ADAPTIVE REAL-TIME IMAGE SMOOTHING USING LOCAL BINARY PATTERNS AND GAUSSIAN FILTERS

Michael Teutsch, Patrick Trantelle, and Jürgen Beyerer

Fraunhofer Institute of Optronics, System Technologies and Image Exploitation (IOSB)
Fraunhoferstr. 1, 76131 Karlsruhe, Germany
{michael.teutsch, patrick.trantelle, juergen.beyerer}@iosb.fraunhofer.de

ABSTRACT

Image smoothing is widely used for enhancing the quality of single images or videos. There is a large amount of application areas such as machine vision, entertainment industry with smart TVs or consumer cameras, or surveillance and reconnaissance with different imaging sensors. In many cases it is not easy to find the trade-off between high smoothing quality and fast processing time. However, this is necessary for the mentioned applications as they are dependent on real-time computing. In this paper, we aim to find a good trade-off. Local texture is analyzed with Local Binary Patterns (LBPs) which are used to adapt the size of a Gaussian smoothing kernel for each pixel. Real-time requirements are met by the implementation on a Graphical Processing Unit (GPU). An image of 512×512 pixels is processed in 2.6 ms.

Index Terms— Image denoising, image enhancement, locally adaptive, variable kernel size, texture analysis, LBPs

1. INTRODUCTION

Image smoothing is a technique used to enhance the quality of images and videos. This is done for different reasons: in machine vision, medical imaging, or automatic surveillance and reconnaissance the performance of processing algorithms is to be improved by higher quality input data. On the other hand, consumers are to be supported with a better visual impression of camera images or videos in entertainment systems. Some of these applications require fast computation to guarantee real-time processing. Thus, one key challenge is to find a good trade-off between smoothing quality and processing time. In this paper, we meet this challenge by using Local Binary Patterns (LBP) and Gaussian filters. Flat image areas are smoothed with a large Gaussian kernel and edges or corners with small kernels. LBPs are used to decide about the kernel size depending on how complex the local texture is. We do not discuss temporal filters [1] which can be applied to videos but have to solve specific problems such as accurate image-to-image registration or handling moving objects.

There is a lot of related work on this topic. Most authors try to remove strong noise with complex methods (image restoration) and only few of them try to find the mentioned trade-off. Bilcu and Vehvilainen [2] propose a modified sigma filter. Images are decomposed by horizontal and vertical low- and high-pass filtering. The sigma filter is applied to each component and image reconstruction is done by linear combination. Chaudhry and Mirza [3] analyzed the effect of different window sizes in Adaptive Fuzzy Punctual Kriging (AFPK) based image restoration. Hammond and Simoncelli [4] developed an orientation-adaptive Gaussian Scale Mixture Model (GSM) in the wavelet domain representing local amplitude and local orientation. Portilla et al. [5] use a GSM only considering the local amplitude. Kervrann et al. [6] present a Bayesian Non-Local (NL) means filter with spatially adaptive dictionaries for better contrast restoration. Wang [7] suggests a modified Lee filter. Haar wavelets and window-based pyramids are used for multiscale image denoising. Jin et al. [8] propose an adaptive Wiener filter using an adaptive weighted averaging (AWA) approach in the spatial and the wavelet domain. Gnanadurai and Sadasivam [9] also present an approach in the wavelet domain based on Generalized Gaussian Distribution (GGD) modeling of subband coefficients which are calculated after the application of Discrete Wavelet Transform (DWT) and used for adaptive thresholding for each subband reconstructing the image with inverse DWT. Some more advanced approaches for texture adaptive smoothing are based on PDEs [10, 11] providing good results for the price of high computation time.

The remainder of this paper is organized as follows: The proposed approach is described in Section 2. In Section 3, the experimental setup and the results are presented. Finally, conclusions and potential future work are given in Section 4.

2. THE PROPOSED APPROACH

Local texture around a central pixel is analyzed with LBPs and the result is used to adjust the size of a Gaussian smoothing kernel. If there is no texture or strong noise, a large kernel is applied. In case of a complex edge and corner structure, a small kernel is chosen. After a brief introduction to the theory of LBPs, the realization of this idea is described.

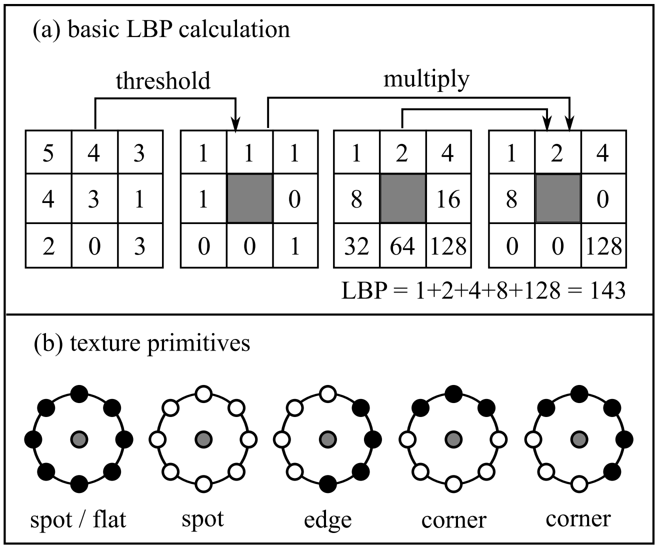

Fig. 1. Theory and interpretation of LBPs [12].

2.1. Theory of LBPs

In recent years, LBPs have become popular for texture analysis. Face detection, background modeling, or texture classification are some of the applications [12]. LBPs are a unique encoding of a pixel neighborhood. Fig. 1a describes the calculation. There are two main parameters: number of neighbors P and radius R. This leads to the following formalization for the gray-values g of central pixel c and neighboring pixels p:

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

In this paper, we are using a special kind of LBPs called *rotation-invariant, uniform* LBPs [12]:

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

U describes the number of bitwise 0/1 and 1/0 transitions. Only two or less are allowed for uniform LBPs. As shown in Fig. 1b, $LBP_{P,R}^{riu2}$ can be interpreted as *texture primitives* [12]. There are nine classes of texture primitives.

2.2. Adaptive Gaussian Smoothing with LBPs

For some example images (Lena, etc.) with different levels and types of artificial noise we calculate $LBP_{8,R}^{riu2}$. For higher robustness, multiscale LBPs [12] with variable radius R for each pixel are used. Fig. 2 shows that the number of corners and edges ($LBP_{8,R}^{riu2}$ classes 3-5) decreases when the noise level increases. As noise is unstructured, other LBPs (mainly classes 0, 8, 9) appear. Class 9 represents all LBPs which are not $LBP_{8,R}^{riu2}$.

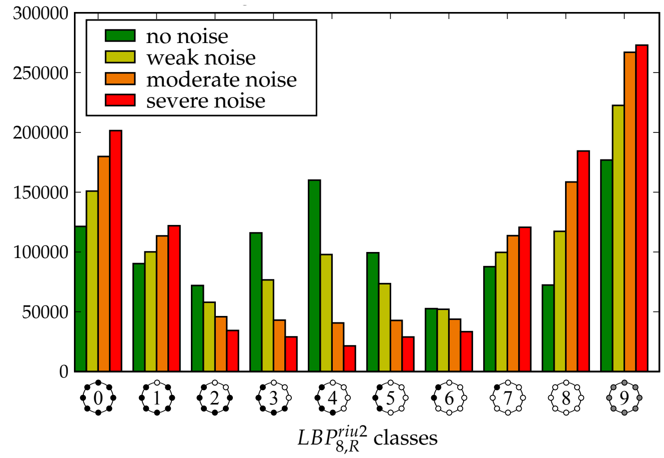

Fig. 2. LBP histogram for Lena with different noise levels.

This approach is used to define a decision criterion for the kernel size of a Gaussian filter. Gaussian filters are effective and widely used for image smoothing but suffer from the effects of a fixed kernel size: while noise reduction may not be strong enough if the kernel is too small, complex edge structures may get blurred if the kernel is too large. Thus, we propose a variable kernel size for each pixel based on the ratio t of edges and corners to the total number of calculated LBPs in its local surrounding area. Let $\mathcal{R}_c$ with the size of $N \times N$ be the local region surrounding a central pixel $c = (x_c, y_c)$ and $\mathcal{C}$ and $\mathcal{E}$ the sets of $LBP_{8,R}^{riu2}$ representing corners (class 3 and 5) and edges (class 4). Ratio t_c is then defined as

$$t_c = \frac{|\{\mathrm{LBP}_{8,r}(i,j) \mid \mathrm{LBP}_{8,r}(i,j) \in \mathcal{C} \cup \mathcal{E}\}|}{|\{\mathrm{LBP}_{8,r}(i,j)\}|} \tag{3}$$

with $(i,j) \in \mathcal{R}_c$ and $r \in \{R_0, \dots, R_k\}$. The smoothed image I_s of a given image I using Gaussian kernels G of size $m \times m$ is then calculated by

$$I_s(c) = \begin{cases} I(c), & \text{if } t_c > t_0 \\ (G_{3\times3} * I)(c), & \text{if } t_0 \geq t_c > t_1 \\ (G_{5\times5} * I)(c), & \text{if } t_1 \geq t_c > t_2 \\ (G_{7\times7} * I)(c), & \text{if } t_2 \geq t_c > t_3 \\ (G_{9\times9} * I)(c), & \text{else.} \end{cases} \tag{4}$$

Ratio t_c is more powerful than analyzing local contrast since it considers the structural character of the local texture. The threshold parameters t_0, t_1, t_2, t_3 have been determined by an automatic parameter optimization using several images with different levels and types of artificial noise maximizing the mean peak-signal-to-noise ratio (PSNR) of the original to the noisy image.

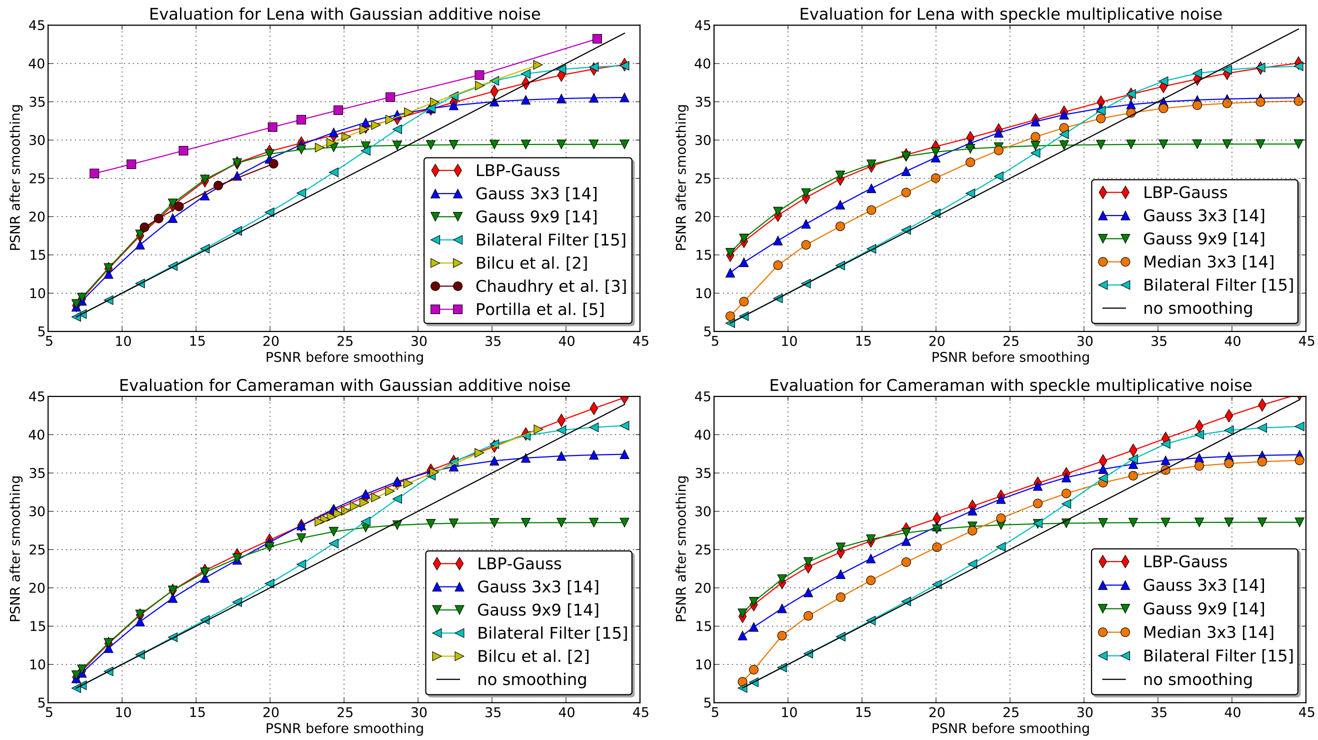

Fig. 3. Evaluation results: PSNR before and after filtering using different approaches.

2.3. GPU Implementation

The proposed approach is well-suited for implementation on a GPU. The Gaussian filter is a separable convolution that runs in 0.2 ms for Lena image using a slightly modified code sample from NVIDIA. LBP performance is mostly dependent on bilinear interpolation (due to the neighbors' circular arrangement) and memory bandwidth. On GPUs bilinear interpolation is implemented in the texture unit [13] and the texture cache is optimized for 2D spatial locality and fits well with the access pattern used by LBPs. With our implementation using CUDA 4.0 and a NVIDIA GeForce GTX 580 GPU we achieved runtimes of about 2.6 ms per image and a one-time initialization time of about 25 ms at application startup. The runtime per image increases linearly with the image size and the initialization time is insignificant for video processing.

3. EXPERIMENTS AND RESULTS

The experimental setup for the quantitative evaluation consists of a set of synthetic and natural test images, an artificial noise generator, and the calculation of PSNR before and after smoothing to measure the smoothing performance. In this paper we present an excerpt of the evaluation with Lena and Cameraman as test images, 19 different levels of Gaussian additive and speckle multiplicative noise for systematic image degradation.

As competitors to the proposed LBP-Gauss approach we use standard Gauss and median filter with fixed kernel size from the OpenCV library [14]. We also evaluate bilateral filtering [14] which is edge-preserving by using means of a nonlinear combination of local pixel values [15]. From the related work we consider the results of Bilcu and Vehvilainen [2] using a modified sigma filter, Chaudhry et al. [3] using Adaptive Fuzzy Punctual Kriging (AFPK), and Portilla et al. [5] using an orientation-adaptive Gaussian Scale Mixture Model (GSM). Most complex methods [4, 6, 8] show similar results as Portilla et al. [5], while [7, 9] perform similar to the proposed approach. However, Bilcu and Portilla provide results for a larger PSNR range of Gaussian additive noise. Speckle noise is not considered in their evaluation at all.

The results are presented in Fig. 3. Complex methods such as [4, 5] work best but have runtimes of 10 or more seconds per image. For Lena with Gaussian additive noise the bilateral [15] and the modified sigma filter [2] slightly perform better than LBP-Gauss for weak noise up to 30 dB PSNR, but worse for moderate and strong noise. LBP-Gauss performs comparably or better in all noise levels for the Cameraman with Gaussian additive noise. For speckle noise the LBP-Gauss provides the most balanced results with the highest mean PSNR along all noise levels. Overall, the LBP-Gauss outperforms Gauss and median filters with fixed kernel size. Some example images are shown in Fig. 4 to demonstrate the benefit of spatially adaptive filtering compared to fixed kernel

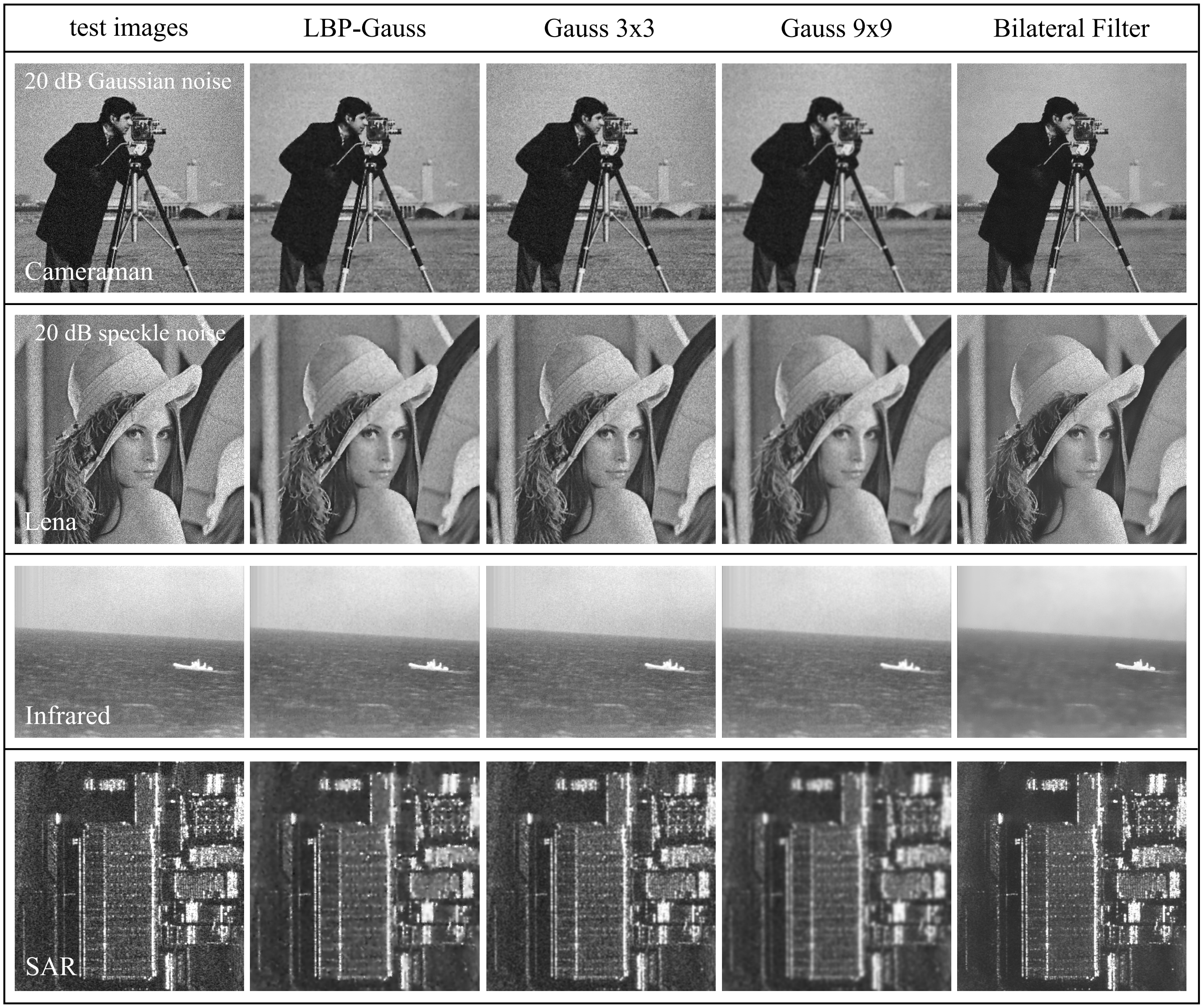

Fig. 4. Examples for image smoothing. Besides Lena and Cameraman we also present one infrared (IR) and one Synthetic Aperture Radar (SAR) image coming from real surveillance data with unknown type and level of noise. While the 3×3 Gauss kernel can not remove the noise well, the 9×9 kernel causes blurring. Due to the strong noise in the example images, parameters for strong denoising have been chosen for the bilateral filter (different than in the evaluation in Fig. 3) causing slower runtimes of about 0.5-1 s. The parameters for LBP-Gauss and bilateral filtering were fixed during the whole test. The LBP-Gauss as proposed in this paper provides the best local adaptivity.

size. The two lower images are coming from real surveillance data with unknown type and level of noise.

4. CONCLUSION

A spatially adaptive image smoothing approach is proposed using LBP statistics and Gaussian filters. Local texture quality and structure are determined for each pixel in the noisy image. Small Gaussian kernels are used to preserve local edge structure, while flat and very noisy image areas are smoothed with large kernels. The results show that the proposed method offers a good trade-off between spatially adaptive image smoothing and real-time processing. Even large images can be processed within few milliseconds. However, the presented approach is outperformed by complex methods with 10 or more seconds of processing time.

Potential future work includes considering the LBP orientation and local contrast for further adjustment of the Gaussian kernel. Local orientation and contrast can be derived directly from the calculated LBPs as well [16].

5. REFERENCES

[1] Thomas Müller and Markus Müller, "CART IV: improving automatic camouflage assessment with assistance methods," in *Proceedings of SPIE Vol. 7662, Infrared Imaging Systems: Design, Analysis, Modeling, and Testing XXI*, Apr. 2010.

[2] Radu Ciprian Bilcu and Markku Vehvilainen, "A Modified Sigma Filter for Noise Reduction in Images," in *Proceedings of the 9th WSEAS International Conference on Communications (ICCOM)*, Vouliagmeni, Athens, Greece, 2005.

[3] Asmatullah Chaudhry and Anwar M. Mirza, "Effect of Neighbourhood Size on Negative Weights in Punctual Kriging based Image Restoration," *International Journal of Applied Mathematics and Computer Sciences*, vol. 4, no. 1, pp. 1–6, 2008.

[4] David K. Hammond and Eero P. Simoncelli, "Image denoising with an orientation-adaptive Gaussian scale mixture model," in *Proceedings of the 13th IEEE International Conference on Image Processing (ICIP)*, Atlanta, GA, USA, 2006.

[5] Javier Portilla, Vasily Strela, Martin J. Wainwright, and Eero P. Simoncelli, "Image Denoising Using Scale Mixtures of Gaussians in the Wavelet Domain," *IEEE Transactions on Image Processing*, vol. 12, no. 11, pp. 1338–1351, Nov. 2003.

[6] Charles Kervrann, Jerome Boulanger, and Pierrick Coupe, "Bayesian Non-local Means Filter, Image Redundancy and Adaptive Dictionaries for Noise Removal," in *Scale Space and Variational Methods in Computer Vision*, vol. 4485 of *Lecture Notes in Computer Science (LNCS)*, pp. 520–532. Springer Berlin Heidelberg, 2007.

[7] Xin Wang, "Lee Filter for Multiscale Image Denoising," in *Proceedings of the 8th International Conference on Signal Processing (ICSP)*, Beijing, China, 2006.

[8] Fu Jin, Paul W. Fieguth, Lowell L. Winger, and Ed Jernigan, "Adaptive Wiener Filtering of Noisy Images and Image Sequences," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, Barcelona, Spain, 2003.

[9] D. Gnanadurai and V. Sadasivam, "An Efficient Adaptive Thresholding Technique for Wavelet Based Image Denoising," *International Journal of Information and Communication Engineering (IJICE)*, vol. 2, no. 2, pp. 1121–1126, 2008.

[10] Joachim Weickert, *Anisotropic Diffusion in Image Processing*, Teubner-Verlag, Stuttgart, 1998.

[11] Pietro Perona and Jitendra Malik, "Scale-Space and Edge Detection Using Anisotropic Diffusion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 7, pp. 629–639, July 1990.

[12] Matti Pietikäinen, Abdenour Hadid, Guoying Zhao, and Timo Ahonen, *Computer Vision Using Local Binary Patterns*, Springer-Verlag London Limited, 2011.

[13] NVIDIA, "NVIDIA CUDA C Programming Guide Version 4.0," 2011, http://docs.nvidia.com/cuda/cuda-c-programming-guide [Last accessed on 2013-01-23].

[14] Gary Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[15] Carlo Tomasi and Roberto Manduchi, "Bilateral Filtering for Gray and Color Images," in *Proceedings of the 6th IEEE International Conference on Computer Vision (ICCV)*, Jan. 1998.

[16] Michael Teutsch and Jürgen Beyerer, "Noise Resistant Gradient Calculation and Edge Detection Using Local Binary Patterns," in *Computer Vision - ACCV 2012 Workshops*, vol. 7728 of *Lecture Notes in Computer Science (LNCS)*, pp. 1–14. Springer Berlin Heidelberg, 2013.