
# OscilloData 2024

Project Documentation

Authors: Thomas Claux and Jad Tijani
École Polytechnique Fédérale de Lausanne (EPFL)

Date: 2024

## 1. Introduction

This document presents the technical documentation for the *OscilloData 2024* project. The project was developed as a collaborative effort between the authors and colleagues from other teams. It integrates three programming environments — C, MATLAB, and LabVIEW — to process, analyze, and visualize oscilloscope trace data extracted from image files. The primary objective was to automate the detection, correction, and mathematical modeling of signal traces obtained from experimental captures.

## 2. Development Environment

The project was executed on an Asus ROG Zephyrus laptop running Windows 10. The following software environments were used for development and testing:

- **C Compiler:** CLion 2024.2.1
- **MATLAB Version:** R2024b (Windows)
- **LabVIEW Version:** 2023Q3 (Patch)

Each programming environment was responsible for a specific stage of data processing, ensuring modularity and interoperability.

## 3. C Algorithm Overview

The C program serves as the computational core for pixel-level image analysis. Its workflow is as follows:

Table 1: C Algorithm Workflow

| Step | Description |
|---|---|
| 1 | Open the binary file `Pixmap.bin`. |
| 2 | Use a histogram function to count the occurrences of each color in the image. |
| 3 | Identify the colors of the four corners and all signal traces through the histogram. |
| 4 | Allow the user to select which traces to retain, recording their colors in the standard output. |
| 5 | Write the coordinates of all points composing each trace to `Traces.txt`. |

This process ensures precise detection of traces and prepares structured data for MATLAB-based post-processing.

## 4. Error Management in C

Error management in C is handled through explicit numeric codes, as shown below:

Table 2: C Error Codes

| Error Code | Description |
|---|---|
| 1 | Pixmap not found |
| 2 | Incorrect depth |
| 3 | Width out of bounds |
| 4 | Height out of bounds |
| 5 | Missing pixels |
| 6 | Too many pixels |
| 7 | Error creating `'To sort'` directory |
| 8 | No traces detected |
| 9 | Too many corners |
| 10 | `'Traces.txt'` not created |
| 11 | Incorrect number of parameters |
| 12 | No corner detected |
| 14 | More than 10 traces |

This rigorous handling guarantees controlled termination and easier debugging.

## 5. LabVIEW Algorithm Overview

LabVIEW acts as the graphical interface of the system. Its main tasks include:

Table 3: LabVIEW Algorithm Workflow

| Step | Description |
|---|---|
| 1 | Import a PNG file and save it as a binary `Pixmap` file. |
| 2 | Create a CLUT (Color Lookup Table) specific to the image and display it. |
| 3 | Pass the Pixmap to the C executable for corner and trace identification. |
| 4 | Generate a MATLAB script based on the C output (trace colors) and user-defined parameters. |
| 5 | Execute the MATLAB script and save the final PDF in the project directory. |

This design allows the user to interact intuitively with the system while maintaining automation in data analysis.

## 6. Error Management in LabVIEW

Error management in LabVIEW is achieved through dialog-based notifications, as summarized below:

Table 4: LabVIEW Error Handling

| Error Case | Action Taken |
|---|---|
| Non-PNG file imported | Displays an error dialog and stops execution. |
| Errors from C code | Displays corresponding error dialog and stops execution. |
| Other MATLAB-related issues | Shows MATLAB error code and message before stopping. |

This system ensures transparency and straightforward user feedback during operation.

## 7. MATLAB Algorithm Overview

MATLAB performs the final analytical and graphical processing steps as detailed below:

Table 5: MATLAB Algorithm Workflow

| Step | Description |
|---|---|
| 1 | Import data from `Traces.txt`. |
| 2 | Identify the four corners and reorder them to correct image perspective. |
| 3 | Rectify the image and reposition trace points accurately. |
| 4 | Use curve fitting (`fit`) to approximate each trace either by a cubic polynomial or a sinusoidal |
| 5 | Select the best fit using the RMSE method and plot the results in their original colors. |

This approach combines computational efficiency with precise visual reconstruction.

## 8. Overview

The *OscilloData 2024* project operates in four main steps. To better illustrate how the code works, we will follow the transformations applied to the file "V6".

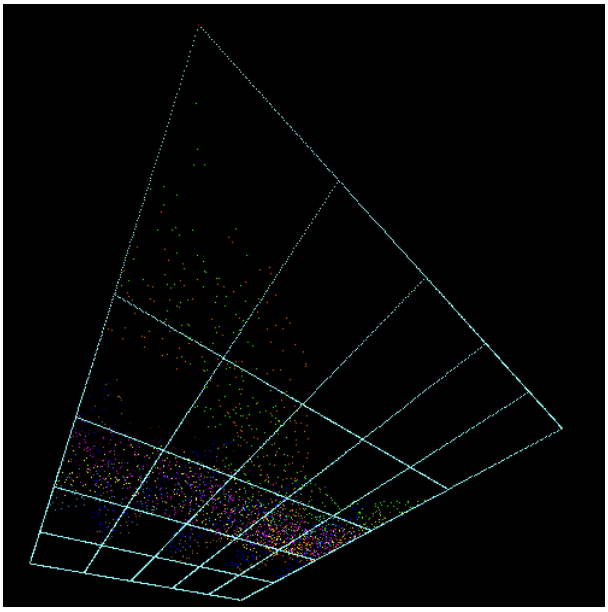

Figure 1: Initial image provided to the LabVIEW program.

First, the C program extracts the "Pixmap" from the PNG file. The pixmap is a pixel-by-pixel representation of the image, where each pixel's color is encoded using 8-bit indexed color.

Next, the program identifies the four corners of the image. The image is formatted such that these corners share a unique color that is not used in any of the plotted traces. Once the corners are detected, the program locates all pixels belonging to each trace and stores them in a list. This process is repeated for every trace found. The five traces with the most pixels are then saved in a text file called `Traces.txt`.

```
Corners= [
162, 483;
464, 148;
22, 36;
197, 6;
];

C0= [
87, 186;
84, 181;
89, 177;
72, 175;
84, 173;
95, 168;
74, 164;
85, 164;
65, 160;
68, 154;
93, 152;
60, 148;
80, 147;
75, 145;
68, 143;
91, 143;
62, 142;
90, 142;
175, 139;
189, 137;
174, 136;
185, 136;
76, 134;
92, 133;
78, 131;
81, 130;
80, 129;
159, 129;
172, 129;
75, 127;
161, 127;
59, 126;
168, 126;
178, 126;
162, 125;
172, 125;
50, 123;
88, 122;
89, 120;
```

Figure 2: Snapshot of the `Traces.txt` file.

The text file is subsequently read by the MATLAB code, which extracts all traces. The coordinates are then transformed using linear algebra techniques to straighten the image. Once the traces are cleaned, the program fits both a sinusoidal function and a

third-degree polynomial to each trace, selecting the best fit for plotting. Finally, the program plots the extracted points along with the fitted curves for all traces and saves the resulting visualization as a PDF file.

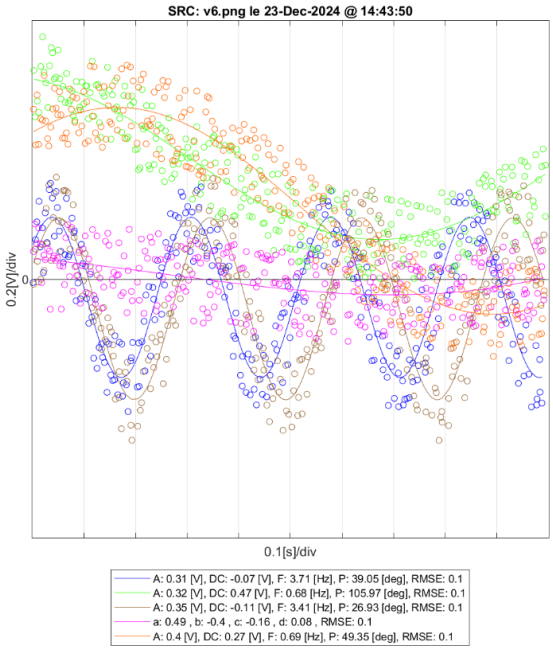


Figure 3: Final PDF output.

## 9. Conclusion

The *OscilloData 2024* project was both technically enriching and highly engaging.

Through the combination of C for data extraction, LabVIEW for automation, and MATLAB for mathematical modeling, the project successfully delivered a coherent system capable of advanced signal trace analysis. This documentation serves as a foundation for future improvements and potential research extensions.