

References

- Adelmann, H., 1998. Butterworth equations for homomorphic filtering of images. *Computers in Biology and Medicine*, 28(2): 169 - 181.
- Aiello, M., 2002. Document understanding for a broad class of documents. *International Journal on Document Analysis and Recognition*, 5: 1 - 16.
- AIM, 2000. Optical character recognition.
URL <http://www.aimglobal.org/technologies/othertechnologies/ocr.pdf>
Date of use: 9 August 2012.
- Anchordoqui, L., Paul, T., Reucroft, S. & Swain, J., 2003. Ultrahigh energy cosmic rays: The state of the art before the auger observatory. *International Journal of Modern Physics*, 18(13): 2229 - 2366.
- ARPANSA, 2012. Understanding radiation.
URL <http://www.arpansa.gov.au/radiationprotection/basics/understand.cfm>
Date of use: 25 June 2012.
- Benett, R., Compton, A. & Wollan, E., 1934. A precision recording cosmic ray meter. *The Review of Scientific Instruments*, 5: 415 - 422.
- Beucher, S. & Meyer, F., 1993. The morphological approach to segmentation: the watershed transformation. In: Dougherty, E.R., ed. *Mathematical Morphology in Image Processing* 34, 433 - 481.
- Burns, N. & Grove, S., 1993. *The practice of nursing research: conduct, critique and utilization*. W.B. Saunders Company.
- Canada, N. R., 2008. Image interpretation & analysis.
URL <http://www.nrcan.gc.ca/earth-sciences/geography-boundary/remote-sensing/fundamentals/1362> Date of use: 9 August 2012.
- Cesarini, F., Francesconi, E., Gori, M. & Soda, G., 1999. A two-level knowledge approach for understanding documents of a multi-class domain. In: *ICDAR '99*. pp. 135 - 138.

References

- Doermann, D., Liang, J. & Li, H., 2003. Progress in camera-based document image analysis. Proceedings of the Seventh International Conference on Document Analysis and Recognition held in Edinburgh, Scotland, 1: 606 - 617.
- Drevin, G. R. & Du Plessis, T., 2009. Evaluating methods to binarize historic cosmic-ray recordings. In: EUROCON 2009. pp. 1386 - 1391.
- Drevin, G.R., 2008. Adaptive frequency domain filtering of legacy cosmic ray recordings. In: 9th International Conference on Computer Vision, Pattern Recognition and Image Processing (CVPRIP 2008), in conjunction with the 11th Joint Conference on Information Sciences.
- Du Plessis, T., 2010. Die versyfering van historiese kosmiese straal-data. Masters thesis, North-West University, South-Africa.
- Ejiri, M., Kashioka, S. & Ueda, H., 1984. The application of image processing technology to industrial automation. *Computers in Industry*, 5(2): 107 - 113.
- Gatos, B., Pratikakis, I. & Perantonis, S., 2006. Adaptive degraded document image binarization. *Pattern Recognition*, 39: 317 - 327.
- Goel, P. & Vidakovic, B., 1995. Wavelet transformations as diversity enhancers. Discussion paper, Institute of Statistics & Decision Sciences, Duke University.
- Gonzalez, R. & Woods, R., 2008. Digital image processing. New Jersey, Pearson Prentice Hall.
- Gurevich, A. & Zybin, K., 2005. Runaway breakdown and the mysteries of lightning. *Physics Today*, 58 (5): 37 - 43.
- Hardy, S. J., 2006. Legacy cosmic-ray data 1935-1960. Washington, DC, Department of Terrestrial Magnetism.
- Heidenreich, P., Cirillo, L. & Zoubir, A., 2009. Morphological image processing for FM source detection and localization. *Signal Processing*, 89(6): 1070 - 1080.
- Hossack, W., 2011. The Fourier Transform, What you need to know: Symmetry of Fourier Transform. URL <http://www2.ph.ed.ac.uk/~wjh/teaching/Fourier/>
Date of use: 9 August 2012.

References

- Jain, A., Ratha, N., Trier, O., Yu, B. & Zhong, Y., 1995. Document processing research at Michigan State University. In: Proceedings of the Symposium on document image understanding held in Bowie, Maryland. Center of automation research, pp. 126 - 140.
- Lange, I. & Forbush, S., 1948. Cosmic-ray results from Huancayo observatory, Peru, June, 1936 December, 1946. Including summaries from observatories at Cheltenham, Christchurch, and Godhavn through 1946. *Researches of the Department of Terrestrial Magnetism*, 14: 182.
- Lorentz, M., 1905. Methods of measuring the concentration of wealth. *Journal of the American Statistical Association*, 9: 205 - 219.
- Mak, K., Peng, P. & Yiu, K., 2009. Fabric defect detection using morphological filters. *Image and Vision Computing*, 27(10):1 585 - 1592.
- Michigan, 2009. Best practices for the capture of digital images from paper or microfilm. State of Michigan. Records Management Services
- Nagy, G., Kanai, J. & Krishnamoorthy, M., 1988. Two complementary techniques for digitized document analysis. In: ACM Conf. Document Processing systems. pp. 169 - 176.
- Niblack, W., 1986. An Introduction to digital Image Processing. New Jersey, Prentice Hall.
- Niczyporuk, M. & Miller, D., 1991. Automatic tracking and digitization of multiple radiopaque myocardial markers. *Computers and Biomedical Research*, 24(2): 129 - 142.
- NWU, 2009. Neutron monitor data.
URL http://www.puk.ac.za/fakulteite/natuur/nm_data/data/nm_de.html
- Otsu, N., 1979. A threshold selection method from gray-level histograms. *IEEE Transactions on systems man and cybernetics*, 9: 62 - 66.
- Paine, S. & Lodwick, G., 1989. Edge detection and processing of remotely sensed digital images. *Photogrammetria*, 43(6): 323 - 336.
- Prewitt, J. M. S., 1970. Object enhancement and extraction. *Picture Processing and Psychopictorics*: 75 - 149.

References

- Realclimate, October 2006. Taking cosmic rays for a spin.
URL <http://www.realclimate.org/index.php/archives/2006/10/taking-cosmic-rays-for-a-spin> Date of use: 25 June 2012.
- Roberts, L. G., 1963. Machine perception of three-dimensional solids. Ph.D. thesis, Massachusetts Institute of Technology.
- Sauvola, J. & Pietikäinen, M., 2000. Adaptive document image binarization. *Pattern Recognition*, 33: 225 - 236.
- Serra, J., 1988. Image analysis and mathematical morphology. New York, *Academic Press*, 2.
- Simpson, J., 2000. The cosmic ray nucleonic component: The invention and scientific uses of the neutron monitor. *Space Science Reviews*, 93: 11 - 32.
- Simpson, J. A., 1983. Elemental and isotopic composition of the galactic cosmic rays. *Annual Reviews of Nuclear and Particle Science*, 33: 323 - 381.
- Sobel, I. E., 1970. Camera models and machine perception. Ph.D. thesis, Stanford University.
- Taneja, H., 2008. Advanced engineering mathematics. Vol. 2. New Delhi, India: I. K. International Pvt Ltd, chapter 18: Fourier integrals and Fourier transforms.
- Valizadeh, M. & Kabir, E., 2012. An adaptive water flow model for binarization of degraded document images. *International Journal of Document Analysis and Recognition*, pp.1 - 12.
- Ziegler, J. F., Curtis, H. W., Muhlfeld, H. P., Montrose, C. J., Chin, B., Nicewicz, M., and others., 1996. IBM experiments in soft fails in computer electronics (1978-1994). *IBM Journal of Research and Development*, 40: 3 - 18.

A. Method description

This appendix describes the methods that are applied during the binarization process. Each method will be referred to by its descriptive name. Input and output images that are used as masks in subsequent methods are referred to as: $M_{Method\ name}$. Input and output images that are not used as mask are referred to as: $I_{Method\ name}$. Empty images used within methods are referred to as: $E_{Method\ name}$.

While some variables will be named in this appendix, the values of those variables and the reasons behind those values are discussed in Chapter 4.

A.1. Crop

At the top and bottom of most of the images there is an empty white area. These areas contain no data and are removed from the image so no time is wasted processing the empty space. This method crops the original image $I_{Original}$ so the entire image is occupied by the recording paper.

The image is roughly binarized by assigning a value of 0 to all pixels with a value below the threshold of 200 and a value of 255 to all pixels with an intensity value above that threshold, as shown in Eq. A.1.

$$I_{BinaryOriginal}(x, y) = \begin{cases} 0 & \text{if } I_{Original}(x, y) < 200 \\ 255 & \text{if } I_{Original}(x, y) \geq 200 \end{cases} \quad (A.1)$$

The binarized image is then scanned row by row starting at the top of the image in the first iteration and then starting at the bottom of the image on the second iteration. If the mean intensity of the entire row is below a second variable threshold, then the position of that row is recorded.

The threshold value of 200 is chosen to compensate for any shadows that may occur along the edge of the data strip within the image.

A buffer value of 20 is added to the vertical position of the identified rows to compensate for any skew that occurs within the image. The image is then cropped to contain only the scanned paper plot between these rows. The output of this method is the cropped image $I_{Cropped}$ as well as the values of the upper and lower boundaries, T_1 and B_1 , respectively.

A.2. Line

The brightest and straightest horizontal line in the cropped image represents the temperature of the ionization chamber during the recording process. This method identifies the rows of the original image $I_{Original}$ that contain that line.

This is done by creating a plot that indicates the number of pixels, which have higher intensity values than the average intensity of the image, in each row of the image. This plot is smoothed by replacing the value of each row with the average value of that row and the two rows above and below it. This is done to prevent a single outlier row from being identified as the temperature line.

The highest peak in the resulting plot represents the set of rows that probably contain the temperature data line. The row with the highest value within this set represents the center of the temperature data line.

Some images do not contain a clearly visible temperature line, while others are so bright that a large number of rows have such high intensities that the position of the line cannot be clearly identified. In such cases, the probable position of the temperature data line is given as output.

The output of this method is an integer value L representing the position of the row within the image that contains the center of the temperature data line.

A.3. Fill

The sprocket holes within the image need to be removed so that these will not affect the accuracy of subsequent methods. Normally the sprocket holes are the only areas within the image that have high enough intensities to be detected by this method, although such bright areas do occur amongst the data in brighter images. This method removes those areas along with the sprocket holes.

The original image $I_{Original}$ is searched using a mask that is small enough to completely fit inside a sprocket hole, even sprocket holes which appear smaller because of shadows along the edges. These shadows appear wherever the original data strip was not held flat against the glass of the digital scanner. The mask is twice as high as it is wide, to properly fit inside the rounded rectangular shape of a sprocket hole.

The image is zero padded to ensure that no pixels within the searching mask fall outside the image while searching its edges and while removing the sprocket holes.

While searching the image, the mean of the mask sized area around each pixel is calculated and compared to a threshold value. If the mean is above the threshold then the entire mask is marked on a second empty binary image E_{Fill} of the same size as the padded input image.

Each pixel that was marked in the empty image E_{Fill} is assigned a filler value within the input image $I_{Original}$. The filler value of 0 (black) is used, as shown in

$$I_{Fill}(x, y) = \begin{cases} 0 & \text{if } E_{Fill}(x, y) = 255 \\ I_{Original}(x, y) & \text{if } E_{Fill}(x, y) = 0 \end{cases} \quad (A.2)$$

The output of this method is the filled image I_{Fill} , without the additional padding.

A.4. Blur

This method is used to decrease the visibility of any noise within the image by applying a mean filter. The method also serves to smooth any lines within the image.

The image is zero padded according to the size of the filter mask to ensure that no part of the mask ever falls outside the image.

This method takes an image I_{Fill} as input along with an integer to indicate the size of the mean filter mask. A 3 by 3 pixel mask provides the best results.

The output of this method is a blurred version I_{Blur} of the input image.

A.5. Scan

This method identifies the pixels that have the highest intensity values in each column of the image matrix. The method takes an integer as argument to specify the tolerance t of the scanning procedure.

The maximum value of each column is determined and each pixel within the column, with an intensity value that falls within the tolerable difference from the maximum value t , has its position marked in an empty image E_{Scan} of the same size as the input image.

It may occur in some darker images that the maximum value within a column is less than the tolerance t . If this occurs then all pixels in that column, including pixels with a value of zero, are recorded to the empty image. This causes the entire column to be marked, which results in data being lost at later stages.

To prevent this from happening, each maximum value is checked to verify if it is larger than the specified tolerance t . If it is not, then the tolerance for that column is set to pass all pixels in the column, except pixels with a value of zero. When the column is dark enough to require this step, most of the pixels in the

A. Method description

column have a value of zero. So by excluding only zero valued pixels, the ideal fewest number of pixels are still marked within that column.

This method can be summarized by the following equation:

$$I_{Scan}(x, y) = \begin{cases} 0 & \text{if } I_{Blur}(x, y) \geq y_{\max} - t \\ 255 & \text{otherwise} \end{cases} \quad (\text{A.3})$$

where I_{Scan} represents the scanned image, I_{Blur} the input image, t the specified tolerance of that iteration and $y_{\max}$ the maximum intensity value of the current column being scanned.

The output of this method is a mask that identifies the pixels that have a high probability of representing data in the original image.

A.6. Remove

This method receives the output of the Scan method I_{Scan} as input.

Some images have such a low intensity that some of the matrix columns are entirely populated by pixels with intensity values of zero. These columns are then added to the scanned image as vertical black lines. All pixels within the scanned image are supposed to have a high probability of being part of the data lines within the image. Thus, these vertical black lines need to be removed.

The Remove method inspects each column of the image matrix and replaces all pixels in columns that are entirely black by white pixels, thus eliminating the effect of the vertical lines in later stages of the process.

The output of this method is a scanned image I_{Remove} without any vertical black lines spanning the height of the image.

A.7. Mark

The purpose of the entire process is to extract the data from the image, specifically the cosmic ray data lines. The pixels representing these lines are usually the brightest in the image.

However, the horizontal data line representing the temperature of the ionization chamber, which remained near constant during the recording process, is also one of the brightest objects in the image. This line is also generally thicker than the cosmic ray data line, making it more likely to be detected by the Scan method, than the cosmic ray data line.

In images with very high average intensities, the scale lines are also often identified by the Scan method instead of the data lines. This results in large segments of straight horizontal lines being identified by the Scan method and only small segments of the actual data.

This method marks the positions of these lines. The method receives the output of the Remove method I_{Remove} as well as an integer value indicating the percentage of a row in the image matrix that must be filled with zeros (black pixels) for that row to be recognized as a line. This percentage must be large enough to ensure that rows containing a large number of disconnected black pixels are not wrongly classified as image wide lines, but at the same time small enough that slightly straight data lines are not marked and erased in later stages.

The image is scanned row by row and if the number of zero valued pixels in a row exceeds the specified cutoff percentage, then that entire row is marked in an empty image of the same size as the input image. The rows are marked by white lines in a black image. This marked mask image M_{Mark} is the output of this method.

A.8. Erase

The output of the Mark method M_{Mark} is used as input for this method along with the output of the Blur method I_{Blur} .

A. Method description

This method scans the output image of the Mark method row by row. If a row is marked then all the pixels in the corresponding row in the blurred image have their intensity values set to a filler value. This filler value F is the intensity mean of the image where the lines are being erased, in this case the blurred image. This filler value F is also applied to the upper and lower neighbor of each corresponding marked pixel, to compensate for the jagged edges of the lines in the image.

This method can be summarized by the following equation:

$$I_{Erase} = \begin{cases} F & \text{if } M_{Mark}(x, y) = 255 \\ IO_{Blur}(x, y) & \text{otherwise} \end{cases}$$

where I_{Erase} represents the image from which lines are to be removed, M_{Mark} the mask output by the Mark method, IO_{Blur} the blurred version of the original data image and F represents the mean value of the blurred image.

When the Scan method is applied to the resulting image a second time, all the high intensity horizontal lines that have been detected will have been replaced by the image mean value, ensuring that they will not be detected again. This allows more of the actual data line pixels to be identified and sent on to subsequent steps in the process.

A.9. Clean

This method receives a binary image as input, in this case the output of the Scan method I_{Scan} . The purpose of this method is to remove small groups of pixels that do not form part of any large image object and are probably not part of the roughly extracted data line.

The method zero pads the input image and counts the number of extracted pixels in a 5 by 5 pixel mask around each pixel in the image as well as the number of pixels in the single pixel wide border around the 5 by 5 pixel mask.

If the number of extracted pixels in the mask is less than five and there are no extracted pixels in the border surrounding the mask, then the small set of pixels probably do not represent a piece of the data line and are removed (replaced by white pixels with a value of 255).

The same process is applied to each individual pixel and if a pixel is not connected to any neighbor then it is also removed.

The purpose of this method is to decrease the number of pixels, in a rough estimation of the foreground of the image, that do not form part of the data that the process is attempting to extract.

The output of this method is the cleaned image M_{Clean} .

A.10. Target

This method receives a binary image from the Clean method M_{Clean} , which indicates the positions of pixels that have a high probability to form part of the data line.

An empty image is created of the same size as the original image. The binary input image M_{Clean} is used as a mask and the 31 by 61 pixel area around each black pixel in the binarized image is marked in the empty image I_{Target} .

Due to the general shape of the data lines, the windows used are twice as high as they are wide. Because an ideal binarized image would have at least one black pixel in every column while some rows are supposed to be empty, this window size ensures good connectivity between windows even when the actual data lines in the original image have a very steep gradient. This window size also promotes connectivity between windows around probable data line pixels while limiting connectivity between more widely spaced windows around pixels that probably do not form part of the data line.

The output of this method is a mask I_{Target} which contains marked areas that indicate areas that have a high probability of containing data in the original image.

A.11. Connect

This method receives the binary image I_{Target} from the Target method as input.

The white vertical rectangles that were marked in the Target method should form several connected components. The largest of these components represent the areas in the original image that contain the data lines.

This method assigns a label to each connected component. A 4-neighbor connected component labeling algorithm is applied to the image to identify any connected components. Each component is given a label with the first label starting at 1.

The output of this method is an image of labeled connected components $I_{Connect}$ as well as the number of labeled components C .

A.12. Scrub

This method receives the output of the Connect method $I_{Connect}$ as input along with the number of labeled components C .

The input image contains connected components. The largest of these components represent data lines. The medium sized components may represent non-connected pieces of data lines or an unwanted image object that has not been completely removed. The small connected components mostly represent unwanted image objects that have not been removed. This method removes those small components from the image.

The number of pixels within each connected component is counted and the average pixel count of the connected components is calculated. All connected components that contain more pixels than the average number are passed to the next method.

The average pixel count of the remaining connected components is calculated and 1.5 times that number is taken as the new cutoff value. All connected components that contain more pixels than this value are passed to the next method while the rest are discarded. This results in all the small connected components that probably do not form part of a data line to be removed from the image.

The output of this method is a mask image I_{Scrub} containing a set of large connected components that mostly represent data lines.

A.13. Identify

This method identifies the vertical area within an image that contains data of interest, discarding all rows above or below this area.

The method plots the number of pixels, that do not form part of the background, in each row. This plot is then smoothed by replacing the count value of each row by the average of row counts within a certain distance from that row. This allows connectivity between peaks in the plot that are separated by a single empty row or two. By smoothing the plot, the effect of outlier rows with very high populations is also decreased.

The plot is then analyzed to determine which peak is the widest (contains the most rows) and which peak is the highest (contains the most pixels). In most cases the peak of interest will be the highest and the widest. All rows in this peak are then passed to the next method while the rest are discarded. If the highest and widest peaks are not the same, then all the rows contained in both are passed while all other rows are discarded.

This method eliminates unwanted objects that do not fall within the data bearing area of the image.

There are 3 variations of this method.

A.13.1. Identify1

This method receives the output of the Scrub I_{Scrub} method as input along with an integer indicating the number of rows to be used when calculating the average row values while smoothing the row population plot during the process. The background of this input image has a value of 0 while the connected components within it have values of 1 or more.

The output of this method is an image $I_{Identify1}$ which contains only the connected components within the vertical area of the image that contains the data lines. All the discarded rows are replaced by black rows.

A.13.2. Identify2

This method receives the output of the Erase I_{Erase} method as input along with an integer indicating the number of rows to be used for calculating the average row values while smoothing the row count plot during the process. The background of this input image has a value of 255 while the pixels of interest within it have a value of 0.

The output of this method is an image $I_{Identify2}$ which contains only pixels within the vertical area of the image that contains the data lines. All the discarded rows are replaced by white rows.

A.13.3. Identify3

This method receives the output of the Define method $I_{Definet}$ as input along with the blurred version of the original image I_{Blur} . The buffer value with this variation of the method is fixed at 20 rows. This method does not discard rows that fall outside the data bearing area of the input image by replacing them, but

removes those rows completely. These discarded rows are cropped out of the image reducing the vertical size of the image.

The output of this image is an image $I_{Identify3}$ with decreased height that only contains rows that contain a piece of a data line. The vertical lengths of the parts of the image that were cropped out, T_2 and B_2 , are also given as output. This smaller image serves as the new original data image for the second iteration of the data extraction process.

Decreasing the image size in this way reduces processing time and eliminates the effect of unwanted image objects in subsequent stages of the process.

A.14. Bind

After all the unwanted components have been removed by the Scrub and Identify1 methods, there still is a possibility that some of the smaller connected components actually do form part of the data line. This method reinserts those lost components into the image.

This method receives the output of the Identify1 method $I_{Identify1}$, the Connect method $I_{Connect}$, the number of connected components in the output of the Connect method C , and another integer value that specifies the areas next to each connected component that will be searched, in this case that value is 25. Each search area is twenty five pixels wide with the same height as the component being searched. Both the left and right sides of the component are searched.

The specified area around each connected component in the Identify1 output is searched in the Connect output. If a connected component is found within this area that was not already in the Identify1 output, then that component is inserted into the image. It is not required that these reinserted components be assigned labels.

The output of this method is an image M_{Bind} containing connected components corresponding to the positions of most of the data line pixels in the original

image.

A.15. Extract

This method receives the output of the Bind M_{Bind} method as input along with the original data image $I_{Original}$.

This input image now contains the positions of the areas within the original image that contain the data lines. This method extracts only those areas of interest to an empty image.

The output of this method is a mostly black image $I_{Extract}$, of the same size as the original, containing all or most segments of the original image that contain any part of the data lines within them.

This method can be summarized by the following equation:

$$I_{Extract}(x, y) = \begin{cases} I_{Original}(x, y) & \text{if } M_{Bind}(x, y) = 255 \\ 0 & \text{otherwise} \end{cases}, \quad (\text{A.4})$$

where $I_{Extract}$ represents the output image which contains the extracted data lines from $I_{Original}$ and M_{Bind} represents the mask used to extract those data lines.

A.16. Purify

The input of this method is the binary output from the Scan method I_{Scan} .

The purpose of this method is to smooth any jagged lines and remove any single pixel noise within the image.

A. Method description

This is done by searching for any pixels within the image that have 4 empty (white) neighbors. If such a pixel is found then its value is also set to white (255). These single pixels are removed to prevent the following dilation from increasing the effect of noise.

The image is then dilated, using a 4-neighbor filter, to thicken any lines extracted by the Scan method, as this is required by the Designate method.

The output of this method is a dilated version of the input M_{Purify} , without any single pixels that have no 4-connected neighbors.

A.17. Designate

The input of this method is the blurred version of the original image I_{Blur} and the output of the Scan method that has been passed through the Remove and Purify methods M_{Purify} that indicates the positions of pixels in the original image that probably form part of the data line.

This method copies only those marked pixels from the original to an empty image. All other pixels in the image have an intensity value of zero (black) as seen in the following equation:

$$I_{Designate}(x, y) = \begin{cases} I_{Blur}(x, y) & \text{if } M_{Purify}(x, y) = 255 \\ 0 & \text{otherwise} \end{cases} \quad (\text{A.5})$$

The output of this method is a roughly extracted version of data lines from the original image $I_{Designate}$.

A.18. Define

This method receives the binary output of the Identify2 method $I_{Identify2}$ as input, where most of the marked pixels within that image represent extracted data. An integer value representing the size of the initial search area around each pixel

is also received as input. A search area of 50 pixels above, below and to the right of each pixel is assigned throughout the process.

The purpose of this method is to remove all marked pixels in the image that do not form part of a data line. The input of this method contains mostly data line pixels. One characteristic of the data is that it is spread over the entire width of the image and so each column is expected to contain some data pixels in close proximity to the data pixels of the previous column. Using this knowledge, each column is searched and a single pixel is marked as the data pixel of that column.

This is achieved by locating the first marked pixel in the image and labeling it as the primary pixel being investigated. This is the pixel with the smallest X coordinate. A table is created listing all other marked pixels within the image, along with their X and Y coordinates and their distance from the primary pixel. Each pixel (excluding the first pixel) is then assigned a label in the table, identifying it as a pixel within the search area around the pixel being investigated or not. Pixels within the search area are referred to as close proximity pixels.

All the pixels that have been labeled as close proximity pixels have their table entries searched for the pixel with the shortest distance from the primary pixel. This pixel is then passed to the output image while all other pixels between the primary pixel and the newly passed pixel are labeled in the list as removable pixels. This newly passed pixel then becomes the new primary pixel. Those list entries labeled to be removed will not be taken into consideration during subsequent calculations and will not be passed to the output image. These removed pixels represent outlying noise pixels or unwanted document objects.

If there are no pixels within close proximity to the primary pixel, then the remaining pixels outside the initial search area are investigated to identify the set of pixels with the shortest horizontal distance between it and the primary pixel. The pixel in this set with the shortest vertical distance between it and the primary pixel is then passed to the output image and becomes the new primary

pixel. Once again all pixels between the initial primary pixel and the new primary pixel are labeled to be removed.

Prioritizing the pixel search in this way, instead of just searching for the pixel outside the initial search area with the shortest distance from the primary pixel, prevents the method from ignoring pixels at the ends of a data line and wrongly identifying a pixel the middle of a following line as its starting point, thus ignoring large segments of both data lines.

Keeping the search area as small as possible also prevents pixels from unwanted image objects from being marked as the next primary pixel, which would render the output of the method useless.

After a pixel has been passed to the output image and that newly passed pixel then becomes the primary pixel, the list of all remaining pixels is calculated again. These calculations exclude those pixels that are labeled to be removed. These calculations list the pixels' distance from the primary pixel in the table as well as the labels indicating whether each pixel is a close-proximity pixel or not.

The method only searches forward. Once a pixel is passed to the resulting image, all pixels with a smaller X coordinate (behind the newly passed pixel) are labeled to be removed and will not be used again in the method.

By identifying pixels which belong to the data line while removing all other pixels, all unwanted data is removed from the image.

The output of this method is a binary image I_{Define} that contains a relatively small number of pixels that form part of data lines.

A.19. Plot

This method receives the output of the Define method I_{Define} as input.

A. Method description

The purpose of this method is to ensure that there is a marked pixel in every column of the image. This is done by identifying all the marked pixels in the image, calculating the vertical and horizontal distance between them and then accurately marking pixels in the empty columns between every pair of pixels to create a plot that is easy to analyze visually, instead of a set of random pixels throughout the image.

If any unwanted data objects have made it through to this stage, those objects are probably pieces of horizontal lines with very high intensities in the original image. This method removes those lines by creating the plot image from the input image, identifying and removing all horizontal lines above a certain length and all pixels within them from the original input image and finally recreating the plot image from the input without the influence of any pixels belonging to unwanted horizontal lines. If any horizontal line segments do remain at the end of this method, then those lines form part of a relatively straight data line.

The output of this method is an image I_{Plot} that has a single pixel in each column.

A.20. Paste

This method receives the output of the Plot method I_{Plot} as input along with all the heights of all the pieces of the original image that was cropped out during the process (T_1, B_1, T_2, B_2) .

This method appends empty space to the top and bottom of the cropped input so that the final resulting image will be the same size as the original. This is done to simplify the process of visually comparing the results.

The output of this method is an image I_{Paste} , of the same size as the original, which contains the final binary results of the extraction process along with an array of the vertical positions of each pixel in the image, thus the numerical data extracted from the original image.

A.21. Insert

This method receives the output of the Line method L and the Paste method I_{Paste} as input.

The method marks the entire row that is identified by a position derived by the Line method, as well as the four rows above and below that line.

The resulting image I_{Insert} now contains the binarized cosmic ray data and either the marked position of the temperature line, or the most probable position of the temperature line.

B. Visual summary of the binarization process

This appendix is a sequential representation of the binarization process of test image F. The image captions describe the methods that were applied to create that image.

Note that these images are the same five hours taken from every step in the binarization process of the test image. The process was originally applied to the complete fifteen hour image. Some images, which do not show improvements in the quality of the results, for this specific image, are not included in this appendix.

Also note that while some method iterations applied during the binarization process of test image F do not show any improvements, these iterations are necessary to ensure the adaptability of the process and will show improvements when the process is applied to other data images with different characteristics.

B.1. Pre-processing

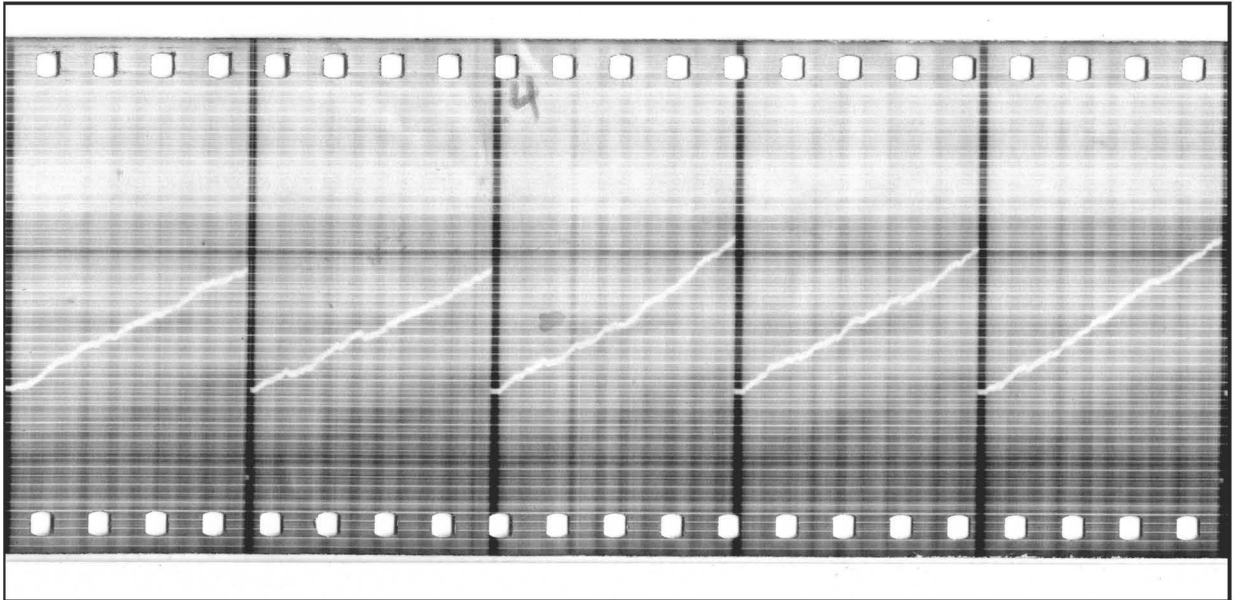


Figure B.1.1.: Original image

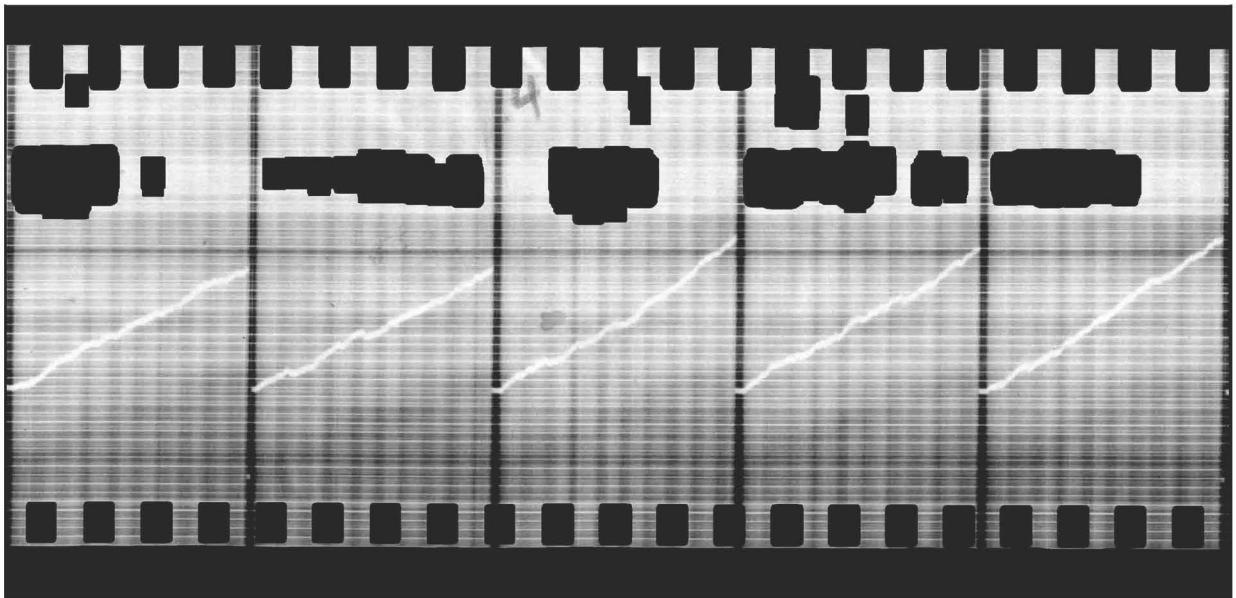


Figure B.1.2.: Filled

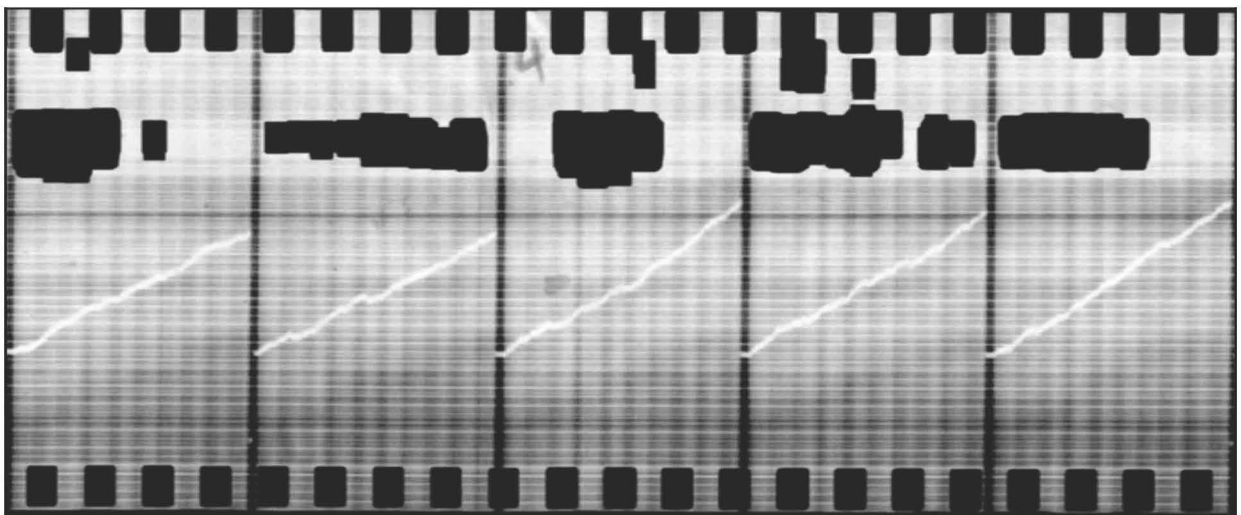


Figure B.1.3.: Cropped and Blurred

B.2. Rough data identification: Iteration 1

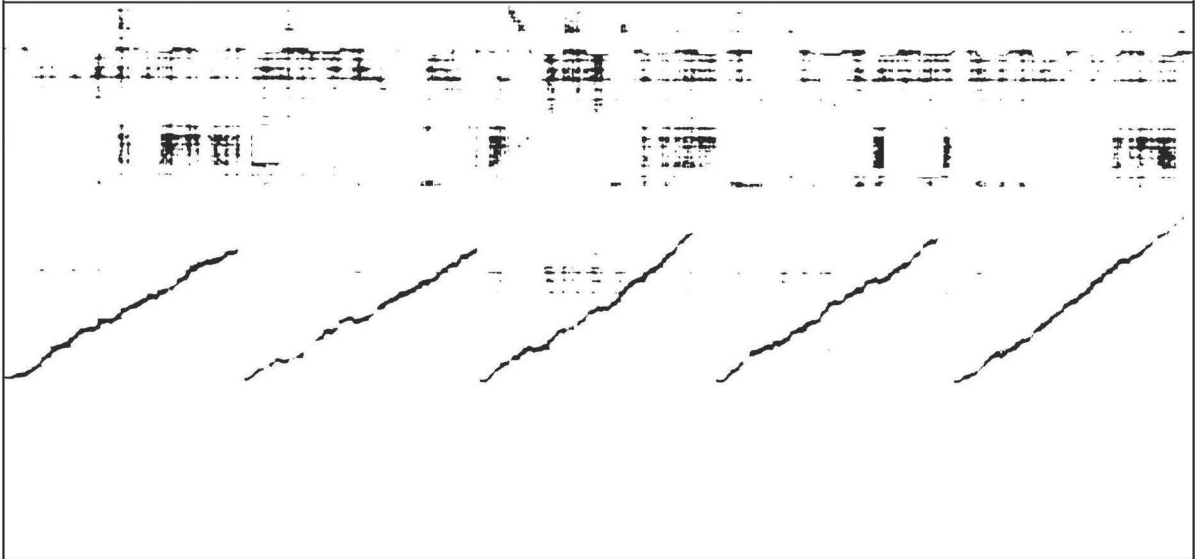


Figure B.2.1.: Scanned and Removed

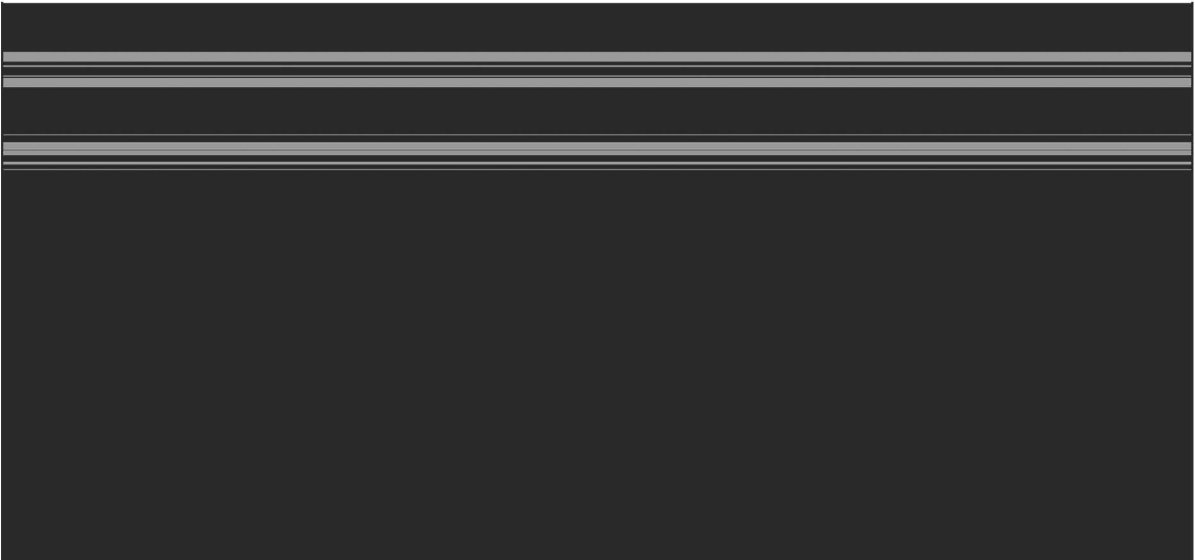


Figure B.2.2.: Marked

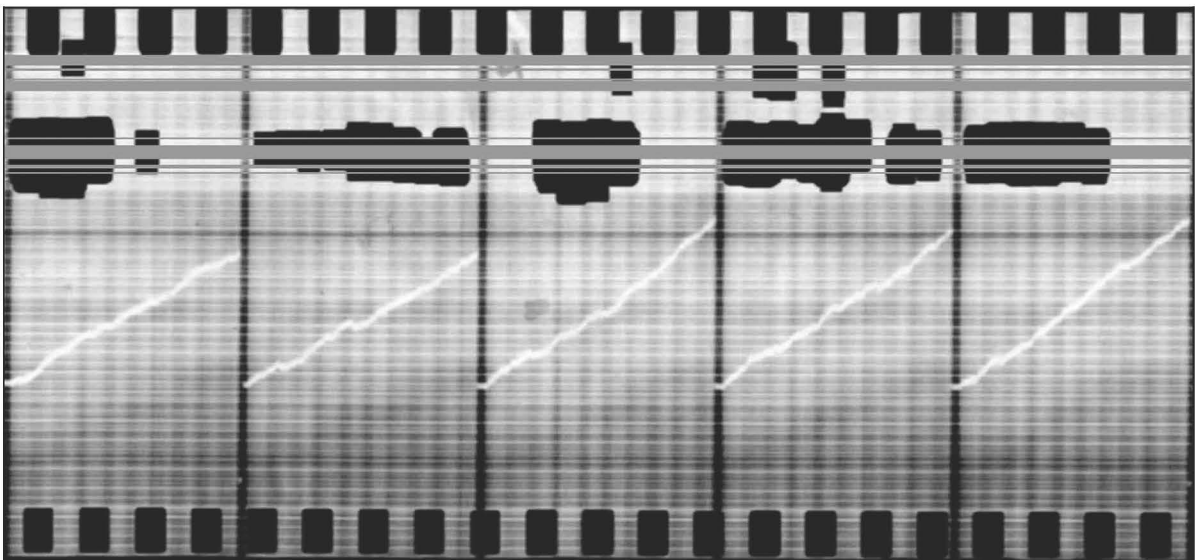


Figure B.2.3.: Erased

B.3. Rough data identification: Iteration 2

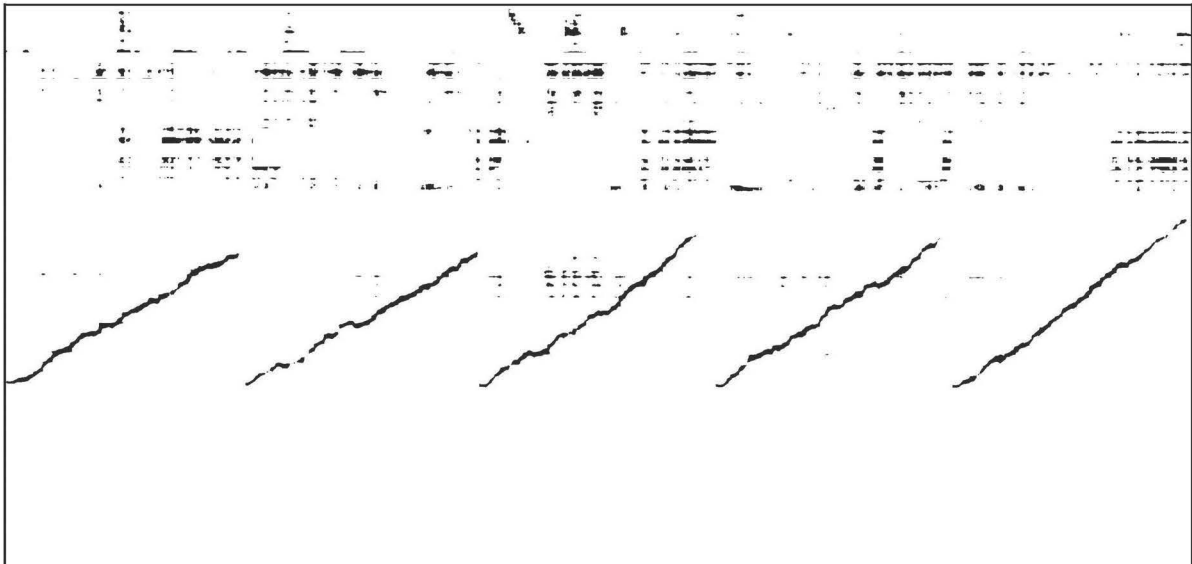


Figure B.3.1.: Scanned and Removed



Figure B.3.2.: Marked

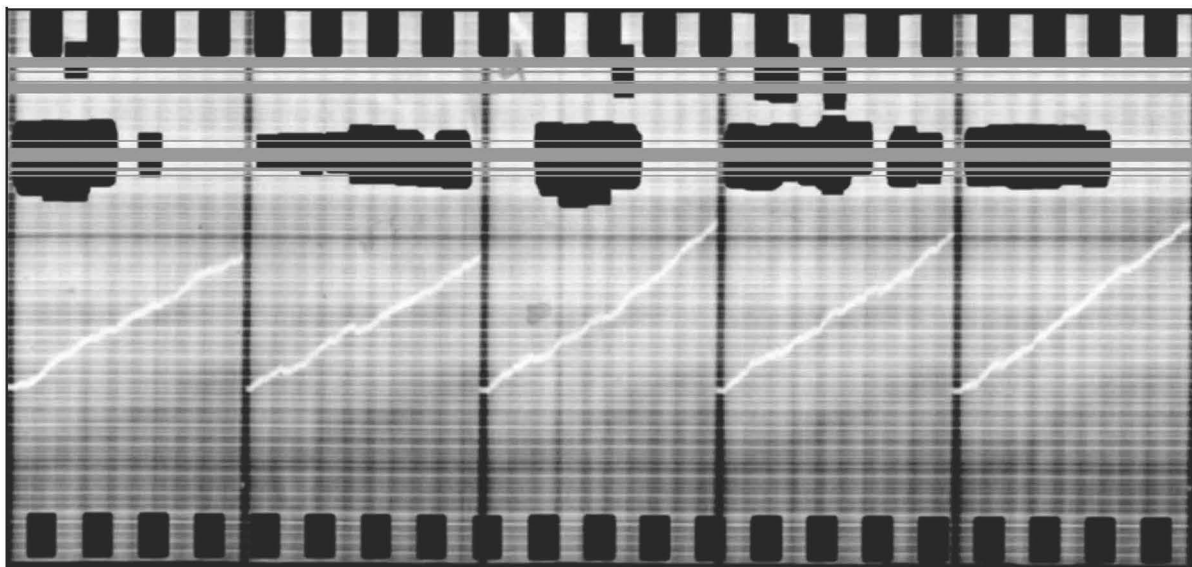


Figure B.3.3.: Erased

B.4. Rough data identification: Iteration 3

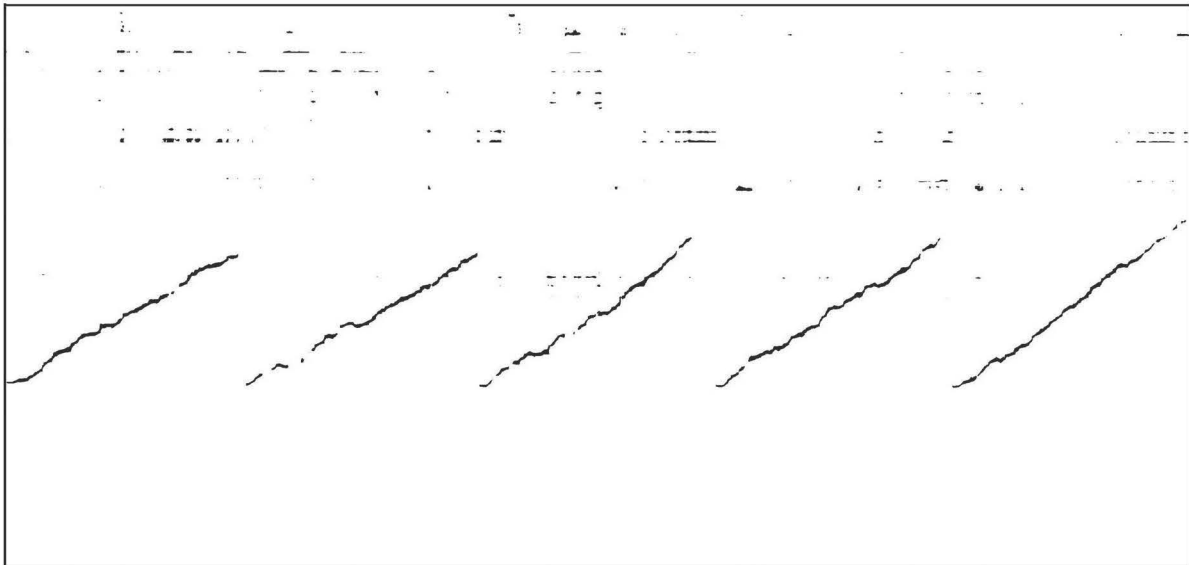


Figure B.4.1.: Scanned and Removed



Figure B.4.2.: Marked

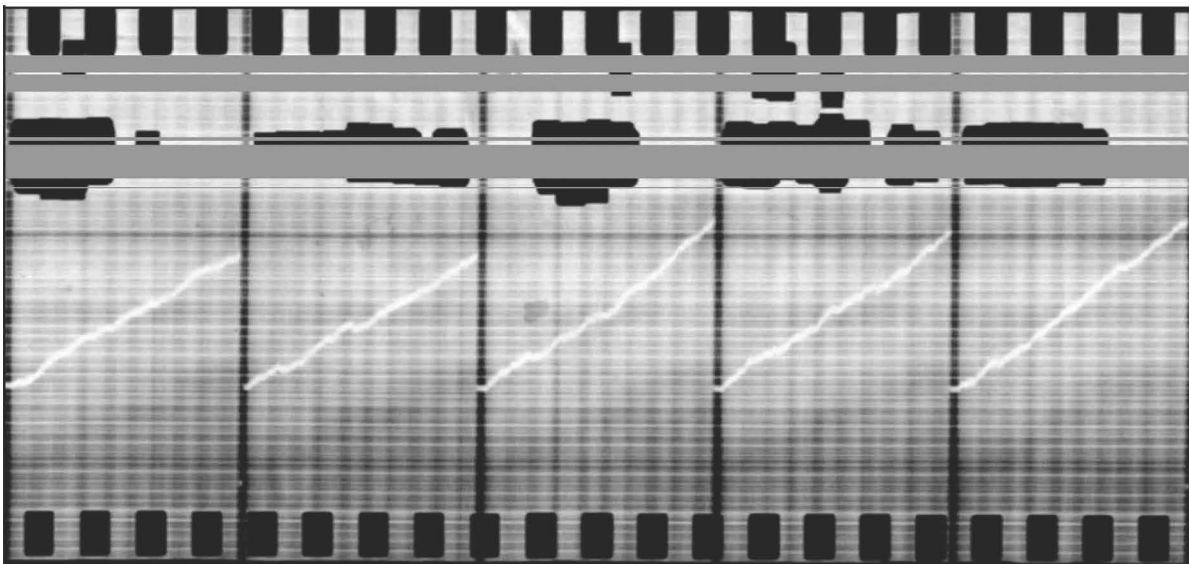


Figure B.4.3.: Erased

B.5. Rough data identification: Iteration 4

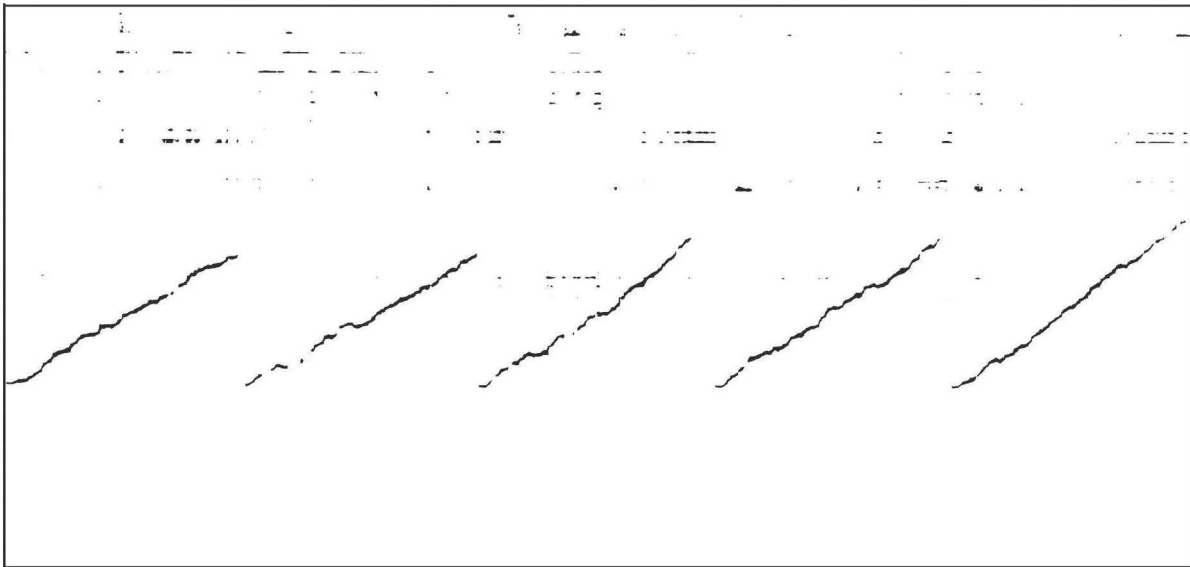


Figure B.5.1.: Scanned and Removed



Figure B.5.2.: Marked

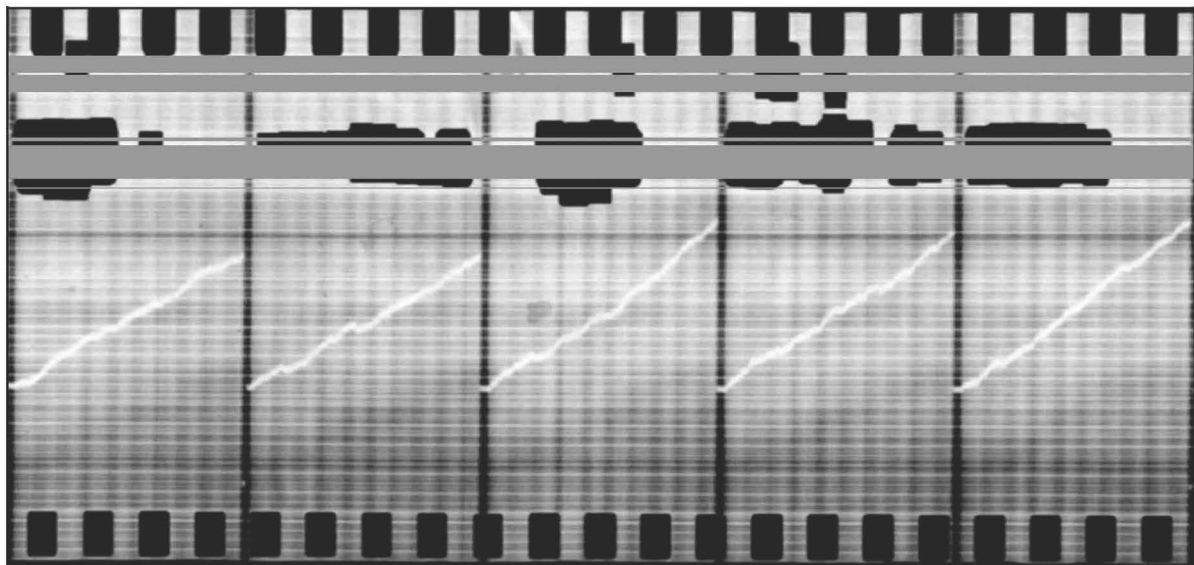


Figure B.5.3.: Erased

B.6. Rough data identification: Iteration 5

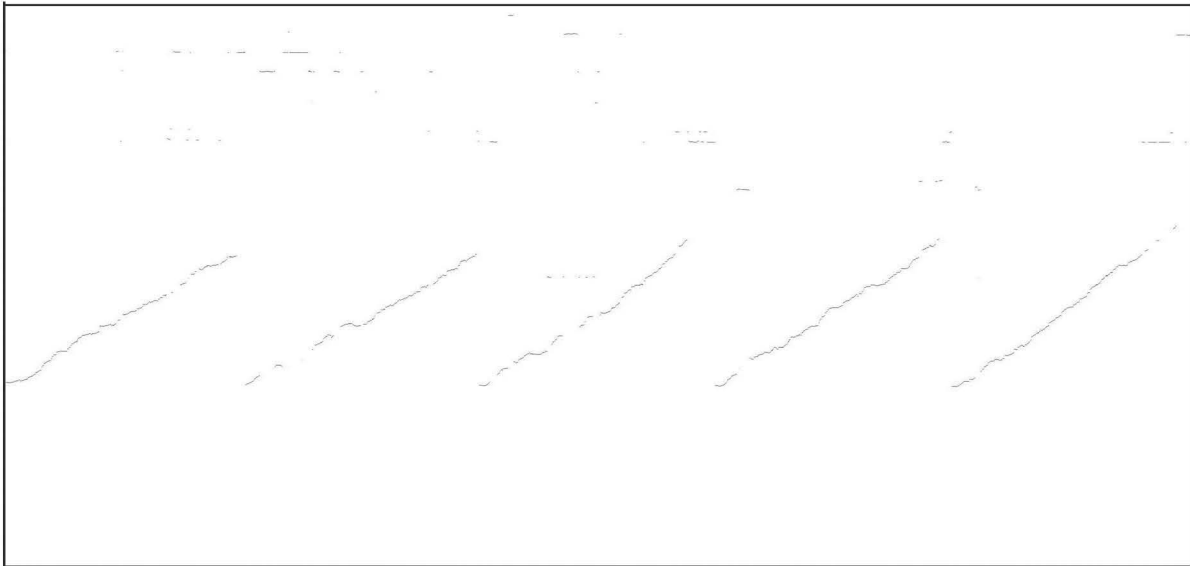


Figure B.6.1.: Scanned and Removed



Figure B.6.2.: Marked

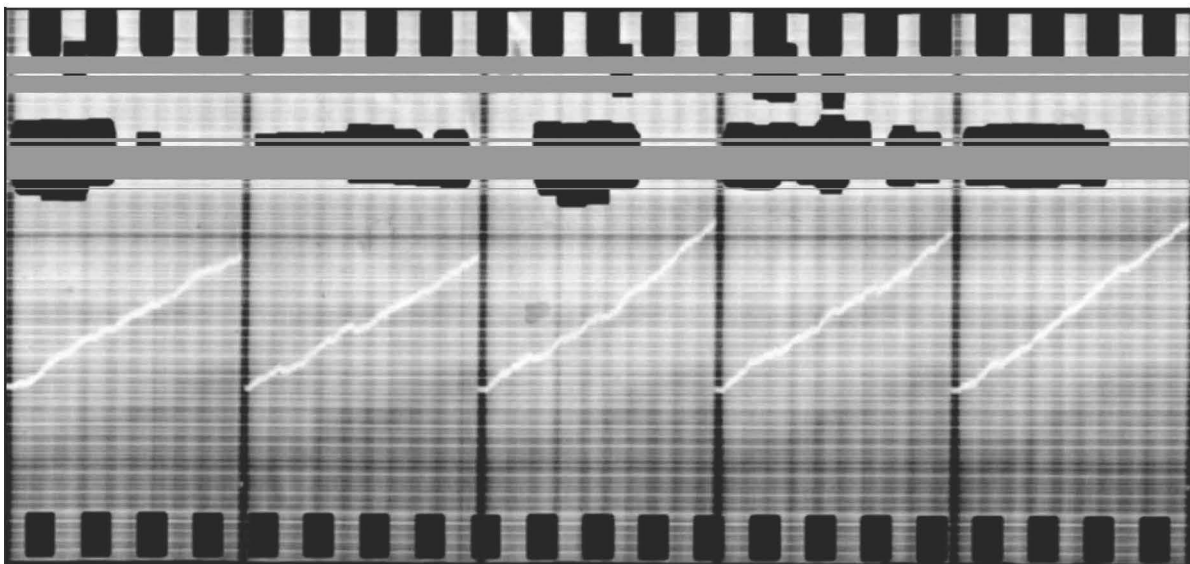


Figure B.6.3.: Erased

B.7. Rough data identification: Iteration 6

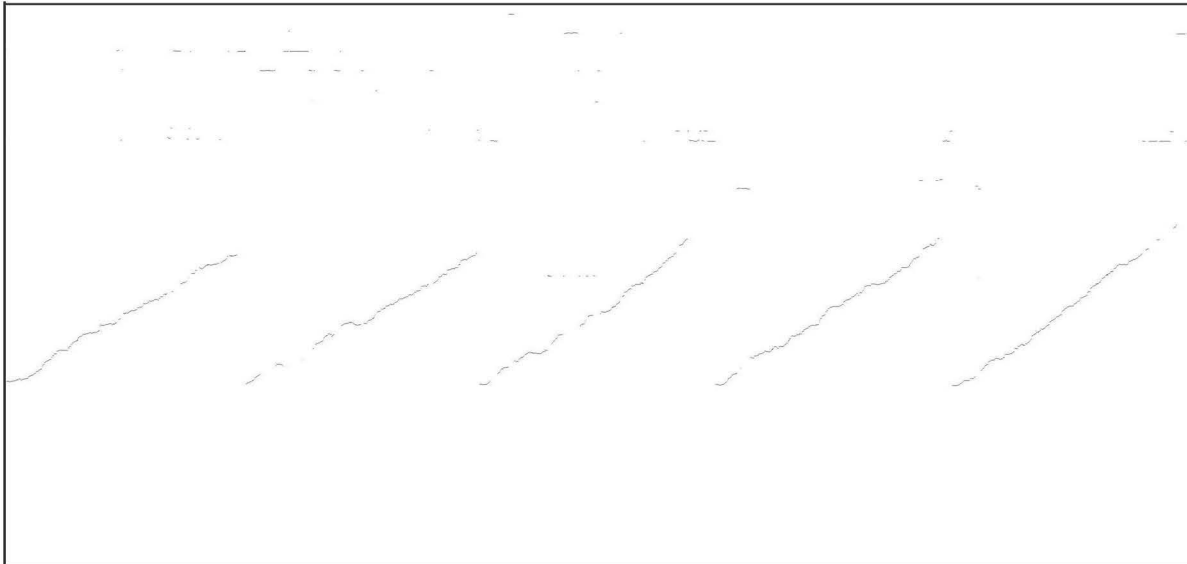


Figure B.7.1.: Scanned and Removed



Figure B.7.2.: Marked

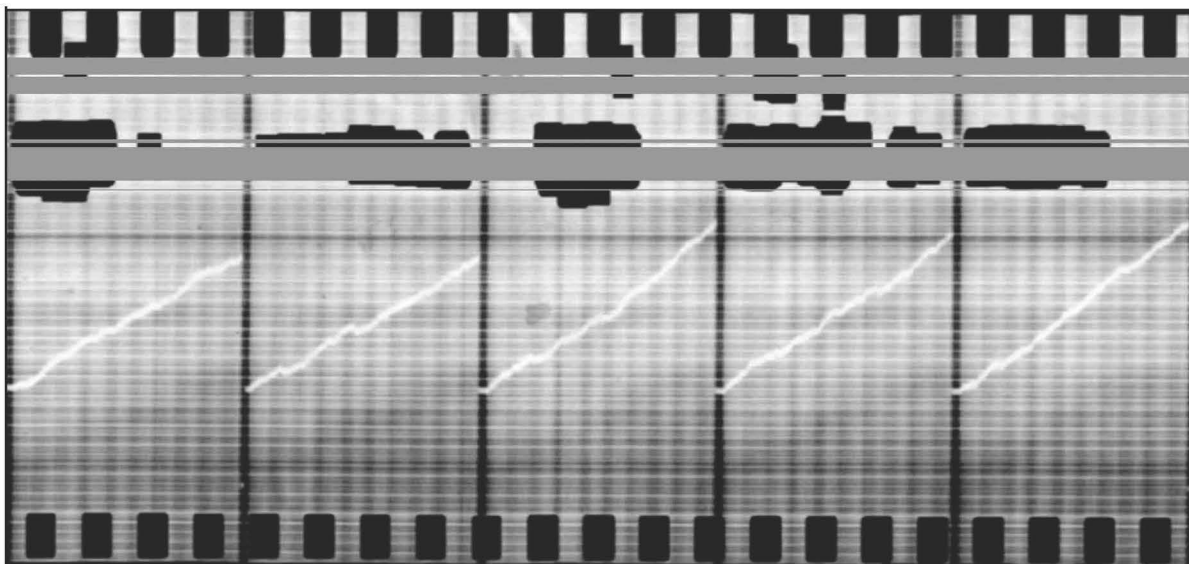


Figure B.7.3.: Erased

B.8. Rough data identification output

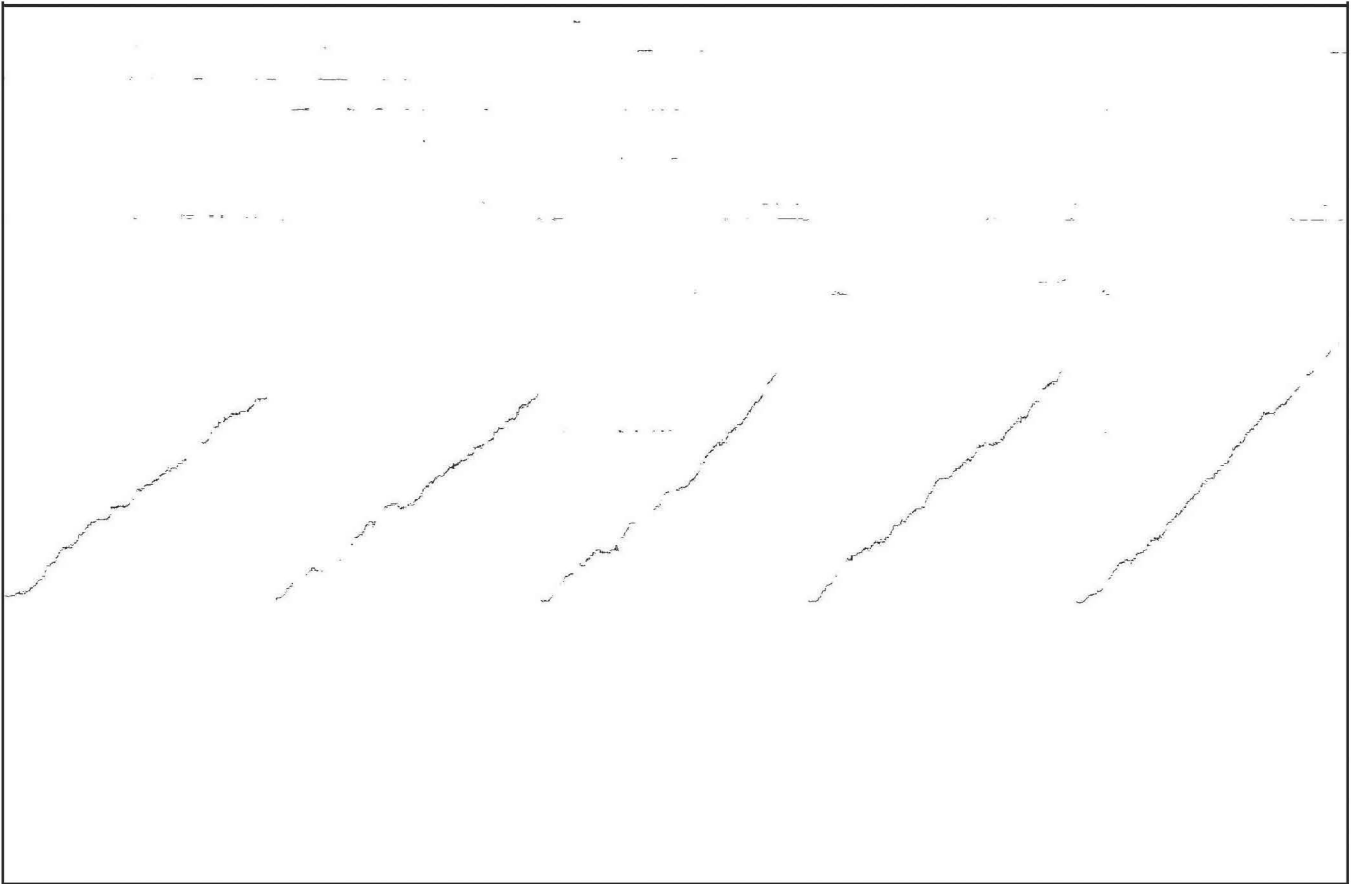


Figure B.8.1.: Scanned and Removed

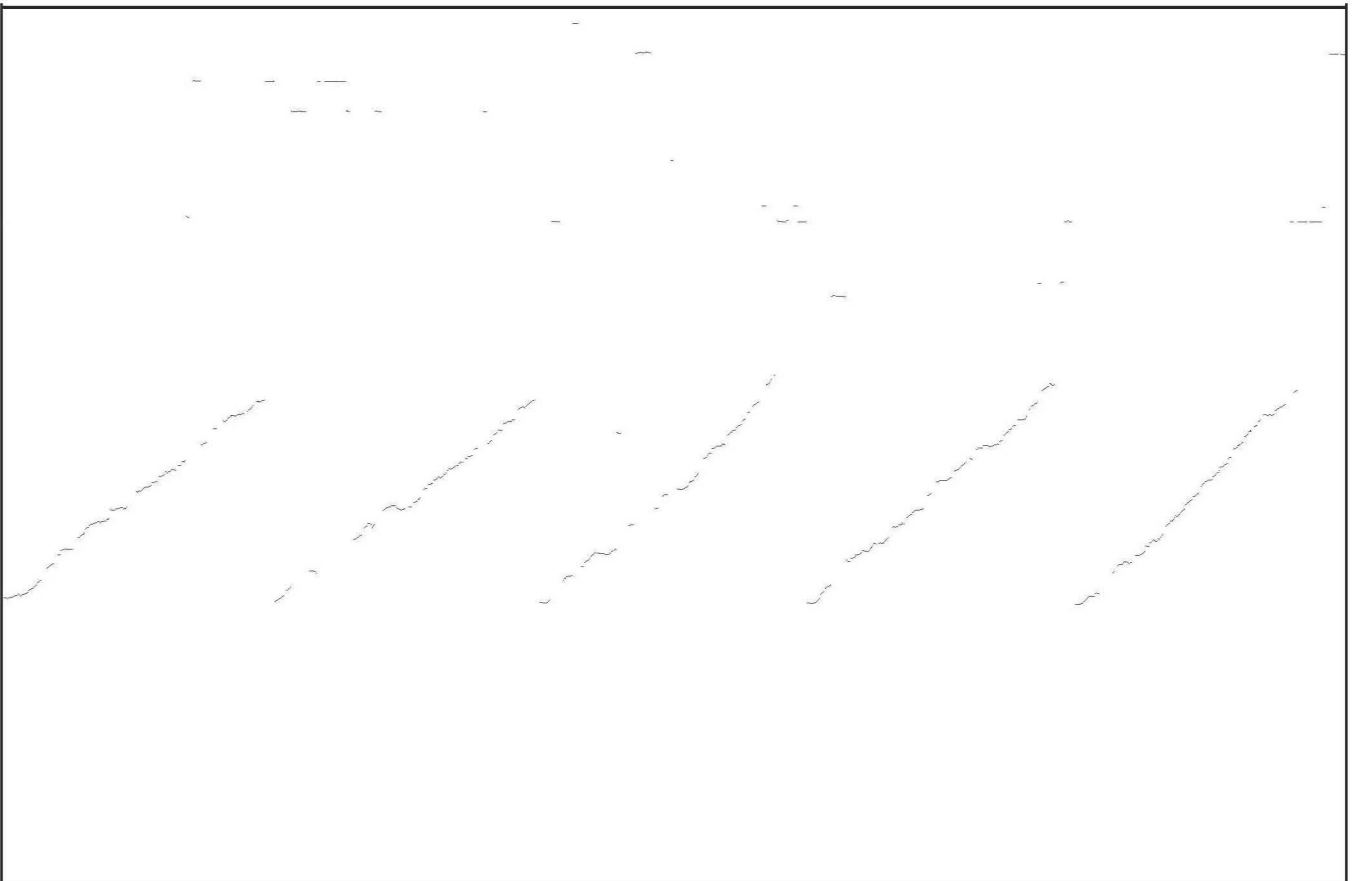


Figure B.8.2.: Cleaned

B.9. Rough data extraction



Figure B.9.1.: Targeted and Connected



Figure B.9.2.: Scrubbed and Identified1

B. Visual summary of the binarization process



Figure B.9.3.: Bound



Figure B.9.4.: Extracted

B.10. Rough data binarization

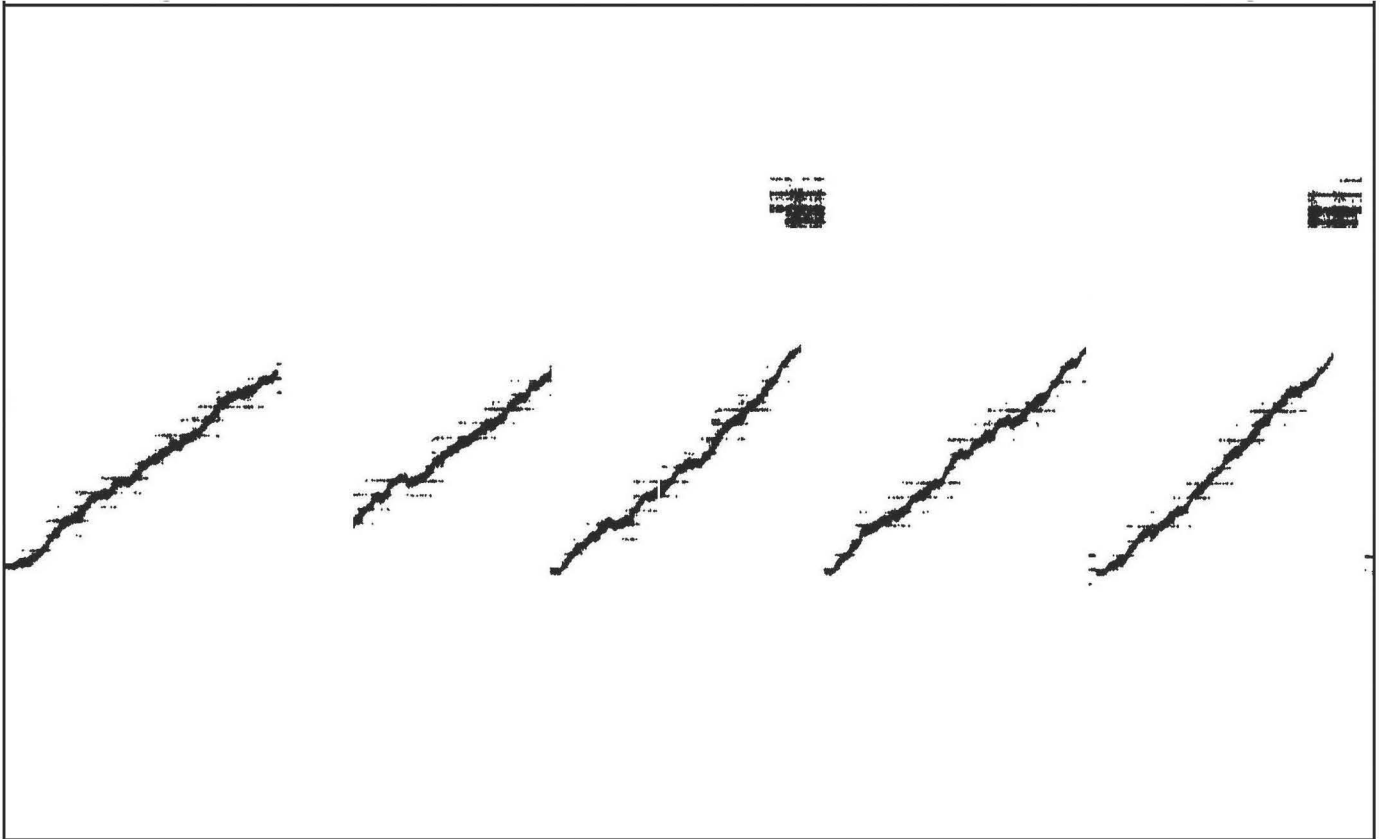


Figure B.10.1.: Rough data binarization: Iteration 1 (Scanned, Removed, Purified, Designated)



Figure B.10.2.: Rough data binarization: Iteration 2 (Scanned, Removed, Purified, Designated)

B. Visual summary of the binarization process

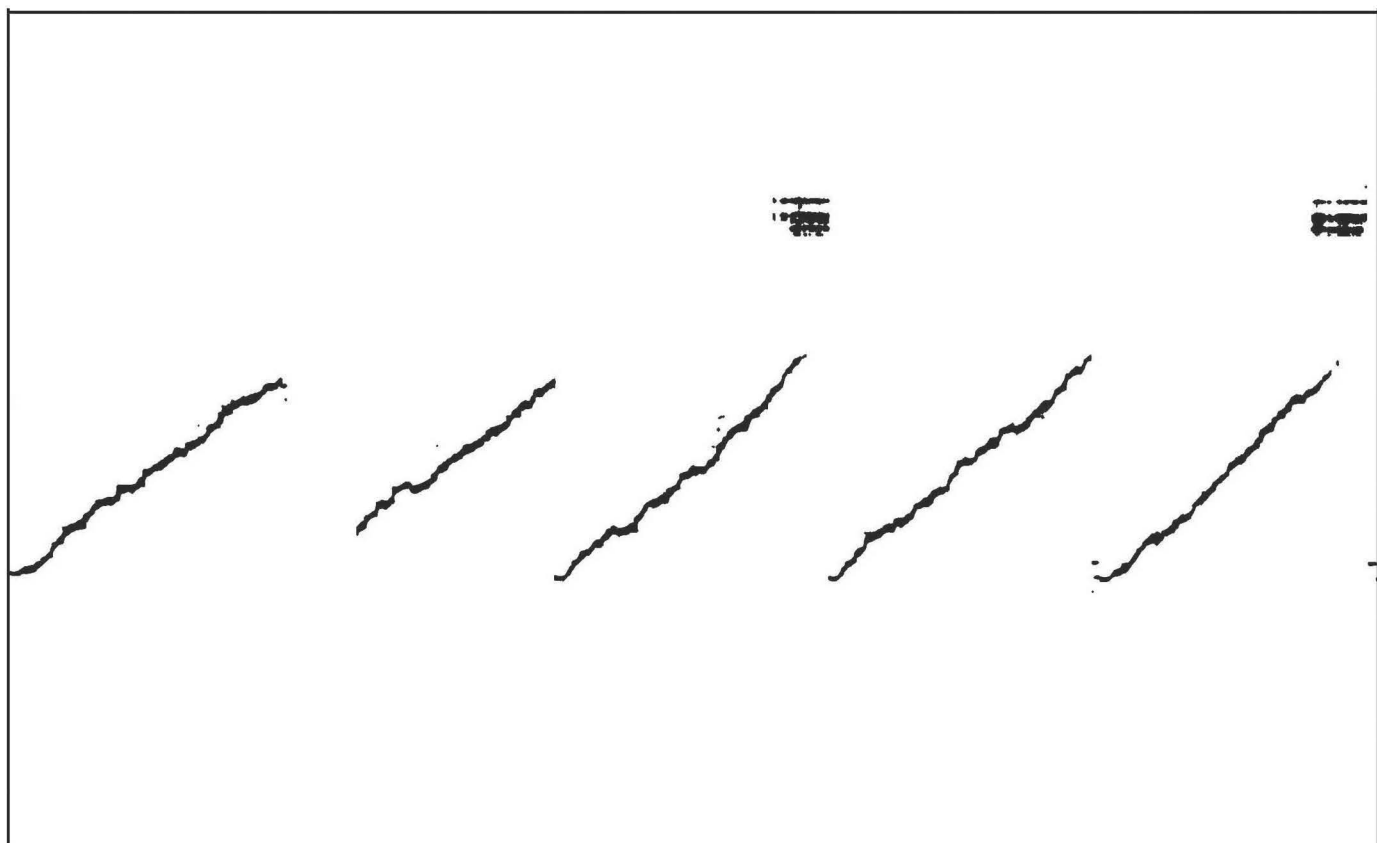


Figure B.10.3.: Rough data binarization: Iteration 3 (Scanned, Removed, Purified, Designated)



Figure B.10.4.: Rough data binarization: Iteration 4 (Scanned, Removed, Purified, Designated)

B. Visual summary of the binarization process

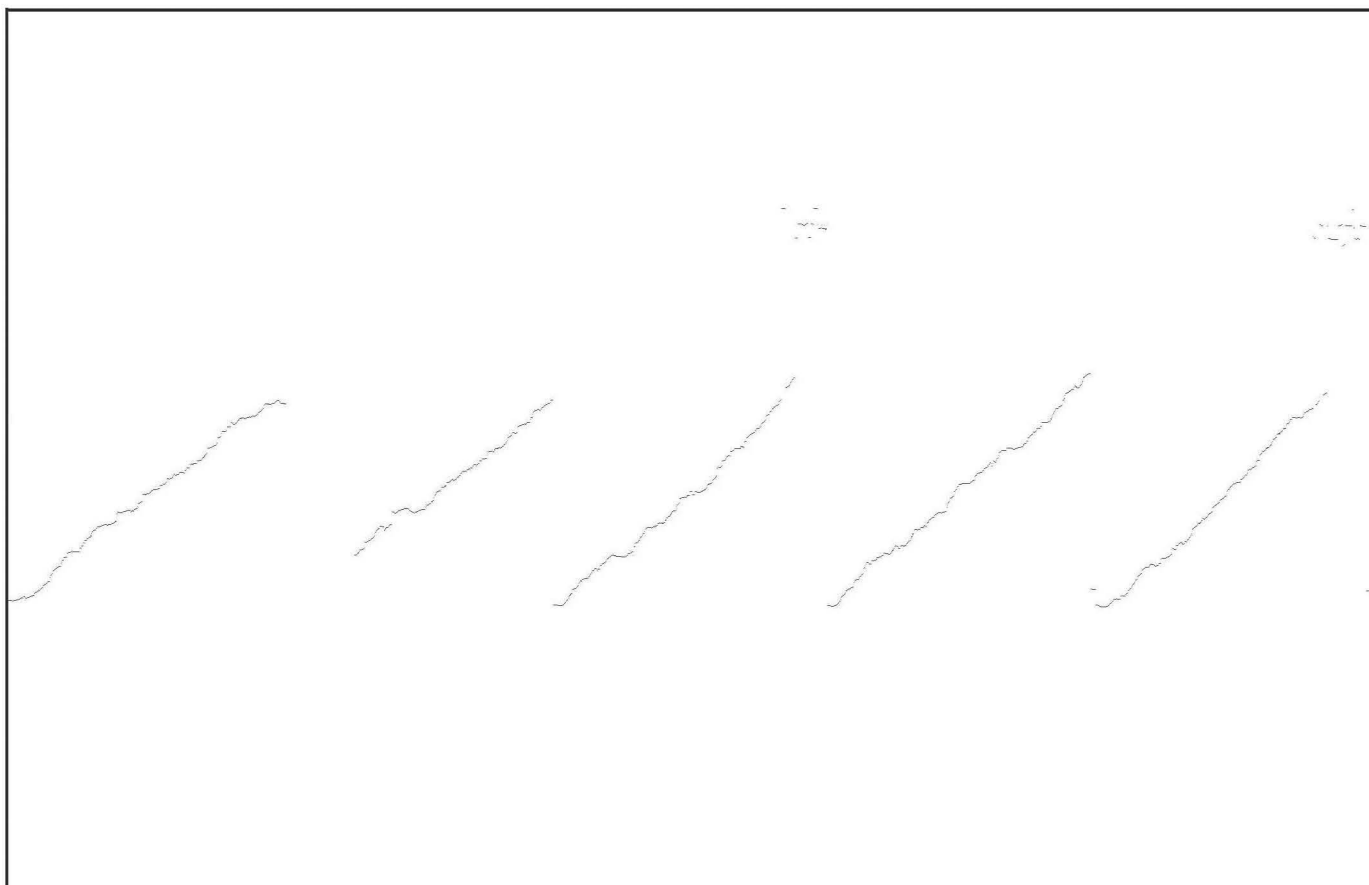


Figure B.10.5.: Scanned

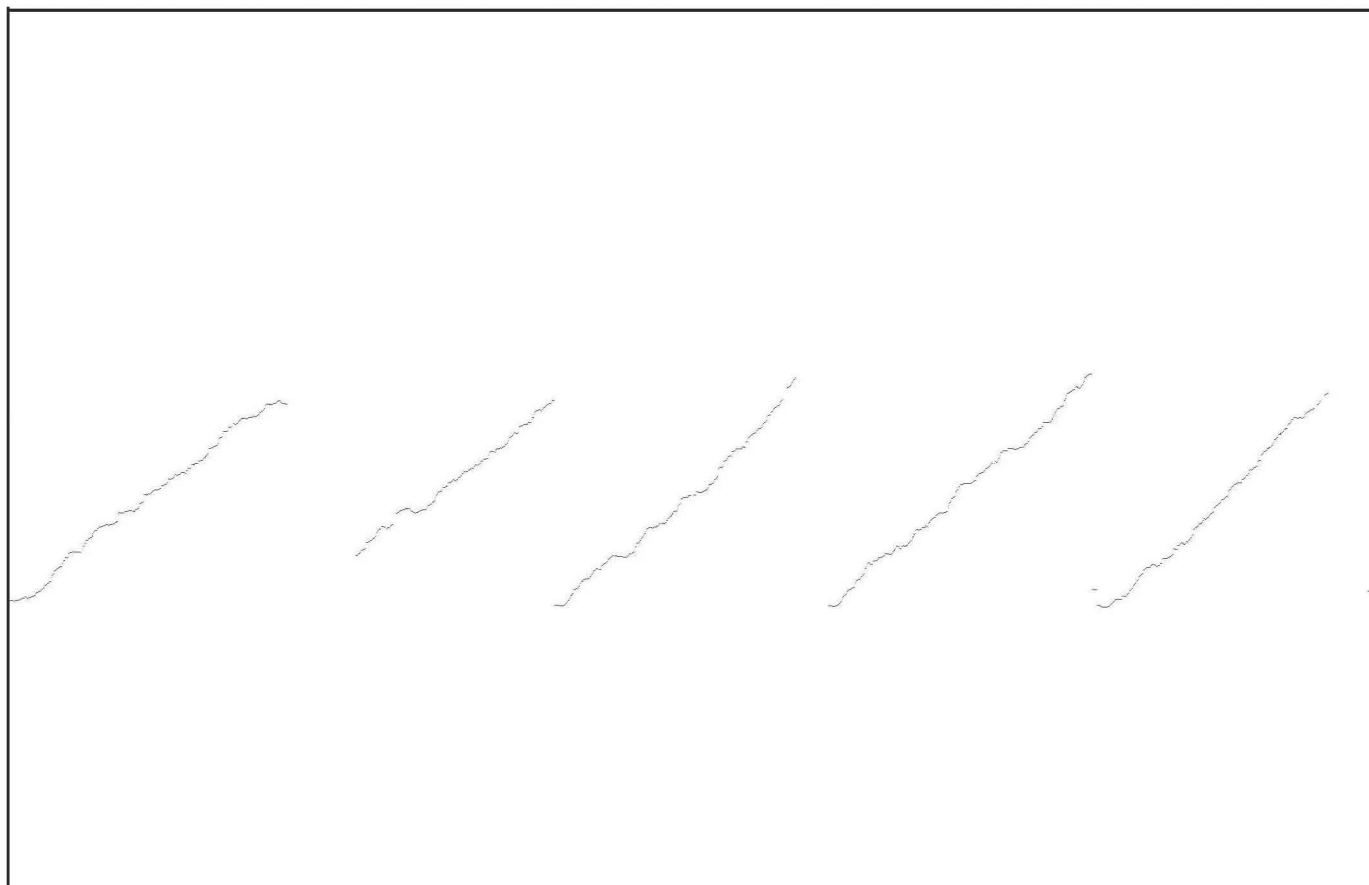


Figure B.10.6.: Identified2 and Defined

B.11. Accurate data identification

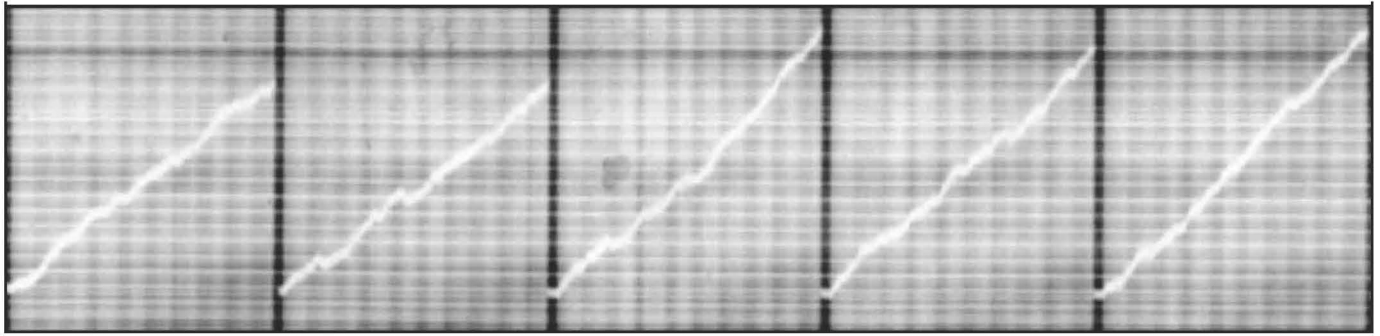


Figure B.11.1.: Identified 3

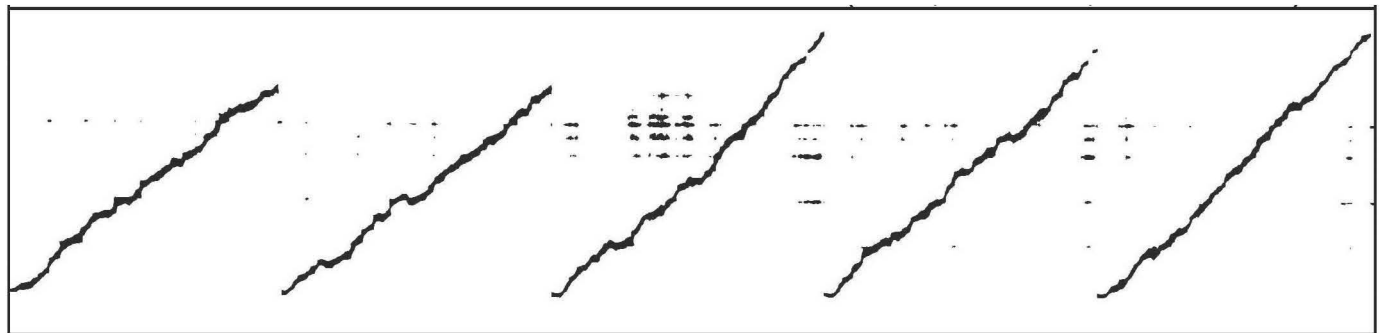


Figure B.11.2.: Accurate data identification: Iteration 1 (Scanned, Removed, Marked, Erased)

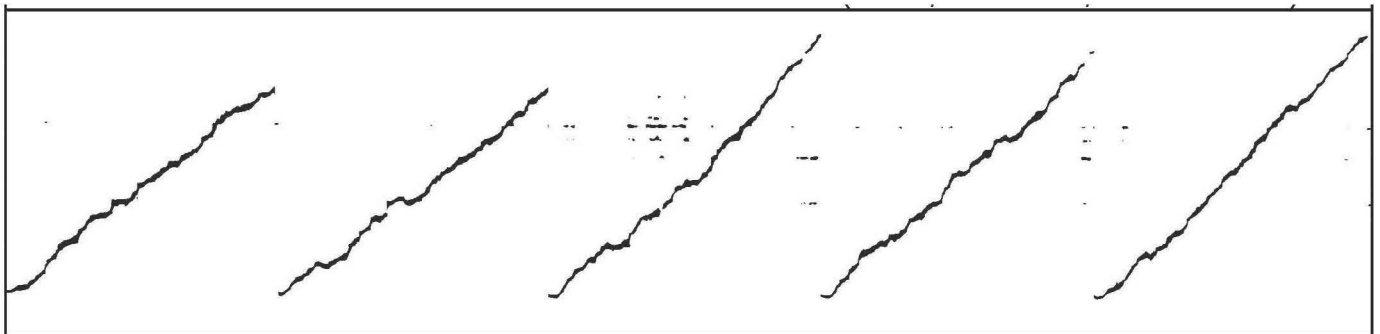


Figure B.11.3.: Accurate data identification: Iteration 2 (Scanned, Removed, Marked, Erased)

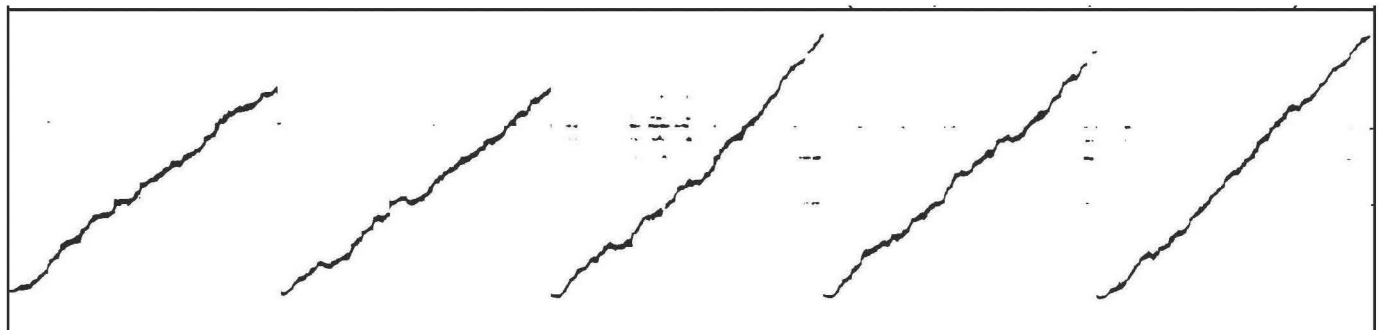


Figure B.11.4.: Accurate data identification: Iteration 3 (Scanned, Removed, Marked, Erased)

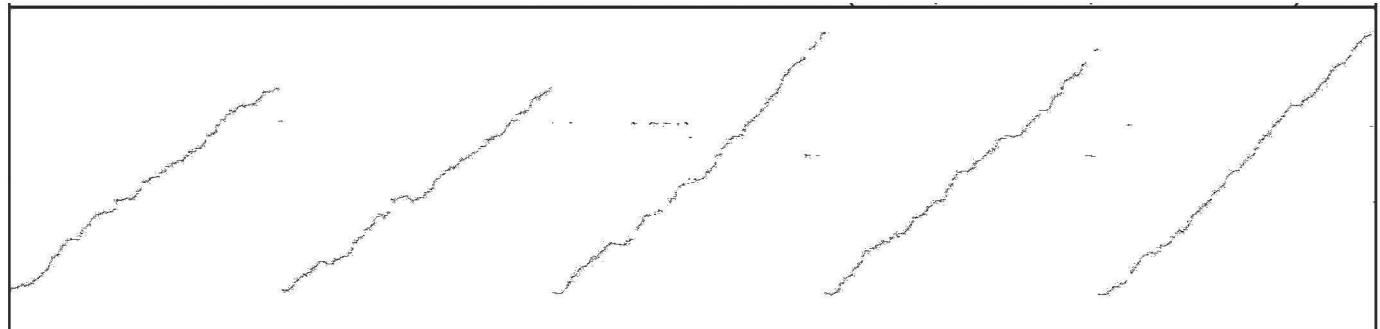


Figure B.11.5.: Accurate data identification: Iteration 4 (Scanned, Removed, Marked, Erased)

B. Visual summary of the binarization process

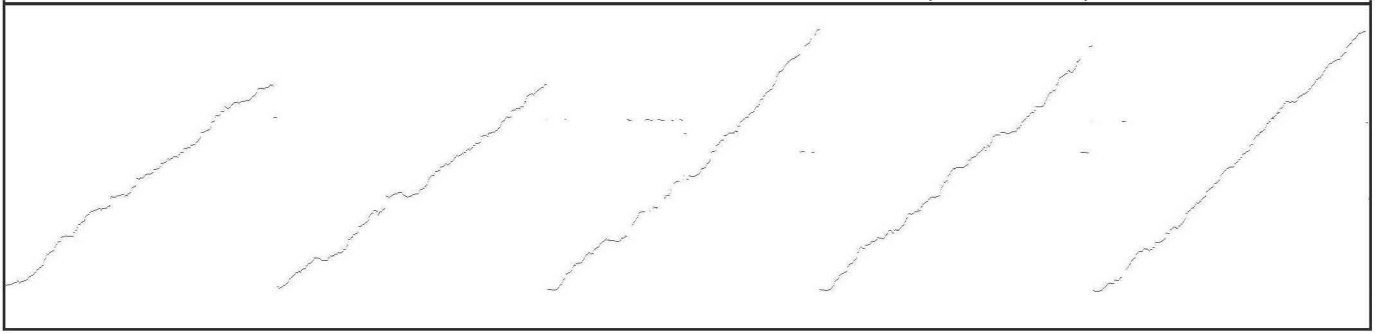


Figure B.11.6.: Accurate data identification: Iteration 5 (Scanned, Removed, Marked, Erased)

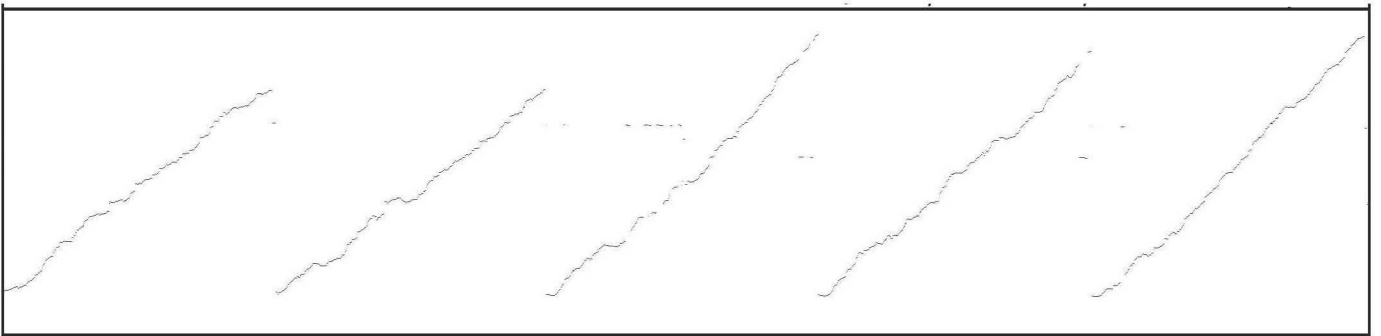


Figure B.11.7.: Accurate data identification: Iteration 6 (Scanned, Removed, Marked, Erased)

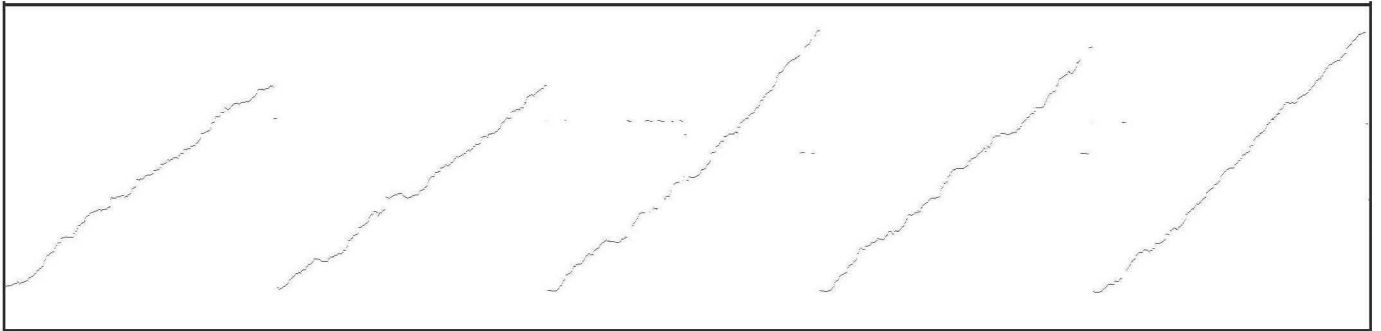


Figure B.11.8.: Scanned

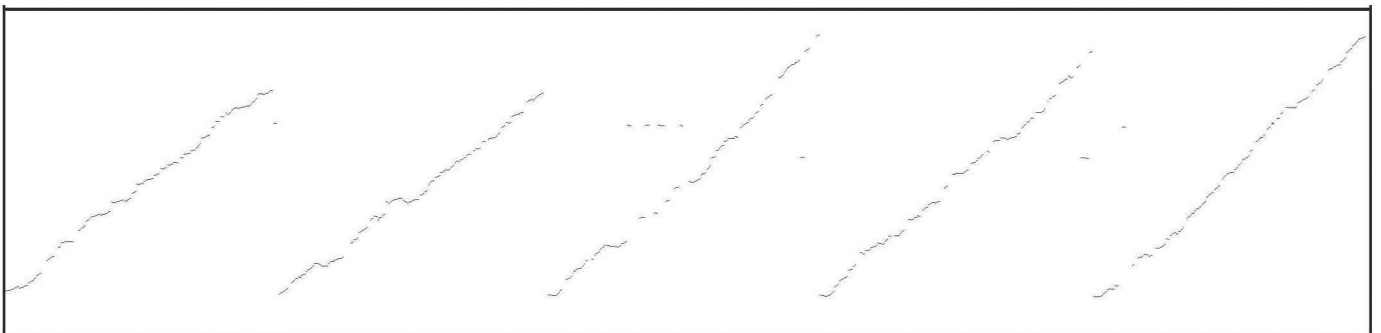


Figure B.11.9.: Cleaned

B.12. Accurate data extraction



Figure B.12.1.: Targeted and Connected

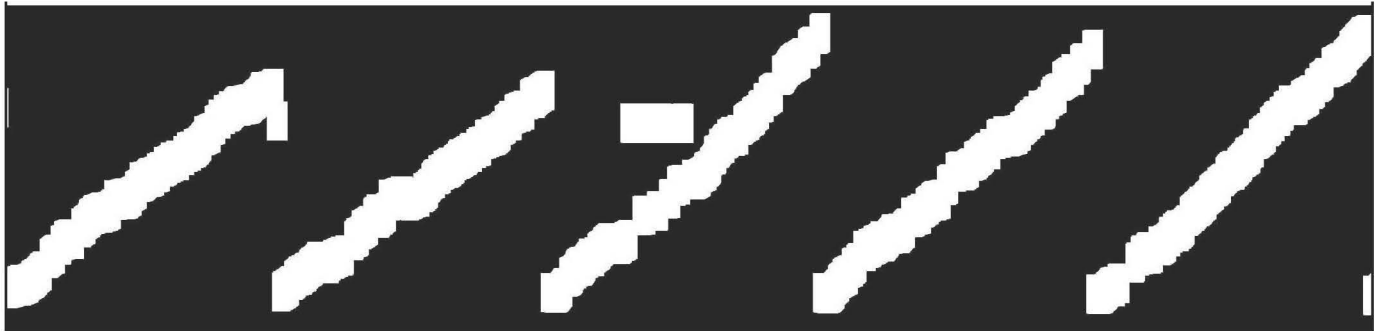


Figure B.12.2.: Scrubbed and Identified



Figure B.12.3.: Bound

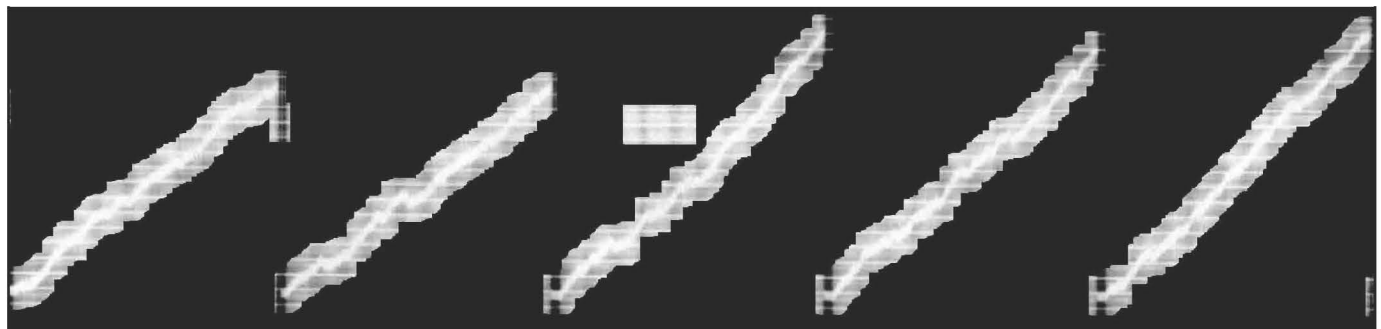


Figure B.12.4.: Extracted

B.13. Accurate data binarization

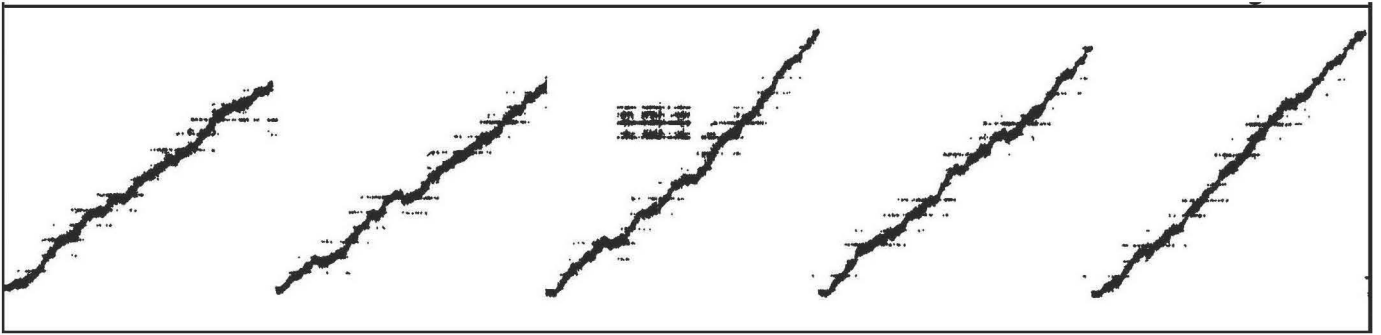


Figure B.13.1.: Accurate data binarization: Iteration 1 (Scanned, Removed, Purified, Designated)

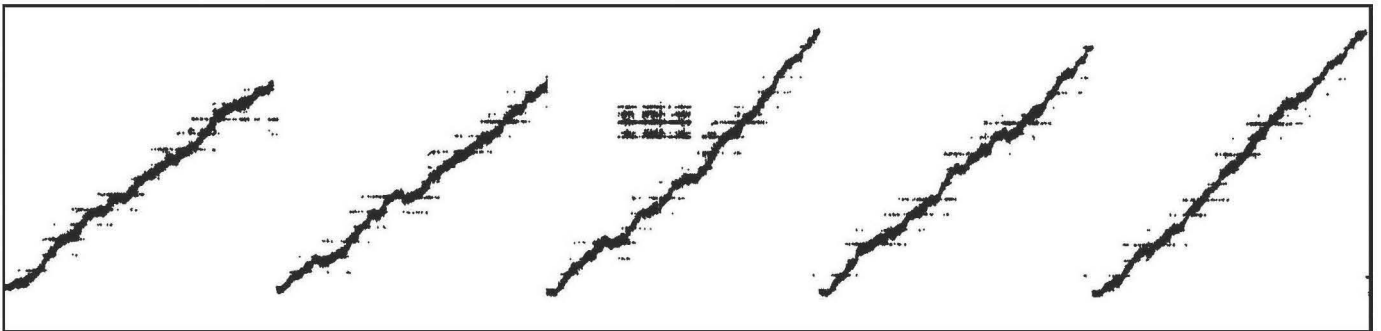


Figure B.13.2.: Accurate data binarization: Iteration 2 (Scanned, Removed, Purified, Designated)

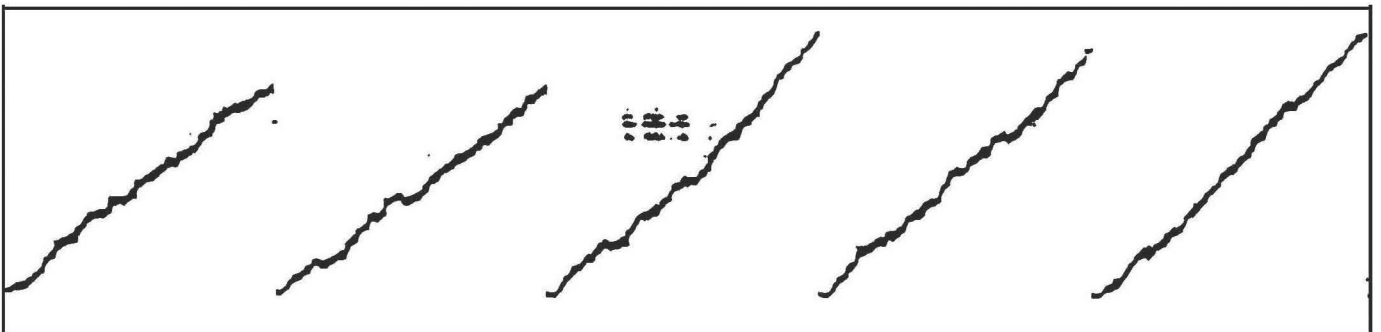


Figure B.13.3.: Accurate data binarization: Iteration 3 (Scanned, Removed, Purified, Designated)

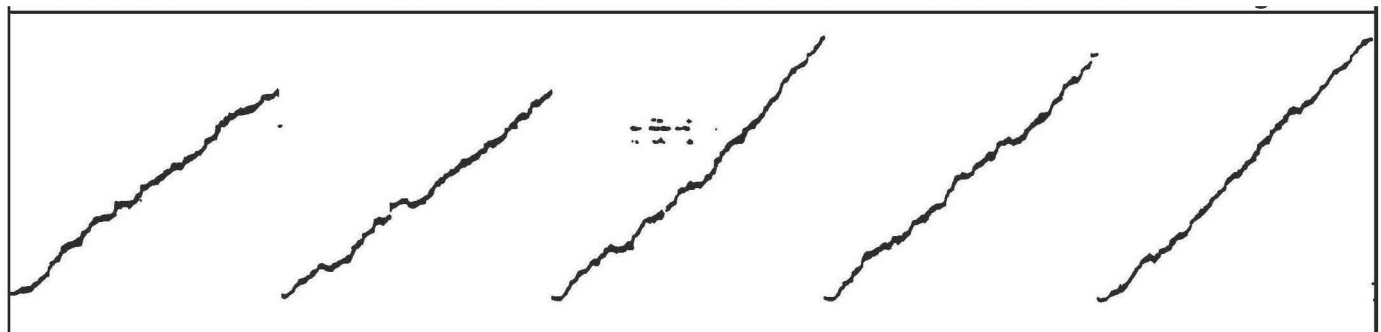


Figure B.13.4.: Accurate data binarization: Iteration 4 (Scanned, Removed, Purified, Designated)

B. Visual summary of the binarization process

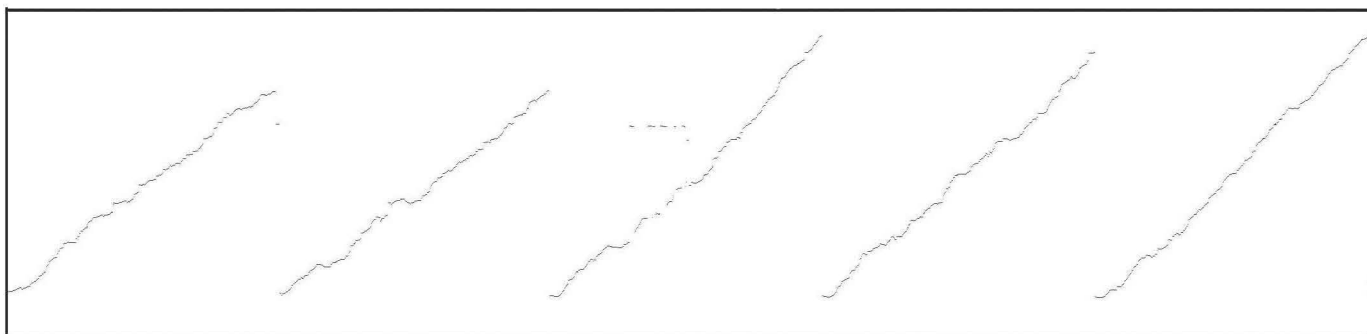


Figure B.13.5.: Scanned, Removed, Marked and Erased

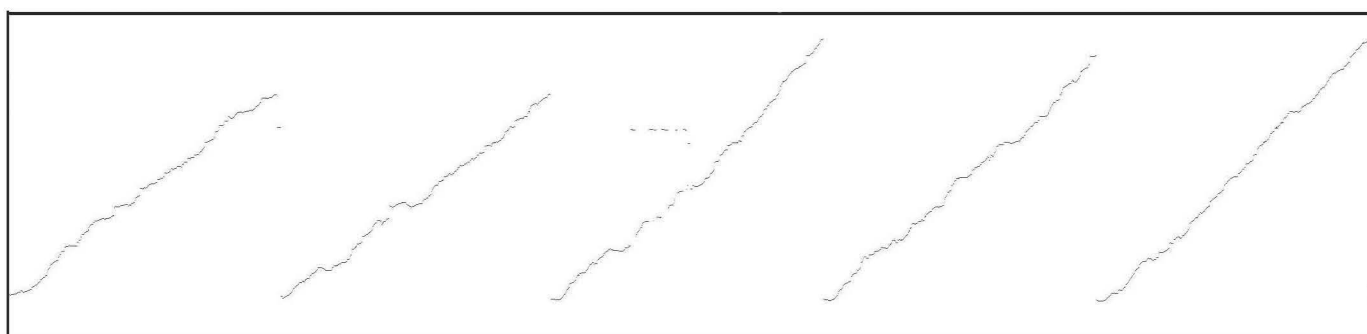


Figure B.13.6.: Identified2

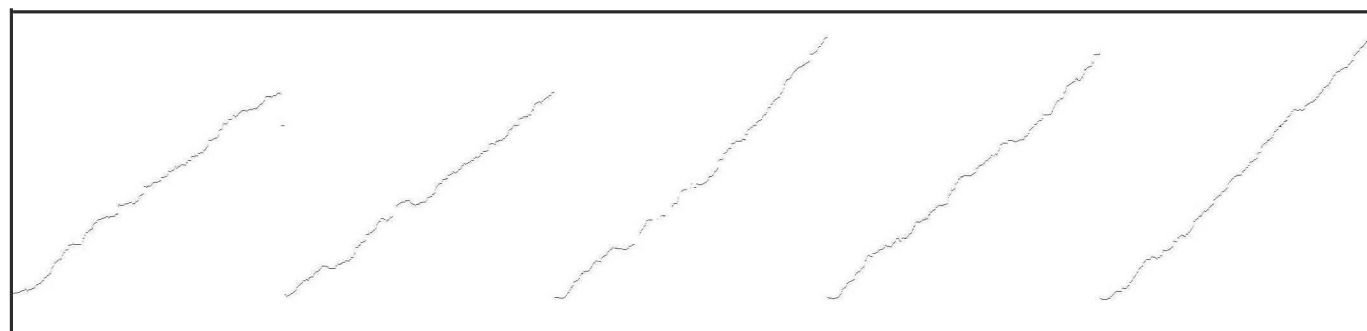


Figure B.13.7.: Defined

B.14. Post-processing

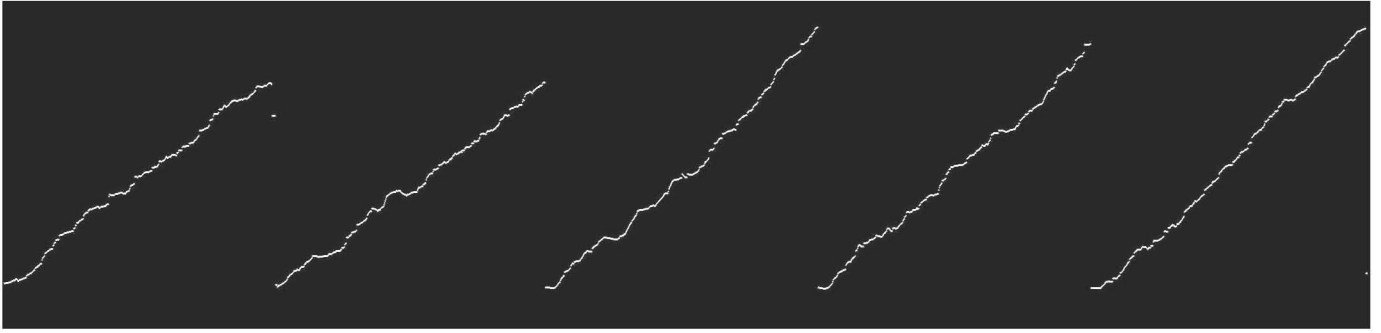


Figure B.14.1.: Plotted

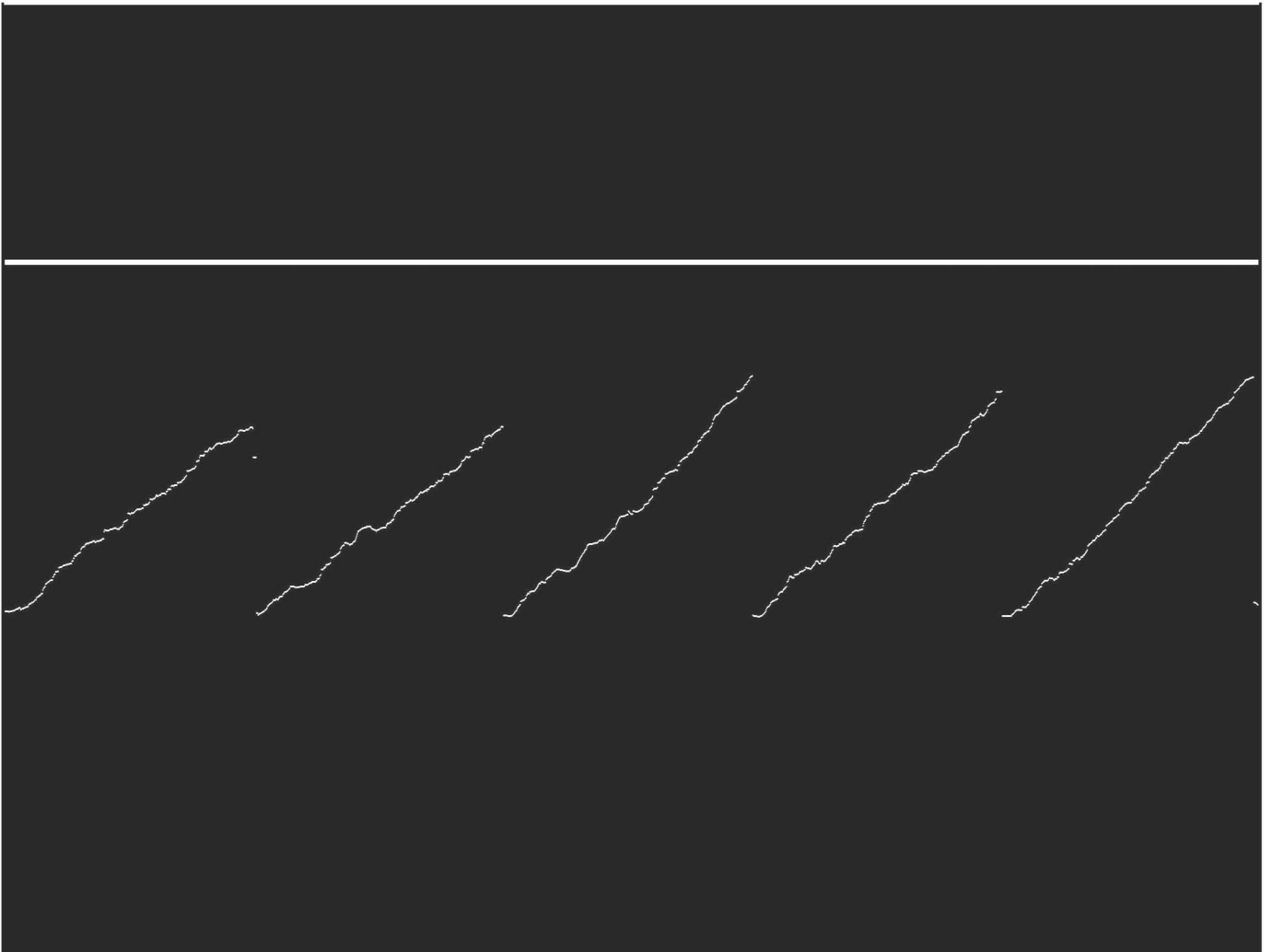


Figure B.14.2.: Pasted and Inserted

C. Letter from the editor

October 12, 2012



To whom it may concern

Re: Letter of confirmation of language editing

The dissertation "*Adaptive binarization of legacy ionization chamber cosmic ray recordings*" by André Steyn (20535341) was language, technically and typographically edited. The sources and referencing technique applied was checked to comply with the APA reference technique. The dissertation is written in English (USA).

Antoinette Bisschoff

Officially approved language editor of the NWU
Member of SA Translators Institute (SATI) Member no. 1001891