Failure of the 2D version of the step and roof edge detector derived from a competitive filter

Leszek Chmielewski

Division of Optical and Computer Methods in Mechanics
Institute of Fundamental Technological Research, PAS
WWW: www.ippt.gov.pl/~lchmiel/

December 1997

1 Introduction

The number of various concepts of step-type edge detectors is very large, whereas quite little has been written on detection of roof-type edges in digital images. This was the reason why an attempt has been made to address this challenging task.

The authors who have managed to design a working roof edge detector must have overcome the problem of finding a discontinuity in a first derivative of a 2-dimensional discrete function with noise, which is a difficult task [BZ87, PH93, Kur94]. The main reason why this task is so difficult is, to say it in simple words, that the derivative of a noisy function is even more noisy.

It seemed that the new type of filter called the *competitive filter* first proposed by Niedźwiecki and Sethares [NS91] and then generalized to two dimensions by Niedźwiecki and Suchomski [NS91] has good potential to become a basis for an edge detector with enough stability to detect discontinuities in the first derivative. The idea of how a detector based on the filter of interest can be constructed has been introduced by Chmielewski for a one-dimensional case in [Chm94] and for a two-dimensional case in [Chm96a]. The preliminary results for a one-dimensional case presented in [Chm96a] were promising, so further studies were carried out [Chm96b, Chm96c].

Unfortunately, in spite of quite good results obtained in the 1D case, the 2D version of the detector appeared not to perform properly.

2 1D case: promising results

The theory of the 1D filter and detector has been reminded in Appendix C. The most mature version [Chm96c] has been described.

With this detector a following experiment has been made to study its performance for noisy data. A series of calculations were made for a single 1D discrete function with no noise, and with added Gaussian noise of gradually increasing standard deviation: $\sigma = 0, 5, 10, 20$ and 30[1]. The function has a step and roof edge in the middle, and in the best case the location of this edge and its two intensities (*jump* for the step and *angle* for the roof, see Fig. 22 in Appendix C) should be constant and equal to that for a noiseless case, irrespective of noise intensity.

The graphs of edge location and intensities versus noise intensity, named the *evolution graphs*, for two values of scale (λ, see Appendix C for explanation) have been shown in Figs. 1-4. Examination of graphs in these Figures reveals that for smaller scale $\lambda = 5$ the results were acceptable up to $\sigma = 10$ and then degraded, whereas good stability has been obtained for larger scale $\lambda = 10$: roof edge value and location remained practically unchanged even for the largest noise, step edge behaved properly up to $\sigma = 20$, and for larger noise the value was incorrect and false edges appeared.

The source graphs of filtering/edge detection results for the 1D case evolution graphs in Figs. 1-4 can be found in Appendix A.

The observation that edge detection with large scale yields positive results has led to expectation that it will be similarly with the 2D version of the detector.

[1] For each σ a new noise realisation from a noise generator was used.

3 2D case: problems and failure

In the implementation of the 2D edge detector tested in this report the concept of *leaves* has been applied (see Fig. 25 in Appendix D).

The experiments with the 1D detector indicate that good results can not be obtained with a small scale λ. The question what scale is small is always related to how much detail is expected to be seen in the image. Many authors attempt to preserve as small details in the filtered image as possible. Therefore, the scale $\lambda = 5$ has been treated here as large, and $\lambda = 3$ as small. The number of pixels in one leaf is 9 and 25, respectively, so in the case of the larger scale the number of data points used for extrapolation is much larger than in the 1D case, where 10 points were enough.

A series of calculations similar to those carried out for a 1D case have been made. For presentation purpose, one row of a test image with a step edge has been selected, to make possible the same presentation technique as previously. The results in the selected row are neither extraneously bad nor good.

The received evolution graphs, that is, edge intensities and location versus noise standard deviation, are shown in Figs. 5-8. The quality of results for scale $\lambda = 3$ is very bad. Even for the smallest noise $\sigma = 5$ the false edges appear, the intensity of the step edge at theoretical location is very erroneous, and many strong false roof edges appear. For scale $\lambda = 5$ the results are slightly better, but still a large number of false roof edges appear even for small noise.

The source images of filtering/edge detection results for the 2D case evolution graphs in Figs. 5-8 can be found in Appendix B.

4 Conclusion

A question arises why the results are unsatisfactory. Two main factors can be indicated here.

- **Bad quality of prediction (extrapolation).** While the linear approximation of the brightness function yields a good approximation in the middle of the leaf (function support), the result of extrapolation is corrupted by the cumulative action of height and angle estimation errors.

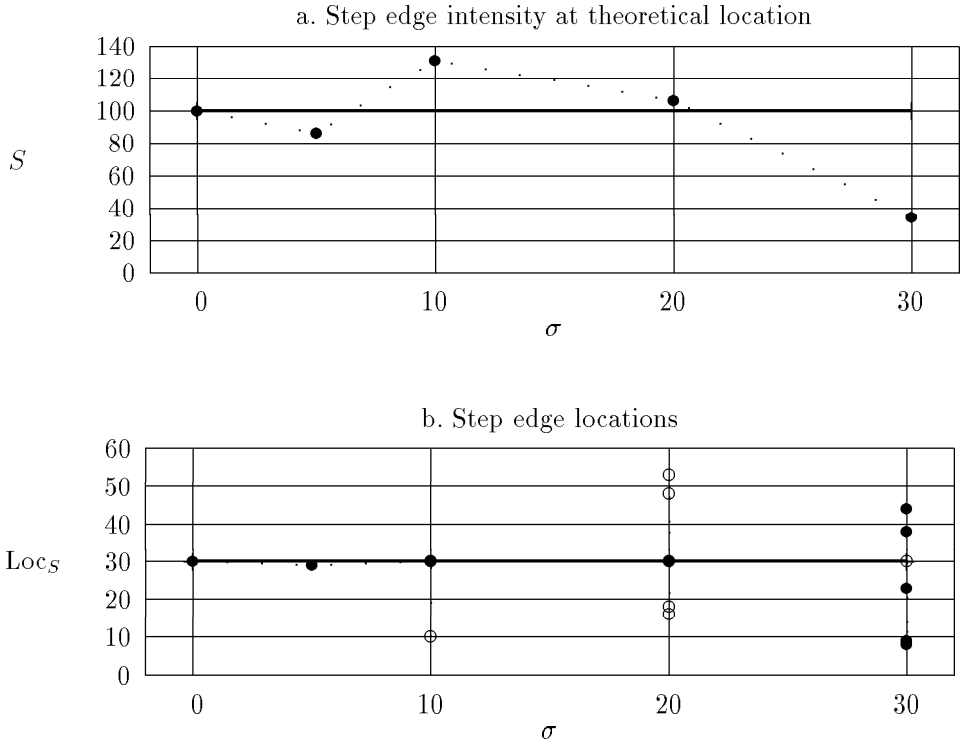


Fig.1: 1D case: evolution of step value and location with noise increase. Scale $\lambda = 5$. Filled points – strong, significant edges; empty points – weaker, but significant edges; thick line – theoretical value.

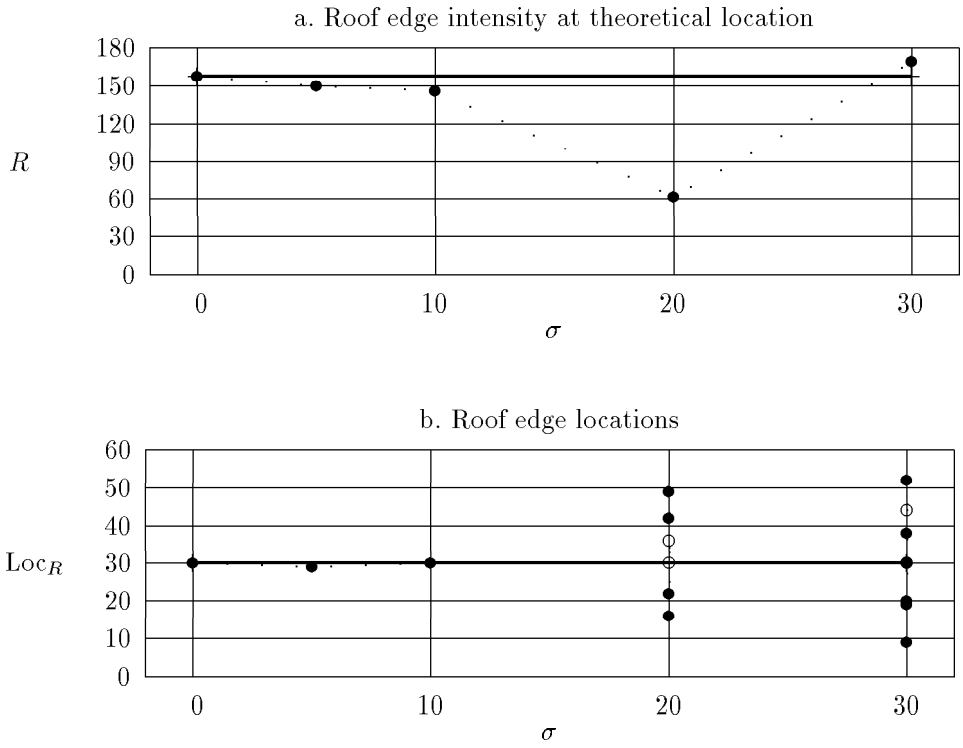


Fig.2: 1D case: evolution of roof value and location with noise increase. Scale $\lambda = 5$. Filled points – strong, significant edges; empty points – weaker, but significant edges; thick line – theoretical value.

- **Weakness of the edge existence condition.** This purely heuristic condition is the result of experiments with a number of various conditions presented in [Chm96a, Chm96b, Chm96c]. The quality of the condition could have been tested in a better way in the 1D case, where the visualisation is easier, while when applied to the 2D case the same condition failed.

Although the amount of work sacrificed to the development of the new edge detector based on a competitive filter has been considerable, the final conclusion is that the overall result is negative, and the work on this concept should be discontinued.

References

[BZ87] A. Blake and A. Zisserman. *Visual Reconstruction.* MIT Press, 1987.

[Chm94] L. Chmielewski. The concept of step and roof edge detector derived from a competitive filter. EPSILON ARG Report, Nov 1994.

[Chm96a] L. Chmielewski. The concept of a competitive step and roof edge detector. *MG&V*, 5(1-2):147–156, 1996.

[Chm96b] L. Chmielewski. The concept of step and roof edge detector derived from a competitive filter – a new condition of edge existence. Report of the Division of Optical and Computer Methods in Mechanics, IFTR PAS, and the Signal Processing and Understanding Group, DIBE, Univ. Genoa, Italy, Mar 1996.

[Chm96c] L. Chmielewski. The concept of step and roof edge detector derived from a competitive filter – new edge intensity measure. Report of the Division of Optical and Computer Methods in Mechanics, IFTR PAS, and the Signal Processing and Understanding Group, DIBE, Univ. Genova, Italy, Sept 1996.

[HW91] R.M. Haralick and L. Watson. A facet model for image data. *CVGIP*, 15:113–129, 1991.

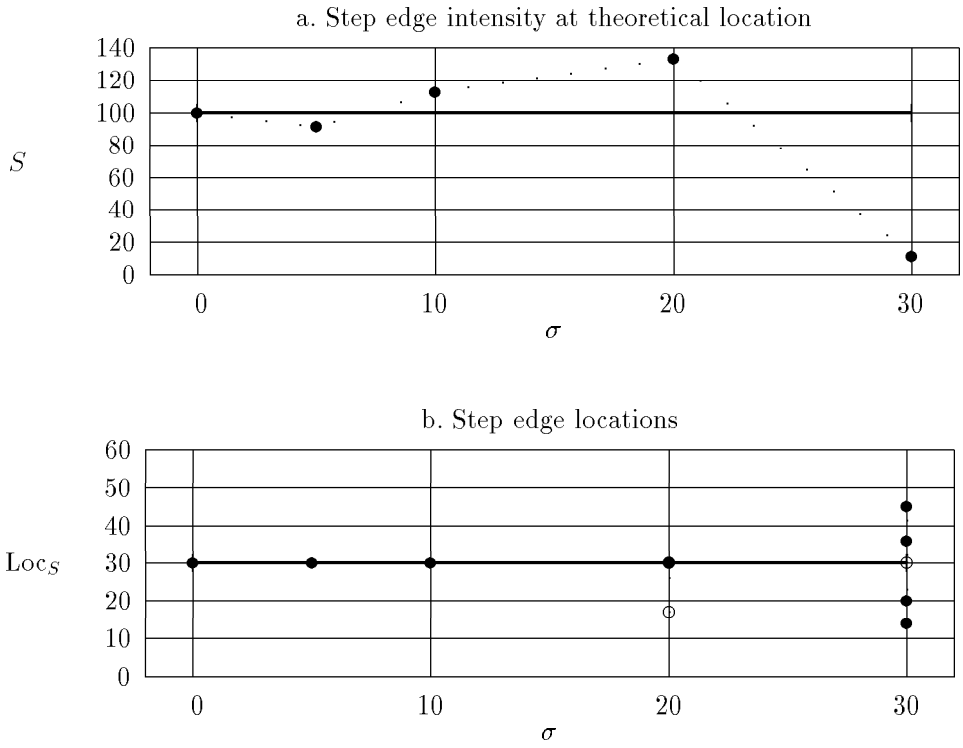


Fig.3: 1D case: evolution of step value and location with noise increase. Scale $\lambda = 10$. Filled points – strong, significant edges; empty points – weaker, but significant edges; thick line – theoretical value.

[KKM+89] D. Y. Kim, J. J. Kim, P. Meer, D. Mintz, and A. Rosenfeld. Robust computer vision: a Least Median of Squares based approach. In *Proc. DARPA Image Understanding Workshop*, pages 1117–1134, Palo Alto, California, May 1989.

[Kur94] A. Kuriański. Detection of image features for stereoscopic scene modelling (in Polish). ICS PAS Reports 765, Inst. of Computer Science, PAS, Warsaw, Dec 1994.

[NS91] M. Niedźwiecki and W. A. Sethares. New filtering algorithms based on the concept of competitive smoothing. In *Proc. 23rd Int. Symp. on Stochastic Systems and Their Applications*, pages 129–132, Osaka, 1991.

[NS94] M. Niedźwiecki and P. Suchomski. On a new class of edge-preserving filters for noise rejection from images. *MG&V*, 1-2(3):385–392, 1994.

[PH93] T. Pajdla and V. Hlaváč. Surface discontinuities in range images. In *5th Int. Conf. Computer Vision*, pages 524–528, Berlin, May 1993. IEEE Computer Society Press.

[RL87] P. J. Rousseeuw and A. M. Leroy. *Robust Regression & Outlier Detection.* John Wiley and Sons, 1987.

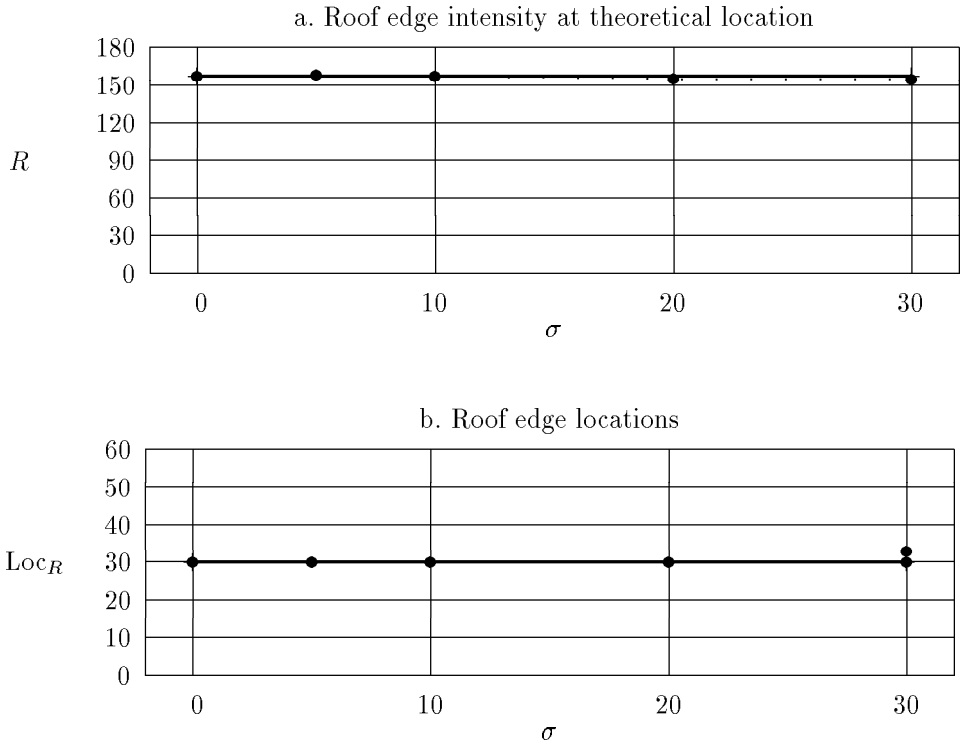


Fig.4: 1D case: evolution of roof value and location with noise increase. Scale $\lambda = 10$. Filled points – strong, significant edges; empty points – weaker, but significant edges; thick line – theoretical value.

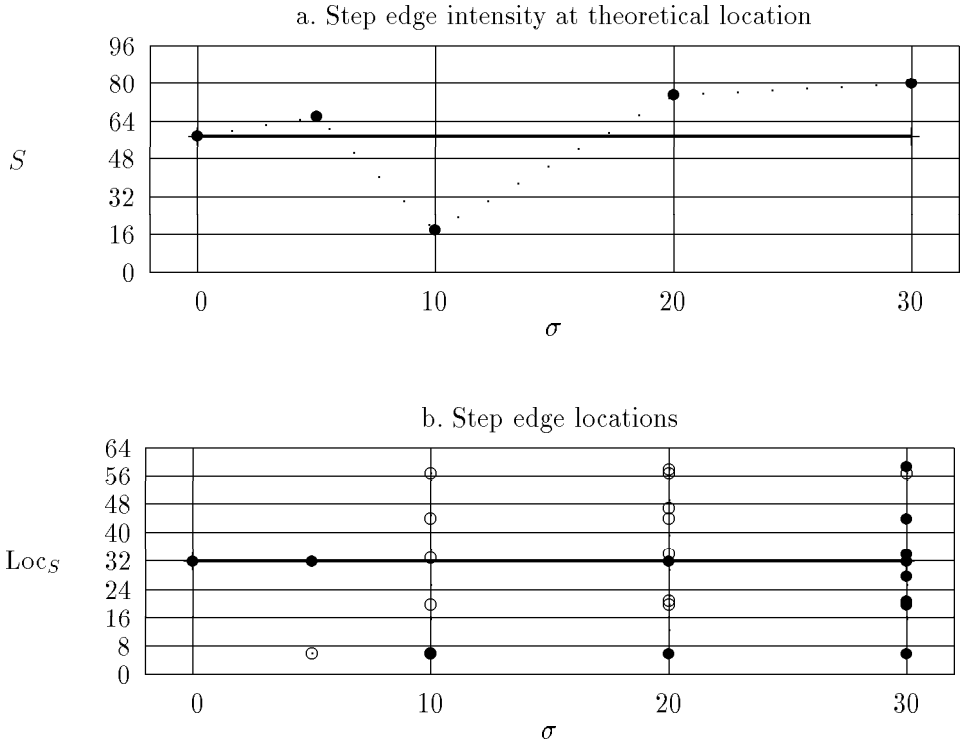


Fig.5: 2D case: evolution of step value and location with noise increase. Scale $\lambda = 3$. Filled points – strong, significant edges; empty points – weaker, but significant edges; thick line – theoretical value.

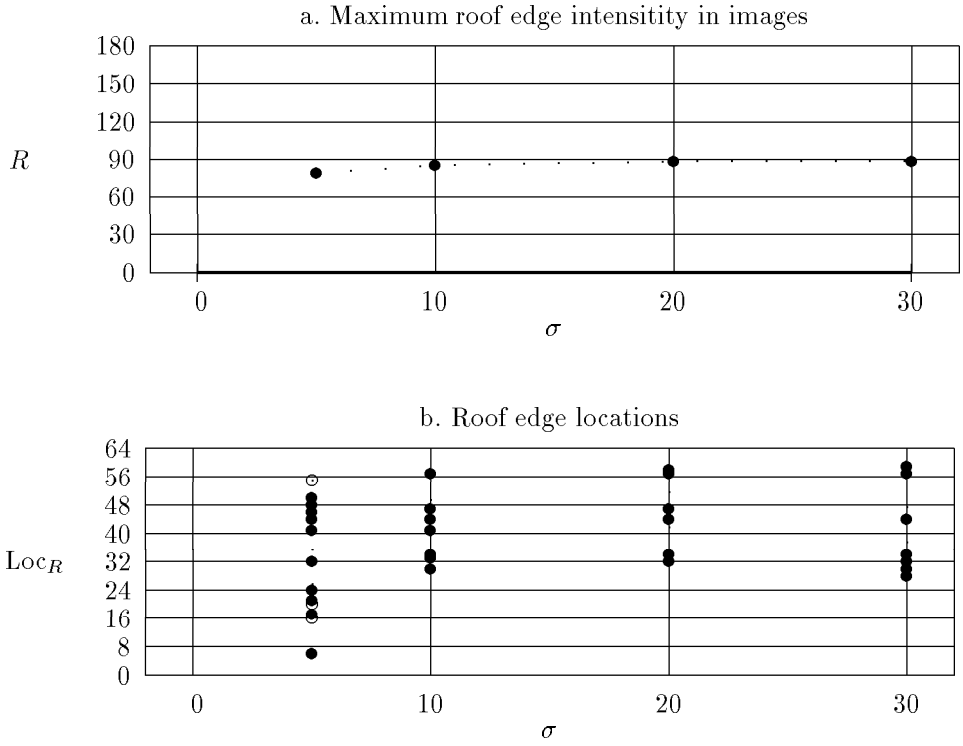


Fig.6: 2D case: evolution of roof value and location with noise increase. Scale $\lambda = 3$. Filled points – strong, significant edges; empty points – weaker, but significant edges; thick line – theoretical value (0); **there is no theoretical location** – image contains no roof.

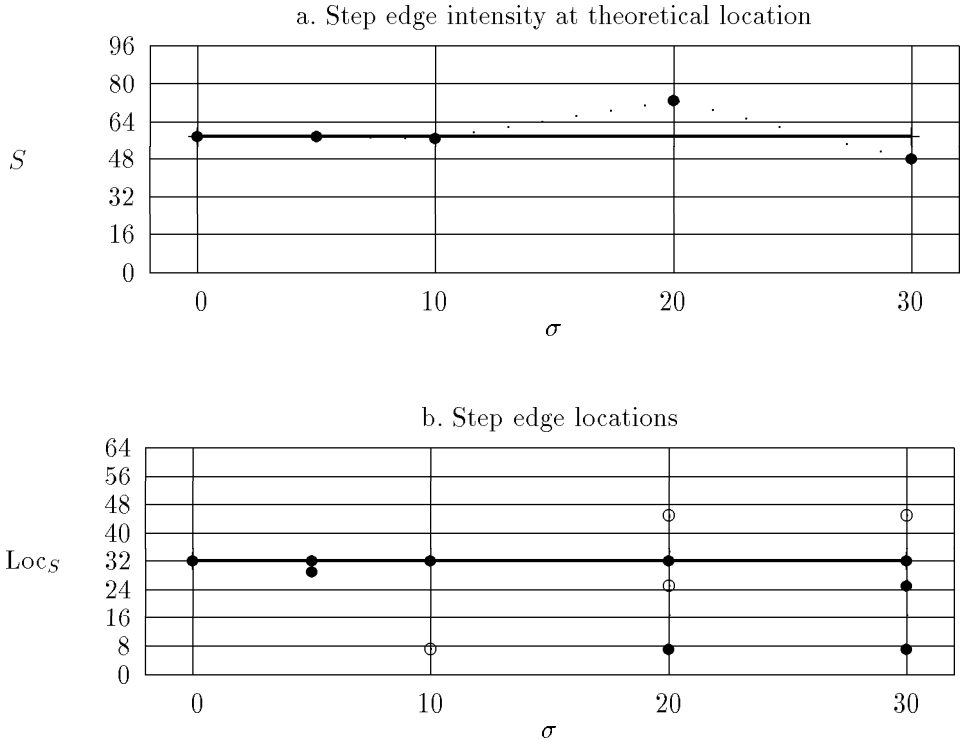


Fig.7: 2D case: evolution of step value and location with noise increase. Scale $\lambda = 5$. Filled points – strong, significant edges; empty points – weaker, but significant edges; thick line – theoretical value.

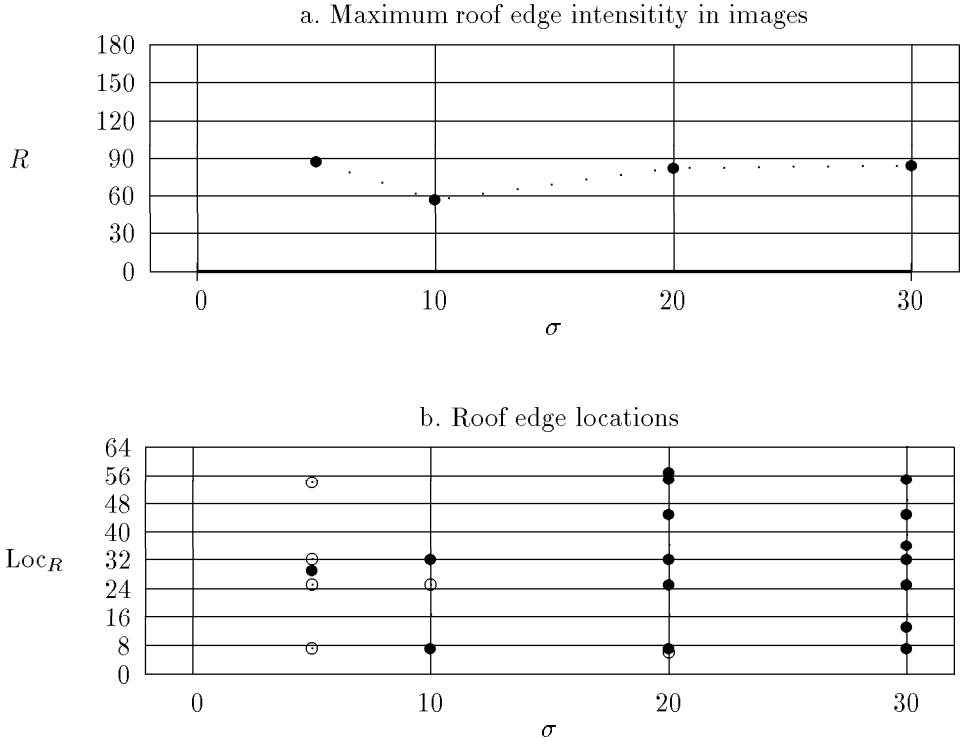


Fig.8: 2D case: evolution of roof value and location with noise increase. Scale $\lambda = 5$. Filled points – strong, significant edges; empty points – weaker, but significant edges; thick line – theoretical value (0); **there is no theoretical location** – image contains no roof.

Appendices

A Sources for the evolution graphs – 1D cases

The source graphs of filtering/edge detection results used to produce the evolution graphs presented in the paper are shown in Figs. 9-13 for scale $\lambda = 5$ and in Figs. 14-18 for scale $\lambda = 10$.

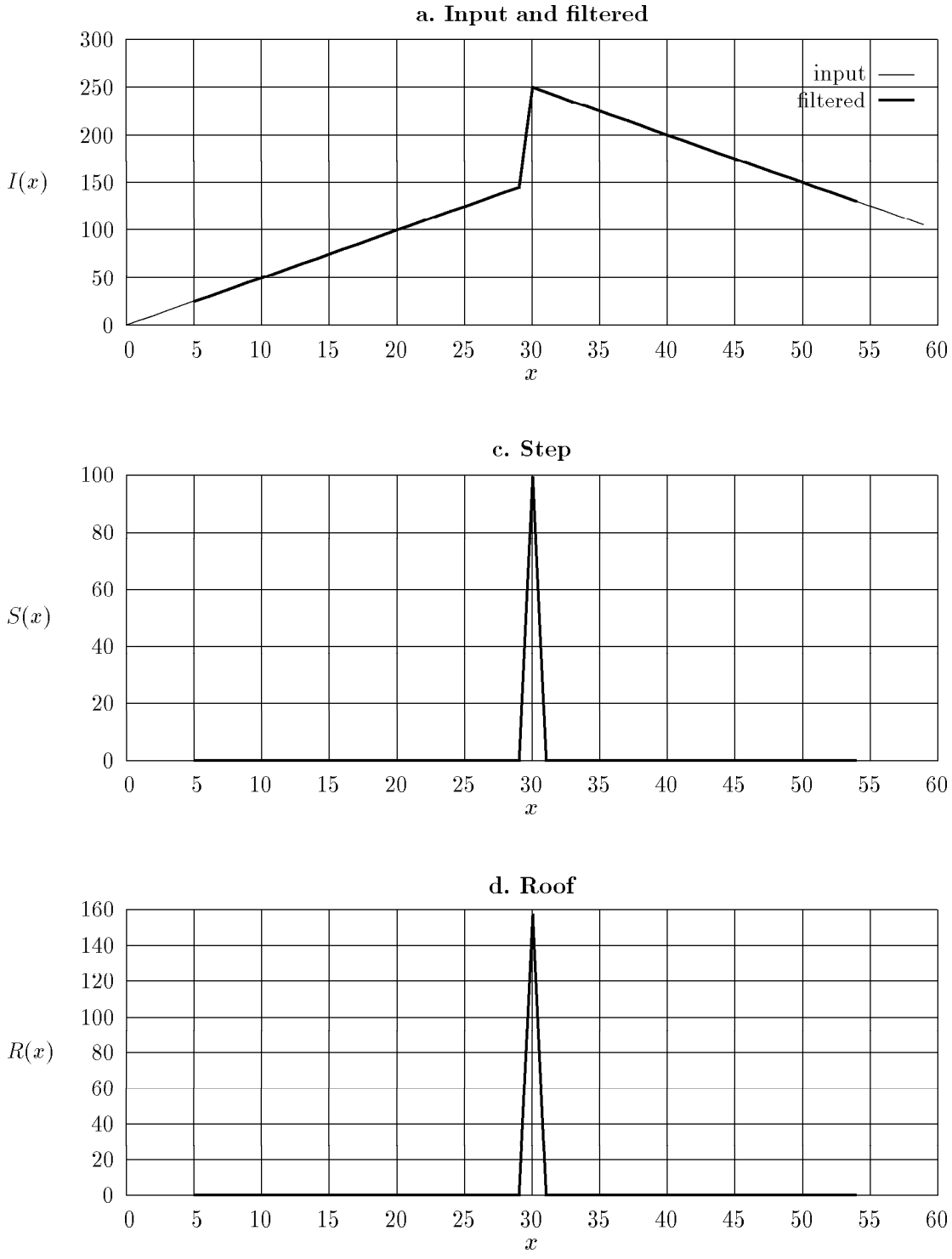


Fig.9: Results for an artificial step-roof with noise $\sigma = 0$ at scale $\lambda = 5$.

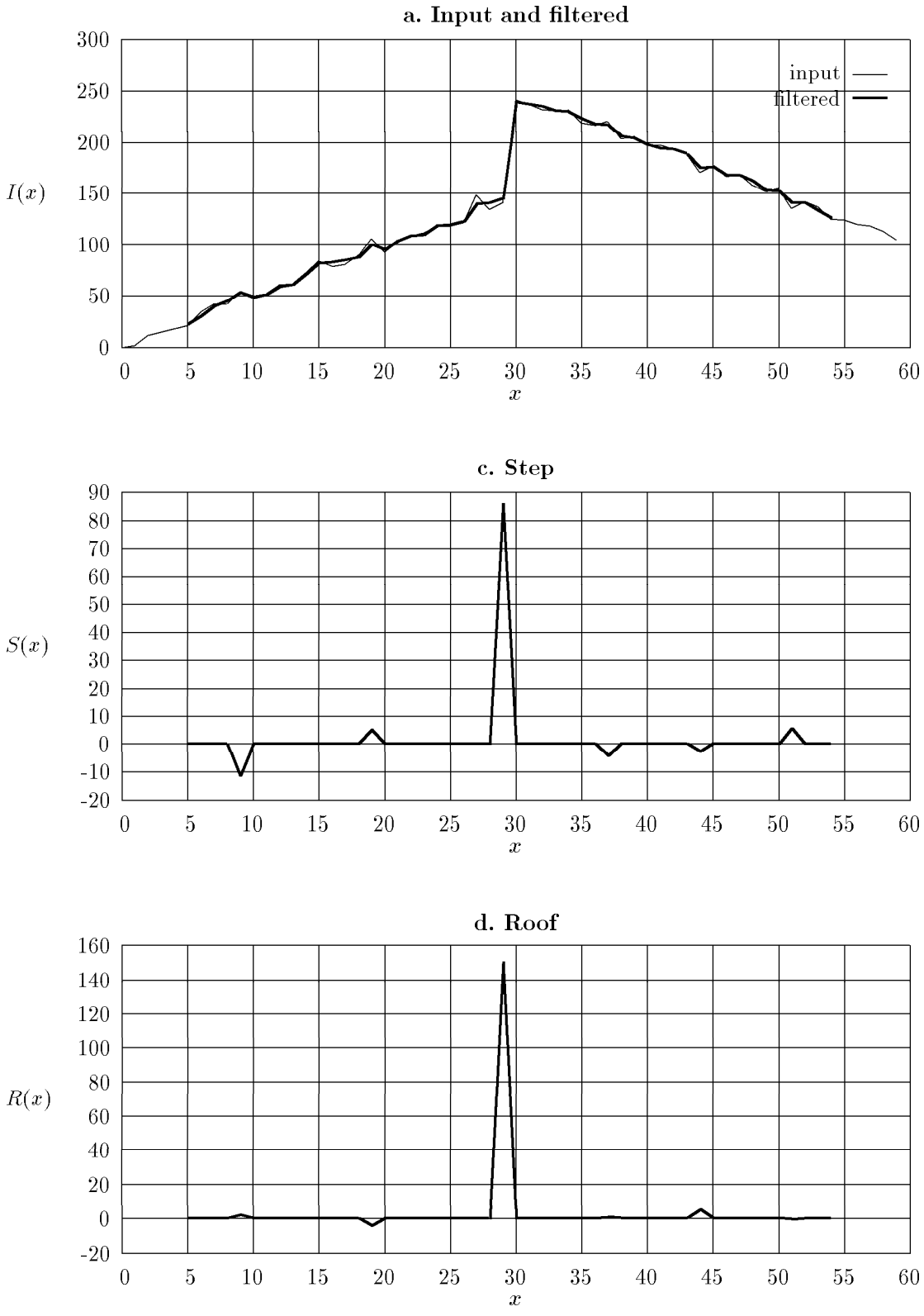
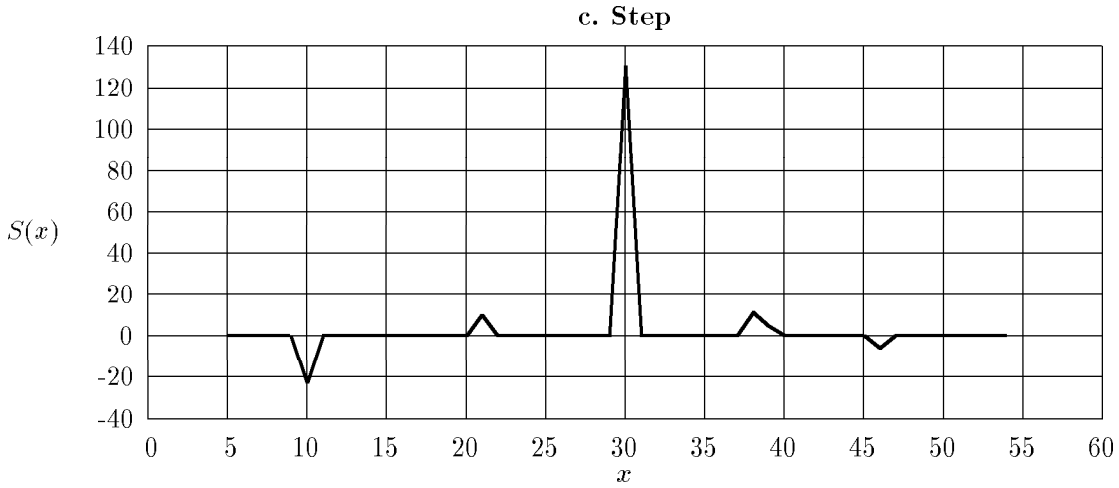


Fig.10: Results for an artificial step-roof with noise $\sigma = 5$ at scale $\lambda = 5$.



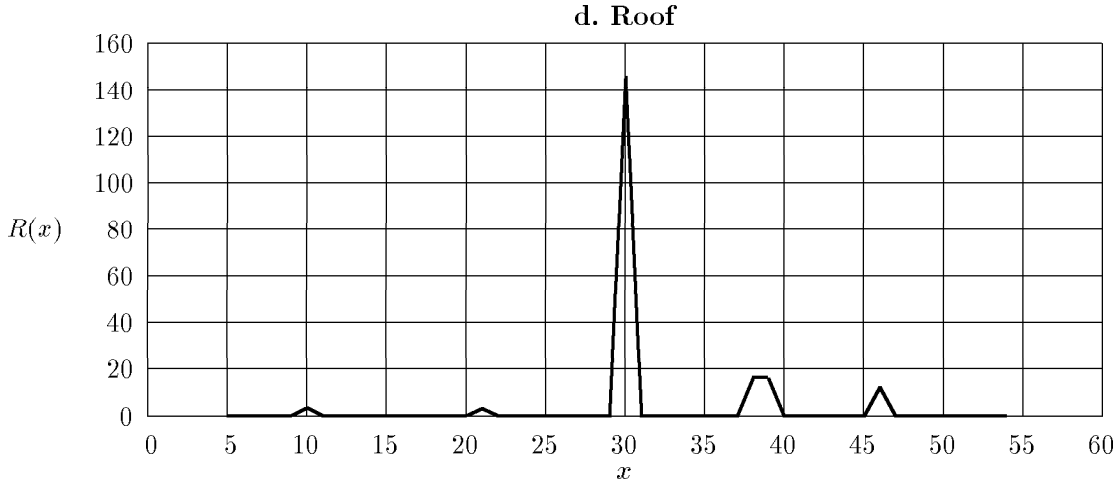


Fig. 11: Results for an artificial step-roof with noise $\sigma = 10$ at scale $\lambda = 5$.

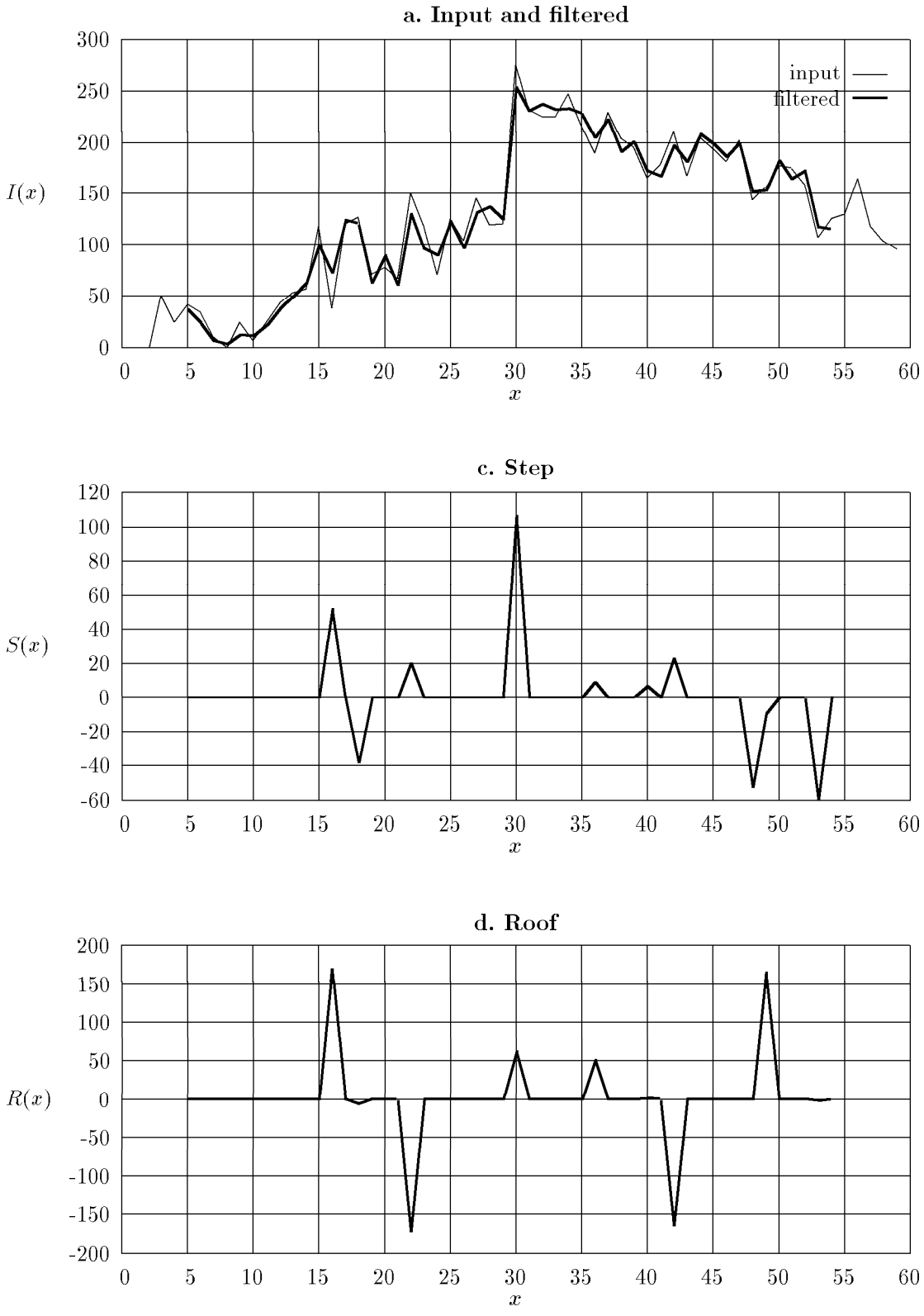


Fig. 12: Results for an artificial step-roof with noise $\sigma = 20$ at scale $\lambda = 5$.

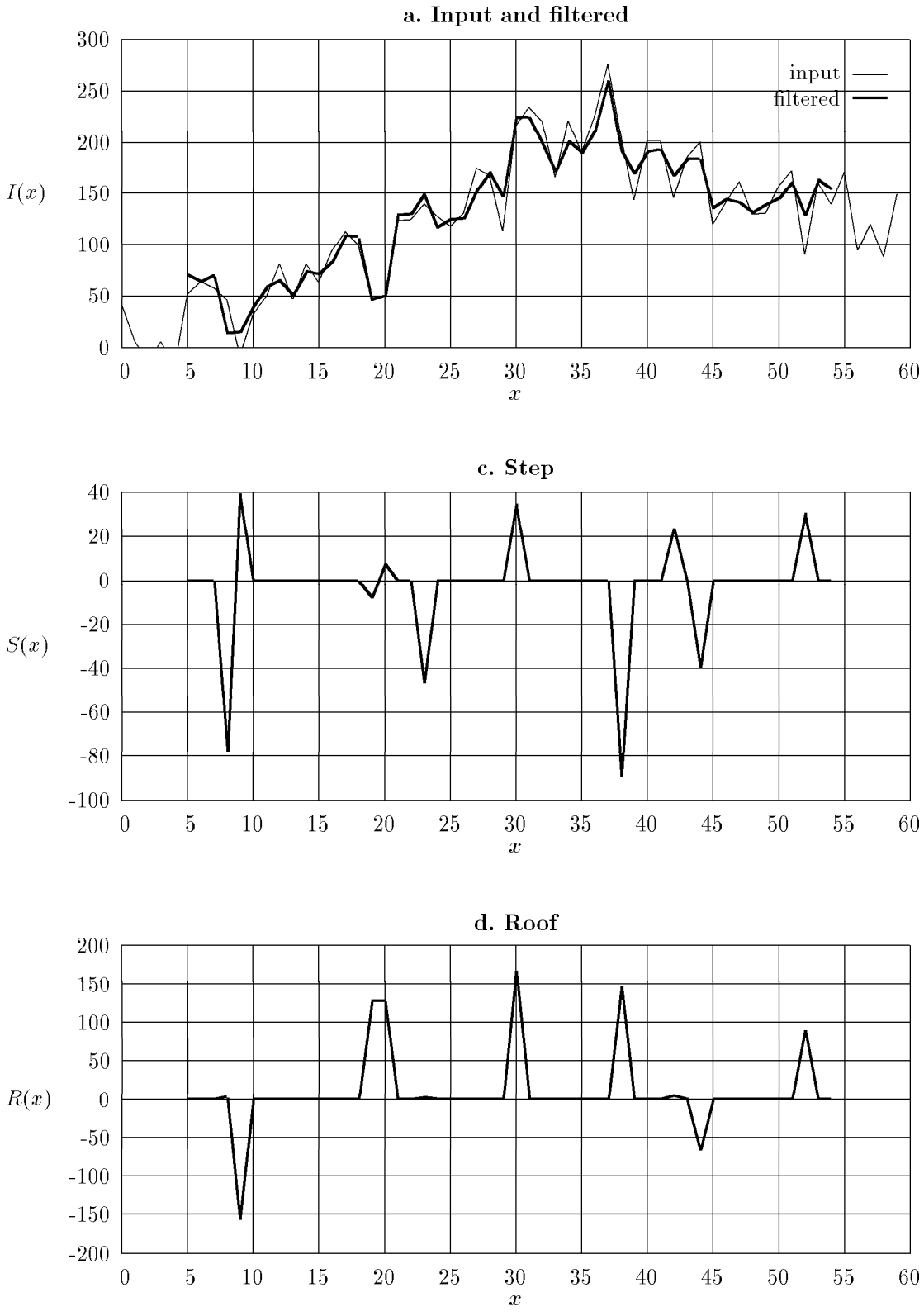


Fig. 13: Results for an artificial step-roof with noise $\sigma = 30$ at scale $\lambda = 5$.

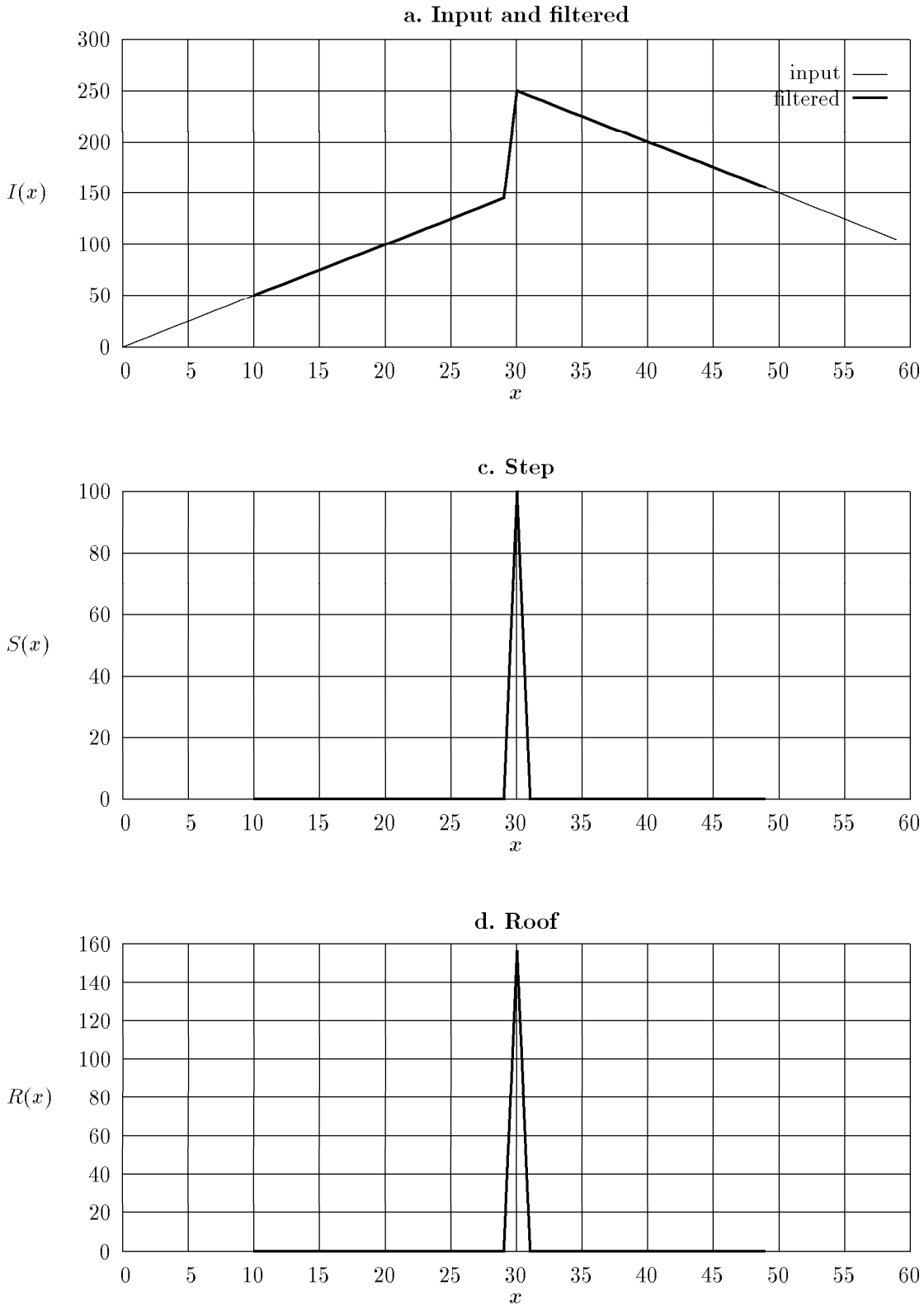
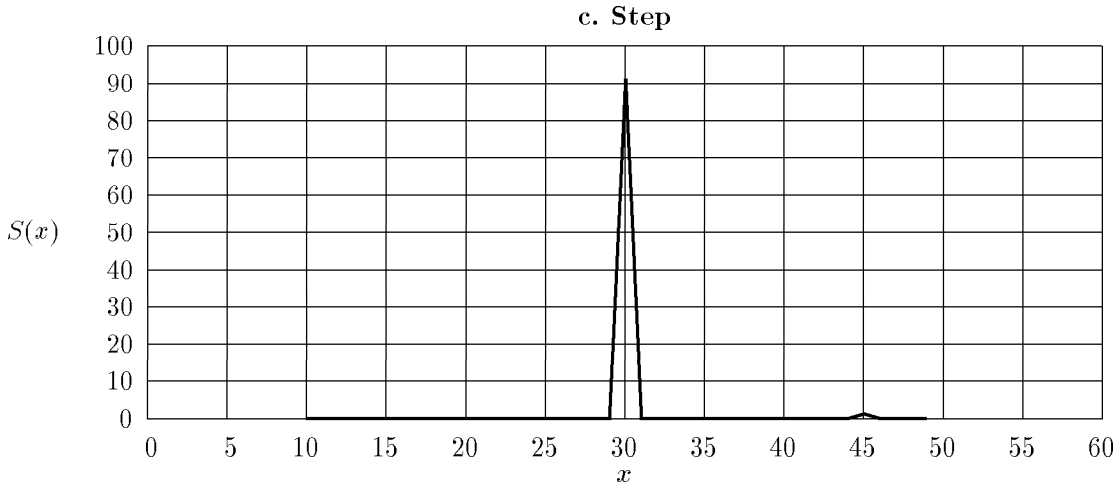


Fig. 14: Results for an artificial step-roof with noise $\sigma = 0$ at scale $\lambda = 10$.



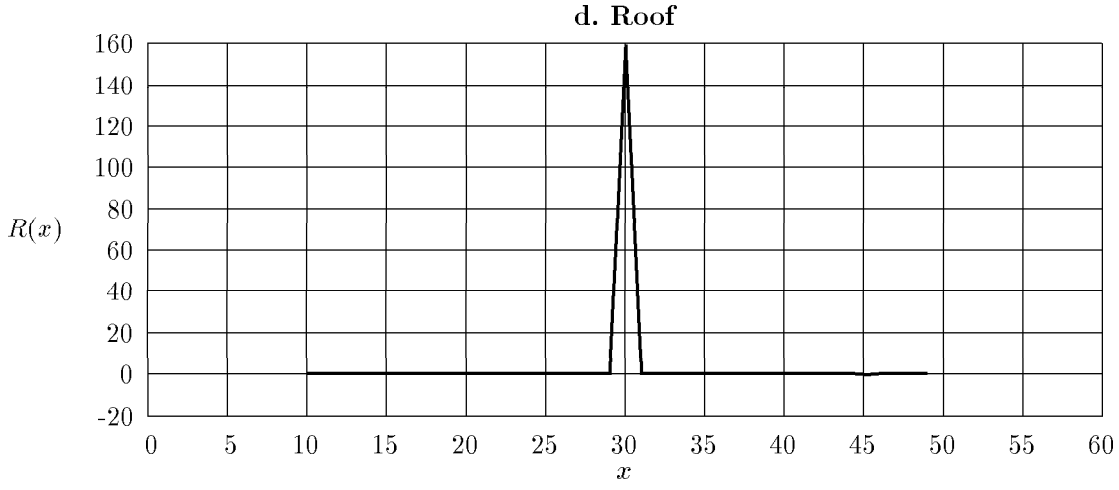
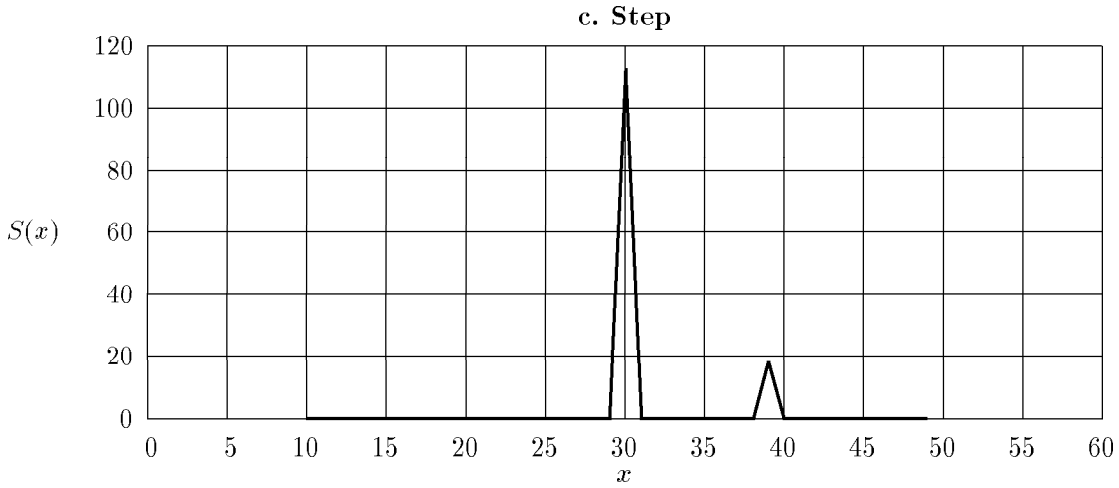


Fig.15: Results for an artificial step-roof with noise $\sigma = 5$ at scale $\lambda = 5$.



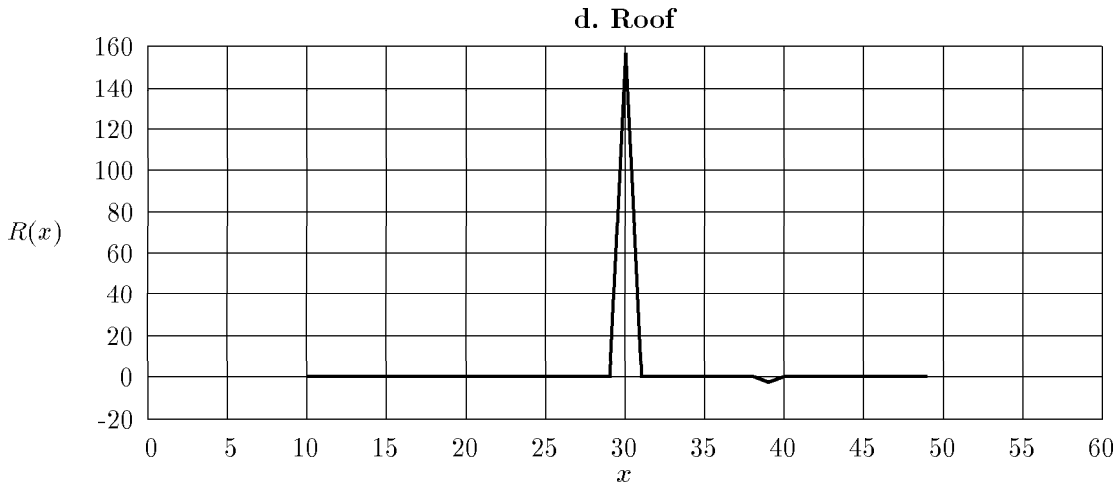


Fig.16: Results for an artificial step-roof with noise $\sigma = 10$ at scale $\lambda = 10$.

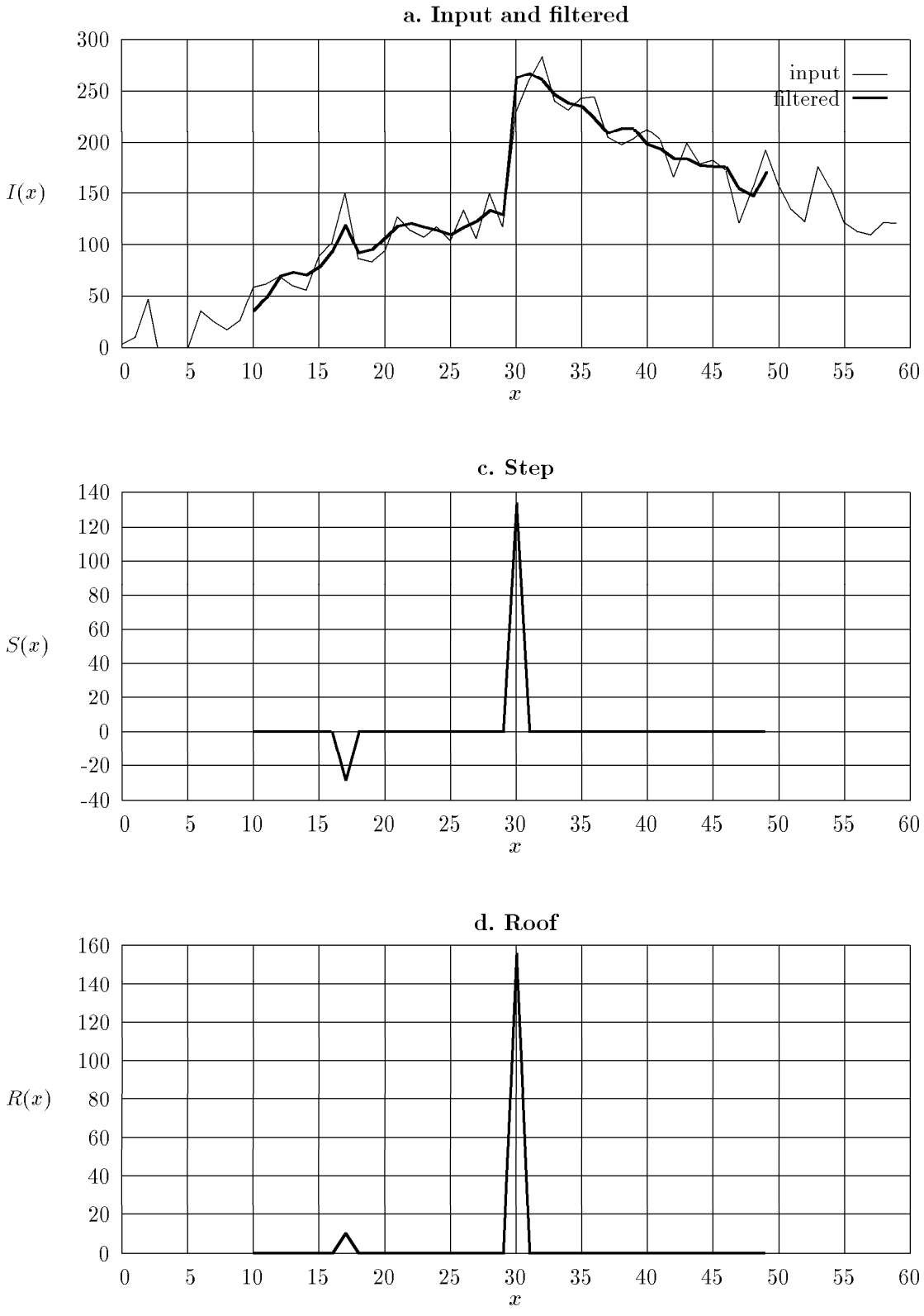


Fig.17: Results for an artificial step-roof with noise $\sigma = 20$ at scale $\lambda = 10$.

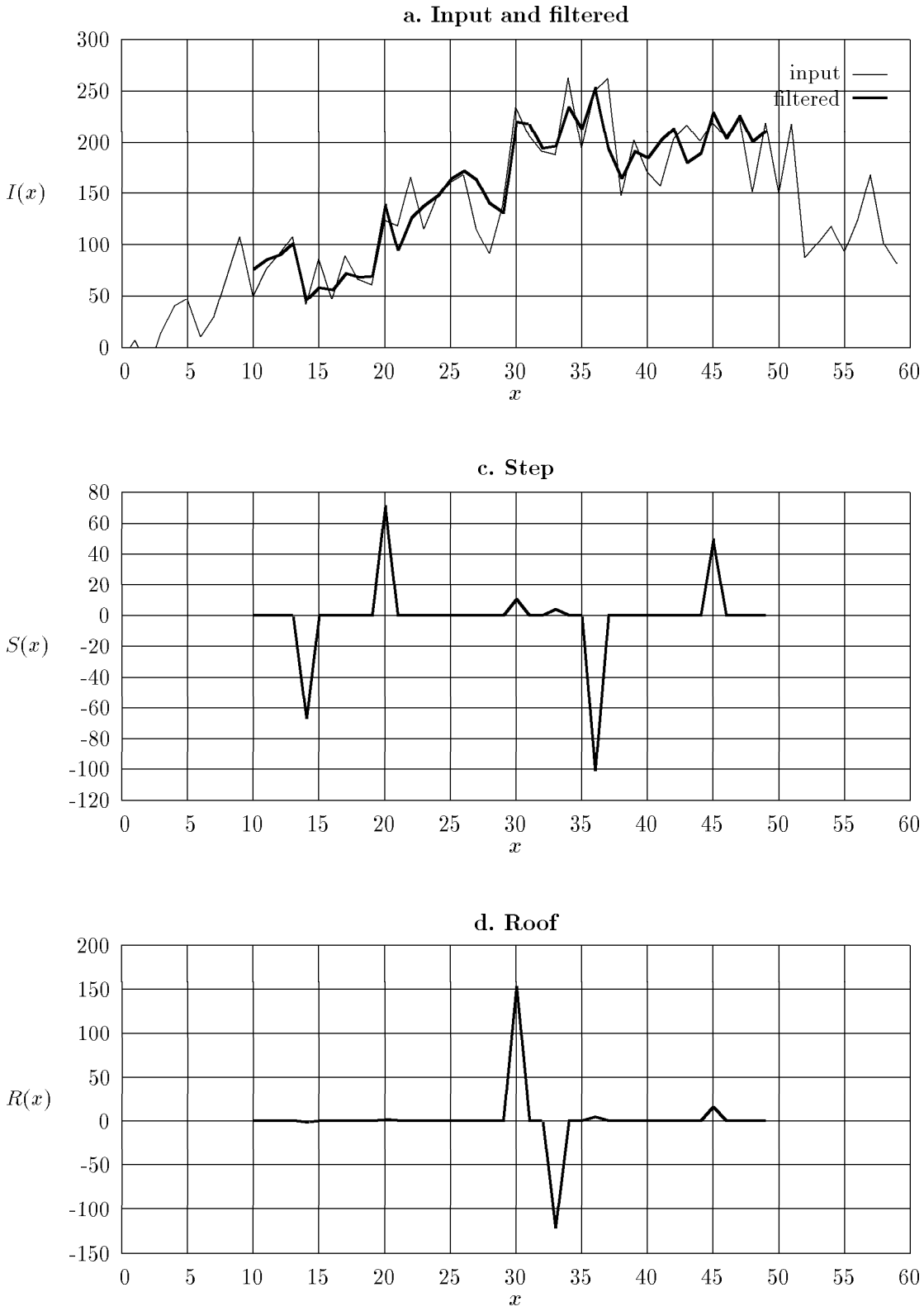


Fig.18: Results for an artificial step-roof with noise $\sigma = 30$ at scale $\lambda = 10$.

B Source images for the evolution graphs – 2D cases

The source images of filtering/edge detection results used to produce the evolution graphs presented in the paper are shown in Fig. 19 for scale $\lambda = 3$ and in Fig. 20 for scale $\lambda = 5$.

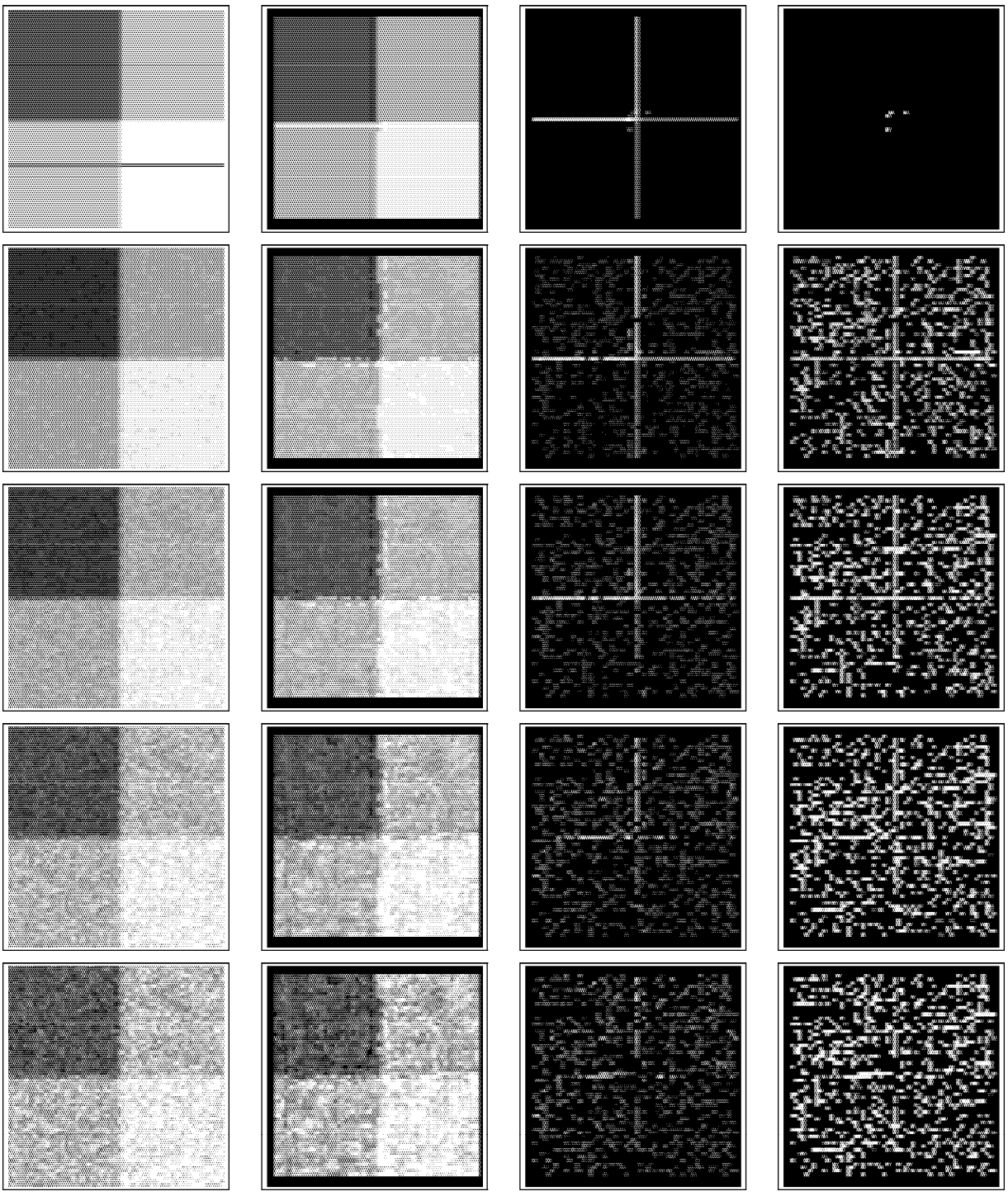


Fig.19: From left to right: input $(64 * 64)$, filtered, step, roof; From top to bottom: noise $\sigma = 0, 5, 10, 20, 30$; scale $\lambda = 3$ (leaf dimensions $3 * 3$).
In upper left image the row 45 used for evolution analysis is shown.

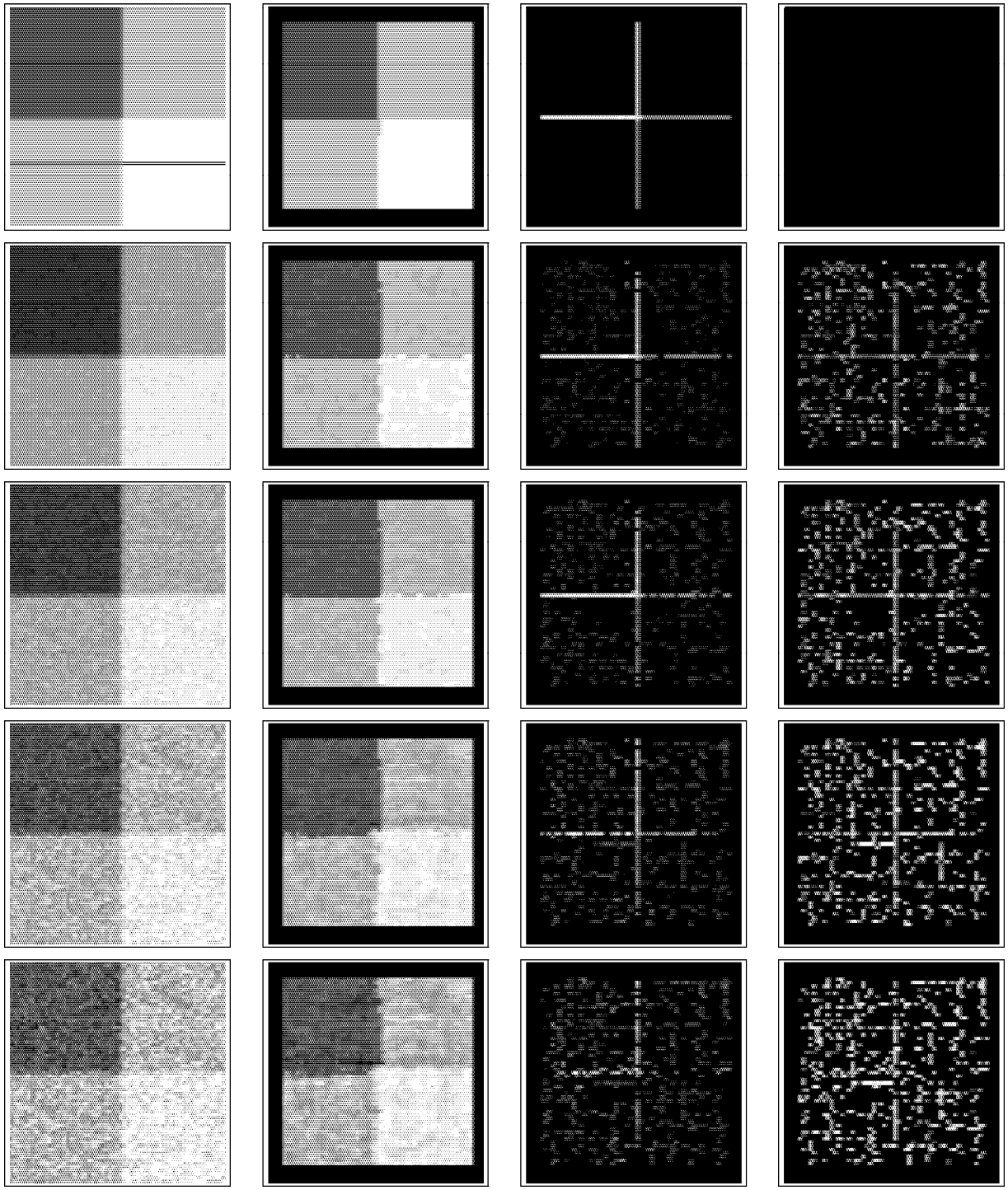


Fig. 20: From left to right: input ($64 * 64$), filtered, step, roof; From top to bottom: noise $\sigma = 0, 5, 10, 20, 30$; scale $\lambda = 5$ (leaf dimensions $5 * 5$).
In upper left image the row 45 used for evolution analysis is shown.

C The Competitive Edge Detector in 1D

C.1 The concept

Let us briefly remind the main idea of a 1D version of a competitive filter according to [NS94]. The aim was to find a time sequence $y(t)$ given the measurements $z(t)$, under the assumption that $z(t) = y(t) + n(t)$, where $n(t)$ was the measurement noise. Time t was considered as discrete, and the whole measurements sequence as given.

The main idea was to run two predictors on the data surrounding the considered point t_0. The first one was running towards future, and yielded $\hat{y}_-(t_0)$ as an estimate of $y(t_0)$, basing on observations from the past with respect to t_0. The second one, running towards past, yielded $\hat{y}_+(t_0)$, basing on the future w.r.t. t_0. As the output of the filter the result of that predictor was chosen which had a lower error index, that is, the one which performed better in a number of its recent predictions in the sense of square mean of prediction errors.

Let us illustrate a simplified competitive filter [NS94], in which as a predicted value the average of the nearest λ (*scale* of the filter) measurements is taken. (Fig. 21). As the error index the mean square fitting errors can be used. In the figure, the future predictor wins.

In the following we shall consider spatial distributions of brightness functions $I(x)$ rather than time sequences, so the notations will change accordingly.

Let us now assume that the fitted functions is affine rather than constant. The method of fitting is arbitrary; at the first thought we would take the least mean squares (LMS) fitting, while for example the least median of squares [KKM+89, RL87] could probably be a better solution. In the present study the LMS fitting has been used throughout. Now, the filtered value will be the extrapolation to the considered pixel x_0 of that affine function which has a better fit.

The fact that an affine function has been chosen in fitting implies that the filter/detector is supposed to work the best with images in which brightness is close to piecewise linear.

Let us examine Fig. 22. The information on the fitted affine functions contains the information on the step edge intensity at x_0. Namely, this is the difference in the predicted values S, if a pixel x_0 is located at a step edge. If there is a roof edge at that pixel, the angle R can be considered as the roof edge intensity.

In the case of our edge detector, unlike in the case of other detectors, it is not guaranteed that the edge intensity will be maximum at the edge location. To see how the precise location of the edge can be found we shall examine the examples in the next section.

C.2 Edge location

We shall consider a one-dimensional artificial image of a combined step-roof edge (Fig. 23), in a noiseless and a noisy version with additive Gaussian noise ($\sigma = 10$).

Let us examine the three graphs in Fig. 23 a, b and c, which show the image brightness data and the intensities of the step and roof edge. The scale for each filter is 10. The step intensity has a maximum at the actual step ($x = 29, 30$), but it also has two minima in the neighbourhood. For the roof edge intensity the situation is similar. This makes it impossible to either detect or locate edges merely on the basis of edge intensity data.

Let us consider the mean square fitting errors of lines used to produce the edge intensity measures in Fig. 23 d. The errors for the filter operating on the *past* with respect to the considered data point are denoted E_-, and the errors for the *future* – E_+.

Let us notice that near each edge the following situation occurs: with the increase of x, the error E_- of the forward predictor increases, and the error E_+ of the backward predictor decreases. At some point

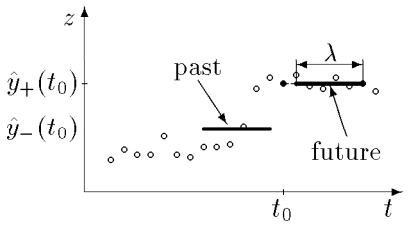


Fig. 21: Averaging competitive filter. λ: scale; $\hat{y}_-(t_0)$, $\hat{y}_+(t_0)$: outputs from predictors working from past and future towards t_0.

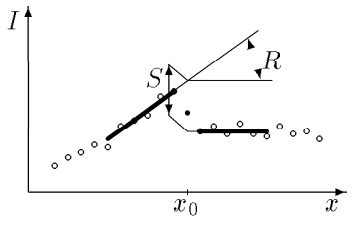


Fig.22: 1D step and roof edge. S, R: step and roof intensities at x_0.

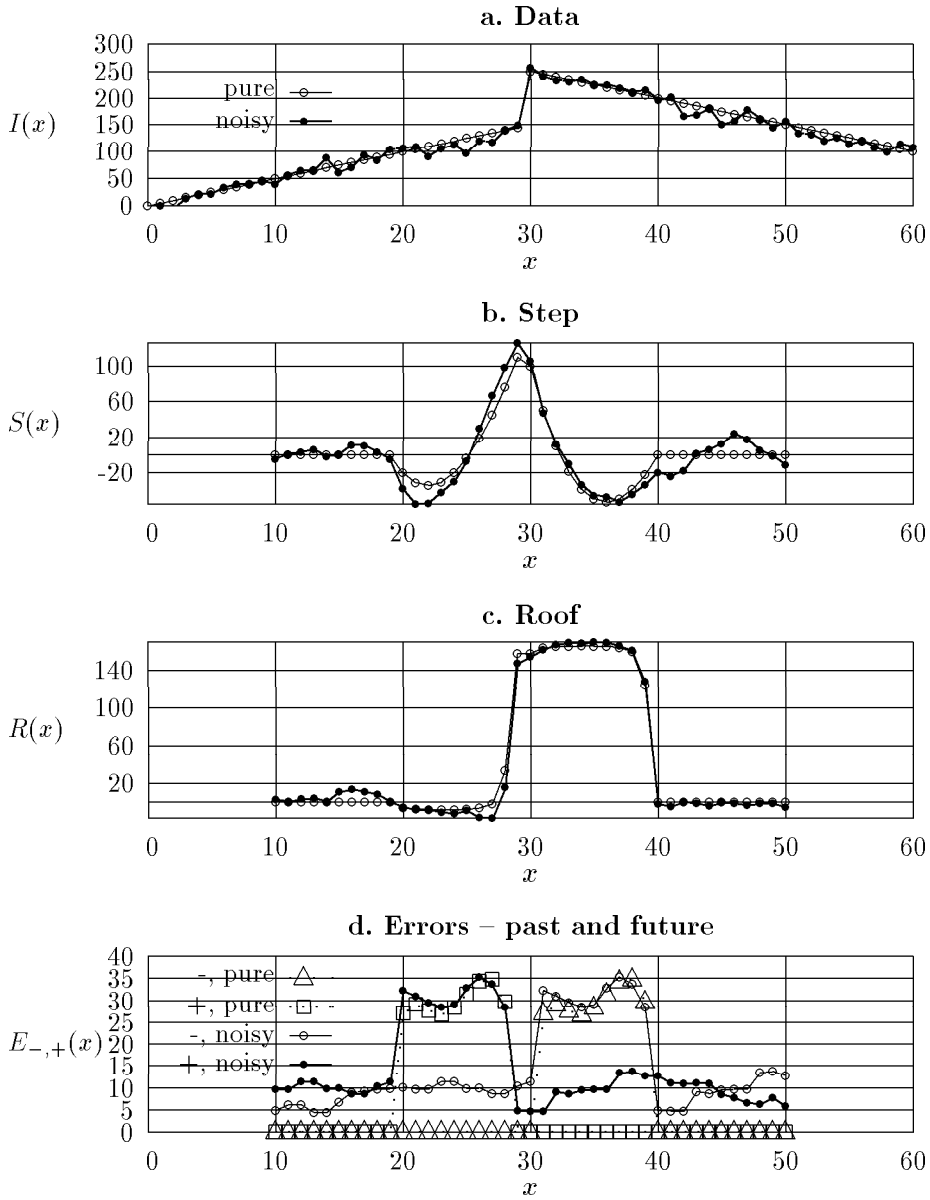


Fig.23: Artificial joint step and roof edge with and without noise.

the graphs of the two errors cross each other.

We shall consider this point as the edge location point x_l.

In the discrete case, the above happens between points x and $x+1$ if

$$\begin{array}{lcll} E_-(x) & < & E_+(x) & \text{and} \\ E_+(x+1) & < & E_-(x+1) & \text{and} \\ E_-(x) & < & E_-(x+1) & \text{and} \\ E_+(x+1) & < & E_+(x)\ . & \end{array} \tag{1}$$

From the two points in which the errors fulfil the condition (1) this one is taken to which the actual crossing point is nearer, that is,

$$\begin{array}{ll} x_l = x & \text{if} \quad E_+(x) - E_-(x) \ < \ E_-(x+1) - E_+(x+1)\ , \\ x_l = x+1 & \text{otherwise}\ . \end{array} \tag{2}$$

This concept is much simpler than that presented in [Chm96b], and in fact it comes back to the method of localizing edges shown in [Chm96a].

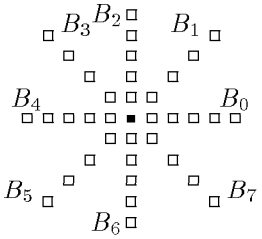


Fig.24: Neighbourhood of a pixel in form of eight branches $B_0 - B_7$ (see text).

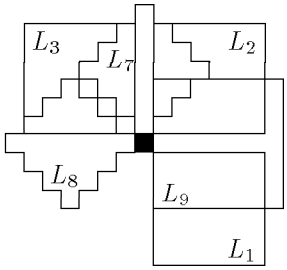


Fig.25: A pixel of a 2D image and examples of its possible square leaves (see text).

C.3 Edge intensity measure

In all the publications except the last report [Chm96c] the measure of step and roof edge intensity was taken directly according to Fig. 22. This means that the parameters of the two predictors – the forward and the backward one – related to the edge location point x_l were taken to calculate the intensities.

Finally [Chm96c], it has been proposed to search the neighbourhood of the point x_l: $x \in \langle x_l - \lambda, x_l + \lambda \rangle$ for the best matching predictors (those with the smallest errors) and to use their parameters in edge intensity calculation.

This last proposition is not very suitable for the 2D implementation, however, as it would require too much search, and the data for a considerable number of *leaves* or *branches* (see Appendix D) would have to be available simultaneously.

D Possible extensions to 2D

The competitive edge detector can be extended to two dimensions in numerous ways. Two of them will be outlined here.

Let us consider the neighbourhood of a pixel in the form of eight sets of pixels lying on eight lines, as shown in Fig. 24. Let us call the pixels lying on each line a branch. Each pair of opposite branches constitutes a 1D filter. As the output of the 2D filter the value coming from that 1D filter can be taken for which the fitting accuracy is the best. As the output edge intensities the maximum intensity found by the 1D filters can be chosen.

In Fig. 25 instead of linear branches a set of possible square neighbouring regions for a pixel are shown. Let us call them leaves. A leaf is a small area which serves as a support over which local estimation of the brightness function is carried out. The results of estimation are used to carry out the prediction (in our case, this is equivalent to extrapolation). This idea is strictly related to the facet model introduced by Haralick and Watson in [HW91].

Two leaves form a filter: not only the opposite leaves, like L_1 and L_3, but also the neighbouring ones, like L_1 and L_2. This is vital for proper behaviour at edge joints. In this version of the detector the same rules as in that with 1D branches are applied.

The advantage of leaves with respect to branches is twofold: better noise resistance can be obtained as more pixels are considered in each filter, and edge orientation can be revealed by taking into account the orientation of the gap between the leaves of the pair which gives the best edge existence measure. For better accuracy, larger leaves inclined by a multiple of less than 45° could be applied. A clear disadvantage is a larger computational complexity of 2D fitting.