METEOROLOGICAL APPLICATIONS
Meteorol. Appl. **23**: 57–64 (2016)
Published online 15 December 2015 in Wiley Online Library
(wileyonlinelibrary.com) DOI: 10.1002/met.1522

RMetS
Royal Meteorological Society

Automatic digitization of pluviograph strip charts

Aleš Jaklič,* Luka Šajn, Gašper Derganc and Peter Peer
Computer Vision Laboratory, Faculty of Computer and Information Science, University of Ljubljana, Slovenia

ABSTRACT: An algorithm for automatic digitization of pluviograph strip charts is presented. The rainfall signal is incrementally extracted from the scanned image of a strip chart by combining the moving average method and the curve edge following method. The mechanical properties of float-based rain gauge were used as constraints in the algorithm design. The algorithm was tested on 58 strip chart images. The comparison between the data derived from the algorithm and the data from the Slovenian Environment Agency shows that the algorithm produces an accurate rainfall time series except for the charts that contain ink smudges. Thus, the algorithm is well suited as a main component of an interactive system that would enable visual inspection of the detected rainfall curve and its possible adjustment.

KEY WORDS pluviograph; rainfall gauge; digitization; computer vision; automatic extraction

Received 5 November 2014; Revised 24 March 2015; Accepted 13 May 2015

1. Introduction

Accurate high-resolution rainfall data are required for many research applications. Besides the studies of precipitation properties (Dirks *et al.*, 1998), rainfall data are essential for the calibration of rainfall-runoff models (Mclntyre and Al-Qurashi, 2008; Bahat *et al.*, 2009). Another important use of these data is in the development of event-based stochastic rainfall generators found in Woolhiser and Osborn (1985), Marien and Vandewiele (1986), Cowpertwait *et al.* (1996) and Koutsoyiannis and Onof (2001). High-resolution precipitation data are also needed for better understanding and statistical analysis of extreme rainfall events (Boni *et al.*, 2006). More on the importance of precipitation data in urban areas can be found in Schilling (1991) and Giesecke and Haberlandt (1998).

Precipitation is measured by either manual (standard rain gauge) or automatic (pluviograph) instruments. While the pluviographs record changes in precipitation intensity with time, standard rain gauges are usually used to measure the amount of precipitation over longer time periods (typically 24 h). The result of pluviograph measurements is a function of cumulative amount of precipitation over time.

The precipitation intensity is measured as an amount of precipitation P over a unit of time t (Bedient *et al.*, 2007):

$$\text{Intensity} = \frac{\mathrm{d}P}{\mathrm{d}t} \approx \frac{\Delta P}{\Delta t} \tag{1}$$

Changes of precipitation intensity with time are usually graphically represented with a hyetograph.

Nowadays, these data are usually acquired by automatic digital pluviographs that make the data immediately available for further analysis. Modern techniques of rainfall estimation are frequently based on usage of electronic scales (Horiuchi *et al.*, 2011).

Before the digital pluviographs came into use, pluviographic data were acquired from paper strip charts (Figure 2) recorded by non-digital automatic pluviograph gauges. There are 38 such float-based gauges (Figure 1) currently operating in Slovenia. The paper strip charts contain a scale depicting time (24 h, from 0700 to 0700) on the x axis and the amount of accumulated water in the accumulator bucket (from 0 to 10 mm) on the y axis. The size of the strip charts is $\sim 422 \times 113$ mm. When the bucket is completely filled, it is automatically emptied, resulting in a sudden drop of the rainfall signal curve (see Figure 2).

The data from the strip charts are converted into digital form by using a digitizer tablet. This process requires a lot of concentration and precision from a human operator. The process is also time-consuming and can be affected by operator errors. After the chart is laid onto the tablet and the borders of the area of interest are marked, the rainfall curve plotline is determined by marking points on the plotline. Rainfall intensities are calculated by a computer program from the resulting point sequences. To process monthly data from a single measuring station, an operator requires from 10 min to an hour, depending on the amount of precipitation recorded on the charts.

In spite of all the obvious benefits that digital pluviographic gauges provide, they are not as robust as their predecessors, which can accurately record data even in events of extreme weather conditions. There are also a lot of pluviographic data recorded on paper strip charts from the times when digital pluviographs were not yet available. By obtaining high-resolution pluviographic data from these charts, many research efforts could benefit by being able to analyse precipitation data in greater detail.

The process of digitization could be greatly improved with the help of a computer program that could automatically acquire these data from the strip charts. This study presents an algorithm that could accomplish this acquisition from scanned images of pluviographic strip charts. The digitization process could become faster and less error prone, and perhaps even more accurate results would be obtained. In addition to the automatic image-processing algorithm described in this study, scanning of the strip charts and simple correction of possible errors are only required.

* Correspondence: A. Jaklič, Computer Vision Laboratory, Faculty of Computer and Information Science, University of Ljubljana, Večna pot 113, 1000 Ljubljana, Slovenia. E-mail: ales.jaklic@fri.uni-lj.si

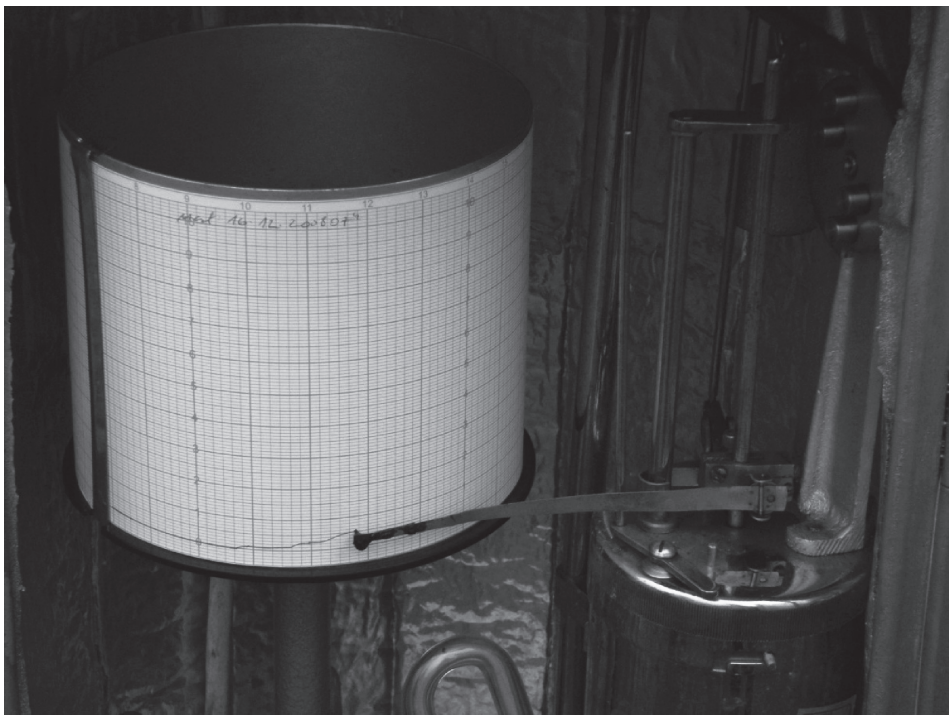

Figure 1. Pluviograph gauge.

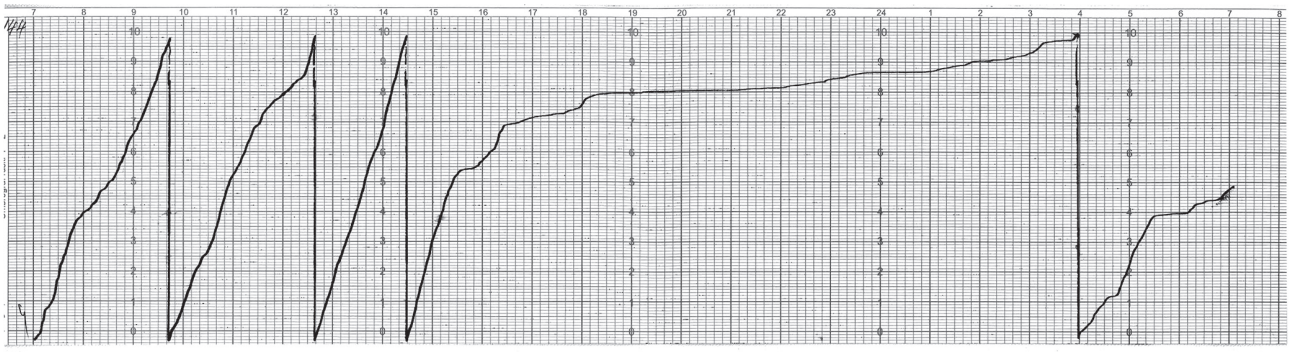

Figure 2. Pluviograph strip chart.

2. Related research

Similar research has been performed at the University of Cagliary in Italy (Deidda *et al.*, 2007). The goal of their research was digitization of the rainfall data from tipping bucket gauge strip charts.

The core steps of their algorithm are: preprocessing, segmentation, automatic signal detection and interactive post-processing.

A 300 DPI resolution image of a pluviograph strip chart serves as input to the algorithm. In the preprocessing step, the image is rotated and transformed (warped) to make the image pixels of the image pixel columns correspond to the same time interval.

The segmentation step that determines which pixels of the image belong to the rainfall plotline consists of thresholding of the R component of RGB colour space step followed by non-hierarchical clustering in the hue-saturation-value (HSV) colour space. The automatic signal recognition step determines the curve plotline.

The similarity of the problems is reflected in the similarity of the basic algorithm structure. The differences between the approach by Deidda *et al.* (2007) and the present approach, which stem from the differences in the input strip charts and the pluviograph pen movement, are as follows:

Strip chart/gauge-based differences in Deidda *et al.* (2007):

- The discontinuous rainfall signal curve. The gauge pen moves vertically in discrete intervals determined by the capacity of a single collector bucket resulting in gaps in the signal curve plotline.
- Warped strip chart scale and rainfall signal curve (circumferential pen movement).
- The *y* co-ordinate axis of the rainfall signal curve is not always proportional to the amount of precipitation collected in the bucket. When the pen reaches the edge of the strip chart, the movement of the pen is inverted.

Processing-based differences:

- In the present study, the presented algorithm uses the Commission internationale de l'éclairage L-a-b (CIELAB) colour space components to separate the rainfall signal curve from the background with an improved version of

thresholding. The algorithm by Deidda *et al.* (2007) performs thresholding in the RGB colour space followed by non-hierarchical clustering in the HSV colour space.
- The curve edge following method that we use in the curve detection step is not appropriate for use with strip charts processed by Deidda *et al.* (2007) due to discontinuous rainfall signal curve.

3. Automatic pluviograph strip chart reading algorithm

The basic steps of the automatic pluviograph strip chart reading algorithm are as follows:

1. separate the rainfall curve from the background, determine which image pixels belong to the curve (segmentation);
2. accurately determine the curve plotline, find a sequence of pixels covering the whole strip chart length with just one pixel *per* image column (curve detection), and
3. calculate the fullness of the collector bucket from pixel co-ordinates and from the differences of these quantities at selected time intervals calculate the corresponding precipitation intensities.

The algorithm takes as input an image in either JPG, PNG or BMP format with a 24 bit colour depth. The resolution of the training and testing set images is 300 DPI, which is a reasonable compromise between accuracy and algorithm execution time. The image resolution is not fixed, but it should not be too low, since curve precision may be lost by lowering the resolution. The input images must satisfy the following constraints.

- All pixels in the *i*th image column correspond to the same time interval. This condition is not met in case of a rotated image or when the pluviograph paper strip chart was not accurately inserted.
- The pluviograph strip chart contains only one rainfall curve. In case of longer periods with no rain, the same strip chart can be used over several days resulting in multiple rainfall curves.
- The colour of the pluviograph pen ink should be perceptually different from the strip chart background.

3.1. Segmentation

In the segmentation step, the rainfall curve is separated from the background. The result of the segmentation is a binary image where the pixels recognized as a part of the curve have a value of 1, while the rest of the image pixels are assigned a value of 0. Because of specific colours of the background and the ink, the CIELAB colour space was used (Forsyth and Ponce, 2002) for presentation of the input images. The colours of the background and the ink are further apart in this colour space than in the RGB colour space and are thus easier to separate.

The pixels belonging to the rainfall curve are selected based on thresholding (Nixon and Aguado, 2002). Thresholding is implemented in a region-growing fashion (Mat-Isa *et al.*, 2005). Seed pixels are those image pixels whose values exceed a threshold that only curve pixels could. These pixels are used as the starting point of a new region. Growing of the region is then based on the lower threshold and the standard deviation of the currently observed pixel neighbourhood. This method improves simple thresholding (that chooses all pixels between two thresholds) by considering the neighbourhood characteristics beside the pixel value. With the use of this method, the amount of falsely detected pixels is reduced and because of the use of standard deviation, local features of the curve are considered whose intensity can vary. Consequently, the accuracy of the segmentation will improve. Figure 3 demonstrates how thresholding with region growing preserves finer details of the image structure.

This method can also be used with swapped threshold roles, where the region begins at a pixel with a value lower than the upper threshold and continues its growth to pixels with a value higher than the lower threshold.

The average of the region values used for segmentation is defined as:

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{2}$$

where x_i is a grey scale value of the ith pixel of a region containing n pixels. The corresponding standard deviation is defined as:

$$\sigma = \sqrt{\frac{\sum_{j=1}^{n}\left(x_j - \bar{x}\right)^2}{n-1}} \tag{3}$$

where x_j is the grey scale value of the jth pixel and n represents the number of all pixels in the jth pixel square-shaped neighbourhood of size 3×3.

This procedure is applied to grey scale a^* and b^* CIELAB components that are smoothed by the use of a mediana filter of size 2×2 (Jähne, 2002). Greyscale images in this algorithm are images with pixel values between 0 and 255. With the a^* greyscale component, region growing begins at pixel values higher than 160. The lower threshold is set to 150. With the b^* greyscale component, region growing begins at pixel values lower than 110. The second threshold is set to 126. The size of the neighbourhood is set to 3×3. These values were obtained empirically from the learning image set. The segmentation process is depicted in Figure 4.

3.2. Curve detection

The result of the segmentation step is a binary image **A** of size $M \times N$. Image pixel a_{ij} ($i \in [1, M]$ and $j \in [1, N]$) contains value 1 in case it was recognized as a part of the curve and value 0 in case it is a background pixel. Due to rainfall curve thickness and possible ink smudges, a single image column in **A** usually contains more than one detected pixel (with value 1). To calculate the rainfall intensity accurately, one must accurately and unambiguously determine the curve plotline, a single pixel *per* image column. This is achieved with a combination of two interleaved methods.

A local curve edge following method quickly and accurately determines the curve plotline in regions, where the curve is connected. There are possible errors in case of a smeared rainfall curve.

Pixels are chosen in sequential fashion, where the next pixel position is considered in the order depicted in Figure 5. This order assures that the resulting pixels always lay at the bottom of the curve. In cases where this algorithm reaches a dead end, it is able to backtrack for a maximum of 10 previously chosen pixels.

When even backtracking cannot help, the procedure temporarily terminates and leaves the processing to the averaging step described below.

A global two-phase averaging calculates the average ordinate value of detected pixels over multiple neighbouring columns (Figure 6). This method is less susceptible to local image irregularities, but does not determine the curve as accurately as the edge

Figure 3. (a) original image, (b) simple thresholding, (c) thresholding with region growing.

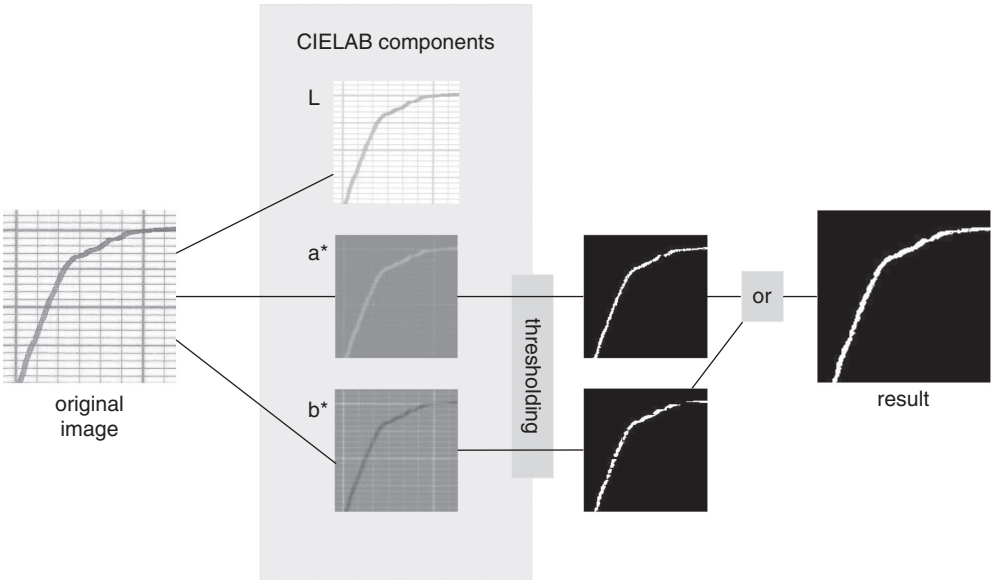

Figure 4. Segmentation steps.

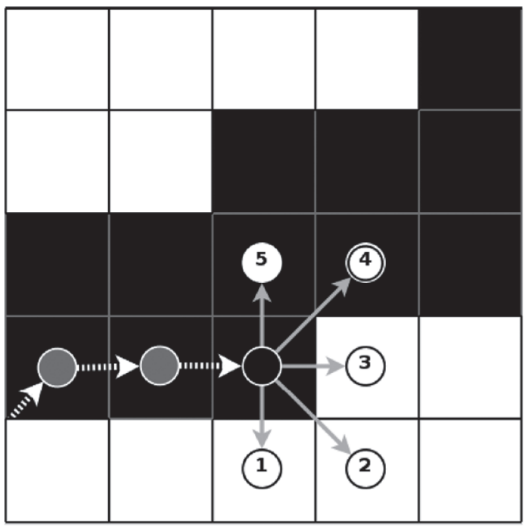

Figure 5. Edge following.

following procedure. For each image column i, the initial estimation $\overline{y}_i$ for that column is first calculated. Its calculation considers multiple image columns using a moving average algorithm:

$$\overline{y}_i = \frac{1}{\overline{N}_i} \sum_{j=1}^{M} \sum_{k=i-\lfloor W/2 \rfloor}^{i+\lfloor W/2 \rfloor} a_{jk}\, j \tag{4}$$

where,

$$\overline{N}_i = \sum_{j=1}^{M} \sum_{k=i-\lfloor W/2 \rfloor}^{i+\lfloor W/2 \rfloor} a_{jk} \tag{5}$$

$\overline{N}_i$ denotes the number of detected pixels in a window of width W and height M with the centre in column i. In our program, the value $W = 7$ was chosen, which equals approximately the curve thickness. In case $\overline{N}_i = 0$, the value of $\overline{y}_i$ is marked as unknown. In spite of all these precautions, ink smudges can lead the average value far away from the curve plotline. That is why $\overline{y}_i$ is an estimation of the plotline position used in the second phase as the centre of a new averaging window with smaller height H_i. The final plotline estimate y_i is then calculated as:

$$y_i = \frac{1}{N_i} \sum_{j=\overline{y}_i-\lfloor H_i/2 \rfloor}^{\overline{y}_i+\lfloor H_i/2 \rfloor} \sum_{k=i-\lfloor W/2 \rfloor}^{i+\lfloor W/2 \rfloor} a_{jk}\, j \tag{6}$$

where N_i denotes the number of detected pixels in a window centred at the pixel $(i, \overline{y}_i)$ and of height H_i. The width W of the window remains the same as in the first phase. The height H_i

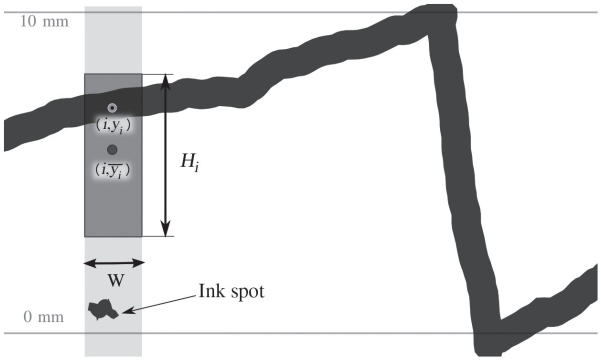

Figure 6. Two-phase averaging.

depends on the value $\overline{N}_i$ from the first window and is calculated using Equation (7), which tries to retain as many pixels from the first window in the new window, while keeping the height small:

$$H_i = \max\left(2W, \frac{\overline{N}_i}{W}\right) \tag{7}$$

In spite of this two-phase calculation, the value y_i can still be inaccurate due to excess ink smudges such as those presented in examples in Figure 7.

From the evidence factor $e_i = N_i / \overline{N}_i$, we can estimate how accurate y_i actually is. Figure 8 depicts the influence of the evidence factor for column i on the continued behaviour of the algorithm and the switching between the edge following and the averaging method.

If the evidence factor e_i is too low (<0.6), the curve plotline position of the corresponding column is marked as unknown. Due to the use of two phases and the evidence factor, the curve plotline can be accurately determined even if the image contains moderate amount of noise-ink smudges.

The combination of the above-mentioned methods does not necessarily produce a sequence of one pixel *per* image column. The edge following method can produce several pixels *per* column, while the averaging can leave empty columns (unknown values). By observing the graphs, one can also deduce that these sequences should obey certain constraints. Since the precipitation is always positive, the resulting pluviograph curve should be monotonically increasing. To be precise, it should be composed of monotonically increasing segments, where each segment spans the time between two consequent bucket-emptying events, marked by sharp vertical pen movement from the top to the bottom of the strip chart (see Figure 2). To produce one pixel *per* column sequence that obeys this constraint, three more steps are required that process only the sequence of elements and need no longer any image information. These steps are noted below.

- Ink spot rejection: spots are identified as large jumps in the sequence in short periods of time.
- Enforcing the constraint that the sequence is composed only of increasing segments: the segments in the sequence should be increasing, except at the points of inversion (when the collector bucket is emptied). For each segment, the optimal curve plotline is determined by the use of the longest increasing subsequence algorithm (Gusfield, 1997). The algorithm is implemented with a dynamic programming method (Cormen *et al.*, 2001). Inversion points are detected as steep falls of the curve from the top to the bottom of the strip chart image.
- Exactly one point *per* image column: where a column contains multiple pixels, these pixels are replaced by a single one with the average ordinate value of these pixels. Empty columns (unknown ordinate values) are filled using linear interpolation over neighbouring columns.

The final result of the curve detection module is a sequence of points (image pixel co-ordinates) composed of increasing segments with a single point *per* image column.

3.3. Calculation of precipitation intensities

The resulting sequence from the curve detection step of length n is transformed into a cumulative sequence of ordinate value differences of adjacent sequence elements (Equation (8)). From this cumulative sequence, the precipitation intensity (at this point in pixels) of any time interval $[i, j]$ can be calculated as $(\mathrm{cum}_j - \mathrm{cum}_i)$:

$$\begin{aligned} \mathrm{cum}_0 &= 0 \\ \mathrm{cum}_i &= \mathrm{cum}_{i-1} + \left(y_{i-1} - y_i\right) \quad 1 \le i \le n \end{aligned} \tag{8}$$

It is important to note in Equation (8) that since the image co-ordinate system origin is in the top left corner of the image, the value y_i is smaller than y_{i-1}.

To transform the rainfall intensities from pixels to mm time^{-1}, the ratio between the image pixels and rainfall quantity (mm) must be known.

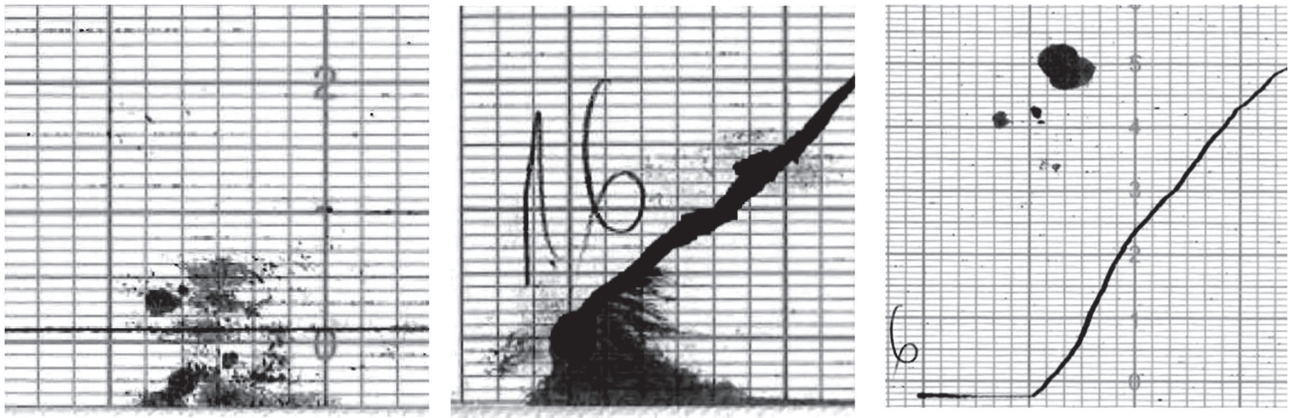

Figure 7. Examples of excess ink smudges.

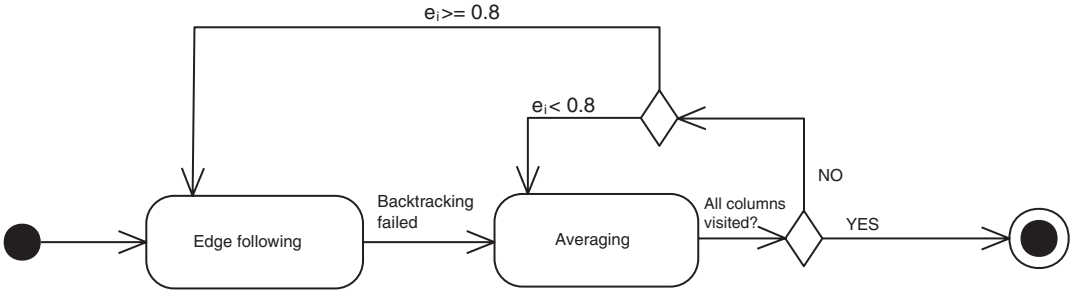

Figure 8. Curve extraction depends on the evidence factor e_i.

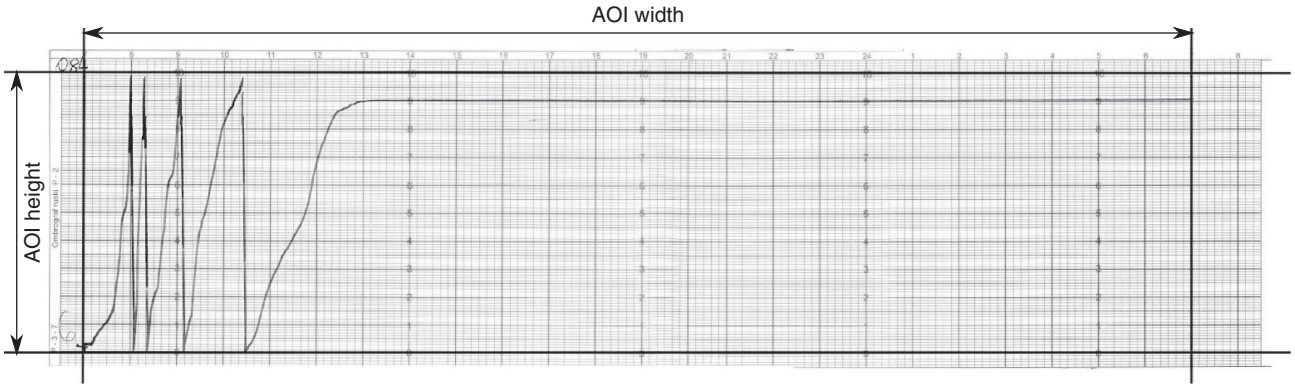

Figure 9. Area of interest.

This ratio can be obtained from the area of interest (Figure 9), where the rainfall curve should lie. This is done programmatically by finding the borders of this area using a simplified Hough transform (Trucco and Verri, 1998; Nixon and Aguado, 2002) to locate horizontal and vertical lines in constant search area of the image. This area is large enough to permit different input strip chart formats. These borders can easily be read from the strip chart image as the lines denoting 0, 10 mm and 0700 and 0700 (on the next day) on the chart axes. When the position and size of the area of interest in the image are found, the precipitation intensity of an arbitrary time interval $[t,\ t+m]$ is calculated as:

$$\text{Intensity}_{[t,\ t+m]} = \frac{\left(\text{cum}_{k+l} - \text{cum}_k\right) * (10\ \text{mm})}{(\text{area of interest height})} \tag{9}$$

where k and $k+l$ are indices of sequence cum corresponding to points in time t and $t+m$, respectively:

$$k = \left\lfloor \frac{t * \text{area of interest width}}{1440\ \text{min}} \right\rfloor$$
$$l = \left\lfloor \frac{m * \text{area of interest width}}{1440\ \text{min}} \right\rfloor \tag{10}$$

The ratio of the collector bucket capacity (10 mm) to the area of interest height in pixels defines the rainfall resolution, minimum rainfall intensity that can be measured. The ratio of the measurement time (1440 min) to the area of interest width in pixels defines the time resolution, the shortest time interval that can be measured. The constant 1440 is the number of minutes in 24 h.

4. Results

Two separate sets of strip chart images were used for the development of the algorithm. The implementation was based on the learning set containing eight images from the weather station in Kal nad Kanalom. Tests were performed on 58 strip chart images from the weather station in Podkraj that were randomly selected from the strip charts for the year 2006. For the quantitative analysis of the results, daily, hourly, half-hourly and 5 min precipitation intensity data were available. These data were registered by the Slovenian Environment Agency (ARSO) and were manually acquired by human operators using digitizer tablets.

4.1. Qualitative test

The qualitative analysis was based on a comparison between the detected curve plotline and the actual rainfall signal plotline. By displaying the pixels of the calculated pixel sequence over the actual image of a strip chart, the differences between the two plotlines were found. The largest difference (error) found was the basis for assigning the calculated result to one of predefined error classes. As the unit of measure, mm marked on the graph scale was chosen, which makes it easy to measure the difference. This unit of measure has no direct connection to rainfall intensity since it does not incorporate any time interval.

The error classes are defined as follows:

- Class 0: either no deviation from the curve plotline was found or the difference between the two curves was smaller than 0.2 mm.
- Class 1: minor deviation from the curve plotline was found. Maximum difference between the curves is between 0.2 and 0.5 mm. An example from this class is shown in Figure 10(a).
- Class 2: major deviation from the curve plotline was found. Maximum difference between the curves is between 0.5 and 10 mm. An example from this class is shown in Figure 10(b).
- Class 3: deviations that lead to more than 10 mm error in the calculation of the precipitation intensity were found (happens

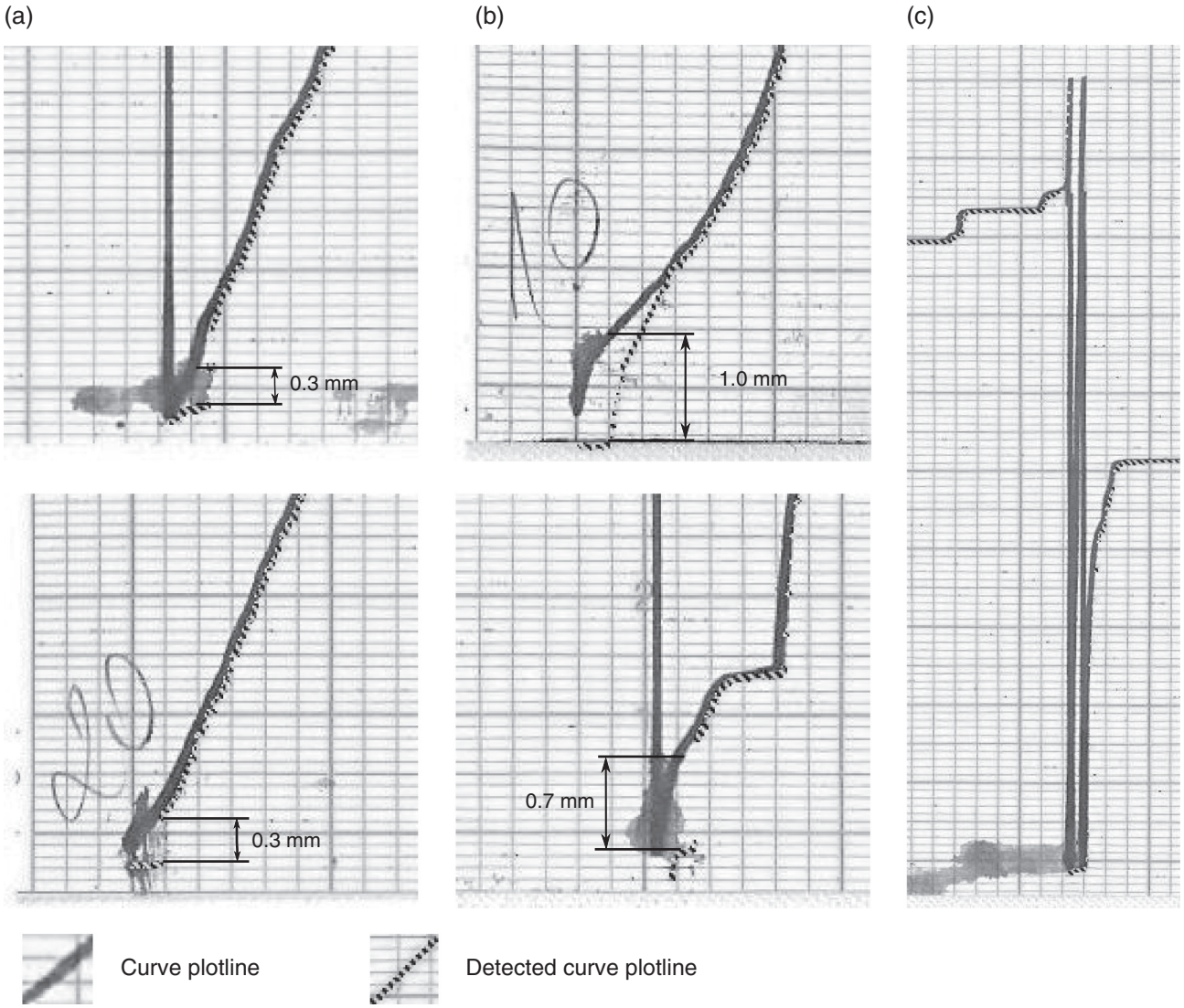

Figure 10. Examples from different error classes (a) error class 1, (b) error class 2 and (c) error class 3.

if the inversion points are not correctly detected). An example from this class is shown in Figure 10(c).

From Table 1, it is evident that 75.9% (Class 0) of strip chart images were correctly processed out of 58 scanned charts. In these cases, the curve plotline was accurately detected, resulting in an accurate rainfall intensity calculation.

4.2. Quantitative test

While the results of the qualitative test were based on visual inspection to determine whether there are any deviations from the rainfall signal curve, the quantitative test was performed to show how these deviations reflect on the computation of the precipitation intensity. The precipitation intensity values returned by our algorithm were compared to the values registered at the ARSO. Class 3 elements of the qualitative test were excluded from this test (they are not representative as they would require manual correction of errors). The strip charts from days that recorded more than 5 cm of snowfall were also excluded; the data on such strip charts do not show actual precipitation intensity. Finally, 45 strip charts were used in this test experiment.

The error was evaluated with measures that are ordinarily used to evaluate the accuracy of a regression curve fit. Mean absolute error (MAE) and relative mean absolute error (RMAE) (Kononenko and Kukar, 2007) are used because our problem can also be viewed as a search for a function that fits best to a set of points. They are calculated as:

$$\mathrm{MAE} = \frac{1}{I}\sum_{i=1}^{I}\left|f\,(i) - \widehat{f}\,(i)\right| \tag{11}$$

$$\mathrm{RMAE} = I\ \mathrm{MAE}\ \Big/ \sum_{i=1}^{I}\left|f\,(i) - \overline{f}\right| \tag{12}$$

where I represents the number of compared intervals, $f(i)$ is a recorded value at the ARSO and $\widehat{f}\,(i)$ is the algorithmically calculated value (for the ith interval). The average value $\overline{f}$ used in Equation (12) is calculated as $\overline{f} = \frac{1}{I}\sum_{i=1}^{I} f\,(i)$.

The MAE tells us for how many mm do the data differ on the average. The RMAE presents the error in the relative fashion, relative to the span of $f(i)$ values. The RMAE values lie between 0 and 1 where 0 means a perfect fit, with no deviation.

Table 1. Qualitative test results.

	Class 0	Class 1	Class 2	Class 3
No. of processed charts	44	6	3	5
Percentage of charts	75.9	10.3	5.2	8.6

Table 2. Quantitative test results.

Resolution	MAE (mm)	RMAE	*I*
Day	0.3844	0.0321	45
Hour	0.1350	0.0960	394
Half-hour	0.1055	0.1379	788
5 min	0.0563	0.3958	4728

MAE, mean absolute error; RMAE, relative mean absolute error.

Test results are shown in Table 2. These results show that the MAE is relatively small and gets smaller with the shortening of the time interval.

The daily and hourly results (RMAE = 0.0321 and RMAE = 0.0960, respectively) are very good. At higher time resolution, the deviation from the official (ARSO) data is larger. This is partly due to the difference in the way the data are acquired. While our algorithm determines the position of the curve plotline in every image column, the points marked with the use of a digitizer tablet are significantly sparser. Intermediate points are obtained by linear interpolation. These differences accumulate over time. The algorithmically acquired data, when the detection of the curve is correct, should thus be more accurate. The tests were performed on a double core Intel 1.7 GHz CPU personal computer running Linux operating system. The processing of the test set composed of 58 images took 5 min and 6 s with average processing time 5.28 s *per* image.

5. Conclusions

Accurate high-resolution rainfall data are needed in many research applications. These data can be acquired by automatic digital pluviometers. However, there remains a significant portion of rainfall data that were or still are recorded by non-digital automatic pluviograph gauges on strip charts. Conversion of these strip charts by human operators using digitizer tablets is a time-consuming and error-prone task.

To automate the digitization of the strip charts produced by float-based gauges, an algorithm that proves to be quite accurate and fast at the same time was developed and tested. The qualitative test shows that in 75.9% of the test cases, the detection of the curve was accurate enough for direct storage of the resulting precipitation intensities. For the rest of the cases, an interactive tool that would allow some manual adjustment of the rainfall curve is needed. The algorithm failure mode is generally due to the excessive ink smudges present on the strip charts.

Even though the algorithm was developed for processing strip charts from float-based gauges, the processing steps can either be adapted or directly used for digitization of other strip charts produced by instruments such as thermographs and hygrographs.

Acknowledgements

We thank the reviewers for their constructive comments and Prof. Franc Solina for his assistance in preparation of this study. For academic performance evaluation, both Aleš Jaklič and Luka Šajn should be considered as lead authors of this study.

References

Bahat Y, Grodek T, Lekach J, Morin E. 2009. Rainfall-runoff modeling in a small hyper-arid catchment. *J. Hydrol.* **373**(1–2): 204–217.
Bedient PB, Huber WC, Vieux BE. 2007. *Hydrology and Floodplain Analysis*, 4th edn. Prentice Hall PTR: Upper Saddle River, NJ.
Boni G, Parodi A, Rudari R. 2006. Extreme rainfall events: learning from raingauge time series. *J. Hydrol.* **327**(3–4): 304–314.
Cormen TH, Leiserson CE, Rivest RL, Stein C. 2001. *Introduction to Algorithms*, 2nd edn. The MIT Press: Cambridge, MA.
Cowpertwait PSP, O'Connell PE, Metcalfe AV, Mawdsley JA. 1996. Stochastic point process modelling of rainfall. II.Régionalisation and disaggregation. *J. Hydrol.* **175**(1–4): 47–65.
Deidda R, Mascaro G, Piga E, Querzoli G. 2007. An automatic system for rainfall signal recognition from tipping bucket gage strip charts. *J. Hydrol.* **333**(2–4): 400–412.
Dirks KN, Hay JE, Stow CD, Harris D. 1998. High-resolution studies of rainfall on Norfolk Island. Part II: interpolation of rainfall data. *J. Hydrol.* **208**(3–4): 187–193.
Forsyth DA, Ponce J. 2002. *Computer Vision – A Modern Approach*. Prentice Hall: Upper Saddle River, NJ.
Giesecke J, Haberlandt U. 1998. Precipitation data requirements for urban hydrology. *Water Int.* **23**(2): 60–66.
Gusfield D. 1997. *Algorithms on Strings, Trees and Sequences: Computer Science and Computational Biology*. Cambridge University Press: New York, NY.
Horiuchi S, Shimaya N, Nurzynska K, Kubo M, Muramoto K, Fujiyoshi Y. 2011. Estimation of the density of the snowfall particle using electric balances and 2DVD. In *2011 Proceedings of SICE Annual Conference (SICE)*, 13–18 September 2011, Tokyo, Japan; 1849–1854.
Jähne B. 2002. *Digital Image Processing*, 5th edn. Springer-Verlag: Berlin, Heidelberg.
Kononenko I, Kukar M. 2007. *Machine Learning and Data Mining: Introduction to Principles and Algorithms*. Horwood Publishing Limited: Woodgate, UK.
Koutsoyiannis D, Onof C. 2001. Rainfall disaggregation using adjusting procedures on a Poisson cluster model. *J. Hydrol.* **246**(1–4): 109–122.
Mclntyre N, Al-Qurashi A. 2008. Performance of ten rainfall-runoff models applied to an arid catchment in Oman. *Environ. Model. Softw.* **24**(6): 726–738.
Marien JL, Vandewiele GL. 1986. A point rainfall generator with internal storm structure. *Water Resour. Res.* **22**(4): 475–482.
Mat-Isa AN, Mashor YM, Othman HN. 2005. Seeded region growing features extraction algorithm; its potential use in improving screening for cervical cancer. *Int. J. Comput. Internet Manage.* **13**(1): 61–70.
Nixon MS, Aguado AS. 2002. *Feature Extraction and Image Processing*. Newnes: Elsevier, Oxford.
Schilling W. 1991. Rainfall data for urban hydrology: what do we need? *Atmos. Res.* **27**(1–3): 5–21.
Trucco E, Verri A. 1998. *Introductory Techniques for 3-D Computer Vision*. Prentice Hall PTR: Upper Saddle River, NJ.
Woolhiser DA, Osborn HB. 1985. A stochastic model of dimensionless thunderstorm rainfall. *Water Resour. Res.* **21**(4): 511–522.

