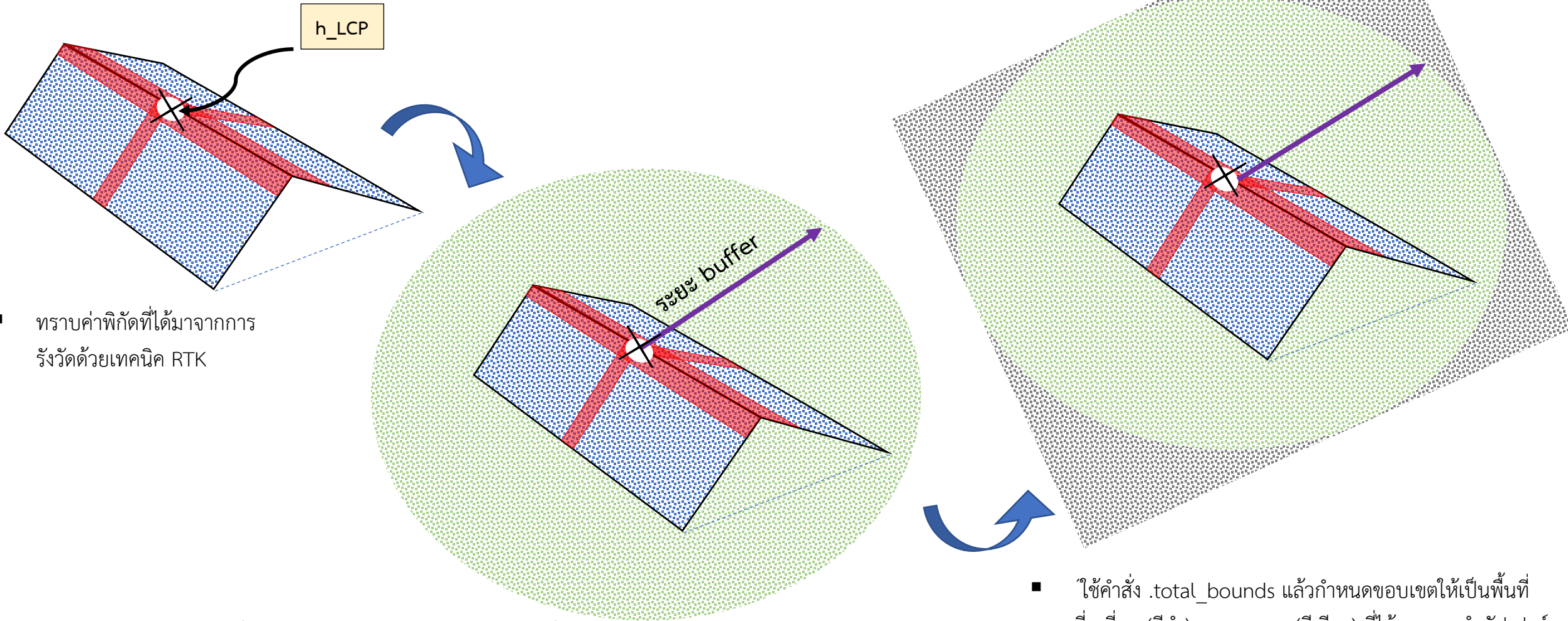
ข้อมูลที่ต้องนำเข้า

- ชุดข้อมูลพอยต์คลาวด์ในแต่ละแนวบิน (Strip) ในรูปแบบไฟล์ .las
- ไฟล์ในรูปแบบ .yaml ที่บันทึกพารามิเตอร์ต่าง ๆ ของเป้าควบคุมไลดาร์
ชื่อไฟล์ PARAMETER_LCP.yaml (./CACHE/PARAMETER_LCP.yaml)

```

VERSION : "0.3"
# configuration and data file for lidar control plane
#
# LCP defintion
BASE : 1.100      # m separation of the two planes from field
WIDTH : 0.65      # m of the future board
LENGTH : 1.220    # m of the future board
#
#
BUFF_RIDGE : 1.2   # %buffer from actual LCP size
BUFF_LFRT : [0.1,0.8] # %buffer to the left&right of LCP ridge line
#
# Lidar sensor response via "pyransac3d"
MINPOINTS : 100
THRESH : 0.05 # meter
MAXITER : 1000
FLIGHT_LINE :
  PointCloud/OneFilePerStrip/Mapper+_CU Sandbox-20230124-180345-F001-S004.las:
    - [ LCP-27,717136.305,1606298.496,0.421,18.3 ]
    - [ LCP-22,717231.352,1606163.019,-0.663,16.6 ]
    - [ LCP-17,717335.724,1606035.136,1.051,10.6 ]
    - [ LCP-12,717389.87,1605865.322,3.341,22.0 ]
    - [ LCP-07,717476.276,1605722.976,2.762,12.1 ]
  PointCloud/OneFilePerStrip/Mapper+_CU Sandbox-20230124-180345-F001-S005.las:
    - [ LCP-27,717136.305,1606298.496,0.421,18.3 ]
    - [ LCP-22,717231.352,1606163.019,-0.663,16.6 ]
    - [ LCP-17,717335.724,1606035.136,1.051,10.6 ]

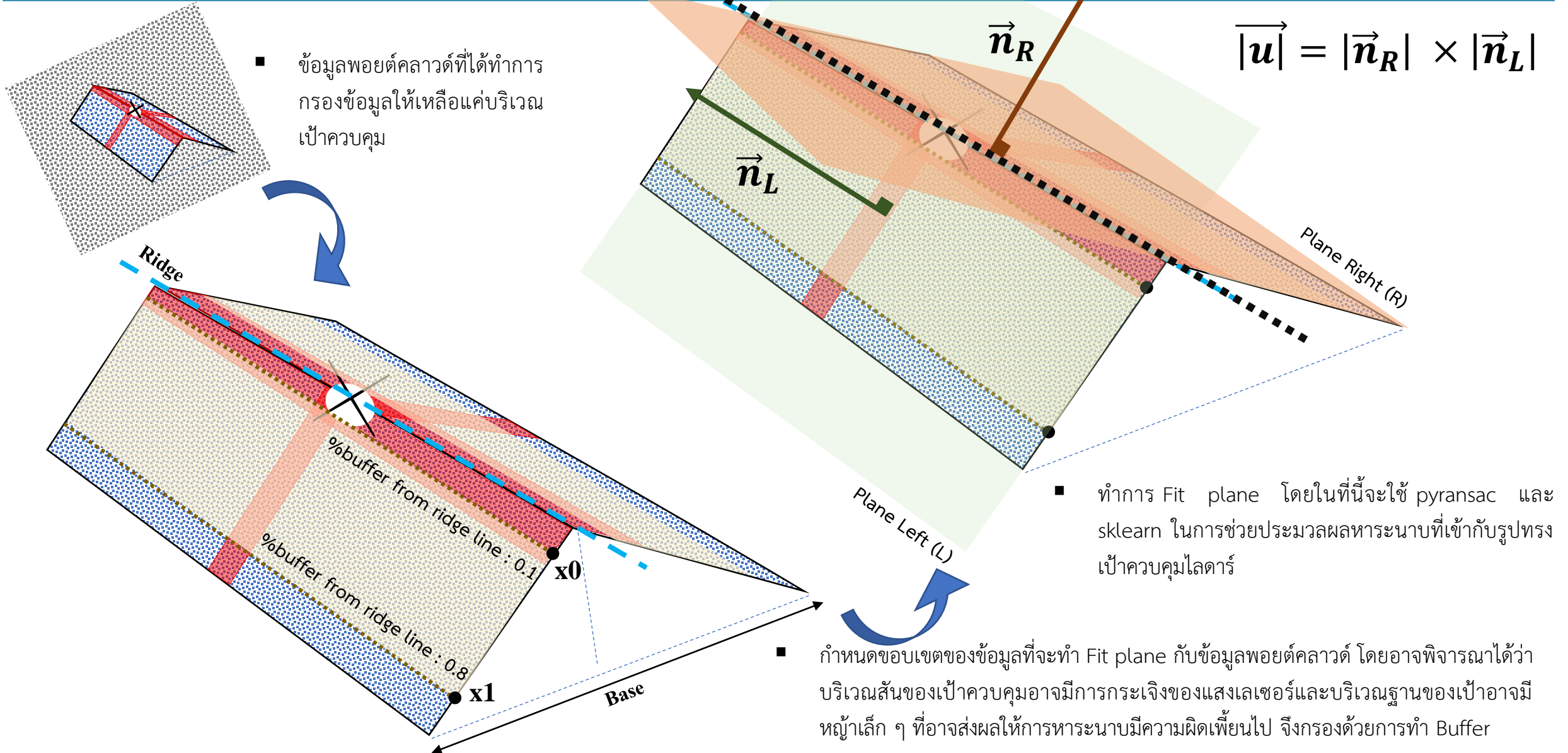
```

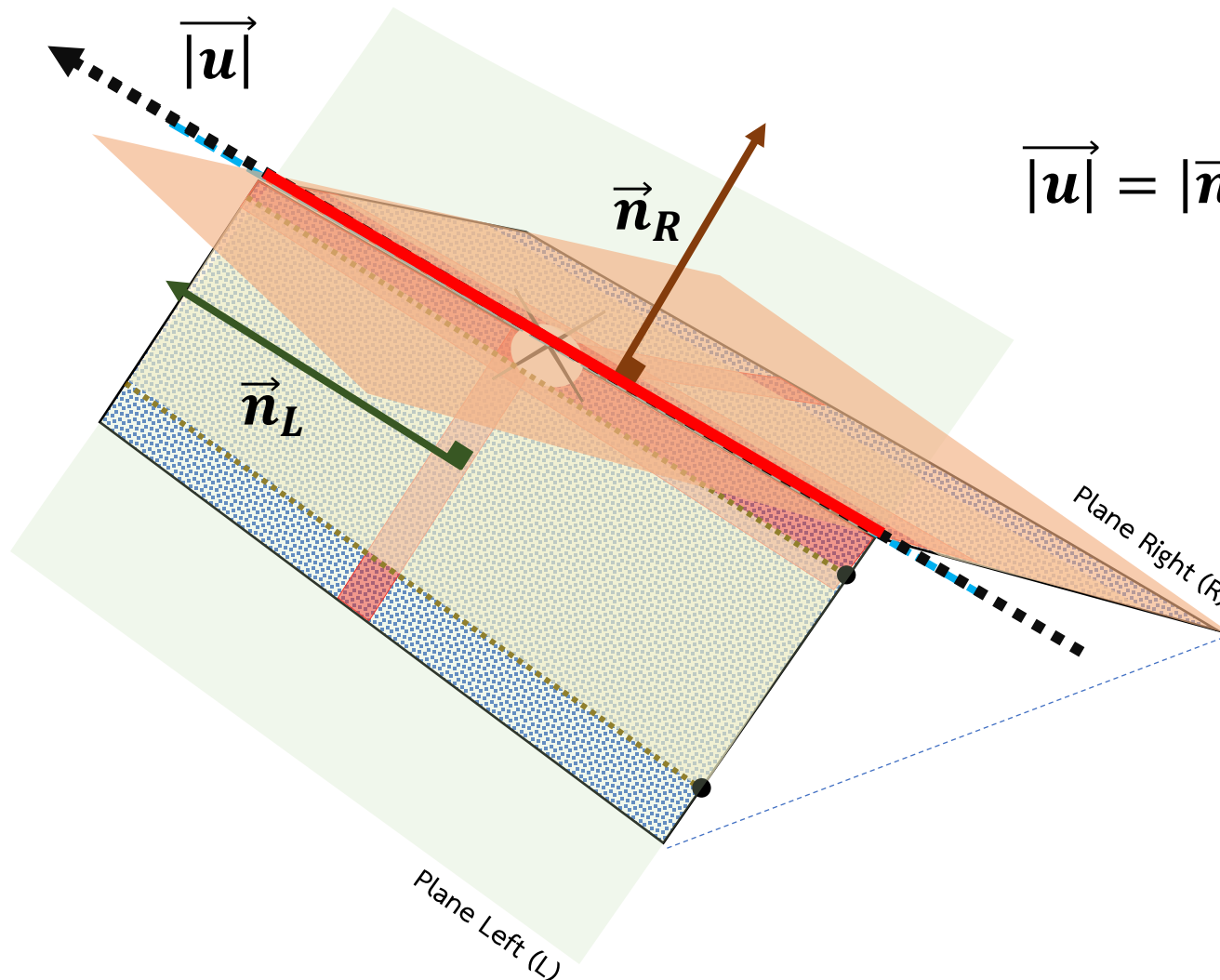


- ทราบค่าพิกัดที่ได้มาจากการ
รังวัดด้วยเทคนิค RTK

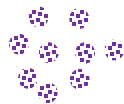
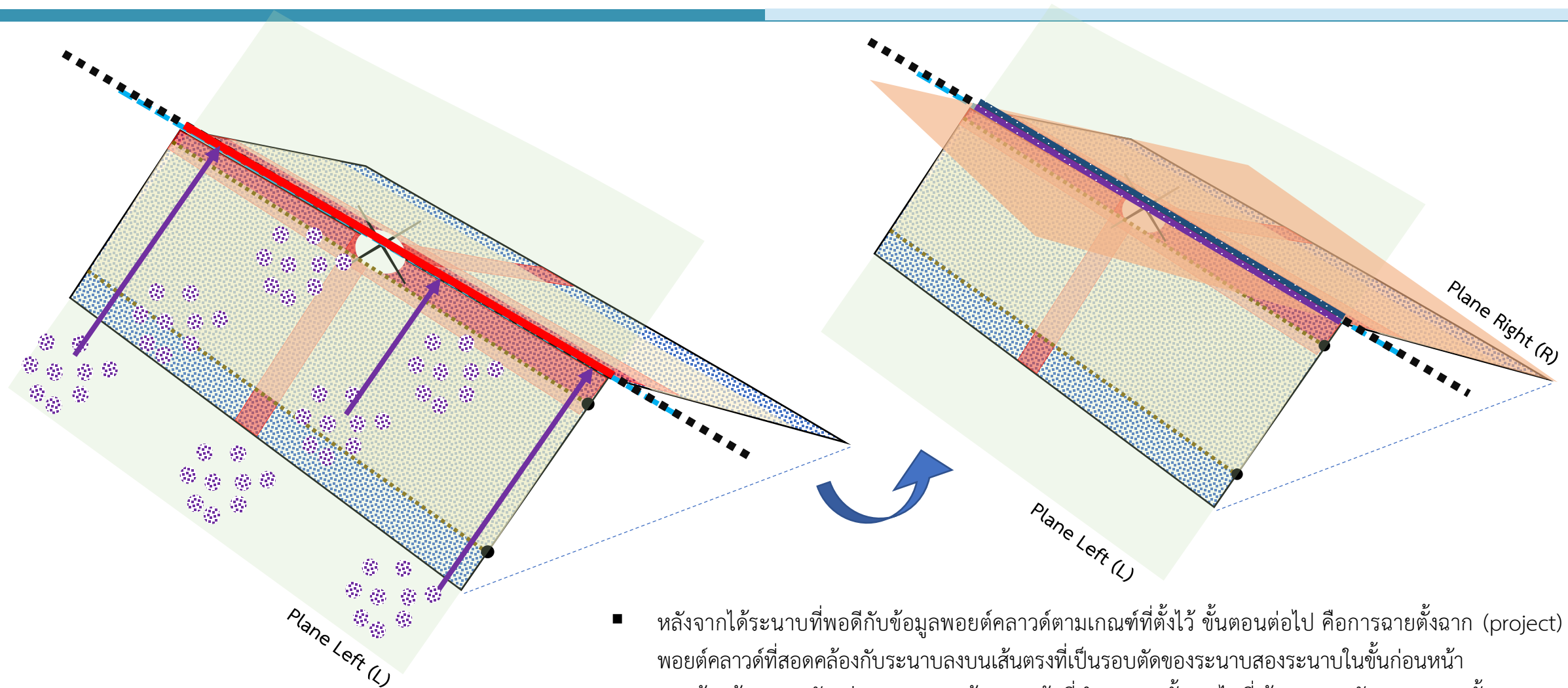
- สร้าง Buffer จากค่าพิกัดที่ได้มาจากการรังวัดด้วยเทคนิค RTK เพื่อเป็นการลดปริมาณข้อมูลพอดียคลาวด์ให้เหลือแค่บริเวณ
เป้าควบคุม ทำให้จัดการข้อมูลสะดวกและรวดเร็วขึ้น โดยระยะของบัฟเฟอร์ควรจะครอบคลุมขนาดของเป้าควบคุม (สีเขียว) ใน
ซอฟต์แวร์จะคำนวณจาก 200% ของขนาดเป้าควบคุมที่ได้ทำการกรอกไว้ในส่วนของพารามิเตอร์ในไฟล์ .yaml

- ใช้คำสั่ง `.total_bounds` แล้วกำหนดขอบเขตให้เป็นพื้นที่
สี่เหลี่ยม (สีดำ) แทนวงกลม (สีเขียว) ที่ได้จากการทำบัฟเฟอร์





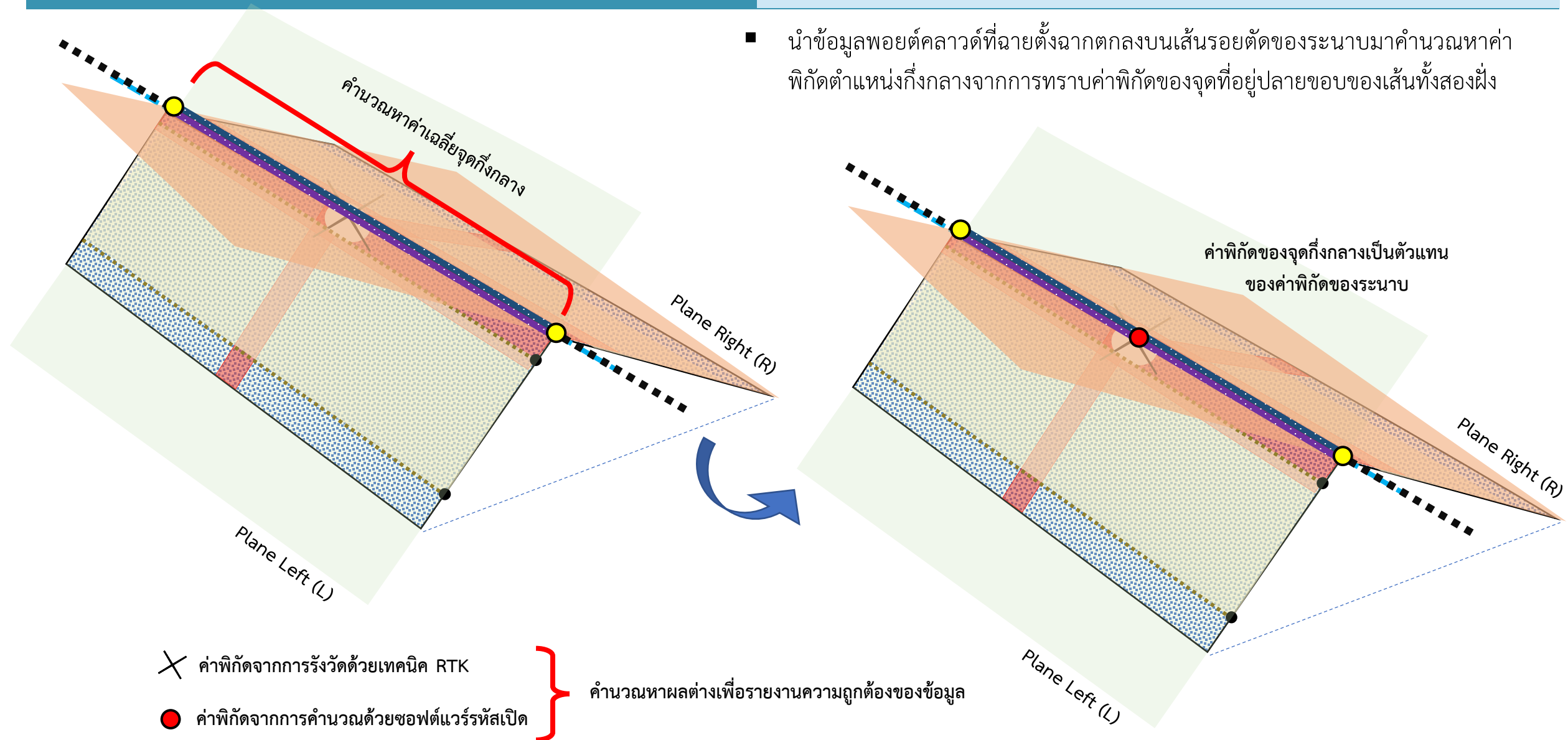
- จากการหาระนาบที่จะมา fit กับเป้าควบคุม ในทางคณิตศาสตร์แปลความหมายได้ว่า ระนาบ 2 ระนาบเมื่อระนาบทั้งสองตัดกัน รอยตัดที่เกิดขึ้นจะเป็นเส้นตรงเส้นหนึ่งที่มีขนาดของความยาวเป็นอนันต์ สอดคล้องกับระนาบที่มีความยาวและความกว้างเป็นอนันต์
- ซึ่งรอยตัดของเส้นนี้ หากพิจารณาตามลักษณะทางเรขาคณิตของเป้าควบคุมจะพบว่ารอยตัดของระนาบคือ สันหลังคาของเป้าควบคุมโลดาร์ และนี่คือสิ่งที่ซอฟต์แวร์จะช่วยประมวลผลเพื่อหาเส้นรอยตัดนี้ก่อนจะไปประมวลผลในขั้นถัดไปเพื่อให้ได้ค่าพิกัดของเป้าควบคุม



กลุ่มข้อมูลพอยต์คลาว์ที่ตกบนระนาบ

- หลังจากได้ระนาบที่พอดีกับข้อมูลพอยต์คลาว์ตามเกณฑ์ที่ตั้งไว้ ขั้นตอนต่อไป คือการฉายตั้งฉาก (project) กลุ่มพอยต์คลาว์ที่สอดคล้องกับระนาบลงบนเส้นตรงที่เป็นรอบตัดของระนาบสองระนาบในชั้นก่อนหน้า
- ภาพด้านซ้ายแสดงตัวอย่างของระนาบซ้ายของเป่าที่ทำการฉายตั้งฉากไปที่เส้นตรงรอยตัดของระนาบทั้งสอง
- ภาพด้านขวาแสดงผลลัพธ์หลังจากฉายตั้งฉากกลุ่มข้อมูลพอยต์คลาว์ที่เข้ากับระนาบลงบนเส้นตรง โดยสีม่วงเป็นตัวแทนของระนาบซ้าย และสีน้ำเงินเป็นตัวแทนของระนาบขวา ลงบนเส้นตรงที่เป็นรอบตัดในที่นี้คือ เส้นสีแดง

- นำข้อมูลพอยต์คลาวด์ที่ฉายตั้งฉากตกลงบนเส้นรอยตัดของระนาบมาคำนวณหาค่าพิกัดตำแหน่งกึ่งกลางจากการหาค่าพิกัดของจุดที่อยู่ปลายขอบของเส้นทั้งสองฝั่ง

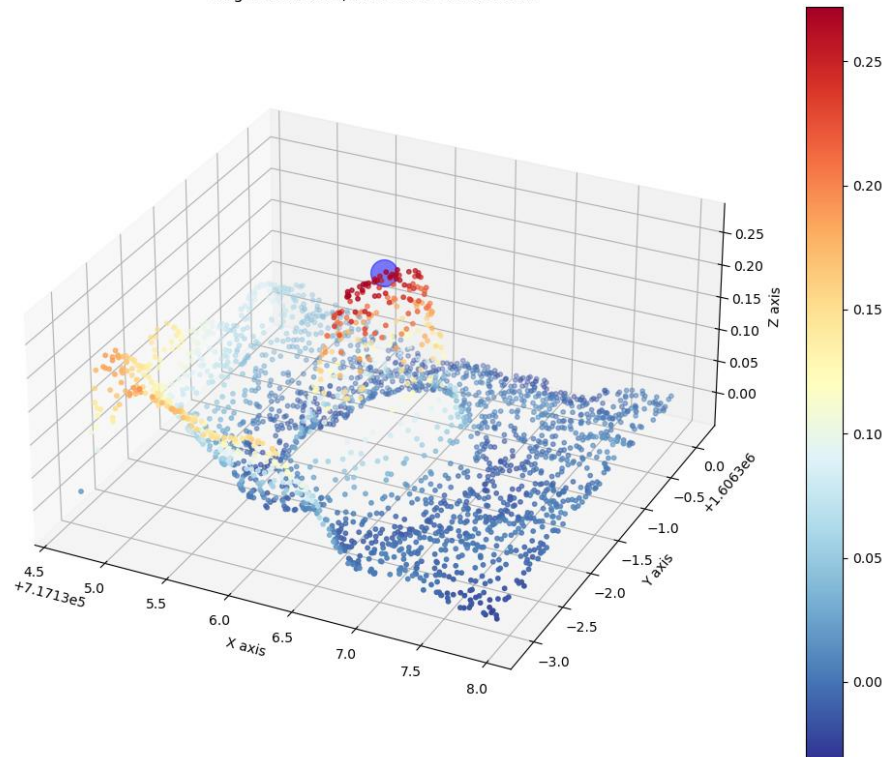


```

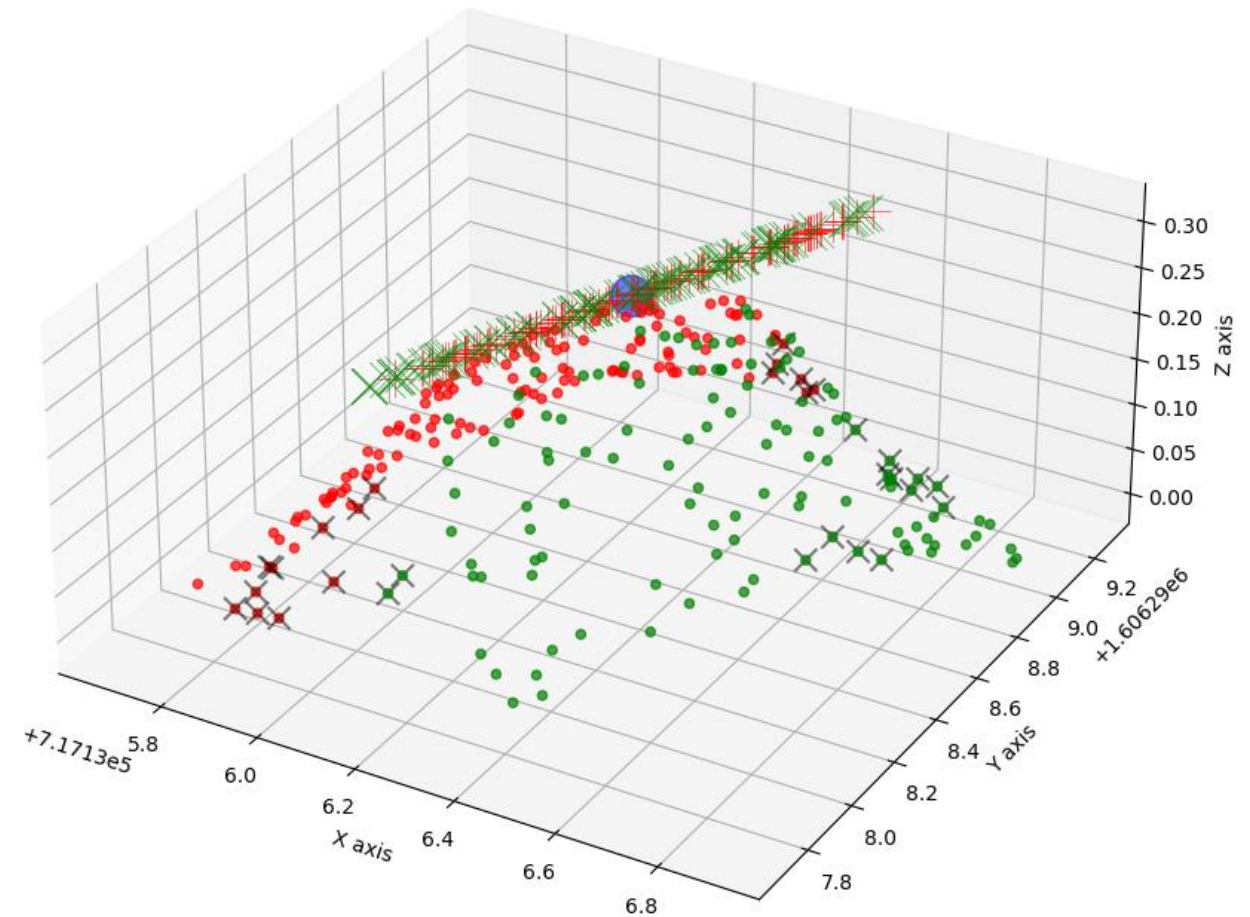
===== LCP : LCP-27 =====
Reading point cloud by flight-line : "Las_File/AA450-001.las"...
Target circle size (meter) : 3.3 x 3.3
Point cloud on target circle : 2,413 (224.0 pnt/sqm)
Fit plane "L" outliers : 11.8% (16/136)
Fit plane "R" outliers : 12.0% (14/117)
LCP-27 : L ridge length = 1.389 m, az = 15.2 deg, slope=-0.03 m
LCP-27 : R ridge length = 1.437 m, az = 15.2 deg, slope=-0.04 m
LCP-27 : LR ridge length = 1.458 m, az = 15.2 deg, slope=-0.04 m
Input LCP-27 L=1.22 : 717,136.305, 1,606,298.496, 0.421 m., AZ:18.3 deg
Best estimate LCP-27 : 717,136.333, 1,606,298.495, 0.299 m
Plotting CACHE/LCP-27_AA450-001.svg...

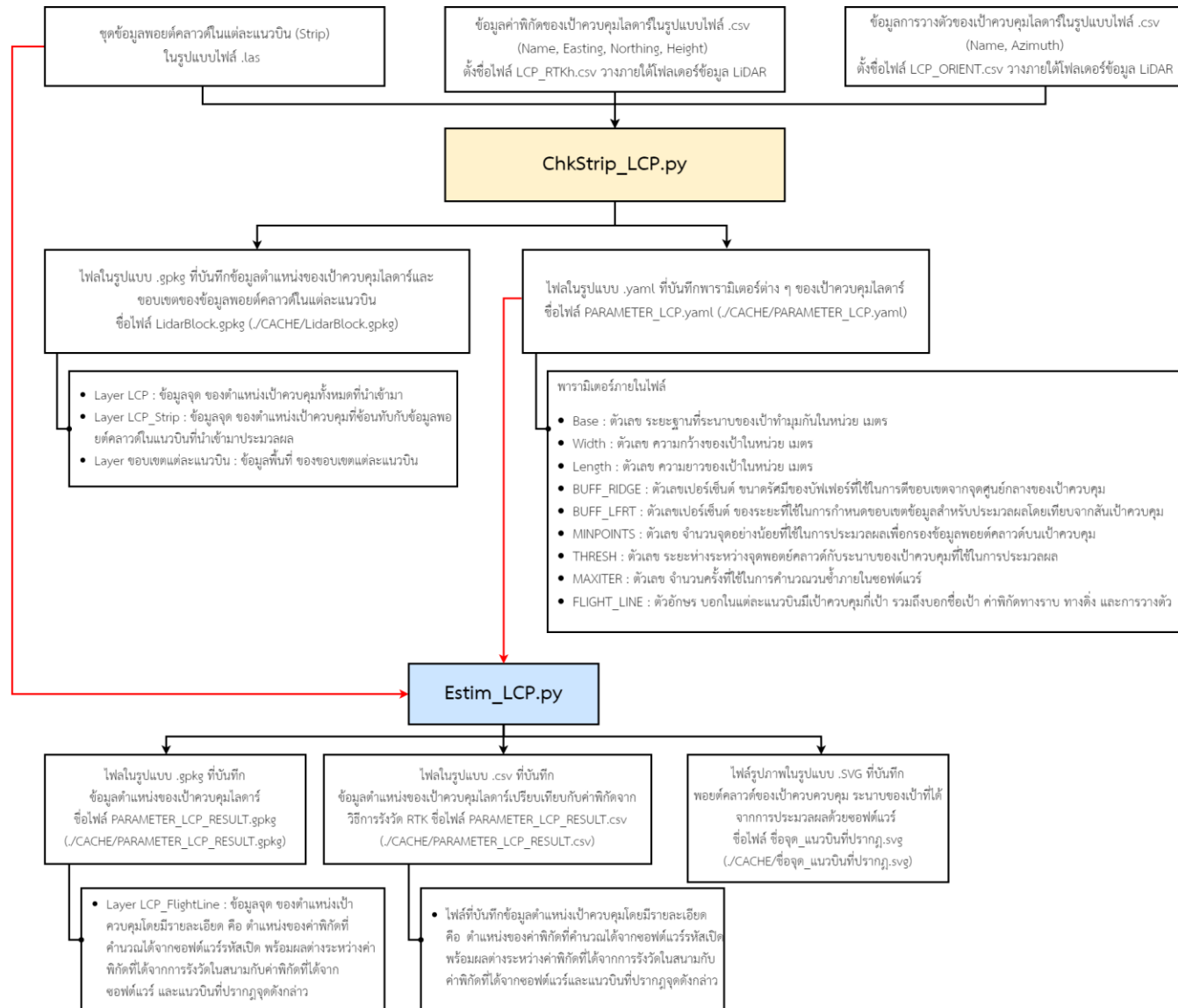
```

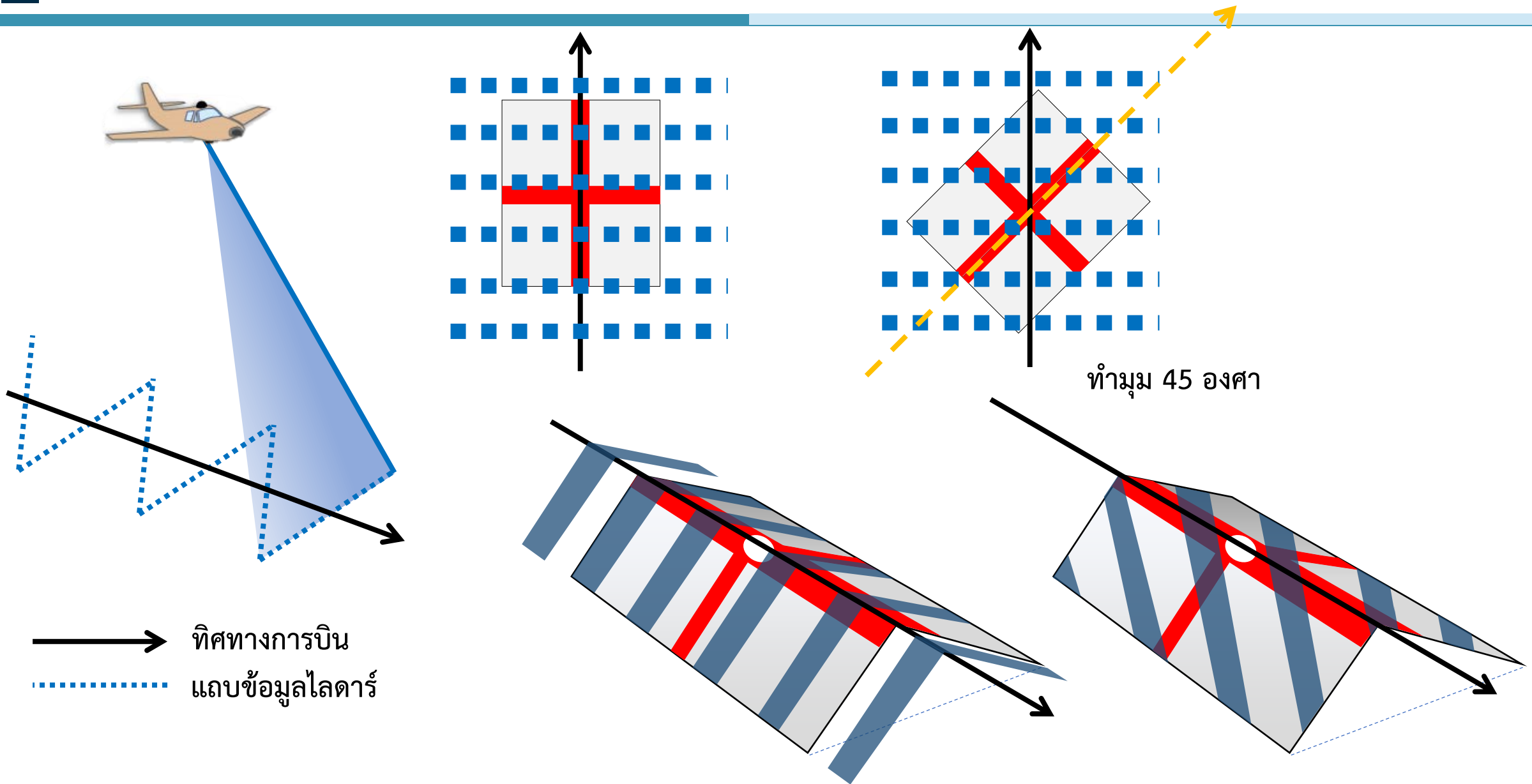
Target 717136.3,1606298.5 Circle 2.0m

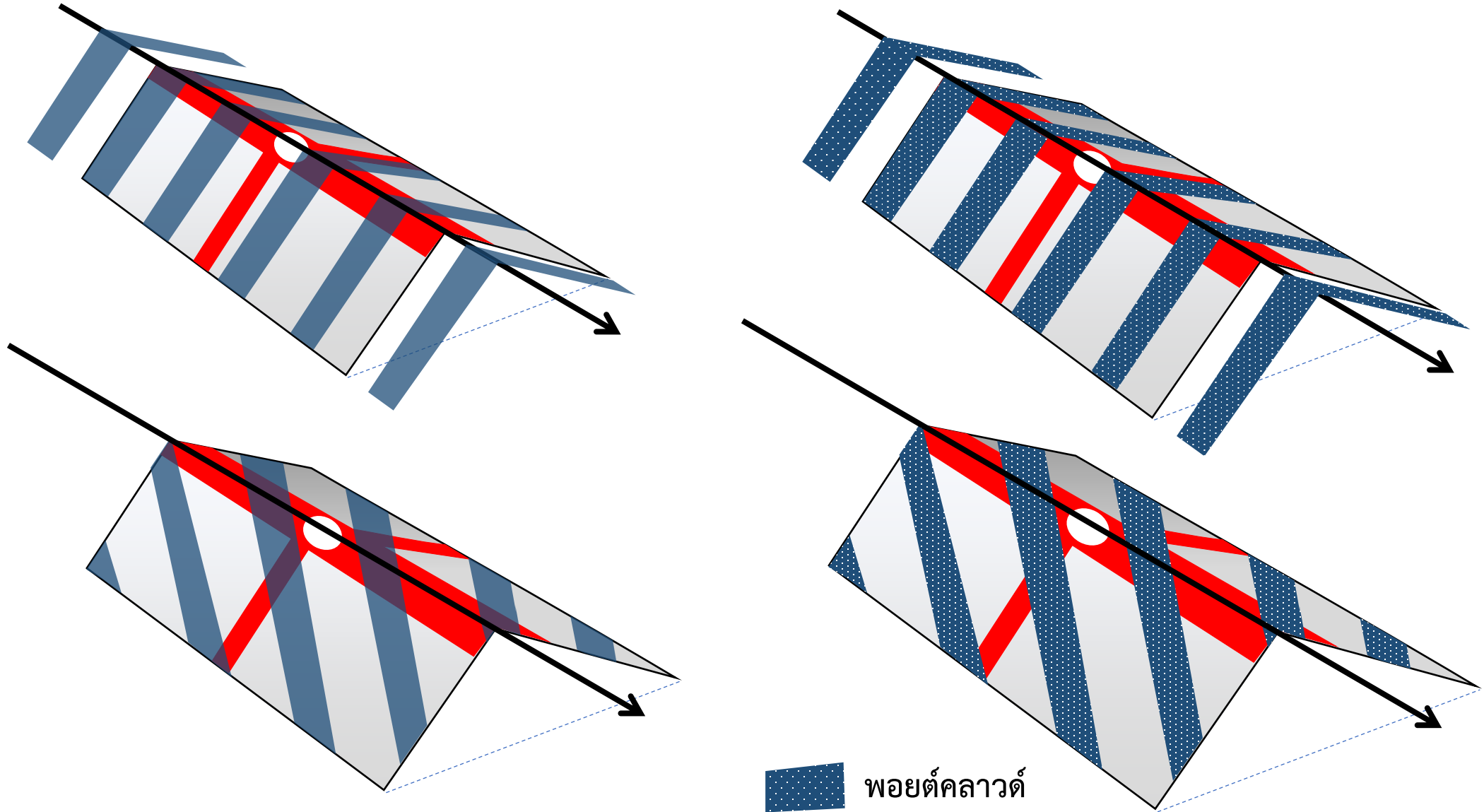


LCP-27@Las_File/AA450-001.las











Thank You