AIR TRAFFIC DATA ANALYSIS

Bachelor Thesis

By

Rohit Kumar Naveen Kumar

A thesis submitted to

Indian Institute of Information Technology Kalyani

for the partial fulfillment of the degree of

Bachelor of Technology
in
Computer Science and Engineering

June, 2020

Certificate

This is to certify that the thesis entitled "**Air Traffic data analysis**" being submitted by **Mr. Rohit Kumar, Mr. Naveen Kumar**, an undergraduate student (Roll No : 39/cse/16031/186 and 39/cse/16019/176) in the Department of Computer Science and Engineering, Indian Institute of Information Technology Kalyani, India, for the award of **Bachelor of Technology** in **Computer Science and Engineering**, is an original research work carried by him under my supervision and guidance. The thesis has fulfilled all the requirements as par the regulation of **IIIT Kalyani** and in my opinion, has reached the standards needed for submission. The works, techniques and the results presented have not been submitted to any other university or Institute for the award of any other degree or diploma.

(**Dr. Sanjay Chatterji**)
Ph.D. [IIT Kharagpur]
Assistant Professor
Department of Computer Science and Engineering
Indian Institute of Information Technology Kalyani
Kalyani, W.B.-741235, India.

Acknowledgments

First of all, I would like to take this opportunity to thank my supervisor Dr. Sanjay Chatterji without whose effort this thesis would not have been possible. I am so grateful to him for working tirelessly after me, answering my doubts whenever and wherever possible. I am most grateful to Department of Computer Science and Engineering, IIIT Kalyani, India, for providing me this wonderful opportunity to complete my bachelor thesis.

And last but the biggest of all, I want to thank my parents, for always believing in me and letting me do what I wanted, but keeping a continuous check that I never wandered off the track from my goal.

Rohit kumar
Roll No.: 39/cse/16031/186
Department of Computer Science and Engineering
Indian Institute of Information Technology Kalyani
Kalyani, W.B.-741235, India.
Naveen kumar
Roll No.: 39/cse/16019/176
Department of Computer Science and Engineering
Indian Institute of Information Technology Kalyani
Kalyani, W.B.-741235, India.

Abstract

The air traffic data analysis is important to manage the routes of flight. It is important to American type economy. It is large and complex traffic system controlled by human decision makers. We studied and learnt about this system how to increase the use of air space effectively in this growing demand of air travelling in this fast and digital life where every person from middle class common people to upper class wants to save time. [**?**][**?**].We studied for better management of air traffic system by human and technology both for effective utilization. Our approach to develop better management system to handle the situation. We are developing better algorithm that enable decision support tool development. In order to do so we are using Python programming language, dynamic programming and machine learning. We will take input and trained that input data and plot that on the graph in order to understand the situation, we discuss our research on three types of decisions in the air traffic management system. The first is faced by supervisors of air traffic controllers: how to configure available airspace, controllers, and other resources to ensure safe and efficient operations in a region of airspace over a period of time. The second type of decision is faced by airlines: how to assign a set of flights to a set of slots in an Airspace Flow Program. The third type of decision is faced by air traffic flow managers: when to implement a Ground Delay Program. We use operational data to build two types of model of the implementation of Ground Delay Programs. The descriptive models we developed can be used to predict Ground Delay Program implementation, which may be of value in decision support tools for stakeholders such as airlines. They also provide insights into current practice that could motivate the development of tools to support the traffic flow managers who decide when to implement Ground Delay Programs.

Keywords: "Air Traffic data analysis"

Contents

List of Tables

Chapter 1

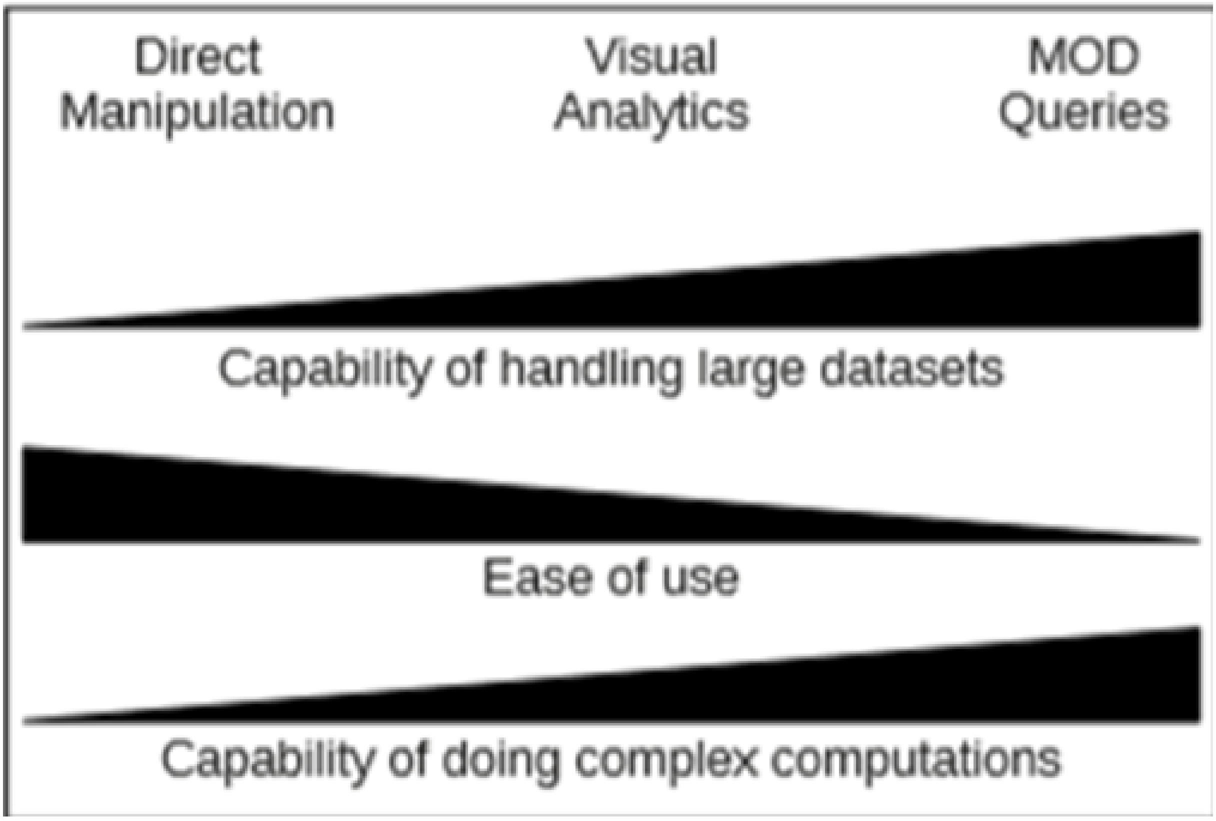
Introduction

Air Traffic data analysis is important to understand about the traffic in the air space. It is based on air traffic control that is known as ATC in short. The goal of Air Traffic Control (ATC) is to maximize both safety and capacity, so as to accept all flights without compromising the life of the passengers or creating delays in flight timing to avoid waste of time of passengers and also effective utilization of air space. It is because air traffic is expected to be double by 2030, new visualizations and analysis tools have to be developed to maintain and further improve the safety level. To do so, air traffic practitioners analyze data from the ATC activity. This multidimensional data includes aircraft trajectories (3D location plus time), flight routes, meteorological data, etc. In this chapter, we detail the relevant tasks of ATC practitioners, and demonstrate recent visualization and query methods to fulfill them. Three techniques are demonstrated: Direct manipulation, Visual analytics, and Moving object database queries.

1.1 Three techniques are demonstrated:

1. Direct manipulation visually represents the raw trajectories, and allows the user to efficiently explore them and highlight interesting Subsets using convenient views and simple mouse interaction. 2. Visual analytics provide a rich tool box of data transformations and visualizations that help a human analyst exploring complex movement events in the data. 3. Moving object database (MOD) defines query operators accessible to the user through textual query languages. They are able to perform complex computations over large data sets efficiently. According to the analysis task, the experience of the human doing

- ***diagram*** : the analysis, and the data set size, any of these three analysis methods can be chosen.



Direct manipulation is good for having a first look at the data. It is intuitive to use. Visual analytics provides more sophisticated transformations and aggregations,andthusitisabletoprocesslargerdatasets,and
analysis.

1.2 Introduction

The AES-128 operates for 10 rounds and encrypts a 128-bit plaintext using a 128-bit key. There is an initial AddRoundKey operation which involves a bitwise xor for each bit of the state with corresponding Round Key's bit. Then for 9 rounds we have the SubBytes, ShiftRows, MixColumn and the AddRoundKey operations. In the last round, there is SubBytes, ShiftRows and AddRoundKey operations only. The details of the above operations are given in below subsections.

Chapter 2

Motivation

Transportation infrastructure of a country is one of the most important indicators of its economic growth. Here we study the Airport Network of India (ANI) which represents India's domestic civil aviation infrastructure as a complex network. We find that ANI, a network of domestic airports connected by air links, is a small-world network characterized by a truncated power-law degree distribution and has a signature of hierarchy. We investigate ANI as a weighted network to explore its various properties and compare them with their topological counterparts. The traffic in ANI, as in the World-wide Airport Network (WAN), is found to be accumulated on interconnected groups of airports and is concentrated between large airports. In contrast to WAN, ANI is found to be having disassortative mixing which is offset by the traffic dynamics. The analysis indicates possible mechanism of formation of a national transportation network, which is different from that on a global scale.

2.1 Introduction

Studies of human factors in aviation have progressed significantly since the Second World War. However, the majority of correspondence has been focussed primarily on Pilots, and the perspective from the flight deck. The development of human factors within Air Traffic Control (ATC) has progressed more slowly than similar studies involving Pilot performance. Isaac and Ruitenberg (1999, p.4) suggest that a possible cause of this is the perceived low profile of ATC personnel who, unlike their uniform-wearing comrades, are less visible to the flying public.

This data is essential for air traffic controllers, inordertomaintainasafedistancebetweenaircraftsandtooptimizetraffic inorder to decrease time gaps between flights, also noise pollution and fuel.Our goal in this chapter is not to provide tools for real-time usages, but rather to detail off-line tools that analyze recorded trajectories in more depth. Without this real-time constraint, ATC practitioners can investigate, in more detail, recorded trajectories and therefore extract relevant information and perform three main tasks: improve safety, optimize traffic, and monitor environmental considerations. Improving safety can be detailed as:

2.1.1 1.analyze and understand past conflicts (when two aircraft fail to meet minimum safety distance) and then improve safety with feedback from past experience,

2.1.2 2 analyzetheaccuracyofdataprovidedbygroundradarwithprobetrajectory comparison (i.e., with GPS tracking and radar test plots), and

2.1.3 3.filter and extract trajectories in order to reuse them for Air Traffic Controllers' training simulations.

Traffic optimization can be detailed as:

1 devisenewairspaceorganizationandflightroutestohandletrafficincrease, 2 study profitability (i.e., number of aircraft on a specific flight route per day, number of aircraft that actually land at a specific airport, etc), 3 calculate the metrics from the traffic: traffic density, spacing quality (mean distance between aircraft), number of holding loops, number of rectilinear trajectories(trajectoriesthatareclos arrival), etc, and 4 measure the activity of each airport: number of take-offs and landings per hour etc.

Finally environmental considerations can be detailed as:

1 Compare trajectories with environmental considerations (fuel consumption, noise pollution, vertical profile comparison),

2.detectmissedapproachtrajectories(whichproducenoise),laptraininglandings(pilotswhotraintotakeof landings consume a lot of fuel)

3 count continuous descending aircraft (since these aircraft maintain a constant descent rate, they reduce their fuel consumption). This list is not exhaustive but it gives the main tasks that ATC practitioners perform. These tasks highlight the need for powerful tools to analyze aircraft trajectories.

Chapter 3

DataSetDescription

In this section, we have taken a data set that is containing the data about the number of flights that have travelled from one destination to other destination with time gap of monthly ,Quaterlly or yearly.And total number of passengers that have travelled in that time period. We have taken names of few flights of national and international airlines and rate of change of passenger traffice montly or quaterly and also top ten airlines having more traffic. We have meantion city1 and city2 as names of airport in foreign and in india.[**?**]. Names of top 100 airports pair with highest inflow and outflow traffic have been taken.

In [**?**], San Francisco International Airport Report on Monthly Passenger Traffic Statistics by Airline. Airport data is seasonal in nature, therefore any comparative analyses should be done on a period-over-period basis (i.e. January 2010 vs. January 2009) as opposed to period-to-period (i.e. January 2010 vs. February 2010). It is also important to note that fact and attribute field relationships are not always 1-to-1. For example, Passenger Counts belonging to United Airlines will appear in multiple attribute fields and are additive, which provides flexibility for the user to derive categorical Passenger Counts as desired. [**?**] :

3.1 This Python 3 environment comes with many helpful analytics libraries installed

```python
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D

# Input data files are available in the "../input/" directory.
# For example, running this (by clicking run or pressing Shift+Enter) will list the files in the input directory

from subprocess import check_output
print(check_output(["ls", "../input"]).decode("utf8"))
```

```
airtrafficcoordinatesindia
international-air-traffic-from-and-to-india
```

Chapter 4

Direct Manipulation of Trajectories

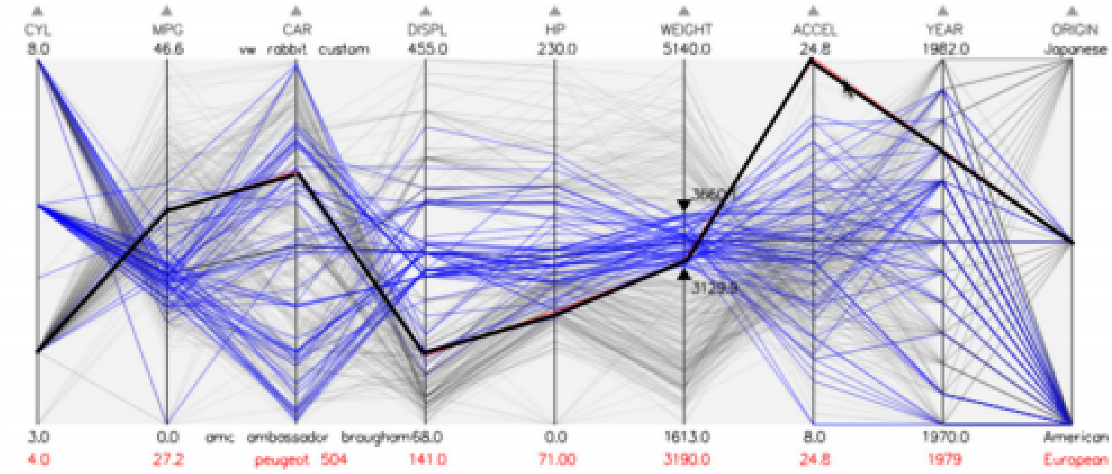
users often explore the queries as much as they explore the data: in the course of exploration, users discover that the set of features they thought relevant has to be adapted, either because they were false, or because they cannot findhow toquery them efficiently. Furthermore, trajectories are numerous and tangled: one-day's traffic over France for example, represents some 20000[?] trajectories.When deali ngwithtra jectories,users mustper for mdynami crequests (response time < 100 ms) on a large multi-dimensional data set (>1 million data) which contains many errors and uncertainties. The problem we address in this section is to find a way to express these queries, simply and accurately, given the constraints of size and uncertainty of the data sets. As a solution, the visualization and direct manipulation of trajectories proposes efficient interactions features. Direct manipulation was introduced by Ben Shneiderman in 1983 (Shneiderman (1983)) within the context of office applications

4.1 Detail an implementation

and thevirtual desktopmetaphor.This term has beenextended to human-computer interaction paradigms. The intention is to allow users to directly manipulate objects presented to them, using actions that correspond to the physical world (e.g., grasp, move objects, etc).Inthe following sections,wefir stdescribe direct manipulation requirements for trajectory exploration, then we detail an implementation instance, and finally we give one scenario of usage.

4.2 Parallel coordinates

Parallel coordinates are a two-dimensional technique to visualize multidimensional datasets Parallel coordinates are one of the most popular visualization techniques where attributes are represented as axes and data items are represented as lines linking the axes. They have been thoroughly studied and Siirtola and Räihä provide a good review of interaction techniques for parallel coordinates Most known techniques are brushing and rearrangement by direct manipulation.



Kandogan proposes Star coordinates [Kandogan, 2001], inspired by both parallel coordinates and the reorderable matrix, for visualizing multidimensional datasets. Conversely to parallel coordinates, the axes are not parallel but laid out radially, and data points are positioned in the 2D space according to their value for each dimension. The technique proposes several interaction techniques by direct manipulation of the axes: brushing an axis to highlight data points whose values according to the associated dimension fall into a range, similarly as for parallel coordinates; rotating an axis to relayout the data points; and scaling an axis to change its influence on the result. All these operations are performed directly on the axes, not the data points.

Chapter 5

Model of Visualization

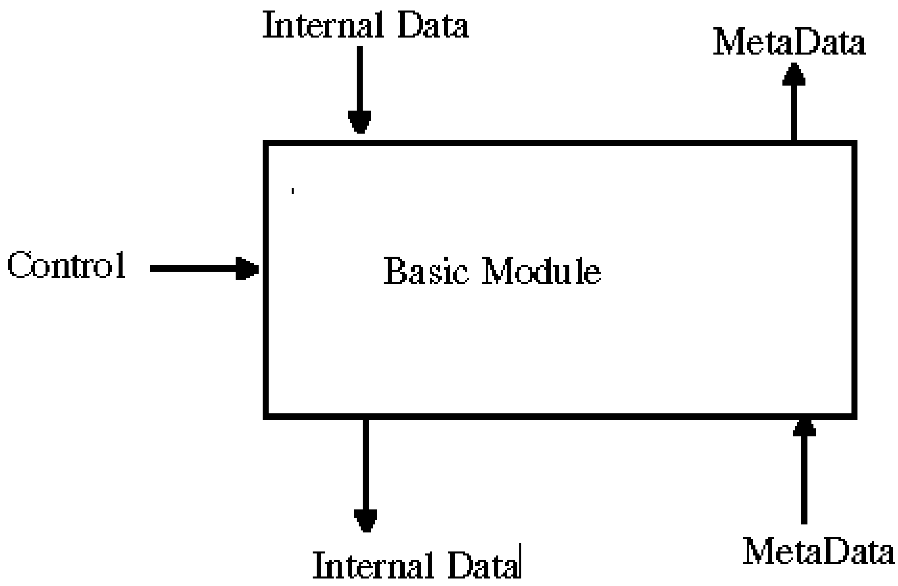
The data the module works on and transforms A visualization system is not just a system to create an image of the data but can be used to manipulate the data to create different types of images. A model of a visualization system should link the system with the model of scientific investigation discussed above. Visualization can help form the link between hypothesis and experiment and between insight and revised hypothesis.

The developed model consists of a set of abstract modules. It is a controlled dual dataflow system. A generic basic module is shown below:

5.1 The Control is perfomed by the User. The internal data is the data the module works on and transforms.

The developed model consists of a set of abstract modules. It is a controlled dual dataflow system. A generic basic module is shown below: Data Types The different data types describe the dataflow between the different modules. The data types are as follows:

Control Data: data that activates and controls all the modules in the system. User Input/Output: all forms of human - computer interaction. Input might be keyboard, mouse, light pen, etc. Output might be screen, hard copy device, sound (for sonification), etc. This is converted into the Metadata for the system modules. Internal Data: the data that is allowed to flow through the system. External Data: data that can be imported/exported to the system. This might be ascii data (from observation or simulation) or images. Storable Data: data that can be stored and retrieved within the system. Graphics Data: a reduced form of internal data that represents graphics primitives (2D or 3D) Picture Data: a reduced form of graphics data (pixel map or 2D primitives for display or hardcopy)

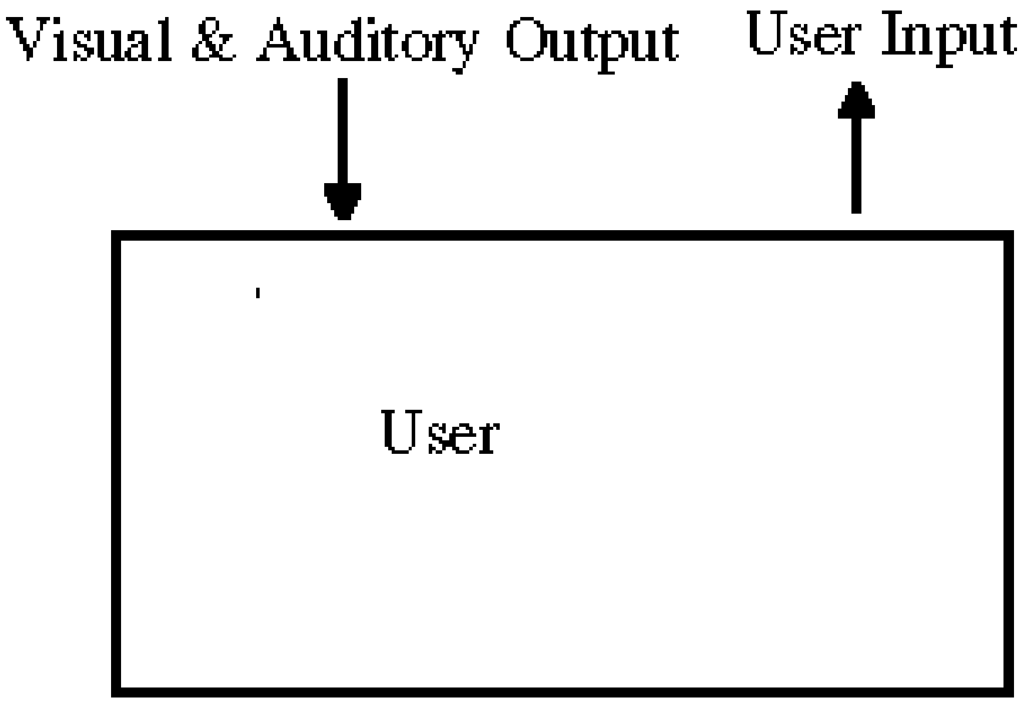


5.2 Module Types.

There are several different module types. Each different type can have multiple sub-types. A visualization system should allow for the user to customize existing modules and to construct new modules.

User Module: The user observes the system output and controls the system.

beginfigure[h]



figure

5.3 User Interface Module: this maps between the user's world and the internals of the visualization system.

To deal with the problems of multiple disciplines in the computational sciences effectively, it is useful to begin by developing a coherent picture of the various steps a scientist takes while simulating a natural process using a computational model.“ ” This way we can capitalize on the similarities between the requirements of each tar- get discipline. The process of numerical simulation (see Figure 1) involves the transformation of basic physical equations (for example, the Navier-Stokes, Schroedenger, or Maxwell equations) into a computer program. These approximations must then be augmented with a speci- fication of the domain to be simulated (that is, a computa- tional grid, initial conditions, boundary conditions, etc.). Together, these constitute a complete description of the problem whose solution can now be computed, typically by numerical simulation.

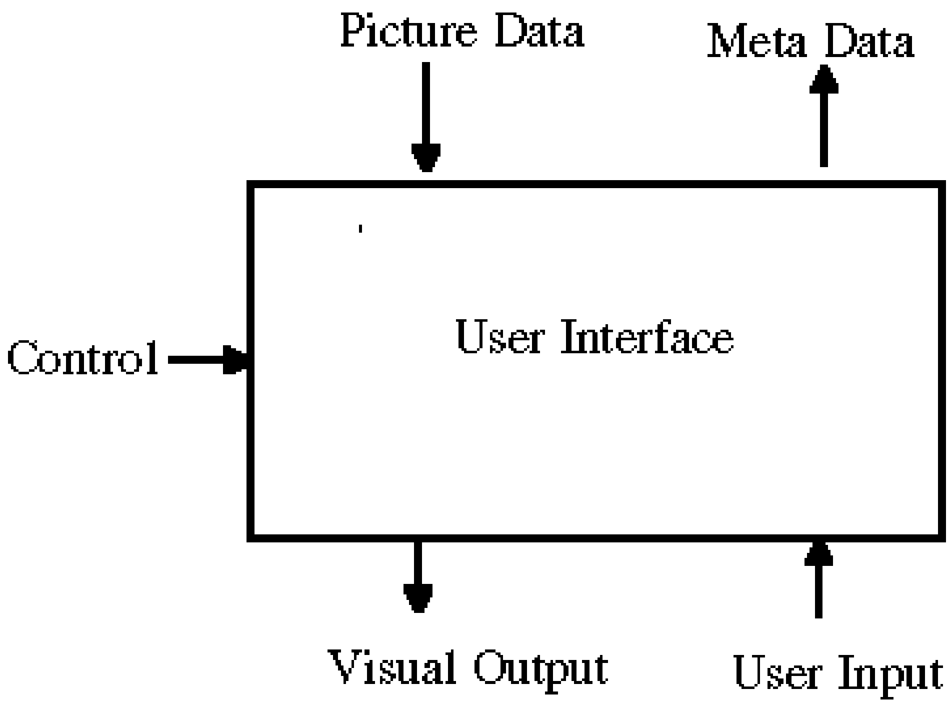
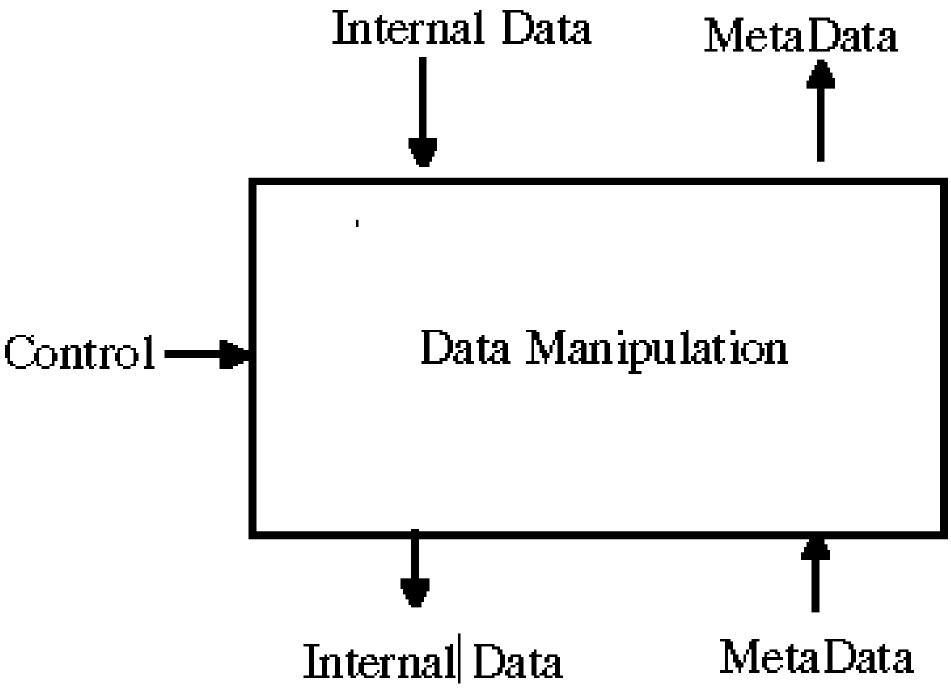


figure figure

5.4 A Visualization Technique module.

The analysis step in the cycle is where computer visualization plays a large role. This step can be broken down into its constituent parts (see Figure 2) to reveal several operations common to all simulations (and a great number of experimental processes). The analysis process is itself a cycle which is executed repeatedly until all questions are resolved. The processes a researcher or engineer executes are the following:



figure

5.5 Data Access modules allow data to be stored and/or retrieved, perhaps in a format specific to the visualization system.

Programmers developing new modules that J)roducc geometry can do so in Fortran, C. or C + + b!. ivriting just a fep̌rocedures. Some of these proc;edures describe to AC'S which parameters will control the geometry that is Jlroduced, lvhat sort of input data the module expects, and how to regenerate the geometry the data on which it depends or one of the module parameters changes. An additional procedure makes PIIIGS+ graphics calls to describe the geometry to AS. One major advantage of this architecture is that only the graphics primitive calls need to be made to describe the geometry-all of the PHIGS+ database and attribute calls are made by AVS directly, controlled by the use1 interface as described above. This dramatically reduces the programming burden on the user.

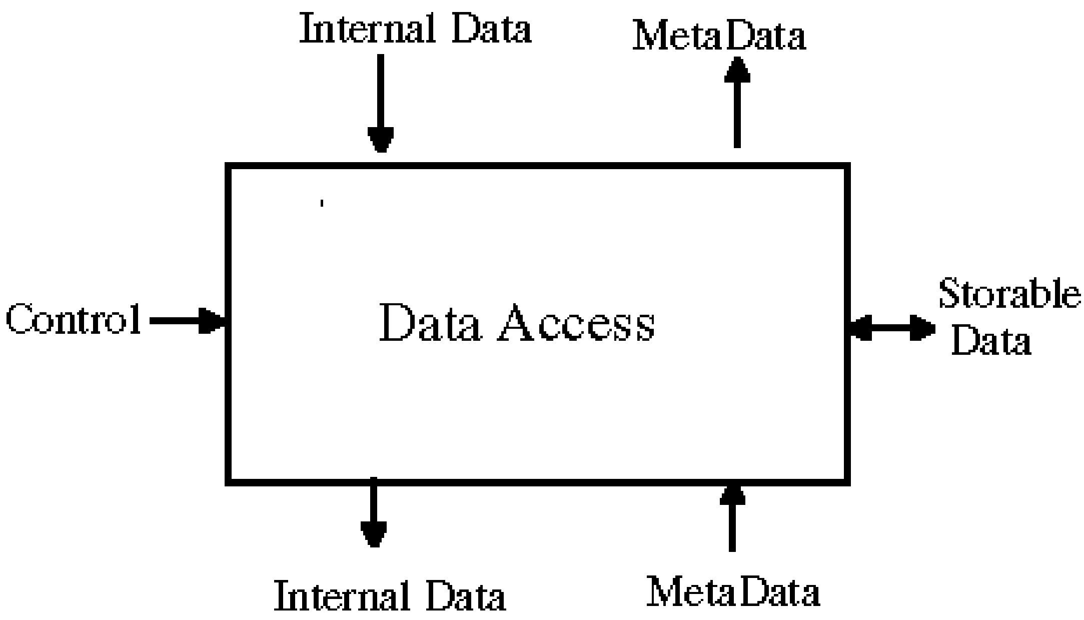


figure The graphics system consists of a user interface and a programmer interface. The user interface presented to a nonprogramming user is defined by the control panels for the render and display image modules supplied with AVS. The render module converts a geometric database into an image, which in turn can be displayed on the screen by the display-image module. III an AVS flow net- work, a render module can have many modules that pro- duce geometric data connected to it. Each image produced as output is a representation from one point of view of the scene defined by the modules connected to the render module and can be displayed on the screen by a display-image module.

Chapter 6

Design Requirements for Trajectory Exploration

Based on trajectory data set characteristics, we extracted the following design requirements to achieve the visual exploration of trajectories:

6.1 .View configuration:

1.View configuration: the system must permit the customization of views so as to offer multiple means of understanding and visually querying the data. It should allow for a change of mapping between data and visual dimensions. The system should also provide smooth transitions between visual configurations. Hence, the user will be able to visually track patterns between different view configurations.

6.2 .View rganization:

2. Views organization and navigation: the system must also permit the display ofmultipleviews.Theuse rmustbeable tovisually comparedifferentvisual configurations of the data set. This can be done with a matrix scatterplot or juxtaposed views.

6.3 .View filtering:

3. View filtering: the system must allow the user to filter out trajectories and then reduce cluttering.

6.4 .Boolean operations:

4. Trajectory selections and Boolean operations: The system must enable the user to select trajectories and combine them in order to perform complex queries. Some systems allow multiple selections sometimes called “layers”. Users can combine layers with Boolean operation by applying an “and” operation when they try to group differently selected trajectories.

6.5 .Graphical depiction of busiest airport pairs:

1.Dubai - Mumbai is arguably the busiest route with a whopping 4.5 million passengers flying both ways in 3 years

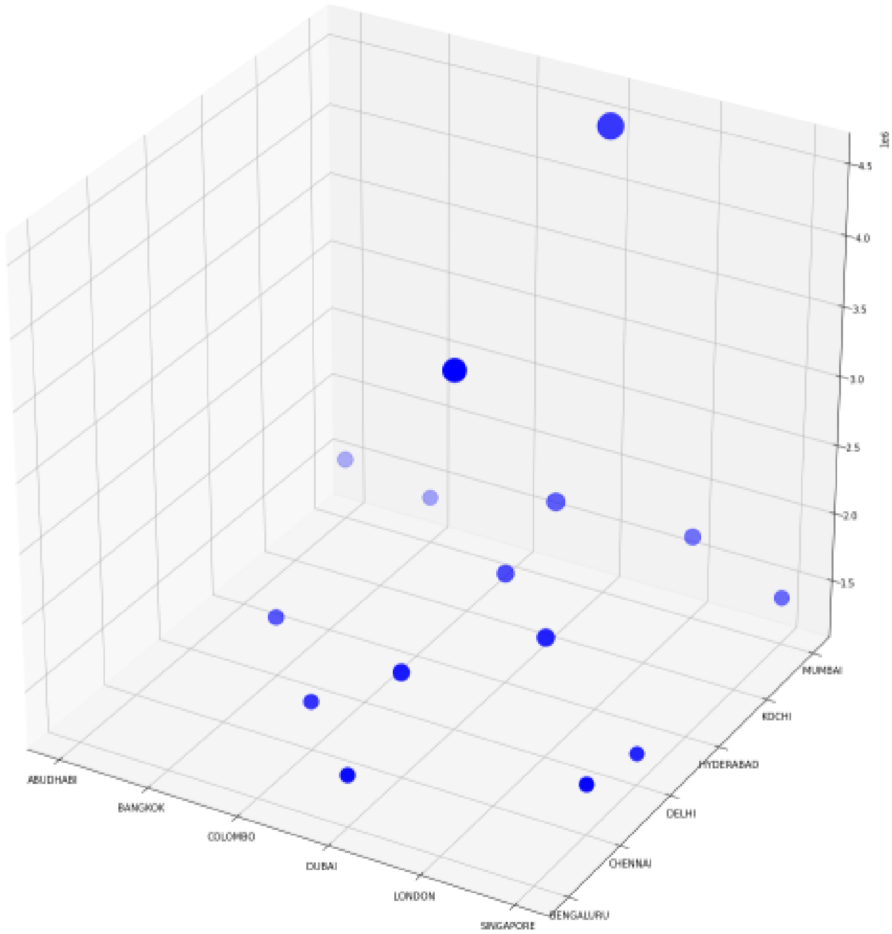
2.Take a look a the table below the graph. Flights fly in and out of Dubai to 5 Indian cities of the top 10 airport pairs.

3. considerable number of flights fly in an out of London and (Delhi, Mumbai)

```
df_citywise_cumulative_traffic = df_citywise.groupby(['CITY1', 'CITY2'])['TOTAL TRAFFIC'].sum().nlargest(15).reset_index()
highest_traffic = np.max(df_citywise['TOTAL TRAFFIC'])
uniqueCity1, city1Ints = np.unique(df_citywise_cumulative_traffic['CITY1'], return_inverse=True)
uniqueCity2, city2Ints = np.unique(df_citywise_cumulative_traffic['CITY2'], return_inverse=True)


df_citywise_cumulative_traffic['TOTAL TRAFFIC SCALED'] = (df_citywise_cumulative_traffic['TOTAL TRAFFIC'] / highest_traffic) * 100


fig = plt.figure(figsize=(20, 20))
ax = fig.add_subplot(1, 1, 1, projection='3d')
ax.scatter(city1Ints, city2Ints, df_citywise_cumulative_traffic['TOTAL TRAFFIC'], s=df_citywise_cumulative_traffic['TOTAL TRAFFIC SCALED
ax.set(xticks=range(len(uniqueCity1)), xticklabels=uniqueCity1,
       yticks=range(len(uniqueCity2)), yticklabels=uniqueCity2)
plt.show()

df_citywise_cumulative_traffic.head(10)
```

3D

It is obvious, that the amount and use of three-dimensional data has rapidly increased in the last few year. Boss and Streilein (2014) observed four major technology and business drivers for 3D: 1. There are massive new sensor hardware capabilities, such as automated data capture and model creation on the sensor side, LIDAR with masses of point clouds and automated photogrammetric workflows and processes. 2. 3D visualisation has now come into mainstream, but 3D analysis not. But there is as yet no mass market with consumer-focused systems. 3. Managing 3D data in enterprise workflows with improved performance and scalability of existing workflows and bridging the gap between point cloud surveys, GIS, CAD, BIM. Traditional file handling moves to database management. 4. There is a necessity for 3D data, where 2D data is not sufficient to describe our world and the consumer expectation demands three dimensions, as we all live and act in a three dimensional environment.



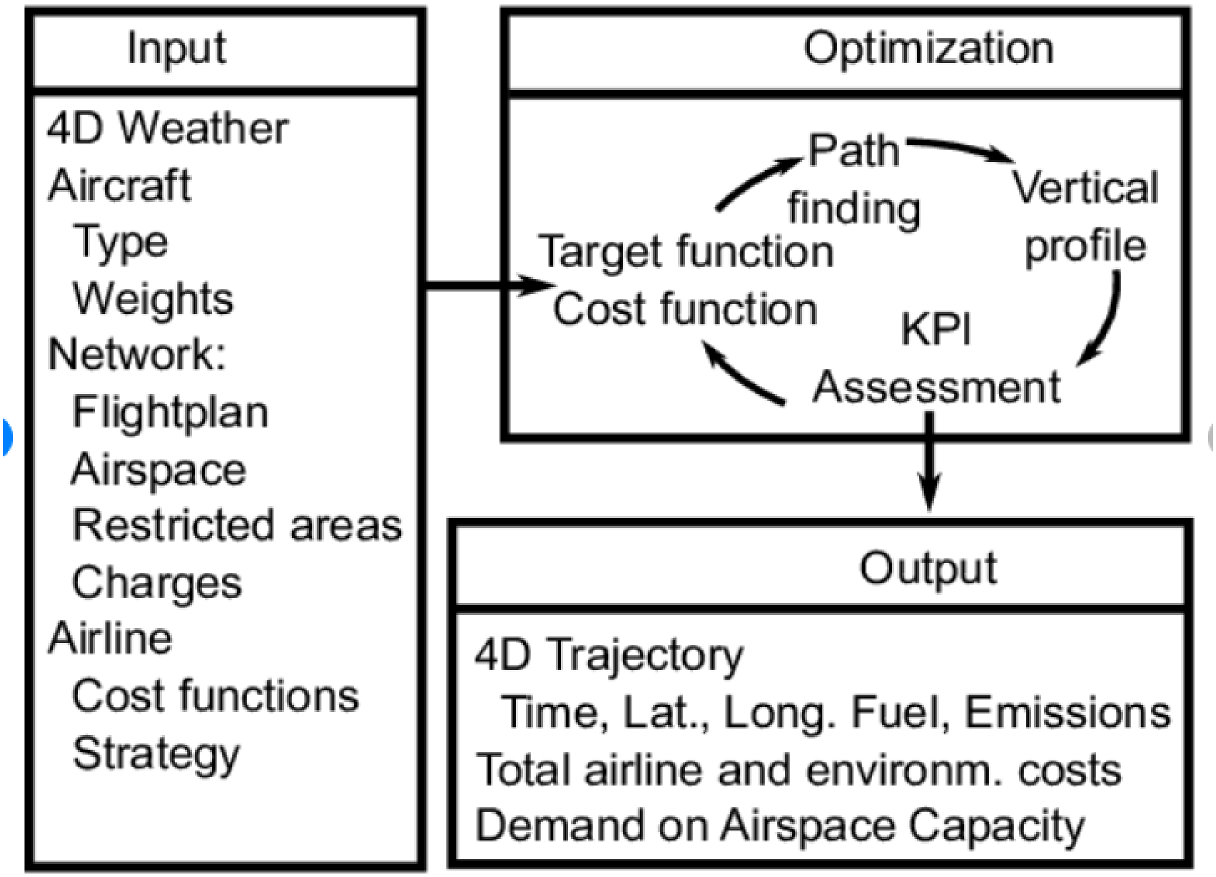
Chapter 6

6.6 Trajectory Manipulation:

We have implemented a simple and efficient direct manipulation technique: trajectory brush, pick and drop. The user selects a subset of the data set by means of a brushing technique. Brushing is an interaction that allows the user to "brush" graphical entities, using a size-configurable or shape-configurable area controlled by the mouse pointer. Each trajectory touched by this area is selected, and becomes gray. The selection can be modified by further brush strokes,or by removing part sofitwith brushstrokes inthe"erase"mode.The display shows a brush trail, so that the user can see and remember more easily how the selection was made. The combination of fast switching between the add/erase mode, trajectory visualization, rapid size-setting, and cursor centered zooming allows for fast and incremental selection.

6.7 4D Trajectory Management:

Airlines are focusing on efficiency by minimizing the number of planes and employees and the fuel burn,whereas the revenue passenger miles, number of departures and passengers and the available ton-milesshall be maximized.17 On a planning level, this strategy will have a reducing effect on the environmentalimpact as well. On the level of a single tra jectory, competitive applications are necessary. Taking safetyrequirements as additional constraints, a multiple-criteria optimization approach becomes unavoidable. Dueto the increasing weight of the environmental compatibility of the flight, the aircraft emissions have to becalculated precisely. For emissions as products of incomplete combustion (NOx, HC, BC), a combustionchamber model is necessary, which requires precise information about thrust, speed, acceleration forces andthe aircraft's attitude.3, 5, 18 The TOolchain for Multi-criteria Aircraft Trajectory Optimization TOMATOhas been exclusively developed for this purpose. This simulation environment includes a multi-criteriatrajectory optimization considering an exact trajectory calculation based on analytically solvable targetfunctions and a combustion chamber model, which is unique at the current state of the art. In summary,airlines' targets are found specifically on the network and tra jectory level and can be applied with simulationenvironments such as TOMATO for different airline business models (e.g. different network structures andcost indices), resulting in diverse target functions in the tra jectory optimization. With the simulation and comparison of the reference scenario with both models, two important stepshave been completed. On the one hand, the simulation environment TOMATO is validated and numericalvariables such as spatial and temporal resolution are assessed (compare Section III). On the other hand,the reference scenario enables a comparison of those conventionally filed flight paths with 4D-optimized freeroute trajectories, which have been calculated with TOMATO in a second step. Especially the impact ofthose optimized trajectories on the ATFM is compared with the impact of the conventionally filed trajec-tories. Figure 4 indicates a more complex airspace in the reference scenario due to a more heterogeneously



Out[11]:

	CITY1	CITY2	TOTAL TRAFFIC	TOTAL TRAFFIC SCALED
0	DUBAI	MUMBAI	4480329	739.172089
1	DUBAI	DELHI	3659375	603.729740
2	DUBAI	KOCHI	2085457	344.062145
3	LONDON	DELHI	1953307	322.259820
4	DUBAI	HYDERABAD	1886814	311.289695
5	DUBAI	CHENNAI	1846427	304.626581
6	LONDON	MUMBAI	1686086	278.173257
7	ABUDHABI	MUMBAI	1524384	251.495394
8	BANGKOK	DELHI	1511137	249.309882
9	DUBAI	BENGALURU	1455623	240.151090

Flight traffic flowing in and out of India

[25]:

```
df_citywise_top100 = df_citywise[df_citywise['TOTAL TRAFFIC'].isin(df_citywise['TOTAL TRAFFIC'].nlargest(100))]
df_citywise_top100.head(20)
```

Out[25]:

	YEAR	QUARTER	CITY1	CITY2	PASSENGERS FROM CITY1 TO CITY2	PASSENGERS FROM CITY2 TO CITY1	TOTAL TRAFFIC
42	2015	Q1	BANGKOK	MUMBAI	90096	94099	184195
93	2015	Q1	DUBAI	CHENNAI	102190	118103	220293
94	2015	Q1	DUBAI	DELHI	218394	218530	436924
96	2015	Q1	DUBAI	HYDERABAD	99989	112952	212941
98	2015	Q1	DUBAI	KOCHI	99921	114833	214754
104	2015	Q1	DUBAI	MUMBAI	280036	300108	580144
141	2015	Q1	KATHMANDU	DELHI	91764	86994	178758
172	2015	Q1	LONDON	DELHI	122349	122506	244855
174	2015	Q1	LONDON	MUMBAI	83375	94554	177929
266	2015	Q1	SINGAPORE	CHENNAI	88291	100794	189085
268	2015	Q1	SINGAPORE	DELHI	86479	97730	184209
272	2015	Q1	SINGAPORE	MUMBAI	79324	94249	173573
324	2015	Q2	BANGKOK	DELHI	85201	89927	175128
328	2015	Q2	BANGKOK	MUMBAI	92270	90960	183230
341	2015	Q2	COLOMBO	CHENNAI	88976	94312	183288
378	2015	Q2	DUBAI	BENGALURU	91594	88399	179993
379	2015	Q2	DUBAI	CHENNAI	117708	119990	237698
380	2015	Q2	DUBAI	DELHI	203483	219902	423385
382	2015	Q2	DUBAI	HYDERABAD	110757	117835	228592
384	2015	Q2	DUBAI	KOCHI	129829	121029	250858

Passenger Traffic monthwise and quarter wise

1 .Q1 is undoubtedly the best quarter for air traffic in India 2 .There is not much difference when we look at other quarter numbers.

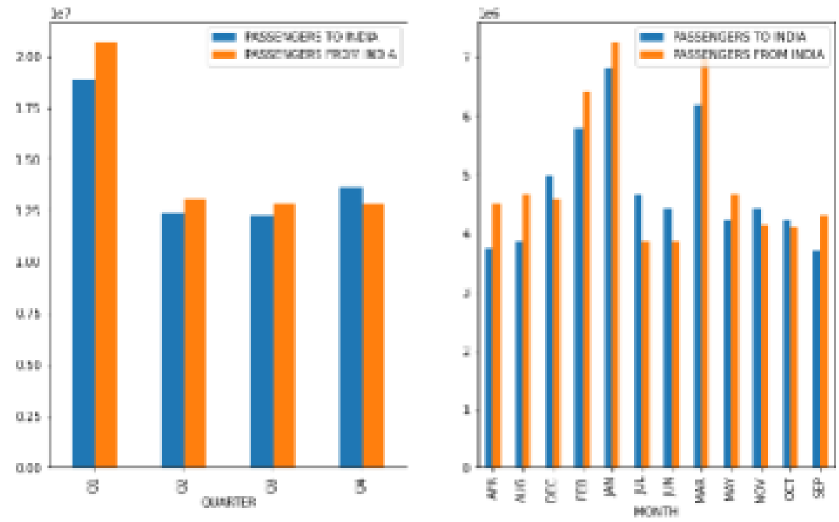
[16]:

```
fig, axes = plt.subplots(nrows=1, ncols=2)

df_qtrly = df_traffic_passengers.groupby(['QUARTER'])['PASSENGERS TO INDIA', 'PASSENGERS FROM INDIA'].sum().reset_index()
df_qtrly.plot(x='QUARTER', y=['PASSENGERS TO INDIA', 'PASSENGERS FROM INDIA'], kind='bar', ax=axes[0], figsize=(12, 7))

df_monthly = df_traffic_passengers.groupby(['MONTH'])['PASSENGERS TO INDIA', 'PASSENGERS FROM INDIA'].sum().reset_index()
df_monthly.plot(x='MONTH', y=['PASSENGERS TO INDIA', 'PASSENGERS FROM INDIA'], kind='bar', ax=axes[1], figsize=(12, 7))
```

```
/opt/conda/lib/python3.7/site-packages/ipykernel_launcher.py:3: FutureWarning: Indexing with multiple keys (implicitly conve
  This is separate from the ipykernel package so we can avoid doing imports until
/opt/conda/lib/python3.7/site-packages/ipykernel_launcher.py:6: FutureWarning: Indexing with multiple keys (implicitly conve
```

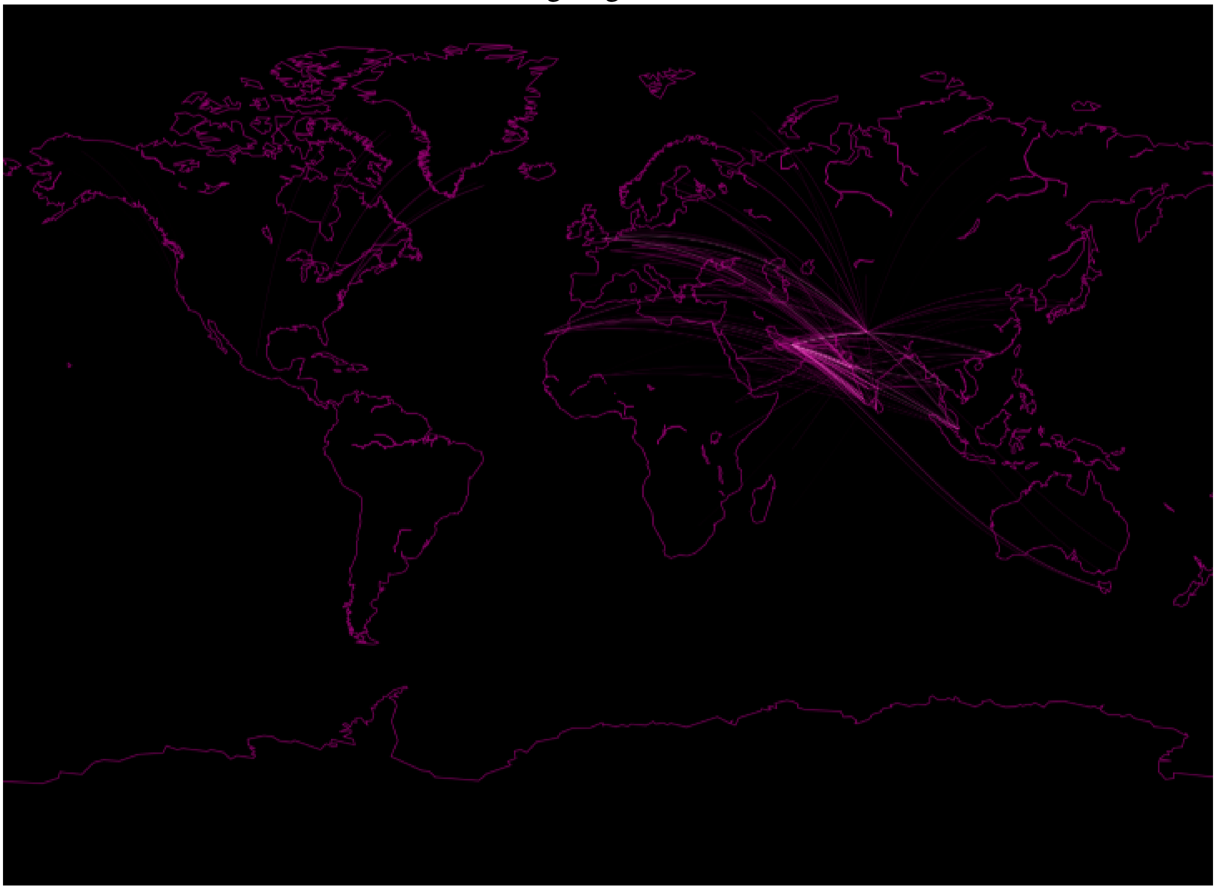
Out[16

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f5580d82910>
```



6.8 Flight map of all traffic flowing in and out of India:

Find out all unique cities from the citywise csv and geocode them to find their latitude and longitude. I used GeoPy for achieving this. Once I have lat, long information of arrival and departure airports, I used the Flight Visualization with Pandas and Matplotlib by Hugo Larcher. Combining the two, the visualization of flight data of our dataset can be viewed. beginfigure[h]



endfigure

6.9 References

6.10 Abul, O., Bonchi, F., and Giannotti, F. 2010. Hiding sequential and spatiotemporal patterns. IEEETransactionsonKnowledgeandDataEngineering,22(12),1709–1723.

6.11 Barabasi, A. L., and Albert, R. 1999. Emergence of Scaling in Random Networks. Science,286, 509.

6.12 Agrawal, R., and Srikant, R. 2002. Mining sequential patterns. Pages 3–14 of: Proceedings of the 11th International Conference on Data Engineering. IEEE.

6.13 Ahas, R., Aasa, A., Roose, A., Mark, U., and Silm, S. 2008. Evaluatingpassive mobile positioning data for tourism surveys: An Estonian case study. Tourism Management,29(3), 469–486.

6.14 Alt, W., and G., Hoffman. 1990. Biological Motion. Springer-Verlag.

6.15 Andrienko, N., and Andrienko, G. 2012. Visual analytics of movement: An overview of methods, tools and procedures. Information Visualization.

6.16 From Wikipedia,