

A TOOLKIT FOR THREE-DIMENSIONAL RECONSTRUCTION AND
VISUALIZATION OF WEATHER RADAR IMAGES

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

MUSTAFA AHMET PESEN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
GEODETIC AND GEOGRAPHICAL INFORMATION TECHNOLOGIES

JULY 2019

Approval of the thesis:

**A TOOLKIT FOR THREE-DIMENSIONAL RECONSTRUCTION AND
VISUALIZATION OF WEATHER RADAR IMAGES**

submitted by **MUSTAFA AHMET PESEN** in partial fulfillment of the requirements for the degree of **Master of Science in Geodetic and Geographical Information Technologies Department, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. Zuhâl Akyürek
Head of Department, **Geod. and Geog. Inf. Tech**

Assoc. Prof. Dr. Uğur Murat Leloğlu
Supervisor, **Geod. and Geog. Inf. Tech., METU**

Examining Committee Members:

Prof. Dr. Zuhâl Akyürek
Civil Engineering, METU

Assoc. Prof. Dr. Uğur Murat Leloğlu
Geod. and Geog. Inf. Tech., METU

Prof. Dr. Mahmut Onur Karşlıođlu
Civil Engineering, METU

Prof. Dr. Ahmet Coşar
Computer Engineering, University of Turkish Aeronautical
Association

Assist. Prof. Dr. Emre Sümer
Computer Engineering, Başkent Üniversitesi

Date: 25.07.2019

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Surname: Mustafa Ahmet Pesen

Signature :

ABSTRACT

A TOOLKIT FOR THREE-DIMENSIONAL RECONSTRUCTION AND VISUALIZATION OF WEATHER RADAR IMAGES

Pesen, Mustafa Ahmet
Master of Science, Geodetic and Geographical Information Technologies
Supervisor: Assoc. Prof. Dr. Uğur Murat Leloğlu

July 2019, 123 pages

Weather radars are among key instruments that feed numerical weather prediction models. They form two-dimensional (2D) images at various elevation angles with radii of tens of kilometers. These images are actually sections from three-dimensional (3D) time-dependent volumetric data. In modern radars, each cell can be assigned a type of hydrometeor after processing. However, the visual interpretation of these images is not easy due to the complicated viewing geometry. Nevertheless, computer graphics can make weather radar images easily observable and analyzable. In this thesis, a method and a toolkit for 3D reconstruction and visualization of weather radar images are presented.

The toolkit enables users to process and visualize NEXRAD Level III hydrometeor classification images in the form of closed-volumes and collected images are converted into volumetric display of precipitation types in following steps: Firstly, continuous and smooth density estimates are obtained for each precipitation type in these 2D images and the type with the maximum density is assigned to the corresponding cell if it is above a threshold. Secondly, the resulting image is segmented using connected components analysis. Then, the boundaries of each segment are extracted and transferred to the 3D space by using metadata.

Subsequently, the 3D polygons from different levels but belonging to the same object are matched.

Finally, a 3D wireframe structure is created from the matched 3D polygons and displayed sequentially to create 3D visual animation using computer graphics algorithms. The toolkit can help weather radar operators to better grasp the complicated weather events in 3D.

Keywords: Meteorology, Three-dimensional Modeling, Three-dimensional Visualization, Image Processing, Weather Radar, Hydrometeor Classification

ÖZ

METEOROLOJİ RADARI GÖRÜNTÜLERİNDEN ÜÇ-BOYUTLU GERİÇATIM VE GÖRSELLEŞTİRME İÇİN BİR ARAÇ TAKIMI

Pesen, Mustafa Ahmet
Yüksek Lisans, Jeodezi ve Coğrafik Bilgi Teknolojileri
Tez Danışmanı: Doç. Dr. Uğur Murat Leloğlu

Temmuz 2019, 123 sayfa

Meteoroloji radarları sayısal hava tahmin modellerini besleyen anahtar enstrümanlar arasındadır. Onlarca kilometre yarıçaplı, çeşitli yükselim açılarında iki boyutlu (2B) görüntüler oluşturur. Bu görüntüler aslında üç boyutlu (3B) zamana bağlı hacimsel verilerden kesitlerdir. Modern radarlarda, her hücreye belirli bir işlem den sonra bir tür hidrometeor atanabilir. Ancak, bu görüntülerin görsel yorumu karmaşık görüntüleme geometrisi nedeniyle kolay değildir. Yine de, bilgisayar grafikleri hava durumu radar görüntülerini kolayca gözlemlenebilir ve analiz edilebilir hale getirebilir. Bu tezde, hava durumu radar görüntüleri için 3B geriçatım ve görselleştirme yöntemi için bir araç takımı sunulmuştur.

Araç takımı kullanıcıların NEXRAD Seviye III hidrometeor sınıflandırma görüntülerini kapalı hacimler halinde işleyebilmelerini ve görselleştirebilmelerini sağlar ve toplanan görüntüler, şu adımlarda hacimsel yağış tipinin göstergesine dönüştürülür: İlk olarak, iki boyutlu (2B) görüntülerde her bir yağış tipi için sürekli ve pürüzsüz yoğunluk tahminleri elde edilir ve eğer eşik değerinin üzerinde ise, karşılık gelen hücreye maksimum yoğunluğa sahip yağış tipi atanır. İkinci olarak, elde edilen görüntü bağılı bileşen analizi kullanılarak bölütlendirilir. Ardından, her bir bölütün sınırları meta veriler kullanılarak çıkarılır ve 3B uzaya aktarılır. Daha sonra, farklı yükseklikteki ancak aynı nesneye ait 3B çokgenler eşleştirilir. Son

olarak, eşleşen 3B çokgenlerden bir 3B kafes yapısı oluşturulur ve bilgisayar grafik algoritmaları kullanılarak 3B görsel animasyon oluşturmak için sırayla görüntülenir. Araç takımı, hava durumu radar operatörlerinin karmaşık hava olaylarını 3B olarak daha iyi anlamalarına yardımcı olabilir.

Anahtar Kelimeler: Meteoroloji, Üç-boyutlu Modelleme, Üç-boyutlu Görselleştirme, Görüntü İşleme, Meteoroloji Radarı, Hidrometeor Sınıflandırma

“The greatness and infinite wisdom of the Creator will be recognized only by those who really endeavor to draw their ideas from the great book we call nature”

Justus von Liebig (1803 – 1873)

ACKNOWLEDGMENTS

First and foremost, I would like to express my deepest gratitude to my advisor, Associate Prof. Uğur Murat Leloğlu, for his continuous understanding and encouragement throughout my Master's Thesis study and research. Before he accepted me as his student, I was taking his classes. During the course breaks, he always prefers to be with his students. Most of the times, there are people surrounding him to ask various questions. There was the time when I finally caught an opportunity to ask him a question. My first question was about "triangulation". He explained this particular theory so striking that at that moment I become able to create three-dimensional systems in my mind. Throughout my entire MSc study, his guidance helped me implement my ideas on the creation of various 3-D systems. I become able to create 3-D systems with the coding languages I have never tried before. Dr. Leloğlu also trained me to grow professionally as a scientist by shaping my research methodology. He gave me opportunities to speak at conferences and present on behalf of my university. I will never forget the time I got the "Best Presentation" prize from the scientific committee in the VII. (National) Remote Sensing and GIS Symposium.

I am completely indebted to my family. I would like to thank especially to my parents and sisters for their unending love, infinite generosity and their respect to my achievements. I am sure my father also walked through this journey with me. Thanks for his friendliness. My uncles, cousins, aunts and the entire Pesen family members will always be remembered.

My sincere thanks go to my close friends Ahmet Tan Fırtına, İrem Özbay, Zafer Durmuş, Onur Arslan, Eren Altıneşik, Talat Taşçı, Taylan Özgür Elma for supporting me spiritually throughout my struggles on the thesis development process.

TABLE OF CONTENTS

ABSTRACT	v
ÖZ	vii
ACKNOWLEDGEMENTS	x
TABLE OF CONTENTS	xi
LIST OF TABLES	xv
LIST OF FIGURES	xvi
CHAPTERS	
1. INTRODUCTION	1
1.1. Radars in Meteorology	1
1.2. Problem Statement	1
1.3. Contribution of the Thesis	4
1.4. Organization of the Thesis	5
2. BACKGROUND THEORY	7
2.1. The Atmosphere	7
2.1.1. The Composition of the Atmosphere	7
2.1.2. Structure of the Atmosphere	8
2.1.3. Formation of Clouds and Precipitation	10
2.2. Weather Radars	14
2.2.1. Radar Principle	14
2.2.2. Scattering of Electromagnetic Waves in the Atmosphere	15
2.2.3. Hydrometeors and Scattering	16
2.2.4. A Brief History of Weather Radars	17

2.2.5. Weather Radar System	18
2.2.6. Types of Weather Radars	19
2.2.6.1. Stationary Radars.....	19
2.2.6.2. Mobile and Deployable Radars	20
2.2.6.3. Airborne Radars.....	21
2.2.6.4. Space-borne Radars	21
2.2.7. Radar Base Moments.....	22
2.2.7.1. Radar Reflectivity Factors	22
2.2.7.2. Mean Radial Velocity	22
2.2.7.3. Spectrum Width	23
2.2.7.4. Differential Reflectivity (ZDR)	23
2.2.7.5. Correlation Coefficient (CC)	23
2.2.7.6. Specific Differential Phase (KDP).....	24
2.2.8. NEXRAD Radars	24
2.2.8.1. History	24
2.2.8.2. Functional Characteristics of WSR-88D	25
2.2.8.3. NEXRAD Data Format.....	26
2.2.8.4. Data Access	27
2.2.8.5. The Hydrometeor Classification Algorithm (HCA)	27
2.3. 3D Reconstruction and Visualization	31
2.3.1. 3D Data Representation.....	32
2.3.2. Surface Reconstruction.....	35
2.3.2.1. Shape Matching	35
2.3.2.2. Similarity Matching	40

2.3.3. 3D Computer Graphics	41
2.4. Similar Works.....	43
3. 3D RECONSTRUCTION OF WEATHER RADAR IMAGES.....	49
3.1. Outline of the Algorithm	49
3.1.1. Development Platform.....	53
3.1.2. Graphical User Interface.....	53
3.2. Segmentation of Hydrometeor Images	54
3.2.1. Image Segmentation	54
3.2.2. Overview of the Segmentation Algorithm.....	56
3.2.3. Density Estimation.....	57
3.2.4. Thresholding	60
3.2.5. Connected Component Analysis.....	62
3.2.6. Segmentation in the GUI	63
3.3. Three-dimensional Boundary Polygons	64
3.3.1. Border Tracking.....	65
3.3.2. Transformation of Image Coordinate to Spatial Coordinate	70
3.3.3. Export of 3D Polygons in the KML Format.....	72
3.4. Matching Polygons.....	73
3.4.1. Matching Polygons from Two Successive Elevations.....	73
3.4.2. Tracking Meshes Through Elevations	78
3.5. Visualization in Three-dimension	83
3.5.1. Pointwise Matching of Polygons	83
3.5.1.1. Coarse Matching	83
3.5.1.2. Fine Matching	87

3.5.2. Triangles to Surfaces	89
3.5.2.1. Shorter Lower Segment Case	90
3.5.2.2. Shorter Upper Segment Case.....	93
3.5.3. Visualization.....	96
3.5.3.1. Lighting as a Feature of Visualization.....	96
3.5.3.2. Control Transparency	97
4. RESULTS AND DISCUSSION	99
4.1. Formation of 3D Objects in the Toolkit.....	99
4.1.1. Three-Dimensional Shapes from Two Polygons.....	99
4.1.2. Three-Dimensional Shapes from Multiple Polygons	101
4.2. GUI for the Toolkit: Meteorology3D.....	106
4.2.1. Read Raw Data and Scale Image	106
4.2.2. Segmentation in the GUI.....	108
4.2.3. Object Creation and Visualization	110
5. CONCLUSIONS	113
5.1. Broader Impact of the Toolkit.....	113
5.2. Future Direction	114
REFERENCES	117

LIST OF TABLES

TABLES

Table 2.1 Composition of pure dry air (Saha, 2008).	7
Table 2.2. WSR-88D specifications (Whiton, Smith, Bigler, & Wilk, 1998)	25
Table 2.3. Ranges of hydrometeors in the corresponding radar products (Dual-Pol Training, 2019).....	29
Table 3.1. Distance matrix of masses between two sequential elevations for label4	75
Table 3.2. Calculated distances; horizontal and vertical minima.....	76
Table 3.3. Intersection values (bold on both sides); after threshold operation, final matches are displayed.	78
Table 3.4. Tracking matches forLabel6 (Light and / or moderate rain).....	80
Table 3.5. Created objects in a resultant view	82
Table 4.1. Properties of the objects (only one label).....	103
Table 4.2. Structure of the cloud objects (all of the labels)	104

LIST OF FIGURES

FIGURES

Figure 1.1. Capabilities of hydrometeor classification (University Corporation for Atmospheric Research, 2009)	2
Figure 2.1. Structure of the atmosphere.....	9
Figure 2.2. Cloud types (Ahrens, 2011, s. 109).....	11
Figure 2.3. Representation of raindrop, cloud droplet and condensation nucleus (Ahrens, 2011, s. 128).....	12
Figure 2.4. Flow diagram of different type of snow formation (Collier, 2016, s. 64)	13
Figure 2.5. The working principle of Radar	15
Figure 2.6. Electromagnetic waves' interaction with matter shown as a unit of an approximate circumference (Nathanson, Reilly, & Cohen, 1991, s. 152).....	16
Figure 2.7. Hydrometeor types (University Corporation for Atmospheric Research, 2009).....	17
Figure 2.8. Radar hardware (Cho, Elkin, & Parker, 2005)	19
Figure 2.9. Incorporation of various Precipitation Mode into a single VCP (Radar Operation Center, 2019)	26
Figure 2.10. Hydrometeor Classification Algorithm (HCA) example	28
Figure 2.11. Object representation systems; Constructive Solid Geometry (CSG) representation (a), solid (b), a boundary representation (c) (Besl & Jain, 1990)	34
Figure 2.12. Surface matching (a), and the parameters used (b) (Winkelbach & Wahl, 2008).....	36
Figure 2.13. Final match; fragment B needs to rotate (a), steps of matching (b) (Winkelbach & Wahl, 2008, s. 9)	37
Figure 2.14. Gradients at the vertices for rotation and bounding box for scale (Shotton, Blake, & Cipolla, 2008, s. 1271)	38

Figure 2.15. The exangle (a), object 1 (O1) and 2 (O2) together will form the model 1 (M1) (b) (Bhanu & Faugeras, 1984, s. 140).....	39
Figure 2.16. Order preservation technique alone (a), preserving order with proximity information (b) (Van Kaick, Hamarneh, Zhang, & Wightton, 2007)	40
Figure 2.17. Low resolution to higher resolution with the Multi Resolution Reeb Graph (MRG) algorithm (Hilaga, Shinagawa, & Kohmura, 2001, s. 206).....	41
Figure 2.18. Texture is applied on the geometric model (Gortler, 2012, s. 8).....	42
Figure 2.19. Logical structures to represent 3D data (a), interpolation and construction of triangular prism instead of hexagonal shapes (b), cutting plane and profile from contours (c) (Lu, Chen, Wang, Yu, & Jiang, 2018, s. 804)	44
Figure 2.20. 3D structure of observed convective cloud by Marching Cubes (MC) Algorithm; 3D visualization (a), vertical profile (b) (Lu, Chen, Wang, Yu, & Jiang, 2018, s. 804).....	45
Figure 2.21. Horizontal section of geopotential heights with the contours and wind with the color (a), same data rendered by Met.3D and mapped (b), 3D isosurface representation with thresholding (c) (Rautenhaus, Kern, Schäfler, & Westerman, 2015, s. 2336).....	46
Figure 2.22. Volume reflectivity for the September 22, 2005 hailstorm along the Michigan/Indiana border (Gibson, 2014).....	47
Figure 3.1. The flowchart of the proposed algorithm (rectangles denote processes while parallelograms denote data)	50
Figure 3.2. Density estimation process and density maps of labels (1, 2, 3, 4, 5).....	51
Figure 3.3. Segmentation for dry ice (blue) and resultant connected components (1, 2, 3, 4, 5, 6)	52
Figure 3.4. Extraction and then matching of boundary polygons	52
Figure 3.5. Meteorology3D GUI Interface Design	54
Figure 3.6. NEXRAD raw data, cropped image, label representation.....	57
Figure 3.7. Cropped raw image (a), segmentation applied only for label 8 (b).....	58
Figure 3.8. Density calculation and density maps.....	60
Figure 3.9. Density maps (a), after applying 10% threshold (b).....	61

Figure 3.10. Connected component analysis	62
Figure 3.11. Top images have radius of 2px, threshold 10% (left), 90% (right); Bottom images have radius of 15px, threshold 10% (left), 90% (right).....	63
Figure 3.12. Segmentation in GUI panel of Meteorology3D	64
Figure 3.13. Directional information of contour pixels; absolute direction (a), relative direction (b) and (c) types of corner pixels (I: inner, O: Outer, IO: Inner outer) (Seo, Chae, Shim, Kim, & Cheong, 2016).....	66
Figure 3.14. Simple Boundary Follower track (Seo, Chae, Shim, Kim, & Cheong, 2016).....	67
Figure 3.15. Track of inner-outer corner pixel at the left-rear pixel. (a) Simple boundary follower (SBF); (b) modified SBF track (Seo, Chae, Shim, Kim, & Cheong, 2016).....	67
Figure 3.16. ISBF (Seo, Chae, Shim, Kim, & Cheong, 2016)	68
Figure 3.17. Radial Sweep Algorithm tracer	69
Figure 3.18. Moore-Neighbor Tracing Algorithm (Seo, Chae, Shim, Kim, & Cheong, 2016).....	70
Figure 3.19. Height calculation (University Corporation for Atmospheric Research, 2009).....	71
Figure 3.20. KML is exported and viewed in Google Earth	72
Figure 3.21. Sample of center of mass table for each elevation angles: Left, bottom to top: 0.5°, 0.9°, 1.5°; right, bottom to top: 1.8°, 2.4°, 3.4°.	74
Figure 3.22. Polygonal coordinates of centers of masses for label4 within elevation angles	75
Figure 3.23. Polygon matching (edited from result).....	77
Figure 3.24. Tracking matches	79
Figure 3.25. Polygon bounding box and pivot points.....	84
Figure 3.26. Numbering of the pivot points	85
Figure 3.27. One vertex may take multiple characteristics	87
Figure 3.28. Merged vertices in a single interval	88
Figure 3.29. Simple “trisurf” example with two triangles.....	90

Figure 3.30. One single vertex at the lower elevation angle interval.....	91
Figure 3.31. Lower interval is shorter than upper interval.....	92
Figure 3.32. Upper interval has one single vertex	94
Figure 3.33. Upper interval is less than lower interval	95
Figure 4.1. 3D view of two consecutively matched polygons (1)	100
Figure 4.2. 3D view of two consecutively matched polygons (2)	101
Figure 4.3. Light and/or Moderate Rain type cloud formation sample (1)	102
Figure 4.4. Light and/or Moderate Rain type cloud formation sample (2)	103
Figure 4.5. Dry snow type cloud formation sample (2)	104
Figure 4.6. Multiple elevation cloud formation of all precipitation types	106
Figure 4.7. Meteorology3D 1 st panel	107
Figure 4.8. Meteorology3D 2 nd panel.....	109
Figure 4.9. Meteorology3D final panel.....	111

CHAPTER 1

INTRODUCTION

1.1. Radars in Meteorology

Meteorology is the study of physical and mechanical nature of mixtures in the air. How the beautiful appearance of the colorful sky has a relation to the seasonal changes provokes curiosity dates back to millennia. Until the second half of the 20th century, meteorology has been seen as a guessing skill rather than a physical science. But things have changed during the wars; pilots' take off, flight and landing safety were a big issue that made pilots and meteorologists establish a very good bond. Following this, the studies carried out by the people who have a mathematical and physical background on atmospheric studies have become a novel resource. For meteorologists and climatologists, each of the theoretical and practical developments in the military field has been used as instruments like computers, radars, aircrafts, and later, satellites (Harper, 2014).

Meteorological radars are used for monitoring severe weather conditions to provide warnings about the location and intensity of the rainfall to meteorological services. In addition to that, they are fundamental sources for researchers. The radars have made a significant progress in forecasting and atmospheric research in the half century after it was adopted specifically for meteorology. Radars are key instruments due to the fact that they bring three-dimensional (3D) real-time data in radii of tens of kilometers.

1.2. Problem Statement

Once dual-polarized weather radars have become practical, it is possible to obtain significantly high accuracy in the estimation of the quantities and types of hydrometeors, which are liquid or solid particles of water in the atmosphere. For some particular classes of hydrometeors, the information that can be obtained from the dual-

polarized weather radar is so striking that it has made the investigating in the formation and the evolution of precipitation easier. The characteristics of hydrometeor classification (phase, size, concentration) are expressed in Figure 1.1. The microphysical origins of the hydrometeor signatures can be calculated with radar moments. The operating frequencies and the particle state distribution functions make the hydrometeor classification precise (Bringi & Chandrasekar, 2001).

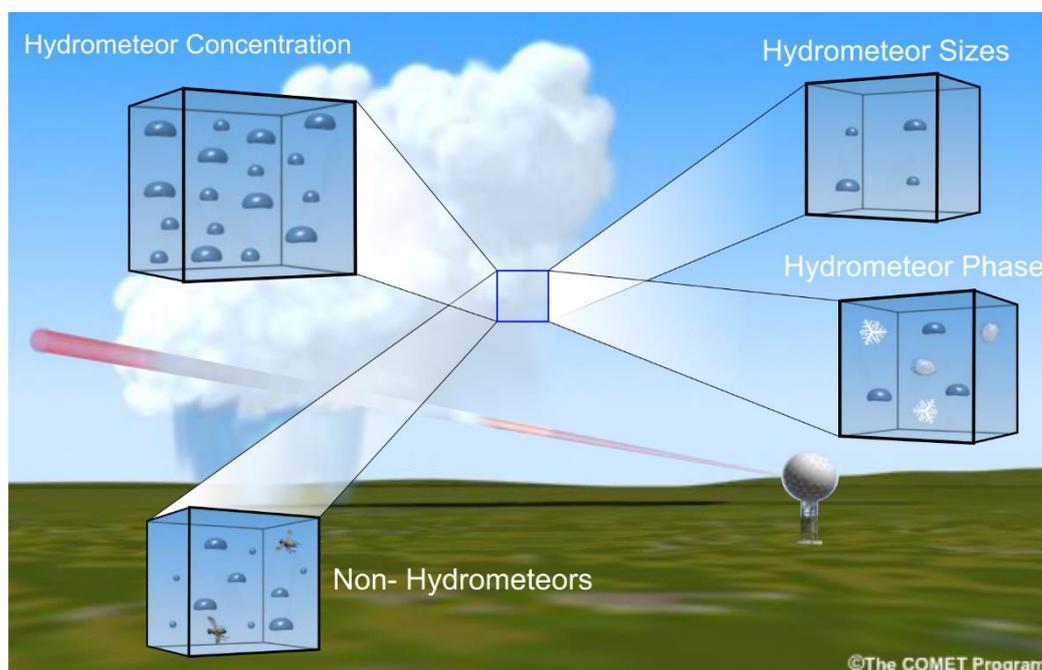


Figure 1.1. Capabilities of hydrometeor classification (University Corporation for Atmospheric Research, 2009)

The process of using computers in meteorology has been a painful process in the beginning. The main parameter in computer-based forecasting was the mean-time-to-failure rate and that was high in the 1950s. On the other hand, computers have become widely used with the application of mathematical prediction models in the field. In addition, computers have started to be a part of different sensing instruments for various purposes.

However, the capability to use visual material from weather radar data on the computer screen for the interpretation of the radar operator still needs to be improved. That is why there are multiple alternatives to the radar software with many different coding languages each having different perspectives on data handling for the creation of this process. Since radars provide images at various elevation angles, their visual interpretation is difficult. With the help of high-speed computers, the radar data have been transferred to the screens as two-dimensional (2D) images for the operators. A lot of data have been monitored instantly with the use of indicators. Although the axes in each indicator are different, (range, elevation, azimuth, etc.) according to each radar scanning pattern, they are not sufficiently understandable, and they cannot contain all the features of the radar images together. Plan-Position Indicator (PPI), Constant Altitude Plan Position Indicator (CAPPI), Pseudo Constant Altitude Plan Position Indicator (PCAPPI) and Range-Height Indicators (RHI) are the most common ones. In PPI, when the radar is placed in the center of the screen, a projection of cloud pattern formation on concentric circles with the same distances to the radar can be seen. In CAPPI, all the elevation angles of the radar are presented together, but an operator needs to determine the altitude in advance. In addition to that PCAPPI represents the interpolation of all data to the pre-determined altitude which probably reshapes the image and can be called as a projection. RHI is completely different from others because the graphical representation samples the range in the horizontal axis and the height on the vertical axis. As mentioned above, although the indicators may exhibit a specific parameter alone, they can only be understood as a 3D radar image when they work together. Therefore, instead of examining all the parameters separately, a 3D visualization would be beneficial to the end-users.

Especially, 3D visualization of hydrometeor classification product can bring a lot of insight into the evolution of weather events, because it is useful for a better understanding of the formation of hydrometeor clusters (objects from this point on) in the 3D environment. By this way, hydrometeor classes can become observable in real time. In order to compose a geometrical model of object in its full dimensionality,

reconstruction, modelling and visualization algorithms are needed. Hence, a toolkit, which is able to parameterize the evolution of interactive 3D object model and display the hydrometeors in their area of interaction is necessary.

A visit was made to the Turkish State Meteorological Service to understand the needs of the operators who perform daily operations and issue warnings for extreme events. It is observed that several screens indicating weather maps, local vertical cross-sections, various meteorological diagrams, satellite imagery, and single or composite weather radar indicators are used and the predictions rely on 2D visualization methods. However, some of the challenging topics still need to be solved such as the vertical and the horizontal extent of the melting layer, sudden fluctuations on the expected path of the convective clouds. A visualization software can be a bridge from proven 2D systems through an interactive way of understanding the complex stages of the precipitation in 3D. The operators stated the importance of seeing the melting layer. They also stressed the importance of real-time analysis, which requires a modular and powerful graphics system architecture to run the algorithms to be developed.

1.3. Contribution of the Thesis

A software package and necessary algorithms are developed to address the defined problem. Specifically, the following is the contribution of this thesis:

- The idea of improving radar's indicator ability by creating and displaying abstract 3D hydrometeor shapes

The information is inferred from a set of 2D classified hydrometeors images for the reconstruction algorithm that creates objects systematically for 3D display and visualizes them using computer graphics methods.

- Density estimation method

2D NEXRAD Level III hydrometeor classification images that are taken from consequent elevations carry various mixed pixels that are mostly disconnected. If we had performed the segmentation directly, it would have led us to have images with

grainy look. To begin with a continuous and smooth segmented image, density estimation method is developed.

- Polygon matching method

Once the borders of objects are obtained from each elevation's image, it is not clear which border polygons (sections) from an elevation corresponds to the border polygons of the same object from other elevations. A method to match the polygons in consecutive elevations created by the same image is developed in this thesis.

- Mesh creation method

To be able to display a 3D model using standard computer graphics algorithms, the objects' surfaces should be represented as meshes composed of triangles. A method is developed to create a 3D wireframe from matched closed polygons.

1.4. Organization of the Thesis

This thesis is organized as follows:

In Chapter 2, background information on atmosphere and hydrometeors are given first. Then, the working principle of the meteorological radar, the reflection and scattering of radar radio waves on the hydrometeors, and radar products are investigated. Finally, the three-dimensional reconstruction and visualization algorithms, which constitute the main subject of the thesis, are explained and some examples from the literature are given.

Chapter 3 describes the proposed algorithm in detail. Estimation of smooth density estimates for each type of precipitation on 2D images, choice of the dominant hydrometeor type in the classification image, the segmentation algorithm, extraction of boundaries of segments, the transform of the boundary polygons into 3D world coordinates, matching of polygons from different elevations belonging to the same object and creation of the wireframe are explained in this chapter. In addition, the

visualization techniques are described in 3D with a triangulated mesh structure created from the polygons of the same object.

In Chapter 4, the 3D reconstruction results of the object formations are shown from different radar sites, on various dates, at distinct hours. Additionally, the Graphical User Interface (GUI) is presented and the use of the algorithm is explained.

Chapter 5 concludes the thesis with interpreting the results and indicating some future research directions.

CHAPTER 2

BACKGROUND THEORY

2.1. The Atmosphere

The air that encompasses the Earth from all sides is called the atmosphere. Atmospheric events are investigated within a wide range of time and space that begins with the average molecules' path (almost 0.1 μm) and extends to the earth's circumference (almost 40,000 km). Sudden turbulent fluctuations take place in micro range as seconds and macro planetary effects may result in months (Markowski & Richardson, 2011).

2.1.1. The Composition of the Atmosphere

Air should not be confused with a chemical compound; yet, it is a mechanical mixture of gasses (Barry & Chorley, 2009). Although the rate of water vapor and other light gasses' existence in a particular volume changes depending on the variations of height, latitude, season and time, the atmosphere composition stays more or less stable. Its nitrogen and oxygen content are more than 99% by volume (Table 2.1).

Table 2.1 *Composition of pure dry air (Saha, 2008).*

Constituent Gas	<i>By Mass (%)</i>	<i>By Volume (%)</i>	<i>Molecular Weight</i>
Nitrogen (N ₂)	75.51	78.09	28.02
Oxygen (O ₂)	23.14	20.95	12
Argon (Ar)	1.3	0.93	39.94
Carbon dioxide (CO ₂)	0.05	0.03	44.01
Neon (Ne)	1.2×10^{-3}	1.8×10^{-3}	20.18
Helium (He)	8.0×10^{-4}	5.2×10^{-4}	4.00
Krypton (Kr)	2.9×10^{-4}	1.0×10^{-4}	83.7

Hydrogen (H ₂)	0.35×10^{-5}	5.0×10^{-5}	2.02
Xenon (X)	3.6×10^{-5}	0.8×10^{-5}	131.3
Ozone (O ₃)	0.17×10^{-5}	0.1×10^{-5}	48.0
Radon (Rn)	-	6.0×10^{-18}	222.0

Composition of the atmosphere tends to stay in equilibrium state close to the surface of the Earth. While a gas decreases in one process, other chemical reactions make the same gas re-synthesis by other processes. For instance, nitrogen's reaction with soil bacteria and planktons in ocean decreases the nitrogen. On the other hand, decaying of living organisms increase the nitrogen. This structure also applies to oxygen. Oxygen decreases with the chemical bonds, creating oxides, while it increases with photosynthesis (Ahrens, 2011).

2.1.2. Structure of the Atmosphere

The atmosphere is composed of concentric layers. The atmosphere expands up to thousands of kilometers from the surface. 30 km is accepted as a boundary because 97% of atmosphere's force on the surface is caused by the air below this level. Earth's gaseous coverage and atmosphere's layers are given in Figure 2.1.

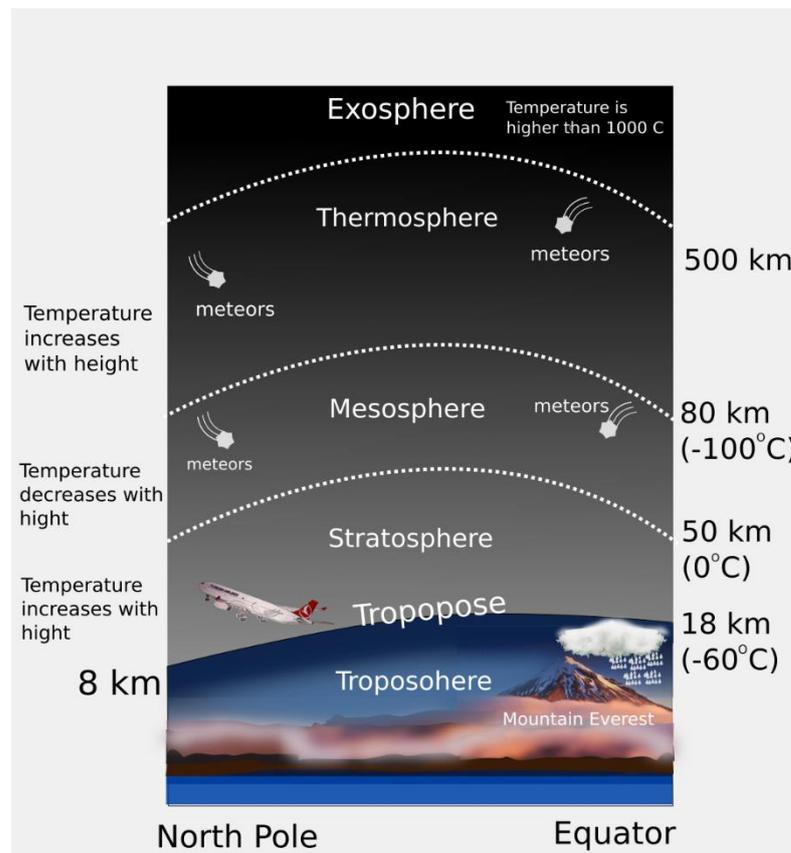


Figure 2.1. Structure of the atmosphere

Layers of the atmosphere are based on temperature, composition, and electrical properties (Ahrens, 2011). The circulation of temperature of air, pressure and humidity is the main concern of meteorology. The first layer, which is called the troposphere, has an altitude that differs from 8 km to 18 km because hot convection currents existing in the equator make the gases move up more with respect to the poles. Many life forms depend on this atmosphere layer and it is the only layer where major weather events appear. An increase in height causes a decrease in the temperature. The tropopause, an intermediate region, is at the top of this layer. It has the attributes of both troposphere and stratosphere. The mixing of air in the tropopause may result in jet streams - squeezed river like high winds that overrun 186 km/h (100 knots). The reason behind the slow rise of the temperature at the higher altitudes of the stratosphere is that the ozone present at this particular level absorbs energetic

ultraviolet (UV) solar energy (Ahrens, 2011). The weather-related events do not occur here. The third layer above stratosphere is the mesosphere. It is known to be the coldest part with the mean temperature of -90°C . After mesosphere, there is thermosphere. The increase in temperature due to the absorption of solar energy by oxygen molecules results in temperatures exceeding 500°C . Although an air molecule can advance freely (without hitting another molecule) less than one millionth of a centimeter at sea level, it takes more than a kilometer at the top of the atmosphere named exosphere because of very low air density. That may result in molecules running off from the gravity (Ahrens, 2011).

2.1.3. Formation of Clouds and Precipitation

Circulation of water is called the hydrological cycle and it is essential to know the quantity of the water to understand the concept of how water molecules travel from atmosphere to surface and back to the atmosphere (Barry & Chorley, 2009). Water vapor is an invisible gas and its unique property is that it can be present in a particular cloud as gas, liquid, and solid forms at the same time. Interestingly, condensation in clear air is a rare event; unless, moisture come across with a proper surface to condense on. The proper surfaces, which help forming of cloud droplets, are the particles suspended in the atmosphere known as Cloud Condensation Nuclei (CCN's). Most of the time, aerosols behave as CCN. Aerosols are both coarse and fine particles and they are categorized not only in size, but also composition and source type (Stevens, 2010). They, on the other hand, have 80% of their mass in the range of the atmosphere from the surface up to 1 km altitude. Aerosols also take part in propagation of light via being a scatterer or absorber.

When the air rises, its temperature decreases so the cooling process begins and water starts to condense at a specific altitude. There are multiple reasons for the ascent of the air. One of them is the birth of convection rise through the topography over heated surfaces. Secondly, air flow due to pressure difference and finally, because of the fronts (Ahrens, 2011). These are the reasons of cloud formations. Clouds are identified

according to (1) the common form, pattern and vertical extent, and (2) their altitude. In Figure 2.2 cloud types are given.

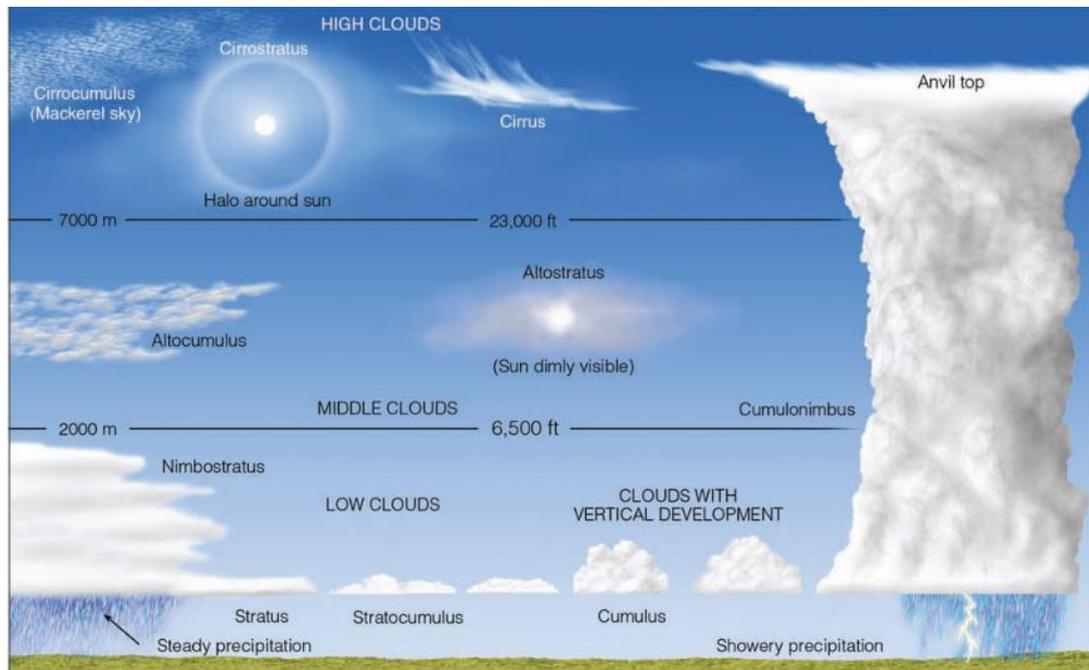


Figure 2.2. Cloud types (Ahrens, 2011, s. 109)

It is a big dilemma that although radius of cloud droplets (1 to 50 μm) and raindrops (more than 1 mm) do not have even the slightest similarities, a cloud to produce rain takes minutes in most of the cases (see Figure 2.3) Ahrens D. (2008) contended that the time it takes for millions of cloud droplets to form a raindrop by condensation would probably take weeks because it is a slow process. Although all the complexity of the precipitation is not yet fully understood, two significant physical attributes draw attention: the collision-coalescence theory and the ice-crystal formation of cold clouds (Barry & Chorley, 2009; Ahrens, 2011).

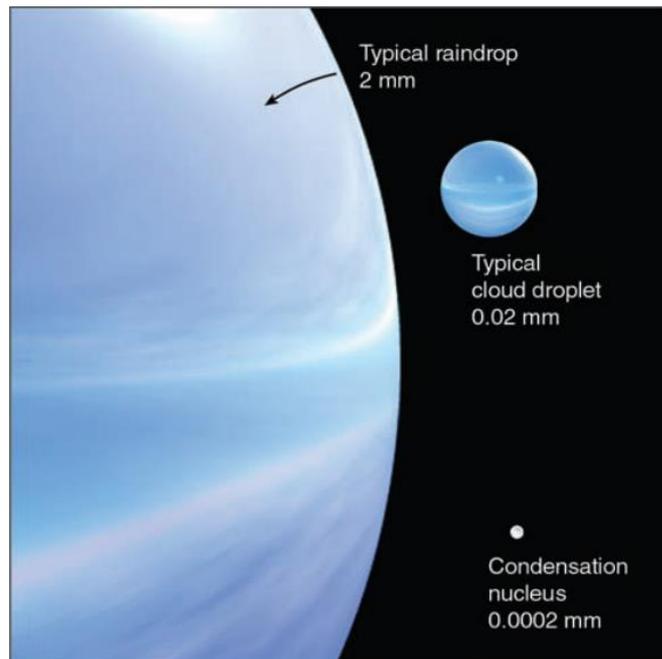


Figure 2.3. Representation of raindrop, cloud droplet and condensation nucleus (Ahrens, 2011, s. 128)

The fundamental influence on the evolution of ice crystals by sublimation is the existence of supercooled water droplets. Once water vapor begins to decrease by this transaction at the dew point, the air gets unsaturated. This is the primary effect of the evaporation of droplets. This process can end in two ways: either the ice crystals become large enough to fall down or no more droplets exist. Moreover, the time when ice crystals begin to melt while falling down, the rain occurs. This is called Wegener–Bergeron–Findeisen precipitation process (Collier, 2016). Commonly, midlatitude rainfall is triggered with the melting process after ice is formed. The phase change of ice (solid to liquid) in the transition layer composes significant discontinuity for radar measurements. The melting layer is the reason of the bright band in the radar observations. Once transition begins at a well-defined height, it appears as an enhanced reflectivity and that interface is called the melting layer (Szyrmer & Zawadzki, 1999).

Most of the situations that trigger the initiation of snow are the clouds below the freezing level that are coinciding with an ascending air. As Figure 2.4 simply shows, there are multiple ways to encounter different types of snow. Commonly small crystals

with diameters less than 75 μm are produced by aerosol nuclei at $-5\text{ }^{\circ}\text{C}$. As ice crystals keep growing by sublimation, snow crystals are formed. When only ice and snow crystals exist, the collision is called aggregation. In addition to that, when ice crystals get into contact with droplets, the droplets freeze. This process is called riming. Once snow crystals rime, they form graupels (Collier, 2016). Ice crystals, graupels and other types of hydrometeors that can be observed in the atmosphere are explained in Subsection 2.2.8.5.

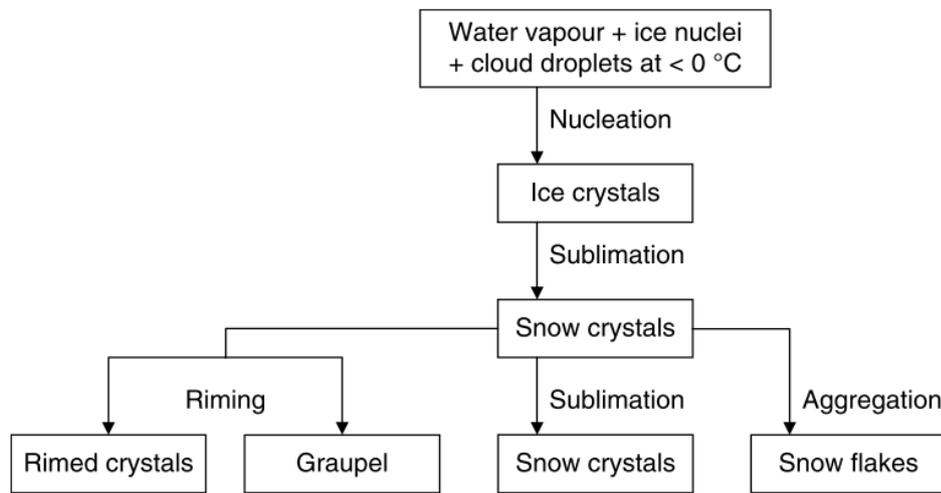


Figure 2.4. Flow diagram of different type of snow formation (Collier, 2016, s. 64)

Estimation of dominant hydrometeor type at the beam height is significant in particular applications. These are the following:

- The determination of occurrence and location of mixed-phase conditions. Specifically, the melting of snow and ice has an impact on the radar based Numerical Weather Prediction (NWP) algorithms.
- Severe weather warning conditions are under the influence of discrimination of heavy rain and hail.

- The life cycles of various storm types have an initialization, which can be predicted by numerical forecast models with the help of hydrometeor phase changes.
- Study of lightning in relation to cloud microphysics.
- Evaluation of microphysical properties and hydrometeors can be compared with different remote sensing instruments (e.g. hydrometeor type and its temperature).

2.2. Weather Radars

2.2.1. Radar Principle

The aim of the radar instrument (RADio Detection And Ranging) is detecting a target and ranging its position, velocity, direction, form, and its specification in three-dimensional space by using electromagnetic radiation. Radar is an electronic system in which a transmitter generates the radio waves. With the help of an antenna, the generated wave is radiated into space. As Figure 2.5 shows, the reflected radio wave from target and lapse of time (round-trip between radar and target) are measured so that the properties of the target can be found (Fukao, Hamazu, & Doviak, 2014). While the lapse of time gives the distance of the target to the antenna, the frequency shift (Doppler shift) gives clues about the velocity of the target. As explained in Subsection 2.2.7, parameters like polarization can be used to determine reflectivity, correlation coefficient, specific differential phase, etc. of the target.

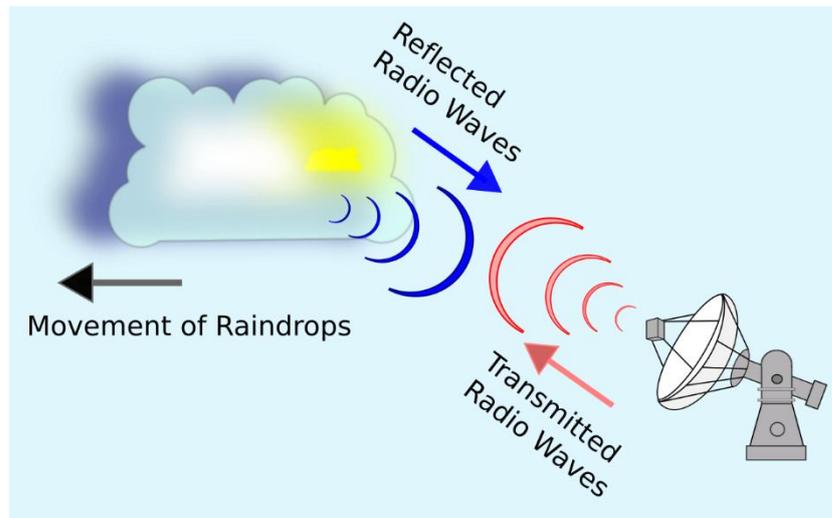


Figure 2.5. The working principle of Radar

2.2.2. Scattering of Electromagnetic Waves in the Atmosphere

One of the most important factors for meteorological radars is the signal reflected from particles, which are known as hard scatterers. Weather radars are investigated under three regions based on the scattering property. The comparison of the wavelength with respect to circumference of the scatterer determines in which region it is going to scatter. In Rayleigh region, signal wavelength is much larger than the diameter of the scatterer (Figure 2.6). On the contrary, in the optical (geometrical) region, the corresponding signal wavelength is much smaller than the diameter of the scatterer. In between of these two, there is the last region called Mie. When precipitation particles are detected with microwave, main scattering of the weather radar is in the Rayleigh region (Fukao, Hamazu, & Doviak, 2014).

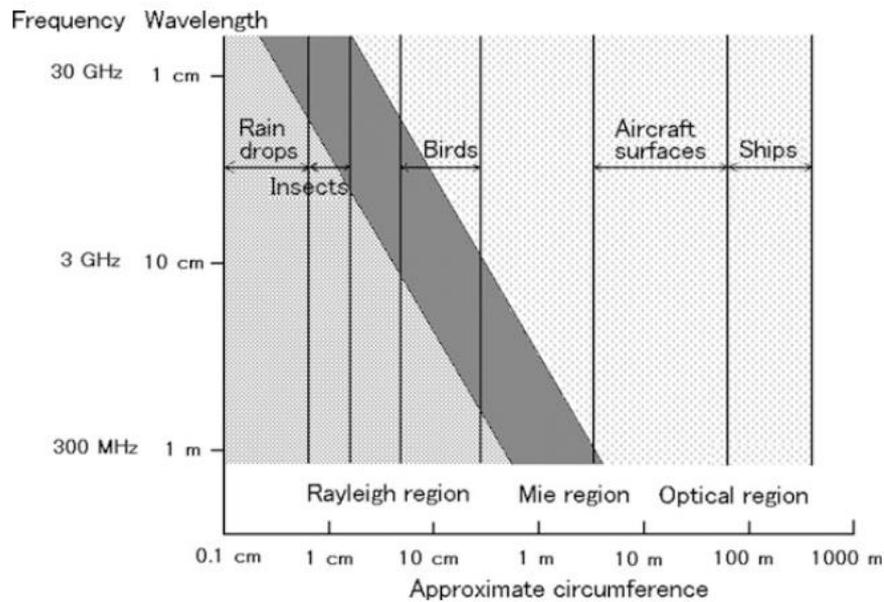


Figure 2.6. Electromagnetic waves' interaction with matter shown as a unit of an approximate circumference (Nathanson, Reilly, & Cohen, 1991, s. 152)

2.2.3. Hydrometeors and Scattering

Using millimeter-wavelength radar is challenging because of non-Rayleigh scattering and gradual loss of wave intensity through the rain (attenuation). The progressing of incident radiation on the Rayleigh scattering regime relies on the dielectric properties of water and ice, and hydrometeor size (Dolan, Rutledge, & Stevan, 2009). As the raindrops coalesce, they get oblique in shape, which results in higher reflection from horizontal polarization. Because of this, different hydrometeor types results in different radar base moments (explained in detail in Section 2.2.7) and different polarizations. Hence, the dual-polarized weather radar has been preferred lately to help distinguishing various scatterers.

Using derived variables like reflectivity factor, differential reflectivity, linear depolarization ratio, cross-correlation coefficient, specific differential phase, classification algorithms can be used to determine the hydrometeor type. Finally, a confidence vector can also be calculated, which simply quantifies the quality of the measurements by parameterizing radar miscalibration, attenuation, non-uniform beam

filling, partial beam blockage and receiver noise that determines the error weight in the classifier (Park, Ryzhkov, Zrnić, & Kim, 2009). As a result, hydrometeor classes like biological attenuation and ground clutter, rain, heavy rain, rain/hail, big drop, dry snow, wet snow, graupel, and ice crystals, are determined. The melting layer property of hail and heavy rain discrimination and season dependent hydrometeor type coverage can also be possible. Figure 2.7 shows some hydrometeor types and their relative sizes (University Corporation for Atmospheric Research, 2009).

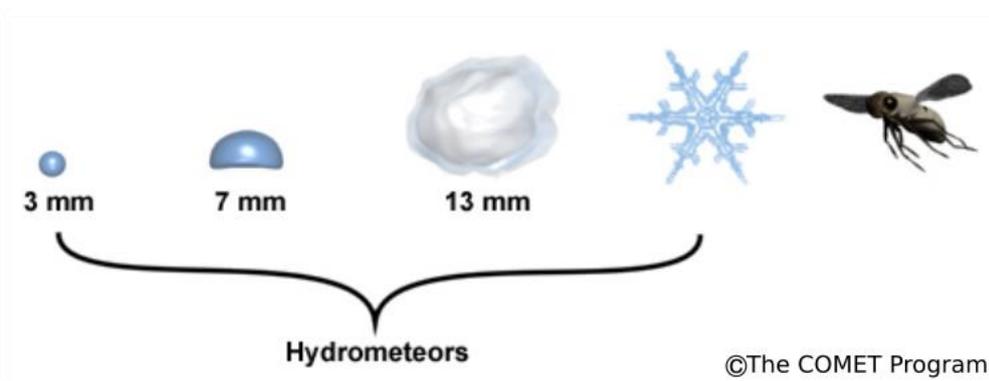


Figure 2.7. Hydrometeor types (University Corporation for Atmospheric Research, 2009)

2.2.4. A Brief History of Weather Radars

Harper (2014) asserts that in the 1930s, British scientists were unable to maintain directional accuracy of their radars because they could only use large beams with long wavelengths. With the invention of the magnetron, the use of millimeter wavelength has enabled the development of the necessary techniques. The use of short wavelengths was predicting the position of their army and opponent's movements during the Second World War (WW-II) with much higher precision. On the other hand, they figured out that while the rain and snow begun to curtain the new radar's indicators, it did not affect the old radars. That is how the name of the weather radar was engraved on scientists' perception. Han (1991) stated that the utilization of radars for forecasting purposes spread from the WW-II. On February 1941, rain-bearing cumulus clouds were observed by an S-band radar established on the English seaside.

Following this event, weather radars have maintained their advance (Han, 1991). The first precipitation tracking and short-range sensing instrument that has been conceived, improved and distributed for meteorologists was the AN/CPS-9 Storm Detection Radar. Raytheon Corporation produced CPS-9 in 1949. Although the radar has been used experimentally to measure rainfall; since then, it is obvious that the operational implementation is still not complete (Whiton, Smith, Bigler, & Wilk, 1998). This, of course, does not alter the fact that meteorological radars provide the best way to measure rainfall remotely, both qualitatively and quantitatively. However, the technological innovations that have been developed in this direction will hopefully make currently used meteorological radars better.

2.2.5. Weather Radar System

Although there are many different arrangements of the radar's hardware, major components involved in a weather radar system can be simplified as shown in Figure 2.8.

The transmitter unit is the supplier of the microwave radiation. Free space propagation of the electromagnetic wave begins from the feed-horn placed at the focal point of the reflector, which is a dish-shaped circular paraboloid used to redirect the energy to the desired point of observation.

The receiver detects echo signals including noise, clutter and interference. Antenna needs to have a protection from both precipitation and the cold. In addition to that, in case of an accident, the spinning radar needs to be stopped. Also, severe wind is another factor that should be minimized when the radar is scanning. As a result, the Radom (RADar DOME) is the primary unit of preservation (Büyükbaş, 2005).

The digital signal processor (DSP) is responsible for clutter filtering and generation of the radar base products. Analog-to-digital conversion (A/D) uses in-phase (I), quadrature (Q), and logarithmic (LOG) as entries and samples them in the form of digital data. Then the signal processor performs algorithms on electrical properties and directs them into meteorological products of concern.

The Radar Data Acquisition (RDA) unit, which includes the receiver and the DSP, generates the Level I and Level II data. Special section is provided for describing base moments of the radar (Subsection 2.2.7), which are part of these data. Following the RPA, Radar Product Generator (RPG) does most of the data processing. The RPG's algorithms convert base moments that are found in RDA into the meteorological products like the hydrometeor classification data that is used in this thesis. The RPG sends the products to Principal User Processors (PUP). PUP is composed of graphics processors for user display systems, communication systems and a workstation. PUP's products are called Level IV data. This data is used internally at Weather Surveillance Radar-1988 Doppler (WSR-88D) sites and mostly for the purpose of operator training (Timothy, 1993). That is why this part is not explained.

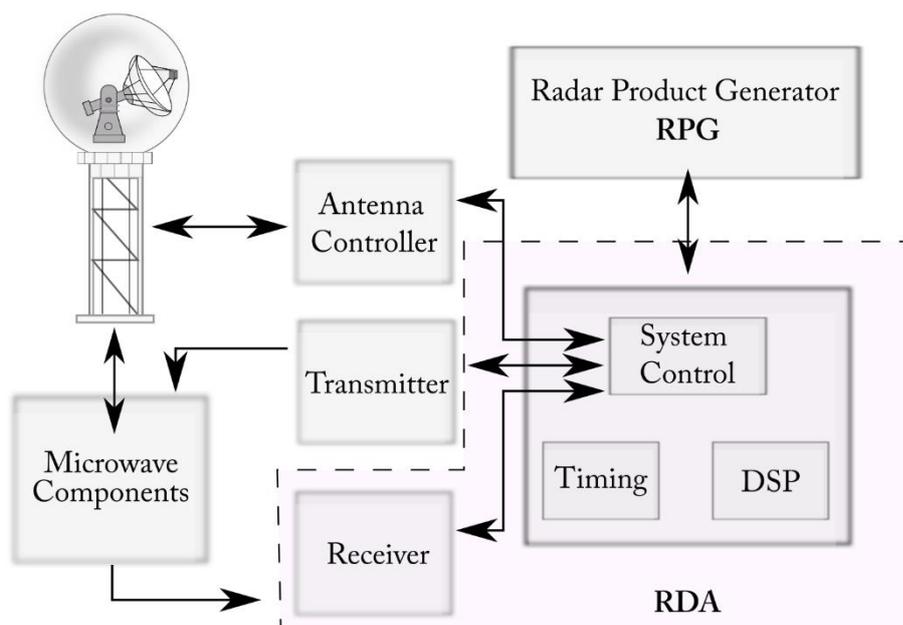


Figure 2.8. Radar hardware (Cho, Elkin, & Parker, 2005)

2.2.6. Types of Weather Radars

2.2.6.1. Stationary Radars

Stationary radars are located at predetermined, carefully chosen locations. Multifunction Phased Array Radar (MPAR) is an example. Following the several years of operational usage by the U.S. Navy, AN/SPY-1 was reproduced as a phased array radar for missile threats in 1973 (Benner, 2007). The phased array radar beam can be electronically steered at various boresights. This helps both to get faster updates and to perform adaptive scan strategies. MPAR's functionality can be described as individual polarimetric Phased Array Radar (PAR) structures working in a network. It senses and chases the aircrafts and at the same time observes upper air formations (Ivić, 2018). The integration of combined requirements of all the existing radar networks is met by the MPAR system, which has been a recent progress on radar observations.

In case of heavy weather conditions, the proper judgment for public safety should contain a minimum number of false alarms, which are only possible with rapid scans. The likelihood of encountering risky weather conditions can be determined by MPAR in just seconds. With the MPAR's capability of tracking storms, meteorologists gained the primary skill of understanding when the storm gets stronger or weaker (NOAA, 2015).

2.2.6.2. Mobile and Deployable Radars

Extraordinary weather conditions, which can simply be presented as storms, cyclones, hurricanes and thunderstorms, have very rare occurrence. Gathering useful data is mainly correlated with the location of severe weather. Therefore, expecting the representative information of unusual formations from stationary weather radars may not be a good idea, if it is not in its range. That is why mobile radars have been known to be the best alternative in risky and hard-to-reach places with the ability of easy installation and lift off. In an open area, to calculate the exact location of the formations, the antenna azimuth and elevation angle need to be known. To be successful, radar-carrying vehicle should be leveled precisely, and the geographic

viewing direction should be determined appropriately. This process would become crucial if this specific radar takes part in a network of radars (Rauber & Nesbitt, 2018).

2.2.6.3. Airborne Radars

Airborne Weather Radar has extra capabilities apart from dual-polarization and Doppler speed measurement, such as a radiometer channel for the measurement of brightness temperature outside. Observation of front development is a way of recognizing and forecasting thunderstorms. Paramore et al. (2006) claimed that the difference of developing, mature and dissipating thunderstorm stages could be better estimated by scanning the thunderstorms vertically. From that point of view, determining the amount of vertical development in the thunderstorm above the freezing level is one significant parameter. In addition to that, deciding if the amount of thunderstorm is above the tropopausal level and understanding the temperature at the thunderstorm's peak, or recognizing the precipitation density at the peaks are vital. Although it is possible to see a serious turbulence area at the top of the storm formation, creating a danger for an aircraft, it is thought that scanning in the aircraft's along-track would save time to draw a new route.

2.2.6.4. Space-borne Radars

The first space-borne weather radar was launched in 1997 by the Tropical Rainfall Measuring Mission (TRMM) as a precipitation radar (PR). Complex downward-looking geometry and usage of higher, attenuating frequencies in rainfall measurement makes space-borne radar challenging due to antenna size constraints. Selection of frequency is a compromise between antenna size and attenuation. Durden et al. (1994) stated that the 13.8 GHz frequency band creates a boundary for space-borne radars. Higher frequencies result in higher attenuation from the rain. This weakening in echo prevents one to have more information from the bottom of the cloud formation. On the contrary, lower frequencies cause lower spatial resolution. This requires a larger antenna that is not very practical on a single space-borne

platform. When scan strategies are compared, the cross-track has better vertical resolution than conical.

2.2.7. Radar Base Moments

2.2.7.1. Radar Reflectivity Factors

Observation of clouds quantitatively is a rather significant part of meteorology. While clouds are the non-precipitating part of the water condensed in atmosphere, the electromagnetic propagation through spatial and temporal structures of the clouds are scattered back. This back-scattering is called the reflectivity (Sauvageot & Omar, 1987). The reflectivity factor and equivalent reflectivity factor should not be confused with each other. While reflectivity factor (dBZ) is related with the rainfall rate, the equivalent reflectivity factor (Z) is a more general implementation of the concentration of particles in a cubic volume. The fact that raindrops are accepted as dielectric, the dipole moment of dielectric polarization effects the type of scattering. Reflectivity factor (in mm^6 per m^3) is applicable when the drop size distribution is certain. Second one approximates it by using the radar range equation. Therefore, both of them are supposed to be same, but they are not; neither drop size distribution is known precisely nor their formation of distribution. Concentration should be consisting of uniformly distributed homogeneous spheres of small raindrops with Rayleigh scattering that depends on the particle size with respect to the wavelength. They need to have the same dielectric constant. If all assumptions were not able to be met, equivalent reflectivity factor is needed. For more detail, see (US Department of Commerce, 2006; Fukao, Hamazu, & Doviak, 2014; Stagliano Jr, 2005).

2.2.7.2. Mean Radial Velocity

When the radar beam hits the particles, their vibrations start to spread electromagnetic energy all around, and some of which would come back to antenna. The echoes backscattered from the particles that are approaching to the radar have higher frequency (Fukao, Hamazu, & Doviak, 2014). The shifts of the frequencies with the velocity of the target is called Doppler shift. Doppler frequency shift is calculated from

the phase of the received signal. Therefore, the radial velocity is based on the phase difference between the pulses through the beam line (Rauber & Nesbitt, 2018).

2.2.7.3. Spectrum Width

From different particles with different radial velocities within a volume of specific resolution, the mean of these velocities are known as the mean Doppler velocity. And as the second moment, the variance of this spectrum implies the spectrum width. This variable is parameterized as wind shear and turbulence. When the cloud formation can easily be detected, estimating the stages of cloud maturation or dissipation by observing the particle's velocities becomes possible. For instance, developing stage of a storm collects particles upwards and dissipating stage may cause precipitation. In addition, ground clutter would probably be the sign of scanning the lowest altitude part of the clouds (Rauber & Nesbitt, 2018).

2.2.7.4. Differential Reflectivity (ZDR)

Differential reflectivity is proportional to the ratio of the power returned from horizontal pulse over the power returned from vertical pulse. Differential reflectivity depends on the median shape and size of the hydrometeors. Horizontally oriented mean profiles of the targets have their ZDR greater than one. On the other hand, vertically oriented targets have their ZDR less than one. When ZDR is equal or close to one, that stands for a hydrometeor of nearly circular mean profile. Therefore, spherical targets (e.g. hail, debris and snow) have values near one; on the contrary, horizontally elongated targets (e.g. raindrops-medium to large) have values greater than one.

2.2.7.5. Correlation Coefficient (CC)

Correlation Coefficient is the correlation of the vertical and the horizontal back-scattered power with respect to scanned sample volume of radar. This product distinguishes precipitation from non-precipitation. It also examines whether the

precipitation particles have the same shape and type as pure hydrometeors or a mixture. The values of CC in the range from 0.9 to 1.0 are most likely precipitation.

2.2.7.6. Specific Differential Phase (KDP)

Specific Differential Phase is a measure of the phase shift in a relation with horizontal and vertical returned signal because of the forward propagation. That means the comparison of horizontal and vertical signals' slow down with respect to each other. This product identifies the area of the heavy rain.

2.2.8. NEXRAD Radars

2.2.8.1. History

Whiton et al. (1998) have described the time of early weather radars in detail until the radars equipped with Doppler capabilities were installed. The radar systems carrying continuous-wave (Rogers, 1990) do not have enough capability to estimate the proper ranges of the fluctuations that occur in the atmosphere. With the pulse-Doppler implementation, the derivation of the radar base moments became possible with a single radar. The weather radar with a 5 cm pulse-Doppler ability, named as Porcupine, has become operational in 1961. After a while, Plan-shear indicator (PSI) (Armstrong and Donaldson 1969) was generated to capture data in real time, which was the beginning of the first mesocyclone observation in 9 August 1968 (Donaldson, 1970). Both in 1971 and in 1973, National Severe Storms Laboratory (NSSL) constructed S-band dual-Doppler system. The full dual-Doppler came into operation one year later, and that was the time of first tornado vortex signature display. The velocity-azimuth display (Lhermitte, 1970; Rabin, 1980) has been extended and became the WSR-88D's significant indicator. In the existence of echo-isolation algorithms, developing an echo-tracking algorithm has become the goal. From 1983-1984 on, that algorithm has been demonstrated and been applied to WSR-88D which is the product of Next Generation Weather Radar (NEXRAD) program. As a result of this joint effort of the Department of Commerce (DOC), Defense (DOD), and Transportation (DOT) WSR-88D was the first radar that enabled storm observation

by the features of segment identification, tracking, positioning and centroid location. In the light of these improvements, NEXRAD was ready to be operational (Forsyth, 1981; Crum, 1993). The First WSR-88D became operational in fall 1990 in the Twin Lakes, Oklahoma. Following that, 158 additional NEXRAD radars have been installed until June 1997 (Whiton, Smith, Bigler, & Wilk, 1998).

2.2.8.2. Functional Characteristics of WSR-88D

Specifications of the WSR-88D are given in Table 2.2.

Table 2.2. *WSR-88D specifications (Whiton, Smith, Bigler, & Wilk, 1998)*

Wavelength (cm)	10.5
Peak Transmitted Power (kW) (at transmitter)	750
Pulse duration (μ s)	Sh pulse: 1.57 Ln pulse: 4.7
Pulse repetition frequency (s ⁻¹)	Sh pulse: 8 Selectable: 318-1304 Ln pulse: 318-452
Linear IF amplifier and response	Dynamic range: 93 dB with AGC
Beamwidth, azimuth ($^{\circ}$)	1
Beamwidth, elevation ($^{\circ}$)	1
Antenna mobility rng ($^{\circ}$)	360 az, -1-60 el
Effective antenna system gain (dB)	45

NEXRAD begins with RDA to decide either it would start in Precipitation Mode or Clear Air Mode depending on the precipitation's range, aerial extent, identity, and condition. These modes can operate both automatically or manually according to the Volume Coverage Patterns (VCP). Precipitation mode is under operation when echoes have enough magnitude. In order to have low maintenance cost, the Clear Air Mode works if there is no observable precipitation, and intensity is under some predetermined thresholds. Although Clear air mode has two VCPs (31-32), as Figure

2.9 depicts, the Precipitation Mode is divided to three main groups and has seven different VCPs (US Department of Commerce, 2017).

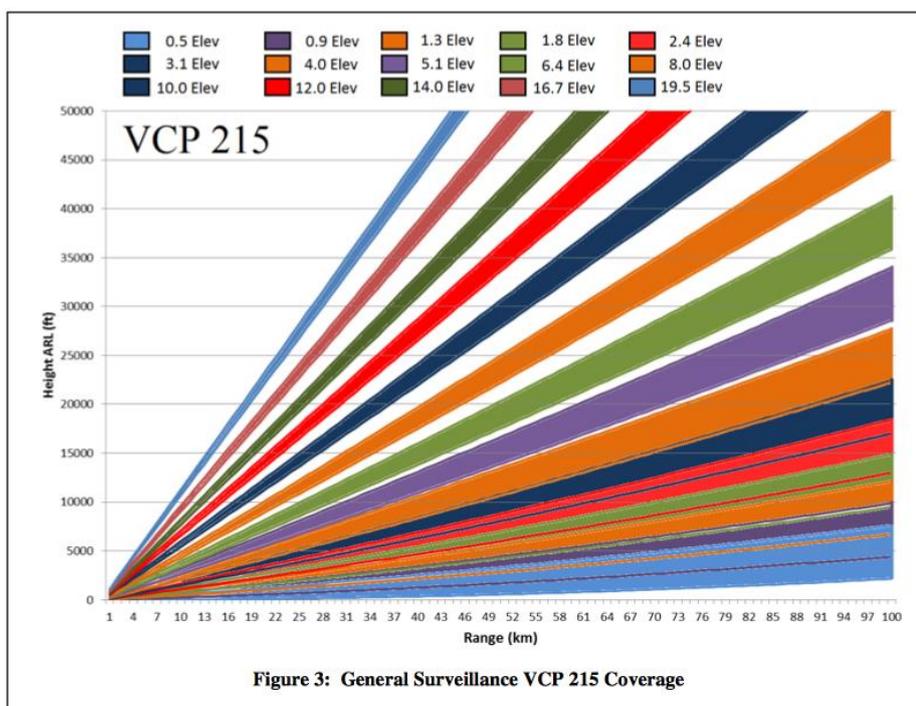


Figure 2.9. Incorporation of various Precipitation Mode into a single VCP (Radar Operation Center, 2019)

2.2.8.3. NEXRAD Data Format

NEXRAD has a unique data output format processed under the Central Collection and Dual Polarization Information Distribution. The system is responsible of communication networks for sizeable amount of data, networks, and servers for both storage and collection. The sub-sections of this particular system generate Level II and Level III data. One of the main reasons of division of output data is that while Level II provides samples of volumetric scans and metadata for the creation of base moments, Level III products are derived from the RPG from Level II data.

135 radars together provide level II data real-time. This information is collected by a center and shared out via websites to external users. In addition, they are stored in

National Climatic Data Center (NCDC). Level II data consist of Reflectivity, Mean Radial Velocity, Spectrum Width, Differential Reflectivity, Correlation Coefficient, and Differential Phase. In addition to that, the status metadata, modes of functions (date, time, and antenna inputs), synchronization and calibration are also part of Level II data (Istok, Fresch, Jing, & Smith, 2009; Radar Operation Center, 2008)

Level III data are the outcome of the RPG. Mainly used for meteorological analysis, forecasting and tracking. Like Level II, Level III data are also proceeding by a center and distributed by Radar Product Central Collection and Dissemination Service (RPCCDS). The data are stored by NCDC and free to be used via FTP download (Istok, Fresch, Jing, & Smith, 2009).

2.2.8.4. Data Access

Extensive amount of datasets is stored in NCDC archives such as information of rain gage measurements, forecast models, radar and satellite imagery. However, binary formats and variables such as point, time series, grids, radial and swath are complicated structures for users working with software programs. For this reason, National Oceanic Atmospheric Administration (NOAA) provides powerful tools in order to decode data into formats that help users gain access to data with common software.

Weather and Climate Toolkit (WCT) is one of the free influential platforms that are able to display and export NEXRAD Level III Hydrometeor Classification data. Many scientific formats can be exported by WCT: Shape file Point and Polygon (P&P), Well-Known Text (P&P), Raw and Gridded NetCDF. If they are mapped, Arc/Info ASCII Grid, and GeoTIFF are also possible (Ansari, 2010). There are two reasons why Gridded NetCDF is chosen in this thesis: It supports MATLABTM and Google Earth and provides mapped array-oriented data access.

2.2.8.5. The Hydrometeor Classification Algorithm (HCA)

HCA uses dual polarization base products to decide observed precipitation types accurately. One of the significant points is that HCA determines the precipitation type at the beam height. Unlike many algorithms, HCA does not use a strict “yes or no” logic because it has fuzzy logic algorithms. The HCA accepts Reflectivity, Velocity, ZDR, Correlation Coefficient (CC) and Specific Differential Phase (KDP) as input. It compares these inputs to defined characteristics of each classification and assigns a likelihood value to each class in the range [0, 1]. After assigning likelihood values to each class, the one with the highest likelihood value is displayed. Figure 2.10 shows the parameters of a pixel that are inserted into the algorithm and the resulting likelihood values. It is observed that HA has the highest likelihood value. Therefore, this pixel is assigned as the HCA class. The distinctive characteristics of the precipitation types are given in Table 2.1.

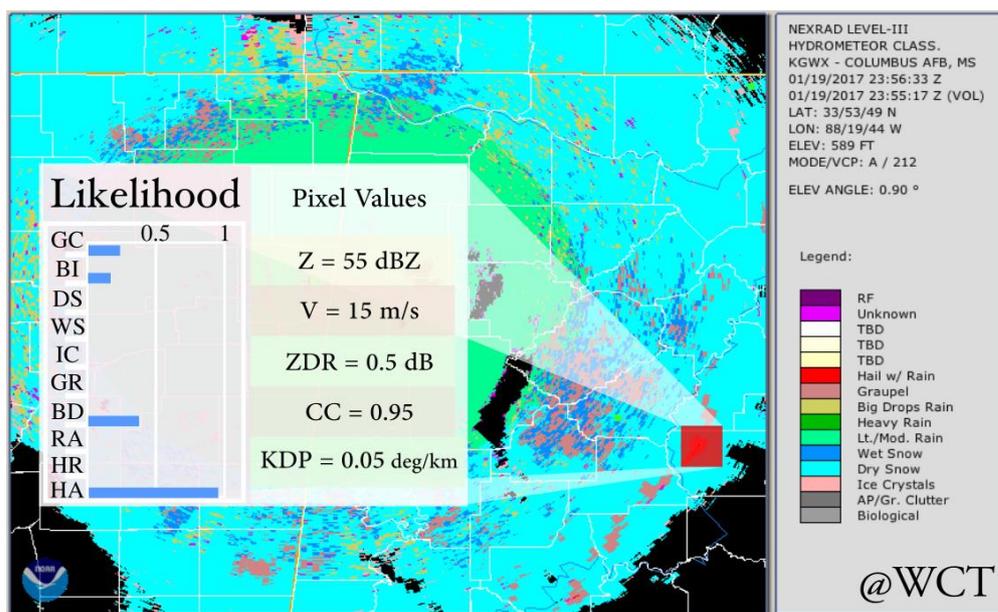


Figure 2.10. Hydrometeor Classification Algorithm (HCA) example

Table 2.3. Ranges of hydrometeors in the corresponding radar products (Dual-Pol Training, 2019)

Classes / Products	Reflectivity dBZ	Correlation Coefficient CC	Differential Reflectivity dB	Specific Differential Phase deg/km
Light to Moderate Rain	5-50	0.97-1.01	0.0-6.0	0.0-1.0
Heavy Rain	40-60	0.95-1.0	0.5-8.0	1.0-5.0
Big Drops	10-50	0.92-1.01	2.5-7.0	0.0-2.0
Hail and Mixture	45-80	0.75-1.0	-0.3-4.5	-2.0-10.0
Graupel	30-55	0.92-1.01	-0.3-2.2	-2.0-2.0
Dry Snow	5-35	0.97-1.01	-0.3-1.5	-1.0-5.0
Wet Snow	25-50	0.88-0.985	-0.5-3.0	-1.0-0.5
Ice Crystals	0-25	0.95-1.01	-1.0-5.0	-1.0-5.0

Radar meteorology is based on the scattering of electromagnetic energy by hydrometeors. Each type of hydrometeor has different characteristics of the returned signal. There are 10 predefined classifications for the output of HCA that are explained below.

Biological Scatterers: These targets include birds and insects that are typically seen at low altitudes because they do not get very high off the ground. One exception to this rule is during migration periods where birds/insects can get hundreds of meters off the ground.

Ground Clutter / Anomalous Propagation: Ground clutter includes any object that is fixed or moving along the ground, including but not limited to buildings, trees, mountains, interstates / roads, and wind farms. These objects typically appear only low altitudes, but can appear at further ranges under a special circumstance called anomalous propagation (AP).

Ice Crystals: These are non-aggregated frozen hydrometeors, formed individually and often found in drier areas of precipitation. Ice crystals include all winter-type precipitation except for dry or melting/wet snowflakes. Ice crystals include the crystals in the form of needles, columns and plates. These precipitation particles typically occur high up in the thunderstorms and in the very cold regions of snowstorms. The densities of ice crystals, from lower to higher order are needles, columns and plates, respectively.

Dry Snow: Dry snow contains collected snowflakes that do not have any liquid water coating. It comprises of what people typically think of as snow. We all associate the hexagonally shaped ice crystal with snowflakes. Dry snow typically occurs just above the melting level in temperatures from 0 degrees Celsius to -15 degrees Celsius.

Wet Snow: Wet snow contains water-coated snowflakes. Wet snow comprises of melting snowflakes only. Therefore, one will only see the wet snow classification near, and just below the melting level.

Light and/or Moderate Rain: Light-to-moderate rain consists of purely liquid hydrometeors that result in all types of rain such as drizzle, and the remaining with rain rates up to 2,5 cm/hour. This is the most common type of precipitation in warm-season type thunderstorms and rain events.

Heavy Rain: Heavy rain is a special category that constitutes large concentrations of liquid hydrometeors found mostly in areas experiencing greater than 2,5 cm/hour rain rates. This class is near and within the highest reflectivity cores inside thunderstorms and rain events.

Big Drops: The big drops classification results when there are a few numbers of very large drops with a lack of smaller drops. This typically occurs near and along the edges of thunderstorms.

Graupel: Graupel occurs when small ice cores high in a thunderstorm fall through a region of super-cooled liquid water, which is an area of liquid water that exists when

temperatures are below freezing. As the small ice particles fall through the super-cooled liquid water the liquid water droplets freeze on contact to the ice core and become small snow pellets, or graupel.

Hail Possibly Mixed with Rain: This category is composed of two situations, pure hail and hail mixed with rain. At this time, there is no distinguishing between small and large hail, but there is some research being done to look into this capability.

Unknown: This class is for the algorithm cannot decisively tell which type of precipitation is in that area. The unknown category holds all radar bins where the following occur: First, if the highest likelihood value from all the possible classifications is below 0.4. Secondly, two or more classifications have likelihood values too close to each other. These situations do not occur in widespread areas, but occur most likely when the dual-polarization data are representative of two or more classifications.

RF: It stands for the range folding. RF occurs when a radar echo from outside the radar's maximum unambiguous range is displayed incorrectly within the radar range. It most likely never affect the HCA product.

2.3. 3D Reconstruction and Visualization

Visualization is a powerful technique of communication in which the information is created in order to highlight the important matters. The rapid growth of capability and scale in computational power has been guided us to a new level of scientific data simulations. Science is no longer interested in proving that simulations can be performed with these calculations. Rather, scientists have begun to create simulations for problem solving, designing, and observation of intriguing environmental variables (e.g. ensemble meteorological visualization, weather nowcasting and forecasting, artificial heart design).

Everyday multivariate and multi-scale datasets are further processed and analyzed to be fully benefitted. In the data processing and analysis phase, visual analytics plays a

vital role in letting researchers explore, explain, examine and predict the behavior of natural processes through integrated and interactive visualization. It also helps them to answer questions arising from the discovery of new concepts, theories, and calculations.

For three-dimensional visualization of the weather radar data, we need to first choose a 3D representation of the reconstruction. Then, the surface should be reconstructed from the 2D radar images. Finally, 3D data should be displayed in an intuitive way. In the following subsections, each subject is visited.

2.3.1. 3D Data Representation

To understand reconstruction, first, users need to begin with describing how a 3D model can be stored in a computer. The understanding of specific objects' appearance in the sensor data context would make it recognizable. Any sensor data context that adequately pictures a real-world object should be functional by a graphics algorithm. There are various object-representation systems for rendering 3D view-independent appearance.

- **Wire-frame representation:** This type of representation offers continuous surfaces by connecting vertex pairs that are stored as a point list and an edge list (Besl & Jain, 1990). These connections can be straight lines or curves. The aim is not to define the area or the volume inside the edges. There is widespread use of wire-frame system because of its simplicity. It also gives an opportunity for the user to construct and manipulate solid surfaces.
- **Constructive solid geometry representation (CSG):** The primitives are the solid objects that have the simplest shapes such as blocks, cylinders, pyramids, cones, and spheres. The objects in the CSG platform consist of primitives within a set of Boolean operators. These operators are known as union, intersection, and difference (Besl & Jain, 1990).
- **Voxel representation:** Voxels are volume-elements or volume-cells. The representation mostly consists of scattered fixed-size cubes into 3-D space.

Although this type of platform is commonly memory intensive, their algorithms are easily applicable. Objects are represented by a list of voxels. They are used mostly in entertainment sector.

- Surface boundary representation (B-Rep): The surfaces are orientable and closed. Combined surfaces become a set of faces of the physical object. Using triangle-faced polyhedron by storing a list of 3-D triangles would be the easiest way to represent boundaries (Besl & Jain, 1990). The number of triangles used holds a desired degree of accuracy. N-sided planar polygons can arbitrarily be replaced to get representation that is slightly more compact. This representation is very common because the surface area and the volume of the object can be specified. Piecewise-planar functions are applicable for object operations (Figure 2.11).
- Multiple 2D projection representation: To provide a clear and complete description of the shape and size of an object a number of views, systematically arranged, are used. To define clearly and completely an object, one 2-D drawing is insufficient. That is why it is appropriate to store a library of 2-D multiple-views (silhouette projections) to indicate 3-D objects. The flat light table offers a number of stable orientations of a 3-D object. To have better recognition, these silhouettes of an object should be different enough.
- Skeleton representation: It is worthwhile to make a shape analysis with the application of skeleton models. The skeleton is a space-curve consists of the spines or axis that is equidistant to objects' boundaries. Skeleton axes underline the topological and geometrical attributes of the 3D shape. It also brings information on length, width, direction, and connectivity. Skeletal shapes appear in literature frequently due to the features of the medial axis (or symmetric axis) transform.

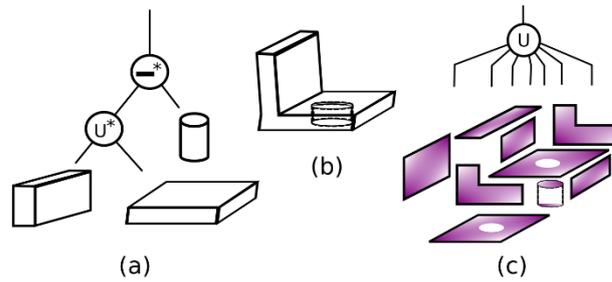


Figure 2.11. Object representation systems; Constructive Solid Geometry (CSG) representation (a), solid (b), a boundary representation (c) (Besl & Jain, 1990)

While choosing an appropriate representation, a unique numerical description of the object is required. Nevertheless, these representations do not assure a single description; instead, the focus is on reorganizing and reordering of the vertices, edges, surfaces, etc. of an object to be able to have a reasonable representation. As a result, the modeler should specify the desired level of detail of the solid objects and maintain the abstract shape.

Regardless of the preference for representation, the surfaces of the object are supposed to be evaluated clearly in one of the visualization systems because both range and intensity images are dependent on surface geometry. In order not to have any unnecessary computation, an explicit surface definition is needed. Scene description consists of feedback loops that are made of data storage mechanism working with vision system automatically. Let us assume a moderate model-based scene is constructed, the entire system would generate a synthetic image and can render it as if the information is taken from a real sensor data. If this process is able to create a better form of the object, this new way of illustration will be a success. On the contrary, once it is not successive illustration, the system will continue to request for a better description (Adan & Huber, 2011; Bhanu & Faugeras, 1984; Shotton, Blake, & Cipolla, 2008; Van Kaick, Hamarneh, Zhang, & Wightton, 2007).

Considering ease of display and better functionality for simulations, wireframe representation is chosen in this thesis.

2.3.2. Surface Reconstruction

Surface reconstruction from various measurements is increasingly being used in many applications areas like cartography, medical imaging, autonomous navigation, archeology, architecture, inverse engineering, etc. The measurements can be very diverse, like: multiple optical images of objects (e.g. stereoscopy), silhouettes of objects from various angles, 3D point cloud (e.g. LIDAR), optical images of the object with changing light (photometric stereo). Each reconstruction method requires a partially different set of skills. In the case of object surface reconstruction from weather radar data, one needs to match shapes, because the sections of an object need to be matched to each other. Because of that reason, relevant work from these domains are reviewed.

2.3.2.1. Shape Matching

Detection of relevant connection between the shapes and matching them is the key issue in many applications such as geometry processing, computer graphics, computer vision, and medical imaging. Matching in medical imaging is mainly based on investigation and composition of anatomy that can be found by point correspondence. Therefore, surface matching can be stated as maximizing the common ground of surface descriptions of shapes after providing their spatial registration. Registration transformations would be the next step because all potential model surfaces/segments are computed to match through the object surface. The control/pivot point selection is vital because in each level, the evaluation of the surface control points would determine the position, orientation and size differences of the surface-segments. Once the surface-segments are met, the merging algorithm creates a new surface for each matched segment.

Winkelbach et al. (2008) have focused on using surface normals in order to reassemble fragments (broken objects) in his research. Without any knowledge of the initial shape of the object, he successfully achieved to match the fragments. This type of 3D puzzle

problem requires some basic mathematical descriptions. First, we need to determine the surface normals of each vertex. As Figure 2.12a depicts, there is a scattered \mathbf{PA} of 3D coordinates $\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_k$ of the first fragment \mathbf{A} and their corresponding surface normals as stated $\mathbf{NA}: \mathbf{n}_1, \mathbf{n}_2, \dots, \mathbf{n}_k$ to that points. The normals are unit vectors that are pointing outward perpendicular to the surface.

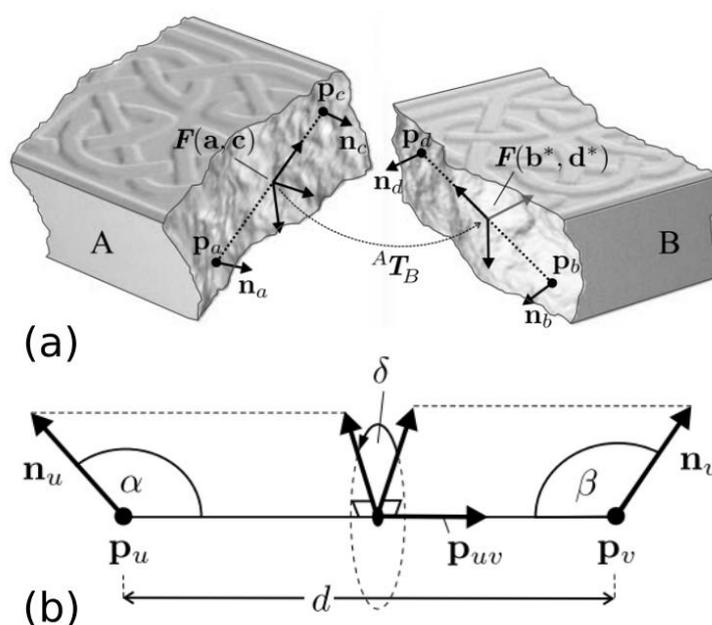


Figure 2.12. Surface matching (a), and the parameters used (b) (Winkelbach & Wahl, 2008)

Computing surface normals is a part of a reconstruction process. Then, the relative pose with the largest surface contact is found. After that, iterative closest point algorithm (ICP) brings out the vertices that would be compared. According to the vertices, the corresponding parameters are defined. These are the distances between each point (d), the angles between the normals and the vectors (α and β) -the unit vector that points each vertex " \mathbf{p}_u and \mathbf{p}_v ", and the signed angle between " \mathbf{n}_u and \mathbf{n}_v " around " \mathbf{p}_{uv} " (δ). Figure 2.12b is a generalized view of the parameters.

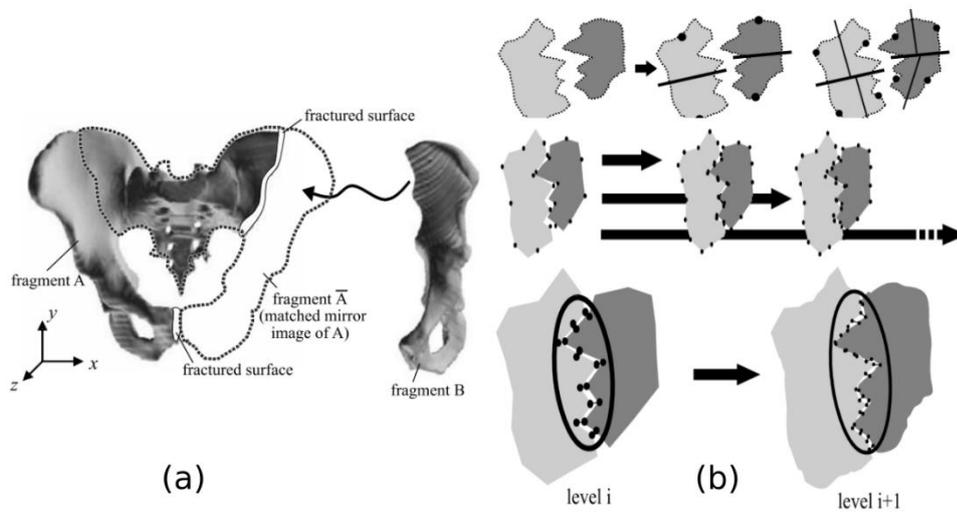


Figure 2.13. Final match; fragment B needs to rotate (a), steps of matching (b)
 (Winkelbach & Wahl, 2008, s. 9)

As the number of contacts increase between the fragments, more robust matching results are acquired. Winkelbach states that two segments are in contact only if there appears more than one vertex, which touches each other tangentially. The final matches are shown in Figure 2.13.

Contour fragments are also an applicable way of finding corresponding matches of shapes. Most of the researchers benefit the internal edges of the objects to describe the contour fragments. Compared to the other matching methods, using contours is more powerful. Because under the light variations the contours still stand accurate. In addition to that, deviation of texture and pose change does not affect contours. Matching object borders is also applicable under varying background. Contour match is frequently used after border tracking and object-background segmentation. The results from these interrelated operations bring out edge maps. Shotton et al. (2008) used local windows instead of the whole image to match contour fragments. Local windows compose of the fragments in a star constellation from an object center called a single fiducial point. These training images are labeled with a bounding box, **b**. The

purpose of using this bounding box is to have information on the scale in order to relate the fragment. The bounding box $b = (b_{tl}, b_{bl})$ that defines

$$\text{the centroid: } \chi = \frac{1}{2}(b_{tl} + b_{bl}) \quad \text{scale: } s = \sqrt{\text{area}(b)}$$

In addition, the information on orientation (rotation) is found by the gradients at the nearest edge calculation. The tangential lines at the edges give the gradients at these points. Moreover, the differences of these gradients would result in the matching edges. If there are minimum angle (gradients) differences, it will state that these are the touching points. Figure 2.14 depicts the parameters clearly.

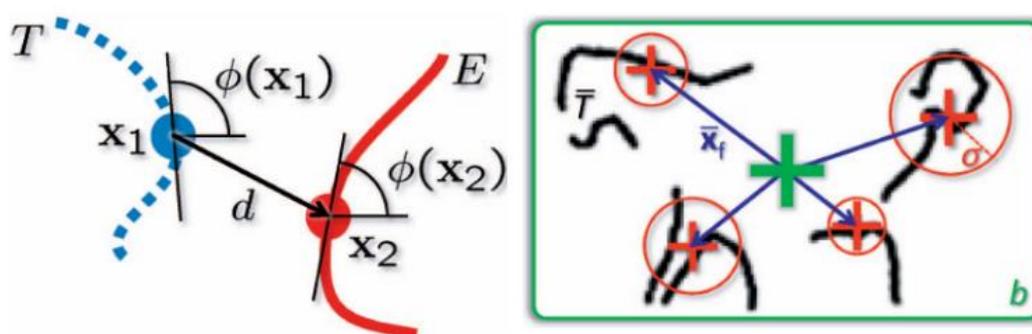


Figure 2.14. Gradients at the vertices for rotation and bounding box for scale (Shotton, Blake, & Cipolla, 2008, s. 1271)

Some shape matching algorithms are based on recognition of segments of objects that are the simple closed curves (approximated polygons) taken from the boundary of the objects. This technique is applicable especially in aerial images and the borders of the biological cells. As stated before, these boundaries are also undergoing rotation, translation, and scaling. Faugeras et al. (1984) began this process by assigning the vertices whose have a higher curvature. Faugeras took advantage of many parameters at the same time such as closest point distance, the entire polygon distance, the slope of each segment, interior angle, and the angle between two segments called exangle. Straight lines are stretched to a meeting point and the angle between them is called exangle (Figure 2.15a). Most of his work of matching segments is based on this angle.

Using exangle together with the other parameters; rotation of the shapes can easily be detected. In Figure 2.15b the model (M1) is composed of first and second object (O1 and O2). These objects form the model. Note that there is also a scale difference of the model and the first object. The algorithm detects the vertices to match the objects with the corresponding models at the determined edges.

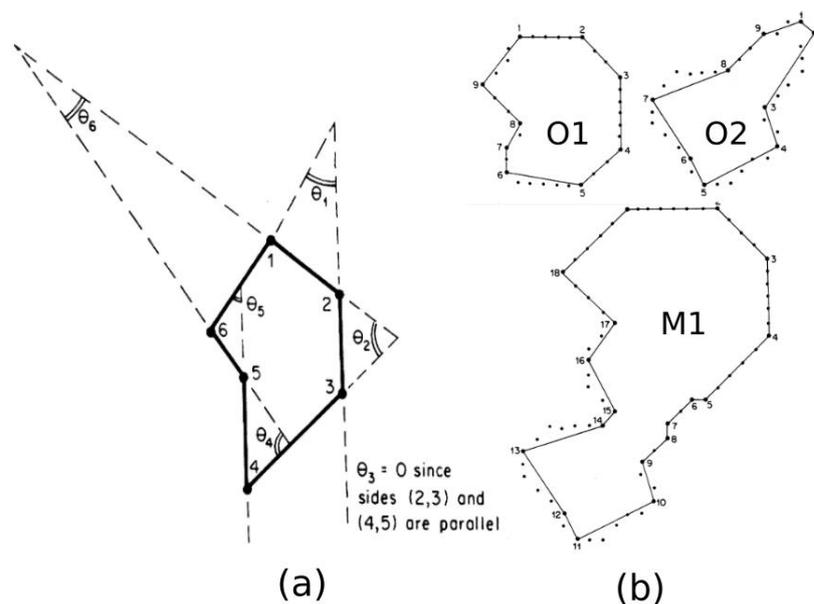


Figure 2.15. The exangle (a), object 1 (O1) and 2 (O2) together will form the model 1 (M1) (b)
 (Bhanu & Faugeras, 1984, s. 140)

It is obvious that the reconstruction of the 3D object successfully via matching projected outlines in a 2D image is possible by point correspondence. Kaick et al. (2007) strengthen the pairwise matching of contours algorithm with proximity information. The fundamental approach of the pairwise vertex matching preserves the vertex order. However, in most of the cases, order preservation technique is not convenient because there are always remaining vertices after pairing. That is why they added an additional condition as proximity information, which is the evaluation of quantitative neighborhood preservation. To be specific, there are points \mathbf{t} from shape

\mathbf{I} and $\hat{\mathbf{j}}$ from shape $\hat{\mathbf{J}}$ are the first vertices of each contour. There must be a close-by neighbor $\hat{\mathbf{i}}'$ of $\hat{\mathbf{i}}$ on \mathbf{I} which is supposed to match with $\hat{\mathbf{j}}'$ on $\hat{\mathbf{J}}$ that is close to $\hat{\mathbf{j}}$. In addition to that, the distance between the vertices must be minimal in each case Figure 2.16 depicts preserving the order only and preserving the order with proximity.

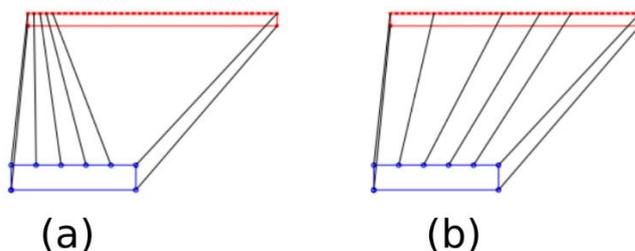


Figure 2.16. Order preservation technique alone (a), preserving order with proximity information (b) (Van Kaick, Hamarneh, Zhang, & Wightton, 2007)

2.3.2.2. Similarity Matching

The techniques we have defined previously for the reconstruction of 3D models are feasible in digitizing and modeling. This has led to an increasing accumulation of 3D models on the internet databases. The new demand is finding a way of searching for a needed object in the dataset. Hilaga et al. (2005) developed a similarity-matching algorithm to explore whether the user can find a compatible model on the internet. This algorithm is based on the topological structure of the object that does not vary subject to resolution. The Reeb Graph technique is a confident path to find a search pattern from the 3D object. Instead of using arbitrary dimensions to determine the topology at the desired level of detail, The Multi Resolution Reeb Graph (MRG) algorithm establishes a certain level. This certainty is described in the paper and is called the function μ . The object is divided into regions based on a height function as a product of μ . Each node becomes the connected components of the object in order to find a connection with the desired object. When the finer level topology meets, the objects are reunited. Figure 2.17 shows the MRG algorithm is applied to a 2D triangle mesh. The height function as an outcome of the μ is represented. The height function

divides each connected component to a new level of resolution. Right after, the nodes are placed and are merged for topology matching. Next step is to have a skeleton of the reconstruction. Finally, nodes are connected and they create the finest resolution.

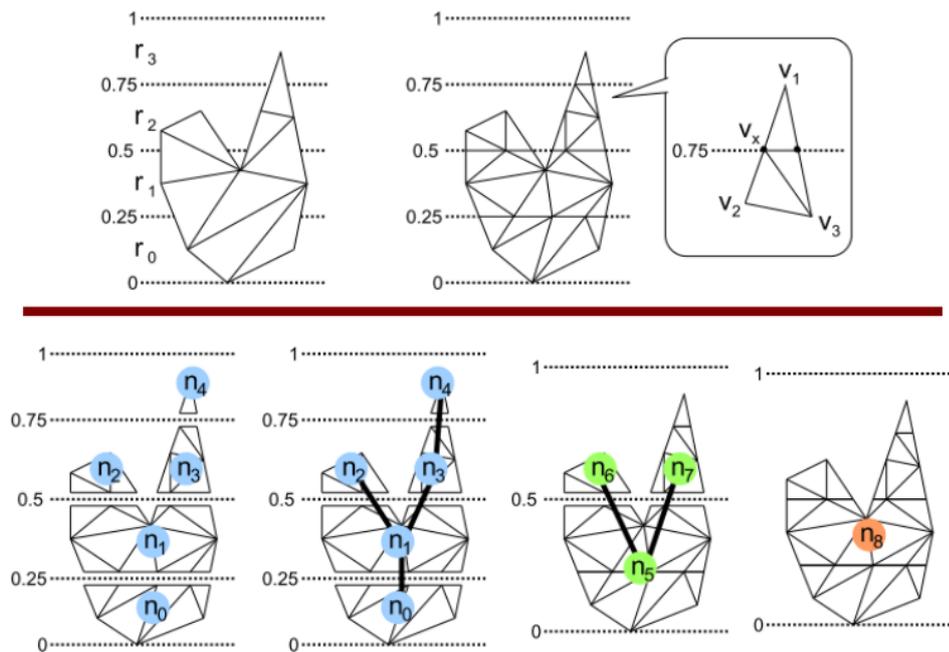


Figure 2.17. Low resolution to higher resolution with the Multi Resolution Reeb Graph (MRG) algorithm (Hilaga, Shinagawa, & Kohmura, 2001, s. 206)

2.3.3. 3D Computer Graphics

3D computer graphics is one of the visualization techniques that are able to use mathematical attributes of any 3D surface of a matter. Computer graphics is increasingly used in many areas like video games, film industry, simulations, scientific visualization, and augmented reality. Because of being so realistic, 3D computer graphics improved the limitations of 2D images. The common elements of the 3D computer graphics are modeling, layout, material, and rendering. The focus of geometric modeling is to have an ability to represent, create and modify shapes. The vertex, edge and face relations are the basic subjects of modeling. In addition, the layout property is responsible for a scene set up by locating and sizing the model.

Moreover, the rasterization may take place whenever the pixel-centres are inside the triangulated model. The vertices replace with the pixels. Another property, material, is investigated when the light hits a material. The ability to control scattering light in desired directions results in different patterns and appearances. Some may appear glossy; on the contrary, some are matte. The realistic outlook is adjustable such as using shiny materials' light attributes; a user is able to fix the incoming light distribution in order to make the object brighter physically. In addition, giving the objects a diffuse effect, they may appear equally bright from all directions like wood. Therefore, the pixel's color can be computed via interaction between the light and materials. Another tool is called the texture. The shader computes color within an auxiliary image that is taken to enclose the model at the desired coordinates. Finally, the 3D scene is transformed into a display via rendering (Buss, 2003).

In Figure 2.18 the geometric model is given on the left. The texture map is in the middle. The assigned parts of the texture are inserted into the corresponding triangles. The result is given on the right.

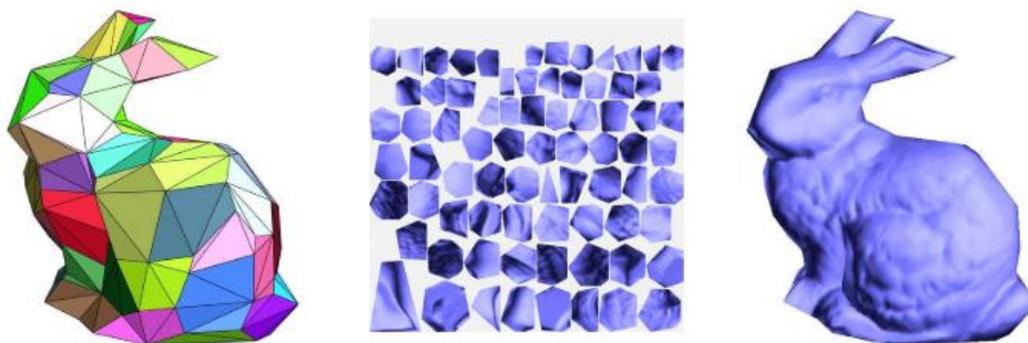


Figure 2.18. Texture is applied on the geometric model (Gortler, 2012, s. 8)

In this thesis, the objects are represented with wireframes and are displayed using the basic computer graphics algorithms.

2.4. Similar Works

The last few decades, the 3D scientific visualization has begun with visual programming systems for mostly commercial applications. These systems that have been used in the atmospheric sciences are able to render higher dimension data sets. However, radar data visualization is still limited to projections of data on 2D-plane.

Most of the software programs that represent the meteorological quantities using 3D rasters (voxels) require a lot of interpolations from real data and cause distortions of 3D structures of convective clouds. Instead, Lu et al. (2018) represented the data using logical objects named radar-point, radar-line, radar sector and radar-cluster (Figure 2.19). The mathematical expression of the logical structure is given in the following equations.

Let the rPoint consists of coordinates and attributes; denoted as x, y, z and A ,

$$\text{rPoint} = P(x, y, z, A) \quad (1)$$

Let the rLine consists of rPoints and attributes; denoted as LI ,

$$\text{rLine} = L(\text{rPoints}, LI) \quad (2)$$

Let the rSector consists of rLines and attributes; denoted as SI ,

$$\text{rSector} = S(\text{rLines}, SI) \quad (3)$$

Let the rCluster consists of rSectors and attributes; denoted as CI .

$$\text{rCluster} = C(\text{rSectors}, CI) \quad (4)$$

Based on the raw radar data, the number of rLines differs in rSectors. That is why rLines needs to be stabilized. Therefore, either existing extra rLines are eliminated or new rLines are generated by interpolation. At the beginning, the continuous volume is built by shifting volumetric cells that are composed of discrete rPoints. This is done by constructing a hexahedron via connecting four consecutive points from the upper

and the lower elevation. These hexahedrons are identified as one special cube grid (Lu, Chen, Wang, Yu, & Jiang, 2018).

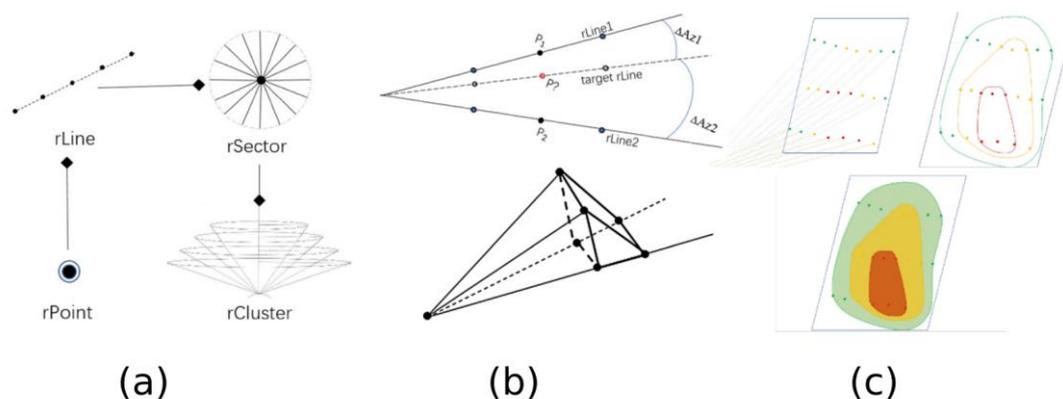


Figure 2.19. Logical structures to represent 3D data (a), interpolation and construction of triangular prism instead of hexagonal shapes (b), cutting plane and profile from contours (c) (Lu, Chen, Wang, Yu, & Jiang, 2018, s. 804)

The marching cubes (MC) algorithm is used to form volumetric reflectivity data (Figure 2.20). The final interpolation is performed while exploring the inner composition of the modeled objects. Arbitrary cutting plane is taken and intersection points are calculated. Then, the points that lay on the plane becomes contours of the objects with different colors with respect to the reflectivity values (Lu, Chen, Wang, Yu, & Jiang, 2018). The method is applied only to reflectivity data.

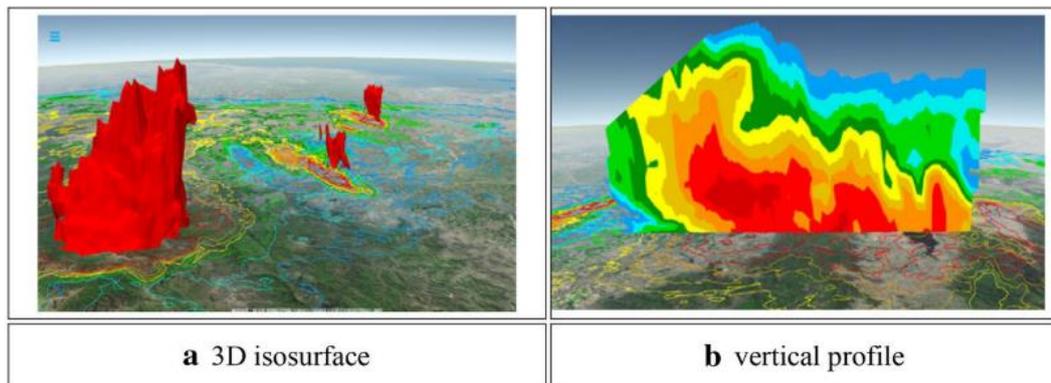


Figure 2.20. 3D structure of observed convective cloud by Marching Cubes (MC) Algorithm; 3D visualization (a), vertical profile (b) (Lu, Chen, Wang, Yu, & Jiang, 2018, s. 804)

Ensemble forecasting is an operation practiced in Numerical Weather Prediction (NWP). Rautenhaus et al. (2015) studies on 3D visualization of ensemble forecasting that focuses on identifying the stability, distinction, and the uniformity among the combination of instruments and their products. The visualization tool (Met.3D) is built up to assist NWP during aircraft-based possible future states of the atmosphere (Rautenhaus, Kern, Schäfler, & Westerman, 2015). Moreover, it is also suitable for research and instruction. The raw datasets are taken from European Centre for Medium Range Weather Forecasts (ECMWF) and Integrated Forecast System (IFS). Met.3D lets the user validate location-based forecast scenarios by navigating through ensemble components at the same grid applying statistical properties such as mean, probability, local minima/maxima or standard deviation. Then, these measures can be visualized with color maps, transparency, texture, and animation. For instance, thresholding wind speed and a cloud cover fraction above or below some level can be done interactively. Example images are shown in Figure 2.21.

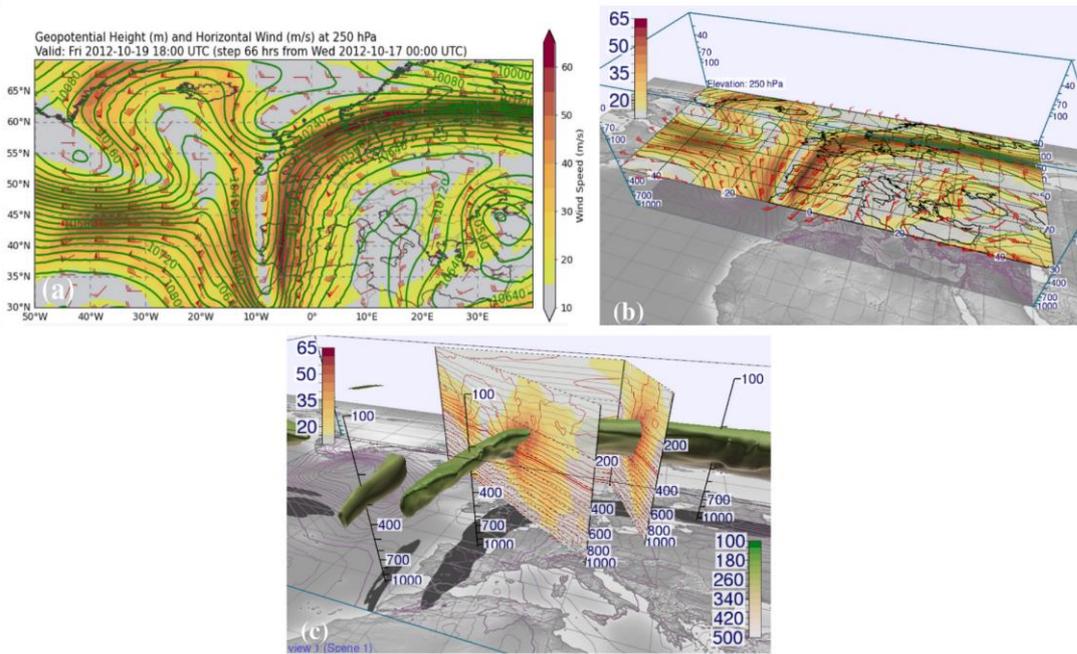


Figure 2.21. Horizontal section of geopotential heights with the contours and wind with the color (a), same data rendered by Met.3D and mapped (b), 3D isosurface representation with thresholding (c) (Rautenhaus, Kern, Schäfler, & Westerman, 2015, s. 2336)

One of the most influential commercial products that can perform 3D visualization of weather Radar images is GR2Analyst. 3D visualization is applied to the NEXRAD Level II base moments with an isosurface process. It enables users to quantify possible storm and/or hail initiations in 3D. With the help of high-resolution maps, the location-based analysis of weather radar data depicts threats and because of its speed and efficiency, the storm path can be displayed in real-time. An example image of GR2Analyst 3D rendering is shown in Figure 2.22.

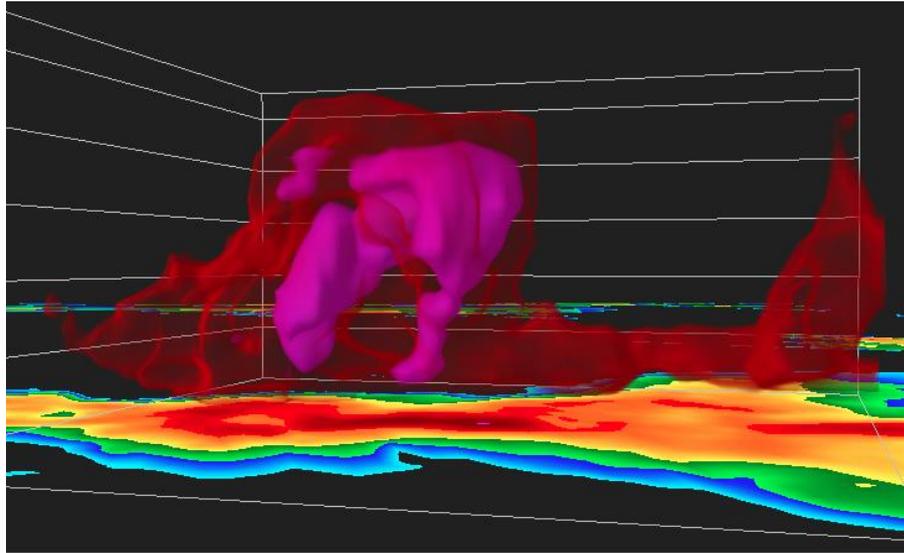


Figure 2.22. Volume reflectivity for the September 22, 2005 hailstorm along the Michigan/Indiana border (Gibson, 2014)

CHAPTER 3

3D RECONSTRUCTION OF WEATHER RADAR IMAGES

3.1. Outline of the Algorithm

The algorithm proposed in this thesis reconstructs the 3D geometry of volumes from radar images that are actually sections of hydrometeors in the air at each elevation angle. The 2D radar images contain labels of hydrometeors. The constructed 3D shapes are abstractions of complicated weather events. Once each time instant is reconstructed, an animation in time can also be obtained. The flow diagram of the reconstruction algorithm is shown in Figure 3.1.

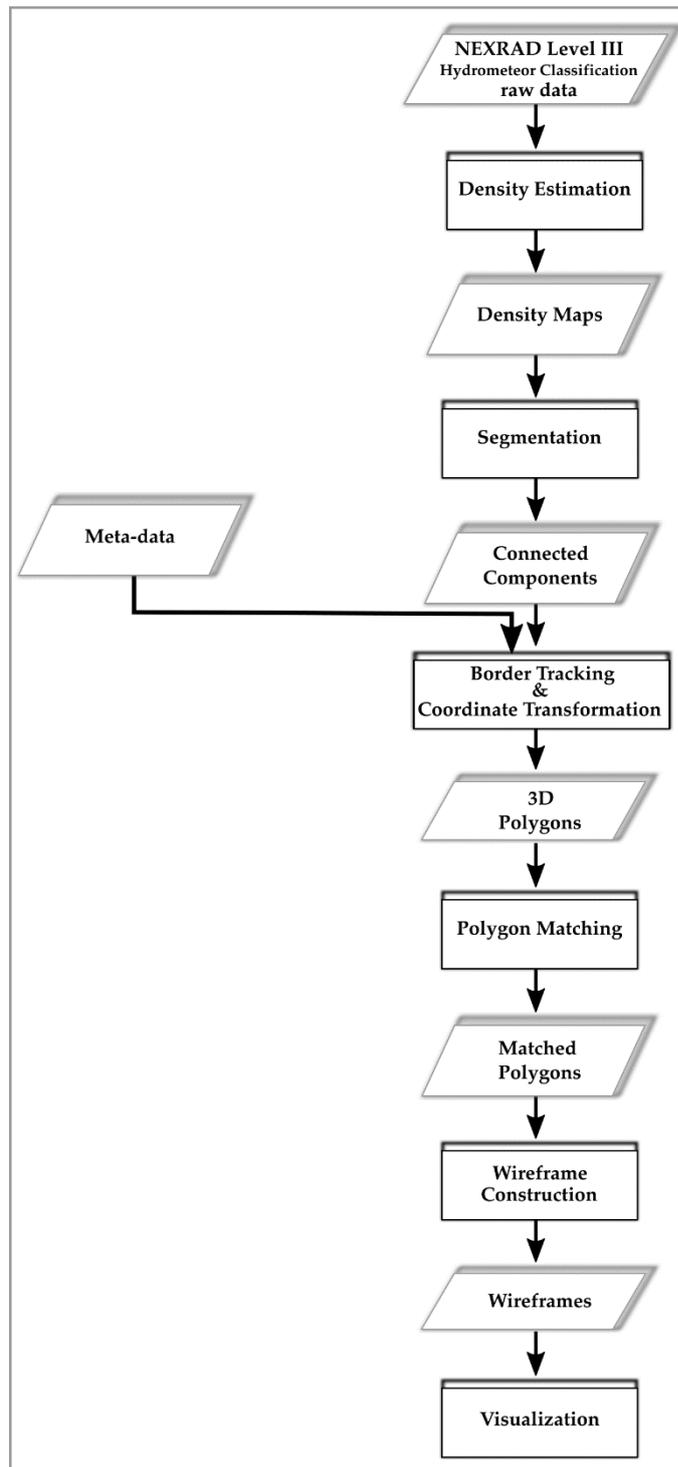


Figure 3.1. The flowchart of the proposed algorithm (rectangles denote processes while parallelograms denote data)

The algorithm begins with estimating the density of each hydrometeor type. Figure 3.2 shows density estimates for four different hydrometeor types obtained from the raw image. In the raw image, gray shows the background and other colors show various hydrometer types. The density is represented by hot colors on the dark blue background. The hydrometeor types are (1) unknown classification, (2) big drops rain, (3) light and/or moderate rain, (4) dry snow, and (5) ice crystals. The details of this part of the algorithm is given in Subsection 3.2.3.

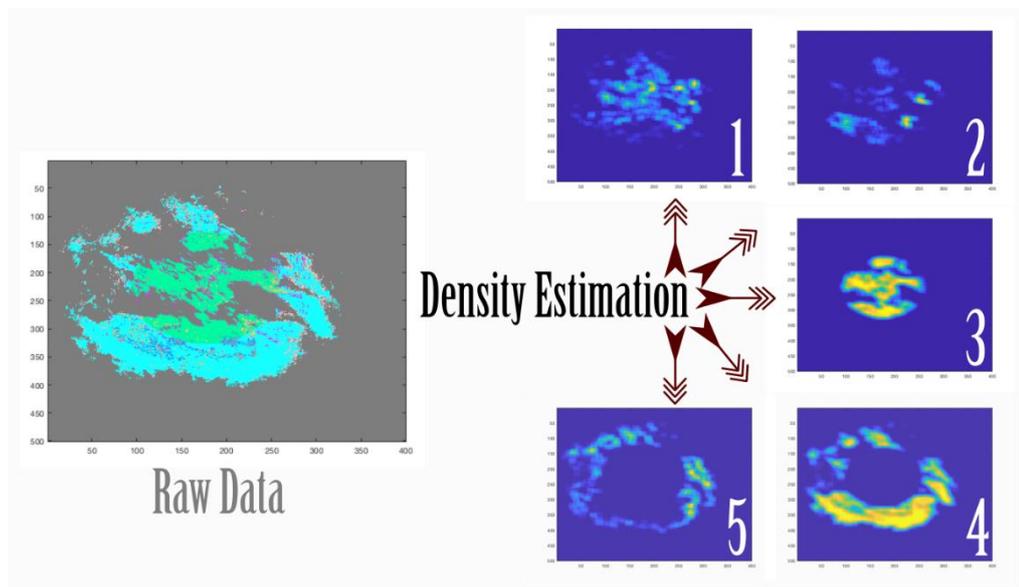


Figure 3.2. Density estimation process and density maps of labels (1, 2, 3, 4, 5)

Once the densities of each type is found, it is possible to find the dominant hydrometeor type in each pixel by choosing the one with the highest density, provided that the level is above a threshold. Then, connected component analysis is performed on the resultant classification image, to identify the pixel that belong to the same object. Each label in the resultant image (shown with different colors) show a distinct segment. Figure 3.3 shows the segmentation process and resultant six connected components belonging only to the dry ice type hydrometeor class. The details of this part of the algorithm is given in Subsection 3.2.5.

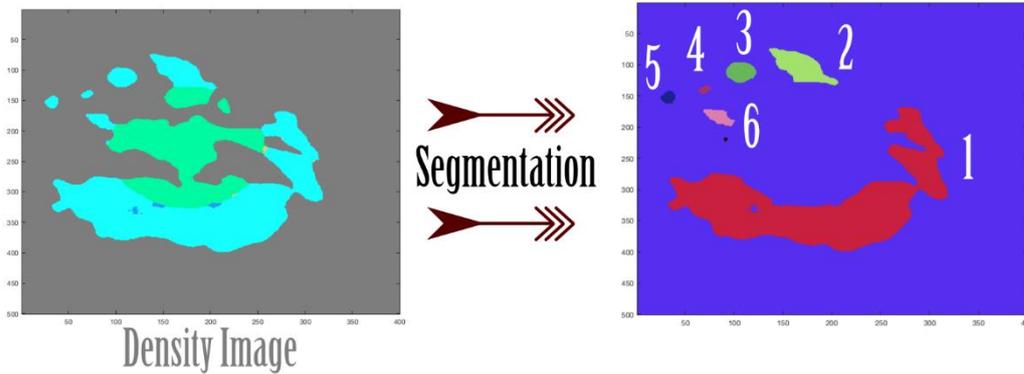


Figure 3.3. Segmentation for dry ice (blue) and resultant connected components (1, 2, 3, 4, 5, 6)

After sections from different objects are segmented, the boundaries of the connected components of the same label are obtained in the form of polygons. With the transformation of the image coordinate system into the world coordinate system, the polygons are transferred to the 3D space. Another algorithm matches the polygons from different levels that are sections from the same object. Figure 3.4 depicts the segments of the label “dry ice” from the "0.9" elevation angle after having been matched with "1.5" elevation angle polygons. Although "1,2,3,6" have found matches at the corresponding elevation, the "4 and 5" could not find any related matches. The details of this part of the algorithm is explained in Subsection 3.4.

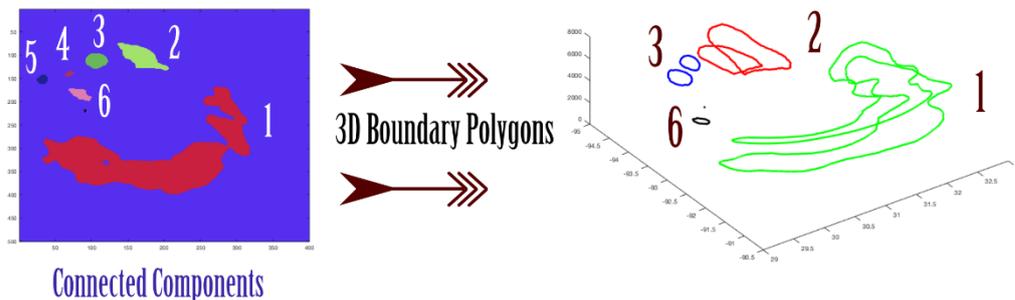


Figure 3.4. Extraction and then matching of boundary polygons

Finally, the matched polygons are converted to wire-meshes so that they can be displayed interactively using computer graphics algorithms. This part is explained in Subsection 3.5.

3.1.1. Development Platform

MATLAB™ (MATrix LABoratory) has its own programming language within a special editor that evaluates codes. Input and output functions are available with explanation for programmers. Computations can be applied both in matrix and vector calculus very efficiently and accurately. Graphical interface is built for interaction and simplicity (Altius Directory, 2019).

In all possible data dimensions, "figure" and "axes" properties direct users to customized MATLAB™ graphs that is why it is very preferable for displaying datasets. Since 3D data has a rare accessible environment in 2D computer screens, MATLAB™ applies very unique techniques for visualization such as structuring slices and meshes, or using transparency attributes. There are two types of volumetric data in MATLAB™. Either data is scalar (only points with magnitude and defined grid) or vector (magnitude and direction). MATLAB™ examples of functions for displaying scalar data are isosurface, trisurface, slice plane and contour slice. For vector data, it is possible to use streamlines and cone plots (Majumdar, 2012). In addition, combinations of these functions are possible.

3.1.2. Graphical User Interface

The purpose of Graphical User Interfaces (GUIs) is automating of a task or a calculation. Point-and-click controls mainly consists of pop-ups, sliders, buttons, menus that enable users run software applications with no required interaction of a coding language knowledge. For building the GUI, MATLAB™ GUIDE (GUI development environment) was chosen, which make a powerful graphical design possible (Mathworks Inc, 2019). In addition to that, GUIDE's interactive development environment provides coding applications in the editor. The main screen of the Toolkit named Meteorology3D is displayed in Figure 3.5.

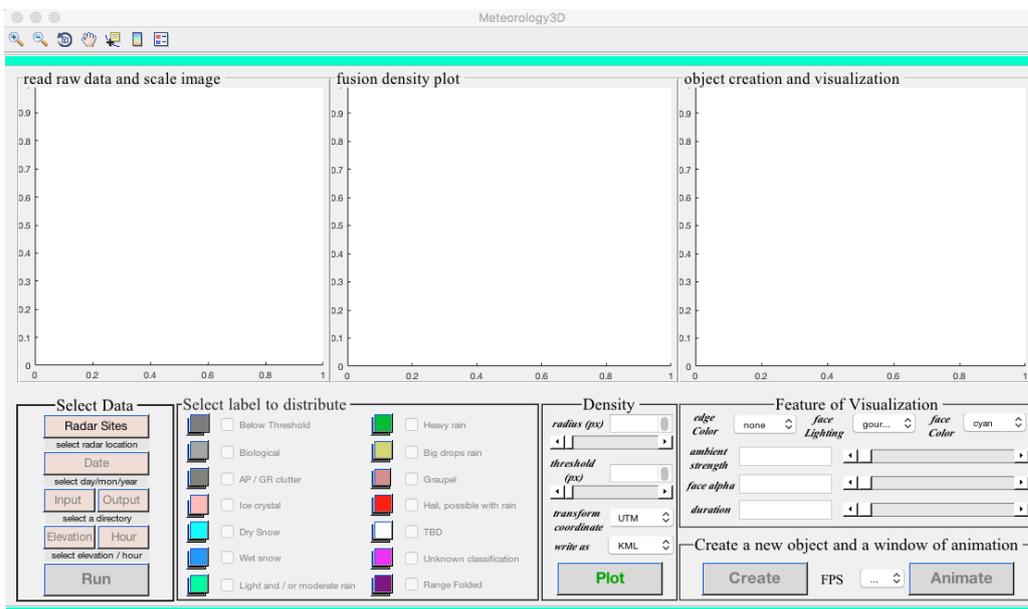


Figure 3.5. Meteorology3D GUI Interface Design

3.2. Segmentation of Hydrometeor Images

3.2.1. Image Segmentation

Image segmentation operation takes place in the scale of pixels or voxels. The digital images can be split into distinct regions to identify objects or other relevant information. These multiple sub-images have the similar attributes, like color or texture (Higgins, 1996). Since there is huge diversity in input data, information that needs to be extracted from an image should be easier to examine and exact to locate. On the other hand, image segmentation is able to collect clusters of pixels no matter what type of distribution (even with the chaotic distributions) or input format they have (Boiangiu & Ioanitescu, 2013). The most frequently used segmentation algorithms are given as in the following:

Histogram Based Segmentation takes an advantage of the histogram of an image in which there are clusters of pixels. Most of the images having separable background

from the foreground use this particular algorithm. In each particular cluster, its local minima and maxima are chosen as the boundaries (Boiangiu & Ioanitescu, 2013).

Region Growing Based Segmentation is a rather effortless algorithm that is applicable to the images, which do not have much varying color elements. It begins with choosing a random pixel and letting this pixel combine with neighboring pixels as the segment grows. When these growing pixels meet non-homogeneity, they become segmented objects in the image (Boiangiu & Ioanitescu, 2013).

Graph Based Segmentation uses the neighboring pixel calculations. As a summary, a graph is built in which pixels are represented as nodes and edges are associated with the neighboring pixel dissimilarity numerically. The segmentation occurs by partitioning the graph with respect to weights between nodes (Felzenszwalb & Huttenlocher, 2004). Edges between two vertices in the same component should have lower weights. On the contrary, edges between vertices across different components should have higher weights.

Watershed Based Segmentation decides whether the clusters of pixels are catchment basins or watershed lines by viewing the image as topography. As the local minima of the basins start to grow, they approach to each other. The touching point would require dams, which stop mixing particular segments. That determines the lines of separation. Therefore, the catchment basins are the objects that can be identified (Beucher, 1992).

Mean Shift Segmentation begins with the conversion of the image into feature space. The spaces L^*u^*v and L^*a^*b are designed in the aim of approximate perceptually uniform color spaces. "L" stands for the Lightness (relative brightness), and remaining coordinates differ only through chromatic adaptation. For each specific cluster, mean of the data samples are computed. The intensities are obtained by shifting the mean values. These procedures keep up until there is no convergence left (Stackoverflow, 2011; Comaniciu & Meer, 2002).

3.2.2. Overview of the Segmentation Algorithm

This thesis takes an advantage of sliding-window (kernel) technique. Quantitative analyses are carried out on the scale of individual pixels that are labeled with precipitation type information. Figure 3.6 is an example of raw data. Depending on the radar site, the size of the raw image varies. In Figure 3.6, the size of the raw data is 800×656 . The size of 24×18 image is cropped from the specific part of the raw data. The numeric representation of the cropped image is demonstrated (same size as 24×18). These numbers in the range zero to sixteen are the hydrometeor labels. As it can be seen in the example raw image, some cells in the radar image are empty while some areas are quite mixed. In the original data, some pixels have the code of a defined precipitation class, while some pixels do not belong to any hydrometeor type. Apart from background pixels, these spurious pixels are mostly distributed irregularly and rarely located in the images. They might be erroneous pixels due to range folding or ground cluttering. Moreover, some pixels are not able to pass the thresholds and there might be others, which cannot be classified. However, the labels between three to 10 correspond to the hydrometeor types.

The density estimation algorithm is developed as the first step. The smoothness of density estimations is also controllable via the size of the kernel. Then the type with the largest density is chosen provided that it is above a threshold. Finally, Connected Component Analysis algorithm creates the segments, which are the building blocks of the following stages. The mentioned techniques are explained in the following subsections.

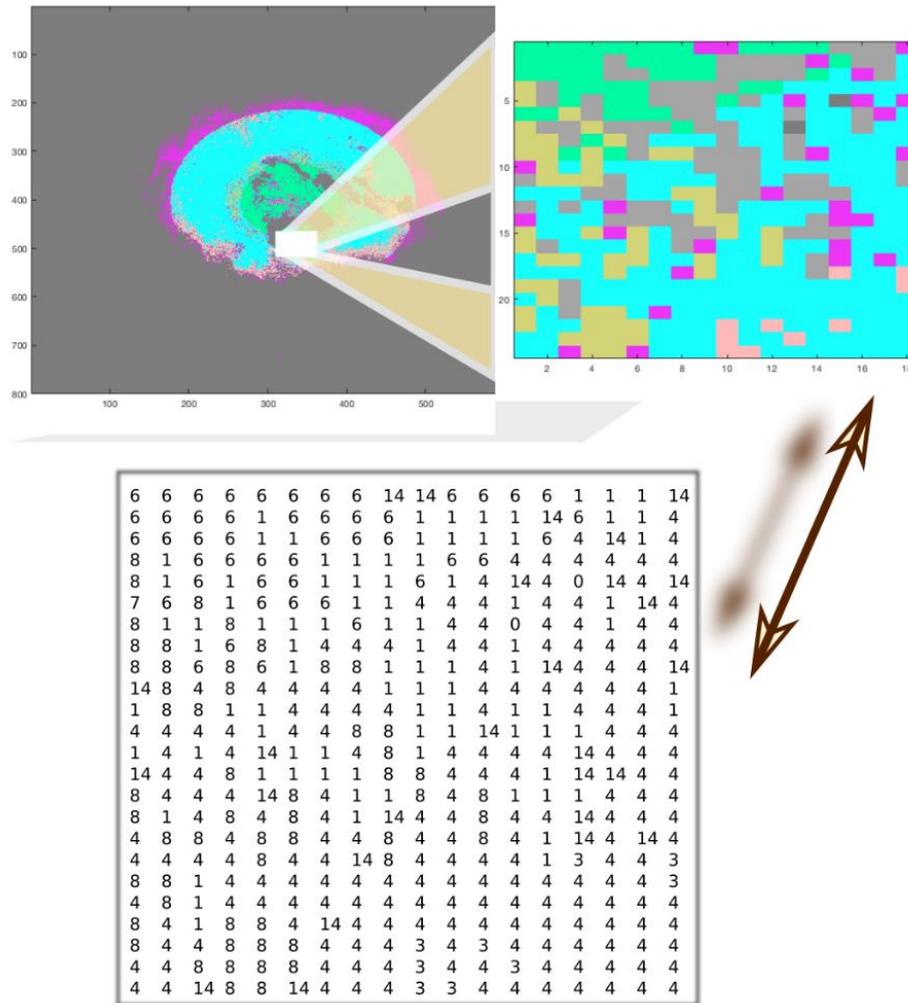


Figure 3.6. NEXRAD raw data, cropped image, label representation

3.2.3. Density Estimation

In the raw image, any segmentation attempt would lead to a highly over-segmented image with a grainy look. It is necessary to estimate the intensity of each type of precipitation as a continuous and smooth function. The significance of the developed algorithm is better understood by analyzing the image pixel-wise. Figure 3.7 shows a cropped raw image with the size of 24×18 . Each label of the pixels can be seen clearly. Assuming that the direct segmentation is applied to the image that have only the label eight; namely, big drops rain type precipitation. In Figure 3.7a labeled pixels are shown with the yellow color and the number eight is written on them. The label

eight is chosen for this test because this label is sparse with respect to other colors. After segmentation, the resulted image Figure 3.7b compose of segments of pixels with only label eight and background pixels. It is clear that most of the segments are formed of not more than six or seven pixels. As a result, the segments with this number of pixels are so small when it is compared with the raw image size. That is why this process is not suitable and workable for connected component analysis.

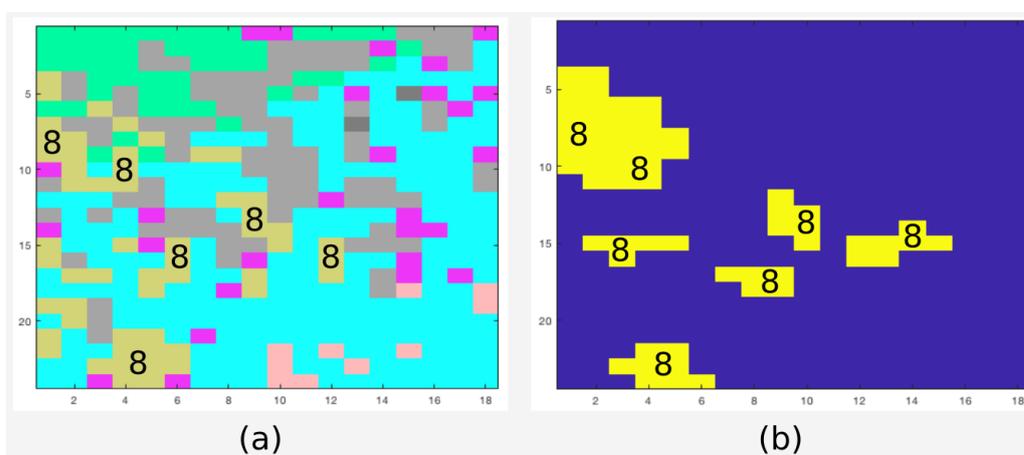


Figure 3.7. Cropped raw image (a), segmentation applied only for label 8 (b)

It is necessary to obtain the segments of each desired hydrometer type as a continuous and smooth image. For this purpose, the labels in each pixel are estimated as the number of pixels containing the same type in a given neighborhood. Naturally, the neighborhood is normalized by the size of the sliding window. For ease of application, the square filter is selected (Figure 3.8). As the filter size increases, the intensity function gets smoother. Therefore, it is possible to control the surface smoothness of the final segments by the size of the window. This method is similar to kernel density estimation (Silverman, 2018) of probability density functions. The overall algorithm is given as Algorithm1.

Algorithm1: Intensity estimate

Input:

- A hydrometeor image from a specific radar site (image)
- The radius of the window (r)

Output:

The 3D image $\text{den}(x, y, \text{label_id})$ with calculated hydrometeor intensities

Algorithm:

- for each label (lb) in the image
 - for each location (x,y)
 - Let c = number of lb's in the image in a neighborhood of $(2r+1) \times (2r+1)$ around (x,y)
 - Let $\text{den}(x, y, \text{lb}) = c / ((2r+1) \times (2r+1))$
-

The parameter of the density estimation is the radius (r) of the filter. The result is normalized with the number of elements of the filter. The resultant images (density maps) represent the density of each hydrometeor type as a smooth function. In Figure 3.8, the radius is chosen as two pixels. Therefore, the number of elements is $(2r + 1)^2 = 25$ pixels. Considering the filter, sample density maps are given for five labels. Within the samples, for instance, one of the missing labels in the filter is six (Light and/or Moderate rain type). That is why the density map for the label six has zero value in that particular position. There are six eights and fours, 10 ones, three 14 in the filter. The ratios of the number of same labeled neighbors are 0.24 (6/25), 0.4 (10/25), 0.12 (3/25) respectively. The final density images are multiplied with 10.000 so that they can be represented in sixteen bits for memory efficiency. The labels are 4, 8, 1, 14 and 6; dry snow, big drops rain, biological, unknown classification, and light and/or moderate rain, respectively.

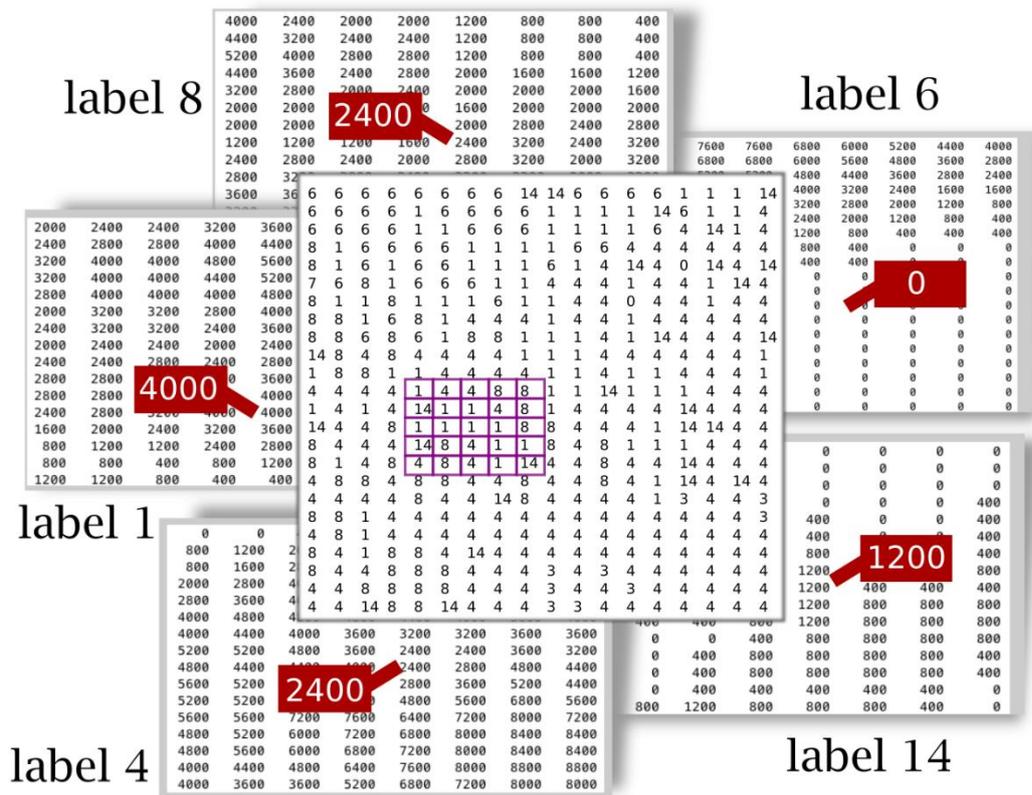


Figure 3.8. Density calculation and density maps

3.2.4. Thresholding

Although the raw image is already classified, the labels are too irregular for being represented geometrically. Once, smooth density maps are obtained for each class, the class with maximum density can be assigned to the corresponding pixel. However, if the density of a class is too small, that pixel should be left as blank. Therefore, the class with maximum density is chosen if the density is above a threshold. This way, the image has labels of a hydrometer when there is a dense and dominant type in an area. The corresponding algorithm is given in Algorithm2.

Algorithm2: Thresholding

Input:

- The density image $den(x,y,lb)$
- The threshold value (T) in percentage

Output:

- labeled image $hydro(x,y)$

Algorithm:

- $hydro(x,y) = \begin{cases} nolabel, & \max(den(lb|x,y)) < T \\ argmax(den(lb|x,y)), & otherwise \end{cases}$
-

In Figure 3.9a, the cropped image and the applied density calculations to the corresponding labels are shown for the image in Figure 3.8. For convenience, the threshold is expressed as percentage of the maximum value.

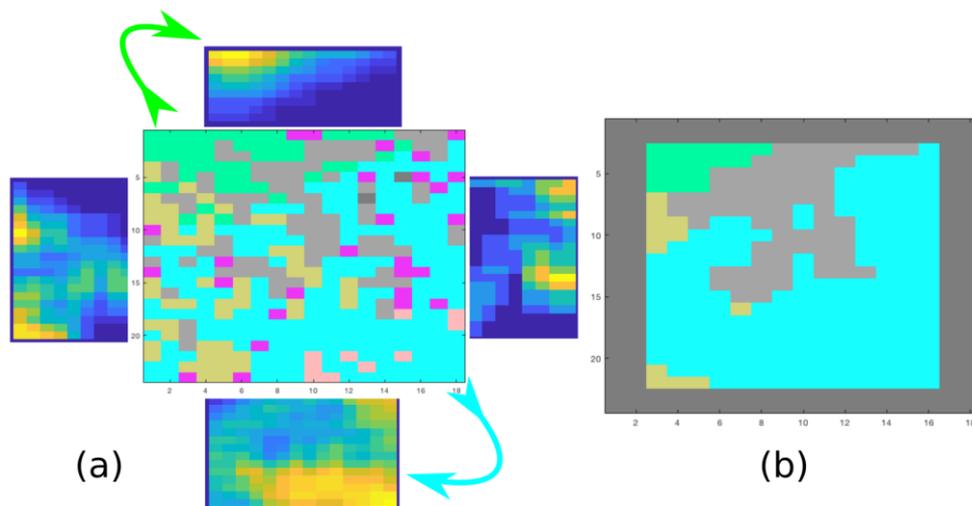


Figure 3.9. Density maps (a), after applying 10% threshold (b)

3.2.5. Connected Component Analysis

Interconnected pixels with the same label are actually a slice of the three-dimensional object cut by the imaging surface. The Connected Component Analysis (CCA) changes the unit of analysis from pixel to region or segment. CCA assigns a unique identifier to each connected region. Figure 3.10 shows the label that belongs to biological (insects, birds, etc.) class in a small raw image on top left. In addition, the binary numbers of this label are shown along with the boundary lines of the connected components on top right. The resultant image is shown on down left using different colors for each connected component, that is, each segment.

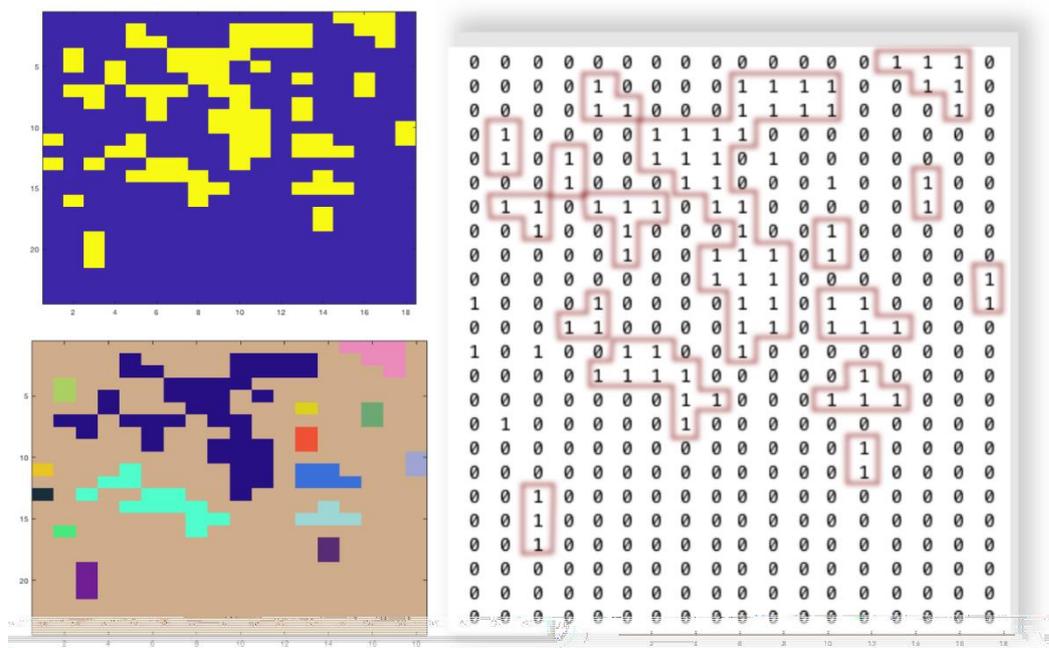


Figure 3.10. Connected component analysis

Two parameters in the process, that is, the radius of the kernel and the density threshold has strong effect on the shapes of the resultant segments. If the threshold is too high, the number and size of the segments are too small. On the other hand, if the

threshold is too low, the segments are larger than required and segments from different objects may merge. In the case of the kernel radius, a too small value results in too many small segments, while a too large radius results in extreme smoothing, hence, loss of details. That is why after many trials in the process of reconstruction of image, the ideal default radius is selected as 15 pixels and the ideal threshold is determined as 10% as shown in Figure 3.11.

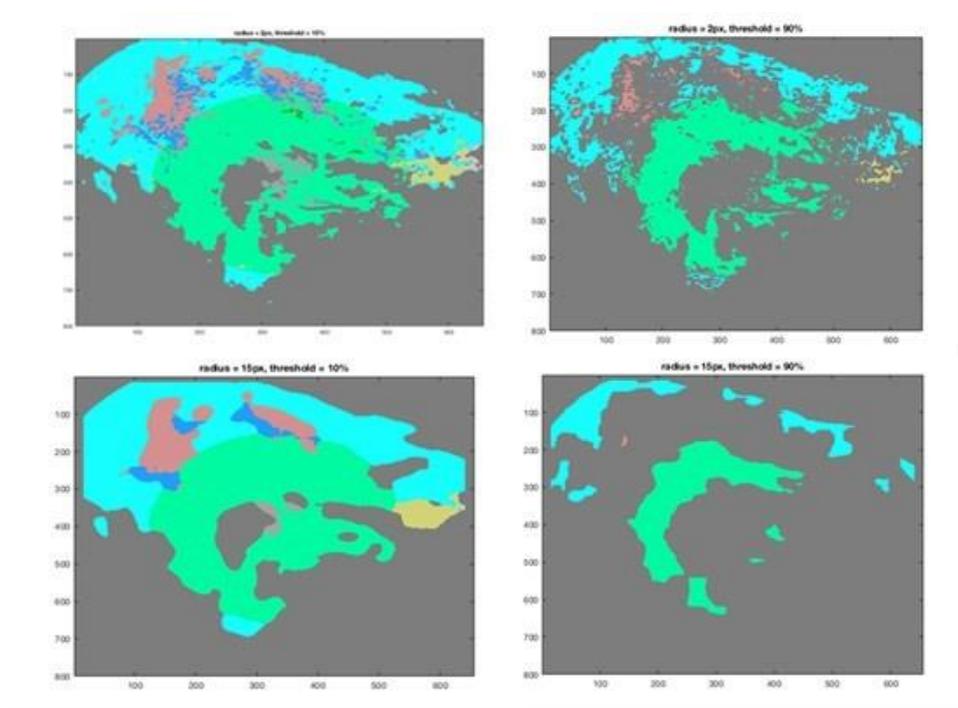


Figure 3.11. Top images have radius of 2px, threshold 10% (left), 90% (right); Bottom images have radius of 15px, threshold 10% (left), 90% (right)

3.2.6. Segmentation in the GUI

The segmentation part of the GUI consists of three main panels as shown in Figure 3.12: "Select Data", "Select label to distribute", and "Density". First, "Select Data" panel is made of pop-ups to select data sets one by one in a specific order. Among all NEXRAD sites, the user should first specify which radar site is required. Then, the date option appears and should be selected in the calendar. The user should also

identify in which directory the Meteorology3D Toolkit is. The output entry is used in later steps. Elevation angle needs to be defined. At last, the time needs to be stated by the user in the “Hour” entry. When the “Run” button gets green, the toolkit is ready to show up the raw data in the "raw data and scale image" axes. The resultant image shows a raw data of hydrometeor classes with colors. Moreover, “Data Cursor “button at the top left of the toolkit can help identify the label values of the classes. The second step for complete segmentation is to decide the radius and threshold values in the "Density" section. The sliders help users to choose both radius threshold values. Currently, “transform coordinate” and “write as” selections are fixed. Finally, when the "Plot" button gets green, the raw image will ready to be segmented. Second axis will display color image according to the legends in the panel.

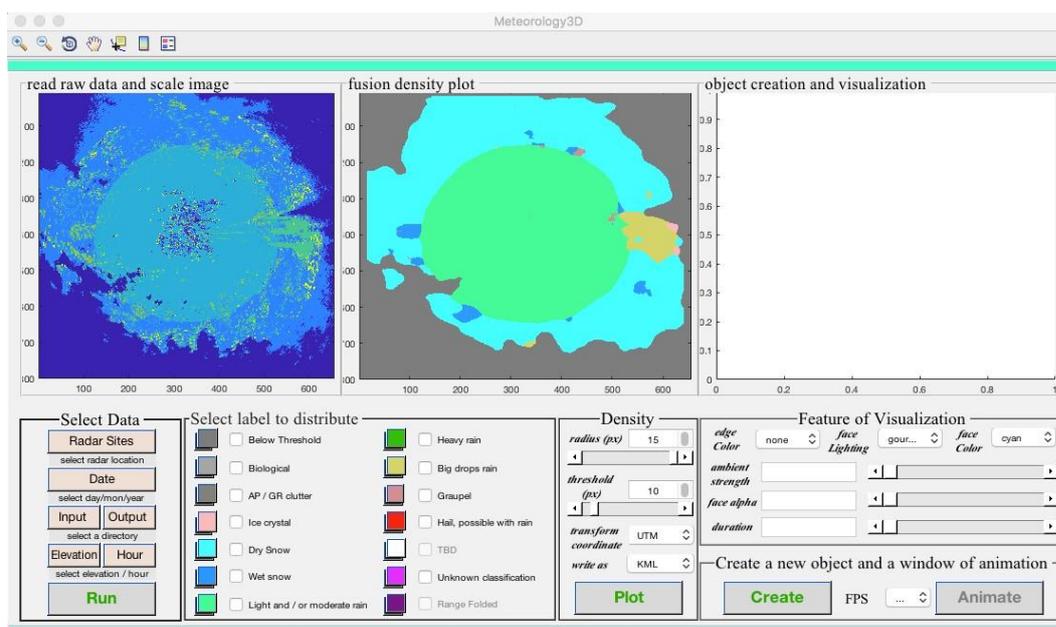


Figure 3.12. Segmentation in GUI panel of Meteorology3D

3.3. Three-dimensional Boundary Polygons

After segmentation, the next step is to have the external characteristics of each object, which is its boundary. When the boundary is drawn for a cluster of pixels, the shape

of an object is identified and can be processed by the computer (R. C. Gonzalez, 2009). Hence, the extraction of the borders of the components (also known as boundary or contour) is a fundamental tool for object recognition. The border is defined as the set of pixels that belong to an object, having one or more non-object pixel neighbors. The pixels that are not the object's portion are known as background pixels; on the contrary, the object's pixels are called foreground pixels in a binary image.

In the proposed algorithm, each segment's boundary is extracted first by boundary following. Then, each 2D boundary polygon is converted to 3D world coordinates. Finally, these 3D polygons are exported to "kml" files for viewing in programs like Google Earth. Each of these steps are explained in following subsections.

3.3.1. Border Tracking

Binary images are composed of black and white pixels. The boundary tracing algorithms should be precise, that is, they should not miss any corner pixel. The algorithm should be fast enough to be applicable for all the variations of an input image.

Before explaining the Thesis' approach known as Moore-neighbor tracing (MNT) algorithm, some common algorithms are explained in the following: (1) simple boundary follower (SBF); (2) modified SBF; (3) improved SBF; (4) the radial sweep algorithm (RSA) and (5) Theo Pavlidis algorithm (TPA) (Seo, Chae, Shim, Kim, & Cheong, 2016).

The basics of pixel-following algorithms are laid by Seymour A. Papert while creating the LOGO program (Solomon & Seymour, 1976). The algorithm is commonly known as Papert's turtle algorithm. It is about a turtle at the beginning of a room that needs to get through the end of the room containing objects distributed randomly. The turtle advances straight. When it faces up with an object, it needs to change direction with crawling and it needs to advance again. This crawling is specified as the directional change. Therefore, while giving an example on tracing a boundary, it is suggested that the following lines need to be thought of like a turtle's advance. Consequently, what

is required are the starting pixels' coordinates, the tracing direction and the next border pixel coordinate. Directions are specified as absolute or relative. The contour pixels are classified according to the local pattern of the contour as shown in Figure 3.13. They are straight line, inner corner pixel, outer corner pixel, and inner-outer corner pixel.

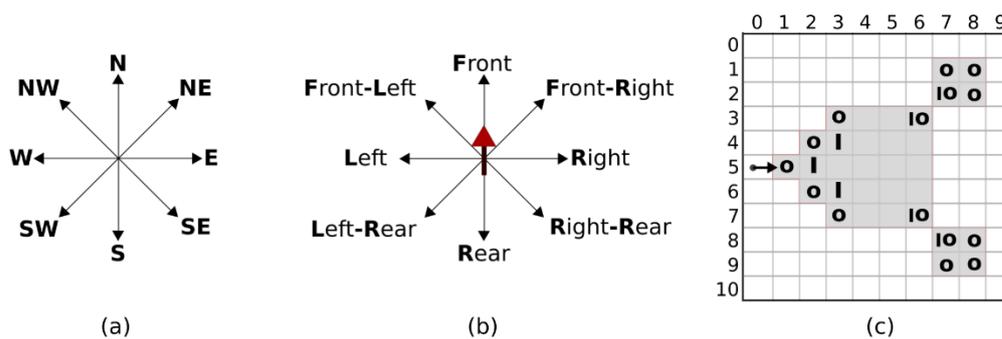


Figure 3.13. Directional information of contour pixels; absolute direction (a), relative direction (b) and (c) types of corner pixels (I: inner, O: Outer, IO: Inner outer) (Seo, Chae, Shim, Kim, & Cheong, 2016)

- Simple Boundary Follower

The initial position of the tracer is stored. The tracer moves either left or right. Once the tracer coincides with an object, it moves left. If the next pixel is the background, it moves right. The tracking direction is displayed in Figure 3.14.

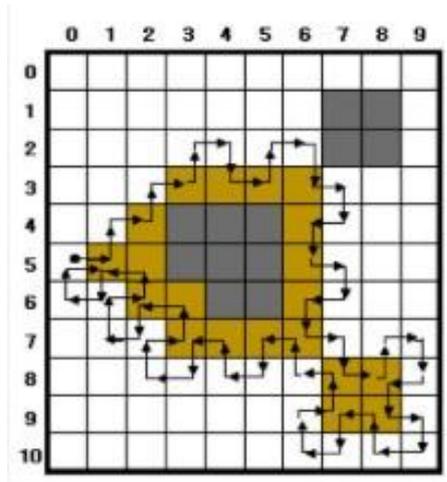


Figure 3.14. Simple Boundary Follower track (Seo, Chae, Shim, Kim, & Cheong, 2016)

- Modified Simple Boundary Follower

As it can be seen from Figure 3.15 SBF misses the inner-outer pixels in the left rear. The main purpose of MSBF is to detect these pixels. The additional condition is this: When there is a black pixel in the left rear, it checks if there are white pixels at both left and rear. Once these conditions are met, the next step is to go left-rear pixel. If the advancing pixel is alone, in order to get rid of a loop the tracer will move back to the same pixel and keeps the same procedure as described in SBF (Seo, Chae, Shim, Kim, & Cheong, 2016). Figure 3.15 depicts the procedure.

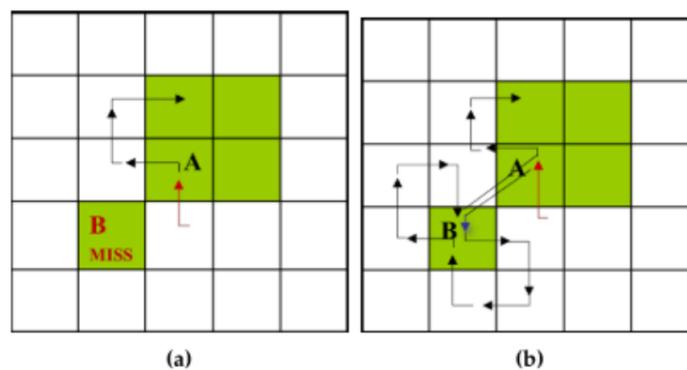


Figure 3.15. Track of inner-outer corner pixel at the left-rear pixel. (a) Simple boundary follower (SBF); (b) modified SBF track (Seo, Chae, Shim, Kim, & Cheong, 2016)

- Improved Simple Boundary Follower

Apart from the previous algorithms, Improved Simple Boundary Follower (ISBF) begins with a contour assuming a tracer is an object. ISBF applies six conditions. As Figure 3.16 represents the tracer moves. If the left neighbor is an object, tracer moves left. If the left-rear is the object and the left and the rear are background, the tracer moves to the inner-outer pixel at the left-rear. If the front-left is black and the left is background, tracer checks the front pixel. If the front is background, the next pixel is the inner-outer corner. If the front is an object, it is the inner corner. If only the front is the object, the next pixel is the front neighbor. If all the neighbors are background, the tracer is on the outer corner.

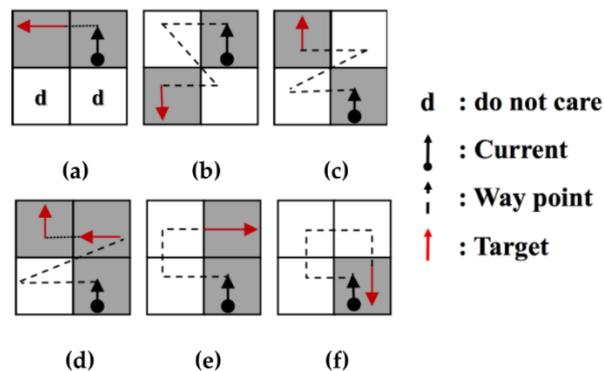


Figure 3.16. ISBF (Seo, Chae, Shim, Kim, & Cheong, 2016)

- Radial Sweep Algorithm

Radial Sweep Algorithm has a tracer that visits the pixels with a sweep and constantly moves without counting the object pixels. To initiate the algorithm, two sequential contour pixels are needed. The following contour pixel is determined by drawing a circle in a clockwise direction with respect to the previously visited pixel. After the next pixel is attained, the first pixel must be the one to draw a clockwise directional circle (Seo, Chae, Shim, Kim, & Cheong, 2016). Figure 3.17 shows the process.

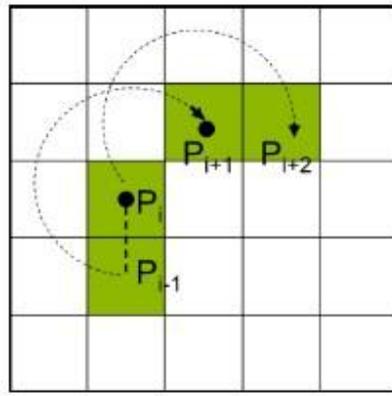


Figure 3.17. Radial Sweep Algorithm tracer

- Theo Pavlidis Algorithm

To decide next contour pixel, the Theo Pavlidis Algorithm (TPA) considers only three adjacent pixels. They are front-left, front and front right. If all three pixels are white, the tracer turns right (Seo, Chae, Shim, Kim, & Cheong, 2016).

- Moore-Neighbor Tracing Algorithm

The algorithms described above accept binary input images as mentioned before, on the other hand, NEXRAD Level III hydrometeor classification images consist of more than one precipitation labels. However, each segment of the same label can easily be converted to a binary image and processed separately. While tracking the boundary, the algorithm also checks if there exist any holes in the segment.

The tracking function that is preferred in this dissertation uses the Moore-Neighbor Tracing (MNT) algorithm. The working principle of MNT is that it detects the advanced pixel by checking the eight connected neighbor pixels. In the beginning, the tracer drives through the T(Prear, drear) and from that point, it proceeds in the clockwise direction. The visited pixels are given in Figure 3.18 (Seo, Chae, Shim, Kim, & Cheong, 2016). This algorithm was chosen because it does not miss any corner

pixel. At the same time, it is very fast. After application of the MNT algorithm, a 2D polygon is obtained for each segment.

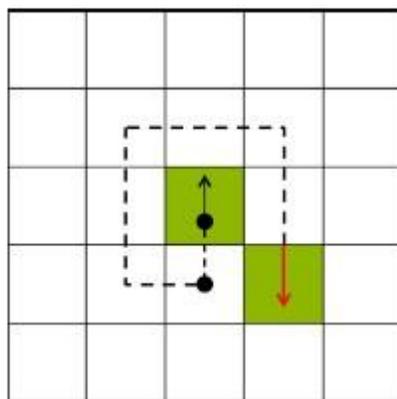


Figure 3.18. Moore-Neighbor Tracing Algorithm (Seo, Chae, Shim, Kim, & Cheong, 2016)

3.3.2. Transformation of Image Coordinate to Spatial Coordinate

The NEXRAD Level III Digital Hydrometeor Classification images exported from Weather and Climate Toolkit (WCT) have their metadata represented in the Geographic Coordinate System. Coordinate systems need to be specified with datum, spheroids, prime meridian, and angular unit. Among many alternatives, the chosen one is North American Datum 1983 (NAD83), which is GPS-compatible. The corresponding Spheroid to NAD83 is the Geodetic Reference System 1980.

The entire process of adding positional information of metadata into vertices (the pixels of the boundary) requires an image coordinate system transformation into the 3D geographic coordinate system before storing as polygonal information.

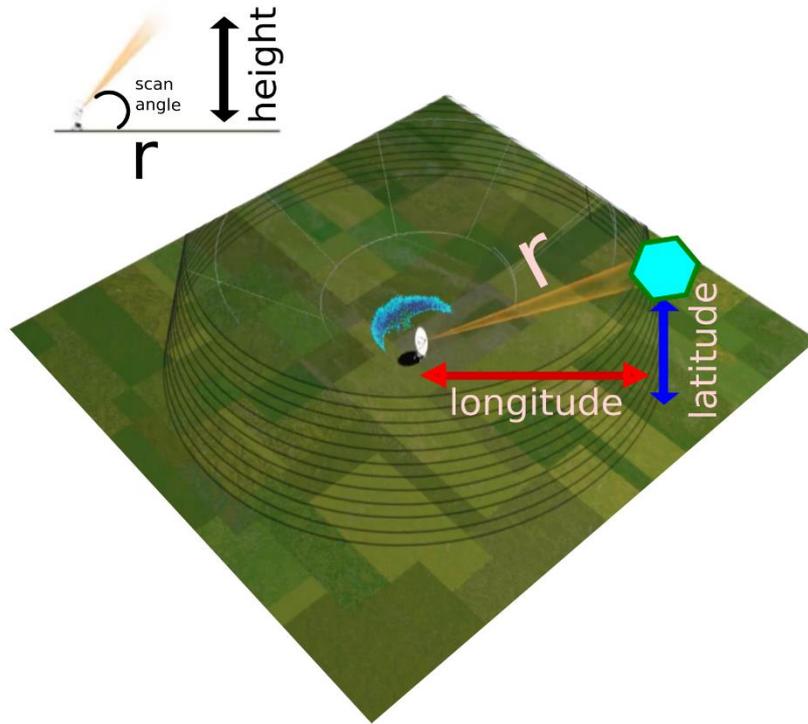


Figure 3.19. Height calculation (University Corporation for Atmospheric Research, 2009)

Let a radar image have the size $H \times W$. Metadata has vectors for Longitudes of size $H \times 1$ and Latitudes of size $W \times 1$. Since the coordinates of the pixels are known, altitude of each vertex can be calculated. Let the radar have the coordinates of R_{lat} , R_{lon} . As shown in Figure 3.19, the altitude of any point with respect to the radar site, H_{dist} , can be calculated using the following equations (Equation 5 and Equation 6):

$$H_r = \sqrt{(d_{lon} \times (R_{lon} - V_{lon}))^2 + (d_{lat} \times (R_{lat} - V_{lat}))^2} \quad (5)$$

$$H_{dist} = (\tan \beta \times H_r) + R_h \quad (6)$$

where

- d_{lon} : Longitude to meter conversion factor
- $V(lon, lat)$: Vertex location
- $R(lon, lat)$: Radar site location
- d_{lat} : Latitude to meter conversion factor
- H_r : Horizontal distance of radar with respect to object (r)

- β : Elevation angle
- R_h : Radar altitude
- H_{dist} : Calculated height of the object.

3.3.3. Export of 3D Polygons in the KML Format

One of the most striking features of the Toolkit is making radar data displayable in the Google Earth in three-dimensions. For this purpose, the polygons taken from border tracking algorithm are exported in Keyhole Markup Language (KML) format, in which each hydrometeor class has a unique name and a unique color.

MATLAB™ has a data structure for representing Geographic Shape Vectors named "geoshape". These structures represent an object like (in this case) polygonal information with defined geographical coordinates including other features that a polygon might have, for example, color, temperature, pressure, etc. "kmlwrite" is used to convert vector vertex data into a format that Google Earth is able to map and display. Figure 3.20 is the depiction of “KML” exported polygons opened in Google Earth.

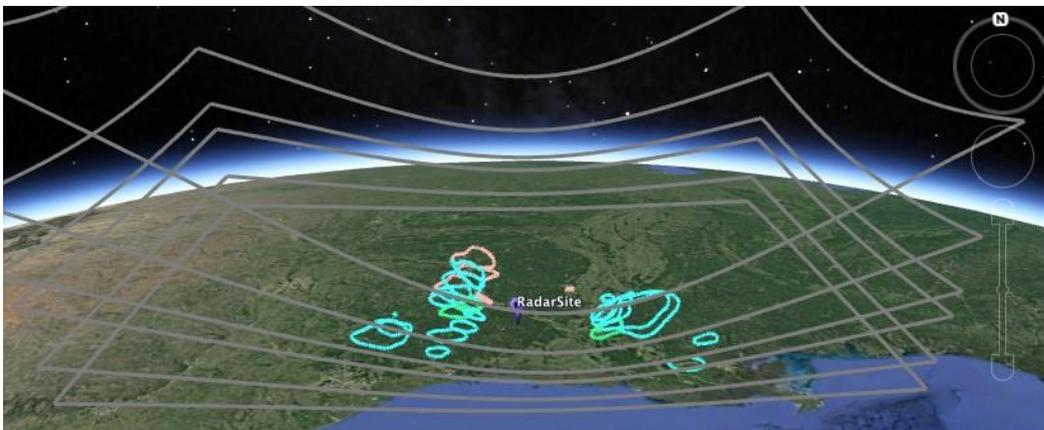


Figure 3.20. KML is exported and viewed in Google Earth

3.4. Matching Polygons

Until this point, the polygons describing each object's three-dimensional profile in the available elevations have been obtained. However, the polygons originating from the same object still need to be matched so that the surface of the three-dimensional object can be created.

3.4.1. Matching Polygons from Two Successive Elevations

Here, the aim is to determine whether polygons of the same type of precipitation at different elevations are the continuation of the same mass. First, only the polygons with the same hydrometeor label can be matched. It is expected that the vertical projections of the sections of the same object overlap considerably. Higher overlap may indicate a higher probability of true matching. The developed algorithm is based on the Euclidian distance of centers of masses. The example centre of the masses data from various elevations can be seen (Figure 3.21).

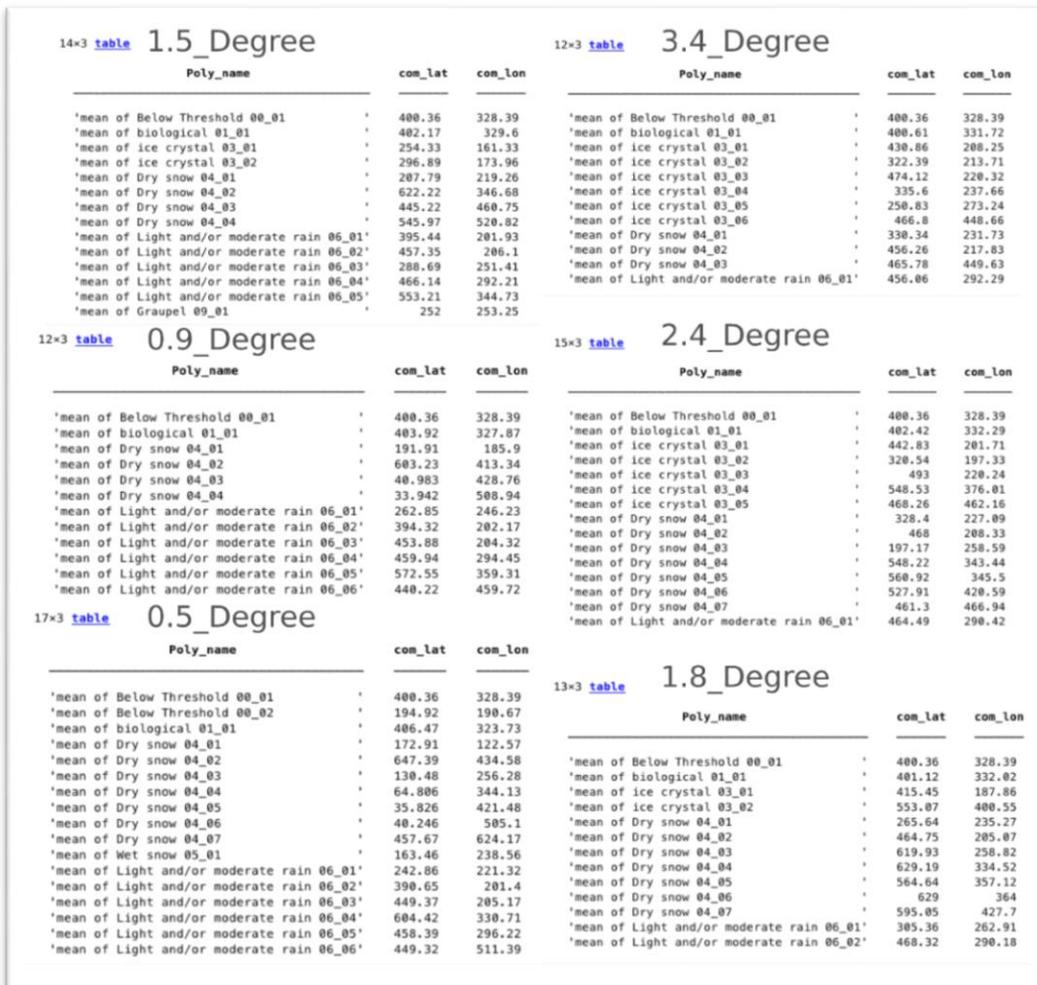


Figure 3.21. Sample of center of mass table for each elevation angles: Left, bottom to top: 0.5°, 0.9°, 1.5°; right, bottom to top: 1.8°, 2.4°, 3.4°.

Only polygons with the same label can be matched to each other. Hence, the algorithm performs matching for each label independently. Figure 3.22 illustrates polygons from successive elevations having the center of mass coordinates carrying just the label 4 (dry snow).

```

K>> 0.9_Degree      K>> 1.8_Degree      K>> 3.4_Degree
ans =
191.9147 185.8961
603.2349 413.3413
40.9831 428.7627
33.9416 508.9416
K>> 0.5_Degree
ans =
172.9089 122.5703
647.3913 434.5780
130.4804 256.2849
64.8065 344.1290
35.8256 421.4767
40.2462 505.1026
457.6667 624.1667

K>> 1.8_Degree      K>> 3.4_Degree
ans =
265.6432 235.2683
464.7541 205.0656
619.9308 258.8231
629.1910 334.5169
564.6400 357.1200
629.0000 364.0000
595.0465 427.6977

K>> 2.4_Degree
ans =
328.4004 227.0857
468.0000 208.3301
197.1656 258.5894
548.2222 343.4444
560.9167 345.5000
527.9140 420.5914
461.3030 466.9394

K>> 1.5_Degree
ans =
207.7938 219.2591
622.2171 346.6771
445.2188 460.7500
545.9706 520.8235

```

Figure 3.22. Polygonal coordinates of centers of masses for label4 within elevation angles

The process of matching polygons from successive elevations starts with the calculation of distances between lower elevation angle polygons and upper elevation angle polygons. The distance is defined as Euclidian distance between centers of mass of polygons in the planimetry (x-y plane). In Table 3.1 0.5° elevation has seven polygons while the next elevation, 0.9°, has four polygons. In the 7×4 distance matrix, rows correspond to the lower image, while columns correspond to the higher image.

Table 3.1. Distance matrix of masses between two sequential elevations for label4

0.5° / 0.9°	<i>P1</i>	<i>P2</i>	<i>P3</i>	<i>P4</i>
<i>P1</i>	D11	D12	D13	D14
<i>P2</i>	D21	D22	D23	D24
<i>P3</i>	D31	D32	D33	D34
<i>P4</i>	D41	D42	D43	D44
<i>P5</i>	D51	D52	D53	D54
<i>P6</i>	D61	D62	D63	D64
<i>P7</i>	D71	D72	D73	D74

For matching two polygons, the minimums in both the rows and columns should be considered. Table 3.2 shows the minima of the columns: the “Vertical Minima” values. 1st polygon from the upper angle may match with 1st polygon from the lower angle, 2nd may match with 2nd, 3rd may match with 5th, 4th may match with 6th. On the other hand, the minimum values of each row need to be determined too. The “Horizontal Minima” values are placed in Table 3.2. Row minima suggest that, the 1st polygon from the lower angle may match with the 1st polygon from the upper angle. The 2nd may match with the 2nd, the 3rd may match with the 1st, the 4th may match with 3rd, the 5th may match with the 3rd, the 6th may match with the 4rd and the 7th may match with the 2nd.

Table 3.2. Calculated distances; horizontal and vertical minima

0.5° / 0.9°	<i>P1</i>	<i>P2</i>	<i>P3</i>	<i>P4</i>	<i>Horizontal Minima</i>
<i>P1</i>	66.1164	519.3536	333.4040	410.6029	66.1164
<i>P2</i>	518.9428	48.9978	606.4362	617.9406	48.9978
<i>P3</i>	93.4278	498.1601	194.3151	270.4721	93.4278
<i>P4</i>	202.9635	542.8587	87.9228	167.6777	87.9228
<i>P5</i>	282.5987	567.4677	8.9266	87.4851	8.9266
<i>P6</i>	353.4064	570.4178	76.3434	7.3814	7.3814
<i>P7</i>	512.5477	256.1981	460.2260	439.1124	256.1981
<i>Vertical Minima</i>	66.1164	48.9978	8.9266	7.3814	

Clearly, there are many inconsistencies to be resolved. Figure 3.23 summarizes the matching procedure. The bold numbers indicate higher elevation angle. The sample distance calculation was held for the sixth lower elevation angle polygon. As there are four polygons in the upper elevation angle, the calculated distances were shown with the arrow and named accordingly (D61, D62, D63, and D64). For the sixth polygon, the minimum distance is D64. Therefore, this polygon will match with upper fourth polygon if this upper polygon has no other minimum distance option. On the other

hand, when the same procedure is applied to remaining polygons, many inconsistencies occur. The upper-to-lower (U for upper, L for lower) matches are 1U-1L, 2U-2L, 3U-5L, 4U-6L and lower-to-upper 1L-1U, 2L-2U, 3L-1U, 4L-3U, 5L-3U, 6L-4U and 7L-2U. The problem is that both 1L and 3L choose 1U. Both 2L and 7L choose 2U. Both 4L and 5L choose 3U. That is why the additional operation was needed applicable for both upper and lower cases at the same time.

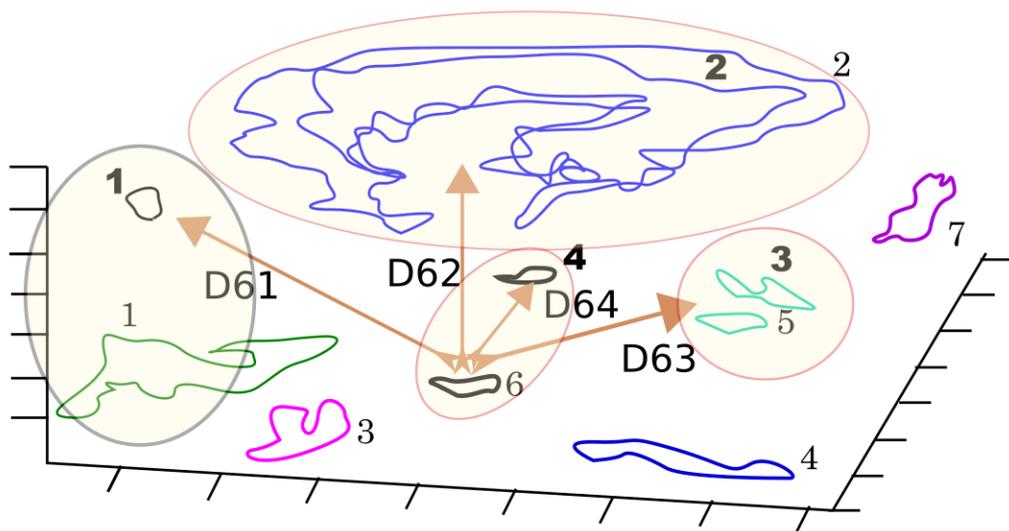


Figure 3.23. Polygon matching (edited from result)

The solution is simple, the intersection of these values are taken. That is, when two polygons match each other, this is accepted as an unambiguous match. When a match is done, corresponding row and column are discarded. This reduces the ambiguity for the remaining polygons. There is one last step to complete the selection process, which is the thresholding. After lots of trial process, the 50 to 70 pixels have been found to be a suitable as a threshold value. This pre-determined value is applied to the remaining values. For instance, in Figure 3.23, the first polygons of two elevation angle's distance to each other is between 50 and 70. It is user's decision to match these polygons. The values below the 50 pixels are good matches. As it approaches to zero, the expected match gets perfect. On the other hand, in some rare cases the threshold

value is re-set to 70 because the number of vertices of polygons in sequential elevation has huge difference when it compared to normal cases. In that case, the polygon with the huge vertex number may take the threshold value as 70 pixels because the objects represented by border vertices are spread over a large area in an image. In Table 3.3, the function applied the 50 pixels threshold value before deciding for final distances. The rest is that to go back to the distance table and find the corresponding row and column values of remaining distances. As shown in Table 3.3 the “Final Match” represents that these are the final matched polygons.

Table 3.3. Intersection values (bold on both sides); after threshold operation, final matches are displayed.

0.5° / 0.9°	<i>P1</i>	<i>P2</i>	<i>P3</i>	<i>P4</i>	<i>Horizontal Minima</i>
<i>P1</i>	66.1164	-	-	-	66.1164
<i>P2</i>	-	48.9978	-	-	48.9978
<i>P3</i>	93.4278	-	-	-	93.4278
<i>P4</i>	-	-	87.9228	-	87.9228
<i>P5</i>	-	-	8.9266	-	8.9266
<i>P6</i>	-	-	-	7.3814	7.3814
<i>P7</i>	-	256.1981	-	-	256.1981
<i>Vertical Minima</i>	66.1164	48.9978	8.9266	7.3814	
<i>Final Match</i>	Over-threshold	2-2	5-3	6-4	

3.4.2. Tracking Meshes Through Elevations

In this step, objects are generated by matching polygons two by two in all elevations. The matching process proceeds from the lowest elevation to the highest. In the end, there might be multiple object formations or just a one depending on the number of

reconstructed polygonal objects. However, when the cloud formation has an oblique shape, the two-dimensional Euclidian distances of the reconstructed polygons can be too large, when it is compared with the polygons from the highest and lowest elevations. That is why; matching begins from the lowest elevation and moving upwards while tracking the object. Figure 3.24 shows a reconstructed object from the boundary polygons. It is clear that matching the lowest elevation angle polygon with the highest elevation angle polygon is not possible. The centre of masses distance of these polygons will be over the threshold value or they will match incorrectly with other polygons. However, when they are matched two by two in sequential elevations, the threshold error will be eliminated. Moreover, incorrect matching will not occur. As a result, the algorithm was designed not to miss any elevation angle while matching polygons.

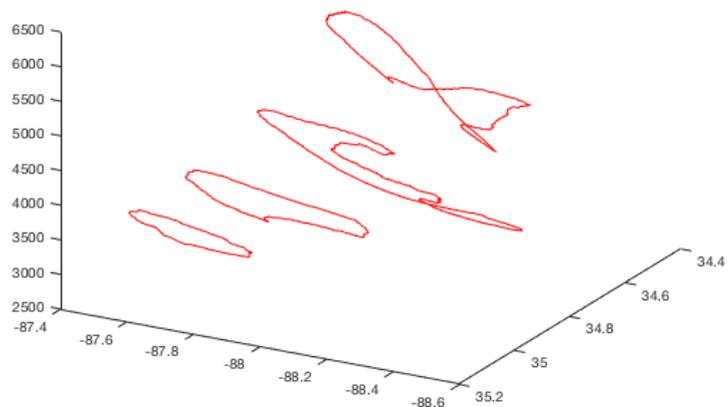


Figure 3.24. Tracking matches

Table 3.4 shows the matched polygons in sequential elevations for the label 6 (Light and / or moderate rain).

Table 3.4. Tracking matches for Label6 (Light and / or moderate rain).

Elevation Pair:	0.5-0.9	0.9-1.5	1.5-1.8	1.8-2.4	2.4-3.4
Object IDs	1_4-2_5	2_5-3_5	3_5-4_2	4_2-5_1	5_1-6_1
	1_1-2_1	2_1-3_3	3_3-4_1		
	1_3-2_3	2_3-3_2			
	1_5-2_4	2_4-3_4			
	1_2-2_2	2_2-3_1			

According to Table 3.4 the object formation occurring at the first row has a layer in all of six elevations. Depending on the number of vertices at each elevation angle, visualization of this cloud will be more clear and notable. Apart from the first rows, the label 6 has four more different object formations, which are less apparent. Because when these four objects are compared with the first one, they have fewer layers. By selecting Light and / or moderate rain from the legends in the Toolkit, algorithm recalls this specific label and begins modeling the objects with the help of this configuration. As it can be seen obviously from the Table 3.4, while approaching to the next row, the repetitions in numbers occur. The second part of one elevation needs to be the corresponding first part of the next elevation couple. Algorithm checks these repetitions, and if they exist, it does not create a new object. It just keeps on adding the new sequential elevation information to the object. Whenever repetition stops, the cloud is created. Therefore, if the next matched pairs come from the same elevation, and then algorithm begins with creating a new cloud. In the following, the algorithm (3) is explained in detail.

Algorithm3: Creating objects as a combination of matched polygons

Input:

- Pairwise matches from successive elevations along with their labels

Output:

- A list of objects each composed of a list of polygons

Algorithm:

- Let $e=1 \dots E$ be the elevation number
- Let $m_{l,e,n}$, $e=1 \dots E-1$ and $n=1 \dots N_e$ be the match list of polygons labeled l from elevation e to elevation $e+1$ and N_e the number of matches between them

1. **for each label lb=1 to L**

2. **for e=1 to E-1**

begin $m_{l,e,n}\{lb\}$ to create 1st object (1st and 2nd elevation)

if $m_{l,e,n+1}\{lb\}$ **exist,**

create object

fill objects with variables

$lb =$ label from (1st step)

$e = 1$ implies 1st and 2nd elevation [1 2]

 polygon numbers = [p q] from $m_{l,e,n}$ (3rd step)

 polygon coordinates = [lon×lat×h]

3. **find** $m_{l,e+1,n}\{lb\}$,

4. **if** it exists

5. **track** matches through N_e

continue adding lb , e , polygons data to the same objects

6. **else**

7. **create** a new object

8. **go through** (3rd step), do the same

9. **end**

end

Table 3.5 shows the created objects (Cobjects) in an order with respect to labels. Each label number corresponds to a different type of precipitation (hydrometeor

class). As it is mentioned previously, there are maximum six elevations and two by two matching of polygons' coordinates are given in the coordinate (coords) column. Table 3.5 depicts the sample data structure of a radar site at a predetermined time with twenty different cloud formations.

Table 3.5. Created objects in a resultant view

Obj_no / Sub-sections	lbl	lvl	polygons	coords
Cobjects {1}	1	[1 2 3 4 5 6]	[1 1 1 1 1 1]	[191×4] [173×4] [132×4] [57×4] [45×4] [46×4]
Cobjects {2}	3	[4 5]	[2 4]	[60×4] [185×4]
Cobjects {3}	3	[4 5 6]	[1 1 1]	[42×4] [89×4] [118×4]
Cobjects {4}	3	[5 6]	[5 6]	[128×4] [295×4]
Cobjects {5}	3	[5 6]	[2 2]	[39×4] [28×4]
Cobjects {6}	3	[5 6]	[3 3]	[67×4] [50×4]
Cobjects {7}	4	[1 2]	[6 4]	[195×4] [137×4]
Cobjects {8}	4	[1 2]	[5 3]	[86×4] [59×4]
Cobjects {9}	4	[1 2 3 4]	[2 2 2 4]	[1270×4] [1260×4] [898×4] [89×4]
Cobjects {10}	4	[1 2 3 4 5 6]	[1 1 1 1 1 1]	[626×4] [645×4] [548×4] [723×4] [502×4] [441×4]
Cobjects {11}	4	[4 5 6]	[2 2 2]	[61×4] [103×4] [76×4]
Cobjects {12}	4	[4 5]	[5 5]	[175×4] [24×4]
Cobjects {13}	4	[4 5]	[7 6]	[43×4] [93×4]
Cobjects {14}	4	[5 6]	[7 3]	[33×4] [89×4]
Cobjects {15}	6	[1 2 3 4 5 6]	[5 4 4 2 1 1]	[107×4] [123×4] [138×4] [127×4] [113×4] [78×4]

Cobjects {16}	6	[1 2 3]	[2 2 1]	[83×4] [76×4] [45×4]
Cobjects {17}	6	[1 2 3]	[3 3 2]	[60×4] [57×4] [48×4]
Cobjects {18}	6	[1 2 3 4]	[1 1 3 1]	[547×4] [311×4] [194×4] [56×4]
Cobjects {19}	6	[1 2 3]	[4 5 5]	[710×4] [265×4] [102×4]
Cobjects {20}	6	[1 2]	[6 6]	[152×4] [50×4]

3.5. Visualization in Three-dimension

3.5.1. Pointwise Matching of Polygons

So far, the matched polygons define the objects, but only as slices at multiple elevations. However, the successive polygons should be connected via edges so that a closed wireframe of the object can be formed. For that purpose, successive polygons are matched point by point in two stages: coarse matching and fine matching.

3.5.1.1. Coarse Matching

In this first step, a small number of points (pivot points) are matched pointwise between two consecutive polygons. In the fine matching step, the segments between the pivot points are matched. The coarse matching algorithm first determines the pivot points and then matches these points.

- Pivot Point Determination

First, the pivot points are chosen in each polygon. In most of the cases, at least four points are chosen using a bounding box method and four points are chosen using corners.

For the first method, the vertices with minimum and maximum coordinates are determined along both axes and a bounding box is drawn, as can be seen in Figure

3.25. It is very common that the polygon touches an edge at multiple points (Vertices in blue color).

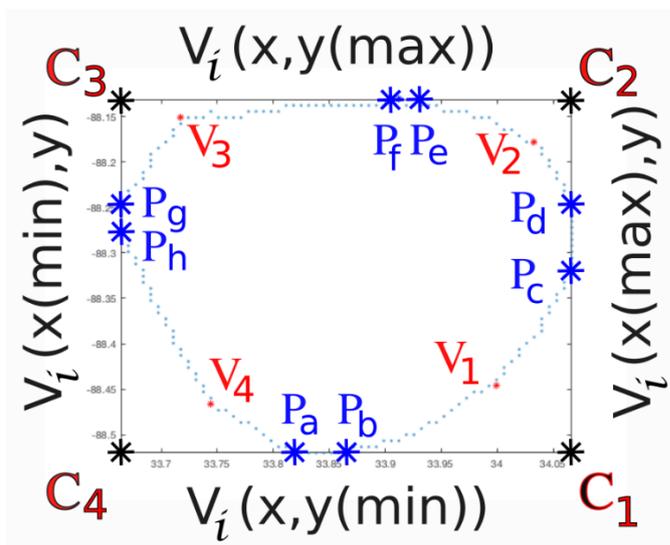


Figure 3.25. Polygon bounding box and pivot points

Because of dealing with hundreds of vertices in most cases, the algorithm checks in each step if there exist multiple vertices with the same latitude or longitude at the maxima or at the minima. For this reason, the pivot points determination algorithm is performed in two dimensions. Instead of using latitude and longitude, it is preferred to name axes as “x” and “y” for convenience. The bottom left is assigned as the origin. The algorithm begins with finding out the first pivot point on the bottom side. Among the vertices, which have the minimum “y” value, there must be a vertex with the minimum “x” value (P_a in Figure 3.25). The second pivot point has the same “y” value (with P_a) but has the maximum “x” value (P_b in Figure 3.25). At the right side, the vertices have the maximum “x” values. Among them, there must be vertices that have the minimum “y” value (P_c in Figure 3.25) and maximum “y” value (P_d in Figure 3.25). For the determination of the next pivot points, two consecutive vertices from two consecutive sides are needed. These two vertices form an interval (for example, P_b and P_c). From this interval, the vertex with the minimum distance to the first corner

(C_1 from Figure 3.25) is chosen as the other pivot point (V_1 in Figure 3.25). With this method, four pivot points corresponding to the corners of the boundary box are found. Hence, up to 12 pivot points are obtained.

- Pivot Point Matching

The fact that boundary tracking algorithm stores the vertices with the same sense of rotation (counter-clockwise), the matching of these pivot points result in a rough 3D model. For this purpose, the pivot points are put in the same order in every elevation. This order has the same orientation with the vertex array. As Figure 3.26 shows, the first pivot point is always at the minimum “y” side of the bounding box.

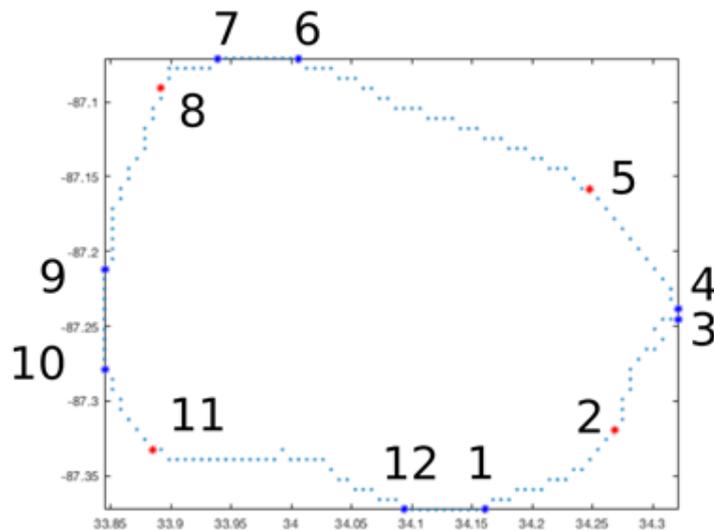


Figure 3.26. Numbering of the pivot points

Figure 3.26 shows all of the pivot points of an example polygon. Although the polygon in Figure 3.26 has 12 pivots, others may have less. In order to match the polygons in successive elevations, the algorithm should not require the existence of exactly twelve pivot points. As the example in Figure 3.27 shows, there are five pivot points on the bounding box (vertices with blue color) and there are three pivot points originating from the corners (vertices with red color). Nevertheless, when it comes to matching, these eight pivot points -in total- carry all the characteristics of the twelve pivot points.

For instance, on the right side of the bounding box, there is just a single pivot point. If the upper elevation has multiple vertices in the same interval (3-4), these vertices should match to this single pivot point. Because this pivot point behaves as both the third and fourth pivot points. When it comes to the left side of the bounding box, the bottom-left corner pivot point pretends as it is the ninth pivot point. Therefore, the top-left corner calculates its minimum distance along the segment that is originated from seventh and ninth pivot points. That gives the eighth pivot point. Then, the same bottom-left corner pivot point behaves as it is the 10th pivot to match with the upper interval. Therefore, the upper (9-10) interval will match through this single vertex. Thereafter, as being the corner pivot point, this vertex becomes the 11th corner pivot point. This time, the upper 10-11 interval matches to this single vertex. Finally, the fact that this vertex is in the same axis with the first pivot point, it behaves as the 12th pivot point. Therefore, the upper elevation (11-12) interval matches to this single vertex again. As a result, the remaining (12-1) interval will match through the same process. However, this time the interval has more than one vertex. Last of all, the polygons can be matched even if they have just a few vertices. Regardless of how irregular they are; they will be able to connect to each other easily. At the same time, they visually begin to create a satisfactory outer surface. However, coming across with this particular situation is very rare.

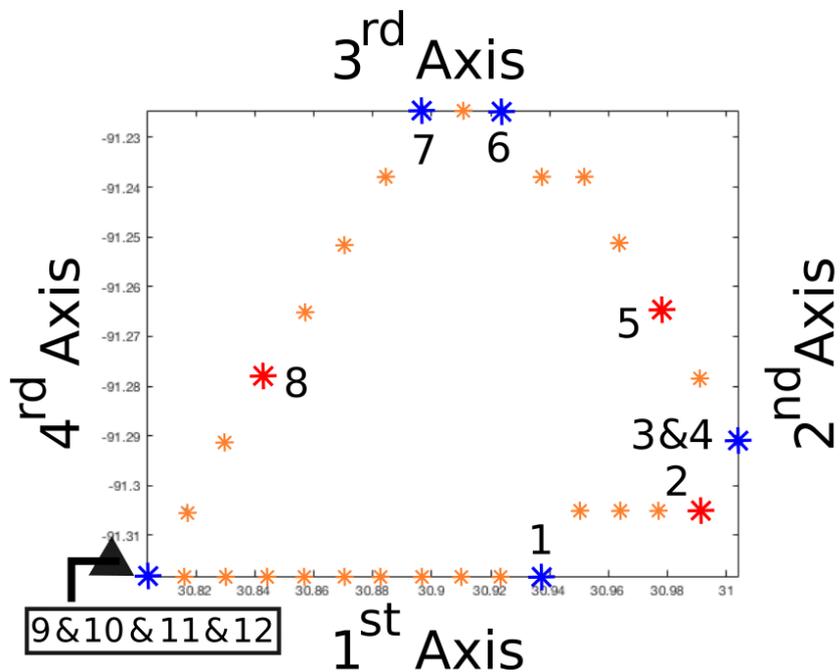


Figure 3.27. One vertex may take multiple characteristics

3.5.1.2. Fine Matching

After coarse matching of the segments (matching of pivot points) has been performed to combine polygons in successive elevations, the next step is to start creating a 3D wireframe structure. The selected polygons are separated into 12 pieces, including the pivot points on the bounding box and the remaining pivot points that are found by using corners of the bounding box. It is time to create triangular surfaces between consecutive elevation's polygons so that a surface that can be rendered by computer graphics algorithms is obtained. In each interval of the lower elevation angle polygon, there will be a corresponding interval to coincide at the higher elevation angle polygon (Figure 3.28).

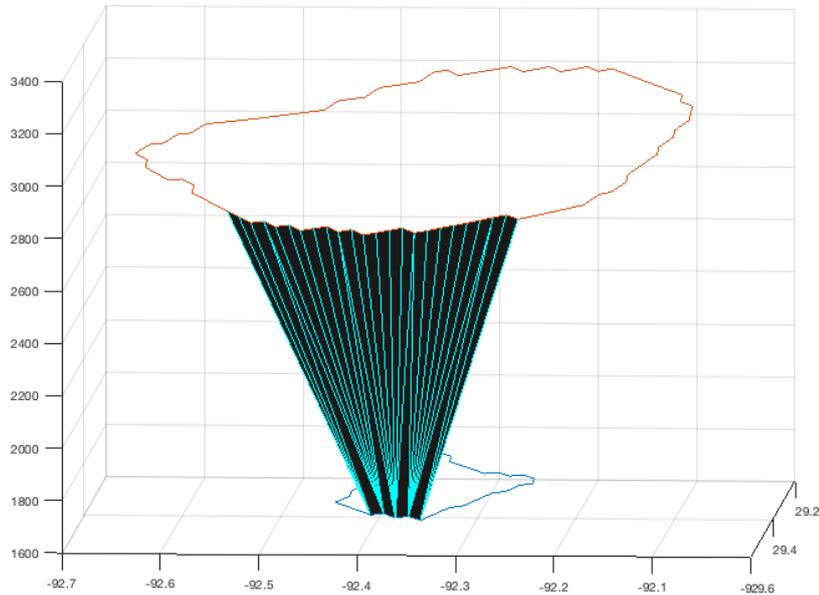


Figure 3.28. Merged vertices in a single interval

In order to match each vertex, the number of vertices in both the lower elevation interval and the upper elevation interval need to be known first. They are denoted as t_1 and t_2 , respectively. Second, the total number of triangles that must be created for the representation of complete mesh structure, n_t , can be found by as $n_t = (t_1+t_2)-2$. The algorithm is given below.

Algorithm4: Triangulated meshes

Input:

- Matched Polygons (P_1 and P_2) in consecutive elevations with coordinates (latitude, longitude, height)

Output:

- Triangle description (Tri) and unified coordinates

Algorithm:

- **draw** bounding box for both P_1 and P_2
- **find** vertex number of pivot points both in P_1 and P_2

- **compose** the intervals with respect to pivot points ($t_1, t_2..t_{12}$)
- **for** interval one to twelve
 - **find** length of t_1 and t_2 (len_1 and len_2)
 - **if** $len_1 < len_2$
 - **if** $len_1 = 1$
 - begin** from $x = 1$ to $x = len_2 - 1$
 - store** [$1, x+2, x+1$]
 - **else**
 - find** mapping function (m_1)
 - begin** from $x = 1$ to $x = len_2 - 1$
 - if** $m_1(x)$ is not equal to $m_1(x+1)$
 - store** [$m_1(x), len_1+x+1, len_1+x$]
 - store** [$m_1(x), m_1(x+1), len_1+x+1$]
 - else**
 - store** [$m_1(x), len_1+x+1, len_1+x$]
 - **else**
 - **if** $len_2 = 1$
 - begin** from $x = 1$ to $x = len_1 - 1$
 - store** [$x, x+1, len_1+1$]
 - **else**
 - find** mapping function (m_2)
 - begin** from $x = 1$ to $x = len_1 - 1$
 - if** $m_2(x)$ is not equal to $m_2(x+1)$
 - store** [$x, x+1, len_1+m_2(x)$]
 - store** [$len_1+m_2(x)+1, len_1+m_2(x), x+1$]
 - else**
 - store** [$x, x+1, len_1+m_2(x)$]

3.5.2. Triangles to Surfaces

The “trisurf” function of MATLAB enables users to use triangles, as surfaces. However, this function requires the triangles in a specific format: Three vectors of x , y and z coordinates of the vertices with length N and a matrix of size $N \times 3$ describing the triangles. Each row of the matrix is composed of the indices of the vertices creating the triangle in the counterclockwise order. As Figure 3.29 shows, there are two vertices in both elevations, which mean there are four vertices and two triangles in total. Therefore, in order to make the function draw triangles along these vertices, the

input array must start from a vertex of one of the triangle and move around in counter-clockwise direction. These might be “1 2 3” and “4 3 2”.

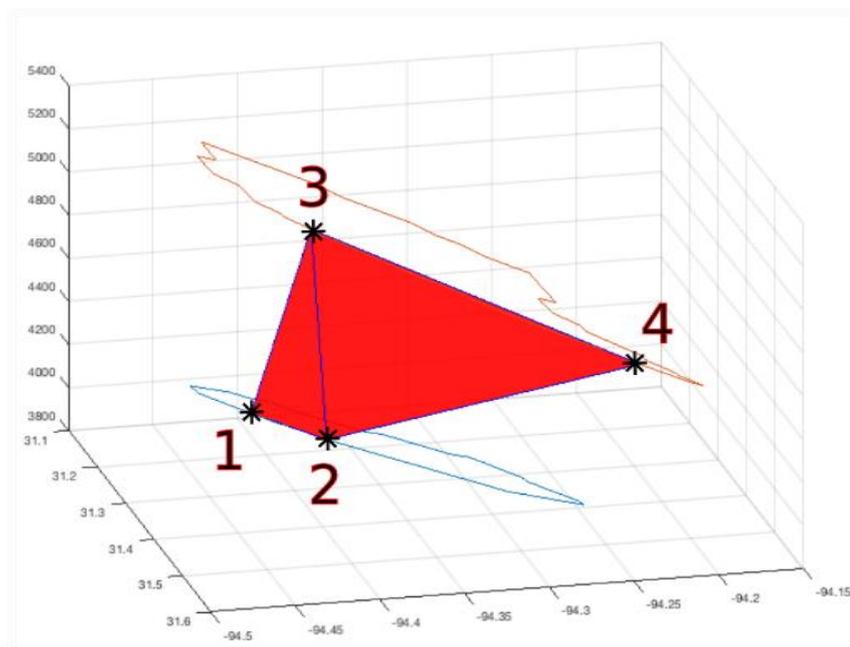


Figure 3.29. Simple “trisurf” example with two triangles

As it is stated previously, the intervals may have a different number of vertices. An algorithm must be designed for creating these triangles in such a way that all of the triangles are created without missing any of them. For this reason, a linear mapping function is created between the segments. Mapping function requires a ratio of the number of vertices for two successive elevations in each interval. In addition to that, there must be the function, which uses this ratio to find out which specific vertices in one interval will match other vertices in the consecutive interval. This structure is examined in two cases: The first one is when the lower segment is shorter than the upper segment, second; the upper segment is shorter than the lower segment as in the following.

3.5.2.1. Shorter Lower Segment Case

In this case, the algorithm needs to distinguish two possible scenarios. Although the lower elevation angle segment is shorter than the upper elevation segment, the

formation of the triangles differs if there is just a single vertex in the lower elevation interval. Therefore, the algorithm first checks if t_1 is equal to one or not.

- Lower Segments with a Single Vertex

When the lower elevation segment has just a single vertex, as expected, all of the vertices in the upper interval matches to this vertex. All of the triangles have the index of this vertex in their representation.

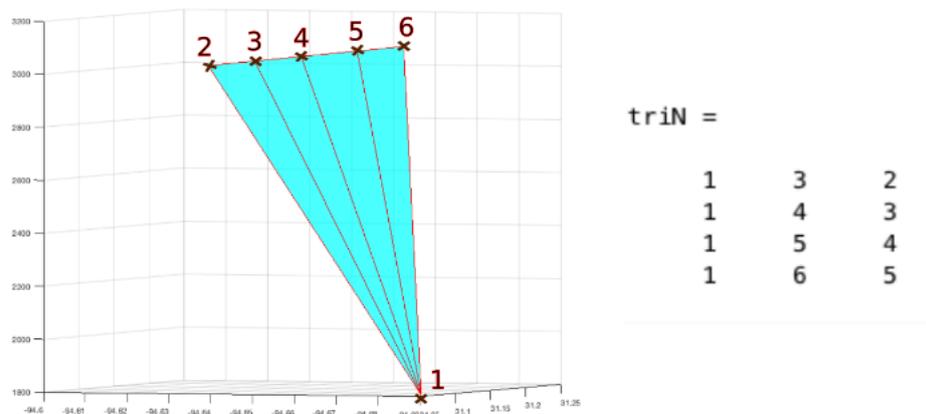


Figure 3.30. One single vertex at the lower elevation angle interval

As the example in Figure 3.30 shows, t_1 is one and t_2 is five; so there are six vertices in total and this will result in four triangles. To align these triangles, the loop is iterated from $x=1$ to t_2-1 and in each loop, it creates the triangle named as $[1, x+2, x+1]$. When “ $x = 1$ ”, the first outcome from the loop will be $[1\ 3\ 2]$. Figure 3.30 shows all possible triangles. After finding out the triangles, the coordinates of these vertices (latitude, longitude, and height) are stored in an array according to the vertex order separately in order to visualize the surface in MATLAB™.

- Lower Segment with Multiple Vertices

When two segments are matched, the first vertex from lower elevation angle needs to match with the first vertex from the upper elevation angle; and the last vertex from the

lower elevation angle need to match with the last vertex from the upper elevation angle. In between these vertices, there is a linear function, which determines connections of other vertices by applying a mapping function to these two intervals. These results will give a vertex number of the lower elevation angle to match. From this attachment of the vertices, a loop is structured and the triangles are drawn accordingly.

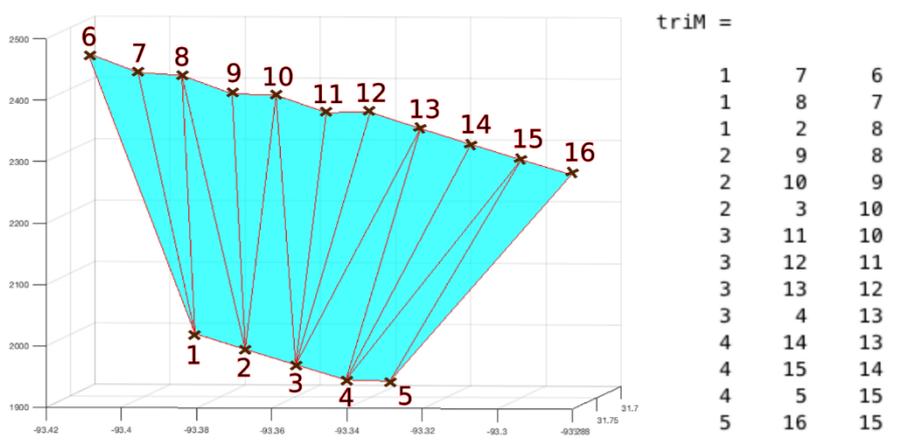


Figure 3.31. Lower interval is shorter than upper interval

As the example in Figure 3.31 shows, t_1 is five and t_2 is 11. Totally, there are 16 vertices, so the number of triangles should be 14. Therefore, the ratio (r) of “ t_1-1/t_2-1 ” is “0.4” and the mapping function is

$$m = (x - 1) \times r + 1 \tag{7}$$

After rounding, the result is $m = \{1, 1, 2, 2, 3, 3, 3, 4, 4, 5, 5\}$. That is, the corresponding vertices are $\{6-1, 7-1, 8-2, 9-2, 10-3, 11-3, 12-3, 13-4, 14-4, 15-5, \text{ and } 16-5\}$. The remaining part of the algorithm depends on the tracking of the mapping results. When two consecutive matching points are the same, one triangle is drawn which is “[$m(x), t_1+x+1, t_1+x$]”. On the contrary, if the two consecutive points differ, it means that a rectangular shape is formed. Therefore, there must be two triangles to be drawn “[$m(x), t_1+x+1, t_1+x$]” and “[$m(x), m(x+1), t_1+x+1$]”. That is; considering the Figure 3.31; the first and second result of the mapping function is the same ($x=1$);

so, the triangle that needs to be drawn is “[1 7 6]”. In addition, the second point and the third point of the mapping function differ; therefore, these two triangles are “[1 8 7]” and [1 2 8]. The remaining triangles are found using the same algorithm and are shown in Figure 3.31 as “triM”. After finding out all triangles, the coordinates of these vertices (latitude, longitude, and height) are stored in an array according to the vertex order.

3.5.2.2. Shorter Upper Segment Case

In this case, two possible scenarios must be considered as mentioned previously. First, there might be just a single vertex at the upper interval. For this particular situation, there is a different algorithm for creating triangles. On the other hand, if the vertex number of the upper elevation is more than one, the algorithm should calculate the mapping function again. Therefore, this section will be examined in two parts. Firstly, the upper segment has one single vertex, and secondly, the upper segment has multiple vertices.

- Upper Segment with a Single Vertex

As the lower elevation angle interval composed of multiple vertices, they will all match with this particular vertex while creating triangles. According to Figure 3.32, t_1 is five, and t_2 is one. There are six vertices in total, which means there must be four triangles.

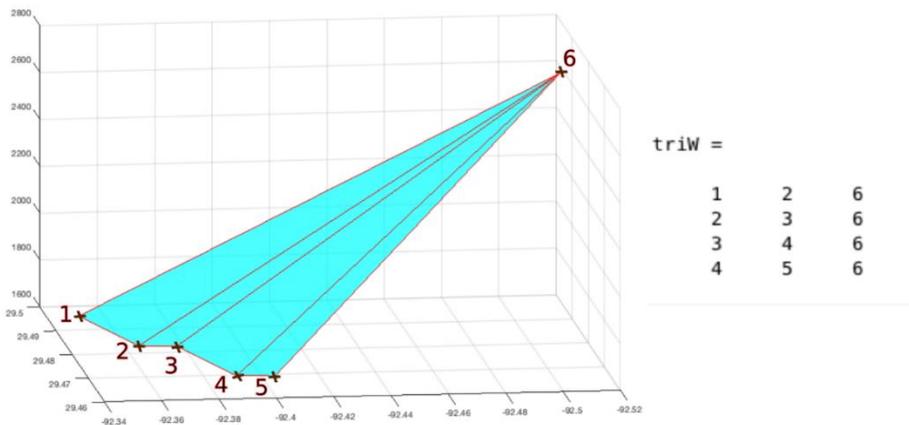


Figure 3.32. Upper interval has one single vertex

To write these triangles, the nested loop begins from “x= 1 to t1-1” and in each loop, it creates the triangle described as [x, x+1, t1+1]. When “x=1”, the first outcome from the nested loop will be: [1 2 6]. Consequently, Figure 3.32 shows all possible triangles. After finding out the triangles, the coordinates of these vertices (latitude, longitude, and height) are stored in an array according to the vertex order separately.

- Upper Segment with Multiple Vertices

The reason why sections 3.5.2.1 and 3.5.2.2 are separated is that when the two segments are compared, wherever there are more vertices, more than one vertex must match to a single vertex in the other interval. This creates confusion in the algorithm if the cases are not separated. Because the function that describes the triangles differs in two parts. First is the description of triangles because of obligatory of counterclockwise direction and second, the order of triangles. Because more than one vertex may attain to a single vertex and this situation may appear in the upper segment or the lower segment. This section’s algorithm is based on the same criteria: the first vertex from the lower elevation angle must match through the first vertex from the upper elevation angle, and the last vertices of both elevation angles must match. In between of them, there is a function, which determines connection of other vertices by applying a mapping function to these two intervals. This mapping function principle is similar to the sub-section 3.5.2.1 case but it differs in parameters. It takes

the inverse of the ratio when it is compared to the previous one, which was “ t_1-1/t_2-1 ”. Furthermore, the algorithm begins taking any number from “1” to “ t_1-1 ” to multiply with the ratio and adds “1” to the result before rounding. This result will give the vertex number of the upper elevation angle to match. Nevertheless, adding t_1 is needed to each result because the upper elevation angle vertices are the continuation of the lower elevation angle vertices. From this attachment of the vertices, the nested loop is structured and the triangles are drawn accordingly.

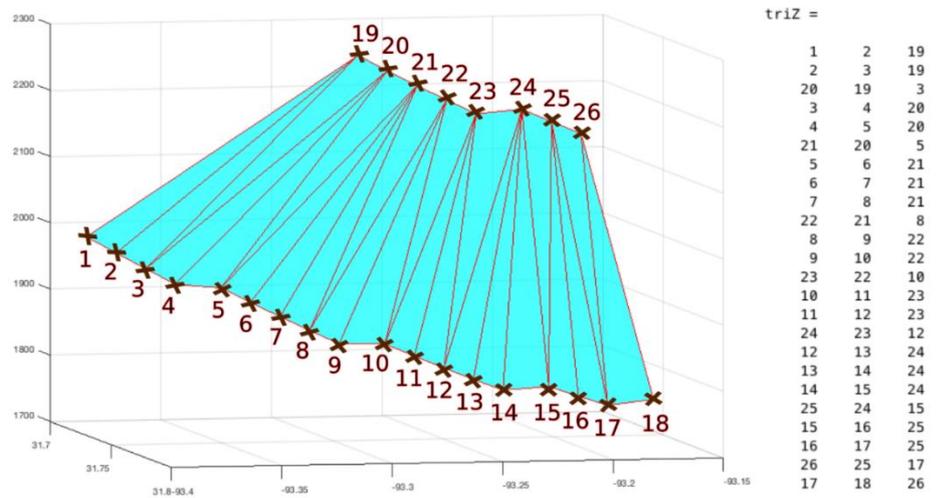


Figure 3.33. Upper interval is less than lower interval

As Figure 3.33 shows the t_1 is eighteen, while the t_2 is eight. There are twenty-six vertices in total, which means there must be twenty-four triangles. Therefore, the inverse ratio (r') of “ t_2-1/t_1-1 ” is “0.4118”. Next step would be the determination of mapping function (m). For every unknown (x) which begin from “1” to “ t_1-1 ”;

$$m = (x - 1) \times r' + 1 \quad (8)$$

After rounding, the result would give us $m = \{1, 1, 2, 2, 3, 3, 3, 4, 4, 5, 5, 6, 6, 6, 7, 7, 8, 8\}$. However, adding t_1 is needed for the results to show the matching clearly. Hereby, vertices are connected. The result is: $\{1-19, 2-19, 3-20, 4-20, 5-21, 6-21, 7-21, 8-22, 9-22, 10-23, 11-23, 12-24, 13-24, 14-24, 15-25, 16-25, 17-26, 18-26\}$. The

remaining part of the algorithm depends on the tracking of the mapping results. When two consecutive matching points are the same, one triangle is drawn, which comes from “[x, x+1, t₁+m(x)]”; on the contrary, if the two consecutive points differ, that means a rectangular shape is formed. Therefore, there must be two triangles to be drawn “[x, x+1, t₁+m(x)]” and “[t₁+m(x)+1, t₁+m(x), x+1]”. That is; considering Figure 3.33; the first and second points of the mapping function are the same (x=1). That is why the triangle that needs to be drawn is “[1 2 19]”. In addition, the second and third points of the mapping functions are different. Therefore, these two triangles are “[2 3 19]” and [20 19 3]. The remaining triangles are found from the same algorithm and are represented in Figure 3.33 as “triZ”. After finding out the all triangles; the coordinates of these vertices (latitude, longitude and height) are stored in an array according to the vertex order separately.

3.5.3. Visualization

After comparing various graphics functions in MATLAB, the "trisurf" command is preferred. Until this section, the algorithm has been dealing with just a single interval among twelve. When all segments are included, the “trisurf” function is run for all segments. It takes the triangle array, the latitudes, the longitudes, and the heights as the input. The model is displayed on the third screen of the Meteorology3D toolkit. Most of the times, depending on how hydrometeor types are distributed in the air, there might appear multiple clouds on the screen. In that case, the ability to demonstrate multiple objects from different views is possible. When the MATLAB command "rotate3d" is on, continuous rotations of axes can be performed by the operator. The users are able to read numeric values of the current azimuth and elevation of the axes. When the view is specified, releasing the mouse makes both animated box and values disappear. Objects protect their location.

3.5.3.1. Lighting as a Feature of Visualization

The shape of an object within a scene gets visible only when the color of a matter and the light source are enabled. The presence of light and shadow settings in computer

graphics exists for various appearances of results to achieve the best display. Hence, some of the settings are made selectable by the user through a pop-up. Because the user might want to analyze the outcomes in various colors. “Flat Lighting” brings on uniformly distributed color on to the target. “Gouraud Lighting” is more applicable in curved surfaces. Because color scattering interpolation at the vertices is computed. In addition to that, “Ambient Light” appears without any direction of a light, which arise uniformly on all targets in the scene. The only way it contributes is when the target has its own color. “Ambient Strength” is a determination of an intensity of light scattered from the object. Moreover, the Toolkit uses “Face Color” and “Edge Color” adjustments to settle the best color for the appearance of the object (MathWorks Inc., 2017).

3.5.3.2. Control Transparency

Transparency has its unique potential on representing graphics while enabling a user to see the degree of detail behind objects. The power of understanding the hidden objects behind the visible front faces of the targets can be adjustable. Eliminating all barriers is possible with alpha function in transparency. Some of the graphics support different alpha settings per objects. In the Toolkit, MATLAB sets transparency into an interval of zero to one. Zero stands for complete transparency, while one is for the opaque surface. Other values in between make object semitransparent (MathWorks Inc., 2017).

CHAPTER 4

RESULTS AND DISCUSSION

4.1. Formation of 3D Objects in the Toolkit

Sharing information via composing reconstructed objects based on their scientific quantities is more helpful than describing them verbally. A human being is able to memorize images more easily when the eyes see and the brain maps (Cho G. , 2012). Three-dimensional rotation on the created figure makes it straightforward to analyze object sizes, classes, volumes and the relations among them. Therefore, the main motivation behind the preparation of 3D visualization is the creation of volumetric object coverage for demonstrating the conditions in an effective way. The screens may be used as serious storm locaters and as instant cloud formation observation platforms.

Among hundreds, three radar sites are chosen for testing. Main concerns were to have radar sites in various parts of the USA and a high frequency of cloud formation in different climatic zones. Therefore, the main parameters are finding the most frequent storm path and time. The first radar site that has been tested was at Miami, Florida (site id KAMX). Dataset was taken on 01/01/2016. The second tested site was at Columbus AFB, near Jackson MS (site id KGWX). The date dataset was taken on 19/01/2017. The last tested site was at Seattle, near Everett WA (site id KATX). Dataset was taken on 09/05/2018. As mentioned previously, the Volume Coverage Patterns change daily. 114 images have been tested per elevation in KAMX, 72 images per elevation in KATX, and 288 images per elevation in KGWX. They all have six elevations individually.

4.1.1. Three-Dimensional Shapes from Two Polygons

Once the sets of maximum six different images (from each elevation) are obtained by the reconstruction algorithm, the toolkit steps up to the modeling by matching

corresponding polygons between each elevation letting to have hull shaped face. But, let us inspect the shapes created by two consecutive elevations first. Figure 4.1 is created by using images taken on 09/05/2018 from the Seattle radar station at 03:21. The top polygon has 409 vertices while the below one has 493. Applied light adjustments are based on the selection of “metal” material, local illumination, ‘gouraud’ face lighting, “0.7” face alpha transparency and “cyan” face color. Each visualized hull shaped object has a specific volume at the space.

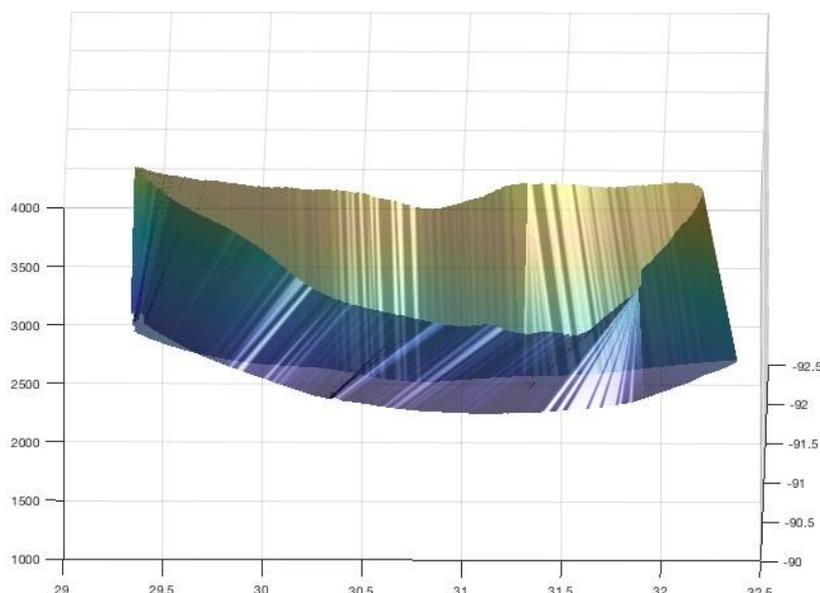


Figure 4.1. 3D view of two consecutively matched polygons (1)

Figure 4.2 was taken on 19/01/2017 from the Columbus AFB radar station at 20:13. The upper polygon has 1924 vertices while the lower polygon has 2231. Applied light adjustments are based on the selection of “shiny” material, local illumination, ‘gouraud’ face lighting, “0.7” face alpha transparency, “magenta” face color, and ambient strength of “1”.

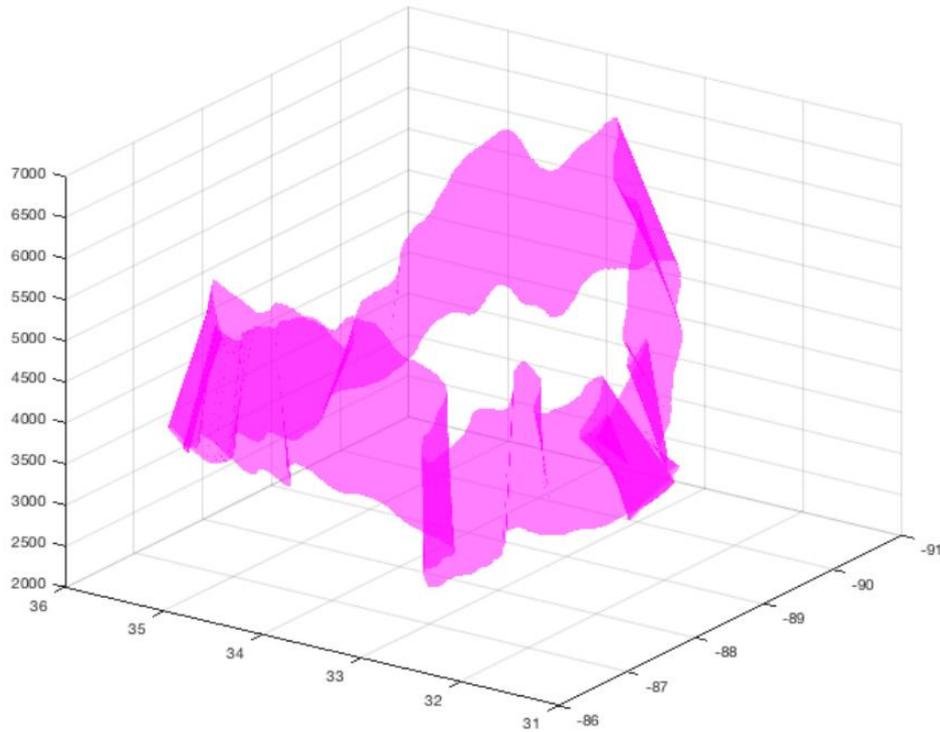


Figure 4.2. 3D view of two consecutively matched polygons (2)

4.1.2. Three-Dimensional Shapes from Multiple Polygons

The connections of the vertices belonging to a specific hydrometeor class create a surface that reconstructs an objects volume (Kleinkort, Huang, Bringi, & Nota, 2017). As mentioned earlier, the polygonal surfaces are presented with face light, transparency and material features. At the same time, the closed volumes of the surfaces with their intrinsic characteristics represent a certain type of precipitation. Moreover, meteorologists can explain what clouds, such as cumulus, stratus, or cirrus, are formed, depending on the structure of height (upper and lower) and volume. There may be many clouds together in the same figure to show the interactions of various hydrometeors. In addition, the location of occurrence let the users compare results

with other instruments. So, the visualization of the objects composed of multiple elevations is important.

Before Figure 4.3 is composed, the visualization code runs for matched polygons two by two until it reaches to a maximum of six elevations. The Toolkit lets the user change the orientation and size of the view interactively. Figure 4.3 is from Seattle radar site. The dataset was taken on 09/05/2018 at 09:00. Four elevations are for light and/or moderate rain type of precipitation. Sequentially 493, 409, 201 and 46 vertices are aligned from bottom to top with the same lighting adjustments as applied in section 3.5.3.1.

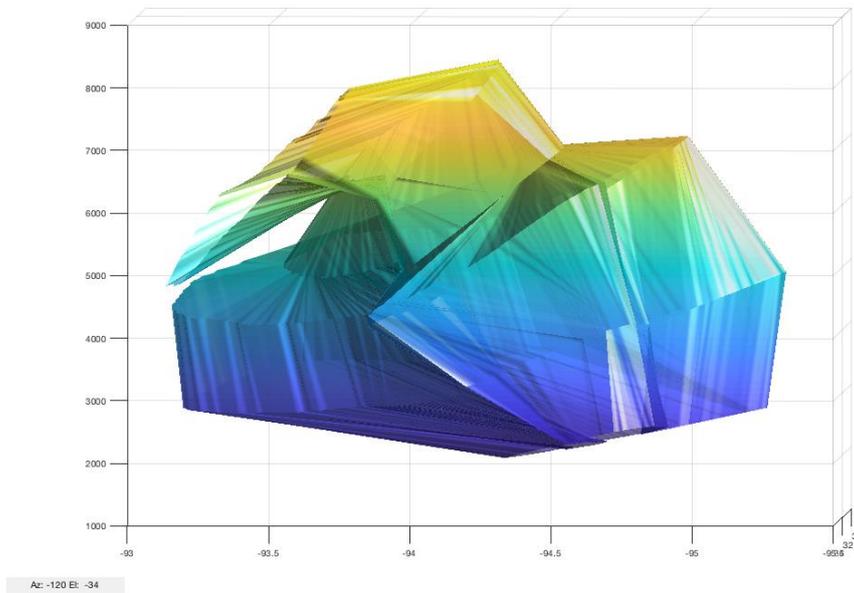


Figure 4.3. Light and/or Moderate Rain type cloud formation sample (1)

Figure 4.4 is from the Seattle radar site. The dataset was taken on 09/05/2018 at 02:26. Six elevations are for light and/or moderate rain type of precipitation. Sequentially 107, 1675, 1224, 805, 537 and 320 vertices are aligned from bottom to top with slightly different lighting adjustments as applied in section 3.5.3.1. These outputs can be exported as an image in various formats (jpg, png, etc.) or as a MATLAB™ figure.

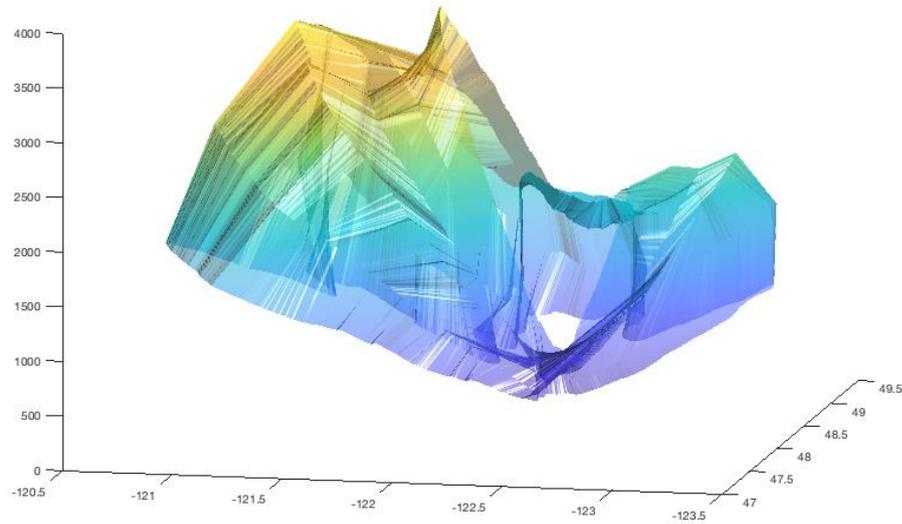


Figure 4.4. Light and/or Moderate Rain type cloud formation sample (2)

The data used in Figure 4.5 was taken on 19/01/2017 from the Columbus AFB radar station at 12:07. Object formations of a single hydrometeor type (dry snow, label 4) are visualized. The structure of the reconstruction algorithm is given in Table 4.1. The “level” column stands for six levels, in which consecutive levels are matched. After the connected component operation on the segmented images, each boundary polygon is numbered (connected component no).

Table 4.1. Properties of the objects (only one label)

Obj_no / Sub-sections	label	level	connected component no	coordinates
Cobjects {1}	4	[1 2 3 4]	[6 5 4 8]	[679×3] [757×3] [382×3] [209×3]
Cobjects {2}	4	[1 2]	[5 4]	[123×3] [38×3]
Cobjects {3}	4	[2 3 4 5]	[1 1 4 2]	[1997×3] [1833×3] [72×3] [161×3]

Cobjects {4}	4	[4 5]	[7 3]	[87×3] [267×3]
Cobjects {5}	4	[4 5]	[3 1]	[419×3] [633×3]
Cobjects {6}	4	[5 6]	[4 3]	[105×3] [257×3]

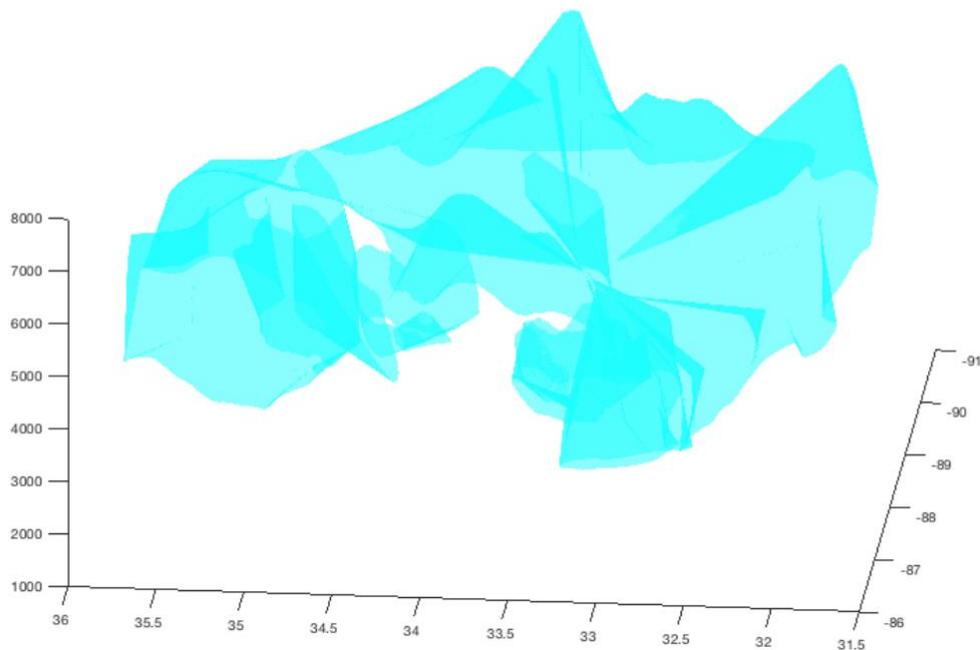


Figure 4.5. Dry snow type cloud formation sample (2)

The data used in Figure 4.6 was taken on 19/01/2017 from the Columbus AFB radar station at 18:38. Objects of existing hydrometeor types (legend at the top-left) are visualized. The structure of the data is given in Table 4.2.

Table 4.2. Structure of the cloud objects (all of the labels)

Obj_no / Sub-sections	label	level	connected component no	coordinates
Cobjects {1}	4	[2 3 4 5 6]	[1 1 1 1 1]	[3500×3] [1898×3] [981×3] [1078×3] [940×3]
Cobjects {2}	5	[1 2]	[7 7]	[51×3] [103×3]

Cobjects {3}	5	[1 2]	[3 2]	[148×3] [55×3]
Cobjects {4}	5	[1 2]	[6 5]	[7×3] [106×3]
Cobjects {5}	5	[2 3 4 5 6]	[4 5 5 2 1]	[66×3] [88×3] [68×3] [3×3] [6×3]
Cobjects {6}	5	[2 3 4]	[6 6 6]	[6×3] [62×3] [26×3]
Cobjects {7}	5	[3 4]	[1 2]	[22×3] [12×3]
Cobjects {8}	5	[3 4]	[2 4]	[26×3] [48×3]
Cobjects {9}	5	[4 5]	[3 1]	[4×3] [42×3]
Cobjects {10}	6	[1 2 3 4 5 6]	[1 1 1 1 1 1]	[1428×3] [1054×3] [815×3] [654×3] [515×3] [413×3]
Cobjects {11}	8	[1 2]	[2 7]	[531×3] [55×3]
Cobjects {12}	8	[2 3]	[2 5]	[98×3] [21×3]
Cobjects {13}	9	[1 2]	[2 1]	[56×3] [105×3]
Cobjects {14}	9	[1 2]	[3 2]	[82×3] [47×3]
Cobjects {15}	9	[2 3 4 5]	[3 3 4 2]	[196×3] [226×3] [141×3] [3×3]
Cobjects {16}	9	[3 4 5 6]	[2 2 1 1]	[132×3] 127×3 [42×3] [25×3]

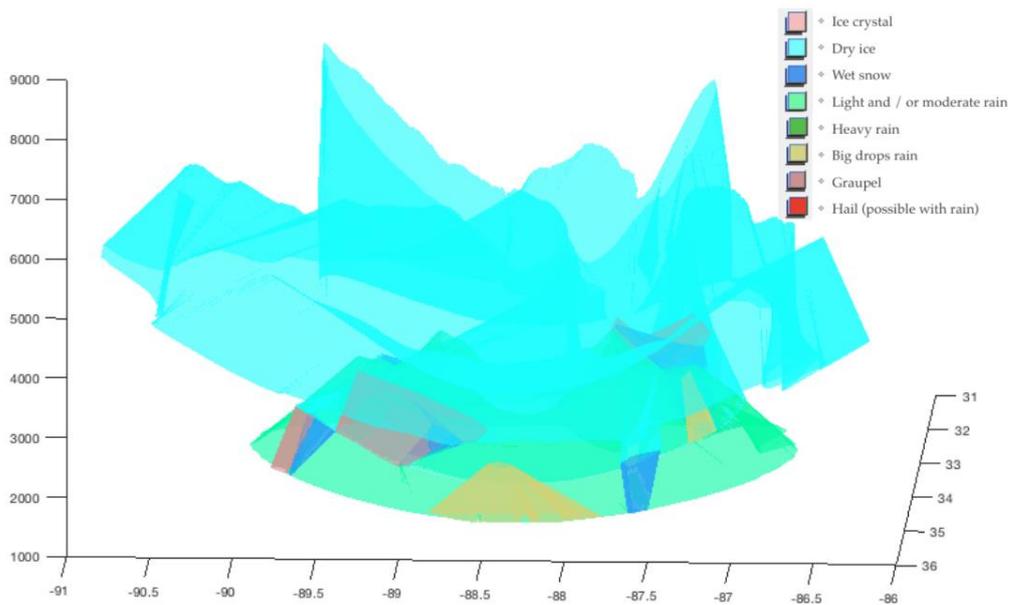


Figure 4.6. Multiple elevation cloud formation of all precipitation types

This example shows how various types of precipitation can be shown simultaneously and hence how the relations between them can be analyzed.

4.2. GUI for the Toolkit: Meteorology3D

Numerical weather models and weather observing systems are producing a vast amount of data. This toolkit can provide efficient access for the benefit of both the meteorologists and the end-users. The way computer graphics is used provides an amazing tool for the availability of the weather data in 3D within a timeline. This toolkit is tuned to the mathematical form of the data, as well as its complexity in the distribution of precipitation types and shape in various 3D formations. Meteorology3D toolkit is a system for interactive visualization of large radar dataset, which can be used for public information, support and analysis purposes. This section shows how to use the basic functionality of the Toolkit.

4.2.1. Read Raw Data and Scale Image

Displaying coordinate and precipitation type parameters within its volume and providing the time evolution are a new approach to understand atmospheric events.

The first part of the toolkit is for loading meteorological radar raw dataset and its visualization. These meteorological raw data exist in the form of scalar fields and were given colors according to the labels in a particular raw image.

As the first part includes the 2D display, the interactivity is carefully increased. The viewer is able to control most of the parameters easily and quickly before applying complex 3D algorithms.

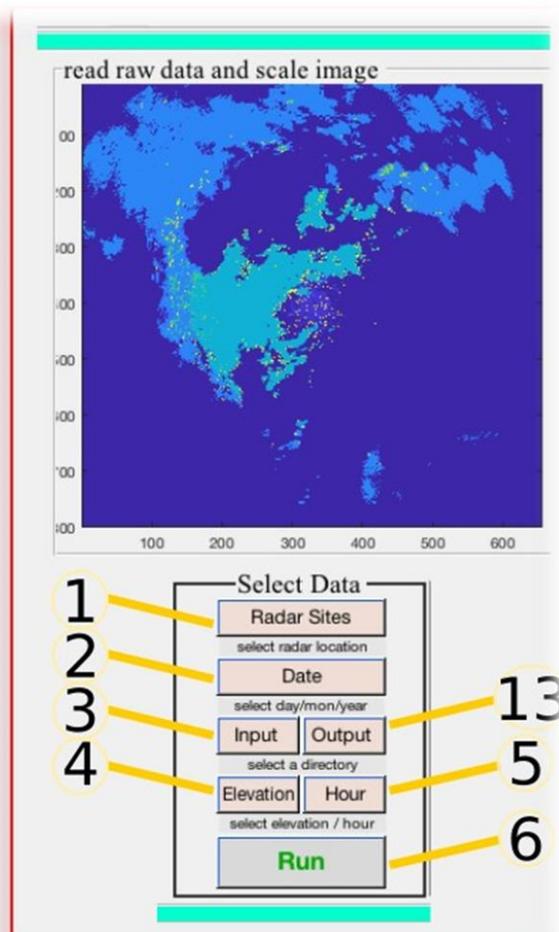


Figure 4.7. Meteorology3D 1st panel

These followings are controlled in this panel and should be selected in order as shown in Figure 4.7.

Procedure:

1. Radar Sites: Selections consist of NEXRAD Radar Sites in the USA, one of which should be chosen.
2. Date: Specifies the Day/Month/Year that the data is taken from the desired NEXRAD Radar
3. Input: This button is responsible for specifying the directory of the Meteorology3D Toolkit.
4. Elevation: Hydrometeor Classification images have six levels of elevations, from which a subset should be selected.
5. Hour: At that specific date, there are multiple images taken depending on the Volume Coverage Pattern. Time of the day should be specified.
6. Run: Once all of these selections are done, the run button gets green, which means the software is ready to display

4.2.2. Segmentation in the GUI

Although the "read raw data and scale image" has one panel, the "Segmentation in the GUI" has two panels. The first panel, called "select label to distribute", has two tasks. While it shows the legend, at the same time it decides which of the specific hydrometeor type will be modeled in the 3D space. On the other hand, the second panel, which is called the "density", is devoted to the selection of density estimation parameters.

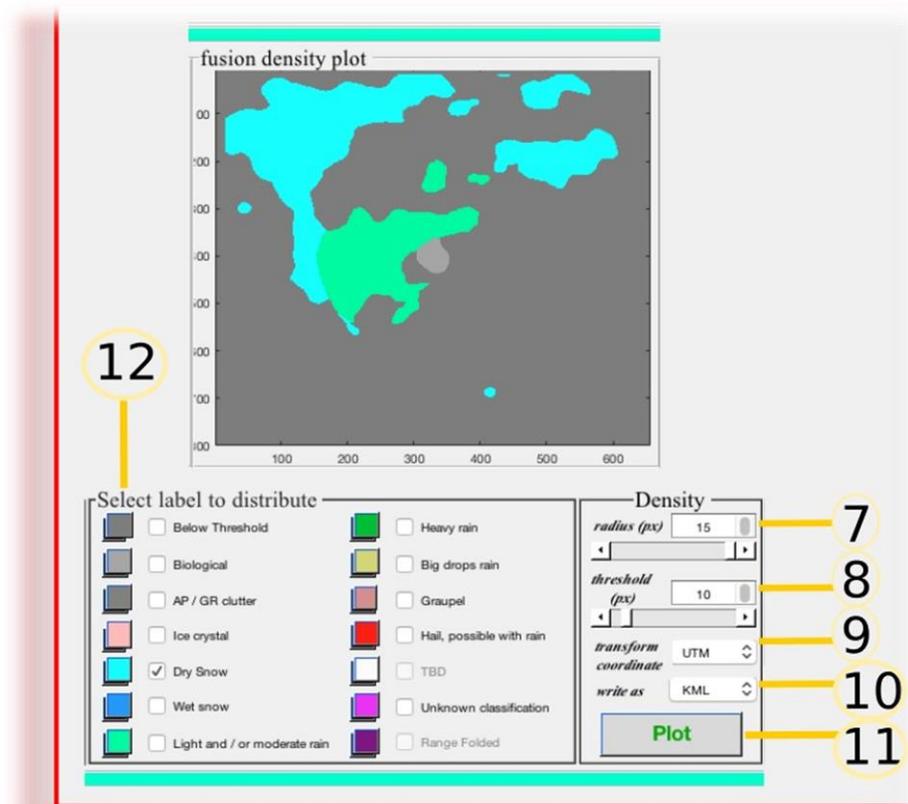


Figure 4.8. Meteorology3D 2nd panel

The following are controlled in this panel and should be selected in order as shown in Figure 4.8:

7. Radius (pixel): The sliding-window radius needs to be determined. The values are between 1 pixel (min 3 is recommended) and 15 pixels.

8. Threshold (%): The size of the objects is specified here. The values are in the range of 1% to 100%.

9. Transform coordinate: The coordinate system can be specified here but currently, the only working option is UTM WGS84.

10. Write as: The selection of file format to be exported. Currently, the only option is KML.

11. Plot: Once all of these selections are done, Plot button gets green, which means the software is ready to display.

12. Legend: Describes the names of the hydrometeors shown in the display.

4.2.3. Object Creation and Visualization

In this panel, located below the "object creation and visualization" figure window, 3D visualization parameters are chosen. The "Segmentation in the GUI" figure window helps users to analyze the connected components. All remaining elevations have raw images, which would be applied to the same image processing steps. After modeling, the connected components merge and gets into the 3D scene.

The following are controlled in this panel and should be selected in order as shown in in Figure 4.9:

12. Select label to distribute: Among 14 labels, the one to be displayed is selected here (Figure 4.8).

13. Output: Within a new window, the directory must be specified for the output files to be sent (Figure 4.7)

14. Edge color: Selection consists of colors for 3D cloud's edges.

15. Face lighting: Selection of lighting arrangement for 3D objects

16. Face color: Selection consists of colors for 3D object's faces.

17. Ambient strength: Ambient light level is between 0 and 1.

18. Face alpha: transparency level is between 0 and 1.

19. Create: Once all of these selections are done, create button gets green, which means the software is ready to display.

20: Duration: Determining the duration of the animation, which has not been implemented yet.

21. Animate: It animates objects according to the duration and FPS (Frame per Second) that has not been implemented yet.

22. Cursors: Once 3D data is displayed, it is important to specify the position of the viewer (ie. camera) relative to the 3D scene. The position can be specified numerically (in the editor); however, it is also possible to use the mouse to navigate in the scene. Clicking and dragging with the mouse in the scene changes the view and it is usually the most convenient. Clicking and dragging the left mouse button results in a rotation, zooming into or out of the scene (getting closer to or further from the rotation center). This rotation is about the current "center of rotation" as specified in the viewpoint panel.

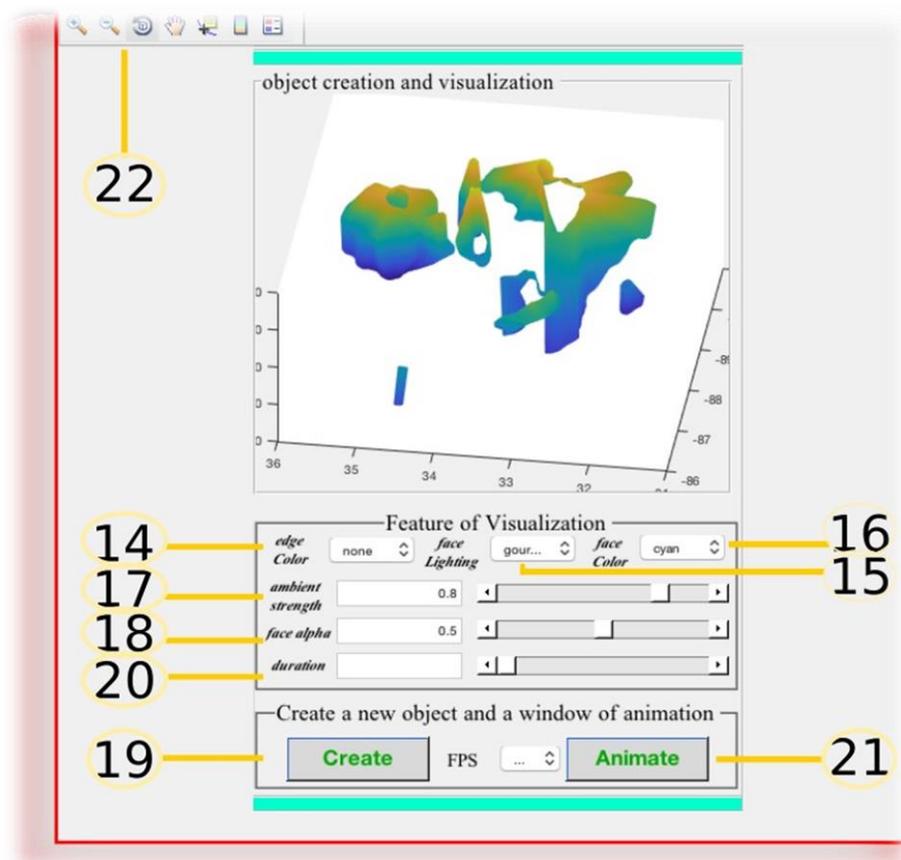


Figure 4.9. Meteorology3D final panel

CHAPTER 5

CONCLUSIONS

5.1. Broader Impact of the Toolkit

This dissertation has proposed and presented a method and techniques for reconstruction and visualization of 3D shapes of hydrometeor clusters based on NEXRAD Level III Hydrometeor Classification images. Among the variety of benefits that our toolkit enables, the main purpose is always having the viewers, specifically, meteorologists, collect information from radar sources to improve the understanding of hydrometeors in 3D with relation to each other. Creating a platform for making management and analysis of weather data easier to the operators, particularly multiple hydrometeor classes in interaction. This will most probably help the operators to generate warnings about hazardous weather conditions with more confidence. Although the numerical models generate significant false alarms in forecasting, the proposed toolkit directly displays the weather radar data enabling the experts to make nowcasting with better understanding the complicated events. To grasp the evolution of the weather system, the meteorologists need to observe the most relevant part of the data with controllable abstraction level.

As the results have shown, the presented algorithms are able to reconstruct 3D shapes and visualize them in a user-friendly manner. The results also underline the capability to get realistic visuals for complicated and diverse shapes. Compositions in various sizes and distributions can be visualized especially when multiple objects are incorporated for displays in Google Earth with KML. When the outcomes are mapped while collecting multiple radar site data, object visuals become beneficial for a survey on geographical conditions. The objects having the longitude, latitude, and altitude

information creates an object formation in 3D space, which would help viewers to figure out the distance to their current location.

Another important feature of the toolkit is that it can display all hydrometeors simultaneously by creating geometric models. By means of 3D computer graphics, the mixture of different hydrometeor objects can show the relationship between the types of precipitation. In this way, the conversion of water bodies in liquid or solid form can be shown. We believe that this software will improve the short-range numerical weather forecasts performed by the radar operators.

5.2. Future Direction

The workflow that has been demonstrated helps the user gain a new perception of the creation of model simulation; at the same time, achieves a better knowledge of atmosphere via discovering 3D patterns. Although reconstruction and visualization are fully functional and the buttons of the animation are created already in the Toolkit, the animation of the created cloud figures in the timeline is not finalized yet. The first step that can be taken is to create the animation part of the toolkit. In addition to that, the background maps can be added to the animations.

Moreover, the shape of 3D formations needs to be improved. The melting layer deserves special attention. Hence, an algorithm to model and visualize this surface in 3D can be added to the library. In addition, the bounding corners of the elevations need better visualization. Because these corners appear sharp in the geometric models. They can be improved by applying smoothing. Furthermore, with respect to the hydrometeor type, the formed clouds may change their appearances (color, light, material, texture, etc.) automatically.

What we have been trying is to make people aware of atmospheric conditions in a better manner by showing them how heterogeneous raw radar data become a 3D model and that can be visualized in Virtual Reality (VR). Hopefully, one can manage to make the same simulation real-time in VR. If this can be realized, people would start using

VR simulations of weather phenomena in their daily life, which would raise the awareness of the atmospheric ever-changing mechanisms.

REFERENCES

- Adan, A., & Huber, D. (2011). 3D reconstruction of interior wall surfaces under occlusion and clutter. *2011 International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission*, 275-281.
- Ahrens, D. C. (2011). *Essentials of meteorology: an invitation to the atmosphere*. Cengage Learning.
- Altius Directory. (2019, 02 19). *MATLAB Programming Language*. Programming Languages List/MATLAB Retrieved from: <http://www.altiusdirectory.com/Computers/matlab-programming-language.php>
- Ansari, S. D. (2010). The weather and climate toolkit. *In AGU Fall Meeting Abstracts*.
- Barry, R. G., & Chorley, R. J. (2009). *Atmosphere, weather and climate*. Routledge.
- Benner, W. T. (2007). MPAR program overview and status. *n Combined Preprints, 87th AMS Annual Meeting, American Meteorological Society*.
- Besl, P. J., & Jain, R. C. (1990). Three-dimensional object recognition. *In Cold Spring Harbor symposia on quantitative biology*, 55, 889-898.
- Beucher, S. (1992). The watershed transformation applied to image segmentation. *Scanning microscopy-supplement*, 299-299.
- Bhanu, B., & Faugeras, O. D. (1984). Shape matching of two-dimensional objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 137-156.
- Boiangiu, C. A., & Ioanitescu, R. (2013). Voting-based image segmentation. *The Proceedings of Journal ISOM*, 7, 211-220.

- Bringi, V. N., & Chandrasekar, V. (2001). *Polarimetric Doppler weather radar: principles and applications*. Cambridge university press.
- Buss, S. R. (2003). *3D computer graphics: a mathematical introduction with OpenGL*. Cambridge University Press.
- Büyükbaş, E. (2005). Training course on weather radar systems. Module B: Radar Hardware. *WMO, Turkey Radar training 1.0*, 64.
- Cho, G. (2012). *3D visualization of in-flight recorded data*. Louisville: The University of Louisville.
- Cho, J. Y., Elkin, G. R., & Parker, N. G. (2005). Enhanced radar data acquisition system and signal processing algorithms for the Terminal Doppler Weather Radar. *In Preprints, 32nd Conf. on Radar Meteorology, Albuquerque, NM, Amer. Meteor. Soc., P4R (Vol. 8)*.
- Collier, C. (2016). *Hydrometeorology*. John Wiley & Sons.
- Comaniciu, D., & Meer, P. (2002). Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 603-619.
- Crum, T. D. (1993). The WSR-88D and the WSR-88D operational support facility. *Bulletin of the American Meteorological Society*, 74(9), 1669-1688.
- Dolan, B., Rutledge, B., & Stevan, A. (2009). A theory-based hydrometeor identification algorithm for X-band polarimetric radars. *Journal of Atmospheric and Oceanic Technology*, 26(10), 2071-2088.
- Donaldson. (1970). Vortex signature recognition by a Doppler radar. *Journal of Applied Meteorology*, 661-670.
- Dual-Pol Training. (2019). *Weather.gov*. Retrieved from: <https://www.weather.gov/media/lmk/soo/DualPol-Flipchart.pdf>

- Durden, S. L. (1994). ARMAR: An airborne rain-mapping radar. *Journal of Atmospheric and Oceanic Technology*, 11(3), 727-737.
- Felzenszwalb, P. F., & Huttenlocher, D. P. (2004). Efficient graph-based image segmentation. *International journal of computer vision*, 167-181.
- Forsyth, D. E. (1981). Modular radar analysis software system/MRASS. *In Conference on Radar Meteorology, 20 th, Boston, MA*, 696-699.
- Fukao, S., Hamazu, K., & Doviak, R. J. (2014). *Radar for meteorological and atmospheric observations*. Springer Japan.
- Gibson, M. S. (2014, Apr). *Gr2Analyst*. Gibson Ridge Software, Retrieved from: http://www.grlevelx.com/gr2analyst_2/volume_renderer.htm
- Gortler, S. J. (2012). *Foundations of 3D computer graphics*. MIT Press.
- Han, D. (1991). *Weather radar information processing and real-time flood forecasting*. University of Salford, UK, Doctoral dissertation.
- Harper, K. (2014). *Weather and climate: decade by decade*. Infobase Publishing.
- Higgins, M. L. (1996). *Autonomous Segmentation of 3D Radar Imagery*. University of Massachusetts. Lowell).
- Hilaga, M., Shinagawa, Y., & Kohmura, T. &. (2001). Topology matching for fully automatic similarity estimation of 3D shapes. *In Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, 203-212.
- Istok, M. J., Fresch, M., Jing, Z., & Smith, S. D. (2009). *15.5 WSR-88D Dual Polarization Initial Operational Capabilities*. NOAA. Retrieved from: roc.noaa.gov.
- Ivić, I. R. (2018). Options for polarimetric variable measurements on the MPAR advanced technology demonstrator. *In 2018 IEEE Radar Conference (RadarConf18)*, pp. 0129-0134.

- Kleinkort, C., Huang, G. J., Bringi, V. N., & Nota. (2017). Visual hull method for realistic 3D particle shape reconstruction based on high-resolution photographs of snowflakes in free fall from multiple views. *Journal of Atmospheric and Oceanic Technology*, 34(3), 679-702.
- Lhermitte, R. M. (1970). Dual-Doppler radar observations of convective storm circulation. *Conference on radar meteorology (preprints)*, 139-144.
- Lu, M., Chen, M., Wang, X., Yu, M., & Jiang, Y. &. (2018). 3D modelling strategy for weather radar data analysis. *Environmental earth sciences*, 77(24), 804.
- Majumdar, N. (2012). *MATLAB Graphics and Data Visualization Cookbook*. Packt Publishing Ltd.
- Markowski, P., & Richardson, Y. (2011). *Mesoscale meteorology in midlatitudes (Vol. 2)*. John Wiley & Sons.
- Mathworks Inc. (2019). *MATLAB GUI*. 02 19, 2019 tarihinde Create Apps with Graphical User Interfaces in MATLAB, Retrieved from: <https://www.mathworks.com/discovery/matlab-gui.html>
- MathWorks Inc. (2017). 3-D Visualization R 2017 a. Natick, MA, USA.
- Nathanson, F. E., Reilly, J. P., & Cohen, M. N. (1991). *Radar design principles-Signal processing and the Environment*. NASA STI/Recon Technical Report A.
- NOAA. (2015). *Research Tools: Multifunction Phased Array Radar*. National Severe Storms Laboratory, Retrieved from: <https://www.nssl.noaa.gov/tools/radar/mpar/>
- Paramore, S. W. (2006). *Washington, DC: U.S. Patent No. U.S. Patent No. 7,109,913. Patent and Trademark Office.*
- Park, H. S., Ryzhkov, A. V., Zrnić, D. S., & Kim, K. E. (2009). The hydrometeor classification algorithm for the polarimetric WSR-88D: Description and application to an MCS. *Weather and Forecasting*, 24(3), 730-748.

- R. C. Gonzalez, R. E. (2009). *Digital Image Processing Using MATLAB®*. Gatesmark Publishing.
- Rabin, R. Z. (1980). Subsynoptic-scale vertical wind revealed by dual Doppler-radar and VAD analysis. *Journal of the Atmospheric Sciences*, 644-654.
- Radar Operation Center. (2008, January 10). *NWS WSR-88D Level II Data Collection and Distribution Network Information*. NOAA. Retrieved from: https://www.roc.noaa.gov/WSR88D/Level_II/Level2Info.aspx
- Radar Operation Center. (2019). *WSR-88D Volume Coverage Pattern (VCP)*. NOAA.
- Rauber, R. M., & Nesbitt, S. L. (2018). *Radar Meteorology: A First Course*. John Wiley & Sons.
- Rautenhaus, M., Kern, M., Schäfler, A., & Westerman. (2015). Three-dimensional visualization of ensemble weather forecasts—Part 1: The visualization tool Met. 3D (version 1.0). *Geoscientific Model Development*, 2329-2353.
- Rogers, R. R. (1990). The early years of Doppler radar in meteorology. *American Meteorological Society, Boston, MA.*, 122-129.
- Saha, K. (2008). *he Earth's atmosphere: Its physics and dynamics*. Springer Science & Business Media.
- Sauvageot, H., & Omar, J. (1987). Radar reflectivity of cumulus clouds. *Journal of Atmospheric and Oceanic Technology*, 264-272.
- Seo, J., Chae, S., Shim, J., Kim, D., & Cheong, C. (2016). Fast contour-tracing algorithm based on a pixel-following method for image sensors. *Sensors*, 16 (3), 353.
- Shotton, J., Blake, A., & Cipolla, R. (2008). Multiscale categorical object recognition using contour fragments. *IEEE transactions on pattern analysis and machine intelligence.*, 30(7), 1270-1281.

- Silverman, B. W. (2018). Density estimation for statistics and data analysis. *Routledge*.
- Solomon, C. J., & Seymour, P. (1976). A case study of a young child doing Turtle Graphics in LOGO. *national computer conference and exposition*.
- Stackoverflow. (2011, February 23). *Image Segmentation using Mean Shift explained*. Retrieved from: <https://stackoverflow.com/questions/4831813/image-segmentation-using-mean-shift-explained>
- Stagliano Jr, J. J. (2005). Polarimetric weather radar base moment algorithm validation. *Conference on Radar Meteorology*.
- Stevens, B. (2010). *Twelve Lectures on Cloud Physics*. mpimet.mpg.de: Retrieved from: <http://www.mpimet.mpg.de/fileadmin/staff/stevensbjorn/teaching/skript-5.pdf>
- Szyrmer, W., & Zawadzki, I. (1999). Modeling of the melting layer. Part I: Dynamics and microphysics. *Journal of the atmospheric sciences*, 56(20), 3573-3592.
- Timothy, C. D. (1993). Recording, archiving, and using WSR-88D data. *Bulletin of the American Meteorological Society*, 74.4, 645-654.
- University Corporation for Atmospheric Research. (2009). *Precipitation Estimates, Part I: Measurement*. The Comet Program, Retrieved from: http://stream1.cmatc.cn/pub/comet/QPF-QPE/PrecipitationEstimatesPart1Measurement/comet/hydro/precip_est/part1_measurement/print.htm
- University Corporation for Atmospheric Research. (2009). *Precipitation Estimates, Part I: Measurement*. The Comet Program, Retrieved from: http://stream1.cmatc.cn/pub/comet/QPF-QPE/PrecipitationEstimatesPart1Measurement/comet/hydro/precip_est/part1_measurement/print.htm

- US Department of Commerce. (2006). *OFCM*. Federal Meteorological Handbook NO. 11: Part D, Retrieved from: <https://www.ofcm.gov/publications/fmh/FMH11/FMH11D-2006.pdf>
- US Department of Commerce. (2017). *Office of the Federal Coordinator for Meteorological Services and Supporting Research*. NOAA, Retrieved from: <https://www.ofcm.gov/publications/fmh/FMH11/fmh11partC.pdf>
- Van Kaick, O., Hamarneh, G., Zhang, H., & Wighton, P. (2007). Contour correspondence via ant colony optimization. *In 15th Pacific Conference on Computer Graphics and Applications*, 271-280.
- Whiton, R. C., Smith, P. L., Bigler, S. G., & Wilk, K. (1998). History of operational use of weather radar by US weather services. Part II: Development of operational Doppler weather radars. *Weather and forecasting*, 244-252.
- Winkelbach, S., & Wahl, F. M. (2008). Pairwise matching of 3D fragments using cluster trees. *International Journal of Computer Vision*, 78(1), 1-13.