LAPPEENRANTA UNIVERSITY OF TECHNOLOGY
DEPARTMENT OF INFORMATION TECHNOLOGY

METHODS FOR REGION BASED PAPER SURFACE ROUGHNESS MEASUREMENT

Master Thesis

The topic of the Master's Thesis was approved by the Council of the Information Technology Department on September 13, 2006.

Examiners: Professor Arto Kaarna and Dr Tech Leena Ikonen

Supervisor: Dr Tech Pekka Toivanen

Lappeenranta, November 27, 2006

Asko Alhoniemi
Ruotsalaisenraitti 3A5
53850 Lappeenranta
Tel. +358 50 4347942
aalhonie@lut.fi

TIIVISTELMÄ

Lappeenrannan teknillinen yliopisto
Tietotekniikan osasto
Asko Alhoniemi
Alueisiin perustuvia menetelmiä paperin pinnan karheuden mittaamiseen
Diplomityö
2006
61 sivua, 21 kuvaa ja 19 taulukkoa

Tarkastajat: Professori Arto Kaarna ja TkT Leena Ikonen

Hakusanat: pinnan karheus, paperin karheus, segmentointi, clusterointi, alueiden yhdistäminen, etäisyysmuunnos

Paperin pinnan karheus on yksi paperin laatukriteereistä. Sitä mitataan fyysisesti paperin pintaa mittaavien laitteiden ja optisten laitteiden avulla. Mittaukset vaativat laboratorio-olosuhteita, mutta nopeammille, suoraan linjalla tapahtuville mittauksilla olisi tarvetta paperiteollisuudessa. Paperin pinnan karheus voidaan ilmaista yhtenä näytteelle kohdistuvana karheusarvona. Tässä työssä näyte on jaettu merkitseviin alueisiin, ja jokaiselle alueelle on laskettu erillinen karheusarvo.

Karheuden mittaukseen on käytetty useita menetelmiä. Yleisesti hyväksyttyä tilastollista menetelmää on käytetty tässä työssä etäisyysmuunnoksen lisäksi. Paperin pinnan karheuden mittauksessa on ollut tarvetta jakaa analysoitava näyte karheuden perusteella alueisiin. Aluejaon avulla voidaan rajata näytteestä selvästi karheampana esiintyvät alueet. Etäisyysmuunnos tuottaa alueita, joita on analysoitu. Näistä alueista on muodostettu yhtenäisiä alueita erilaisilla segmentointimenetelmillä. PNN -menetelmään (Pairwise Nearest Neighbor) ja naapurialueiden yhdistämiseen perustuvia algoritmeja on käytetty. Alueiden jakamiseen ja yhdistämiseen perustuvaa lähestymistapaa on myös tarkasteltu.

Segmentoitujen kuvien validointi on yleensä tapahtunut ihmisen tarkastelemana. Tämän työn lähestymistapa on verrata yleisesti hyväksyttyä tilastollista menetelmää segmentoinnin tuloksiin. Korkea korrelaatio näiden tulosten välillä osoittaa onnistunutta segmentointia. Eri kokeiden tuloksia on verrattu keskenään hypoteesin testauksella.

Työssä on analysoitu kahta näytesarjaa, joiden mittaukset on suoritettu OptiTopolla ja profilometrillä. Etäisyysmuunnoksen aloitusparametrit, joita muutettiin kokeiden aikana, olivat aloituspisteiden määrä ja sijainti. Samat parametrimuutokset tehtiin kaikille algoritmeille, joita käytettiin alueiden yhdistämiseen. Etäisyysmuunnoksen jälkeen korrelaatio oli voimakkaampaa profilometrillä mitatuille näytteille kuin OptiTopolla mitatuille näytteille. Segmentoiduilla OptiTopo -näytteillä korrelaatio parantui voimakkaammin kuin profilometrinäytteillä. PNN -menetelmän tuottamilla tuloksilla korrelaatio oli paras.

ABSTRACT

Lappeenranta University of Technology
Department of Information Technology
Asko Alhoniemi
Methods for Region Based Paper Surface Roughness Measurement
Master's thesis
2006
61 pages, 21 figures and 19 tables.

Examiners: Professor Arto Kaarna and Dr Tech Leena Ikonen

Keywords: surface roughness, paper roughness, segmentation, clustering, region merging, distance transform

Paper surface roughness is one of the paper quality measures. It is measured with devices that measure a paper surface physically and with optical measuring devices. Measurements require laboratory conditions, but there is a need for faster on-line measurements in the paper industry. Paper surface roughness can be expressed as one roughness value per sample. In this work, a processed image is divided into meaningful regions and a separate roughness value is calculated for each of the regions.

There are several methods used for roughness measurements. Generally accepted statistical methods are applied in this work, as well as a recent distance transformation method. In the paper surface roughness measurement, there has been a need to divide an analyzed sample into regions that are based on their roughness. Based on the division, rough areas can be extracted from a sample. The distance transformation produces regions that are further analyzed. These regions are used to form homogeneous regions with different segmentation techniques. Pairwise Nearest Neighbor based methods and the method based on merging neighboring regions are used for the segmentation. The split and merge approach of the regions is also studied.

Validation of the segmented image is generally done visually. The approach of this work is to compare a generally accepted statistical method with the segmentation results. High correlation between the results of these two methods indicates successful segmentation. The results of different experiments are further compared using hypothesis testing.

Two sets of images of cardboard surfaces, which were measured with a profilometer and OptiTopo measuring devices, were analyzed in this work. The changed starting parameters for the distance transform were the amount and location of the starting points. The same starting parameter changes were applied on all region merging algorithms. The correlation for the profilometer samples was generally stronger than for the OptiTopo samples after the distance transform. The segmentation results of the OptiTopo images had higher improvement in correlation than the profilometer images. The Pairwise Nearest Neighbor transform based algorithm produced results with the strongest correlation.

Contents

ABBREVIATIONS AND SYMBOLS

$c(i,j)$	Mean gray level value
$d(p)$	Distance value
$G(p)$	Gray level of a pixel
H_0	Test hypothesis
H_1	Alternative hypothesis
p	Pearson's coefficient of correlation
R_q	RMS-roughness of a profile
$S(x,y)$	Gaussian weighting function
S_a	Arithmetic mean deviation of a surface
S_a(x,y)	Moving arithmetic mean deviation of a surface
S_q	RMS-roughness of a surface
z	z-value associated with probability in normal distribution function
z'	Fisher's conversion
z_0	Mean height value
$z(x)$	Profile height function
$Z(x,y)$	Surface height function
$cost(i,j)$	Merge cost
$cost_{tot}$	Total merge cost

$(\lambda_{xc}, \lambda_{yc})$	Cutoff wavelengths
μ	Mean
ρ	Correlation
σ^2	Variance
σ	Standard Deviation
σ'_z	Approximated standard deviation for Fisher's conversion

CCD	Charge coupled device
DTOCS	Distance Transform on Curved Space
Hex	Hexagon grid type
LWC paper	Light weight coated papers
Net	Net grid type
PNN	Pairwise Nearest Neighbor Algorithm
RGM	Region Merging Algorithm
RMS	Root Mean Square
SC paper	Supercalendered papers
SDTOCS	Split DTOCS Algorithm
SPM	Split and Merge Algorithm
Tile	Tile grid type
WDTOCS	Weighted DTOCS
1D	One dimensional
2D	Two dimensional
3D	Three dimensional

PREFACE

This thesis was completed in 2006 at the Laboratory of Information Processing, Department of Information Technology, Lappeenranta University of Technology, Finland. The thesis was a part of PapSurface-project funded by National Technology Agency of Finland (TEKES), Stora Enso, UPM, Metso Paper and Metso Automation.

I would like to thank Dr Tech Pekka Toivanen for the opportunity to prepare this Master's Thesis. I also would like to thank Dr Tech Leena Ikonen, Dr Tech Jarno Mielikäinen and Toni Kuparinen for the support and guidance during the preparation of this thesis. I also thank professor Arto Kaarna and all those, who were not mentioned here, but were involved in the process of preparing this work.

Lappeenranta, November 2006

Asko Alhoniemi

1 Introduction

Paper roughness is an important quality measure in the paper industry. Equality of the paper roughness during the production process ensures high quality of produced paper. High paper quality guarantees paper usability in the purpose it is made for.

At the moment paper roughness measurements require laboratory conditions. There is a need for faster implementations of roughness measurements. An automatic, on-line roughness measurement during the production process could cause significant economical savings. Immediate detection of change in paper quality could decrease the amount of low quality paper that is produced every time quality problems occur. On the way to this goal, optical measuring devices must be improved, and fast algorithms for processing measurement data must be developed.

In roughness measurement, the total roughness of the sample can be expressed by one roughness parameter. However, it may occur that the sample is mostly smooth, containing only relatively small rough areas. Those features may be lost, if the sample is considered as a whole. To find those features as well, there is a need to divide the sample into meaningful regions according to their qualities.

Paper surface can be considered as a texture. A selection of methods exists to extract different texture features from the image. In paper roughness measurement, these methods can be used to categorize the paper surface. Statistical methods have been applied to paper roughness measurement as well as distance transform methods. These methods are employed in this thesis for the roughness measurements of the samples. The segmentation of the distance transformation results for the paper samples are the main focus in this work. It is not known to be studied in any other research before this work. Several region growing methods are applied for post processing of the images. These methods are PNN (Pairwise Nearest Neighbor) based thresholding methods and an algorithm for merging neighboring regions. The split and merge approach is applied as well.

This thesis is divided into eight chapters. Chapter two introduces used roughness measurement methods, chapter three describes roughness analysis methods and chapter four explains used segmentation methods. In chapter five the validation of achieved results is discussed. Chapter six presents implemented experiments, the results are discussed briefly in chapter seven and drawn conclusions are presented in chapter eight.

2 Roughness Measurements

There are a couple of ways to measure paper surface roughness. These methods are used to find a characterization of the paper surface that is not flat. Roughness can be seen as a deviation from a flat reference plane, where all surface elements are in one plane in the ideal case [1]. Comparing a flat surface to the paper surface is in general a starting point for measuring devices.

At the moment there are three different paper surface roughness measurement approaches in the paper industry. They are air leak, optical and profilometry methods. They will be introduced shortly.

In air leak methods, the general principle is to measure the ability of the paper surface to resist an air stream flowing between the paper surface and another surface or measuring head. The measures affecting the results are air pressure, pressure of pressing the measuring head against the paper surface and the area of the measuring head [2]. This paper roughness measurement method is standardized [3].

The idea of a profilometer is to actually measure the surface variation. There is either a real stylus moved along the surface or a laser beam used as stylus in optical profilometers. Movements of stylus are based on actual surface variation in the path it is moved. Optical methods have reported [4] higher roughness values than mechanical. In mechanical methods the stylus size restricts the resolution to $10\ \mu m$ at the smallest. Therefore smaller variations on the surface are "low pass filtered" due to the stylus size.

A laser profilometer works in the following way. Laser light is focused on the paper surface. Light reflected from the surface is analyzed by a detector. An analyzer measures deviation from the ideal focus position [4].

The functioning principle of OptiTopo, described in [5], is different than in the previous methods. If position, direction and illumination of the light source are known, the topography of an image can be calculated. The charge coupled device (CCD) camera captures two images of the sample. A measured surface is illuminated from two opposite directions by grazing light. Paper is assumed to scatter diffuse light. Shadows and highlights caused by illumination are measured. The partial derivative of the surface results from the measurements, which is then integrated in the frequency domain to achieve the 3D topography image.

3 Computational Methods for Roughness Analysis

There are several methods for analyzing paper textures. Rodionov [6] found a correspondence between optical and physical roughness measurements. In his Master Thesis, laser specle patterns and microscopic images were compared. Kuparinen [7] has studied 2D (two dimensional) roughness analysis methods in his Master Thesis work. He compared statistical processing methods and fractal dimension analysis methods. These methods were applied on LWC (Light weight coated papers), SC (Supercalendered papers), coated cardboard and base board. Significant correlations were found between the different methods, but there was no clear evidence which of the methods produced the best results.

Ikonen [8] suggests the DTOCS (Distance Transform on Curved Space) method to be applied for paper roughness measurement. DTOCS and statistical methods have been compared for anisotropic grids in [9]. In that research, correlation has been found between DTOCS and statistical methods. In this work approach is based on that method. Both statistical methods and DTOCS are applied to the test data. In this chapter, these roughness analysis methods are described in detail.

3.1 Distance Transforms

In the image, subject to the analysis, the height values are considered as gray levels. The DTOCS method calculates distances along the image surface. Total distances are formed by summing local distances. Local distance means distance between two adjacent pixels. The distance consists of a horizontal component and a vertical component, which are summed. The horizontal component is the height difference between values, and the vertical component is always 1 because of the adjacency in local distance calculation. The calculation of the local distances is given by:

$$d(p_i, p_{i-1}) = |G(p_i) - G(p_{i-1})| + 1,\ p_{i-1}\ \in N_8(p_i) \tag{1}$$

where p is pixel coordinates and $G(p)$ is gray level of the pixel. As explained in [10], there are other local distance calculation methods, which improve accuracy in both horizontal and vertical component calculations. WDTOCS calculates the 3D (three dimensional)

Euclidean distance between adjacent pixels. The optimal DTOCS, introduced in [11], uses weighting factors for the horizontal local distance components, which improve the approximations of global distance values.

To calculate the global distances, local distances are used for optimal path construction from seed points to every pixel in the image. Seed points are pixels, where calculations are started. Their location and the amount is defined manually, except in the split and merge approach, which is introduced in the next chapter. Naturally, their distance values are 0 because the distance from a pixel to itself is 0. The priority pixel queue algorithm, presented in [10] is used in this work. It is used to calculate global distances for all pixels in the image. These distances are stored in the distance image. The algorithm is the following:

```
1.  Put all seed pixels with distance value 0 to priority queue.
2.  Initialize every other pixel than seed pixel to infinity.
3.  While queue is not empty.
    3.1.  Dequeue pixel with the smallest distance.
    3.2.  If dequeued value is higher than the value currently in that pixel, continue
          from 3.
    3.3.  Distance value $d(p, s)$ of a pixel $p$ from a seed pixel $s$ becomes final value in
          the distance image
    3.4.  Calculate distance $d(p, s) + d(p, x)$ to the adjacent pixels and enqueue them.
```

The distance image results from the calculation. The nearest neighbor transform is calculated simultaneously with the DTOCS. The distance image contains the distances from all the pixels to the nearest possible seed points. In order to know where the distance of each pixel originates from, a tessellation image is created during the calculations. In the tessellation image, each seed pixel has a label, and that label is given also to all those pixels that are closest to the seed pixel. Thus pixels labeled with the same label form a region.

The next step in roughness calculation is to calculate the roughness value for each of the regions. Values in the distance image are cumulative. To achieve the average height variation along the path, the distance image is normalized. Each pixel is divided by the distance along a flat surface, using exactly the same path along which shortest path calculations were made. From the normalized image, mean values are calculated for each region. This value is considered as the roughness value of the region in the image. By calculating the mean value for all regions in the image, one roughness value is given to the analyzed image.

3.2 Statistical Methods

The most common roughness measure is the root mean square deviation of surface height. In 1D case it is given in discrete form by

$$R_q^2 = \frac{1}{N}\sum_{j=1}^{N}(z(x_j) - z_0)^2, \tag{2}$$

where R_q is RMS-roughness of a profile, N is the amount of values considered in roughness calculation, $z(x)$ the local height value and z_0 the mean height value for given area [12]. This equation has the same form as the mean square error function (Equation (8)), introduced later with the PNN based method. In the 2D case, the roughness measure is

$$S_q^2 = \frac{1}{N_x N_y}\sum_{i=1}^{N_y}\sum_{j=1}^{N_x}(Z(x_j, y_i) - z_0)^2. \tag{3}$$

where S_q is RMS-roughness of a surface and Z(x,y) is the local height value on a surface.

To measure only the roughness, waviness must be extracted from the sample by using Gaussian filtering. The filter is calculated by a direct convolution with the Gaussian weighting function, and it is given by

$$S(x,y) = \frac{1}{\beta\lambda_{xc}\lambda_{yc}} exp\Bigg\{ -\frac{\pi}{\beta}\Bigg[\bigg(\frac{x}{\lambda_{xc}}\bigg)^2 + \bigg(\frac{y}{\lambda_{yc}}\bigg)^2\Bigg]\Bigg\} \tag{4}$$

where x and y are the positions from the center of the function, λ is the cutoff wavelength, which is 50% of the attenuation ratio and $\beta = ln2/\pi$ [13]. Recommended wavelengths are 0.25mm, 0.8mm and 2.5mm [14]. These wavelengths are also used in [9] to the paper roughness measurement.

To estimate the results of the methods applied in this work, a generally accepted method is needed for comparison. As suggested in [7], the statistical roughness parameter, which is the arithmetic mean deviation of the surface, is used to indicate sample roughness. The statistical roughness parameter is called a S_a roughness parameter. The calculation of the parameter in discrete form is given by

$$S_a = \frac{1}{N_x N_y} \sum_{i=1}^{N_y} \sum_{j=1}^{N_x} |Z(x_j, y_i)|. \tag{5}$$

In Equation (5), N_x and N_y are the number of data points in x and y direction and $Z(x, y)$ is the height value from the reference surface plane.

Above is explained how to calculate one roughness value. The same method can be applied within the sample as well, calculating the roughness value for each pixel. It is done by using a moving kernel. If using 5×5 kernel for example, as in this work, the equation would look like

$$S_a(x, y) = \frac{1}{5 \cdot 5} \sum_{i=-2}^{2} \sum_{j=-2}^{2} |Z((x-j), (y-i))|. \tag{6}$$

The pixel is the centroid of the kernel. The calculation averages the roughness in the neighborhood of a pixel. If the used kernel is enlarged, the significance of a single height value decreases. More exceptional values are lost with the larger kernel.

When the roughness value is calculated for all of the pixels within an image, these values can be further used for roughness parameter calculation. For example, for one region within an image one moving S_a roughness value can be calculated. It is the mean value of the roughness values within a region.

4 Segmentation Methods

Image segmentation is a technique used to improve image processing results. According to the definition in [15], the segmentation process means partitioning of the image into non-intersecting, homogeneous regions. In the segmented image, there are no two regions that would form a homogeneous region, if they were merged.

Image segmentation is based on gray levels, textures or edges. In this work, only texture based methods are used. This is due to the fact that the initial regions are produced by used roughness measurement method (DTOCS). From the resulting regions different features are calculated, and these features can be used to find similar regions. This approach naturally leads to use texture based methods. Another reason for selected approach is that no single pixel values can be used but rather regions that describe the surface roughness of the area.

4.1 Region merging

The amount of seed points determines how many resulting regions are achieved by using the DTOCS method. However, it is not clear how many regions should be set to achieve the best results. There are two possible approaches to the problem: to automatize the setting of seed points or to compose the result from the existing regions.

In this work both approaches mentioned above are used. DTOCS regions are used to initialize region merging algorithms. The split and merge approach tries to determine automatically, how many seed points are needed. Region merging is used to improve the results achieved by the DTOCS method. It is assumed that several similar regions are produced by that method. However, the regions that represent the similar data should be grouped into one region. Methods for grouping the regions spatially as well as methods which are not using spatial information are used in this work.

4.1.1 PNN based multiple thresholding

Gray [16] describes vector quantization as encoding of a set of data into a vector. Compression of data is achieved in the encoding process. Data can be reproduced later by decoding a vector. These vectors are called code vectors. A set of these vectors forms a codebook. As given in [17], the codebook initialization process starts by minimizing error within a cluster, which means finding the centroid of a cluster. If there are N data sets and N corresponding code vectors in the initial state, the data sets can be reduced to $N-1$ data sets by merging two regions. Regions to be merged are those that cause the minimal increase in the total error. A new code vector is chosen to minimize the error caused by encoding the new area into one code vector. Thus the new code vector is the centroid of two vectors representing merged regions.

As in [17], the increase in error when merging two clusters is given by

$$cost(i,j) = \frac{n_i n_j}{n_i + n_j} ||\mathbf{c_i} - \mathbf{c_j}||^2. \tag{7}$$

In the equation, n is the number of pixels, i and j are merged clusters and **c** is the centroid of the cluster. As in [18], the total error of the clustered image is given by

$$cost_{tot} = \frac{1}{N} \sum_{i=1}^{N} ||\mathbf{x_i} - \mathbf{c_{p_i}}||^2. \tag{8}$$

In the equation, N is the number of clusters, **x** represents input vectors in the image and p defines mapping of an input vector to the nearest code vector. In other words, calculating the total cost before and after merging of two clusters produces values that differ as much as $cost(i,j)$.

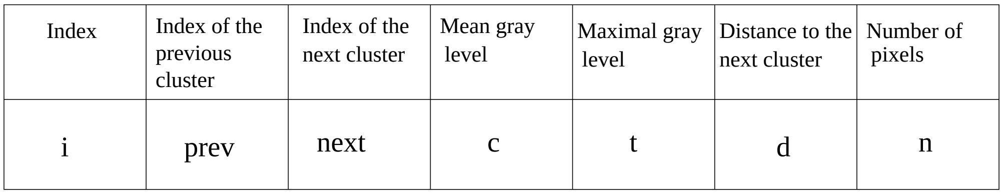
In the PNN based method, a code book is needed for merge operations. Data is encoded to code vectors. There is no need to decode vector data, since the code book is used only for merging decisions. Encoding is also the initialization stage of the PNN based algorithm. Merge decisions are made according to quantization error values. In every merge step are merged a pair of clusters that gives the smallest error value. This approach makes value differences smoother. Each merge step causes the range of the mean values to become smaller.

The fast pairwise nearest neighbor based algorithm introduced by [18], is a 1D multilevel thresholding algorithm. The initialization stage of the algorithm is based on the histogram, where values are divided to as many categories as there are values. The fact that only 1D thresholding is needed, makes it possible to design a more efficient algorithm than is the case with more dimensions. In 1D case, it is possible to set values along a line segment, instead of a matrix, which is the case for 2D data. Instead of grouping pixels with same value according to the histogram, DTOCS formed regions are grouped and sorted from the lowest mean value to the highest, as is visualized in Figure 2. After the initialization, the regions must be linked to each other to form a linked list. The linked list structure looks like in Figure 1. Every region must have link to previous and to next region, except the first and the last regions. The mean value must be saved as well as the cost value to the next region.

When all the regions are put and linked in the list, region merging can be started. An example that contains five regions is presented in Figure 2. Region merging can be seen as the removal of the threshold between two regions in the list. Every time two regions are merged, one threshold is removed. In the example, it would be T2 or T4, if the regions had equal amount of pixels. Merging is complete, when the amount of pixels in two regions is summed, links are updated and new mean value is calculated. New cost values must be calculated for the previous region and the new region. The amount of pixels is needed to calculate the new cost value (Equation (7)) and the new mean value, which is given by

$$c(i,j) = \frac{n_i c_i + n_j c_j}{n_i + n_j}. \tag{9}$$

The maximum value field is needed to define the range of values within one region. In the implementation with DTOCS method, there is no use for that value field (maximum gray level in Figure 1), since all pixels are marked in the tessellation image to belong to a certain region.



Index	Index of the previous cluster	Index of the next cluster	Mean gray level	Maximal gray level	Distance to the next cluster	Number of pixels
i	prev	next	c	t	d	n

Figure 1: Fields in the linked list [18]

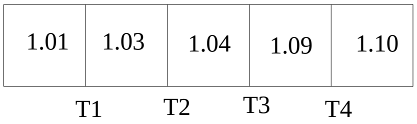


Figure 2: Thresholding the pixel range

The algorithm used in this research is the following. Prior to processing the image, the number of resulting regions must be set.

1. Form initial regions by using the DTOCS method.
2. Set regions to a linked list from the lowest mean value to the highest. Set initial values.
3. Calculate merge costs for each region to the next region in the list.
4. Put indexes into a minimum heap according to the cost value.
5. While the amount of regions is higher than defined,
    - 5.1. Remove region(i), region(prev(i)) and region(next(i)) from the minimum heap.
    - 5.2. Give region(i) and region(next(i)) the same label in the tessellation image.
    - 5.3. Calculate the new mean for the merged region.
    - 5.4. Calculate the new cost for the merged region and region(prev(i)) and put them into the heap.

The complexity of the algorithm is considered by dividing it into three steps. The initialization of the algorithm takes $O(NlogN)$ time, because there is only the 1D list of thresholds to be considered, and each of these values are put into heap. As stated in [19], all basic operations of the heap structure take less than $2logN$ comparisons. Because of the minimum heap structure, finding cluster pair to be merged takes only $O(1)$ time. Updating already known indexes in the linked list and the heap structure takes $O(logN)$ time. The total complexity of the algorithm is $O(NlogN)$.

The approach in this work consisted of an additional step for the PNN based algorithm. Before initialization, regions must be sorted from the lowest value to the highest (sorting required for step 2 in the algorithm presented above). If using the heapsort algorithm, for example, the complexity of sorting regions becomes $O(NlogN)$ [19]. The update of different images takes constant time with different amount of regions, since the complexity of updating images is only dependent on the sizes of processed images. Therefore, also modified algorithm used in this work has total complexity $O(NlogN)$.

The working principle of the PNN method is shown in Figure 3. Each image shows each merge step. The labels of the regions are shown as well. The image and the regions represent the lower left corner of the Figure 15, which is presented later in the Experiments chapter.

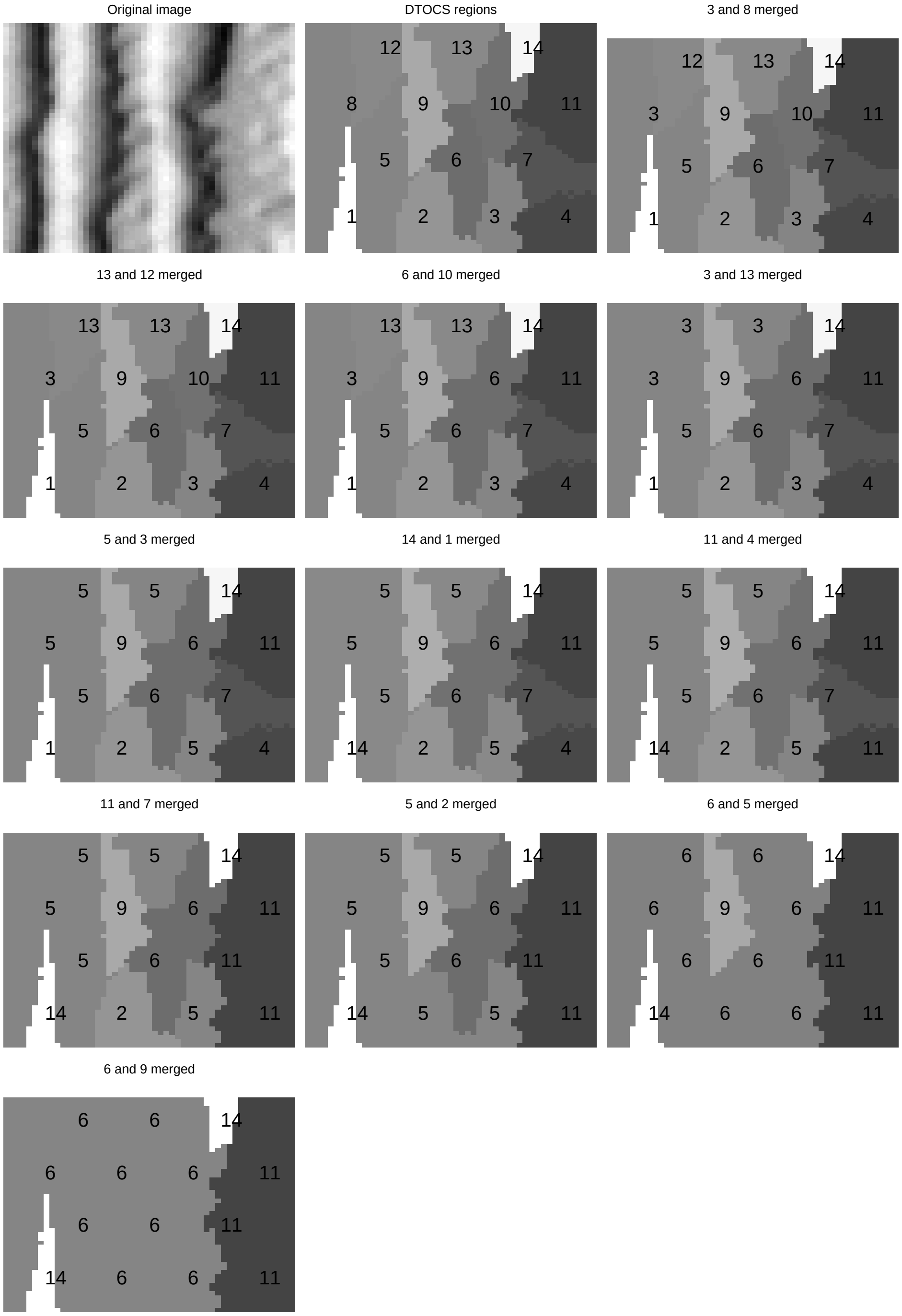


Figure 3: A PNN based merge procedure, step by step.

4.1.2 Lazy PNN

The PNN method becomes more complex, if the approached problem can not be considered as 1D data. In the 2D case, merging regions can not be considered as removing a threshold between two regions in the list, as was described in previous section. Instead, the distance matrix is used to restore distances from all regions to all other regions. In the matrix distances are symmetrical. Therefore, a triangular matrix is sufficient to present all distance values [20]. Virmajoki presents a few improvements to the PNN method. The fast exact PNN saves a pointer to the nearest neighbor for each region. Only the new region and those regions that had pointers to the old regions, need to have their distance values and nearest neighbor pointers updated, when two regions are merged. The method can be even further developed by introducing lazy update. It means that after each merge step old values are marked as out of date, and updated only when it is necessary. Therefore immediately after merging two regions, only the nearest neighbor of the new region must be updated. Pointers pointing to old regions are marked as out of date. As proved by Kaukoranta [21], the new merge cost can never be lower than the old merge costs. For example, consider 3 regions a, b and c, which have their merge cost calculated according to their mean values. If cost values $d(a,b) \leq d(a,c) \leq d(b,c)$, and regions a and b are merged, for resulting regions $d(a,c) \leq d(a+b,c)$ is always true. Other methods to improve PNN algorithm exists, but they are not discussed in this work.

In this work, the PNN based multiple threshold algorithm was modified to handle more than 1D data with some minor changes. The algorithm presented in the previous section would look like the following after the changes:

```
1.   Form initial regions by using the DTOCS method.
2.   Set regions to a list. Set initial values.
3.   Calculate merge costs from each region to all other regions.
4.   Choose the minimum cost.
5.   Put indexes into a minimum heap according to the cost value.
6.   While the amount of regions is higher than defined,
     6.1.  Remove region(i), from the minimum heap.
     6.2.  If region(i) has cost value that is "out of date", update it, put it into the heap
           and go to 5.Else continue
     6.3.  Give region(i) and region(next(i)) the same label in the tessellation image.
     6.4.  Calculate the new values for the merged region.
     6.5.  Calculate the new cost for the merged region, mark regions that point to
           region(i) and region(next(i)) as out of date. Put the new region into the heap.
```

The modification of the algorithm for 2D data increases its complexity. In the initial stage, for every region must be calculated the distance to every other region. This makes the complexity of the initialization to be $O(N^2)$. The heap initialization takes the same amount as in the 1D case: $O(NlogN)$. The update of the regions takes in the worst case

$O(N)$ time. Therefore, the total complexity of the algorithm increases to $O(N^2)$ in the 2D case.

4.1.3 Split and merge

To be able to better evaluate the DTOCS method, an alternative approach for region growing was chosen. The algorithm presented by [22] was used to find comparable results. It is called the split and merge algorithm (SPM) in this work. It consists of a split phase and a merge phase. The algorithm is also modified by replacing the split phase with the DTOCS, which is called the split DTOCS (SDTOCS) algorithm in this work.

Splitting of the image is very straightforward. The image is simply divided into four equal sized squares. Each of these squares is split again to four, and it is continued recursively as long as they fulfill the split criterion. The split criterion can be variance within the region, or as presented by [22], it can be the difference between the highest and the smallest value in the region. A square region is split as long as its variance exceeds a certain value or as long as the value difference exceeds preset value.

If a single value difference is used as a split criterion, there is a risk of doing the wrong split decision, if there is only one distorted pixel value that differs significantly from other pixels in the region. However, splitting should form a small region around that pixel, and if the merge phase works well, other regions nearby should be re-merged, if they are similar. The disadvantage of using variance for splitting decisions is the loss of small features within a large region, because in large areas they do not increase the variance enough.

The SDTOCS algorithm starts with calculating initial DTOCS regions. It is reasonable to choose relatively large regions to save processing time and let the algorithm decide when there is a need for further attention within an area. One initial region at a time is taken into consideration, and 4 new seed points are placed on it by using the same grid type as was used in the initial stage, if the split criterion was met. It is not always possible to place all 4 seed points, if the region is very irregular. After placing the seed points, new regions are calculated using the DTOCS within a region. New regions will have their boundaries inside the original old region. This approach is continued as long as the split criterion is met and the region is not too small.

Both algorithms (SDTOCS, SPM) are dependent on some statistical method, for example

S_a roughness. Split decisions are based on it rather than on the DTOCS normalized image. The value comparison in S_a roughness map is a reliable way to find variation within a certain area.

The split phase in both algorithms is manual, which means the split criterion must be set manually. They require human interaction: one has to evaluate how many initial regions are chosen before entering to the merge phase. The exact amount of initial regions is not known beforehand, since splitting is based on exceeding a certain threshold.

The merge phase of the algorithm is called the Region Merging Algorithm (RGM). In the merge phase the neighborhood criterion is followed: regions A and B are neighbors, if there exists at least one pixel in region A and one pixel in region B that are adjacent to each other. Adjacent pixels for a pixel are all the pixels that surround it, so there are 8 adjacent pixels at most. To find out all neighborhood relations for all regions, the whole image is scanned using a kernel, which is presented in Figure 4. With that kernel scanning of the image is started from the upper left corner. The black square is the starting pixel that is compared to its neighbors, which are white squares in the figure.

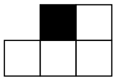
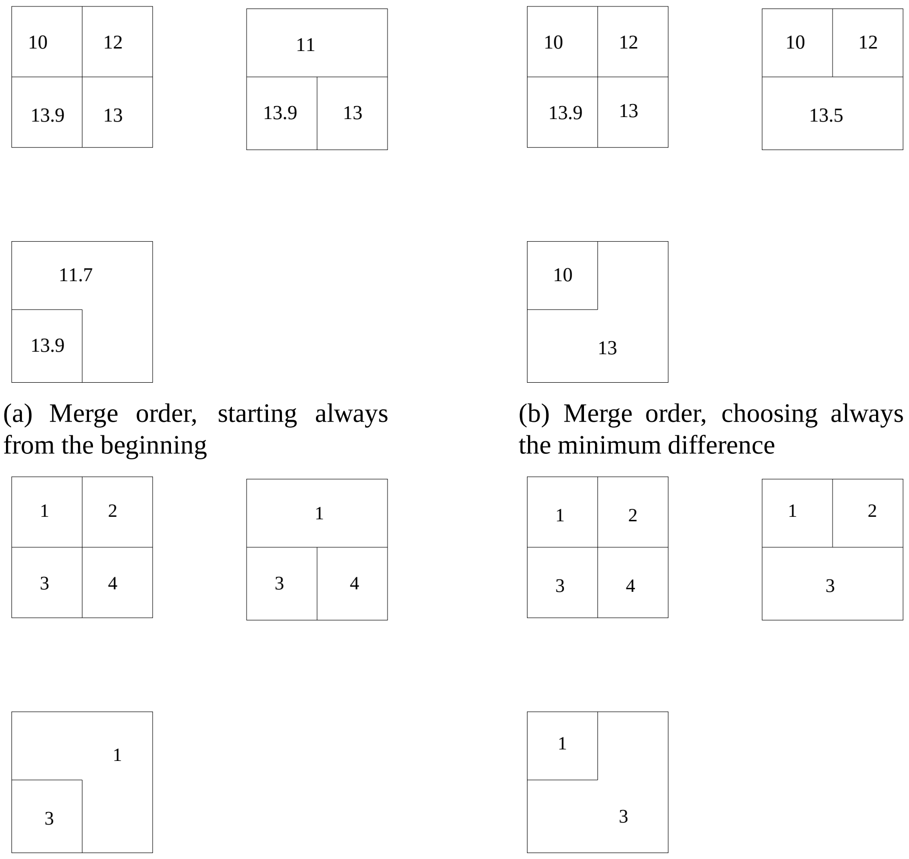


Figure 4: Scanning directions.

The result of the scanning is the neighborhood list that consists of all neighborhood relations to all regions. The next task is to determine the pair of regions to be merged. There are several approaches to the problem. One straightforward way is to start from the beginning of the regions until the end. If a pair of regions is found that have a lower merge cost than the threshold, the regions are merged and the list is started again from the beginning. After the merge operation, the necessary structures are updated and the scanning of the list is started from the beginning. The approach in this work is to find the minimum in every round, so the closest possible pair of regions is always merged. Willebeck [22] mentioned another approach, where the threshold value is increased in stages, to prevent wrong order of region merging.

In Figure 5 an example of merging order in region merging is shown. In image a) and b) are presented 3 merge stages: initial stage, the first merge operation and the second merge operation. Image c) presents labeling for three steps in image a) and image d) presents the labeling for three steps in image b). Values in the images are mean values. The regions have the same size in initial stage. Threshold is 2.1, and merge operation is done, if the difference between mean values of the regions is under the threshold. In image a) the merge order is always from the beginning of the list. First it is checked if regions labeled with 1 and 2 can be merged. As their difference is lower than the threshold, they are merged. Combined region will have label 1, and the new mean value. After the merge operation checking of the regions is started from the beginning again. Regions 1 and 4 are merged next. After the update no more regions are merged.

In the figure, in image b) regions with the smallest difference are chosen. Regions with labels 3 and 4 are chosen, and after update 2 and 3. This example shows how merging order does matter in some cases. The merging threshold is another human guided parameter in the approach that is used in this work. To achieve desired results, the appropriate threshold value needs to be found.



(a) Merge order, starting always from the beginning

(b) Merge order, choosing always the minimum difference

(c) Merge order, starting always from the beginning, labels

(d) Merge order, choosing always the minimum difference,labels

Figure 5: An example of different merge order.

The split algorithm used in this work is the following:

```
1. Calculate a $S_a$ roughness map.
2. Split into four regions.
    2.1. If the difference between the highest and the lowest value exceeds the threshold, and the size of the region is larger than the minimum allowed size for regions, go to 2. Else relabel region, calculate the mean value of the region and update the roughness map.
```

Merge algorithm (RGM) used in this work is the following:

```
1. Scan the tessellation image.
   If $(x, y)$ and $(x + 1, y - 1)$, $(x + 1, y)$, $(x, y - 1)$ or $(x - 1, y - 1)$ have different labels, the regions indicated by the labels are neighbors. Add to the neighborhood list, if not yet there.
2. For each pair in the neighborhood list, calculate the difference of the mean values.
3. Sort values according to the difference, from the lowest to the highest.
    3.1. If the lowest value is lower than the present threshold, merge regions that have the lowest value. Else finish.
    3.2. Update neighbors, update the tessellation image and the roughness map. Go to 3.
```

For the merge phase, initialization of the neighborhood list used in the form presented here has complexity $O(N^2)$, since preventing the redundant neighbors in the list requires scanning of the complete list every time a neighbor is added. Sorting of regions would take no more than $NlogN$ as was with the PNN based algorithm. Update of the neighborhood list takes the length of neighborhood list in the worst case, which has the complexity $O(N)$. Since neighbor pairs are not stored into heap structure, the minimum value must be found in every round. This takes $O(N)$ time. The total complexity of the merge phase is $O(N^2)$.

In Figure 6, an example of RGM is shown. Each image show each merge step. The labels of the regions are shown as well. The image and the regions represent the lower left corner of Figure 15. It is the same original image as in Figure 3. In this case it produces very similar result to the PNN based method, when there are 3 resulting regions for the PNN and 4 resulting regions for the RGM algorithm. The only difference is the mean value, which is considered as the roughness value for the region. It is different for regions 1 and 14, whereas it is the same in PNN for those regions, since the PNN algorithm united them as one region.

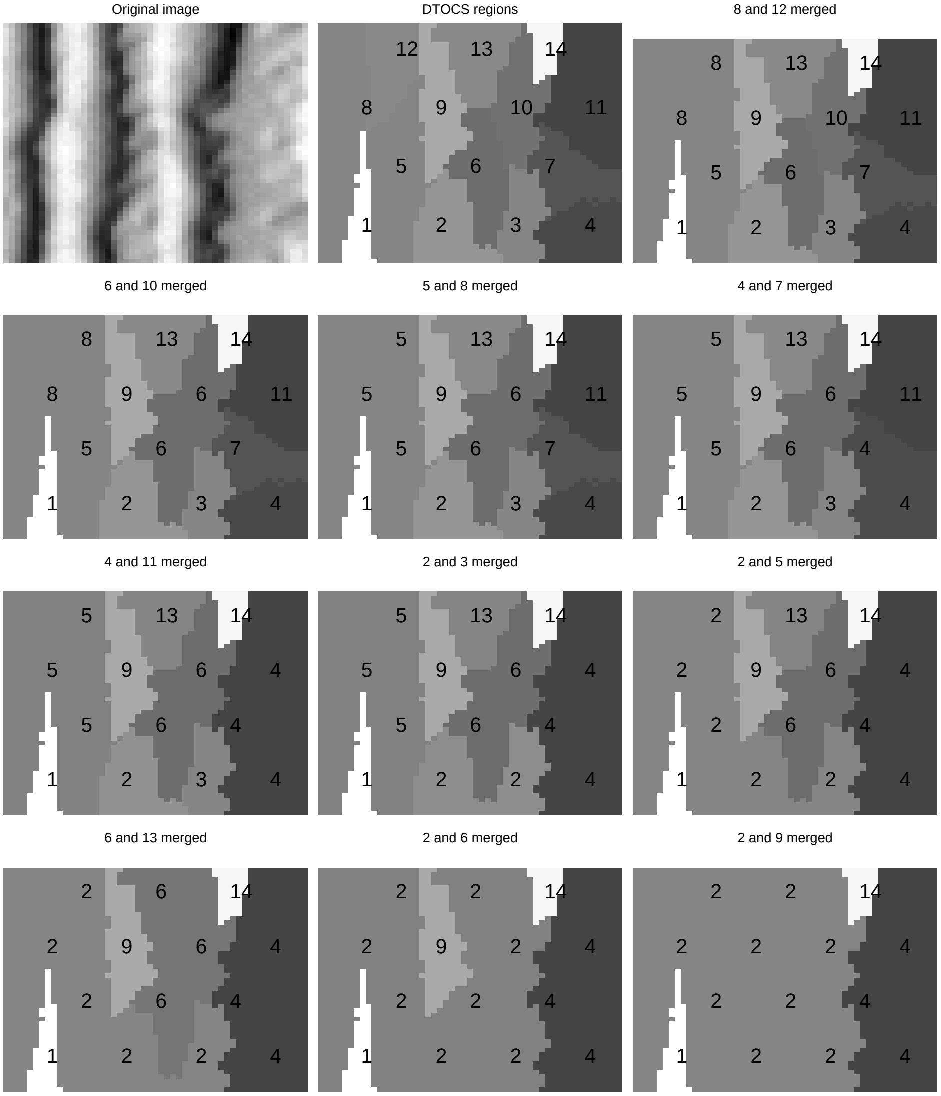


Figure 6: A RGM based merge procedure, step by step.

5 Result Validation

In many segmentation problems, the results are evaluated visually, comparing them to the original image. In roughness measurement, in many occasions it is impossible to compare the results and to decide which of the results are better or even if the result is good enough. Therefore some method is needed to evaluate the results. In this work statistical methods are used in validation of the results.

The Pearson coefficient of correlation, marked with p, is used to evaluate two pairs of values. For any random pair of values $-1 \leq p \leq 1$ is always true. Values near 1 mean strong positive correlation, values near -1 mean strong inverse correlation. The value of p is given by

$$p = \frac{\sum x \cdot y - n \cdot \overline{X} \cdot \overline{Y}}{\sqrt{(\sum x^2 - n \cdot \overline{X}^2) \cdot (\sum y^2 - n \cdot \overline{Y}^2)}} \tag{10}$$

where x and y are the values of the different samples, $\overline{X}$ and $\overline{Y}$ are the mean values of the samples, and n is the number of value pairs. Pearson's coefficient of correlation is used to indicate correlation between samples. As stated in [23], it is not able to show that no correlation exists. There can always be found some kind of correlation for dependent samples. Therefore the compared samples should be independent.

Plotting all value pairs to xy plane shows to human observer, if there is a trend with value pairs. If a value increase in the first value of value pair is followed by the increase in the second value as well, strong positive correlation has been found. If there is a strong correlation between the values in each pair, a regression line can be drawn on the xy plane to visualize the trend. The equation of the regression line is expressed in the form

$$y = m \cdot x + b. \tag{11}$$

For that equation, the slope is found by

$$m = \frac{\sum x \cdot y - n \cdot \overline{X} \cdot \overline{Y}}{\sum x^2 - n \cdot \overline{X}^2} \tag{12}$$

and the constant b by

$$b = \overline{Y} - m \cdot \overline{X}. \tag{13}$$

It is not sufficient to calculate the correlation between samples. It must be evaluated, if the correlation is significant. For this purpose, hypothesis testing is used, as described in [24]. The correlation increases, when the sample size increases and variation of sample pairs decreases. The first step is to make a null hypothesis: $H_0 : p = 0$, which means there is no correlation between samples. If the null hypothesis can be shown to be wrong, significant correlation has been proved between samples, and the alternate hypothesis $H_1 : p \neq 0$ is accepted. The null hypothesis can be proved to be wrong, if the evaluated value p is not between the significant confidence levels of H_0. Usually a $95\ \%$ confidence level is high enough.

The confidence interval calculation requires some additional operations compared to a normally distributed set of data. The calculation of confidence intervals for Pearson's correlation described in [25] is introduced in detail next. Before starting the calculation, Pearson's correlation p must be converted to normal distribution. It is done by Fisher's z' conversion, which is given by

$$z' = 0.5 \cdot [\ln(1 + p) - \ln(1 - p)]. \tag{14}$$

After the conversion, the hypothesis testing is done assuming the peak of the used t-distribution is in the converted correlation value. With t-distribution $n - 2$ degrees of freedom is used, where n is the number of data pairs. With sample sizes greater than 30, normal distribution is assumed. The calculation of confidence intervals is given by

$$z'_{\pm} = z' \pm z \cdot \sigma_{z'}, \tag{15}$$

where z tells how many standard deviations it is below the mean or above the mean. z is calculated by

$$z = \frac{x - \mu}{\sigma}, \tag{16}$$

where μ is the mean value of the used distribution, x is a data point above or below the center of the distribution and σ is the standard deviation of the used distribution. The standard deviation of the correlation data is not known, but it is approximated by

$$\sigma_{z'} = \frac{1}{\sqrt{N-3}}. \tag{17}$$

After the calculation of $z'_{\pm}$, inverse conversion must be done to have confidence interval in Pearson's scale. By solving p from the Equation (14) the inverse operation is done. The resulting equation is given by

$$p = \frac{e^{2z'}-1}{e^{2z'}-1}. \tag{18}$$

For example, the $95\ \%$ confidence level for correlation 0.4716 with 184 samples is calculated the following way:

1. Convert the correlation to z' (Equation (14)), $z' = 0.5121$.
2. Calculate z (Equation (16)). For example, for a normal distribution, z is 1.96 when covering $95\ \%$ of the distribution.
3. Calculate $\sigma_{z'}$ (Equation (17)), $\sigma_{z'} = 0.036$
4. Calculate confidence intervals z'_{+} and z'_{-} (Equation (15)), $z'_{-} = 0.3664$ and $z'_{+} = 0.6578$.
5. Convert the results z'_{-} and z'_{+} back to Pearson's scale (Equation (18)), which results to confidence interval, where lower end point is $p_{-} = 0.35$ and upper end point is $p_{+} = 0.58$.

As explained in [23], the results are evaluated for example by comparing two sets of data with each other. Again, Fisher's transform is needed (Equation (14)). For both of the compared correlations z' is calculated. After that value z is calculated according to equation

$$z = (z'_1 - z'_2) / \sqrt{\frac{1}{N_1-3} + \frac{1}{N_2-3}}. \tag{19}$$

The result gives probability to the question: "Does these two samples have the same correlation". And again, null hypothesis is rejected if the rejection probability exceeds the $95\ \%$ confidence level.

For example, if comparing correlation in the previous example to set of 760 pairs with correlation 0.4876 the hypothesis testing would look like the following:

1. Calculate z' to the other data set, the same way as in the first example.
   $z'_2 = 0.5329$.
2. Calculate z (Equation (19)), $z = 0.2512$.
3. Finally z is put to the normal distribution, which results 80% of the total area of the probability density function. It is the probability for the data sets to have the same correlation.

There are some restrictions for using correlation in the estimation of measure of similarity between two samples. In the split and merge approach the results can not be verified by using the Pearson's correlation coefficient. Comparison with S_a roughness would be misleading, since both split and merge algorithms (SPM, SDTOCS) already use the Gaussian filtered roughness map to make split decisions. The SPM algorithm uses it also for merge decisions. As stated in [23], samples (S_a roughness and split and merge results) have causal relationship: results are partly derived by certain operations from S_a roughness map. Therefore the results for SPM and SDTOCS are not validated by correlation coefficients.

6 Experiments

The results of the experiments that were carried out by using previously introduced algorithms are presented in this chapter. The purpose of these tests was to find out if S_a roughness correlates with the results of different algorithms. S_a roughness is a generally accepted method to measure surface roughness. High correlation with S_a roughness is considered to prove that used method is able to measure surface roughness.

All the necessary conditions of different experiments are introduced first. They are such as the amount of seed points used to calculate the DTOCS (grid size), grid type (how seed points are placed on the image). A detailed study of the values produced by the DTOCS method is needed to better understand the results. Another issue is to choose a type of a statistical method for comparison. There were a few statistical methods, introduced previously in this work, and they may correlate differently with DTOCS roughness.

After the introduction to the experiments there is an excursion to the normalization process of the DTOCS algorithm. Experiments are started by examining different S_a roughness values of the samples. The next step is to examine the DTOCS method, which is also an initialization stage for the segmentation methods. Experiments continue with segmentation methods. First is tested merge operations and then the split and merge approach. Finally, certain comparisons between different test sets are done.

6.1 Introduction

This section has the following structure. First is described, how the test data was measured. Material and test area sizes as well as different parameters that were used in the experiments are described. The manner of representation of the results is introduced as well.

6.1.1 Measuring devices

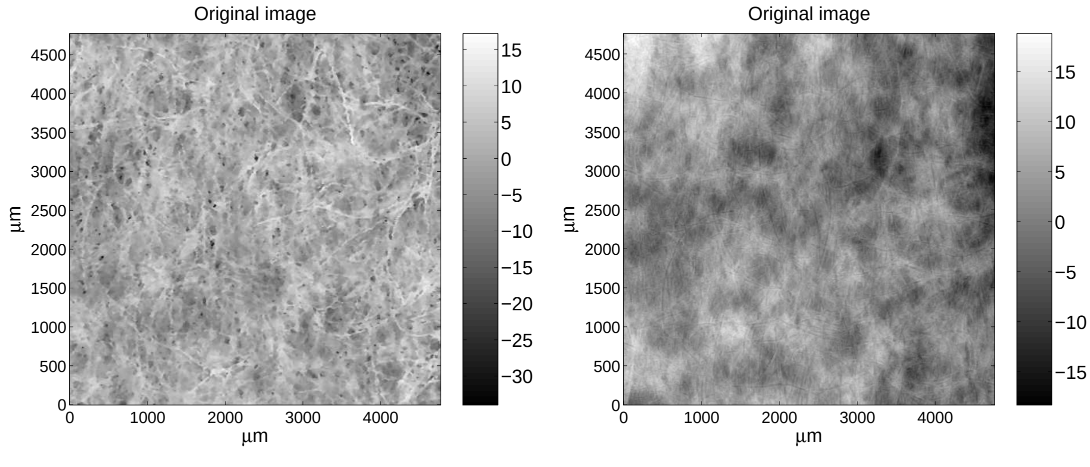
The profilometer used to measure the samples in this work was Rodenstock RM-600 3-D/C laser profilometer. The topography was measured by laser sensor LS-10. The wavelength of the laser beam was $790\ nm$. In the measurement method, a collimated beam of light, coupled to an optical microscope, is split and focused on the test surface. Height differences are seen as light and dark fringes on a video camera or in a detection system.

Unlike in a normal configuration, OptiTopo images presented in this work have been illuminated with four lights. The topology is calculated using these four images. The achieved resolution is 2048×2048 pixels for $15 \times 15\ mm^2$ area.

6.1.2 Test samples

Two sets of test data are used in all of the tests. In both sets, there are 8 images. The material for both of the sets was base cardboard. Exactly the same test areas are measured by using two different measuring devices. The first set is measured using the profilometer. The measurement area was $16 \times 16\ mm$. The distance between pixels was $10\ \mu m$ in x and y -directions. In other words, a sample image is a 3D matrix, which has a height value every $10\ \mu m$.

The second data set is measured using OptiTopo. The measurement area was $15 \times 15\ mm$. The distance between pixels was $7.32\ \mu m$ $(15000\ \mu m/2048)$ in x and y -directions. An example of both measured data sets is seen in Figure 7. In the figure the same sample is measured with OptiTopo and profilometer.



(a) Profilometer

(b) OptiTopo

Figure 7: Samples from the used data sets.

The sample images were not used as a whole in the experiments. A certain area was cut from the center of the used images. In Table 1 are shown different test area sizes for different tests presented. Gaussian filtering reduces the size of the area. Two test sets contain different resolution, therefore the amount of pixels is also different for the given test area. For more complex algorithms, smaller test areas are used to reduce the processing time.

Table 1: The sizes of the test areas.

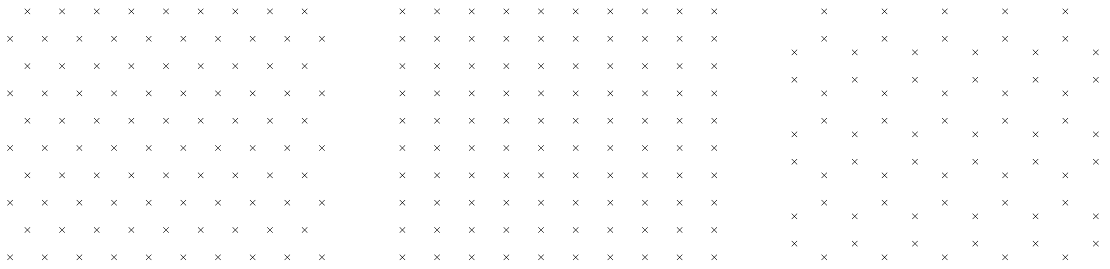
	$250\ \mu m$
Profilometer	
Tables 2 - 17	$4.76 \times 4.76\ mm^2$, $476 \times 476\ pixels$
Tables 18 - 19	$2.26 \times 2.26\ mm^2$, $226 \times 226\ pixels$
OptiTopo	
Tables 2 - 5	$4.75 \times 4.75\ mm^2$, $649 \times 649\ pixels$
Tables 18 - 19	$2.25 \times 2.25 mm^2$, $307 \times 307\ pixels$

6.1.3 Test parameters

The grid sizes vary between 5×5 to 41×41 seed point grids, which results to $100\ \mu m$ - $1904\ \mu m$ distance between the seed points. In the grid definition the equal distance between the seed points was preserved. Therefore, the actual amount of seed points varies due to rounding of the distance between two points. For example, a test image side is 476 pixels. When it is divided by 30 (30×30 grid), the result is 15.87. It is rounded to 16, but each row can have only 29 seed points, because $30 \cdot 30 = 480$.

In Figure 8 are presented three different test grids, which are called a net grid, a tile grid and a hexagon grid in this work. These grids show three different ways that were used to set the seed points for the DTOCS method on an image. If using a tile grid, regions will most likely be scattered in such way, that the grid is suitable for those methods that take adjacency into account, like the RGM algorithm. For example, if using a net grid, regions formed by the DTOCS method become regular in flat surfaces, because there are no obstacles causing irregularities. In that case the regions would look almost like squares. Those regions that are the closest to the corners of a region, are adjacent with it in many cases. But if those regions are regular having a form of a square or some other geometrical shape, sometimes some minor irregularities in the corner area could make them not to be adjacent. In that case, quite often it is up to one pixel that two regions are not adjacent.

In Figure 9 is presented an example that shows a corner area of four regions. Each square represents one pixel, and all pixels are numbered with a label of a region they belong to. If the regions are completely regular, the corner area looks like in image a). But if some minor irregularity in the original image causes labeling as in image b), where region 1 has one pixel more than the other regions, adjacency criterion is not met with all of the regions. In image b) only region one is adjacent with three other regions, while in image a) all the regions were adjacent to each other. With large regions, one pixel can have too much significance, if it is situated in the corner of other regions. That is most likely avoided by setting the seed points as a tile grid, where regularly formed regions have two adjacent regions above and below, and one on both sides. No border areas of one or two pixels are formed, if regions are close to regular.


(a) Tile grid

(b) Net grid

(c) Hexagon grid

Figure 8: Used grids.

1	1	2	2
1	1	2	2
3	3	4	4
3	3	4	4

(a) Completely regular regions

1	1	2	2
1	1	2	2
3	1	4	4
3	3	4	4

(b) Minor irregularity in the corner

Figure 9: A corner area of four regions that have a square form.

There is a different distance between the seed points in different grid types. In the hexagon grid there are areas, where the seed points are more densely placed than in the other areas. In the tile grid the distance between the seed points is a bit shorter than in the net grid, and on the sides of images there are larger areas.

The wavelength of the Gaussian filter was chosen to be $250\ \mu m$, since it is one of the recommended wavelengths. Measured material was rougher than the normal paper samples. Therefore slightly greater variations were preserved with the chosen wavelength in the S_a roughness map than would have been preserved with the shorter wavelength, such as $80\ \mu m$.

6.1.4 Presenting the correlation of the test data

Two ways to calculate the correlation were used in this work. The first way was to calculate the DTOCS roughness value and the S_a roughness value for the entire image. The second approach was to have the regional correlation. It means, that there are two values per region: the S_a roughness value and the DTOCS roughness value. The regions are defined by the results. For example, if the PNN based method produces 5 regions and the DTOCS roughness value for each of them, the S_a roughness value is calculated as the mean value from the S_a roughness map for the corresponding pixels.

Two types of tables are used to present the results. The first table type lists the correlation data for different tests. Besides the correlation values, there are presented the confidence intervals for $95\ \%$ confidence level. The amount of used seed points is given. The vertical and the horizontal distances between the seed points are also given, and the amount of sample pairs, if it is not easily derived from the results and from the sample size.

The second table type consists of one value per sample. It is a matrix that contains comparisons between all of the samples in the table. For each value pair, the hypothesis testing is done. A null hypothesis is an assumption that the value pair has the same correlation. As a result, the probability for the null hypothesis is achieved. For example, if comparing samples 1 and 2, row 1 is chosen, and from that row the first column (columns start always from sample 2) is chosen to find out the probability of the same correlation in samples 1 and 2. Therefore, in the experiments following expression is used: "Sample A correlates better than sample B", if the correlation coefficient of sample A is higher than of sample B, and the pairwise hypothesis testing shows that the probability of A and B to have the same correlation is below 5%. These tables are presented in Appendix I.

6.1.5 Test structure

Tests are divided into three phases. The first phase is the evaluation of the initial stage, which means the formation of regions by the DTOCS method. DTOCS regions were formed with different parameters, and the results were compared to each other. The way to compare the results was the pairwise hypothesis testing that was explained above.

In the second phase, different segmentation results were evaluated. It was done by changing exactly the same parameters as was changed in the first phase. If the results do not change similarly as in the first phase, it may give information about the quality of the algorithm. Change in the results is detected by comparing the correlation coefficients the same way as in phase one.

The last phase is to compare different segmentation results with each other and with initial regions. The goal is to find out, which of the methods gives the best results. An objective of this study is also to find out, how much the results change after the segmentation, and does initial regions and final segmented regions both correlate with S_a roughness.

The split and merge algorithms are not evaluated the same way as the other algorithms. As was explained earlier, those methods are not independent when compared to S_a roughness, because they use the S_a roughness map for the split decisions and the SPM uses it also for the merge decisions. These algorithms are judged visually from the processed images after the split operations and after the merge operations.

6.2 Emphasizing properties of the DTOCS algorithm

A small example of the DTOCS roughness map calculation is in Figure 10 and in Figure 11. In this example, in both tables there are four images shown: an original image, projection distance image, distance image and normalized image. One seed point is situated in the lower right corner and it is marked with P in the images. The calculation starts from pixel P. The resolution of the example is 10×10 (the distance between neighboring pixels is 10). The same height pattern is set right next to the seed value in the upper table and to the left corner in the lower table. The resulting normalized image shows how much higher values are achieved in case of seed value being close to the pattern and in case of seed value being more distant. The mean value for the upper normalized image is 1.07 and for the lower image 1.01. Thus, the location of seed points affects the final result. It is obvious that the method emphasizes the values that are close to seed points. Therefore even after the normalization, pixel values are not comparable with each other.

a) Original image					b) Distance image				
0	0	0	0	0	42	42	42	42	42
0	0	0	0	0	42	32	32	32	32
0	0	0	0	0	42	32	22	22	22
0	0	0	1	1	42	32	22	11	11
0	0	0	1	P	42	32	22	11	P

c) Projection distance image					d) Normalized image				
40	40	40	40	40	1.05	1.05	1.05	1.05	1.05
40	30	30	30	30	1.05	1.07	1.07	1.07	1.07
40	30	20	20	20	1.05	1.07	1.1	1.1	1.1
40	30	20	10	10	1.05	1.07	1.1	1.1	1.1
40	30	20	10	P	1.05	1.07	1.1	1.1	P

Figure 10: An example of DTOCS calculation

a) Original image					b) Distance image				
0	0	0	0	0	40	40	40	40	40
0	0	0	0	0	40	30	30	30	30
0	0	0	0	0	40	30	20	20	20
1	1	0	0	0	41	31	20	10	10
0	1	0	0	P	42	31	20	10	P

c) Projection distance image					d) Normalized image				
40	40	40	40	40	1.00	1.00	1.00	1.00	1.00
40	30	30	30	30	1.00	1.00	1.00	1.00	1.00
40	30	20	20	20	1.00	1.00	1.00	1.00	1.00
40	30	20	10	10	1.03	1.03	1.00	1.00	1.00
40	30	20	10	P	1.05	1.03	1.00	1.00	P

Figure 11: An example of the DTOCS calculation.

The DTOCS method overestimates the diagonal values in the normalized image. This results from the calculation method of the distances. The method calculates the diagonal distances exactly the same way as the horizontal and the vertical components. However, in the normalization process there is the same denominator, which would be greater if using Euclidean distances. If there is a continuing slope in the image, a height value in a pixel is a diagonal distance away. More distance in the slope results to higher or lower value, and as it finally gets divided by lower than the real value, the final value gets overestimated. This results to the higher normalization values. An example of overestimation is in Figure 12. A normalized image (Distance image divided by 2D distances) is presented in the figure. Gray levels represent the pixel values, high values are light gray levels and dark values are low values. In the figure, the seed points are seen in the middle of "the white X letters". In the normalization process, the highest values are always close to the seed points, and they are in 45 degree angle of the horizontal level to four directions.

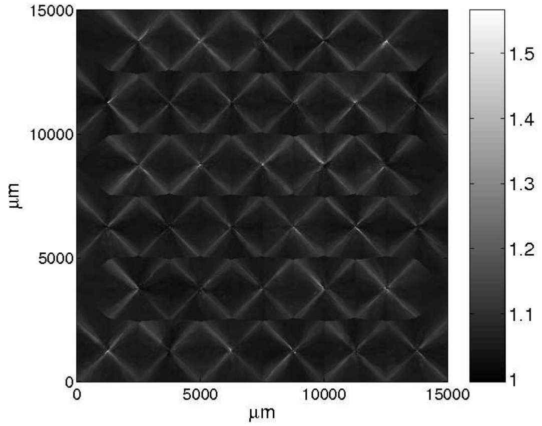


Figure 12: Values of the normalized image

6.3 Comparison of statistical methods

Three statistical methods to calculate the roughness surface were compared on regional basis to DTOCS regions. The results are in Table 2. They were moving S_a roughness after the Gaussian filtering, only an image after the filtering, and a moving S_a roughness map without Gaussian filtering. Moving S_a roughness was calculated using 5×5 kernel.

Gaussian filtering turned out to be crucial for the correlation between DTOCS and S_a roughness. Without the removal of waviness surface there was no correlation detected (the correlation coefficients were low for both data sets) between DTOCS roughness and moving S_a roughness. In the pairwise hypothesis testing probability for the Gaussian filtered (both a moving S_a roughness map and only an original image after the Gaussian filtering) and unfiltered data to have the same correlation was clearly below $5\ \%$. It proves that DTOCS roughness and statistically processed image correlates only after the Gaussian filtering. Therefore it can be concluded that larger scale roughness is not detected with the DTOCS. There was no difference between two different results after the filtering. This experiment directs to use moving S_a roughness after the Gaussian filtering in the following experiments.

Table 2: The correlation coefficients between DTOCS and regional S_a roughness in 30×30 tile grid for the cardboard data sets.

	Grid,	$Corr.$	$Conf.$ $interval,$ $lower$	$Conf.$ $interval,$ $upper$	$Seed$ $points$	$Distance$ $between$ $seedpoints,$	$Total$ $sample$ $size$
Profilometer, wavelength 250 μm							
1	Moving S_a	0.6876	0.6751	0.6997	885	$160 - 320\mu m$	7080
2	Gauss. only	0.6795	0.6667	0.6918	885	$160 - 320\mu m$	7080
3	Gauss. off	0.1890	0.1659	0.2119	841	$170 - 340\mu m$	6728
OptiTopo, wavelength 250 μm							
4	Moving S_a	0.5155	0.4977	0.5328	841	$161 - 322\mu m$	6728
5	Gauss. only	0.5060	0.4880	0.5236	841	$161 - 322\mu m$	6728
6	Gauss. off	0.0095	-0.0138	0.0328	885	$168 - 336\mu m$	7080

6.4 Comparison of DTOCS roughness and S_a roughness

In order to be able to evaluate different segmentation methods, they must be compared to the initial stage of the segmentation procedure. The initial stage means that there are regions formed by the DTOCS method, and those regions are not yet further processed. As the result of the DTOCS method may be affected by different starting parameters, tests are applied by changing them. The changed parameters are grid size and grid type.

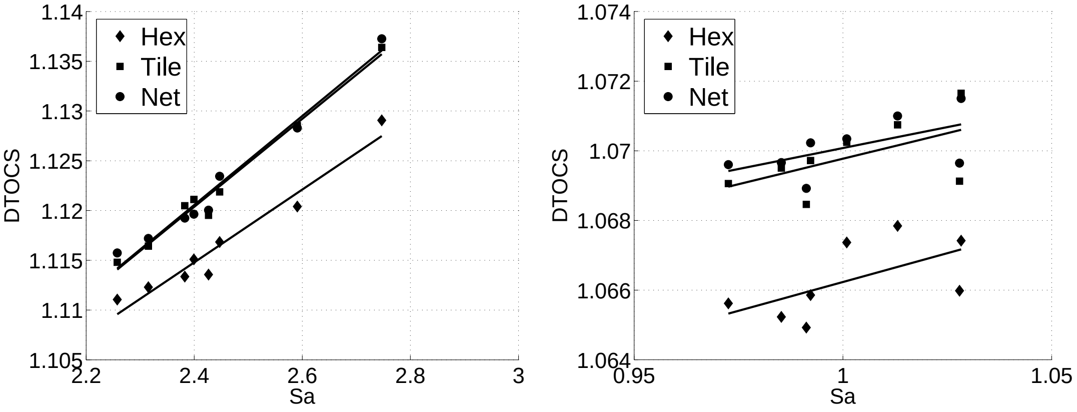
When calculating one roughness value for an image, the result does not give sufficient information on regions within an image. Therefore the same experiments applied for the entire images are repeated by using the regional approach and the results are presented later in this section. In the regional approach each region forms a value pair: DTOCS roughness and moving S_a roughness is calculated for it, the same way as was done earlier in the statistical methods testing. The regional approach results to the high amount of value pairs that can be used in the hypothesis testing.

6.4.1 Grid type comparison for a whole image

In this experiment, moving S_a roughness is calculated for the entire image. It is compared with the DTOCS roughness value, calculated for the entire image as well. The experiment resulted to 8 value pairs. Due to the small sample size, the hypothesis testing does not produce reliable results and it is not used for the comparison of the correlation coefficients.

Instead, the value pairs are placed into xy plane, where the DTOCS roughness values are in y axis and the moving S_a roughness values in x axis. A regression line is drawn on the plane on the basis of the value pairs.

In Figure 13 are drawn values resulting from the grid type comparison. In each of the images there are 3 grid types represented. In image a) are presented the results for profilometer and in image b) for OptiTopo. In both images the drawn regression lines are almost identical for different grid types. In the hexagon grid DTOCS values differ slightly from the other grid types, but its regression line has almost the same slope as the others. That is most likely caused by the smaller amount of seed points. When there are less seed points, there are less high values, which cause also the mean values to be lower.



(a) Profilometer.

(b) OptiTopo.

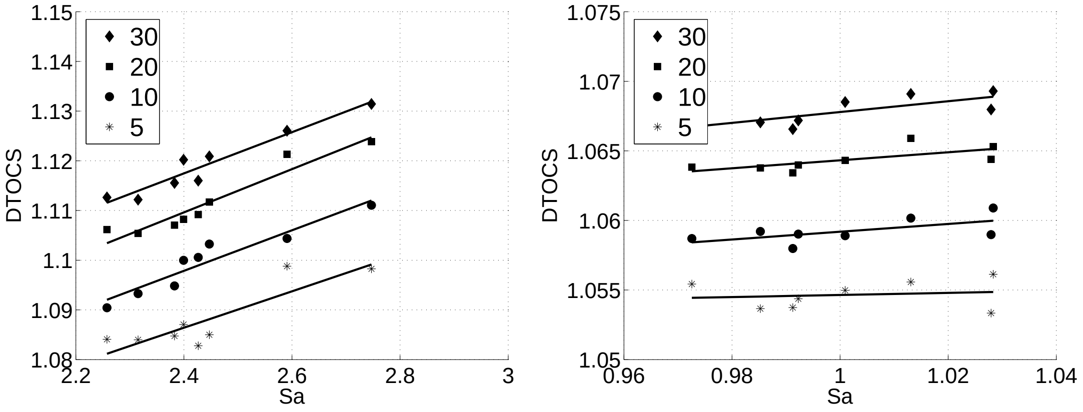
Figure 13: The comparison of different grid types (grid size 41×41) between DTOCS roughness and S_a roughness for the cardboard data sets.

According to these tests it looks quite obvious, that different grid types do not correlate differently with moving S_a roughness. For tile and net grid, DTOCS roughness value range is almost identical. According to these tests, the OptiTopo samples can not be shown to correlate with Sa roughness, since lower bound of the confidence intervals in all of the samples is below 0 (Table 14, Appendix II).

6.4.2 Grid size comparison for a whole image

In Figure 14 are shown scatter plots for different grid sizes. Also in this experiment, the slopes are near the same for all grid sizes. In both of the data sets higher grid size results to higher DTOCS roughness value. Increase in the amount of seed points increase also the correlation coefficient. However, the test sample size is too small to be able to see the significance of the difference. Excluding the results for 30×30 grid size, correlation of

OptiTopo data was not shown to exist. (Table 15, Appendix II).



(a) Profilometer.

(b) OptiTopo.

Figure 14: The comparison of the different grid sizes (net grid) between DTOCS roughness and S_a roughness for the cardboard data sets.

6.4.3 Regional grid type comparison

In Table 3 are listed the correlation coefficients from the grid type comparisons. In addition to previously introduced grid types, a tile grid that has been shifted 4 pixels right and 4 pixels down along x and y axis was used. There is no significant difference between the values, but with the profilometer data the hypothesis testing shows that the hexagon grid has weaker correlation than the other samples (probability of different correlation is over $95\ \%$). As was in the previous experiments, there are less regions, when the hexagon grid is used. That may also cause the lower correlation. The same difference is not found in the OptiTopo results. Therefore, according to this experiment, it may be suggested that the hexagon grid has slightly weaker correlation. Unlike in the previous experiments, the DTOCS results from OptiTopo data was proved to correlate with S_a roughness, although the correlation is not very strong.

6.4.4 Regional grid size comparison

In Table 4 the correlation coefficients from the grid size comparison are shown. The pairwise hypothesis testing for the correlation coefficients of the profilometer data shows that for 40×40 grid the correlation is higher than for all the other samples. For the other grid sizes, the same trend is seen when comparing it to two steps (30×30 to 10×10 etc.) smaller grid. The results of OptiTopo does not support any conclusions about the effect of

Table 3: The correlation coefficients between DTOCS grid type comparison and regional moving S_a roughness in 30×30 grid for the cardboard data sets.

	Grid,	*Corr.*	*Conf.*	*Conf.*	*Seed*	*Distance*	*Total*
			interval,	*interval,*	*points*	*between*	*sample*
			lower	*upper*		*seedpoints,*	*size*
Profilometer, wavelength 250 μm							
1	Net	0.6894	0.6770	0.7013	900	$160\mu m$	7200
2	Tile	0.6876	0.6751	0.6997	885	$160 - 320\mu m$	7080
3	Hex	0.6651	0.6498	0.6799	660	$160 - 320\mu m$	5280
4	Tile, shift	0.6949	0.6823	0.7070	841	$160 - 320\mu m$	6728
OptiTopo, wavelength 250 μm							
5	Net	0.5139	0.4961	0.5313	841	$161\mu m$	6728
6	Tile	0.5155	0.4977	0.5328	841	$161 - 322\mu m$	6728
7	Hex	0.5127	0.4926	0.5322	663	$161 - 322\mu m$	5304
8	Tile,shift	0.5091	0.4912	0.5266	841	$161 - 322\mu m$	6728

the grid size or the grid type changes to the results. Generally, the correlation coefficients for OptiTopo data are not very high in this experiment.

Table 4: The correlation coefficients between DTOCS grid size comparison and regional moving S_a roughness in tile grid for the cardboard data sets.

		Corr.	*Conf.*	*Conf.*	*Seed*	*Distance*	*Total*
			interval,	*interval,*	*points*	*between*	*sample*
			lower	*upper*		*seedpoints,*	*size*
Profilometer, wavelength 250 μm							
1	40x40	0.7105	0.7018	0.7190	1580	$120 - 240\mu m$	12640
2	30x30	0.6876	0.6751	0.6997	885	$160 - 320\mu m$	7080
3	20x20	0.6662	0.6462	0.6852	390	$240 - 480\mu m$	3120
4	10x10	0.6414	0.5975	0.6814	95	$480 - 960\mu m$	760
5	5x5	0.5418	0.4358	0.6330	25	$950 - 1900\mu m$	200
OptiTopo, wavelength 250 μm							
6	40x40	0.4971	0.4840	0.5099	1620	$117 - 234\mu m$	12960
7	30x30	0.5155	0.4977	0.5328	841	$161 - 322\mu m$	6728
8	20x20	0.5047	0.4785	0.5301	400	$234 - 468\mu m$	3200
9	10x10	0.4876	0.4315	0.5400	95	$476 - 952\mu m$	760
10	5x5	0.4716	0.3508	0.5769	23	$952 - 1904\mu m$	184

6.5 PNN based method

In this section, the roughness values are calculated for the regions resulting from the PNN based method. The first task was to evaluate the segmentation results with a different amount of final regions. Identical experiments with the previous section were applied to the samples in order to compare the segmented images with the images in the initial stage (regions that are formed by DTOCS algorithm). If the grid size changes or the grid type changes do not produce similar results as was in the initial stage, it means that PNN based regions either improve or weaken the result.

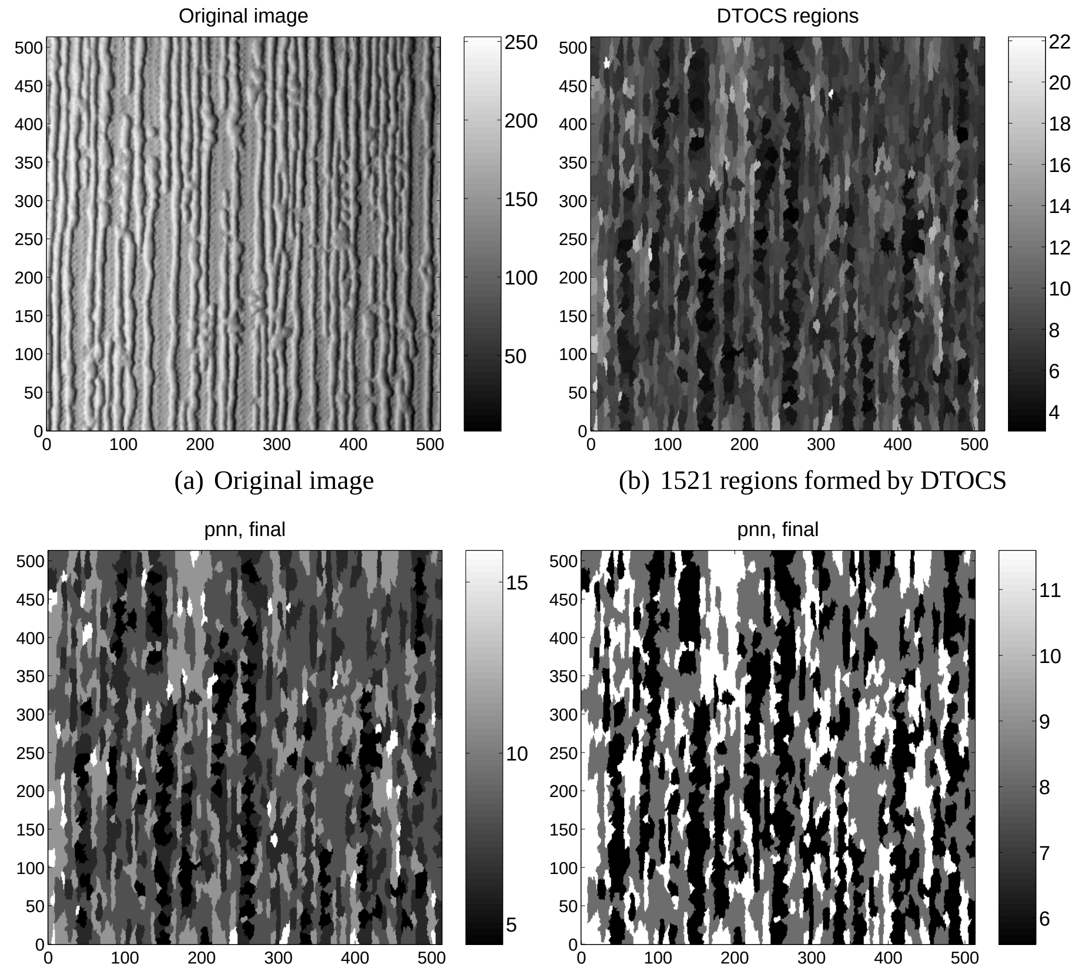
6.5.1 Different amount of final regions

The PNN based algorithm was tested by using example textures as well as using the test data sets. The first experiment was done for different amount of resulting regions. The method was applied to the example texture to visualize, how the method works in combination with the DTOCS. In Figure 15 in image a) the example texture has been divided into the regions according to their roughness values, which in this example means mostly vertical stripes, which are seen in the figure. The example texture consists of 255 gray levels, which are interpreted as height values by the DTOCS method. Initialization is done setting 40×40 seed points to the original image as tile grid. The size of the original image was 512×512 pixels. The first step is to form DTOCS regions, which is visualized in image b). The division into 5 resulting regions is shown in image c) and the division into 3 resulting regions in image d). It is quite clear in images c) and d), how more dense areas with vertical lines belong to the white area and the areas with less variation belong to the black area. In the resulting images rough areas are marked with lighter gray level in image c) and white color in image d).

PNN based method was applied to the cardboard data sets as well, and 3, 5 and 10 resulting regions were compared. The experiment did not produce significantly different correlation values. The detailed results are in Table 16 in Appendix II.

6.5.2 Different grid types

Figure 16 shows visually the effect of different grids used in the selection of DTOCS seed points and how it affects on the final result of the PNN based merge operation. Three grid



(a) Original image

(b) 1521 regions formed by DTOCS

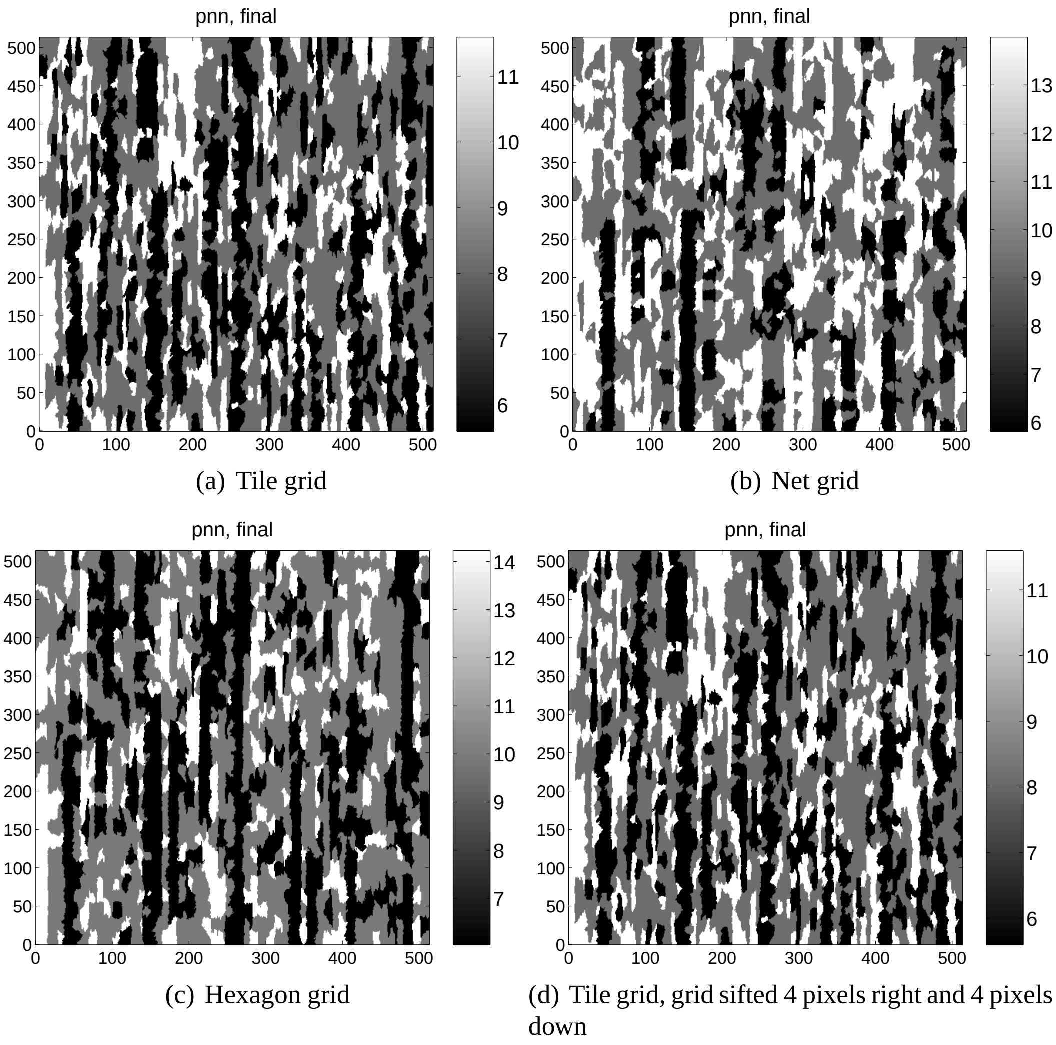
(c) 5 resulting regions by PNN

(d) 3 resulting regions by PNN

Figure 15: The PNN based method for 3 and 5 resulting regions.

types and the shifted tile grid can be visually compared by looking at these images. In the figure it is notable that the light and dark areas are not similar. However, there are still connection with the flat areas and light gray levels in all of the images. It must be pointed out that the value range is not the same in these images. Therefore the dark areas have different DTOCS roughness values in different images. Since the final values are formed by the quantization process, they may produce not easily comparable results. If the final roughness value of the roughest area in one of the images is not the same as in the other image, it naturally means, that those regions are not representing exactly the same feature.

Although different grid types do not produce identical looking results, they do not differ significantly according to the test sets. It is notable, that there is no difference in the correlation with the hexagon grid and the other grid types, while there was a difference in the initial regions. For both profilometer and OptiTopo data the correlation coefficients were significantly high. The values can be seen in Table 17 in Appendix II.



(a) Tile grid

(b) Net grid

(c) Hexagon grid

(d) Tile grid, grid sifted 4 pixels right and 4 pixels down

Figure 16: The PNN based method with different grid types, grid size 40×40.

6.5.3 Different grid sizes

The effect of the grid size change did not look very significant for the results of the profilometer data. If comparing 40×40 grid size to 20×20 grid size in Table 5, the correlation coefficient is lower for the latter in the profilometer results. As can be seen in the table, the correlation coefficients of OptiTopo data do not differ significantly.

6.6 RGM

The method to evaluate the segmentation results of the RGM is identical to the previous section. Grid types and grid sizes are changed, and then the results are compared. Two thresholds are applied to some of the grid sizes to see the effect of the threshold change as

Table 5: The correlation coefficients between different grid types in PNN (3 resulting regions) and regional moving S_a roughness in tile grid for the cardboard data sets.

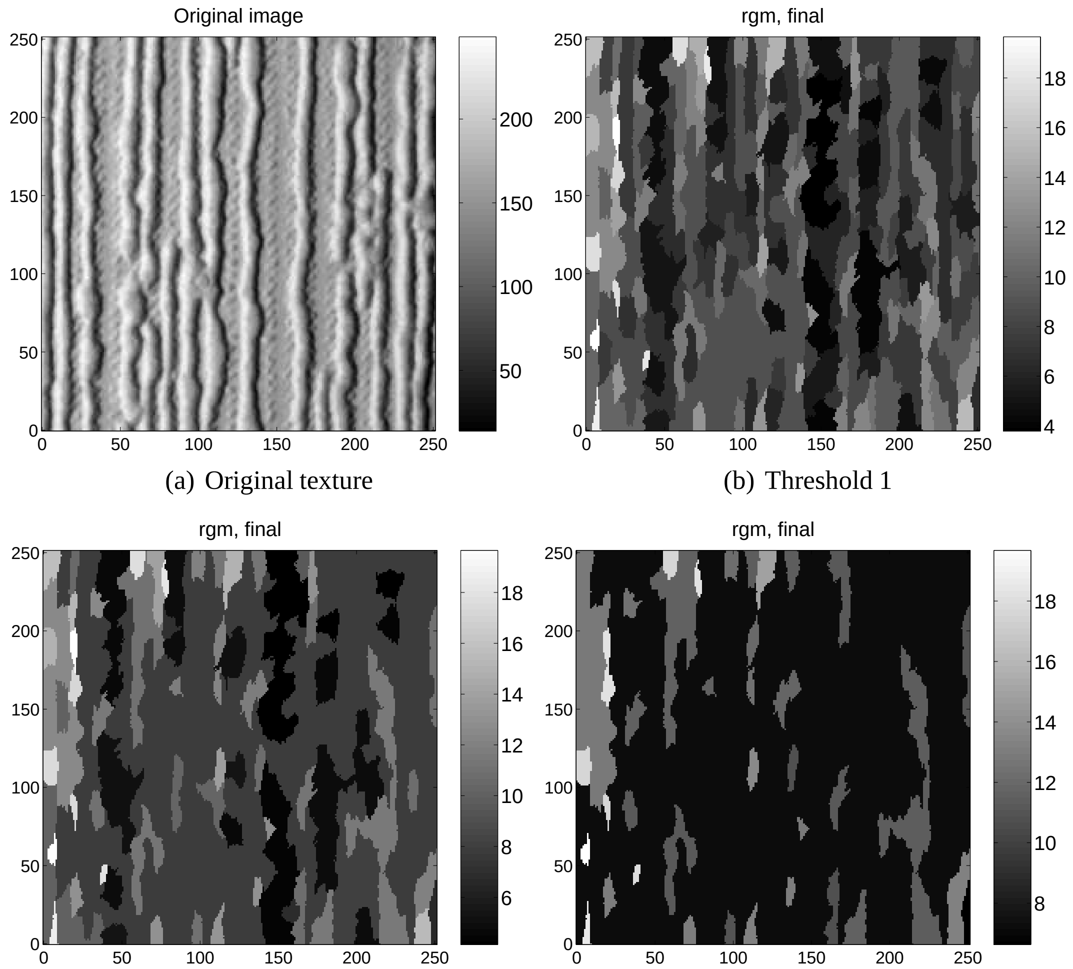
	Grid, 250 μm	$Corr.$	$Conf.$ $interval,$ $lower$	$Conf.$ $interval,$ $upper$	$Seed$ $points$	$Distance$ $between$ $seedpoints,$
Profilometer, wavelength 250 μm						
1	40×40	0.9710	0.9331	0.9876	1580	$120 - 240\mu m$
2	30×30	0.9392	0.8626	0.9737	885	$160 - 320\mu m$
3	20×20	0.8853	0.7497	0.9496	390	$120 - 240\mu m$
OptiTopo, wavelength 250 μm						
4	40×40	0.9789	0.9510	0.9910	1620	$117 - 234\mu m$
5	30×30	0.9731	0.9379	0.9885	841	$161 - 322\mu m$
6	20×20	0.9536	0.8942	0.9800	400	$234 - 468\mu m$

well. The size of the test area is smaller than in the previous experiments, but the distance between the seed points is preserved.

6.6.1 Different thresholds

In Figure 17 is shown an example texture that was split into regions by the DTOCS method, and the formed regions were further merged using the RGM algorithm. The results are shown with different thresholds. The threshold value is the mean value difference between a pair of regions. The texture has 255 height values. In the figure the spatial criterion is obviously visible: there are no scattered regions, because each pair of the merged regions must be connected. Therefore the regions remain connected throughout the merging process.

When comparing the results to the original image, it is easy to notice that increasing the threshold results to larger areas. Unfortunately, the larger the areas are, the rougher areas are merged into them, and the distortion is increased within the region. In image d) there is no very clear connection between the dark areas and the rough areas in the original image. Unlike image d), image b) is not so much distorted and it is easier to find the connection between the original image and the segmentation result in image b).



(a) Original texture

(b) Threshold 1

(c) Threshold 2

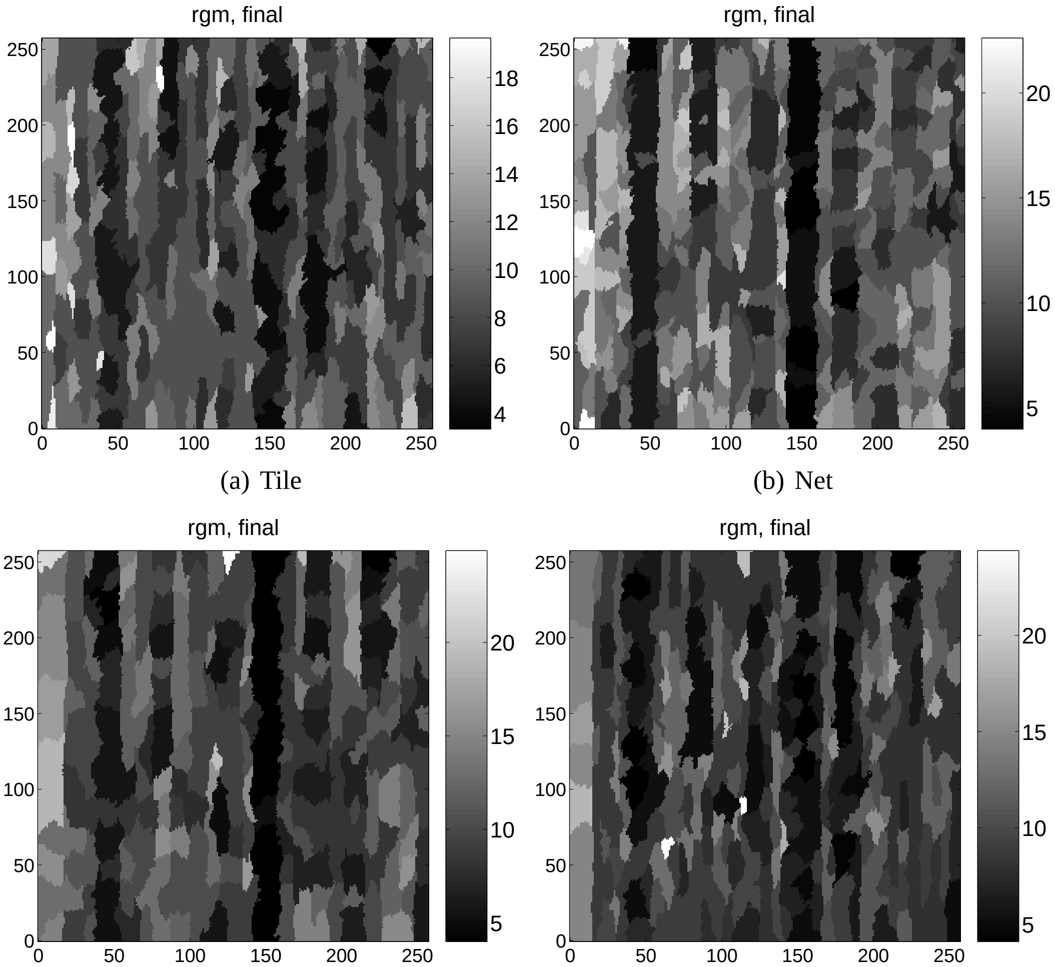
(d) Threshold 3.5

Figure 17: The RGM algorithm with different thresholds, grid size 20×20, image size 250×250 pixels.

6.6.2 Different grid types

In Figure 18 grid types are changed. The RGM algorithm is able to detect only some of the vertical plane areas (regions with dark gray levels). The rough areas (light gray levels) are quite significantly affected by the grid type change: they are scattered differently in every image. According to the texture, the final regions are strongly affected by the grid type change.

Cardboard samples were compared with these grid types as well. Generally, the correlation coefficients were low, especially for the profilometer images. The grid type changes did not affect the results significantly. The detailed results are in Table 18 in Appendix II.



(a) Tile

(b) Net

(c) Hexagon

(d) Tile, shift

Figure 18: The RGM algorithm with different grid types, threshold 1, grid size 20×20, image size 256×256 pixels.

6.6.3 Different grid sizes

The experiment procedure was more complicated than before. Finding the suitable thresholds required manual work, and even in the final results, thresholds were not necessarily suitable for all of the images. The same threshold produced different amounts of regions. To have the best possible results, the threshold should have been set for each image separately. The threshold had to be increased for a smaller amount of regions. Otherwise not enough merged regions would have resulted from the tests.

For the profilometer data, grid sizes 20×20 and 15×15 produced better results with lower threshold (more resulting regions). The best results were achieved with 15×15 and 10×10 grid, which both had one result that showed better correlation than 20×20 and 5×5 grids. The experiments with the OptiTopo data produced the highest correlation coefficients for

10×10 grid, where both of the results had higher correlation than 20×20 grid, and one of them had higher correlation than 15×15 grid. The results of the experiments are listed in Table 19 in Appendix II.

6.7 Comparison of different samples

In this section the difference between the used algorithms is visualized by presenting one cut of the image from both data sets. All the algorithms are applied to both images. The algorithms, excluding the SDTOCS and the SPM, are also compared by using the pairwise hypothesis testing between different results of the previous experiments. This testing is done in order to compare different methods with each other. These tests may suggest which of the methods do not distort the result too much. If the correlation is significantly lower after the segmentation process than prior to it, the algorithm may not provide enough information from the original image.

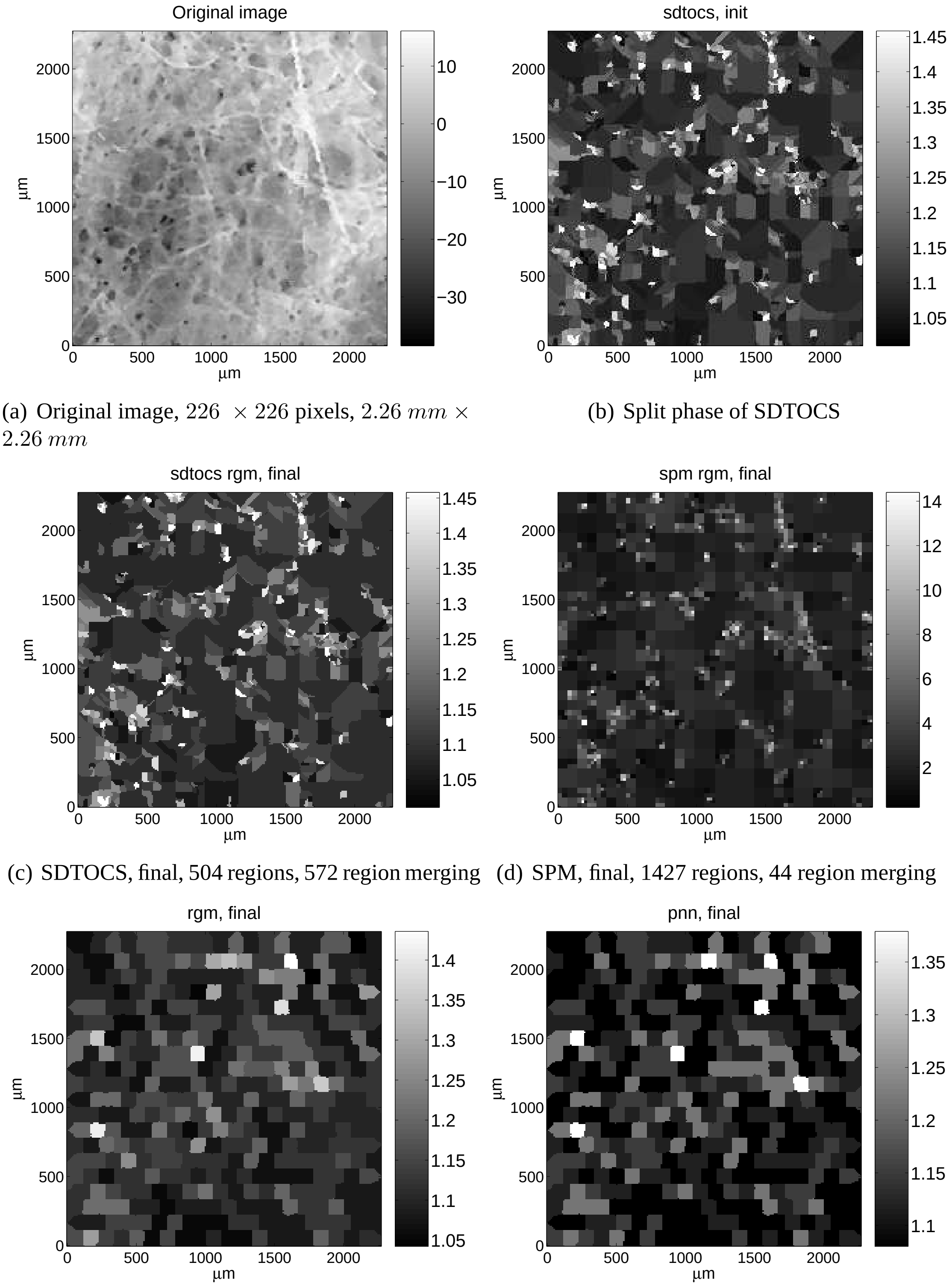
6.7.1 Split and Merge

In Figure 19 is presented one profilometer image and in Figure 20 one OptiTopo image. They had the 5×5 tile grid that was further split by using the SDTOCS and SPM algorithms, and the 10×10 tile grid that was segmented by using the PNN and RGM algorithms. The results can be estimated by comparing how well the areas in the original picture with a lot of variation correspond with the areas presented with light gray levels or white color, and how well the areas with less variation correspond with the darker areas in the segmented images. The SDTOCS algorithm itself should find the suitable amount of seed points, and the amount of initial regions defines the maximum size of a region. The SPM algorithm starts splitting from 4 equal sized squares.

In Figure 19 single lines in image a), which are base cardboard fibers, are lost inside regions in image c). For a region to be further split there needs to be several fibers in the small area. Significant value variations are isolated by splitting the areas containing high values, which can be seen as clusters of small white areas in the image. In Figure 20 in image a) fibers can not be see so clearly. Therefore the gray level variations must be observed. Areas with a lot of variation in image a) are seen as clusters of small regions in image c).

The algorithms can be compared through an example. In Figure 19 in image a) there is a black circular spot in the center ($1250\ \mu m$ right and $1300\ \mu m$ up from the lower left corner of image a)). The dark gray levels represent the low values and the light gray levels the high values in image a). In the image, the black round area is surrounded with white color. In image c) its border regions are white, indicating the area with rough values. The same circular area is easily detected also in image d). Both RGM and PNN algorithms found the black spot as well, but they did not consider it as the roughest area in the image. When comparing the images, it is seen that not all clusters of small regions in the SDTOCS image and in the SPM image match with the light gray levels in the RGM image and in the PNN image.

The main difference between the SDTOCS and SPM algorithms seems to be in large areas, which are more dependent on the merge phase, since there are most likely not so much difference between the mean values of the large areas than there is with the small areas. Unlike large regions, clusters of small regions seem to be almost identical for both of the algorithms.



(a) Original image, $226\ \times 226$ pixels, $2.26\,mm \times 2.26\,mm$

(b) Split phase of SDTOCS

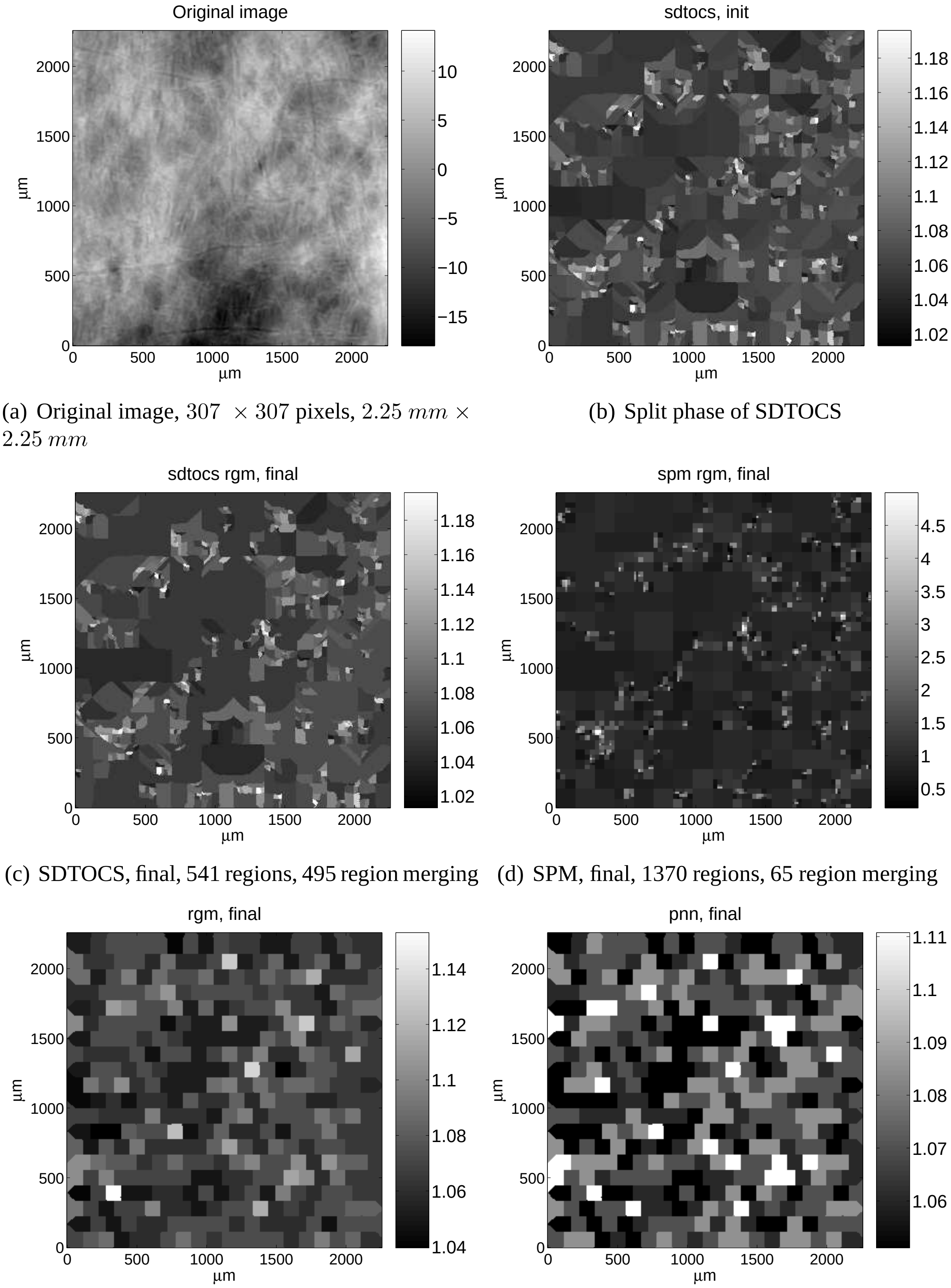
(c) SDTOCS, final, 504 regions, 572 region merging

(d) SPM, final, 1427 regions, 44 region merging

(e) RGM, 150 regions

(f) PNN, 5 regions

Figure 19: The profilometer data set.



(a) Original image, $307 \ \times 307$ pixels, $2.25\,mm \times 2.25\,mm$

(b) Split phase of SDTOCS

(c) SDTOCS, final, 541 regions, 495 region merging

(d) SPM, final, 1370 regions, 65 region merging

(e) RGM, 153 regions

(f) PNN, 5 regions

Figure 20: The OptiTopo data set.

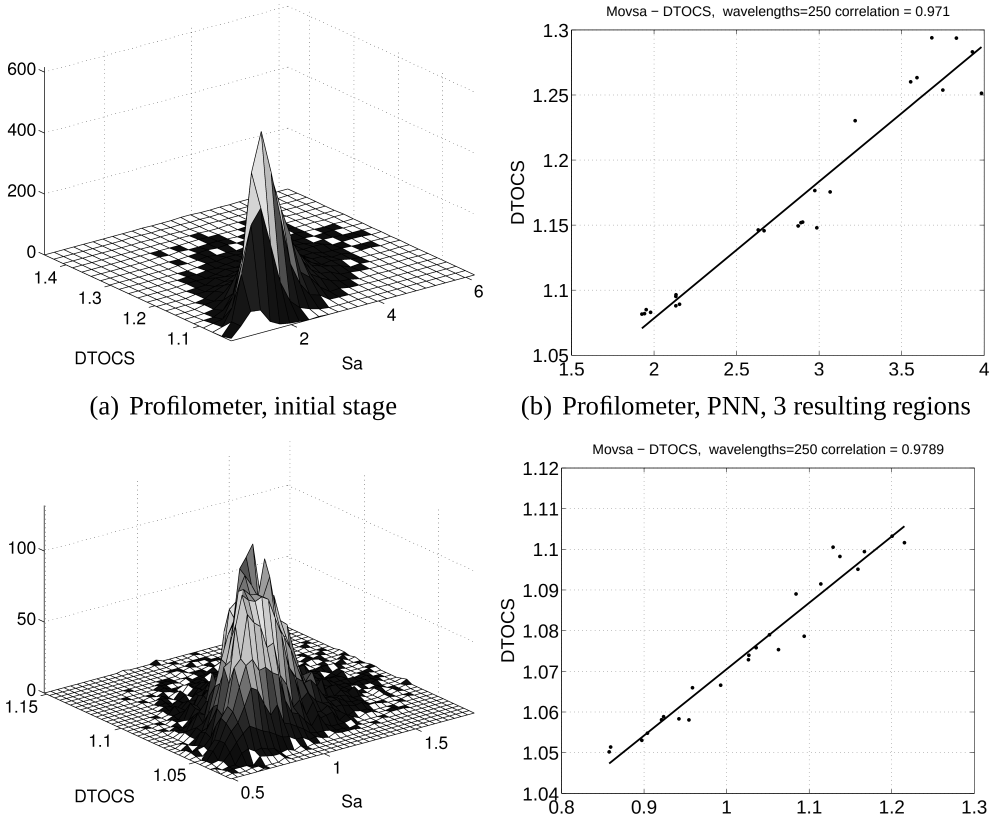
6.7.2 Comparison of correlations

In Table 13 in Appendix I are presented comparisons between different experiments that were presented earlier. From each table certain data sets have been chosen to be compared. All the significant differences are pointed out.

In the profilometer data it is shown that there is a difference between the PNN based method, the RGM and original regions. The correlation is higher for the PNN than for the original regions and the same or lower for the RGM. The PNN has significantly higher correlation than the RGM. Equivalent correlations for 30×30 grid, grid type tile are 0.94 for the PNN (3 resulting regions), 0.69 for initial regions and 0.55 ($20 - 41$regions) and 0.70 ($60 - 94$ regions) for the RGM.

For the OptiTopo data the results are almost equal. The resulting regions formed by the PNN have significantly higher correlation than the original regions. It is even higher than in the profilometer images for 5 and 10 resulting regions. The RGM method has weaker correlation values than the PNN and slightly better correlation than the initial regions. The equivalent correlations for the 30×30 tile grid are 0.97 for the PNN (3 resulting regions), 0.52 for initial regions, and 0.57 ($100 - 119$ regions) and 0.63 ($43 - 58$ regions) for the RGM. The regional correlation coefficients for the segmented OptiTopo images were higher or on the same level than for the segmented profilometer images. The correlation in the initial DTOCS images was weaker for the OptiTopo images than for the profilometer images.

In Figure 21 on the left are shown histogram plots of DTOCS regions, formed by using the 40×40 tile grid. The DTOCS roughness values are on one axis and the moving Sa roughness values on the other. All values are set into the 50×50 2D histogram, and they are presented in the 3D image. In other words, x and y axes are divided into 50 categories, and each value falling in to a certain category increases the histogram value by one. It is seen as increase in z axis, which represents the amount of values in one category. The most distant single high values from both axes are removed to show the dense areas clearly. On the right side there is a 2D representation of scatter plots and a regression line, having the same x and y axis as a 3D representations on the left side. Images b) and d) show values after the PNN based algorithm has merged original regions into 3 resulting regions. The corresponding correlation coefficients are 0.97 for PNN and 0.71 for the initial regions for the profilometer data set and 0.98 and 0.50 for the initial regions for the OptiTopo data set. These two images show, how densely or widely values are scattered along the plane. The values follow surprisingly accurately the regression line in images on the right. The hypothesis testing prove the difference that is seen between the left side images and the right side images. The probability for them to have the same correlation is less than 0.1% for both OptiTopo and profilometer images.



(a) Profilometer, initial stage

(b) Profilometer, PNN, 3 resulting regions

(c) OptiTopo, initial stage

(d) OptiTopo, PNN, 3 resulting regions

Figure 21: The comparison of the value range before and after the PNN based segmentation.Regional DTOCS roughness is presented in y axis and regional moving S_a roughness in x axis.

7 Discussion

In the experiments it was shown by detailed examples, how different placing of seed points results to different DTOCS roughness values. However, the comparison of different grid types did not produce significantly different results in the initial stage. If there are enough equally distributed seed points, they represent the image surface well. In rough surfaces there is a higher probability to place a seed point onto the extremely high or low value and emphasize it than in flat surfaces. Increasing the amount of seed points seemed to improve the results. When there are more seed points, normalized values are higher, but the difference between the highest and the lowest value decreases, and the neighborhoods of seed points are less emphasized.

In the RGM method, there was no difference in correlation after the grid changes. However, in the example texture the segmentation results were significantly different. The grid type change affects to the neighborhood list, and different regions are merged. Regions to be merged are chosen relatively well, and although differently looking, the results are not weaker in any of the grid types.

The threshold selection determines the amount of resulting regions in the RGM method. In order to achieve acceptable results, there should be enough regions to avoid the loss of important features. However, if there are too many regions, an image can not be considered fully segmented, and most probably there are region pairs left that could be merged into one homogeneous region.

The PNN based method adapted to the grid changes better than the RGM. When applying the PNN based method, achieved good results may suggest that the roughness does not show up differently in different areas of the image. Applying the spatial criteria for merging DTOCS formed regions does not improve the results and some of the searched features may even be distorted.

The split and merge approach had disadvantages, which were realized during the experiments. Achieving the optimal results is impossible, because of the complexity reasons, but achieving reasonably good results requires also a lot of human work in finding the right thresholds. If thresholding is not optimal in the SDTOCS and SPM algorithms, the large areas may be erroneously merged much easier than the small areas. In these algorithms, small areas usually have the high mean values. That prevents them from being easily merged with other regions.

8 Conclusions

In this work, cardboard samples were analyzed. There were two data sets, each containing 8 images. The material was base cardboard, and the measurements were done with two devices from the same test area. The images in the first test set were profilometer measurements and in the second test set OptiTopo measurements.

This work concentrated on forming meaningful areas within an analyzed image. An applied method to form the initial regions was the DTOCS algorithm. These regions were further manipulated by the PNN based algorithm and another region merging algorithm. Split and merge algorithms were also applied to the samples. The split phase was done either by the DTOCS algorithm or by simply recursively dividing the image in each step to 4 squares. The region merging algorithm was used for the merge phase.

Generally accepted method was used to compare and to evaluate the quality of the results. Used methods were compared to S_a roughness to find the linear correlation between them. Correlations were analyzed by hypothesis testing. Different samples were set as pairs and it was tested, if they had the same correlation. For the split and merge approach correlation was not used for the estimation of the results, because these methods were based on the S_a roughness map, and therefore had causal relationship with S_a roughness.

Correlation was found for almost all of the samples. Correlation was generally higher in DTOCS formed regions for the profilometer images than for OptiTopo images. The DTOCS algorithm produced the strongest correlations for the biggest grid sizes. No correlation was found between the statistical methods and the DTOCS roughness calculation, when the Gaussian filtering was not used. The result was as expected, since it proves that the DTOCS method measures only smaller scale roughness, and the correlation with S_a roughness is distorted, if waviness is not extracted from the surface before the S_a roughness calculation.

Good results were achieved in this work, when using the PNN based algorithm in combination with the DTOCS method. The PNN based method seemed to improve the correlation in both data sets, when compared to the correlation of initial DTOCS formed regions. These results were better for OptiTopo data. Also for the profilometer images the results were good. The PNN based method does not form large regions without the initial regions, which also define the minimum size for the regions. The objective to divide an image into meaningful regions would not be met without the initialization that is done with the DTOCS algorithm.

The RGM algorithm produced mixed results. The strongest correlation was achieved in the middle size grids, unlike in the other experiments. It seems that the RGM is highly dependent on the correct threshold selection. The results from OptiTopo images had higher correlation than the results from profilometer images. Generally, the correlation for the RGM algorithm was lower than for the PNN.

It is likely, that the used split and merge algorithms detect high variations on a surface relatively well. However, they may leave unrecognized most of the variation below the threshold value. The RGM and PNN algorithms are able to classify also smaller scale roughness to the separate regions, but they hide small exceptional features within the regions, unlike split and merge algorithms.

The results of the RGM algorithm were too volatile, and generally the correlation was not very strong. As being more complex algorithm than the PNN, it is not suggested as such to be used for merging DTOCS formed regions. All the segmentation algorithms that included manually guided parameters should be further developed to automatize the segmentation process. Usability of these algorithms in on-line measurement is highly dependent on the size of the analyzed area, the frequency of the measurements and the computing power of the used equipments. The results of the OptiTopo, which is the only possible measurement device in on-line mesurement, were partly not in the satisfying level. Further reasearch is needed to improve the accuracy of the OptiTopo measurements. The future work of the studied segmentation methods may contain subjects that are based on the PNN algorithms, since the PNN based method provided the most promising results in this work. Efficiency of the PNN based method can be further studied by using code vectors, which contain multiple values. Detection of roughness according to the test samples that have been used to train the implemented system can be studied. These topics may be the steps on the way to the on-line roughness measurement system, which is the ultimate objective of the research project.

REFERENCES

[1] H. Löytty. *Kokoonpuristuva huokoinen paperi, kirjallisuuskatsaus.* Oy Keskuslaboratorio, Espoo, 1998.

[2] J-E. Levlin and L. Söderhjelm. *Papermaking Science and Technology: Pulp and Paper Testing*. Fapet Oy, Helsinki, 1999.

[3] ISO 8791/1-1986 (E). Paper and board - determination of roughness and smoothness (air leak methods). part 1: General method.

[4] P. Wagberg and P. A. Johansson. Surface profilometry - a comparison between optical and mechanical sensing on printing papers. *Tappi Journal*, 76(12):115–121, 1993.

[5] G. G. Barros. *Influence of Substrate Topography on Ink Distribution in Flexography*. PhD thesis, Karlstad University, 2006.

[6] O. Rodionov. Paper surface image processing for roughness quality measurement. Master's thesis, Lappeenranta University of Technology, 2002.

[7] T. Kuparinen. Paper surface roughness measurement using machine vision. Master's thesis, Lappeenranta University of Technology, 2005.

[8] L. Ikonen. *Distance Transforms on Gray-Level Surfaces*. PhD thesis, Lappeenranta University of Technology, 2006.

[9] L. Ikonen, T. Kuparinen, E. Villanueva, and P. Toivanen. Distance transforms on anisotropic surfaces for surface roughness measurement. In *Discrete Geometry for Computer Imagery (DGCI), LNCS 4245*, pages 651–662, Szeged, Hungary, 25-27 Oct 2006.

[10] L. Ikonen and P. Toivanen. Distance and nearest neighbour transform of gray-level surfaces using priority pixel queue algorithm. In *Advanced concepts for intelligent vision system: 7th International Conference*, pages 308–315, Antwerpen, Belgium, 20-23 Sep 2005.

[11] L. Ikonen and P. Toivanen. Shortest routes on varying height surfaces using gray-level distance transforms. *Image and Vision Computing*, 23(2):133–141, 2005.

[12] K. Niskanen. *Papermaking Science and Technology: Paper Physics.* Fapet Oy, Helsinki, 1998.

[13] K. J. Stout, editor. *Characterisation of Roughness*. Penton Press, London, 1993.

[14] National Measurement System Programme for Length Metrology. Softgauges for surface topography. WWW-page. Available from http://161.112.232.32/softgauges/SP_GassianFilter.htm, referenced 4.10.2006.

[15] N. R. Pal and S. K. Pal. A review on image segmentation techniques. *Pattern Recognition*, 26(9):1277–1294, 1993.

[16] R. M. Gray. Vector quantization. *ASSP Magazine, IEEE*, 1:4–29, 1984.

[17] W. H. Equitz. A new vector quantization clustering algorithm. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 37:1568–1575, 1989.

[18] O. Virmajoki and P. Franti. Fast pairwise nearest neighbor based algorithm for multilevel thresholding. *Journal of Electronic Imaging*, 12(4):648–659, 2003.

[19] R. Sedgewick. *Algorithms in C*. Addison Wesley Publishing Company, Inc, Reading(MA), 1990.

[20] O. Virmajoki. *Pairwise Nearest Neighbor Method Revisited*. PhD thesis, University of Joensuu, 2004.

[21] T. Kaukoranta, P. Fränti, and O. Nevalainen. Vector quantization by lazy pairwise nearest neighbor method. *Optical Engineering*, 38(11):1862–1868, 1999.

[22] M. Willebeck-LeMair and A. P. Reeves. Region growing on a hypercube multiprocessor. In *Proceedings of the third conference on Hypercube concurrent computers and applications*, volume 2, pages 1033–1042, Pasadena(CA), USA, Jan 1989.

[23] Amsterdam Institute of Phonetic Sciences. Statistical tests. WWW-page. Available, http://www.fon.hum.uva.nl/Service/Statistics.html, referenced 27.9.2006.

[24] D. C. Montgomery and G. C. Runger. *Applied Statistics and Probability for Engineers*. John Wiley & Sons, Inc, New York, 1994.

[25] D. M. Lane. Hyperstat online statistics textbook. WWW-page. Available, http://davidmlane.com/hyperstat/B8544.html, referenced 18.9.2006.

Appendix I

Table 6: The comparison of the correlation in the statistical methods and DTOCS roughness in Table 2.

	2	3	4	5	6
Profilometer					
1. Moving S_a	0.366	0.000	0.000	0.000	0.000
2. Gauss. only		0.000	0.000	0.000	0.000
3. Gauss. off			0.000	0.000	0.000
OptiTopo					
4. Moving S_a				0.456	0.000
5. Gauss. only					0.000
6. Gauss. off					

Table 7: The comparison of the correlation in different grid types in Table 4.

	2	3	4	5	6	7	8
Profilometer							
1. Net	0.838	0.013	0.534	0.000	0.000	0.000	0.000
2. Tile		0.023	0.412	0.000	0.000	0.000	0.000
3. Hex			0.003	0.000	0.000	0.000	0.000
4. Tile, shift				0.000	0.000	0.000	0.000
OptiTopo							
5. Net					0.900	0.929	0.706
6. Tile						0.836	0.615
7. Hex							0.791
8. Tile, shift							

Table 8: The comparison of the correlation in different grid sizes, in Table 3.

	2	3	4	5	6	7	8	9	10
Profilometer									
1. 40×40	0.003	0.000	0.001	0.000	0.000	0.000	0.000	0.000	0.000
2. 30×30		0.066	0.030	0.001	0.000	0.000	0.000	0.000	0.000
3. 20×20			0.285	0.007	0.000	0.000	0.000	0.000	0.000
4. 10×10				0.054	0.000	0.000	0.000	0.000	0.003
5. 5×5					0.394	0.614	0.486	0.356	0.358
OptiTopo									
6. 40×40						0.100	0.607	0.737	0.656
7. 30×30							0.497	0.331	0.441
8. 20×20								0.575	0.569
9. 10×10									0.802
10. 5×5									

Table 9: The comparison of the correlation in different grid types in the PNN based method, in Table 17.

	2	3	4	5	6
Profilometer					
1. Net	0.903	0.384	0.180	0.141	0.153
2. Tile		0.321	0.222	0.177	0.191
3. Hex			0.027	0.019	0.021
OptiTopo					
4. Net				0.897	0.931
5. Tile					0.966
6. Hex					

Table 10: The comparison of the correlation in different amount of resulting regions in the PNN based method, in Table 16.

	O 3	P 5	P 10	O 5	O 10
Profilometer, 3 regions	0.177	0.154	0.088	0.386	0.652
Optitopo 3, regions		0.003	0.001	0.511	0.215
Profilometer, 5 regions			0.882	0.007	0.012
Profilometer, 10 regions				0.001	0.001
OptiTopo, 5 regions					0.529
OptiTopo, 10 regions					

Table 11: The comparison of the correlation in different grid sizes in the PNN based method (3 resulting regions), in Table 5.

	O 30x30	P 40x40	P 20x20	O 40x40	O 20x20
Profilometer 30x30	0.177	0.220	0.283	0.081	0.653
OptiTopo 30x30		0.902	0.015	0.690	0.368
Profilometer 40x40			0.021	0.602	0.438
Profilometer 20x20				0.005	0.128
OptiTopo 40x40					0.194

Table 12: The comparison of different grid sizes in the RGM algorithm, in Table 19.

	2	3	4	5	6	7	8
Profilometer							
1. 20×20, Th. 0.075	0.002	0.492	0.000	0.113	0.000	0.835	0.307
2. 20×20, Th. 0.05		0.023	0.294	0.847	0.145	0.190	0.253
3. 15×15, Th. 0.05			0.001	0.261	0.001	0.868	0.629
4. 15×15, Th. 0.025				0.489	0.482	0.084	0.077
5. 10×10, Th. 0.05					0.300	0.349	0.530
6. 10×10, Th. 0.025						0.049	0.042
7. 5×5, Th. 0.025							0.641

	9	10	11	12	13	14	15
Profilometer							
1. 20×20, Th. 0.075	0.134	0.034	0.020	0.191	0.001	0.001	0.316
2. 20×20, Th. 0.05	0.178	0.098	0.373	0.007	0.263	0.663	0.406
3. 15×15, Th. 0.05	0.428	0.234	0.134	0.682	0.005	0.012	0.589
4. 15×15, Th. 0.025	0.026	0.003	0.051	0.000	0.677	0.619	0.178
5. 10×10, Th. 0.05	0.552	0.589	0.796	0.321	0.385	0.676	0.634
6. 10×10, Th. 0.025	0.016	0.005	0.033	0.000	0.848	0.304	0.101
7. 5×5, Th. 0.025	0.544	0.463	0.351	0.713	0.071	0.139	0.602
8. 5×5, Th. 0.01	0.891	0.772	0.556	0.794	0.073	0.168	0.913
OptiTopo							
9. 20×20, Th. 0.025		0.850	0.553	0.555	0.040	0.102	0.996
10. 20×20, Th. 0.01			0.558	0.223	0.021	0.049	0.916
11. 15×15, Th. 0.01				0.117	0.080	0.213	0.714
12. 15×15, Th. 0.005					0.003	0.004	0.724
13. 10×10, Th. 0.01						0.458	0.147
14. 10×10, Th. 0.005							0.297
15. 5×5, Th. 0.005							

Table 13: The comparison of different experiments. Each value represents the probability of a pair of correlations (a row and a column) to have the same correlation.

		1.	2.	3.	4.	5.	6.
	Profilometer		T 17:2	T 16:1	T 16:2	T 19:3	T 19:4
1.	DTOCS, 30×30, tile, Table 4:2		0.000	0.003	0.000	0.001	0.458
2.	PNN, 30×30, tile, 3 regions, Table 17:2			0.154	0.088	0.000	0.000
3.	PNN, 30×30, tile, 5 regions, Table 16:2				0.882	0.000	0.006
4.	PNN, 30×30, tile, 10 regions, Table 16:3					0.000	0.000
5.	RGM, 15×15, tile, th. 0.05, Table 19:3						0.001
		7.	8.	9.	10.	11.	12.
	Profilometer	T 3:2	T 17:5	T 16:3	T 16:4	T 19:11	T 19:12
1.	DTOCS, 30×30, tile, Table 4:2	0.000	0.000	0.000	0.000	0.064	0.000
2.	PNN, 30×30, tile, 3 regions, Table 17:2	0.000	0.177	0.386	0.652	0.000	0.000
3.	PNN, 30×30, tile, 5 regions, Table 16:2	0.000	0.003	0.007	0.012	0.001	0.000
4.	PNN, 30×30, tile, 10 regions, Table 16:3	0.000	0.001	0.001	0.001	0.000	0.000
5.	RGM, 15×15, tile, th. 0.05, Table 19:3	0.455	0.000	0.000	0.000	0.134	0.682
6.	RGM, 15×15, tile, th. 0.025, Table 19:4	0.000	0.000	0.000	0.000	0.051	0.000
	OptiTopo						
7.	DTOCS, 30×30, tile Table 4:7		0.000	0.000	0.000	0.001	0.023
8.	PNN, 30×30, tile, 3 regions, Table 17:5			0.511	0.215	0.000	0.000
9.	PNN, 30×30, tile, 5 regions, Table 16:5				0.529	0.000	0.000
10.	PNN, 30×30, tile, 10 regions, Table 16:6					0.000	0.000
11.	RGM, 15×15, tile, th. 0.01, Table 19:11						0.117
12.	RGM, 15×15, tile th. 0.005, Table 19:12						

Appendix II

Table 14: The correlation coefficients of the grid type comparison for the cardboard data sets, Figure 13.

	Corr.	*Conf. interval, lower*	*Conf. interval, upper*	*Seed points*	*Distance between seedpoints,*
Profilometer, wavelength 250 μm					
Net 41×41	0.987	0.93	1.00	1600	120 μm
Tile 41×41	0.990	0.94	1.00	1580	$120 - 240$ μm
Hex 41×41	0.974	0.86	1.00	1192	$120 - 240$ μm
OptiTopo,wavelength 250 μm					
Net 41×41	0.580	-0.21	0.91	1681	117 μm
Tile 41×41	0.574	-0.22	0.91	1661	$117 - 234$ μm
Hex 41×41	0.603	-0.18	0.92	1242	$117 - 234$ μm

Table 15: The correlation coefficients of the grid size comparison for the cardboard data sets, Figure 14.

	Corr.	*Conf. interval, lower*	*Conf. interval, upper*	*Seed points*	*Distance between seedpoints,*
Profilometer, wavelength 250 μm					
30×30	0.963	0.81	0.99	900	160 μm
20×20	0.960	0.79	0.99	400	240 μm
10×10	0.949	0.74	0.99	200	480 μm
5×5	0.874	0.44	0.98	100	950 μm
OptiTopo, wavelength 250 μm					
30×30	0.762	0.12	0.95	900	161 μm
20×20	0.693	-0.02	0.94	400	234 μm
10×10	0.625	-0.14	0.92	200	476 μm
5×5	0.147	-0.62	0.77	100	952 μm

Table 16: The correlation coefficients between different amount of resulting regions (3 - 10) of PNN and regional moving S_a roughness in 30×30 tile grid for the cardboard data sets.

	Grid,	$Corr.$	$Conf.$	$Conf.$	$Seed$	$Distance$
			$interval,$	$interval,$	$points$	$between$
			$lower$	$upper$		$seedpoints,$
Profilometer, wavelength 250 μm						
1	3 regions	0.9392	0.8626	0.9737	885	$160 - 320$ μm
2	5 regions	0.8720	0.7695	0.9307	885	$160 - 320$ μm
3	10 regions	0.8647	0.7962	0.9113	885	$160 - 320$ μm
OptiTopo, wavelength 250 μm						
4	3 regions	0.9731	0.9379	0.9885	841	$161 - 322$ μm
5	5 regions	0.9617	0.9282	0.9797	841	$161 - 322$ μm
6	10 regions	0.9510	0.9244	0.9684	841	$161 - 322$ μm

Table 17: The correlation coefficients between different grid types in PNN (3 resulting regions)and regional moving S_a roughness in 30×30 grid for the cardboard data sets.

	Grid,	$Corr.$	$Conf.$	$Conf.$	$Seed$	$Distance$
		$interval,$	$interval,$	$points$		$between$
			$lower$	$upper$		$seedpoints,$
Profilometer, wavelength 250 μm						
1	Net	0.9346	0.8528	0.9717	900	160 μm
2	Tile	0.9392	0.8626	0.9737	885	$160 - 320$ μm
3	Hex	0.8906	0.7604	0.9520	660	$160 - 320$ μm
OptiTopo, wavelength 250 μm						
4	Net	0.9709	0.9328	0.9875	841	161 μm
5	Tile	0.9731	0.9379	0.9885	841	$161 - 322$ μm
6	Hex	0.9724	0.9362	0.9882	663	$161 - 322$ μm

Table 18: The correlation coefficients between grid type comparison in RGM and regional moving S_a roughness in 15×15 grid for the cardboard data sets.

	Type	$Corr.$	$Conf.$ $int.$ low	$Conf.$ $int.$ $high$	$Dist.$ $seed$ $points, \mu m$	$Samples,$ $total$	$Regions,$ $final$
Profilometer, wavelength 250 μm, threshold 0.05							
1	Tile	0.5522	0.4535	0.6375	$150-300$	222	$20-40$
2	Net	0.5773	0.4745	0.6646	150	192	$16-32$
3	Hex	0.5351	0.4086	0.6414	$150-300$	147	$4-24$
OptiTopo, wavelength 250 μm, threshold 0.01							
4	Tile	0.6338	0.5712	0.6891	$146-292$	399	$42-55$
5	Net	0.6672	0.6093	0.7180	146	404	$43-58$
6	Hex	0.6299	0.5522	0.6967	$146-292$	271	$26-42$

Table 19: The correlation coefficients between grid size comparison in RGM and regional moving S_a roughness in tile grid for the cardboard data sets.

	Grid, 250 μm	$Corr.$	$Conf.$ $int.$ low	$Conf.$ $int.$ $high$	$Dist.$ $seed$ $points, \mu m$	$Samples,$ $total$	$Th.$	$Regions,$ $final$
Profilometer, wavelength 250 μm								
1	20x20	0.5061	0.4067	0.5936	$110-220$	246	0.075	$22-40$
2	20x20	0.6692	0.6146	0.7174	$110-220$	448	0.05	$41-81$
3	15x15	0.5522	0.4535	0.6375	$150-300$	222	0.05	$21-41$
4	15x15	0.7043	0.6603	0.7434	$150-300$	570	0.025	$60-94$
5	10x10	0.6545	0.4887	0.7748	$230-460$	65	0.05	$6-15$
6	10x10	0.7325	0.6607	0.7910	$230-460$	199	0.025	$22-41$
7	5x5	0.5321	0.2758	0.7177	$450-900$	43	0.025	$3-12$
8	5x5	0.5920	0.4469	0.7068	$450-900$	99	0.01	$7-19$
OptiTopo, wavelength 250 μm								
9	20x20	0.6028	0.5110	0.6809	$110-220$	219	0.025	$23-36$
10	20x20	0.6119	0.5674	0.6528	$110-220$	828	0.01	$96-123$
11	15x15	0.6338	0.5712	0.6891	$146-292$	399	0.01	$43-58$
12	15x15	0.5734	0.5267	0.6166	$146-292$	860	0.005	$100-119$
13	10x10	0.7228	0.6390	0.7896	$227-454$	159	0.01	$17-23$
14	10x10	0.6864	0.6245	0.7398	$227-454$	327	0.005	$34-45$
15	5x5	0.6033	0.4246	0.7368	$447-894$	67	0.005	$6-11$