Olli Lahdenoja

Local Binary Patterns in Focal-Plane Processing

Analysis and Applications

Turku Centre *for* Computer Science

TUCS Dissertations
No 194, June 2015

Local Binary Patterns in Focal-Plane Processing

Analysis and Applications

Olli Lahdenoja

To be presented, with the permission of the Faculty of Mathematics and Natural Sciences of the University of Turku, for public criticism in Auditorium PHA1 on June 6, 2015, at 12 noon.

University of Turku
Department of Information Technology &
Technology Research Center (TRC)
20014 Turun Yliopisto

2015

Supervisors

Professor Ari Paasio
Technology Research Center
University of Turku
FIN-20014 University of Turku
Finland

Docent Mika Laiho
Technology Research Center
University of Turku
FIN-20014 University of Turku
Finland

Reviewers

Professor Ákos Zarándy
SZTAKI - Hungarian Academy of Sciences
Institute for Computer Science and Control
13-17 Kende Street
Budapest, H-1111
Hungary

Associate Professor Fernando Corinto
Polytechnico di Torino
Department of Electronics and Telecommunications
Corso Duca degli Abruzzi, 24
10129-Torino
Italy

Opponent

Associate Professor Victor Manuel Brea Sánchez
Department of Electronics and Computer Science
University of Santiago de Compostela
Rúa de Jenaro de la Fuente Domínguez
15782 - Santiago de Compostela
Spain

ISBN 978-952-12-3197-1
ISSN 1239-1883

Abstract

Feature extraction is the part of pattern recognition, where the sensor data is transformed into a more suitable form for the machine to interpret. The purpose of this step is also to reduce the amount of information passed to the next stages of the system, and to preserve the essential information in the view of discriminating the data into different classes. For instance, in the case of image analysis the actual image intensities are vulnerable to various environmental effects, such as lighting changes and the feature extraction can be used as means for detecting features, which are invariant to certain types of illumination changes. Finally, classification tries to make decisions based on the previously transformed data.

The main focus of this thesis is on developing new methods for the embedded feature extraction based on local non-parametric image descriptors. Also, feature analysis is carried out for the selected image features. Low-level Local Binary Pattern (LBP) based features are in a main role in the analysis. In the embedded domain, the pattern recognition system must usually meet strict performance constraints, such as high speed, compact size and low power consumption. The characteristics of the final system can be seen as a trade-off between these metrics, which is largely affected by the decisions made during the implementation phase. The implementation alternatives of the LBP based feature extraction are explored in the embedded domain in the context of focal-plane vision processors.

In particular, the thesis demonstrates the LBP extraction with MIPA4k massively parallel focal-plane processor IC. Also higher level processing is incorporated to this framework, by means of a framework for implementing a single chip face recognition system. Furthermore, a new method for determining optical flow based on LBPs, designed in particular to the embedded domain is presented. Inspired by some of the principles observed through the feature analysis of the Local Binary Patterns, an extension to the well known non-parametric rank transform is proposed, and its performance is evaluated in face recognition experiments with a standard dataset. Finally, an a priori model where the LBPs are seen as combinations of n-tuples is also presented.

Tiivistelmä

Piirreirroituksessa pyritään muuntamaan sensoreista saatu informaatio muotoon, jonka tietokone voi helpommin käsitellä. Tämän hahmontunnistuksen vaiheen tarkoituksena on myös vähentää järjestelmän seuraaville tasoille lähetettävän datan määrää, sekä pyrkiä säilyttämään oleellinen tieto luokittelua silmälläpitäen. Esimerkiksi kuva-analyysin yhteydessä varsinainen intensiteettitieto on tyypillisesti herkkä ympäristön muutoksille, kuten valaistukselle, jolloin piirreirroitusta voidaan käyttää havaitsemaan sellaisia piirteitä, jotka eivät riipu tietyntyyppisistä valaistuksen muutoksista. Lopulta luokittelun avulla pyritään tekemään varsinaisia päätelmiä sensori-informaation perusteella.

Väitöskirjassa kehitetään uusia menetelmiä piirreirroitukseen käyttäen sulautettuja tietokonejärjestelmiä yhdistettynä ei-parametrisiin kuvapiirteisiin. Valituille piirteille suoritetaan myöskin piirreanalyysiä keskittyen paikallisiin binäärikuvio piirteisiin (LBP, Local Binary Pattern). Kun kyseessä on sulautettu järjestelmä, hahmontunnistusta tekevän systeemin täytyy yleensä myös saavuttaa tietyt tehokkuusvaatimukset, kuten nopeus, pieni fyysinen koko ja alhainen tehonkulutus. Valmiin järjestelmän ominaisuudet muodostuvat siten osaltaan siitä, millaisia valintoja järjestelmän toteutusvaiheessa on tehty. Erilaisia toteutusvaihtoehtoja LBP pohjaiseen piirreirroitukseen on työssä tutkittu sulautetun järjestelmän tapauksessa, keskittyen kuvatasossa laskentaa tekeviin prosessoreihin.

Työssä havainnollistetaan mm. LBP piirteiden laskentaa MIPA4k rinnakkaisprosessorilla. Työssä esitetään myös viitekehys yhdellä IC piirillä toimivan kasvontunnistusjärjestelmän toteuttamiseksi. Samoin esitetään uusi LBP pohjainen menetelmä optisen vuon määrittämiseen, joka on soveltuu erityisesti sulautettuun järjestelmään. Piirreanalyysin yhteydessä, pohjautuen LBP piirteiden ominaisuuksiin, esitetään myöskin yleistetty versio hyvin tunnetusta kuvan järjestyslukumuunnoksesta (engl. Rank transform). Sen tehokkuutta mitataan kasvontunnistuksessa yleisesti käytetyllä FERET testiaineistolla. Lopuksi myös esitetään malli paikallisille binäärikuvioille, joissa ne nähdään kuvatasoon sijoittuvien järjestyslukujonojen kombinaatioina.

Acknowledgements

The financial support from TUCS Graduate School, The Academy of Finland (no. 254430), The Finnish Scientific Advisory Board for Defence (MATINE), Nokia Foundation, Tekniikan edistämissäätiö and Ulla Tuomisen säätiö is gratefully acknowledged.

I would like to thank professor Ari Paasio and adjunct professor Mika Laiho for providing the possibility to conduct this work and also important guidance and supervision. They have given me valuable ideas and freedom to carry out my own ideas during this work. Our laboratory, currently known as Technology Research Center (TRC) at University of Turku has been a pleasant place to work.

The important contribution of the other co-authors Esa Alhoniemi, Janne Maunu, Tero Säntti, Jonne Poikonen and Valery Ipatov to the publications in this thesis is acknowledged. The former and current staff in our laboratory also deserves my thanks. I would also like to thank my parents and little brother and friends for support and encouragement during the thesis work.

The reviewers professor Ákos Zarándy and associate professor Fernando Corinto also deserve my thanks for reviewing this thesis.

Psalm 23:3 (New International Version),
He guides me along the right paths for his names sake.

List of original publications

1. O. Lahdenoja, M. Laiho, A. Paasio, *Reducing the feature vector length in Local Binary Pattern based face recognition*, IEEE International Conference on Image Processing, pp. 914-917, 2005.

2. O. Lahdenoja, E. Alhoniemi, M. Laiho, A. Paasio, *A Shape-preserving non-parametric symmetry transform*, 18th International Conference on Pattern Recognition, pp. 373-377, 2006.

3. O. Lahdenoja, *A Statistical approach for characterising Local Binary Patterns*, TUCS Technical Report, No. 795, 2006.

4. O. Lahdenoja, M. Laiho, J. Maunu, A. Paasio, *A Massively parallel face recognition system*, EURASIP Journal on Embedded Systems - Special Issue on Embedded Vision Systems, Article ID 72316, 2007.

5. O. Lahdenoja, M. Laiho, A. Paasio, *On the spatial distribution of local non-parametric facial shape descriptors*, 16th Scandinavian Conference on Image Analysis, pp. 351-358, 2009.

6. O. Lahdenoja, M. Laiho, *Regional image correspondence matching method for SIMD processing*, 19th European Conference on Circuit Theory and Design, pp. 802-805, 2009.

7. O. Lahdenoja, J. Poikonen, M. Laiho, *Extracting Local Binary Patterns with MIPA4k vision processor*, 12th International Workshop on Cellular Nanoscale Networks and Their Applications, 2010.

8. O. Lahdenoja, J. Poikonen, M. Laiho, *Towards Understanding the Formation of Uniform Local Binary Patterns*, ISRN Machine Vision, Article ID 429347, 2013.

9. O. Lahdenoja, T. Säntti, J. Poikonen, M. Laiho, A. Paasio, *Characterizing spatters in laser welding of thick steel using motion flow analysis*, 18th Scandinavian Conference on Image Analysis, pp. 675-686, 2013.

10. O. Lahdenoja, T. Säntti, M. Laiho, J. Poikonen, *Spatter Tracking in Laser- and Manual Arc Welding with Sensor-level Pre-processing*, 22nd International Conference on Computer Graphics, Visualization and Computer Vision (WSCG), 2014.

The originality of this thesis has been checked in accordance with the University of Turku quality assurance system using the Turnitin OriginalityCheck service.

Contents

Glossary of Terms

ACE processor family *a family of mixed-mode CNN processors by the Institute of Microelectronics of Seville*
A child LBP *an LBP derived from an n-tuple by using a threshold (instead of the actual center)*
Adaptive integration time *adjusting the integration time of a camera to compress the high intensity variations of a scene by image dynamic range compression*
A priori model *model of data interpretation (features) which holds with random (i.i.d.) data*
ASIC *Application Specific Integrated Circuit*
b **(binary pattern space)** *all individual LBPs of length M*
Block histogram *LBP histogram defined for a certain image subregion*
Census transform *An image transform which compares local neighborhood windows among different images using a bit-wise XOR() operator*
CMOS *Complementary Metal-Oxide-Semiconductor*
CMU+MIT dataset *The face detection evaluation database from the CMU and MIT universities*
CNN *Cellular Nonlinear Network (an array processor)*
CNN template *a globally controlled neighborhood map applied to each PE of a CNN to control its programming*
Complete uniform (n-tuple) *an n-tuple containing at most one increasing and one decreasing run*
CSU evaluation system *A C++ library for evaluation of face recognition algorithms*
Descriptor k *k is specific to a certain LBP and it is calculated as the minimum between the total number of one and zero bits*
Device mismatch *random variation in IC device properties*
Dynamic range *The number of separate intensity levels in an image, e.g. 256 (8bit)*
EBGM *Elastic Bunch Graph Matching*
Embedded system *computer and software designed to operate under restrictions set by environment*

EYE-RIS platform *a commercial mixed-mode CNN processor by Anafocus*

FERET *The Facial Recognition Technology Database*

Focal-Plane Processor *Integrated circuit (IC) which performs computation on a regular 2D array of PEs (Processing Elements)*

FPGA *Field-Programmable Gate-Array*

GPU *Graphics Processing Unit*

IC *Integrated Circuit*

i.i.d. *independent and identically distributed*

Image transform *an image is transformed into some another form with the same dimensions*

I **(intensity space)** *All possible instances of intensity combinations within a local neighborhood (e.g. for 3x3 local neighborhood and 8-bit valued image the dimension becomes* 256^9*)*

Intermediate unit permutation *an n-tuple derived from the LBP neighborhood including the center pixel's rank*

Interpolation *the intensity value of an exact image location in between four neighboring pixels is calculated by bilinear weighting (e.g. in the case of circular neighborhood window)*

Keypoint descriptor *an image location and a corresponding feature vector*

KOVA1 *a 96x96 PE mixed-mode array processor by Kovilta*

LBP(M**,** r**)** *Local Binary Pattern with* M *samples and radius* r

LBP-AM *LBP-Adaptive Matching method (based on occurrence maps)*

LDA *Linear Discriminant Analysis*

LIOP *Local Intensity Order Pattern (keypoint descriptor)*

LLM *Local Logical Memory - a local digital memory in CNN PE*

W_i^ψ **(local neighborhood)** *a window of the nearby pixels defined around a certain image location, where the neighborhood* ψ *can be e.g. circular or square shaped.* i *is the location within this window*

LPQ *Local Phase Quantization*

ϕ_{MAP} **(mapping operator)** *A mapping operator from the permutation space into the binary pattern space*

ϕ_{MAP}^{-1} **(mapping operator)** *A mapping operator from the binary pattern space into the permutation space*

MIPA4k *a 64x64 PE mixed-mode processor array by Microelectronics laboratory at University of Turku*

Non-parametric descriptor *image descriptor where the data interpretation is derived without assumptions on the underlying probability distribution (e.g. Gaussian distribution)*

N-tuple *local image feature which uses the ranks of the instances among a local neighborhood*

Occurrence Map *A method to describe LBPs in an image by binary images (one binary image reserved for each LBP)*

OpenCL *Open Computing Language*
Order statistics *area of mathematics which considers the relative ordering of data samples*
Parametric descriptor *image descriptor where the data interpretation is derived with assumptions on the underlying probability distribution*
PCA *Principal Component Analysis*
Photodiode *integrated diode with light induced current*
Performance constraint *some of the main performance constraints of an embedded system are power consumption, speed, physical size etc.*
PE *Processing Element*
P **(permutation space)** *all possible permutations of the LBP neighborhood*
Rank 1 recognition rate *the accuracy of a pattern recognition system where the nearest match is assumed to be the exact output of the classifier*
Rank (rank ordering) *Determining the rank of a pixel in its local neighborhood. For smallest pixel value in a local neighborhood the rank is 1, and for second smallest 2 etc.*
Rank transform *An image transform which changes the intensity valued image into a local rank measure valued image*
Ranked order extraction unit *A circuit which operates by extracting the ranks of its inputs*
Runs level 2 n-tuple *An n-tuple with at most two circular runs (a run is a monotonic increasing or decreasing sequence of rank values in a local circular neighborhood)*
ROC curve *The number of true positives as a function of false positives*
Root permutation *an n-tuple which is formed by removing the rank corresponding to the LBP center pixel from unit permutation and then re-ranking it (into interval 1...M)*
S(), symmetry transform *An image transform which changes the local intensity values into symmetry measures*
SAD *Sum-of-Absolute Difference*
SCAMP processor family *a family of mixed-mode CNN microprocessors by the University of Manchester*
SIMD *Single-Instruction Multiple-Data, a type of microprocessor*
SISD *Single Instruction Single Data, a type of microprocessor*
SMQT *The Successive Mean Quantization Transform*
Symmetry level *the minimum between the total number of zero and one bits in a local binary neighborhood window (see also Descriptor k)*
Uniform LBP *An LBP with at most two 0-1 and 1-0 circular transitions*
Unit permutation *an instance of permutation space, containing the rank for each of the elements in a local LBP neighborhood (including the center)*

Chapter 1

Introduction

Machine vision tries to interpret the optical sensor data obtained from the environment by the means of computers and embedded devices. The purpose is to assist or replace human beings in tasks which are either too time consuming, difficult, or inconvenient for humans to execute. In the past years, progress in many applications which previously have been either too unreliable or too time consuming for machines have been made. This has been facilitated by the development of new algorithmic methods for image analysis and by the rapid increase in available processing power provided by digital computers. For instance, face detection which has traditionally been both too inefficient and too time consuming for machines to perform, can today be carried out reliably even with portable devices such as smartphones and digital cameras [1].

Other applications for machine vision include medical image analysis, automated video surveillance, 3D pose estimation, stereo matching, and many more [2]. In the industrial domain, machine vision can be used for instance to automate some of the procedures in Printed-Circuit-Board assembly lines [3]. Although machine vision has been used successfully in many applications and significant progress has been made, there still exists many challenges in this field. Especially, if comparing the current automated algorithms to the capabilities of human vision, it is evident that a lot of further research needs to be done.

In general, machine vision can be seen as a special type of the more general discipline of pattern recognition [4]. A typical pattern recognition system first preprocesses the captured sensor data, such as image data, by removing additional noise. Feature extraction follows next, with the aim at extracting meaningful and salient parts of the data (see Figure 1.1). The purpose of this step is to reduce the amount of information passed to the next stages of the system. Feature analysis can be used to improve the efficiency of the feature extraction stage in order to improve the overall

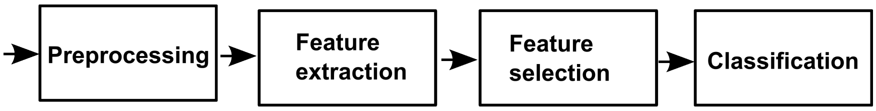


Figure 1.1: A simplified diagram of a typical pattern recognition system [4].

system performance. Finally, the system performs classification based on the extracted features, the purpose of which is to make decisions based on the sensed data. Then, the system usually performs proper procedures depending on the decisions made by the classifier [4].

1.1 Introduction to Local Binary Patterns

In the feature extraction stage of a machine vision system Local Binary Pattern (LBP) features [5], [6] can be used to represent the low-level intensity structure of the image data in a form of short binary codes. A histogram representation is then used to represent the texture within an image region, capturing the actual surface microstructure within the image. Typically, LBPs are fast to compute and their performance in classification tasks has been shown to improve over previous methods [7], [8], [9]. Therefore, the LBPs have recently gained significant popularity in the field of computer vision.

The high performance of the LBPs have been considered to be influenced by its capability to tolerate the changes in image intensity level bias. This is a consequence of the non-linear characteristics of its threshold function, which is unaffected by actual intensity value. The non-linear characteristics of the threshold function implies, that the LBPs belong to the class of *non-parametric* descriptors, where no assumptions on the actual intensity distribution are made. On the contrary, a *parametric* method may assume that the underlying probability distribution of the image pixels is, for instance, Gaussian. Also, LBPs have been interpreted being a unifying approach between the structural and statistical image representations [10], where the actual low-level image structure is incorporated into regional histogram representation.

1.1.1 Derivation of the LBPs

The LBPs are derived from an intensity image by comparing each pixel to its circular neighbours within a certain radius. The original LBP used 8-connectivity nearest neighborhood. The idea is to assign a bit for each of the quantized directional gradients and to simply combine these bits in a decimal form. As a consequence, each pixels local neighborhood can be represent

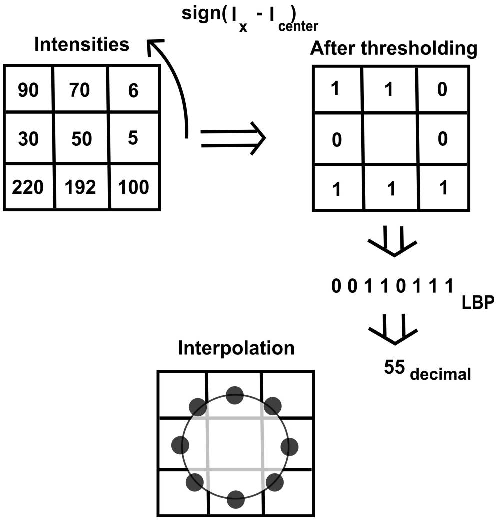


Figure 1.2: Local Binary Pattern with radius r=1 (top) and with interpolation (bottom) with $M = 8$ samples.

by a decimal number, which are used as bins of the LBP histogram. The derivation of the LBP is shown in Figure 1.2. In the case where the exact location of the contour pixel lies between the 4 neighboring pixels, the value of the contour pixel is calculated by bilinear interpolation.

It has been observed, that uniform LBPs containing at most two circular 0-1 and 1-0 transitions are among the most frequent ones with natural images [10], [11]. By selecting the subset of uniform LBPs into final LBP histogram, the length of the feature vector can be reduced while actually obtaining a better recognition accuracy [11]. For example patterns 00010000, 11110000 and 110011 are uniform and patterns 01010000, 11100110 and 010101 are not uniform.

1.1.2 Applications

In the embedded industrial domain the LBPs have been previously applied, for instance, to automatic characterization of paper in an unsupervised manner [12]. Recently, the LBPs were proposed for face recognition in [8]. Later, the original LBP has been extended in various ways. These LBP variants have shown an increase in terms of recognition accuracy in many applications in comparison with the original LBP [6], [13], [14], [15]. Some of these variants modify the original LBP operator [13], while the other may add some additional pre-processing steps before the actual LBP feature extraction [14]. Gabor filtering was used in [14] as a pre-processing step in face recognition by dividing the intensity representation into a plurality of subpresentations. Then, LBPs for each subpresentation were extracted separately. Finally a multi-level histogram was obtained as the feature vector.

A keypoint descriptor based on center-symmetric LBPs was presented in [16]. Recently, Local Phase Quantization (LPQ) descriptor [17] possessing a high insensitivity to image blur was proposed. The LPQ descriptor takes advantage of the blur invariance of the phase of the image Fourier spectra in certain circumstances. The LPQ descriptor is related to the LBP in that short binary codes are used in constructing the final feature vectors. A more comprehensive list of LBP applications can be found in [6].

1.1.3 Other related descriptors

Rank transform, which was originally presented in [18] is based on ordering the intensities of the local square neighborhood window into an ascending order, and determining the rank of the centermost pixel within the window. The transformed value within the transformed image is then replaced by the rank of the corresponding pixel. The rank transform was extended in [19] for estimating stereo correspondence by means of a robust local correlation measure. Lately, perhaps inspired by the success of the LBP methodology, ordinal keypoint descriptors have also been proposed [20], [21]. However, the ordinal rank order representation, in general, seems vulnerable to noise, which was already indicated in [22]. Other non-parametric descriptors and transforms include, for instance, ordinal co-occurrence matrices [23] and Successive Mean Quantization Transform (SMQT) [24].

1.2 Description of the work

1.2.1 The objectives of the thesis

This thesis concentrates on theoretically and practically oriented development of methods for machine vision with non-parametric image descriptors. The theoretically oriented part of the thesis aims at increasing the understanding on the low-level non-parametric descriptors and especially on the LBPs. The relation between LBPs and n-tuples has been studied, and an a priori model for the LBPs has been developed in this context. Practical aspects are considered within the context of non-parametric feature extraction with focal-plane processors. The research in our laboratory has been concentrated on practical implementation of focal plane processors (also known as CNN (Cellular Nonlinear Networks)) and the focus of my work has been to widen this perspective to more practical machine vision applications.

The contribution of the author has been concentrated, for instance, to *bottom-up* type effort - searching for answers to the question - which kind of applications and methods would be benefited most by the already existing focal-plane processor hardware, and *top-down* effort related to the question - which kind of computation should be incorporated to the focal-plane processors in order to facilitate their usage in the future. The main assumptions has been that the theoretically oriented analysis would also give support to the development of the practically oriented algorithms and also that the knowledge of higher level image analysis would facilitate the development of novel low-level image processing circuitry.

1.2.2 Contributions of the thesis

In the following three main contributions of this thesis are briefly summarized. The first contribution is theoretical analysis of LBPs covering papers *1,2,3* and *8*. The second contribution is algorithmic methods for focal-plane processors covering papers *4,5,6,9,10* and the third contribution is LBP extraction with focal-plane processors covering papers *4* and *7*. Paper *4* takes advantage of aspects based on all three contributions.

In the first contribution, an a priori model for the theoretical analysis of LBPs is presented [25]. It is based on studying the distribution of the LBPs with i.i.d. (independent and identically distributed) data and by seeing the pixels as instances of random variables of order statistics [25]. This approach is also linked to the method for characterizing the uniform LBPs in order to reduce their feature vector length presented in [26]. An extension of the aforementioned methods called non-parametric symmetry transform [27] was also developed. It combines some of the main properties of different non-parametric descriptors. Image intensity and rank order representations are replaced in this transform by a more compact measure, providing interesting invariance properties. Finally, the theoretical aspects of uniform LBPs are studied in [28], which summarizes and extends some of the earlier theoretical work.

In the second contribution algorithmic methods for real-time LBP based image analysis are presented. LBP extraction method with CNN-UM is presented in [29]. Frameworks for face localization [30] [31], face recognition [29] and optical flow analysis [31] [32] based on focal-plane processors are also presented. An application of B/W optical flow analysis is demonstrated in laser welding application using the KOVA1 focal-plane processor in [32] and spatter tracking algorithm for laser and manual arc welding applications is presented in [33].

Finally, the third contribution consists of dedicated LBP feature extraction building blocks including the schematics level design of LBP extraction [29]. LBP extraction on MIPA4K processor is also demonstrated by taking advantage of its ranked-order extraction unit [34].

The author's contribution to the papers included

O. Lahdenoja, M. Laiho, A. Paasio, "Reducing the feature vector length in Local Binary Pattern based face recognition", IEEE International Conference on Image Processing, pp. 914-917, (2005)

The main idea and the performed experiments and paper writing for this paper were from the author, while the other authors contributed in improving the presentation and explanation of the used methods.

O. Lahdenoja, E. Alhoniemi, M. Laiho, A. Paasio, "A Shape-preserving non-parametric symmetry transform", 18th International Conference on Pattern Recognition, pp. 373-377, (2006)

The main ideas and experiments and paper writing were from the author, while the other authors contributed in improving the overall quality and presentation of the paper.

O. Lahdenoja, "A Statistical approach for characterising Local Binary Patterns", TUCS Technical Report, No. 795, (2006)

The author was responsible for the ideas for the paper, while Prof. Valery Ipatov kindly provided support in the mathematics and also the attached proof in the appendix.

O. Lahdenoja, M. Laiho, J. Maunu, A. Paasio, "A Massively parallel face recognition system", EURASIP Journal on Embedded Systems - Special Issue on Embedded Vision Systems, Article 72316, (2007)

The main idea of using LBPs with focal-plane processing was from the author. Also the design of the face recognition algorithm was mostly performed by the author and the CNN template design was performed in co-operation with Dr. Laiho. The circuit scematics for the paper and the mismatch simulation were from Dr. Laiho. Mr. Maunu provided assistance in developing and simulating the face recognition algorithm and Prof. Paasio assisted in improving the overall quality of the paper.

O. Lahdenoja, M. Laiho, A. Paasio, "On the spatial distribution of local non-parametric facial shape descriptors", 16th Scandinavian Conference on Image Analysis, pp. 351-358, (2009)

The ideas and experiments for this paper were from the author, while the role of the other authors was to improve the presentation and explanation of the used methods.

O. Lahdenoja, M. Laiho, "Regional image correspondence matching method for SIMD processing, 19th European Conference on Circuit Theory and Design, pp. 802-805, (2009)

Most of the ideas for the paper were from Dr. Laiho, while the author contributed the experiments and paper writing. The face localization algorithm for SIMD processors was a modified version from the one presented in the previous article.

O. Lahdenoja, J. Poikonen, M. Laiho, "Extracting Local Binary Patterns with MIPA4k vision processor", 12th International Workshop on Cellular Nanoscale Networks and Their Applications, (2010)

The measurements with the MIPA4k processor were performed by Dr. Poikonen and Dr. Laiho, while the author contributed the Matlab analysis and most of the paper writing. The explanation of the usage of ranked order statistics unit in LBP extraction was contributed by Dr. Poikonen.

O. Lahdenoja, J. Poikonen, M. Laiho, "Towards Understanding the Formation of Uniform Local Binary Patterns", ISRN Machine Vision, Article ID 429347, 2013.

The main ideas and the experiments as well as the paper writing were made by the author, while the other authors gave support in improving the presentation, illustration and the explanation of the used methods.

O. Lahdenoja, T. Säntti, J. Poikonen, M. Laiho, A. Paasio, "Characterizing spatters in laser welding of thick steel using motion flow analysis", 18th Scandinavian Conference on Image Analysis, pp. 675-686, 2013.

The optical flow based segmentation algorithm and the image analysis algorithms were made by the author, while the other authors performed the welding experiments and contributed to the implementation of the KOVA1 based edge detection and its description.

O. Lahdenoja, T. Säntti, M. Laiho, J. Poikonen, "Spatter Tracking in Laser- and Manual Arc Welding with Sensor-level Pre-processing", 22nd International Conference on Computer Graphics, Visualization and Computer Vision (WSCG), 2014.

All the new results in the paper were made by the author. The laser welding experiments were carried out jointly by the authors, while the other authors performed the arc welding experiments and contributed to the description of the KOVA1 array and the discussion on potential FPGA acceleration of the tracking algorithm.

Chapter 2

Local Binary Pattern Feature Analysis and Applications

In a pattern recognition system after obtaining the sensor data and extracting the features, feature selection may be used to reduce the dimensionality of the feature vector [4]. In order to carry out the feature selection efficiently, the analysis of the features need to be performed. By conducting this step, the performance of the overall pattern recognition system may be increased. As a part of feature analysis, representing the local descriptors in terms of one another can be used to increase the understanding on their discriminative performance.

In [35], the LBPs were represented as combinations of vector quantized linear filters. It is essential to understand the difference and limitations of these descriptors in terms on one another in order to provide necessary means for developing better feature descriptors and or to improve the way in which they are applied. This section describes a study of the feature analysis of LBPs in terms of their occurrence probability and applies this model to a study of the properties of n-tuples and uniform LBPs. Several applications inspired by the study are presented.

2.1 Feature Analysis of Local Binary Patterns

2.1.1 An a priori model

It was noted in [22], that the distribution of n-tuples is largely uneven with natural image data. A tendency that low-order (i.e. monotonic) n-tuple profiles dominate with different kinds of textures was observed. The a priori model described here, is extended in a more comprehensive study of the distribution of n-tuples with natural images. A unifying framework between an order statistics based i.i.d. model [25], n-tuples [22] and uniform LBPs

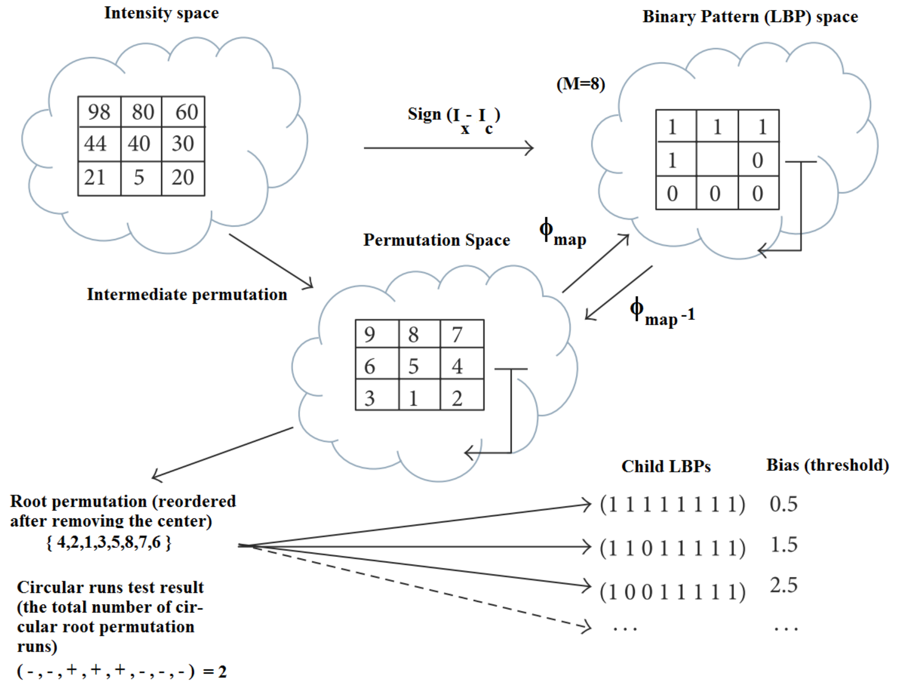


Figure 2.1: The concept of permutation space and circular runs test for permutations [28].

is further presented [28].

A permutation space is defined in LBP(M, r) neighborhood to contain all possible $(M+1)!$ permutations within a circular window. The permutation space instances are equally probable to occur with i.i.d. data [37], if interpolation is not considered. An example of the permutation space is shown in Figure 2.1.

The a priori probability of a LBP with M samples $(M \geq 3)$ to occur with continuously distributed i.i.d. data [25] [38], without considering interpolation, given descriptor k (which is the minimum between the total number of ones and zeros in an LBP) is,

$$P_{LBP}(M, k) = \frac{k!(M-k)!}{(M+1)!} \tag{2.1}$$

A unit permutation is defined as an instance of the permutation space. The mapping operations described in [28] for the intensity space I, the permutation space P and binary pattern space b are described in the following:

Definition 2.1. *Let the intensity space* $\mathbf{I}$ *be defined as sets consisting of intensities* $\{\ I_1,\ I_2,\ I_3,\ \ldots,\ I_M\ \}$ *around the center in addition to the center point* $\{\ I_c\ \}$ *within a local LBP neighborhood. A unit permutation is defined as an individual permutation* $P_n = \{\ R_1,\ R_2,\ R_3,\ \ldots,\ R_M\ \}$ *around the center in addition to the center point rank* $\{\ R_c\ \}$ *formed by rank ordering the intensity samples* I_n *as* R_n *so that the smallest intensity is assigned a rank of 1. In the case of tied intensities, the ranks of the tied intensities are taken from an i.i.d distribution. The length of the unit permutation is then* $(M+1)$*, where the number of contour samples in the corresponding LBP is equal to M [28].*

Definition 2.2. *a) The LBP mapping operator* ϕ_{MAP} *is defined between the instances of spaces* $I_n \Rightarrow b_n$*, or* $P_n \Rightarrow b_n$ *as 1 for instances which have a magnitude greater or equal than the center and 0 for instances which have magnitude smaller than the center. In the case of mapping* $P_n \Rightarrow b_n$ *and ties between the center intensity* $\{\ I_c\ \}$ *and contour intensities* I_n*, the rank of the center* $\{\ R_c\ \}$ *is assigned the minimum within the combined set of the tied ranks* $\{\ R_c\ ,\ R_n\ \}$*. This preserves the uniformity of the LBP also when the permutation space is used. The resulting LBP code is a concatenation* $\{b_1,\ b_2,\ b_3, \ldots b_M\}$ *of the individual bits.*

b) A mapping from intensity space to binary pattern space is defined by applying the LBP mapping operator ϕ_{MAP} *for intensity set* I_n *resulting in binary pattern* b_n*.*

c) A mapping from an instance of intensity space I_n *to an instance of permutation space* P_n *is defined as* $R(I_n)$*, where an operator R extracts the rank ordering of intensity samples among the intensity set* I_n *so that the smallest element will be assigned to value 1.*

d) A mapping from the permutation space into the binary pattern space is defined by applying the LBP mapping operator ϕ_{MAP} *to the set of ranks* P_n *resulting in a binary pattern* b_n *among the binary pattern space.*

e) A reverse mapping ϕ_{MAP}^{-1} *from an instance of binary pattern space* b_n *into permutation space* P_n *is defined indirectly as forming all the* P_n *elements according to the criteria* ϕ_{MAP} *results in a match (from all elements of* P_n $\Rightarrow b_n$*) [28]. In other words,* ϕ_{MAP}^{-1} *generates all unit permutations which correspond to a certain LBP code.*

Uniform patterns are defined formally in circular grayscale neighborhood $I_n = \{I_1 \ldots I_M\}$, including the center pixel I_c with patterns of value U ≤ 2, where $U(I_n)$ is defined in the following [10]:

$$U(I_n) = |s(I_M - I_c) - s(I_1 - I_c)| + \sum_{m=2}^{M} |s(I_m - I_c) - s(I_{m-1} - I_c)| \quad (2.2)$$

The $s(x)$ function gets a value of 1 if $x \geq 0$ and 0 otherwise. Rotation invariant categories for the LBPs and uniform LBPs in particular are defined using $ROR\{x, i\}$ operator, which circularly shifts M-bit binary number x representing the LBP i times to the right,

$$LBP^{ri}_{M,r} = min\{ROR(LBP_{M,r}, i)|i = 0, 1...M - 1\} \quad (2.3)$$

For example, uniform $LBP^{ri}_{6,r}$ pattern class 000011 contains also circularly rotated uniform patterns $LBP_{6,r}$ of 100001, 110000, 011000, 001100 and 000110 [10].

The total number of uniform patterns with respect to k and M is (see an example of the number of uniform patterns in different LBP categories characterized by their total number of 1 bits in Figure 2.15),

$$\#uniform,\, even\, M\ = \begin{cases} 2 & k = 0, \\ 2 * M & 0 < k < M/2, \\ M & k = M/2. \end{cases} \quad (2.4)$$

$$\#uniform, odd\, M\ = \begin{cases} 2 & k = 0, \\ 2 * M & 0 < k <= (M - 1)/2. \end{cases} \quad (2.5)$$

and the a priori probablity of uniform patterns becomes,

$$P^{All}_{u2}(M) = \frac{2}{(M + 1)} + \frac{2M}{(M + 1)!} \sum_{k=1}^{M/2-1} k!(M - k)! \quad (2.6)$$
$$+\frac{M}{(M + 1)!}[(\frac{M}{2})!]^2$$

for even M. In Figure 2.2 the a priori probability of uniform patterns is shown according to the equations 2.4-2.6.

2.1.2 Applying the a priori model for the analysis of LBPs and n-tuples

In natural images there is a tendency, that n-tuples are biased towards lower order monotonic profiles [22]. Here we use this observation to model the

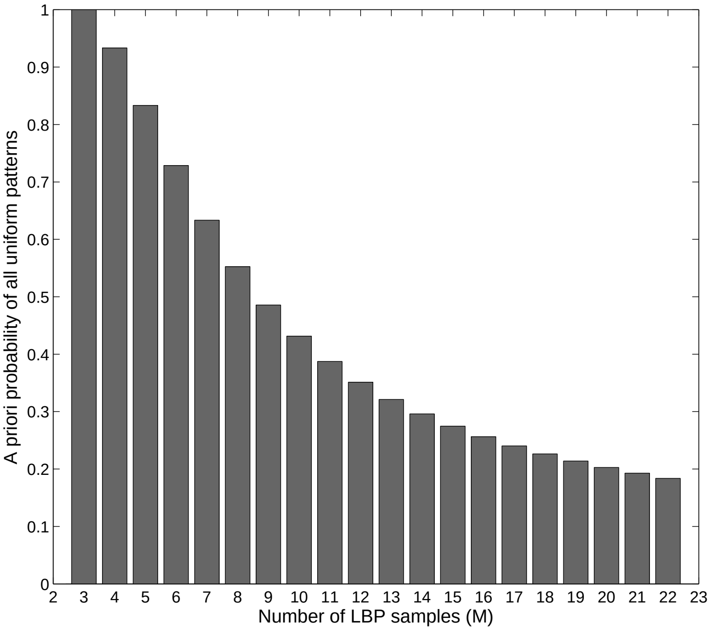


Figure 2.2: A priori probability for uniform patterns with respect to number of samples M and equations 2.4-2.6 [28].

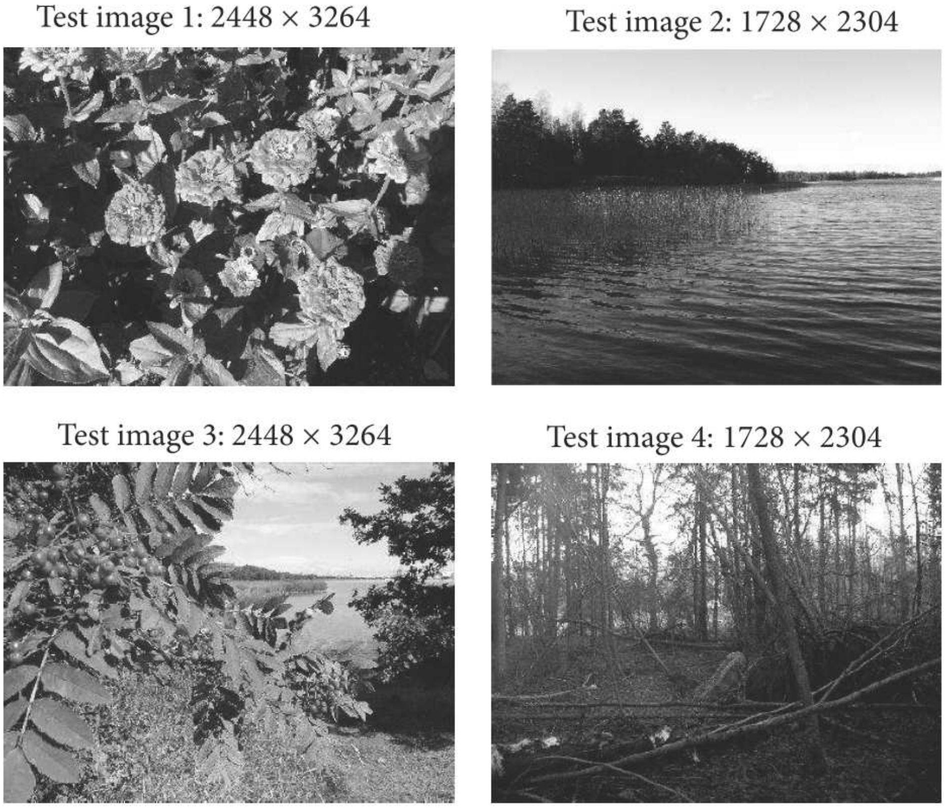

Test image 1: 2448 × 3264
Test image 2: 1728 × 2304
Test image 3: 2448 × 3264
Test image 4: 1728 × 2304

Figure 2.3: Natural test images [28]. Each image was analyzed in which kind of n-tuples they contain. The average neighborhoods of these n-tuples are further shown in Figures 2.5, 2.6, 2.7, 2.8 and 2.9, corresponding to test images 1,2,3,4 and 4 (with a larger radius), respectively. The extraction of the average neighborhoods is described in Figure 2.4. The experiment illustrates, which kind of n-tuple neighborhoods contribute to the formation of uniform LBPs in particular.

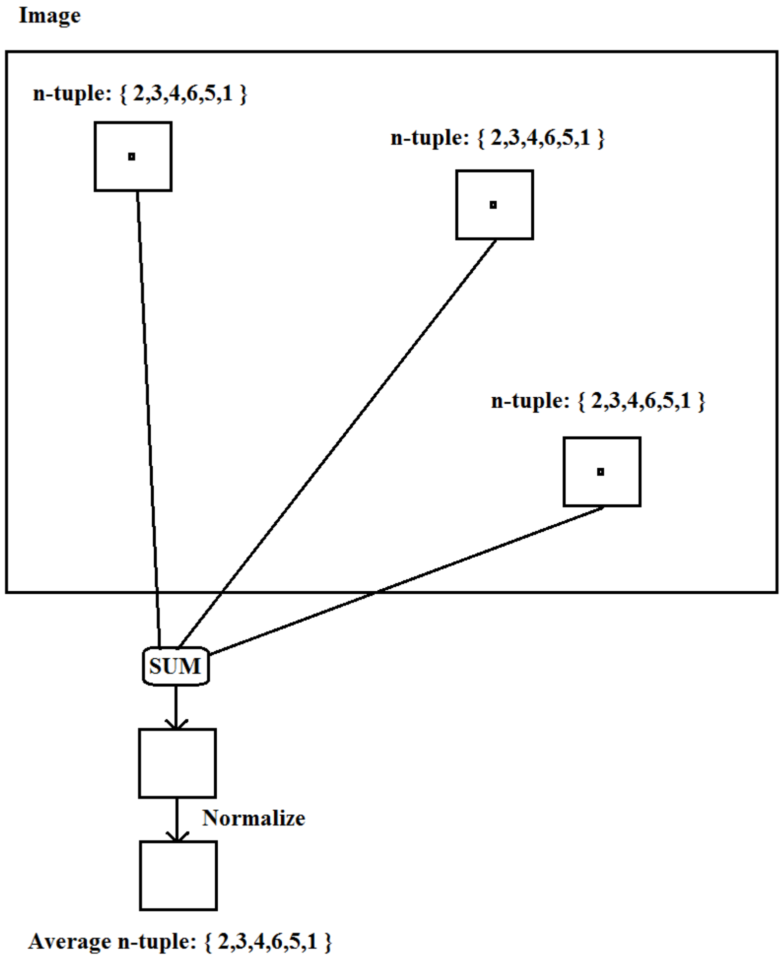


Figure 2.4: Extracting the average n-tuple neighborhoods. Each individual grayscale surrounding (of size 35x35) specific to a certain n-tuple are collected from the test image. They are then placed upon each other and summed together. Normalization is finally performed according to maximum and minimum value of the summed neighborhood. Thus, the produced average grayscale neighborhood describes the selected n-tuple in the specific test images.

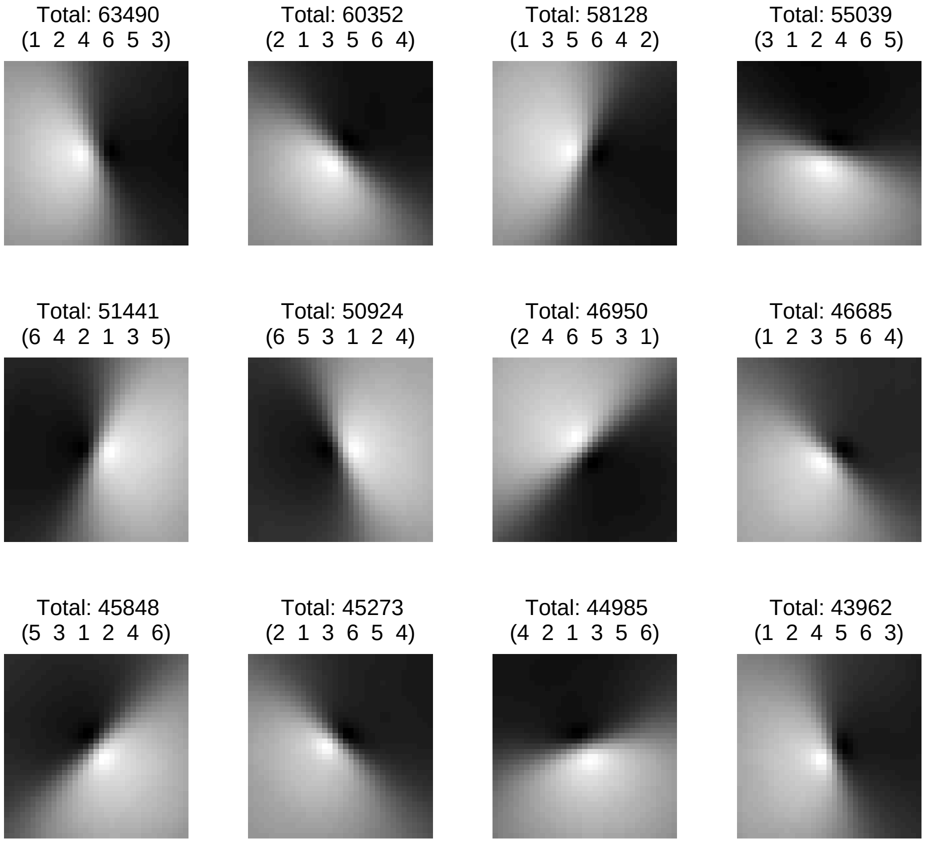


Figure 2.5: Most common n-tuples using LBP (6,2) neighborhood of the test image 1. The number of occurrences for each patch is also shown. It can be observed that the orientations of the n-tuples follow the overall directions of the image gradient.

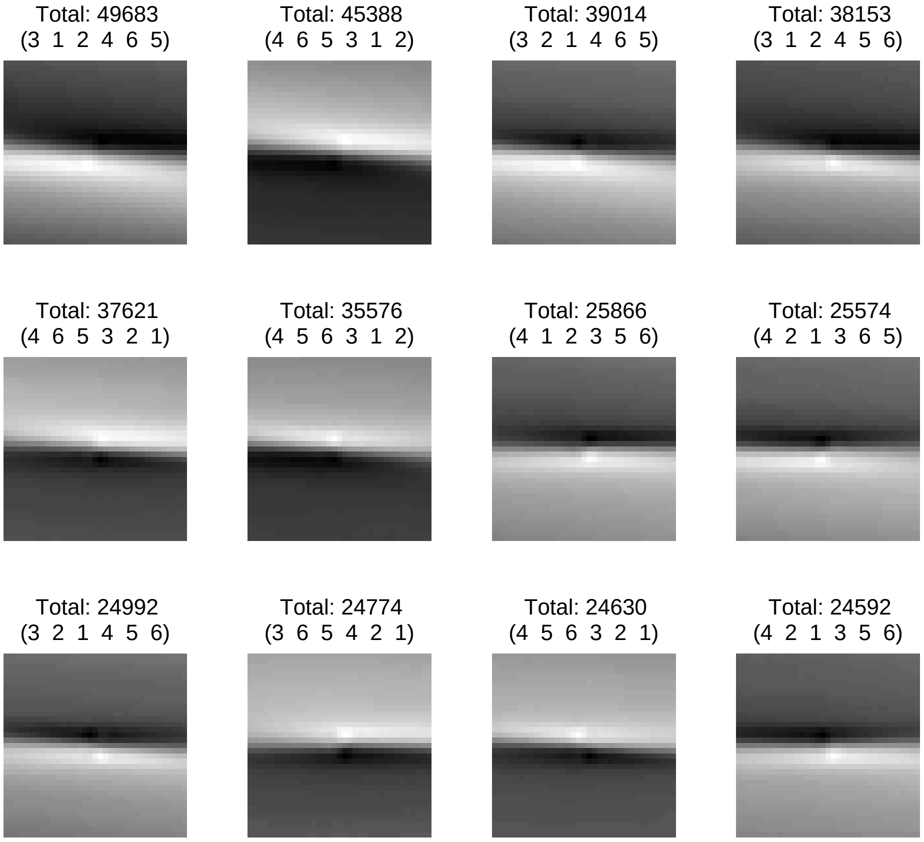


Figure 2.6: Most common n-tuples using LBP(6,2) neighborhood of the test image 2. The number of occurrences for each patch is also shown. It can be observed, that the horizontal gradients which describe the water and the sky dominate the most common n-tuples.

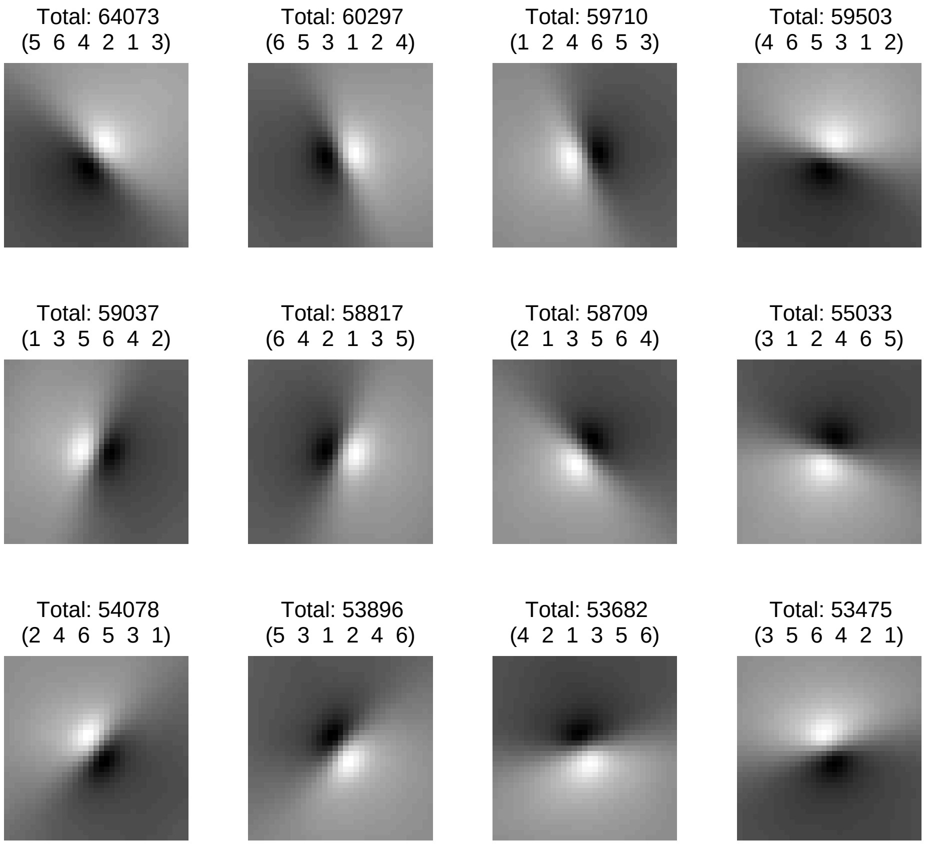


Figure 2.7: Most common n-tuples using LBP(6,2) neighborhood of the test image 3. The number of occurrences for each patch is also shown. As in Figure 2.5, the directions of the n-tuples are distributed in all orientations.

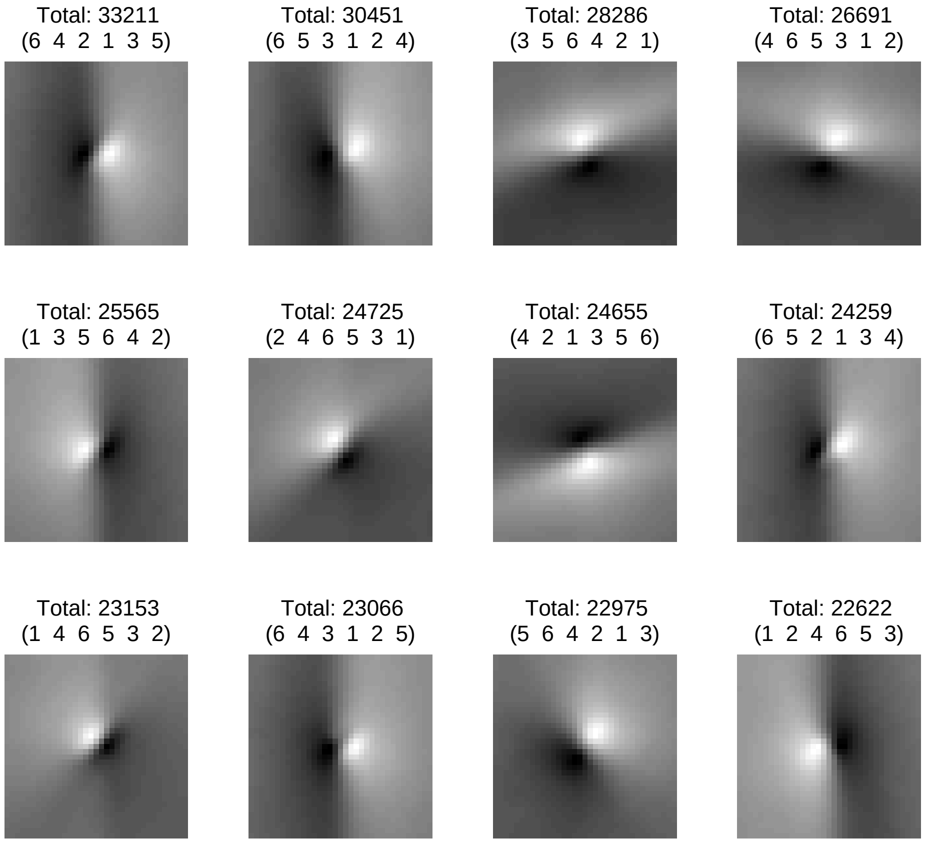


Figure 2.8: Most common n-tuples using LBP(6,2) neighborhood of the test image 4. The number of occurrences for each patch is also shown. It can be observed, that the vertical gradients corresponding to the trees are captured by some of the most common n-tuples.

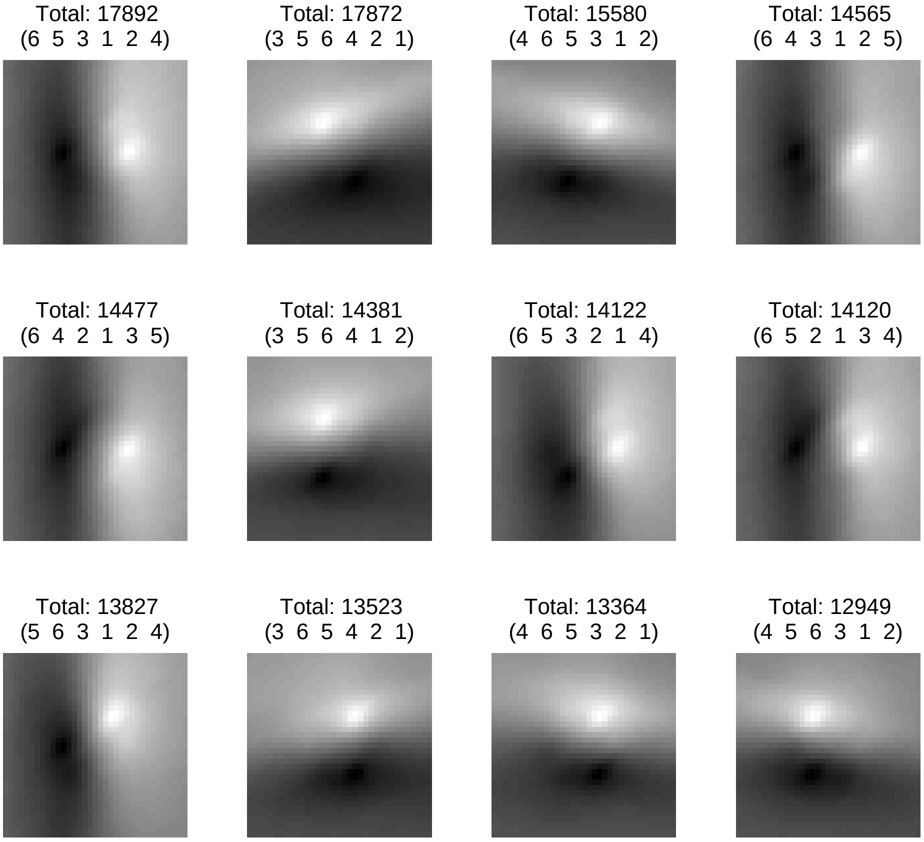


Figure 2.9: Most common n-tuples using LBP(6,6) neighborhood of the test image 4. The number of occurrences for each patch is also shown. It can be observed, that by increasing the radius of the n-tuples the average surroundings become smoother.

correlation of the local neighborhood in the context of n-tuples and LBPs. Thus, the introduced a priori model allows studying the formation process of uniform patterns in particular using the n-tuple representation. This is possible, if the LBPs are seen as a result of the mapping operation from the intensity space through the permutation space to the binary pattern space. In this case the produced *intermediate* unit permutation (*intermediate* in the sense that it is produced from the intensity representation directly towards the binary pattern space, see Figure 2.1) can also be represented as an n-tuple, if the center pixel is not accounted for (this is also denoted as an intermediate root permutation [28]).

Definition 2.3. *An intermediate root permutation is defined as the corresponding n-tuple of the LBP neighborhood when the center pixel is not accounted for [28]. A root permutation is formed if the center pixel's rank of an unit permutation is removed and the set of ranks is re-ordered. For instance the root permutation of unit permutation* P_n*={ 3 } (center pixel), { 1, 2, 6, 5, 4 } contour pixels, is { 1, 2, 5, 4, 3}.*

Figures 2.5, 2.6, 2.7, 2.8 and 2.9 represent average n-tuple surroundings of the test images in Figure 2.3. The surroundings are formed by extracting (a 35x35 pixel sized) grayscale image patch around each individual n-tuple among the specific test image and by combining them as described in Figure 2.4. The number of occurrences of the selected most common n-tuples are also shown. For instance, in Figure 2.5 the most common n-tuple (in the upper left corner) is $\{1, 2, 4, 6, 5, 3\}$. By studying the n-tuples in these specific figures it can be easily observed, that the most common n-tuples are typically of runs level 2 when M is 6. Certain n-tuples with a runs test result of 2 also appear to capture some of the most common edges within the test image (for instance, see in Figure 2.8 and the corresponding test image).

2.1.3 Monotonic n-tuples and uniform patterns

In the following, the a priori probability of the runs level 2 n-tuples is considered [28]. As mentioned, when considering the a priori model with an i.i.d. data, all the n-tuples are of equal probability (without considering interpolation). However, when considering natural images, as it was already illustrated, the distribution becomes biased towards monotonic profiles. Hence, to quantitatively model how the monotonic n-tuple profiles affect to the overall share of uniform patterns with natural images, the a priori model is here applied to the monotonic profiles in particular. For n-tuples only (i.e. without considering the center pixel) the permutation space can be modified so that the center pixel is not accounted for. This way, the overall number of

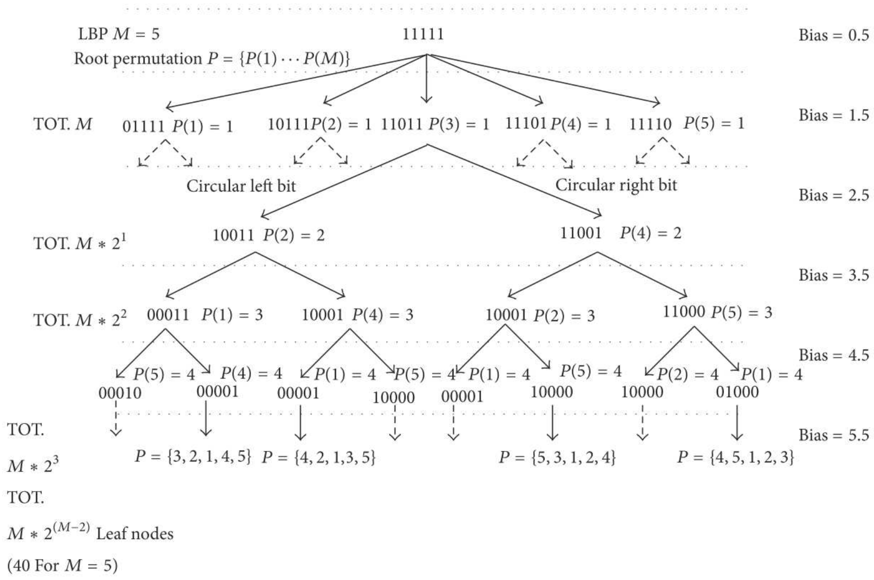


Figure 2.10: Complete uniform permutation tree [28].

runs level 2 n-tuples can be compared to the overall number of all n-tuples (without the center), producing a total dimension of $M!$ for all n-tuples.

An n-tuple could be seen to produce an LBP if (an adjustable) threshold value called bias is assigned to the center location. If considering ranks from $1, 2...M$ the bias could have values of $0.5, 1.5...M + 0.5$, for instance. A child LBP is any of the LBPs produced by this kind of mapping. Next, we define complete uniform n-tuples as n-tuples which changes from the smallest rank towards the largest rank and from the largest rank towards the lowers rank always happen next to each other. They are denoted as complete uniform n-tuples (or *complete uniform root permutations*) because they always map to an uniform LBP independently of the center pixel chosen (called *bias* in this context). For complete uniform n-tuples all the bias values chosen produce a uniform pattern [28].

Definition 2.4. *Complete uniform (intermediate) root permutations are root permutations whose all child LBPs are uniform. The number of complete uniform root permutations is given by $M * 2^{M-2}$, ($M \geq 3$), where M is the number of contour samples in LBP and the length of a root permutation (see Figure 2.10).*

In Figure 2.10 the formation of complete uniform n-tuples is illustrated. Thus, the total number of complete uniform n-tuples can be calculated from the number of leaf nodes in a perfect binary tree of height M-2 (see Fig-

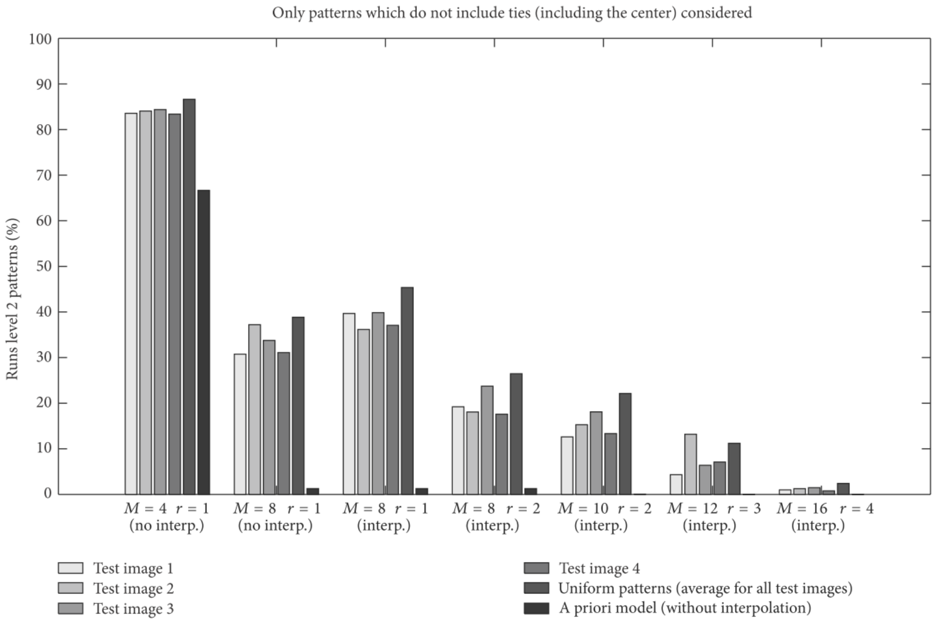


Figure 2.11: The distribution of runs level 2 n-tuples within the test images [28].

ure 2.10). In Figure 2.11 the distribution of runs level 2 n-tuples is shown on each of the test images in Figure 2.3. The images containing monotonic changes would appear to contain slightly more runs level 2 patterns than the others, however not significantly. In Figure 2.11 the a priori probability of the runs level 2 n-tuples is also shown (without interpolation). The purpose of the figure is also to illustrate how natural images increase the number of monotonic n-tuples in comparison with the a priori model. For example, with $M = 8$ and $r = 1$ the a priori probability of the runs level 2 n-tuples is below 5%, while increased to above 30% with the test images. Note that the number of runs level 2 patterns among uniform patterns is normalized to the number of uniform patterns. Thus, the slight increase in the number of runs level 2 patterns among uniform patterns may also be caused by this.

The number of permutations resulting into uniform patterns is further plotted in Fig. 2.12. The leftmost image represents the permutation space of uniform patterns so that all unit permutations within the LBP neighborhood are counted once (an equal probability for each permutation according to the applied i.i.d. model). The image on the right represents the permutations with natural test images. It can be observed, that in the a priori model with i.i.d data (on the left) the low value of k uniform LBPs are produced very frequently for permutations with longest run of 3,4 and 5. This is because

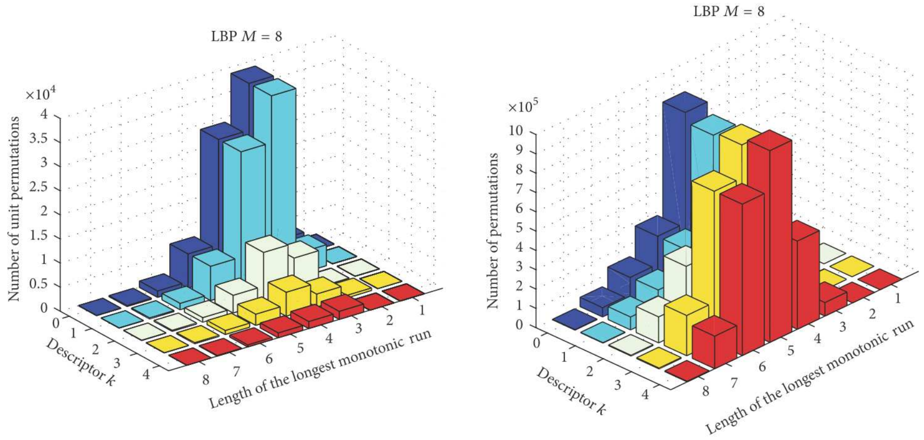


Figure 2.12: The distribution n-tuples with respect to descriptor k and the monotonicity of the permutation [28]. The same test is applied to the a priori model (on the left) and natural test images (on the right). With natural image data the high k uniform LBP patterns dominate the occurrence probability. Note also that monotonic profiles are more common and that the monotonic permutations tend to populate especially the high k patterns.

of the high a priori probability of uniform patterns 00000000 and 11111111 as well as uniform patters with 7 zero bits and 1 one bit and 7 one bits and 1 zero bit (their a priori probability is the largest). Due to a lower number of possible combinations the high k patterns are more rare in this case. With natural image data the high k uniform LBP patterns dominate the occurrence probability. Note also that monotonic profiles are more common and that the monotonic permutations tend to populate especially the high k patterns.

2.2 Applications of the LBP Analysis

In this section, the methods presented for the feature analysis of the LBPs are further applied to practical applications. The objective is to improve the performance of the LBP and to develop new methods inspired by the original LBP representation. It is shown, that by seeing the LBPs as a part of the more general class of non-parametric descriptors, a new non-parametric image transform can be introduced.

2.2.1 On the discriminative quality of the high k uniform patterns

An illustrating example of the feature selection in the context of the LBP methodology is to select only uniform patterns for the final feature vector. In an empirical study in [26] the uniform patterns were further analyzed by dividing them into different categories based on a measure derived from their total number of zero and one bits. It appeared in [26], that in face recognition this kind of separation criteria was also supported by the discriminative capability the LBP. A symmetry level L_{sym} for the uniform LBPs was defined as the minimum between the cardinalities of the sets consisting of zero and one bits. The facial image database used in [8] and [26] was the FERET (The Facial Recognition Technology) database by NIST (U.S. National Institute of Standards and Technology) [36].

The LBPs can also be divided into rotation invariant categories by their number of zero and one bits (see previous equation 2.3). This is illustrated in Figure 2.15, where one row of codes always represent one rotation invariant category with a number of samples M of 8. In this figure the symmetry levels L_{sym} can be seen to contain two rotation invariant categories, the first one consisting of the original codes and the second one from their negated counterparts (changing the 1 bits to 0's and vice versa). In [26], if considering the rotation invariant LBP classes separately, the classes with higher L_{sym} performed better. Therefore, the concept of symmetry levels for the uniform LBPs was defined. This was used in [26] to reduce the length of the final LBP feature vector. Figure 2.13 shows the recognition rate in the FERET [36] set with respect to each of the rank levels (number of ones in the LBPs) in LBP(12,2) neighborhood. The test protocol was the same as in [8] except that only the selected bins of the corresponding rotation invariant category were used. The main concept in reducing the feature vector length in [26] was to assign a bin of their own for the higher symmetry level patterns and to combine the lower symmetry level pattern categories. Thus, the number of ones in each pattern defined which patterns were combined.

Table 2.1 shows the final recognition results before and after the feature vector length reduction using the recognition protocol proposed in [8] and the FERET database. The original length of the feature vector of both the query face and the stored database faces were 2301 bins in [8] achieving a recognition rate of 76.4% using the LBP$(8, 2)$ neighborhood. With the proposed method the corresponding feature vector length was 1170 histogram bins achieving a recognition rate of 76.1% in the LBP$(8, 2)$ neighborhood (the other L_{sym} categories than 3 and 4 were combined). In LBP$(12, 2)$ neighborhood a recognition rate of 77.4% was achieved with 1794 bins (the other L_{sym} categories than 5 and 6 were combined). Thus a slight increase in the recognition rate was obtained with a reduced feature vector length.

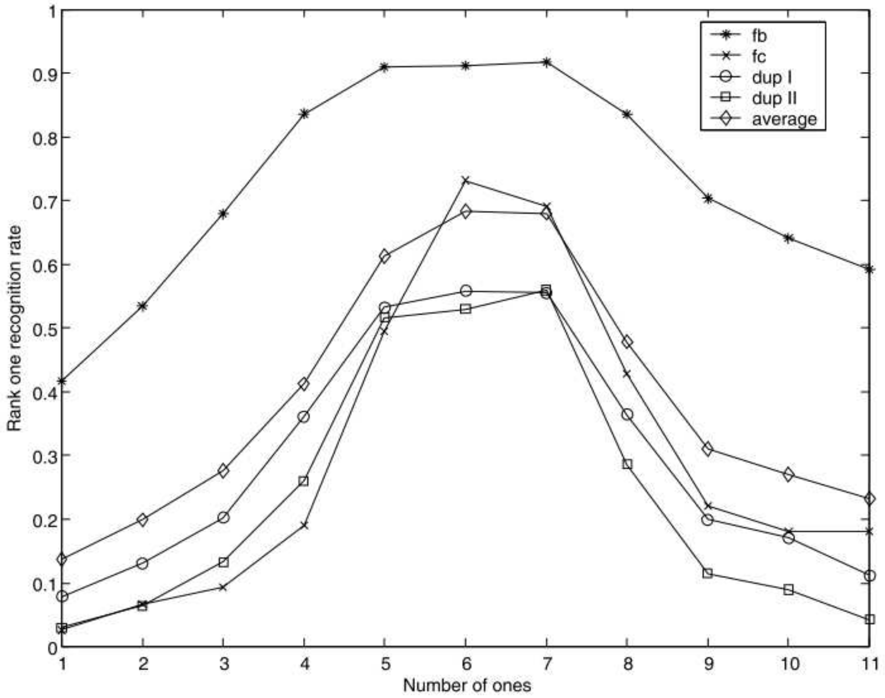


Figure 2.13: The recognition rate in FERET set [36] with respect to rank level of the LBPs [26]. **(c) 2005 IEEE.**

Method	fv length	fb	fc	dup1	dup2	average
LBP(8,2)	9984	97%	75%	66%	60%	74.6%
LBP^{u2}(8,2)	2301	97%	79%	66%	64%	76.4%
LBP^{u2}(8,1)	2301	95%	75%	57%	45%	70.0%
LBP^{u2}(12,2) L_{sym}=[5,6]	1794	95%	85%	64%	66%	77.4%
LBP^{u2}(8,2) L_{sym}=[3,4]	1170	96%	79%	65%	64%	76.1%
LBP^{u2}(8,1) L_{sym}=[3,4]	1170	94%	76%	58%	50%	69.4%
LBP^{u2}(8,2) L_{sym}=[4]	624	94%	81%	60%	56%	72.9%
LBP^{u2}(8,1) L_{sym}=[4]	624	91%	74%	53%	43%	65.2%
PCA MahCosine	N.A.	85%	65%	44%	22%	54.2%
Bayesian MAP	N.A.	82%	37%	52%	32%	50.8%
EBGM optimal	N.A.	90%	42%	46%	24%	50.5%
LDA ldasoft	N.A.	73%	47%	45%	18%	45.8%

Table 2.1: Results with FERET database [36]. The FERET database contains a gallery set of 1196 images, and four probe sets; fb, fc, dup1 and dup2, which are evaluated against the gallery set. The fb set contains changes in facial expression (a total of 1195 images), the fc set contains changes in the illumination of the faces (a total of 194 images) and the dup1 (a total of 722 images) and dup2 (a total of 234 images) sets contain the aging of the subjects. **(c) 2005 IEEE.**

2.2.2 Negation invariance

Further analysis on the symmetry level concept reveals, that it considers an image and its negated counterpart in a similar way. Image negation is a decreasing mapping of intensities. Let the circular grayscale neighborhood be again denoted as I_n={ $I_1...I_M$ } in addition to center pixel's grayscale value I_c. For example, consider the LBP intensity set I_c { 50 } for the LBP center pixel and I_n { 10, 20, 100, 60, 70, 130, 190, 55 } for the contour pixels (i.e. $M = 8$). The corresponding LBP is 00111111. If all of the intensities are negated (i.e. applying a mapping of 255-I_n), the resulting I'_c is { 205 }, and the resulting contour intensities I'_n are { 245, 235, 155, 195, 185, 125, 65, 200 }. The negated LBP is then 11000000, while the symmetry level (L_{sym}, k) of 2 is preserved. In Figure 2.15, for instance, $L_{sym} = 1$ category contains all uniform patterns with 7 zero bits and 1 one bit, which form the rotation invariant category of the pattern 00000001 (see the definition in previous equation 2.3), and also the negations of these patterns (all uniform patterns with 7 one bits and 1 zero bit). Thus, two rotation invariant categories are in this case included to the same L_{sym} category. Figure 2.15 further illustrates how the distribution of the bins for certain facial locations change with different LBP codes, with the rotation invariant categories and with the symmetry level categories.

2.2.3 Non-Parametric Intensity Symmetry Transform

When understanding the relation between the LBPs and other non-parametric descriptors, such as the rank transform, it is possible to take advantage of the feature analysis of the LBPs as an inspiration for developing new non-parametric descriptors. In the following a modification to the well known rank transform proposed in [18], is explained. In the rank transform the location of the center pixel within each local window is replaced by the pixels rank. In the proposed non-parametric symmetry transform the center pixel is always replaced by the symmetry level measure of the corresponding neighborhood, which is the minimum between the cardinalities of the number of ones and zeros within the window. The main difference to the symmetry of the LBP neighborhood is that now arbitrary neighborhood (e.g. a square window) can be used. Hence, the concept of symmetry levels is modified into a non-parametric transform [27].

Let a local window be defined as W_i^{ψ}, where i is the index within the window W (the window can be 1D, 2D, 3D etc.) in neighborhood ψ. The instances i that have values above the center element of the window are denoted with '1' bits, and the instances i that have values below the center element as '0' bits, respectively, i.e,

$$W_i^{\psi} = \begin{cases} 1 \ \ (0), & I_i^{\psi} > I_{\text{center}}^{\psi} \\ 0 \ \ (1), & I_i^{\psi} < I_{\text{center}}^{\psi} \\ \text{undefined}, & i = \text{center}. \end{cases} \tag{2.7}$$

Then the generalized definition for local intensity symmetry level is,

$$L_{sym} = \min\{Card(W_i^{\psi} =' 1'), Card(W_i^{\psi} =' 0')\}. \tag{2.8}$$

where $Card(W_i^{\psi} =' 0')$ denotes the total number of elements with value '0' and $Card(W_i^{\psi} =' 1')$ the total number of elements with value '1'. In words, the minimum between the number of occurrences of '1's and '0's is chosen as the transformed value. The transformed values vary from $[0, Sum_i/2]$ for even Sum_i and for odd Sum_i, between $[0, (Sum_i - 1)/2]$, where Sum_i is the number of elements i in window W_i^{ψ}. As in rank transform, the whole image except its borders is searched through with the local window, so that the pixel in the centermost location of the local window is always replaced by the transformed value.

Properties of the Non-parametric Symmetry Transform

Non-parametric symmetry transform is always symmetric around the median of the local neighborhood. As a consequence, the interval of the possible output values is reduced by a half in comparison with the rank transform. As the rank transform and LBPs, symmetry transform is invariant to increasing monotonic changes in image intensity bias. An interesting additional property is invariance to any monotonic decreasing image transformation such as image negation [27]. Hence, the intensity symmetry transform is invariant to any monotonic mapping of the intensities. In Figure 2.14, the negation invariance property of the symmetry transform is illustrated. Being a non-parametric transform, the intensity symmetry transform should also be robust against illumination changes.

Applications of Non-parametric Symmetry Transform

In [27], the non-parametric symmetry was proposed for correspondence matching and finding salient image locations. With saliency, in this case areas which were monotonically changing were meant. The considerations in the previous sections could indicate, that the symmetry transform (as well as the rank transform) would also possess insensitivity to local rotation, since in the case of LBPs the rotation invariant categories are included into the symmetry level categories. In the case of the largest symmetry level in Figure 2.15, the negated LBP patterns also belong to the same rotation

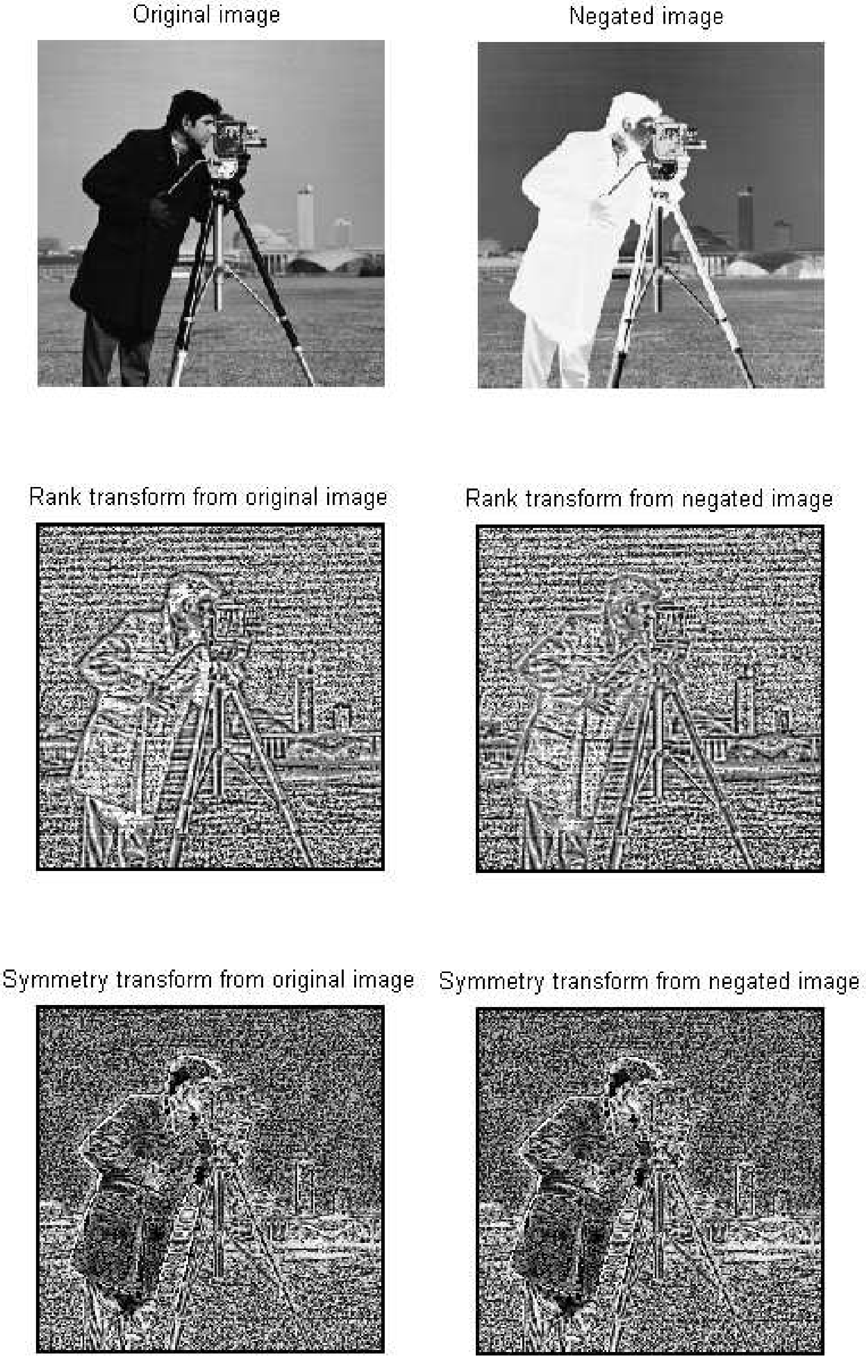

Original image
Negated image
Rank transform from original image
Rank transform from negated image
Symmetry transform from original image
Symmetry transform from negated image

Figure 2.14: Original and negated image and their rank and symmetry transforms. It can be observed that the symmetry transform is not changed when negating the image.

invariance class. In [27], the clusters of the highest level of intensity symmetry image locations remained largely unchanged in certain experiments of rotation and intensity bias changes [27].

Being a local measure, the intensity symmetry naturally can not be as such used for interest point matching, since it does not incorporate a means for deriving a discriminating feature vectors like the interest point detectors, such as [39]. Also, it could be expected that certain information is lost when reducing the dynamic range of the rank transform. In a face recognition experiment using the FERET set, however, an average increase of 15.0% compared with rank transform was obtained with all sets (without applying weighting) in [27]. Absolute valued distance $\sum_{image} abs(S(image1) - S(image2))$ (L1 distance) was used as the distance metric between two symmetry transformed images ($S()$ denotes the symmetry transform). The recognition results are shown in Table 2.2.

The initial assumption of the authors was that better performance of the symmetry transform in face recognition could be linked to its invariance properties, but the later work in [40] would seem to indicate, that it may also be caused by a better tolerance of the intensity symmetry transform to noise. This could also explain why heavy averaging filter was required for both the rank transform and the intensity symmetry transform in order to obtain the best recognition results. However, it is still possible that these two properties are linked to each other. Thus, further investigation of this issue would be interesting in the future. The rank transform and its other kind of modifications have been further studied in [40], where it was proposed also for edge detection.

2.2.4 Illustrating the average LBPs in face images

The LBP images in Figure 2.15 were extracted in the following way; *(a)* A sufficiently large subset Z (e.g. Z = 1000) of spatially normalized facial images was selected (in this case the images were taken from the FERET database),

(b) To all of the selected images an LBP transform was applied in (8,1) neighborhood (the neighbourhood could also be some other), which produced an MxNxZ matrix, where MxN is the dimension of the LBP transformed images and Z is the total number of images,

(c) A MxNxD histogram was constructed to represent the LBP distribution in each facial location (x, y) over the MxNxZ locations. In this case, D is the total number of uniform LBP bins,

(d) Finally, the basis images shown in Figure 2.15 were extracted by intersecting one uniform LBP bin magnitude at a time from the MxNxD matrix into a MxN density image of the specific LBP code. In these images (Figure 2.15) the LBP density increases as the image gets brighter. An

Subset	fb	fc	dup1	dup2	avg.
Images	1195	194	722	234	-
Rank transform Non-weighted	72.9%	55.2%	41.6%	35.9%	51.4%
Symmetry transform Non-weighted	88.0%	70.1%	54.6%	53.0%	66.4%
LBP(8,2) [9] Non-weighted	93%	51%	61%	50%	64%
LBP(8,2) [9] Weighted	97%	79%	66%	64%	76%
PCA MahCosine	85%	65%	44%	22%	54%
Bayesian MAP	82%	37%	52%	32%	51%
EBGM optimal	90%	42%	46%	24%	51%
LDA ldasoft	73%	47%	45%	18%	46%

Table 2.2: Rank 1 recognition rate by using the FERET database [27]. The symmetry transform performed clearly better than the rank transform in the face recognition experiment. However, the weighted LBP approach [8] still performed better.

additional step for the basis images was applied in [30] so that the dynamic range of each image was normalized by the total number of bins within that image. Hence, the basis images represent the probability density of the LBP bins accross the training set on average. The basis image framework could be used to study how the different LBP codes are distributed across the overall area of the human face.

2.2.5 Face localization with normalized LBP histograms

The basis image approach was further applied in [30] to face detection with a moderate detection accuracy. The detector used a look-up table based brute force search of 21x21 sized basis images across all scales. One of the reasons for not achieving a state-of-the-art the performance could lie in the alignment procedure before step 1, since it does not account for faces in different orientations. Also different illumination conditions caused difficulties to the detector in some cases. The performance of the detector in terms of number of false positives against and true positive rate (ROC curve) is shown in Figure 2.16. Still, the approach would seem to provide simple and efficient means for face localization (See also Chapter 3).

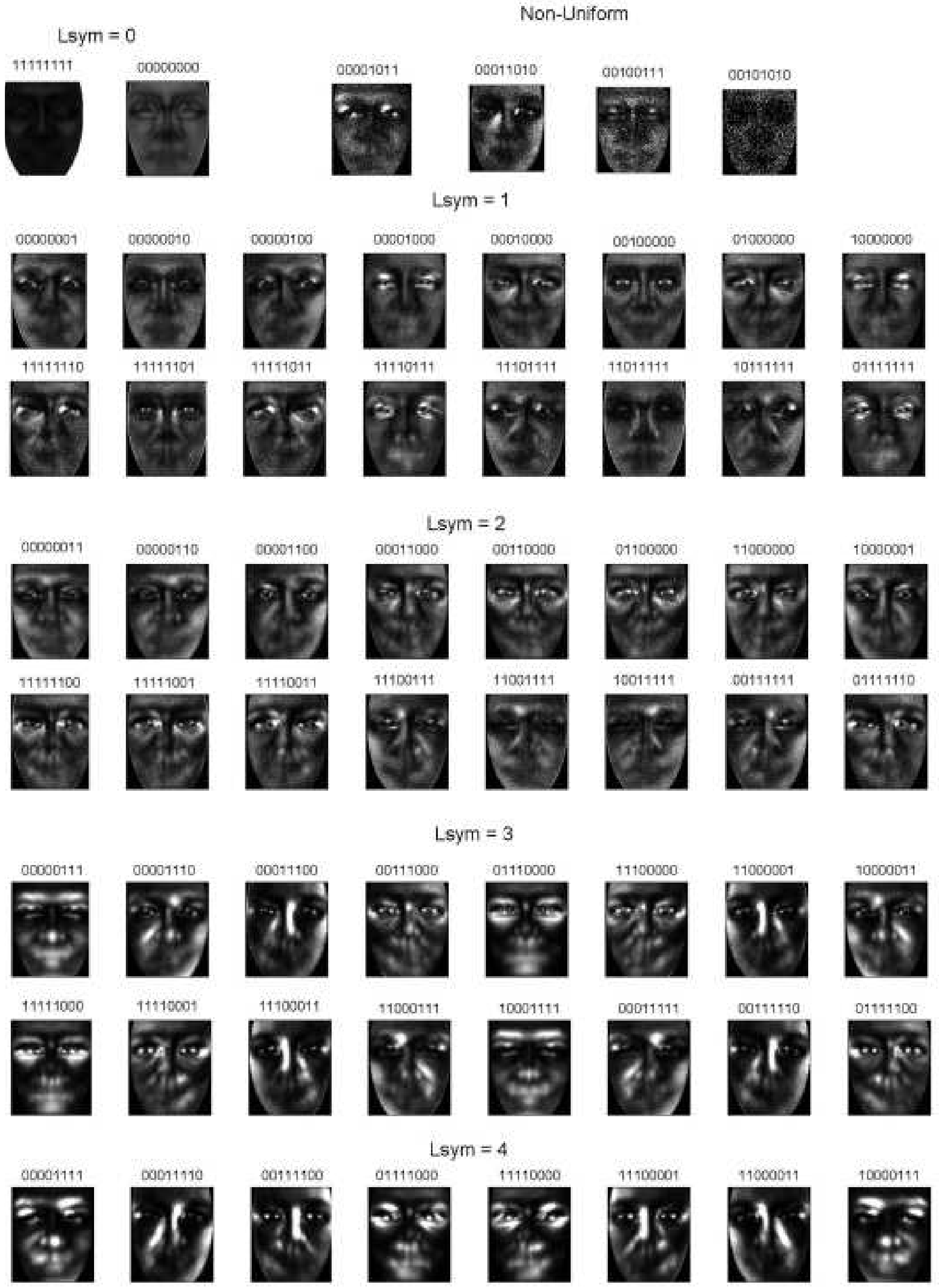


Figure 2.15: LBP basis images for uniform and non-uniform patterns in (8,1) neighborhood [30]. **Reproduced with kind permission from Springer Science and Business Media, 2009.**

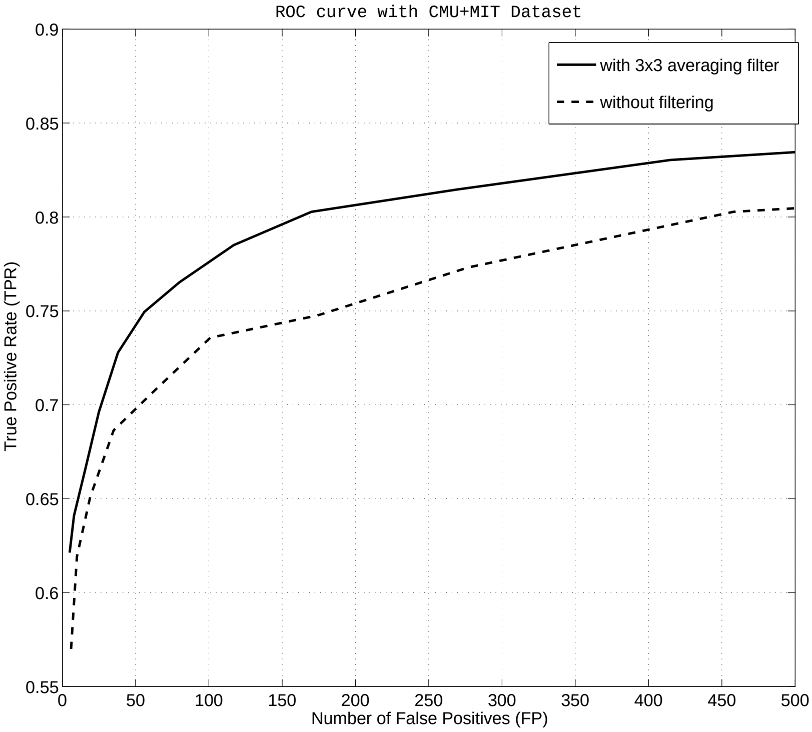


Figure 2.16: ROC curve of the proposed basis image face detector.

2.3 Discussion

In the recently proposed LIOP (Local Intensity Order Pattern) [20] descriptor, n-tuples were proposed as the main low-level feature representation of a rotation invariant keypoint descriptor. Therefore, the analysis of n-tuples and their relation to LBPs is also of major current interest. Representing the LBPs in a more general framework of n-tuples was used to point out that circularly monotonic n-tuples contribute to the increase of the number of uniform pattern bins in the LBP histogram.

The application of the feature analysis was to take advantage of the discriminative power of the high descriptor k uniform patterns to reduce the length of the LBP feature vector and to widen this interpretation into a non-parametric symmetry transform in arbitrary neighborhoods. As the LBP (see Chapter 3), the symmetry transform could be efficiently performed in parallel, for instance using a FPGA implementation. For Census transform these kind of applications have been already proposed. Previously in [19] the rank order statistics were used to derive a robust correlation measure between local image patches in two images for stereo matching. The main difference of the non-parametric symmetry transform to the one in [19] is that instead of calculating the distance between two local pathes in two images separately, in this case both images are transformed independently with a simpler procedure, and then a comparison between the transformed images is carried out. Since the local ψ neighborhood can be arbitrary, also temporal, spatiotemporal or 3D analysis could potentially be performed with the intensity symmetry transform.

Chapter 3

Application to Focal-Plane Processors

Focal-Plane processors belong to the class of SIMD (Single-Instruction Multiple-Data) processors. They are typically designed as full custom ASICs, where a single processing element performs analog, mixed-mode and digital processing. The most obvious target usage of the focal-plane processors lies in high-speed image analysis, since the photodiodes are efficiently embedded onto the same silicon die with the actual processor circuitry [41]. The difference between SIMD and more traditional SISD (Single Instruction Single Data) microprocessors lies in the amount of processing performed in parallel, which potentially can give a positive impact on the processing speed and power efficiency.

In this chapter, the usage of LBPs in the embedded domain is studied within the context of focal plane processors. Although the extraction of the LBPs is relatively fast and efficient even with SISD hardware, the benefit of the proposed approach could be in increased suitability of the focal-plane processors into new applications. Also, the focal-plane processors could provide better performance in comparison with SISD hardware in applications where high speed, power efficiency and compact size are required.

3.1 Introduction to Focal-Plane processors

The instructions are performed in focal plane processors in SIMD fashion i.e. for the whole array simultaneously, by applying successive array-wise operations on the currently available data. The neighborhood connectivity of the focal plane processors is usually limited to a 8-nearest-neighborhood connectivity or to the second neighborhood at most, which limits their usage in many cases. However, by applying successive operations, the execution of more complex image processing beyond the local neighborhood can also be performed. The basic principles of the cellular array processors (e.g. Cellular Neural Networks, CNNs) were proposed already in [42]. Still, many focal-plane chips use some of the concepts of the traditional CNN paradigm.

Examples of modern SIMD type focal plane processors include the ACE-16k processor [43], the EYE-RIS platform [44], ASPA and SCAMP-5 processors [45], [46], and the MIPA4k processor [47]. Some of these are discussed in the following in more detail.

The CNN-UM (Cellular Neural Network - Universal Machine) was introduced in [48] and [49]. The ACE processor family consists of ACE400, ACE-4k and ACE16k processors [43]. ACE16k consists of an array of 128x128 processing elements (PE) and was implemented in 0.35um semiconductor process, while delivering the performance of 330GOPS. The commercial EYE-RIS system, which is a follower of ACE16k was proposed in [44].

The SCAMP processor family has been presented in [50], [51], [46]. An important characteristic of the SCAMP family is the reduction of the cell area by time-multiplexing the neighborhood connections. Operations such as pixel level snakes have been demonstrated on SCAMP-3 [51]. SCAMP-5 chip is a 256x256 cell array [46] capable of asynchronously propagating array operations. The tracking of 5 closed-shape objects at 25kfps and single object at 100kfps was demonstrated using the SCAMP-5 vision chip in [46]. The ASPA processors presented in [45] also provide the capability of asynchronously propagating operations.

A QCIF resolution CNN chip was presented in [52]. Later, a 64x64 element MIPA4k [47] mixed-mode processor array was proposed with the capabilities of asynchronously propagating morphological reconstruction, adaptive image brightness control, and binary programmable B/W cell templates. Some of the early work within this thesis has been done before our research team implemented the MIPA4k processor, which in general, have influenced into some of its implemented features. For instance, there are possibilities for LBP extraction in the MIPA4k processor. The successor of the MIPA4k is a commercial 96x96 element KOVA1 array processor [53].

3.2 Extracting Local Binary Patterns with Focal Plane Processors

In many applications of the LBPs the first and the second order local neighborhoods have shown to be capable of providing state-of-the art recognition rates [9]. In fact, the LBP was originally proposed for the 8-nearest neighborhood and only later it has been extended to other radiuses (r) and sample sizes (M). The possibility of extracting the LBPs with focal plane processors was first proposed during this work included in references [54], [55] and [29]. In [55] the LBP extraction was proposed by applying directional comparison templates into each of the local neighborhood directions successively, and then applying the piecewise linear output non-linearity. A dedicated hardware for parallel extraction of the LBPs was described in [54].

Next, the basic LBP extraction with CNN in (8,1) neighborhood is explained. The input image is read from the cell input u and the state x is assigned to zero. The comparison is carried out by the following equation [42],

$$\dot{x}_{ij} = -x_{ij} + a \cdot y_{kl} + \sum_{kl \in N_r} B_{ij,kl} u_{kl} \tag{3.1}$$

where a is the center element of the A-template. In the A template all elements are zero, except the center element with a value of 1. The output nonlinearity is,

$$y_{ij} = \varphi(x_{ij}) = \frac{|x_{ij} + 1| - |x_{ij} - 1|}{2} \tag{3.2}$$

Equation 3.3 and Table 3.1 further shows the comparison templates for the LBP(8,1) neighborhood without interpolation.

$$B_{r=1} = \begin{bmatrix} b_1 & b_2 & b_3 \\ b_8 & b_0 & b_4 \\ b_7 & b_6 & b_5 \end{bmatrix} \tag{3.3}$$

In the following, an example of the LBP extraction with the embedded MIPA4k processor is further shown [34]. In order to make the comparison between the neighboring pixels more robust (several non-idealities were present in the MIPA4k processor test environment), the ranked order statistics extraction unit of the MIPA4k processor [47] was used. The ranked order statistics extraction unit of the MIPA4k is capable of indicating the

Template	b_0	b_1	b_2	b_3	b_4	b_5	b_6	b_7	b_8
T_1	1	-1	0	0	0	0	0	0	0
T_2	1	0	-1	0	0	0	0	0	0
T_3	1	0	0	-1	0	0	0	0	0
T_4	1	0	0	0	-1	0	0	0	0
T_5	1	0	0	0	0	-1	0	0	0
T_6	1	0	0	0	0	0	-1	0	0
T_7	1	0	0	0	0	0	0	-1	0
T_8	1	0	0	0	0	0	0	0	-1

Table 3.1: Threshold templates for LBP with r of one and M of eight [29].

relative ordering of any of the main vertical or horizontal neighbors including the center. In this case, the bits in directions (N, W, S, E) were first extracted by a comparison to the center pixel directly. Since the diagonal neighbors cannot be accessed directly by the order statistics unit, the following procedure was applied; for instance to get the NE bit the image was first shifted to the west and then the ordering between the eastern and the northern directions was determined. In this case the analog non-idealities in MIPA4k D/A conversion and order statistics extraction reduced the number of matching bits to 81.6% [34] in a (8,1) neighborhood, in comparison with the ideal bits obtained from Matlab. Figure 3.1 shows the contents of the local digital memories of the 64x64 cells with MIPA4k after performing the thresholding operation.

The total estimated time for pixel-parallel LBP extraction with MIPA4k was less than 10us. In [56] a standard CPU implementation of LBP extraction with an image size of 1280x720 pixels took 1.8 - 29ms depending on the configuration and the best GPU based LBP extraction result was 1.1ms (optimized OpenCL implementation, without considering the time for I/O). The total processing time per pixel with MIPA4k was 2.44ns, for standard CPU 1.95ns-31.5ns and for optimized GPU implementation 1.19ns. Since this was an initial implementation of the LBP extraction and it was implemented indirectly via the ranked order extraction unit, the performance could be held sufficient.

3.3 Optical flow with Local Binary Patterns

Representing the image texture as regional LBP histograms suites well to be used in combination with the traditional computer architecture. However, in order to make the LBPs better suitable for SIMD type computing, also other alternatives which would better utilize the available parallelism should be considered. Next, the LBPs are represented as bit-layers (see Figure 3.2), so that each LBP thresholding direction is considered separately without the decimal conversion and histogramming [31]. This makes it possible to apply the full descriptiveness of the LBP to the SIMD type processors. In this section, this approach is used to determine optical flow.

The approach has also the following advantages. First, if analog mismatch is present in the computation of the LBP bits, in the case of a wrong bit, the effect of the error can be restricted to that individual bit in the specific bit-layer (if the decimal conversion would be applied, the LBP might be assigned into a wrong bin). Second, by performing global array-wise shifting operations on all LLMs (i.e. to the digital Local Logical Memories within each of the array cells), the LBPs within a reference area can be compared efficiently to the sensor input which is not shifted.

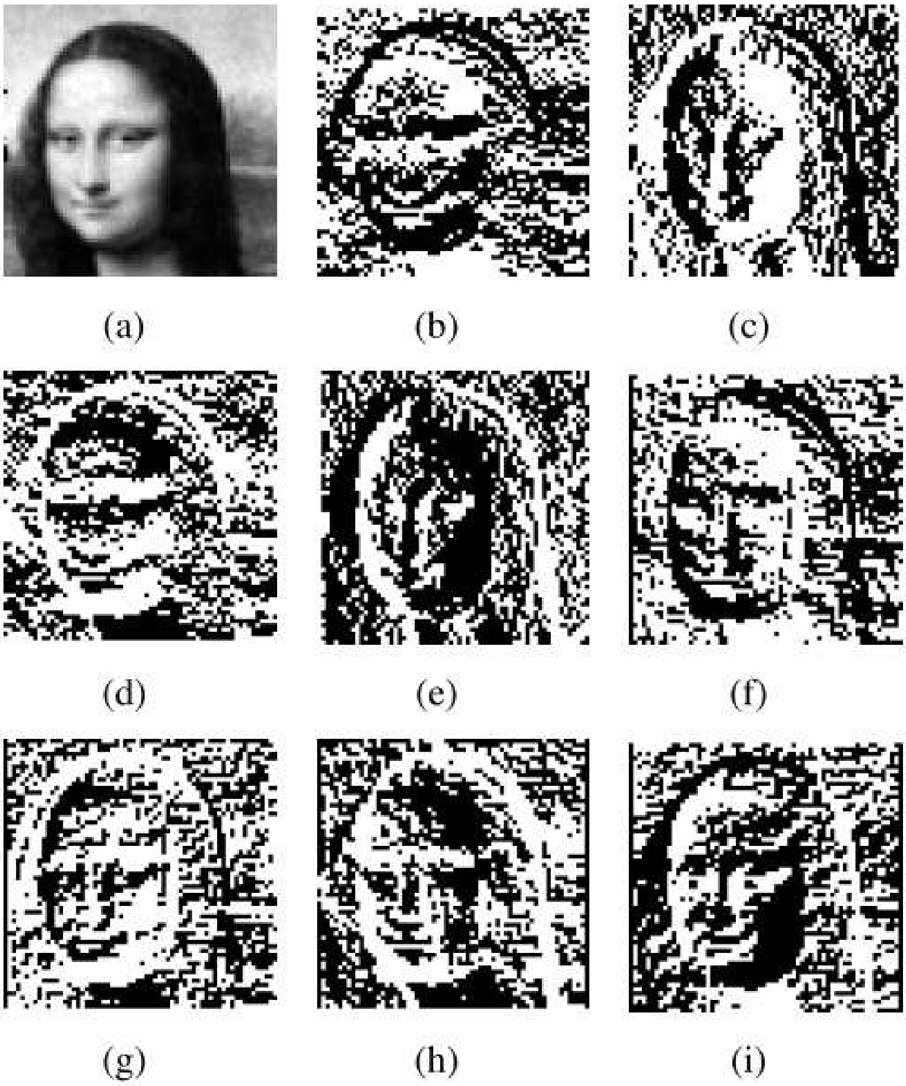


Figure 3.1: Original image (a) and directional LBP bits to (b) North, (c) East, (d) South, (e) West, (f) North-East, (g) South-East, (h) South-West and (i) North-West [34]. **(c) 2010 IEEE.**

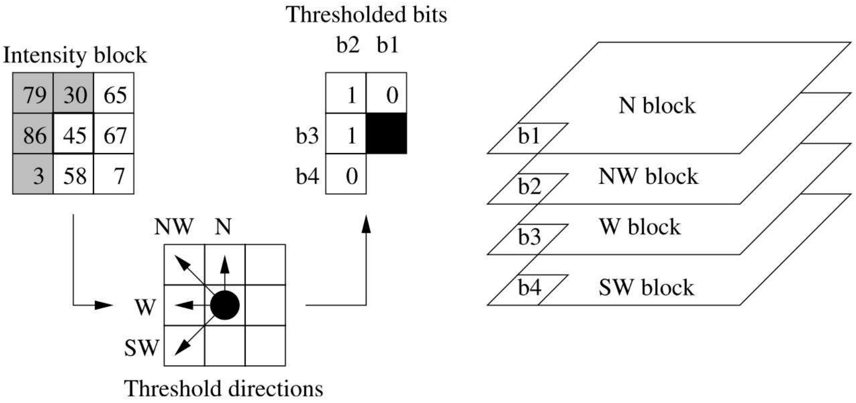


Figure 3.2: Bit-layer LBP representation [31]. **(c) 2009 IEEE.**

When the LBP bit directions are considered separately without the decimal conversion, it can be observed that there are redundancy in the informative content of the LBP bits. For instance, it is illustrated in Figure 3.2, that the comparison of the center bit with an intensity value of 45 into NW direction with an intensity value of 79 effectively provides the same information, than if comparing the NW direction bit with an intensity value of 79 as a center to its SE bit with an intensity value of 45. Therefore, only bits b1-b4 shown in Figure 3.2 are used for representing the local neighborhood. The method for comparing the bits within an image region into a reference region was simply to count the sum of matching bits within the regions by using a cell specific XNOR operation. This can be effectively implemented with CNN type computing with array-wise operation applied on each of the the LLMs, if a global sum operation is available [57]. Figure 3.3 shows an example of motion vectors extracted in Matlab between two frames of a video sequence by using the bit-layer method. Observe also, that the proposed novel optical flow method should be robust against environmental effects, such as temporal changes in illumination due to its non-parametric nature.

3.4 Application to the Analysis of Welding Process

The CNN approach was proposed for the closed-loop control of a laser welding process in [58]. The the keyhole dynamics (i.e. full penetration vs. partial penetration mode) in the welding area were analyzed on-line to control the laser power to optimize the welding quality. It was illustrated, that

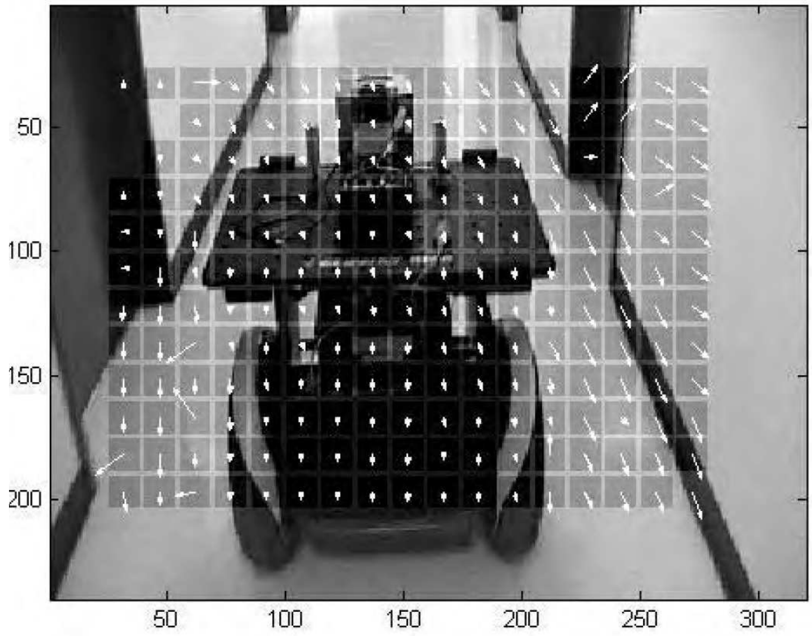


Figure 3.3: Optical flow from two succeeding frames using a search area of 25x25 pixels and a block size of 15x15 pixels [31]. Both the camera and the robot are moving to forward direction. **(c) 2009 IEEE.**

better weld quality could be achieved with this setup. Using thick section steel is an emerging research area, and to obtain a satisfying welding quality the parameters (such as welding speed, laser power, focal position and the type of shielding gas) need to be optimised. In this subsection, methods targeted to support the automated analysis of the welding process are considered.

The KOVA1 array processor was applied in the proposed application to extract edges in laser welding application when using a second order LBP neighborhood. The edges were extracted by comparing the analog current values in the main vertical and horizontal directions to an adjustable threshold. The result of the image capture was a B/W image sequence in welding of thick ($\geq$ 8mm) steel at approximately 3500 Fr/s frame-rate. Before starting the image capture, the camera was set to an adaptive integration mode [47], which means that the integration time of the local pixels was adjusted by the average of its neighboring pixels. This allowed the compression of the dynamic range of the images, so that the very bright and very low intensity regions in the imaging area could be captured. The laser power applied was 5kW, resulting into partial penetration with welding speeds of 2m/min and 3m/min [32].

3.4.1 Spatter segmentation

Laser welding

The first considered approach [32] for spatter segmentation was to extract block based motion vectors between the successive B/W images by using a binary pixel-wise sum of XNOR operation as a distance metric between two matching blocks. The motivation for this was, that when capturing the test sequences, the camera was set to a burst mode so that four successive frames were always extracted at a high speed and then there was an idle time in-between. The motion vectors were grouped according to their amplitude and direction, and connected component labeling was used to segment the spatters based on these properties. An example of the segmented spatters is shown in Figure 3.4.

Arc welding

In manual arc welding application, spatter segmentation was implemented by using Hough transform. The test sequence consisted of B/W images captured at high speed (approximately 1408 Fr/s). In this case, a continuous frame capture was used, where the whole image sequence without idles was obtained. Figure 3.5 shows spatters extracted off-line by using the Hough transform. Naturally, Hough transform can also be implemented on an embedded FPGA platform.

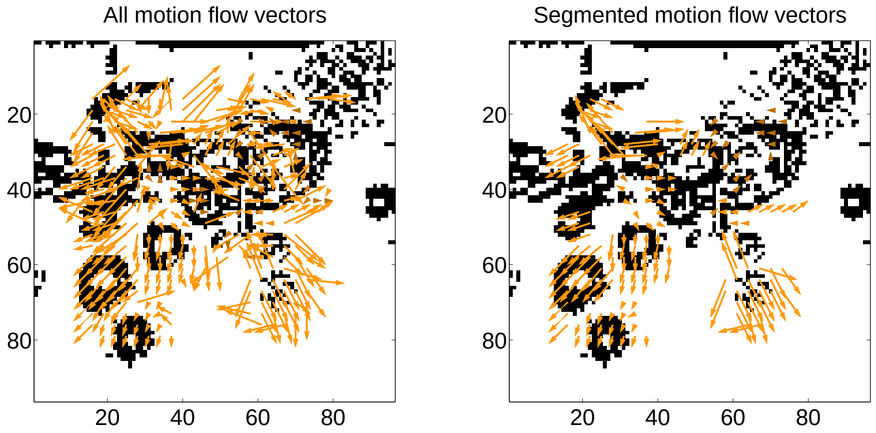


Figure 3.4: Illustration of the method for extracting spatters using optical flow of B/W images [32]. **Reproduced with kind permission from Springer Science and Business Media, 2013.**

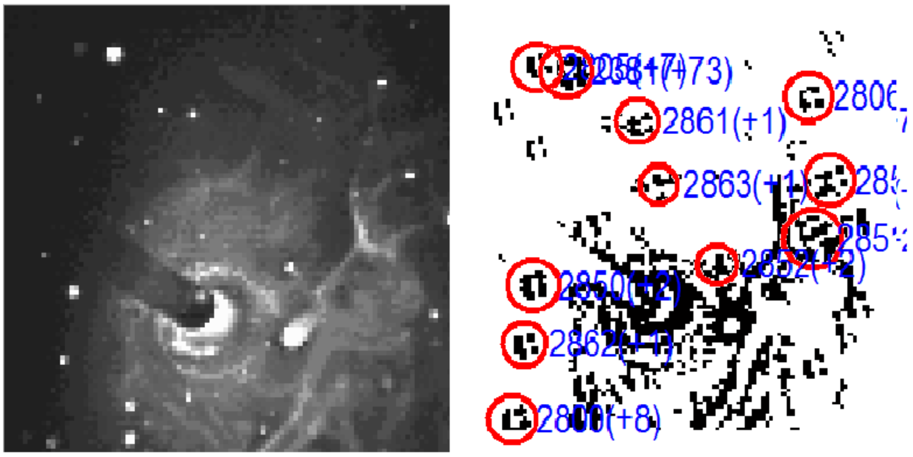


Figure 3.5: Adaptive integration image captured with KOVA1 processor from arc welding test and the corresponding B/W image in Matlab where the spatters have been annotated automatically [33].

3.4.2 Spatter tracking

A spatter tracking algorithm was implemented in Matlab to enable counting of individual spatters. The algorithm was tested in manual arc welding and laser welding scenarios. The implementation was based on looping through a list structure in Matlab, so that each spatter was assigned multiple properties, including their velocity and size [33]. After going through all frames, all the spatter instances were collected for further analysis.

Figure 3.6, shows the overall number of the spatters (the ground truth) and the number of extracted spatters in the arc welding sequence. The x-axis in this figure represents the total number of spatters in 100 frame intervals at frames 1-1500. It can be observed that the overall number of spatters (TP+FP) follows the ground truth. The error between the ground truth and the overall result can still become quite large in some intervals, but for an application such as calculating the statistics on the overall distribution of the spatters (e.g. the movement direction) across the whole welding sequence the current method could be sufficient. The advantage of the Hough transform based segmentation is that the distinction between the vapour plume and the actual spatters could be made more efficient in comparison with the B/W optical flow method. Connected component labeling based segmentation after a binary closing with a 3x3 template of all ones is shown as a reference method in Figure 3.6. The main limitation of the tracking algorithm was that it required the tracked spatter to be visible in every frame, which could not always be achieved in the B/W sequence due to sensitivity of the B/W threshold to disturbances.

3.5 Face Analysis with Focal Plane Processors

The objective of this section is to study, whether the LBP based face recognition proposed in [9] could also be implemented efficiently on massively parallel processors. The motivation for this lies in not only showing that focal plane processors could be capable to this, but also to investigate which kind of new capabilities would be required from the focal plane processors to achieve a state-of the-art performance for face recognition. Previously, in [59], a method for normalizing face images with the 64x64 cell ACE4k CNN has been implemented. That method was based on detecting edges from the input image and by calculating the axis of symmetry of the face images from the vertical and horizontal projections of the edge pixels.

3.5.1 Proposed Face Localization Procedure

The proposed face localization procedure for focal-plane processors is a modification from one presented earlier (in Chapter 2 sections 2.2.4-2.2.5). The

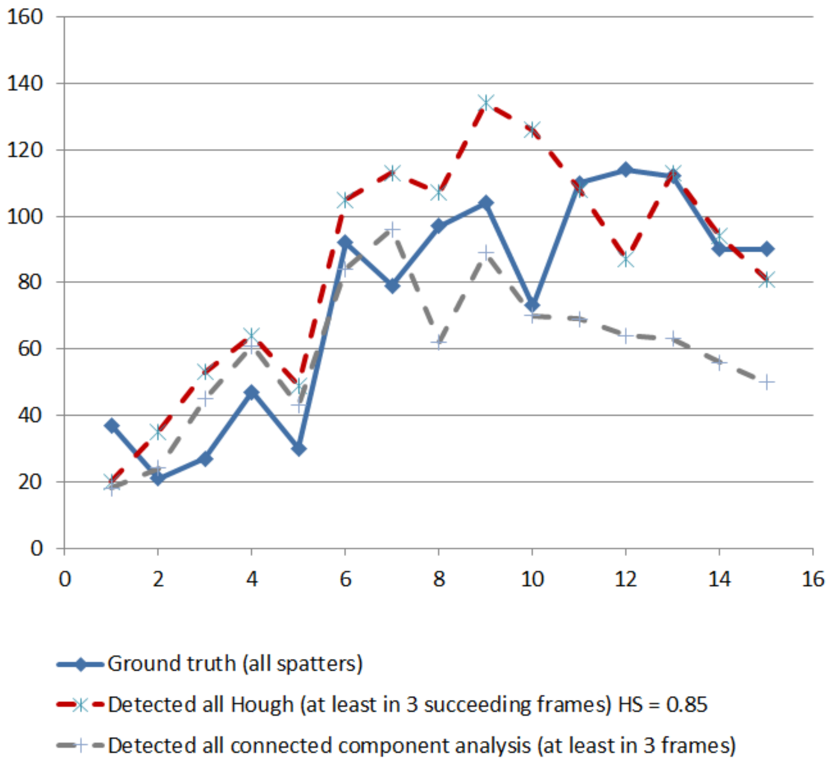


Figure 3.6: The manually annotated ground truth and the number of individual spatters detected by the segmentation and tracking algorithm in the manual arc welding test [33]. The x-axis represents 100 frame captions taken from the test sequence, starting from interval 1-100 (x value of 1) and ending to the interval 1401-1500 (x value of 15). The y-axis represents the total number of spatters in each interval.

difference is that instead of considering individual decimal LBPs, the bit-layer presentation of Figure 3.2 is used. First, a set of prototype faces is selected, in practice they were again collected from the normalized FERET set. Second, the binary images representing the bit-layers of the prototype faces are extracted. For example, if there are Z grayscale prototype faces (as in Section 2.2.4), then Z bit-layer images are extracted for each of the 4 bit-layers. The specific Z bit-layer images are then placed upon each other, and the sum of bits in integer form is calculated for each facial location across all prototype faces. The resulting prototype faces are shown in Figure 3.7. There is one prototype face for each of the bit directions (N, NW, W and SW). The binary bit-layer prototypes (bottom row of Figure 3.7) were then thresholded from the integer form ones.

The resized versions of the binary prototype faces in Figure 3.7 are then slided through the array, by shifting the LLM bits within the array cells and by calculating the number of matching bits by using a cell-specific XNOR operation. The resulting bits are then applied to the flexible read-out of unit currents from the specific cells. Figure 3.8 shows the results of face localization with bit-layers [34]. The actual bit layers were extracted from the MIPA4k processor after D/A conversion and ranked order statistics based LBP extraction. The effect of erroneous bits can be seen if the ideal match is compared to the one obtained using the non-ideal bits in Matlab (the matching percentage indicates the minimum localization threshold in 1% steps in accuracy). It can be observed, that also in the case of mismatch the face localization procedure gives correct results.

3.5.2 Proposed Face Recognition Algorithm

The face recognition algorithm, which was proposed in [60] and [29] extracts first the locations of each individual LBPs. For each LBP a separate binary image is generated, which represents in which facial locations (a 1 bit if LBP is in that location and 0 otherwise) the LBP exists. These binary images are denoted as occurrence maps. As a consequence, the locations of the individual LBPs are preserved. Lately, also a distance transform on specific occurrence maps [13] has been proposed, which support the assumption that the actual locations of the LBPs matter. The facial area is divided also in 3x2 regions as shown in Figure 3.9. This is done for two reasons. First, the regions enable weighting the face comparison procedure so that the facial areas which are more discriminative are given more importance [9]. Second, it allows using different orientation selective operations on each block.

The matching of two faces (one acquired and one stored to the memory) is done by searching for minimum distance between the acquired face and the stored face. First, each occurrence map of a face is dilated into each of the 8-neighboring directions a pre-determinated number of times (e.g. by 3 times).

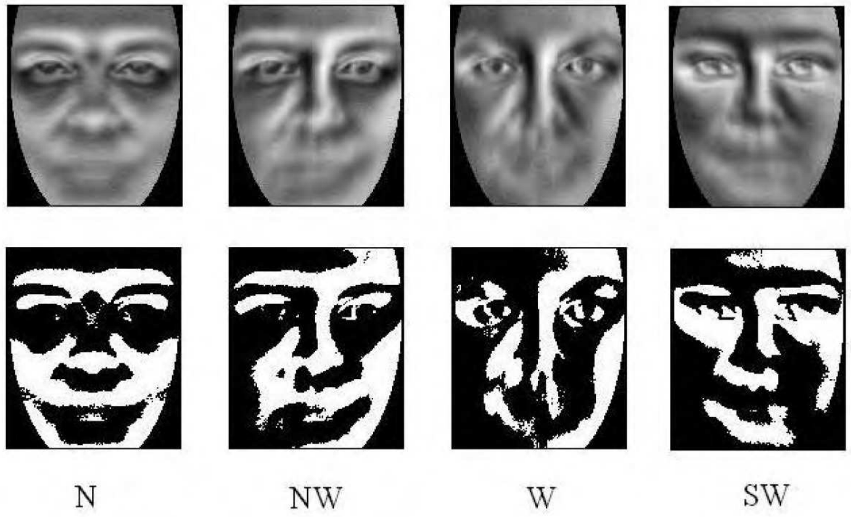
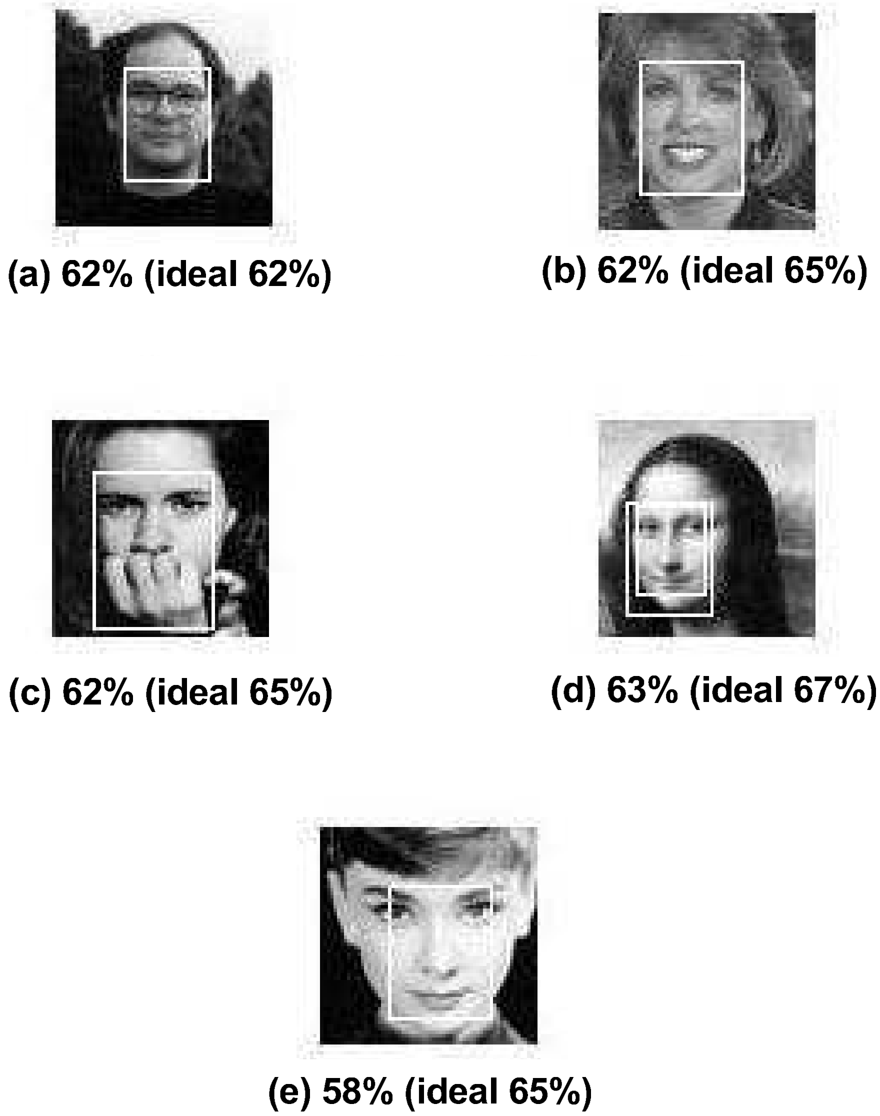


Figure 3.7: Average prototype faces to each of the four bit-layer directions and their thresholded versions [31]. The thresholded face prototypes were resized in sizes 20x20, 25x25, 30x30, 35x35, 40x40, 45x45 and 50x50 to detect faces in different spatial scales with the bits extracted from the MIPA4k processor. **(c) 2009 IEEE.**

Then a block specific dilation direction is selected for each of the 3x2 blocks separately so that the distance to the stored face is minimized (thus the identity of the stored face may or may not represent in this step the acquired face). Then, the sum of matching bits between the acquired occurrence maps and the occurrence maps of the stored faces is calculated. The recognition result is then the identity of the person stored to the memory, whose number of the matching bits from the acquired face is the maximum. If there are more than one faces stored to the memory, the acquired face is compared to each of them separately, and the identity of the stored face which maximizes the similarity is chosen as the recognition result.

Table 3.2 shows the rank 1 recognition rate of the occurrence map algorithm (LBP-AM) in the (8,2) LBP neighborhood in comparison with the original LBP algorithm in [9] with the FERET set [36]. The rank 1 recognition rate means that the acquired image always has to be correctly identified as the nearest image to the set of images stored to the memory. Thus, the larger the number of the stored faces, the more difficult it is to obtain a rank 1 match. The performance of the reference algorithms are standard implementations of the CSU evaluation system [61]. The proposed method could provide an average increase in the recognition accuracy over [9] of approximately 4% with weighting. The method proposed in the previous chap-



(a) 62% (ideal 62%)

(b) 62% (ideal 65%)

(c) 62% (ideal 65%)

(d) 63% (ideal 67%)

(e) 58% (ideal 65%)

Figure 3.8: The percentages above show the number of matching bits to prototype faces in order to localize the faces correctly with a minimum possible threshold using 1% accuracy [34]. The detections represent peaks in the 'faceness' map, where a perfect match (100%) fully corresponds to the bitlayer prototypes. The leftmost percentage represents the actual detection with bits extracted from the MIPA4k, and the percentage in the parenthesis (ideal) represents the detection with bits extracted ideally in Matlab. **(c) 2010 IEEE.**

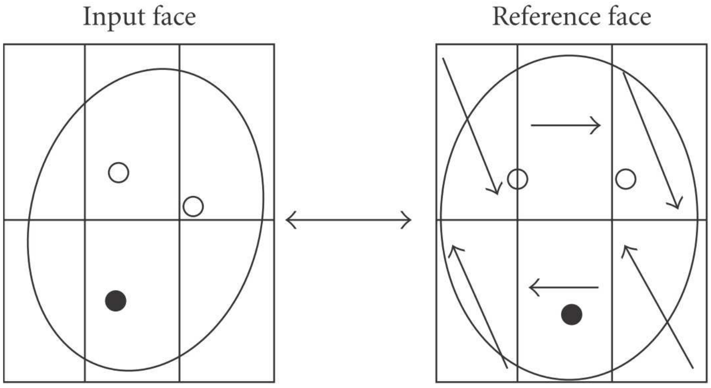


Figure 3.9: Adaptive face alignment of the occurrence map algorithm [29].

ter (section 2.2.1) for feature vector length reduction was also used in this experiment (applied to reduce the number of occurrence maps). Without weighting, interestingly, the algorithm outperformed its counterpart in [9] by 14%. A possible future direction for improving the algorithm could be to modify it to operate rather with bit-layers (as in the case of the proposed optical flow algorithm), than with the entire LBP code, which could provide further insensitivity to bit errors in case of non-idealities in the LBP extraction.

3.5.3 LBP processing hardware

The proposed face recognition algorithm is described in a framework of a massively parallel face recognition system in [29]. The transistor level designs for the LBP processing using a neighborhood comparison unit, and LBP matching unit were described in [54], [29]. In Table 3.3 the effect of mismatch to the LBP based face recognition is estimated using the framework proposed in [9]. Roughly a mismatch value of one percent could be tolerated without significant effect to the final recognition rate.

The proposed cell structure in [29] is described in a high abstraction level in Figure 3.10. The redundancy in the occurrence maps is encoded/decoded by memory code/decode unit, since most of the bits in the occurrence maps are zeros. Since 30 uniform and symmetrical LBPs were selected, one occurrence map location could be encoded into 5 bits. Unit current (see Figure 3.10) is needed for counting the number of matching bits by applying a flexible cumulative analog read-out from the local regions [57]. On an image database of 1000 faces the elapsed time for recognizing a face was estimated acceptable in [29]. A main disadvantage of the face representa-

Method	fb	fc	dup1	dup2	average
LBP-AM(8,2) Weighted	97%	86%	71%	67%	80%
LBP(8,2) [9] Weighted	97%	79%	66%	64%	76%
LBP(12,2) [26] Weighted	95%	85%	64%	66%	77%
LBP-AM(8,2) Non-weighted	95%	85%	69%	63%	78%
LBP(8,2) [9] Non-weighted	93%	51%	61%	50%	64%
PCA MahCosine	85%	65%	44%	22%	54%
Bayesian MAP	82%	37%	52%	32%	51%
EBGM optimal	90%	42%	46%	24%	51%
LDA ldasoft	73%	47%	45%	18%	46%

Table 3.2: Rank 1 face recognition accuracy of the occurrence map algorithm [29]. At the time of publication the results were among the best for LBP. However, recently advanced methods reaching significantly higher recognition rates have been reported (e.g. [14], [13])

Mismatch %	fb	fc	$dup1$	$dup2$	average
0	97%	80%	66%	64%	77%
1%	96%	73%	66%	65%	75%
2%	95%	69%	64%	58%	71%
3%	94%	63%	62%	57%	69%
5%	92%	55%	57%	49%	63%

Table 3.3: Mismatch effects for histogramming with weighting [29] according to recognition procedure in [9].

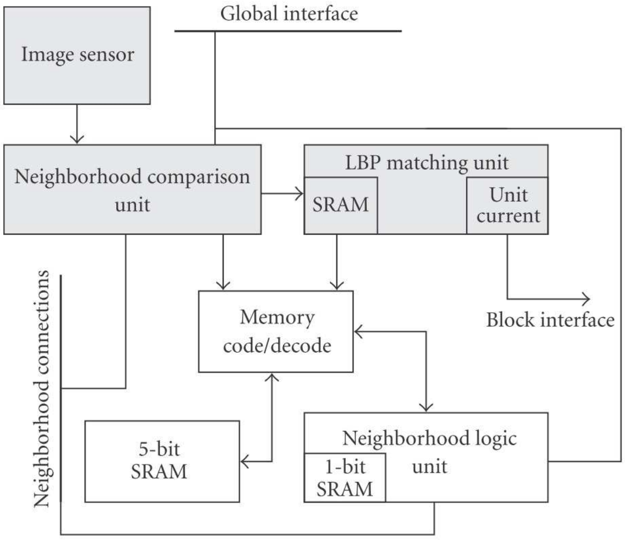


Figure 3.10: Dedicated LBP processing cell structure [29].

tioin in this case lies in its higher feature vector length (approximately 4 times longer than one in [9]) if using 130x150 sized images. This is because the exact directions of the LBPs are stored to the occurrence maps, instead of only considering the regional histograms. The implementations of the neighborhood comparison unit and LBP matching unit are shown in Figures 3.11 and 3.12. In Figure 3.13 a mismatch value of roughly one percent was achieved with larger transistors, which could be considered sufficient for face recognition.

3.6 Discussion

In this section, methods for extracting LBPs with SIMD hardware were proposed. The proposed methods could be expected to give a positive impact on better applicability of the focal-plane processors for e.g. high speed industrial process control and other embedded applications, which require compact size, high speed and high power efficiency. As a summary, the focal-plane approach would seem to provide a simple means for implementing a face recognition system or equivalent into a very compact size by taking advantage of the existing focal-plane processor hardware. One of the main challenges of this approach lies in the inaccuracy of the analog thresholding operation. Therefore, some of the proposed methods were further designed to cope with this limitation.

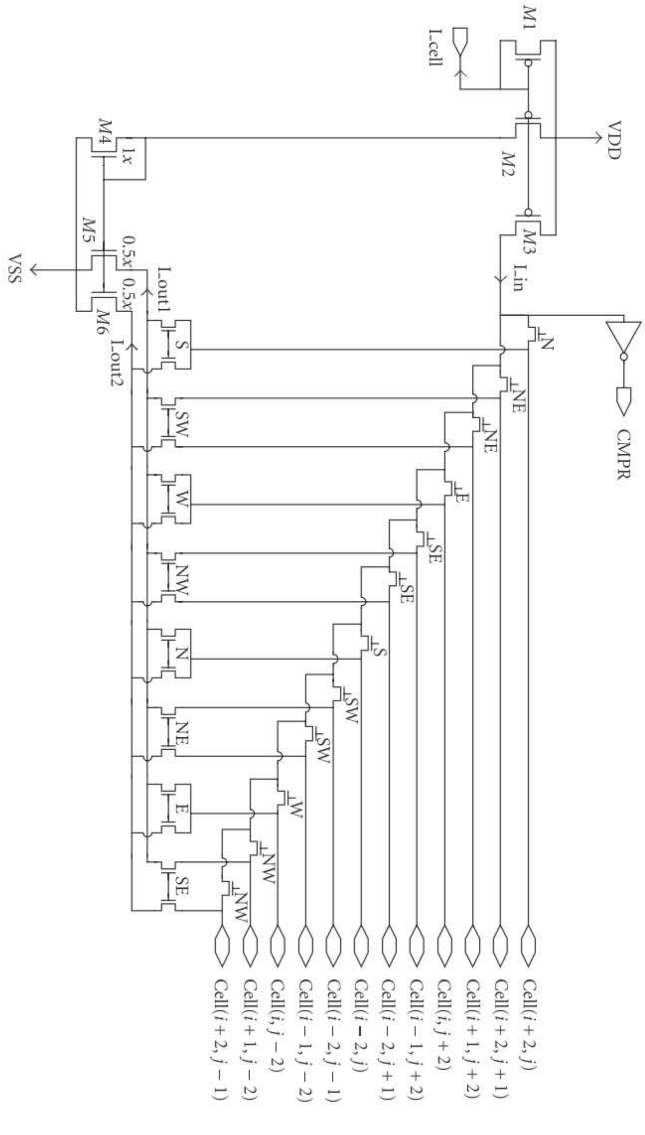


Figure 3.11: Neighborhood comparison unit (analog part) of the LBP matching hardware [29]. Transistors M1-M6 provide the current to the unit and the other transistors operate as switches. For instance, when the North bit of the LBP is extracted the switch N is turned on and the I_cell coming from e.g. in-cell photodiode is compared to the one coming two cells above. Simultaneously the circuit provides its photodiode current to two rows below, so that also the cell two rows below can extract its N bit. For directions NE, SE, SW and NW the currents are interpolated between two neighboring cells.

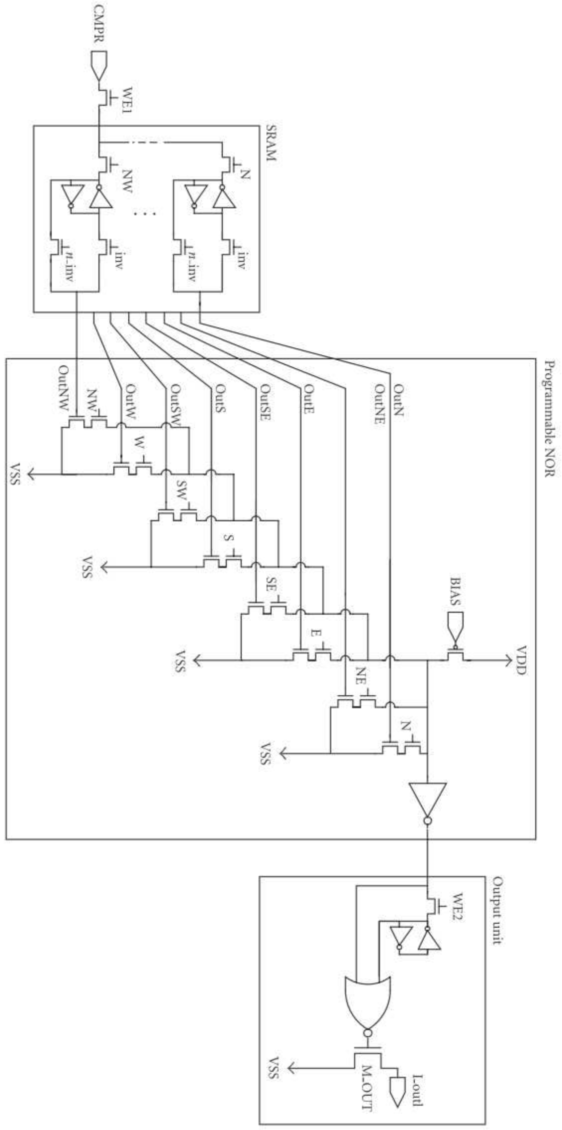


Figure 3.12: Digital part of the LBP matching hardware [29]. The LBP bits are stored to 8 SRAMs. The LBP matching is a two step process, first the LBP bits with a value of one are checked and then the bits with value of zero. In the first step, the control bits corresponding to LBP bits that are required to be one are set HI and an inverted LBP pattern is fed to the programmable NOR unit. The result fed to the output unit is zero if all LBP bits with value one match. Second, the LBP bits with value zero are checked by using the non-inverted LBP as input to the NOR block and by setting the control bits which are required to be zero to a HI value. If the results of the first and the second step after the NOR inverter are both 0, the two input NOR port at the output unit gives result one (match).

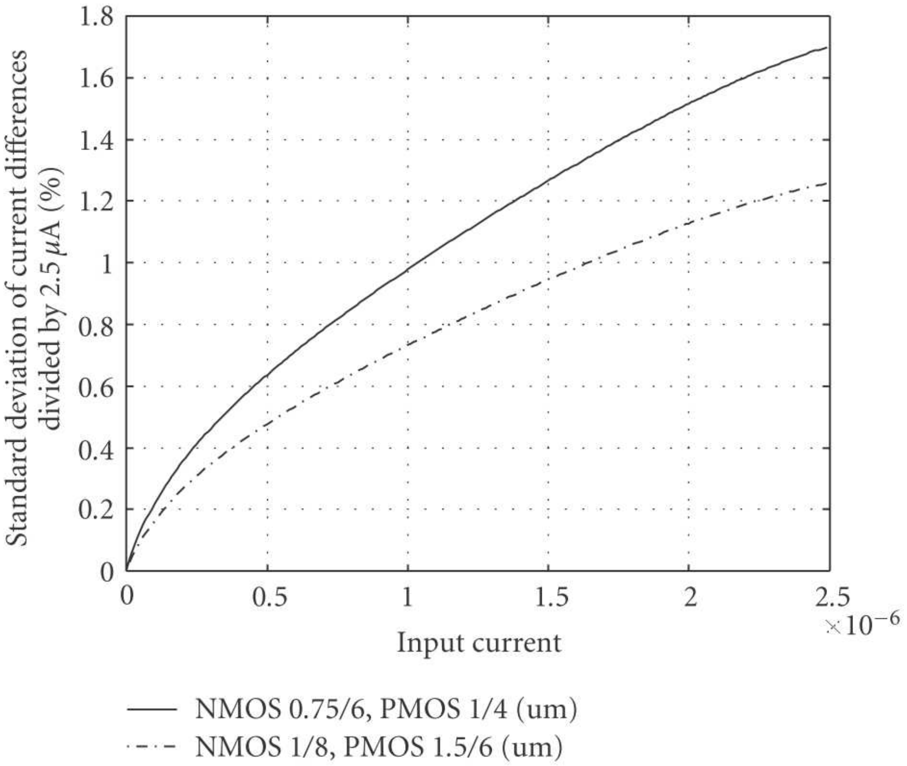


Figure 3.13: Effect of mismatch to the analog transistors of the neighborhood comparison unit [29]. A 0.13um CMOS process was used.

In face detection or localization, the lack of efficient image scaling functionality from the focal-plane processors caused the need for resizing the sliding window instead of scaling down the image size. This was circulated by changing (increasing) the sliding window size in this work, but in general, the scaling functionality - if implementable - would be very beneficial keeping the applicability of the focal-plane processors in mind.

The proposed optical flow method was designed for SIMD processors in particular and it could provide further robustness in applications where the environmental effects (e.g. temporal illumination changes) cause difficulties. Although the proposed optical flow method was not implemented on-chip, it can be expected to reach very high frame rates with the existing focal-plane processors, when a flexible global sum readout functionality is provided. The time needed for the calculation of the global sum could be expected to be the main limiting factor when considering very high frame operation rates (e.g. up to 1000fps) [62].

Chapter 4

Discussion and Conclusions

The thesis provided an introductory section to the LBP and the used concepts in the first chapter. In the second chapter, an a priori model for n-tuples, LBPs, and uniform LBPs was presented. Since the a priori occurrence probability of n-tuples with i.i.d. data is constant (if not considering interpolation), the approach could be seen as a combination of the proposed i.i.d. model, the LBP representation and ordinal n-tuple based image representation. The runs test for permutations which is a well known non-parametric test for randomness was linked to the uniform pattern selection of the LBP methodolody. In particular, the patterns with few runs were shown to contribute to the increase of the number of uniform patterns with natural images in comparison with the a priori model. When generated by monotonic n-tuples, the response of uniform LBPs is enhanced, which could indicate that the uniform patterns also correspond to the deterministic structures of the image.

High descriptor k uniform patterns were shown to be very rare according to the a priori model, while becoming quite common with natural image data. However, it should be noted that although the a priori probability of an individual high descriptor k window is small, the overall probability of all high k windows (e.g. uniform and non-uniform) is the same as for lower k. These patterns are typical to edges and monotonically changing intensity structures. Non-parametric image descriptors, in general, suffer from inherent sensitivity to noise [40], [22], and it seems that at least partly the selection of uniform patterns for the LBP histogram could provide additional robustness for the LBP representation to noise [28], [63].

Inspired by the applied feature analysis, a new non-parametric image transform, called non-parametric symmetry transform was proposed. The negation invariance property of the transform calculates the difference of the original pixels rank from the rank of the median. The term symmetry in this context was originally related to the local micro-symmetry of the

LBPs. This approach showed better performance than the rank transform in a face recognition experiment and reduced also the dynamic range of the output values. The usage of the previously proposed rank transform, which is closely related to the proposed symmetry transform has been further studied in [40]. In [40] an additional convergence property for the rank transform was proposed and the sensitivity of the rank transform to noise was explained. This could also be one factor behind the better performance of the symmetry transform in the face recognition experiment. However, an interesting observation based on the recognition results between the symmetry transform and the rank transform was that the overall rate was increased by roughly an equal amount between each test set, which could indicate that noise robustness is not the only factor behind the higher recognition rates (for example, the symmetry transform could include tolerance to rotation, facial mirror symmetry or some other characteristics).

In the third chapter, the Local Binary Pattern approach was proposed for massively parallel focal-plane processor arrays. Focal-plane processors can potentially provide a very high image analysis throughput due to a high bandwidth between the photodiodes and the local processing elements. In the authors knowledge, the LBPs have not been applied to this domain before. The research questions posed in the beginning of the thesis - which kind of features should be incorporated to the massively parallel arrays to promote their applicability to real world problems - was hence confronted. In the proposed framework, methods for extracting the LBPs, an optical flow algorithm based on a modified LBP representation and methods for localizing, and recognizing faces with massively parallel processor arrays were described. A coarse estimation on the performance constraints of a dedicated system performing face recognition was also performed. The proposed face recognition algorithm also utilized one of the previous experimental result in the earlier work [26], in order to reduce the final feature vector length. Thus, the proposed methods could be expected to have an impact on wider applicability of focal-plane processors to practical applications. Also, the research question - which kind of applications would benefit the most from focal plane processors was confronted in the context of analysis of laser welding process with the KOVA1 camera. In the future, methods towards on-line control of laser welding process of thick steel are targeted.

Appendix A

Appendix of paper contents

A.1 Reducing the Feature Vector Length in Local Binary Pattern based Face Recognition

The local binary pattern method was proposed for face recognition in [8]. The length of the feature vectors was denoted in that paper as one drawback of the method. This paper presents a method for reducing the length of the feature vectors in LBP based face recognition. The idea is to divide the uniform LBPs into classes by the number of their zero and one bits. The minimum between the total number of one and zero bits of a LBP was denoted as the symmetry level of the pattern. It was empirically observed with the standard FERET set, that the classes with higher symmetry performed better than the patterns with low level of symmetry. Also qualitative analysis showed that the facial regions could be efficiently represented by the high level of symmetry patterns.

The analysis was taken advantage of in reducing the final length of the LBP feature vectors. The idea was to use a bin of their own for higher level of symmetry uniform patterns and to combine the bins of each rotation invariant category of the lower level of symmetry patterns. As a result the overall number of bins in an individual uniform LBP block histogram could be reduced in half with only 0.3% negative effect to the final recognition rate.

A.2 A Shape-preserving Non-parametric Symmetry Transform

This paper extends the idea of the previous article into a new non-parametric transform beyond the LBP neighborhood. It is shown, that the proposed new non-parametric image transform can be seen as a modification of the rank transform, while showing an interesting additional property of negation invariance. The performance of the proposed transform was compared to the rank transform in face recognition, and a major improvement in the recognition accuracy was obtained with the standard FERET database. It also performed quite well in comparison with original LBP, although the performance of the weighted LBP was higher.

An another application scenario proposed was searching of salient points in images. In this context, saliency relates to high symmetry level of the local neighborhood, i.e. the center pixels are near to the local median of the window. In empirical tests the high symmetry level locations showed also tolerance to image transformations such as local rotation.

A.3 A Statistical Approach for Characterising Local Binary Patterns

The paper presents an a priori model for the LBPs, where the image pixels are seen as instances of order statistic variables. The analysis in the paper is based on the observation, that when the number of ones (or zeros) in the pattern increases, the overall a priori probability of the class of patterns belonging onto each category remain constant. As a consequence, the number of pattern combinations within each class determines the overall probability of each pattern. The distribution of the LBPs thus follow roughly the binomial distribution.

A more general interpretation is, that the symmetry level concept used in articles 1 and 2 also characterizes the a priori probabilities of the non-parametric local windows (e.g. in Census transform). With i.i.d. data the occurrence probability of a certain local window which possess a high level of symmetry is significantly lower, than the lower level of symmetry locations. Also, an illustration of the LBP distribution on a natural test image is shown. In Equation 3, the more compact form in references [38] and [28] can be obtained by rewriting the $\binom{M}{k}$ as $M!/(k! * (M-k)!)$. There is also a typo in the Proof of Theorem I after the sentence "The recursion follows:" the next $J(M,k)$=$k/(m-k+1) * J(M,k+1)$ should be $J(M,k)$=$k/(m-k+1) * J(M,k-1)$.

A.4 A Massively Parallel Face Recognition System

This paper presents a framework for implementing an embedded single chip LBP based face recognition system, including the implementation and analysis of its main building blocks. The paper covers and extends the contents of the previous conference publications [55], [54] and [60], which are therefore not included to this reprint (in order to avoid repetition). Two alternatives for extracting the LBPs with focal-plane processors were proposed. The first was to use a standard CNN type comparison templates and to store the LBP bits into the LLMs of a CNN. The other method was to design dedicated citcuitry for the extraction and matching of the LBP bits. These included a neighborhood comparison unit, which produce the LBP bits to the LLMs of the array, and a neighborhood matching unit which was used to determine in which locations a certain LBP exists in the array.

The trade-offs in implementing the proposed system were also discussed. The face recognition algorithm was a modified version of the one presented in [8], taking into consideration the actual locations of each LBP on the focal-plane array. Thus, a histogram representation of the LBP was replaced with spatial representation, where the location of an individual LBP could be preserved. If not considering mismatch effects, the proposed algorithm performed slightly better than the one in [8]. The earlier method presented for reducing the feature vector length was also taken advantage in this algorithm. In the example of Equation 7 the interpolation weights are only approximative. An example of exact weights using bilinear interpolation can be found e.g. in [35] in the case of (8,1) neighborhood.

A.5 On the Spatial Distribution of Local Non-parametric Facial Shape Descriptors

This paper presents a method to form an average image of local non-parametric descriptors, which can be used for the analysis of the distribution of each individual LBP across the area of human faces. The idea is to normalize the density of each individual LBP histogram specific to each pixel position in the facial area accross a large database of cropped face images. It is illustrated, that the density of the low level of symmetry uniform patterns is most typical to the eye areas, while the higher level of symmetry patterns typically occur in areas of the nose, cheek and eye surroundings. Thus, the concentration of LBPs in various orientations change accross the area of human face in average.

The proposed method is applied to face detection, by implementing a face detector which searches accross different spatial scales by accumulating the evidence of each LBP in its expected location. The detector can achieve a moderate detection accuracy (i.e. $>$ 80% with 200 false positives) using the standard full CMU+MIT database. Considering the simplicity of the approach, the detector performs quite well, although not achieving a state-of-the-art detection performance (e.g. compared to machine learning approaches such as in [1]).

A.6 Regional Image Correspondence Matching Method for SIMD Processing

The paper presents a new kind of LBP representation, which is applied for optical flow analysis and face localization. The proposed methods are designed for SIMD type focal-plane processors in particular. The new LBP representation is based on comparing the center pixel not to its all eight neighbors, but only to four neighbors next to each other. The redundancy of the LBP representation is thus reduced, because adjacent pixel pairs are not compared twice to each other, while the center pixel is also taken into consideration.

The optical flow method is based on shifting the bits inside the array cells with respect to reference bits which are not shifted. The method requires means for global readout of the matching bits within the array. The face localization method is based on a similar principle as the previous publication, except of the different LBP representation. The prototype faces are produced in integer form by assigning the LBP bit images in top on each other and by summing the bits. These images are then again converted from integer form through a threshold operation to a binary form and then slided through the array. The performance of the face localization method is estimated with the full CMU+MIT set, and it is concluded that it could perform quite reliably in the proposed application with limited image size. It should be noted, that the optical flow vectors for SAD method of Fig. 3 in the publication are over pessimistic due to an unfortunate coding bug.

A.7 Extracting Local Binary Patterns with MIPA4k Vision Processor

The paper presents a method for extracting LBPs with MIPA4k vision processor. The method is based on using the ranked-order statistics unit of the MIPA4k for deriving the LBPs. The selected method was considered to be the most practical with the actual MIPA4k prototyping system. The number of correct LBP bits in comparison with LBPs affected by MIPA4k mismatch is also analyzed. The LBP bits read from MIPA4k were used in Matlab to analyze whether the face localization procedure presented in the previous publication also work in the presence of mismatch. It is concluded that with applied test images the face localization is robust to circuit non-idealities.

A.8 Towards Understanding the Formation of Uniform Local Binary Patterns

The purpose of this paper is to increase the understanding on the Local Binary Patterns by seeing them as combinations of permutations. The occurrence probability of each of the permutations with i.i.d. data is constant, so that a permutation space can be defined between the intensity and LBP representations. Mapping operators from the intensity space to the permutation space, and from the permutation space to the LBP space are presented. The main contribution of the paper to the analysis of uniform patterns is to describe how circularly monotonic permutations tend to further increase the occurrence frequency of uniform patterns with natural image data, and that these permutations are also among the most common individual permutations with natural image data.

The paper extends the previous technical report of the author, and can provide new perspectives to the relation between LBPs and n-tuples. Note that the test of the number of ties among uniform and non-uniform patterns in Fig. 20 is affected by the larger relative share of uniform patterns in comparison with non-uniform patterns. If the normalization would be done against all patterns in both cases, the number of ties within non-uniform patterns would naturally be smaller.

A.9 Characterizing Spatters in Laser Welding of Thick Steel Using Motion Flow Analysis

The paper presents an approach to characterize spatters in laser welding. The methods are based on the analysis images captured at high speed with the KOVA1 focal-plane processor. The B/W images were captured at multi kFr/s rate in bursts, and the off-line analysis step in Matlab consisted of determining the motion vectors between image pairs. The approach was based on determining the motion vectors in a fine grid of predetermined size, and to reduce the representation of the motion vectors in quantized amplitude (length) and direction (angle) components. A connected component analysis step was then applied to segment the spatters according to their characteristics.

Depending on the search window size and the block size, the proposed method could also be implemented in FPGA for high speed analysis. The purpose of the paper is to represent first steps toward a system allowing automated analysis of the characteristics of the welding process of thick steel.

A challenge related to the implemented approach is the distinction between the actual spatter objects and the plume, which could cause false positives. It was proposed in the paper, that future work could contain tracking the detected spatters in a longer time-scale.

A.10 Spatter Tracking in Laser- and Manual Arc Welding with Sensor-level Pre-processing

This paper extends the previous paper considering the characterization of spatters in laser welding by introducing a spatter tracking algorithm to enable the counting of individual spatters in laser and arc welding. As in the previous paper, the KOVA1 processor was again used to extract the edges of the spatters and to provide a robust image representation by adjusting the pixel's integration time by the average of the neighboring pixels. Hough transform and connected component analysis were considered as alternatives to perform the spatter segmentation as an off-line process in Matlab. The principle of the tracking algorithm is to compare the linear estimate of the next spatter location based on frames t-2 and t-1 to each of the new spatters in frame t, enabling the update of the tracked spatters as a list structure. All of the spatters during the tracking are afterwards collected to a list for further analysis. This approach can give information on the directions and velocities of the spatters within the welding image sequences.

The methods were tested against a manually generated ground truth in two welding sequences, the first of which was manual arc welding and the second laser welding. It was concluded, that the algorithm (TP+FP rate) could roughly follow the actual spatter behaviour within the test sequences. In the future, it is planned that some of the methods proposed could be implemented on an FPGA by in-camera processing.

Bibliography

[1] P. Viola, M. Jones, *Rapid object detection using a boosted cascade of simple features*, Proceedings of the IEEE conference of Computer Vision and Pattern Recognition (CVPR), pp. 511-518, vol. 1, (2001)

[2] G. Stockman, L. G. Shapiro, *Computer Vision*, Prentice Hall, (2001)

[3] E. N. Malasama, E. G. M. Petrakisa, M. Zervakisa, et. al, *A survey on industrial vision systems, applications and tools*, Image and Vision Computing, 21(2), pp. 171-188, (2003)

[4] S. Theodoridis, K. Koutroumbas, *Pattern recognition*, Second edition, Elsevier, (2003)

[5] T. Ojala, M. Pietikäinen, D. Harwood, *A comparative study of texture measures with classification based on featured distribution*, Pattern Recognition, 29(1), pp. 51-59, (1996)

[6] M. Pietikäinen, A. Hadid, G. Zhao, et. al, *Computer vision using Local Binary Patterns*, Springer, (2011)

[7] T. Ojala, M. Pietikäinen, T. Mäenpää, *Multiresolution gray-scale and rotation invariant texture classification with local binary patterns*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 24(7), pp. 971-987, (2002)

[8] T. Ahonen, A. Hadid, M. Pietikäinen, *Face recognition with local binary patterns*, Proceedings of the European Conference on Computer Vision, ECCV, pp. 469-481, (2004)

[9] T. Ahonen, A. Hadid, M. Pietikäinen, *Face description with Local Binary Patterns: Application to face recognition*, IEEE Transactions on Pattern Analysis and Machine Intelligence 28(12), 2037-2041, (2006)

[10] T. Mäenpää, The local binary pattern approach to texture analysis extensions and applications, Dissertation, Infotech Oulu, University of Oulu, (2003)

[11] T. Mäenpää, T. Ojala, M. Pietikäinen, *Robust texture classification by subsets of Local Binary Patterns*, in Proceedings of the 15th International Conference on Pattern Recognition, pp. 947 - 950, (2000)

[12] M. Turtinen, O. Silvén, M. Pietikäinen, et. al, *Characterisation of paper*, WO2004/023398, Honeywell, (2004)

[13] X. Tan, B. Triggs, *Enhanced Local Texture Feature Sets for Face Recognition Under Difficult Lighting Conditions* , IEEE Transactions on image processing, 19(6), pp. 1635-1650, (2010)

[14] W. Zhang, S. Shan, W. Gao, et. al, *Local Gabor binary pattern histogram sequence (LGBPHS): a novel non-statistical model for face representation and recognition*, in IEEE International Conference on Computer Vision, ICCV, Vol. 1, pp. 786 - 791, (2005)

[15] A. Ernst, B. Froba, *Face detection with the modified census transform*, in IEEE International Conference on automatic Face and Gesture recognition, pp. 91-96, (2004)

[16] M. Heikkilä, M. Pietikäinen, C. Schmid, *Description of interest regions with local binary patterns*, Pattern Recognition , 42(3), pp. 425-436, (2009)

[17] V. Ojansivu, J. Heikkilä, *Blur Insensitive Texture Classification Using Local Phase Quantization*, International Conference on Image and Signal Processing, LNCS 5099, pp. 236-243 (2008)

[18] R. Zabih, J. Woodfill, *Non-parametric Local Transforms for Computing Visual Correspondence*, in European Conference on Computer Vision (ECCV), pp. 151-158, (1994)

[19] D.N. Bhat, S. K. Nayar, *Ordinal measures for visual correspondence*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 20(4), pp. 415 - 423, (1998)

[20] Z. Wang, B. Fan, F. Wu, *Local Intensity Order Pattern for feature description*, in IEEE International Conference on Computer Vision (ICCV), pp. 603-610, (2011)

[21] B. Fan, F. Wu, Z. Hu, *Rotationally Invariant Descriptors Using Intensity Order Pooling*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 34(10), pp. 2031-2045 (2012)

[22] L. Hepplewhite, T.J. Stonham, *Texture Classification Using N-Tuple Pattern Recognition*, in International Conference on Pattern Recognition (ICPR), Vol. 4, pp. 159-163, (1996)

[23] M. Partio, B. Cramariuc, M. Gabbouj, *An Ordinal Co-occurrence Matrix Framework for Texture Retrieval*, EURASIP Journal on Image and Video Processing, Article ID 17358, (2007)

[24] M. Nilsson, M. Dahl, I. Claesson, *The successive mean quantization transform*, in IEEE International Conference on Acoustics, Speech, and Signal Processing, (ICASSP), Vol. 4, pp. 429-432, (2005)

[25] O. Lahdenoja, *A Statistical approach for Characterizing Local Binary Patterns*, TUCS Technical report, no. 795, (2006)

[26] O. Lahdenoja, M. Laiho, A. Paasio, *Reducing the feature vector length in Local Binary Pattern based face recognition*, in IEEE International Conference on Image Processing (ICIP), pp. 914-917 (2005)

[27] O. Lahdenoja, E. Alhoniemi, M. Laiho, A. Paasio, *A Shape-preserving non-parametric symmetry transform*, 18th International Conference on Pattern Recognition, pp. 373-377, (2006)

[28] O. Lahdenoja, J. Poikonen, M. Laiho, *Towards Understanding the Formation of Uniform Local Binary Patterns*, ISRN Machine Vision, Article ID 429347, (2013)

[29] O. Lahdenoja, M. Laiho, J. Maunu, A. Paasio, *A Massively parallel face recognition system*, EURASIP Journal on Embedded Systems - Special Issue on Embedded Vision Systems, Article 72316, (2007)

[30] O. Lahdenoja, M. Laiho, A. Paasio, *On the Spatial Distribution of Local Non-parametric Facial Shape Descriptors*, 16th Scandinavian Conference on Image Analysis, SCIA, pp. 351-358 (2009)

[31] O. Lahdenoja, M. Laiho, *Regional image correspondence matching method for SIMD processing*, 19th European Conference on Circuit Theory and Design, pp. 802-805, (2009)

[32] O. Lahdenoja, T. Säntti, J. Poikonen, M. Laiho, A. Paasio, *Characterizing Spatters in Laser Welding of Thick Steel Using Motion Flow Analysis*, 18th Scandinavian Conference on Image Analysis (SCIA), pp. 675-686, (2013)

[33] O. Lahdenoja, T. Säntti, M. Laiho, J. Poikonen, *Spatter Tracking in Laser- and Manual Arc Welding with Sensor-level Pre-processing*, 22nd International Conference on Computer Graphics, Visualization and Computer Vision (WSCG), (2014)

[34] O. Lahdenoja, Jonne Poikonen, Mika Laiho, *Extracting Local Binary Patterns with MIPA4k Vision Processor*, IEEE International Workshop on Cellular Nanoscale Networks and Applications, (2010)

[35] T. Ahonen, M. Pietikäinen, *Image description using joint distribution of filter bank responses*, Pattern recognition letters, 30(4), pp. 368-376, (2009)

[36] P.J. Phillips, H. Wechsler, J. Huang, P. Rauss, *The FERET Database and Evaluation Procedure for Face Recognition Algorithms*, Image and Vision Computing, 16, 295-306, (1998)

[37] J. D. Gibbons, *Nonparametric Statistical Inference*,McGraw-Hill, (1975)

[38] F. Bianconi, A. Fernández, *On the occurrence probability of Local Binary Patterns: A Theoretical Study*, Journal of Mathematical Imaging and Vision, 40(3), pp. 259-268 (2011)

[39] D. G. Lowe, *Distinctive image features from scale-invariant keypoints*, International Journal of Computer Vision, 60(2), pp. 91-110 (2004)

[40] J. Mukherjee, *Local rank transform: Properties and applications.* Pattern Recognition Letters, 32(7), pp. 1001-1008 (2011)

[41] Zarándy Á. (ed.): Focal-Plane Sensor-Processor Chips, Springer (2011)

[42] L. O. Chua, L. Yang, *Cellular Neural Networks: Theory*, IEEE Transactions on Circuits and Systems, 35(10), pp. 1257-1272, (1988)

[43] A. Rodríguez-Vázquez, G. Liñán-Cembrano, L. Carranza, et. al, *ACE16k: The Third Generation of Mixed-Signal SIMD-CNN ACE Chips Toward VSoCs*, IEEE Transactions on Circuits and Systems-I: Regular Papers, 51(5), pp. 851-863, (2004)

[44] A. Rodríguez-Vázquez, R. Domínguez-Castro, F. Jiménez-Garrido, et. al, *The Eye-RIS CMOS Vision System*, Analog Circuit Design : Sensors : Actuators and Power Drives : Integrated Power Amplifiers from Wireline to RF: Very High Frequency Front Ends (2), Springer (2008)

[45] A. Lopich, P. Dudek, *ASPA: Focal Plane Digital Processor Array with Asynchronous Processing Capabilities*, IEEE International Symposium on Circuits and Systems, ISCAS, pp. 1592-1595, (2008)

[46] S. J. Carey, D. R. W. Barr, B. Wang, A. Lopich, P. Dudek, *Locating High Speed Multiple Objects using a SCAMP-5 Vision-Chip*, International Workshop on Cellular Nanoscale Networks and Their Applications (CNNA), (2012)

[47] M. Laiho, J. Poikonen, A. Paasio, *MIPA4k: Mixed-Mode Cellular Processor Array*, Zarandy A. (ed.): Focal-Plane Sensor-Processor Chips, Springer (2011)

[48] T. Roska, L. O. Chua, *The CNN Universal Machine: An Analogic Array Computer*, IEEE Transactions on Circuits and Systems II, Analog and Digital Signal Processing, 40(3), 163-174, (1993)

[49] T. Roska, Á. Zarándy, S. Zöld, et. al, *The Computational Infrastructure of Analogic CNN ComputingPart I: The CNN-UM Chip Prototyping System*, IEEE Transactions on Circuits and Systems I: Fundamenta Theory and Applications, 46(1), pp. 261-268, (1999)

[50] P. Dudek, P. J. Hicks, *A General-Purpose Processor-per-Pixel Analog SIMD Vision Chip*, IEEE Transactions on Circuits and Systems I, Regular Papers, 52(1), pp. 13-20, (2005)

[51] P. Dudek, D. R. W. Barr, A. Lopich, S. J. Carey, *Demonstration of real-time image processing on the SCAMP-3 vision system*, International Workshop on Cellular Neural Networks and Their Applications, (2006)

[52] A. Paasio, A. Kananen, K. Halonen, et. al, *A QCIF resolution binary I/O CNN-UM chip*, Journal of VLSI Signal Processing Systems for Signal, Image and Video Technology, (23)2, pp. 281-290, (1999)

[53] www.kovilta.fi

[54] M. Laiho, O. Lahdenoja, and A. Paasio, *Dedicated hardware for parallel extraction of local binary pattern feature vectors*, Proceedings of the 9th IEEE International Workshop on Cellular Neural Networks and Their Applications (CNNA), pp. 27-30, (2005)

[55] O. Lahdenoja, M. Laiho, and A. Paasio, *Local binary pattern feature vector extraction with CNN*, IEEE International Workshop on Cellular Neural Networks and Their Applications (CNNA), pp. 202-205, (2005)

[56] M. Bordallo López, A. Nieto, J. Boutellier, et. al, *Evaluation of real-time LBP computing in multiple architectures*, Journal of Real-Time Image Processing, 2014 (in press)

[57] P. Dudek, *A Flexible Global Readout Architecture for an Analogue SIMD Vision Chip*, Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS), vol. 3, pp. 782-785, (2003)

[58] L. Nicolosi, R. Tetzlaff, F. Abt, et. al, *New CNN based algorithms for the full penetration hole extraction in laser welding processes: Experimental results*, IJCNN, pp. 2256-2263 (2009)

[59] Z. Szlávik, T. Szirányi, *Face identification using CNN-UM*, Proceedings of the European Conference on Circuit Theory and Design, (ECCTD), vol. 2, pp. 81-85, (2003)

[60] O. Lahdenoja, J. Maunu, M. Laiho, and A. Paasio, *A massively parallel algorithm for Local Binary Pattern based face recognition*, IEEE International Symposium on Circuits and Systems (ISCAS), pp. 3730-3733, (2006)

[61] D.S. Bolme, J.R. Beveridge, M. Texeira, B.A. Draper, *The CSU Face Identification Evaluation System: Its Purpose, Features and Structure*, Third International Conference on Computer Vision Systems, 304-311, (2003)

[62] J. Yang, C. Shi, L. Liu, N. Wu, *Heterogeneous vision chip and LBP-based algorithm for high-speed tracking*, IEEE Electronics letters, 50(6), pp. 438-439, (2014)

[63] J. Chen, V. Kellokumpu, G. Zhao, et. al, *RLBP:Robust Local Binary Pattern*, Proceedings of the British Machine Vision Conference (BMVC), (2013)

ARTICLE 1

O. Lahdenoja, M. Laiho, A. Paasio, Reducing the feature vector length in Local Binary Pattern based face recognition, IEEE International Conference on Image Processing, pp. 914-917, 2005.

ARTICLE 2

O. Lahdenoja, E. Alhoniemi, M. Laiho, A. Paasio, A Shape-preserving non-parametric symmetry transform, 18th International Conference on Pattern Recognition, pp. 373-377, 2006.

ARTICLE 3

O. Lahdenoja, A Statistical approach for characterising Local Binary Patterns, TUCS Technical Report, No. 795, 2006.

Turku Centre for Computer Science.

A Statistical Approach for Characterising Local Binary Patterns

Olli Lahdenoja

University of Turku, Department of Information Technology
Joukahaisenkatu 3-5, FIN-20014 Turku, Finland
olanla@utu.fi

TUCS Technical Report

No 795, October 2006

Abstract

Local Binary Patterns (LBPs) are a class of local non-parametric image descriptors, which have been applied to several applications such as texture analysis, motion detection and biometric authentication with a high discriminative performance. In many applications, such as in face recognition the length of a feature vector becomes high, since it consist of information on the distribution of different LBP combinations in varying spatial locations. We analyse the properties of different LBPs in a statistical view and derive a new LBP characterisation method called symmetry. It is based on our observation that the occurrence probabilities of the LBPs follow roughly the binomial distribution.

Keywords: Local Binary Pattern, LBP, natural images, statistical

TUCS Laboratory
Microelectronics

1 Introduction

Non-parametric local image descriptors allow tolerance against various image distortions, such as intensity bias changes and other monotonic transformations [1]. Local Binary Patterns (LBPs) belong to the class of non-parametric local image descriptors and were first proposed for texture analysis in [2]. Recently, the LBPs have been applied also to several other applications such as motion detection [3] and human biometric authentication; face detection [4] and analysis [5], and palmprint identification [6].

Efficient selection of LBP features is of major importance due to the relatively large number of combinations needed in the LBP representation. For example, in [3] each pixel in a scene was modelled as a group of adaptive LBP histograms. Two main approaches for feature vector length reduction has been proposed. First, the LBPs can be divided into shift or rotation invariant classes, which also provides local rotation invariance [8]. Second, it has been empirically observed that the LBPs with a low number of circular 1-0 and 0-1 transitions are more frequent and discriminative than the others [9]. In this paper we present a finding, that the binomial probability distribution governs the average LBP occurrence probabilities. As a concequence, the discriminative efficiency of the LBPs is dependent on a measure called symmetry [10], which characterises the binomial property.

2 Background

In the derivation of non-parametric descriptors a local window is usually formed, which values are determined by intensity comparisons to a center pixel (Figure 1). This process is usually also used in the derivation of more traditional non-parametric measures, such as the rank transform [1] and the census transform [7]. In the census transform a square 2D window is used which is compared using the hamming distance to another window in an another image to find a level of correspondence. With the LBPs a 1D circularly shaped window is used instead of 2D, reducing the amount of possible window combinations. A smaller number of window combinations (different LBPs) allows a more compact representation of an image or an image region by a LBP histogram.

2.1 Derivation of a Local Binary Pattern

In Figure 1, a sampled neighborhood of a pixel is shown. If a sample point is located between pixels its value is the average of the neighboring pixels. Each pixel in the image is used as center point x_c so that M samples are taken in a contour of a circle with radius r. This is denoted by $LBP(M, r)$. The Local Binary Pattern is derived so that, starting from a certain angle (shown in Figure 1 by straight line), following the circle on a fixed direction (clockwise or counterclockwise), sample

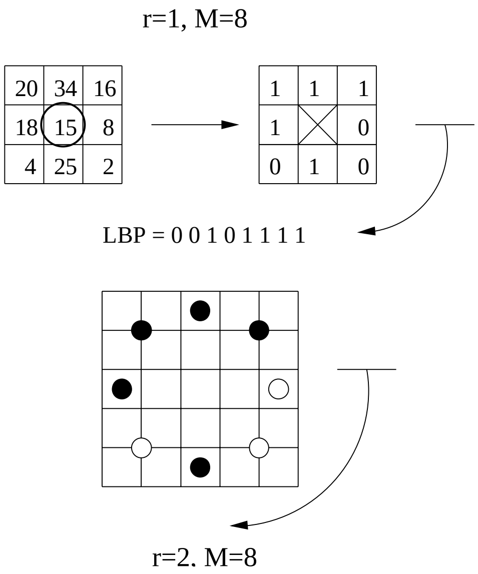


Figure 1: Local Binary Pattern with radius r=1 (up) and r=2 (down) with M samples

points are compared to the center point and set to binary 1 or 0, depending on the intensity of the sample compared to the center point. If the grayscale value of a sample pixel is greater than the center point, the binary value 1 (0) is selected and 0 (1) is selected otherwise. If the radius r is one, a pixel can be compared with all of its eight nearest neighbours shown in Figure 1 (top). A binary pattern with M bits is generated by storing the consecutive binary values. The patterns can also be converted into a decimal form. Total of 2^M different values can be generated using M bit patterns.

As a result, each pixel is replaced with a binary pattern (except the borders of the image where the neighbours can not be explicitly determined). The feature vector of the image usually consists of the histogram. The initial length of the histogram is 2^M. If there are multiple (N) regions in an image, the histograms can be concatenated into a single histogram with length $N \cdot 2^M$. The purpose of dividing an image into regions is to include more spatial information into the histogram. For the histogram comparison there exists several methods. These include histogram intersection, log-likelihood statistics, and Chi-square statistics. It is also possible to use multiple LBPs simultaneously (with possible different radiuses) to describe a certain location, which on the other hand, increase further the feature vector length [11].

2.2 Applications of the LBP methodology

Recently the LBP methodology has been applied on several other computer vision applications beyond texture analysis [8], for which it was first proposed. In motion tracking [3] adaptive LBP histograms were used in natural scenes to model the background and to detect moving objects (e.g. humans and cars) using background substraction. In comparison to other common approaches the method was shown to be very competitive. In the field of face analysis, the LBP methodology has also been shown to be very competitive against the state-of-the art [5] in the recognition accuracy. In LBP based face recognition a face image is divided into several regions. For each region a LBP histogram is generated and the concatenated histogram of all the face regions is used as the face feature vector. In [12] a step towards recognizing all the faces in the FERET (The Facial Recognition Technology) database was taken by using LBPs with Gabor filtering. However, in addition to improving the recognition accuracy, generating the histograms for each Gabor filtered orientation faces further increases the length of the face feature vector compared to [5]. In the other fields of face analysis the LBP has succesfully been applied for face detection [4] and pose estimation [13]. LBPs have also been applied to other human biometrics such as palmprint identification [6].

2.3 Existing feature selection methods

A major difficulty in taking advantage of the LBP representation is the length of the feature vectors which, for example in face recognition, assess demands on the memory storage with larger facial databases [5], [12]. Fortunately there exist feature selection methods for further characterisation of the LBPs allowing a feature vector length reduction. Two known methods are rotation invariance [8] and uniformity [9]. Reducing all the LBPs which are rotation invariant (or shifting invariant, considering the leftmost and the rightmost LBP bits as neighbours) with respect to each other to a single histogram bin can be used to make the representation locally rotation invariant, simultaneously reducing the amount bins. However, in addition to feature vector length reduction this method usually also reduces the recognition accuracy. Another method is uniformity [9], in which the LBPs are characterised by their number of 1-0 and 0-1 transitions. For example, the LBP 10111011 has four transitions and the LBP 11110000 has two circular transitions (the leftmost and the rightmost bits considered as neighbours). Using the LBPs with at most two circular transitions, in addition to a decrease in the histogram length, has shown also to maintain the recognition accuracy unaltered [5]. The usage of uniformity is based the fact that locally there exist strong mutual correlations among natural images.

Clearly, in most applications certain LBPs are more discriminative than the others. It is likely that for a some degree the discriminatory power is also application dependent. Therefore also search methods have been proposed to find an optimal subset of LBPs [11]. Furthermore, in some applications such as motion tracking from video it could be possible to adaptively boost the optimal subset of selected features [14]. However, it is clear that these last two methods also increase the computational complexity of the algorithm.

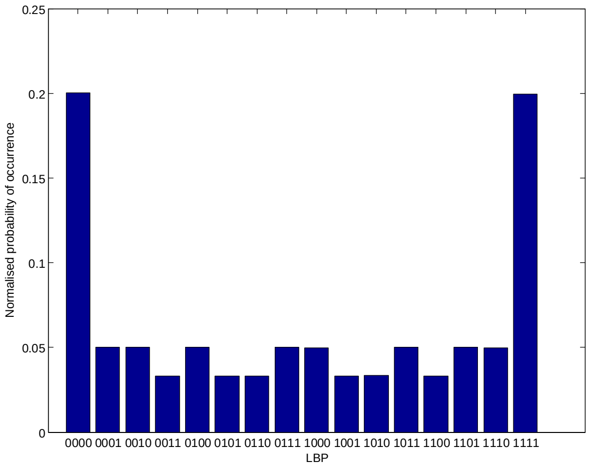


Figure 2: LBP(4,1) distribution with gaussian i.i.d. image data

3 Characterising the LBP sampling

To investigate the distribution of the LBPs the data samples are first considered to be i.i.d. Although with natural images there exist strong local dependencies, our intention is to investigate whether the LBP sampling process itself has anything to do with the LBP distribution. This can be justified by the fact that even natural images are not fully dependent, but in a statistical view contain some degree of independency [15]. Also, it can be assumed that the pixels locally follow some probability distribution, which may vary globally.

To illustrate our approach, we generated a gaussian distributed i.i.d. random image with a size of over one million pixels (LBPs). The standard deviation used was 255. The LBP operator used was $LBP(4,1)$ i.e. with radius of one and sample number of four (samples taken in the vertical and horisontal directions). The distribution of the occurrence probabilities of different LBPs is shown in the Figure 2. First, it can be observed that the LBPs are not distributed uniformly, yet rather symmetrically. It can also be observed, that there exist regions of equal probability, which are formed by the following LBPs (0000, 1111 - probability of 0.2), (0001, 0010, 0100, 0111, 1000, 1011, 1101, 1110 - probability of 0.05) and (0011, 0101, 0110, 1001, 1010, 1100 - probability of 0.033). It is evident that the distribution can be determined by using the binomial distribution so that there exist $2 \cdot \binom{4}{0} = 2$ elements of probability 0.2, $2 \cdot \binom{4}{1} = 8$ elements of probability 0.05, and $\binom{4}{2} = 6$ elements of probability 0.033. Furthermore, we see that the total sum probability of each of the first two LBP classes is equal to 0.4, and the sum of the probabilities of the third class is 0.2. Hence, the probilities sum up to one.

3.1 The occurrence probability of LBPs with i.i.d. data

In the following we show the generality of the previous illustration with any i.i.d. data distribution. A rank of a LBP is either the number of ones (e.g. number of samples above the center pixel x_c) or the number of zeros (e.g. number of samples below or equal the center pixel x_c) in that pattern. A rank class consist of all LBPs with the same rank. We show using the order statistics [16] that the probability for a certain LBP rank $P(M, k)$ is constant with any i.i.d. data distribution. As a consequence, the probability for an individual LBP to occur $P_{LBP}(M, k)$ with i.i.d. data distribution follows the binomial distribution.

Definition 1 *The probability $P(M, k)$ for a certain rank class is, i.e. that* some k *out of $M + 1$ samples are above (or below) a certain sample x_c is,*

$$P(M, k) = \int_{-\infty}^{\infty} \binom{M}{k} F(x)^k [1 - F(x)]^{M-k} p(x) dx \tag{1}$$

where $F(x)$ is the cumulative probability distribution c.d.f. and $p(x)$ is the probability density distribution p.d.f. of all the samples.

Theorem 1 *The probability $P(M, k)$ for a certain rank class is constant,*

$$P(M, k) = \frac{1}{M + 1} \tag{2}$$

A proof is provided in the Appendix I.

Corollary 1 *The probability of a LBP to occur with any i.i.d. data, with a rank (or symmetry level) of k is,*

$$P_{LBP}(M, k) = \frac{1}{\binom{M}{k}(M + 1)} \tag{3}$$

Note that if we leave $\binom{M}{k}$ out of Definition 1 we obtain the Corollary 1 directly by following the proof. This result also extends to discrete distributions, where it is possible to obtain two instances for a same value. In that case a randomization procedure (determining the order of samples using uniform distribution) can be used.

To obtain the probability for each LBP we have to divide $P(M, k)$ with the number of binomial combinations related to the rank classes. Hence, the probability for a $certain$ LBP to occur $P_{LBP}(M, k)$ is presented in the Equation 3. This result states that the occurrence probabilities of the LBPs with i.i.d. data distribution change binomially (the symmetry which is later defined can also be used instead of k).

3.2 Definition of the symmetry

The number of ones or the number of zeros (the rank of a LBP) could be used as such to discriminate the LBPs into classes. However, if we are interested in characterising the LBP occurrence probabilities in a statistical view, this description is not unique. The level of symmetry [10] L_{sym} is the minimum between the total number of zeros $|B_i =' 0'|$, $| \cdot |$ being the cardinality of the set, and the total number of ones $|B_i =' 1'|$ in a LBP (Equation 4), where index i goes through all bit locations

$$L_{sym} = \min\{|B_i =' 1'|, |B_i =' 0'|\}. \tag{4}$$

If the number of LBP samples is M, the range of the symmetry values obtained using Equation 4 is $[0, M/2]$ for even M. For odd M the range is $[0, (M-1)/2]$.

For clarity, let us again consider Figure 2. The overall probability for symmetry class $L_{sym} = 0$ is 0.4. This probability consist of two rank classes, one for LBP 0000 and the other for LBP 1111. Both these rank classes possess the probability $1/(M+1) = 1/5$ to occur. Hence the overall probability for the symmetry class $L_{sym} = 0$ is $2 \cdot 1/5 = 0.4$. The overall probability for symmetry class $L_{sym} = 1$ is also 0.4, which consist of four LBPs belonging to a rank class with one 0, and four LBPs belonging to a rank class with one 1. Also both these rank classes possess the probability $1/(M+1) = 1/5$ to occur. Hence the overall probability for the symmetry class $L_{sym} = 1$ is also $2 \cdot 1/5 = 0.4$. However, with the symmetry class $L_{sym} = 2$ there is only one rank class left and therefore the overall probability for this class is only $1/(M+1) = 1/5 = 0.2$. This is for even M. If M were odd, also the middle rank (or later, spacing) would need to be considered twice.

3.3 Properties of the symmetry

The measure of symmetry L_{sym} is invariant with respect to any increasing mototonic transformation in the same way as the the rank and LBP's rank classes. Also, symmetry is invariant with respect to any decreasing monotonic transformation, and as a consequence, with respect to any monotonic image transformation (mapping) [17]. Symmetry is also locally rotation invariant, in that for a certain pattern with a certain symmetry level all the rotations and their negations (including the negation of the pattern itself) exist among the same symmetry level. For uniform patterns (of second order) the symmetry includes uniquely all the rotations of a pattern and their negations.

3.4 Spacings

Spacings can be used to illustrate the expectations of the rth order statistics [16]. The expected length for a spacing is $E_r = E[x_{(r+1)} - x_{(r)}]$, where $x_{(r)}$ is the rth ordered sample. If we use a two-valued window, or LBP to represent a order

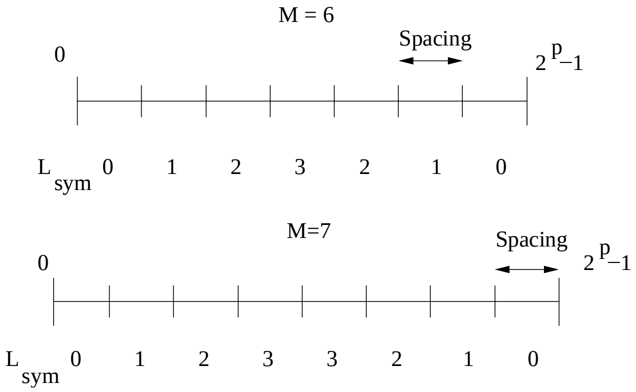


Figure 3: Illustration of spacings related to even and odd (M) LBP symmetry

statistics, its number of ones (or number of zeros) i.e. the rank tells how many samples were above (or below) the center pixel x_c. If the LBP contour samples with i.i.d. data are arranged in the order of their magnitudes, the expectations E_r can be used to represent their distribution. For simplicity, let us consider i.i.d. data with uniform distribution (Figure 3). The probability for each spacing is $1/(M+1)$, which is the probability that a random center sample x_c falls into a certain spacing. The number of spacings is the same as the number of the rank classes. It can be seen from the Figure 3 that one symmetry class usually covers two rank classes or spacings. However, if a LBP (or its center pixel x) represents a median i.e. the highest symmetry with an even number of samples M, there is only one rank class or spacing to deal with (Figure 3).

Given some LBP (with i.i.d. sample data) the determination of its occurrence probability goes as follows; determine LBP's symmetry using the Equation 4 and then determine its occurrence probability using the symmetry level L_{sym} as k in the Equation 3. In the following section we will consider the LBP probabilities with natural image data.

4 Experimental Study of LBP Distribution with a Natural Image

In this section we will consider the occurrence probabilities of the LBPs with natural image data with respect to their symmetry levels. In order to characterise the distribution of the LBPs with natural images the occurrence probabilities of each symmetry class is measured.

In each rank class the amount of combinations is $\binom{M}{L_{sym}}$. In each symmetry class, except the case of even M and the median spacing, there exist total number of $2\binom{M}{L_{sym}}$ combinations, which are used for normalising the $P(M,k)$ to obtain $P_{LBP}(M,k)$. In the case of even M and the median spacing (the highest symme-

Figure 4: Test image ’city’, 1600x1200 pixels (grayscale)

try) the normalising factor is $\binom{M}{L_{sym}}$. Rather than showing the normalised probabilities, which follow the binomial distribution we show the average probabilities of the symmetry classes.

In Table I, uniform i.i.d. noise is added to test image. The $LBP(12, 2)$ operator is used, for which the average occurrence probabilities of the symmetry levels are shown. The test image intensities vary between 0 and 255. Noise value of n% is added by calculating a zero-mean uniformly disitributed matrix, which varies between $-0.5 \cdot n/100 \cdot 255$ and $0.5 \cdot n/100 \cdot 255$.

In the following, we give two examples of how to interpret Table I (and the following Tables). The total probability for symmetry class $L_{sym} = 4$ with 15% i.i.d. uniform noise added, in the Table I, is 0.161. To obtain the average $P_{LBP}(M, k)$ the 0.161 have to be divided with $2\binom{12}{4} = 990$. The average $P_{LBP}(M, k)$ is then 0.000163. With $L_{sym} = 6$ in Table I and 15% noise level the symmetry class probability is 0.092. In this case (the highest symmetry level) the 0.092 is normalised (divided) with $\binom{12}{6} = 924$. The result for average $P_{LBP}(M, k)$ is then 0.0000996.

It can be observed from Table I, that also with a natural image the distribution of the LBP occurrence probabilities remain binomially distributed, since the normalisation evidently leads to this (the occurrence probabilities of the symmetry classes are relatively near each other before the normalisation). We have noticed this behaviour with several kinds of natural images, representing different LBP fields of applications. Table I also illustrates that by adding an independent component to an image slowly changes the distribution of the symmetry levels to that expected of i.i.d. data.

L_{sym} probabilities with LBP(12,2), (%) uniform noise added								
	0%	5%	10%	15%	20%	25%	30%	40%
0	0.170	0.118	0.126	0.131	0.136	0.138	0.141	0.145
1	0.094	0.123	0.134	0.140	0.144	0.147	0.148	0.150
2	0.116	0.137	0.145	0.149	0.151	0.152	0.153	0.154
3	0.128	0.147	0.152	0.155	0.155	0.156	0.156	0.156
4	0.143	0.160	0.161	0.161	0.160	0.159	0.158	0.158
5	0.203	0.196	0.182	0.173	0.168	0.164	0.162	0.158
6	0.146	0.119	0.101	0.092	0.087	0.084	0.082	0.079

Table 1: Average percentage of patterns.

5 Discussion

According to the information theory, the information content of an event is a decreasing function of its occurrence probability [18]. This has to do also with the discriminative efficiency of the LBPs. If we are able to match a LBPs (e.g. in two separate face images) with having a low occurrence probability (i.e. high level of symmetry L_{sym}), the match seems to be be more discriminative than if we had obtained a match of a LBPs having a high occurrence probability (i.e. low level of symmetry L_{sym}).

Symmetry is a measure of how much the LBP sampling neighbourhood deviates from the local median. We could also draw an analogy from the symmetry to the well-known non-parametric one-sample sign test, where random samples taken from a continuous population are used to test whether a known median corresponds to the unknown median of the population. The degree of confidence is then determined from the binomial distribution. With the LBP symmetry we use a hypothesis, that the median of the sample distribution is fixed, i.e. the center sample.

In feature vector length reduction for improving the performance the LBP based systems, symmetry suites to be used together with the previously known uniformity [9] method. Actually, these two measures complete each other in that uniformity measures the circular correlation i.e. dependency between the samples, while symmetry is based on a degree of statistical independency.

6 Conclusion

The Local Binary Pattern (LBP) occurrence probabilities were shown to be characterised by the binomial distribution. As a consequence, the occurrence probabilities of certain LBPs are significantly lower than the others with natural images. The average occurrence probability of a LBP is determined by its level of symmetry, which can be easily calculated for each individual LBP. We used the fact that high level of symmetry patterns are the most discriminative in [10] for face recognition. Due to its statistical nature, also LBP applications beyond face recognition can take advantage of this consept.

6.1 Appendix: Proof of theorem 1

$$
\begin{aligned}
P(M,k) &= \int_{-\infty}^{\infty} \binom{M}{k} F(x)^k [1-F(x)]^{M-k} p(x)dx \\
&= \int_{-\infty}^{\infty} \binom{M}{k} F(x)^k [1-F(x)]^{M-k} dF(x) \\
\text{denote} \;\; & F(x) = y \\
&= \int_0^1 \binom{M}{k} y^k (1-y)^{M-k} dy \\
&= \binom{M}{k} J(M,k) \\
& \text{For} \;\; k=0, \\
& \quad J(M,0) = \frac{1}{M+1} \\
& \text{For} \;\; k>0, \; \text{integrate by parts :} \\
& \quad J(M,k) = udv = 0 - \int_0^1 vdu \\
&= \frac{M-k}{k+1} \int_0^1 y^{k+1}(1-y)^{M-k-1} dy \\
&= \frac{M-k}{k+1} J(M,k+1) \\
& \text{The recursion follows :} \\
& \quad J(M,k) = \frac{k}{M-k+1} J(M,k+1) \\
P(M,k) &= \binom{M}{k} J(M,k) \\
&= \binom{M}{k} \frac{k!}{M(M-1)...(M-k+1)} J(M,0) \\
&= \frac{1}{M+1} \\
P_{LBP}(M,k) &= J(M,k) = \frac{1}{\binom{M}{k}(M+1)}
\end{aligned}
\tag{5}
$$

Acknowledgment

The author would like to thank Prof. Valery Ipatov for his valuable contribution.

References

[1] Bhat, D.N., Nayar, S.K. *Ordinal Measures for Image Correspondence*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 20, pp. 415-423, 1998.

[2] T. Ojala, M. Pietikainen, D. Harwood. *A comparative study of texture measures with classification based on featured distribution*, Pattern Recognition, Vol. 29, Issue 1, pp. 51-59, 1996.

[3] M. Heikkila, M. Pietikainen. *A Texture-based Method for Modeling the Background and Detecting Moving Objects*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 28, Issue. 4, pp.657-662, 2006.

[4] A. Hadid, M. Pietikainen, T. Ahonen. *A Discriminative Feature Space for Detecting and Recognizing Faces*, IEEE Conference on Computer Vision and Pattern Recognition, CVPR, pp. 797-804, 2004.

[5] T. Ahonen, A. Hadid, M. Pietikinen M. *Face description with local binary patterns: Application to face recognition*, IEEE Transactions on Pattern Analysis and Machine Intelligence, In press. Available at www.ee.oulu.fi/mvg

[6] Xianji Wang, Haifeng Gong, Hao Zhang, Bin Li, Zhenquan Zhuang. *Palmprint Identification using Boosting Local Binary Pattern*, 18th International Conference on Pattern Recognition (ICPR), pp. 503-506, 2006.

[7] R. Zabih, J. Woodfill. *Non-parametric Local Transforms for Computing Visual Correspondence*, Third European Conference on Computer Vision, Stockholm, Sweden, 1994.

[8] T. Ojala, M. Pietikainen, T. Maenpaa *Multiresolution Gray-scale and Rotation Invariant Texture Classification with Local Binary Patterns*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 24, Issue. 7, pp. 971-984, 2002.

[9] T. Maenpaa, T. Ojala, M. Pietikainen, M. Soriano. *Robust Texture Classification by Subsets of Local Binary Patterns*, 15th International Conference on Pattern Recognition (ICPR), Vol. 3, pp. 939-942, 2000.

[10] O. Lahdenoja, M. Laiho, A. Paasio. *Reducing the feature vector length in local binary pattern based face recognition*, IEEE International Conference on Image Processing (ICIP), Vol. 2, pp. 914-917, 2005.

[11] T. Maenpaa. *The Local Binary Pattern Approach to Texture Analysis - Extensions and Applications*, Dissertation, Acta University of Oulu C 187, 2003.

[12] Shiguang Shan, Wenchao Zhang, Yu Su, Xilin Chen, Wen Gao. *Ensemble of Piecewise FDA Based on Spatial Histograms of Local (Gabor) Binary Patterns for Face Recognition*, 18th International Conference on Pattern Recognition (ICPR), pp. 606-609, 2006.

[13] X. Feng, M. Pietikainen, A. Hadid. *Facial expression recognition with local binary patterns and linear programming*, Pattern Recognition and Image Analysis, Vol. 15, Issue. 2, pp. 546-548, 2005.

[14] H. Grabner, H. Bischof. *On-Line Boosting and Vision*, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Volume 1, pages 260-267, 2006.

[15] J. G. Daugman. *High confidence visual recognition of persons by a test of statistical independence*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 15, Issue. 11, pp. 1148 - 1161, 1993.

[16] A. Stuart, J.K. Ord. *Kendall's Advanced Theory of Statistics*, Chapter 14: Order-Statistics. Charles Griffin, LTD, London. 1943.

[17] O. Lahdenoja, E. Alhoniemi, M. Laiho, A. Paasio. *A Shape-Preserving Non-Parametric Symmetry Transform*, 18th International Conference on Pattern Recognition (ICPR), Vol. 2, pp. 373-377, 2006.

[18] C.E. Shannon. *A Mathematical Theory for Communication*, The Bell System Technical Journal, Vol. 27, pp. 379-423, 623-656. 1948.

ARTICLE 4

O. Lahdenoja, M. Laiho, J. Maunu, A. Paasio, A Massively parallel face recognition system, EURASIP Journal on Embedded Systems - Special Issue on Embedded Vision Systems, Article ID 72316, 2007.

Hindawi Publishing Corporation
EURASIP Journal on Embedded Systems
Volume 2007, Article ID 72316, 13 pages
doi:10.1155/2007/72316

Research Article

A Massively Parallel Face Recognition System

Olli Lahdenoja,[1, 2] Mika Laiho,[1] Janne Maunu,[1, 2] and Ari Paasio[1]

[1] *Department of Information Technology, University of Turku, Joukahaisenkatu 3-5, 20014 Turku, Finland*
[2] *Turku Centre for Computer Science (TUCS), University of Turku, Joukahaisenkatu 3-5 B, 6th floor, 20520 Turku, Finland*

Received 12 April 2006; Revised 31 August 2006; Accepted 5 October 2006

Recommended by Heinrich Garn

We present methods for processing the LBPs (local binary patterns) with a massively parallel hardware, especially with CNN-UM (cellular nonlinear network-universal machine). In particular, we present a framework for implementing a massively parallel face recognition system, including a dedicated highly accurate algorithm suitable for various types of platforms (e.g., CNN-UM and digital FPGA). We study in detail a dedicated mixed-mode implementation of the algorithm and estimate its implementation cost in the view of its performance and accuracy restrictions.

1. INTRODUCTION

Face recognition is easy for humans but extremely difficult to perform with computer systems reliably in varying environmental conditions. Traditionally, the development of computer-based face recognition systems has concentrated on algorithm design, while the implementation cost in the hardware level in the form of, for example, speed and power consumption has been on a lower priority.

Methods such as PCA (principal component analysis) [1], also called as the eigenface method, and LDA (linear discriminant analysis) [2] aim at a more compact face representation (feature vector). This is performed by minimizing the mutual dependencies between the samples producing more uncorrelated and compact features. A variant of the PCA, two-dimensional PCA (2DPCA), was presented in [3]. As a consequence, the length of a face feature vector was increased with the overall recognition rate. After the computation of the feature vectors, a classifier, such as an SVM (a support vector machine) [4], is used to determine a distance measure between certain faces. This is performed by maximizing the margin between the different sample classes. In elastic bunch graph matching (EBGM) [5], the face is represented as a graph consisting of nodes which are represented by *jets* and edges describing the facial features. The *jet* is a descriptor of a local image region and it can be constructed using a wavelet approach. Some variants of the above-mentioned methods are not well suitable for high-speed implementations with large facial databases [6].

Nonparametric features called local binary patterns (LBPs) have recently shown high discriminative performance in many applications, for example, in face recognition [7]. The results in face recognition were achieved in comparison to state-of-the-art recognition methods including PCA, LDA, Bayesian classifier, and EBGM using a standard CSU (Colorado State University) [8] and FERET (the facial recognition technology) [9] environment.

The term nonparametric refers primarily to the fact that no assumptions are made on the local probability distributions of the image pixel intensities. The main advantages of nonparametric LBP features are invariances against various transformations, such as lighting bias and rotation. The advantages of massively parallel processing include high image processing performance, since the operations are performed simultaneously for a large number of processing units connected together into an array. We intend to use massively parallel processing to accelerate the performance of the LBP-based face recognition, allowing a compact implementation, including the imaging device (CMOS image sensors). Many real world applications can be predicted for this embedded face recognition system, for example, biometric face authentication in security applications.

The cellular nonlinear network (CNN) technology [10] is a powerful tool for high-speed massively parallel image processing. The concept of a programmable massively parallel CNN-UM (CNN-universal machine) processor was proposed in [11] where programming capabilities and memory were to be integrated to the same chip. Several chip implementations have been made to show that extremely high computation power can be included in a single chip [12] and a QCIF (176 × 144 pixels) resolution binary I/O CNN-UM [13].

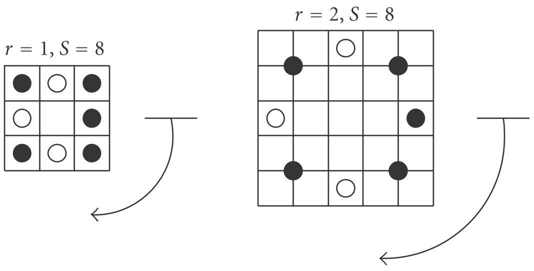
This paper describes a massively parallel face recognition system. The system consists of three parts, which are mixed-mode or CNN-UM-based LBP sampling methodology [14], a LBP-based massively parallel face recognition algorithm [15], and a dedicated mixed-mode hardware for the proposed algorithm (partly described in [16]). The sampling method which we present has the advantage of a speed increase up to 5 times compared to a modern standard computer, with on the other hand, some decrease in sampling accuracy and flexibility in the LBP sampling neighborhood size. In addition to face recognition, where the sampling speed is not critical, other high-speed LBP applications are likely to benefit from this sampling concept.

The presented algorithm is very competitive against the previous LBP-based face recognition algorithms in the recognition accuracy, while a trade-off between the face description length and recognition accuracy remains. The algorithm is flexible in that it can be applied beyond mixed-mode implementations for a massively parallel digital FPGA which does not introduce a significant decrease in the sampling accuracy. The dedicated mixed-mode hardware implementation for the proposed face recognition algorithm is presented in detail in schematic level with simulations considering mismatch and A/D conversion accuracy restrictions. This information can be used for determining whether it is appropriate for certain specific applications to use a fully digital or a mixed-mode implementation. At this point, the dedicated hardware is not fabricated, instead its performance is estimated using simulation tools.

2. THE LOCAL BINARY PATTERN METHODOLOGY

Local binary pattern (LBP) methodology was presented in [17] as a texture measure. It is based on comparing each grayscale pixel to its nearby samples and producing unique binary patterns based on the relative intensities of the pixels. The neighborhood is defined to be circular allowing invariance against rotation. First versions of the LBP were implemented using an eight pixel nearest neighborhood, but later circular neighborhoods with an arbitrary radius have been used. Also neighborhoods with multiple radiuses have been suggested in [18]. As a texture descriptor, the LBPs have shown to be very efficient in the view of computational complexity and recognition accuracy (see [18, 19]).

Recently, the LBP methodology has been applied also on several other computer vision tasks beyond the face recognition and the texture analysis. The LBP was used in [20] for modeling the image background and detecting moving objects. A combined face detection and recognition system was implemented in [21]. Also, many other computer vision applications for the LBP have been implemented, such as pose detection, context-based image retrieval, and industrial paper quality inspection.



Local binary pattern: 11010101
Decimal: 213

Figure 1: Local binary patterns with radiuses r of 1 and 2 and sample number S of 8.

2.1. Deriving the LBP feature vectors

Figure 1 represents the derivation (sampling) of an LBP. Comparison of the contour pixel intensities with the center pixel is started from a certain predetermined angle and proceeded in a certain direction, for example, clockwise. The LBP is generated by going through the whole local circle contour. If a circle contour sample is located between two pixels, the actual value of the sample is found out by interpolation (e.g., the average). As a result, each original pixel is replaced by its LBP representation, which is invariant with respect to monotonic grayscale changes and, in certain circumstances, rotation [19]. A histogram of length 2^S, where S is the number of local contour samples, is generated to describe an image or a region in an image. Several methods exist for histogram comparison, such as the chi-square statistics employed in [7].

The number of bins in the histogram can be reduced from 2^S using uniform [22] patterns which consist of at most two circular 0-1 and 1-0 transitions. For example, patterns 11000011 and 00011100 are uniform patterns (the leftmost and rightmost bits are considered neighbors), and patterns 01010100 and 01101101 are not uniform patterns. Further reduction on the number of histogram bins can be achieved by using a discrimination concept called symmetry [23].

The level of symmetry is the minimum between the total number of zeros $|B_i =' 0'|$, $|\cdot|$ being the cardinality of the set and the total number of ones $|B_i =' 1'|$ in an LBP (1), where index i goes through all bit locations

$$L_{\text{sym}} = \min\{|B_i =' 1'|, |B_i =' 0'|\}. \qquad (1)$$

By using uniform patterns with a high level of symmetry (e.g., 00111100 and 11110000), the number of histogram bins can be reduced significantly (for S of 8, the reduction is 88%) from 2^S.

In [19] rotation invariant categories for the local binary patterns were defined. Certain local binary patterns can be rotated from each other to a minimum value so that each of the patterns in a certain class produces the same decimal value. For example, patterns 01100000, 00110000, and 00011000 belong to the same rotation invariance class since the minimum value that can be extracted by shifting all these patterns is 00000011.

3. LBP-BASED FACE RECOGNITION

In general, the process of recognizing, whether a specific input face (e.g., sensed by a camera or integrated sensors) matches one of the N reference faces stored into the memory, consists of many separate comparison steps. Two faces, the input face and the stored reference face, are compared with each other in each step. A distance measure (e.g., an integer number) is derived based on the comparison result according to (2),

$$d_{\text{match}} = \underset{d}{\operatorname{argmin}}\, d(\text{face}_{\text{input}}, \text{face}_{\text{stored},1:N}). \tag{2}$$

Depending on the value of the distance measure d, the probability of the individuals to be matched can be determined. In the LBP face recognition, the face is usually divided into regions that can be used with or without weighting to enhance the spatial accuracy.

3.1. The histogram-based method

In [7] the face recognition was based on dividing face images into several spatially neighboring (e.g., 7 × 7 with 130 × 150 8-bit grayscale images) histogram regions, and an LBP histogram was constructed to represent each of these regions. A face descriptor was then a concatenated histogram of all the regions. A weighted chi-square distance (3) between the concatenated histograms was used with w denoting weights and S and M denoting the sample and model distributions, respectively. Index j is specific to a block and index i to a histogram bin,

$$\chi_w^2(S, M) = \sum_{i,j} w_j \frac{(S_{i,j} - M_{i,j})^2}{S_{i,j} + M_{i,j}}. \tag{3}$$

Not all the regions were considered equally important, but when comparing the feature vectors (concatenated histograms) the more informative face areas such as the eyes and the mouth were weighted (multiplied) by a factor that was considered to be optimal for the overall recognition results. In practice, the weighting factors were determined from the effect of that specific region on the overall recognition rate by neglecting all the other regions at a time.

3.2. The occurrence map method

A problem with the histogram approach [7] is related to the block division, since the borders between the blocks may lose information in the face comparison. If an LBP is slightly moved near the border of two blocks, it can move from one block into another. This causes a relatively large effect on the histograms of specific blocks. If the number of blocks is large, the occurrence probability of this block mismatch increases. More importantly, the spatial relations of the LBPs inside a block are not preserved by the histogram representation.

To improve the recognition accuracy of the LBP, we presented the occurrence map method for face recognition in [15]. Since each pixel can be replaced by a unique pattern in the LBP methodology, we chose to represent the occurrences of a certain pattern in an image (or an image region) by a binary occurrence map. For each histogram bin (LBP), a separate binary occurrence map is generated (with the size of the original image), that represents the locations for which that specific pattern occurs. The occurrence maps are compared with each other so that the other map is first dilated into certain directions. With D LBPs there will be an equal number of occurrence maps for describing their locations. The representation with D occurrence maps is possible with a tolerable feature vector length due to the two LBP compression methods, uniformity and symmetry.

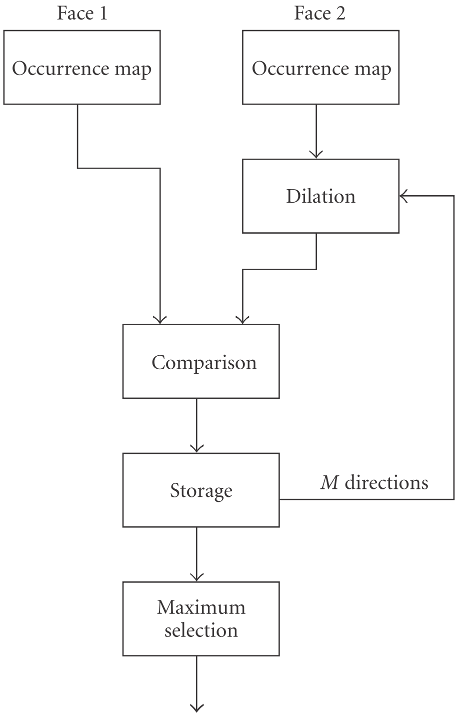


Figure 2: The matching algorithm.

4. FACE RECOGNITION WITH THE OCCURRENCE MAP METHOD

The flow diagram for comparing two faces using the proposed algorithm is shown in Figure 2. The two collections of occurrence maps, one representing the input face and the other representing the stored reference face, are given as the inputs to the matching process. A certain LBP is chosen to be processed. As a consequence, one occurrence map is chosen at a time, from both occurrence map collections (for face one and face two). Also, the image has been divided into blocks so that a spatially corresponding block is selected from both, the input face and the stored reference face.

The binary occurrence map of the face number two is dilated in a predetermined direction (e.g., N, W, S, E, etc.) a predetermined number of times. The dilated occurrence map is compared to the corresponding block of the other

```
SELECT the input face,
FOR each $N$ stored faces,
    FOR all occurrence map pairs (total of $D$),
        FOR each block,
            FOR each direction $M$,
                $K$ times Dilations,
                ADD (AND (occurrence maps)),
                STORE (add),
            END FOR,
        END FOR,
        ADD (block specific sums),
    END FOR,
    SELECT overall block direction
    ADD (block and direction specific sums)
END FOR,
MAX(add)
```

Algorithm 1

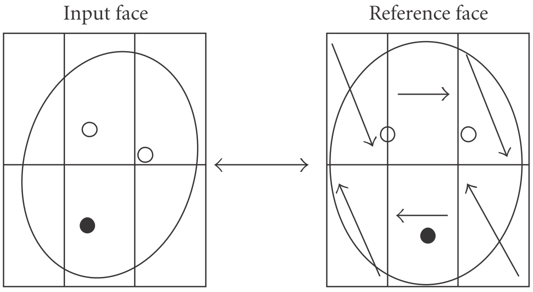


Figure 3: Examples of block matching vectors for matching two faces.

occurrence map (of face number one) by using a (local) AND operator for the whole array simultaneously. The dilation is repeated for M directions. Comparison results representing the similarities between the occurrence maps for specific dilation directions are stored into memory storage. Block specific occurrence map matching results are summed together, so that an optimal dilation direction is searched for each block. This means that the matching direction is specific for the whole block consisting of D occurrence maps. The dilation is performed only for either one of the faces under comparison. Following pseudocode illustrates the steps within each other performed during the comparison algorithm, see Algorithm 4.

Figure 3 represents an example of determining the block matching vectors, indicating the optimal dilation direction. Our method allows tolerance against slight changes in rotation and scaling adaptively between the images. In the case of comparing face images which represent different individuals, the optimal dilation direction is used to match the most informative regions (e.g., the eyes) into corresponding locations. In the case of comparing different images taken from the same individual, the optimal dilation direction is able also to match the features specific to that individual.

4.1. Simulation environment

The performance of the proposed algorithm was simulated in Matlab environment. Face images (130 × 150 × 8 bit) normalized by the CSU (Colorado State University) face recognition evaluation system were used [8]. In [7] the same system was used since it allows comparison to the other existing state-of-the-art face recognition methods. FERET (the facial recognition technology) [9] database is used by the CSU system. The FERET database is divided into five image sets. The gallery set contains frontal images from 1196 people and it is compared against four probe sets. The fb probe set contains 1195 face images with alternative expressions. The fc set contains 194 photos taken in different lightning conditions. The $dup1$ set is taken later in time and it contains 722 images. The $dup2$ set contains 234 face images taken at least a year after the corresponding gallery image. The CSU system uses the eye coordinates of images for normalization. After that, the actual recognition process is performed for the different methods.

4.2. Feature vector generation

Sample number S of eight was used in [7] with the radius of two to obtain the best recognition accuracy. This resulted in the initial length of the histogram representing a certain region (block), of 256. This could be reduced to 59 by using only uniform patterns. By taking the advantage of the symmetry, the number of bins of a region could further be reduced to 30 with a negligible effect on the average recognition rate [23].

A total of 30 binary occurrence maps (of size 130 × 150) can be used to describe the locations of each LBP (or a group of LBPs) in an image uniquely. In a certain location (array cell) there is only a single unique LBP at a time. These 30 occurrence maps can therefore be encoded losslessly using 5 bits of accuracy, resulting into the initial length of the face feature vector of 126 × 146 × 5 bits. The total length (including all face regions) of the histogram representation in [7] with S of eight was 2301 8-bit bins. Compared to this, the feature vector generated by using the binary occurrence maps and the proposed algorithm will be approximately four times larger.

One alternative for generating more compact feature vectors would be to subsample the images or the LBP images into a lower vertical and/or horizontal resolution, since the feature vector length is reduced quadratically with respect to the sampling ratio. However, eventually, there will exist a trade-off between the recognition accuracy and the resolution.

A suitable amount of consecutive dilation operations was experimentally found to be three. The images were sized according to the CSU normalization procedure as in [7] and 30 histogram bins (or equivalently 30 occurrence maps) were used. Using more dilations caused the dilated occurrence

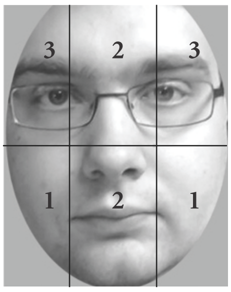


Figure 4: Division into 3 × 2 blocks and the respective weights.

map of the face number two to cover an unnecessarily large portion of the other occurrence map and the occurrence probability of false matches during the comparison operation was increased. The normalized face images were divided in the simulations to a total of six blocks as shown in Figures 3 and 4.

We performed simulations with and without block weighting. The weighting was applied for each specific block according to values shown in Figure 4. The optimal comparison results for the blocks were multiplied by the weights to enhance the effect of face areas which are considered as the most significant in the view of the recognition (e.g., the eyes) and decrease the effect of less significant areas on the final result. The weights chosen are close to that used in [7]. A proper size for the regions was determined from iterative simulations. The sizes of the blocks that resulted in the best recognition accuracy were much larger (total of 6) compared to that of [7] where 49 blocks were used for the same sized images. Effect of slight displacement in face images to the weighted recognition accuracy should therefore be less in our method than in [7]. Furthermore, the number of block dilations can be tuned in our algorithm in order to take into account larger displacements, if needed.

As the amount of dilation directions was increased from four to eight, also the recognition accuracy was improved, since the adaptivity of the algorithm was increased. Therefore, eight directions were used in the implementation of the algorithm (N, NE, E, SE, etc.).

4.3. *Simulation results*

The results of the simulations for $LBP(S, radius)$ are shown in Table 1. They consist of rank one recognition rates, which means that the exact match of the input image and the reference individual is required. The LBP-AM (adaptive matching) is used to denote the proposed occurrence map algorithm. The reference results are the same as in [7, 23], since the same CSU normalization procedure has been used, allowing an objective comparison. The LDA, PCA, EBGM, and Bayesian recognition rates are the standard implementations of the CSU system.

Table 1: FERET results with the occurrence map algorithm.

Method	fb	fc	dup1	dup2	Average
LBP-AM(8,2) weighted	97%	86%	71%	67%	80%
LBP(8,2) [7] weighted	97%	79%	66%	64%	76%
LBP(12,2) [23] weighted	95%	85%	64%	66%	77%
LBP-AM(8,2) nonweighted	95%	85%	69%	63%	78%
LBP(8,2) [7] nonweighted	93%	51%	61%	50%	64%
PCA MahCosine	85%	65%	44%	22%	54%
Bayesian MAP	82%	37%	52%	32%	51%
EBGM optimal	90%	42%	46%	24%	51%
LDA ldasoft	73%	47%	45%	18%	46%

Table 1 shows that the proposed algorithm outperforms the previous LBP-based algorithms, [7, 23] in the recognition accuracy and also the other standard CSU implementations of PCA, LDA, Bayesian MAP, and EBGM for each FERET set. The increase in recognition rate with weighting is approximately 4% compared to [7] and 3% compared to [23]. The improvement in the recognition accuracy without weighting is as much as 14.0% compared to [7]. If the algorithm is to be implemented as an embedded FPGA (digital sampling applied) the recognition rates are not expected to decrease, since mismatch is not present. However, analog sampling exposes the recognition rates for a slight decrease as demonstrated in the later sections.

5. HARDWARE ALGORITHM CODESIGN

For the LBP extraction and face recognition, two different mixed-mode massively parallel hardware architectures are considered. First, a standard CNN-UM can be used which has the advantage of being a flexible general-purpose system allowing larger manufacturing volumes with a lower cost. On the other hand, the performance of the face recognition system in the terms of silicon area, power consumption and speed can be optimized by a dedicated massively parallel hardware implementation. The dedicated implementation requires simultaneous hardware algorithm codesign. Both implementations mentioned above consist of an array of processing cells which operate in mixed mode, that is, carrying out computations based on both digital and analog processings. Some issues in mixed-mode processing have to be carefully taken into account, to ensure that the system functions properly.

5.1. *Internal accuracy*

Mixed-mode processing is well suitable, for example, for integrated near-sensor processing, where the sensed phenomena are continuously valued. Then there is no need for time-consuming A/D conversion in the data acquisition phase. In practice, the limited internal accuracy of semiconductor devices affects the results of continuously valued array operations. A main source of internal inaccuracy is device mismatch, which affects even basic current thresholding and multiplication operations. The effect of mismatch can be reduced, roughly speaking, by increasing the size of the devices. Therefore, there exists a design trade-off between the processing accuracy and the array resolution, since the number of cells in an array depends on the size of a single cell (assuming a fixed array size). In a dedicated massively parallel implementation, mismatch is used as a design constraint. It should be small enough to ensure correct (enough) operation, but on the other hand, oversizing the cells should be avoided to achieve a proper array resolution.

5.2. *Read-out of results*

Massively parallel mixed-mode systems can gain extremely high computation power, but in addition to the internal accuracy, the read-out of results is usually a performance bottleneck. One alternative is to use embedded A/D converters inside each array cell, but this has the disadvantage of increasing the area of the cells. Another approach is to use a limited number of A/D converters so that the read-out from the cells is performed sequentially. On the other hand, this decreases the overall computation speed. In certain applications, we are not interested in the exact results among each cell, but, for example, the sum of the results among a certain subset of cells. In this case it is possible to use a read-out scheme such as proposed in [24], where a unit current switch is placed into each cell and these are wired together into a common output. The output current is then A/D converted. This functionality is not embedded into a traditional CNN-UM, since it requires, for example, extra wiring, but can relatively easily be implemented on a dedicated massively parallel array.

5.3. *Architectural issues*

The architecture of the massively parallel platform assesses strict limitations on the types of algorithms that can be reasonably mapped on it. Most importantly, since the array consists of identical cells connected usually only on each other in the local 8 neighborhood (e.g., in the case of CNN-UM), only operations that can be defined by a template describing this kind of neighborhood can be executed. In image processing, most of the operations needed can be mapped into a local form. It would be helpful in many applications, if the size of the neighborhood could be increased. However, this causes a need for excessive wiring in the layout of a massively parallel array, and is therefore difficult. Also, a large number of inputs into each cell would increase the cell size and might affect the internal accuracy of the computations. The LBP methodology is relatively well suited on massively parallel processing, since in most applications, neighborhoods with eight to twelve neighbor connections are enough for optimal performance [7].

Sizes of CNN arrays currently in use are sufficient for a large variety of image processing applications, for example, face recognition (a common size of the arrays is becoming to be at least 128 × 128). If resolution is limited or larger images need to be processed, it is possible to read image data one or a few blocks at a time and process these blocks separately with the CNN or a dedicated array processor. The speed and the complexity of the implemented algorithm, however, are dependent on how many processing cells are available assuming a certain input frame resolution.

6. CNN-UM TEMPLATE ALGORITHMS FOR LBP PROCESSING

In the following, we show either how the CNN-UM can be used only for LBP sampling with a large variety of different potential applications or how it can be used for face recognition with either the histogramming or with the occurrence map method. A sample number of eight is a practical maximum for radius one LBP transform. Consequently, eight logical memories (LLMs) must be allocated for LBPs, since a standard CNN operates in 8 nearest neighborhood.

6.1. *Obtaining the local binary pattern*

In order to extract the local binary patterns, the input image is written to cell input and the state is initialized to zero. S threshold operations are used to generate the S-bit local binary pattern (one bit per direction) and the result of each comparison is stored in an LLM (local logic memory). As a result, LLMs of individual cells contain the LBP pattern in that specific location. The LBP process of comparing sample points one by one to the center point can be modeled with [10],

$$\dot{x}_{ij} = -x_{ij} + a \cdot y_{kl} + \sum_{kl \in N_r} B_{ij,kl} u_{kl}, \tag{4}$$

where a is the center element of the A-template. The cell inputs are denoted with u_{kl} which are multiplied and summed with the B-template. The x_{ij} and $\dot{x}_{ij}$ are the cell state and the rate of change of the cell state, respectively. The center element equals unity and all other elements of A are zero. The output nonlinearity here is the threshold function

$$y_{ij} = \varphi(x_{ij}) = \frac{|x_{ij} + 1| - |x_{ij} - 1|}{2}. \tag{5}$$

The comparison is applied circularly for S different directions.

The bias I is zero for all the following comparison templates. The templates in Table 2 are constructed according to (6) and are used to implement the LBP transform with the radius of one. With the CNN-UM, the LBP radius can be one (or in some cases two) covering most of the practical purposes. Increasing the sample number increases the amount of comparison templates so that the number of comparison template operations is equal to the number of samples S, for one input frame. For example, using a neighborhood of one with a sample number of four, only templates T_2, T_4, T_6, and T_8 are used with a certain input image. Interpolation between two adjacent pixels is shown in the template of (7) with the

Table 2: Threshold templates for LBP with r of one and S of eight.

Template	b_0	b_1	b_2	b_3	b_4	b_5	b_6	b_7	b_8
T_1	1	−1	0	0	0	0	0	0	0
T_2	1	0	−1	0	0	0	0	0	0
T_3	1	0	0	−1	0	0	0	0	0
T_4	1	0	0	0	−1	0	0	0	0
T_5	1	0	0	0	0	−1	0	0	0
T_6	1	0	0	0	0	0	−1	0	0
T_7	1	0	0	0	0	0	0	−1	0
T_8	1	0	0	0	0	0	0	0	−1

radius of two and sample point direction of threshold template T_3. In this equation, the average of the two pixels with weights of a half is thresholded against the center pixel. Also, a larger area can be used for the interpolating neighborhood in a similar manner. If the radius is two, the algorithm works exactly the same way, with a radius of one, except that the center pixel is now thresholded against a circle with a larger radius,

$$B_{r=1} = \begin{bmatrix} b_1 & b_2 & b_3 \\ b_8 & b_0 & b_4 \\ b_7 & b_6 & b_5 \end{bmatrix}, \tag{6}$$

$$B_{r=2} = \begin{bmatrix} 0 & 0 & 0 & -1/2 & 0 \\ 0 & 0 & 0 & 0 & -1/2 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{7}$$

6.2. LBP occurrence map generation and histogramming

The LBP occurrence maps are directly extracted from the results of the thresholding operations. The operations such as dilation and the AND operation are readily available in the CNN-UM.

Pattern matching can be used to construct a histogram of the LBP patterns of an image or a region of an image. One option of creating the histogram of LBPs is to match them using the local logic unit (LLU) of a CNN-UM. In the beginning of the comparison operation, LBPs are stored in S local memories. A logic OR operation is applied bit-serially for the contents of those LLMs that are required to be LO. The result of this is inverted and written into a vacant LLM. Furthermore, a logic AND operation is applied bit-serially for the result of the inversion and the contents of those LLMs that are required to be HI. The result is HI when the LBP matches the required pattern. This operation requires a relatively large number of bit-serial logic operations for each bin of the histogram. For example, if S is 8, and the number of ones in the desired pattern is 3, we need 4 LLU OR-operations (for the bits that are to be zeros), one LLU inversion operation (for the result), and 3 LLU AND-operations (for the bits that are to be ones).

The sum of the ones (one indicating an LBP match) in the binary occurrence maps that result after the subsequent pattern matchings equals the amplitude of the histogram bin of a certain local binary pattern. If the input image fed to the CNN contains only a certain block or blocks of the actual input image, the pattern matching has to be performed for each block or a subset of blocks separately. Then, the decrease in parallelism increases the overall execution time. Assuming that the input frame is processed by P blocks at a time and the image is divided into total of R blocks, the amount of pattern matching and comparison operations is multiplied by R/P. The ratio of R and P can be decided based on the input resolution and speed requirements of the application.

6.3. Selecting symmetrical and uniform patterns

The selection of uniform and symmetrical patterns can be carried out by applying the pattern matching only for these specific patterns. A category of LBPs can be unified into a single occurrence map by using a cell specific AND operator applied for the whole array simultaneously. The patterns that are neither uniform nor symmetrical can be concatenated into the same occurrence map by a similar approach, using "do not care" conditions in pattern matching, or by summing the A/D converted results outside the array for these specific LBPs. A similar approach can be used for the dedicated mixed-mode hardware implementation for unifying the occurrence maps, by using AND operator implemented in the neighborhood logic unit.

6.4. A modified CNN-UM cell for LBP sampling

A slightly modified CNN cell can perform the pattern matching using a CNN pattern matching template [25]. Figure 5 shows the modified part of a CNN cell with $S = 8$. The first modification is that S LLMs need to be accessible simultaneously so that they can act as multiplier inputs instead of cell input. Whether the inputs to the multipliers come from the memories or from the cell input is programmable via switches. Moreover, the outputs of the multipliers can be programed to be redirected either to neighbors (normal CNN operation) or to the state node of the cell itself. In other words, the multipliers normally used for neighborhood operations are utilized in pattern matching operations.

A 3×3 pattern matching template for second-order uniform LBP of 00111000 is shown in the template of (8). The B template consists of the pattern to be matched so that a minus one corresponds to a white pixel, a plus one to a black pixel, and a zero to a "do not care" condition. The bias z is defined as 0.5-N where N is the number of pixels required to be either black or white,

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \qquad B = \begin{bmatrix} -1 & -1 & -1 \\ 1 & 0 & -1 \\ 1 & 1 & -1 \end{bmatrix}, \qquad z = -7.5. \tag{8}$$

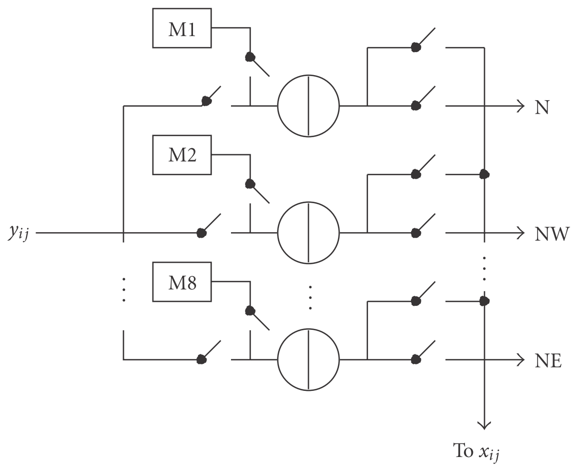


Figure 5: CNN cell modifications.

6.5. *Effect of mismatch on histogram-based face recognition*

Determination of the LBP of a pixel requires a comparison of the pixel value to its neighbors. Comparison can be performed easily in the analog domain by subtracting two currents from each other. However, since the comparison is performed in the analog domain, the mismatch of the analog devices corrupts the result. We chose to simulate the effect of mismatch in the face recognition with histogramming.

We carried out simulations as to how much analog mismatch is tolerated in the comparison operation. The standard FERET database [9] of different sets of facial images normalized by the CSU system [8] was used. The simulations were carried out so that prior to determining the feature vector of an image, normally distributed noise generated with Matlab was added to the image so that a noisy pixel value becomes

$$P_{i,j;\text{noisy}} = P_{i,j} + n_{i,j} \cdot \frac{\%}{100} \cdot 255, \qquad (9)$$

where $n_{i,j}$ is an element of a zero mean, standard deviation one, normally distributed noise matrix, and 255 is the maximum pixel intensity. Table 3 shows the results of the face recognition simulation with different percentages of mismatch. A radius of two with $M = 8$ was used. The results show that the mismatch affects mostly those images that were taken in different illumination conditions. The results shown in Table 3 were simulated so that all images, both the reference images and the probe images in fb, fc, dup1, and dup2, were corrupted by noise. If noise was only added to the probe images, the degradation in the average recognition accuracy was about half of that shown in Table 3.

6.6. *Effect of quantization in analog read-out*

We also examined the effect of quantization in the analog read-out on the recognition accuracy in Table 4. We chose to use feature extraction without weighting and used the histogram method [7]. With the block size that we used, the maximum dynamic range was between zero and 378 (21 ∗ 18). With the occurrence map algorithm, the block size is larger, but also the magnitude of the sum of ones after the comparison phase is much smaller due to the coding into occurrence maps. We noticed that the bin amplitudes of more than 255 corresponding to eight-bit accuracy had no effect on the recognition accuracy. The upper limit of the dynamic range was then divided by a scaling factor, for example, four when six-bit accuracy was used. In practice the recognition rate did not decrease even with seven-bit accuracy. With six-bit accuracy, the average recognition result was decreased only by 0.5%. Six- or seven-bit accuracy can rather easily be reached by the ADC. With five-bit accuracy, the effect on recognition rate was larger resulting in a total decrease of 3.3%. This means that summing up the currents from all cells in a block and converting with a 7-bit A/D converted give practically an unaltered performance.

Table 3: Mismatch effects for histogramming with weighting.

Mismatch %	fb	fc	dup1	dup2	Average
0	97%	80%	66%	64%	77%
1%	96%	73%	66%	65%	75%
2%	95%	69%	64%	58%	71%
3%	94%	63%	62%	57%	69%
5%	92%	55%	57%	49%	63%

Table 4: FERET histogram-based face recognition results with quantization.

Method	fb	fc	dup1	dup2	Average
LBP optimal nonweighted $r = 2, M = 8$	93%	52%	61%	49%	63.8%
LBP 6 bits nonweighted $r = 2, M = 8$	93%	49%	61%	50%	63.3%
LBP 5 bits nonweighted $r = 2, M = 8$	92%	47%	56%	47%	60.5%

7. A DEDICATED LBP HARDWARE

A dedicated hardware architecture for the adaptive occurrence map matching algorithm consists of external memory, a massively parallel processor array (including integrated current-mode imaging sensors), a control unit, and a memory interface which includes a cache. The organization of these units is illustrated in Figure 6.

The size of the external memory depends on the number of stored face feature vectors. With reference face database of 100 images, the size of the external memory becomes approximately 9.76 Mbits or 1.22 Mbytes (100 images ∗ 150 ∗ 130 pixels ∗ 5 bit/pixel). The memory interface performs the

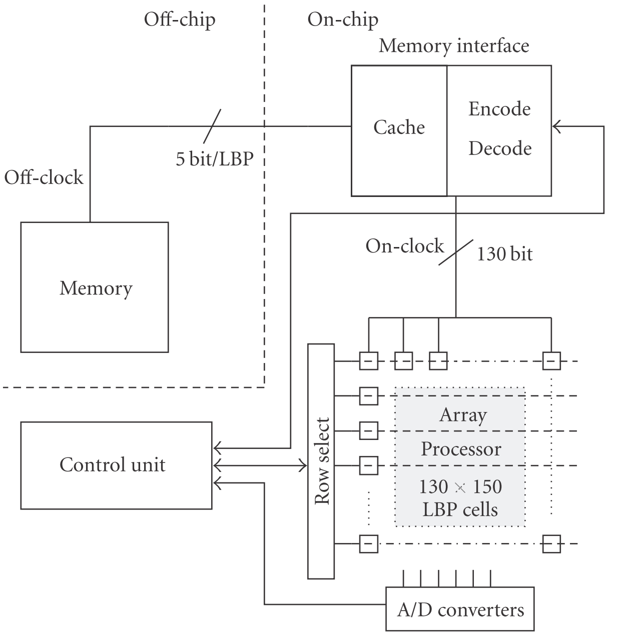


FIGURE 6: Architecture of the recognition system.

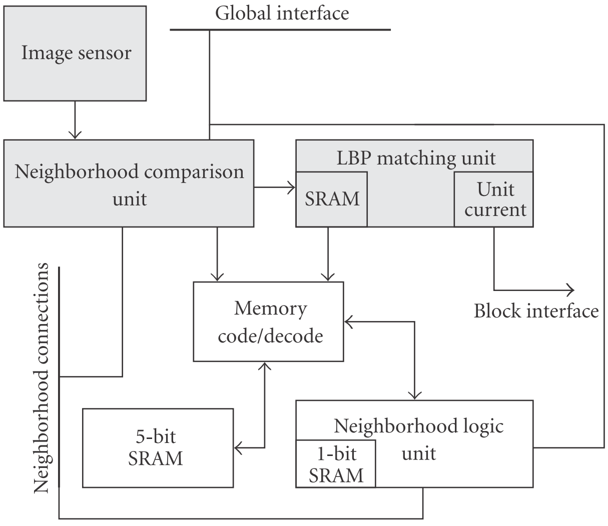


FIGURE 7: LBP processing cell for the adaptive matching algorithm.

decoding of LBP images to binary occurrence maps which are used by the array processor. We propose using a cache of one feature vector (reference face) which results the size of approximately 97.5 kbits or 12.2 kbytes. The cache is integrated on the same chip with the array. The array processor itself performs only binary operations, in addition to the LBP feature extraction (sampling). The control unit is integrated on the same chip and sends instructions and receives data to/from the array processor and to/from the memory interface. The control unit also processes the (A/D converted) results from the array.

7.1. A dedicated LBP processing cell

The dedicated LBP processing cell in Figure 7 can be used only for LBP sampling and possibly histogram generation, or it can be used for the dedicated occurrence map algorithm. The parts of the LBP cell which are targeted for LBP sampling are highlighted with gray. The full LBP cell for the face matching algorithm based on the occurrence maps includes CMOS image sensor, instruction code/decode unit, neighborhood comparison unit, LBP matching unit, memory decode/code unit, SRAM memory, and neighborhood logic unit. The neighborhood comparison unit, image sensor, and the LBP matching unit form the architecture of the general dedicated LBP processing hardware. Therefore, they are discussed in more detail in the later sections.

The neighborhood comparison unit performs the LBP sampling with a radius of two using eight samples (four of which are interpolated from their neighbors' values). The functionality and implementation of this unit are explained later in this section. The matching unit performs comparison of an LBP stored into the cell with a certain specific globally selected 8-bit LBP. If an exact match is recorded, the matching unit activates the current source specific to that cell, which is routed to the block interface. The block denotes a certain subset of LBP cells for which the weighting and A/D conversion are applied later. The memory code/decode unit performs coding of the LBP occurrence maps into a binary weighted form. For example, the binary weighted representation for a maximum of 32 occurrence maps consists of 5 bits which are needed for the local memories to store the input face. For each cell, the occurrence map indicates whether a specific LBP occurs in that cell or not. This unit also decodes the occurrence maps from the 5-bit SRAM memory into a total of 30 occurrence maps. The neighborhood logic unit is used for the directional dilation operation for specific directions. A 1-bit SRAM is included into the neighborhood logic unit for storing one occurrence map of the other face image under comparison. This occurrence map is read from the on-chip memory interface, as the full LBP image stored into the 5-bit SRAM (the input image) can be read using the on-chip sensors.

7.2. Implementation of the neighborhood comparison unit

Figure 8 shows the neighborhood comparison unit. The current $I_$cell feeds the input current of the pixel. It could be obtained, for example, from an in-cell image sensor or a D/A converter. Transistors $M1$–$M6$ are analog transistors and the rest of the transistors shown are minimum-sized switches. The neighborhood comparison unit has 12 connections to neighbors on the circle of radius 2. The neighbors are coupled so that when, for example, control signal N is active, the unit receives input current from a cell two rows above and conveys its current to the cell two rows below (see the indexes in the figure). Notice that the current of transistor $M4$ is mirrored to transistors $M5$ and $M6$ at a ratio of 0.5. This

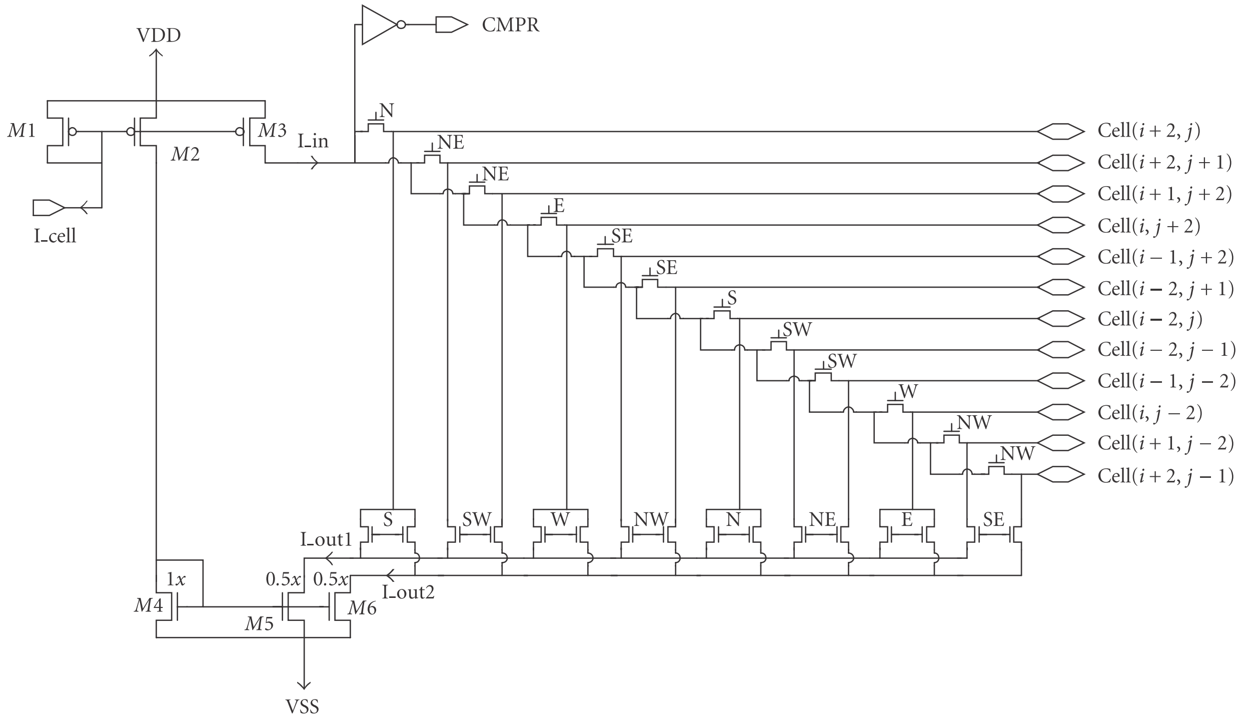


Figure 8: Neighborhood comparison unit.

is because when processing directions NE, NW, SE, and SW the cell should receive inputs from two different neighbors, the currents of which are interpolated. An inverter serves as a comparator and the comparison result is available at *CMPR*.

7.3. Implementation of the LBP matching unit

Figure 9 shows the proposed circuit for matching LBPs within the pixels in order to generate the occurrence map representation. The comparator output *CMPR* is fed to eight SRAMs. All these are read out simultaneously and they provide either an inverted or noninverted output. After the comparison circuit has written the LBP into the SRAMs, the matching using the programmable multi-input NOR is started. Notice that the SRAM and the NOR share eight control signals (N, NE, E, SE, S, SW, W, and NW) with the neighborhood comparison unit. These control signals can be used to program which pull-down paths of the NOR are possible. The weak pull-up transistor is turned on with *BIAS* during evaluation.

The matching is a two-phase process with the proposed circuit. First, the SRAMs provide the NOR with the inverted LBP. The control bits associated with LBP bits that are required to be HI are taken HI. In other words, only selected pull-down paths are enabled. Now, if all inverted LBP bits that are connected to activated pull-down paths in the NOR are LO, the result is HI. This is written to an SRAM in the output unit through an inverter. Second, the noninverted LBPs are fed to the NOR, the control bits enable the pull-down paths associated with those LBP bits that are required to be LO and the result is fed to the lower NOR input in the output unit. If the LBP matches the pattern under search, both inputs of the NOR in the output unit are zero and the gate voltage of analog output transistor M_OUT is HI.

7.4. Optimizing the performance of the LBP cell

In order to get an idea of how large the transistors $M1$–$M6$ of the neighborhood comparison unit should be, simulations were performed. The simulations were carried out with Eldo level 53 parameters of a 0.13 um CMOS process. Ideally, the standard deviation of the difference current would be zero. Figure 10 shows the standard deviation of the difference current divided by 2.5 μA (in percents) for two different combinations of the sizes of the analog transistors. The upper curve was simulated with the transistor $M4$ and the combination of $M5$ and $M6$ sized to 0.75/6, while the PMOS transistors were sized to 1/4 (sizes are in micrometers). The corresponding transistor sizes for the lower curve were 1/8 and 1.5/6, respectively. The standard deviations were determined from 50 Monte Carlo iterations. If the intensity of the pixel would be represented with currents ranging from zero to 2.5 μA, the standard deviation would be around one percent with the larger transistors (see Table 3).

8. LBP PROCESSING PERFORMANCE

Embedding the LBP sampling with the recognition system allows a compact integrated solution for face recognition, with either a CNN-UM or a dedicated massively parallel hardware. The sampling of LBPs is relatively fast even without a massively parallel hardware [7] and, in a practical view, the

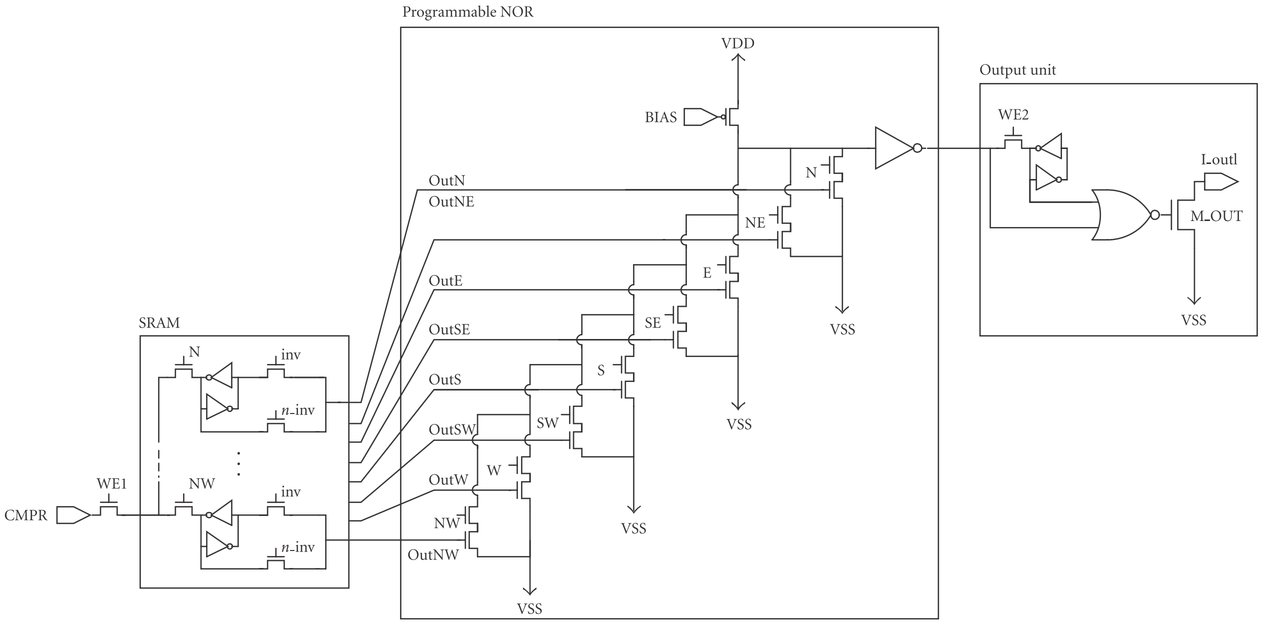


Figure 9: LBP matching unit.

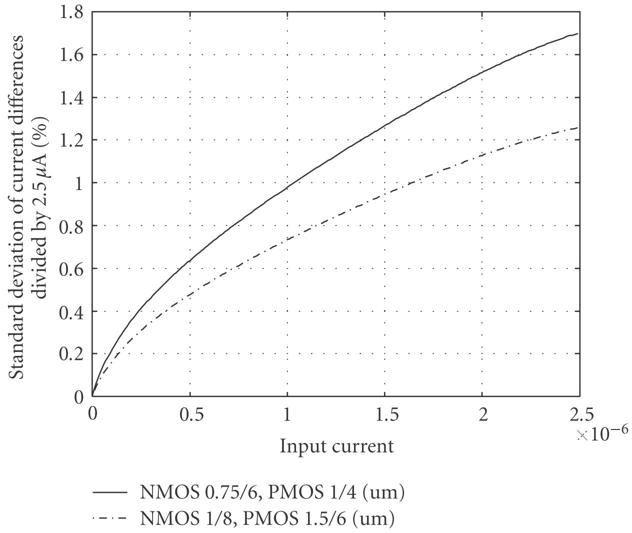


Figure 10: Effect of mismatch on the sampling accuracy.

accuracy improvements in the occurrence map-based face matching are more significant than the improvement in the sampling phase, which however, could be taken as an advantage in other applications.

8.1. Performance of the CNN-UM histogramming

Using the template analyzer [26] the estimated execution time for each of the comparison templates is 2.00τ and 2.88τ for pattern matching templates (τ being the CNN time constant). Excluding the memory delays and template programming delays, the total time used by the threshold templates is 16τ (assuming a sample number of eight). If all the 256 patterns are to be extracted, the total time for pattern matching operations will become 737.3τ $(2.88 * 256\tau)$ assuming that all the blocks are processed in a single frame. However, it was mentioned that it is possible to use a "do not care" condition to reduce the amount of pattern matching operations when, for example, only uniform and symmetrical patterns are used which will further reduce this time.

If the input image is read directly from in-cell sensors, early A/D conversion can be avoided. Assuming that there is a total of 30 symmetrical uniform patterns corresponding to bins and that the image is divided into blocks as proposed, the total amount of successive A/D conversions will be 30 if it is assumed that there is a peripheral A/D converter for each block. The D/A conversion is implemented automatically by summing of binary unit currents from the cells. The integration of the A/D converters will not cause problems since we have shown that the accuracy requirements for the converters are not severe (see Table 4). If the time for a single A/D conversion is 0.5 μs, the total time for conversion will be 15 μs.

By using τ of 1 μs, the total time for LBP sampling becomes 768.3 μs (1302 faces/s). As a reference, for the same images that we used in the simulations, the LBP feature vector extraction time for one face image for AMD Athlon 1800 MHz was 3.49 ms (285 faces/s) with weighting (blocks outside face are ignored) and 4.14 ms (240 faces/s) without weighting [7].

8.2. Performance of the dedicated mixed-mode algorithm

Using the read-out scheme proposed in [24], there will be a need for executing $D * M$ A/D conversions (in parallel for each block) at each comparison where D is the number of occurrence maps and M is the number of search directions. Let us assume the elapsed time for the parallel A/D conversions being approximately 0.5 microseond. With a reference image database of 100 images, this results the elapsed time of 12 ms for comparing a single input image with all the stored reference images (assuming D of 30 and M of 8). The total comparison time for one stored reference image would be about 120 μs. This could be further improved by searching the block specific optimal dilation direction only for certain patterns and calculating the comparison result then for the best direction estimate. In [27] it was demonstrated that with the binary programmable CNN, the measured propagation time (for dilation) was short, only 16.3 ns with 3.3 μA being unit current and 0.7 V being operating voltage.

When estimating the overall performance of the system, the external (off-chip) memory speed and bandwidth need to be considered. We propose using 40 data pins from the chip to interface with the memory. The on-chip memory interface encodes this data to a total of 130 lines which are routed to the cell array. The memory structure of the cell array consists of row and column decoders, which are used to select one row at a time from the 130 × 150 array for writing. If the external memory would operate with a 7 MHz off-chip clock, the total bandwidth through the 40 pins becomes 280 Mbits/s. With a 100 image, database, this would result 0.035 s (29 fps) elapsed when scanning through the whole memory. For a single image the reading time from the off-chip memory would become about 348 μs. Since a cache is used, the occurrence map matching can be performed simultaneously with the memory read operations. The on-chip clock of 15 MHz is enough for synchronizing the 130 lines routed to a certain array row through the column decoder. This would result an elapsed time of 0.03 s for processing through the whole external memory of 100 faces. For a single face the reading time from cache would be about 300 μs. As a conclusion, the system could operate beyond 25 recognized faces per second with a 100 image reference database. If the database was larger, say 1000 faces, it can be estimated that the time used for recognizing a face would be about 0.35 s.

9. DISCUSSION

Power supply, clock generation, and user interface as well as possible camera and camera interface are needed for the dedicated face recognition system to function. Integrating the image sensors will reduce power consumption, since direct A/D conversion is not needed in the imager. Furthermore, the read-out of face matching results can be performed in parallel with a dedicated hardware for each block with small unit currents, which should be more efficient compared to, for example, an FPGA implementation. Whether an FPGA, a CNN-UM, or a dedicated hardware gives the best performance depends on the targeted application and it depends on the final application whether the improvement on the recognition accuracy is worth implementing a special hardware, or whether to use a general purpose computer hardware with standard LBP histogramming. For applying the occurrence map algorithm, a massively parallel hardware seems the only practical alternative.

The occurrence map representation of LBPs with the capability for adaptive block search could also be benefited in other applications beyond face recognition. In LBP-based motion analysis [20], the occurrence maps would be an elegant way to search, for example, motion specific to a certain direction from the image scene.

In the following, we address two different scenarios where the system could be used as such or with some external hardware and software. First, controlling of people transpassing through a security check. In this scenario the recognition time for one face could be while waiting, say seconds, and the database of stored faces would be larger, say 1000 images. A separate face detection and normalization system would not necessarily be required. In another scenario, an individual among a mass of people would be searched without necessarily informing the individual of the surveillance. Then a separate face detection and normalization system would be needed.

10. CONCLUSIONS

This paper described a framework for implementing a massively parallel face recognition system. The system consists of three parts, which are mixed-mode or CNN-UM-based LBP sampling method, a massively parallel face recognition algorithm, and a dedicated mixed-mode hardware for the proposed algorithm. The LBP sampling process of CNN-UM or a dedicated hardware utilizing current-mode operation can be generalized also for other high-speed LBP applications beyond the face recognition. An example of a potential application where the sampling speed would be important is industrial paper quality inspection using computer vision and local binary patterns [28].

The adaptive face matching algorithm could also be implemented as an embedded FPGA implementation. The implementation of the neighborhood logic unit, memory code/decode unit, and instruction code/decode unit, which has been so far considered only in the architectural level, would then be relatively straightforward since digital logic design could be used. Also, the mismatch would not cause any significant decrease to the recognition accuracy if the LBP sampling was performed digitally.

The dedicated massively parallel face recognition algorithm was shown to perform accurate face recognition, with a maximum increase in the recognition accuracy with weighting of 4% compared to [7] and 3% compared to [23]. The improvement in the recognition accuracy without weighting was as much as 14.0% compared to [7]. Furthermore, the face recognition algorithm is adaptively tolerant to slight changes in face orientation and scaling.

ACKNOWLEDGMENT

This work was funded by the Academy of Finland Projects no. 205746 and no. 106451.

REFERENCES

[1] M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of Cognitive Neuroscience*, vol. 3, no. 1, pp. 71–86, 1991.

[2] A. M. Martinez and A. C. Kak, "PCA versus LDA," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 2, pp. 228–233, 2001.

[3] J. Yang, D. Zhang, A. F. Frangi, and J.-Y. Yang, "Two-dimensional PCA: a new approach to appearance-based face representation and recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 1, pp. 131–137, 2004.

[4] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 121–167, 1998.

[5] L. Wiskott, J.-M. Fellous, N. Krüger, and C. D. von Malsburg, "Face recognition by elastic bunch graph matching," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 775–779, 1997.

[6] J. Ruiz-del-Solar and P. Navarrete, "Eigenspace-based face recognition: a comparative study of different approaches," *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, vol. 35, no. 3, pp. 315–325, 2005.

[7] T. Ahonen, A. Hadid, and M. Pietikäinen, "Face recognition with local binary patterns," in *Proceedings of 8th European Conference on Computer Vision (ECCV '04)*, vol. 3021 of *Lecture Notes in Computer Science*, pp. 469–481, Prague, Czech Republic, May 2004.

[8] D. S. Bolme, J. R. Beveridge, M. Teixeira, and B. A. Draper, "The CSU face identification evaluation system: its purpose, features, and structure," in *Proceedings of 3rd International Conference on Computer Vision Systems (ICVS '03)*, pp. 304–313, Graz, Austria, April 2003.

[9] P. J. Phillips, H. Wechsler, J. Huang, and P. J. Rauss, "The FERET database and evaluation procedure for face-recognition algorithms," *Image and Vision Computing*, vol. 16, no. 5, pp. 295–306, 1998.

[10] L. O. Chua and L. Yang, "Cellular neural networks: theory," *IEEE Transactions on Circuits and Systems*, vol. 35, no. 10, pp. 1257–1272, 1988.

[11] T. Roska and L. O. Chua, "The CNN universal machine: an analogic array computer," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 40, no. 3, pp. 163–173, 1993.

[12] G. L. Cembrano, A. Rodríguez-Vázquez, R. C. Galán, F. Jiménez-Garrido, S. Espejo, and R. Domínguez-Castro, "A 1000 FPS at 128 × 128 vision processor with 8-bit digitized I/O," *IEEE Journal of Solid-State Circuits*, vol. 39, no. 7, pp. 1044–1055, 2004.

[13] A. Paasio, A. Kananen, K. Halonen, and V. Porra, "A QCIF resolution binary I/O CNN-UM chip," *Journal of VLSI Signal Processing Systems for Signal, Image, and Video Technology*, vol. 23, no. 2, pp. 281–290, 1999.

[14] O. Lahdenoja, M. Laiho, and A. Paasio, "Local binary pattern feature vector extraction with CNN," in *Proceedings of the 9th IEEE International Workshop on Cellular Neural Networks and Their Applications (CNNA '05)*, pp. 202–205, Hsinchu, Tawian, May 2005.

[15] O. Lahdenoja, J. Maunu, M. Laiho, and A. Paasio, "A massively parallel algorithm for local binary pattern based face recognition," in *Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS '06)*, p. 4, Kos, Greece, May 2006.

[16] M. Laiho, O. Lahdenoja, and A. Paasio, "Dedicated hardware for parallel extraction of local binary pattern feature vectors," in *Proceedings of the 9th IEEE International Workshop on Cellular Neural Networks and Their Applications (CNNA '05)*, pp. 27–30, Hsinchu, Tawian, May 2005.

[17] T. Ojala, M. Pietikäinen, and D. Harwood, "A comparative study of texture measures with classification based on feature distributions," *Pattern Recognition*, vol. 29, no. 1, pp. 51–59, 1996.

[18] T. Mäenpää, "The local binary pattern approach to texture analysis—extensions and applications," Dissertation, University of Oulu, Oulu, Finland, 2003.

[19] T. Ojala, M. Pietikäinen, and T. Mäenpää, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 971–987, 2002.

[20] M. Heikkilä and M. Pietikäinen, "A texture-based method for modeling the background and detecting moving objects," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 4, pp. 657–662, 2006.

[21] A. Hadid, M. Pietikäinen, and T. Ahonen, "A discriminative feature space for detecting and recognizing faces," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '04)*, vol. 2, pp. 797–804, Washington, DC, USA, June-July 2004.

[22] T. Mäenpää, T. Ojala, M. Pietikäinen, and M. Soriano, "Robust texture classification by subsets of local binary patterns," in *Proceedings of 15th International Conference on Pattern Recognition (ICPR '00)*, vol. 3, pp. 935–938, Barcelona, Spain, September 2000.

[23] O. Lahdenoja, M. Laiho, and A. Paasio, "Reducing the feature vector length in local binary pattern based face recognition," in *Proceedings of IEEE International Conference on Image Processing (ICIP '05)*, vol. 2, pp. 914–917, Genova, Italy, September 2005.

[24] P. Dudek, "A flexible global readout architecture for an analogue SIMD vision chip," in *Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS '03)*, vol. 3, pp. 782–785, Bangkok, Thailand, May 2003.

[25] CNN Software Library, Ver. 1.1, Chapter 3, http://lab.analogic.sztaki.hu.

[26] CNN Template Analysator, TemInfo, http://lab.analogic.sztaki.hu.

[27] J. Flak, M. Laiho, A. Paasio, and K. Halonen, "VLSI implementation of a binary CNN: first measurement result," in *Proceedings of the 8th IEEE International Workshop on Cellular Neural Networks and Their Applications (CNNA '04)*, p. 129, Budapest, Hungary, July 2004.

[28] M. Turtinen, M. Pietikäinen, and O. Silvén, "Visual characterization of paper using isomap and local binary patterns," *IEICE Transactions on Information and Systems*, vol. E89-D, no. 7, pp. 2076–2083, 2006.

ARTICLE 5

O. Lahdenoja, M. Laiho, A. Paasio, On the spatial distribution of local non-parametric facial shape descriptors, 16th Scandinavian Conference on Image Analysis, pp. 351-358, 2009.

ARTICLE 6

O. Lahdenoja, M. Laiho, Regional image correspondence matching method for SIMD processing, 19th European Conference on Circuit Theory and Design, pp. 802-805, 2009.

ARTICLE 7

O. Lahdenoja, J. Poikonen, M. Laiho, Extracting Local Binary Patterns with MIPA4k vision processor, 12th International Workshop on Cellular Nanoscale Networks and Their Applications, 2010.

ARTICLE 8

O. Lahdenoja, J. Poikonen, M. Laiho, Towards Understanding the Formation of Uniform Local Binary Patterns, ISRN Machine Vision, Article ID 429347, 2013.

Hindawi Publishing Corporation
ISRN Machine Vision
Volume 2013, Article ID 429347, 20 pages
http://dx.doi.org/10.1155/2013/429347

Research Article

Towards Understanding the Formation of Uniform Local Binary Patterns

Olli Lahdenoja, Jonne Poikonen, and Mika Laiho

Business and Innovation Development BID Technology, University of Turku, 20014 Turku, Finland

Correspondence should be addressed to Olli Lahdenoja; olanla@utu.fi

Received 29 May 2013; Accepted 30 June 2013

Academic Editors: O. Ghita and N. A. Schmid

The research reported in this paper focuses on the modeling of Local Binary Patterns (LBPs) and presents an a priori model where LBPs are considered as combinations of permutations. The aim is to increase the understanding of the mechanisms related to the formation of uniform LBPs. Uniform patterns are known to exhibit high discriminative capability; however, so far the reasons for this have not been fully explored. We report an observation that although the overall a priori probability of uniform LBPs is high, it is mostly due to the high probability of only certain classes of patterns, while the a priori probability of other patterns is very low. In order to examine this behavior, the relationship between the runs up and down test for randomness of permutations and the uniform LBPs was studied. Quantitative experiments were then carried out to show that the relative effect of uniform patterns to the LBP histogram is strengthened with deterministic data, in comparison with the i.i.d. model. This was verified by using an a priori model as well as through experiments with natural image data. It was further illustrated that specific uniform LBP codes can also provide responses to salient shapes, that is, to monotonically changing intensity functions and edges within the image microstructure.

1. Introduction

The Local Binary Pattern (LBP) methodology [1] was first proposed as a texture descriptor, but it has later been applied to various other fields of computer vision: for example, face recognition, facial expression recognition, modeling motion and actions, as well as medical image analysis. Numerous modifications and improvements have been suggested to the original LBP methodology for various applications, while the LBPs have also been proposed for signal processing tasks beyond image processing (e.g., [2]). A detailed list of various applications and papers related to the LBP methodology is available in CMV Oulu pages [3].

Before the introduction of Local Binary Patterns, co-occurrence statistics descriptors based on binary features and n-tuples [4], as well as the texture unit and texture spectrum (TUTS) method [5], have been studied. N-tuples have been studied in, for example, [4, 6] for texture retrieval. It was discovered that the distribution of individual n-tuples could not reach the classification accuracy of quantized binary features such as BTCS [4].

The possibility of using only uniform and rotation invariant binary patterns distinguishes the Local Binary Pattern methodology from its predecessors, because it enables a more compact image representation. It has been widely accepted that uniform LBPs, which contain at most two circular 0-1 or 1-0 transitions, are highly applicable and thus have been frequently used in various applications—not only in texture analysis. While many modifications to the original LBP have been proposed, most image analysis applications still take advantage of a combination of LBP and uniform patterns, despite other modifications in sampling, such as applying Gabor filtering as a preprocessing step [7]. However, it has been unclear how these particular uniform patterns contribute to increasing the discriminative capabilities of the LBPs. It was shown in [8] that uniform patterns are a priori very frequent even with random data. The observations from the existing research raise naturally the question "Why are uniform patterns so discriminative?". It was also shown in [8] that the percentage of uniform patterns further increases with natural image data compared to a priori model. In this paper, some of the mechanisms related to this increase in occurrence probability will be addressed.

LBPs are represented in this paper as compositions of individual n-tuples, that is, permutations. We denote the set of all possible permutations as the "permutation space." The permutation model represents a middle ground in terms of the complexity between the image intensity representation and the binary pattern feature representation. The total number of permutations in the permutation space is smaller than the number of instances in the intensity space but higher than the amount of instances in the binary pattern space. The permutation based approach is applied here both through an a priori model and through experiments with natural images to examine the particularly discriminative quality of uniform LBPs. The aim of this study is also to better understand the relationship between uniform LBPs and the properties of deterministic nonrandom image data.

This paper is composed of eight sections. Section 2 contains an introduction to LBP methodology as well as background and related work. Section 3 defines the permutation space used and a priori probability model for the uniform patterns. In Section 4, a modification to the original permutation space is defined for modeling purposes, while in Section 5 the previously defined concepts are used to analyze the uniformity of Local Binary Patterns. In Section 6 qualitative and quantitative experiments with a priori model and natural image data are performed. Sections 7 and 8 provide further discussion and conclusions.

2. Background and Related Work

2.1. Derivation of Local Binary Patterns. A Local Binary Pattern is derived for a specific pixel neighborhood radius r by comparing the intensities of M discrete circular sample points to the intensity of the center pixel (clockwise or counterclockwise), starting from a certain angle. The comparison determines whether the corresponding location in the Local Binary Pattern of length M is 1 or 0. A value 1 is assigned if the center pixel intensity is smaller than the sample pixel intensity and 0 otherwise. Sample number $M = 8$ is the most commonly used, with circle radius $r = 1$; however, also other values for the radius and sample numbers can be used. If a sample point is located between pixels, the intensity value used for the comparison can be determined by *bilinear interpolation* (see Figure 1). Using this sampling procedure, sweeping over the whole image is denoted by $\mathrm{LBP}(M, r)$ [9, 10].

After the LBP extraction, each pixel in an image is replaced by a binary pattern, except at the borders of the image where all of the neighbour values do not exist. The *feature vector* of an image then consists of a histogram of the pixel LBPs. The initial length of the histogram is 2^M since each possible LBP is assigned a separate bin. If there are N regions in an image (e.g., a normalized face image could be divided into 7×7 blocks for enhancing the spatial accuracy of the histograms [9]), the histograms can be combined into a single histogram with a length of $N \cdot 2^M$.

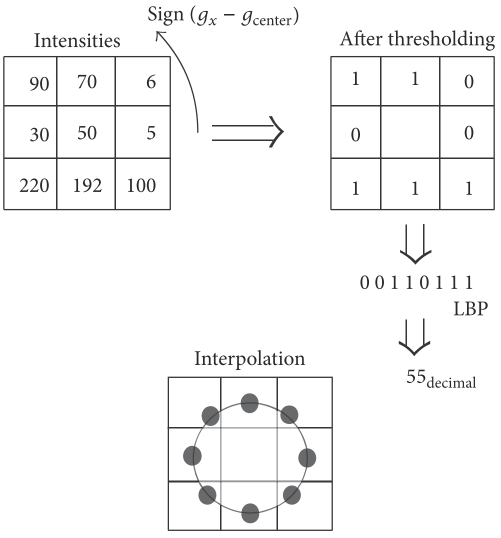


Figure 1: Derivation of Local Binary Patterns.

2.2. Uniform LBPs. Uniform Local Binary Patterns are patterns with at most two circular 0-1 and 1-0 transitions. For example, patterns 00111000, 11111111, 00000000, and 11011111 are uniform, and patterns 01010000, 01001110, or 10101100 are not uniform. Selecting only uniform patterns contributes to both reducing the length of the feature vector (LBP histogram) and improving the performance of classifiers using the LBP features (see [1, 9–11]). Uniform LBPs can also be applied to obtain rotation invariance [10]. In [12–14] global rotation invariance for the LBPs was achieved by applying a Discrete Fourier Transform to the uniform bins of the LBP histograms. In [13], the rotation invariance of LBP variants was also analyzed.

There are several methods for performing LBP histogram comparison. These include histogram intersection, log-likelihood statistics, and Chi-square statistics [9]. It is also possible to use multiple LBPs simultaneously, also with different radiuses, to describe a certain image location. The natural disadvantage of this is the further increase in the length of the feature vector.

2.3. Related Research. The use of uniform Local Binary Patterns was proposed in [15] as a way to reduce the high dimension of the original LBP feature vector. The use of uniform patterns can be seen as a *filter type* feature selection method [16], since it is related directly to the image data. Also, a *beam search* method was proposed in [15] using feedback from a classifier (for texture images), extending to a *wrapper type* feature selection [16]. Later, numerous other filter and wrapper type methods have been proposed for LBP feature selection, including Fisher separation criterion (FSC) based learning, Boosting (AdaBoost), LDA, and PCA; see [1] for a complete description. Face recognition has typically been used as a benchmark application. In [11] a machine learning approach was chosen to study which individual LBP bins were most discriminative in facial expression recognition. A boosting classifier was used, and 91.9% of the most discriminative patterns turned out to be uniform. However, it has been unclear how these particular uniform patterns contribute to increasing the discriminative capabilities of the LBP methodology.

In this paper, the space of all $n!$ possible n-tuples is constructed, and its relation to individual uniform and nonuniform LBP codes and to the class of all uniform LBP codes is modeled. Hence, we propose forming the LBPs as a result of an intermediate nonlinear rank ordering operation in order to facilitate the understanding of the Local Binary Patterns. Rank ordering and the census transform were introduced in [17] as nonparametric descriptors. Since then, many variants of descriptors based on ordinal intensity representation have been proposed. Recently, the local intensity order pattern (LIOP) descriptor showed excellent performance [18] in keypoint matching. It was further developed in [19]. In LIOP the numbers of occurrences of n-tuples among local patches are selected as histogram bins in a rotation invariant manner. The methods proposed in this paper can also support the design of new descriptors and promote understanding of the existing descriptors based on n-tuple processing, for example, [18, 19].

LBPs have also been described as vector quantized responses to linear filters [20]. This allows the analysis of the properties of the LBP operator and the modeling of its relationship to other filter bank based descriptors. The focus of this paper is also to provide an alternative approach to the filter bank based LBP decomposition [20] and to suggest an alternative approach to the previous work in [8] for analyzing the formation of uniform patterns.

In [8] the a priori distribution of uniform LBPs was studied, and it was observed that their a priori probability is rather high also with independent identically distributed (i.i.d) data. This indicates that these patterns do not necessarily relate only to image structures such as small edges, corners, and line-ends, as was previously thought, but also to the LBP sampling process itself. The percentage of uniform patterns has been also shown to further increase from the estimated a priori probability [8] in applications using natural image data. The exact distribution of LBPs was studied by calculating the volume of multidimensional polytopes in [8]. It has also been shown that the minimum between the total number of zeros and ones in a LBP can be used to uniquely characterize the occurrence probability of the LBPs with i.i.d. data [21]. However, in [21] the a priori probabilities were not linked to the occurrence frequencies of uniform patterns. In both studies [8, 21], a link between information theory and a priori occurrence probabilities of the LBPs was speculated.

In [8] the LBPs were modeled using a space partitioning approach, where the pixel intensities (see Definition 1) were mapped into LBP binary pattern space. In practice this means that, for example, for the LBP(8, 1) operator the dimension of the intensity space is 256^9, for 8-bit pixels, and a certain location of an image (consisting of a set of intensities) would represent an individual point in this space. This particular intensity set could then be further mapped into the LBP space.

Definition 1. The intensity space $\mathbf{I}$ used in derivation of Local Binary Patterns consists of sets of instances in space $\{I_1, I_2, I_3, \ldots, I_M\}$ ordered circularly around the center in addition to the center point $\{I_c\}$. Its dimension is $I_{\mathrm{RANGE}}^{(M+1)}$, where I_{RANGE} represents the range of the intensities.

The a priori probabilities of LBPs in the case of continuously distributed i.i.d variables (as intensities) were considered already in [21]. The a priori probability of individual LBPs (for i.i.d data distribution without interpolation) is completely determined by its descriptor k (see Definition 2) according to (1), following the binomial distribution [8, 21].

Definition 2. The descriptor k [8, 21, 22] for a Local Binary Pattern is calculated as the minimum between the cardinalities of the sets consisting of 1 bits and 0 bits.

For example, the descriptor k for pattern 00110011 is four, for pattern 00000000 zero, and for pattern 01010100 three.

Theorem 3. *The probability of an LBP with M contour samples ($M \geq 3$) to occur with continuously distributed i.i.d. data [8, 21], without considering interpolation, given descriptor k is*

$$P_{LBP}(M, k) = \frac{k!\,(M-k)!}{(M+1)!}. \tag{1}$$

3. Constructing the Permutation Space

3.1. Definition of the Permutation Space. We propose adding a “mid-space” between the intensity space and the LBP space, the permutation space illustrated in Figure 2. The concepts of root permutations and child LBPs in Figure 2 will be explained in the later sections. This provides an alternative approach to [8] in modeling the uniform patterns and allows modeling some of the fundamental differences between n-tuple and LBP based approaches for low level image representation [1, 18, 19]. An LBP is here modeled to consist of multiple instances of *unit permutations* located among the permutation space. The a priori probability of each unit permutation is equal with i.i.d. data, which is readily well known within nonparametric statistics [23].

Definition 4. Let the intensity space $\mathbf{I}$ be defined as sets consisting of intensities $\{I_1, I_2, I_3, \ldots, I_M\}$ around the center in addition to the center point $\{I_c\}$ within a local LBP neighborhood. A unit permutation is defined as an individual permutation $\{R_1, R_2, R_3, \ldots, R_M\}$ around the center in addition to the center point rank $\{R_c\}$ formed by rank ordering the intensity samples I_n as R_n so that the smallest intensity is assigned a rank of 1. In the case of tied intensities, the ranks of the tied intensities are taken from an i.i.d distribution. The length of the unit permutation is then $(M + 1)$, where the number of contour samples in the corresponding LBP is equal to M.

The permutation space $\mathbf{P}$ contains all possible unit permutations for the circular neighborhood. It can be derived from the intensity space so that the number of instances is reduced (since, in practice $M \ll I_{\mathrm{RANGE}}$). The number of

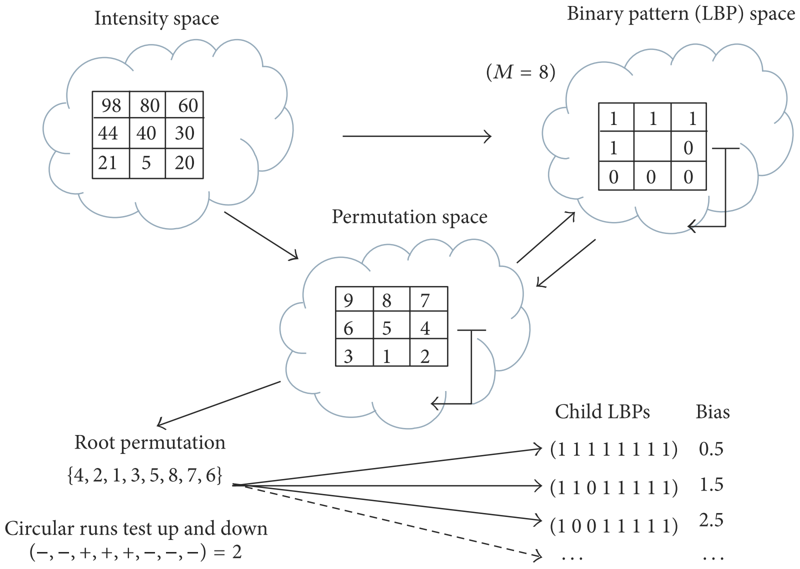


Figure 2: The concept of intensity space, permutation space, and binary pattern space.

instances in the permutation space is $(M + 1)!$. The binary pattern space $\mathbf{b}$ consists of instances of sets composed of bits $\{b_1, b_2, b_3, \ldots b_M\}$ excluding the center, for M being the number of contour samples among the LBP, respectively. The dimension of the binary pattern space is 2^M. The binary pattern space $\mathbf{b}$ can be produced directly from the high-dimensional intensity space as in [8]. It can also be derived indirectly through the permutation space consisting of reduced number of instances. Each individual LBP can also be projected back to the permutation space from the binary pattern space into a collection of unit permutations (see Figure 2).

In this paper, when considering the a priori probabilities, interpolation is not used. However, its effect to the final LBP distribution is evaluated in the experimental section using natural image data. Interpolation in the context of studying the uniform patterns has been considered more in depth in [8]. In general, the number of uniform patterns has shown to grow with interpolation [8], due to increased dependency and correlation between the neighboring sample points as they are averaged using bilinear weighting from their four neighbors.

3.2. Local Binary Pattern Operator for Permutations and Reverse Mapping. The mapping operator ϕ_{MAP} between the intensity space and the LBP space and between the permutation space and the LBP space is defined in the following. The mapping operator ϕ_{MAP} can be used for both permutation space and intensity space to derive an LBP. In the case of the permutation space, instead of intensities, the magnitudes of the ranks are considered. The mapping from an instance of the LBP space (a particular LBP code) into the permutation space (ϕ_{MAP}^{-1}) is defined indirectly as forming the set of all unit permutations which result in this particular LBP according to the ϕ_{MAP} operator.

Definition 5. (a) The LBP mapping operator ϕ_{MAP} is defined between the instances of spaces $\mathbf{I_n} \Rightarrow \mathbf{b_n}$, or $\mathbf{P_n} \Rightarrow \mathbf{b_n}$ as 1 for instances which have a magnitude greater or equal than the center and 0 for instances which have magnitude smaller than the center. In the case of mapping $\mathbf{P_n} \Rightarrow \mathbf{b_n}$ and ties between the center intensity $\{I_c\}$ and contour intensities $\mathbf{I_n}$, the rank of the center $\{R_c\}$ is assigned the minimum within the combined set of the tied ranks $\{R_c, R_n\}$. This preserves the uniformity of the LBP also when the intermediate permutation space is used. The resulting LBP code is a concatenation $\{b_1, b_2, b_3, \ldots b_M\}$ of the individual bits.

(b) A mapping from intensity space to binary pattern space is defined by applying the LBP mapping operator ϕ_{MAP} for intensity set $\mathbf{I_n}$ resulting in binary pattern $\mathbf{b_n}$.

(c) A mapping from an instance of intensity space $\mathbf{I_n}$ to an instance of permutation space $\mathbf{P_n}$ is defined as $\mathbf{R}(\mathbf{I_n})$, where an operator $\mathbf{R}$ extracts the rank ordering of intensity samples among the intensity set $\mathbf{I_n}$ so that the smallest element will be assigned to value 1.

(d) A mapping from the permutation space into the binary pattern space is defined by applying the LBP mapping operator ϕ_{MAP} to the set of ranks $\mathbf{P_n}$ resulting in a binary pattern $\mathbf{b_n}$ among the binary pattern space.

(e) A reverse mapping ϕ_{MAP}^{-1} from an instance of binary pattern space $\mathbf{b_n}$ into permutation space $\mathbf{P_n}$ is defined indirectly as forming all the $\mathbf{P_n}$ elements according to the criteria ϕ_{MAP} results in a match (from all elements of $\mathbf{P_n} \Rightarrow \mathbf{b_n}$).

As an example, consider an arbitrary Local Binary Pattern, for example, $M = 8$ pattern 00110011. It is a result of applying the ϕ_{MAP} operator to a unit permutation, where the rank of the center pixel (ordinal value) is always 5 (rank of smallest being 1). An example of a permutation which could produce this particular LBP could be {5} for center pixel and {1, 2, 9, 8, 3, 4, 6, 7} for the other pixels.

This is not the only possible unit permutation for this particular LBP. The degree of freedom related to the unit permutations for a given LBP is determined by the following elements: the rank of the center pixel, the number of bits above the center pixel, that is, number of 1s in LBP code, and number of bits below the center, that is, number of 0's. The a priori probability of occurrence for an individual unit permutation is then constant $1/(M+1)!$, with i.i.d. data, where $M + 1$ is the length of the unit permutation and M is the number of contour samples in an LBP. Hence, the intensity space is divided into larger fractions of equal unit probability, still allowing the derivation of Local Binary Patterns.

In the case of LBP 00110011, the number of unit permutations invoked by the restriction "four locations above the center" is 4!, and the number of unit permutations invoked by the restriction "four locations below the center" is also 4!. As a consequence, the total occurrence probability of the LBP under consideration becomes $4! * 4!/[(8 + 1)!]$, from which $1/[8+1]!$ is assigned for each of the unit permutations. As another example, consider the LBP with $M = 5$, 01000, containing, for example, the unit permutation $\{5\}$ center (since four zeros are below it), $\{1, 6, 2, 3, 4\}$ contour, which results in a total cumulative probability of $1! * 4!/[(5 + 1)!]$ for all the unit permutations.

3.3. Modeling a Priori Distribution of Uniform Patterns Only with i.i.d. Data. Next we consider the total occurrence probability of all uniform patterns with i.i.d. data with respect to all LBPs. The number of uniform patterns with respect to descriptor k is described completely in (2) and (3) with respect to M and descriptor k. For (2) (even M) k varies between 0 and $M/2$, and for (3) (odd M) k varies between 0 and $(M - 1)/2$.

$$\#\text{uniform, even } M = \begin{cases} 2 & k = 0, \\ 2 * M & 0 < k < M/2, \\ M & k = M/2. \end{cases} \tag{2}$$

$$\#\text{uniform, odd } M = \begin{cases} 2 & k = 0, \\ 2 * M & 0 < k <= (M-1)/2. \end{cases} \tag{3}$$

For example, consider an arbitrary LBP with $M = 4$. The case of $k = 0$ consists of uniform patterns 0000 and 1111. When $k = 1$, the uniform patterns are 0001, 0010, 0100, 1000, and the inversions of these patterns. In these cases all patterns are uniform. Descriptor value $k = 2$ (i.e. $M/2$) leads to the uniform patterns of 1100, 0110, 0011, and 1001. The other nonuniform patterns for the highest possible k value for $M = 4$ are 0101 and 1010.

With LBPs having $M \geq 4$ the total occurrence probability of the set of all uniform patterns, for i.i.d. data distribution and even M is,

$$P_{u2}^{\text{All}}(M) = \frac{2}{(M+1)} + \frac{2M}{(M+1)!} \sum_{k=1}^{M/2-1} k!\,(M-k)! + \frac{M}{(M+1)!} \left[\left(\frac{M}{2} \right)! \right]^2. \tag{4}$$

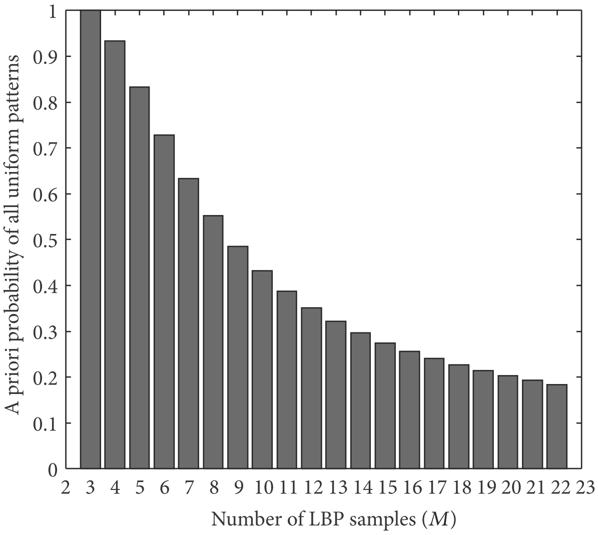


Figure 3: The total a priori probability of all uniform patterns with LBP sample number M according to (4) (for odd M the summation in (4) was adjusted according to (3)).

The first term of (4) includes the a priori probability of all-zero and all-one uniform LBPs, the second term is the sum of probabilities of patterns where k varies between 1 and $M/2-1$ for even M, and the third term includes the highest descriptor k uniform patterns.

According to the permutation concept proposed here, the number of all unit permutations for uniform patterns becomes then $(M + 1)! * P_{u2}^{\text{All}}$, while the total number of all permutations for modeling LBPs with sample number M becomes $(M+1)!$. In Figure 3 the total occurrence probability of uniform patterns is plotted with respect to the LBP sample number M using the (4) (for odd M the summation in (4) was adjusted according to (3)).

4. Modified Permutation Space

To analyze the formation of LBPs and uniform patterns in particular, we propose a modification to the permutation space by removing the center rank of a unit permutation. This requires the definitions of *an intermediate unit permutation*, *a root permutation*, and *an intermediate root permutation*.

A certain LBP can be composed of multiple unit permutations, but if an instance of the intensity space set is mapped to the LBP binary pattern space, a single unique intermediate unit permutation can be assigned to the permutation space and it can be uniquely used for determining the resulting LBP (neglecting ties).

Definition 6. An intermediate unit permutation is defined as a unit permutation $I_n \Rightarrow P_n \Rightarrow b_n$ as the permutation P_n. The directions of the arrows describe the order in which the rank ordering (R) and mapping (ϕ_{MAP}) are performed.

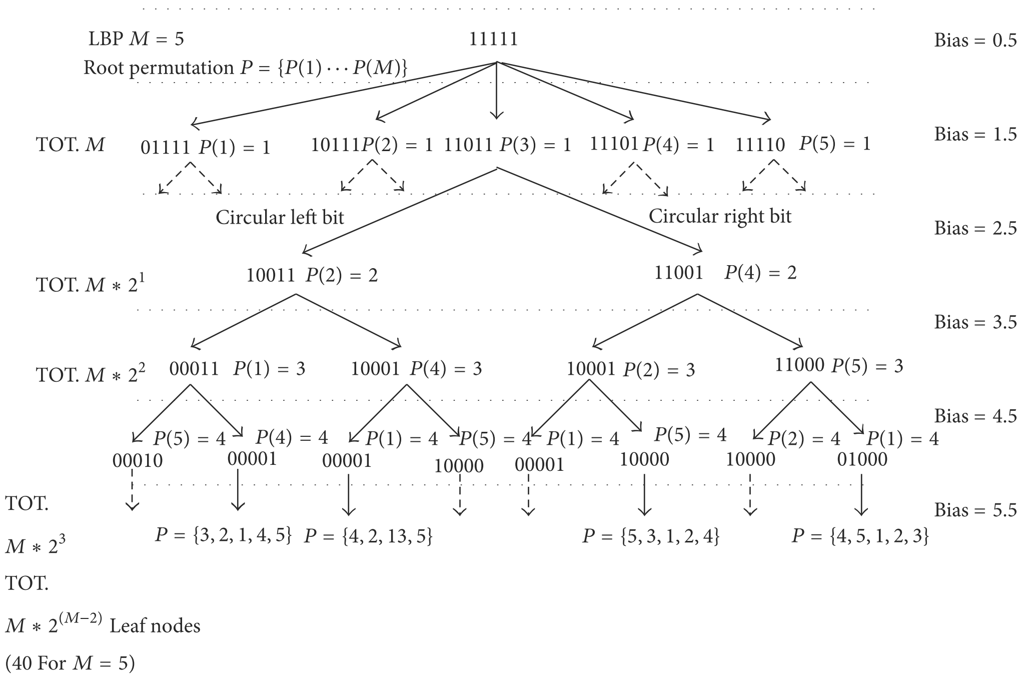


FIGURE 4: An example of forming the LBP $M = 5$ complete uniform root permutations.

TABLE 1: Root permutations.

Case	(A)	(B)	(C)
Permutations			
Center	9	4	9
Contour	3, 8, 2, 5, 4, 7, 1, 6	5, 8, 6, 9, 1, 2, 3, 7	2, 3, 4, 5, 6, 7, 8, 1
Root permutations			
Contour	3, 8, 2, 5, 4, 7, 1, 6	4, 7, 5, 8, 1, 2, 3, 6	2, 3, 4, 5, 6, 7, 8, 1
Child LBPs (Bias)			
0.5	11111111	11111111	11111111
1.5	11111101	11110111	11111110
2.5	11011101	11110011	01111110
3.5	01011101	11110001	00111110
4.5	01010101	01110001	00011110
5.5	01000101	01010001	00001110
6.5	01000100	01010000	00000110
7.5	01000000	00010000	00000010
8.5	00000000	00000000	00000000
Total uniform	4	6	9

Depending on the context, a root permutation may be derived from an intermediate unit permutation, which is simply extracted by rank ordering the intensity set without the center, or in more abstract case it may refer to a normal unit permutation from which the rank of the center pixel is removed (see Definition 7 and Table 1).

Definition 7. A root permutation is a unit permutation produced by removing the rank of the center. The rank of the center pixel is first removed from the unit permutation, and the rank ordering ($\mathbf{R}(\mathbf{P_n})$) is applied again for the magnitudes of the remaining ranks. An intermediate root permutation refers simply to an n-tuple formed by rank ordering the LBP contour sample intensities (without considering the center). The total length of the root permutation or the intermediate root permutation is then $\mathrm{Card}(P_n) - 1$, that is, equal to M.

Two different LBPs or unit permutations can produce the same root permutation. For example, LBPs with $M = 4$, 0000 and 1111 could be composed of instances of permutations $\{5\}$ center, $\{1, 2, 3, 4\}$ contour, and 1 center, $\{2, 3, 4, 5\}$ contour, respectively, the root permutation for both being $\{1, 2, 3, 4\}$. Given an arbitrary LBP, only if the full intensity set or the intermediate unit permutation is known, a certain root permutation can be used to describe that particular LBP. However, it will not be unique, since the information from the center intensity value or rank is missing.

Given a root permutation or an intermediate root permutation, multiple *child* LBPs can be generated if a new center pixel (called *bias*) is assigned. A root permutation is allowed to generate child LBPs by setting up a new bias level (instead of the center pixel). In practice, any bias value between or above the magnitudes of the ranks could be used, but for clarity we use threshold values in the middle of the integer ranks (0.5, 1.5, 2.5, 3.5, etc.). In Table 1, three root permutations are shown, and the child LBPs which can be generated by these root permutations are described by changing the center bias. One of the child LBPs will always represent the original LBP.

Definition 8. A child LBP is an LBP generated from the root permutation or from the intermediate root permutation by comparing successively the bias level to the magnitude of the root permutation ranks (one by one) and assigning bit 1 if the magnitude of the bias level is below the magnitude of the rank permutation and 0 otherwise (in the corresponding location). That is, the LBP mapping operator ϕ_{MAP} is applied to the root permutation by using the set bias as a center.

In Section 6, when performing the experiments, we will use an approach where all possible $M!$ root permutations (corresponding to contour samples of the LBPs) are pregenerated and the bias is then adjusted to each of its $M+1$ possible locations for each root permutation.

4.1. Monotonicity of the Root Permutation. Consider fully ordered (with respect to some circular shift) root permutations of length M containing only one long increasing or decreasing run, in addition to the transitions from the smallest to the largest or from largest to the smallest. We describe next that for these root permutations, all of the child LBPs (generated by changing the center pixel bias) are uniform (See Table 1).

For the lowest and the highest bias value, the resulting child LBPs derived from the root permutation are all uniform since they consist of only all-zero and all-one LBPs. For the second lowest bias and the second highest bias, the resulting patterns are uniform since they include only one instance of one or zero (the descriptor k being one). The remaining bias values (between 2.5 to $M-1.5$) for $M \geq 4$ are considered next.

The first observation is that each fully ordered permutation can be circularly shifted left or right H times in order to make it a monotonically increasing (or decreasing) permutation starting with the lowest (highest) element. It can also be shifted back so that the generated child LBP is not altered. Let us focus only on the length M (longest) up (or down) run while also omitting the circularity. Let $\{\mathbf{B}\}$ be the set of the root permutation instances below the bias value and $\{\mathbf{A}\}$ the set of the root permutation instances above the bias value. It is evident that for the set $\{\mathbf{B}\}$, if the ordering of the permutations is monotonically increasing, then they are fully ordered in increasing order also in $\{\mathbf{A}\}$. The same applies for the decreasing permutations.

When changing the bias by one (increasing or decreasing), the instance where the bit transition occurs, generated by the bias, also shifts by one to the right or to the left. While acknowledging the initial conditions which hold for descriptor k values 0 and 1 (as described before), it is evident that only one bit right next to the transition point can change and it will change to the direction which preserves the uniformity. See, for example, Table 1 Case C. The ordered root permutations are likely to be common with natural image data as pointed out in [4]. Also, statistically their likeliness to occur with random data should be lower than with deterministic data, which will be considered in the next section.

5. On the Relation between Runs Test for Permutations and Uniform Patterns

The simplest tests of randomness for two-valued data (e.g., coin-toss data) are based on estimating the total number of instances of each value or to count the number of successive instances of each [23]. For example, the sequence (T T T T T T T) is not likely to be generated randomly, while the sequence (T F T T F F T T) would be more likely to result from a random process. If there are too many or too few instances of each value, the generating process is not likely to be random (i.i.d.).

The runs up and down test can be applied for numeric data, such as intensities or ranks, to examine the number of monotonically increasing (decreasing) sequences (runs). According to the runs test, monotonicity is the strongest indication of nonrandomness. If the length of the runs is high (few runs within the data), the data is not likely to be generated by a random process. In other words, the hypothesis of randomness is rejected.

Given a root permutation, if the changes from the smallest towards the largest element, or vice versa, occur always next to the set of elements which have previously been changed, all of the child LBPs will be uniform. We denote these root permutations as complete uniform root permutations (see Table 1 case C as an example). Fully ordered root permutations described in the previous section form a subset of complete uniform root permutations. For instance, permutations $\{6, 4, 1, 2, 3, 5\}$, $\{1, 2, 3, 4\}$, and $\{1, 2, 3, 4, 8, 7, 6, 5\}$ are complete uniform root permutations (see also Figure 4).

Definition 9. Complete uniform root permutations are root permutations for which all of the child LBPs are uniform. The number of complete uniform root permutations is given by $M * 2^{M-2}$, $(M \geq 3)$, where M is the number of contour samples in LBP and the length of a root permutation.

The total number of complete uniform root permutations can be examined through the following example: consider the bias changing from its lowest level towards the highest (Figure 4). For the lowest level, only one uniform LBP can be found (of all ones, e.g., 11111 in the case of $M = 5$). For the second lowest bias level, all the child LBPs are also uniform, since only one bit is changed in comparison with the previous bias, and the other bits are 1s. For the following bias levels, uniform patterns will be generated if and only if the successive change among the child LBPs is always next to previous changed bits (either circularly to the left or to the right). For example, for an LBP of 11000111 (indexes $1, \ldots, 8$) the next change to zero could occur only on index location 2 (to the left) or location 6 (to the right) in a circular manner.

The changing bit will also indicate the successive value of the root permutation formed among the path of the permutation tree (see Figure 4). The number of leaf nodes for this tree is then equal to the total number of complete uniform root permutations, which is $M * 2^{M-2}$, $(M \geq 3)$. It consists of M patterns for the second lowest bias and for each of these patterns, a perfect binary tree of height $M-2$.

Definition 10. Runs up and down test result for the root permutations is defined as the total number of successive increasing or decreasing circular runs within a certain root permutation.

It was observed that all of the complete uniform root permutations with $M \leq 10$ matched to category two patterns (n-tuples) according to the circular runs test. However, in this paper the proof of the equivalence of the runs level 2 patterns and the complete uniform root permutations is omitted for $M > 10$. We emphasize that the number of runs according to the runs up and down test for permutations is not the same as the uniformity level of the pattern. The runs test for permutations is a more flexible and general test of randomness, and it can only be applied if the rank order statistics of the successive samples in a pattern are known, since a fixed threshold (bias) is not used as in LBP. The derivation of the rank order statistics is not necessary for extracting the LBPs, since the thresholding according to the center pixel's intensity value determines the pattern uniquely. However, the LBP can still be uniquely determined from the rank permutations. Hence, changing the bias level for generating the child LBPs from root permutations can be seen as a unifying approach between the LBPs and permutation space, where the contribution of the center pixel is adjusted by using the bias.

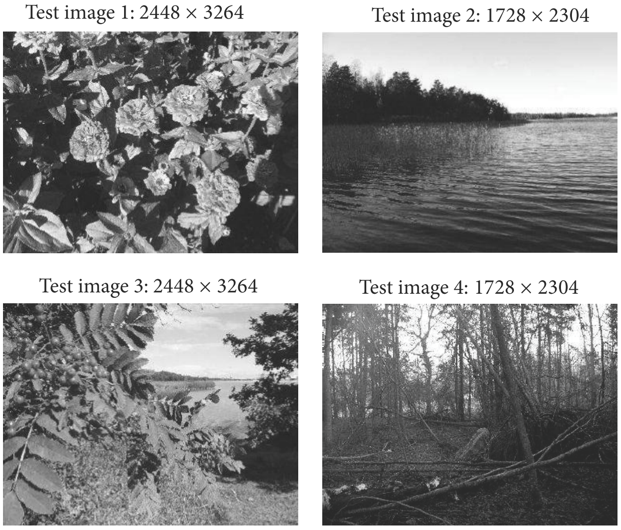

Test image 1: 2448 × 3264
Test image 2: 1728 × 2304
Test image 3: 2448 × 3264
Test image 4: 1728 × 2304

Figure 5: Test images used in the experiments. Test image 1 was chosen due to its fine texture within the leafs and the flowers in order to compare it with the Test images 2 and 3, which contain areas of monotonic changing intensity (representing the sky and the water). Test image 4 was chosen in order to study the effect of edge gradients to the n-tuples and LBPs.

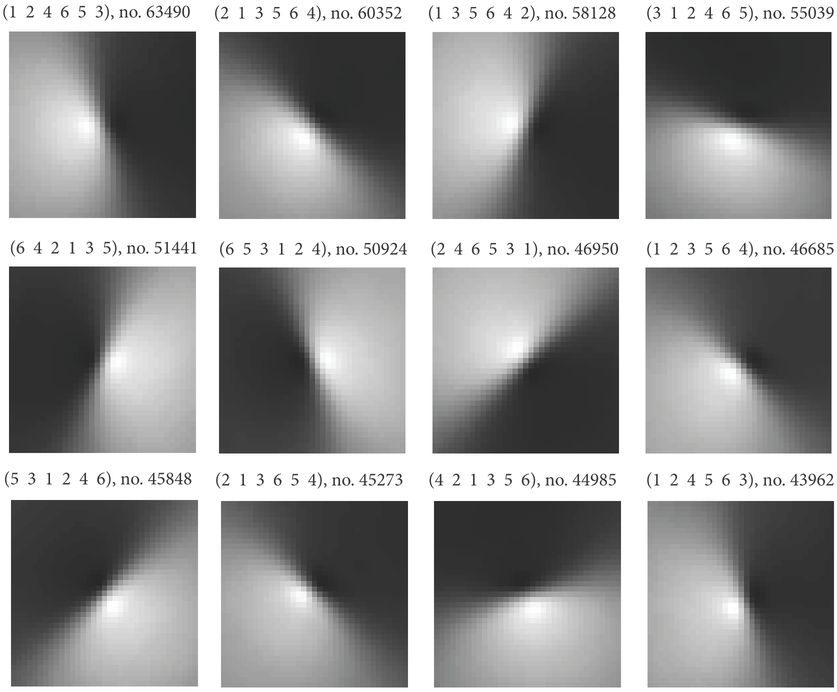
6. Experiments

6.1. Qualitative Tests on n-Tuples and LBPs. In this subsection, the distribution of the n-tuples (intermediate root permutations) is studied with natural image data. The objective is to characterize which individual n-tuples are most common with natural image data and to make implications on their role in the formation of uniform patterns. Also, the spatial response of the n-tuples to different image structures is studied with different kind of images. The runs level 2 intermediate root permutations described in the previous section will be shown to be among the most common n-tuples with natural image data. This observation can promote the understanding of the uniform patterns and their high occurrence probability with natural images in particular.

In Figures 6, 7, 8, and 9 the most common intermediate root permutations are shown for different test images of Figure 5. The total number of occurrences of each permutation is also shown. For each instance of a certain permutation in the corresponding test image, a neighborhood of 35 × 35 pixels was extracted and all of the local intensity blocks for the given permutation were combined, that is, added together. The intensity scale was then normalized based on the minimum and maximum values within the sum of the permutation blocks. In order to better distinguish between the true monotonic runs level 2 n-tuples in nonflat image areas, only patterns which did not contain ties were extracted.

When comparing the responses in Figures 6 and 9, it can be observed that with small neighborhood radius ($M = 6$ with $r = 2$) the n-tuples appear to correspond to edges in various orientations. It can also be observed that the most common intermediate root permutations are typically of runs level 2. Hence, the theoretical analysis in the previous sections is also supported by occurrence statistics of the natural images. It should be noted that the property described in the previous sections, stating that the runs level 2 root permutations will always produce a uniform pattern independently of the bias chosen, holds for the intermediate root permutations as well. Independently of the center chosen, these intermediate root permutations will always produce a uniform LBP (to the direction $I_n \Rightarrow P_n \Rightarrow b_n$). The orientations of the detected edges follow the direction of the most common gradients among the test images (see e.g., Figure 9 and the corresponding test image 4). In Figure 7, the local neighborhood is extended to M of 8 and radius of 8 using test image 2. It can be observed that now the monotonically changing image structures specific to the sky and to the water dominate the average intensity blocks. In Figure 8, M of 6 and radius of 6 are used. It can be observed that the intensity structures corresponding to the n-tuples become smoother compared to the lower radius. In this case the n-tuples capture larger scale changes.

Tests with repeated textures were also performed. The images shown in Figure 10, from the Outex [24] dataset, were used. In Figure 11, the response of the most common n-tuples within $(8, 2)$ neighborhood with interpolation is shown using the Outex images. The most common permutations in Figure 11 correspond to the structures present in wood_012 texture sample. It consists of a gradually changing monotonic texture pattern. According to this experiment it would seem that especially monotonic changes contribute to the formation of runs level 2 patterns.



(1 2 4 6 5 3), no. 63490

(2 1 3 5 6 4), no. 60352

(1 3 5 6 4 2), no. 58128

(3 1 2 4 6 5), no. 55039

(6 4 2 1 3 5), no. 51441

(6 5 3 1 2 4), no. 50924

(2 4 6 5 3 1), no. 46950

(1 2 3 5 6 4), no. 46685

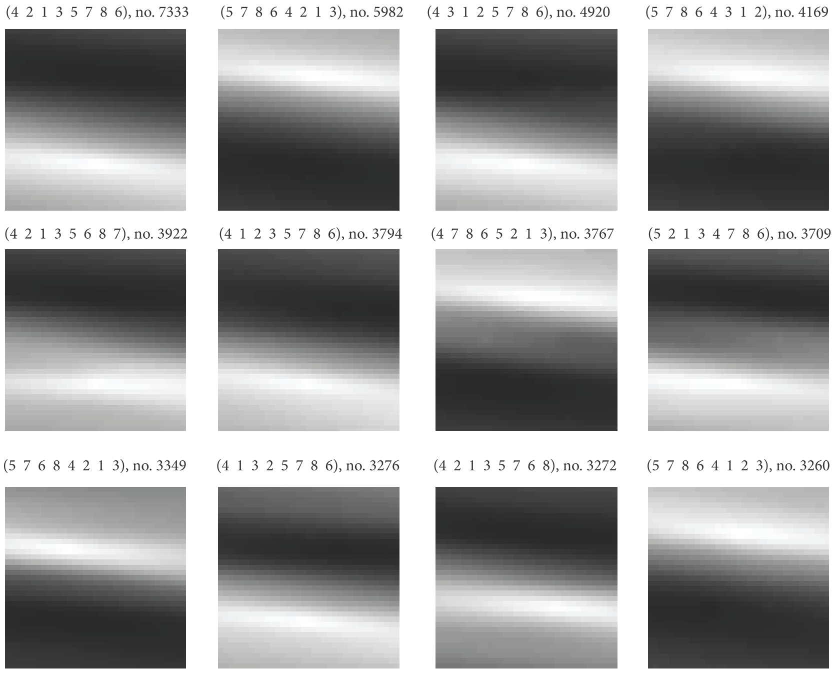
(5 3 1 2 4 6), no. 45848

(2 1 3 6 5 4), no. 45273

(4 2 1 3 5 6), no. 44985

(1 2 4 5 6 3), no. 43962

Figure 6: Average intensity patch of the 12 most common intermediate root permutations in (6, 2) neighborhood (with interpolation) using test image 1 of Figure 5. The first rank from the left corresponds to the Eastern direction, and the following ranks are formed to the counterclockwise direction (North-East, North, etc.).



(4 2 1 3 5 7 8 6), no. 7333

(5 7 8 6 4 2 1 3), no. 5982

(4 3 1 2 5 7 8 6), no. 4920

(5 7 8 6 4 3 1 2), no. 4169

(4 2 1 3 5 6 8 7), no. 3922

(4 1 2 3 5 7 8 6), no. 3794

(4 7 8 6 5 2 1 3), no. 3767

(5 2 1 3 4 7 8 6), no. 3709

(5 7 6 8 4 2 1 3), no. 3349

(4 1 3 2 5 7 8 6), no. 3276

(4 2 1 3 5 7 6 8), no. 3272

(5 7 8 6 4 1 2 3), no. 3260

Figure 7: Average intensity patch of the 12 most common intermediate root permutations in (8, 8) neighborhood (with interpolation) using test image 2 of Figure 5. The first rank from the left corresponds to the Eastern direction, and the following ranks are formed to the counterclockwise direction (North-East, North, etc.) It can be observed that the most common intermediate root permutations with this radius correspond to monotonically changing edge functions characterizing the horizontal gradients of the input image.

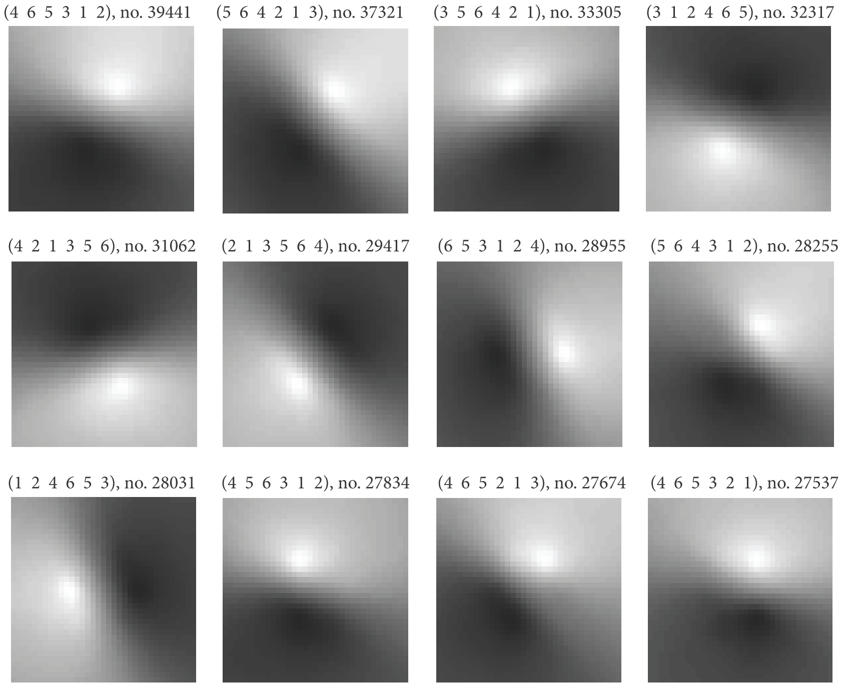


Figure 8: Average intensity patch of the 12 most common intermediate root permutations in (6, 6) neighborhood (with interpolation) using test image 3 of Figure 5. The first rank from the left corresponds to the Eastern direction, and the following ranks are formed to the counterclockwise direction (North-East, North, etc.).

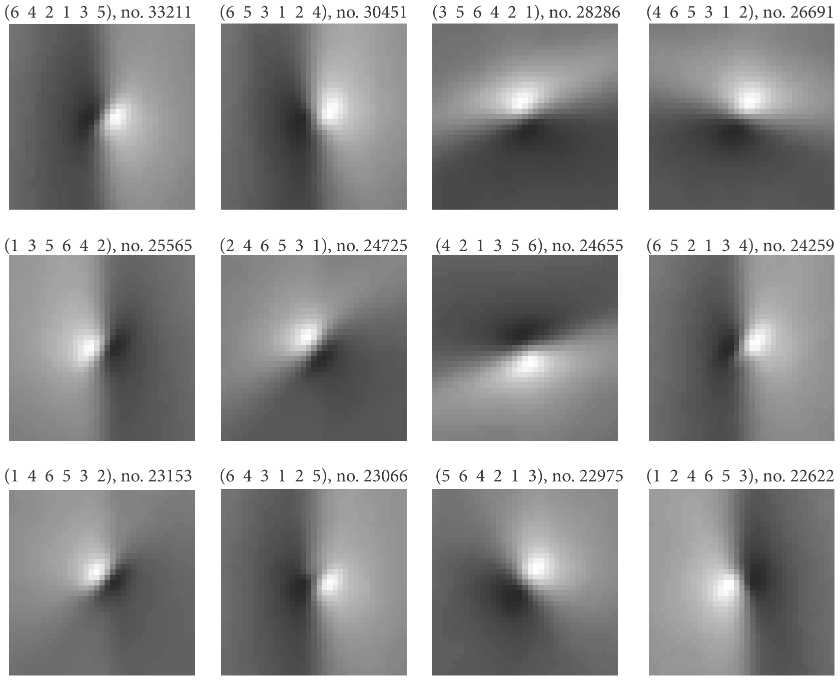


Figure 9: Average intensity patch of the 12 most common intermediate root permutations in (6, 2) neighborhood (with interpolation) using test image 4 of Figure 5. The first rank from the left corresponds to the Eastern direction, and the following ranks are formed to the counterclockwise direction (North-East, North, etc.) It can be observed that the most common intermediate root permutations correspond now to the main edge directions specific to the test image.

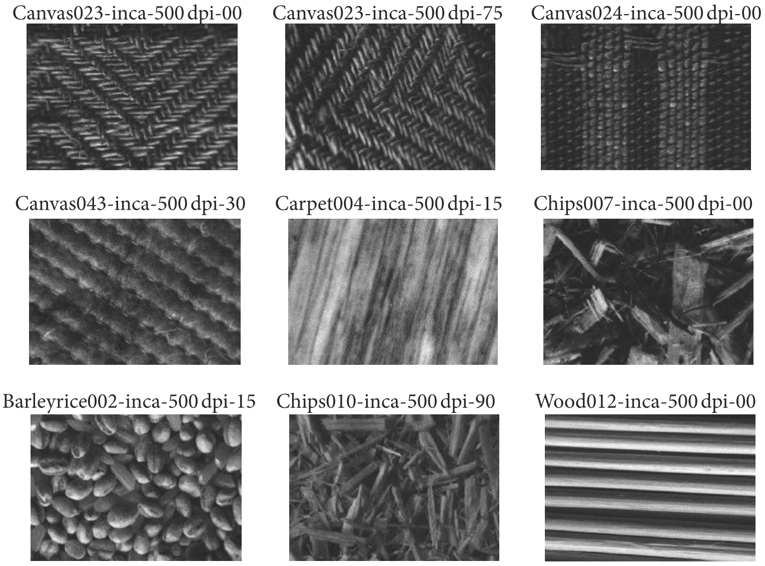

Canvas023-inca-500 dpi-00 Canvas023-inca-500 dpi-75 Canvas024-inca-500 dpi-00
Canvas043-inca-500 dpi-30 Carpet004-inca-500 dpi-15 Chips007-inca-500 dpi-00
Barleyrice002-inca-500 dpi-15 Chips010-inca-500 dpi-90 Wood012-inca-500 dpi-00

Figure 10: Outex [24] test images.

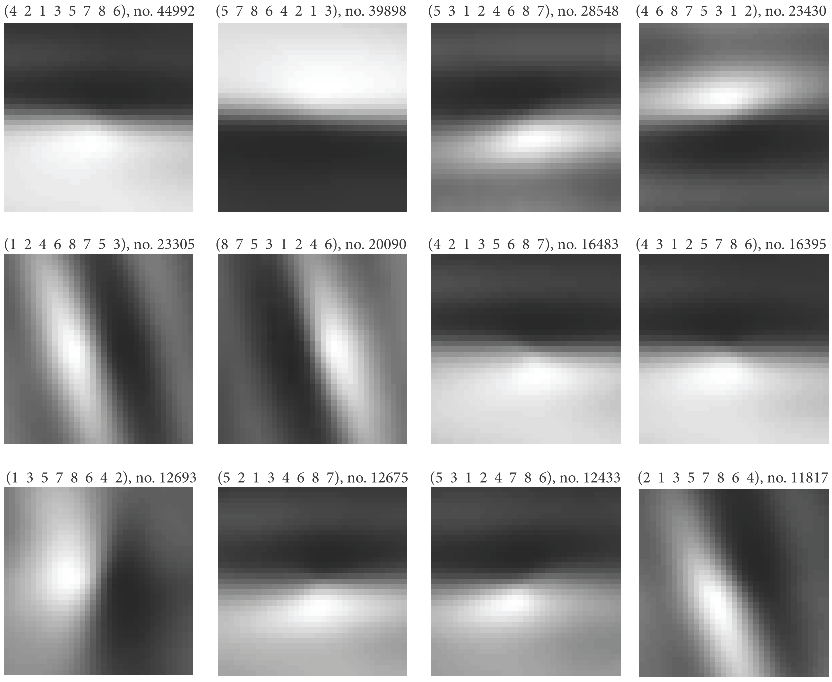

(4 2 1 3 5 7 8 6), no. 44992 (5 7 8 6 4 2 1 3), no. 39898 (5 3 1 2 4 6 8 7), no. 28548 (4 6 8 7 5 3 1 2), no. 23430
(1 2 4 6 8 7 5 3), no. 23305 (8 7 5 3 1 2 4 6), no. 20090 (4 2 1 3 5 6 8 7), no. 16483 (4 3 1 2 5 7 8 6), no. 16395
(1 3 5 7 8 6 4 2), no. 12693 (5 2 1 3 4 6 8 7), no. 12675 (5 3 1 2 4 7 8 6), no. 12433 (2 1 3 5 7 8 6 4), no. 11817

Figure 11: Average intensity patch of the 12 most common intermediate root permutations in (8, 2) neighborhood (with interpolation) using all of the selected Outex test images. The first rank from the left corresponds to the Eastern direction, and the following ranks are formed to the counterclockwise direction (North-East, North, etc.) It can be observed that the most common intermediate root permutations (n-tuples) seem to capture intensity patches corresponding to wood_012. The selected (M, r) combination might be too sensitive to noise. See, for example, similar intensity patches related to permutations (4 2 1 3 5 7 8 6) and (4 2 1 3 5 6 8 7).

Figure 12 shows the average intensity blocks of the LBPs in (6, 2) neighborhood related to the most common intermediate root permutation of test image 4, that is, permutation $\{6, 4, 2, 1, 3, 5\}$. The shown LBPs correspond to situation where the bias is changed gradually from its minimum value to the maximum value. It can be noted that as the descriptor k of the LBPs grows, the spatial support for the edges becomes stronger. With small k, the detected structure becomes limited to the proximity of the center pixel. If the permutation tree representation of Figure 4 is used (in this

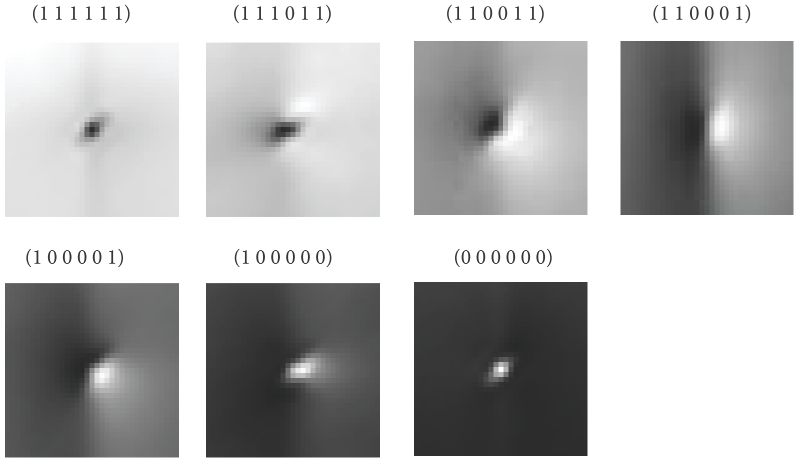


Figure 12: Average intensity patches of uniform LBPs in (6, 2) neighborhood corresponding to the most common intermediate root permutation in Figure 9, that is, permutation (6 4 2 1 3 5). Test image 4 was used. The first bit from the left corresponds to the Eastern direction, and the following bits are formed to the counterclockwise direction.

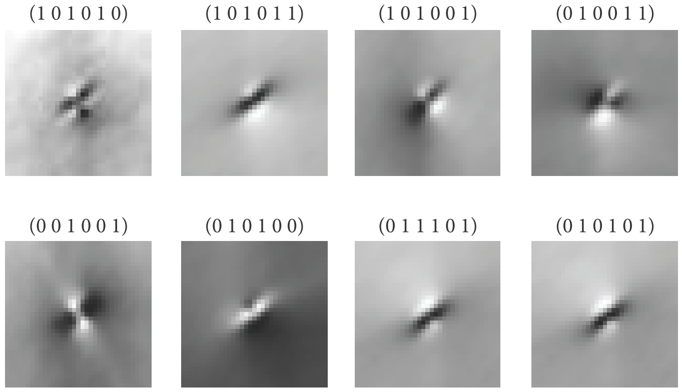


Figure 13: Average intensity patches of certain nonuniform LBPs. Test image 4 is used. The first bit from the left corresponds to the Eastern direction, and the following bits are formed to the counterclockwise direction.

case $M = 6$), the corresponding path for this runs level 2 permutation is $\{LEFT, RIGHT, LEFT, RIGHT\}$ in a circular manner. With similar tests using nonuniform patterns, the response becomes partly limited to the area of the pattern itself, without significant edge support (see Figure 13).

6.2. Tests with a Priori Model. In this subsection, the a priori distribution of LBPs is further studied. The objective is to better understand which factors contribute to the formation of uniform patterns in particular. The proposed approach can also give new perspectives on the previous studies in [8, 21]. Based on the examples in Section 4 (e.g., Table 1), we hypothesize that root permutations which include monotonically ordered subsets would produce more uniform patterns than permutations of arbitrary order. This is analyzed first.

The total number of unit permutations (and final LBPs) was $(M+1)!$, corresponding to the number of instances in the constructed permutation space. The following experimental procedure was then applied.

(1) A table containing all possible $M!$ intermediate root permutations was constructed.

(2) Using these permutations as root permutations, the center bias was changed for each permutation $M + 1$ times in order to generate all the child LBPs for the permutation space corresponding to all unit permutations in given LBP neighborhood M.

In Figure 14, the length of the longest monotonic run among the root permutations is plotted, along with the total share of permutations resulting in uniform child LBPs. For example, in the case of $M = 10$, if the length of the longest monotonic run among the root permutations is 3, roughly 40% of the resulting LBPs are uniform, and if the length of the longest monotonic run is 7, roughly 70% of the resulting LBPs are uniform. These results would seem to support the given hypothesis.

Next, we studied the correspondence between the runs test and the relative share of uniform Local Binary Patterns.

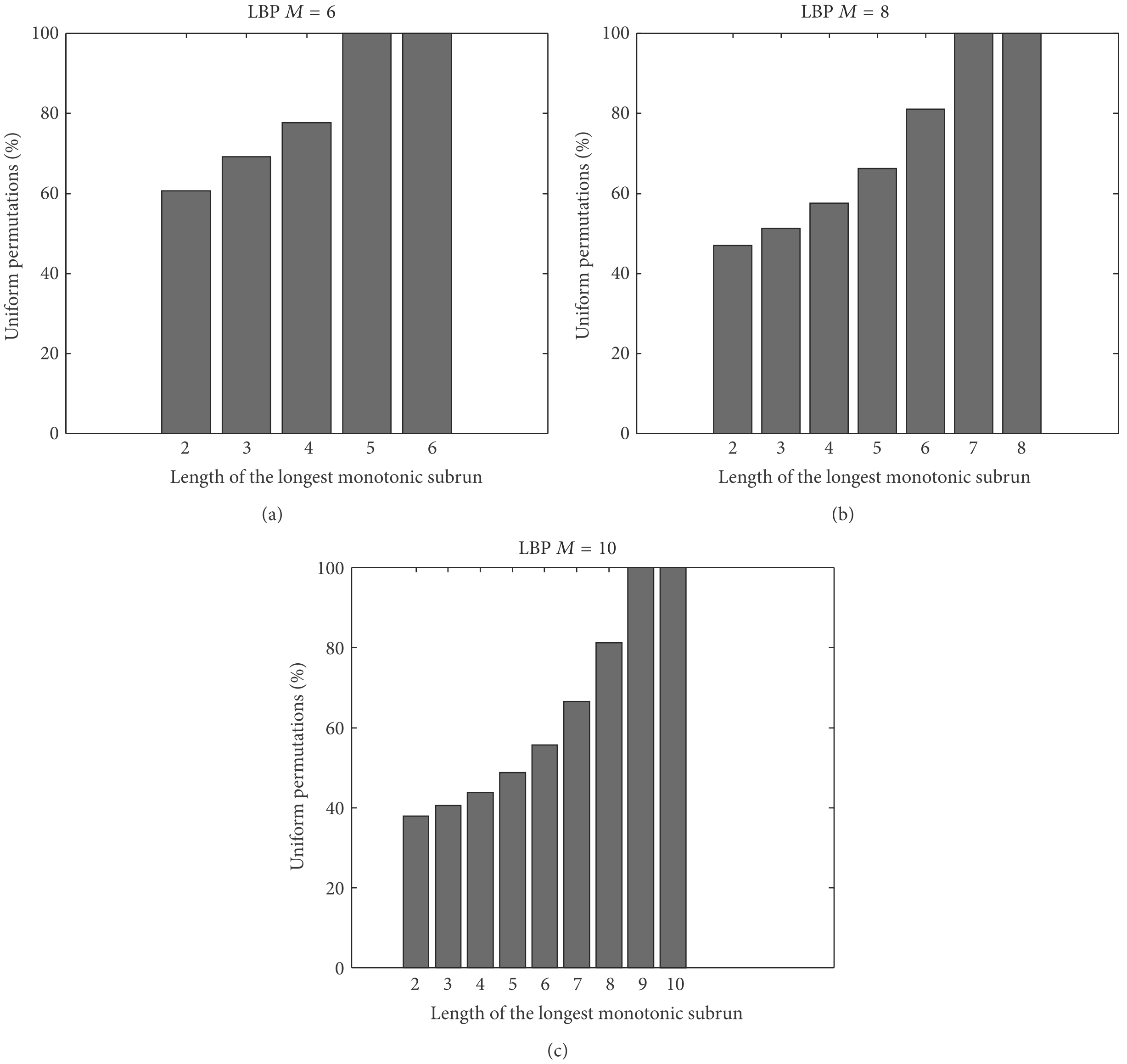


FIGURE 14: Length of the longest monotonic consecutive run among the permutations with respect to the total share of uniform child LBPs while going through all unit permutations.

In Figure 15, the setup was the same as previously used. The percentage of permutations resulting in a uniform child LBP is plotted, with respect to the total number of LBPs within the current category. The result of the circular up and down runs test for randomness is indexed on the x-axis. It can be observed that as the result of the runs up and down test for permutations decreases, the share of uniform patterns increases. It can also be observed that for the runs test result 2, all the LBPs are uniform.

In Figure 16 the data corresponding to the previous experimental setup is plotted with $M = 10$ and also with respect to descriptor k. It can be observed that, as k decreases, the share of permutations resulting in uniform LBP increases. Also, as the result of the circular up and down runs test for randomness decreases, the relative share of uniform permutations (frequency of permutations resulting in uniform patterns) increases. This implies that uniform patterns become more frequent with a stronger hypothesis of nonrandomness according to the runs test. According to Figure 16, for the descriptor k values 0 and 1, all of the root permutations result in uniform patterns, since all patterns (of all-zero LBP, all-one LBP, all-zero LBP including a single 1, and all-one LBP including a single 0) are then uniform. When the result of runs up and down test is 2, the patterns remain uniform despite the increase in k.

We also examined the relationship between the runs test and individual LBPs by considering their root permutations. It is clear that, for example, LBP 01010101 must contain at least 8 runs. However, LBP 10000 could contain various numbers of runs according to runs up and down test for randomness,

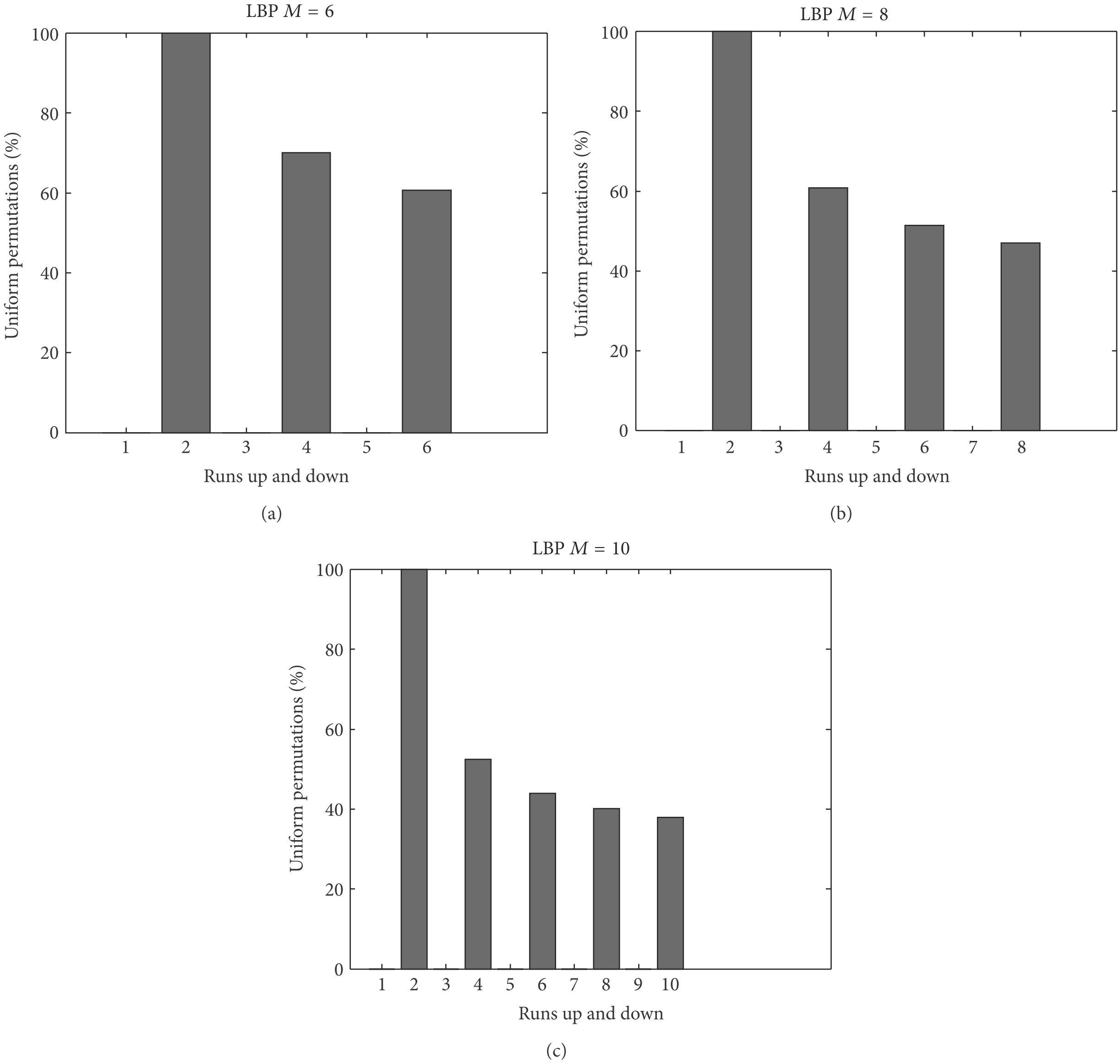


FIGURE 15: Number of root permutations resulting in uniform child LBPs with respect to total amount of permutations. The runs up and down test result for the randomness of the permutation is plotted on the x-axis.

since now the leftmost LBP bit (or the bit change next to it) alone restricts the degree of freedom of the underlying root permutations. The remaining long sequence of all zeros allow a large number of combinations considered as possible root permutations.

In the following, a reverse mapping ϕ_{MAP}^{-1} of Definition 5(e) is used. For LBPs with $M = 6$, 000101, 101110 the number of all circular runs among the root permutations varied between 4 and 6, with a mean of 4.667 and variance 0.908. For uniform $M = 6$ patterns 011100 and 111000 the runs test result varied between 2 and 6 with a mean of 3.33 and variance 1.829. The rotation of the patterns did not seem to affect the result, as was expected. For uniform pattern 00000000 the number of runs varied between 2 and 8 with a mean of 5.333 and variance 1.4223 (in this case, the total number of root permutations was 8!). For uniform $M = 8$ pattern 01100000 the result of the runs test varied between 2 and 6 with a mean of 4.667 and variance 1.245. The results from the ϕ_{MAP}^{-1} test indicate that nonuniform patterns, in average, contain more runs than the uniform patterns, as can be expected.

In Figure 17, the number of uniform patterns (permutations resulting into uniform patterns) is shown, with respect to the length of the longest monotonic run and descriptor k, while using the same approach as in the previous figures of this subsection. It can be observed that maximum run lengths around 3, 4, and 5 provide most of the contribution to uniform Local Binary Patterns with LBP neighborhood of $M = 8$ in the case of the a priori model. As expected, as the descriptor k grows, the number of permutations resulting

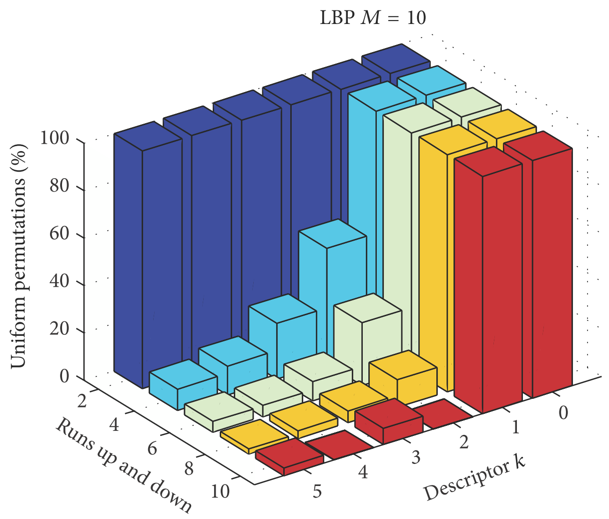


Figure 16: Number of runs among the root permutations, plotted with respect to descriptor k. Z-axis represents the probability of permutations resulting in uniform child LBP.

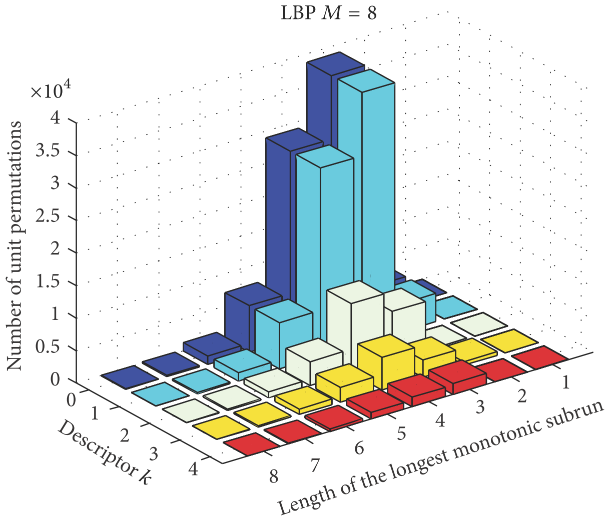


Figure 17: The longest monotonic run among root permutations, plotted with respect to descriptor k and only considering permutations resulting in uniform child LBPs.

in uniform LBPs also decreases due to reduced number of binomial combinations. In general, the number of particular unit permutations related to high descriptor k LBPs is smaller (see (1)), emphasizing the relative effect of these particular unit permutations on the final LBP histogram.

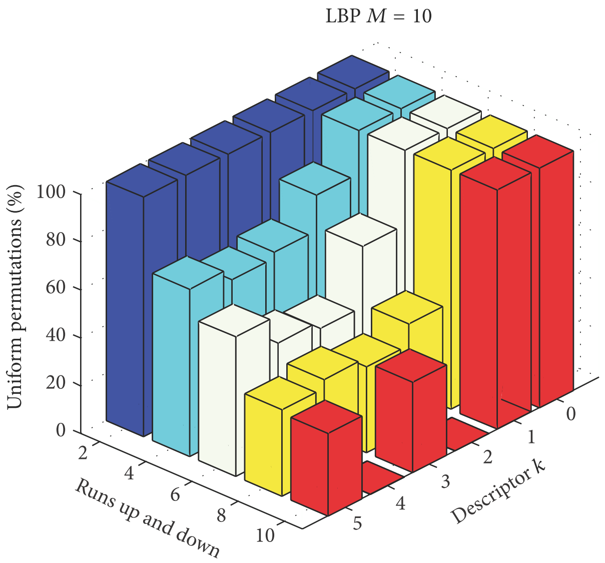


Figure 18: Percentage of uniform patterns with respect to the runs up and down test result and descriptor k. Natural images in LBP(10, 2) neighborhood with interpolation are used.

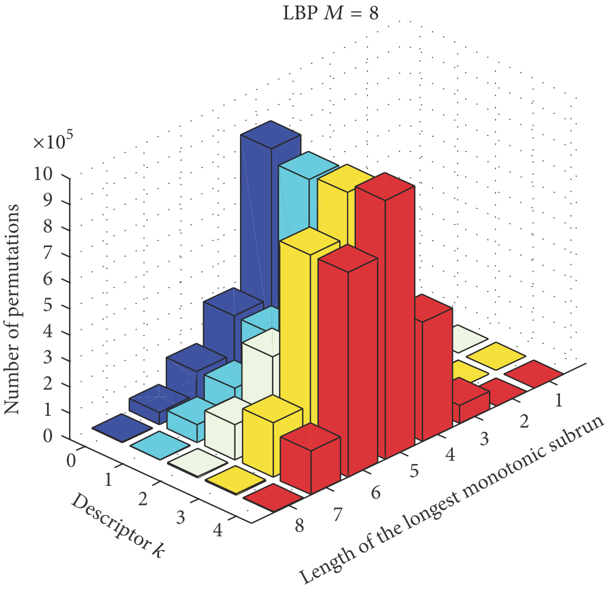


Figure 19: Number of intermediate root permutations resulting in uniform patterns with respect to the longest monotonic subrun and descriptor k. Natural images in LBP(8, 2) neighborhood with interpolation are used.

6.3. Experiments with Natural Image Data. The experiments of the previous subsection with the a priori model are next considered in the case of natural image data, by using the test images of Figure 5. In the following, the effect of ties and the effect of runs level 2 permutations are considered separately. This allows studying the role of these two in the formation of uniform patterns in particular. The grayscale range of all test images is 8 bits. Figures 18 and 19 correspond to the experiments with the a priori model shown in Figures 16 and 17, respectively. In Figure 18, the percentage of uniform patterns with respect to runs up and down test and descriptor k is shown. An increased percentage of uniform patterns with larger k values, in comparison with the a priori model, can be observed. These patterns correspond to monotonic edge

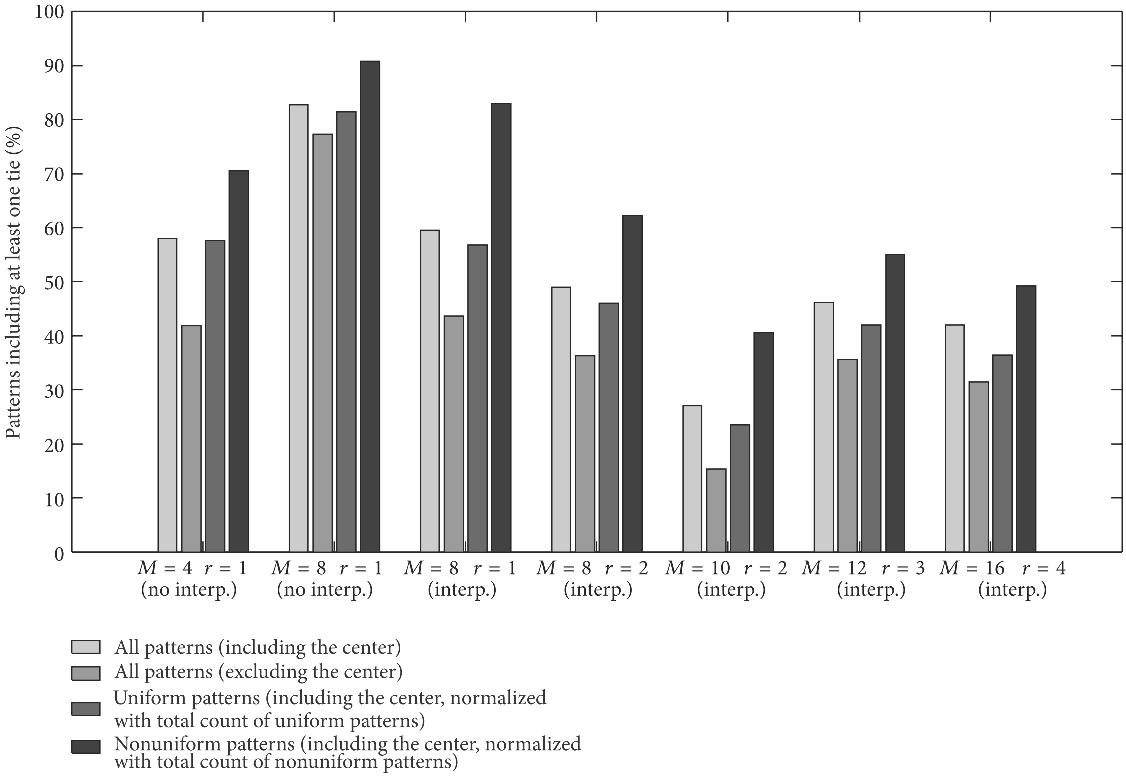


Figure 20: The effect of ties (neighboring pixels being of equal magnitude) to the formation of uniform patterns. It can be observed that on average, uniform patterns tend to contain less ties than nonuniform patterns.

structures in the images. In Figure 19 the number of uniform patterns is increased, especially for larger descriptor k values and longer run lengths. While being a priori is extremely rare, the high k patterns are actually among the most common ones with natural image data when only uniform patterns are considered. The results of Figures 17 and 19 also indicate that natural image data increases the monotonicity of the root permutations for the uniform patterns. As a consequence, the number of uniform patterns is increased.

The effect of ties is considered next. In Figure 20, the percentages of local neighborhoods containing at least one tie are shown with respect to neighborhood radius and the possible usage of interpolation using all of the test images of Figure 5. For uniform patterns, the percentage share of patterns containing at least one tie is also shown. It can be observed that ties are not, on average, more common among uniform patterns than with the other patterns. Similar results were obtained with the Outex textures of Figure 10. This implies that although ties could cause some minor increase in certain LBP bins (e.g., all-one bin or all-zero bin) within flat image areas, as could be predicted from the a priori model used in [8], it seems that they cannot explain most of the increase related to the occurrence frequency of uniform patterns, when using natural images. A possible explanation to this could be that ties tend to occur within flat image areas where the relative amount of i.i.d. noise is more significant.

In Figure 21, the percentage share of runs level 2 patterns is shown for each of the test images of Figure 5. A minor increase in the number of these patterns can be observed for test images 2 and 3. This could be related to the large monotonic areas of the sky and the water present in these images. The a priori estimate of the number of runs level 2 permutations (corresponding to $M * 2^{M-2}/(M!)$, see Definition 9) is also shown in Figure 21. Natural images seem to increase the share of the runs level 2 permutations significantly, compared to the a priori estimate. The percentage share of runs level 2 permutations among uniform patterns is further shown for all of the test images in Figure 21 (as an overall average for test images 1–4). It can be observed that also with natural image data, the occurrence frequency of these permutations increases further among uniform patterns.

When studying Figure 21, it can be observed in the case of $M = 8$ that interpolation increases the share of runs level 2 permutations in each of the test images. The share of uniform patterns naturally increases also when using interpolation. It can be observed that while the number of runs level 2 patterns is significantly larger with the test images than with the a priori model, increasing the neighborhood radius decreases the number of runs level 2 patterns (see e.g., the results in (8, 1) and (8, 2) neighborhoods with interpolation). This could be explained by reduced correlation among the pixel intensities within the local neighborhood, as the radius increases.

6.4. Quantitative Tests. We also performed initial experiments with the FERET facial recognition database [25] (FAFB, FAFC, DUP1, and DUP2) sets and Outex TC 0012 [24]

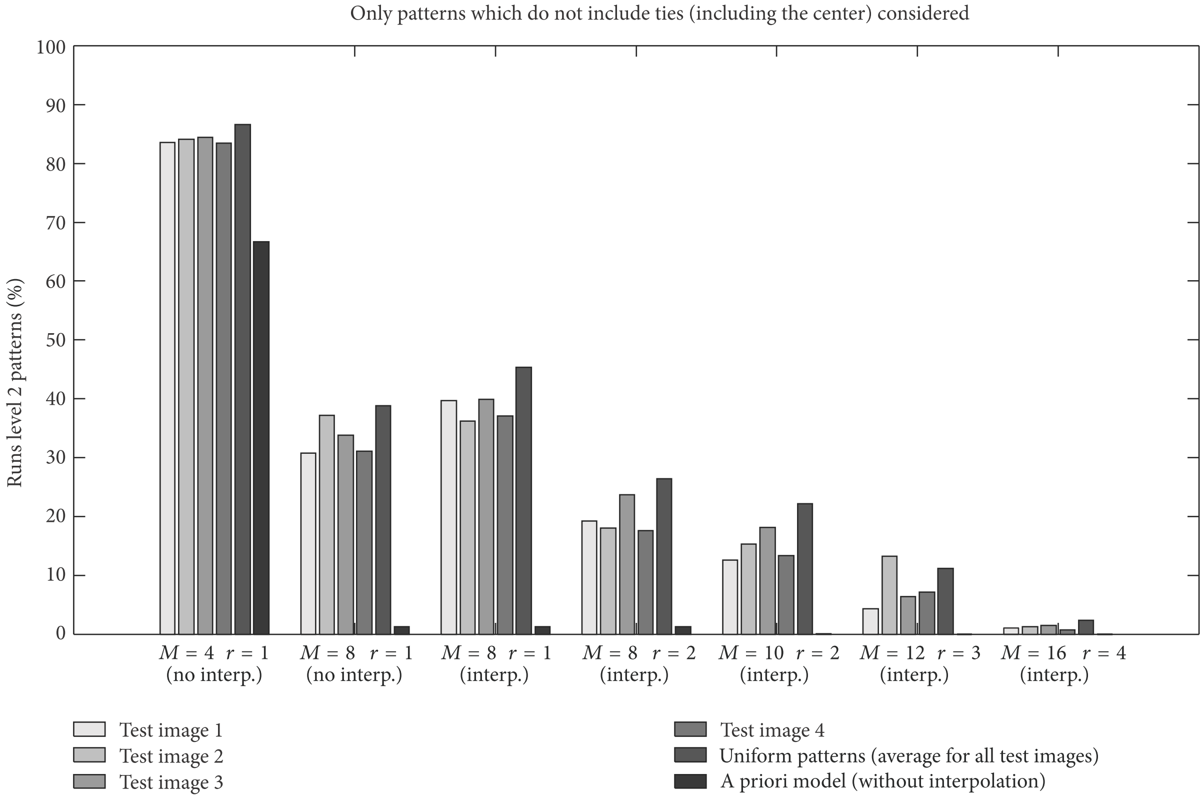


Figure 21: Percentage share of runs level 2 permutations among images 1–4 in various permutation neighborhoods. The bar corresponding to uniform patterns indicates the occurrence frequency of runs level 2 patterns among uniform patterns only. It can be observed that the average frequency of runs level 2 permutations increases compared to the a priori model in all neighborhoods and even further when the uniform patterns are considered.

rotation invariant texture categorization set to test whether the intermediate root permutations of runs level 2 would perform better in classification than traditional LBP with uniform patterns. In all of the following tests, if not explicitly noted, ties were coded in an increasing order of the rank magnitudes among the n-tuples, so that a neighborhood containing only tied intensities was assigned to the increasing permutation $\{1, 2, 3, \ldots, M\}$.

In Figure 22, the recognition rates for the FERET sets are shown. The length of the feature vector in $M = 6$ was 720 (i.e., factorial of 6). It can be observed, that using only runs level 2 n-tuples (total of 96 out of 720), the recognition rate was not significantly lower than with the full permutation histogram. Increasing the permutation neighborhood to (8, 2) with interpolation and using runs level 2 patterns only (total of 512 patterns out of 40320) an average recognition accuracy close to the reported LBP(8, 2) accuracy [9] was obtained. However, it appears that for larger M (say 8 or more) and small radiuses the number of runs level 2 permutations drops even with natural image data. This can be related to the effect of noise, the high dimension of the permutation histogram, and the substantially low a priori probability of the runs level 2 patterns. The a priori probability of runs level 2 patterns in $M = 8$ neighborhood (without considering interpolation) is as low as 1.27%, emphasizing the significant role of these permutation bins in the overall permutation histogram. To increase the number of runs level 2 patterns, we used a 4 × 4 averaging filter as a preprocessing step with M of 8.

Table 2: Outex TC 0012 rotation invariant texture classification experiment.

Outex TC 0012	Recognition rate (%)	FV length
N-tuples (6, 2) ROT-INV interp. (runs 2 only)	47.6	16 bins
N-tuples (6, 2) ROT-INV interp. (all)	55.5	120 bins
N-tuples (8, 2) ROT-INV interp. (runs 2 only)	**58.8**	64 bins
LBP (8, 1) ROT-INV interp.	**64.6**	10 bins

The effect of ties to the performance of n-tuples can also be estimated from the results of Figure 22 with $M = 6$. It can be observed that neglecting ties reduces the recognition performance for radius of 2, but with larger radiuses the recognition accuracy is not significantly altered. The length of the individual block histogram in [9] for the LBPs was 59, which is smaller than for the permutations.

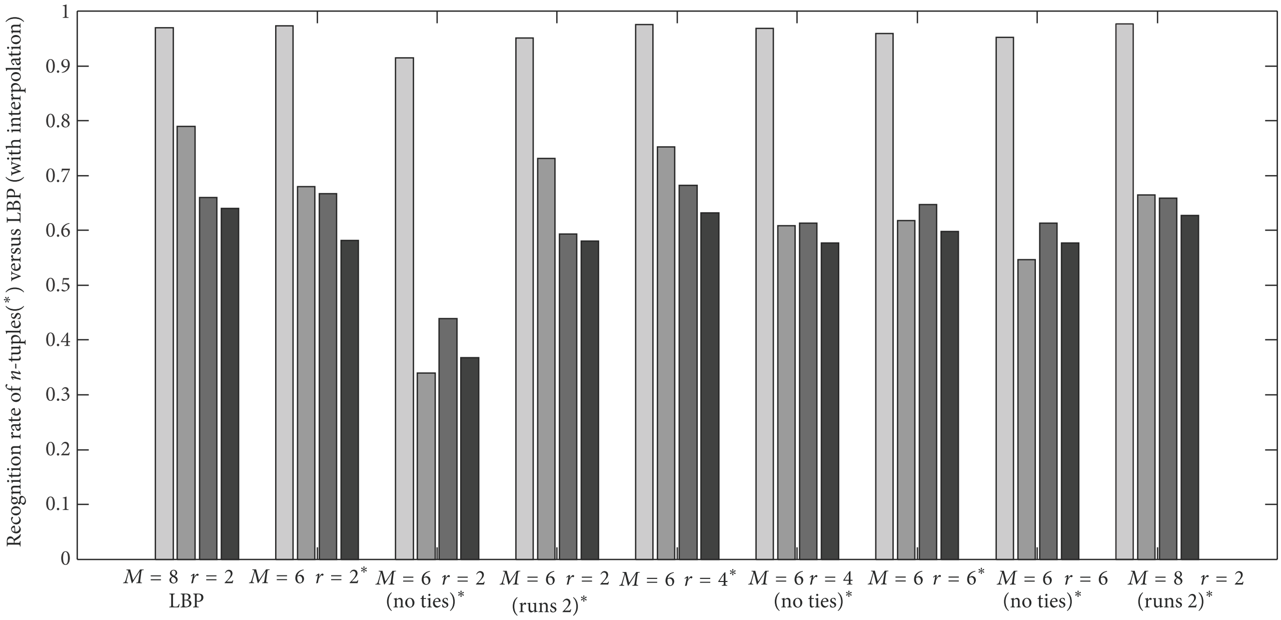


Figure 22: FERET face recognition results with n-tuples are shown in comparison with LBP. Test sets FAFB, FAFC, DUP1, and DUP2 are considered (from the left to the right). It can be observed that LBP outperforms n-tuples. Also for LBP the length of the feature vector is smaller.

In tests on rotation invariant texture classification (see the results of Table 2) with the Outex TC 0012 set [24], a rotation invariant mapping of intermediate root permutations (n-tuples) was used. For each available permutation pattern, M possible rotations were assigned, and these were then combined in a rotation invariant manner [10]. In the case of runs level 2 patterns only, a similar procedure was also performed. Using (6, 2) neighborhood the total number of runs level 2 patterns was 96, and for each rotation invariant bin, 6 rotated bins were assigned. The final length of the feature vector was then 16 bins. All rotation invariant n-tuples (i.e., not only runs level 2) in the (6, 2) neighborhood resulted in a feature vector length of 120 bins. In the (8, 2) neighborhood with interpolation using only runs level 2 patterns, by always assigning 8 patterns into the same rotation invariant category the final length of the feature vector became 64 bins for the n-tuples. The highest recognition rate for the n-tuples was obtained with these parameters using the runs level 2 patterns only. Also the feature vector length in the (8, 2) neighborhood was shorter than with the (6, 2), which would seem to indicate that the runs 2 n-tuples are among the most salient ones, if the n-tuples alone are considered. However, traditional rotation invariant LBP in a (8, 1) neighborhood with interpolation resulted in a better recognition rate (64.6% accuracy) in [14] with the same test set. Also, the length of the feature vector for the LBPs was shorter (10 bins). Log-likelihood distance metric with 1-nearest neighbor classification was used in these experiments.

The previous results raise naturally the question, why are LBPs more discriminative than n-tuples, despite the usage of a similar feature vector length reduction method (i.e. the usage of runs level 2 patterns) than the uniform patterns. According to the previous qualitative and quantitative analysis (see also Figure 22), it appears that the performance of the n-tuples is decreased in comparison with LBPs for two reasons. First, their performance is decreased due to large number of ties among small local radiuses. However, this is changed if a larger radius is used. Second, a small change in the intensity order (i.e., on the permutation) for larger M (e.g., more than 6) changes the bin placement of the histograms and increases susceptibility to noise, which is not the case in LBP. Above-mentioned issues should be considered carefully when designing descriptors based on n-tuple processing. As a rule of thumb for selecting the number of samples in an n-tuple, the radius of the local neighborhood should be at least equal to the number of circular sample points M (see Figure 22). In the recently proposed LOCP descriptor [26] a binary representation of successive circular pairs in a local neighborhood was used. This approach avoids the negative effect of ties by changing the following circular bit only if the intensity is changed, that is, the bin placement of the permutation histogram does not change due to possible minor intensity order differences. As a consequence, robustness to noise could be achieved in [26] by only considering the neighboring circular pairs when deriving the binary pattern.

7. Discussion

In this work, the observation that Local Binary Patterns can be modeled as compositions of rank permutations was used to study some of the mechanisms related to the formation of uniform patterns. As previously observed, uniform patterns are a priori very frequent (with i.i.d. data), but this seems to be mostly explained by how they are located according to the descriptor k (see (1)–(4)). For most k values there are exactly $2 * M$ instances of uniform LBP rotations and inversions, while the number of all LBPs increases rapidly with descriptor k. As a consequence the relative contribution of uniform LBPs is larger at low k values. Therefore, according to a priori model, uniform LBPs contain a larger share of unit permutations compared to other LBPs and are very frequent.

According to our investigation, the contribution of uniform patterns seems to get even stronger with a stronger evidence of nonrandomness among the underlying data. This issue was quantitatively analyzed in this paper using the runs up and down test for permutations. Monotonicity among the root permutations would be a strong indicator of nonrandomness. In tests with natural images, the majority of the most common circular permutations belonged to category 2 according to the runs test, corresponding to monotonically changing intensity structures. Using natural image data also increased the length of monotonic runs among the permutations, compared to the a priori model.

With real-world images occurrence percentages of uniform patterns of 70–90% and beyond are typical [10]. For example, with $M = 16$ the percentage of uniform patterns in textures was in the range of 57.6–79.6% in [10], while the a priori probability that we estimated in this paper was less than 30% (see Figure 3) for i.i.d. data without interpolation. A considerable portion of this increase can be explained by the bilinear subsampling (interpolation when deriving the LBP code) as described in [8], but we also propose that a portion of this increase could be explained by the capability of the uniform patterns to respond to deterministic properties within the image microstructure.

The relatively high occurrence probability of uniform patterns a priori, and even higher occurrence probability with natural images, could be compared also with the relative occurrence probabilities of individual n-tuples [4], since we showed that LBPs could be seen as compositions of n-tuples. Monotonic n-tuples dominate the occurrence statistics of natural images [4], which tends to increase the share of uniform patterns. This behavior was modeled quantitatively in this paper (see Figure 14).

Permutations with runs test result 2 can be used to capture monotonic edges and monotonic spatial image features, as was shown in Section 6. By increasing the radius of the local neighborhood, also the extent of the detected change could be increased. The formation of these patterns was examined in Figure 4. The understanding of this behavior could facilitate the development of new image descriptors inspired by the uniformity principle of Local Binary Patterns. Initial tests on the performance of runs level 2 permutation histograms were performed in Section 6, and it appeared that their performance did not exceed the original LBP. However, since the formation of the runs level 2 permutations can also be modeled as a tree structure, a graph based matching approach for the permutations could also be possible.

The use of only certain permutations inspired by the uniform pattern principle would seem to enhance the properties of n-tuples (see the quantitative experiments in Section 6). However, for larger M the feature extraction cost is higher for the n-tuples and the required feature vector becomes longer. Therefore, in these cases the traditional LBP with uniformity is preferable. Future work includes examining the possibilities to enhance the performance of n-tuple based descriptors, for example, [18, 19] based on the principles proposed. We showed that the uniform pattern principle can be at least partly extended to n-tuples as well. This could provide a variety of alternatives for increasing the performance of n-tuple based descriptors. For example, the pooling scheme in [18, 19] could be adjusted, not only to take into account rotation but also to select n-tuples according to their runs test score. This could also allow increasing the number of neighborhood samples among pooled n-tuples without increasing the descriptor length significantly.

8. Conclusions

We proposed the modeling of LBPs through nonlinear intermediate mapping into permutations. The permutation set was further modified in order to gain a more flexible LBP model by removing the effect of the center pixel on the actual permutations and by modulating the effect of the center pixel by introducing an adjustable bias term. The approach proposed in this paper was intended to provide further understanding of the LBPs and of the uniform LBPs in particular. The notion of root permutations was introduced in order to model the formation process of uniform patterns. Monotonicity among the root permutations was shown to be in an important role in the increased share of uniform patterns with natural images. The possible relationship between the runs up and down test for randomness for permutations and the selection of uniform patterns for LBP histograms was also considered. According to our investigation, the response of uniform patterns is enhanced when the result of the runs test is low, that is, indicating nonrandomness and correlation.

The a priori occurrence probability of uniform LBPs is high. This has been previously observed to be a result from the sampling process itself as well as from the use of bilinear interpolation. In addition to this, we provided quantitative analysis on how the Local Binary Pattern methodology, together with selecting only the uniform patterns, can be seen as a process which further enhances the response of such deterministic underlying image intensity structures, which are not likely to be formed by statistically random distribution or phenomena. Being also shape primitives, the uniform Local Binary Patterns naturally embody response to various microshapes.

Acknowledgment

The research was funded by the Academy of Finland Project no. 254430.

References

[1] M. Pietikainen, A. Hadid, G. Zhao, and T. Ahonen, *Computer Vision Using Local Binary Patterns*, Springer, Berlin, Germany, 2011.

[2] N. Chatlani and J. J. Soraghan, "Local Binary patterns for 1D signal processing," in *Proceedings of the 18th European Signal Processing Conference*, pp. 95–99, Aalborg, Denmark, 2010.

[3] http://www.cse.oulu.fi/CMV/LBP_Bibliography/.

[4] L. Hepplewhite and T. J. Stonham, "Texture classification using N-tuple pattern recognition," in *Proceedings of International Conference on Pattern Recognition (ICPR '96)*, vol. 4, pp. 159–163, Vienna, Austria, 1996.

[5] L. Wang and D.-C. He, "Texture classification using texture spectrum," *Pattern Recognition*, vol. 23, no. 8, pp. 905–910, 1990.

[6] L. Hepplewhite and T. J. Stonham, "N-tuple texture recognition and the zero crossing sketch," *Electronics Letters*, vol. 33, no. 1, pp. 45–46, 1997.

[7] W. Zhang, S. Shan, W. Gao, X. Chen, and H. Zhang, "Local Gabor Binary Pattern Histogram Sequence (LGBPHS): a novel non-statistical model for face representation and recognition," in *Proceedings of the 10th IEEE International Conference on Computer Vision (ICCV '05)*, vol. 1, pp. 786–791, October 2005.

[8] F. Bianconi and A. Fernández, "On the occurrence probability of local binary patterns: a theoretical study," *Journal of Mathematical Imaging and Vision*, vol. 40, no. 3, pp. 259–268, 2011.

[9] T. Ahonen, A. Hadid, and M. Pietikäinen, "Face description with local binary patterns: application to face recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 12, pp. 2037–2041, 2006.

[10] T. Ojala, M. Pietikäinen, and T. Mäenpää, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 971–987, 2002.

[11] C. Shan and T. Gritti, "Learning discriminative LBP-histogram bins for facial expression recognition," in *Proceedings of British Machine Vision Conference (BMVC '08)*, Leeds, UK, 2008.

[12] T. Ahonen, J. Matas, C. He, and M. Pietikäinen, "Rotation invariant image description with local binary pattern histogram fourier features," in *Proceedings of Scandinavian Conference on Image Analysis (SCIA '09)*, vol. 5575 of *Lecture Notes in Computer Science*, pp. 61–70, 2009.

[13] A. Fernández, O. Ghita, E. González, F. Bianconi, and P. F. Whelan, "Evaluation of robustness against rotation of LBP, CCR and ILBP features in granite texture classification," *Machine Vision and Applications*, vol. 22, no. 6, pp. 913–926, 2011.

[14] G. Zhao, T. Ahonen, J. Matas, and M. Pietikäinen, "Rotation-invariant image and video description with local binary pattern features," *IEEE Transactions on Image Processing*, vol. 21, no. 4, pp. 1465–1477, 2012.

[15] T. Maenpaa, T. Ojala, and M. Pietikainen, "Robust texture classification by subsets of Local Binary Patterns," in *Proceedings of the 15th International Conference on Pattern Recognition*, vol. 3, pp. 947–950, Barcelona, Spain, 2000.

[16] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *Journal of Machine Learning Research*, vol. 3, pp. 1157–1182, 2003.

[17] R. Zabih and J. Woodfill, "Non-parametric local transforms for computing visual correspondence," in *Proceedings of European Conference on Computer Vision*, pp. 151–158, Stockholm, Sweden, 1994.

[18] Z. Wang, B. Fan, and F. Wu, "Local intensity order pattern for feature description," in *Proceedings of IEEE International Conference on Computer Vision (ICCV '11)*, pp. 603–610, November 2011.

[19] B. Fan, F. Wu, and Z. Hu, "Rotationally invariant descriptors using intensity order pooling," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 10, pp. 2031–2045, 2012.

[20] T. Ahonen and M. Pietikäinen, "Image description using joint distribution of filter bank responses," *Pattern Recognition Letters*, vol. 30, no. 4, pp. 368–376, 2009.

[21] O. Lahdenoja, "A statistical approach for characterizing local binary patterns," TUCS Technical Report 795, 2006.

[22] O. Lahdenoja, M. Laiho, and A. Paasio, "Reducing the feature vector length in local binary pattern based face recognition," in *Proceedings of IEEE International Conference on Image Processing (ICIP '05)*, pp. 914–917, Genova, Italy, September 2005.

[23] J. D. Gibbons, *Nonparametric Statistical Inference*, McGraw-Hill, New York, NY, USA, 1975.

[24] T. Ojala, T. Maenpaa, M. Pietikainen et al., "Outex—new framework for empirical evaluation of texture analysis algorithms," in *Proceedings of the 16th International Conference on Pattern Recognition*, vol. 1, pp. 701–706, 2002.

[25] P. J. Phillips, H. Wechsler, J. Huang, and P. J. Rauss, "The FERET database and evaluation procedure for face-recognition algorithms," *Image and Vision Computing*, vol. 16, no. 5, pp. 295–306, 1998.

[26] C. H. Chan, B. Goswami, J. Kittler, and W. Christmas, "Local ordinal contrast pattern histograms for spatiotemporal, lip-based speaker authentication," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 2, pp. 602–612, 2012.

ARTICLE 9

O. Lahdenoja, T. Säntti, J. Poikonen, M. Laiho, A. Paasio, Characterizing spatters in laser welding of thick steel using motion flow analysis, 18th Scandinavian Conference on Image Analysis, pp. 675-686, 2013.

ARTICLE 10

O. Lahdenoja, T. Säntti, M. Laiho, J. Poikonen, Spatter Tracking in Laser- and Manual Arc Welding with Sensor-level Pre-processing, 22nd International Conference on Computer Graphics, Visualization and Computer Vision (WSCG), 2014.

Spatter Tracking in Laser- and Manual Arc Welding with Sensor-level Pre-processing

Olli Lahdenoja
Technology Research Center
Brahea Center, University of Turku, 20014 Turku Finland
olanla@utu.fi

Tero Säntti
Technology Research Center
Brahea Center, University of Turku, 20014 Turku Finland
teansa@utu.fi

Mika Laiho, Jonne Poikonen
Technology Research Center
Brahea Center, University of Turku, 20014 Turku Finland
mlaiho@utu.fi, jokapo@utu.fi

ABSTRACT

This paper presents methods for automated visual tracking of spatters in laser- and manual arc welding. Imaging of the welding process is challenging due to extreme conditions of high radiated light intensity variation. The formation and the number of spatters ejected in the welding process are dependent on the parameters of the welding process, and can potentially be used to tune the process towards better quality (either on-line of off-line). In our case, the spatter segmentation is based either on test on object elongatedness or Hough transform, which are applied on pre-processed image sequences captured by a high-speed smart camera. Part of the segmentation process (adaptive image capture and edge extraction) is performed on the camera, while the other parts of the algorithm are performed off-line in Matlab. However, our intention is to move the computation towards the camera or an attached FPGA.

Keywords

Laser welding, Thick Steel, Metal arc welding, Spatter tracking, Smart camera.

1. INTRODUCTION

One phenomenon in laser welding and welding in general is the formation of spatters, which represent the ejection of melt from the interaction zone caused by extreme conditions of material interaction. Increased spatter formation may indicate that the process parameters need to be adjusted to improve the quality.

In previous works [Nic11a], [Nic09a] the spatter formation was related to the on-line control of process parameters in welding of aluminum and 1mm thick steel sheets. In [Nic11a] the spatter segmentation was based on blob analysis including a threshold operation with a suitable mask. In [Jäg08a] a spatter tracking system incorporating a Kalman filter for trajectory analysis was proposed. In [Fen09a] spatter tracking in the context of laser MAG (metal active gas) welding was proposed. An off-line Kalman filter based approach for tracking was implemented with an SVM based spatter extraction.

In this paper, we present an approach for spatter tracking in the scenario of welding thick steel, by a combination of a smart camera for on-line processing and spatter event tracking as an off-line process, enabling automated analysis of the spatter behavior in welding. Different alternatives for spatter segmentation and tracking are considered in the context of high power laser welding and traditional manual metal-arc welding.

2. SPATTER EXTRACTION AND SEGMENTATION

On-line analysis

The huge difference between the intensity of light radiated from the welding zone compared to the surrounding areas causes a challenge for welding spatter segmentation and tracking. The objective is to enable the spatter object segmentation both within the very low and the very high intensity regions. This can be difficult to achieve because of the high intra-scene dynamic variation. Successful image capture may still be performed with the aid of active laser illumination [Fen09a]. Alternatively, a smart camera adapting to the intensity variation locally can be used.

Our approach is to use a high-speed smart camera, which enables the imaging of the welding zone by adjusting the integration time of the individual pixels according to their nearby local surroundings [Lah13a]. After this, the camera is used to extract the edge information from the second order spatial neighbors against an adjustable threshold. The KOVA1 array [KOV] provides a 96x96 resolution pixel parallel mixed-mode imaging processor, with a processor per pixel image processing functionality. In order to achieve a high enough frame rate for this application, only binary (1-bit) image sequence was used. Using gray scale images would limit the frame rate by filling up the bandwith of the 100 Mb/s Ethernet connection from the camera to the PC. Additionally, the required processing (edge detection) is performed faster on the camera than on the FPGA or the PC.

Off-line analysis

The spatter segmentation was performed as an off-line process in Matlab. Two slightly different methods for spatter segmentation are considered. The first is to apply Hough transform directly on the extracted edge images. This allows capturing the center location and perimeter of the spatters. The disadvantage of the method is that spatters which are not circular cannot be detected correctly. However, the method can also be implemented on FPGA for high-speed analysis.

The second considered approach is first to apply an additional morphological processing step to the extracted binary images (a morphological closing), in combination with connected component labeling. The closing operation could also be implemented on-line. The extracted object regions are then tested by their eccentricity (‘regionprops’ in Matlab R2012b). An additional step considered in this paper is to require that the center pixel of the object is of adjacent polarity compared to the edge pixels. This may be used to refine the circularity of the object, since it can be assumed that the spatters are, in general, symmetric and the edge image provided by the adaptive integration of the camera provides an object with a hole in the center.

3. SPATTER TRACKING

The proposed tracking algorithm is implemented in Matlab, and described in the following. First, the segmented objects are extracted as described in the previous section. A list containing the elements of initial spatter properties (SPL1) is created. Each spatter is assigned an individual ID number (in phase 3) and spatter area property, which describes a circular region in the case of Hough transform, and otherwise a rectangular region. TSPL_PRE (previous tracked spatter list) is initialized also to zero in this step. The steps of the tracking algorithm are explained below (1-5).

1) Initialize TSPL = 0 (Tracked spatter list) and read SPL2 (Spatter list 2) based segmented spatters at this frame.

2) Compare SPL2 to TSPL_PRE according to criterion IS_NEAR and update the matching spatters to TSPL. Remove these spatters from SPL2.

3) Compare the remaining SPL2 to SPL1, initialize and update those with property IS_NEAR_INIT to TSPL. Initialize spatter ID.

4) SPL1=SPL2, TSPL_PRE=TSPL

5) If frames left, Go back to phase 1

To initiate the tracking, the test IS_NEAR_INIT considers the spatter locations and areas only, without considering the previous displacement vector. In this phase, we use Euclidean maximum distance requirement of below 15 pixels. The test IS_NEAR performs the following action. The previous displacement vector (See Fig. 1) of the spatter under tracking has been stored and it is compared by using the IS_NEAR criteria to each of the new spatter observations. We use a Euclidean distance requirement of < 5 pixels. The assumption is that the spatters roughly propagate with constant velocity. An additional property which is checked is that the size of the new spatter has to be above 0.5 times and below 2 times of the size of the old spatter. The time which the spatter has been tracked (in number of frames) is also stored. All of the tracked spatters are collected to a list for later analysis. The post-processing step incorporates also a possibility to consider the average sizes and velocities of the spatters under test.

Observe that we use the same tracking algorithm in both cases i.e. with the connected component labeling based segmentation and with the Hough transform based segmentation. Thus, the segmentation algorithm, which could be performed on the camera or on an FPGA, is not affected by the tracking algorithm. Further possibilities of HW accelerating the tracking algorithm are discussed in section 6.

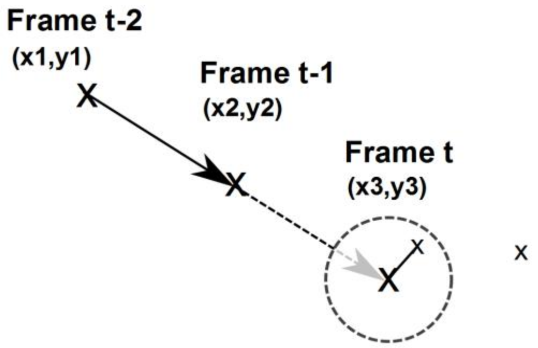


Figure 1. The test for proximity between two spatters. The estimated new center location of the spatter according to previous frames is compared to each of the candidate spatters in Frame t according to Euclidean distance (partly reminding the Kalman filter [Jäg08a]).

4. EXPERIMENTAL SETUP

Manual metal-arc welding set-up

The first tests were performed with manual arc welding, where the number of spatters is very large. The experimental setup in this case was initially targeted to test the properties of the camera, but the excessive spatter formation was shown to be suitable also for testing the spatter tracking algorithm. The frame rate used was 1408 Fr/s (with a difference between two frames of approximately 710us). Fig. 2 represents an example of the adaptive camera capture and the corresponding edge image captured on-line.

Laser welding set-up

The second test case used was a coaxial setup of full-penetration mode laser welding of 4mm thick steel. The laser power applied was selected so that full penetration was achieved, 4.5kW on 12mm/s and 6kW on 20mm/s welding speeds. The number of spatters was very low, which created challenges for the automated spatter analysis due to a relatively large portion of vapor plume. A partial penetration mode is expected to produce more spatters, which will be studied in the future. The frame-rate applied was approximately 3500 Fr/s.

5. EXPERIMENTS

Manual metal-arc welding tests

Hough transform was found to suite well to the case of manual metal arc welding, since the spatters were typically circular (See Fig. 2). The considered radiuses were from 2 to 10, with a Hough sensitivity parameter (HS) of 0.85 (the available implementation in Matlab R2012b). However, large spatters (e.g. r>6) were very rare, due to larger imaging area in comparison with the laser welding. Connected component labeling, which was applied after a binary closing with a 3x3 mask of all ones was used as a reference method. The object elongatedness was required in this case to be below 0.9.

Since it was not possible to manually label the whole welding sequence, the performance of the tracking was estimated so that a ground truth was generated and the overall spatter count returned by the algorithm (i.e. TP+FP) was determined at constant 100 frames steps (see Fig. 4, index 1 indicates frames 1-100 etc.). Note also that the ground truth contains all spatters, but the tracking algorithm only counts spatters included in at least 3 succeeding frames.

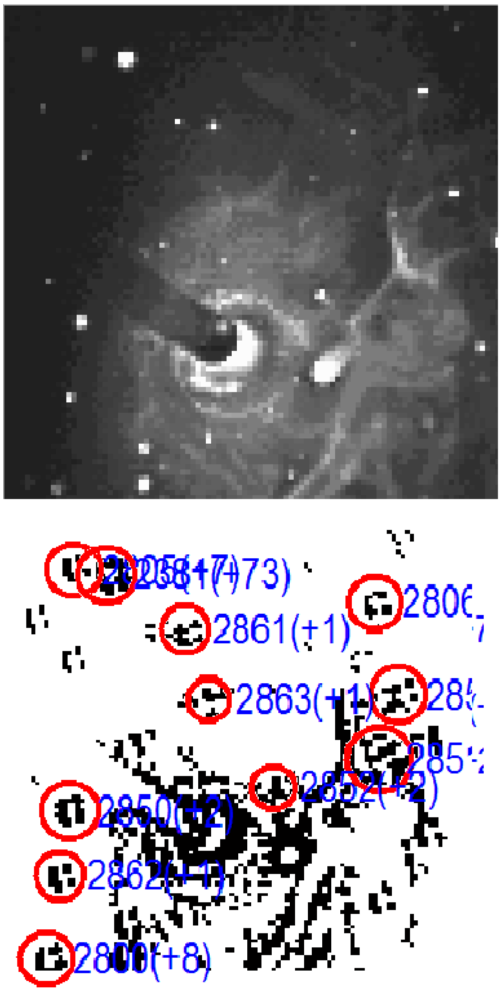


Figure 2. The arc welding set-up. The image on the top represents the welding zone captured by the adaptive integration. On the bottom the edges extracted on-line and the tracked spatter IDs extracted off-line.

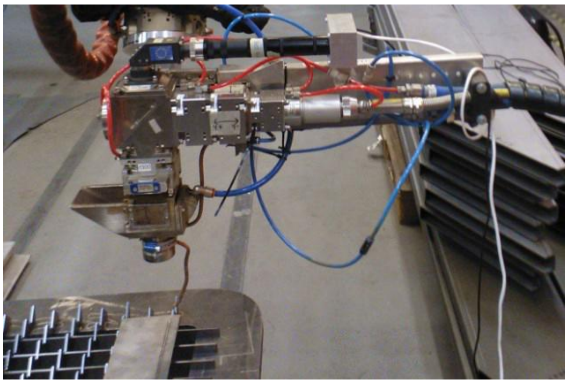


Figure 3. The laser welding test setup. The camera is attached to the welding head (from Precitec) through a 90 degree mirror and focusing lens. The fibre laser is brought from a collimator through 90 degree mirror optics to the work piece. The focal position of the laser was -4mm (below the surface).

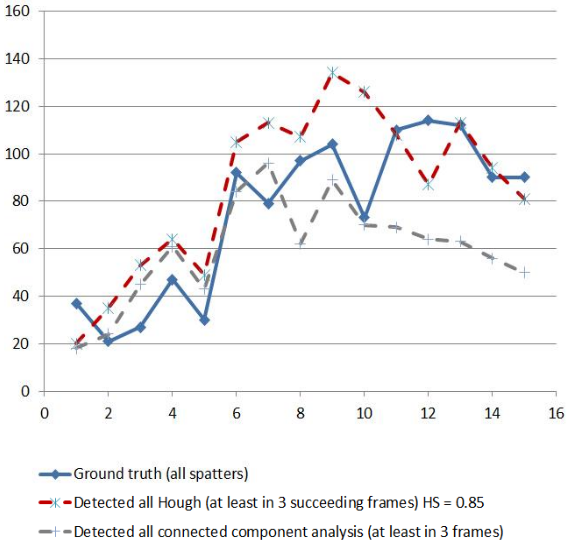


Figure 4. An indicative chart on the ground truth and the overall number of detected spatters in arc welding in frames 1-1500 at 100 frame intervals.

In the tests, increasing the requirement of the number of consecutive tracked frames could be used to filter out false positives caused by the vapor plume and to make it possible to increase the sensitivity of the segmentation. Also, a rectangular area near to the arc end was masked to reduce the effect of the plume. The ground truth was generated by three human experts in 500 frame intervals, which was quite challenging and could induce certain bias. It can be observed, that results follows the manually extracted spatter behavior. It appears in Fig.4, that when the number of spatters (and likely the amount of plume) is higher, the Hough transform gives better results. In Fig. 5, an angular histogram plot of the spatter directions for the same sequence is shown with Hough transform (3 succeeding tracked frames required).

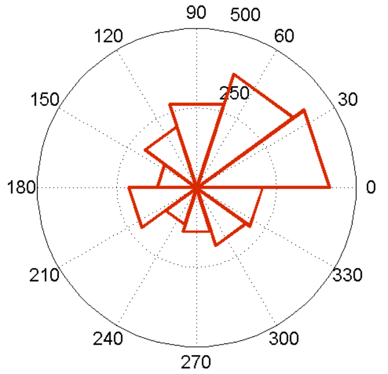


Figure 5. An angular histogram plot of the most common spatter directions in the arc welding test using Hough transform. The location of the arc was approximately in the location shown in Fig. 2.

Laser welding tests

Since the spatters in laser welding were not always circular, the connected component analysis in combination with spatter elongatedness test was applied with the same parameters as in arc welding. An additional requirement was that the center location of the spatters was white, which was shown to reduce the false alarm rate. The size of the detected spatters was limited to a square of minimum edge dimension of 5 pixels and maximum edge dimension of 20 pixels.

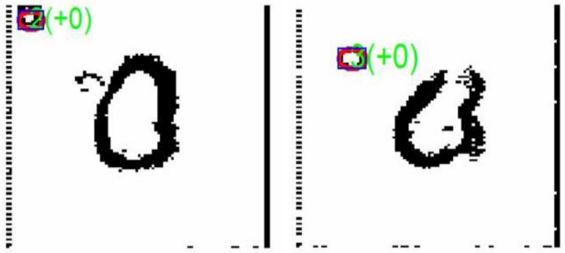


Figure 6. Examples of tracked spatters in laser welding scenario.

A ground truth of first 1000 frames of the sequence was manually collected, including only 13 spatters. The total amount of spatters was very low due to the used full penetration mode. A total of 9 spatters were detected correctly in comparison with the ground truth with one false alarm. Thus, a detection rate of 69.2% was obtained with false positive rate of 0.001/frame. Fig. 6 represents examples of tracked spatters in the laser welding scenario.

In [Nic11a] the false positive rates of spatter segmentation varied between 0.009/frame and 0.028/frame. The detection rate is not directly comparable, since the authors in [Nic11a] consider the detection of spatter events as an occurrence of one or more spatter within a single image. If there were one or more spatters present in the ground truth and also in the segmentation result, the match was considered as positive. In our case, individual spatters, which may locate across multiple succeeding frames are considered. However, it should be noted, that the tracking option used in our paper relaxes the requirement for spatter counting, since if a spatter is not found from an individual frame, the correct detection can still be attained in the succeeding frames.

6. ON POSSIBILITIES OF HW ACCELERATING THE ALGORITHMS

We are currently studying options for implementing the segmentation and the tracking algorithm on FPGA or by in-camera processing. While examining seam tracking, we have already demonstrated Hough transform based line detection on the FGPA attached to the camera [Sän14a]. The unit will be modified to detect circles, and then used as the detection engine for the tracking algorithm presented in this paper. Also the FPGA based acceleration of the tracking algorithm is possible. Currently on laptop PC with Matlab the algorithm runs approximately 30fps (without considering the time for the segmentation).

FPGA based processing

In order to implement the tracking algorithm on an FPGA the number of segmented spatters (elements in the lists) sets the limit for the efficient implementation. This is because the parameters of the spatter elements (e.g. 8-bit values) should be, preferably, mapped to the internal FPGA Block RAMs and not to external memory, which introduces extra latency. Assuming a BRAM size of 36kbit and list element size of 6*8 = 48 bits, the maximum number of elements in a list can be estimated. When mapped to a single BRAM the maximum length of a list would be 768, which should be enough for the application considered. The collection of the final spatter properties for further analysis could then be implemented by sending the data to a PC. At this point, as mentioned, only the adaptive image capture and spatter edge extraction were performed on-line.

Camera based processing

The applied smart camera (KOVA1) includes pixel-level processing functionality based on Cellular Nonlinear Network (CNN)-type [Ros93a] parallel processing architecture. In the KOVA1 CMOS sensor chip, analog and digital (mixed-mode) computational circuitry, as well as binary (static) and analog (dynamic) memory is integrated physically together with each sensor diode, creating a focal plane cell [Lai11a]. The additional pixel-level circuitry naturally imposes some limits on practical image resolution vs. chip size (cost/yield), e.g. the size of the commercial KOVA1 sensor-processor array is 96x96 pixels, while the largest published research implementation with a similar cellular processing architecture is the SCAMP-5 with a 256x256 pixel array [Car13a].

Each pixel cell in the sensor array of the KOVA1 is directly connected to its nearest neighborhood and in some operations even to the second neighborhood. Information propagation over larger distances is also possible in an asynchronous or synchronous manner. The cellular connectivity together with local, pixel-level, memory and processing circuitry allows the implementation of multiple-stage image analysis operations on the pixel plane. The pixel-level parallel processing and the avoidance of data transfer to an external processor result in reduced energy consumption and more importantly considering the present application, greatly reduced processing delay. The pixel-level processing functionality of the KOVA1 includes both grayscale and binary (1 bit) operations. The grayscale processing was used in this case to implement the image dynamic compression used in the capture of the welding data. The preprocessed grayscale image is segmented into a 1 bit binary representation to allow more efficient pixel-level shape and object analysis.

At this stage an edge detection operation has been used to extract relevant detail. The pixel level processing circuitry enables the implementation of sensor-level binary mathematical morphology and other similar analysis, as described in [Lai08a], as well as normal Boolean logic operations. By storing and combining intermediate result images in local pixel memory even complex spatial processing sequences can be performed on the sensor plane, including time-domain inter-frame analysis. Asynchronous binary propagation on the sensor plane can be used to perform regional operations such as holefilling, or morphological reconstruction very efficiently. By applying more binary analysis on the sensor-level, the processing demands at later stages (e.g. on an FPGA) can be greatly reduced, while retaining sufficient overall analysis speed.

If edge image based spatter segmentation is applied, one possibility to perform more computation at the camera level would be to use a holefiller template with pixel-level processing to detect roundish spatter objects, and then to extract the center locations of these objects for further processing. However, the holefiller template requires, that the edges of the objects are fully connected, which may not be always achievable. This could be avoided by using orientation selective propagation templates and by combining them or by applying a combination of dilation and skeleton extraction. Also blob based region extraction could be considered in the future.

7. DISCUSSION

The test set used in laser welding was challenging, since the spatters were not always circular, and due to second order neighborhood used in smart camera based segmentation. A second order neighborhood was used to better obtain the smooth edges of the spatters. Due to this, the edges of the spatters were thick, thus reducing the freedom in applying a larger closing mask. The purpose of the closing mask is to avoid a situation, where an individual spatter is detected as two distinct objects in the connected component labeling step. Also, if using too large closing mask, the centroid of the spatters becomes filled, and the segmentation process becomes less robust.

An approach for spatter segmentation and tracking in two welding scenarios was proposed. Our approach extends the work in [Nic11a] in that it also enables the spatter tracking and the analysis of the individual spatter properties. Future work consists of developing methods towards on-line spatter analysis. In comparison with work in [Lah13a] the methods proposed provide further robustness to false positives induced by the vapor plume. A limitation of the proposed approach in comparison with traditional multi object tracking is that the collision of two spatters may lead to erroneous tracking. This issue should be addressed in the future.

8. CONCLUSIONS

Methods for spatter segmentation and tracking in two welding scenarios were proposed in this paper. Hough transform was proposed for extracting spatters in the manual-arc welding scenario, while object elongatedness analysis was proposed for laser welding. The advantage of the Hough transform based segmentation is that it inherently avoids merging two distinct objects (or plume) nearby into the same object in the segmentation phase. The main advantage of the connected component analysis approach and the elongatedness test was that the objects need not to be circular, allowing flexibility for segmentation. Checking the polarity of the center pixel of the extracted object also increased the robustness of the segmentation algorithm.

9. ACKNOWLEDGMENTS

We would like to acknowledge the support from Lappeenranta University of Technology (LUT) and Machine Technology Center Turku in conducting the tests. The research was funded by Academy of Finland project no. 254430.

10. REFERENCES

[Nic11a] Nicolosi, L., Tetzlaff, R, Blug, A. , et.al, A Monitoring System for Laser Beam Welding Based on an Algorithm for Spatter Detection. Conf.proc. ECCTD'11, pp. 25-28.

[Nic09a] Nicolosi, L, Tetzlaff, R., F. Abt, et. al, New CNN based algorithms for the full penetration hole extraction in laser welding processes: Experimental results, IJCNN'09, pp. 2256-2263.

[Jäg08a] Jäger,. M, Humbert, S., Harmrecht, F. A. , Sputter Tracking for the Automatic Monitoring of Industrial Laser-Welding Processes. IEEE Trans. Industrial Electronics, 55, No.5, 2008.

[Fen09a] Fennander, H., Kyrki, V., Fellman, A., Salminen, A., Kälviäinen, H., Visual measurement and tracking in laser hybrid welding, Machine Vision and Applications 20, pp. 103–118, (2009).

[Lah13a] Lahdenoja, O., Säntti, T, Poikonen, J., Laiho, M., Paasio, A., Characterizing Spatters in Laser Welding of Thick Steel Using Motion Flow Analysis, SCIA'13, pp. 675-686.

[KOV] http://www.kovilta.fi

[Sän14a] Säntti, T., Lahdenoja, O., Paasio, A., Laiho, M., Poikonen, J., Line Detection on FPGA with parallel sensor-level segmentation, CNNA'14, to Appear.

[Ros93a] Roska, T., Chua, L. O., The CNN Universal Machine: An Analog Array Computer, IEEE Transactions on Circuits and Systems-II: Analog and Digital Signal Processing. 40(3), pp. 163-173, (1993)

[Lai11a] Laiho, M., Poikonen J., Paasio A.: MIPA4k: Mixed-Mode Cellular Processor Array. In: Zarandy A. (ed.): Focal-Plane Sensor-Processor Chips, Springer (2011)

[Car13a] Carey, S.J., Lopich, A., Barr, D.R.W., Wang, Bin, Dudek, P., A 100,000 fps vision sensor with embedded 535GOPS/W 256×256 SIMD processor array, Symposium on VLSI Circuits (VLSIC), pp. 182-183, (2013)

[Lai08a] Laiho, M., Paasio, A., Flak, J., et. al, Template Design for Cellular Nonlinear Networks With 1-Bit Weights. IEEE Trans. on Circuits and Systems 55(3), pp. 904-913 (2008)

Turku Centre for Computer Science
TUCS Dissertations

1. **Marjo Lipponen**, On Primitive Solutions of the Post Correspondence Problem
2. **Timo Käkölä**, Dual Information Systems in Hyperknowledge Organizations
3. **Ville Leppänen**, Studies on the Realization of PRAM
4. **Cunsheng Ding**, Cryptographic Counter Generators
5. **Sami Viitanen**, Some New Global Optimization Algorithms
6. **Tapio Salakoski**, Representative Classification of Protein Structures
7. **Thomas Långbacka**, An Interactive Environment Supporting the Development of Formally Correct Programs
8. **Thomas Finne**, A Decision Support System for Improving Information Security
9. **Valeria Mihalache**, Cooperation, Communication, Control. Investigations on Grammar Systems.
10. **Marina Waldén**, Formal Reasoning About Distributed Algorithms
11. **Tero Laihonen**, Estimates on the Covering Radius When the Dual Distance is Known
12. **Lucian Ilie**, Decision Problems on Orders of Words
13. **Jukkapekka Hekanaho**, An Evolutionary Approach to Concept Learning
14. **Jouni Järvinen**, Knowledge Representation and Rough Sets
15. **Tomi Pasanen**, In-Place Algorithms for Sorting Problems
16. **Mika Johnsson**, Operational and Tactical Level Optimization in Printed Circuit Board Assembly
17. **Mats Aspnäs**, Multiprocessor Architecture and Programming: The Hathi-2 System
18. **Anna Mikhajlova**, Ensuring Correctness of Object and Component Systems
19. **Vesa Torvinen**, Construction and Evaluation of the Labour Game Method
20. **Jorma Boberg**, Cluster Analysis. A Mathematical Approach with Applications to Protein Structures
21. **Leonid Mikhajlov**, Software Reuse Mechanisms and Techniques: Safety Versus Flexibility
22. **Timo Kaukoranta**, Iterative and Hierarchical Methods for Codebook Generation in Vector Quantization
23. **Gábor Magyar**, On Solution Approaches for Some Industrially Motivated Combinatorial Optimization Problems
24. **Linas Laibinis**, Mechanised Formal Reasoning About Modular Programs
25. **Shuhua Liu**, Improving Executive Support in Strategic Scanning with Software Agent Systems
26. **Jaakko Järvi**, New Techniques in Generic Programming – C++ is more Intentional than Intended
27. **Jan-Christian Lehtinen**, Reproducing Kernel Splines in the Analysis of Medical Data
28. **Martin Büchi**, Safe Language Mechanisms for Modularization and Concurrency
29. **Elena Troubitsyna**, Stepwise Development of Dependable Systems
30. **Janne Näppi**, Computer-Assisted Diagnosis of Breast Calcifications
31. **Jianming Liang**, Dynamic Chest Images Analysis
32. **Tiberiu Seceleanu**, Systematic Design of Synchronous Digital Circuits
33. **Tero Aittokallio**, Characterization and Modelling of the Cardiorespiratory System in Sleep-Disordered Breathing
34. **Ivan Porres**, Modeling and Analyzing Software Behavior in UML
35. **Mauno Rönkkö**, Stepwise Development of Hybrid Systems
36. **Jouni Smed**, Production Planning in Printed Circuit Board Assembly
37. **Vesa Halava**, The Post Correspondence Problem for Market Morphisms
38. **Ion Petre**, Commutation Problems on Sets of Words and Formal Power Series
39. **Vladimir Kvassov**, Information Technology and the Productivity of Managerial Work
40. **Frank Tétard**, Managers, Fragmentation of Working Time, and Information Systems

41. **Jan Manuch**, Defect Theorems and Infinite Words
42. **Kalle Ranto**, Z_4-Goethals Codes, Decoding and Designs
43. **Arto Lepistö**, On Relations Between Local and Global Periodicity
44. **Mika Hirvensalo**, Studies on Boolean Functions Related to Quantum Computing
45. **Pentti Virtanen**, Measuring and Improving Component-Based Software Development
46. **Adekunle Okunoye**, Knowledge Management and Global Diversity – A Framework to Support Organisations in Developing Countries
47. **Antonina Kloptchenko**, Text Mining Based on the Prototype Matching Method
48. **Juha Kivijärvi**, Optimization Methods for Clustering
49. **Rimvydas Rukšėnas**, Formal Development of Concurrent Components
50. **Dirk Nowotka**, Periodicity and Unbordered Factors of Words
51. **Attila Gyenesei**, Discovering Frequent Fuzzy Patterns in Relations of Quantitative Attributes
52. **Petteri Kaitovaara**, Packaging of IT Services – Conceptual and Empirical Studies
53. **Petri Rosendahl**, Niho Type Cross-Correlation Functions and Related Equations
54. **Péter Majlender**, A Normative Approach to Possibility Theory and Soft Decision Support
55. **Seppo Virtanen**, A Framework for Rapid Design and Evaluation of Protocol Processors
56. **Tomas Eklund**, The Self-Organizing Map in Financial Benchmarking
57. **Mikael Collan**, Giga-Investments: Modelling the Valuation of Very Large Industrial Real Investments
58. **Dag Björklund**, A Kernel Language for Unified Code Synthesis
59. **Shengnan Han**, Understanding User Adoption of Mobile Technology: Focusing on Physicians in Finland
60. **Irina Georgescu**, Rational Choice and Revealed Preference: A Fuzzy Approach
61. **Ping Yan**, Limit Cycles for Generalized Liénard-Type and Lotka-Volterra Systems
62. **Joonas Lehtinen**, Coding of Wavelet-Transformed Images
63. **Tommi Meskanen,** On the NTRU Cryptosystem
64. **Saeed Salehi**, Varieties of Tree Languages
65. **Jukka Arvo**, Efficient Algorithms for Hardware-Accelerated Shadow Computation
66. **Mika Hirvikorpi**, On the Tactical Level Production Planning in Flexible Manufacturing Systems
67. **Adrian Costea**, Computational Intelligence Methods for Quantitative Data Mining
68. **Cristina Seceleanu**, A Methodology for Constructing Correct Reactive Systems
69. **Luigia Petre**, Modeling with Action Systems
70. **Lu Yan**, Systematic Design of Ubiquitous Systems
71. **Mehran Gomari**, On the Generalization Ability of Bayesian Neural Networks
72. **Ville Harkke**, Knowledge Freedom for Medical Professionals – An Evaluation Study of a Mobile Information System for Physicians in Finland
73. **Marius Cosmin Codrea**, Pattern Analysis of Chlorophyll Fluorescence Signals
74. **Aiying Rong**, Cogeneration Planning Under the Deregulated Power Market and Emissions Trading Scheme
75. **Chihab BenMoussa**, Supporting the Sales Force through Mobile Information and Communication Technologies: Focusing on the Pharmaceutical Sales Force
76. **Jussi Salmi**, Improving Data Analysis in Proteomics
77. **Orieta Celiku**, Mechanized Reasoning for Dually-Nondeterministic and Probabilistic Programs
78. **Kaj-Mikael Björk**, Supply Chain Efficiency with Some Forest Industry Improvements
79. **Viorel Preoteasa**, Program Variables – The Core of Mechanical Reasoning about Imperative Programs
80. **Jonne Poikonen**, Absolute Value Extraction and Order Statistic Filtering for a Mixed-Mode Array Image Processor
81. **Luka Milovanov**, Agile Software Development in an Academic Environment
82. **Francisco Augusto Alcaraz Garcia**, Real Options, Default Risk and Soft Applications
83. **Kai K. Kimppa**, Problems with the Justification of Intellectual Property Rights in Relation to Software and Other Digitally Distributable Media
84. **Dragoş Truşcan**, Model Driven Development of Programmable Architectures
85. **Eugen Czeizler**, The Inverse Neighborhood Problem and Applications of Welch Sets in Automata Theory

86. **Sanna Ranto**, Identifying and Locating-Dominating Codes in Binary Hamming Spaces
87. **Tuomas Hakkarainen**, On the Computation of the Class Numbers of Real Abelian Fields
88. **Elena Czeizler**, Intricacies of Word Equations
89. **Marcus Alanen**, A Metamodeling Framework for Software Engineering
90. **Filip Ginter**, Towards Information Extraction in the Biomedical Domain: Methods and Resources
91. **Jarkko Paavola**, Signature Ensembles and Receiver Structures for Oversaturated Synchronous DS-CDMA Systems
92. **Arho Virkki**, The Human Respiratory System: Modelling, Analysis and Control
93. **Olli Luoma**, Efficient Methods for Storing and Querying XML Data with Relational Databases
94. **Dubravka Ilić**, Formal Reasoning about Dependability in Model-Driven Development
95. **Kim Solin**, Abstract Algebra of Program Refinement
96. **Tomi Westerlund**, Time Aware Modelling and Analysis of Systems-on-Chip
97. **Kalle Saari**, On the Frequency and Periodicity of Infinite Words
98. **Tomi Kärki**, Similarity Relations on Words: Relational Codes and Periods
99. **Markus M. Mäkelä**, Essays on Software Product Development: A Strategic Management Viewpoint
100. **Roope Vehkalahti**, Class Field Theoretic Methods in the Design of Lattice Signal Constellations
101. **Anne-Maria Ernvall-Hytönen**, On Short Exponential Sums Involving Fourier Coefficients of Holomorphic Cusp Forms
102. **Chang Li**, Parallelism and Complexity in Gene Assembly
103. **Tapio Pahikkala**, New Kernel Functions and Learning Methods for Text and Data Mining
104. **Denis Shestakov**, Search Interfaces on the Web: Querying and Characterizing
105. **Sampo Pyysalo**, A Dependency Parsing Approach to Biomedical Text Mining
106. **Anna Sell**, Mobile Digital Calendars in Knowledge Work
107. **Dorina Marghescu**, Evaluating Multidimensional Visualization Techniques in Data Mining Tasks
108. **Tero Säntti**, A Co-Processor Approach for Efficient Java Execution in Embedded Systems
109. **Kari Salonen**, Setup Optimization in High-Mix Surface Mount PCB Assembly
110. **Pontus Boström**, Formal Design and Verification of Systems Using Domain-Specific Languages
111. **Camilla J. Hollanti**, Order-Theoretic Mehtods for Space-Time Coding: Symmetric and Asymmetric Designs
112. **Heidi Himmanen**, On Transmission System Design for Wireless Broadcasting
113. **Sébastien Lafond**, Simulation of Embedded Systems for Energy Consumption Estimation
114. **Evgeni Tsivtsivadze**, Learning Preferences with Kernel-Based Methods
115. **Petri Salmela**, On Commutation and Conjugacy of Rational Languages and the Fixed Point Method
116. **Siamak Taati**, Conservation Laws in Cellular Automata
117. **Vladimir Rogojin**, Gene Assembly in Stichotrichous Ciliates: Elementary Operations, Parallelism and Computation
118. **Alexey Dudkov**, Chip and Signature Interleaving in DS CDMA Systems
119. **Janne Savela**, Role of Selected Spectral Attributes in the Perception of Synthetic Vowels
120. **Kristian Nybom**, Low-Density Parity-Check Codes for Wireless Datacast Networks
121. **Johanna Tuominen**, Formal Power Analysis of Systems-on-Chip
122. **Teijo Lehtonen**, On Fault Tolerance Methods for Networks-on-Chip
123. **Eeva Suvitie**, On Inner Products Involving Holomorphic Cusp Forms and Maass Forms
124. **Linda Mannila**, Teaching Mathematics and Programming – New Approaches with Empirical Evaluation
125. **Hanna Suominen**, Machine Learning and Clinical Text: Supporting Health Information Flow
126. **Tuomo Saarni**, Segmental Durations of Speech
127. **Johannes Eriksson**, Tool-Supported Invariant-Based Programming

128. **Tero Jokela**, Design and Analysis of Forward Error Control Coding and Signaling for Guaranteeing QoS in Wireless Broadcast Systems
129. **Ville Lukkarila**, On Undecidable Dynamical Properties of Reversible One-Dimensional Cellular Automata
130. **Qaisar Ahmad Malik**, Combining Model-Based Testing and Stepwise Formal Development
131. **Mikko-Jussi Laakso**, Promoting Programming Learning: Engagement, Automatic Assessment with Immediate Feedback in Visualizations
132. **Riikka Vuokko**, A Practice Perspective on Organizational Implementation of Information Technology
133. **Jeanette Heidenberg**, Towards Increased Productivity and Quality in Software Development Using Agile, Lean and Collaborative Approaches
134. **Yong Liu**, Solving the Puzzle of Mobile Learning Adoption
135. **Stina Ojala**, Towards an Integrative Information Society: Studies on Individuality in Speech and Sign
136. **Matteo Brunelli**, Some Advances in Mathematical Models for Preference Relations
137. **Ville Junnila**, On Identifying and Locating-Dominating Codes
138. **Andrzej Mizera**, Methods for Construction and Analysis of Computational Models in Systems Biology. Applications to the Modelling of the Heat Shock Response and the Self-Assembly of Intermediate Filaments.
139. **Csaba Ráduly-Baka**, Algorithmic Solutions for Combinatorial Problems in Resource Management of Manufacturing Environments
140. **Jari Kyngäs**, Solving Challenging Real-World Scheduling Problems
141. **Arho Suominen**, Notes on Emerging Technologies
142. **József Mezei**, A Quantitative View on Fuzzy Numbers
143. **Marta Olszewska**, On the Impact of Rigorous Approaches on the Quality of Development
144. **Antti Airola**, Kernel-Based Ranking: Methods for Learning and Performace Estimation
145. **Aleksi Saarela**, Word Equations and Related Topics: Independence, Decidability and Characterizations
146. **Lasse Bergroth**, Kahden merkkijonon pisimmän yhteisen alijonon ongelma ja sen ratkaiseminen
147. **Thomas Canhao Xu**, Hardware/Software Co-Design for Multicore Architectures
148. **Tuomas Mäkilä**, Software Development Process Modeling – Developers Perspective to Contemporary Modeling Techniques
149. **Shahrokh Nikou**, Opening the Black-Box of IT Artifacts: Looking into Mobile Service Characteristics and Individual Perception
150. **Alessandro Buoni**, Fraud Detection in the Banking Sector: A Multi-Agent Approach
151. **Mats Neovius**, Trustworthy Context Dependency in Ubiquitous Systems
152. **Fredrik Degerlund**, Scheduling of Guarded Command Based Models
153. **Amir-Mohammad Rahmani-Sane**, Exploration and Design of Power-Efficient Networked Many-Core Systems
154. **Ville Rantala**, On Dynamic Monitoring Methods for Networks-on-Chip
155. **Mikko Pelto**, On Identifying and Locating-Dominating Codes in the Infinite King Grid
156. **Anton Tarasyuk**, Formal Development and Quantitative Verification of Dependable Systems
157. **Muhammad Mohsin Saleemi**, Towards Combining Interactive Mobile TV and Smart Spaces: Architectures, Tools and Application Development
158. **Tommi J. M. Lehtinen**, Numbers and Languages
159. **Peter Sarlin**, Mapping Financial Stability
160. **Alexander Wei Yin**, On Energy Efficient Computing Platforms
161. **Mikołaj Olszewski**, Scaling Up Stepwise Feature Introduction to Construction of Large Software Systems
162. **Maryam Kamali**, Reusable Formal Architectures for Networked Systems
163. **Zhiyuan Yao**, Visual Customer Segmentation and Behavior Analysis – A SOM-Based Approach
164. **Timo Jolivet**, Combinatorics of Pisot Substitutions
165. **Rajeev Kumar Kanth**, Analysis and Life Cycle Assessment of Printed Antennas for Sustainable Wireless Systems
166. **Khalid Latif**, Design Space Exploration for MPSoC Architectures

167. **Bo Yang**, Towards Optimal Application Mapping for Energy-Efficient Many-Core Platforms
168. **Ali Hanzala Khan**, Consistency of UML Based Designs Using Ontology Reasoners
169. **Sonja Leskinen**, m-Equine: IS Support for the Horse Industry
170. **Fareed Ahmed Jokhio**, Video Transcoding in a Distributed Cloud Computing Environment
171. **Moazzam Fareed Niazi**, A Model-Based Development and Verification Framework for Distributed System-on-Chip Architecture
172. **Mari Huova**, Combinatorics on Words: New Aspects on Avoidability, Defect Effect, Equations and Palindromes
173. **Ville Timonen**, Scalable Algorithms for Height Field Illumination
174. **Henri Korvela**, Virtual Communities – A Virtual Treasure Trove for End-User Developers
175. **Kameswar Rao Vaddina**, Thermal-Aware Networked Many-Core Systems
176. **Janne Lahtiranta**, New and Emerging Challenges of the ICT-Mediated Health and Well-Being Services
177. **Irum Rauf**, Design and Validation of Stateful Composite RESTful Web Services
178. **Jari Björne**, Biomedical Event Extraction with Machine Learning
179. **Katri Haverinen**, Natural Language Processing Resources for Finnish: Corpus Development in the General and Clinical Domains
180. **Ville Salo**, Subshifts with Simple Cellular Automata
181. **Johan Ersfolk**, Scheduling Dynamic Dataflow Graphs
182. **Hongyan Liu**, On Advancing Business Intelligence in the Electricity Retail Market
183. **Adnan Ashraf**, Cost-Efficient Virtual Machine Management: Provisioning, Admission Control, and Consolidation
184. **Muhammad Nazrul Islam**, Design and Evaluation of Web Interface Signs to Improve Web Usability: A Semiotic Framework
185. **Johannes Tuikkala**, Algorithmic Techniques in Gene Expression Processing: From Imputation to Visualization
186. **Natalia Díaz Rodríguez**, Semantic and Fuzzy Modelling for Human Behaviour Recognition in Smart Spaces. A Case Study on Ambient Assisted Living
187. **Mikko Pänkäälä**, Potential and Challenges of Analog Reconfigurable Computation in Modern and Future CMOS
188. **Sami Hyrynsalmi**, Letters from the War of Ecosystems – An Analysis of Independent Software Vendors in Mobile Application Marketplaces
189. **Seppo Pulkkinen**, Efficient Optimization Algorithms for Nonlinear Data Analysis
190. **Sami Pyöttiälä**, Optimization and Measuring Techniques for Collect-and-Place Machines in Printed Circuit Board Industry
191. **Syed Mohammad Asad Hassan Jafri**, Virtual Runtime Application Partitions for Resource Management in Massively Parallel Architectures
192. **Toni Ernvall**, On Distributed Storage Codes
193. **Yuliya Prokhorova**, Rigorous Development of Safety-Critical Systems
194. **Olli Lahdenoja**, Local Binary Patterns in Focal-Plane Processing – Analysis and Applications

Turku Centre *for* Computer Science

Joukahaisenkatu 3-5 B, 20520 Turku, Finland | www. tucs.fi

University of Turku
Faculty of Mathematics and Natural Sciences
- Department of Information Technology
- Department of Mathematics and Statistics

Turku School of Economics
- Institute of Information Systems Science

Åbo Akademi University
Faculty of Science and Engineering
- Computer Engineering
- Computer Science

Faculty of Social Sciences, Business and Economics
- Information Systems

ISBN 978-952-12-3197-1
ISSN 1239-1883

Olli Lahdenoja

Local Binary Patterns in Focal-Plane Processing

Olli Lahdenoja

Local Binary Patterns in Focal-Plane Processing

Olli Lahdenoja

Local Binary Patterns in Focal-Plane Processing – Analysis and Applications