Lab 3 - Parallel Coordinates

Problem Description -

Implement parallel coordinates using processing. Design and implement interaction mechanisms to support data exploration and evaluate the effectiveness of parallel coordinates on a variety of multidimensional datasets.

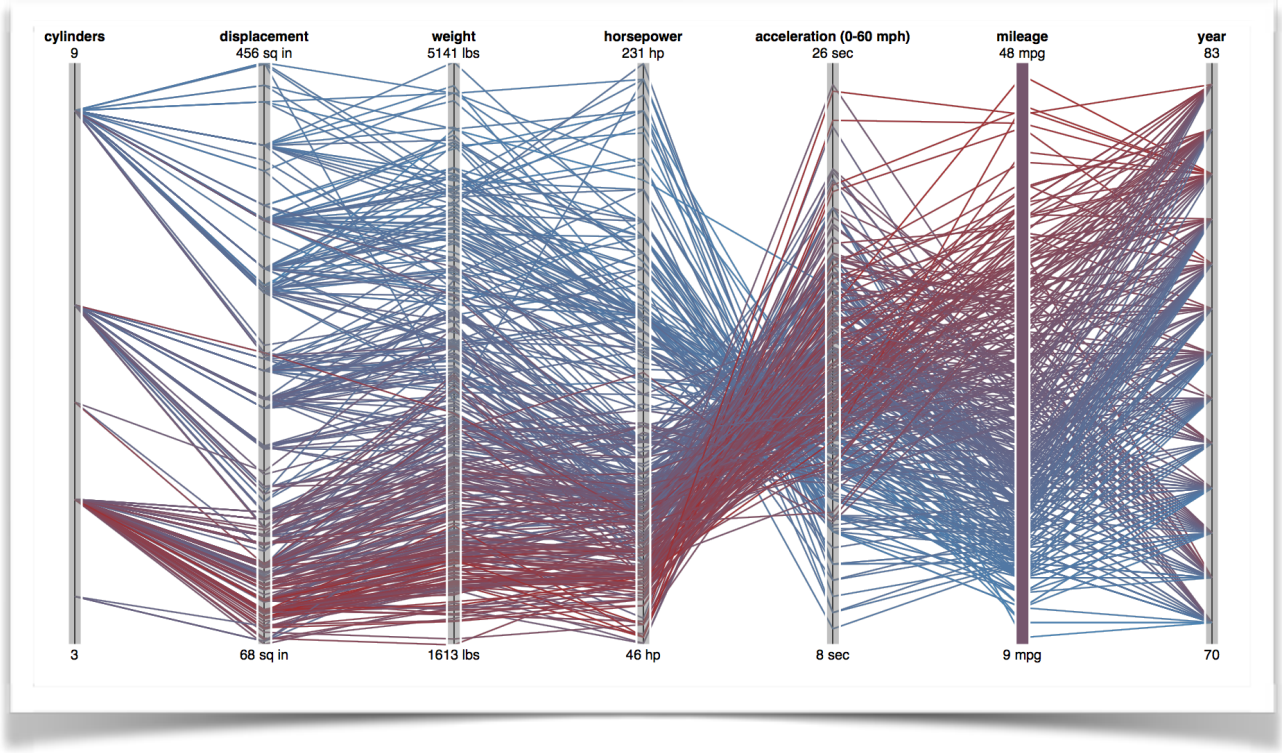
Solution -

Part a)

Background Research -

As instructed I studied a couple of already existing implementations of parallel coordinates and noted what was important according to me. I have briefly mentioned my analysis below,

Link no. 1 - *http://mbostock.github.com/protovis/ex/cars.html*



1.1) Interaction techniques -

a) Filtering along a particular dimension by clicking on it - A ruler is placed with every vertical line that represents attributes. By clicking on the ruler anywhere you can filter the data. A cursor that has four arrows is displayed when you can select an attribute.
b) Select a particular region on the ruler - You can select a ruler assigned for a particular attribute and drag your mouse vertically to select a region (range of values) to display the lines only for that particular range of values for that particular filtered attribute. A cursor of a plus symbol is displayed when you can select a range for the selected attribute.
c) Create a range and drag it up and down to display data only for that range of values. And you can do this for more than one attribute at a time.
d) If a range of values is selected on the ruler for a particular attribute then to select the entire ruler again just click on the vertical line anywhere. Notice that if a range of values is selected on a ruler then the min and max value for that particular ruler disappears and only the min and max for the selected region is displayed.

1.2) Colors used -

a) Red and Blue - They are easy on the eyes, but slightly dull, goes from blue to red as the values increase. If it is in the blue region then blue lines are displayed or else red or if it involves both the regions then the lines are purple.
b) Grey - for disabled lines, when a range is selected.
If a ruler is selected completely then it is a shade of purple or else if a range of values is selected not the ruler then it is a shade of red or blue or purple depending on where the window is placed on the ruler.
c) White Background - The safe bet.
d) Black for the text - It goes well with the white background and the dull shades of red, blue and purple that we are using.

1.3) Other Features -

a) Thickness of lines
It is optimum, not too thick so the intersections between two lines are not visible but enough to be clearly visible.
b) Font - Legible fonts are used. Mentioning the attribute names in bold and values in regular helps distinguish between the two.

Link no. 2 - *http://syntagmatic.github.io/parallel-coordinates/*

This is not really one vis. that they have implemented. It is more like what are the various things that can be done with the parallel coordinates. I have listed the features that I liked or the ones that helped me.

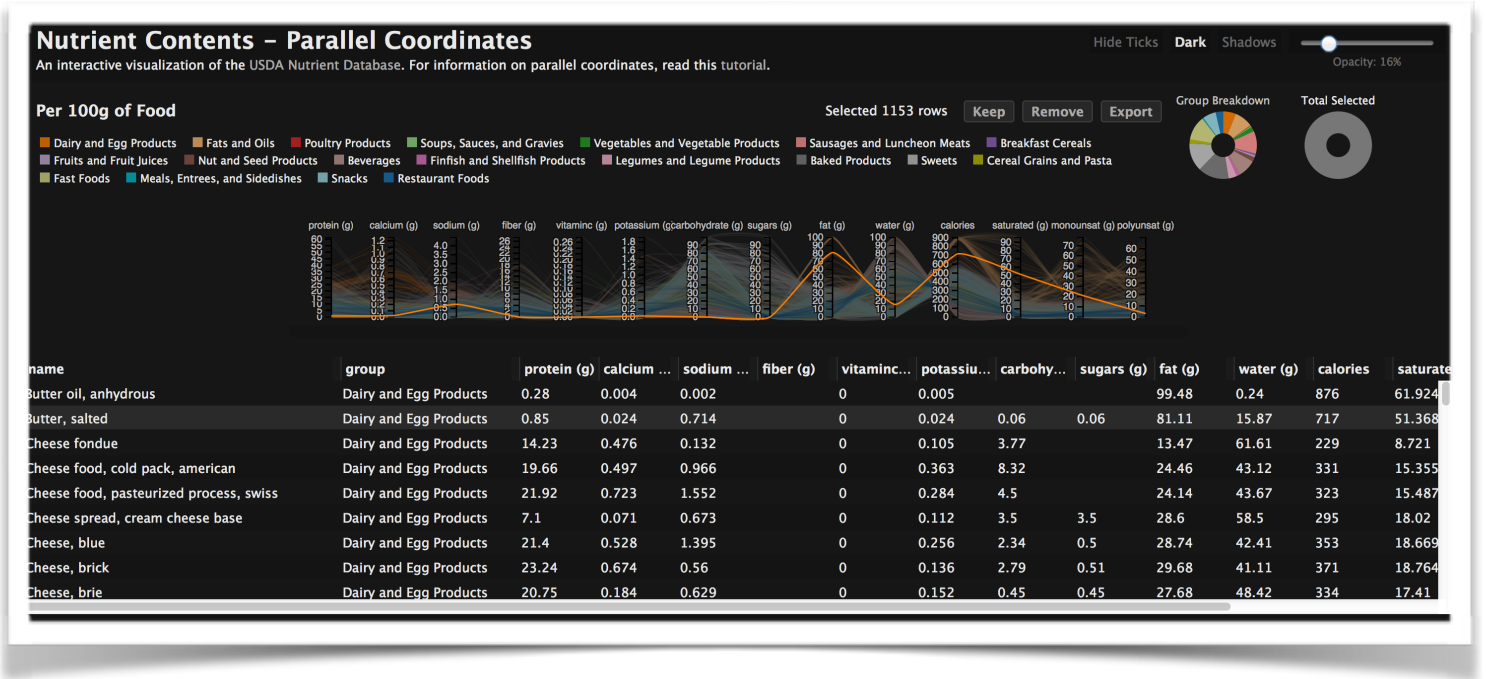
1) Basic (helps you understand the basic understanding of parallel coordinates) -
They have used vertical black lines, numbers 0 to 5 mentioned instead of attribute names, values after an interval along with ticks mentioned on the vertical lines and blue lines indicating the relation/ connection.

2) Brushing -
Same as protovis, almost! An extra feature here is the reset brushes button, it resets all the buttons to full scale at once without having to do them separately.

3) Reordering -
Move the position of vertical axes to study the correlation. Used for shuffling dimensions.

4) Progressive Rendering -
For multiple data points the interaction and filtering works really slow. To improve that, progressive rendering is applied. It is a very good feature to save the user from all the frustration that he/ she has to undergo because of a slow interaction mechanism.

Link no. 3 - *http://exposedata.com/parallel/*



This one is my favorite of all. But since it has a lot of features I will list only the ones that were striking in either a positive or a negative sense.

1.1) Interaction Techniques -

a) Command buttons for minute details like showing/hiding ticks, making the background black or white according to the user's wish (I would totally keep the black background though, It looks much more powerful than the white one). Since white is so much neater it is only sensible to use a command button and leave it unto the user.
b) Slider provided for the opacity - To deal with so many overlapping lines. I think the opacity slider makes so much sense.
c) The entire table provided at the bottom of the screen. For someone who is detailed oriented or specific about the data that they are looking at the table at the bottom helps as the user can select one record and look at the parallel coordinate for that particular record only but the table is taking more screen space than the idiom itself that it might not be the best decision to implement this.
d) Missing progressive rendering feature which they talked about in the syntagmatic page, since there are a lot of data points in this particular vis. the brushing interaction is slow. It seems like if they would have implemented the missing progressive rendering feature they could have improved the vis.

1.2) Colors used -

Multiple colors are used and the shades of colors used are quite pleasant. I liked looking at the visualization because of the color combination used. As mentioned in the previous section I would prefer a black background firstly because it is my favorite color so I am biased towards it and secondly because I feel that it makes the vis. so much more powerful, but then white is neater. Overall, the blend of white, black, various shades of grey and the hues used for the lines makes it look pretty. The legends used for grouping also make looking at the data a little simple.

1.3) *Other Features* -

a) Fonts- Legible and clear.
b) Font Size - Varying font sizes for various labels which makes distinguishing between them easy.

Part b)

Requirements and its solutions -

Requirement no. 1 -

Download the car data set and inspect the data. Process the data into a form that can be used with the FloatTable.pde file reader or modify the reader.

Solution -

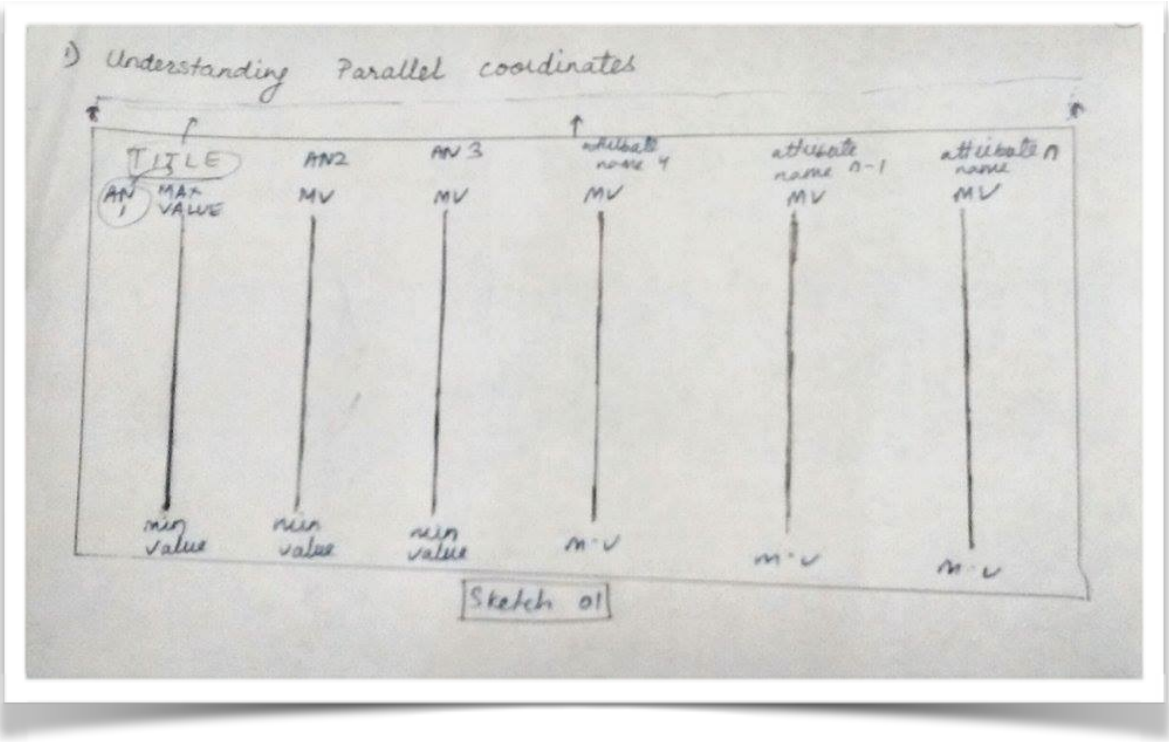
The car data set is given in formats like .okc, .cg and .cf. None of them could be opened on my machine using applications like Pages or TextEdit. I tried to look for some online converters to convert .okc to .csv or .tsv but apparently there are no converters available. So, I started working on the next requirements using the cameras.csv data set for the initial stages of my application. Then I just changed the extension of the .okc file to .tsv to see if that worked and it did for a little part. Finally I looked for sample cars.tsv online and manually modified the .okc file. It was the most tedious task ever. There was an extra row that had to be removed and a column indicating the car names had to be added. And tabs and spaces had to be added at the right places and the data values had to be truncated. I am pretty sure this is not the best way to do it as it was very time consuming and felt like donkey work. The converted file is in the data folder for your perusal.

Requirement no. 2 -

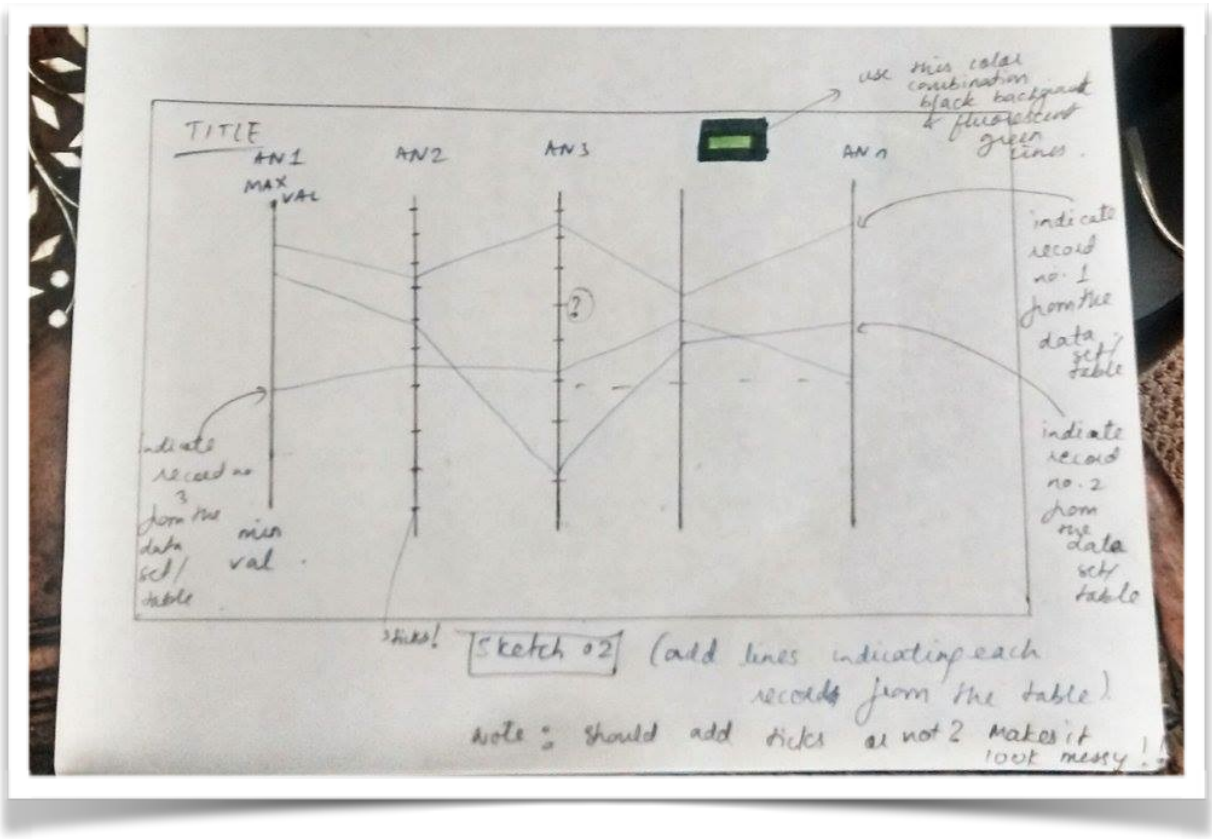
Sketch on paper the concept of your basic visualization and encoding mechanisms. Make sure to include things like labels and titles in your sketch. Think especially about what you might do to deal with with the numerous overlapping lines that are common to parallel coordinates. Summarize your plan in your report, perhaps including scans of drawings that show what you would like to develop in Processing.

Solution -

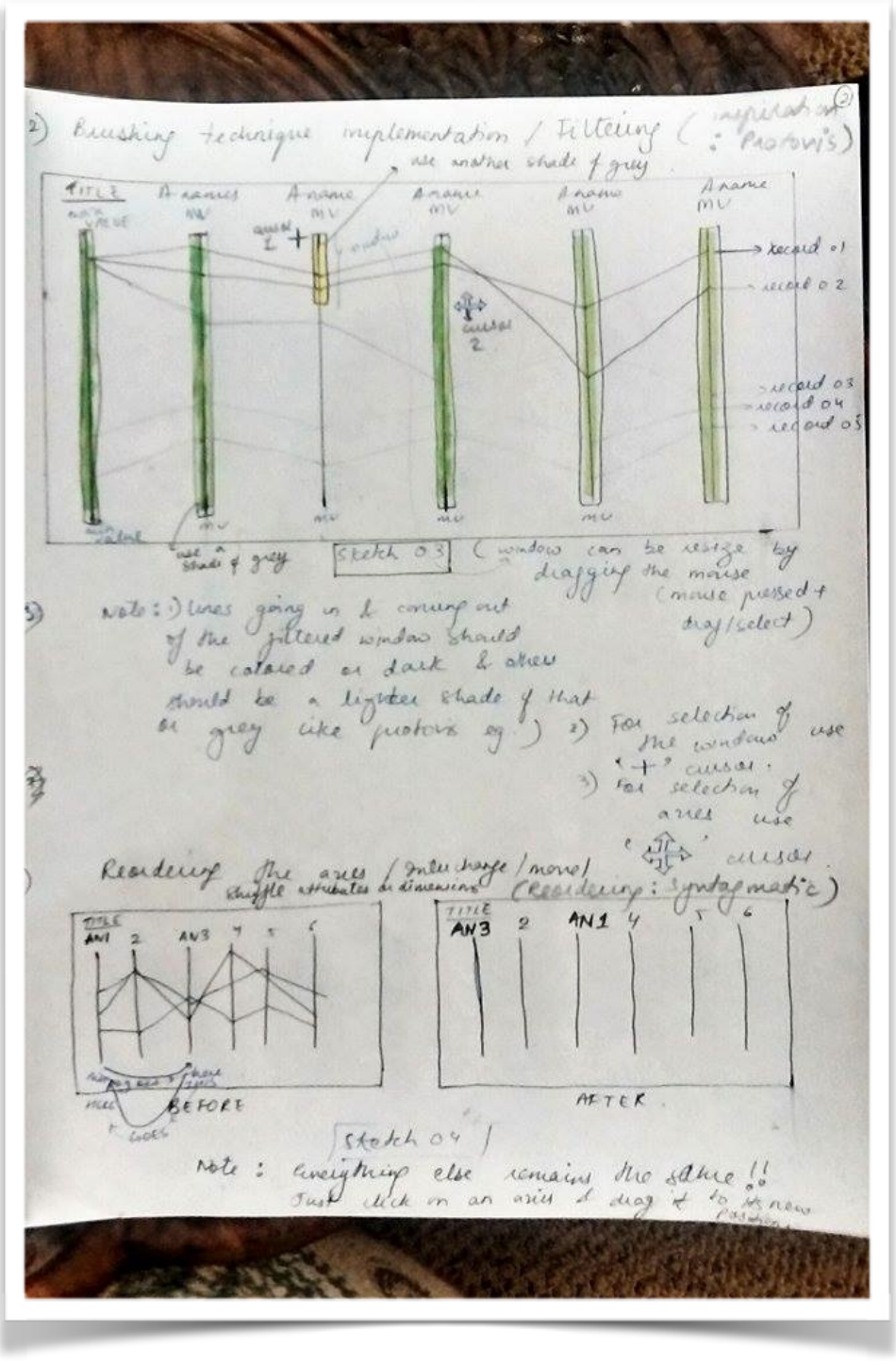
My favorite part of the whole assignment apart from looking at the output screen for a long time was creating these sketches. I tried really hard not going overboard with the colors used for the sketches otherwise it would have been really time consuming. Below are my very ambitious sketches that I confidently and enthusiastically created after looking at the examples. It was disappointing to not be able to all of it in processing though. Pardon me for the poor quality of pictures.



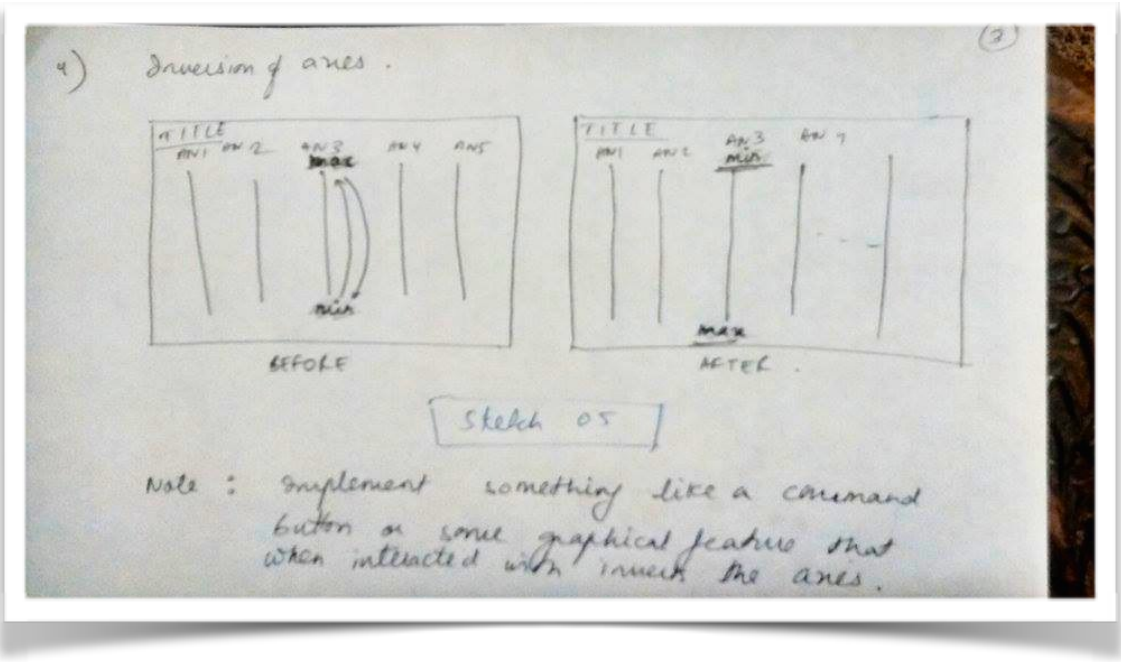
Sketch 01 - This one is a basic representation of how the various axes and labels associated with them would look. It only has a representation of the columns of the dataset. Note that the rows haven't been added yet.



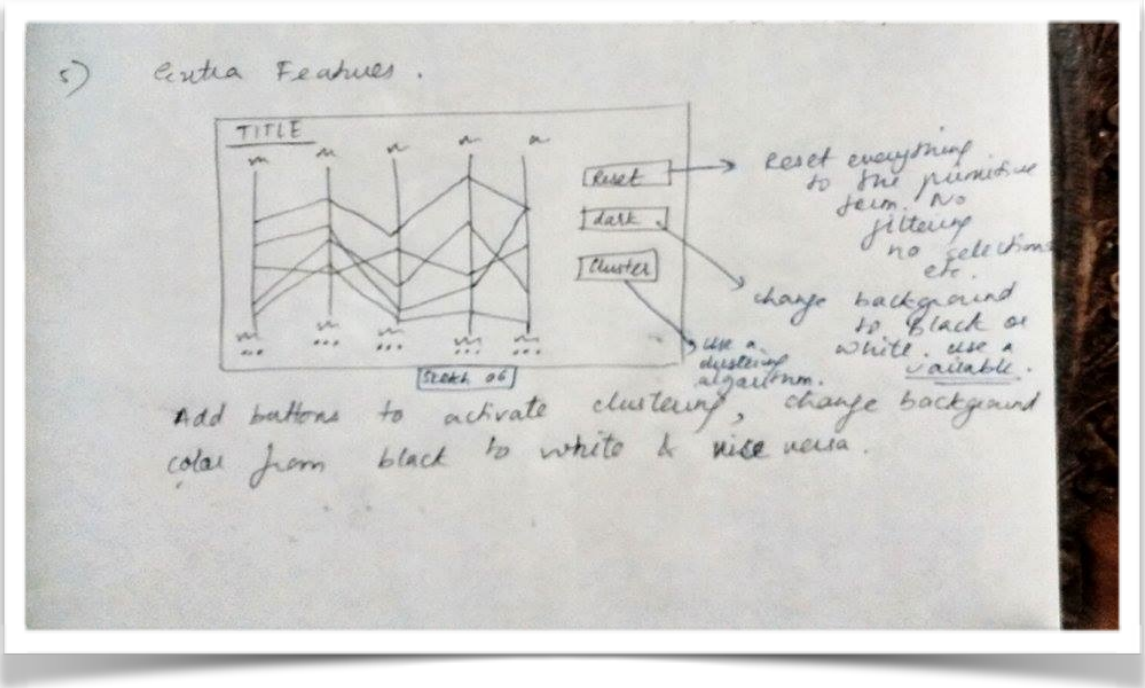
Sketch 02 - This one has the rows added to it. The color codes were decided at this stage and it still wasn't finalized if the ticks were to be added or not hence, half of the axes have ticks and the other half don't.



Sketch 03 and sketch 04 - Sorry for putting them on the same page. Cropping them separately wasn't really convenient. Anyway, sketch 03 is a representation of the brushing or filtering technique where the user would be given a choice to select the range of data for one or more than one columns. In sketch 04, I tried to imagine how the reordering of axes would look like while working on its logic in my head. Programming it was not as easy as drawing it.
sketch 05 - Just reversing the maximum and minimum values of an axis on the basis of a mouse click somewhere on the screen (undecided).



sketch 06 - In this one I basically just listed down the extra features that I wanted to implement. Never knew until the end that I was being ambitious considering the amount of time I had and the time that was needed to actually implement the basic requirements



Requirement no. 3 -

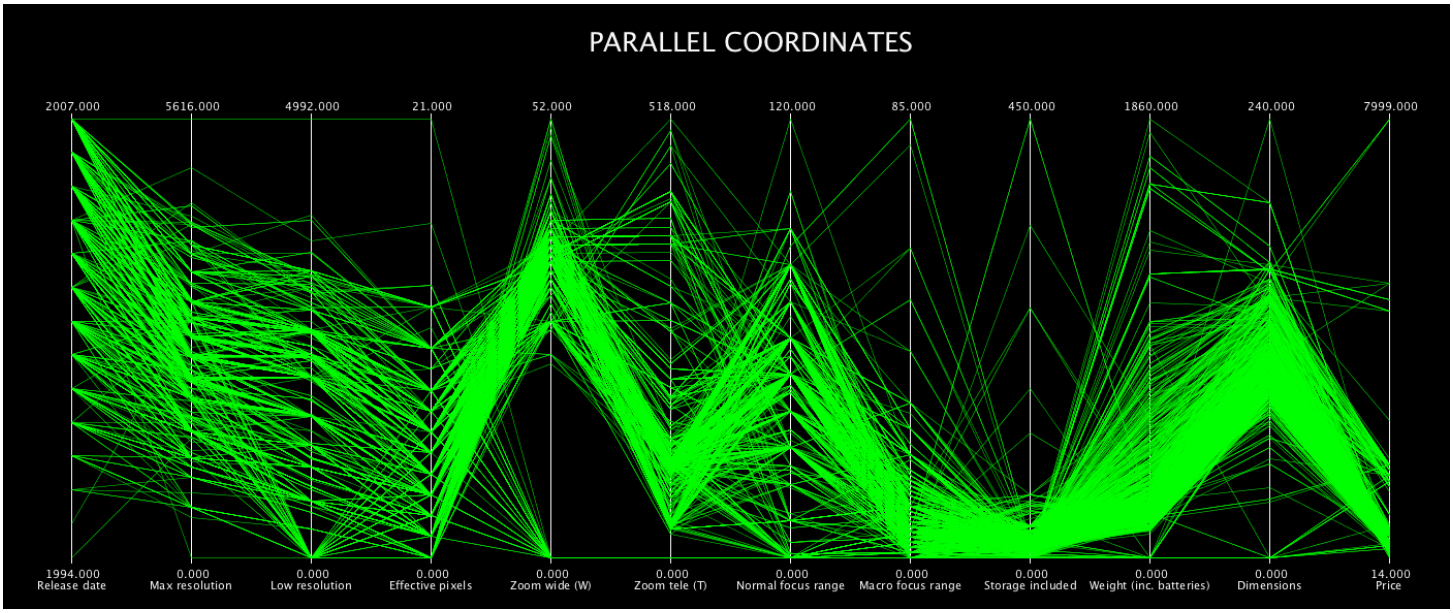
Create a processing project of your initial concept. Write a basic renderer which loads the data and displays it as needed. Be sure to label everything you can: titles, tick marks, ranges, etc. Comment about what was not included from sketch from your initial concept.

Solution -

Please refer to sketch01.pde to see the code for the implementation of this part. For this all I did was set a background in the draw() function and called various functions that created the vertical lines, displayed title for the entire page, displayed axis labels and obtained minimum and maximum values of each column of the database and display it at the ends of each line representing that particular column. This is a static representation of the dataset.

Screenshot of the initial concept -

Data set used - cameras.tsv



What was not included from the sketch?

Ticks - I had initially sketched ticks on half of the axes to see if it worked but I did not implement it because it just added to the clutter.
I implemented everything else from the basic sketch that I had created.

Requirement no. 4 -

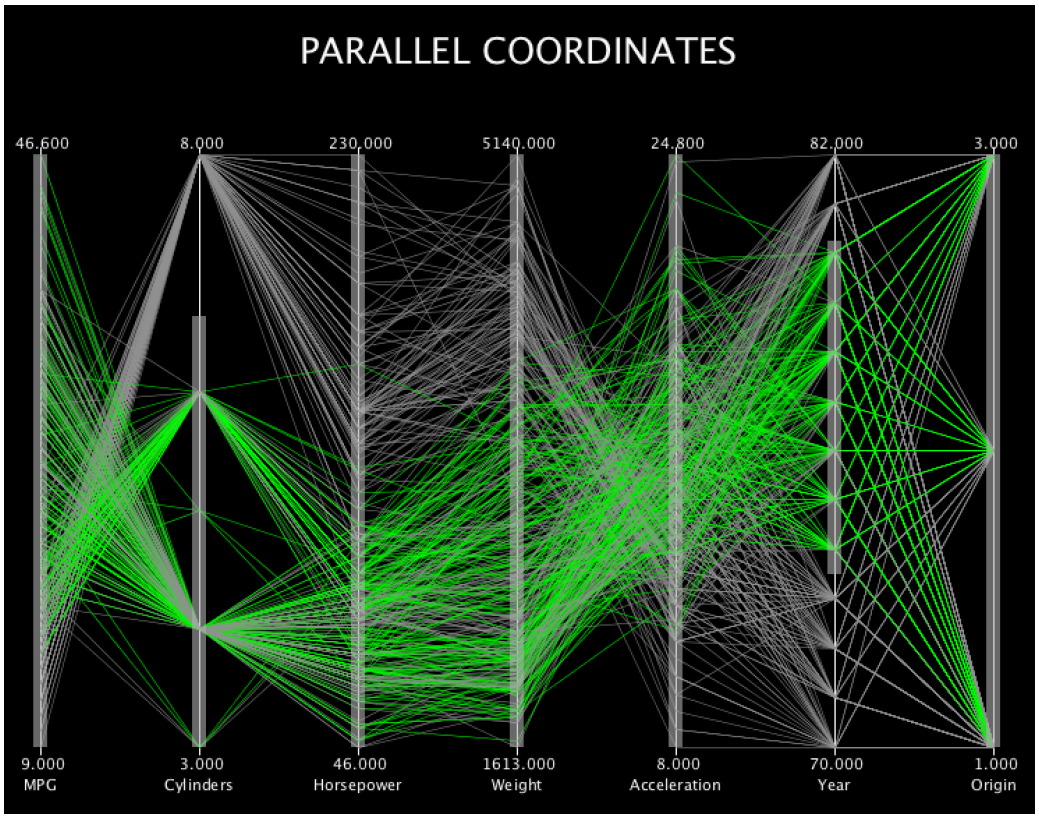
Work on interactivity. You must support the following basic tasks:
• filter the data across multiple attributes
- reorder the axes
- invert the axes
  Consider sketching out the concepts of these interactions before coding. Think about how you want to interact with the axes and the data. How would you design these interactions to make them effective? In your report, you must both explain all interactions that you implemented AND justify your decisions for which interaction mechanisms you chose.

Solution -

1.1) Interaction techniques -

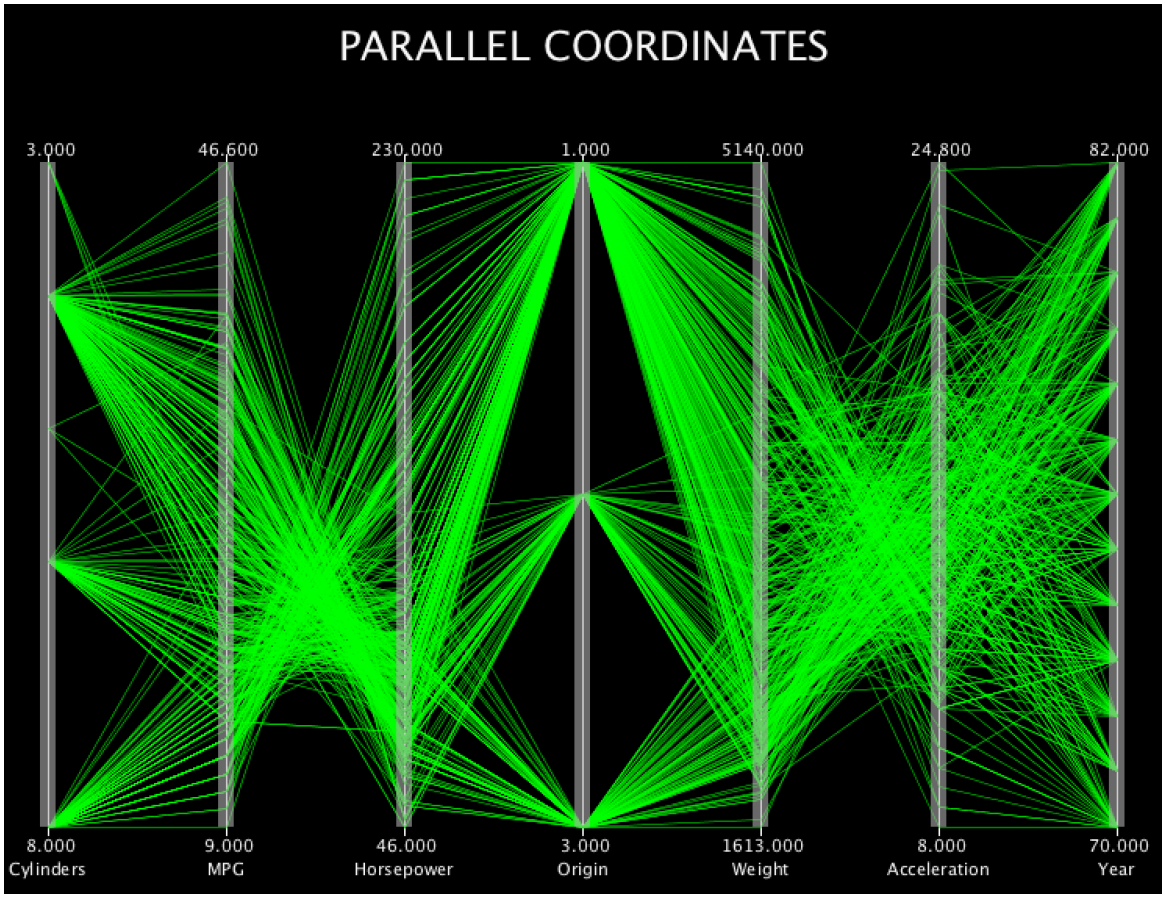
a) Filter the data across multiple attributes -
If the user clicks on the axis at a particular location the range of that axis changes and the minimum and maximum value for that particular axis is changed to the value of the point at which the user clicks. This can be done for all the axes. I chose clicking on the axis instead of dragging the mouse over it because it seemed like a simpler task to do. Observe in the screenshot that the axis cylinders and year are filtered as visible from the reduced size of the grey boxes on the axis. The rows which are not in that filtered range are grey in color and the rest are green.

Screenshot of brushing -
Data set used - cars.txt



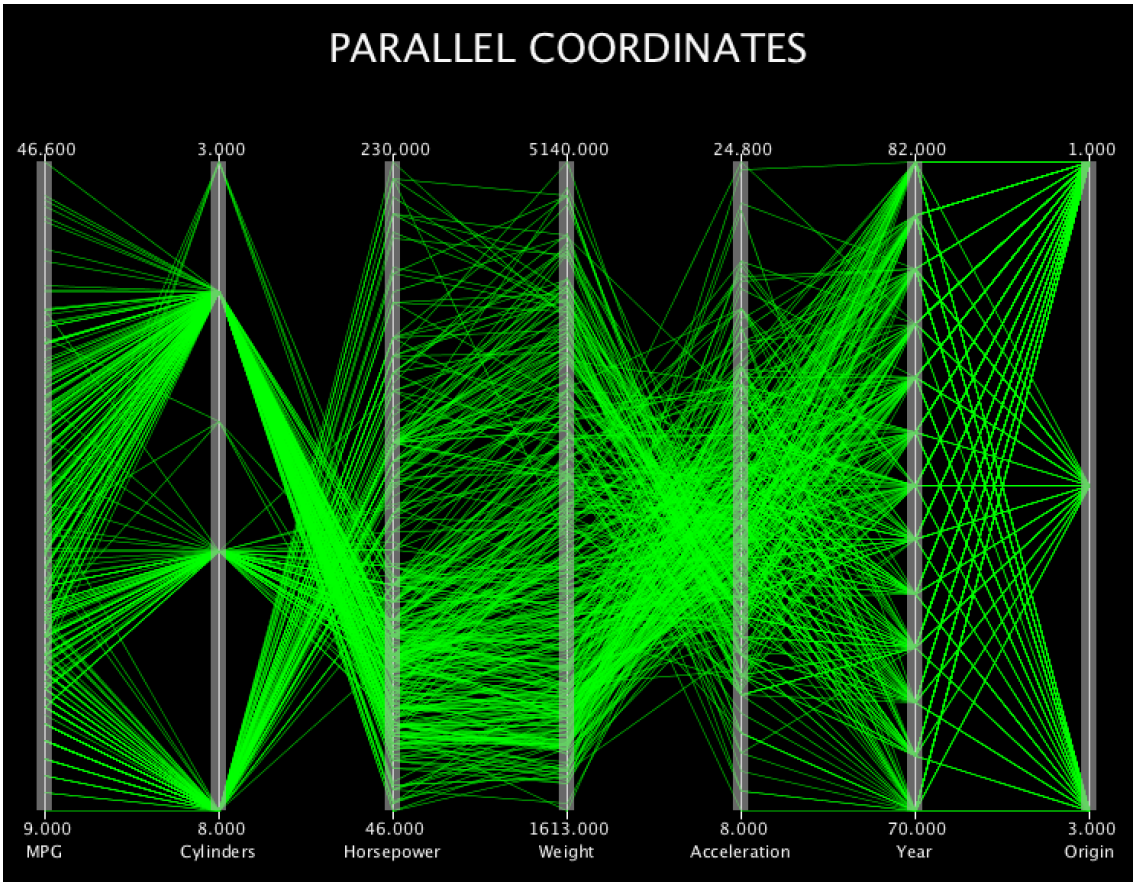
b) Reordering the Axis -
The user can click on an axis and drag it to a new location horizontally. It replaces the axis that is already present at that location. Simple way of interacting. Nothing fancy or complex. I chose this because it was simple and even its implementation was doable. Observe in the screenshot that origin is not in the exact middle and the ones on it's right hand side have shifted to the right by one position each and cylinder has been shifted to the extreme left exchanging places with MPG.

Screenshot of reordering the axis -
Data set used - cars.txt



c) Inverting the axis - Click at the bottom of every axis on it's name to invert the maximum and minimum values of that particular axis. In the screenshot below if you observe then the axis cylinders and origin have been reordered as is visible that their minimum values are at the top of the axes.

Screenshot of inverting the axis -
Data set used - cars.txt



1.2) Choice of Colors - I really wanted to use a black background because I thought it made the visualization so much more powerful. And the version of green used is bright enough and stands out really well. The labels are displayed using a vey light shade of gray which also works well with black and green. Not harsh on the eyes, I really like the color combination. It somehow reminds me of Electronic Dance Music.

Colors used -
a) Active data lines, green - (RGB - 0,255,0)
b) Inactive data lines, grey - (RGB alpha - 144,144,144,255)
c) Filtered box for range, grey - (RGB alpha - 176,176,176,150)
d) Axis Labels, Grey - (RGB - 248,248,248)
e) Background, Black - (RGB - 0,0,0)

1.3) Choice of Font - I used Ariel Narrow for everything because I like the clarity of the font. It doesn't have too many curves so that kind of helps with the black background.

Requirement no. 5 -

Consider improving the visualization further, potentially using either clustering of the data, interactive color legends, or any other techniques that we have discussed in class. Make sure you document any additional features that you added and how they were helpful for data analysis.

Solution -

Unfortunately, I could not add any additional features but I would love to implement clustering. I would have used colors for its implementation. It seems like an important feature in the analysis of the data. And I wanted to add a couple of command buttons for resetting the data and switching between a dark background setting and a white background setting.

Requirement no. 6 -

Use your finished tool to investigate two different datasets and in your report include conclusions you were able to draw from the tool as well as which interaction feature you found to be most fruitful towards data exploration. Include screenshots from your tool that illustrates these conclusions. Finally, critique the utility of parallel coordinates - did you find your implementation easy or hard to use? For the features you identified, could you have found them using other means or without interactivity?

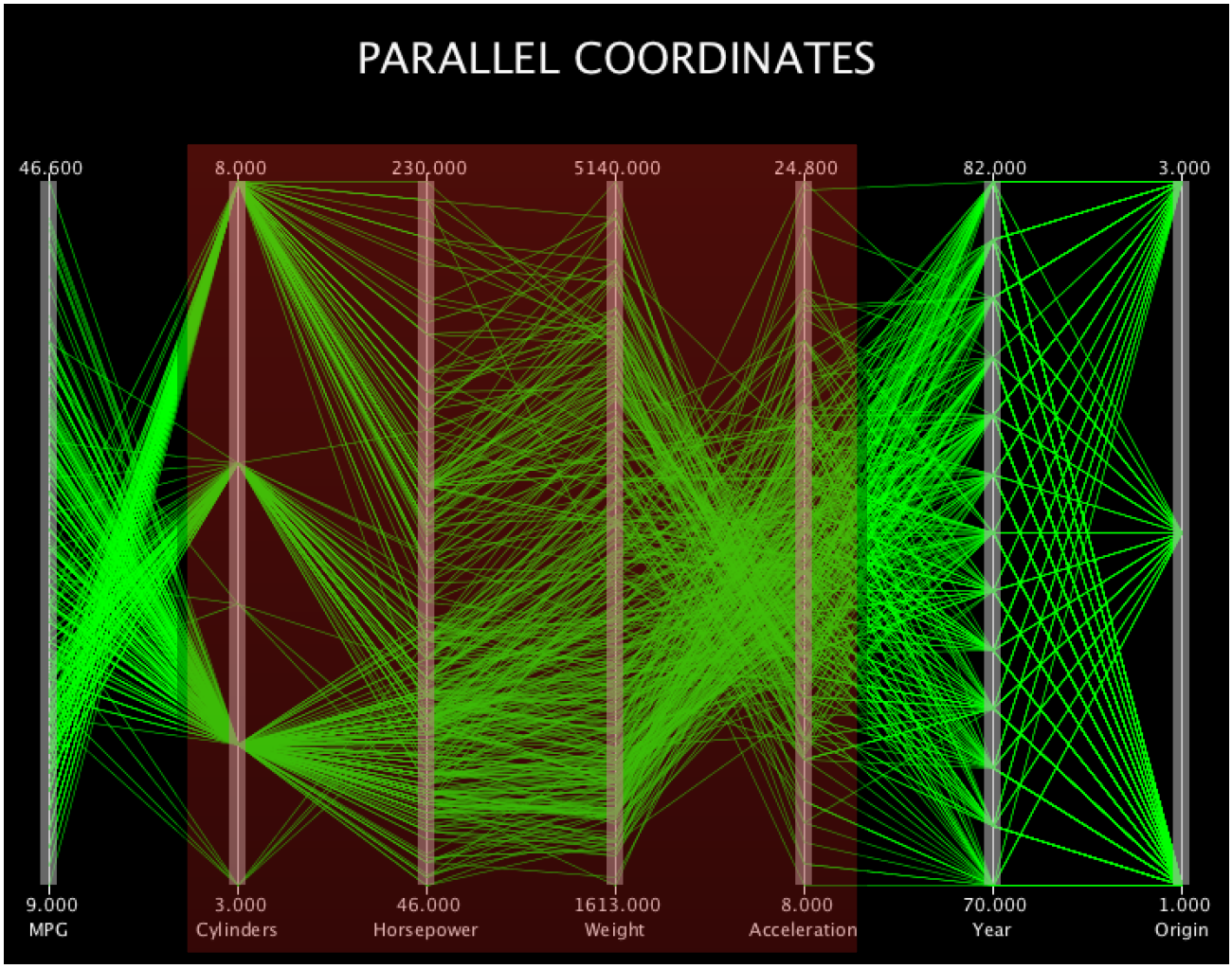
Solution -

Part a) Investigation of two different data sets and drawing conclusions.

Data set used - cars.txt

> I tried using all the interactions that I have implemented for this section to investigate the dataset. Here are some observations and conclusions that I could make using the data given and my application.
>
> (fig no. 1) Firstly, I studied the graph from left to right axis wise and concluded certain things like there is a variety of mileage options available for the cars in this data set. The lines coming out of the axis are well spread throughout the axis unlike no. of cylinders which are fixed and there are only 3 or 4 values for the no. of cylinders in the entire graph. There are a variety of weight options for the car. So if the buyer is specific about the weight of the car then he can look the axis for information. The other prominent axis the one that indicates the year. The cars have always been in production it seems and also periodically. I don't really know what origin means because first I thought it was the year in which the car was produced but the year axis does not represent the same information.

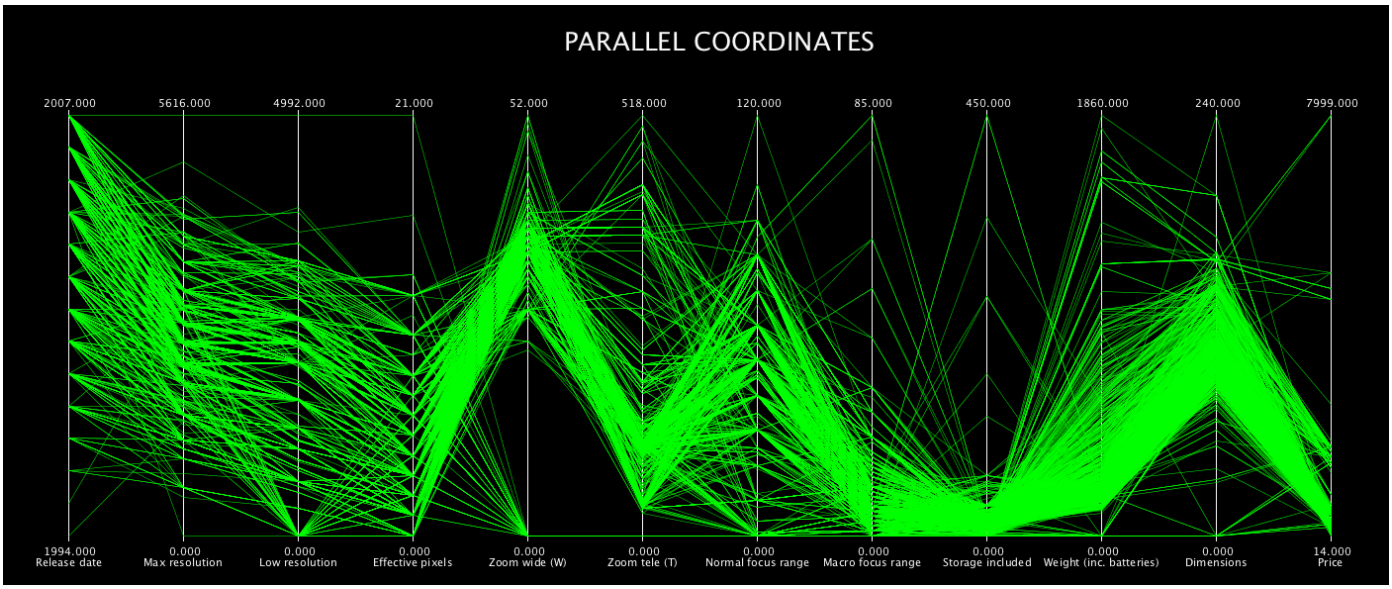
fig no. 1



Let us look at the correlations between a couple of axis for our data. Observe the highlighted portion of the screenshot. You can observe that there is a direct correlation between the no. of cylinders and the horsepower. As the number of cylinder increases so does the horse power. Similar kind of a relation is seen between weight and horsepower. They are almost strongly correlated. But you can't conclude the same about weight and acceleration. Because there are a lot of crossing lines between both the axes.

Next thing that I was curious about if the other parameters had changed with year. So what I did was I reordered the year axis and placed it next to almost every axes and observed the correlation. I concluded that they have started making lighter weighing cars over the years. Between year and the number of cylinders it was observed that there is not such a strong correlation.

Data set used - cameras.tsv



To see if parallel coordinates were any good without the interactivity involved I decided to study the data related to cameras without any interactivity involved. I am just going to paste the same old screenshot again so it is easier to see what I talking about.

What I can see here prominently are the outliers. There is just one camera that was released in 2007 that had a max resolution value and a maximum of all the low resolution values and the most of the effective pixels unlike all the other cameras that were released in 2007. You can also observe that after a certain year the cameras were regularly and periodically released. Then you can observe that there is a slight (not very evident) correlation between the weight of the camera and its dimensions. But there are outliers where the even if the dimensions were more the weight is less. They must have used some high technology light weight fibre for making that camera and it's lenses. Such are the conclusions you can draw from the static representation of the cameras dataset using this tool.

Part b) Critique the utility of parallel coordinates.

From the information that I could gather before and after actually working on the implementation I concluded that visualizing high dimensional data (without aggregating the data) is not a piece of cake. You end up with a cluttered screen and creating that cluttered screen takes up a lot of your time. There are a couple of slightly efficient ways to implement multidimensional data and one of them is parallel coordinates. Cluttering of data represented is a huge problem here because in depth analysis of the data set using parallel coordinates is kind of not possible. Sure, parallel coordinates help you see the trends and correlations in the data and also if there are any outliers

in the data set and if the right colors are used then there is something mesmerizing about so many lines on a single screen trying to convey some information to the viewer or analyst.
While I was switching between the cars dataset and the cameras dataset I realized that as the number of axes increases the screen size also has to be increased and if there are a lot of attributes in a particular table then it is not a very good idea to implement it on a laptop screen. the vertical height is not really a huge problem (even though it is) because the range of values is mapped or can be scaled according to the screen size if you wish.

I think size of the data is a huge parameter for parallel coordinates and it's implementation. I don't think it can handle a lot of data records otherwise the jagged lines or polylines will just end up overlapping heavily making the whole visualization slightly pointless. So for a fairly large dataset parallel coordinates works fine and if clustering is applied then analysis can be successfully done but it is not suitable for a super huge data set is what I have concluded.

Another factor is that since I created this visualization and attended the lectures where it was taught and read books, articles, papers and tutorials about parallel coordinates, I know what it actually is and how to work with it so it was easy for me to use this tool but from my experience after sending the screenshots to my friends and family (showing off your work is one of the most important benefits of studying Computer Graphics and fields related to it), I got reactions like "WOW! Looks pretty! But what is it?" and "Nice colors..but I don't get it". So, I concluded that the user needs to be trained or taught a little before being able to interact with this form of visualization. But don't all the visualization techniques require a little background study?

Scatter plots are a good alternative to parallel coordinates but personally I like (read as love) parallel coordinates. Studying correlation can be done in both scatterplots as well as parallel coordinates but because of the lines and the visible slope of the lines it is easier to study correlation using parallel coordinates. But without interactivity parallel coordinates are not really a sensible approach to look at large data sets. For example finding the correlation between two axes that are not aligned next to each other can be impossible without reordering the axes. Similarly, finding outliers can be difficult if filtering the axes wasn't possible. Interactivity is a must, only then we can fully explore the best of parallel coordinates. I think I love parallel coordinates and I think I already said that a couple of times already.