COMP2043.GRP Final Group Report

Bootstrap Particle Filter simulator

Team 6

2018.04.17

Supervisor: Dr. Liang Dai

Member:

Hejia Qiu zy18720

Cong Liu zy19426

Zexi Song zy15777

Xiang Zhang zy18743

Kaiwen Zhang zy18672

Contents

1 Introduction

1.1 Background of the Project

Sequential Monte Carlo(SMC, also called particle filter), is a widely-used statistical sampling method, which can be exploited to sequentially sample from a sequence of target densities. It deals with the problem of recursively estimating the probability density function $p(x_t|Y_s)$.

Befort the short journey of Introduction to Particle filter, a brief Background of particle filter will be explained in advanced. In our present world, the ability of solving the problem of estimating various quantities in nonlinear dynamic systems plays a significant role in many practical applications. In order to understand how a system performs, we need to obtain certain important quantities related with the system. However, in most cases, we have to estimate them based on various noisy measuremets available from the sytem due to the fact that we do not have direct access to these factors. The state estimation problem is addressed mianly within a probabilistic framework. More specifically, the approach is heavily affected by the Bayesian view of estimation, which implies that the complete solution to the estimation problem is provided by the probability density function $p(x_t|Y_s)$. This density function contains all available information about the state variable x_t, given observations Y_t(Thomas B.Schon, 2006).

1.2 Motivation & Wide implmentation of state estimation

This section illustrates the kind of problems that can be handled using state estimation, by explaining two applications. For instance, first of all, the automotive industry's focus is recently shifting from mechanics to electronics and software. This promote the practical application of estimation theory.

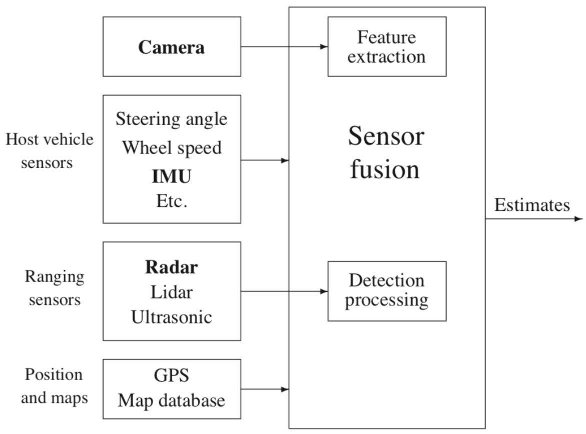


Figure 1: The most important factors enabling future automative safety systems is the availablility of accurate information about the state. (Thomas B. Schon, 2006)

It has been stated above that the data from different sensors can be analyzed to generate the more optimal estimate of the state. This is the idea that is illustrated in Figure 1. The data from Camera, Rador, and GPS are gathered jointly to produce the best possible estimate of the car system.

1.2.1 Automotive Navigation-Example

The objective of this example is to calculate estimates of the road geometry, which are important in several advanced control systems such as lane guidance and collision avoidance. The idea exemplified here follows from the general framework introduced in Figure 1.

Figure 2: when entering a curve, all vehicles start moving in the lateral direction. This information can be used to support the road geometry estimate. (Thomas B. Schon, 2006)

Here information from several sensors is used to obtain better perfomance, than separate use of the sensors would allow for.The main assumption is that the leading vehicles will keep following their lane, and their lateral movement can thus be used to support the otherwise difficult process of road geomtry estimation. For instance, when entering a curve as in Figure 2 the vehicles ahead will start moving to the right and thus there is a high probability that the road is turning to the right. Assuming that the surrounding vehicles will keep following the same lane, is in discrete-time expressed as $y^i_{t+1} = y^i_t + w_t, \qquad w_t \sim \mathcal{N}(0, \mathcal{Q}_{lat})$. Here, y^i denotes the lateral position of vehicle i and w_t denotes Gaussian white noise which is used to accout for model uncertainties. The estimate of the road curvature during an exit phase of a curve is illustrated in Figure 3. The true reference signal was generated using the method proposed by Eidehall and Gustafsson (2006).

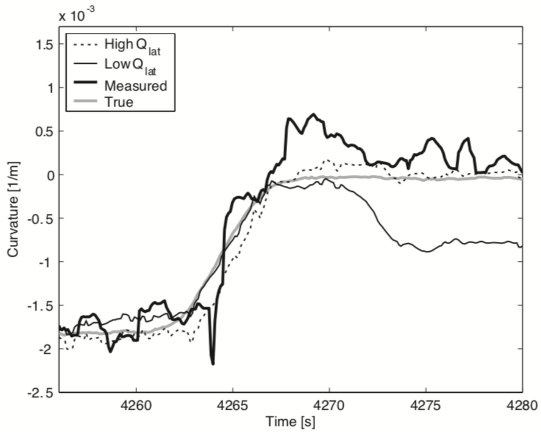


Figure 3: Comparision of estimation performance from two filters, one with a large $\mathcal{Q}_{lat}$ and one with a small $\mathcal{Q}_{lat}$. The raw measuremet signal from the image processing unit is also included. Comparing this raw vision measuremet to the result from the filters clearly illustrates the power of a model based sensor fusion approach

1.3 Scope of the project

This subsection describes the scenario where the product will be used for.

1.4 Introducing two kinds of Auxiliary Particle Filters(APF)

1.4.1 Sequential Monte Carlo Methods

As we have mentioned above that Sequential Monte Carlo methods, or particle filter, provide the solution for state estimation. Then, what does it constitue? It is constituted by a combination of sequential importance sampling and resampling steps. The key idea underlying the particle methods is to represent the probability density function using a set of samples(also referred to as particles, hence the name particle methods) and its associated weights. The density function $p(x_t|Y_s)$ is approximated with an empirial density function,

$$p(x_t|Y_s) \approx \sum_{i=1}^{M} \tilde{q}_t^{(i)} \delta(x_t - x_{t|s}^{(i)}), \qquad \sum_{i=1}^{M} \tilde{q}_t^{(i)} = 1, \qquad \tilde{q}_t^{(i)} \geq 0, \forall i \tag{1}$$

where $\delta(\cdot)$ represents the Dirac delta function and $\tilde{q}_t^{(i)}$ denotes the weight associated with particle $x_{t|s}^{(i)}$. For intuition we can think of each particle x_t^i as a possible system state and the corresponding weight $\tilde{q}_t^{(i)}$ contains information about how probable that particular state is.

There are three core steps for making up sequential Monte Carlo including Resampling, Propagation, and Weighting.

1.4.2 Auxiliary particle filter(APF)

As a principled solution to compute the filtering PDFs $\{(x_t|y_{1:t})\}$ sequentially in time is provided by the following two recursively equations:

$$p(x_t|y_{1:t}) = p(x_t|y_t, y_{1:t-1}) = \frac{p(y_t|x_t)p(x_t|y_{1:t-1})}{p(y_t|y_{1:t-1})} \tag{2}$$

$$p(x_t|y_{1:t-1}) = \int p(x_t|x_{t-1})p(x_{t-1}|y_{1:t-1})d_{x_{t-1}} \tag{3}$$

These two equations are coming from Forward filtering, and the explaination will be included in the Appendixes in detail. Further, the main barriers here are the integral in equation (3) and $p(y_t|y_{1:t-1})$ in equation (2), which are in general not analytically tractable. Nevertheless, the integral can be approximated using an importance sampler targeting the filtering distribution at time $t-1$. This give us the impetus to proceed in an inductive fashion. Thus,assuming we have an empirial approximation of the filtering distribution at time t-1, constituted by N weighted samples, $\{x^i_{t-1}, v^i_{t-1}\}^N_{i=1}$, i.e.

$$\hat{p}^N(x_{t-1}|y_{1:t-1}) = \sum_{i=1}^{N} v^i_{t-1}\delta_{x^i_{t-1}}(x_{t-1}) \tag{4}$$

At time $t = 1$, it is able to obtain a point-mass approximation according to eqution (4), by targeting $p(x_1|y_1) \propto p(y_1|x_1)\mu(x_1)$ with an importance sampler. Inserting the approximation $\hat{p}^N(x_{t-1}|y_{1:t-1})$ into equaiton(3), results in

$$\hat{p}^N(x_t|y_{1:t-1}) = \int p(x_t|x_{t-1})\sum_{i=1}^{N} w^i_{t-1}\delta_{x^i_{t-1}}(x_{t-1})d_{x_{t-1}} = \sum_{i=1}^{N} w^i_{t-1}p(x_t|x^i_{t-1}) \tag{5}$$

Using equation(5) and equation(2), we can evaluate an approximation of the filtering PDF $p(x_t|y_{1:t}$ up to proportionality

$$p(x_t|y_{1:t}) \approx \frac{1}{p(y_t|y_{1:t-1})}\sum_{i=1}^{N} w^i_{t-1}p(y_t|x_t)p(x_t|x^i_{t-1}) \tag{6}$$

As for this opens up for targeting $p(x_t|y_{1:t})$ with an importance sampler. Then, choosing a similar type of mixture as proposal density, namely

$$q(x_t|y_{1:t}) = \sum_{i=1}^{N} w^i_{t-1}q(x_t|x^i_{t-1}, y_t) \tag{7}$$

There are many different options when it comes to choosing the component $q(x_t|x^i_{t-1}, y_t)$ in this mixture. Note that, in general, the proposal density at time t is allowed to depend on the current observation y_t, as indicated by the notation used in equation (7).

Therefore, Using auxiliary variable in the form of a discrete random variable a_t which takes values on the set of integers $\{1, ..., N\}$ in the $q(x_t|y_{1:t})$ can take y_t into account, which make use of this information already when simulating the particles $\{x_t^i\}_{i=1}^N$, to increase the probability of producing samples in the most relevant parts of the state space. This is why we indicate a possible dependence on y_t in the mixture components of the proposal density in (7).The key in this development is to target the joint distribution of (x_t, a_t) with an importance sampler, instead of directly targeting the marginal distribution of x_t

Analogously to above, the mixture proposal (7) can be interpreted as a joint proposal distribution for the pair (x_t, a_t), given by

$$q(x_t, a_t|y_{1:t}) = w_{t-1}^{a_t} q(x_t|x_{t-1}^{a_t}, y_t) \tag{8}$$

Here, a_t should be thought of as an index selecting one of the componets in the sum (7). Generating, independently, N realizations from this joint proposal distribution can be done as follows.

1. Sample the ancestor indices $\{a_t^i\}_{i=1}^N$ according to

$$\mathbb{P}(A_t = i|\{x_{t-1}^j, w_{t-1}^j\}_{j=1}^N) = w_{t-1}^i \qquad i = 1, ..., N \tag{9}$$

   This corresponds to the resampling step of the algorithm. The resampled particles are given as $\bar{x}_{t-1}^i = x_{t-1}^{a_t^i}$ for $i = 1, ..., N$.

2. Propagate the particles to time t by simulating $x_t^i \backsim q(x_t|\bar{x}_{t-1}^i, y_t)$ for $i = 1, ..., N$

The advantage of explicitly introducing and implmenting auxiliary variable lies in the computation of the importance weight. From (6), we have that the joint target distribution for (x_t, a_t) is proportional to what called unnormalized joint target density:

$$w_{t-1}^{a_t} p(y_t|x_t) p(x_t|x_{t-1}^{a_t}) \tag{10}$$

In fact, it is possible to use the information available in the current observation y_t, not only when proposing the new state x_t, but also when proposing its ancestor index a_t. The principle is that we can thereby increase the probability of resampling particles at time $t-1$ that are in agreement with the observation y_t.

It is free to use any proposal distribution that we find appropriate to obtain the pair (x_t, a_t) (Thomas B. Schon, Fredrik Lindsten, 2017). For example, let v be the function that will be used to adapt the proposal distribution for the ancestor indices. For each particle , i.e. for $i = 1, ..., N$, we then compute the quantities

$$v_{t-1}^i := v(x_{t-1}^i, y_t), \tag{11}$$

referred to as adjustment multipliers. The adjustment multipliers are used to construct a proposal distribution for the ancestor index variable a_t

$$\mathbb{P}(A_t = i|\{x_{t-1}^j, w_{t-1}^j\}_{j=1}^N, y_t) = \frac{w_{t-1}^i v_{t-1}^i}{\sum_{t=1}^N w_{t-1}^l v_{t-1}^l} \qquad i = 1, ..., N \tag{12}$$

From the above expression, it is clear that v acts as adjusting the original weights w via multiplication. The underlying ideas of implementation of adjustment multipliers is that by carefully choosing the function v, we can adapt the sampling of the ancestor indices in equation (12) to make use of the information available in the current measuremet y_t
Then, we get the following expression for the i importance weight as the use of adjustment multipliers:

$$\bar{w}_t^i = w(x_{t-1}, x_t, y_t) = \frac{w_{t-1}^{a_t^i} p(y_t|x_t^i) p(x_t^i|x_{t-1}^{a_t^i})}{w_{t-1}^{a_t^i} v(\bar{x}_{t-1}, y_t) q(x_t^i|x_{t-1}^{a_t^i}, y_t)} \tag{13}$$

This completes the algorithm, since these weighted particles in turn can be used to approximate the filtering PDF at time t+1, then at time t+2 and so on.

1.4.3 Bootstrap particle filter

After we have known the Auxiliary particle filter, a pragmatic solution for bootstrap particle filter is to choose the proposal density according to

$$w_{t-1}^{a_t^i} v_{t-1}^{a_t} = w_{t-1}^{a_t}, \tag{14}$$

$$q(x_t|x_{t-1}^{a_t}, y_t) = p(x_t|x_{t-1}^j), \tag{15}$$

$$q(x_t|y_{1:t}) = \sum_{j=1}^{N} w_{t-1}^j p(x_t|x_{t-1}^j) \tag{16}$$

Therefore, it follows that the weight function is given by

$$\bar{w}_t^i = \frac{w_{t-1}^{a_t^i} p(y_t|x_t^i) p(x_t^i|x_{t-1}^{a_t^i})}{w_{t-1}^j p(x_t|x_{t-1}^j)} = p(y_t|x_t) \tag{17}$$

This completes the algorithm.

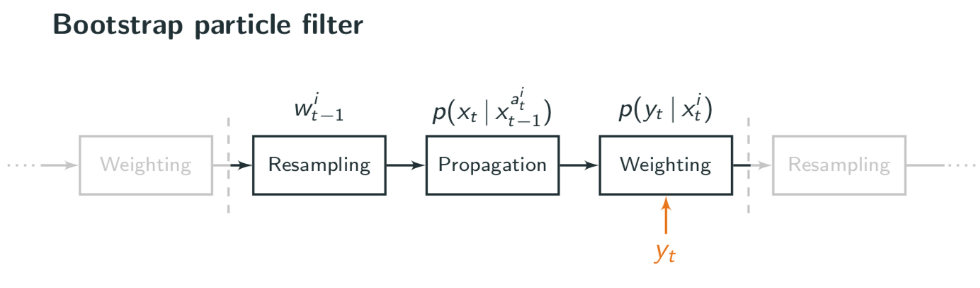


Figure 4: bootstrap particle filter

1.4.4 Fully adapted particle filter

In order to adapt the proposals to the information that is available in the current observation y_t. A natural choice as for the proposal for the state x_t is to use

$$q(x_t|x_{t-1}, y_t) = p(x_t|x_{t-1}, y_t), \tag{18}$$

From Bayes'rule, we can write

$$p(x_t|x_{t-1}, y_t) = \frac{p(y_t|x_t)p(x_t|x_{t-1})}{p(y_t|x_{t-1})}, \tag{19}$$

Plugging this expression into the denominator of equation (13) we obtain

$$\bar{w}_t^i = w(x_{t-1}, x_t, y_t) = \frac{p(y_t|x_{t-1})}{v(x_{t-1}, y_t)}, \tag{20}$$

When it confront the choice to decide the adjustment multipliers. In particular, we see the choice $v(x_{t-1}, y_t) = p(y_t|x_{t-1})$ will lead to a weight function that is identically equal to 1. In other word, we obtain particles that are generated in such a way that they are equally informative about the target distribution.

$$\bar{w}_t^i = 1. \tag{21}$$

This completes the algorithm.

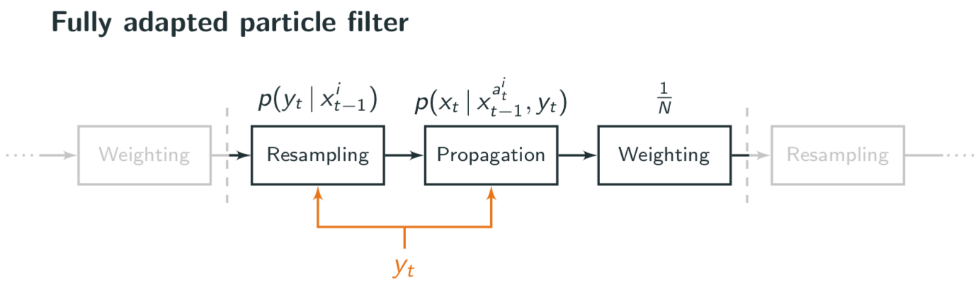


Figure 5: Fully adapted particle filter

2 Requirement Specification

2.1 Introduction

Particle Filter Simulator is a lightweight application which is mainly used by college students and teachers. Because Particle Filter Simulator is aimed for teaching, thus, in order to meet the main needs of users and provide positive user experience, a large amount of significant requirements should be carefully considered and implemented. In this part of report, it will illustrate the process of gathering and implementing requirement and provide a list of different types of requirements. For each specific requirement, it will be analyzed and explained in more details.

2.2 User case Diagram

Use case diagram could describe how users interact with system and it could be the first step from requirement specification to software implementation. The actors, use cases and system boundary will be shown as follow. It could help developers implement the main function of Particle Filter Simulator and provide a clear diagram for users to understand each parts of the software.

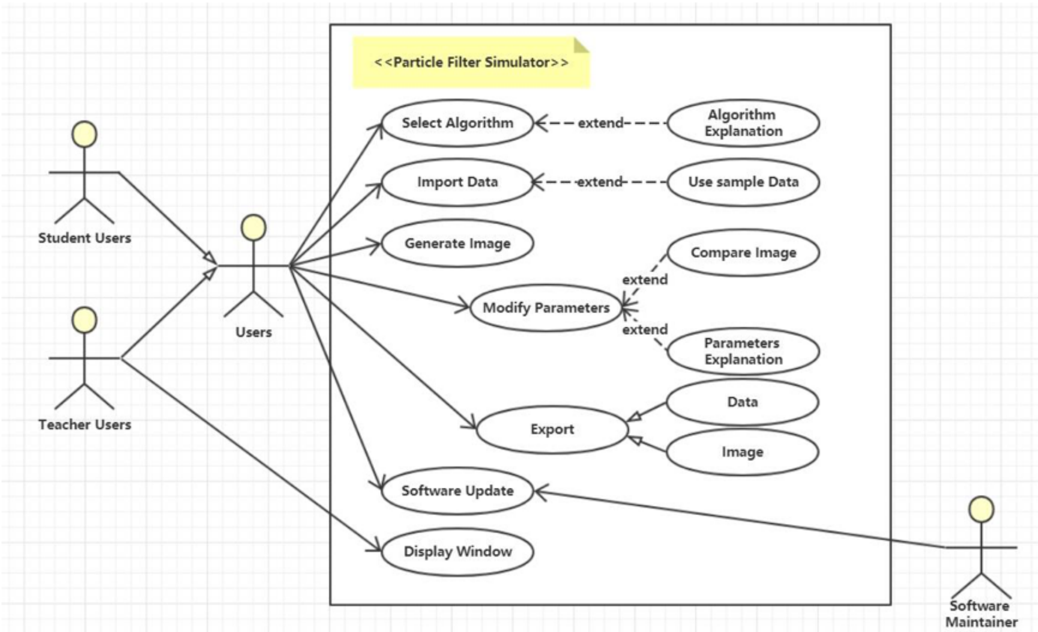


Figure 6: User case diagram includes the actors, use cases, and system boundary required in Particle Filter Simulator System.

2.3 Prototype

Software prototype provides a brief user interface and it has been divided the whole software into several parts. Developer could better implement division and cooperation by focusing on each different component in prototype. It also provides a feasible option to validate if the software design achieves expected user experience.

1. Start interface: After the software is opened by user, user should first select different algorithms. It is the first impression of user experience and help users gets start easily.

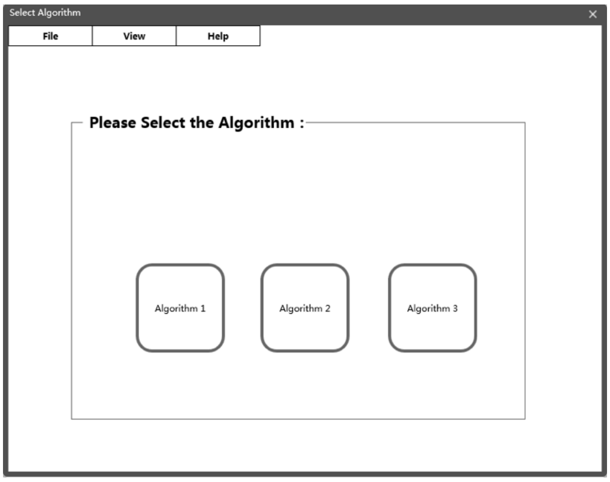


Figure 7: Start interface

2. Navigating bar: Navigating bar is placed on the top of window and it contains three main options: File, View and Help.
   (a) File options include the function of operating input and output file.
   (b) View options include the function of setting window on the top or changing the software appearance.
   (c) Help options include the help document and software update.

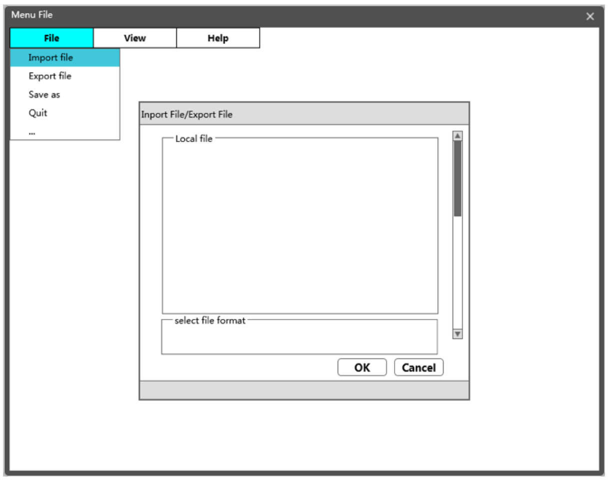


Figure 8: Navigating bar-File

Figure 9: Navigating bar-View

Figure 10: Navigating bar-Help

3. Main interface(Figure 11 & 12 & 13): Main interface of generating data and image, which includes three components: function button, outcome image and parameter controller. Function button includes several functional buttons to deal with data and image. Outcome image includes an area to generate a plot image with coordinate axis and two scroll bar. Parameter controller contains a list containing parameter name, several slider used to adjust parameter value and a start and refresh button.

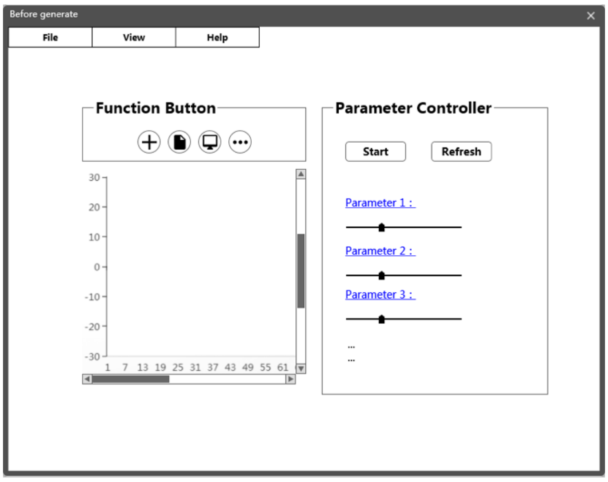


Figure 11: Main interface-prepare

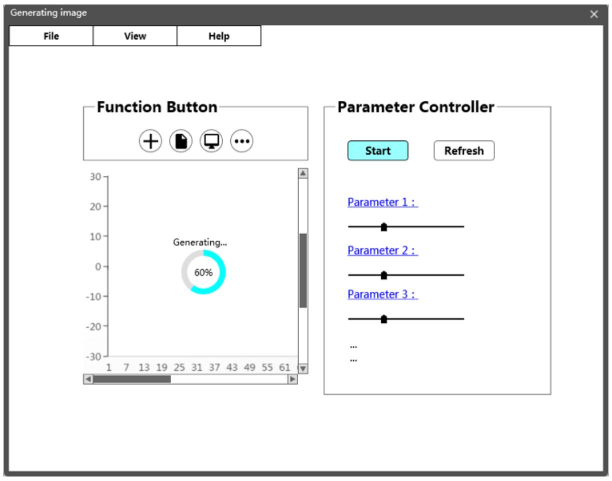


Figure 12: Main interface-executing

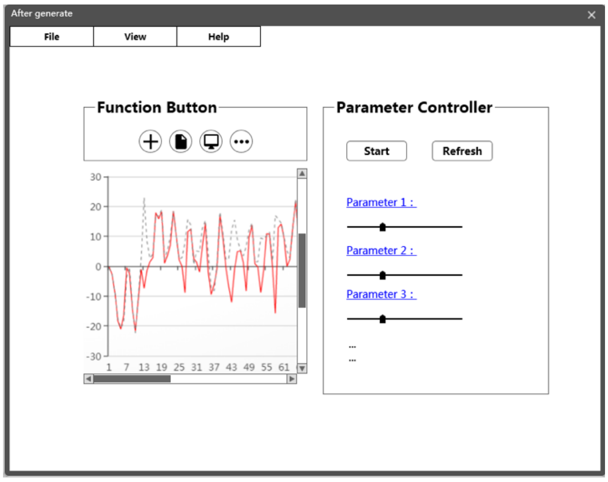


Figure 13: Main interface-complete

2.4 Requirements list

During the procedure of software design and requirements Implementation, developers found that some requirements is difficult to implement and there are also some new requirements need to be considered. In this section, it mainly focusing on the requirements that has been appeared in the software design process and which of them have been implemented or not.

2.4.1 Functional requirements

Particle Filter Simulator is using the algorithm selected by user to generate the outcome image. These are the main functional requirements appearing during the software design process:

1. The software should illustrate thecorrectlyusagemethod in help document.
2. The software should provide the function of selecting different algorithms
3. The software should display the brief explanation of algorithms when user is selecting an algorithm.
4. The software should provide some existing sample data to users without other specific data.
5. The software needs prompt and check the format of the file when users import specific sample data.
6. The software should use the algorithm which is selected by user to deal with the import sample data and generate the outcome image.

7. The software should provide modifiable parameter list and brief explanations of each parameter.

8. The software should have the function of modifying parameters which are initially set as default value.

9. The software should implement the function of exporting images, data, or both of them according to the users decision.

10. The software should permit users to combine two images together to compare the differences between them.

11. The software should provide the function of clearing the current data and results if users want to import another sample data or select another algorithm.

12. The software should provide a function of putting the software window into the top.

13. A restart button should be provided to handle the situation of software crashing.

14. The software need to check version update automatically and submit bugs which are collected by users to software maintainer.

15. When the software is using algorithm to generate image, a progress bar or a waiting hint should be provided which is used to display the running state of software.

16. The range of permanents should be limited to avoid the situation that increasing running time of algorithm and software crashing.

2.4.2 Non-functional requirements

Performance & security & Environment: The non-functional requirements are given as follow which include the environment requirements, performance requirements and data requirements:

1. The window size of the software should be suitable for demonstration.

2. The user interface should be convenient for user activity.

3. The running time of using the algorithm to deal with the sample data should be as quick as possible, for instance, it should be less than 5 seconds with the consideration of the computability of Javascript.

4. The system resources occupied by software should be limited.

5. The software needs to run on major platforms and operating systems.

6. The software should be convenient for update and extension.

7. The outcome image should have positive distributing accuracy when the value of several parameters is limited.

8. The software requires appropriate software capacity to be convenient for users download and use in a short period of time.

9. It should be convenient to passing parameter data between each component in different part of software.

10. The software is free, it is better that the development and maintenance costs should be properly reduced.

3 Development Strategies

3.1 Agile

Stakeholders:

1. Teachers: Teachers might use this software to help explaining particle filter for teaching purpose. Main requirement of this system is to visualize the output of the algorithm. Before the visualization, teacher will be able to set the parameters of particle filter, then the algorithm will run in the background.

2. Students: Students require same functions as teacher and expect that this software will be able to provide data import/export functions. By those functions they can compare output of the algorithm with different parameters, in the form of both data and visual image. Those functions can also be used to submit the output to teacher if required.

3.2 Plan & Task assignments

1. For this group project, as it is founded on deep mathematical knowledge, two members mostly focus on the realization of the algorithm, including the implementation of the algorithm in background code and related documents. (Cong Liu, Kaiwen Zhang)

2. One member is responsible for choosing developing techniques, including developing language, framework and tool components. He is also supposed to set up the software development framework and help other members build the develop environment and solve technique problems in developing. (Hejia Qiu)

3. One member will cooperate with the former one to develop the software, both in design and code, and supposed to complete some functions and test work. Prototype and interface design is also supposed to be completed by this member with Hejia. (Xiang Zhang)

4. As this group project requires meeting and documents, one member is supposed to record meeting minutes and necessary documents, for both developers and users. (Zexi Song)

4 Design & Implementation

4.1 Introduction

Considering the requirements and developing cost of the whole project, we chose Node.js and Electron as develop platform and Vue.js as front-end framework. Those techniques help to develop a light-weight and cross-platform app, which works as an algorithm simulator.
Other tools, such as Echarts.js, are used for satisfying specific requirements.

4.2 Techniques

As this software is designed as an academic teaching tool and might run on schools or users own machine, we are supposed to choose techniques that are available to develop light-weight and cross-platform PC application.
The implement of data visualization and algorithm computation is the basic requirement to be considered. Learning cost and the flexibility for cooperative development will also be considered as the group is supposed to learn and develop as a team.

4.2.1 Programming Language for Development

- JavaScript, HTML and CSS
  As this project requires data visualization function, front-end is a suitable choice as there exists visualizing frameworks like Echarts.js based on JavaScript and canvas in front-end development. Front-end developing also decrease the cost of interface development and can easily improve UI by CSS tools, such as Bootstrap. In addition, group members have basic skill and developing experience on it so use front-end language can decrease learning cost.

4.2.2 Developing framework

- Electron
  Electron is an open-source framework to build cross-platform desktop Apps with front-end language. It is based on Node.js and npm source.

- Vue.js
  Vue.js is a progressive JavaScript framework. It is approachable, versatile and performant.

The main advantage of this framework is that with it we can divide the software into components. It is much easier for group project that group members will not get confused by reading others code, but just focus on their own components.

4.2.3 Developing Tools

- Vue-cli
  Vue-cli is a full system for scaffolding Vue.js project and a popular tooling baseline for the Vue ecosystem. With this tool, the components structure becomes much clearer.
  In Vue-cli, we do not need to register vue components in HTML files but have specific space for them. JavaScript and CSS code are also wrote in component files.

- Electron-Vue
  With this tool we have no need to manually set up electron apps, based on vue-cli system.

4.2.4 Package Tools

- Electron-builder
  This is a useful tool to package an Electron app based on npm/yarn source. By using this tool, we can easily package the software into cross platform installation package and installation-free software. The size of package will also be small. In addition, auto-update function is supported by Electron-builder.

4.2.5 Data Visualization Tools

- Echarts.js
  ECharts.js is a powerful, interactive charting and visualization library for browser.
  The API of this framework can satisfy all requirement of the software, such as data zooming and image exporting. It provides a strong library of data visualization for our further work.

4.2.6 Math Tools

- Math.js
  Math.js is an extensive math library for JavaScript and Node.js and is used in this project to do matrix calculation.
  However, it still lacks many functions for matrix calculation, comparing with MATLAB library. With this problem, we have made a lot of efforts for the implementation of the algorithm.

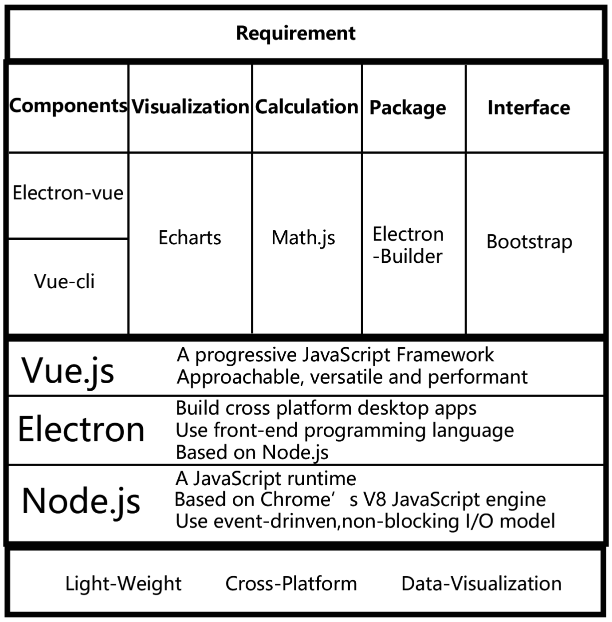


Figure 14:

4.3 Schedule

09.28 Hejia Qiu shard how to use Git GUI and Github in the group then the whole group learned how to use git to develop and version control.
10.13 We have the first formal meeting with supervisor Dr.Liang Dai and he provide us some basic reading materials and give guidance on math.
10.13-10.18 We keep learning with existing material and guidance, try to understand how Sequential Monte Carlo works.
10.18-10.30

We find MATLAB code of particle filter and use different pairs of parameters to understand how them works in a visible way. Cong made considerable progress here and explained to group members.
10.30-11.14 During the former meeting we decided to divide into two small groups, one for algorithm (Cong, Zexi) and another for software design (Hejia, Xiang, Kaiwen). Every small group can focus on their work and that is an efficient way. Software design contains software positioning, requirement confirmation, programming language choosing and function UI design.
11.14-11.23 We designed the software prototype in detail together with the understanding of algorithm. Kaiwen leaded the team finishing the requirement document.
11.23-12.07 We gathered together for interim group report writing and communicated about software design in detail. Xiang provided the first draft of prototype.
12.07-01.10 Hejia tried several develop frameworks and showed in the group to compare advantages and disadvantages. Cong made progress on algorithm and Kaiwen, Xiang and Zexi keep completing software design.
01.10-2.28 Discuss which language should be used to develop our project. Decided to use an open source framework on Github - Electron, which is based on Node.js technique.
During those days, group members started reading relative documents and download examples to learn project structures. Hejia also helped all members to build develop environment.
2.28-3.6 Based on former technical selection work, the group decided to use those two develop tools: Electron-Vue and Electron-Builder. Electron-Vue is based on Vue.js a popular opensource front-end framework to make software structure as components. Electron-Builder is an efficient software packaging tool for Electron based on webpack and npm. We had made tests to ensure those two frameworks works as we expected.
3.6-3.13 Echarts.js, a data visualization framework by Baidu, is chosen by the group for chart drawing on GUI. Hejia learned Echarts.js, made some demos and discussed with Cong and Kaiwen, made sure this framework can satisfy all requirement of the algorithm.
3.13-3.20 Cong wrote algorithm in Python based on former MATLAB reference and math knowledges. We discussed the efficiency and user cost of adding Python environment into the install package. Hejia and Cong made the demo based on Electron-Vue and the group made details assign.
3.20 Detailed assign each persons task. Hejia QIU: Software structure, data visualization and image processing functions. Cong LIU: Core algorithm complement and MATLAB algorithm transformation. Kaiwen ZHANG: Repetition algorithm from MATLAB to JavaScript with Cong. Xiang ZHANG: Write menu bar and complete data import/export functions. Zexi SONG: Minutes of meetings and related documents of the software.
3.21-3.27 Xiang finished the first draft of menu bar without I/O functions.
Hejia reset the software structure for decrease develop cost and import the chart

component into the software then set basic window parameters.
Cong and Kaiwen fixed some bugs of the algorithm with the help of supervisor and finished the first draft of algorithm on JavaScript.
3.28-3.31 Optimized former functions and fixed bugs.
4.1-4.4 Cong added data store function in software. Hejia finished first draft of interface and completed interactive function. Xiang updated I/O functions on menu bar. Zexi finished part of documents.
4.7 Hejia import Bootstrap-Vue component for improve UI. Xiang fixed type error in importing data with the help of Cong and Hejia.
4.8 Xiang organized the group to have final test together.

5 Test

The test of Particle Filter Simulator software aims to discover the functional and service performance-related problems through variety test approaches. According to apposite approaches, basic errors and drawbacks on functions and logic can be detected and reduced. Ensure that software can implement its functions as expected.
In this software, the whole functions will be tested including importing a json file, exporting data, drawing chart of algorithm running result, refreshing the chart and altering the parameter value of Particle Filter algorithm. The correctness of algorithm implementation will be tested particularly. Plus, parts of non-functional features which are significant to the software will be tested as well, for instance, system compatibility, installation and uninstallation, and performance in response time.

5.1 Pass/Fail Criteria

For functional features, the pass criterion applies to all of them: implement correctly without any error and warning. And if the condition do not reach, the feature is failed. For non-functional features, pass or fail criteria should be considered separately.

1. System compatibility: The criterion of system compatibility is whether the software can implement its functions well as a cross-platform software. Two mainstream operating systems must be satisfied, Windows and Mac OS.

2. Installation: installation pass criterion is verification that software works properly after installation. It is failed if installation error occurs.

3. Uninstallation: the pass criterion of uninstallation is the software can be uninstalled clearly and has no harmful effects on the system.

4. Performance(response time): Miller(1968) indicated there are three levels of response time for users. 0.1 to 0.2s: users may think it is immediate response.1 5s: The users feel that the interaction is basically smooth. The users will notice the delay. 8s or more: Users will follow the dialog. A prompt or progress bar is required to confirm that the system is still in process. Treat the

information as a reference, the pass criterion is set to most response time of operations should be under 5s, otherwise, a progress bar should be showed to tell users the computer does work. Over 5s and no progress bar or other prompt will be considered as failure.

5.2 Approach

The test will use black box testing to check functions with equivalence partition testing and boundary testing, while white box testing and unit testing to check the source code. Matlab will be applied to support mathematical principles.
Performance will be tested by Chrome devtools to present the response time of every operation. All test cases are manual tests. The testing table is attached on the Appendices.

6 Evaluation and Reflection

6.1 Assessment of the product

After two semesters study and work, we finally completed our software that realizes the data visualization of Bootstrap particle filter, however, the fully adapted particle filter has not been realized. The following assessments of the product analyzes both achievements and failure.

1. The software of our group has completed most of the requirements.
2. The size of our software packages is very small, and it is easy for customers to download and use.
3. Our software can be cross-platform.
4. Our software has more than one installation approaches for customers to choose. They could download an installation package or portable version to start using our program.
5. Our software implements data visualization, the result of each change in data is straightforward on the graphics.
6. Our software can import and export data. We can not only use an existing data to complete data visualization, but also export data according to the image that we have generated.
7. The speed of the software is not optimized well. The software would run for 1.6 seconds if we set the number of particles to 100. But if the number is increased to 500, it would run for 7.9 seconds.

8. Our softwares language choice is not the optimal choice, we had tried some other languages such as python. Using python could optimize operation time but it adds new problems which increase user load. For the sake of the user, we use JavaScript as the language of our software.

9. Our software has not been tested by Alpha text, also named informal acceptance test

6.1.1 Further improvement

1. We still have a lot of room for improvement in language selection. After lots of steps screening, we chose JavaScript. We do not think it is the optimal solution, although it is a result considering the balance of code difficulty, package size, and some other factors. But from the aspect of the functionality and software instead of development difficulty and the burden of the customers, we would choose multithreading JavaScript or other advanced languages.

2. Because the softwares loading time will become longer as the number of the particle increasing, a progress bar can let our customers see its loading process intuitively.

3. If the parameters have been modified, the software would generate a new image according to the new data. Sometimes customers may need both the old image according to the old data and the new image with its data to contrast. If there is a contrast window which could contrast the data and images between old and new, the software will give customers a more intuitive experience

6.2 Assessment of the process

Our teammates cooperate better and become more efficient after nearly one year's effort and adjustment. It is no doubt that the accomplishment of the project is a result of our concerted effort. From the perplexed start to the efficient cooperation later, we experienced a lot. During this period, there is something rewarding and also something that needs improvement. The followings are the assessments of our project.

1. The five members of the process united in a concerted effort, maintained a high degree of activity and attendance

2. Team members communicated frequently and contacted our supervisor when we met problems, and we were all hard-working and willing to work.

3. At first, all the members were learning about the basic theory of the Particle filter. We waste too many too much time on the foundation of mathematics and it is not necessary.

4. The progress of software design was slowed down because in the initial stage we spend a few weeks learning math basic together in low efficiency.

5. We divided our group into two teams, one is responsible for algorithm implementation of Particle filter, and the other is responsible for software design. This had greatly speeded up our process.
6. Our project nearly had no progress during the Chinese Spring Festival holiday. This affected the subsequent progress
7. Though we communicated with our supervisor positively and frequently, the math problem of Particle filter is still hard for year two student majored in computer science, which is responsible to the incomplete function implementation of our final software.

6.2.1 Further improvement

At first few weeks, all the meetings with our supervisor were like math lectures, every member of the group wanted to understand what Particle filter is and its mathematical principles, that was meaningless. If we divided our team in an inchoate time and have a clear distribution of responsibilities, we would save a lot of time and catch up progress.

6.3 Assessment of the technique

6.4 Summary

Although we encountered many obstacles in the development process, due to the lack of experience, we took many detours, but we finally made meaningful achievements and gained valuable experience. In this project, we improved our understanding of algorithms, learned new mathematical knowledge, and tried various code languages and frameworks. From the very beginning, we didn't know what Monte Carlo was, until we could transform the particle filter into the algorithm that we were familiar with. And then we made the software which completes data visualization. Most important of all, we understand the importance of a team to software engineering. Through the constant grinding, joint efforts and collaboration among our team members, we've come to realize how powerful a team can be in this project. We know that the software has many shortcomings and needs improvement, but we are still proud of our software. This is the fruit of our team. We are going to express gratitude to our supervisor Dr. Liang and our tutor Dave s strong support and help for our project. Their rich experience and deep understanding on algorithm, schedule management and software development are undoubtedly the key to our project. Without their help, we cannot succeed.

Appendix A Reference

Thomas B., S., Fredrik, L. (August 23, 2017). Learning of dynamical systems-Particle filters and Markov chain methods, 129. Retrieved from
Thomas B. Schon (2006). Estimation of Nonlinear Dynamic Systems-Theory and Application. Department of Electrical Engineering Linkopings Universtiy, Sweden

Appendix B Forward Filtering

As we discussed above, the filtering problem estimates the state X_t given the information present in the measuremets $Y_{1:t}$ by computing the filtering PDF $p(x_t|y_{1:t})$. It can be obtained from the Bayesian theorem :

$$p(x_t|y_{1:t}) = p(x_t|y_t, y_{1:t-1}) = \frac{p(y_t|x_t, y_{1:t-1})p(x_t|y_{1:t-1})}{p(y_t|y_{1:t-1})} \tag{22}$$

Because the measuremets from an Space State Model are conditionally independent, therefore,

$$p(x_t|y_{1:t}) = \frac{p(y_t|x_t)p(x_t|y_{1:t-1})}{p(y_t|y_{1:t-1})} \tag{23}$$

The equation(23) requires one step prediction PDF $p(x_t|y_{1:t-1})$, which we are likely to find based on objects we have. Let's consider the joint PDF $p(x_t, x_{t-1}|y_{1:t-1})$, which in other form is as,

$$p(x_t, x_{t-1}|y_{1:t-1}) = p(x_t|x_{t-1}, y_{1:t-1})p(x_{t-1}|y_{1:t-1})$$

(24)

Based on the Markov property, we get

$$p(x_t, x_{t-1}|y_{1:t-1}) = p(x_t|x_{t-1})p(x_{t-1}|y_{1:t-1})$$

(25)

The prediction PDF $p(x_t|y_{1:t-1})$ can be obtained by marginalizing equaiton(25) w.r.t. X_{t-1},

$$p(x_t|y_{1:t-1}) = \int p(x_t, x_{t-1}|y_{1:t-1})dx_{t-1} = \int p(x_t|x_{t-1})p(x_{t-1}|y_{1:t-1})dx_{t-1} \tag{26}$$

Summarizing the above analysis we get

$$p(x_t|y_{1:t}) = \frac{p(y_t|x_t)p(x_t|y_{1:t-1})}{p(y_t|y_{1:t-1})} \tag{27}$$

$$p(x_t|y_{1:t-1}) = \int p(x_t|x_{t-1})p(x_{t-1}|y_{1:t-1})d_{x_{t-1}} \tag{28}$$

Appendix C User-Manual

Overview:
Bootstrap Particle Filter Simulator is an interactive software that implements the data visualization to realize Bootstrap Particle Filter Algorithm.

C.1 Functions

C.1.1 Basic function

1. There are four parameters controller whose value will be passed to the underlying algorithm can be adjusted. Users can drag the SeekBar to adjust them.

2. When all of the parameters have been set already, User can click the start button to initialize the computation. The software will compute the algorithm on the Background.

3. Once the computation is done, the image of plotting the algorithm will present on the left side of the window.

4. There is a default image presenting on the initial window before any users manipulation.

5. The image of plotting differs from time to time, since the algorithm runs differently each time.

C.1.2 I/O Functions

1. The software provides the function to import or export data that set on the File menu. Users can use the export function to store the data of the presenting image and parameters.

2. Users are able to import data out of the software. When the process of import is done, the refresh button has to be clicked to plot a new image on the left side.

3. The format of export and import file is in .JSON, which contains the matrix result from the algorithm and the parameters infromation.

C.1.3 Advanced Functions

1. Users are able to view the specific part of the image along the x or y axis by manipulate the zoom bars. They just need to click "Area zooming", and then drag the mouse to the specific region they want to look at.

2. Users are able to click "Always on Top" in view menu for keeping the software window setting on the top.

3. Restore area zooming provides the users with restoring the image back to the normal image without any zomming effect by clicking "Restore area zooming".
4. The button "Restore" will eliminate all the special effect.
5. The button "Save as image" allows users to save the image that will be exported as the PNG format.
6. The button "Data view" allows users to view the concrete data of the specific point on the plotting line.

For more information, please visit our website:
https://github.com/HenryJaQiu/GRP-17-18-Group6

Appendix D Testing Table

ID	Category	Aim	Test steps	Expected result	Actual result	Pass/ Fail
1	Code correctness	Check the correctness of randn function in Algorithm.js	1. Execute randn function by variety parameter.	Jbtest shows ans = 0	As expected	Pass
			2. Output every data in returned matrix mat.			
			3. Input these data into Matlab as a matrix.			
			4. Execute jbtest in Matlab.			
2	Code correctness	Check the correctness of other function in Algorithm.js	1. Insert a line of code to output the return value to console.	Output data is the same as expected	As expected	Pass
			2. Give actual value of parameters and call the function.			
3	Install	Check if software can work after installation	1. Double click installation package and install the software.	1. Install successfully.	As expected	Pass
			2. Open PF simulator.	2. Software can work well.		

4	System compatibility	check if software is supported by Windows and MacOs	1. Install the software on Windows and MacOs. 2. Open software check the UI and functions.	1. Software can be installed on different systems. 2. UI and functions work correctly.	As expected	Pass
5	Functions	Software is able to implement PF algorithm and show result with a chart	Click start button to running the algorithm by presetting data.	1. Two lines in the chart. 2. There are different charts after clicking start.	As expected	Pass
6		Changing Particles value in Parameter Setting will change data running in algorithm	1. Insert console.log(data) in source code to show the number of particles. 2. Change particles value to 200 and click Start in software.	Data showing in console is 200.	As expected	Pass
7		Changing Initial Noise Covariance value in Parameter Setting will change data running in algorithm	1. Insert console.log(data) in source code to show the value of Initial Noise Covariance. 2. Change Initial Noise Covariance value to 50 and click Start in software.	Data showing in console is 50.	As expected	Pass

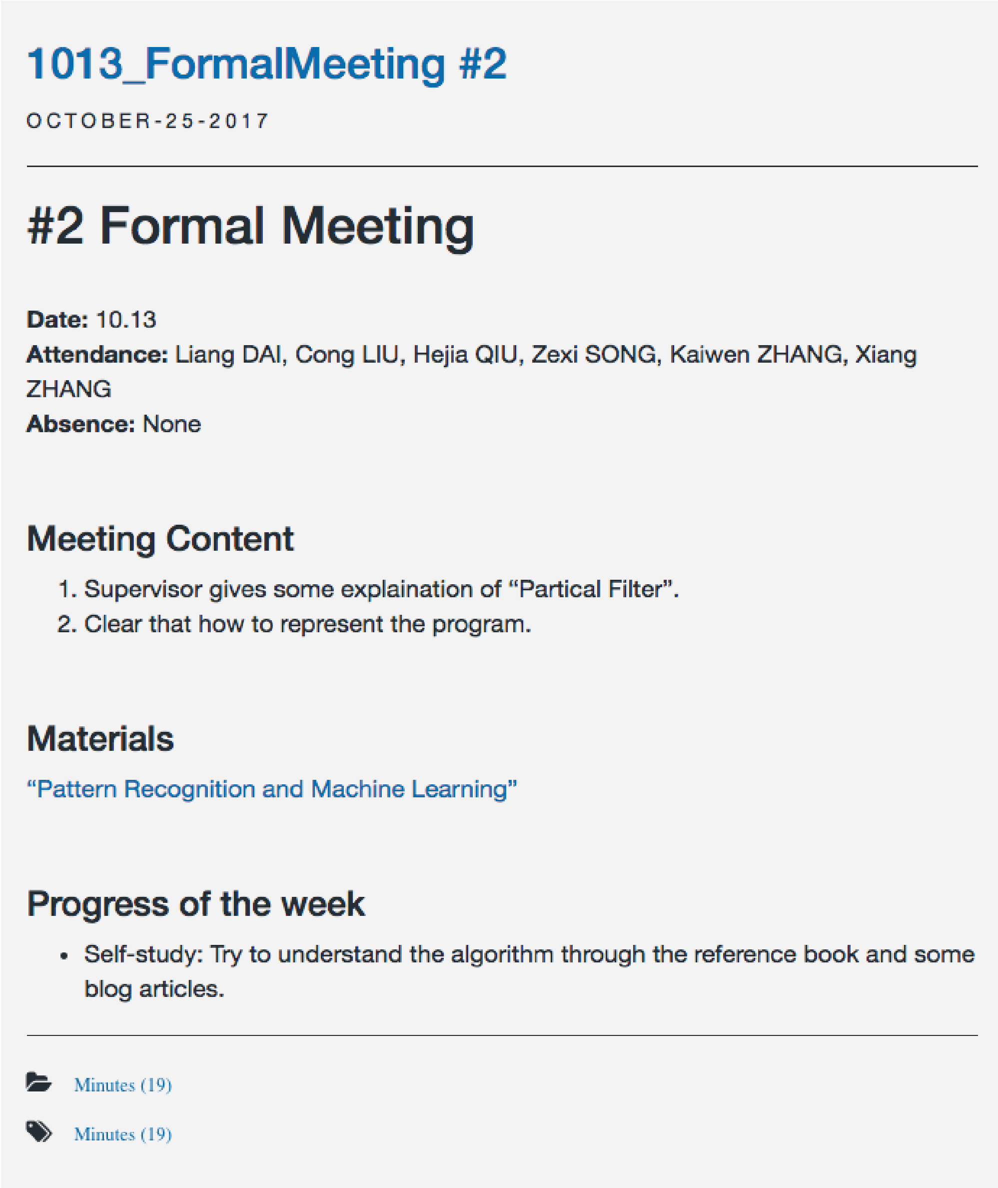
8		Changing Process Noise Covariance value in Parameter Setting will change data running in algorithm	1. Insert console.log(data) in source code to show the value of Process Noise Covariance. 2. Change Process Noise Covariance value to 20 and click Start in software.	Data showing in console is 20.	As expected	Pass
9		Changing Measurement Noise Covariance value in Parameter Setting will change data running in algorithm	1. Insert console.log(data) in source code to show the value of Measurement Noise Covariance. 2. Change Measurement Noise Covariance value to 30 and click Start in software.	Data showing in console is 30.	As expected	Pass
10		Save chart	Click the button Save as on the top of chart.	Chart can be saved as a png file in local.	As expected	Pass
11		Export data	Click Export data in menu File.	Related data will be save as a json file.	As expected	Pass
12		Import a json file	1. Click Import a json file in menu File. 2. Click Refresh.	Chart will be drawn by given data in json file.	As expected	Pass

13		Software window always on top	1. Click Always on Top in menu View. 2. Open other windows in the screen.	Particle Filter Simulator will always on the top and not be covered by other windows.	As expected	Pass
14		Show Help window	Click Help in menu Help	A new window is created and show help document.	As expected	Pass
15		Show About message dialog	Click About in menu Help	A message dialog shows including brief introduction and version information	As expected	Pass
16		Quit the software	Click Quit in menu bar	Quit the software	Under MacOs system, window is closed but not quit.	Fail
17	Performance	Test the response time of drawing chart with default data	1. Open chrome-devtools and choose Performance to start record. 2. During recording, click Start button and waiting the chart. 3. Repeat step 2 for 5 times.	3The response time is less than 5s, otherwise there should be a prompt.	3Average time is 1530ms.	3Pass

18		Test the response time of drawing chart with max data	1. Set all parameter values to maximum. 2. Open chrome-devtools and choose Performance to start record. 3. During recording, click Start button and waiting the chart. 4. Repeat step 2 for 5 times	The response time is less than 5s, otherwise there should be a prompt.	Average time is 7963.6ms and no prompt.	Fail
19	Uninstallation	Uninstall the software	Uninstall	No components remain and no effect to system	As expected	Pass

Appendix E Formal Meeting Timeline

1. The first formal meeting on 2017.10.13



1013_FormalMeeting #2

OCTOBER-25-2017

#2 Formal Meeting

Date: 10.13
Attendance: Liang DAI, Cong LIU, Hejia QIU, Zexi SONG, Kaiwen ZHANG, Xiang ZHANG
Absence: None

Meeting Content

1. Supervisor gives some explaination of “Partical Filter”.
2. Clear that how to represent the program.

Materials

“Pattern Recognition and Machine Learning”

Progress of the week

- Self-study: Try to understand the algorithm through the reference book and some blog articles.

Minutes (19)

Minutes (19)

Figure 15:

2. The second formal meeting on 2017.10.18

1018_FormalMeeting #3

OCTOBER-25-2017

#3 Formal Meeting

Date: 10.18
Attendance: Liang DAI, Cong LIU, Hejia QIU, Zexi SONG, Kaiwen ZHANG, Xiang ZHANG
Absence: None

Meeting Content

1. Liang shows the Mathematical inference process of Kalman Filter & Particle Filter. View details

Materials

- "Pattern Recognition and Machine Learning"
- "schon2006_thesis"

Progress of the week

- To understand the algorithm through the reference book and some blog articles.
- To understand/run PF algorithm code on matlab.

Minutes (19)

Minutes (19)

Figure 16:

3. The third formal meeting on 2017.11.14

1114_FormalMeeting #4

NOVEMBER-14-2017

#4 Formal Meeting

Date: 11.14
Attendance: Liang DAI, Cong LIU, Hejia QIU, Zexi SONG, Kaiwen ZHANG, Xiang ZHANG
Absence: None

Meeting Content

1. Solve some PF algorithm problem.
2. Divide report into two parts.
    - **algorithm**: focus on figure out the algorithm issue including math & matlab code etc. And finish the related part of report.
    - **software Design**: focus on the design of software including prototype and user requirments. And finish the related part of report.

Materials

None

Progress of the week

- Learn to use LaTex.
- To understand PF algorithm continuous.

Minutes (19)

Minutes (19)

Figure 17:

4. The fourth formal meeting on 2017.11.23

1123_FormalMeeting #5

NOVEMBER-23-2017

#5 Formal Meeting

Date: 11.23
Attendance: Liang DAI, Cong LIU, Hejia QIU, Zexi SONG, Xiang ZHANG
Absence: Kaiwen ZHANG

Meeting Content

1. Solve some PF matlab code problem.
2. Communicate the progress of two groups.
3. Determine the work should be finished before 12.1.
   - **algorithm**: finish algorithm-related part of report: intro.,application, adjustable arguments etc.
   - **software Design**: finish the design-related part of report: software intro.,User interface design, user requirements etc.

Materials

None

Progress of the week

- finish simple prototype design, parts of software instruction.
- To understand PF algorithm continuous.

Minutes (19)

Minutes (19)

Figure 18:

5. The fifth formal meeting on 2018.3.13

0313_FormalMeeting #11

MARCH-13-2018

#11 Formal Meeting

Date: 3.13

Attendance: Hejia QIU, Zexi SONG, Cong LIU

Absence: Xiang ZHANG, Kaiwen ZHANG

Meeting Content

1. Show the last word of vue-cli, electron-vue and electron-builder.
2. Changes: Original setting is re-load images after the data, parameter or operator changes but this cost a lot of source. So we decide to use the two-way binding function of echarts and vue in order to auto update instead of re-load every time data changes. This method could decrease cost and increase software efficiency and fluency.

Materials

https://molunerfinn.com/electron-vue-1/#%E5%89%8D%E8%A8%80

Progress of the week

Minutes (19)

Minutes (19)

Figure 19:

6. The sixth formal meeting on 2018.3.21

0321_FormalMeeting #14

MARCH-21-2018

#14 Formal Meeting

Date: 3.21

Attendance: Hejia QIU, Zexi SONG, Cong LIU， Kaiwen ZHANG

Absence: Xiang ZHANG

Meeting Content

1. Show our latest code and algorithm to Dr.LIANG.
2. Debug some bugs of the code with Dr.LIANG's help.

Materials

None

Progress of the week

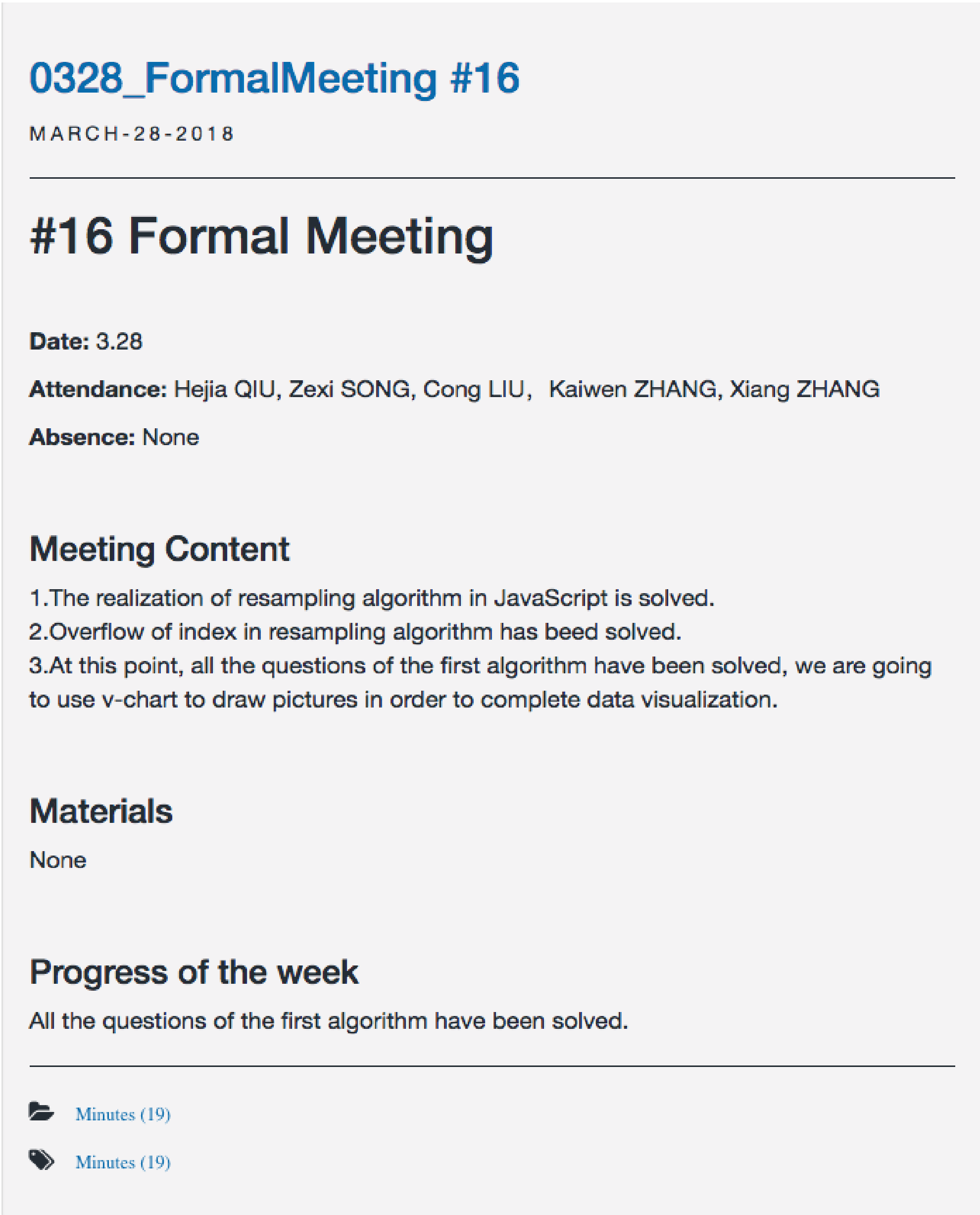
Realize repetition algorithm for some parts of Matlab code with Python and Javascript.

Minutes (19)

Minutes (19)

Figure 20:

7. The seventh formal meeting on 2017.3.28



0328_FormalMeeting #16

MARCH-28-2018

#16 Formal Meeting

Date: 3.28

Attendance: Hejia QIU, Zexi SONG, Cong LIU, Kaiwen ZHANG, Xiang ZHANG

Absence: None

Meeting Content

1.The realization of resampling algorithm in JavaScript is solved.
2.Overflow of index in resampling algorithm has beed solved.
3.At this point, all the questions of the first algorithm have been solved, we are going to use v-chart to draw pictures in order to complete data visualization.

Materials

None

Progress of the week

All the questions of the first algorithm have been solved.

Minutes (19)

Minutes (19)

Figure 21:

8. The eighth formal meeting on 2018.4.4

0404_FormalMeeting #19

APRIL-4-2018

#19 Formal Meeting

Date: 4.4

Attendance: Hejia QIU, Cong LIU, Kaiwen ZHANG, Xiang ZHANG

Absence: Zexi SONG

Meeting Content

1.Solved a problem related to data formate. After checking, normal array format is DenseMatrix, but the imported one is an object, which will result in a failure of presenting the image because of wrong format.
2.Use bootstrap-vue as css editing library
3.Determine the division of the final report.
Cong LIU:Introduction.
Hejia QIU:Design,implement and strategy
Kaiwen ZHANG:Requirement
Xiang ZHANG:Text
Zexi SONG:Evaluation and reflection

Materials

None

Progress of the week

Determine the division of the final report.

Minutes (19)

Minutes (19)

Figure 22:

9. The ninth formal meeting on 2018.4.13

Formal Meeting 20

APRIL-13-2018

---

layout: post title: “0413_FormalMeeting #20” date: 2018-4-13 19:09:55 +0800 categories: Minutes tags: Minutes description: This is the minute of NO.20 formal meeting.

#20 Formal Meeting

Date: 4.13

Attendance: Hejia QIU, Cong LIU, Kaiwen ZHANG, Xiang ZHANG, Zexi SONG

Absence: None

Meeting Content

1.When the number of particle is 100,the response time is 1.7s. But if the number of particle grows to 500, it needs 8s to response, we decide to design something to prompt the customer that “Please waiting for a while”.
2.Solved normal distribution problem by taking uniform distribution random numbers and converting them into normal distribution by box-muller, and tested the result by normality test.
3.The efficiency of the algorithm is proposed by the supervisor.
4.Some questions are answered by the supervisor for the idea that we want to add an algorithm.
5.We report the progress and detailed division of labor so far.

Materials

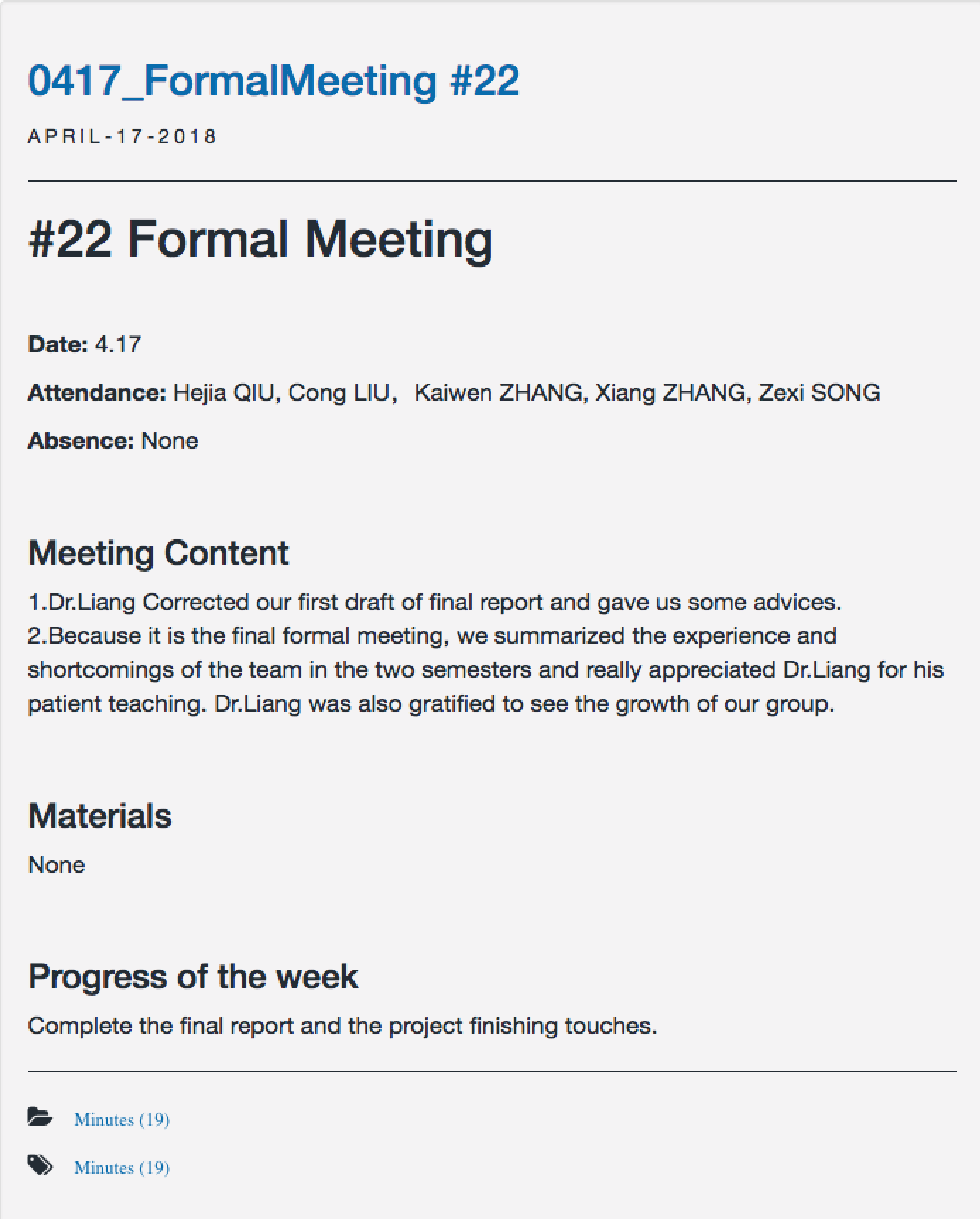
None

Progress of the week

Improved algorithms and software code.

Figure 23:

10. The tenth formal meeting on 2018.4.17



0417_FormalMeeting #22

APRIL-17-2018

#22 Formal Meeting

Date: 4.17

Attendance: Hejia QIU, Cong LIU， Kaiwen ZHANG, Xiang ZHANG, Zexi SONG

Absence: None

Meeting Content

1.Dr.Liang Corrected our first draft of final report and gave us some advices.
2.Because it is the final formal meeting, we summarized the experience and shortcomings of the team in the two semesters and really appreciated Dr.Liang for his patient teaching. Dr.Liang was also gratified to see the growth of our group.

Materials

None

Progress of the week

Complete the final report and the project finishing touches.

Minutes (19)

Minutes (19)

Figure 24: