# 4. Process description

## 4.1. Introduction

**This chapter describes the process of binarizing and extracting data from a scanned image of a legacy ionization chamber cosmic ray recording strip. This explanation will include the reasons why certain approaches were followed and why certain values were assigned to variables.**

**Segments of test images will be used to better explain the results of some methods used during different stages of the process. These segments are taken from the full sized result images and are not the results of applying the methods to segments of the original images.**

**The descriptions of the methods applied during this process can be found in Appendix A. This chapter contains images showing how the methods affect small sections of the cosmic ray image. Images showing how a larger section of a cosmic ray image is affected are shown in Appendix B. The contents of these appendices were not included in this chapter for ease of reading.**

**Multiple iterations of the same methods are applied to the test images and most of the process is applied twice. This is done to maximize accuracy and to enable the process to handle severely distorted images.**

**The process can be summarized as follows:**

1. Pre-Processing: Empty white space at the top and bottom of the image are removed along with all sprocket holes within the image. The empty spaces are cropped out of the image while the sprocket holes are erased. The image is also blurred to make subsequent steps in the process more effective and to decrease

the visibility of any noise in the image. The location of the temperature line within the image is also recorded;

2. Rough data identification: This phase identifies all pixels that probably form part of a data line, while ignoring pixels that form part of scale lines or temperature lines. At the end of this phase, steps are applied to remove small groups of pixels that probably represent noise in the image;

3. Rough data extraction: Using the results from the previous phase, the data lines are roughly extracted from the original image by extracting all pixels that fall within a small area around each identified pixel from the rough data identification phase. By only extracting specific areas of the original image, the effect of any unwanted image objects, such as sprocket holes or scale lines, is greatly reduced during subsequent phases of the process;

4. Rough data binarization: The extracted parts of the original image are scanned using a similar set of methods as in the rough data identification phase. Instead of identifying the positions of data lines within the image, this phase extracts thick versions of those lines to an empty image. This image is then scanned several times using the same set of methods, but with higher sensitivities each time. Each time the image is scanned, the thickness of the data lines decreases. After the image has been scanned several times, the result is a very thin line that mostly represents cosmic ray data. To ensure that no pixels from unwanted image objects are still present in the image, a method is applied to ensure that there is only one pixel in each column of the image matrix and that the distance between two pixels in two columns is kept to a minimum. All remaining pixels have a very high probability of being part of a data line, which allows parts of the image which contain a very small number of pixels, or no pixels at all, to be removed. This reduces the vertical size of the image. At this point, the extracted data lines are broken and the effect of unwanted image objects may have prevented some data line pixels from being identified and extracted;

5. Accurate data identification: The result of the previous phase is a smaller image that contains the data lines from the original image and much fewer unwanted image objects. This greatly increases the accuracy of the binarization process. The same process is followed as in rough data identification. The sensitivities of some parameters are increased to further improve the accuracy of the methods

applied to the smaller image;

6. Accurate data extraction and binarization: The same processes are followed as in rough data extraction and binarization;

7. Post Processing: The pixels in the image from the previous phase are connected to form an accurate plot representing the cosmic ray data in the original image. The temperature line that was identified early in the process is inserted into the final image. All rows that were cropped out during the process are added to the image as empty rows to return the final image to its original size;

**The inspiration behind the process comes from the work done by Gatos *et al*. (2006) and Valizadeh and Kabir (2012). The method designed by Gatos *et al*. (2006) roughly estimates the foreground and background of the image before binarization. By roughly estimating the foreground of the data images, the accuracy of the cosmic ray data binarization process is greatly increased. The method designed by Valizadeh and Kabir (2012) provided the idea of how to target and extract data pixels from the image, by searching for local maxima.**

**The entire process will now be described in detail.**

## 4.2. Pre-processing

**The images all have white space above and below the photographic strip in the image. These spaces are removed using the Crop method (A.1 on page 103) . By removing these spaces the processing time of the method is decreased, due to the smaller size of the image, and the high intensity pixels in these spaces cannot negatively influence the accuracy of the method.**

**The cropping method reduces the size of the image by approximately 10%. The threshold value is set at 200, meaning the first rows from the top and bottom of the image with a mean value below this threshold are taken as the boundaries of the white spaces. A buffer of 20 rows is added to these boundaries to compensate for any skew and to ensure that these white areas are cropped out completely.**

**The sprocket holes along the edges of the images are removed using the Fill method (A.3 on page 105). These sprocket holes also contain high intensity pixels that may be wrongly identified as data pixels.**

**The fill method is applied to the image after the crop boundaries have been calculated, but before the image is actually cropped. Doing so ensures that if skew or shadows cause a piece of the white area to remain in the cropped image, that area will be black instead of white. This ensures that such occurrences will not affect the results of the process. These shadows occur wherever the the photographic strip was not held flat against the scanning surface while scanning the original cosmic ray recording strip.**

**The mask size used to identify the sprocket holes is 41 pixels high and 21 pixels wide. The mask size used to remove any holes is 81 pixels high and 41 pixels wide. The smaller size of the searching mask ensures that even sprocket holes that contain some shadows will be detected and the larger filling mask size ensures that the entire sprocket hole is removed.**

**Some of the data images are so bright that there are patches of white pixels within the image that are also identified by the Fill method. These patches are also removed and this serves to improve the accuracy of the process rather than decrease it. The mask sizes ensure that no part of the data line can be directly targeted and removed by the method.**

**In several data images, the data lines pass through the sprocket holes, the size of the filler mask is kept as small as possible to prevent unnecessary loss of data when this occurs.**

**The final step of the pre-processing phase is to apply the Blur method (A.4 on page 106) to the image. The image is blurred by a 3 by 3 pixel mean filter. This process decreases noise in the image and smoothes any lines within the image, making them much more detectable by subsequent methods. The difference between applying the Scan method to a blurred and unblurred image is shown in Figure 4.1.**

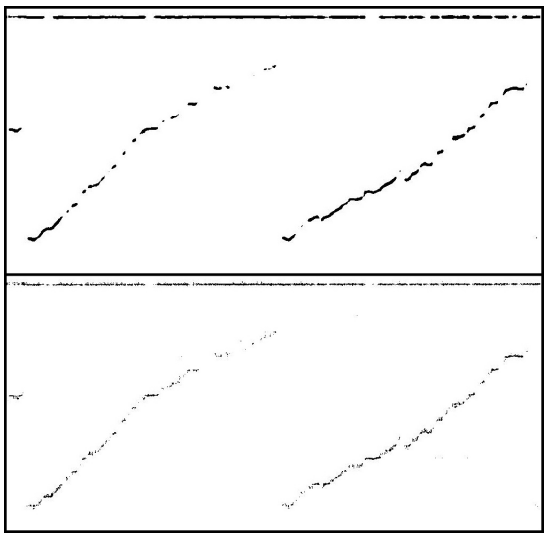

Figure 4.1.: Difference between scanning a blurred image (Top) and an unblurred image (Bottom).

**The mask size of 3 by 3 pixels was chosen because larger filter windows start to decrease the visibility of the relatively thin data lines while increasing the intensity values of unwanted image objects such as bright patches and very wide hour markers, making these objects more difficult to remove in subsequent phases of the process.**

**The results of the pre-processing phase are shown in Figure 4.2 and the steps of the process can be seen in figures B.1.1, B.1.2 and B.1.3 on page 122.**

## 4.3. Rough data identification

**The wide variety of image characteristics and quality of the data images posed the greatest challenge to accurately identifying which pieces of the image contain data and which pixels within those pieces are actually part of a data line and not part of an unwanted object, scale line or hour marker.**

**The solution to this problem was inspired by the work done by Valizadeh and Kabir (2012), who designed an adaptive water flow model for binarization of degraded document images. This binarization method searches for the lowest intensity pixels within different regions of a document image. These pixels**

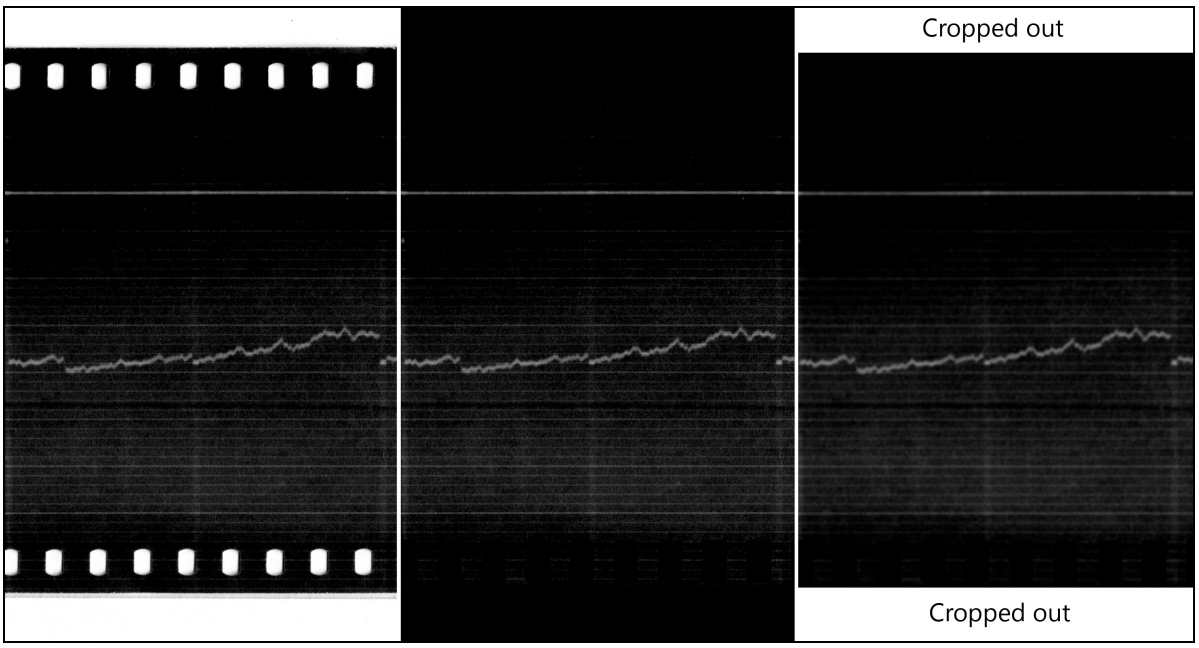


Figure 4.2.: Results of pre-processing. (A) Original (B) Filled (C) Cropped and Blurred.

**represent the text and document objects that need to be extracted from the document.**

**The data lines are some of the brightest (highest intensity) objects in a cosmic ray recording image. More specifically, the pixels within a data line (especially the pixels at the core of these lines) are usually the brightest pixels in any column of the image where the data line is visible. By identifying these pixels, a set of pixels that have a high probability of being part of the data line can be extracted.**

**The concepts used by Valizadeh and Kabir (2012) are implemented in the Scan method, which is applied to the image after it has been pre-processed.**

**Instead of searching for the lowest intensity pixels in defined regions of a degraded document image, the Scan method (A.5 on page 106) searches for the brightest pixel within each column of the cosmic ray image. By varying the tolerance of the Scan method, a binarized version of the image is created where connectivity between data line pixels is increased as the tolerance is increased.**

**However, as the tolerance increases, the number of pixels that are included in the binarized image, that do not form part of the data line, increases.**

**The tolerance of the Scan method should be set at a value that extracts as much data line pixels as possible, but as few non-data line pixels as possible. It was found that the tolerance should originally be set at 10, meaning all pixels with values that differ from the value of the highest intensity pixel by less than 10 are passed to the binary image. This value provides the best extraction of data pixels while limiting the number of additional objects that have to be removed at later stages.**

**The results of Scanning a preprocessed image using different tolerances are shown in Figure 4.3.**

**The Remove method (A.6 on page 107) is applied to the output of the Scan method. The method removes any vertical lines, spanning the height of the image, caused by columns in which all pixels have a value of zero.**

**All rows that contain more than a defined number of scanned pixels are identified by the Mark method (A.7 on page 108) and all those marked lines are then removed from the original image by replacing those rows with filler rows. The marked rows are replaced by using the Erase method (A.8 on page 108).**

**The output of these four methods is a blurred version of the original where some of the brightest horizontal lines in the image have been removed. This allows more data line pixels to be detected.**

**This group of methods (Scan, Remove, Mark and Erase) is applied to the image six times, each time extracting a more accurate data line.**

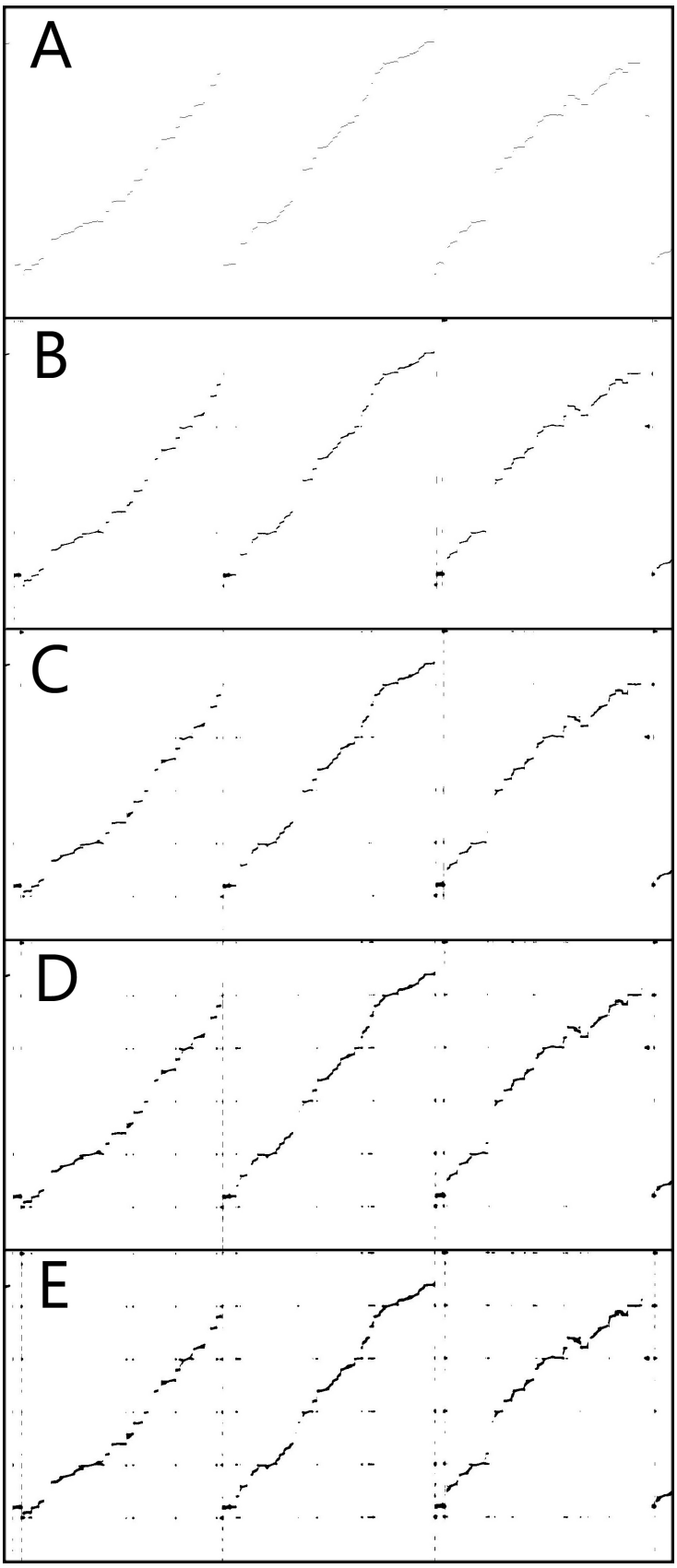


Figure 4.3.: Results of Scan method at different tolerances. A = 0. B = 5. C = 10. D = 15. E = 20.

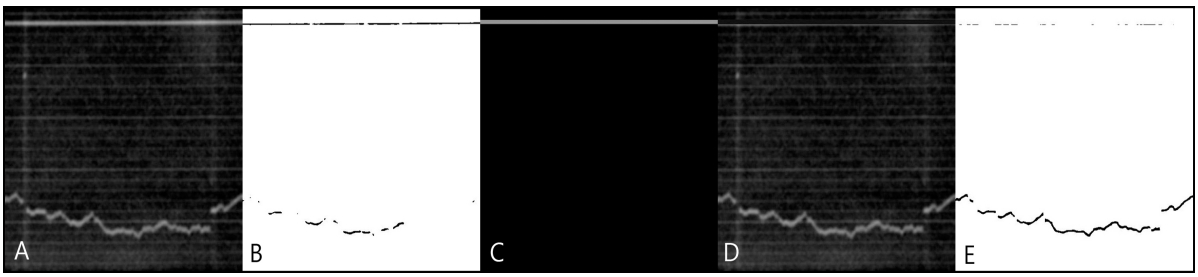


Figure 4.4.: The effect of a single iteration of the Scan, Remove, Mark and Erase methods. (A) Original image. (B) Scan and Remove output. (C) Mark output. (D) Erase output. (E) Scan output of following iteration.

**The variables of these six iterations are as follows:**

Iteration 1: Scan tolerance = 10 and Mark row population = 10% (Figures B.2.1, B.2.2 and B.2.3 on page 123);

Iteration 2: Scan tolerance = 10 and Mark row population = 10% (Figures B.3.1, B.3.2 and B.3.3 on page 124);

Iteration 3: Scan tolerance = 5 and Mark row population = 7.5% (Figures B.4.1, B.4.2 and B.4.3 on page 125);

Iteration 4: Scan tolerance = 5 and Mark row population = 7.5% (Figures B.5.1, B.5.2 and B.5.3 on page 126);

Iteration 5: Scan tolerance = 0 and Mark row population = 5% (Figures B.6.1, B.6.2 and B.6.3 on page 127);

Iteration 6: Scan tolerance = 0 and Mark row population = 5% (Figures B.7.1, B.7.2 and B.7.3 on page 128);

**The values of these variables were derived through experimentation.**

**Each variable set is applied twice to ensure that all detected lines are removed properly and that all detectable data pixels are detected. The sensitivity of the Mark method decreases as the tolerance of the Scan method decreases, because fewer and fewer pixels are extracted as these iterations are applied to the image, reducing the extracted pixel population of each row. The improvement of data visibility after a single iteration is shown in Figure 4.4.**

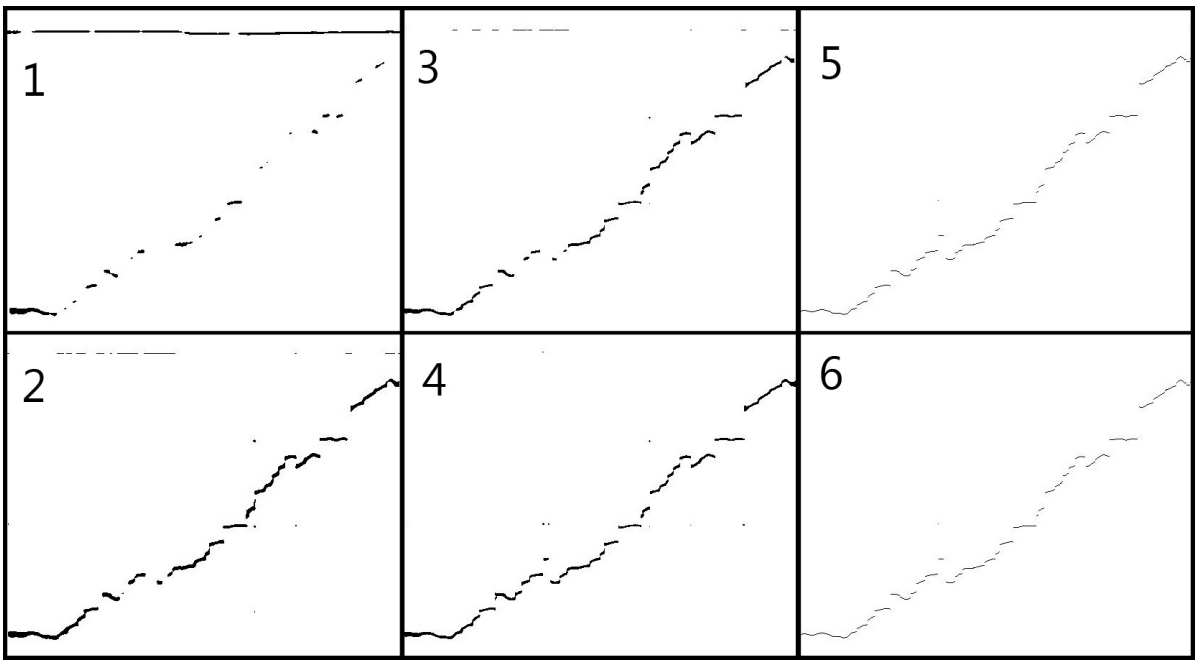


Figure 4.5.: Results of six Scan, Remove, Mark and Erase iterations.

**The population threshold values used in the Mark method were chosen so this extraction process would remove any prominent lines while not removing pieces of a relatively straight data line. If a set of rows are removed which contain data line pixels, then those pixels are recovered later.**

**The result of applying this group of methods six times is shown in Figure 4.5.**

**After these six iterations, all lines preventing the detection of data line pixels have been removed, the Scan method is applied to the image once more with a tolerance of 0 (Figure B.8.1 on page 129). The marked pixels in the resulting binary image consist mostly of data line pixels, while some pixels representing noise and high intensity unwanted image objects are still present.**

**Most of the extracted pixels that are not part of the data line form small sets that are separated from any clearly defined object or appear as random non-connected pixels. These unwanted pixels are removed by the Clean method (A.9 on page 109) as shown in Figure B.8.2 on page 129.**

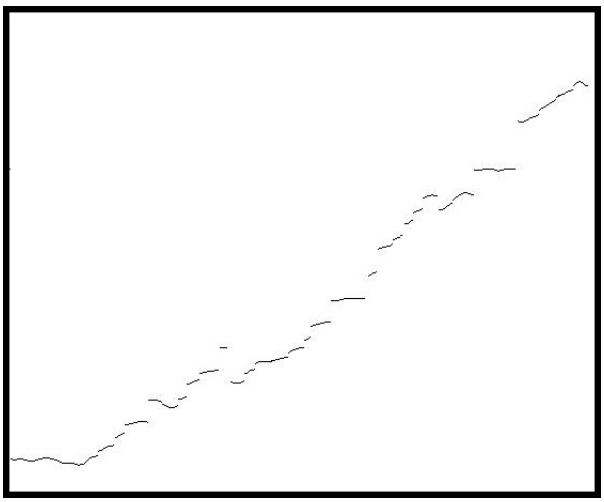

Figure 4.6.: Final output of the rough data identification phase.

**The result of this phase of the process is a binary image (Figure 4.6) that contains pixels representing the brightest objects in the image, mostly data line pixels. In most images, the data lines are only partly extracted, consisting of non-connected segments.**

**The purpose of this detection and line removal phase is not to extract the data, but to identify the probable locations of data lines within the image.**

## 4.4. Rough data extraction

**Once the probable positions of data lines within the image have been identified, methods can be applied to extract these data lines from the original image.**

**The cleaned binary image contains a set of pixels that probably represent part of the data line in the original image. This image is converted to a binary image that contains a set of areas, which probably contain parts of the data line in the original image. This is done using the Target method (A.10 on page 110).**

**The size of the mask that is targeted around each pixel, in the cleaned image, is 31 pixels wide and 61 pixels high. This size provides the best connectivity between separated data line segments while limiting connectivity between unwanted image objects.**

**The areas are labeled as connected components by the Connect method (A.11 on page 111), from which all smaller than average components are removed by the Scrub method (A.12 on page 111). The result is an image which indicates the areas in the original image that probably contain data line segments (Figure B.9.1 on page 130).**

**By removing most of the smaller components, that probably represent noise and unwanted image objects, the Identify1 method (A.13.1 on page 113) can accurately indicate the set of rows containing the data lines. All components outside these rows are discarded, thus removing most of the components representing unwanted image objects (Figure B.9.2 on page 130).**

**If any components, which form part of the data line, are wrongly removed then those components are reinserted in the image using the Bind method (A.14 on page 114) as shown in Figure B.9.3 on page 131.**

**The resulting image is a set of areas that represent the locations of data lines in the original image. At this stage of the process, the connected components resemble the shape of the data line. The Extract method (A.15 on page 115) is used to create an image with a black background to which all the areas of the original image, that are identified by the remaining connected components, are extracted.**

**The foreground of the resulting image now mostly consists of the extracted data lines (Figure B.9.4 on page B.9). The extraction process is summarized in Figure 4.7. Take note of the reinserted component indicated in image D.**

## 4.5. Rough data binarization

**The Scan method (A.5) is applied to extracted data lines just as in the rough data identification phase, and once again the Remove method (A.6) is applied to eliminate any empty columns that result in vertical black lines in the output image.**

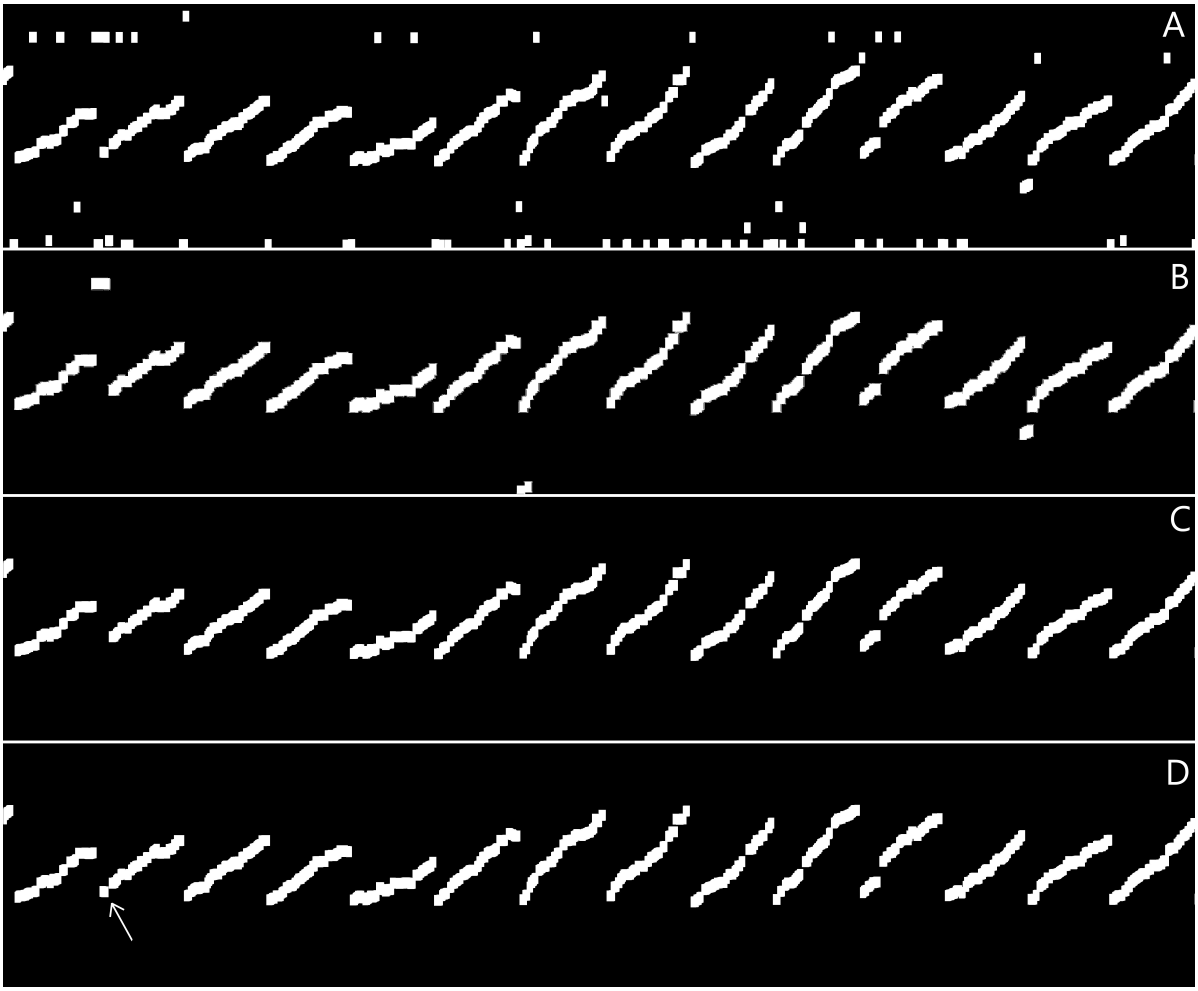


Figure 4.7.: Rough data extraction. (A) Output of Target. (B) Output of Scrub. (C) Output of Identify1. (D) Output of Bind. A reinserted component is shown at the arrow on the left. .

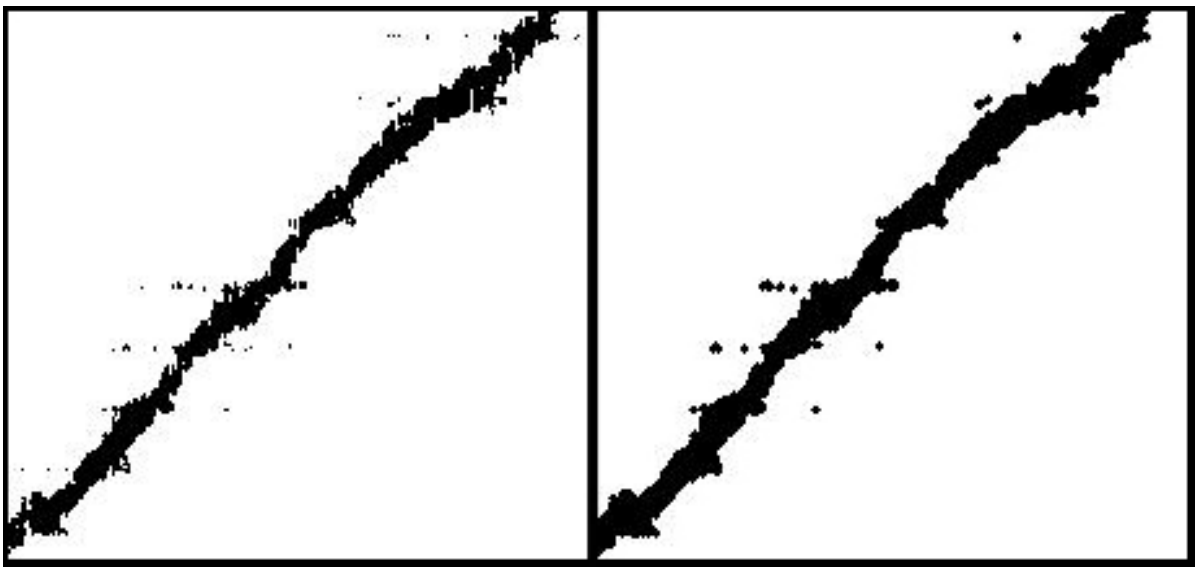

Figure 4.8.: Thickening effect of the Purify method.

**The Purify method (A.16 on page 115 ) is applied to the scanned image to remove single non-connected pixels and dilate the remaining lines in the image. The image containing these slightly thicker lines is used as a mask, by the Extract method (A.15 on page 115), to create an image consisting of a black background and only the pixels from the original image that are represented in the output of the Purify method.**

**Previously the binary output of the Scan method was used to eliminate horizontal lines in the image and allow subsequent Scan iterations to identify the data line more clearly. Now the output of the Scan method is used by the Designate method (A.17 on page 116 ) to extract only the marked pixels from the original image. This process reduces the set of large segments of the original image, to an image containing the non-binarized extracted data line from the original image. The resulting image is used as the input for the following iteration of the process just described (Scan, Remove, Purify and Designate). The effect of the Purify method can be seen in Figure 4.8, where it is shown how the method smoothes the data lines while not increasing the number of single non-connected pixels. These thicker lines ensure cleared data lines when the output of the Designate method is scanned.**

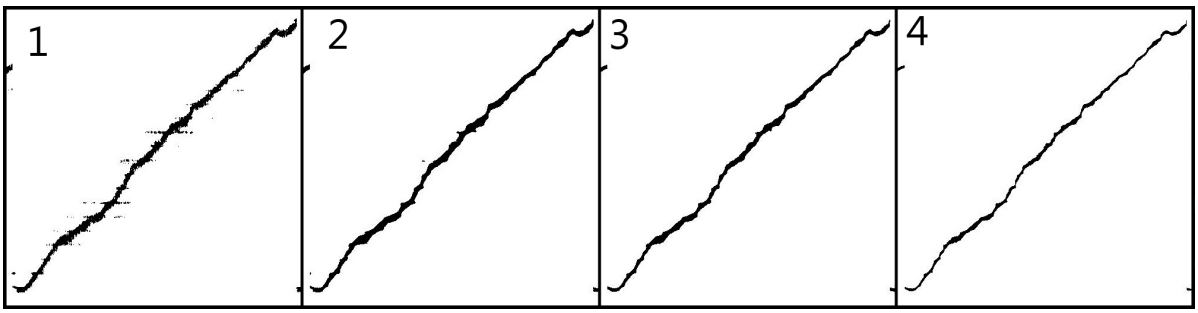


Figure 4.9.: Results of 4 Scan, Remove, Purify and Designate iterations. From left to right, iterations 1 through 4.

**Four iterations of this scanning process are applied to the image. The tolerances of the Scan methods are:**

Iteration 1: Scan tolerance = 20 (Figure B.10.1 on page 132);

Iteration 2: Scan tolerance = 15 (Figure B.10.2 on page 132);

Iteration 3: Scan tolerance = 10 (Figure B.10.3 on page 133);

Iteration 4: Scan tolerance = 5 (Figure B.10.4 on page 133);

**The starting Scan tolerance of 20 allows a thick representation of the data line to be extracted, from which a more accurate data line can be extracted during each following iteration, as can be seen in Figure 4.9.**

**The result of these four iterations is scanned once more with a tolerance of 0 (Figure B.10.5 on page 134), to extract an accurate binary version of the data line. If there are any horizontal lines remaining at this stage, which occurs rarely, then these are removed by the Mark and Erase methods. Any row that has more than 2% of its pixels marked is removed. Because the current image is a binary image, the filler value used to remove the lines is 255 (white).**

**The resulting data line can be seen in Figure 4.10.**

**The Identify2 method (A.13.2 on page 113) is applied to identify the set of rows that contain the extracted data lines and discard all other pixels in the image. The result is a binary image that contains only the data lines. Depending on**

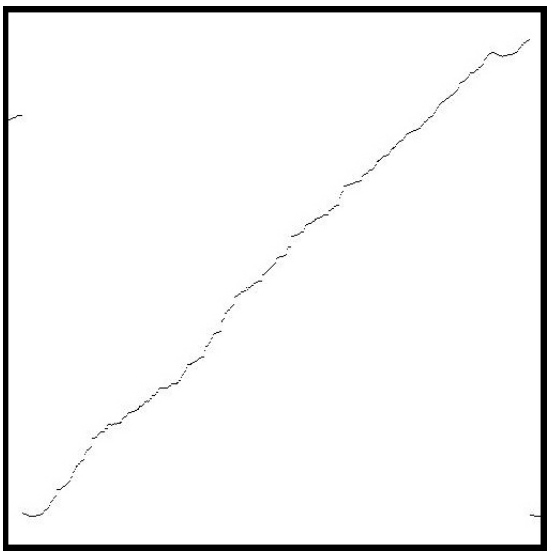

Figure 4.10.: Result of applying Scan method to output from Figure 4.9.

**the quality of the original image, these data lines could be near perfect or still severely disconnected with many segments missing.**

**The Define method (A.18 on page 116) is applied to ensure that only one pixel is marked within each column. Each defined pixel is the pixel in the current column that is closest to the marked pixel in the previous non-empty column. The initial search area is set at 50, consisting of the area 50 pixels above, 50 pixels below and 50 pixels in front of the primary pixel being investigated. All pixels within this area are considered to be within close proximity to the primary pixel.**

**This search area size was chosen to be large enough to allow pixels in severely disconnected data lines to fall within the search area, while being small enough to prevent pixels within segments of the following (different) data line from being chosen as the new primary pixel.**

**The Define method removes pixels that do not form part of a data line, as can be seen is Figure 4.11.**

**The resulting image is an extracted binary version of the data lines contained in the original image (Figure B.10.6 on page 134), where some or many segments**

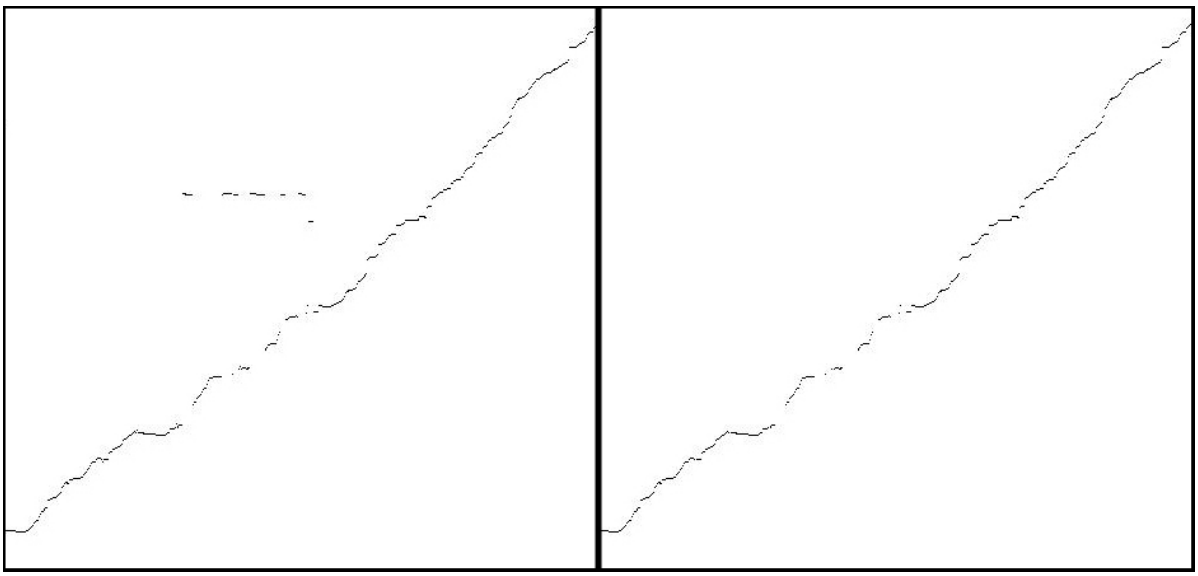

Figure 4.11.: Removing the remnants of a bright scale line by applying the Define method.

**may be missing due to them not being detected in earlier phases of the process because of high intensity unwanted objects in the original image.**

**Because the define method clearly identifies which rows contain data line pixels, the Identify3 method (A.13.3 on page 113) can be applied to accurately extract only that set of rows that contain the data lines, from the original image. This cropped version of the original will contain only a fragment of the unwanted image objects of the full sized original image. The smaller image will also greatly reduce processing time. This smaller image is used as input for the next phase of the process (Figure B.11.1 on page 135).**

## 4.6. Accurate data identification

**The entire data identification process is repeated using the cropped version of the original image that only contains the data lines. All unwanted image objects that are not among the data lines have thus been removed. The absence of these objects greatly improves the results of the process.**

**The variables used in the 6 iterations of the Scan phase are as follows:**

Iteration 1: Scan tolerance = 10 and Mark row population = 7.5% (Figure B.11.2 on page 135);

Iteration 2: Scan tolerance = 10 and Mark row population = 7.5% (Figure B.11.3 on page 135);

Iteration 3: Scan tolerance = 5 and Mark row population = 5% (Figure B.11.4 on page 135);

Iteration 4: Scan tolerance = 5 and Mark row population = 5% (Figure B.11.5 on page 135);

Iteration 5: Scan tolerance = 0 and Mark row population = 2.5% (Figure B.11.6 on page 136);

Iteration 6: Scan tolerance = 0 and Mark row population = 2.5% (Figure B.11.7 on page 136);

**The results of the Scan and clean methods at the end of the accurate data identification phase can be seen in Figures B.11.8 and B.11.9 on page 136.**

**The increased sensitivity of the Mark method is due to the fact that the output images at this stage of the process will contain much less horizontal lines, because mostly pixels within the data lines will be marked by the Scan method. So any horizontal lines that are detected will have lower row populations than when scanning the original full sized image. Thus higher sensitivities are needed to detect them.**

## 4.7. Accurate data extraction and binarization

**The entire data extraction and data binarization process is repeated on the output of the data identification process.**

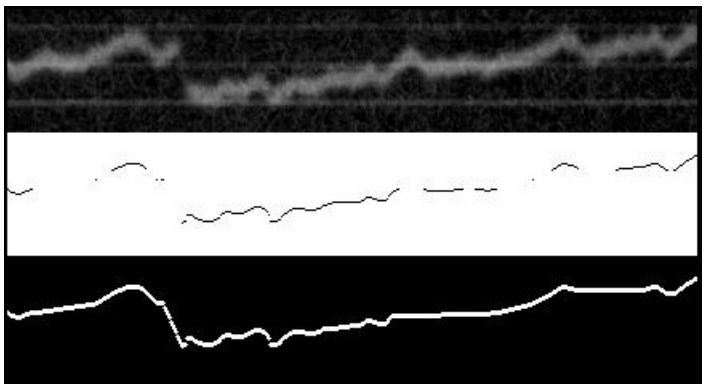

Figure 4.12.: The effect of the Plot method on a set of disconnected data line segments. Top: Original data line. Middle: Extracted data line. Bottom: Output of Plot method.

**The results of accurate data extraction can be seen in Figures B.12.1, B.12.2, B.12.3 ad B.12.4 on page 137. The results of accurate data binarization can be seen in Figures B.13.1, B.13.2, B.13.3 ad B.13.4, B.13.5, B.13.6 and B.13.7 on pages 138 and 139.**

**The result is an accurate binary image containing the data lines. Some disconnectivity may still occur, which is rectified in the post processing phase.**

## 4.8. Post-processing

**The Plot method (A.19 on page 118) is applied to ensure that there is a marked pixel in each column of the image (Figure B.14.1 on page 140). If two segments of the data line are disconnected, then it would be safe to assume that the small gap between them would follow a similar gradient. So visually connecting these segments by marking pixels at appropriate positions in the empty columns between them is an acceptable solution. The Plot method also connects the last pixel of one data line with the first pixel of the following data line, completing the plotted version of the results as shown in Figure 4.12.**

**The Paste method (A.20 on page 119) is called to append empty rows to the top and bottom of the resulting plot image so the final image will be exactly the same size as the original (Figure B.14.2 on page 140).**

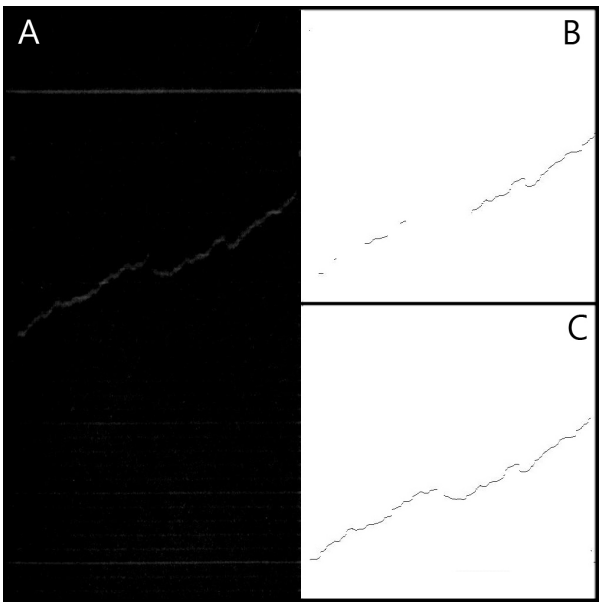


Figure 4.13.: Difference between applying the binarization process once and twice. (A) A segment of the original image. (B) Extracted data after single application. (C) Extracted data after double application.

**The number of rows appended depends on the output of the Crop and Identify3 methods. This makes the task of visually comparing the results of the process much easier.**

**Although applying the process twice may seem redundant, this redundancy ensures the greatest accuracy and prevents lost data line segments. This is shown in Figure 4.13.**

## 4.9. Process results

**The resulting images can be seen in Figures 4.14 through 4.19. To improve visibility, only the first eight hours of each data image is shown. An array containing the numerical value of each marked pixel is also included in the output.**

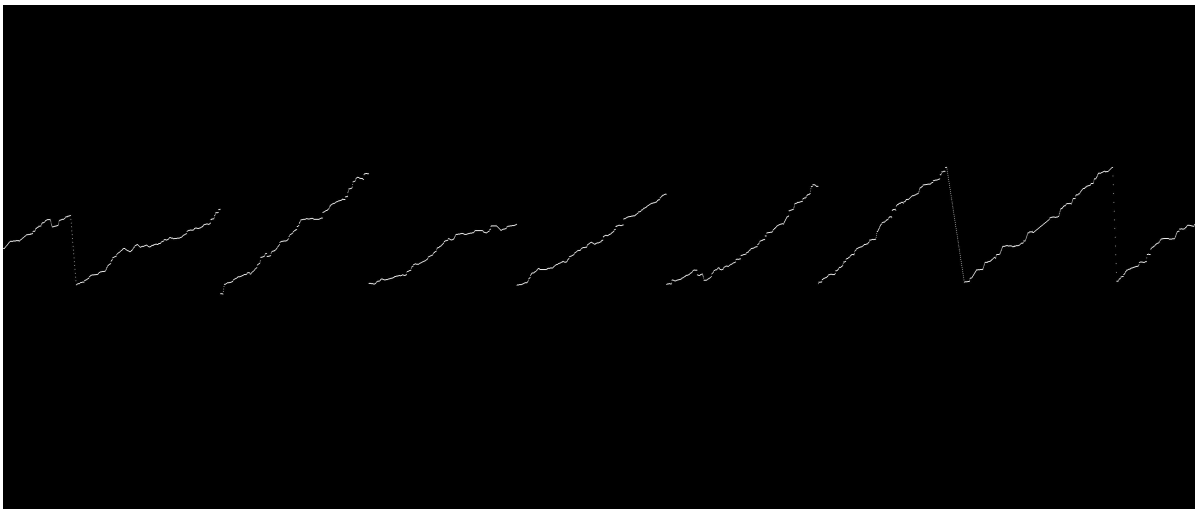

Figure 4.14.: The data extracted from image A.

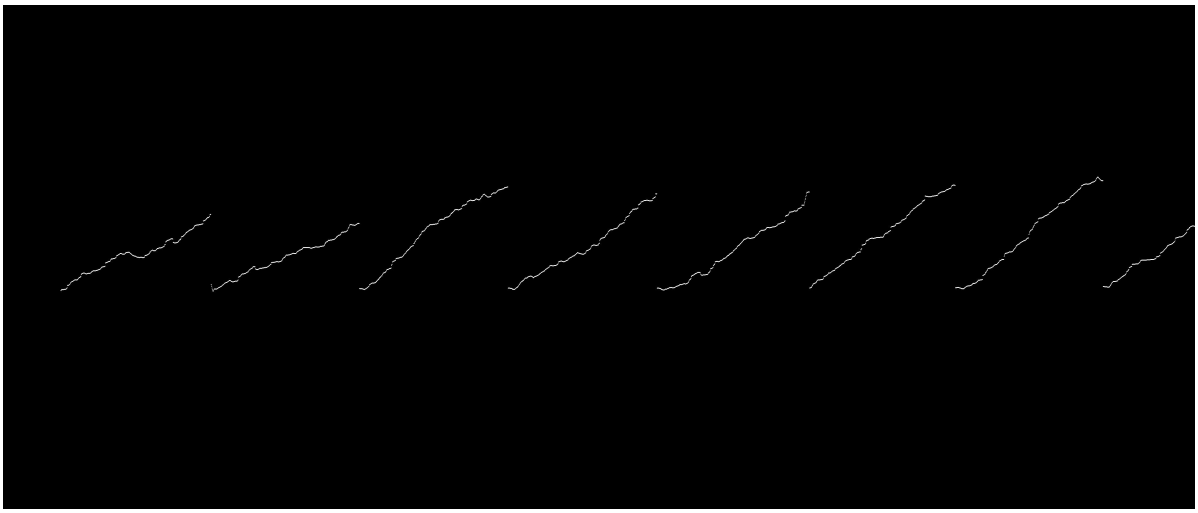

Figure 4.15.: The data extracted from image B.

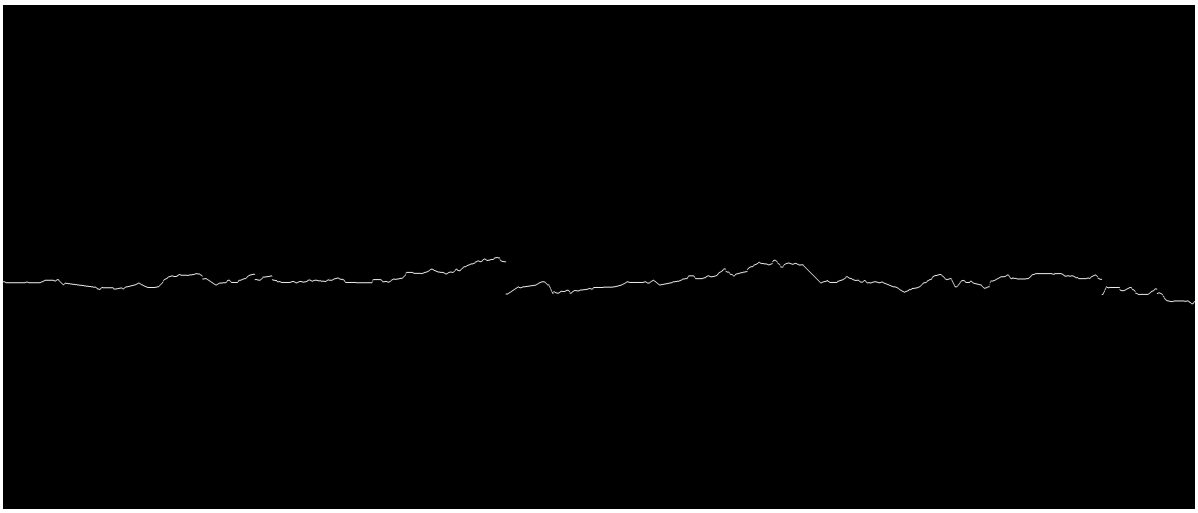

Figure 4.16.: The data extracted from image C.

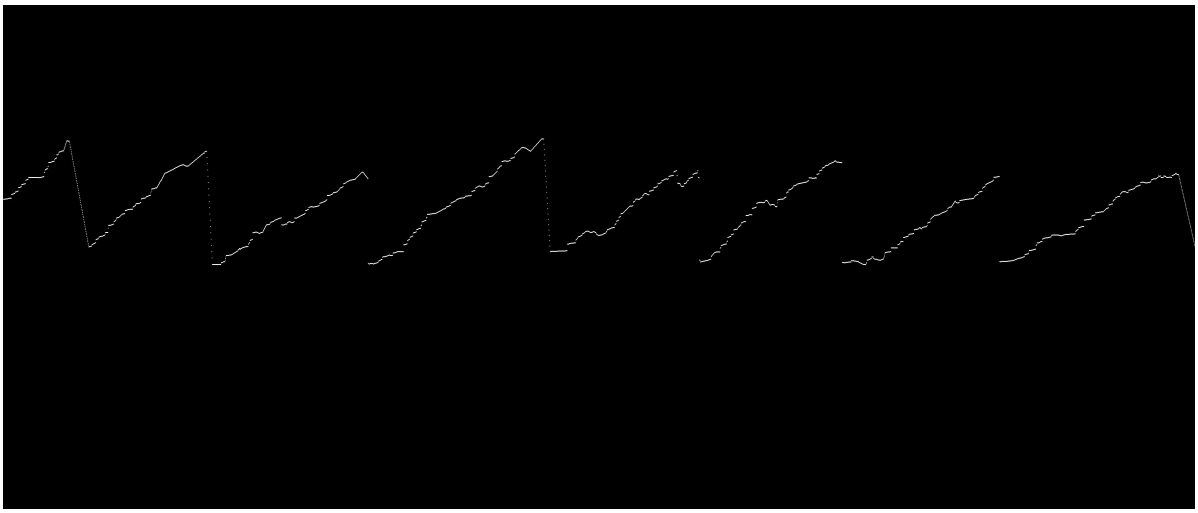

Figure 4.17.: The data extracted from image D.

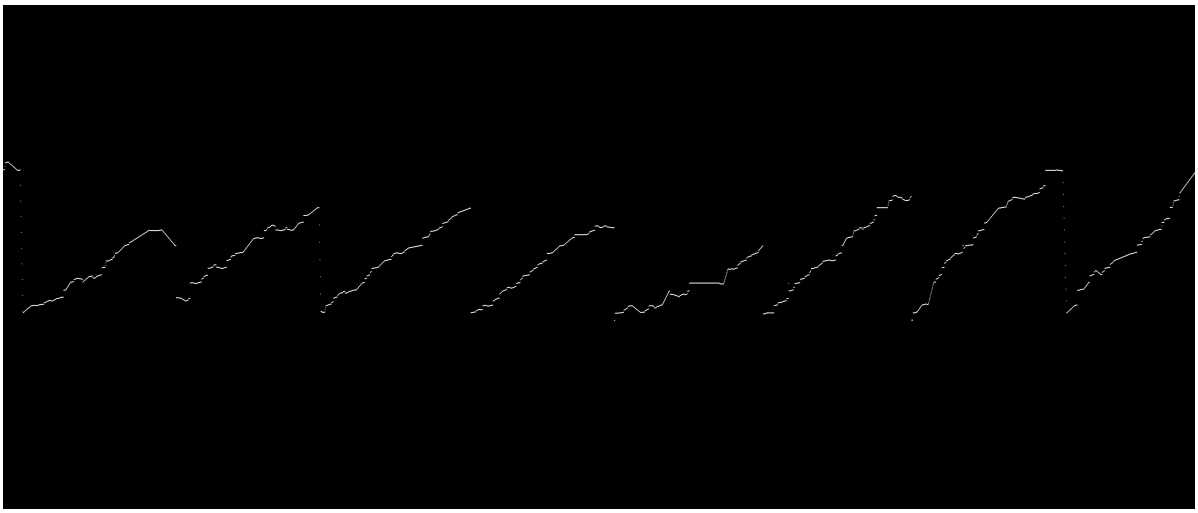

Figure 4.18.: The data extracted from image E.

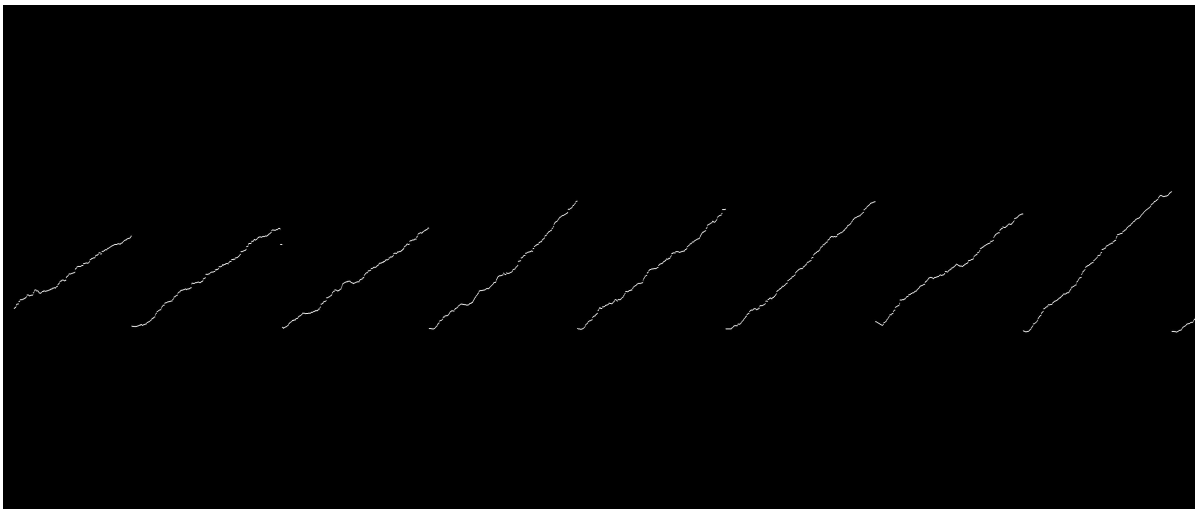

Figure 4.19.: The data extracted from image F.