

Noise Resistant Gradient Calculation and Edge Detection Using Local Binary Patterns

Michael Teutsch and Jürgen Beyerer

Fraunhofer IOSB
Fraunhoferstr. 1, 76131 Karlsruhe, Germany
{michael.teutsch,juergen.beyerer}@iosb.fraunhofer.de

Abstract. Gradient calculation and edge detection are well-known problems in image processing and the fundament for many approaches for line detection, segmentation, contour extraction, or model fitting. A large variety of algorithms for edge detection already exists but strong image noise is still a challenge. Especially in automatic surveillance and reconnaissance applications with visual-optical, infrared, or SAR imagery, high distance to objects and weak signal-to-noise-ratio are difficult tasks to handle. In this paper, a new approach using Local Binary Patterns (LBPs) is presented, which is a crossover between texture analysis and edge detection. It shows similar results as the Canny edge detector under normal conditions but performs better in presence of noise. This characteristic is evaluated quantitatively with different artificially generated types and levels of noise in synthetic and natural images.

1 Introduction

Gradient calculation and edge detection are topics still worth to discuss as many image processing applications have to deal with input images affected by noise or weak signal-to-noise-ratio (SNR). In automatic surveillance and reconnaissance, difficult environmental conditions, high object distance, moving sensors, and sensor-specific noise lead to images, which are challenging to process. The quality of such image data can vary strongly even for the same sensor over time. Being an important step for applications such as object and image segmentation, line detection, texture analysis, contour extraction, or model fitting, edge detection has to be precise but at the same time robust against noise. The Canny algorithm [1] is a good choice to handle this problem, but it also reaches its limits when the noise level is getting high or alternates. We want to show, that in such cases, our proposed approach can outperform Canny and other tested methods. The original Canny processing chain consists of Gaussian smoothing, gradient calculation, directional non-maximum-suppression for gradient magnitudes and hysteresis thresholding to determine edge pixels. In this work, this chain is taken as a template and its modules are modified using Local Binary Patterns (LBPs). The main innovation is the filtering and gradient calculation strategy. The decrease of edge detection performance with increasing level of noise is slowed down compared to original Canny, while still providing a similar

performance in case of low noise level. Experiments with standard evaluation methods on synthetic and natural image data support this observation.

The presentation and discussion of related work is focused on other approaches towards noise resistant edge detection. Many authors altered the original Canny processing chain modifying mainly the smoothing or gradient calculation strategy or both, while adopting the idea of non-maximum-suppression and subsequent thresholding. Korn [2] combines smoothing and gradient calculation in only one convolution matrix which is an approximation of the first normalized Gaussian derivatives. Different scales are applied and the best one chosen automatically. Evaluation is performed visually on natural image data. Kitanovski et al. [3] use multi-scale undecimated Haar wavelet transform to emphasize edges. Their approach tracks for edges existing at several scales favoring edges at larger scale. Thus, robustness against noise is achieved but detailed edge structure may get lost. Agaian and Almuntashri [4] aim to segment MRI brain images. The typical Canny filter matrices for smoothing and gradient calculation are replaced to better deal with the impulsive noise of MRI images. This seems to work at least as well as normal Canny but no quantitative evaluation is given. Sun and Sun [5] consider two windows around each image pixel. In these windows, gray-value mean and variance are calculated and a difference measure determines the pixel's edge strength. The edge direction is derived from the four different possible window arrangements which describe rotations in steps of 45° . Only a visual evaluation is given. Panetta et al. [6] introduce an adaptive switching function choosing the appropriate smoothing filter (Gaussian, Median, etc.) based on some performance evaluations. Then, a shape-dependent convolution is proposed using kernels of different size and shape (circle, ellipse, hexagon, diamond, etc.) for joint gradient magnitude calculation. Standard Canny and Sobel are outperformed on synthetic images with respect to Abdou and Pratt's figure of merit [7]. For natural images slightly better performance than Canny is visible in presence of noise.

Some authors developed a new edge detection processing chain independent of the Canny algorithm. Chen and Das [8] aim to detect edges and corners in noisy images. A pattern classification algorithm automatically identifies the noise type and chooses the right image restoration technique. After gradient calculation, fuzzy k -means based adaptive thresholding determines the edge or corner pixels. No quantitative evaluation is given but the results on natural data look convincing. Hou and Wei [9] use discrete singular convolution (DSC) to generate different filters for multi-scale edge detection in noisy images. With various levels of Gaussian noise the figure of merit [7] is taken as evaluation on a synthetic image. The performance is very similar to Canny. Chang [10] proposes contextual Hopfield neural networks (CHNNs) for edge detection. For segmentation of MRI images, CHNNs are able to perform better than various other approaches such as Canny or Wavelets in presence of strong salt-and-pepper noise.

In this paper, the focus lies on processing data as it appears in surveillance applications. Strong noise of unclear type, and small objects, which may

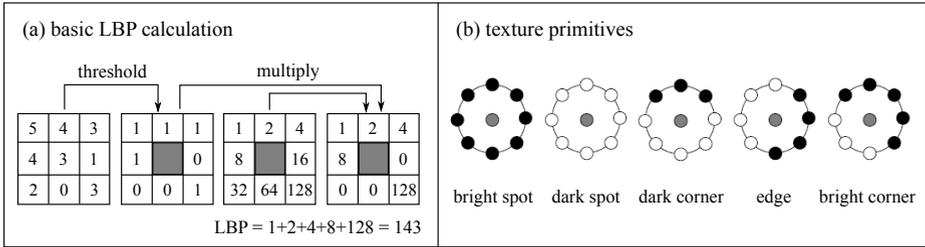


Fig. 1. Calculation and interpretation of Local Binary Patterns (LBPs) [17]

disappear after smoothing characterize the images and, thus, an approach is presented dealing with noise but keeping sophisticated edge structure. The remainder of the paper is organized as follows: the theory of LBPs and the edge detection approach using LBPs are introduced in Section 2. A description of the experimental setup as well as a demonstration of the results and some examples for processing natural image data are given in Section 3. Conclusions are presented in Section 4.

2 The Proposed Approach

The application of LBPs is widely spread in image processing research. Some examples are texture classification [11,12], face detection [13], background modeling [14], structure emphasizing filter [15], or setting up a SIFT descriptor [16]. After a short introduction to the theory of LBP, the modifications and their applicability for edge detection are presented.

2.1 Theory of LBP

LBPs describe a unique encoding for local pixel neighborhood. They are easy to implement, fast to compute, and characterized to be high-performance and robust features in the abovementioned approaches. In the following, we refer to the work of Mäenpää [17] and Ojala et al. [11] for the theory of LBP. In Fig. 1 (a), the typical way of LBP computation is shown. The gray-value of the central pixel is compared to each of the eight neighbors. In case of a higher or equal gray-value, its position will be highlighted with a 1 and, thus, considered for the LBP computation. LBP encoding is calculated by multiplying all highlighted positions with their related weights and summing them up afterwards. The result is a value between 0 and 255 describing a specific neighbor constellation. There are two basic design parameters: number of neighbors P and radius R , since neighbors are ordered circularly around the central position c . This leads to the equation:

$$LBP_{P,R} = \sum_{p=0}^{P-1} s(g_p - g_c)2^p, \text{ where } s(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0. \end{cases} \quad (1)$$

A specialization of LBPs, which will be important for edge detection, is the set of rotation-invariant, uniform LBPs:

$$LBP_{P,R}^{riu2} = \begin{cases} \sum_{p=0}^{P-1} s(g_p - g_c), & \text{if } U(LBP_{P,R}) \leq 2 \\ P + 1, & \text{else.} \end{cases} \quad (2)$$

$LBP_{P,R}^{riu2}$ can be interpreted as texture primitives [17] as seen in Fig. 1 (b). Description *riu2* stands for rotation-invariance and uniformity measure U of 2 or less. U returns the number of bitwise 0/1 and 1/0 transitions in a LBP [11]. With $P = 8$ and $U \leq 2$, only the 58 texture primitives among the 256 LBPs are considered. Rotation-invariance is achieved by assigning all potential rotations of an uniform LBP to the same equivalence class, for example *edge*, *bright corner*, or *dark corner*. As seen in Fig. 2 (top left), there are nine equivalence classes and eight LBPs in each class, each LBP corresponding to a rotation in steps of 45° . An exception is given by the classes *bright spot* and *dark spot* with only one representative each.

LBPs are gray-scale invariant [11] as only the sign of the gray-value difference is considered. However, further information is available in the neighbor's gray-values. To extract this information, the rotation-invariant variance measure VAR is introduced:

$$VAR_{P,R} = \frac{1}{P} \sum_{p=0}^{P-1} (g_p - \mu)^2, \quad \text{where } \mu = \frac{1}{P} \sum_{p=0}^{P-1} g_p. \quad (3)$$

Ojala et al. [11] point out that the combination of *LBP* and *VAR* turned out to be a powerful feature for texture classification.

2.2 Gradient Calculation and Edge Detection with LBPs

The first question is, why LBPs should be suitable for gradient calculation or edge detection? The answer is given in Fig. 2. For some example images, the LBPs are calculated for three different radii $r \in \{1.0, 2.0, 3.0\}$ and accumulated in LBP histograms. Each histogram has ten bins: nine for the different $LBP_{P,R}^{riu2}$ equivalence classes, which are displayed in Fig. 2 (top left), and one for all other LBPs. Since the accumulation values were highly varying along the bins, the vertical axis is visualized in logarithmic scale. In a first experiment, all pixels of a synthetic image are set to value 127. There are no edges in this image. Four different kinds of artificially generated noise are added subsequently, but not mixed, to find out which LBP distribution appears for which kind of noise. It is obvious that for salt-and-pepper noise mostly spot-like LBPs (classes 0, 1, 8) appear, while the accumulation of class 3 is a result of bilinear interpolation during LBP calculation. The other kinds of noise mainly produce LBPs of classes 0, 1, and 9. If pixel positions of such LBPs are not considered for gradient calculation, noise can be suppressed. For the second experiment, a synthetic image with various edges and well-known natural image *Lena* are considered. The LBPs are calculated for each original image as well as for each image with

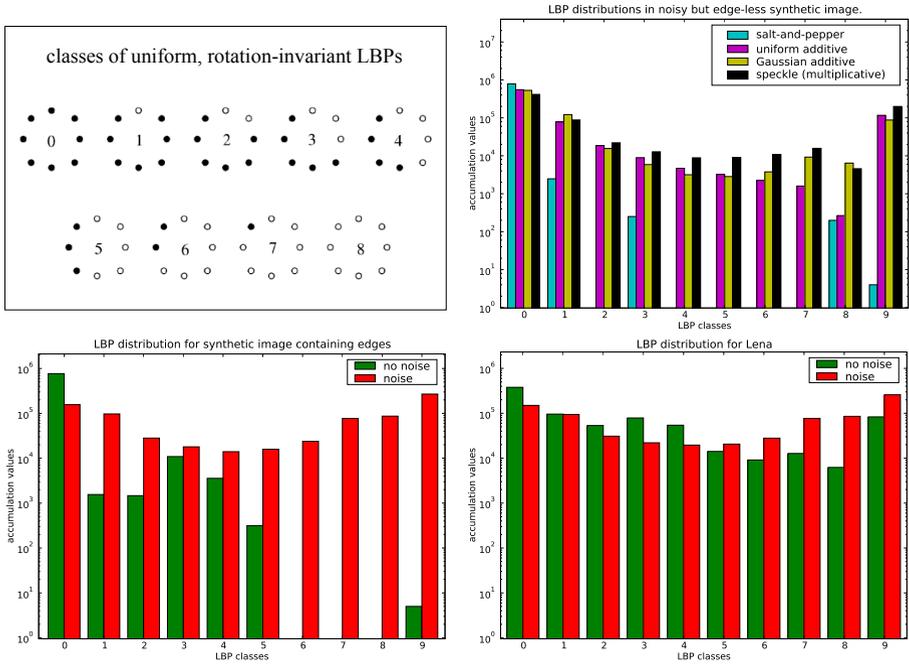


Fig. 2. LBP distributions for an edge-less image with different kinds of noise (top right) and the comparison of LBP distributions for a noise-free and noisy synthetic image with edges (bottom left) and Lena (bottom right)

added combination of noise (Gaussian, speckle, and salt-and-pepper). Class 0 appears most often as it stands for flat, homogeneous image areas. But it also can be seen that classes 3 and 4 mainly represent edges while classes 7 and 8 represent noise.

The basic idea in using LBPs for gradient calculation is to generate a filter rejecting pixel positions of LBPs which are likely to be produced by noise. Thereafter, gradient magnitudes are calculated at the accepted pixel positions using the local variance $VAR_{P,R}$. For easy embedding and testing, the Canny edge detection processing chain is taken as template and modified. The aim of each component is kept, but solved in a different way using LBP. The original Canny algorithm [1] consists of:

1. Noise suppression by smoothing with Gaussian kernel.
2. Calculation of gradient magnitude and direction for each pixel by convolution with a filter matrix.
3. Non-maximum-suppression of the gradient magnitudes in dominant gradient direction.
4. Determination of edge pixels using hysteresis-thresholding.

In the following, the modifications to gain more robustness towards noise are presented step by step.

Noise Suppression: Not only Gaussian noise is the target of noise suppression here, but also salt-and-pepper, uniform additive, and speckle noise. Salt-and-pepper is likely to occur in infrared, while speckle noise is symptomatic for SAR imagery. Especially in applications using such data, another important parameter is used to enhance the robustness of standard LBPs in flat image regions and for edge detection in noisy images: the gray-value threshold T , which was introduced and also used in [14]. Eq. 1 is adapted as follows:

$$LBP_{P,R,T} = \sum_{p=0}^{P-1} s(g_p - g_c - T)2^p. \quad (4)$$

With this modification, a binary decision function f is defined and applied pixelwise to all image pixel positions $c = (x, y)$. f accepts only pixels with related LBPs, which fulfill three criteria:

$$U(LBP_{P,R,T}) = 2 \quad (5)$$

$$LBP_{P,R,T} \neq 2^p, p \in \mathcal{P} = \{0, \dots, P-1\} \quad (6)$$

$$LBP_{P,R,T} \neq 2^P - 1 - 2^p, p \in \mathcal{P} \quad (7)$$

This means, only uniform LBPs are allowed, which are not spots (5) or spot-like (6), (7). The assumption is that all non-uniform LBPs and all uniform LBPs violating one of the three criteria are the result of noise. Thus, they should be suppressed before gradients are calculated. This leads to the following formulation of f for each pixel position c :

$$f(c) = \begin{cases} 1, & \text{if (5) and (6) and (7)} \\ 0, & \text{else.} \end{cases} \quad (8)$$

Only pixel positions with $f(c) = 1$ will be considered for the next step.

Gradient Magnitude and Direction: Convolution with Sobel, Prewitt or other filters is an approximation of partial derivatives. Here, the gradients are calculated using $VAR_{P,R,T}$ and $LBP_{P,R,T}$. The gradient magnitudes $G(c)$ for each pixel position c are computed using the equation:

$$G(c) = \begin{cases} \sqrt{VAR_{P,R,T}}, & \text{if } f(c) = 1 \\ 0, & \text{else.} \end{cases} \quad (9)$$

Variance tends to focus too much on bright objects. So, standard-deviation is used instead of variance as it produces more homogeneous edge images. The robustness against noise can be increased significantly using multi-resolution LBPs [11]. In the literature, they are also known as multi-scale LBPs. For the same pixel position c , several LBPs are calculated varying the parameters P and R . In this work, only variations of radius R are considered and P is fixed to

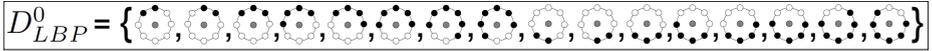


Fig. 3. Set D_{LBP}^0 with LBPs of orientation 0°

$P = 8$. For each LBP accepted by f , $VAR_{P,R,T}$ is calculated and summed up for the gradient magnitude:

$$G(c) = \begin{cases} \sum_{r=R_1}^{R_n} \sqrt{VAR_{P,r,T}}, & \text{if } f(c) = 1 \\ 0, & \text{else.} \end{cases} \quad (10)$$

For each texture primitive in Fig. 1 (b), which is not a spot, eight different rotations are possible. Hence, gradient directions are already available by storing the *not* rotation-invariant $LBP_{P,R,T}^{u2}$ at every pixel position c with $f(c) = 1$.

Non-Maximum-Suppression: In Canny algorithm, the atan2-function is used to calculate gradient directions. These directions are rounded off to only four discretization steps: 0° , 45° , 90° , and 135° . For each pixel's gradient magnitude, a directed non-maximum-suppression is applied in its gradient direction. The remaining maxima describe a skeleton of gradient magnitudes, from which the edge pixels can be determined by hysteresis-thresholding. In this work, the non-maximum-suppression is performed in exactly the same way, but the direction discretization is already given. All uniform LBPs accepted by f are assigned to one of the four direction sets D_{LBP}^0 , D_{LBP}^{45} , D_{LBP}^{90} , or D_{LBP}^{135} . LBPs with an even number of 0s and 1s are ambiguous and, thus, assigned to two sets. The set D_{LBP}^0 is shown in Fig. 3. To find out about its direction, the currently considered LBP just has to be re-found in one of the sets.

Determination of Edge Pixels: The last step is to generate a binary edge pixel image B . Therefore, hysteresis-thresholding is used. Two thresholds t_1 and t_2 are determined with $t_1 < t_2$. If a gradient magnitude $G(c)$ exceeds t_2 , it is accepted as edge pixel. Then, all pixels are considered, which are connected to this edge pixel, and also accepted if their gradient value is greater than t_1 . This approach was adopted with a minor change: a minimum threshold for edge length was introduced. B is the final result of the edge detection approach.

3 Experiments and Evaluation

In this section, the performance of gradient calculation and edge detection with LBPs is evaluated. Before presenting the results, first the experimental setup and the evaluation approach are described and discussed. Finally, the processing of some natural example images coming from standard datasets as well as special surveillance applications is demonstrated.



Fig. 4. Evaluation images with reference edge pixel images

3.1 Experimental Setup

Four parts are needed for the experimental setup: a test image database, a synthetic noise generator with measured signal-to-noise-ratio (SNR), a figure of merit to evaluate the performance, and other edge detection algorithms as competitors. To guarantee repeatability of these experiments, standard approaches have been chosen for each part.

Two test images are used: a synthetic one and Lena. There is a ground-truth edge pixel image for the synthetic image and a sensed-truth for Lena. Sensed-truth means that a humanly sensed good result of Canny edge detection was manually corrected and amended by straightening the typical sinuous lines produced by Canny in case of blurred edges. This is the only way to lay a foundation for a quantitative evaluation on a natural image, which is publicly accessible. The four images have a resolution of 512×512 and are shown in Fig. 4.

A random noise generator was implemented to manipulate the original image with salt-and-pepper, uniform additive, Gaussian additive, or speckle (multiplicative) noise. The level of noise was varied and measured by applying two different methods: peak-signal-to-noise-ratio (PSNR) and structural similarity (SSIM). PSNR is widely used to calculate the difference between original and noisy image in decibel (dB) for evaluating noise resistant edge detectors [9,3]. Structural similarity [18] is a rather new approach much more related to the human noise sensing. It is a value between 1 (no noise) and 0 (strong noise) and not depending on the image peak which is a disadvantage of PSNR. Thus, it was decided to present the results using SSIM, although both methods have been applied. However, the results to be shown later were clearly noticeable with both PSNR and SSIM.

As figure of merit F , the proposal of Abdou and Pratt [7] was chosen:

$$F = \frac{1}{\max(I_I, I_A)} \sum_{i=1}^{I_A} \frac{1}{1 + \alpha d_i^2}, \quad (11)$$

where I_I and I_A is the number of edge pixels in the ideal and the actually detected edge image, d_i denotes the distance between a detected edge pixel and the nearest edge pixel in the ideal image, and α the penalty constant set to $\frac{1}{9}$ as

proposed in [7]. The result is a value between 1 for perfect detection and 0 for poor detection.

The proposed LBP edge detector was compared to three other approaches: two variations of the original Canny algorithm as it is implemented in OpenCV [19] and the edge detector proposed by Korn [2]. Canny in OpenCV doesn't include any smoothing, so two different smoothing strategies are tested with Gaussian and Median filter. Gaussian filter is well-known to be powerful against additive noise, while Median filter is useful against salt-and-pepper and speckle. The Korn edge detector is similar to Canny but using a normalized filter matrix for smoothing and gradient calculation with automatically determined size which is slightly adaptive to varying noise levels.

3.2 Results

Before generating the results, an automatic parameter optimization was applied for each tested algorithm using images with different kinds and levels of noise. It is possible to adjust the parameters for each noise level, of course, but this was not considered since it is not suitable for a real surveillance application.

The OpenCV Canny algorithm has four parameters: size of the smoothing filter, size of the Sobel filter, and the two hysteresis thresholds. For Korn edge detection, filter size factor σ , gradient threshold t , and the two hysteresis parameters have to be set. The LBP edge detector has four parameters as well: LBP radius R , gray-value threshold T , and the two hysteresis thresholds. For all studies in this paper, the number of LBP neighbors was fixed to $P = 8$. Optimization was performed by maximizing the mean figure of merit across the original synthetic and Lena image as well as all different noise level images.

With the best set of parameters, the evaluation was run and its results are shown in Fig. 5. The noise-related plots of F against SSIM show the decreasing performance of edge detection with increasing noise level. *G-Canny* denotes Canny with Gaussian, and *M-Canny* with Median smoothing. M-Canny has its advantages only for salt-and-pepper noise. In all other cases, the LBP edge detector is performing similar to the three other approaches for weak noise but better than them for stronger noise starting at a SSIM value of about 0.9. The convergence of Canny and Korn to $F = 0.4$ when processing the Lena image is a weakness of the figure of merit F . Canny and Korn tend to produce false positive edge pixels in case of strong noise, while the LBP edge detector generates false negatives. Since the reference edge pixel image for Lena contains many edge pixels in general, many false positives cause a better result of F than many false negatives in this situation. Without artificial noise, the figure of merit was nearly the same for all four approaches with about 0.95 for the synthetic image and 0.9 for Lena.

It should be mentioned, that no smoothing was necessary for LBP edge detection. This is an advantage as edge detection normally deals with the problem that smoothing with big filter size can suppress desired sophisticated edge structure but small filter size might not be sufficient enough to suppress strong noise.

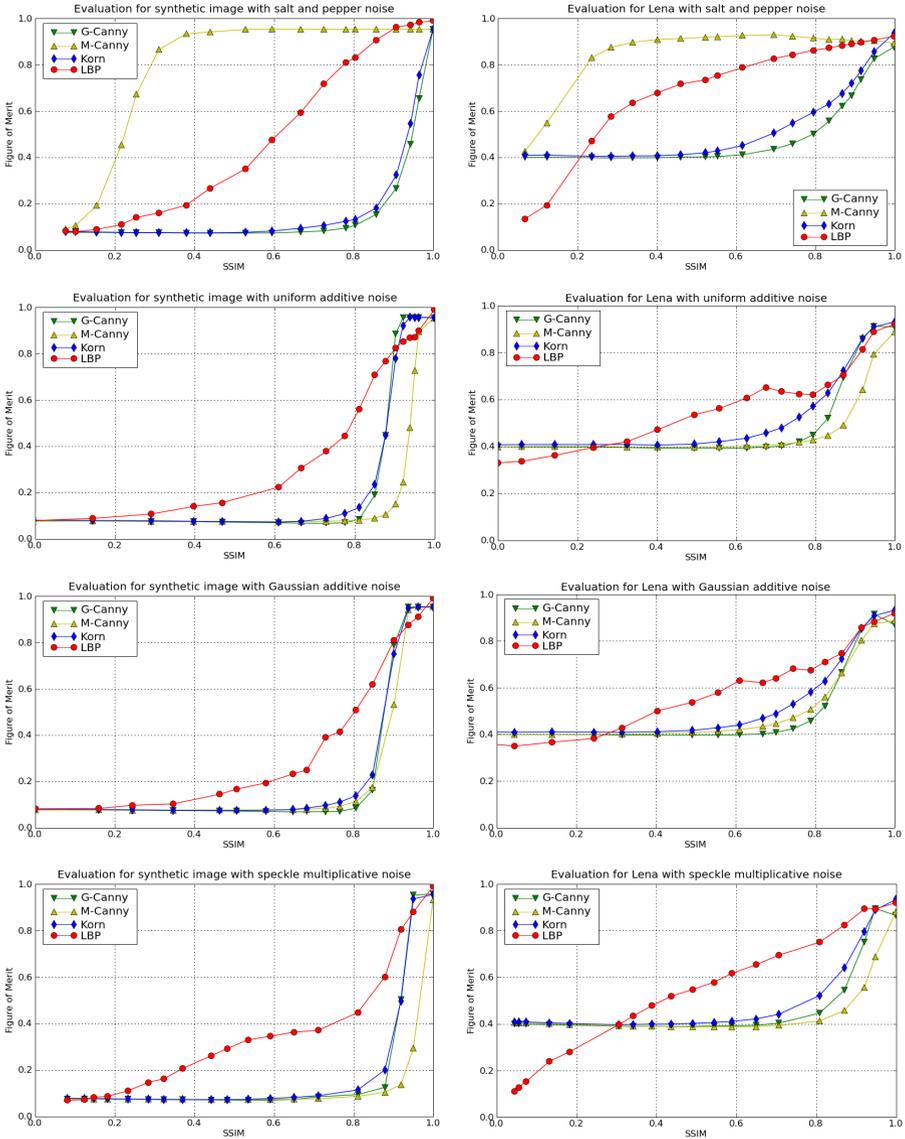


Fig. 5. Evaluation results: figure of merit F plotted against SSIM for synthetic image and Lena with different types and levels of noise

3.3 Examples

For a visual impression of the results, some standard images such as the cameraman or the golf cart have been chosen along with some images from real surveillance applications. They are processed using the LBP approach, OpenCV

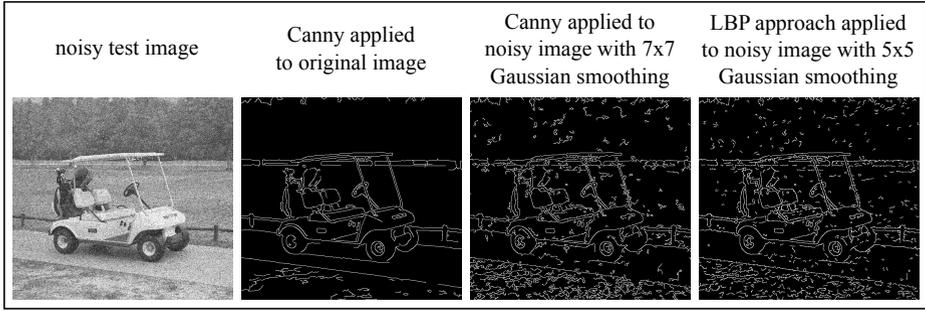


Fig. 6. Example for optional Gaussian smoothing in combination with LBP edge detection

Canny with Gaussian smoothing, and the Korn approach. The parameters were adopted from the automatic parameter optimization applied in Section 3.2 with one minor change: since high smoothing parameters were chosen for Canny and Korn, only Canny parameter with a 7×7 filter matrix was directly adopted and σ for Korn filter matrix was lowered to $\sigma = 1.5$ for a better comparison of strong smoothing effects. The examples are visualized in Fig. 7. In the most left column, the original images are located followed by the edge detection results of LBP, Canny, and Korn. In the first row, the original cameraman image is processed. All edge detection results are similar, but due to strong smoothing, some details are lost in the Canny image such as the tower in the background. A combination of Gaussian, speckle, and salt-and-pepper noise is added with a SSIM of 0.75 to the original image and visualized in the second row. The tendency to produce false negative edge pixels rather than false positives is clearly visible in the LBP edge image especially in comparison to the Korn result, which is strongly affected by noise. The Canny edge image is better due to strong Gaussian smoothing. A potential drawback of the LBP approach is shown in the third row. The noisy golf cart has a SSIM of 0.68 to the original image. Nearly all of the edge pixels found by the LBP approach are correct but due to the high false negatives rate, the Canny result visually looks better although it is also affected by noise. If a subsequent algorithm such as line detection with Hough transform is able to handle false negatives better than false positives, this can be an advantage for the LBP approach.

However, it is possible to support the LBP approach with smoothing, too. This is an application dependent alternative method to the proposed one. An example is visualized in Fig. 6. The same noisy golf cart as in Fig. 7 is shown on the left position. The edge detection result of Canny algorithm applied to the original image (without additional noise) and to the noisy image is displayed in the second and third column. Finally, the right image is the result of the LBP approach with previous 5×5 Gaussian smoothing. Visually, the results of Canny and LBP look similar. But when calculating Abdou and Pratt's figure of merit with the Canny result of the original image as reference, the LBP approach

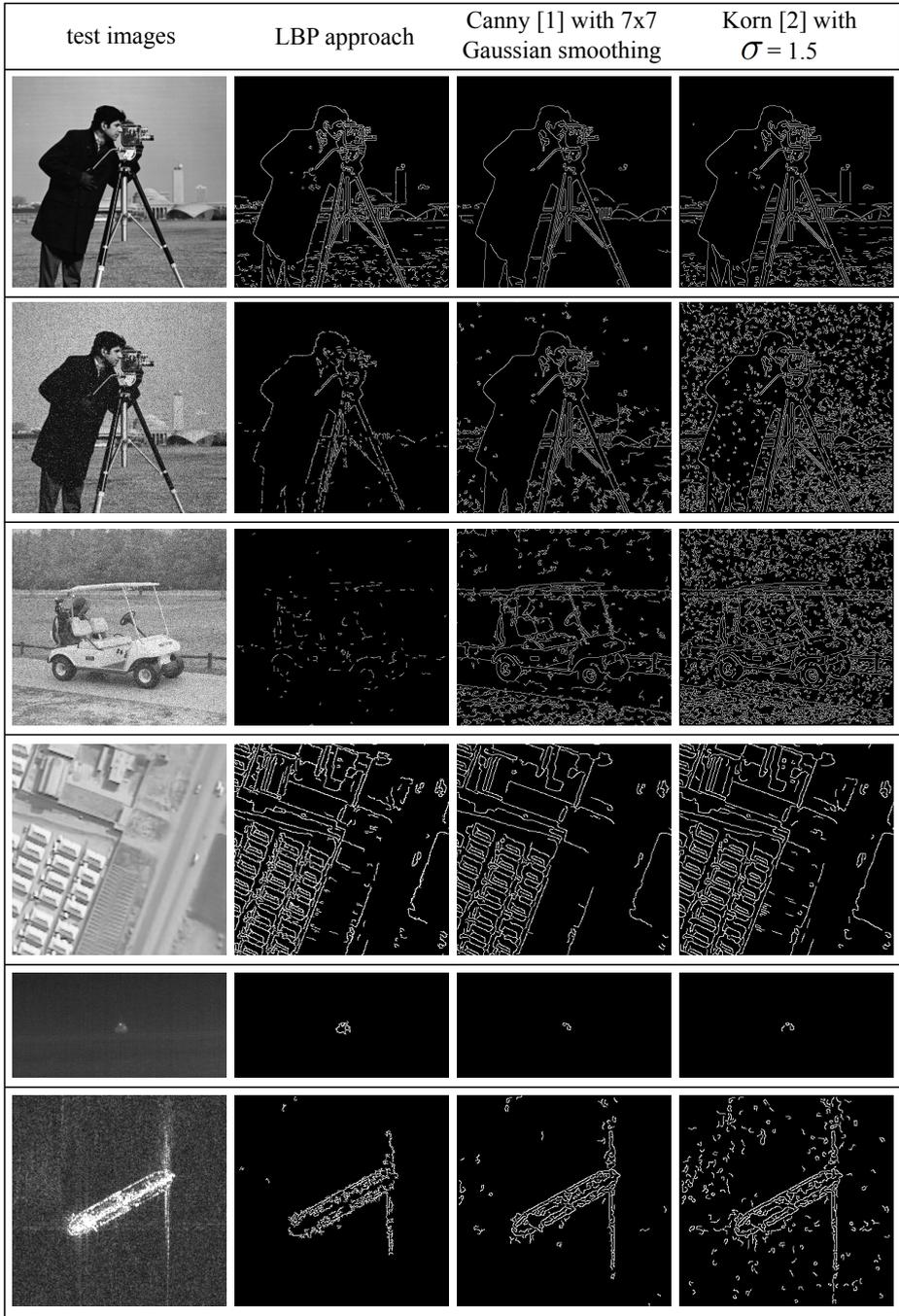


Fig. 7. Examples for real data processing: original images and edge detection using LBPs, Canny, and Korn (from left to right column)

performs consistently better with a figure of merit difference between 0.05 and 0.1 along various test images. This result is achieved with a smaller smoothing filter matrix than for Canny and with a Canny result (for original image) as reference for comparison.

The last three examples in Fig. 7 originate from real surveillance applications and were our motivation to develop the proposed approach. The upper one is an image coming from a visual-optical camera mounted on an unmanned aerial vehicle (UAV). In the mid row a ship in a thermal infrared image originating from a buoy camera is shown. Finally, in the lower row a Synthetic Aperture Radar (SAR) image coming from TerraSAR-X satellite with an observed oil tanker is displayed. The vertical smearing effect is typical SAR noise besides the strong speckle. These images are the result of different sensors with different view angles, have different content, different gray-value distribution, as well as different types and levels of noise. LBP edge detection provides the best edge completeness and noise suppression, while both Canny and Korn produce more false negatives in the infrared image and more false positives in the SAR image. This effect was observed in many more example images from different surveillance sensors. Of course, it is possible to find different sets of parameters for Canny and Korn to produce good edge detection results for each of the original images in Fig. 7 separately, but with these examples the robustness of LBP edge detection is demonstrated.

4 Conclusion

A novel approach for gradient calculation and edge detection is presented using Local Binary Patterns (LBPs). The main innovation is the way of noise suppression with a binary decision function f , which only accepts a special subset of LBPs, namely texture primitives, assuming, that all other LBPs are affected by noise. Gradient magnitude is calculated at pixel positions accepted by f with the gray-value variance of LBP neighbors and robustness is achieved by using multi-scale LBPs. For a good testing environment, the LBP approach is embedded to the Canny processing chain consisting of smoothing, gradient calculation, directional non-maximum-suppression, and hysteresis-thresholding. The structure of the processing chain is adopted but implemented using Local Binary Patterns (LBP) instead of Gaussian smoothing and Sobel filter. The evaluation is performed using one synthetic and one natural image with added artificial noise of different types and levels. With increasing noise level, LBP edge detection outperforms the approaches of Canny and Korn concerning Abdou and Pratt's figure of merit. This effect is visually demonstrated with different natural image examples.

In general, inaccuracies occurring at an early stage of a processing chain can significantly affect the performance of all subsequent modules as well as the overall performance. The assumption is, that with higher robustness against noise, not only edge detection but also other algorithms, which rely on gradients instead of edges, can be improved.

References

1. Canny, J.: Computational Approach to Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 8, 679–698 (1986)
2. Korn, A.: Toward a Symbolic Representation of Intensity Changes in Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 10, 610–625 (1988)
3. Kitanovski, V., Taskovski, D., Panovski, L.: Multi-scale Edge Detection Using Undecimated Wavelet Transform. In: *Proceedings of the IEEE International Symposium on Signal Processing and Information Technology, ISSPIT* (2008)
4. Agaian, S., Almuntashri, A.: Noise-Resilient Edge Detection Algorithm for Brain MRI Images. In: *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBC* (2009)
5. Sun, X., Sun, G.: A New Noise-resistant Algorithm for Edge Detection. In: *Proceedings of the Second International Workshop on Education Technology and Computer Science, ETCS* (2010)
6. Panetta, K.A., Agaian, S.S., Nercessian, S.C., Almunstashri, A.A.: Shape-dependent canny edge detector. *Optical Engineering* 50 (2011)
7. Abdou, I.E., Pratt, W.K.: Quantitative design and evaluation of enhancement/thresholding edge detectors. *Proceedings of the IEEE* 67, 753–763 (1979)
8. Chen, Y., Das, M.: Robust edge and corner detection using noise identification and adaptive thresholding techniques. In: *Proceedings of the IEEE International Conference on Electro/Information Technology* (2007)
9. Hou, Z.J., Wei, G.W.: A new approach to edge detection. *Pattern Recognition* 35, 1559–1570 (2002)
10. Chang, C.Y.: Contextual-based Hopfield neural network for medical image edge detection. *Optical Engineering* 45 (2006)
11. Ojala, T., Pietikäinen, M., Mäenpää, T.: Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24, 971–987 (2002)
12. Guo, Z., Zhang, L., Zhang, D.: A Completed Modeling of Local Binary Pattern Operator for Texture Classification. *IEEE Transactions on Image Processing* 19, 1657–1663 (2010)
13. An, K.H., Park, S.H., Chung, Y.S., Moon, K.Y., Chung, M.J.: Learning discriminative multi-scale and multi-position LBP features for face detection based on Ada-LDA. In: *Proceedings of the IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 1117–1122 (2009)
14. Heikkilä, M., Pietikäinen, M.: A Texture-Based Method for Modeling the Background and Detecting Moving Objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28, 657–662 (2006)
15. Teutsch, M., Saur, G.: Segmentation and Classification of Man-Made Maritime Objects in TerraSAR-X Images. In: *Proceedings of the IEEE International Geoscience and Remote Sensing Symposium, IGARSS* (2011)
16. Heikkilä, M., Pietikäinen, M., Schmid, C.: Description of Interest Regions with Local Binary Patterns. *Pattern Recognition* 42, 425–436 (2009)
17. Mäenpää, T.: *The Local Binary Pattern Approach to Texture Analysis - Extensions and Applications*. Dissertation, University of Oulu, Finland (2003)
18. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing* 13, 600–612 (2004)
19. Bradski, G.: *The OpenCV Library*. Dr. Dobb's Journal of Software Tools (2000)