Making LAS an Open Standard (with compression)

Martin Isenburg, rapidlasso GmbH

No “universal” Point Standard

airborne (and mobile) LiDAR

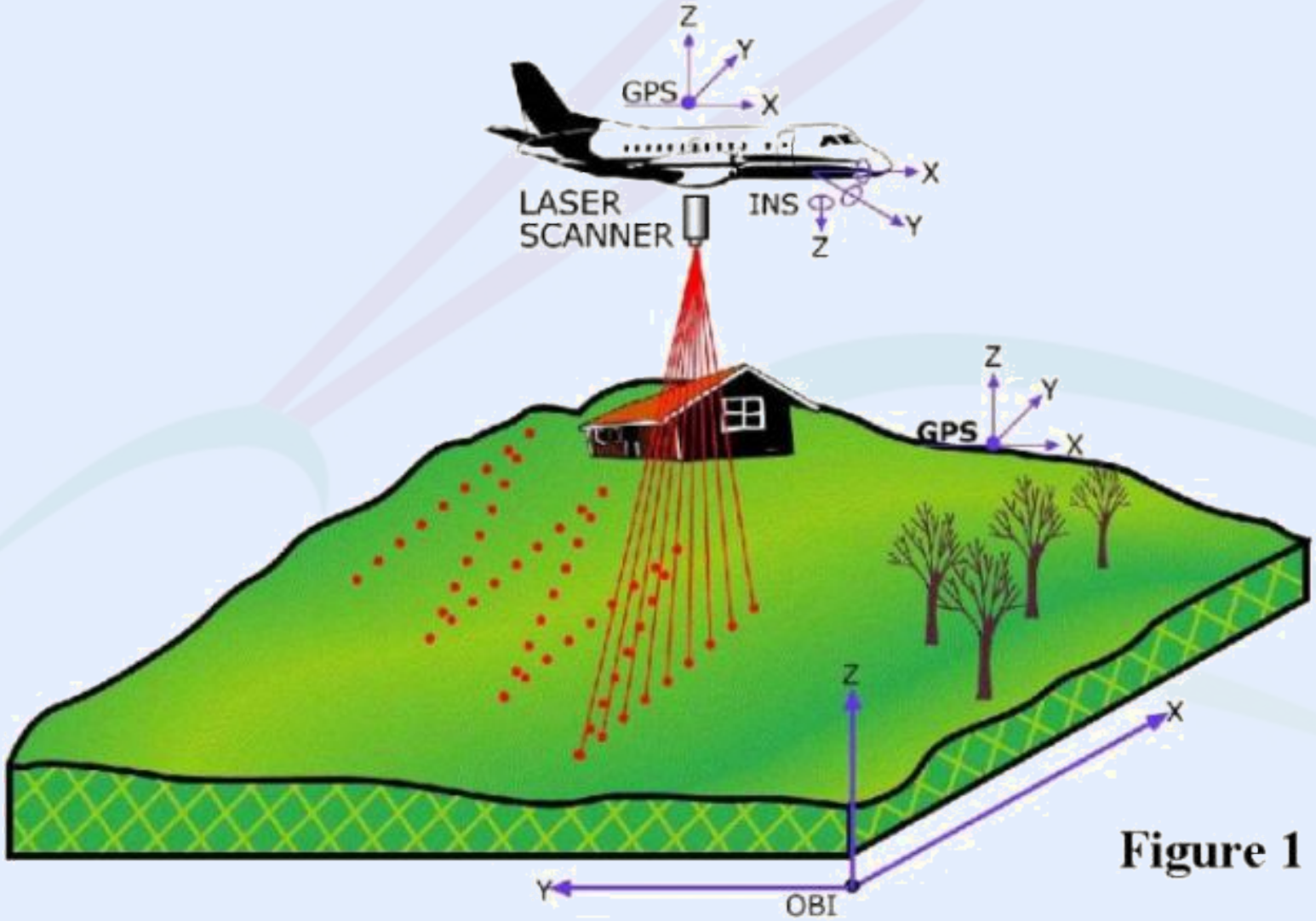
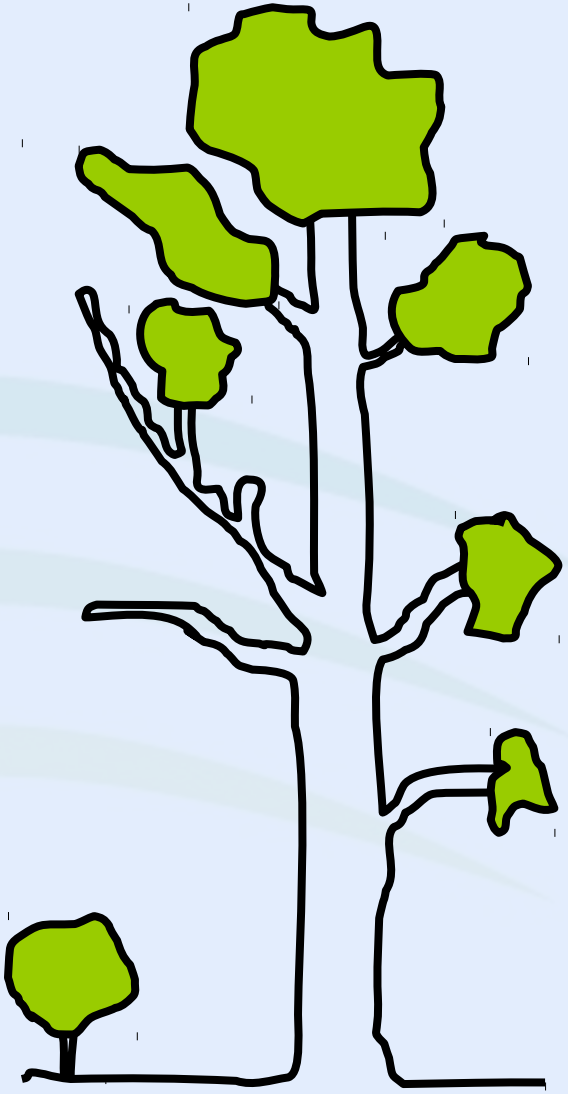
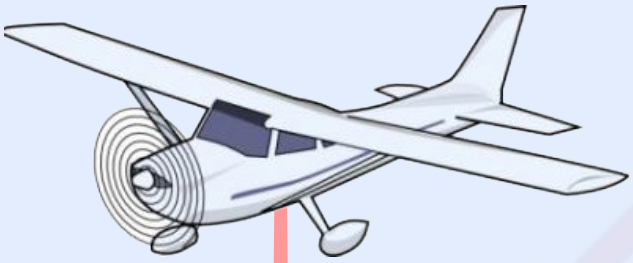


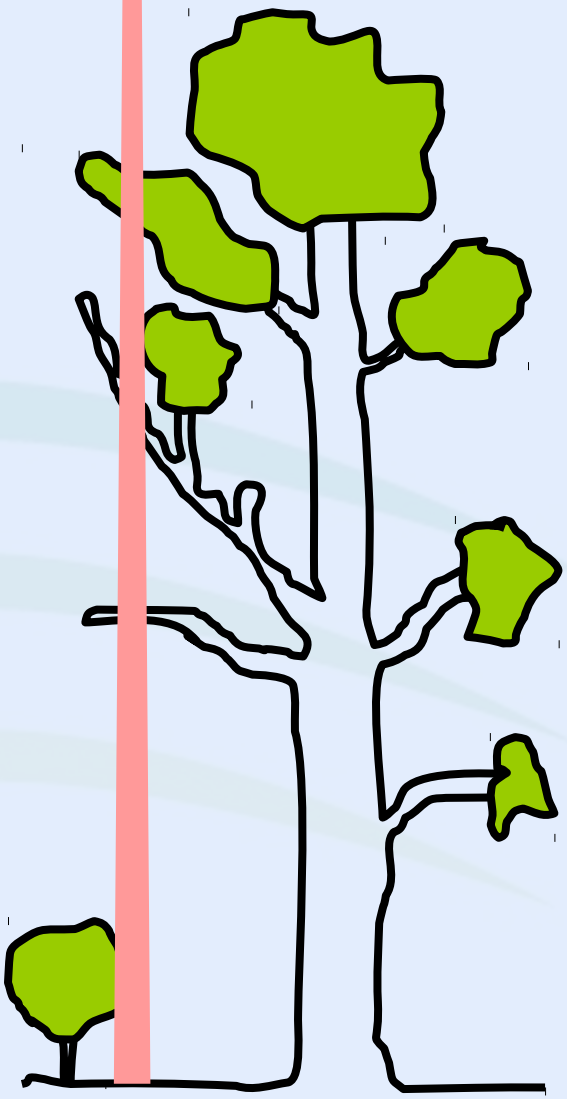
Figure 1

LiDAR Details

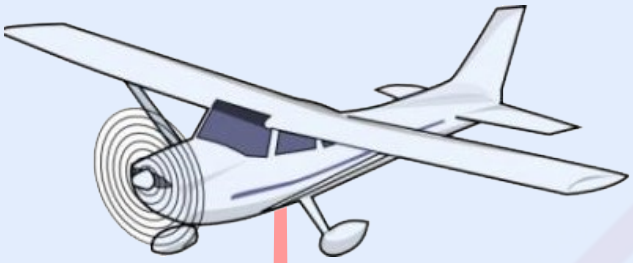


LiDAR Details



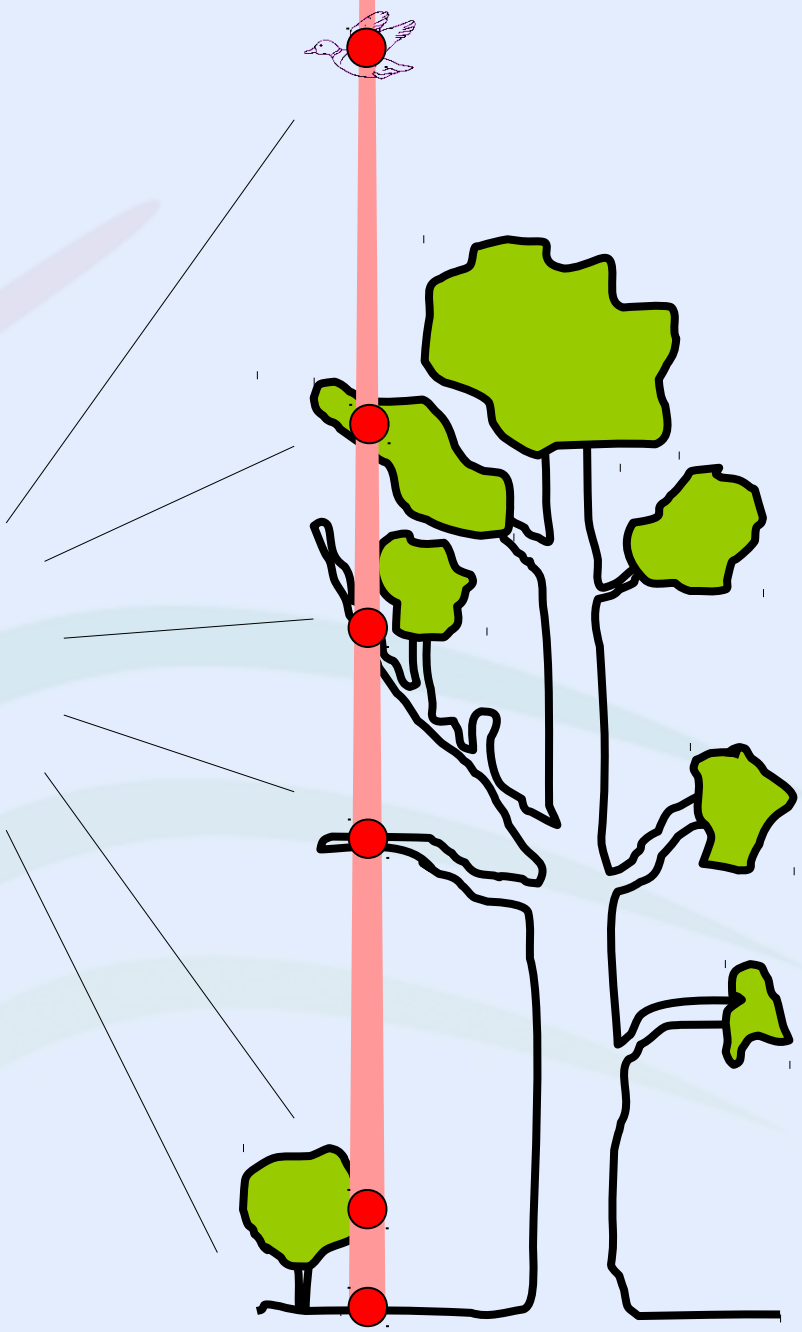


LiDAR Details



one laser shot
can produce
multiple
discrete returns:

$x_1\ y_1\ z_1$ (1^{st} of 6)
$x_2\ y_2\ z_2$ (2^{nd} of 6)
$x_3\ y_3\ z_3$ (3^{rd} of 6)
$x_4\ y_4\ z_4$ (4^{th} of 6)
$x_5\ y_5\ z_5$ (5^{th} of 6)
$x_6\ y_6\ z_6$ (6^{th} of 6)



The LAS format

small header . . . followed by many million point records

Item	Format	Size	Required
File Signature ("LASF")	char[4]	4 bytes	*
(1.1) File Source ID	unsigned short	2 bytes	*
(1.1) Reserved	unsigned short	2 bytes	
(1.1) Project ID - GUID data 1	unsigned long	4 bytes	
(1.1) Project ID - GUID data 2	unsigned short	2 byte	
(1.1) Project ID - GUID data 3	unsigned short	2 byte	
(1.1) Project ID - GUID data 4	unsigned char[8]	8 bytes	
Version Major	unsigned char	1 byte	*
Version Minor	unsigned char	1 byte	*
(1.1) System Identifier	char[32]	32 bytes	*
Generating Software	char[32]	32 bytes	*
(1.1) File Creation Day of Year	unsigned short	2 bytes	
(1.1) File Creation Year	unsigned short	2 bytes	
Header Size	unsigned short	2 bytes	*
Offset to point data	unsigned long	4 bytes	*
Number of variable length records	unsigned long	4 bytes	*
Point Data Format ID (0-99 for spec)	unsigned char	1 byte	*
Point Data Record Length	unsigned short	2 bytes	*
Number of point records	unsigned long	4 bytes	*
Number of points by return	unsigned long[5]	20 bytes	*
X scale factor	double	8 bytes	*
Y scale factor	double	8 bytes	*
Z scale factor	double	8 bytes	*
X offset	double	8 bytes	*
Y offset	double	8 bytes	*
Z offset	double	8 bytes	*
Max X	double	8 bytes	*
Min X	double	8 bytes	*
Max Y	double	8 bytes	*
Min Y	double	8 bytes	*
Max Z	double	8 bytes	*
Min Z	double	8 bytes	*

http://lasformat.org

POINT DATA RECORD FORMAT 0:

20 bytes

Item	Format	Size	Required
X	long	4 bytes	*
Y	long	4 bytes	*
Z	long	4 bytes	*
Intensity	unsigned short	2 bytes	
Return Number	3 bits (bits 0, 1, 2)	3 bits	*
Number of Returns (given pulse)	3 bits (bits 3, 4, 5)	3 bits	*
Scan Direction Flag	1 bit (bit 6)	1 bit	*
Edge of Flight Line	1 bit (bit 7)	1 bit	*
(1.1) Classification	unsigned char	1 byte	*
(1.1) `Scan Angle Rank (-90 to +90) – Left side	char	1 byte	*
(1.1) User Data	unsigned char	1 byte	
(1.1) Point Source ID	unsigned short	2 bytes	*

POINT DATA RECORD FORMAT 1:

28 bytes

Item	Format	Size	Required
X	long	4 bytes	*
Y	long	4 bytes	*
Z	long	4 bytes	*
Intensity	unsigned short	2 bytes	
Return Number	3 bits	3 bits	*
Number of Returns (given pulse)	3 bits	3 bits	*
Scan Direction Flag	1 bit	1 bit	*
Edge of Flight Line	1 bit	1 bit	*
(1.1) Classification	unsigned char	1 byte	*
Scan Angle Rank (-90 to +90) – Left side	unsigned char	1 byte	*
(1.1) User Data	unsigned char	1 byte	
(1.1) Point Source ID	unsigned short	2 bytes	*
GPS Time	double	8 bytes	*

POINT DATA RECORD FORMAT 3:

34 bytes

Item	Format	Size	Required
X	long	4 bytes	*
Y	long	4 bytes	*
Z	long	4 bytes	*
Intensity	unsigned short	2 bytes	
Return Number	3 bits	3 bits	*
Number of Returns (given pulse)	3 bits	3 bits	*
Scan Direction Flag	1 bit	1 bit	*
Edge of Flight Line	1 bit	1 bit	*
(1.1) Classification	unsigned char	1 byte	*
Scan Angle Rank (-90 to +90) – Left side	unsigned char	1 byte	*
(1.1) User Data	unsigned char	1 byte	
(1.1) Point Source ID	unsigned short	2 bytes	*
GPS Time	double	8 bytes	*
Red	unsigned short	2 bytes	*
Green	unsigned short	2 bytes	*
Blue	unsigned short	2 bytes	*

Dutch National LiDAR (AHN2)

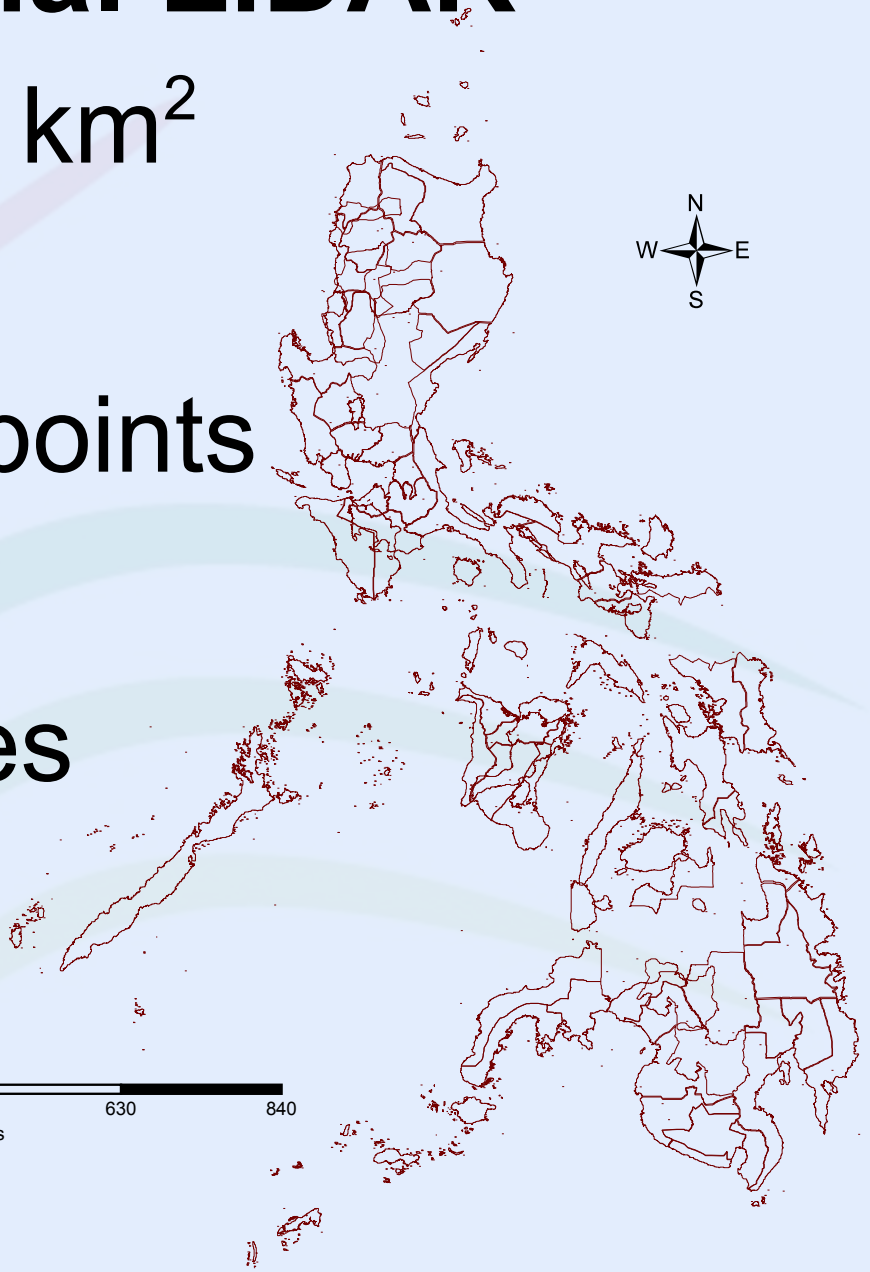
- land area: 41,526 km^2
- density: 8 to 10 shots / m^2
- total: ~ 400,000,000,000 points

* 28 bytes/point = 11.2 TB

Philippine National LiDAR

- land area: 298,170 km^2
- 2 – 4 shots per m^2
- approx. 1.2 trillion points

  ~ 33.6 TB
- often multiple copies need to be stored



LASzip (or LAZ)

LASzip

http://laszip.org

- LAZ = lossless compressed LAS
- fastest encoding / decoding
- highest compression
- used across TB of data
- integrates with LAX spatial indexing
- winner of innovation award 2012
- open source
- LGPL-licensed

} **FREE !!!**

LASzip

http://laszip.org

- LAZ = lossless compressed LAS
- fastest encoding / decoding
- h
- u
- i
- v
- open source
- LGPL-licensed

search for “LASzip” on

http://youtube.com

FREE !!!

Software with native LAZ support

- QT Modeler (7.1.6 and up) by Applied Imagery
- ERDAS IMAGINE (14.1 and up) by Hexagon Geospatial
- FUSION (3.40 and up) by Bob McGaughey of the USDA
- Global Mapper (13.1 and up) by Blue Marble Geo
- Trimble RealWorks (8.1 and up) by Trimble Navigation Limited
- ENVI LiDAR (5.1 and up) by Exelis VIS
- FME (2012 and up) by Safe Software
- TopoDOT by Certainty3D
- Pointools by Bentley Systems
- Pix4uav by Pix4D
- CloudCompare by Daniel Girardeau-Montaut
- SURE by nframes
- LAStools by rapidlasso - fast tools to catch reality
- RiProcess by RIEGL LMS GmbH
- ZEB1 by 3D Laser Mapping
- OCAD (11.4.0 and up) by OCAD Inc.
- Gexcel R3 by Gexcel
- Voyager (1.3 and up) by Voyager GIS
- Scanopy by imagination
- ScanLook by LiDAR USA
- GRASS GIS (7.0 and up) by Open Source Geospatial Foundation (OSGeo)
- OPALS by TU Vienna
- EspaEngine by ESPA Systems
- ReportGen (2.9.0 and up), by PDF3D
- OrbitGIS by Orbit
- K2Vi by AAM Group
- LiS by LASERDATA
- Geoverse by euclideon
- PointCloudViz by mirage

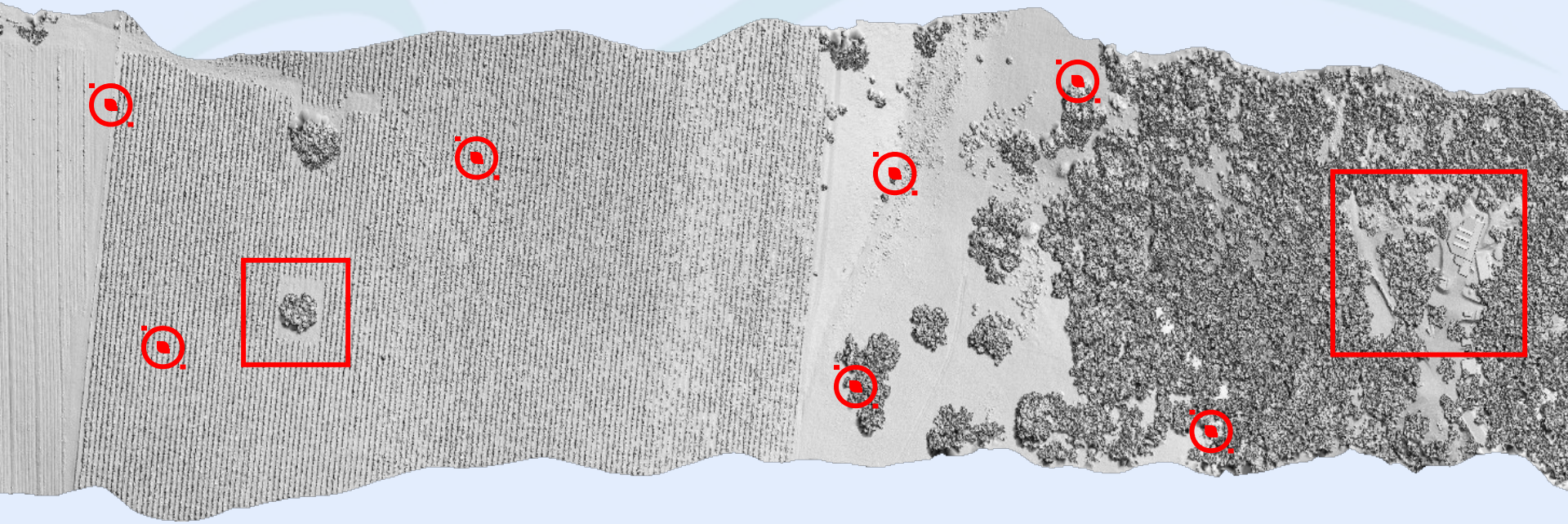
Download LAZ data

- Sonoma County LiDAR from SonomaVegMap Project
- WMS served LiDAR by Government of La Rioja in Spain
- Puget Sound LiDAR Consortium
  - 2007: sumpter
  - 2009: douglasco, snohoriver, umpqua, wenas, wenatchee
  - 2011: kittitas, quinault, rattlesnake
  - 2012: chehalis, hoh, jefferson_clallam, quinault, upper_naches
- open LiDAR data strategy of the National Land Survey of Finland
- nationwide open LiDAR data of the Netherlands
- Digital Coast LiDAR by NOAA (+ how to download)
  - Batch Download Folders: 12, 13, 15, 16, 17, 18, 19, 20, 23, 24, 25, 26, 27, 28, 29, 31, 32, 33, 34, 35, 36, 37, 39, 40, 41, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 62, 63, 64, 65, 66, 67, 68, 69, 71, 76, 77, 78, 79, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 111, 112, 114, 115, 116, 117, 501, 502, 503, 504, 505, 506, 507, 508, 509, 510, 511, 512, 513, 514, 515, 516, 517, 518, 519, 520, 521, 522, 523, 524, 525, 526, 527, 528, 529, 530, 531, 532, 533, 534, 535, 536, 537, 538, 539, 540, 541, 542, 543, 544, 545, 546, 547, 548, 549, 550, 551, 552, 553, 554, 555, 556, 557, 558, 559, 560, 561, 562, 563, 564, 565, 566, 567, 568, 569, 570, 571, 572, 573, 574, 575, 576, 577, 578, 579, 581, 582, 583, 584, 586, 587, 588, 1061, 1063, 1064, 1065, 1066, 1069, 1070, 1071, 1072, 1073, 1076, 1077, 1078, 1079, 1080, 1117, 1118, 1119, 1120, 1121, 1122, 1123, 1124, 1125, 1132, 1133, 1158, 1159, 1170, 1171, 1172, 1174, 1175, 1176, 1178, 1179, 1198, 1199, 1381, 1382, 1389, 1390, 1391, 1392, 1393, 1397, 1398, 1399, 1403, 1404, 1406, 1407, 1408
- GRAFCAN LiDAR of the Canary Islands

LASindex (or LAX)

Spatial Indexing

- efficient access to subset of points
  - query area-of-interest
  - sample elevation at some position
  - get data from neighbor tiles



Simple Extension: LAX files

- access LiDAR in original place and format (e.g. LAS, LAZ, SHP, BIN ...)
  - create tiny LAX files with the same file name that end in *.lax
  - size around 0.1 – 0.01 % of LiDAR
  - presence accelerates read queries
- assumptions / opportunities
  - data should be spatially coherent
  - resort points into z-order if needed

Simple Extension: LAX files

- access LiDAR in original place and format (e.g. LAS, LAZ, SHP, BIN ...)

search for "LASindex" on

http://youtube.com

- a
  - data should be spatially coherent
  - resort points into z-order if needed

Optimized LAS (or zLAS)

Mash-up of LAZ + LAX + sort

- also lossless compressed LAS
- also fast encoding / decoding
- also high compression
- also spatial indexing
  - but stored in one file
  - also optional sorting (“z-order”)
- additional statistics and histograms
- maybe selective decoding … ? … ?

Unique Opportunity: LAS 1.4

New Point Types 6 - 10

- represent “natural break”
- opportunity: improve & add features
  - selective decompression
  - rewritable classifications & flags
  - add point attributes without rewrite
  - <your idea>
- prevent format fragmentation
  - add whatever else “folks” need

5 Step Plan

Step 1: Make LAS a Standard

- current: LAS is “de-facto” standard
  - successful open format
  - ASPRS holds specification's copyright
  - overseen by LAS Working Group
    - no transparent processes
    - no protocols or “going-on-record”
- future: LAS becomes real standard
  - reorganization of (active) LWG team

Step 2: Make LAZ a Standard

- current: LAZ is “de-facto” standard
  - successful open format
  - Martin maintains open source code
  - guided by community via “LAS room”
    - sort of a “voluntary consensus”
    - google groups and github records
- future: LAZ becomes real standard
  - documenting LAZ (for point types 0 - 5)

Step 3: LAS 1.4 Compressor

- current: no rush, let us get this right
  - NOAA-sponsored compatibility-mode
  - open “call for input” since 2014
  - anticipated features
    - selective decompression
    - partial rewrite (flags & classifications)
- future: joint format **with** all “folks”
  - exploit existing “LAS room” process

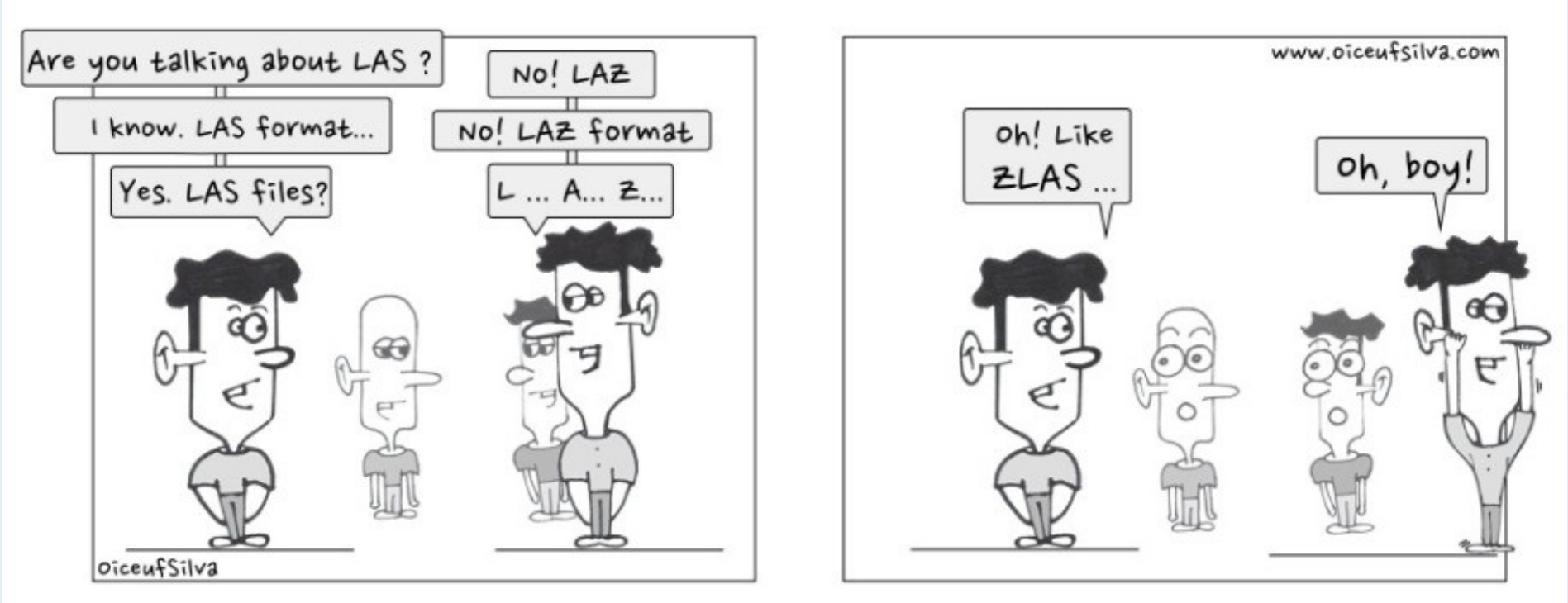
Step 4: Make zLAZ a Standard

- current: not much LAS 1.4 content
  - both LAZ and zLAS can handle it
  - compatibility-mode “buys us time”
- future: work together, test, donate
  - seize opportunity of “natural break”
  - avoid quick solution and “format war”
  - exploit “LAS room” process to finalize
  - let community test just like in the past
  - follow steps (1) & (2) to standardize

Step 5: Pick an Organization

- current: LAS Working Group
  - guided industry to one open format
  - big “Thank You!” for achievement
  - success has out-grown capacity
- future: same folks but re-organized
  - OGC but very open to public (github?)
  - maybe ASTM like E57
  - maybe ISPRS … ? maybe … ?

Let us avoid two near-identical flavors of “optimized LAZ” … (-:



Thank You.