COMENIUS UNIVERSITY IN BRATISLAVA
FACULTY OF MATHEMATICS PHYSICS AND INFORMATICS

3D POSITION RECONSTRUCTION OF REENTRY OBJECTS FRAGMENTS USING TWO VIDEO RECORDINGS.

Master thesis

2024

Bc. Damián Gorčák

COMENIUS UNIVERSITY IN BRATISLAVA
FACULTY OF MATHEMATICS PHYSICS AND INFORMATICS

3D POSITION RECONSTRUCTION OF REENTRY OBJECTS FRAGMENTS USING TWO VIDEO RECORDINGS.

Master thesis

Study program:	Applied informatics
Branch of study:	Informatics
Department:	Department of Applied Informatics
Supervisor:	Mgr. Daniel Kyselica
Consultant:	Mgr. Jiří Šilha, PhD.

Bratislava, 2024

Bc. Damián Gorčák

54062234

Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta:	Damián Gorčák
Študijný program:	aplikovaná informatika (Jednoodborové štúdium, magisterský II. st., denná forma)
Študijný odbor:	informatika
Typ záverečnej práce:	diplomová
Jazyk záverečnej práce:	anglický
Sekundárny jazyk:	slovenský

Názov: 3D position reconstruction of reentry objects fragments using two video recordings.
3D rekonštrukcia polohy fragmentov vtupujúcich objektov pomocou dvoch videozáznamov.

Anotácia: Na obmedzenie vesmírneho odpadu na obežnej dráhe Zeme by sa mali odstrániť nefunkčné satelity a telesá rakiet.
Jedným z mechanizmov, ako to urobiť, je manévrovanie objektu do zemskej atmosféry, keď pomaly klesá. Tento pád sa nazýva reentry a pozostáva zo svetelnej fázy, ktorú môžu astronómovia pozorovať. Účinok veľmi podobný meteoru/ohnivej guli často pozorovateľný voľným okom. Videozáznamy tej istej udalosti z viacerých observatórií poskytujú informácie o 3D polohe jednotlivých fragmentov objektu. Takéto informácie môžu astronómovia použiť na určenie mnohých informácií o padajúcom objekte: dynamické vlastnosti, štartovacia dráha atď. Naša fakulta disponuje záznamami viacerých udalostí opätovného vstupu asteroidov a umelých objektov. V minulosti bola vyvinutá experimentálna metóda na nájdenie párov segmentov medzi dvoma videonahrávkami. Tento postup využíva klasické metódy na sledovanie a párovanie prvkov. Grafové neurónové siete ukázali dobrý výkon na dátach so silnou priestorovou štruktúrou, ktorá dobre vyhovuje nášmu problému so zhlukom pohyblivých segmentov.

Cieľ: Študovať literatúru o 3D mapovaní, grafových neurónových sieťach, 3D rekonštrukcii z videa. Nájdite a implementujte nové metódy
rekonštruovať 3D pozície pohyblivých segmentov vrátane generovania trénovacích dát.

Literatúra: Reliable Feature Matching AcrossWidely Separated Views (https://ieeexplore.ieee.org/document/855899)
StickyPillars: Robust and Efficient Feature Matching on Point Clouds using Graph Neural Networks (https://arxiv.org/abs/2002.03983)

Kľúčové slová: Graph neural networks, space debris

Vedúci:	Mgr. Daniel Kyselica
Konzultant:	Mgr. Jiří Šilha, PhD.
Katedra:	FMFI.KAI - Katedra aplikovanej informatiky
Vedúci katedry:	doc. RNDr. Tatiana Jajcayová, PhD.

54062234

Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

Dátum zadania: 20.09.2022

Dátum schválenia: 20.09.2022

prof. RNDr. Roman Ďurikovič, PhD.
garant študijného programu

študent

vedúci práce

I hereby declare that I have written this thesis by myself, only with help of referenced literature, under the careful supervision of my thesis advisor.

. . . .

Bratislava, 2024 Bc. Damián Gorčák

Acknowledgement

First, I would like to express my gratitude to Mgr. Jiří Šilha, PhD. for his guidance during the whole thesis and invaluable expertise in astronomy that made this thesis possible. I'd also like to thank my supervisor Mgr. Daniel Kyselica for his valuable advice and assistance. A special thanks to prof. RNDr. Roman Ďurikovič, PhD. for his insightful feedback and organization of YACGS seminars. And lastly, I'd like to express my deepest thanks to my partner and my family for supporting me during my academic years.

Abstract

Keywords: space debris, machine learning, space object classification

Abstrakt

Kľúčové slová: vesmírny odpad, strojové učenie, klasifikácia vesmírnych objektov

Contents

List of Figures

List of Tables

Terminology

Terms

- **Star field tracking (sidereal)**
  Ground-based tracking mode in which, telescope is moving in the same direction and speed as the apparent motion of stars.
- **Object tracking**
  Tracking mode, where the focus is aimed at the moving object of interest and the telescope is moving in the same way.
- **Survey**
  Observation of a region of the sky when no specific target is defined.
- **Star catalog**
  A list of stars with its positions and magnitude.
- **Star tracker**
  An optical device usually used to determine the orientation of satellite using positions of the stars.
- **Deblending**
  The process of separating overlapping objects.

Abbreviations

- **CCD** - Charge-Coupled Device.
- **IAA** - International Academy of Astronautics.
- **USSSN** - US Space Surveillance Network.

- **CNN** - Convolutional Neural Network.
- **FC** - Fully-Connected.
- **RSO** - Resident Space Object.
- **ML** - Machine Learning.
- **SDSS** - Sloan Digital Sky Survey.
- **PCA** - Principal Component Analysis.
- **ANN** - Artificial Neural Network.
- **NN** - Neural Network.
- **MLP** - Multi-Layer Perceptron.
- **R-CNN** - Region-based Neural Network.
- **MS COCO** - Microsoft Common Objects in Context.
- **AGO** - Astronomical and Geophysical Observatory in Modra.
- **AGO70** - The Newtonian telescope at AGO, with 70 cm parabolic mirror.
- **ESA** - European Space Agency.
- **PECS** - Plan for the European Cooperating States.
- **FMPI** - Faculty of Mathematics, Physics and Informatics.
- **FITS** - Flexible Image Transform System.
- **RADEC** - Right Ascension and Declination.
- **FOV** - Field Of View.
- **PSF** - Point-Spread Function.
- **FWHM** - Full Width at Half Maximum.
- **ADU** - Analogue-to-Digital Unit.
- **ADC** - Analog to Digital Converter.
- **SVM** - Support-Vector Machine.
- **ResNet** - Residual Neural Network.
- **ILSVRC** - ImageNet Large Scale Visual Recognition Challenge.

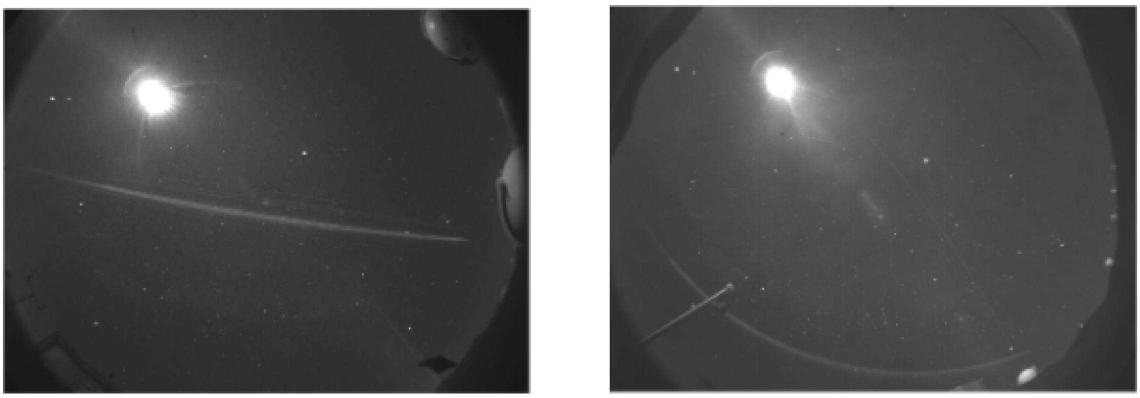
- **RELU** - Rectified Linear Unit.
- **TSV** - Tab-Separated Values.
- **CLI** - Command Line Interface.
- **YAML** - YAML Ain't Markup Language.

Chapter 1

Introduction

1.1 Problem definition

Space debris is a huge problem for humanity. Each year, many objects, whether small or large, reenter the Earth's atmosphere[1]. A very small number of these reentries have been recorded by Amos cameras . Since reentry recordings are not typical meteor recordings (for which there is already a good tool for object detection in time), and reentry recordings contain far more fragments than the current meteor reduction approach can handle, there is a need to find a new way of reducing reentry objects. For this purpose, we will introduce our new method of processing pairs of video recordings with the help of Superglue.



(a) (b)

Figure 1.1: Example of Summary images of reentry space rocket. 1.1a and 1.1b shows images from two Amos cameras situated in Hawaiian islands- AMOS-HK and AMOS-MK. Taken from [2]

1.2 Space debris

In this section, we will introduce the issue of space debris. Then definition of what it is and an overview of the different types of debris will be discussed. As follow uncontrolled reentry problem will by discuss. Finally, we will introduce a solution for reducing uncontrolled reentry space debris.

1.2.1 Definition

> Space debris is any type of space object that is man-made, no longer in active use, and in Earth's orbit.[1]

Those can be whole space aircraft, tools, or waste that was left with some space. As space debris is also considered, many fragments, which are produced through collisions. Involving larger existing debris stemming from either accidental or intentional. Debris can be of all sizes from microscopic particles to large-sized space aircraft. These space debris exist in range from 160 to 36,000 km above our Earth's surface. And can be very dangerous even small particles due to its enormous speed.[1]

According to [3] there are these options how to clean space debits:

- Tug for uncontrolled reentry- object is caught, and adjusted its orbit so it can reenters the atmosphere freely into some place which is not predesignated.
- Tug for controlled reentry- object caught, and adjusted its orbit so it can reenters the atmosphere at specific angle and land in some specific area.
- Space laser nudge- This technique uses laser to move object without physical contact from space.
- Recycling debris - Gather and process debris in space for using it as fuel or other utilities.
- Just-in-time collision avoidance - Rapid response rocket would meet with specific debris and alter the target debris orbit (Can be also done with laser nudge). This technique is meant for prevent collision between large orbital debris.
- Ground laser nudge - uses a laser to move an object without physical contact from the surface of the earth.

- Physical sweeping - directly impacting debris for relocating or moving

1.2.2 Reentry

Is one of the biggest engineering problem. If engineers wants to space aircraft land at earth ground in one piece (unlike smaller debris which burns up when it encounters atmosphere) The spacecraft must stay cool, which is not simple task. Due to high speed which object reentry object is generating [4].

In the article [5] the authors mentions of as of 15 July 2021, a total number of 3646 payloads, 3993 rocket bodies and 17,880 orbital debris have reentered in the Earth's atmosphere. They also said that since humanity started to send object into space, there was an average of 1 intact object (rocket body or payload) every 3 days plus one piece of debris reenters each 31 hour.

Uncontrolled reentry can be very dangerous. If we consider time slot during first of January 2010 to thirty first of December 2020 there re-entered on average 67 payloads, 42 rocket bodies and 287 debris per year. When we consider just object classified as large(radar cross section $>$ 1 m2). 214 payloads and 417 rocket bodies with combined mass of 1113 tones was reentering uncontrolled. This can be count on average 100 tones of debris was reentering uncontrolled.[5]

In [6] authors found two ways of counting future rocket body reentry risk. The first of them is to find all rocket bodies which have a perigee less than 600 km. Those objects are considered to reenter the atmosphere coming decades. Objects with perigee more than 600 km require much longer timescales. In the year where the article was published (2022), there were 651 rocket bodies which satisfies the perigee limit. With this method, they got results of 0.01 casualties per square meter of casualty area. The second method was to take the trend of rocket body reentries over the past 30 years and apply this to the next 10 years, with an output of casualty risk of 0.006 per square meter of casualty area. When we assume that each reentry spreads fragments of debris into an area 10 m2 they got 10 percent chance of hitting one or more humans with space debris in the next 10 years (meaning from 2022 to 2032). They also discussed that consequences could be more fatal for example if debris falls in another country it can invoke political tension, or if the airplane is hit more people can die as only a 300 g piece of debris could catastrophically damage an airliner in flight. With more and more planned space crafts launch casualty risk is increasing.

1.2.3 Phases of reentry

There are several steps in space reentry. according to [2] there are these steps of space reentry:

- Partial fragmentation- external objects with high resistance (low weight or large area) such as solar panels will begin to fall off. this happens usually 100km above Earth's surface.
- Main fragmentation/explosion- Cloud of fragments are created when parent body is fully fragmented. This is caused by thermodynamic forces.
- Light flight (luminous trajectory)- As the fragments undergo ablation within the Earth's atmosphere, interacting with air molecules, a mesmerizing meteor-like phenomenon is generated. This captivating event can be observed from the ground observer
- Dark flight- when fragments slower down, no ablation is present and fragments starts to free fall.
- impact- when fragments fall down to Earth's surface. Size and speed is depend on previous steps.

1.2.4 AMOS all-sky system

AMOS (All-sky MeteorOrbit System) is meteornetwork consisting of all-sky meteor video cameras. Its been developed by Comenius University in Bratislava since 2007. Amos consists of four major components:

- fish-eye lens
- image intensifier
- projector
- digital video camera - with an image resolution of 1600x1200 pixels with 20 fps (frames per second).

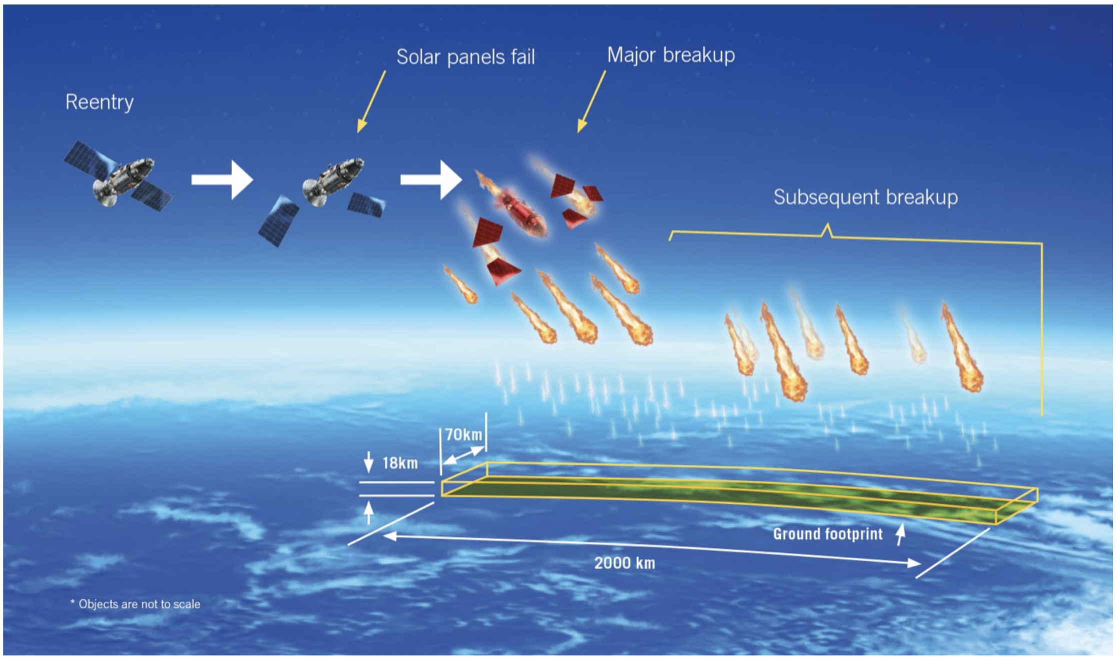


Figure 1.2: Space reentry phases. Taken from [7]

. Amos cameras are deployed all over the world. These cameras are usually installed in pairs of two, for purpose of securing the triangulation for meteor trajectory measurement. Distance between these cameras varies from 80 to 150 km and are operated remotely. There are five cameras are in Slovakia, two on Canary Islands (La Palma, Tenerife), two in Chile (San Pedro de Atacama, Paniri Cau - Chiu-Chiu), two in the USA (Hawaiian Islands) and three in Australia and two in South Africa. These locations establish a global network of stations to comprehensively monitor the influx of meteoroid particles and derbis into the Earth's atmosphere.

1.3 features matching

Feature matching or generally image matching is in general estimating correspondences between of features in two sets of pictures. It improves abilities of systems which uses analysis and interpretation of visual data. Feature matching is and important part in various tasks. Such as image registration, camera calibration and object recognition, which is the task of establishing correspondences between two images of the same scene/object. In [8] are mentioned these steps, how is local feature matching generally performed:

- detecting interest points (keypoints)

- computing visual descriptors
- matching these with a Nearest Neighbor (NN) search
- filtering incorrect matches.
- estimating a geometric transformation.

1.3.1 Feature

Feature is an informational element which is relevant for solving computational task related to a certain application. Features in image can be specific structures such as points, edges or objects. Features can be detected using general neighborhood operation or via feature detection applied to the image. Features can be classified into two main categories: First type are Features located in specific locations of the images. Those can be mountain peaks, building corners or doorway. Second type of features are those which can be matched based on their orientation and local appearance. These may also serve as indicators for identifying object boundaries and occurrences of occlusion events within the sequence of images.[9]

1.3.2 Feature descriptors

Feature descriptors are vectors which contains interesting information information about feature. They are like fingerprints that differentiate one feature from another. Ideally descriptors would be invariant invariant under image transformation. So when picture is transformed we will be able to make same feature again.[10]

1.3.3 Existing solutions

In [11] are mentioned this solutions FAST (Features from accelerated segment test), ORB (Oriented FAST and Rotated BRIEF), SURF (Speeded-Up Robust Features), SIFT (Scale-Invariant Feature Transform).

There is also one relatively new solution for feature matching. It is Superglue.

1.4 SuperGlue: Learning Feature Matching with Graph Neural Networks

Superglue is a new technique of matching local features in two sets of pictures. It is learnable middle-end which takes as an input keypoints. Keypoints are local featrures which consists of descriptors (brightness, color etc.) and position of keypoint. Output of Superglue are matches between those keypoints. It uses learnable (automated) looking for keypoint, where no heuristic or human interaction is no needed. Superglue architecture can be decomposed into two parts frond-end and back-end. Front-end takes as input set of two images and decomposes it into keypoints locations and their descriptors (descriptors could be for example: light in particular pixel color and so on.). This method is trained end-to-end from image pairs – priors for pose estimation are learned from a large annotated dataset, enabling SuperGlue to reason about the 3D scene and the assignment.

** context agregation ->refers to the process of gathering and combining information from different sources or contexts to make more informed decisions or improve the understanding of a particular problem. ** bundle adjustment -> technique of estimating point location in 3d based on camera location and visual ray

1.4.1 Architecture of Superglue

SuperGlue architecture can be seen on image 1.3. It can be decomposed into two major components: the attentional graph neural network and the optimal matching layer.

First component uses keypoint encoder for mapping descriptors d and positions p into a single vector. This encoder is implemented as a multilayer perceptron, which is a type of neural network architecture. Vector mentioned above is passed into graph neural network[link] with attention GNNa. GNNa computes both self attetion and cross attention. Purpose of self attention is to boost receptive field of local descriptors. Cross attention is inspired by the way humans compare photos (by looking back-and-forth) and it enables cross-image communication. Result of this is representation of each point (in figure 1.3 it is labeled as f).

Second component optimal matching layer creates an M by N score matrix, which is augmented with dustbins. In the final it finds the optimal partial assignment using the Sinkhorn algorithm [8].

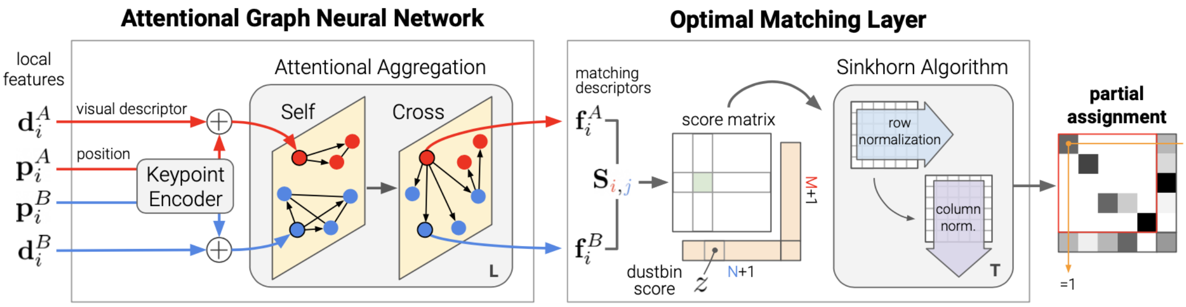


Figure 1.3: Architecture of superglue library in python. Taken from [8]

1.5 Graph neural network (GNN)

> A Graph Neural Network is an optimizable transformation on all attributes of the graph (nodes, edges, global-context) that preserves graph symmetries (permutation invariances) [12].

Nowadays, attention to the research of graph analysis using machine learning is increasing due to the great expressiveness of graphs. Graphs are used to express a large number of different systems, such as: social science (social network), natural science (physical systems), protein-protein interaction networks, and many others. [13].

1.5.1 Graphs

We can interpret many things using graphs, such as interpersonal relationships, social networks, interactions between individuals, the interaction of drugs with people, and even public transportation (public transit). Graphs are an effective tool for visualizing and analyzing relationships between different entities. For example in interpersonal relationships, graphs could depict how people are connected through family, friendship, or work relationships. Other example could be social network where each connection is subset of whole network of connections.

In this article [14], the authors address graphs of the following types:

- Undirected graphs- is a pair (V, E), where (V) is a finite set of vertexes, and E is a subset of $P_2(V)$ and is called edge. $P_2(V)$ is the set of all two element subset of V. The text form of such graphs is:

$$(\{2, 1, 4, 5\}, \{\{1, 2\}, \{1, 4\}, \{4, 2\}\}).$$

. Such graphs are used to express the relationships in finite set. More specifically said relations that are binary, symmetric and ireflexive. [15] With additional information of direction we can provide more information about relation as for example representing parent (start node) and child (end node) relationship.[14]

- Directed graphs is a pair (V, A), where (V) is a finite set of vertexes, and A is a subset of VxV where VxV is the matrix which is expressing directions of edges. Simply said digraphs are graphs in which edges have directions. In those graphs each edge have specified a starting point and ending point. edges in such graphs are drawn as arrows. [15]

- Heterogeneous Graphs. Those type of graphs consists of different node types. for this purpose method Graph Inception was created,. This method groups and clustered different neighbours to be utilised as a whole. Those cluster are called sub-graphs and are used for parallel calculation.[15]

- Dynamic Graph. Structure of this kind of graphs is changing over time. This mean that vertexes are added, updated or deleted the same for edges. Also inputs of such graphs might be dynamic.[15]

- Attributed Graph. Edges in graphs include additional information like weights or type of edge. having graph like this is more manageable when working with relational data[15]

1.5.2 Prediction tasks on graphs

The authors in this [12] classify prediction tasks on graphs into three main types: graph-level, node-level, and edge-level.

Goal in graph-level task is to predict property of whole graph. for example lets assume molecule which is represented as graph. There we can predict what the molecule smells like or how it affects receptors implicated in desease.[12]

Node-level tasks involve predicting the identity or role of each node in the graph. This type of tasks can be used for image segmentation where we are trying to label role of each pixel in the whole image.[12]

Edge level tasks. Purpose of them is to understand and predict relationships between pairs of nodes in a graph. One example can be image scene understanding, where beyond object recognition is important to define (label) relationship between objects.[12]

1.5.3 General GNN

The main building block of a graph neural network is message propagation. It iteratively aggregates neighbors for a central node based on graph neural networks. Message propagation usually consists of two steps: message passing and state update [16]. Message propagation is in [16] formulated as follows: in each layer of GNN we assume that state of node u at time t is $\mathrm{h}_u^t \in R^d$. N_u denotes set of neighbours user u and $edge_{uw} \in R^d$ is representation of edge between u and one of it's neighbour w. Then the state of user u can be updated as:

$$\mathrm{m}_u^{t+1} = \sum_{w \in N_u} M(\mathrm{h}_u^t, \mathrm{h}_w^t, edge_{uw}) \tag{1.1}$$

$$\mathrm{h}_u^{t+1} = U(\mathrm{h}_u^t, \mathrm{m}_u^{t+1}) \tag{1.2}$$

Where $m_u^{t+1} \in \mathbb{R}^d$ is message received from user u at time t and d is denotes dimension. So the state $\mathrm{h}_u^{t+1} \in \mathbb{R}^d$ is computed by aggregating it's own state at time t and received message. Example of message propagation is shown in figure below 1.4.

In Graph Neural Networks (GNNs), There is multiple layers. GNN is able to aggregate a central node multiple steps of neighbours in the graph. Aggregation operation and update operation are used by GNN for learning high-level representations of node. For purpose of extracting representation of central node aggregation is used. Aggregation collects data from its neighbours. When representation of central node, and it's neighbors is determined, then update operation can be used for computing the latest representation of the center node.[16]

In [16] author discussed this strategies for aggregation operation: mean-pooling and attention mechanism. In update operation they mentioned these strategies for

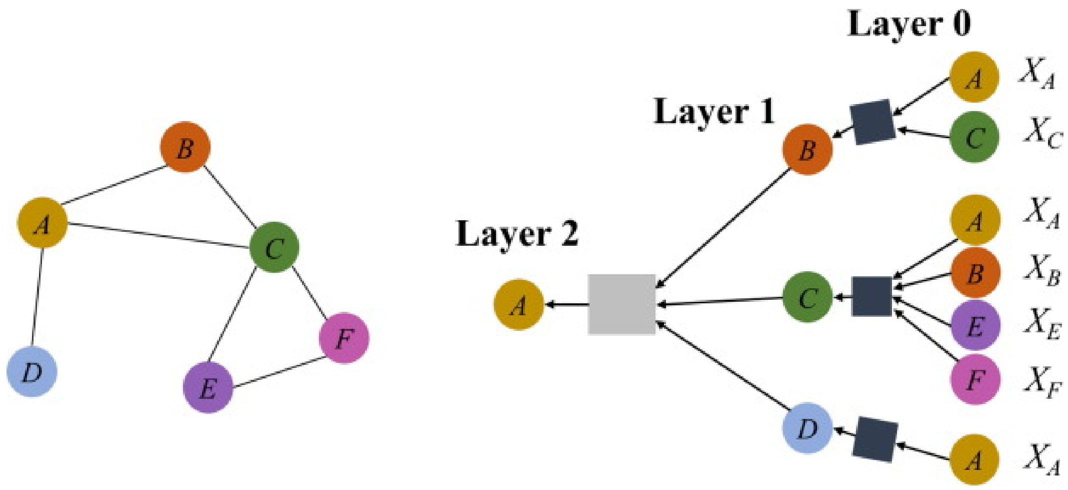


Figure 1.4: Example of message propagation on simple 2 layer network. Taken from [16]

obtaining central node's latest representation: concatenation operation, sum operation and GRU mechanism.

There are two main types of GNN models: Spectral models and spatial models. Spectral models apply graph convolution to solve graph tasks. Spatial models gathers features directly through aggregation. Both of these models firstly itteratively collect a central node's neighbors information. After this step they aggregate these information to derive high-level representation of this node.[16]

The authors of [16] point out these models in their work: GCN, GraphSage, GAT, HetGNN.

- **GCN**: is a spectral model. It combines graph convolution and neural networks to obtain a representation of higher order representation. Node representations are updated by:

$$H^{l+1} = \delta\left(\widetilde{D}^{-\frac{1}{2}}\widetilde{A}\widetilde{D}^{-\frac{1}{2}}H^l W^l\right) \tag{1.3}$$

  where δ is and activation function, $H^l \in \mathbb{R}^{|V|\times d}$ is the representations in the l-th layer (V is the set of items and d is dimension). $W^l \in \mathbb{R}^{d\times d}$ is the transformation matrix for the l-th layer. $A \in \mathbb{R}^{|V|\times|V|}$ is the adjacency matrix of the graph and $\widetilde{D}_{ii} = \sum_j \widetilde{A}_{ij}$

- **GraphSage**- Goal of this model is to learn representation of (each) node based on it's neighbours. It consists of two steps: aggregation an update. Agregation can be expressed as:

$$a_v = f_{aggregate}(\{h_u|\} \in N(v)) \tag{1.4}$$

Where $f_{aggregate}$ is aggregator (i.e., mean, sum and max pooling). h_u is feature vector of node u and $N(v)$ are all neighbours of node v.

Update is next step which is based on node's neighbourhood aggregated representation and node v's previous representation can be expressed as:

$$h_v^k = f_{udpate}(a_v, h_v^{k-1}) \tag{1.5}$$

Where h_v^{k-1} is previous feature vector of node v and a_v is aggregation of node v.[17]

- **GAT**- The main feature distinguishing GAT from GCN is how information from neighboring nodes of a given node is aggregated. GAT utilizes the attention mechanism for computing updated node embedding. This mechanism learns different nodes weights based on their importance. [16]

  Equations detailing the computation of node embeddings h_i^{l+1} for layer $l+1$ based on the embeddings of layer l:

$$z_i^{(l)} = W^{(l)} h_i^{(l)} \tag{1.6}$$

  Where this equation is a linear transformation of the lower layer features $_i^{(l)}$ and $W^{(l)}$ is its learnable weight matrix.

$$e_{ij}^{(l)} = \text{LeakyReLU}(\vec{a}^{(l)^T}(\mathbf{z}_i^{(l)}||\mathbf{z}_j^{(l)})) \tag{1.7}$$

  In this equation is computed pair-wise attention score between two neighbors. $z_i^{(l)}||z_j^{(l)}$ represent concatenation of two nodes features. Then dot product of it and learnable weight vector $\vec{a}^{(l)^T}$ is taken and LeakyReLU is applied in the end. This whole is called additive attention.

$$\alpha_{ij}^{(l)} = \frac{\exp(e_{ij}^{(l)})}{\sum_{k \in \mathcal{N}(i)} \exp(e_{ik}^{(l)})} \tag{1.8}$$

Applies a softmax to normalize the attention scores on each node's inbound edges.

$$h_i^{(l+1)} = \sigma\left(\sum_{j\in\mathcal{N}(i)} \alpha_{ij}^{(l)} z_j^{(l)}\right) \quad (4) \tag{1.9}$$

This equation is similar to GCN. The embeddings from neighbors are aggregated together, and scaled by the attention scores.

- **GCN**: is a spectral model. It combines graph convolution and neural networks to obtain a representation of higher order representation. Node representations are updated by:

$$H^{l+1} = \delta\left(\widetilde{D}^{-\frac{1}{2}}\widetilde{A}\widetilde{D}^{-\frac{1}{2}}H^l W^l\right) \tag{1.10}$$

  where δ is and activation function, $H^l \in \mathbb{R}^{|V|\times d}$ is the representations in the l-th layer (V is the set of items and d is dimension). $W^l \in \mathbb{R}^{d\times d}$ is the transformation matrix for the l-th layer. $A \in \mathbb{R}^{|V|\times|V|}$ is the adjacency matrix of the graph and $\widetilde{D}_{ii} = \sum_j \widetilde{A}_{ij}$

- **HetGNN** is used for heterogeneous Graph Neural Networks. It first divide neighbours of central node by their type. Then agregation is used on them. [16]

1.6 Celestial coordinates

Celestial coordinates are coordinates that helps astronomers specify the location of objects in the sky.

These coordinate system are oriented to either the ecliptic (with origin in the center of either the Sun or Earth) or the celestial equator (projection of Earth's equator onto the celestial sphere). In [18] are mentioned two types of celestial coordinates. First - the former system (defined by degrees of celestial latitude and longitude) and later system (expressed in degrees of declination and hours for right ascension)[18]

1.6.1 Longitude and Latitude

This coordinate system is 'fixed to the Earth's surface'. We can use it to describe uniquely any location on earth. It uses two coordinates Longitude and Latitude. The demarcation of the longitude coordination is done with meridians. Longitude ranges from 0° to 180° east and 0° to 180° west. The zero point of longitude is defined as a point in Greenwich, England called the Prime Meridian. The demarcation of the latitude coordination is done by lines parallel to equatorial. [19]

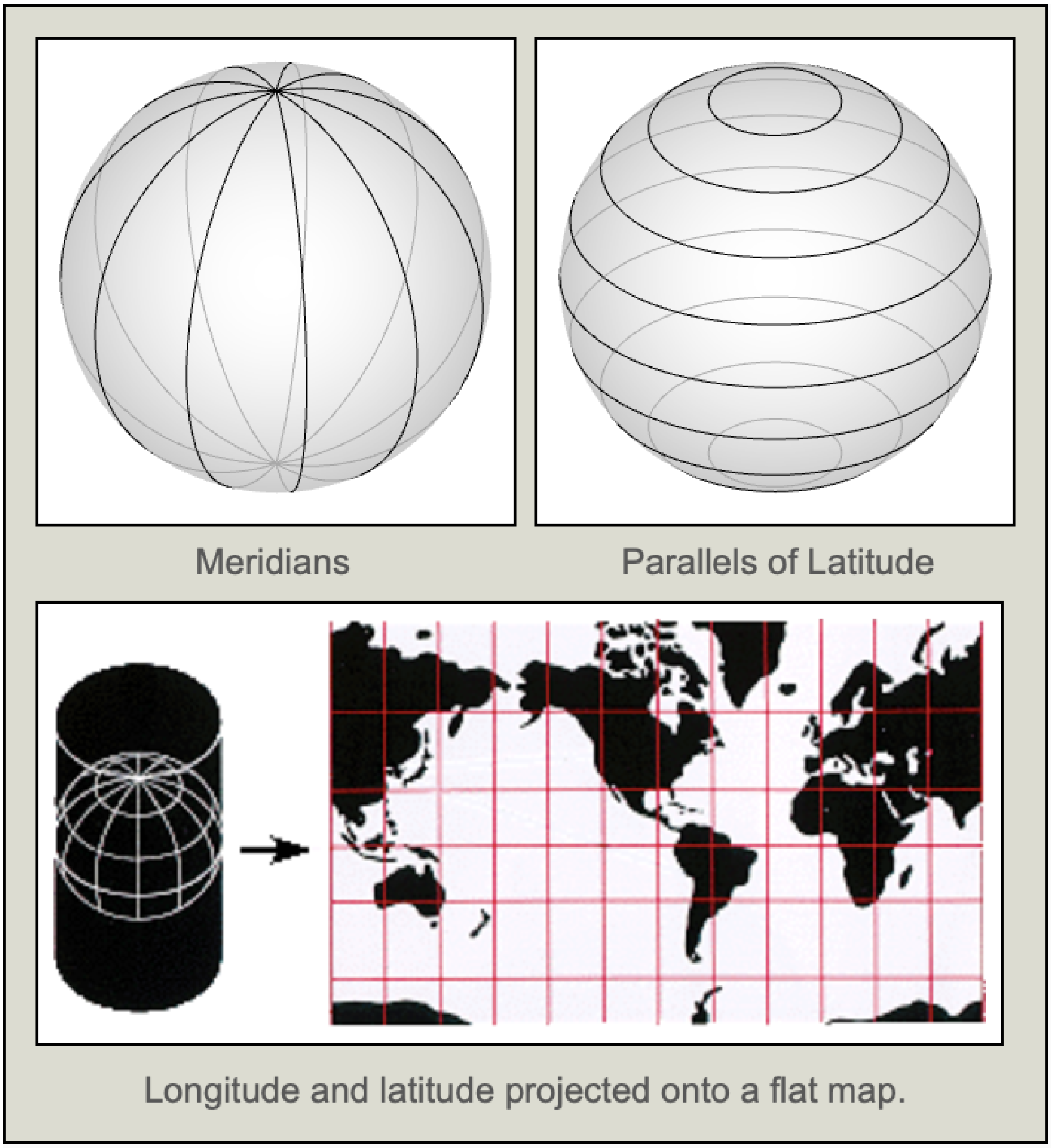


Figure 1.5: Longitude and latitude coordinate system. Taken from [20]

1.6.2 Equatorial Coordinate System

This method is used to locate objects on the celestial sphere. Just like longitude and latitude it describe unique location of object. The equatorial coordinate system is projection of the latitude and longitude coordinate system which is used on earth onto celestial sphere. So basically latitude become declination (indicate how far north or south of the celestial equator the object lies). Longitude then is projected to right ascension. Instead of degrees minutes and seconds as is used in measurement in longitude hours, minutes and seconds (east from where the celestial equator intersects the ecliptic) is used to measure right ascension. [21]

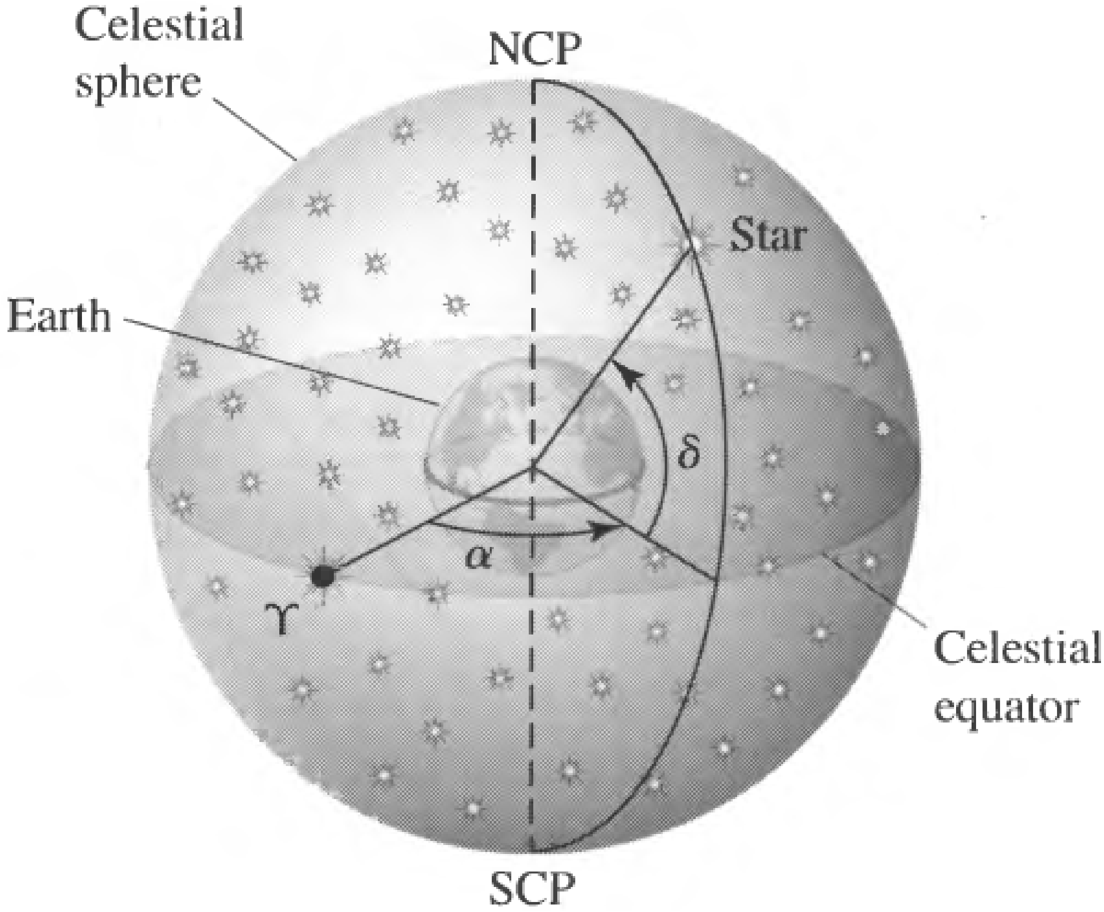


Figure 1.6: A representation of the Equatorial Coordinate System. The right ascension, declination and location of vernal equinox is represented α, δ, and γ. Taken from [22]

1.6.3 Horizon, or Altitude-Azimuth Coordinates

This coordinate system is 'not fixed to the Earth's surface', which means that location of object can be different from different locations. Altitude is the angular distance above the horizon and can range from 0° (horizon) to 90° (zenith). Azimuth is the angular distance measured in a clockwise direction from the north, parallel to the horizon. The azimuth changes from 0° for an object due North to 90° (due East) to 180° (due South) to 270° (due West) to 360° . This system is fixed to the Earth, so unlike Equatorial Coordinate System it coordinates changes also in time. [23]

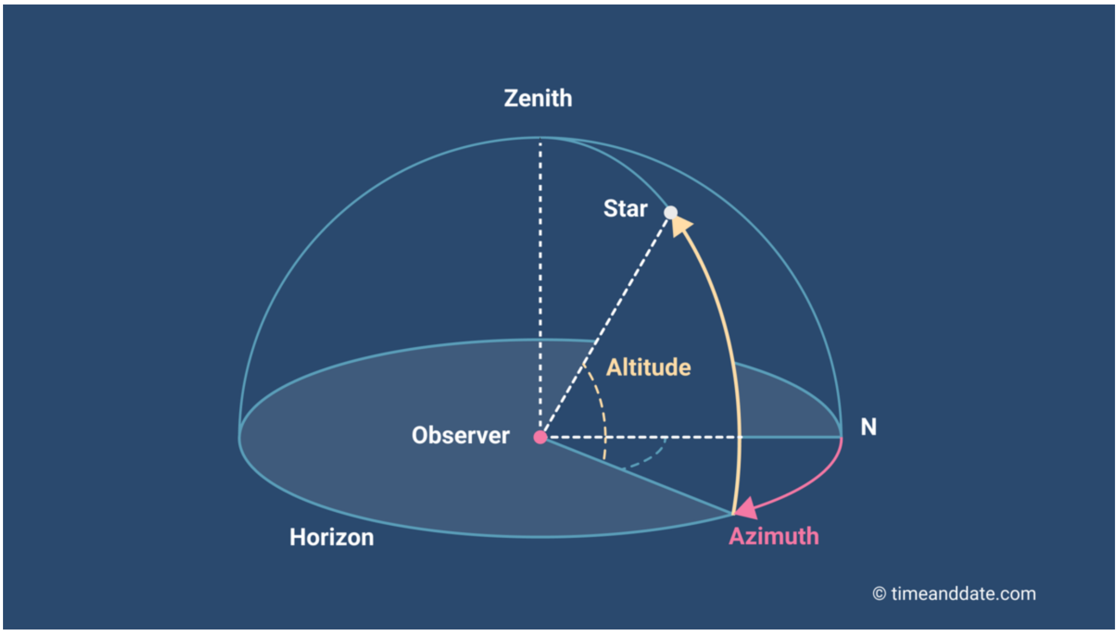


Figure 1.7: Horizon coordinate system. Taken from [24]

Chapter 2

Astronomical data

Chapter 3

Implementation

Chapter 4

Data generation

Chapter 5

Results

Conclusion

Bibliography

[1] H. Sylvestre and R. Parama, "Space debris: Reasons, types, impacts and management," *Indian Journal of Radio and Space Physics*, vol. 46, Mar. 2017.

[2] D. Kyselica, J. Šilha, R. Ďurikovič, D. Bartková, and J. Tóth, "Towards image processing of reentry event," *Journal of Applied Mathematics, Statistics and Informatics*, vol. 19, no. 1, pp. 47–60, 2023.

[3] T. J. Colvin *et al.*, "Cost and benefit analysis of orbital debris remediation," *ntrs.nasa.gov*, Mar. 2023.

[4] R. D. Launius and D. R. Jenkins, *Coming home: reentry and recovery from space.* Government Printing Office, 2012, pp. 8–9.

[5] C. Pardini and L. Anselmo, "The kinetic casualty risk of uncontrolled re-entries before and after the transition to small satellites and mega-constellations," *Journal of Space Safety Engineering*, vol. 9, no. 3, pp. 414–426, 2022, ISSN: 2468-8967.

[6] M. Byers, E. Wright, A. Boley, and C. Byers, "Unnecessary risks created by uncontrolled rocket reentries," *Nature Astronomy*, vol. 6, pp. 1–5, Jul. 2022.

[7] P. Marks. "Dodging debris." Accessed 17 Jan. 2024. (Jul. 2021), [Online]. Available: `https://aerospaceamerica.aiaa.org/features/dodging-debris/`.

[8] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superglue: Learning feature matching with graph neural networks," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 4937–4946.

[9] A. Marmol, T. Peynot, A. P. Eriksson, A. T. Jaiprakash, J. M. Roberts, and R. Crawford, "Evaluation of keypoint detectors and descriptors in arthroscopic images for feature-based matching applications," *IEEE Robotics and Automation Letters*, vol. 2, pp. 2135–2142, 2017.

[10] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *Computer Vision – ECCV 2006*, A. Leonardis, H. Bischof, and A. Pinz, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 404–417, ISBN: 978-3-540-33833-8.

[11] S. Li, "A review of feature detection and match algorithms for localization and mapping," *IOP Conference Series: Materials Science and Engineering*, vol. 231, no. 1, p. 012 003, Sep. 2017.

[12] B. Sanchez-Lengeling, E. Reif, A. Pearce, and A. B. Wiltschko, "A gentle introduction to graph neural networks," *Distill*, 2021, https://distill.pub/2021/gnn-intro.

[13] J. Zhou, G. Cui, S. Hu, *et al.*, *Graph neural networks: A review of methods and applications*, 2021. arXiv: `1812.08434 [cs.LG]`.

[14] L. Waikhom and R. Patgiri, *Graph neural networks: Methods, applications, and opportunities*, 2021. arXiv: `2108.10733 [cs.LG]`.

[15] D. Grinberg, *An introduction to graph theory*, `https://synthical.com/article/da00b019-c8b9-478d-a385-9c7292147b6d`, Jul. 2023. arXiv: `2308.04512 [History and Overview]`.

[16] X. Li, L. Sun, M. Ling, and Y. Peng, "A survey of graph neural network based recommendation in social networks," *Neurocomputing*, vol. 549, p. 126 441, 2023, ISSN: 0925-2312.

[17] W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive representation learning on large graphs," *CoRR*, vol. abs/1706.02216, 2017. arXiv: `1706.02216`.

[18] N. Kanas, "Star and solar system maps: A history of celestial cartography," *Research Notes of the AAS*, vol. 5, no. 4, p. 69, Apr. 2021.

[19] M. Kennedy, *Understanding Map Projections*. Redlands, Calif.: ESRI, 2000, ISBN: 9781589480032.

[20] NAAP. "Units of longitude and latitude - basic coordinates and seasons." Accessed on 10-12-2023. (2019), [Online]. Available: `https://astro.unl.edu/naap/motion1/tc_units.html`.

[21] V. A. Golovashchenko and S. V. Kolomiyets, "Meteor Observational Data Visualisation in the Equatorial Coordinate System Using Information Technology," *Odessa Astronomical Publications*, vol. 30, p. 166, Jan. 2017.

[22] A. Govind, D. Choudhury, B. Mathew, and K. Paul, "Astrometry: The foundation for observational astronomy," *Mapana - Journal of Sciences*, vol. 17, pp. 1–10, Jan. 2018.

[23] A. Roy and D. Clarke, *Astronomy: Principles and Practice*, 4th ed. CRC Press, 2003.

[24] K. Bikos. "The horizontal coordinate system." Accessed 17 Jan. 2024. (2019), [Online]. Available: `https://www.timeanddate.com/astronomy/horizontal-coordinate-system.html`.