WebPlotDigitizer User Manual
Version 2.6

Ankit Rohatgi*

October 20, 2013

Contents

*E-Mail: ankitrohatgi@hotmail.com

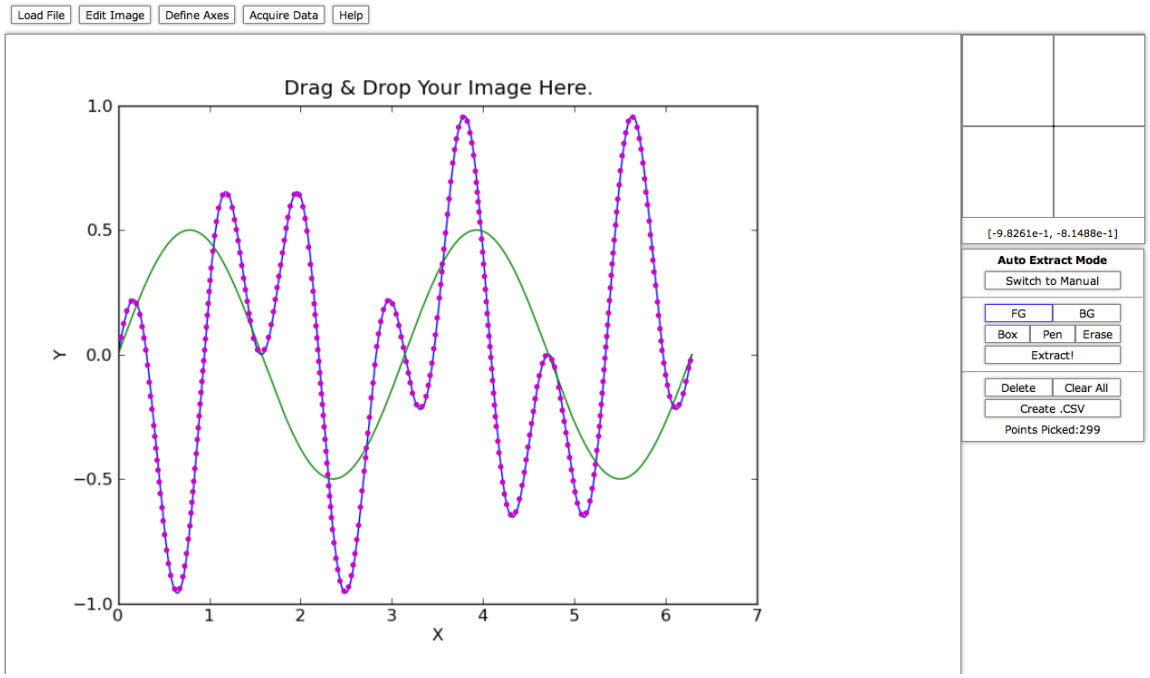


Figure 1: Screenshot of WebPlotDigitizer showing the data points recovered on a plot via automatic detection.

1 Introduction

A large quantity of useful information is available only as plotted data points in an image. In these images, it is easy to determine the relationship between the variables involved, but recovering the exact numerical values of the data is usually a tedious and error prone process. To aid this time consuming task of data recovery, several digitization softwares have been developed over the years[1]. However, even with the abundance of free and commercial softwares, this task remains daunting and prone to errors. Many of the existing softwares are either designed to work only on specific operating systems or work with a limited variety of plots. Some are just difficult to use or prone to errors. Finally, many require a paid license which prevents their widespread use by students, independent researchers or organizations with limited resources.

Because of the above limitations in current digitizing softwares, WebPlotDigitizer was developed to facilitate easy and accurate data extraction from a variety of plot types and also maps. This program has been built using HTML5 which allows it to run within most popular web browsers and does not require an installation process that is performed by the user. This is also distributed free of charge as an opensource software. A screenshot of a typical session of the software is shown in Figure 1.

[1]For a list of existing softwares, visit `http://en.wikipedia.org/wiki/Converting_scanned_graphs_to_data`

1.1 History

WebPlotDigitizer was initially developed while working on my graduate studies at the University of Notre Dame. Having to pull out data from several publications for comparing and contrasting my own findings in the lab was too painful even for someone studying fluid-particle interactions in suspensions flowing at low Reynolds numbers. The search for a tool to aid this process usually ended in realizing that most of the existing softwares for this purpose did not fulfill many of the requirements. I was faced with a similar task as an undergraduate student. Back then, just writing a simple Java based code for picking a few points by clicking on them manually was sufficient. My advisor at Notre Dame also had a Matlab code to do this job for two dimensional XY plots, but the points still had to be picked manually. Both the options were not very convenient.

Some of the experimental work in the lab required me to learn some basic image processing techniques which eventually formed the basis of the automatic detection algorithms used here. Image processing knowledge along with some interest in learning the much publicized HTML5 APIs at the time were a perfect match to create a web based data extraction tool like this.

Considering the significant interest in this software, I have continued to refine the software in my spare time even after completing my graduate studies in 2012.

1.2 User Manual and Tutorials

This user manual describes the various capabilities of the software and aims to help the user in making an effective use of the software. This manual may be updated continuously to match the latest deployed version of the software. A few video tutorials for the previous version of the software are available at `http://arohatgi.info/WebPlotDigitizer`. Other resources to find technical information about the software are also being considered.

1.3 License

WebPlotDigitizer is distributed under GNU General Public License version 3 by Ankit Rohatgi. For complete terms and conditions, please refer to `http://www.gnu.org/copyleft/gpl.html`

1.4 Source Code

WebPlotDigitizer is an open source software (see above). The source code can be obtained from GitHub (`https://github.com/ankitrohatgi/WebPlotDigitizer/`). Feel free to contact via email if you wish to contribute to this project.

1.5 Availability

The latest released version of the software can be used directly from the website `http://arohatgi.info/WebPlotDigitizer`. For the Google Chrome web browser, an *app* pointing to the online software is also available at the Chrome App Store (`https://chrome.google.com/webstore/category/apps`).

1.6 Supported Browsers

Version 2.6 was ensured to work without major issues on the following browsers:

- Safari 6.0.5 on Mac OS 10.8.5
- Google Chrome on Mac OS 10.8.5
- Google Chrome on Windows 7 32-bit
- Firefox 24.0 on Windows 7 32-bit
- Internet Explorer 10 on Windows 7 32-bit
- Firefox 24.0 on Xubuntu Linux 13.04 32-bit

It is expected that browsers similar in functionality and support for the HTML5 API should not have any major problems executing the version 2.6.

1.7 Citing WebPlotDigitizer

If you wish to cite WebPlotDigitizer in any of your works, then please use the following information:

Author	Ankit Rohatgi
Title	WebPlotDigitizer
Website	`http://arohatgi.info/WebPlotDigitizer`
Version	2.6
Date	October, 2013
E-Mail	ankitrohatgi@hotmail.com

1.8 Reporting Issues

In case of issues with the data recovery, access to the software or general technical questions, feel free to contact via e-mail. Issues specific to bugs in the software can also be reported on the issues page on GitHub: `https://github.com/ankitrohatgi/WebPlotDigitizer/issues`

1.9 Data Privacy

WebPlotDigitizer's image analysis code runs entirely on the user's computer and does not store the loaded images or data on to any server. When *Graph in Plotly* option is selected, the digitized data is transmitted to Plotly (`http://plot.ly`) servers.

1.10 Funding

WebPlotDigitizer is not a funded project and is supported mainly by my spare time and effort. PayPal donations via the website have helped keep the project afloat and are much appreciated.

2 Loading Plots

The image file containing the figure to be analyzed can be loaded into the software in the following ways:

Figure 2: The top row of buttons in WebPlotDigitizer.

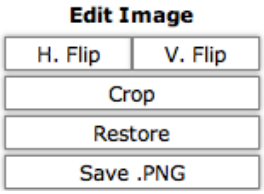


Figure 3: A few basic tools for editing the loaded image are available.

1. **Drag & Drop Operation:** Click down on the file icon in the file browser and move the mouse over the area containing the previous image in WebPlotDigitizer. Release the mouse button to load the file in the software.
2. **Browse File on Hard Drive:** Click on the *Load File* button on the top left of WebPlotDigitizer (Figure 2). A popup should show up with a control labelled *Choose File*. Use this control to look for the desired file.
3. **Copy-Paste from Clipboard:** This is only supported in Google Chrome web browser. An image selected by copying in a PDF or an image viewer can be pasted on to the software via a simple copy-paste operation.

2.1 Supported Image Formats

WebPlotDigitizer relies on the image formats supported by the HTML5 *canvas* element. Most browsers support common image formats such as JPEG, PNG, BMP and GIF. Since the support for an image format depends on the browser used with the software, please refer to your browser's manual for details. For popular browsers, you can also refer to Wikipedia (`http://en.wikipedia.org/wiki/Comparison_of_web_browsers#Image_format_support`).

3 Edit Image

If the loaded image needs to be inverted or if only a small area of the image is relevant, then you may wish to use some of the editing tools available by clicking the *Edit Image* button in the top toolbar (Figure 2). Clicking this button will make a set of buttons appear in the sidebar as shown in Figure 3. The buttons available here are for the following purpose:

1. **H. Flip:** Invert the image horizontally.
2. **V. Flip:** Invert the image vertically (not the same as 180° rotation).
3. **Crop:** Crop a rectangular region from the image. The user can draw the desired rectangular region on the image after clicking this button.

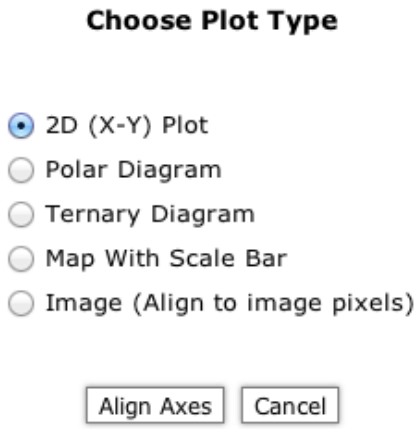


Figure 4: Popup with plot types that are supported in the software.

4. **Restore:** Restores the originally loaded image.

5. **Save .PNG:** Opens a new tab containing the edited image.

4 Define Axes

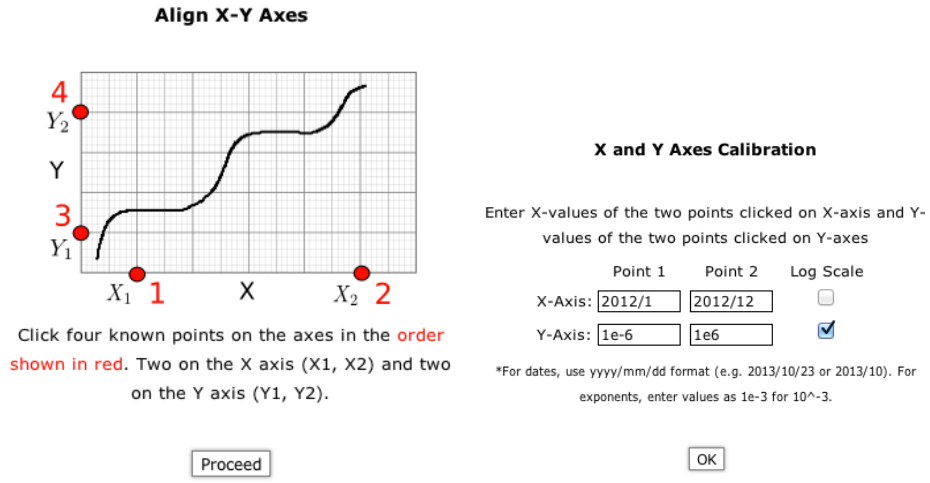
After editing the image as needed (or not at all), you should specify the type of axes used in the plot. This step is required for the software to correctly map the image pixels to possible data points in the image. Depending on the plot type, you will have to select a few known points on the axes. On clicking the *Define Axes* button (see Figure 2), you should be presented with the menu shown in Figure 4.

4.1 2D (X-Y) Plot

Most plots that are on a two dimensional cartesian coordinate system fall under this category. Two dimensional plots that are skewed such that the axes are not mutually perpendicular will also work. Also, neither the horizontal or the vertical axes need to be perfectly aligned with the horizontal and vertical lines of the computer. An image rotated by an angle should also work.

On selecting this option, you should be presented with a popup window which asks you to click on two points on the horizontal axis and two points on the vertical axis. After clicking *Proceed* on that popup, click on two points on one of the two axes (x_1, x_2) and two on the other (y_1, y_2). For better accuracy during the digitization process, pick the points that are as far away from each other as possible. Also, remember the (x_1, x_2) and (y_1, y_2) values on the axes as you will be required to enter those once four points have been clicked on.

After the four points that are required have been clicked on, another popup window will appear where you will be required to enter the values at these points. This helps the software map the image pixels corresponding to data points to their actual values when the image is digitized.



(a) Mark four points to align axes

(b) Specify values

Figure 5: Alignment for 2D (X-Y) Plot.

4.1.1 Format of Calibration Values

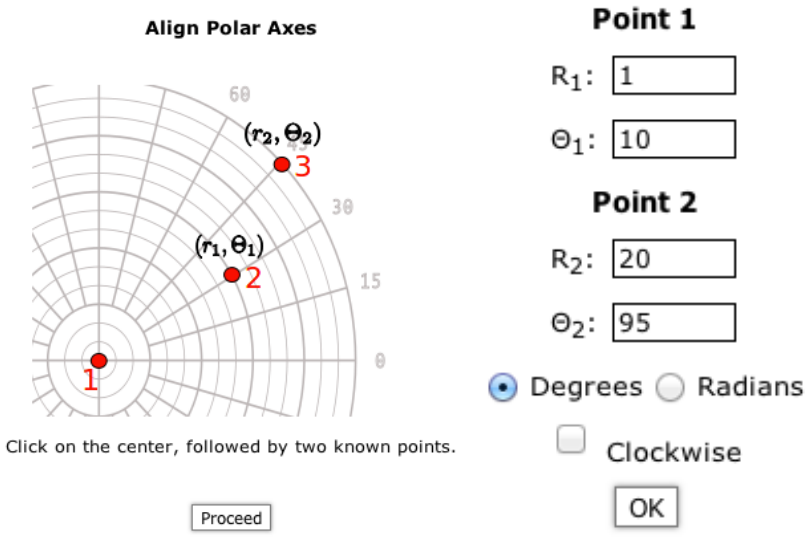
Like most computer programs, WebPlotDigitizer accepts integers (e.g. 1, 2, 3 etc.) or floating point numbers (e.g. 3.14159). Some extra things to keep in mind are as follows:

1. Fractions (e.g. 1/2) are not computed as numbers.
2. For exponentials, the caret symbol (^) is not recognized and the values have to be entered as 1.45e-10 for 1.45×10^{-10} (for example).
3. **Dates:** This is enabled only for 2D (X-Y) Plots. At the time of calibration, the dates have to be entered in the format shown below. With the final digitized data, however, results can be formatted in many different ways (see section 6.2).

Date	Format	Examples
Year, Month and Date	YYYY/MM/DD	2012/10/23, 2012/10/5 or 2012/10/05
Year, Month	YYYY/MM	2012/10 or 1989/5
Year	YYYY	2012 (treated as any integer)

4.2 Polar Diagram

Select this option if the data points in the image are plotted on a polar axes. On selecting this, you will be required to click on three known points including the center of the polar diagram (Figure 6). After clicking on 3 points, you can also select the axes orientation and select Degrees or Radians for the angle. The values entered here also have to follow the format similar to 2D (X-Y) Plots. Dates are not parsed here.



(a) Mark three points to align axes (b) Specify values

Figure 6: Alignment for Polar Diagram.

4.3 Ternary Diagram

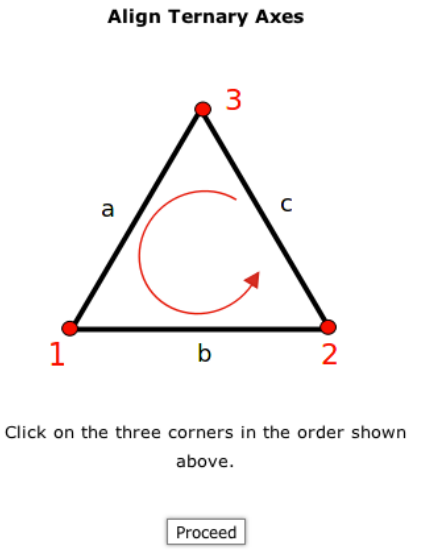
Ternary phase diagrams are typically harder to read than simple two dimensional cartesian or polar plots. Using this software to recover data makes the process of data recovery extremely straightforward and thus reduces the possibility of misinterpreting the data. For this type of plot, simply mark the three corners as shown in the instructions and then specify the range of variables and orientation of the diagram (Figure 7).

4.4 Map With Scale Bar

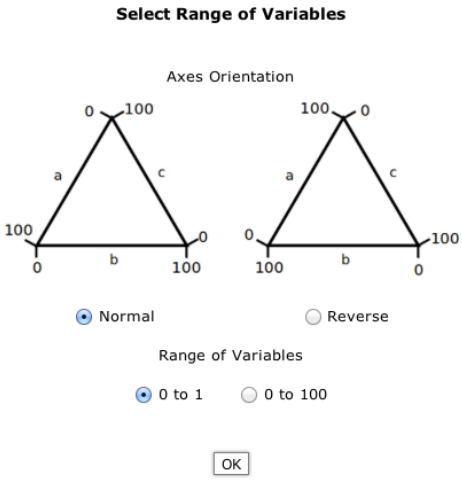
This plot type is similar to 2D (X-Y) Plots and is provided only as a convenience for images that only have scale information (e.g. microscope images or maps). To calibrate this plot type, simply click on the two ends of the scale bar and enter the scale value without units (Figure 8). The coordinates reported by the software assume the origin to be located at the bottom left of the image. The (x,y) values that are generated are scaled by the scaled factor entered during calibration.

4.5 Image (Align to Image Pixels)

This plot type is similar to the Map plot type and is also provided as a convenience over the generic 2D (X-Y) plots. If you select this type, no calibration information is required as the software will calibrate the on-screen pixels to the pixels on the image automatically. This plot type is useful for looking up exact pixel location of a feature in an image.

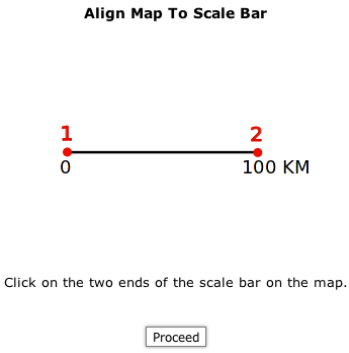


(a) Mark three points to align axes



(b) Specify values

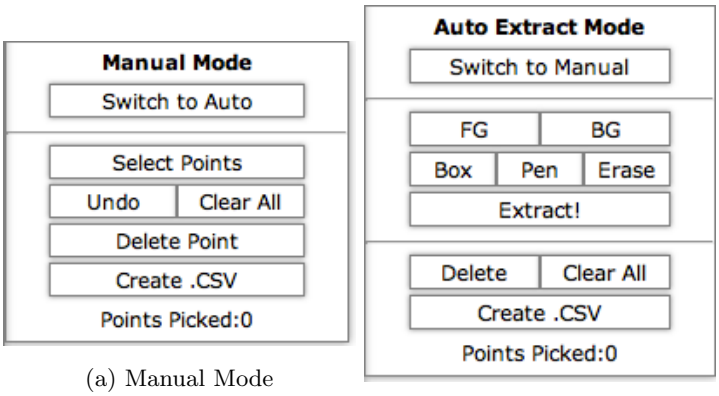
Figure 7: Alignment for Ternary Diagram.



(a) Mark ends of the scale bar

(b) Specify scale value

Figure 8: Alignment for maps and microscope images.



(a) Manual Mode

(b) Automatic Mode

Figure 9: Data acquisition controls.

5 Acquire Data

Once the plot axes have been calibrated, you can begin selecting data points on the image. Also note that the numbers below the zoom window reflect actual data coordinates corresponding to your mouse position on the image. If you see incorrect numbers here, then perhaps incorrect calibration values were entered. You must repeat the axes calibration in this situation.

WebPlotDigitizer should also show a side panel with the data acquisition controls (Figure 9) when the axes are aligned. This sidebar can also be brought up by clicking on the *Acquire Data* button. The data acquisition can be done in either manual or automatic mode. You can alternate between the two modes at any time. In the manual mode you can click on the image to add data points or remove some of the digitized data points. In the automatic mode, you can set up an extraction algorithm that can differentiate between the data points and the image background and mark several data points in a short time.

5.1 Manual Mode

The buttons available in this mode are as follows

1. **Switch to Auto:** Switch to automatic extraction mode.
2. **Select Points:** After clicking this button, you can click on the image area to add data points. Any point added on the image will automatically be converted from on-screen pixels to data values utilizing the axes calibration.
3. **Undo:** This removes the last data point that was added.
4. **Clear All:** This deletes all data points added on the image. This does not clear the axes calibration.
5. **Delete Point:** After clicking this button, you can click on a previously added data point to delete it.

6. **Create .CSV:** If one or more data points are present, this brings up a popup with the digitized data.

5.2 Automatic Mode

The controls available in the automatic mode are used for providing the extra set of inputs that are required by the automatic extraction algorithms. The steps required to extract data in this mode are described later in this section. The purpose of the various buttons available in this mode is as follows:

1. **Switch to Manual:** Switch to manual mode
2. **FG:** Select foreground color of the data point or curves. This is used for foreground based extraction described later. You do not need to specify this if you using background based extraction.
3. **BG:** Select background color of the data. This is used only when using background based extraction. This is not required if foreground based extraction is used.
4. **Box:** This is used to mark a rectangular region to be used during the search for data points.
5. **Pen:** This is used for marking areas of the image to be included in the search for data point via free hand drawing.
6. **Erase:** This is used to unmark the areas marked using the *Box* or *Pen* tools.
7. **Extract!:** Brings up a window containing options to select the automatic extraction algorithms and specify the remaining parameters.
8. **Delete:** This button has the same function as the *Delete Point* button in the manual mode. After clicking this button, you can click on a previously chosen data point to remove it.
9. **Clear All:** This deletes all the data points and the marked region for data extraction. Axes calibration is not affected by this.
10. **Create .CSV:** If one or more data points are present, this brings up a popup with the digitized data.

5.2.1 Differentiating Data From Background

Automatic data extraction relies on separating the color of the data points or curves from the background in the image. The extraction algorithms can work in two modes of color extraction: Foreground mode and Background mode. In the foreground mode, the algorithms look for the foreground color specified for the data and ignore everything else. In the background mode, the algorithms include everything except the background color as potential data points. If the data points or curves of interest are uniformly colored (approximately), then the foreground mode may be more suitable. Otherwise if the background is uniformly colored (approximately) and the curve or data points are not then the background mode may me more suitable.

To specify the foreground or the background color, use the *FG* or *BG* buttons described above. After clicking either of these buttons, you can either specify the red, green and blue (RGB) values of

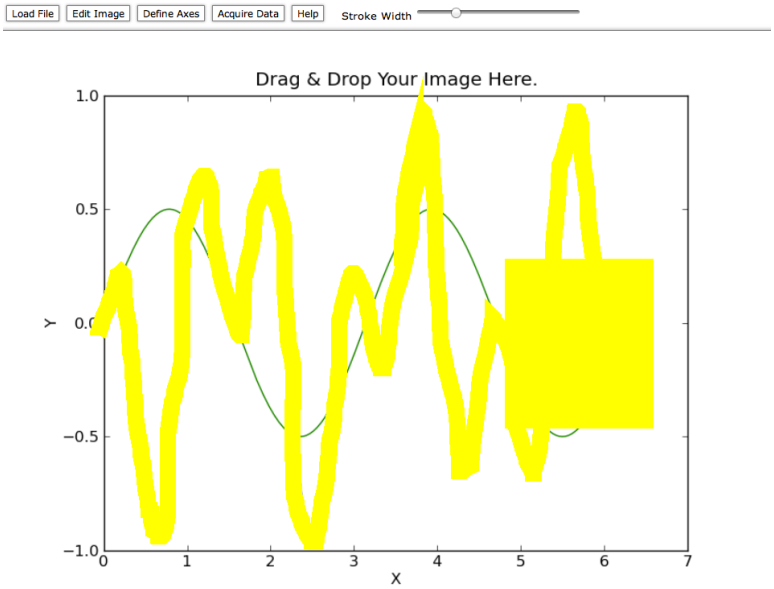


Figure 10: Use the *Box*, *Pen* and *Erase* tools to mark the region containing the required data (automatic extraction).

the color. Alternatively, you click the *Color Picker* button and pick up the color of the pixel that is under the subsequent click made on the image.

5.2.2 Mark Region of Interest

The extraction algorithms also need to know the region of the image to be searched for the specified colors. The software does not search the entire image as on many occasions, the data point or curve colors may be repeated in non-data parts of the image. To specify the region of interest, use the *Box*, *Pen* and *Erase* tools to paint over a yellow mask over the data part of the image as shown in Figure 10. The extraction algorithms will look for data only under the yellow colored region.

5.2.3 Data Extraction

After setting the data colors and marking the region of interest, you can click on the *Extract!* button. This brings up a popup window as shown in Figure 11.

In the left part of this window (under *Fine Tuning*), you can select the color detection mode to be used by the algorithms. You can also specify the extent of variability that is allowed for while comparing colors (*Color Distance*). The black and white (binary) image shown is the final result of the color extraction. It is normal to require a non-zero amount of variability even for seemingly uniformly colored data. You can adjust the *Color Distance* value here and click *Re-scan* to generate another binary image. Adjust the color distance value till you obtain a clear picture of the data of interest only.

In the right part of the window (*Choose Algorithm*), you can select the algorithm to be used and specify the algorithm specific set of parameters. These algorithms are described in detail in the next

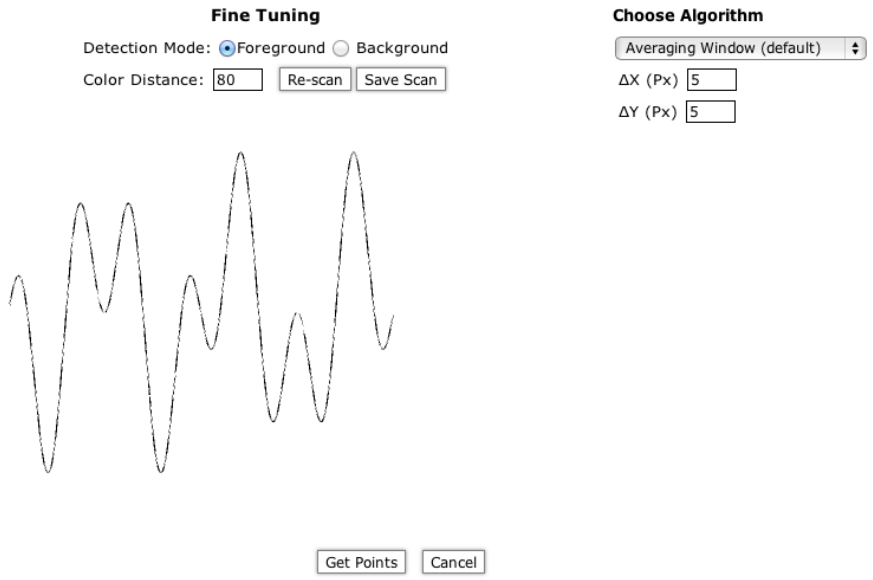


Figure 11: Define color selection mode, allowed variation in colors and select the algorithm for automatic data extraction.

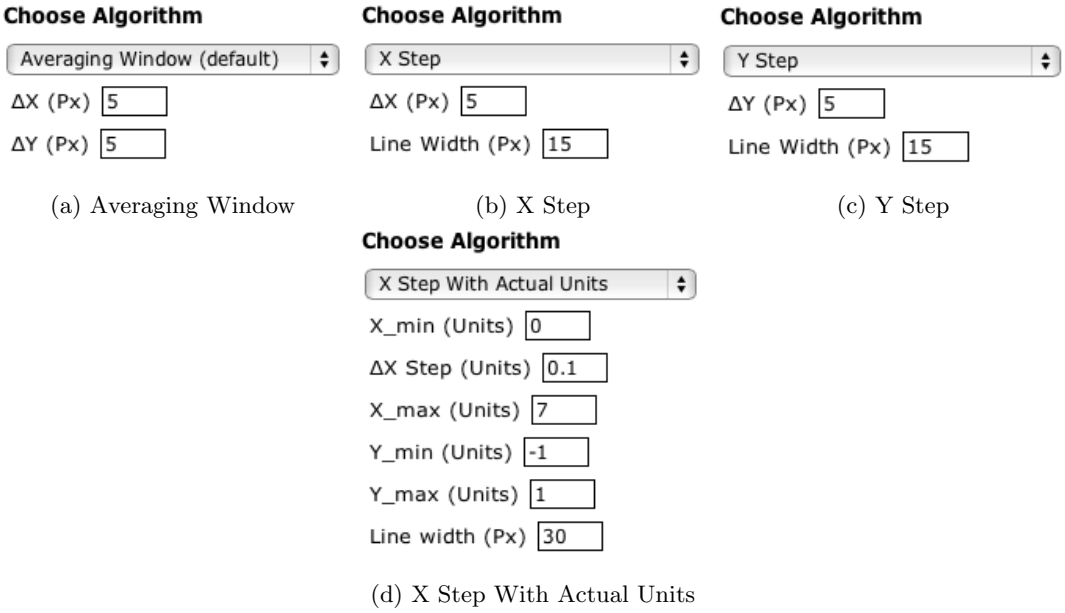
section. After selecting the algorithm and specifying the required parameters, click *Get Points* to allow the automatic extraction to finally yield the data points. If you wish to repeat the extraction process, then you will have to mark the region of interest again and then select *Extract!*. After obtaining data points from automatic mode, you can add or remove specific points using the manual mode.

5.3 Digitization Algorithms

Four different digitization algorithms are available in WebPlotDigitizer as shown in Figure 12. The *Averaging Window* algorithm is set as the default as it is usually suitable for many plot types. The other algorithms are more suited for 2D (X-Y) plots, but may be useful in other cases too. These algorithms are under active development and a few shortcommings with the current algorithms should get addressed in the future versions.

5.3.1 Averaging Window

As mentioned above, this is probably the most useful algorithm and is useful across many plot types. This algorithm finds the mean position of a colored region that can fit in a rectangular region that is ΔX pixels (on-screen) wide and ΔY pixels (on-screen) tall. As a user, you should increase the size of the box for thick lines or large data points and decrease it for thin lines. If you see multiple points incorrectly detected across the width of a thick data curve, then you need to increase the numbers specified here. The fact that this requires on-screen pixels may be changed in the future so that the values in actual units in the current axes can be specified.



(a) Averaging Window

(b) X Step

(c) Y Step

(d) X Step With Actual Units

Figure 12: Four automatic extraction algorithms are available. These are subject to change in the future releases.

5.3.2 X Step

This algorithm starts on the left side of the image and scans in the vertical direction. After detecting the first set of data, this jumps ΔX pixels (on-screen) to the right. In the vertical direction, it picks points by finding average position of colored pixels that fall within the *Line Width* of each other. So for thick curves or data points, increase the value specified for *Line Width*.

Due to limited utility of this algorithm, this may be removed or replaced in the future versions of the software.

5.3.3 Y Step

This is similar to the X Step algorithm, but this starts scanning from the bottom of the image and moves up scanning horizontally. Again, due to somewhat limited utility of this algorithm, this may be removed or replaced in the future versions of the software.

5.3.4 X Step With Actual Units

This algorithm works only for 2D (X-Y) plots at the moment. This is an improved version of the X Step algorithm as the exact scanning positions on the X axis can be specified in the data units. This is often useful while comparing values from two sets of data on the same image. Specify the range of values in X and Y axis to look for data by setting $X_{min}, X_{max}, Y_{min}$ and Y_{max}. Set ΔX to set the step size in X direction. The width of the curve (*Line Width*) still needs to be specified in terms of on-screen pixels.

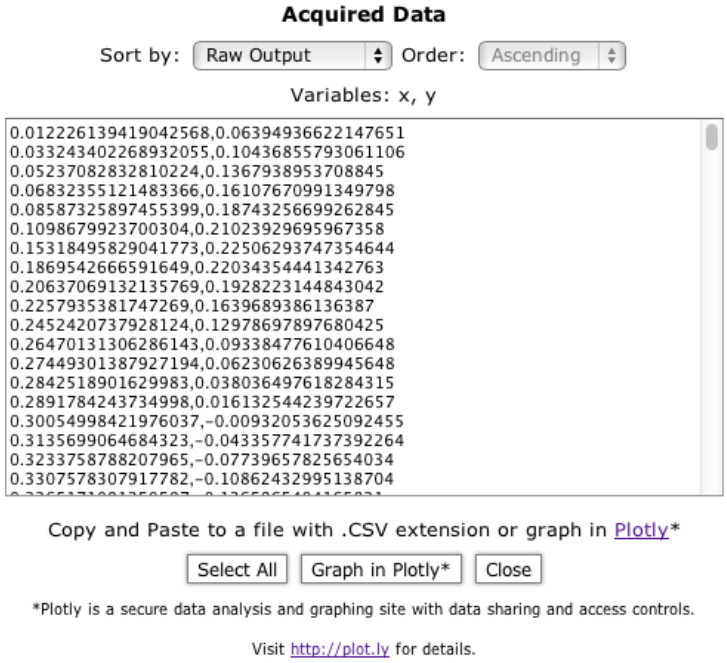


Figure 13: CSV formatted digitized data. This can be sorted, pasted to a .CSV file or graphed in Plotly.

6 Handling Digitized Data

Once the required data points are marked on the image using the manual mode, automatic mode or a combination of both, the digitized values can be seen by clicking the *Create .CSV* button. This presents a popup window as shown in Figure 13. Here, the digitized values can be sorted by the variable or in order of the distance between the points (Nearest Neighbor). The values can also be copied and used in common data analysis softwares. Recently, an option to send these values over to another cloud based data analysis and graphing software called Plotly (`http://plot.ly`) has also been added.

6.1 Sort Data

The digitized data can be left unsorted (Raw Output) or by one of the axes variables in ascending or descending order. The Nearest Neighbor option sorts the data depending on the distance of the points from each other.

6.2 Formatting Dates

If one or both of the axes in a 2D (X-Y) plot contain dates then fields to specify the output format of the values are also shown (Figure 14). In these fields, the following pieces of text are replaced with the corresponding part of the date to format the text (case insensitive):

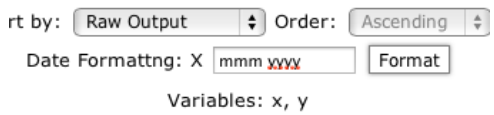


Figure 14: Field to specify formatting of dates. This appears on X, Y or both axes depending on which variables contained dates at the time of axes calibration.

Text	Replaced With	Example
YYYY or yyyy	Year, all digits	2012
YY or yy	Year, last two digits	98 for 1998
MMMM or mmmm	Month, full name	January
MMM or mmm	Month, short name	Jan
MM or mm	Month, numeric	10
DD or dd	Date	23

A few examples are shown below:

Format Field Text	Date Shown (Example)
dd-mm-yyyy	23-10-2012
mmm-yyyy	Oct-2012
yyyy abc mmmm	2012 abc October
mmm 'yy	Oct '12

6.3 Export to .CSV File

Comma-Separated Values (CSV) format files are simple text files containing tabular data. Each row of line corresponds to a table row and the column values are separated by a character or a string (usually just a comma)[2]. Due to its simplicity, CSV format is supported in most data analysis softwares like Microsoft Excel, Matlab etc.

To save the values from WebPlotDigitizer into a CSV file, all you have to do is open your favorite text editor (e.g. Notepad on Microsoft Windows) and Copy-Paste the values shown in the results popup window. Save the file with a .CSV file extension. You can now use this file in your favorite data analysis software.

[2]For more information on CSV format, check out `http://en.wikipedia.org/wiki/Comma-separated_values`