VENTSPILS AUGSTSKOLA

INFORMĀCIJAS TEHNOLOĢIJU FAKULTĀTE

Projekta atskaite kursā "IS vai pētniecisko projektu izstrāde"

Automatizēta ēku plakņu noteikšana, izmantojot LIDAR datu apstrādi

Autors: Ventspils Augstskolas
Informācijas tehnoloģiju fakultātes
maģistra studiju programmas
"Datorzinātnes"
2. kursa students
Roberts Ķēniņš

Zinātniskā vadītājā: Ph.D. Linda Gulbe

Ventspils

2021

Saturs

Ievads 3
1. Programmatūra, pētāmais apgabals un izmantotie dati 4
1.1. Izmantotā programmatūra 4
1.2. Pētāmais apgabals un izmantotie dati. 4
2. Metodoloģija 6
2.1. Daudzstūru iezīmēšana 6
2.2. NDSM izgriešana no daudzstūra. 7
2.3. 3D modeļa ģenerēšana 9
2.4. CesiumJS 10
3. Rezultāti 11
Secinājumi 13
Izmantotie avoti 14

Ievads

Ņemot vērā to, ka pareiza pilsētas 3D karte var būt ļoti noderīga pilsētas plānošanā, dažādu notikumu vai procesu modelēšanā, kā arī tūrisma nolūkos, ēku plakņu noteikšana ir ļoti aktuāla problēma. Manuāla 3D ēku modeļu veidošana (pat ja, iespējams, detalizētāka) ir tik laikietilpīga, ka plašām teritorijām tas vienkārši nav praktiski iespējams bez milzīga darbaspēka. Daudz praktiskāk ir attālināti iegūt datus, izmantojot aerolāzerskenēšanu (LIDAR - LIght Detection And Ranging) un pielietojot uz tiem konkrētus algoritmus un metodes iegūt derīgu informāciju automātiskai 3D modeļu izveidei plašai teritorijai. Modeļiem vajadzētu būt arī pēc iespējas mazāk kļūdainiem un pēc iespējas pilnīgākiem, jo katram modelim manuāli veikt labojumus būtu ļoti laikietilpīgi un neproduktīvi automatizēta risinājuma gadījumā. Risinājumam būtu jābūt pietiekami robustam, tādam, kuru iespējams atkārtoti pielietot dažādās līdzīgās teritorijās, nesaskaroties ar nopietnām problēmām. Risinājuma izpildes laiks nav visai būtisks kritērijs, jo ideālajā gadījumā, ja risinājums darbojas ar pietiekami augstu precizitāti, apgabala ēku plaknes tiek aprēķinātas vienu reizi un atjaunojumus nepieciešams veikt, ja fizisko ēku forma būtiski fiziski izmainās vai klāt ir nākušas jaunas ēkas vai nojauktas vecās.

Pētījuma mērķis ir daļēji, bet pēc iespējas vairāk automatizēt ēku plakņu noteikšanu, lai būtu iespējams izveidot 3D modeļus no NDSM un manuāli izveidota jumta plakņu SHP faila.

Pētījuma rezultātā tiek izstrādāti nesarežģīti, ātri ģenerējami 3D modeļi, kurus iespējams apskatīt WEB vidē.

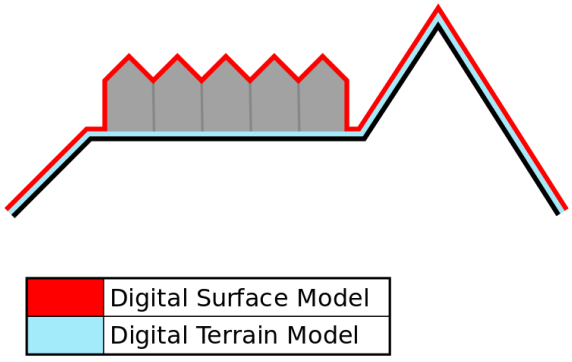
1. Programmatūra, pētāmais apgabals un izmantotie dati

1.1. Izmantotā programmatūra

- Python – augsta līmeņa programmēšanas valoda, pētījumā tiek izmantota, lai strādātu ar attēliem un SHP failiem skriptos, kuru funkcijas tiek rakstītas šajā valodā.
- QGIS – Quantum GIS (ģeogrāfiskā informācijas sistēma). Programma tiek lietota, lai izveidotu pētījumā izmantotos SHP failus un pēc nepieciešamības nolasītu NDSM vai citas vērtības.
- JavaScript – programmēšanas valoda, kura visbiežāk pielietota WEB aplikācijas. Pētījumā nepieciešama, lai strādātu ar tālāk minēto CesiumJS.
- CesiumJS - javascript bibliotēka, kas paredzēta 3D karšu un globusu izveidei un apskatei. Rīks ir pieejams bezmaksas atvērtai lietošanai ar ierobežotu funkcionalitāti, kā arī iespējams iegūt papildus bezmaksas funkcionalitāti (atļauju lietot Cesium Ion karšu serveri) personīgai lietošanai vai komerciālai par samaksu. Pētījumā CesiumJS tiek pielietots 3D modeļu precīzai attēlošanai uz kartes un to interaktīvai apskatei [2].

1.2. Pētāmais apgabals un izmantotie dati.

- TIFF formāta Ventspils NDSM (normalizēts digitālais virsmas modelis), kopējais izmērs 6416x7159 pikseļi. Katrs pikselis atbilst 1m^2. Ar šo failu tiek aprakstīts zemes virsmas augstums, iekļaujot visus objektus (ēkas, kokus, automašīnas).



Att. 1.1. Digitālā virsmas modeļa salīdzinājums ar digitāla reljefa modeli [1].

Digitālais virsmas modelis veidots pielietojot aerolāzerskenēšanu LIDAR (light detection and ranging). Tā ir tālizpētes tehnoloģija, ar kuras palīdzību tiek noteikts zemes virsmas augstums. Uz lidaparāta ir novietots skeneris, un no šī skenera tiek mērīts attālums līdz mērķim, izmantojot lāzera stara impulsu. Šajā procesā tiek analizēta no mērķa atstarotā gaisma [3].

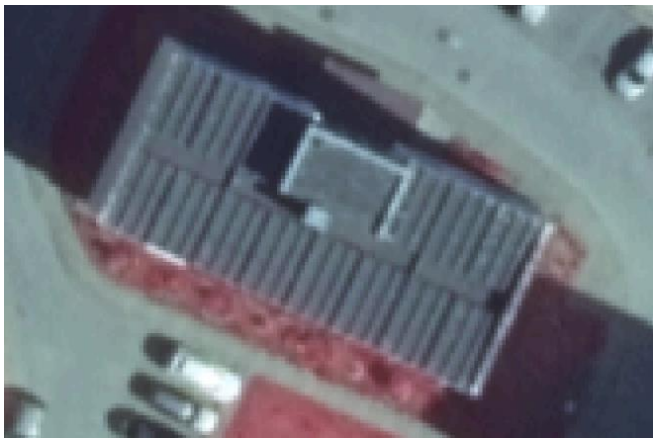
- SHP (shapefile) ir ģeotelpisks vektordatu formāts, ko pielieto ĢIS (ģeogrāfiskās informācijas sistēmās). Shapefile var saturēt vai aprakstīt punktus, līnijas un daudzstūrus, kā arī var tikt piešķirts nosaukums katrai no šīm figūrām.
- TIFF formāta Ventspils CIR (color infrared) karte izmērā 25661x28636 tiek izmantota, lai precīzāk izvēlētos SHP failos iezīmēto daudzstūru koordinātas, paši dati no šīm kartēm pētījumā izmantoti netika.
- KML (keyhole markup language) – formāts, kāds balstīts uz XML un ir paredzēts ģeogrāfisku datu uzglabāšanai. Populārs, jo to iespējams apskatīt vairākās bezmaksas aplikācijās, kā arī Google Earth. Formāts var saturēt punktus, līnijas, daudzstūrus, kā arī attēlus, HTML formatējumu un citus atribūtus. Pētījumā šajā formātā tiek saglabāti rezultējošie 3D modeļi.

2. Metodoloģija

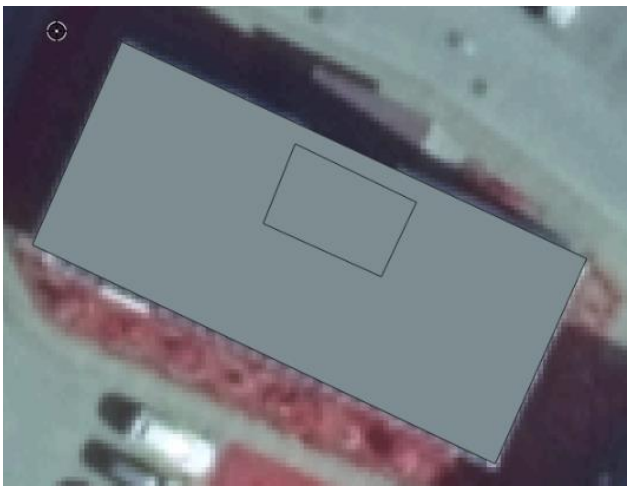
Pētījuma ietvaros tiek veikts process, kura rezultātā no divdimensionāla SHP faila, kas satur ēku jumtu plakņu daudzstūrus un NDSM tiek iegūti 3D modeļi KML formātā, kurus iespējams apskatīt WEB vai jebkurā citā vidē, kas atbalsta šo failu formātu.

2.1. Daudzstūru iezīmēšana

Ēku jumta plakņu daudzstūru iezīmēšana ir pirmais solis pētījuma procesā. Tas tiek darīts manuāli, izmantojot programmatūru QGIS, paņemot kā pamatu CIR attēlu.



Att. 2.1. CIR attēls, kurā redzams ēkas jumts.



Att. 2.2. CIR attēls, kurā redzami atbilstošie daudzstūri.

Attēlā 2.1 parādīts izvēlētais ēkas jumts. Kā redzams, tad jumts sastāv no 2 plaknēm, plaknes, kas pārklāj jumtu un plaknes, kas ir kā mazāks paaugstinājums uz otras

plaknes. Abas plaknes nepieciešams iezīmēt kā atsevišķus daudzstūrus, lai tās tiktu pareizi attēlotas 3D modelī.

Att. 2.3. Cita tipa jumta plaknes un tām atbilstošie daudzstūri.

Attēlā 2.3 attēlots cita veida jumts un tam atbilstošais SHP faila saturs. Šajā gadījumā jumts sastāv no 2 plaknēm, sīkāki objekti uz jumta netiek ņemti vērā, jo to detaļas nebūtu precīzi vai pat vispār nolasāmas NDSM failā.

2.2. NDSM izgriešana no daudzstūra.

Python ļoti vienkārši iespējams izmantot SHP failā esošos daudzstūrus, lai veiktu izgriezumu no NDSM attēla, pietiek ar 3 koda rindiņām:

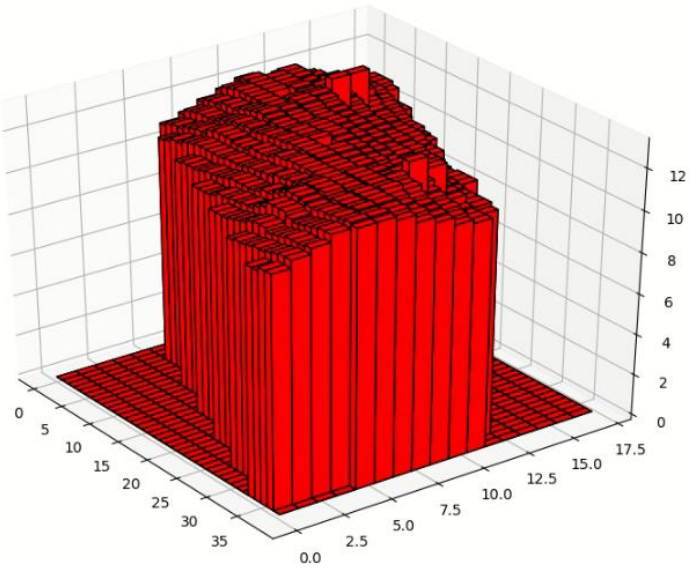
```
from osgeo import gdal, ogr
OutTile = gdal.Warp("path\\to\\output",
     "path\\to\\ndsm",
     cutlineDSName="path\\to\\shapefile",
     cropToCutline=True,
     dstNodata = 0)
OutTile = None
```

Alternatīvi šo izgriezumu var veikt arī pašā QGIS vidē, bet mērķis ir pēc iespējas automatizēt 3D modeļu izveidi, līdz ar to katrā solī ir vēlams pēc iespējas samazināt manuālas darbības, kuras nepieciešams veikt lietotājam.

Att. 2.4. Rezultējošais NDSM attēla izgriezums.

Rezultātā tiek iegūts attēlā 2.4. parādītais izgriezums. Attēlā saskatāmi daži balti pikseļi, kas ir balti (nav datu) vai arī pikseļi, kas ir tumšākas krāsas (zemāka virsma par jumtu), tomēr gadījumos, kad šādi pikseļi nesakrīt ar daudzstūru stūriem (kam pie pareizas daudzstūru zīmējuma izveides arī vajadzētu notikt) tie rezultātu negatīvi neietekmē.

Šo NDSM izgriezumu varam arī vizualizēt kā 3D grafiku, jo katra NDSM pikseļa vērtība atbilst augstumam konkrētajā punktā. Līdz ar to, ja attēlojam šo attēlu kā 3D stabiņu diagrammu (att. 2.5.), tad varam aptuveni gūt ieskatu, kādā forma būs nākamajā solī ģenerētajai ēkai.



Att. 2.5. NDSM attēls kā stabiņu diagramma.

2.3. 3D modeļa ģenerēšana

KML failu sintakse paredz to, ka lai izveidotu konkrētas ēkas 3D modeli, nepieciešams zināt katras ēkas plaknes stūru punktu koordinātas. Ņemot vērā to, ka jumta plakņu koordinātas ir lietotāja noteiktas SHP failā, tās ir iespējams nolasīt un no šīm koordinātām uzģenerēt KML failu. Ģenerēšanu var veikt ar vienkāršu skriptu, kas pa rindiņai sastāda KML failu, attiecīgajās vietās iekļaujot vērtības no SHP faila. Sākotnēji tiek iekļautas abas lietotāja iezīmētās plaknes un pēc tam tiek veidotas visas nepieciešamās sānu plaknes, izvēloties 2 blakus esošus punktus un kā nākamos 2 punktus izvēloties tās pašas koordinātas, bet nomainot augstumu uz 0 (tātad līdz ar zemi). Zemāk apskatāms KML faila izgriezums. Galvenokārt uzmanību nepieciešams pievērst tam, lai abās pusēs koordinātām būtu iekļauti visi nepieciešamie marķējumi. Katrai ēkai iespējams dot nosaukumu ar <name> marķējumu, kā arī krāsu vai caurspīdīgumu pēc nepieciešamības.

```
<?xml version="1.0" encoding="utf-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
  <Document>
    <name></name>
    <Style id="solidRed">
      <PolyStyle>
        <color>641400FF</color>
      </PolyStyle>
    </Style>
    <Placemark>
      <name>building_model1</name>
      <styleUrl>#solidRed</styleUrl>
      <MultiGeometry>
      <Polygon>
        <extrude>0</extrude>
        <altitudeMode>relativeToGround</altitudeMode>
        <outerBoundaryIs>
          <LinearRing>
            <coordinates>
              21.54927659,57.3882228,11.298963
              21.54937813,57.38821224,12.487901
              21.54945561,57.38855084,12.030616
              21.5493751,57.3885496,11.024592
            </coordinates>
          </LinearRing>
        </outerBoundaryIs>
      </Polygon>
      <Polygon>
        <extrude>0</extrude>
        <altitudeMode>relativeToGround</altitudeMode>
        <outerBoundaryIs>
          <LinearRing>
            <coordinates>
              21.54937813,57.38821224,12.487901
              21.54945561,57.38855084,12.030616
              21.5495276,57.3885396,11.11605
              21.5494476,57.3882096,11.11605
```

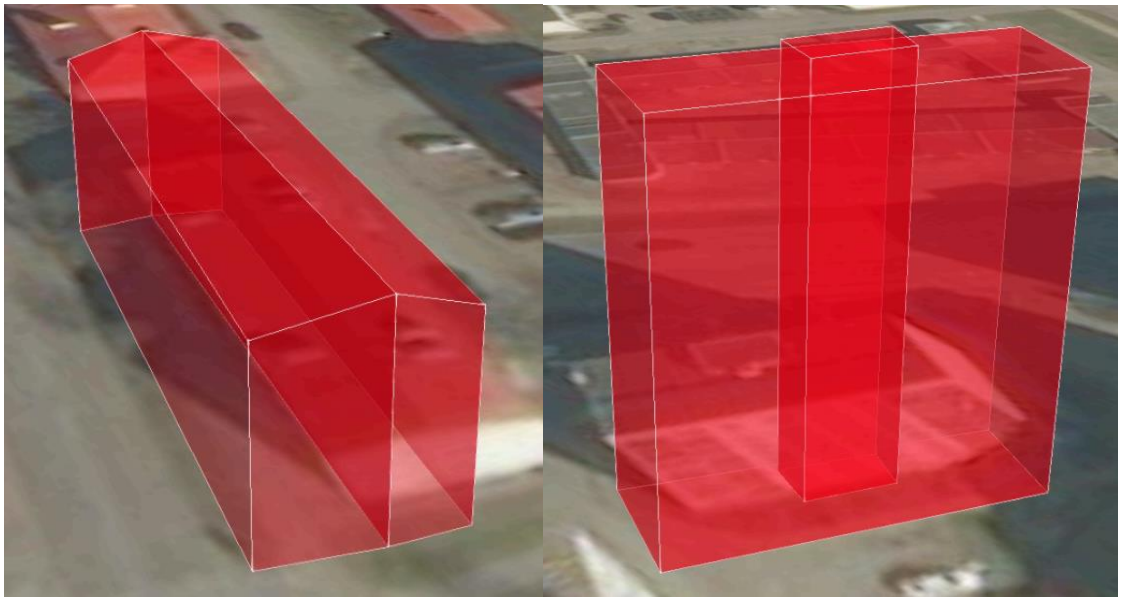
Iespējams alternatīvs un vēl ātrāks risinājums, kura rezultātā tiek iegūti mazāki faili ir nomainīt marķējumu <extrude> uz 1, kas veic dotās plaknes izstiepšanu līdz zemei, kas šajā gadījumā un ņemot vērā metodi, kādā tiek veidoti šie modeļi ir pietiekami. Gadījumos, kad nepieciešams veidot sarežģītākas ēku sienas šī metode varētu būt nepiemērota.

2.4. CesiumJS

Rezultējošie KML faili tiek pielikti CesiumJS, kur ar nelielu konfigurēšanu un izmaiņām index.html failā tos ir iespējams apskatīt. Rezultātu sadaļā redzamie attēli ir iegūti no šī pārlūka un nākotnē paredzēts automatizēti izveidot pēc iespējas vairāk 3D modeļus, lai tie būtu apskatāmi kā kopēja pilsēta.

3. Rezultāti

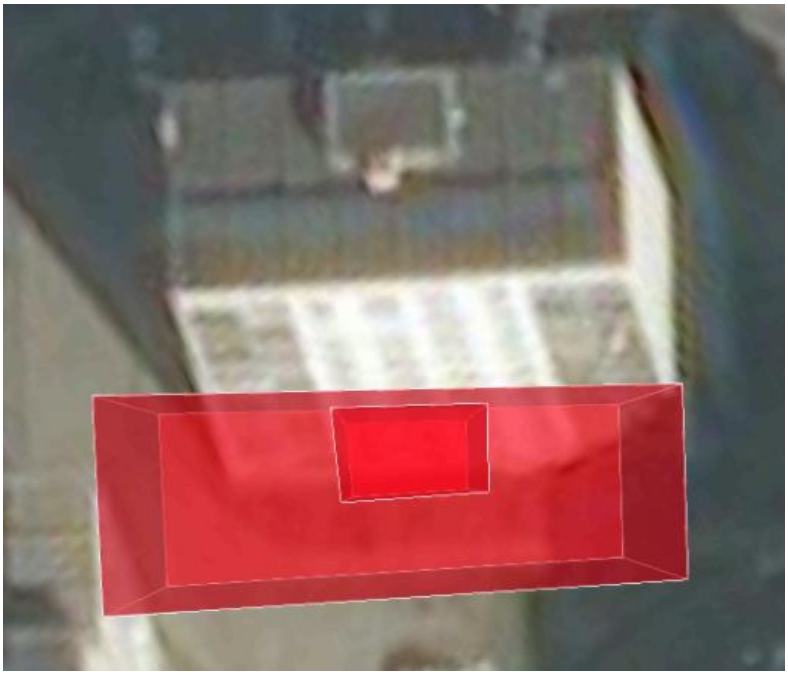
Rezultātā iegūti Cesium uzņemtie ekrānšāviņi. Kā redzams, tad ēkas pēc formas viennozīmīgi līdzinās tam, kādas tās saskatāmas dzīvē (att.3.1).



Att. 3.1. Rezultējošo 3D modeļu skats no sāna.

Attēlos 3.2. un 3.3, kuros parādīts skats uz abām ēkām no augšas, varam redzēt neprecizitātes ēku platumos un jumta konstrukcijā, kā arī redzam nobīdi uz zemes virsmas salīdzinājumā ar ēkām, kas redzamas kartē zem tām. Tomēr šo nobīdi, iespējams, ir nepamatoti ņemt vērā, jo iespējama kļūda savietojamībā starp pētījumā izmantotajām kartēm un kartēm, kuras piedāvā Cesium. Pārējās kļūdas ir radušās no lietotāja neprecīzas jumtu iezīmēšanas ar daudzstūriem.

Att. 3.2. Skats no augšas uz ēku 1.



Att. 3.3. Skats no augšas uz ēku 2.

Ēkas šajos attēlos apzināti attēlotas caurspīdīgas, lai uzskatāmāk būtu redzams process, kādā tās tiek veidotas. Kā redzam attēlos, tad modeļos pastāv iekšējas sienas, kuras nebūtu redzamas, ja ēkas būtu necaurspīdīgas. Ēkas šādi tiek veidotas, lai mazinātu aprēķinu skaitu un atļautu <extrude> marķējuma izmantošanu KML failā.

Šajā aprakstā apskatīti 2 ēku veidi, bet šādus modeļus ir iespējams izveidot salīdzinoši ļoti ātri, pieņemot, ka tiek atvēlēts laiks jumta plakņu individuālai iezīmēšanai.

Secinājumi

1. Ar pētījumā aprakstīto metodi iespējams ģenerēt ēku 3D modeļus salīdzinoši vienkārši un ātri, tomēr nepieciešama samērā precīza ievade no lietotāja.
2. Process nav pilnībā automatizēts un viennozīmīgi lielākā daļa no darba ir ēku jumtu plakņu iezīmēšana, kas lielam ēku daudzumam var prasīt pietiekami daudz laika. Tomēr šī metode vienalga ir daudz ātrāka nekā 3D modeļu manuāla izveide no sākuma.
3. Ēku modeļi nekad netiks izveidoti pilnībā precīzi, ņemot vērā ierobežoto NDSM izšķirtspēju un lietotāja kļūdu ēku jumta plakņu iezīmēšanā.
4. Procesu, iespējams, var turpināt automatizēt tālāk, automātiski iezīmējot ēku jumta plaknes bez cilvēka ievades. Šāds risinājums būtu daudz sarežģītāk izveidojams ar pietiekami labu precizitāti un noteikti nefunkcionētu ar visiem ēku jumtu tipiem, tomēr, iespējams, samazinātu darbu, kas cilvēkam jāveic manuāli, pieņemot, ka neprecīzi izveidotās ēkas nav nepieciešams pilnīgi rekonstruēt no sākuma.
5. 3D modeļi, kas tiek izveidoti ar šo metodi ir ļoti vienkārši un tikai aptuveni parāda ēkas uzbūvi. Ar šo metodi nav nekāda veida kā detalizēti attēlot, piemēram, ēku sienas, toties ja mērķis ir uzģenerēt pēc iespējas vairāk vienkāršas ēkas, tad šī metode ir ļoti noderīga. Ēkas ir pietiekami detalizētas, lai gūtu kopēju ieskatu par pilsētu vai apdzīvotu vietu, uz kuru metode tiek pielietota.

Izmantotie avoti

1. Surfaces represented by a Digital Surface Model and Digital Terrain Model by Yodin. Licensed under the Creative Commons Attribution-Share Alike 4.0 International license (https://creativecommons.org/licenses/by-sa/4.0/deed.en). https://commons.wikimedia.org/wiki/File:DTM_DSM.svg [sk. 09.02.2021.]

2. 3D Geospatial Visualization for the Web https://cesium.com/cesiumjs/ [sk. 11.02.2021.]

3. Jamie Carter, K Schmid, K Waters, L Betzhold, B Hadley, R Mataosky, and J Halleran. Lidar 101: An introduction to lidar technology, data, and applications. National Oceanic and Atmospheric Administration (NOAA) Coastal Services Center, Charleston, South Carolina, https://coast.noaa.gov/data/digitalcoast/pdf/lidar-101.pdf. [sk. 11.02.2021.]