UNIVERSITÄT DUISBURG-ESSEN

BACHELOR THESIS

Development and Comparison of Overview Techniques for Extreme Resolution Datasets

Author:
Danyun LEI

Supervisor:
Prof. Dr. Jens KRÜGER

Examiners:
Prof. Dr. Jens KRÜGER
Prof. Dr. Josef PAULI

A thesis submitted in fulfillment of the requirements
for the degree of Bachelor of Science

in

Computer Engineering
International Studies in Engineering (ISE) PO08
Fakultät für Ingenieurwissenschaften

for

The High Performance Computing Group
Department Engineering

September 11, 2019

Versicherung an Eides Statt

Ich, Danyun LEI, versichere an Eides statt durch meine untenstehende Unterschrift,

- dass ich die vorliegende Arbeit - mit Ausnahme der Anleitung durch die Betreuer - selbstständig ohne fremde Hilfe angefertigt habe und
- dass ich alle Stellen, die wörtlich oder annähernd wörtlich aus fremden Quellen entnommen sind, entsprechend als Zitate gekennzeichnet habe und
- dass ich ausschließlich die angegebenen Quellen (Literatur, Internetseiten, sonstige Hilfsmittel) verwendet habe und
- dass ich alle entsprechenden Angaben nach bestem Wissen und Gewissen vorgenommen habe, dass sie der Wahrheit entsprechen und dass ich nichts verschwiegen habe.

Mir ist bekannt, dass eine falsche Versicherung an Eides Statt nach §156 und nach §163 Abs. 1 des Strafgesetzbuches mit Freiheitsstrafe oder Geldstrafe bestraft wird.

Ort, Datum

Unterschrift

"To see a world in a grain of sand, hold infinity in the palm of your hand. "

William Blake

UNIVERSITÄT DUISBURG-ESSEN

Abstract

Fakultät für Ingenieurwissenschaften

International Studies in Engineering (ISE) PO08

Computer Engineering

Bachelor of Science

Development and Comparison of Overview Techniques for Extreme Resolution Datasets

by Danyun LEI

In this thesis, three main overview techniques and two secondary preview techniques of the overviews are developed and implemented using the Mandelbrot set as the source of extreme resolution datasets. For these datasets, a hierarchical structure is used to present the index information of the current region of interest. Using the developed software in this thesis, it is possible to intuitively understand the hierarchical state of the current observing area with the whole. This thesis also compares the different overview techniques which in combination consists of six different ways.

It is worth mentioning that the Mandelbrot set that are implemented in this thesis is a very good example for extreme high resolution datasets because it is a dataset that can theoretically provide infinitely high resolution.

The technology stack used in this thesis is pure web technology, a classic combination of `HTML` / `JavaScript` / `CSS`, and the program offers the possibility to adapt and replace the pure front-end technology with front and back end separation solution easily.

Acknowledgements

I hereby express my deep sense of gratitude and indebtedness to Prof. Dr. Jens Krüger, for your valuable guidance, encouragement and support. Your patience and faith in me was a key reason that I could finish the work.

Also I express my thanks to Prof. Dr. Yunqi Lei, Dr. Bixia Wu and Dr. Franz-Josef Schmitz, in no particular order. Without your support, I could not have finished the task successfully.

Contents

List of Figures

List of Tables

List of Abbreviations

HTML	**H**ypertext **M**arkup **L**anguage
JS	**J**ava**S**cript
CSS	**C**ascading **S**tyle **S**heets
DOM	**D**ocument **O**bject **M**odel
URL	**U**niform **R**esource **L**ocator
JSON	**J**ava **S**cript **O**bject **N**otation
UI	**U**ser **I**nterface

Physical Constants

Speed of Light	$c_0 = 2.997\,924\,58 \times 10^8\,\mathrm{m\,s^{-1}}$ (exact)

List of Symbols

a	distance	m
P	power	W (J s^{-1})
ω	angular frequency	rad

Special thanks to Dr. Zhonghua Xu, Ms. Meng Wang and Mrs. Vivian E. Rice, I wish I could share the joy of this achievements with all of you - here or in Azeroth.

Chapter 1

Introduction and Objectives

1.1 Main Section 1

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam ultricies lacinia euismod. Nam tempus risus in dolor rhoncus in interdum enim tincidunt. Donec vel nunc neque. In condimentum ullamcorper quam non consequat. Fusce sagittis tempor feugiat. Fusce magna erat, molestie eu convallis ut, tempus sed arcu. Quisque molestie, ante a tincidunt ullamcorper, sapien enim dignissim lacus, in semper nibh erat lobortis purus. Integer dapibus ligula ac risus convallis pellentesque.

1.1.1 Subsection 1

Nunc posuere quam at lectus tristique eu ultrices augue venenatis. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Aliquam erat volutpat. Vivamus sodales tortor eget quam adipiscing in vulputate ante ullamcorper. Sed eros ante, lacinia et sollicitudin et, aliquam sit amet augue. In hac habitasse platea dictumst.

1.1.2 Subsection 2

Morbi rutrum odio eget arcu adipiscing sodales. Aenean et purus a est pulvinar pellentesque. Cras in elit neque, quis varius elit. Phasellus fringilla, nibh eu tempus venenatis, dolor elit posuere quam, quis adipiscing urna leo nec orci. Sed nec nulla auctor odio aliquet consequat. Ut nec nulla in ante ullamcorper aliquam at sed dolor. Phasellus fermentum magna in augue gravida cursus. Cras sed pretium lorem. Pellentesque eget ornare odio. Proin accumsan, massa viverra cursus pharetra, ipsum nisi lobortis velit, a malesuada dolor lorem eu neque.

1.2 Main Section 2

Sed ullamcorper quam eu nisl interdum at interdum enim egestas. Aliquam placerat justo sed lectus lobortis ut porta nisl porttitor. Vestibulum mi dolor, lacinia molestie gravida at, tempus vitae ligula. Donec eget quam sapien, in viverra eros. Donec pellentesque justo a massa fringilla non vestibulum metus vestibulum. Vestibulum in orci quis felis tempor lacinia. Vivamus ornare ultrices facilisis. Ut hendrerit volutpat vulputate. Morbi condimentum venenatis augue, id porta ipsum vulputate in.

Curabitur luctus tempus justo. Vestibulum risus lectus, adipiscing nec condimentum quis, condimentum nec nisl. Aliquam dictum sagittis velit sed iaculis. Morbi tristique augue sit amet nulla pulvinar id facilisis ligula mollis. Nam elit libero, tincidunt ut aliquam at, molestie in quam. Aenean rhoncus vehicula hendrerit.

Chapter 2

Background

2.1 Main Section 1

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam ultricies lacinia euismod. Nam tempus risus in dolor rhoncus in interdum enim tincidunt. Donec vel nunc neque. In condimentum ullamcorper quam non consequat. Fusce sagittis tempor feugiat. Fusce magna erat, molestie eu convallis ut, tempus sed arcu. Quisque molestie, ante a tincidunt ullamcorper, sapien enim dignissim lacus, in semper nibh erat lobortis purus. Integer dapibus ligula ac risus convallis pellentesque.

2.1.1 Subsection 1

Nunc posuere quam at lectus tristique eu ultrices augue venenatis. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Aliquam erat volutpat. Vivamus sodales tortor eget quam adipiscing in vulputate ante ullamcorper. Sed eros ante, lacinia et sollicitudin et, aliquam sit amet augue. In hac habitasse platea dictumst.

2.1.2 Subsection 2

Morbi rutrum odio eget arcu adipiscing sodales. Aenean et purus a est pulvinar pellentesque. Cras in elit neque, quis varius elit. Phasellus fringilla, nibh eu tempus venenatis, dolor elit posuere quam, quis adipiscing urna leo nec orci. Sed nec nulla auctor odio aliquet consequat. Ut nec nulla in ante ullamcorper aliquam at sed dolor. Phasellus fermentum magna in augue gravida cursus. Cras sed pretium lorem. Pellentesque eget ornare odio. Proin accumsan, massa viverra cursus pharetra, ipsum nisi lobortis velit, a malesuada dolor lorem eu neque.

2.2 Main Section 2

Sed ullamcorper quam eu nisl interdum at interdum enim egestas. Aliquam placerat justo sed lectus lobortis ut porta nisl porttitor. Vestibulum mi dolor, lacinia molestie gravida at, tempus vitae ligula. Donec eget quam sapien, in viverra eros. Donec pellentesque justo a massa fringilla non vestibulum metus vestibulum. Vestibulum in orci quis felis tempor lacinia. Vivamus ornare ultrices facilisis. Ut hendrerit volutpat vulputate. Morbi condimentum venenatis augue, id porta ipsum vulputate in.

Curabitur luctus tempus justo. Vestibulum risus lectus, adipiscing nec condimentum quis, condimentum nec nisl. Aliquam dictum sagittis velit sed iaculis. Morbi tristique augue sit amet nulla pulvinar id facilisis ligula mollis. Nam elit libero, tincidunt ut aliquam at, molestie in quam. Aenean rhoncus vehicula hendrerit.

Chapter 3

Requirements and Concepts

3.1 Main Section 1

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam ultricies lacinia euismod. Nam tempus risus in dolor rhoncus in interdum enim tincidunt. Donec vel nunc neque. In condimentum ullamcorper quam non consequat. Fusce sagittis tempor feugiat. Fusce magna erat, molestie eu convallis ut, tempus sed arcu. Quisque molestie, ante a tincidunt ullamcorper, sapien enim dignissim lacus, in semper nibh erat lobortis purus. Integer dapibus ligula ac risus convallis pellentesque.

3.1.1 Subsection 1

Nunc posuere quam at lectus tristique eu ultrices augue venenatis. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Aliquam erat volutpat. Vivamus sodales tortor eget quam adipiscing in vulputate ante ullamcorper. Sed eros ante, lacinia et sollicitudin et, aliquam sit amet augue. In hac habitasse platea dictumst.

3.1.2 Subsection 2

Morbi rutrum odio eget arcu adipiscing sodales. Aenean et purus a est pulvinar pellentesque. Cras in elit neque, quis varius elit. Phasellus fringilla, nibh eu tempus venenatis, dolor elit posuere quam, quis adipiscing urna leo nec orci. Sed nec nulla auctor odio aliquet consequat. Ut nec nulla in ante ullamcorper aliquam at sed dolor. Phasellus fermentum magna in augue gravida cursus. Cras sed pretium lorem. Pellentesque eget ornare odio. Proin accumsan, massa viverra cursus pharetra, ipsum nisi lobortis velit, a malesuada dolor lorem eu neque.

3.2 Main Section 2

Sed ullamcorper quam eu nisl interdum at interdum enim egestas. Aliquam placerat justo sed lectus lobortis ut porta nisl porttitor. Vestibulum mi dolor, lacinia molestie gravida at, tempus vitae ligula. Donec eget quam sapien, in viverra eros. Donec pellentesque justo a massa fringilla non vestibulum metus vestibulum. Vestibulum in orci quis felis tempor lacinia. Vivamus ornare ultrices facilisis. Ut hendrerit volutpat vulputate. Morbi condimentum venenatis augue, id porta ipsum vulputate in.

Curabitur luctus tempus justo. Vestibulum risus lectus, adipiscing nec condimentum quis, condimentum nec nisl. Aliquam dictum sagittis velit sed iaculis. Morbi tristique augue sit amet nulla pulvinar id facilisis ligula mollis. Nam elit libero, tincidunt ut aliquam at, molestie in quam. Aenean rhoncus vehicula hendrerit.

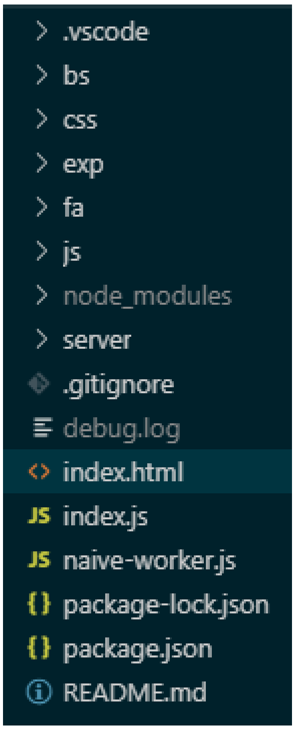
Chapter 4

Implementation

In this chapter, the overall structure of the project, how the files are arranged and the functionalities of each components, will be described in details.

4.1 Files And Folders

The folder names and file names are mostly self-explanatory or conventional in this project. They'll be described briefly in this section.



```
> .vscode
> bs
> css
> exp
> fa
> js
> node_modules
> server
  .gitignore
  debug.log
  index.html
  index.js
  naive-worker.js
  package-lock.json
  package.json
  README.md
```

FIGURE 4.1: A glimpse of files and folders.

4.1.1 Folders

Folder `./.vscode` The configured Visual Studio Code workspace settings file. This file is included and stored inside the workspace and only apply when the workspace is opened which overrides Visual Studio Code's default user settings. The author tweaked this file to make some parts of VS Code's editor, user interface, and functional behavior more fitting to review or to base future work upon this project.

```
// https://code.visualstudio.com/docs/getstarted/settings
```

Folder `./js` All the third-party open source JS dependencies are stored in this folder. Sometimes third-party open source projects include a bundle of JavaScript and CSS files, here only the pure JS projects' files are included.

Folder `./css` The CSS files of the projects are included. Firstly there is a `./css/common.css` file, which sets the overall styles of the project, basically whatever the users can see at the very first glance when they open this project. Then there are several other CSS files, each sets a specific portion of the styles in this project. These files include:

- CSS File `./css/dock.css` sets the iOS-Dock look-like styles, making the focused item larger with larger margins and adjacent items smaller and smaller margins with their corresponding nearby items.
- CSS File `./css/minibar.css` sets the customed scrollbar styles that's being added upon the default styles of the dependency *MiniBar* which is used to create custom scrollbars.
- CSS File `./css/stacked.css` sets the styles of the stacked cards effect.
- CSS File `./css/tabs.css` sets the related styles of the tabs effect.

Note that most of the effects require not only the CSS stylings but also JS actions in order to work.

Folder `./fa` Assets of the dependency *Font Awesome*, including all resources of the open source part. This dependency is used for the fonts of the icons in this project.

Folder `./bs` Assets of the open source project *Bootstrap* by *Twitter*. This dependency is used for the stylings of the web elements inside the control panel, such as input boxes, dropdown menus and font styles in control panel. It also comes with some nice utilities for general web elements style setting.

Folder `./node_modules` Packages pulled from the JS dependency management tool *npm* are stored in this folder. The required dependency here is the package `minibarjs` under this folder – in folder `./node_modules/minibarjs`. Conventionally this folder shouldn't be included or committed to the version control system[1], because all the packages info are recorded in the file `package.json` and `package-lock.json` and if any dependencies are missing, running the *npm* command `npm install` should be able to pull all necessary dependencies into this folder, however, considering this project sometimes can be run in an environment without internet connection, this folder is included in the final static zipped package.

[1] This is actually also what this project is following.

```
// https://www.npmjs.com/
```

Folder `./exp` Some trivial *Python*, JS and HTML codes left from the prototypes of implementation at the beginning of this project. Some of them are using different algorithms and different scripts trying to achieve similar results to this project. They are not in use anymore and only kept for future references.

4.1.2 Top Level Files

File `index.html` This entry HTML file of this project. When a server is being run on the local machine, this is the first file getting executed. When a different implementation of the back end using techniques other than a web worker, for example a `WebSocket`, is developed and being adapted to this project, double-clicking on this file should also start this project.

File `index.js` The main JS script file of the project. This file gets included at the very end of the HTML file `index.html`.

File `naive-worker.js` The back end calculation JS script. The only job of this script is to receive information of the image the front end is asking for, and post the result message back to the front end. This piece of scripts not only post the complete results back, but also slices of results when the calculation takes longer than a certain amount of time and let the front end decide what to do with the partial results[2].

File `package.json` A description file of the JS package management tool *npm*. This file can have many descriptions about what *npm* should do for this workspace but here it most importantly specifies which packages to pull from the global repository.

```
// Description of npm - package.json
// https://docs.npmjs.com/files/package.json
```

File `package-lock.json` A generated file from *npm* package manager which locks the version of the dependencies of this specific workspace. Take the current project as an example, in file `package.json` there is this part in the JSON body:

```
{
  ..
  "dependencies": {
    ..
    "minibarjs": "^0.4.0",
    ..
  },
  ..
}
```

[2] In this project, what the front end will do after receiving partial results is that it will still render the slices of images onto the canvas and high light the painted partial image with green borders.

This piece of code only specified that the version of the package `minibarjs` that we require will match all `0.x.x` releases including `0.5.x`, but will hold off on `1.x.x`. This file `package-lock.json` will "lock" the version inside current workspace to a specific version with a hashed fingerprint of the files, in the current project with a version number of `0.4.0` and a hash fingerprint `sha512-iCUE/YVWn+0ht+NV2fLBS8bAVxED/916A5i1qJ20csCrc0tXHamgpWCo7uL+23HQ0UyFPvpw1izw213vzVKkXg==`.

File `README.md` A brief introduction file for the global version control system *GitHub*. Trivial.

File `.gitignore` Version control settings file, telling which files should not be committed to *Git* system. Not relavant to the project but the version control during the development phase of this project. Trivial.

4.2 Front End

Since this project is a pure web project, the front end occupies a large portion of the codes.

4.2.1 HTML Entry `index.html`

The entry of the project is where this program gets started, in similar concept of the `main()` function in `C` or the `public static void main(String[] args)` function in `Java`. The entry point is a HTML file and as expected named `index.html`. It introduces the front end structure of the project in raw.

First part of the HTML file is the `<head>` part. In this part, the character set of this web page is defined as *UTF-8*, the size of the entire HTML document as fullscreen size, scaling not allowed and not shrinking to display its content.

```
<meta charset="utf-8">
<meta name="viewport" content="width=device-width,
    initial-scale=1, shrink-to-fit=no">
```

And then all the needed CSS files are included to end the `<head>` part. Besides the CSS files which will be described in subsection 4.2.3 CSSs For Overview Effects, the necessary CSS files from third-party open source vendors are also included, including *Bootstrap*'s CSS part, *FontAwesome* and *MiniBar* CSS assets.

The `<body>` part is the essential part of the HTML entry, which describes the structure of what users can "actually see". It begins first with three `<div>` tags for the most important three parts of this project, the container for main background canvases, the container for mini-maps, and the container for the control panel floating on the top right corner of the UI screen. The positioning, sizes and container behaviours of these `<div>`s are defined in the CSS files which are already included. Before users set any effects up, these properties mostly come from the file `./css/common.css`.

After the visual `<div>` part, several `<script>` tags come after it to include what's necessary for the essential coding part. Here firstly are the dependencies of the project,

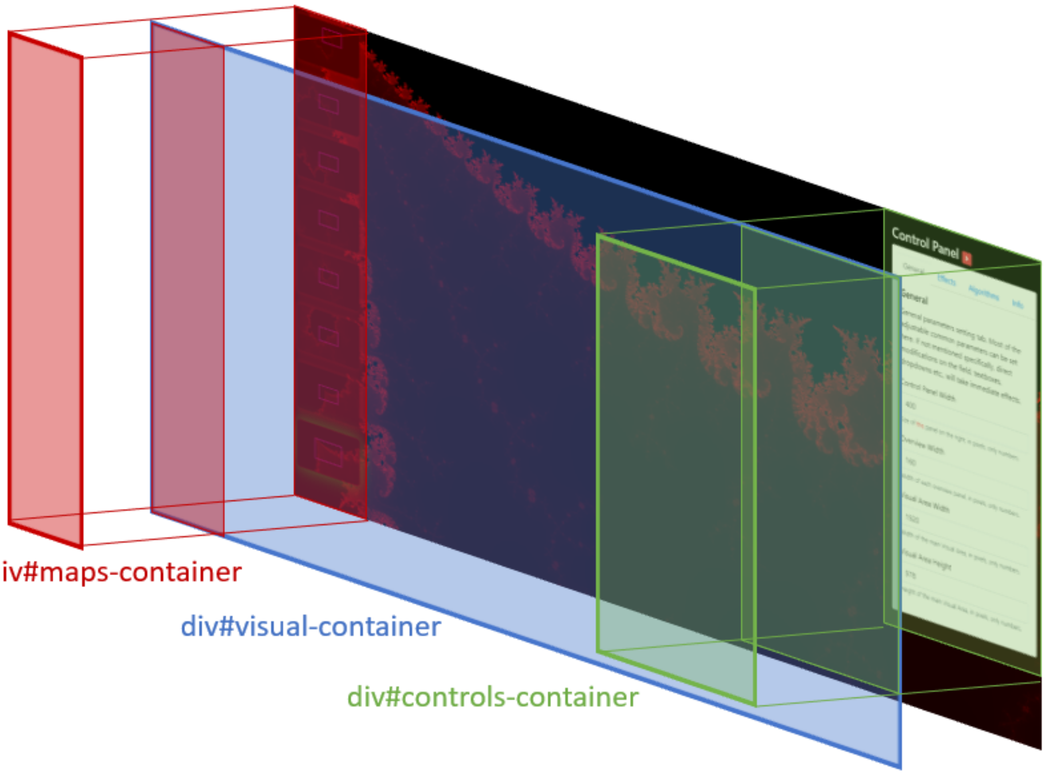


FIGURE 4.2: DOM structure in `<body>` tag.

including *jQuery*, *Bootstrap*'s JS part, and *MiniBar*'s JS part. And then at the very end the main JS file `index.js` is included and all the core programs of this project goes in there.

Worth noting that conventionally all JS files should be included at the very end of the page as what we are doing now, unless the JS file is needed before the render phase of the web page. This way if the JS file is a little bit bigger than usual, the loading of the JS files won't affect the rendering process of the DOM documents.

4.2.2 Main JavaScript `index.js`

The main JS file `index.js` is where the core codes are. In this file there is firstly the definition of required classes from bottom level to the top, then the instantiation of them and putting the front end HTML elements into action to display the overall results.

There in total four classes defined.

4.2.2.1 Class `MandelWorker`

The class `MandelWorker` is in charge of sending a message to the back end and when a result is sent back, doing some other actions.

When instantiated, an instance of a native *Web Worker*[3] will also be created as a private property of this class. `MandelWorker` instantiates the *Web Worker* by the script `naiveworker.js`, which means that the script `naiveworker.js` will be the core of the worker and this worker will be doing whatever in that script when it is asked to.

Function `work(params...)`

The function `work(params...)` is the interface between `MandelWorker` and the outside invoker. To get an image from the source, one must invoke this function with the needed parameters as follows:

- `magnif` The magnification level of the result image to be expected from the *Web Worker*.
- `centerX` The `x` component of the center coordinates on the mathmatical plane of the result image to be expected from the worker.
- `centerY` The `y` component of this coordinates.
- `width` The width in pixels of the result image.
- `height` The height in pixels of the result image.
- `callback` The function to execute when a result message is received.
- `callbackThis` The "this" context where the `callback` function should be executed under.

Here what's worth mentioning is the parameter `magnif`. The magnification level is a number representing the number of pixels that together has a length of 1 on the mathmatical axis. As shown in Figure 4.3 Magnification Level is an image with the magnification level of 2, since 2 pixels have the length of 1 on the mathmatical axis.

Once this function is invoked, `MandelWorker` will tell its own *Web Worker* to start working on datasets fetching[4], and if any sorts of results come through that worker, hit the `workerResponse(e)` function of the current `MandelWorker`.

Function `workerResponse(e)`

The function `workerResponse(e)` will be called when a response from the *Web Worker* is sent back. It basically does one thing: checking if the parameters of `callback` and `callbackThis` were set when function `work(params...)` got invoked in the first place. If they were set to any function, call it under the conext of the parameter `callbackThis`.

Function `destroy()`

The function `destroy()` as the name implies is the method to destroy and release the resources for current `MandelWorker`. It terminates the *Web Worker*, sets the response method to `null` so no responses will be dealt furthermore and sets all other references to `null` as well so the internal JS engine can garbage collect[5] all these instance to avoid memory leakage when the calculation gets heavy.

[3] Web Workers are a simple means for web content to run scripts in background threads. // https://developer.mozilla.org/en-US/docs/Web/API/Web_Workers_API/Using_web_workers

[4] In the context of the current project, is actually image generation.

[5] Include content of https://en.wikipedia.org/wiki/Garbage_collection_(computer_science)

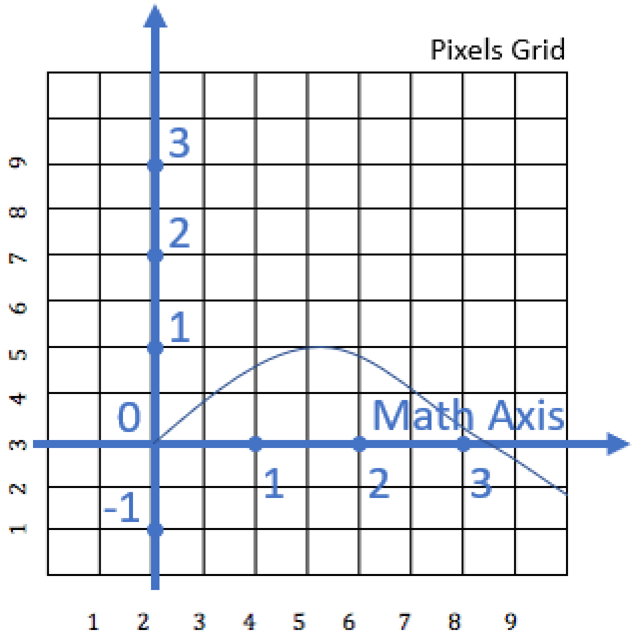


FIGURE 4.3: Magnification level in aspect of mathmatical axis.

4.2.2.2 Class `MapVisualPair`

An abstract concept of pairing a minimap[6] with an active focus region with higher resolution.

`init(mapCanvas,previewCanvas,visCanvas=null)`

`destroy()`

`drawMapHoverArea(offsetRealX=0,offsetRealY=0)`

`drawPreviewHoverArea(offsetRealX=0,offsetRealY=0)`

`moveTo(x=null,y=null)`

4.2.2.3 Class `MinimapManager`

The manager class of all the minimaps, controlling all their behaviours on the top level.

Inits

`initMaps(visCanvas,previewCanvas,mapsContainer,visualContainer,hoverX=0, hoverY=0)`

`initPairMainDrag()`

Mouse Down(drag to browse around)

`pairMainMouseDown(e)`

[6] Same concept in current project as an *overview*.

`pairMainMouseMove(e)`

`pairMainMouseUp(e)`

`pairMainStepDrag(timestamp)`

Mouse Wheel(zoom in and out)

`initPairMainWheel()`

`pairMainWheel(e)`

`pairMainTimeoutWheel()`

`pairMainStepWheel()`

4.2.2.4 Class `EffectManager`

This class manages the behaviours, activation and deactivation of the overview effects and the preview effects of the overviews.

General functions: `init()`, `getInfo(el)`

Fade in / out related functions: `fadeMouseOver(e, currentPair)`, `fadeMouseOut(e, currentPair)`

Zoom in / out through related functions: `zoomMouseOver(e, currentPair)`, `zoomStep(timestamp)`, `zoomMouseOut(e, currentPair)`

General preview effects functions: `updatePreview()`, `updateFadePreview()`, `updateZoomPreview()`, `destroyPreview()`

General overview effects functions: `destroy()`, `update()`

Scrollbar + Dock effects specific functions: `initScrollbar()`, `destroyScrollbar()`, `updateScrollbar()`

Stacked Cards effects specific functions: `initStacked()`, `destroyStacked()`, `updateStacked()`

Tabs effects specific functions: `initTabs()`, `destroyTabs()`, `updateTabs()`

4.2.2.5 Instantiation, Variables And the Rest

4.2.3 CSSs For Overview Effects

Folder `./css` includes five CSS files, each setting up some visual effects of the project.

File `./css/common.css` first sets up all general appearance of the elements on the web page when no parameters or effects are set. File `./css/dock.css` sets up the appearance when *Scrollbar* + *Dock* is activated, only the iOS Dock part and file `./css/minibar.css` sets up the scroll bar part. File `./css/stacked.css` sets up the effects of stacked cards. File `./css/tabs.css` sets up the effects of the tab selection on the top.

4.2.4 Scrollbar + Dock Effect

4.2.5 Stacked Cards Effect

4.2.6 Tabs Effect

4.3 Back End Calculation

The back end calculation is done in the JS file `naive-worker.js`. This file is being used for initializing the *WebWorker*s inside `index.js` dynamically. Whenever a calculation or extraction for a specific region of a dataset is needed, the main JS file `index.js` is going to send a message to `naive-worker.js` with desired parameters and this back end will respond with corresponding image data.

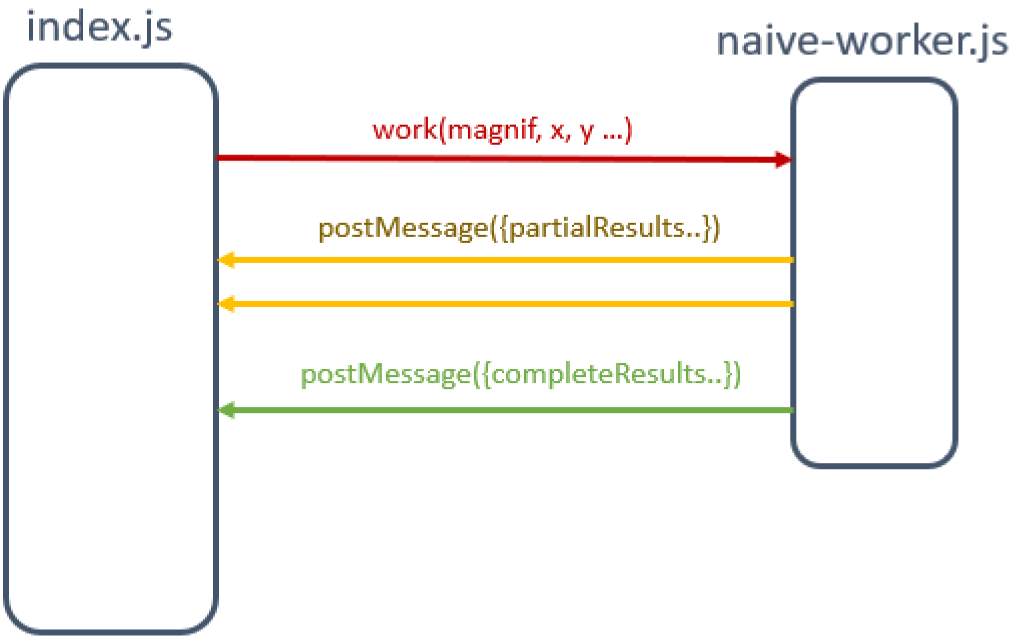

index.js
naive-worker.js
work(magnif, x, y ...)
postMessage({partialResults..})
postMessage({completeResults..})

FIGURE 4.4: Message exchange between `index.js` and `naive-worker.js`.

4.3.1 Global Scope

In the global scope of this file, the following things were done.

Includes The `decimal.js` dependency is included for high-precision floating points calculation. Default parameters for the dependency is set.

Constants Constants of default screen width and default screen height are defined in case the front end doesn't give these parameters.

Canvas An `OffscreenCanvas` instance is created and instantiated with the dimensions of by default the values of the defined constants. The `OffscreenCanvas` will be used as the canvas to generate the desired image on, and since it's not being shown on the screen, will occupy less system resources and boost the calculation speed. Corresponding variables is declared after the instantiation, respectively `canvas` for the `OffscreenCanvas` itself and `ctx` as the 2d context of the canvas.

4.3.2 Message Reception

See Figure 4.4 Message Exchange.

4.3.3 Iteration Limit

4.3.4 Iteration Count for One Point

4.3.5 Image Generation

4.3.6 High Precision Version

4.4 Utility Assets

Other open source third-party utilities lie in different folders with corresponding names.

4.4.1 Folder `./js`

In `./js` folder, all JS third-party files are here, including:

- File `decimal.min.js` is for high-precision floating points calculation for JavaScript.
- File `jquery-3.4.1.min.js` is for DOM traversal and manipulation, event handling and animation.
- File `bootstrap.bundle.min.js` is for some basic styling of the control panel sitting on top right corner of the screen.

4.4.2 Folder `./fa`

4.4.3 Folder `./bs`

4.4.4 Folder `./css`

Chapter 5

Discussion

5.1 Main Section 1

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam ultricies lacinia euismod. Nam tempus risus in dolor rhoncus in interdum enim tincidunt. Donec vel nunc neque. In condimentum ullamcorper quam non consequat. Fusce sagittis tempor feugiat. Fusce magna erat, molestie eu convallis ut, tempus sed arcu. Quisque molestie, ante a tincidunt ullamcorper, sapien enim dignissim lacus, in semper nibh erat lobortis purus. Integer dapibus ligula ac risus convallis pellentesque.

5.1.1 Subsection 1

Nunc posuere quam at lectus tristique eu ultrices augue venenatis. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Aliquam erat volutpat. Vivamus sodales tortor eget quam adipiscing in vulputate ante ullamcorper. Sed eros ante, lacinia et sollicitudin et, aliquam sit amet augue. In hac habitasse platea dictumst.

5.1.2 Subsection 2

Morbi rutrum odio eget arcu adipiscing sodales. Aenean et purus a est pulvinar pellentesque. Cras in elit neque, quis varius elit. Phasellus fringilla, nibh eu tempus venenatis, dolor elit posuere quam, quis adipiscing urna leo nec orci. Sed nec nulla auctor odio aliquet consequat. Ut nec nulla in ante ullamcorper aliquam at sed dolor. Phasellus fermentum magna in augue gravida cursus. Cras sed pretium lorem. Pellentesque eget ornare odio. Proin accumsan, massa viverra cursus pharetra, ipsum nisi lobortis velit, a malesuada dolor lorem eu neque.

5.2 Main Section 2

Sed ullamcorper quam eu nisl interdum at interdum enim egestas. Aliquam placerat justo sed lectus lobortis ut porta nisl porttitor. Vestibulum mi dolor, lacinia molestie gravida at, tempus vitae ligula. Donec eget quam sapien, in viverra eros. Donec pellentesque justo a massa fringilla non vestibulum metus vestibulum. Vestibulum in orci quis felis tempor lacinia. Vivamus ornare ultrices facilisis. Ut hendrerit volutpat vulputate. Morbi condimentum venenatis augue, id porta ipsum vulputate in.

Curabitur luctus tempus justo. Vestibulum risus lectus, adipiscing nec condimentum quis, condimentum nec nisl. Aliquam dictum sagittis velit sed iaculis. Morbi tristique augue sit amet nulla pulvinar id facilisis ligula mollis. Nam elit libero, tincidunt ut aliquam at, molestie in quam. Aenean rhoncus vehicula hendrerit.

Appendix A

Frequently Asked Questions

A.1 How do I change the colors of links?

The color of links can be changed to your liking using:

`\hypersetup{urlcolor=red}`, or

`\hypersetup{citecolor=green}`, or

`\hypersetup{allcolor=blue}`.

If you want to completely hide the links, you can use:

`\hypersetup{allcolors=.}`, or even better:

`\hypersetup{hidelinks}`.

If you want to have obvious links in the PDF but not the printed text, use:

`\hypersetup{colorlinks=false}`.