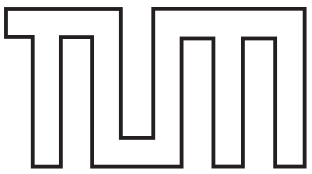


Technische Universität München
Chair of Media Technology
Prof. Dr.-Ing. Eckehard Steinbach

Master Thesis

Image Based Haptic Feature Extraction

Author:	Albert Iepure
Matriculation Number:	03650065
Address:	Arcisstr. 21
	D-80333, München
Advisor:	Matti Strese
Begin:	17.09.2015
End:	03.12.2015

With my signature below, I assert that the work in this thesis has been composed by myself independently and no source materials or aids other than those mentioned in the thesis have been used.

München, November 24, 2015 _____

Place, Date Signature

München, November 24, 2015 _____

Place, Date Signature

Abstract

Images of different object surfaces convey important information about haptically perceptible textures. The extraction of tactile information of different materials by making use of inexpensive technologies can have practical and commercial applications in e-commerce or robotics. However, differences in distance, rotation, lighting and focus conditions are hurdles which need to be overcome to extract robust image-based features that will allow a successful surface classification task.

In this work, eleven haptically relevant features are introduced, which have a low to invariant dependency on different camera conditions. These are used for a robust machine learning-based approach for surface classification.

A database of 690 images, corresponding to 69 different textures, is used to extract haptically relevant features. Perceptually-relevant image features such as roughness, softness and regularity are used to correctly classify the textures. The extracted features are perceptually relevant so that they can be also used in future work for the retrieval of the most similar textured surface to a classified one.

Experimental results and the evaluation of a cross-validated naive Bayes classifier show that the proposed approach allows for the successful classification of textured surfaces under varying camera conditions, a maximum prediction accuracy of 85.8% being achieved. When a subset of 6 features is selected, a classification accuracy of 82.5% is obtained.

Kurzfassung

Bilder von verschiedenen Objektoberflächen vermitteln wichtige Informationen über deren haptisch wahrnehmbare Texturen. Die Extraktion taktiler Informationen aus unterschiedlichen Materialien durch den Einsatz kostengünstiger Technologien ermöglicht die Oberflächenklassifikation, welche im e-Commerce oder der Robotik kommerzielle Anwendung finden kann. Unterschiede in der Distanz, der Drehung, der Beleuchtung und des Fokus während der Bildaufnahme sind jedoch Hürden, die zu überwinden sind, um robuste bildbasierte Merkmale für eine erfolgreiche Oberflächenklassifikation zu ermöglichen.

In dieser Arbeit werden haptisch relevante Merkmale definiert, die eine geringe bis invariante Abhängigkeit von verschiedenen Aufnahmebedingungen haben. Diese werden für einen robusten Maschinellen Lernansatz zur Oberflächenklassifizierung verwendet.

Eine Datenbank von 690 Bildern mit entsprechend 69 unterschiedlichen Texturen wird verwendet, um haptisch relevanten Merkmale zu extrahieren. Wahrnehmungsbezogene Bildmerkmale wie Rauheit, Weichheit und Regelmäßigkeit werden verwendet, um die Oberflächen korrekt zu klassifizieren. Die extrahierten Merkmale sind perzeptuell relevant, sodass sie auch bei zukünftigen Arbeiten zur Suche von den ähnlichsten texturierten Oberflächen auf den verwendeten Klassifikator zurückgegriffen werden kann.

Experimentelle Ergebnisse und die Auswertung eines Naive-Bayes-Lernansatzes zeigen, dass der vorgeschlagene Ansatz die erfolgreiche Klassifizierung von texturierten Oberflächen unter variierenden Aufnahmebedingungen mit einer Klassifikationsgenauigkeit von 85.8% erreicht. Wenn eine Teilmenge von sechs der definierten Merkmale ausgewählt wird, erreicht das Klassifikationssystem 82.5%.

Contents

Chapter 1

Introduction

The ability to determine the tactile feeling of different materials by making use of inexpensive technologies (e.g. visual inspection of a given surface) is a topic of utmost interest in the image processing research field. This also has considerable practical applications in the web-based e-commerce. Such an application, which enables users to see and feel a variety of materials through a touchscreen becomes an increasingly popular topic for different fields of research nowadays.

The human tactile system relies not only on direct touch but also on the visual inspection of a given textured surface. This work presents a set of eleven touch-based features, e.g. roughness, coarseness, edginess, which match the human perception. These are extracted from a newly created database of 690 images, corresponding to 69 textures, that is used for the surface classification experiments. In addition, this work also presents the results obtained after the feature extraction on a second newly created database of 690 magnified images, where also 10 magnified images per texture are considered.

Humans are able to visually examine a given structure and evaluate its roughness. Therefore, a visio-haptic subjective experiment is performed to evaluate the visual and tactile roughness of the considered materials. The outcome is used to define an image-based roughness feature and to evaluate the perceived roughness mismatch of touching a surface compared to merely seeing it.

Evidence is presented that surface classification should not only be limited to the evaluation of image data simply because humans can also use more than one sense for the classification among textured surfaces. However, experimental results show that the proposed approach allows for the successful classification of textured surfaces under varying conditions, a maximum prediction accuracy of 86% being obtained when a tenfold cross-validated naive Bayes classifier is used.

Chapter 2

Related Work

2.1 Textural Properties

The texture of an image is an important source of information that humans employ when analyzing a scene. Basically, it refers to the arrangement of the basic building blocks of a material. For example, in a digital image, the texture is described by the spatial interrelationships among the image pixels. Visually, these spatial dependencies are perceived as changes in the intensity patterns or the so-called gray tones of the image. Therefore, it has long been an important research topic in image processing. The end result of texture analysis is represented by its application in industrial, biomedical and remote sensing areas. Therefore, appropriate texture descriptors could provide powerful means in applications of texture browsing, coarse texture classification and texture retrieval.

In [TMY78], a series of psychological measurements on human subjects are performed and six basic textural features, e.g. coarseness, contrast, directionality, linelikeness, regularity, roughness are determined. These features were approximated to obtain a computational form which can be used by a machine to classify and select different textures.

In [HSD73], Haralick et al. developed a computational method for a set of 14 features based on image properties, which do not have any visual correspondence. Even if these parameters can be used to successfully classify images, the lack of perceptual significance demands more effort to be invested such that features that are general and familiar to any human being are identified.

With respect to the perception of textures, considerable research work was undergone in the psychological fields. The most notable observation is that Julesz has verified that the discrimination of textures depends primarily on the difference in lower order statistics [Jul62]. For example, first and second moments, as used in the arithmetic mean and variance are examples of low order statistics.

2.2 Structure of Texture Patterns

A visual texture can be regarded as what constitutes a macroscopic region. The structure it shows is attributed to the patterns which repeat, whose elements or primitives are arranged obeying a so called "placement rule", which can be expressed mathematically as $f = R(e)$, where R represents a specific placement rule and e models an element. In this way, a texture f can be obtained if R is a function which satisfies the previously mentioned conditions. A "microtexture" can be described by recursively replacing e by f.

There are two levels of considering a texture, either statistical or structural. From the statistical point of view, the texture is inspected from the macroscopic level as a functional combination of R and e. In this way, the difference between the textured regions is measured by employing the statistics of the local properties. From the structural point of view, more information is needed about both R and e to successfully describe a texture.

When the computational analysis of a given texture is considered, it is fairly cumbersome to parse its structure on a general level because the factors R and e are heavily intertwined and they both influence the texture pattern. In the case of naturally occurring textures, where R and e usually have a considerable variation, it is impossible to determine them separately. Therefore, a statistical approach is needed to solve the problem, where a set of statistics is extracted from a local set of measurements.

As previously mentioned, the second order statistics produce the difference among textures, but there exists no general method to pinpoint whether the R or e are the cause for the total difference and what kind of statistics determine the variation of each.

In [TMY78], Tamura et al. describes a set of features through which all input patterns are measured and yield well distributed results. The features which successfully describe all visual textures taken into account are presented in the following section.

2.3 Features Defined by Tamura

In [TMY78] a series of psychological experiments were conducted to construct a series of psychometric prototypes against which the computational features can be compared.

A set of 16 digital pictures of different textures was photographed, scanned, quantized into 64 gray levels and displayed on a CRT monitor. These pictures were shown to the human subjects, who were 28 men and 20 women. After performing the tests and interpreting the data, the following remarks were made:

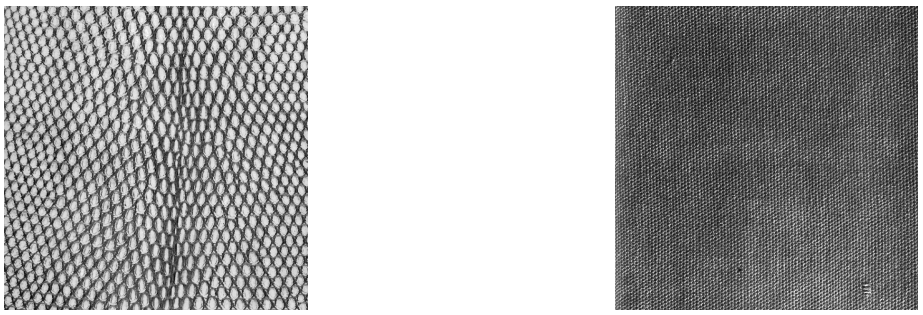
- A considerable correlation among coarseness, contrast and roughness exists.
- High roughness in a texture is perceived by humans when it is coarse, high contrast, nondirectional, blob-like and irregular.

- Coarseness and contrast greatly influence roughness, because of the possible intrinsic ambiguity between the rough and course features when dealing with untrained subjects.
- The correlation between the directionality and linelikeness features is extremely high, because most of the directional textures consist of linelike elements.
- Linelikeness is interpreted by subjects as a global property.
- Regularity is interpreted as the variance of texture elements.

By considering these aspects, difficulties occur in describing and explaining the visual specifications. Several deficiencies were caused by the imperfect understanding of the definitions. The same measurement for all human subjects in the visual examination part cannot be expected, since individual differences will occur. Moreover, the computational techniques are likely to achieve a lower level of correspondence with respect to the human perception than the psychological experiments.

2.3.1 Coarseness

Coarseness is, along with contrast and directionality, one of the most fundamental textural features as mentioned in [TMY78] and is related to the appearance of large structured elements in an image. For example, in the case of two patterns differing only in scale, the magnified one is coarser, whereas when it comes to patterns that have different structures, the bigger its element size is or the less often its elements are repeated, the coarser it is perceived by humans, as shown in Figure 2.1.



(a) (b)

Figure 2.1: Different levels of coarseness. a) Coarse texture. b) Fine texture.

The algorithm starts by computing and storing the average gray level (AGL) of subwindows of size $2^k \times 2^k$, with $k \in \{1, 2, 3, 4, 5\}$, which are centered at every pixel within the image,

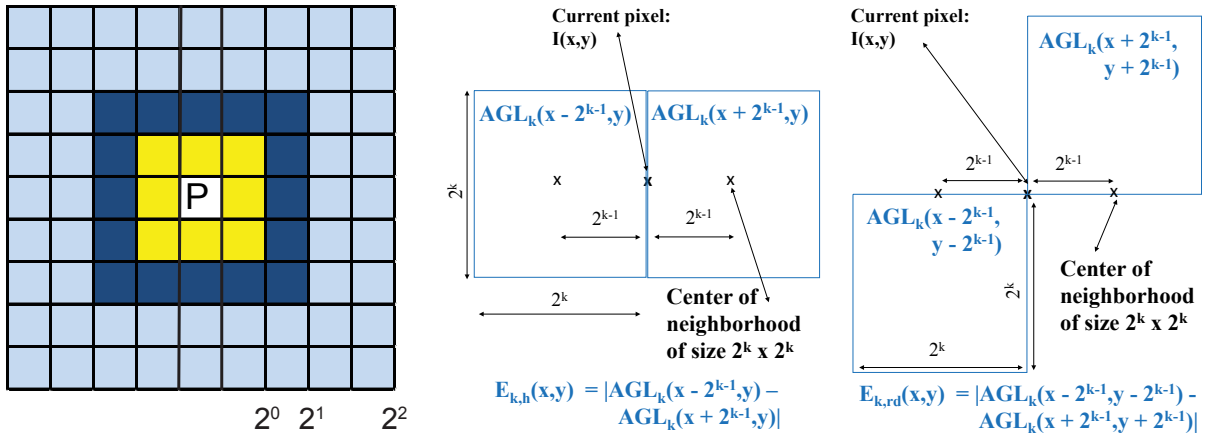
where the AGL is computed as:

$$AGL_k = \frac{\sum_{i=x-2^{k-1}}^{x+2^{k-1}-1} \sum_{j=y-2^{k-1}}^{y+2^{k-1}-1} I(i,j)}{2^{2k}} \tag{2.1}$$

Afterwards the symmetric subtraction of average gray levels is undergone. This is performed for non-overlapping, adjacent subwindows in the horizontal and vertical directions with respect to the current pixel for the complete range of sizes, as shown in Figure 2.2b. Afterwards, the results are stored. For example, in the vertical direction, this is computed as $E_{k,v}(x,y) = |AGL_k(x, y+2^{k-1}) - AGL_k(x, y-2^{k-1})|$. For each pixel, the k value which corresponds to the maximum gradient of average gray levels of the adjacent subwindows in all directions is selected and $S_{best}(x,y) = 2^k$ is obtained. The coarseness feature (F_{crs}) is determined as the average of S_{best} over the entire picture:

$$F_{crs} = \frac{1}{m \times n} \sum_{i=1}^{m} \sum_{j=1}^{n} S_{best}(i,j) \tag{2.2}$$

where m and n are the effective width and height of the picture, respectively. An image will contain textures at different scales and through S_{best} it is intended to select the largest detected size for the structured element, e.g. a large size for the texture element is chosen when coarse texture is present, even in the presence of a microtexture, and a small size for the texture element is chosen when only a fine texture is present.

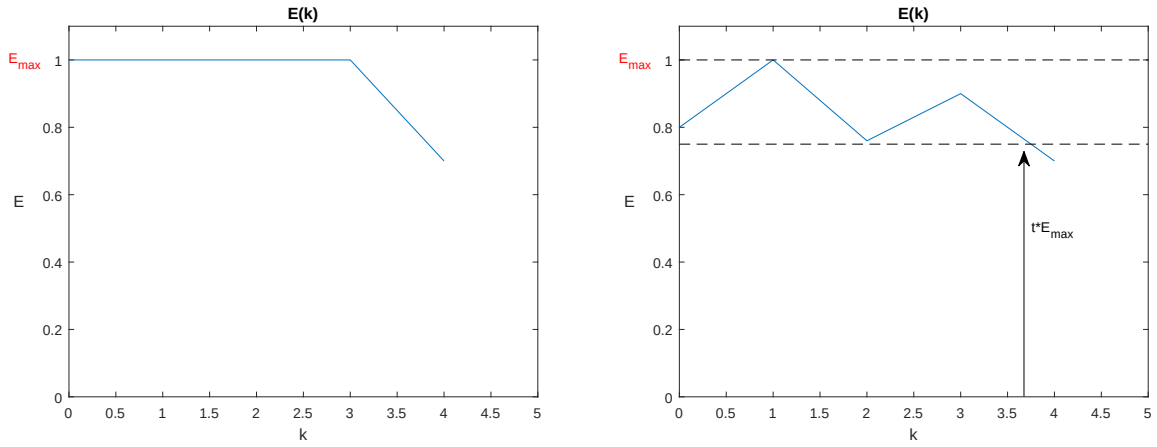


(a) (b)

Figure 2.2: a) Subwindows centered at current pixel P of sizes $k \in \{1,2,3\}$ are shown with different colors. b) The adjacent windows with respect to the current pixel in the horizontal (left) and right diagonal direction (right).

A refinement in the stage of determining the best size of the subwindow is implemented to take into account the case when the maximum of E appears in more than two operator

sizes, the largest size can be considered. To illustrate this, a black and white checkerboard can be considered. At a point on an edge of a square, the value of E stays constant as the averaging size increases (E is the gray level difference of alternate squares). Just after the averaging size exceeds the size of a square element d, a decrease in the output of E occurs. Consequently, the best size is determined to be d. This is shown in Figure 2.3a. For natural textures there are considerable variations in gray levels, element size, and the placement rule. In such a situation, differences between adjacent large neighborhoods tend to be canceled out to some extent by performing the averaging operation, even though macrotexture still produces clear differences in a small size operator. A case exists, which is shown in Figure 2.3b, where the maximum decision can produce a significant change in the best size even though the difference between the maximum and the second maximum values of E_k is extremely small. Due to this, as suggested in [TMY78], the procedure of determining the best size of the subwindow is modified. For each point, if there exist some k such that $k > k_{max}$ (which yields E_{max}) and $E_k > t \cdot E_{max}$, the largest k for S_{best} is taken (the parameter t is a certain constant less than 1). Otherwise, the original procedure is retained. The most perceptually relevant results are obtained for $t = 0.9$, as discussed in [TMY78]).



(a) (b)

Figure 2.3: Behaviour of the best size detectors. a) Ideal case. b) Noisy case. By using the threshold t, the maximum decision does not produce any more significant changes in the best size k.

To test the algorithm and determine whether the obtained results are in accordance with what the humans perceive, two textures from the Brodatz database [Bro66], namely $D35$ in Figure 2.1a and $D77$ in Figure 2.1b were selected. After running the algorithm, the coarseness feature values of $F_{crs_{D35}} = 16.74$ and $F_{crs_{D77}} = 11.17$ are obtained, which coincides with the human perception.

2.3.2 Contrast

The simplest way of varying picture contrast is to shrink or stretch its gray scale. Through this means a change in picture quality, not in picture structure, is achieved. Therefore, when two different patterns differ only in terms of their gray level distribution, the difference between their contrast can be measured and quantified, as shown in Figure 2.4. As it is mentioned in [TMY78], a plethora of factors influence the contrast difference between two texture patterns which have different structures, the most notable being:

- The dynamic range of gray levels has a major influence on contrast.
- The ratio of black and white areas also influences how a texture is perceived.
- The sharpness of edges play an important role when comparing two images with the same gray level distribution, the image with sharp edges usually has higher contrast.
- The period of the pattern repetition also influences how contrast is perceived.

(a) (b)

Figure 2.4: Different levels of contrast. a) High contrast. b) Low contrast.

To develop a relevant contrast feature, the four factors previously presented are attempted to be approximated step by step. The first factor can be simply expressed by the range of gray levels. For the second factor, the variance σ^2 or the standard deviation σ about the mean of the gray levels probability distribution is a better descriptor than contrast, because σ and σ^2 measure the dispersion in the distribution, approximating the second factor to some extent. The result is not actually the desired one for the cases where the distribution of gray levels has a single peak which is highly biased either to the black or white ends of the histogram, as shown in Figure 2.5d. Therefore, the kurtosis α_4, which is the fourth standardized moment, is used to measure the polarization of black and white areas on the gray level histogram as follows:

$$\alpha_4 = \frac{\mu_4}{\sigma^4} = \frac{E[(\mathbf{H} - \mu)^4]}{(E[(\mathbf{H} - \mu)^2])^2} \tag{2.3}$$

where $\mathbf{H}$ is the histogram vector of the image I, μ_4 is the fourth moment about the mean of $\mathbf{H}$ and σ is the standard deviation of $\mathbf{H}$. Kurtosis measures the width of the peak of

the probability distribution function of a real-valued random variable. For symmetric and unimodal distributions, the kurtosis measures both the width of the peak of the distribution and the heaviness of its tail, as described in [BM88]. A unimodal distribution is a distribution with a single mode, the mode being the value at which the probability density function reaches its maximum value. A heavy-tailed distribution is a probability distribution whose tails are not exponentially bounded, which means that they have heavier tails than the exponential distribution. The kurtosis has the lower bound of the squared skewness plus 1, this lower bound being achieved when twin peaks occur and there does not exist any upper limit for it.

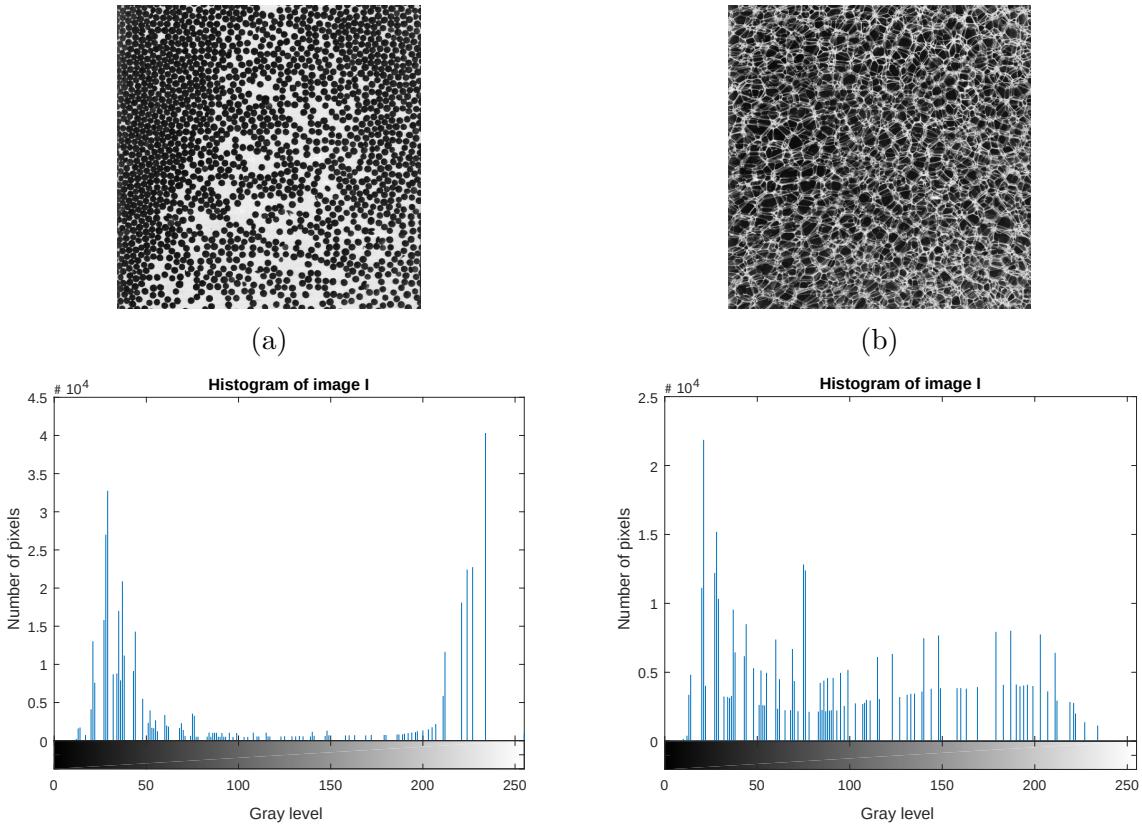


Figure 2.5: Texture images from the Broadatz database and their histograms [Bro66]. a) Texture $D67$. b) Histogram of $D67$. c) Texture $D111$. d) Histogram of $D111$.

Combining σ and α_4, the contrast computational feature (F_{con}) is obtained:

$$F_{con} = \frac{\sigma}{\alpha_4^n} = \frac{\sigma}{(\frac{\mu_4}{\sigma^4})^n} = \frac{\sigma^2}{\mu_4^n}, \text{ where } n = 0.25 \tag{2.4}$$

In [TMY78] it was experimentally concluded that $n = 0.25$ yielded the best correlation coefficient between the computational and perceptional contrast feature.

Even if only the first two factors are considered to obtain this formula, the other two factors, which stand for the pictorial structure of the image, are not accounted for because the experiments performed in [TMY78] yielded results which were considered good enough.

After running this algorithm on the images which are shown in Figure 2.4a and Figure 2.4b, the contrast feature values of $F_{con_{D67}} = 0.2080$ and $F_{con_{D111}} = 0.0707$ are obtained, which coincides with the observation that $D67$ has a higher contrast than $D111$. Thus, the algorithm outputs values which are perceptually accurate.

2.3.3 Directionality

This feature represents a global property over the considered surface, involving both element shape and periodicity. It is not a measure of the orientation, but of the presence of it in the image, and is rotation invariant, meaning that two patterns which differ only in orientations will have the same degree of directionality, as it is stated in [TMY78].

Directionality is one of the most significant texture features which are well perceived by the human visual system. The directionality of a texture can also be obtained from the Fourier spectrum of its picture, but it is a time-consuming process to compute it. This is why the histogram of local edge probabilities against their directional angle is used as a measure of directionality. It was shown that this histogram represents global features of the input picture such as long lines and simple curves, as in [MMM73]. This makes use of the gradient, which is a vector, having therefore both magnitude and direction.

The algorithm starts by computing for each pixel the magnitude $|\Delta G|$ and direction θ of the gray level gradient, which are approximated as follows:

$$|\Delta G| = \frac{|G_H| + |G_V|}{2} \text{ and } \theta = tan^{-1}(\frac{G_V}{G_H}) + \frac{\pi}{2} \tag{2.5}$$

where G_H and G_V are the image gradients along the X and Y axes. The Prewitt kernels for the horizontal and vertical directions are described by the following 3×3 operators:

$$P_H = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \text{ and } P_V = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} \tag{2.6}$$

To obtain the gradients, the two 3×3 kernels are convoluted with the original image I, obtaining $\mathbf{G}_H = \mathbf{P}_H * \mathbf{I}$ and $\mathbf{G}_V = \mathbf{P}_V * \mathbf{I}$, where $*$ stands for the 2D convolution. The Prewitt operator computes an approximation of the gradient of the image intensity function, giving the direction of the largest possible increase from low values to high values of the intensity level and the rate of change in that direction. The obtained result shows the magnitude of change of the image at that specific point and also how that edge is likely to be oriented. The gradient of a two-variable function, e.g. the image intensity function, is at each point a 2D vector whose components are given by the derivatives in the X and

Y directions. As it is mentioned in [Pre70], the Prewitt kernels are separable filters that compute the gradient with smoothing, because they can be decomposed as the products of an averaging and differentiation kernel, as it is the case for example of $P_H = [111]^T \cdot [-101]$.

The θ values, which lay in the interval $[0, \pi]$, are used to form a $n = 16$ bin histogram which is obtained by counting the number of points with magnitude $|\Delta G| > t$, where t is a threshold whose value is $t = 12$, used to avoid counting unreliable directions which cannot be regarded as edge points. As in [TMY78], the desired histogram H_D is obtained as follows:

$$H_D(k) = \frac{N_\theta(k)}{\sum_{i=0}^{n-1} N_\theta(i)}, \; k \in \{0, 1, \; ... \; , n-1\} \tag{2.7}$$

where $N_\theta(k)$ is the number of points that fulfill the following two conditions:

$$\frac{(2k-1)\pi}{2n} \leq \theta \leq \frac{(2k+1)\pi}{2n} \text{ and } |\Delta G| \geq t \tag{2.8}$$

Various values of t were tested and it was concluded that the shape of the histogram H_D is not influenced by the value of the threshold t.

To measure the directionality quantitatively from H_D, the sharpness of the peaks is computed by summing the second moments around each peak from valley to valley, if multiple peaks are determined to exist. The directionality feature (F_{dir}) is defined as follows:

$$F_{dir} = 1 - r \cdot n_p \cdot \sum_{p}^{n_p} \sum_{\Phi \in w_p} (\Phi - \Phi_p)^2 \cdot H_D(\Phi) \tag{2.9}$$

where n_p is the number of peaks, Φ quantized direction angle, Φ_p is the p^{th} peak position of H_D, w_p is the p^{th} peak between two adjacent valleys and r is a normalizing factor which is related to the quantization levels of of Φ.

The inner summation in the formula for the directionality feature is for each local maxima. The bins which are far away from the peak position have more contribution to the directionality feature than the bins which are near the peak, due to the term $(\Phi - \Phi_p)^2$. It also follows that the peak bins do not contribute to the feature, because the correspondent angle difference is zero. However, any directionality measurement can consider the contribution of each individual peak to truly match the perceived reality. The term $(\Phi - \Phi_p)^2$ biases the directionality feature in terms of rotation, because a rotation of the image implies that the position of the peaks in the histogram will be changed, determining that the two directionality values obtained for the two images will be different.

No more than two peaks are considered in [TMY78] and for this reason, in the proposed algorithm, the decision whether n_p is either 1 or 2 has to be taken. Therefore, if $H_D(\nu_{12})/H_D(\Phi_2) < 0.5$, $H_D(\nu_{21})/H_D(\Phi_2) < 0.5$ and $H_D(\Phi_2)/H_D(\Phi_1) > 0.2$ then $n_p = 2$, if not $n_p = 1$. Here, ν_{12} and ν_{21} are the positions of valleys from the first peak Φ_1 to the second peak Φ_2 and vice versa, respectively, as it is highlighted in Figure 2.6.

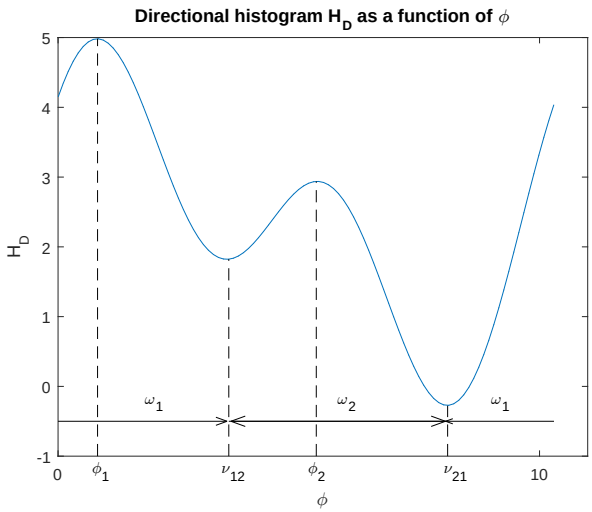
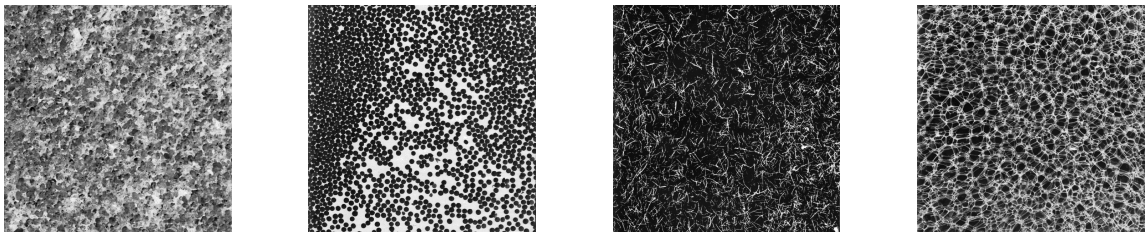


Figure 2.6: Definition of symbols for bidirectional the case.

2.3.4 Linelikeness

This feature determines the amount of line occurrences in an image and is intended to reinforce the three features which were described beforehand, especially when the patterns cannot be discriminated in terms of directionality. With regard to tactile perception, it distinguishes between irregular, coarser structures and those that have visual lines e.g. fur, wooden surfaces or surfaces which contain fibers. For example, textures $D28$, $D67$, $D109$ and $D111$ are all considered nondirectional. However, $D109$ and $D111$ are considerably more linelike than $D28$ and $D67$, which are blob-like, as shown in Figure 2.7.

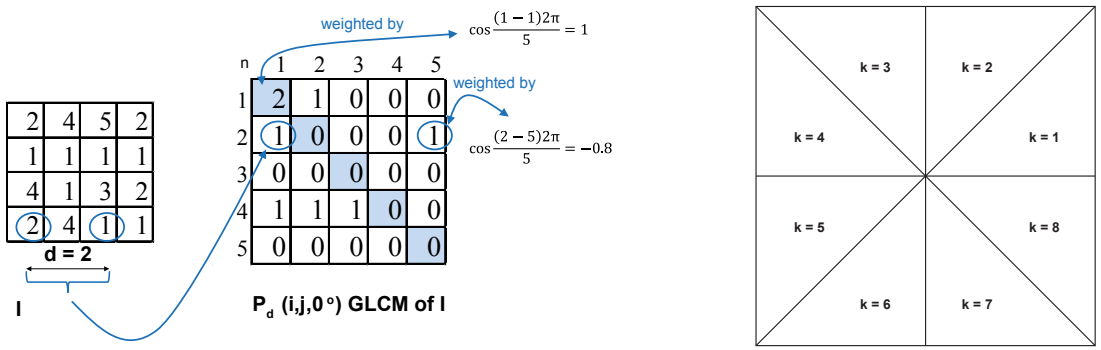


(a) (b) (c) (d)

Figure 2.7: Natural texture patterns from the Brodatz database [Bro66].
a) $D28$. b) $D67$. c) $D109$. d) $D111$.

Linelikeness represents an element of texture that is composed of lines. As suggested in[TMY78], the gray-level co-occurence matrix ($GLCM$) is employed for the calculation, which is defined in [HSD73]. The element $P_d(i, j)$ of the $GLCM$ is defined as the relative frequency, with which two image intensities (at positions i and j) are separated by a distance d as it is illustrated in Figure 2.8a. Here, the 0° argument of P_d indicates that the

$GLCM$ is computed for the horizontal direction, its element in position $(1, 1)$ represents the number of horizontally adjacent pixels in I that are separated by the distance $d = 2$ and have the values $(1, 1)$. This matrix can be either symmetric, when $P(i, j) = P(j, i)$, or nonsymmetric, depending on whether the pair of pixels is counted once or twice (once forward and once backward). Before proceeding with the actual computation, the $GLCM$ values are normalized in order for its elements to represent probabilities instead of numbers. The normalization process implies the division of the number of counts by the total number of counted pixel pairs. In the horizontal direction, for $d = 1$, there will be $2 \cdot (N_x - 1)$ neighboring resolution cells on each row, times N_y rows, yielding a total of $2 \cdot N_y \cdot (N_x - 1)$ nearest horizontal neighbor pairs. For the nearest right diagonal neighbor ($d = 1$, $\theta = 45°$) there will be $2 \cdot (N_x - 1)$ diagonal resolution cell pairs for each row except the first, for which none exists, and there are N_y rows, yielding $2 \cdot (N_x - 1) \cdot (N_y - 1)$ nearest right diagonal neighbor pairs. Similarly, there will be $2 \cdot (N_x - 1) \cdot (N_y - 1)$ nearest left diagonal neighbor pairs and $2 \cdot N_x \cdot (N_y - 1)$ nearest vertical neighbor pairs. The matrix is normalized by dividing each element in the matrix by the corresponding number of neighboring pairs.



(a) (b)

Figure 2.8: a) Illustration of the $GLCM$ P_d matrix for $d = 2$ that is computed based on the image I. b) The eight directions for which the linelikeness feature is computed.

Through this matrix, a line is modeled as a pair of neighboring pixels with the same values, which represent the entries on the main diagonal of P_d, or with similar values, which represent the rest of the entries. The values of the weighting factor represented by the cosine function decrease as the elements of the $GLCM$ are further apart from one another, diminishing thus the contribution of the pairs of pixels with radically different intensity values, which are very likely not to represent a line.

The image linelikeness feature (F_{lin}) is defined as:

$$F_{lin} = \sum_{k=1}^{8} \frac{\sum_{i}^{n} \sum_{j}^{n} P_d(i, j, k) \cdot cos(\frac{(i-j)2\pi}{n})}{\sum_{i}^{n} \sum_{j}^{n} P_d(i, j, k)} \tag{2.10}$$

where n is the width of the P_d matrix. In this equation, the distance $d = 4$, as suggested in [TMY78].

2.4 Other Features

2.4.1 Complexity

In the past 30 years, a number of texture analysis methods have been suggested. For example, in [Har79] various methods are grouped in statistical and structural techniques. The main disadvantage of both these techniques is that they do not have general applicability, in the sense that they cannot be applied to different classes of textures with a reasonable success rate. Statistical techniques are usually good for microtextures and are poor performers in the case of macrotextures, the opposite is the case for the structural techniques. In contrast to these, the human perception mechanism seems to adapt to any kind of texture, the main properties that are used by the humans to differentiate between different textural patterns involve coarseness, complexity, contrast, directionality, shape and texture strength. Therefore, when trying to obtain a system for the automatic recognition and classification of textures, computational measures of the previously mentioned textural properties can be implemented.

In [AK89], computational measures corresponding to the complexity perceptual attribute are presented. The computational form for each feature was obtained by expressing a perceptual description of the property in terms of spatial changes in intensity and/or its dynamic range, because the amount to which a given texture exhibits a specific property is heavily dependent on these two factors. When taking a digital image, the information concerning the spatial changes in intensity can be obtained by looking at the difference between the gray level of each pixel and at the gray levels of its neighboring pixels.

Complexity refers to the visual information content of a texture, a texture being more complex if the information content is high, which happens when there are many patches or primitives present in the texture, and also when the primitives have different average intensities. For example, a texture with a large number of sharp edges and/or lines will be considered complex, as it is shown in Figure 2.9. All these depend on the spatial period of the pattern repetition and on the dynamic range of the gray scale, making thus complexity to be partly correlated with contrast.

Usually the textures whose spatial rate of change in intensity is slight tend to have a couple of different values of gray tones, but with a high probability of each value to occur. As a consequence, in this kind of textures there can be a small number of patches with different average intensity levels, but these patches will be large. Another consequence of the high level of local uniformity in the intensity values is that a few edges are going to occur. This is why a texture in which there are very rapid spatial changes in the intensity values has

(a)

(b)

Figure 2.9: Different levels of textural complexity.
a) Complex texture. b) Simple texture.

a higher likelihood of being complex than a texture that has a high degree of uniformity when it comes to the local intensity. Moreover, the rapid changes in intensity usually imply a large number of intensity values, each having a low probability to occur. This is why the size of primitives and occurrence probabilities of gray tone values tend to have an inverse relationship with complexity.

Considering these, primordial to the development of the previously mentioned feature is the computation of a so called neighborhood gray tone difference matrix ($NGTDM$) for an image, in which the i^{th} entry is the sum of differences between the gray level of all pixels with gray level i and the average gray levels of its neighbors.

Let $g(k,l)$ be the gray tone level of the pixel found at position (k,l) which has the gray tone value i. The average gray tone over a neighborhood centered at (k,l), but excluding it can be is denoted by $\overline{A}(k,l)$ and defined as:

$$\overline{A_i} = \overline{A}(k,l) = \frac{1}{W-1}\left[\sum_{m=-d}^{d}\sum_{n=-d}^{d} g(k+m,l+n)\right], (m,n) \neq (0,0) \tag{2.11}$$

where d stands for the neighborhood size and W is the averaging window, computed as $W = (2d+1)^2$. Afterwards, the i^{th} entry in the $NGTDM$ is computed as:

$$s(i) = \begin{cases} \sum \left| i - \overline{A_i} \right|, & \text{for } i \in N_i \text{ if } N_i \neq 0 \\ 0, & \text{otherwise} \end{cases} \tag{2.12}$$

where N_i represents the set of all pixels having gray tone level i, excluding the peripheral regions of width d.

To illustrate this, the 5×5 input image shown in Figure 2.10 is considered. Taking $d = 1$ a 3×3 neighborhood is obtained, which implies that a border of width 1 around the image exists, where the neighborhood cannot be centered (because otherwise pixels which are outside of the image will be considered).

1	1	4	3	1
3	3	0	1	1
5	3	2	2	2
2	1	1	3	4
0	2	2	5	1

(a)

i	s(i)
0	2.375
1	0.705
2	0.500
3	0.625

(b)

Figure 2.10: Sample image a) and its corresponding $NGTDM$ b).

In order to compute the third entry of the $NGTDM$ denoted by $s(i)$, the two pixels within the indicated region with a gray tone level of 2 are considered. Thus:

$$s(2) = \left|2 - \frac{14}{8}\right| + \left|2 - \frac{14}{8}\right| = 0.5$$

is obtained. In a similar fashion all the other values for $s(i)$ are obtained and shown in Figure 2.10b.

The complexity computational feature (F_{com}) is obtained as:

$$F_{com} = \sum_{i=0}^{G_h} \sum_{j=0}^{G_h} \frac{|i-j|\,(p_i s(i) + p_j s(j))}{n^2(p_i + p_j)},\ p_i \neq 0 \text{ and } p_j \neq 0 \tag{2.13}$$

For a $N \times M$ image, the probability p_i of occurrence of gray tone value i is computed as:

$$p_i = \frac{N_i}{n^2}, \text{ where } n^2 = (N - 2d) \cdot (M - 2d) \tag{2.14}$$

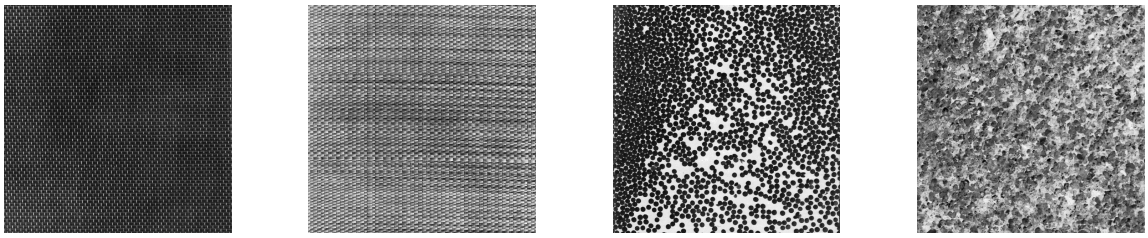
The complexity computational feature is a sum of normalized weighted differences between intensity values which are taken in pairs. The weights are represented by the sum of the probability-weighted entries in the $NGTDM$ corresponding to the two intensity values under consideration. The normalizing factor takes high values for coarse textures and small values for busy or fine textures and it is used to represent the inverse relationship between complexity and the sizes of primitives and/or probabilities of intensity values. The absolute difference between the two different intensity values is used to express the influence of contrast variations on complexity. A high value of this computational feature indicates a high degree of information content.

This feature is derived from a matrix whose elements measure actually the spatial rate of change in intensity. A big difference between the gray level of a pixel and the average gray levels of the surrounding neighboring pixels over a small neighborhood would indicate a greater spatial rate of change in intensity than for exactly the same difference, but over a larger neighborhood, yielding thus higher sensitivity to intensity spatial changes when small values of d are considered. Thus, when computing features based on the $NGTDM$ for small values of d, better results are expected to be obtained than in the case for larger d

values. In addition, a small d means less computations, making thus the computation more time efficient. This idea is also proven by the results obtained from comparing the levels of correspondence between the human ranking of the textures which have were considered in [AK89] and the corresponding computational feature. High values for the similarity between the two types of features were obtained for $d = 1$ and not $d = 2$.

2.4.2 Regularity

The human texture perception study described in [RL+93] has concluded that the three most important perceptual dimensions when discriminating natural textures are represented by periodicity, directionality and complexity. According to this study, regularity, which implies the existence of a structure which is periodic, is the strongest perceptual cue in texture discrimination.



(a) (b) (c) (d)

Figure 2.11: Example of regularity classification.
a) Highly regular. b) Regular. c) Slightly regular. d) Irregular.

Regularity indicates the variation of a placement rule. It is inferred that the variation of elements, especially in the case of natural textures, reduces its regularity. Moreover, a fine texture tends to be perceived as regular.

The analysis of the regularity of naturally occurring textures, which are difficult to describe mathematically represents a challenge without using any information related to element size or shape. The different period signals contained in the image of a periodic texture are mixed together in the time domain representation. Therefore it is hard to analyze the texture directly and this is why the frequency domain is chosen, where the periodicity of the texture can be analyzed. In the Fourier domain, there are also noise signals which disturb the periodicity of the image, such as uneven illumination or the hairiness of the texture.

The Fourier spectrum has the following properties, which are illustrated in Figure 2.12d - Figure 2.12f:

- Regular textures usually have dominant harmonic components which appear as spectral peaks following a pattern in the Fourier spectrum.

- Even if certain local inhomogeneities (e.g uneven surface or viewpoint distortion) slightly spread out or change the frequencies of the spectral peaks, the intrinsic structure of these peaks remains unchanged [LP96].
- For textures with a high degree of directionality, the directionality will be preserved in the Fourier domain, where the high spectral values will lie in a line in a direction which is perpendicular to the direction of texture pattern.
- For irregular textures, the spectral values are not concentrated in any direction.

In [LC05] it is shown that a texture with a well-defined directionality, even if a perceivable micropattern is absent, is considered more regular than a pattern that has no periodicity and directionality, even if the individual micropatterns can be identified. The four scales of regularity are shown in Figure 2.11, where Figure 2.11b is perceived as being more regular than Figure 2.11c, even if in the latter there are clear circular patterns. This emphasizes that periodicity and directionality are two equally important factors when determining regularity.

To compute the regularity of textures, as suggested in [LC05], the 2D Fourier transform is firstly performed on the image I of the texture, obtaining the Fourier spectrum F which is smoothed afterwards with a 2D square Gaussian kernel with a standard deviation $\sigma = 0.5$, obtaining F_{smooth}. Afterwards, F_{smooth} is treated as a new image and the Fourier transform is employed again to produce a so called *enhanced Fourier spectrum*, denoted by E.

The Fourier spectrum of highly regular textures consists of prominent peaks which are regularly scattered in certain directions, as shown in Figure 2.12d, whereas for textures with low regularity, the spectral values are not concentrated on specific directions, as illustrated in Figure 2.12f. As suggested in [LC05], by applying the Fourier transform on the Fourier spectrum, an *enhanced Fourier spectrum* is obtained, where these properties can be further enhanced, as shown in Figure 2.12g - Figure 2.12i. In Figure 2.12d the Fourier spectrum of Figure 2.12a is shown, where the spectral peaks originate from the contribution of those pixels with the same period in the original image, which are spread out regularly along certain directions. This pattern is made more visible in the enhanced Fourier spectrum. The reason for this is that those peaks in the Fourier spectrum which are periodic, as shown in Figure 2.12g, contribute to the same frequency when the Fourier transform is applied again, making the peaks in the enhanced Fourier spectrum more visible. On the other hand, for those remaining pixels in the Fourier spectrum which are not peaks, since they are not periodic, by applying the Fourier transform again, they do not contribute to the same frequency and therefore they do not yield prominent peaks.

The Radial Wedge Distribution Variance ($RWDV$) proposed in [LC+02] is extracted from the enhanced Fourier spectrum, called $E(u, v)$, which is divided in a number of wedges as shown in Figure 2.13. The radial wedges are denoted by RW_i, with $i = 1, \ldots, m$, and $m = 360/\Delta_\theta$, where Δ_θ is the size of each wedge. In the implementation of $RWDV$ a wedge size of $\Delta_\theta = 1^\circ$ is considered. Each value $E(u, v)$ can be accumulated to the corresponding RW_i. To do this, the angle θ_i formed by the current pixel with the origin is

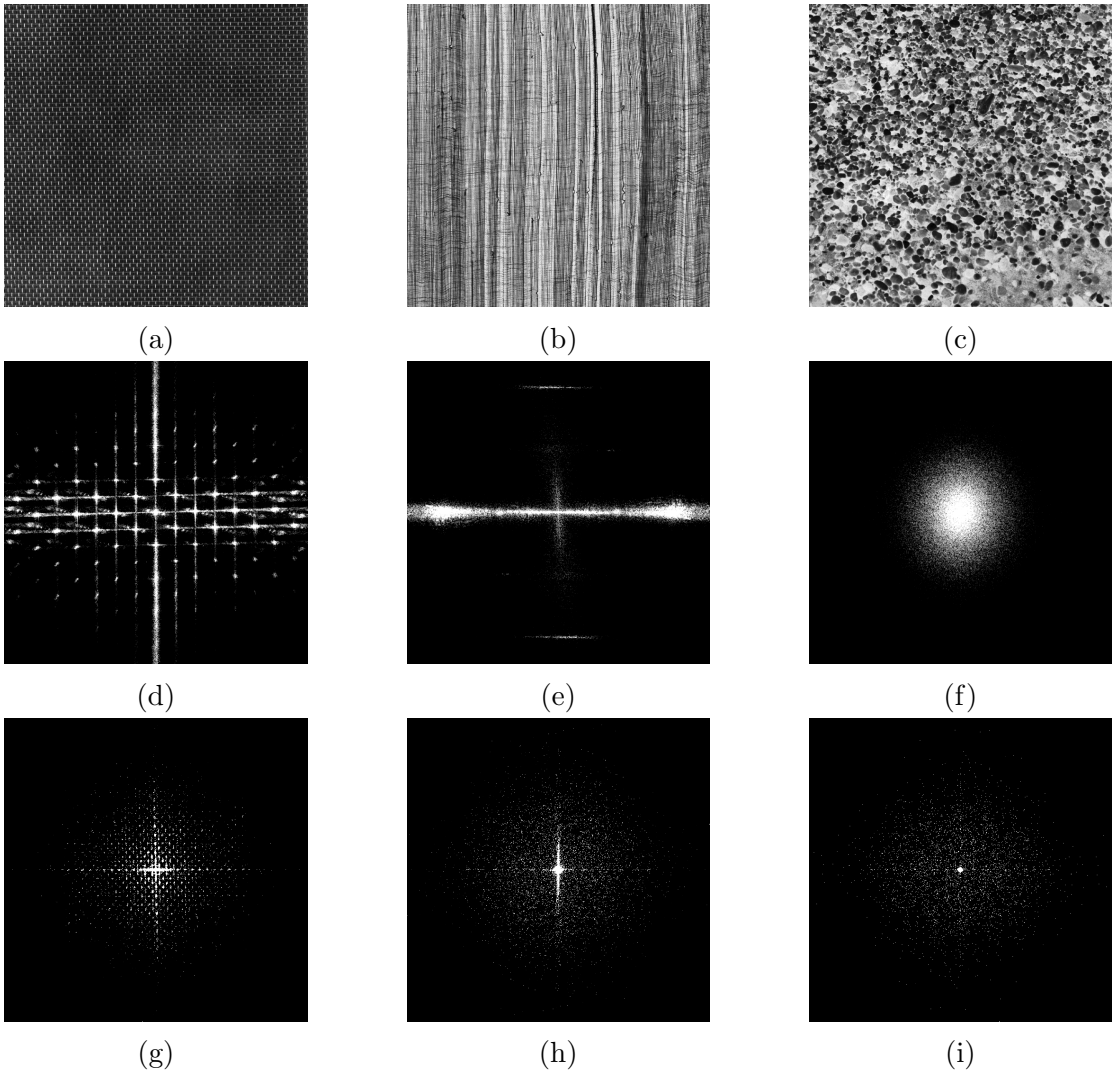


Figure 2.12: Examples of Fourier and enhanced Fourier spectra for regular, directional and irregular textures. a) Regular texture - $D6$. b) Directional texture - $D106$. c) Irregular texture - $D54$. d) Fourier spectrum of $D6$. e) Fourier spectrum of $D106$. f) Fourier spectrum of $D54$. g) Enhanced Fourier spectrum of $D6$. h) Enhanced Fourier spectrum of $D106$. i) Enhanced Fourier spectrum of $D54$.

computed as:

$$\theta_i = atan\left(\frac{u - y_c}{v - x_c}\right) \tag{2.15}$$

where y_c and x_c are the coordinates of the center of the image, which also acts as the origin of the axes used to measure the angles. Afterwards, the quadrant the current pixel belongs to is determined by comparing the position of the (u, v) pair with respect to the (y_c, x_c) pair and the angle which was previously computed is adjusted by taking into consideration the quadrant it belongs to:

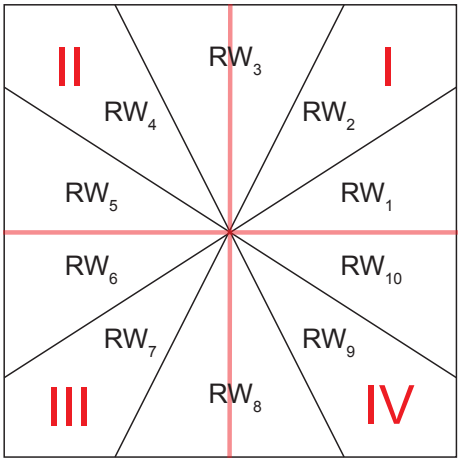


Figure 2.13: Radial wedges.

- **First quadrant**: $\theta_i = |\theta_i|$;
- **Second quadrant**: $\theta_i = 180° - \theta_i$;
- **Third quadrant**: $\theta_i = 180° + |\theta_i|$;
- **Fourth quadrant**: $\theta_i = 360° - \theta_i$;

In the last step $E(u, v)$ is accumulated in the corresponding RW_i, where the position i is computed as $i = floor(\theta_i) + 1$.

The energy of each wedge is normalized by the total energy of all wedges. Considering ERW_i, $i = 1, \ldots, m$ to be the normalized energy of the radial wedge RW_i, then the $RWDV$ is defined as:

$$RWDV = \frac{1}{m}\sum_{i=1}^{m}(ERW_i - \overline{ERW})^2, \text{ where } \overline{ERW} = \frac{1}{m}\sum_{i=1}^{m} ERW_i \tag{2.16}$$

In the case of highly regular textures, because the spectral peaks spread out regularly along certain directions, the variance of all radial wedge energies will be larger than that of a low regularity texture, allowing the $RWDV$ to be used to measure textural regularity. A large value of $RWDV$ indicates textures with highly structured periodic patterns, while small values indicate irregular or random textures.

2.5 Shortcomings

The experimental database which was considered by Tamura et al. consists of 16 textures with an image resolution of 256×256 pixels. These images were quantized to 64 gray levels. Thus, the major shortcoming of all features which are defined based on this database is that the resolution of the images is very small, namely a few orders of magnitude lower than the 8 MP images considered in this work. In addition, the reduced number of gray levels (64 instead of 256) can indicate that the results which were obtained at the end of the study are not generic enough. Moreover, certain parameters are considered in the algorithms, e.g. the distance d between two pixels for which the $GLCM$ is computed, which is then used to determine the linelikeness of the texture. However, these hard coded values are resolution-dependent, meaning that for increased resolutions of the considered images higher values for these parameters will have to be considered to correctly capture the desired feature.

A number of features, which were presented until now, do not provide a high interclass variance when applied to the textures in the considered database, as is the case for the directionality and regularity features. This is not desirable, because it makes the features hardly usable for a machine learning approach. This is the main motivation why improvements of these features are suggested, such that their discriminating power is maximized.

Nonetheless, the values which were obtained when considering the image database described in Chapter 3 for the directionality and regularity features for example do not correspond to the human perception. This is another reason to obtain an improvement of these features, such that haptically-relevant features are obtained.

Chapter 3

Recording Procedure

3.1 Original Database

To create a procedure which matches the human perception, a database of 69 textures is considered, as shown in Figure 3.1. This collection contains a large variety of natural textures, including many inhomogenous ones which are not normally included in texture studies.

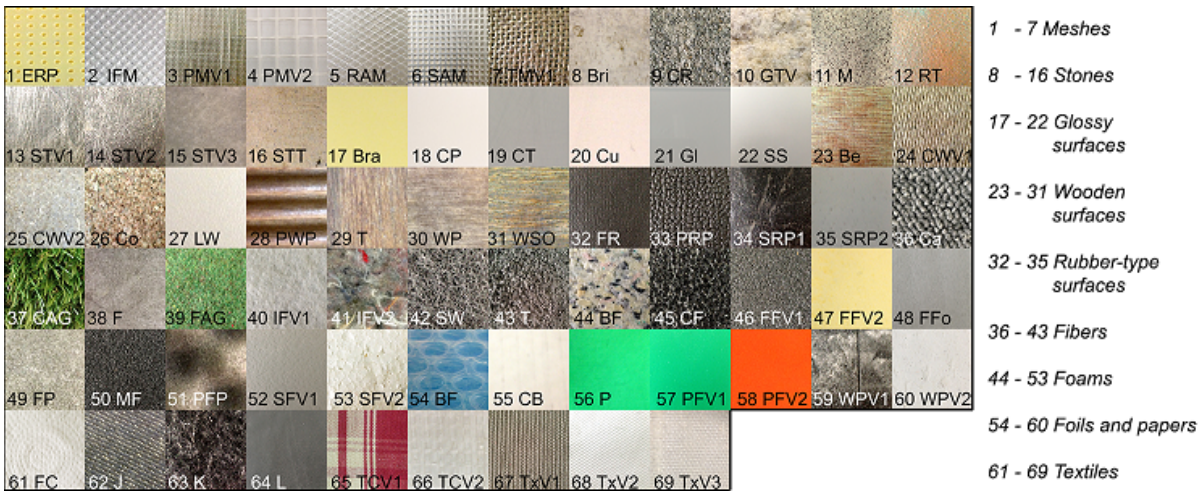


Figure 3.1: Materials included in the haptic surface database, which are freely accessible at *http* : *//www.lmt.ei.tum.de/texture/*. The surface names are shown in Appendix A in Table A.3.

Ten images per texture were captured by using the rear camera of a common smartphone (Samsung S4 Mini), with a resolution of 8 MP. The illumination and capturing conditions (distance, viewing angle) vary for different images of the same surface. For each surface five images under daylight and five images under ambient light conditions were captured,

Figure 3.2: A texture under different angles and illumination conditions.

using also the smartphone flashlight to increase the range of possible illumination conditions. Viewing directions and camera distances were chosen arbitrarily for each picture, resulting in substantial variations within each individual class, as it is shown in Figure 3.2. While these images carry important information about the surface, differences in distance, rotation, lighting and focus conditions complicate the extraction of robust image-based features for the surface classification task. This also models the real life situations, where illumination or focus are not ideal.

3.2 Magnified Database

The major drawback when taking pictures from afar is that fine details, e.g. the intrinsic periodicity within a woven fabric or the fine hairs that furs exhibit, cannot be captured by the camera, despite the high resolution. To counteract this and be able to extract more information out of images, a novel approach is considered, that makes use of an affordable and easy to use magnifying glass, as it is shown in Figure 3.3. This can be simply attached to the camera of the smartphone to be able to capture photos with an increased level of detail. To ensure that the developed algorithms are able to deal with a wide range of image qualities offered by different devices, another smartphone (an Iphone 6) was chosen

(a) (b)

Figure 3.3: Setup for capturing magnified images. a) Magnifying glass with clip. b) The complete setup, with the magnifying glass attached to the camera of the smartphone.

for taking the magnified version of the images, which the same sensor resolution of 8MP.

Ten images were taken per texture, all under ambient illumination conditions, as shown in Figure 3.4. The flashlight could not be used in this scenario, because the images were all blurred and very bright when using it. The distance was kept constant for all pictures to be able to obtain a well focused image and different viewing angles were considered. All images which were taken with the magnifying glass attached to the camera of the smartphone have blurred and unfocused edges, which makes only the central part of the image contain useful texture information.

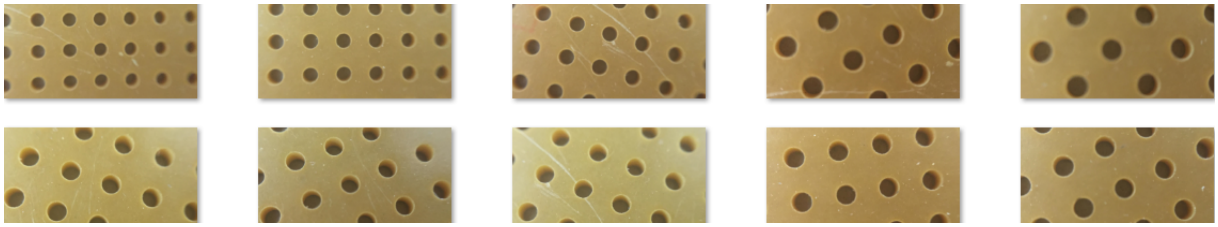


Figure 3.4: A magnified texture under different angles and illumination conditions.

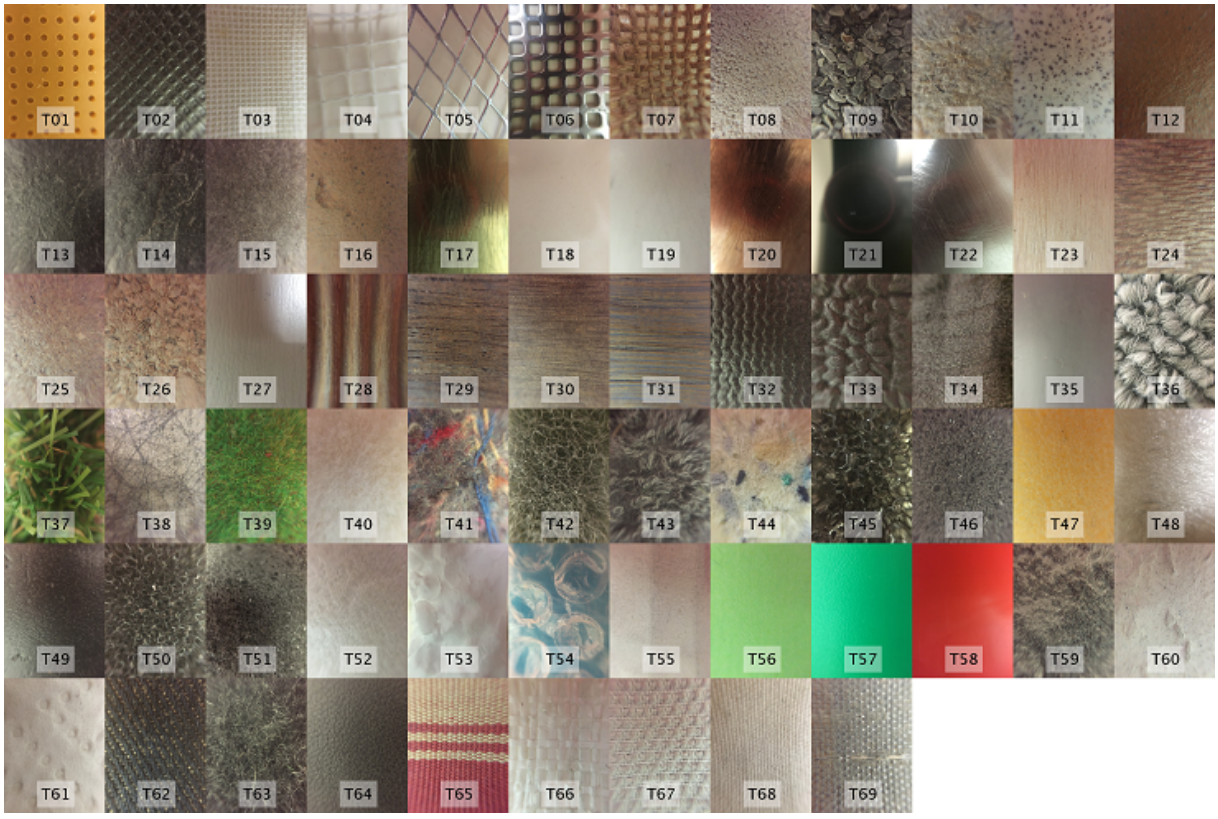


Figure 3.5: Images of textures taken with the magnifying glass attached to the smartphone.

Chapter 4

Haptically Related Textural Features

4.1 Improved Features

In this chapter, a series of improvements are presented with respect to the original features described in Chapter 2. The features are then tested on the surface texture database, which is described in greater detail in Chapter 3. The surface names are shown in Appendix A in Table A.3.

Boxplots are used to graphically represent the patterns of the considered data and can be employed to compare the behavior of the features on the proposed database. The body of the boxplot consists of a box which is drawn from the first quartile to the third one. Within the box, a line is drawn at the median of the data set. Two vertical lines, which represent the whiskers, extend from the first and third quartile to the smallest and largest non-outlier, respectively. Outliers are extreme values which differ significantly from other values within a dataset. A value is considered to be extreme, if it is at least 1.5 interquartile ranges (IQR) below the first quartile or above the third quartile. The IQR is a measure of variability, which is defined as the difference between the third and the first quartile.

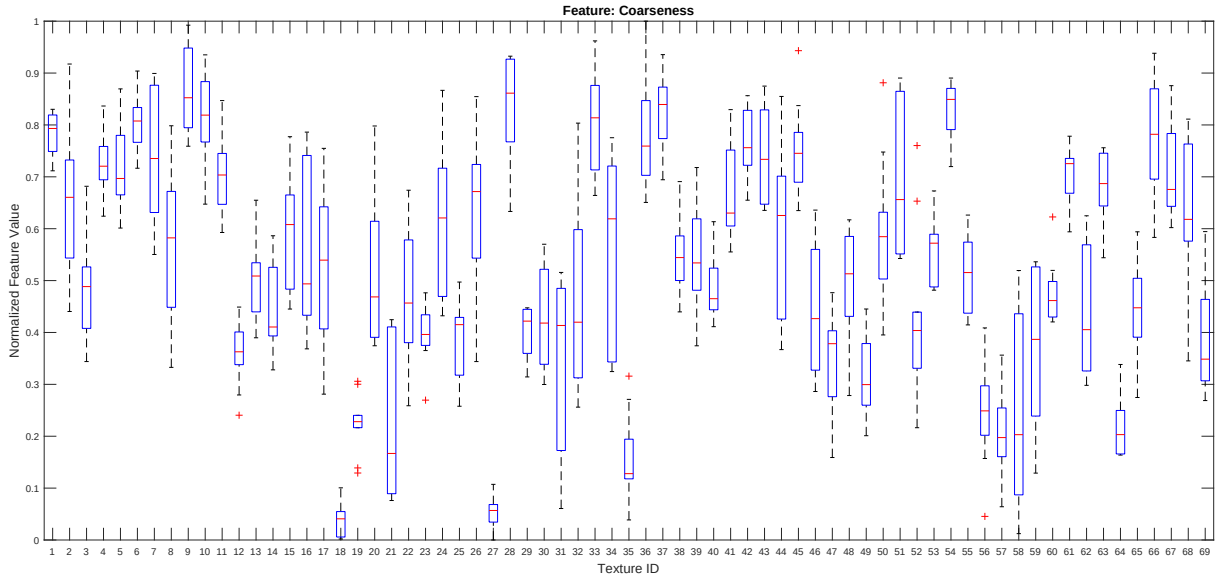
The computations were performed using MEX files created in Matlab R2015b and a laptop with an Intel Core i7(3rd Gen) 3612QM CPU with a clock frequency of 3GHz and 8 GB of RAM.

4.1.1 Coarseness

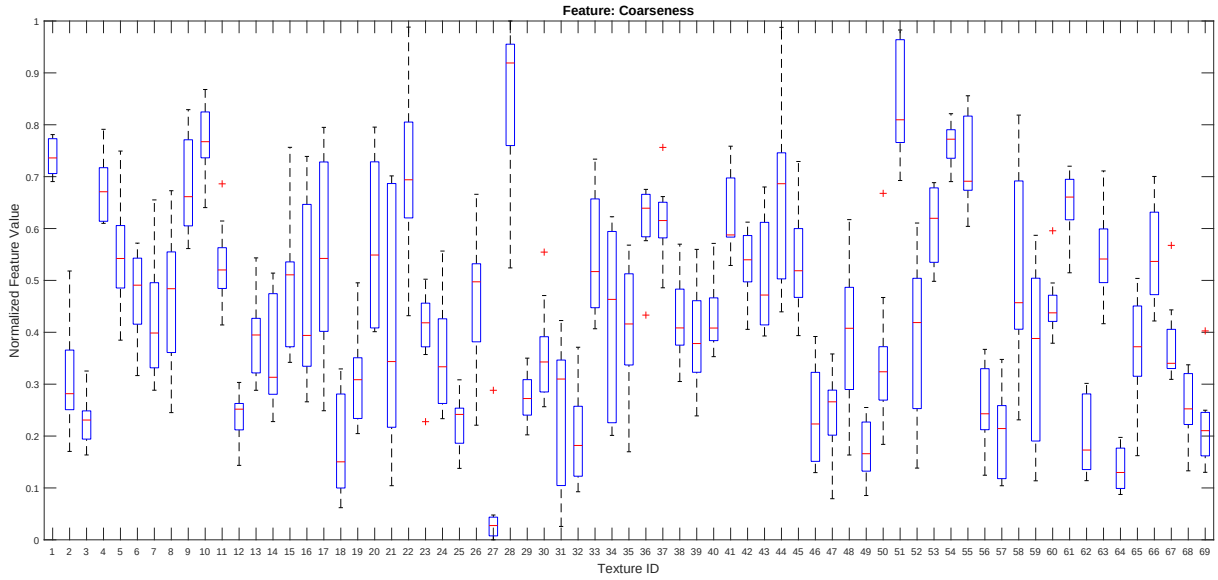
To detect coarseness patterns with a higher degree of perceptual correspondence than in the initial implementation described in [TMY78], a modified version of it is considered.

The original search range of $k \in \{1, 2, 3, 4, 5\}$ needs to be increased to adapt the problem to the current scenario of high resolution images. However, with $k > 7$, the computational

time becomes prohibitively large, as the number of computations which need to be performed increases by a factor of four when the search range is doubled. Thus, to obtain a trade off between the processing time and perceptual accuracy, the range $k \in \{1, 2, 3, 4, 5, 6\}$ is considered to compute the AGL.



(a)



(b)

Figure 4.1: The coarseness feature as suggested by Tamura et al. a) and considering the suggested improvements b). In terms of intraclass variance, represented by the height of the boxes, and interclass variance, represented by the amount of overlapping among boxes, improvements are obtained.

Another improvement which was implemented concerns the directions in which the symmetric subtraction of average gray levels is undergone. Initially, only the vertical and horizontal directions are considered. In addition, the diagonal directions are also added (left and right diagonal), to be able to detect texture elements which have also other orientations and to make this feature less rotation-dependent.

When considering the original coarseness feature applied to the considered image database, most of the textures are interpreted by the algorithm as having a medium to high degree of coarseness. This is suggested by the observation that most feature values lie in the range $[0.4, 1]$, a coarseness value of 1 indicating a very coarse surface. As it can be deducted from the boxplot in Figure 4.1a, the interclass variance is not as high as in Figure 4.1b, with a considerably high degree of overlapping in the former case. This is not desired, since a machine learning approach needs radically different features for different textures to be able to identify new data.

The boxplot in Figure 4.1b corresponds to the enhanced feature that has an extended search range in comparison to the one suggested in [TMY78] and also additional search directions for patterns (using a 45° resolution instead of a 90° one). It is shown that a greater degree of interclass variance among all textures exists. However, this comes at the expense of a slightly increased intraclass variance, the height of the boxes in Figure 4.1b being on average slightly higher than in Figure 4.1a. With the new approach, textures like crushed rock, profiled wood plate, profiled foam and bubble foil are correctly identified as the coarsest textures, whereas surfaces like laminated wood, ceramic plate and fine rubber are classified among the least coarse ones.

This indicates that a larger search range and a finer quantization for the angles for which patterns are searched for needs to be adopted to correctly capture the coarseness properties of a texture. However, this comes at the expense of increased computational complexity, which is primarily influenced by the search range, where an increase by a factor of two causes the computational time to become four times longer.

4.1.2 Directionality

One major drawback of the directionality feature suggested in [TMY78] is that it does not consider the contribution of each individual peak in the directional histogram. Thus, results which are not in accordance to the human perception are obtained. To illustrate this, Figure 4.2 shows the textures which were used as a reference for the algorithm and their corresponding directional histogram H_D are shown. In Figure 4.2g it is shown that *Cloth* is highly directional, with two major peaks of the directional histogram, one being located at the angle $\pi/16$ radians, and the other one at $15\pi/16$ radians, which represents the same angle. On the other hand, from Figure 4.2i, it follows that *Ground* has a relatively weaker directionality in comparison with *Cloth*, because its peak is not so sharp and does not have such a high amplitude. The histogram of *Sawdust* shown in Figure 4.2j is nearly

flat, implying that this structure has no directionality. It can be said that these descriptions concerning the three textures are perceptually relevant. Thus, H_D can reflect directionality very faithfully.

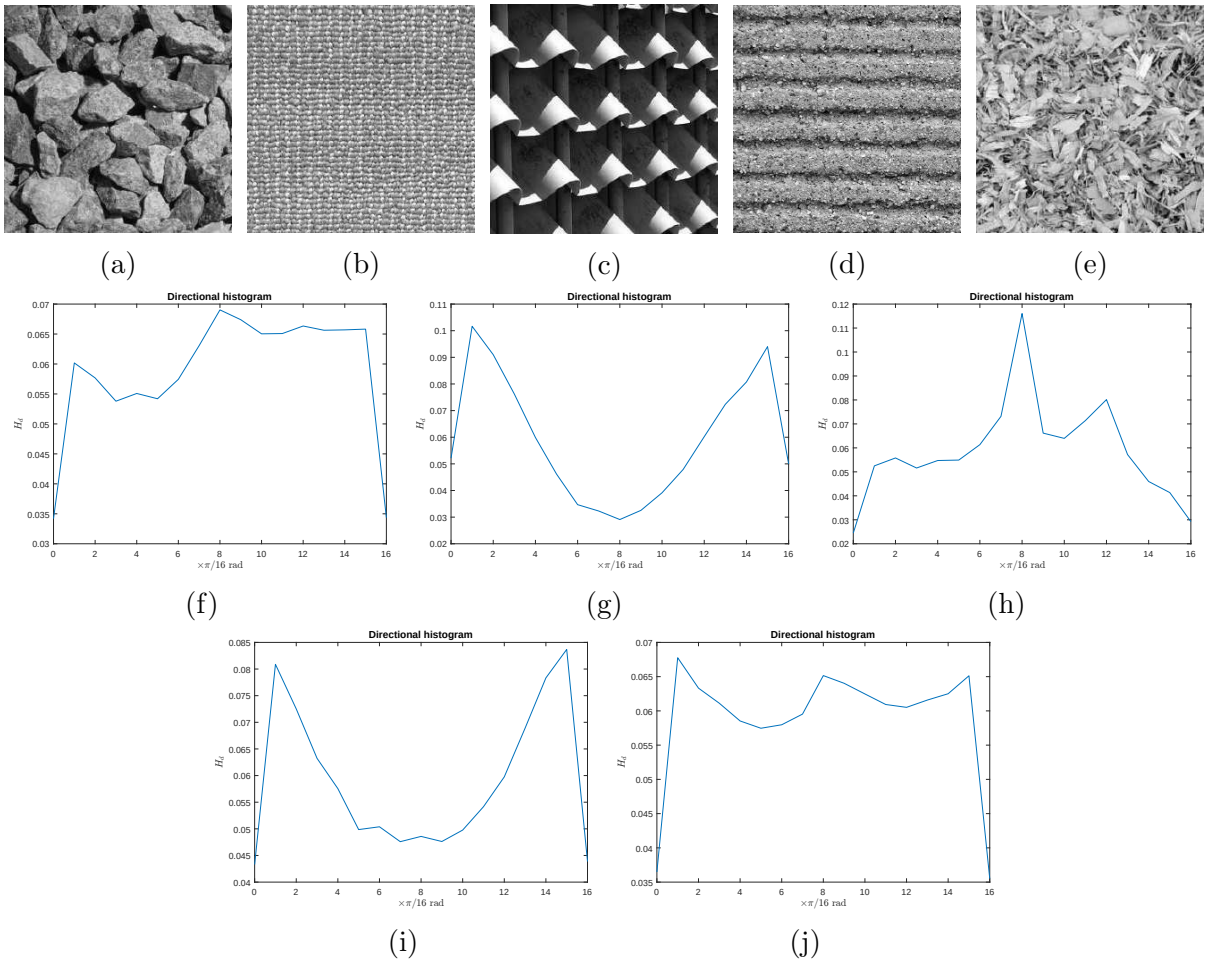


Figure 4.2: Different textures and their corresponding directional histograms [MS12]. a) Rocks. b) Cloth. c) Roof. d) Ground. e) Sawdust. f) H_D of Rocks. g) H_D of Cloth. h) H_D of Roof. i) H_D of Ground. j) H_D of Sawdust.

The results which were obtained after applying the algorithm suggested in [TMY78] on the previously mentioned images are shown in the first column of Table 4.1.

Upon inspection of these results, *Cloth* is considered to be the most directional texture among the ones considered, whereas *Ground* is the second most directional, taking precedence over *Roof*, which is not perceptually intuitive. The directional histogram of *Ground* shows two spikes, which have a depth of approximately 0.03 and an absolute value of 0.08, whereas *Roof*, which is perceptually more directional than *Ground*, has also two spikes, one with a depth of 0.05 and an amplitude of almost 0.12, and the second one with a depth of 0.02 and an amplitude of 0.08. Considering these values and the final value of the direc-

Texture	Directionality (Tamura)	Directionality (improved)
Rocks	0.0427	0.0793
Cloth	0.1997	0.1414
Roof	0.1272	0.1963
Ground	0.1835	0.1366
Sawdust	0.0690	0.0799

Table 4.1: Directionality feature computed according to Tamura et al. and the improved version which is more perceptually relevant.

tionality feature for the two textures, it can be concluded that the algorithm favorizes the amplitude of the peaks more than their depth, which appears to be different from human perception of texture directionality. In [HZLL05] it is evidenced that this method fails to capture directionality properly, assigning higher directional score to random images than real directional images.

After testing the algorithm on the surface texture database, it follows that the directionality values tend to fall within the same range for a high number of textures. This translates to a very low interclass variance, which is not desirable when trying to conduct a machine learning approach. Therefore, to obtain a directionality feature which better matches the human perception and is characterized by a higher discriminating power than in the considered case, a new approach has to be taken.

In the paper [IZL08] a new method is proposed to compute the directionality of an image by making use of the geometric property of the directional histogram, unlike the classical Tamura et al. method which uses the statistical property of it. This method was proven to outperform the conventional method, both in subjective and objective analyses, having also a superior retrieval performance.

Initially, the directional histogram which was previously obtained is considered. This is made up of 16 bins, which correspond to the angles from [0,π]. Each pair of bins is joined, obtaining a line segment, whose slope s_k is computed as:

$$s_k = \frac{H_{D_{k+1}} - H_{D_k}}{\phi_{k+1} - \phi_k} \tag{4.1}$$

In the next step, the number n of peaks within the histogram is determined, together with the number of segments k_n which help form the local maximum. Then, for each segment belonging to a local maximum its weight $w_{l_{k_n}}$ is computed as follows:

$$w_{l_{k_n}} = 2^{-\left|p_{l_{k_n}} - p_n\right|+1} \tag{4.2}$$

where $p_{l_{k_n}}$ is the position of the current segment and p_n is the peak of the n^{th} local maximum whose segments is considered. In this way, it is ensured that the weights diminish as the distance towards the valley of a local maximum increases. The sharpness S_n of the n^{th}

local maxima is then computed as the weighted sum of slopes of all line segments joining the bin summits:

$$S_n = \sum_{i=1}^{l_{k_n}} w_{l_{k_n}} \cdot s_{l_{k_n}} \tag{4.3}$$

Considering these, the directionality feature (ID) is computed as the weighted sum of all sharpness values from all the local maxima in the histogram, multiplied by the number of peaks n:

$$ID = n \sum_{i=1}^{n} W_i \cdot S_i \tag{4.4}$$

where the weight W_i for the i^{th} local maxima, whose height is denoted by H_{p_i}, is computed as follows:

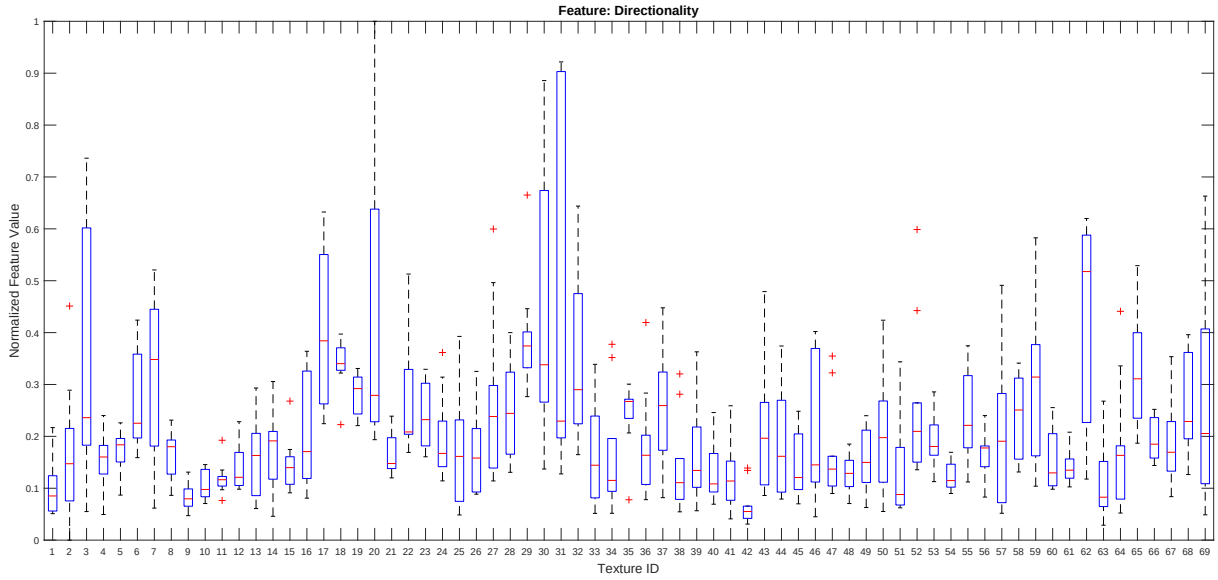
$$W_i = \frac{H_{p_i}}{\sum_{i=1}^{n} H_{p_i}} \tag{4.5}$$

In this formula, all bins including the ones corresponding to the peaks contribute to the sharpness value. Through the weighting scheme which is implemented, the slope of a line segment near the peak has a greater contribution to the overall directionality than the slope of a line segment near the bottom of the local maximum. The results which are obtained by applying the new algorithm to the same set of pictures as before can be seen in the second column of Table 4.1.

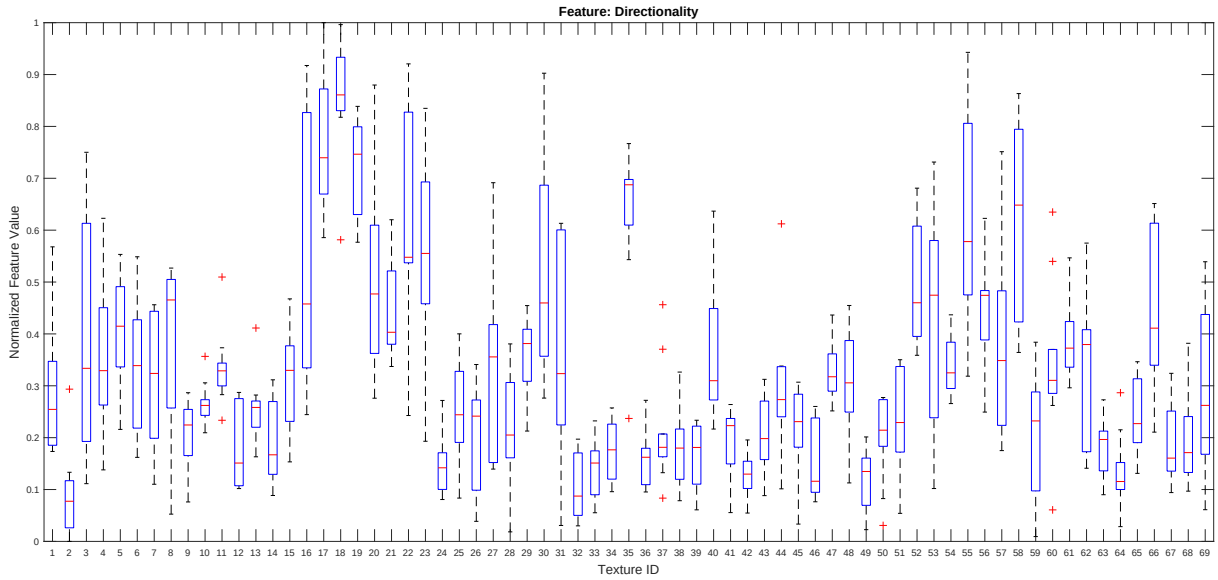
Considering these new values, the most directional of the textures is *Roof*, followed by *Cloth* and *Ground*, and only the *Rocks* and *Sawdust* have very close directionality values which can make it difficult to differentiate them. However, this is acceptable, since they are both nondirectional.

In Figure 4.3a, it is shown that most of the textures have directionality values which span roughly the same range, which is undesirable, especially from a machine learning point of view. What also stands out from the boxplot is that there are very high discrepancies among textures which have similar degrees of directionality, e.g. the two versions of plastic meshes. Even if perceptually the first version is more directional than the second one, as it has a larger number of lines in the same direction due to its smaller mesh size, the two value ranges are not be so distinct from one another, as shown in Figure 4.3a. When considering the same two textures in Figure 4.3b, it follows that the first version has indeed a higher value for this feature, and the second version has a slightly smaller value, which is perceptually accurate. In both cases, surfaces like brass, copper, stainless steel score high values for this feature, even if they are not perceptually linelike. The reason for this can be due to the scratches these surfaces exhibit, which are interpreted as lines by the algorithm. Excluding these textures, the highest values for the directionality feature are obtained for textiles and meshes, which is in correspondence with what the humans perceive.

Considering the the goodness of features criterion (GFC) values, which is described in Section 6.1.3, a major improvement is obtained, since initially $GFC = 0.1705\%$ and after



(a)



(b)

Figure 4.3: The directionality feature as suggested by Tamura et al. a) and with the presented improvements b). Even if the intraclass variance does not change significantly, visible improvement in terms of interclass variance can be seen, the data having a more uniform distribution in the interval $[0, 1]$.

the improvements the $GFC = 0.8951\%$. The latter value is still small, meaning that there is still a considerable amount of interclass variance. However, this result is much better than the initial one, which is also shown in the boxplots in Figure 4.3.

4.1.3 Linelikeness

In [TMY78], when computing the linelikeness feature, only distance $d = 4$ is used for building the corresponding $GLCMs$ upon which algorithm is based. By doing this, the algorithm is restricted to detect lines of a certain size. Another drawback for selecting a fixed d value is that the pixels which fall between the ends of the pixel pair of interest are not considered, because it is assumed that no sudden changes between the pixel values of neighboring pixels exist.

Texture	**D28**	**D67**	**D109**	**D111**
Linelikeness	1.0992	2.3480	3.0317	4.1202

Table 4.2: The linelikeness feature values for textures $D28$, $D67$, $D109$, $D111$ from the Brodatz database.

Instead of selecting a certain value for d, a range of values for the distance d is more appropriate. Therefore, the feature is computed for a range of distances, considering $d \in \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$. The linelikeness feature (IL) is obtained by summing up the corresponding values:

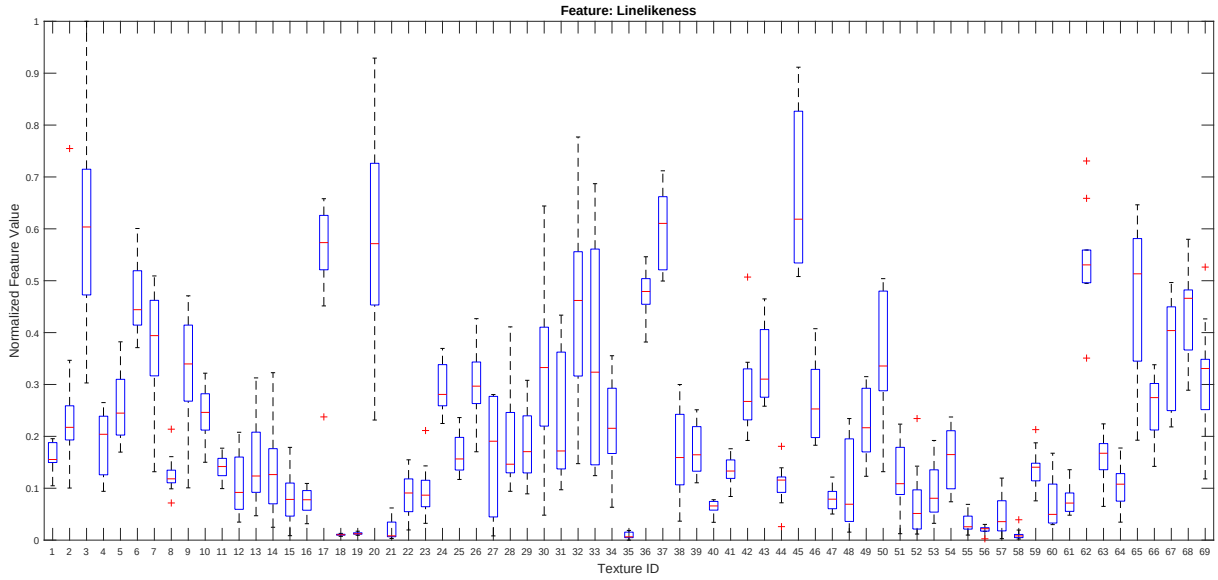
$$IL = \sum_{k=1}^{8} \sum_{d=1}^{10} \frac{\sum_{i}^{n} \sum_{j}^{n} P_d(i,j,k) \cdot cos(\frac{(i-j)2\pi}{n})}{\sum_{i}^{n} \sum_{j}^{n} P_d(i,j,k)} \tag{4.6}$$

where n is the width of the P_d matrix. The reason behind choosing to sum the linelikeness values corresponding to the eight directions instead of choosing the largest one, is because a measure of the linelike patterns which occur in an image regardless of the direction these patterns appear in is desired to be obtained. Therefore, the more linelike patterns occur in an image, without any particular orientation, the bigger the sum gets and in the end, the larger the value of IL.

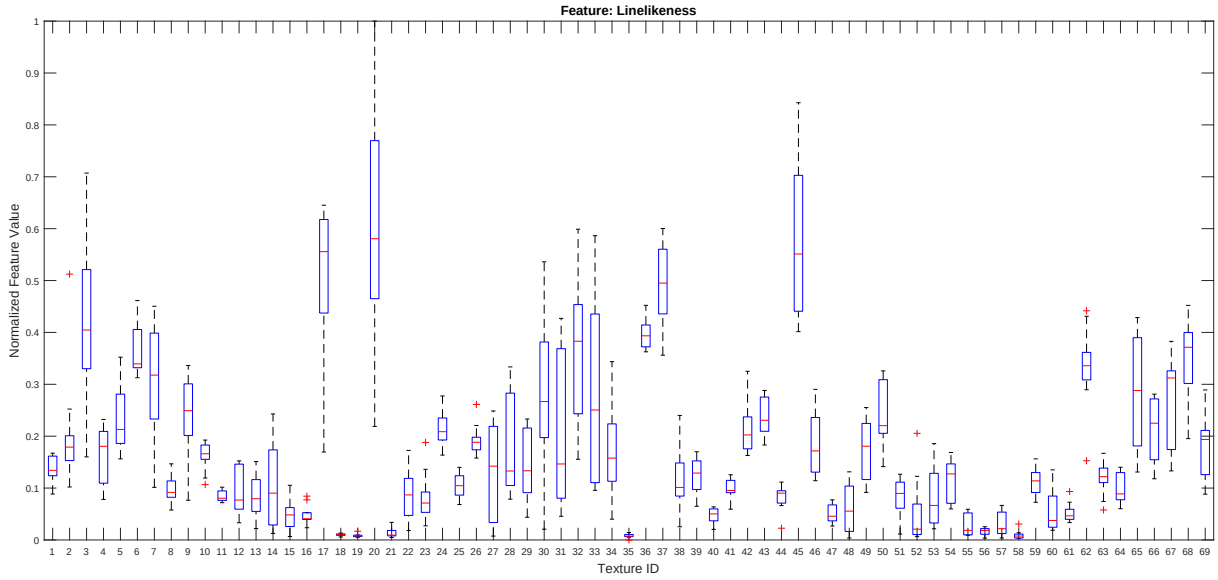
By using this range, consistent and coherent line patterns inside the image are clearly searched for, being able to detect the lines of different sizes which can be found within an image. This, however, comes at the expense of increased computational complexity.

The proposed algorithm is tested on the four images presented in Figure 2.7, obtaining the results which are shown in Table 4.2. The results are in perfect accordance with what the humans perceive, since textures $D109$ and $D111$ have much greater values for the linelikeness feature as $D28$ and $D67$.

The first observation concerning the two boxplots in Figure 4.4 is that they look very much alike. However, differences also arise. For example, the intraclass variance of the improved version of this feature is smaller than the one corresponding to the linelikeness implemented as described in [TMY78]. This observation can be drawn from the two boxplots presented



(a)



(b)

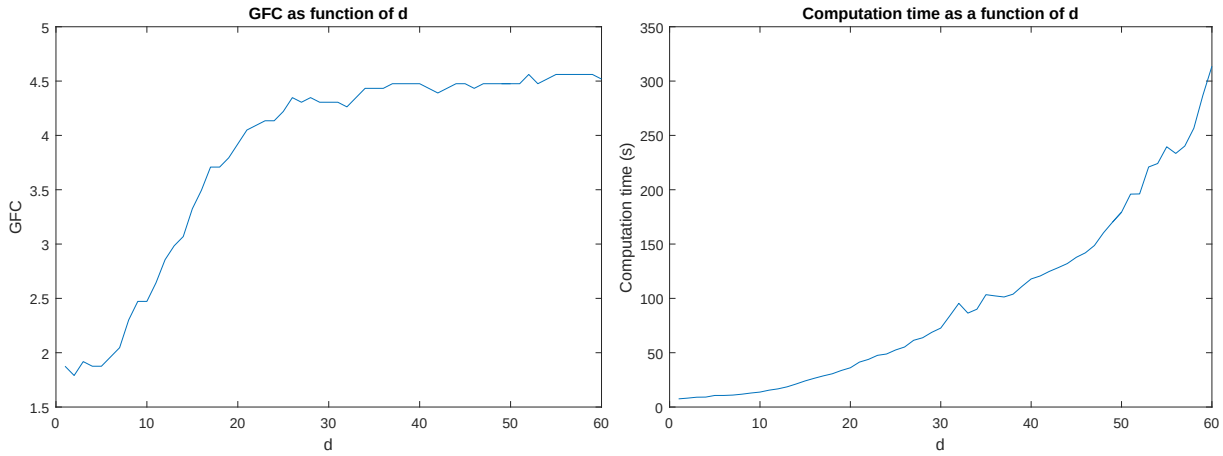
Figure 4.4: The linelikeness feature as suggested by Tamura et al. a) and with the implemented improvements b). The interclass variance remains basically unchanged, whereas the intraclass variance slightly decreases.

in Figure 4.4, where the height of the boxes in Figure 4.4b is on average lower than for the ones in Figure 4.4a. The interclass variance is very similar for the two versions. These observations are supported by the values GFC metric values. For the Tamura et al. version of the algorithm, $GFC = 3.2822\%$, whereas for the improved version, $GFC = 3.3248\%$. The most prominent outliers in both implementations are the brass and copper textures,

which according to the boxplots score the highest linelikeness values, due to the scratches which are to be found on their surface. Excluding these outliers, the most linelike textures in the database are correctly identified as being artificial grass fibers and plastic meshes, which is perceptually accurate.

4.1.4 Complexity

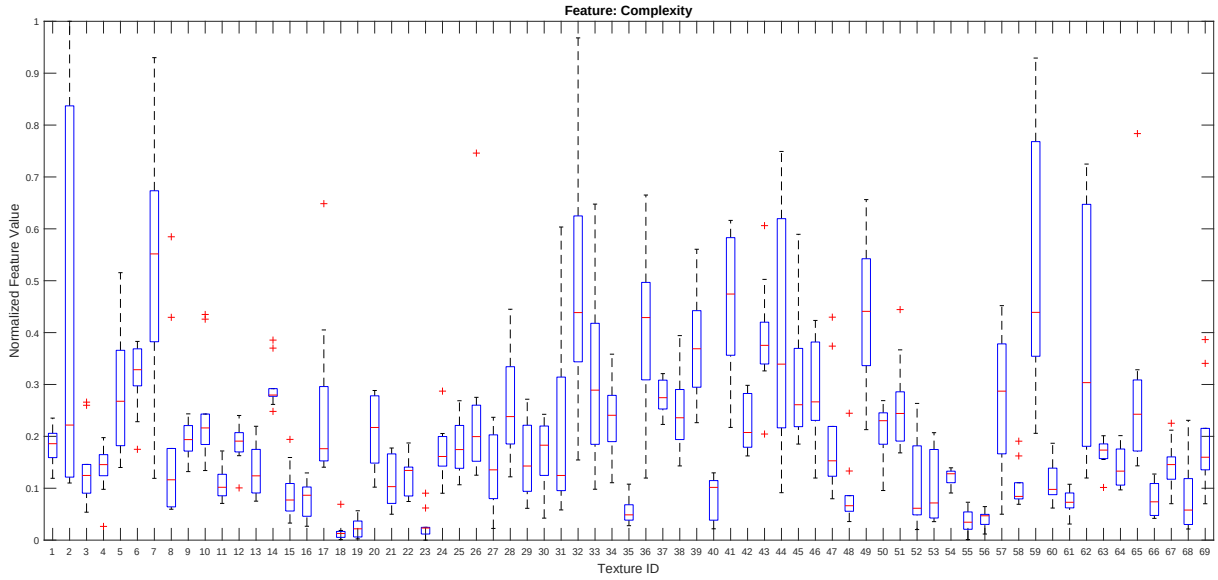
One major drawback of the method described in [AK89] is that the fixed $d = 1$ value is specific to the resolution of the images which were considered in the experiments, namely 384×384. This resolution is a few orders of magnitude less than the one corresponding to the images in the database on which the defined features were tested. Thus, the search range which will be obtained with the previously mentioned d value will simply be too narrow to capture relevant texture characteristics.



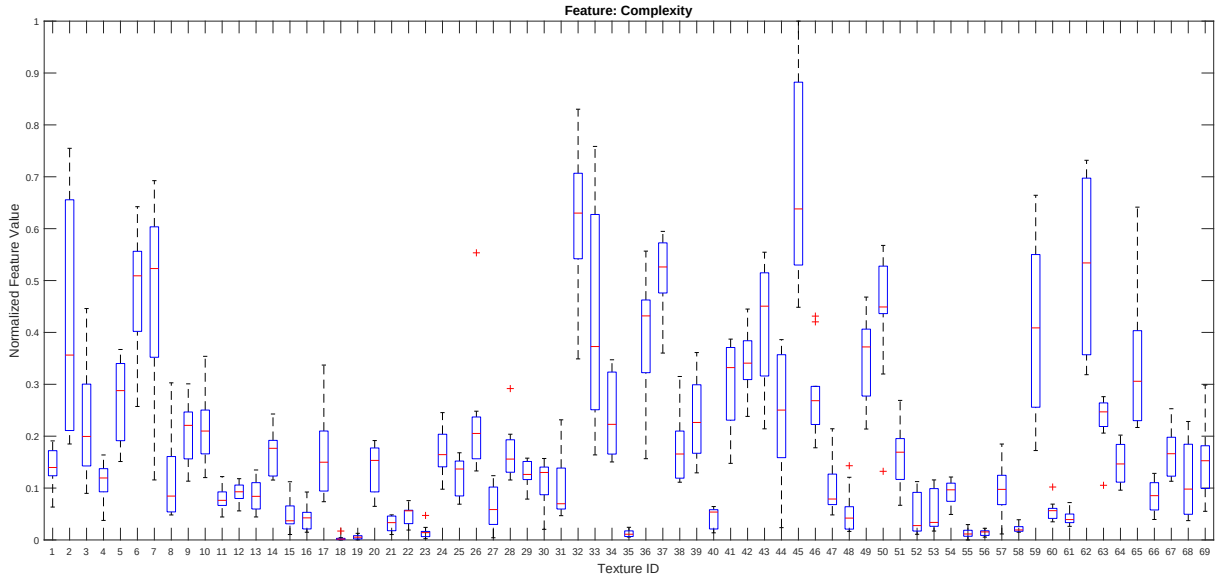
(a) (b)

Figure 4.5: a) GFC as a function of the range d. b) Computation time per image as a function of the range d.

To find the distance value that yields results that best match the human perception, the complexity feature was computed for $d \leq 60$. In Figure 4.5a the GFC values as a function of d is shown. The GFC values tend to increase as d increases. A large value of the GFC shows that the considered values are significantly different from one another, which is one strong desiderate from a machine learning point of view. The average computation time as a function of d is shown in Figure 4.5b. As expected, due to the increased computational complexity determined by a large d, the time required to compute the complexity feature increases exponentially with d. As it is shown in Figure 4.5a, the GFC has the tendency to increase with d until a local maximum at $d = 26$ is achieved. Afterwards, the GFC values slightly decrease as the search range further increases, and starting with $d = 32$ a plateau of the GFC values is achieved. However, in terms of the GFC metric, the increase



(a)



(b)

Figure 4.6: The complexity feature for $d = 1$ a) and for $d = 26$ b). The latter is more perceptually-relevant and has better discriminative power than the former.

is negligible after $d = 26$, whereas the average computation time spent per image increases by a factor of 6. Thus, $d = 26$ offers a good trade off between computation time and the discriminative power of the considered feature.

In Figure 4.6, the boxplots corresponding to $d = 1$, as suggested in [AK89], and $d = 26$ are shown. The latter offers much more interclass variance than the former, whereas

the intraclass variance is similar for the two cases. These observations are in complete agreement with the feature quality metric defined in Section 6.1.3, since for $d = 1$ a $GFC = 1.8755\%$ is obtained, while for $d = 26$ a $GFC = 4.348\%$ is obtained. Also, the boxplot in Figure 4.6b is highly correlated with the human perception, since the most complex textures which are identified are meshes, grass fibers, foams and jeans, whereas the least complex ones are ceramic plates, plastic foils, paper and cardboard. The algorithm correctly identifies these textures whose information content is high, e.g. with a high number of lines or edges, as the most complex and the more uniform ones as not complex.

4.1.5 Regularity

The main drawback of the RWDV algorithm proposed in [LC+02] is that the interclass variance of the feature values is very low, which makes a machine learning approach not feasible in this case. The values which are obtained are not correlated with what the humans perceive. Thus, a new approach is suggested to make the regularity feature be more perceptually relevant.

To determine the harmonic structure of an image, the frequencies and the magnitudes of the harmonic spectral peaks need to be determined. To do this, the image is firstly filtered using a bilinear filter, which is a nonlinear, edge-preserving and noise-reducing smoothing filter, which replaces the intensity value of each pixel by a weighted average of intensity values from the neighboring pixels. As suggested in [LSAR10], a bilinear filter with a half-width $w = 5$, standard deviations $\sigma_1 = 3$ and $\sigma_2 = 30$ and a tolerance of $t = 0.01$ is used. In the next step, histogram equalization is performed to improve contrast. Afterwards, the Fourier spectrum of the filtered image is computed. The texture image and the corresponding Fourier transform are represented by $I(x, y)$ and $F(u, v)$, where (x, y) are the coordinate variables in time domain and (u, v) are the coordinate variables in the frequency domain. The Fourier spectrum which is thus obtained is considered as a new image and the Fourier transform is applied again, obtaining the enhanced Fourier spectrum $E_F(u, v)$. In the next step, the power spectrum $P(u, v)$ of $E_F(u, v)$ is computed as follows:

$$P(u, v) = \sqrt{\mathrm{Re}^2(E_F(u, v)) + \mathrm{Im}^2(E_F(u, v))} \tag{4.7}$$

The numerical range of the power spectrum is considerably large. Therefore, to conveniently visualize $E_F(u, v)$, a simple linear transform is used to map $P(u, v)$ to the default gray level range of $[0, 255]$. As by using the following expression:

$$P_L(u, v) = floor\left[\frac{P(u, v) - P_{min}}{P_{max} - P_{min}} \cdot 255\right] \tag{4.8}$$

where $P_L(u, v)$ is the amplitude spectrum after the linear transform, P_{max} and P_{min} represent the maximum and minimum values of it $floor()$ represents the function which rounds down its argument to the nearest integer towards minus infinity.

The peaks due to the periodic information in the image are in the low frequency regions, which are inconveniently placed on the borderlines of the amplitude spectrum. To mitigate this, the center of the image is considered as the origin of the axes of coordinates. The first and third quadrants, and the second and fourth quadrants are respectively interchanged, which is called frequency shift. In this way, the low frequency components are shifted into the center of the image, whereas the high frequency components lie now at the boundary of the power spectrum.

The local maxima of the magnitudes are determined by searching a 5×7 neighborhood of each frequency sample. This size is chosen such that the resulting local maxima are separated from each other by at least two frequency samples. The number of maxima points which can be obtained is limited to 1024 to avoid misleading results. Bright surfaces generate lines of high intensity value pixels which are perpendicular to each other and start at the origin of the spectrum, as shown in Figure 4.7b. The maxima which were previously found that are within a ± 3 pixel range from the center of the image are removed, both horizontally and vertically, to make sure that the shape of the cluster of points is not modified by the illumination conditions. The rest of the coordinates of the extrema points are saved. In the next step, a binary image is obtained by inserting a 1 at the corresponding location where a maximum occurs. Thus, the pattern of the local maxima of the enhanced Fourier spectrum is obtained.

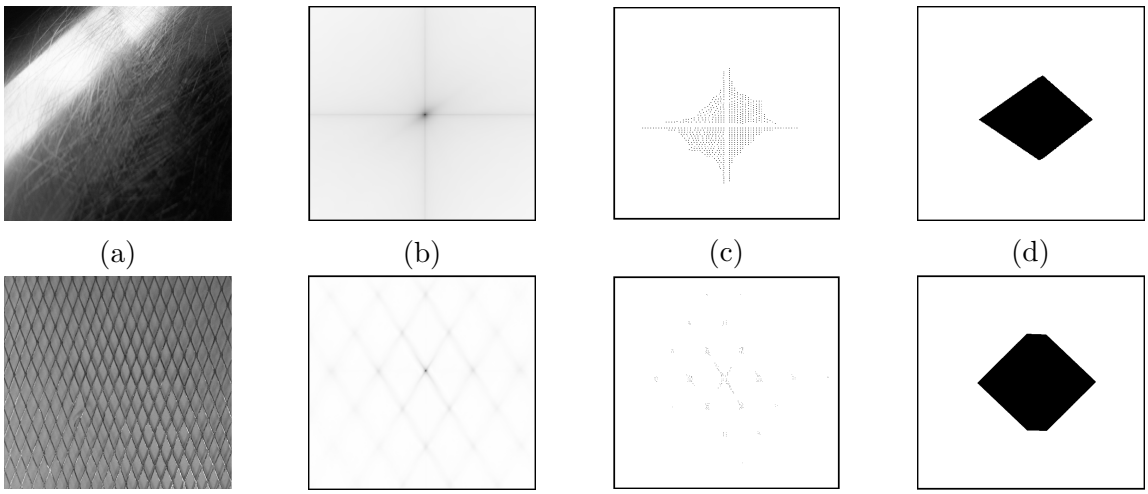
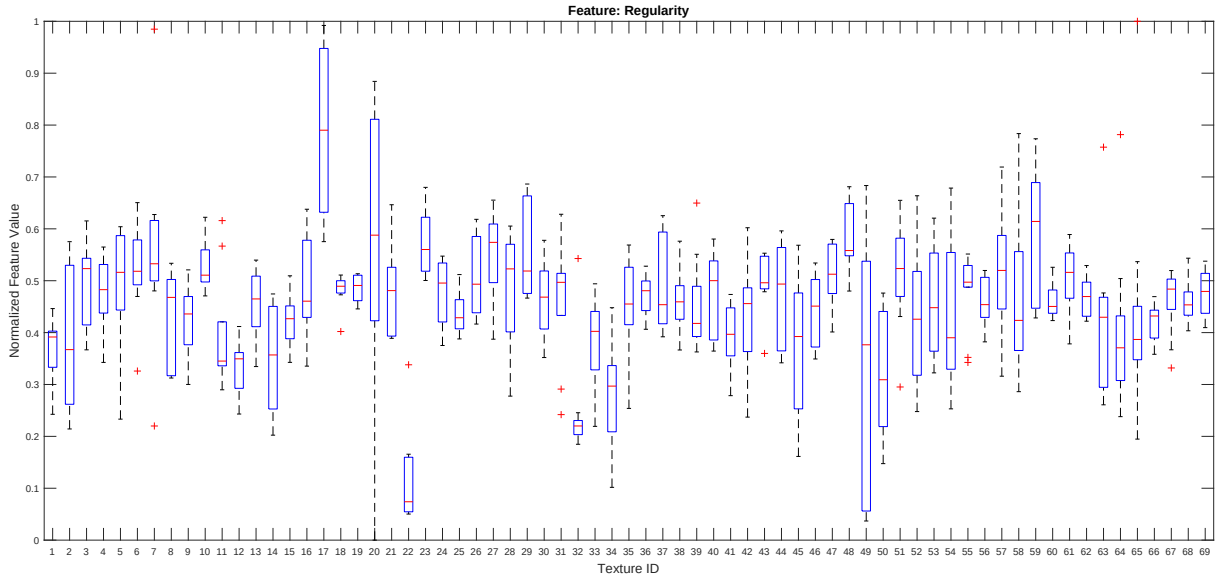
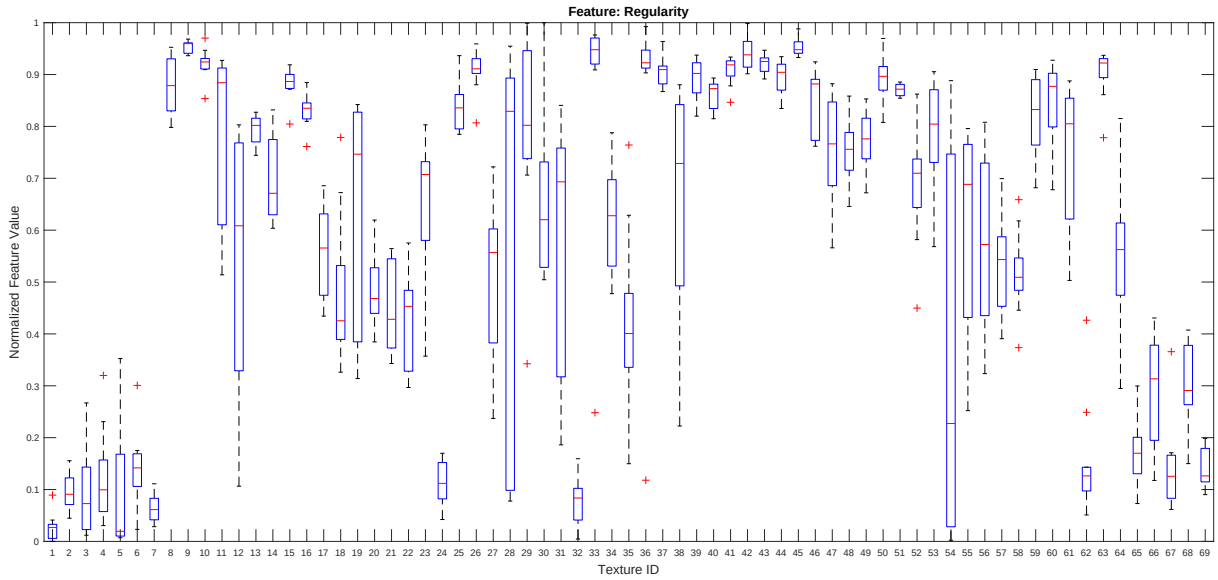


Figure 4.7: Example of two images with irregular (a) and regular (e) patterns. The colors for the enhanced Fourier spectrum (second column from the left), maxima (third column from the left) and convex hull (fourth column from the left) images are inverted for better readability.

The regularity feature ($IREG$) is defined to differentiate the periodic patterns corresponding to highly structured textures from the nonperiodic, cloud-like patterns specific to the



(a)



(b)

Figure 4.8: The regularity feature implemented with the *RWDV* algorithm a) and with the suggested method b). In the former case, a greater discrimination among different classes is achieved, which makes this implementation more suitable for a machine learning approach than the former one.

unstructured textures in the following way:

$$IREG = A \cdot C_A^{-1} \tag{4.9}$$

where A represents the total number of on pixels (with value 1) in the region, and C_A

returns the number of pixels which fall within the smallest convex polygon that contains the region. The reasoning behind this definition is that for the highly regular textures, the pattern of the maximum values of the enhanced Fourier spectrum of the image contains points which are periodically placed with respect to one another, extending substantially in the vertical and horizontal directions. This is why, for this kind of textures, the smallest convex polygon which contains the pattern is usually a parallelogram. The points of the pattern that fall within this convex polygon are sparse due to the periodicity of the points. Consequently, the previously mentioned ratio is a measure of the density of the pattern. For irregular textures, the pattern appears to be cloud-like, with almost all points being placed close to one another, without any outliers. Therefore, the convex polygon does not span wide empty areas and the corresponding ratio of the two areas has a higher value than in the previously discussed case. Considering these, low values of the regularity feature correspond to highly regular textures, whereas high values of the feature correspond to irregular textures.

In Figure 4.8, the boxplots corresponding to the $RWDV$ method and to the improved version are shown. In Figure 4.8a, the overlap among boxes which correspond to different textures is significant, suggesting a very low interclass variance. This is an undesired behavior from the machine learning point of view, because the amount of information which can be extracted from such a feature is very limited. The boxplot in Figure 4.8b, which corresponds to the improved regularity feature, shows that a greater discrimination among different classes is achieved. This observation is sustained by the feature quality measurement described in Section 6.1.3, in this case a $GFC = 3.6658\%$ is achieved, whereas in the first case $GFC = 0.3410\%$. This feature is not only more suitable for a machine learning approach, but it has also a high degree of perceptual correspondence. Textures like meshes and textiles are correctly identified as highly regular, whereas textures like stones, cork, artificial grass fibers, foam and foils are identified as irregular.

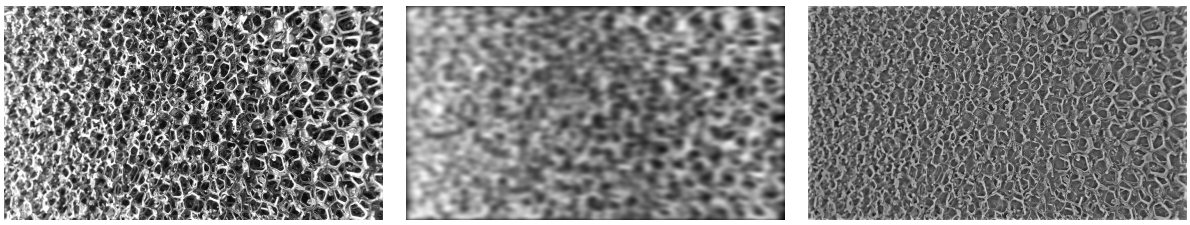
4.2 New Features Definitions

In this section, five touch-relevant features are proposed and described. They are meant to be complementary to the ones which were previously described and to be able to capture new information, which can be later on used by a machine learning approach to train a model that will predict with reasonable accuracy the type of texture under consideration.

4.2.1 Edginess

The visual properties of textured surfaces are highly dependent on the illumination conditions. The edges within a texture, for example, will be enhanced in the presence of light, because of the light-shadow pattern that appears. To implement the edginess feature, given an input image I, a low-pass filter with cutoff frequency of 1% of the width of the Fourier

transform of I is used to obtain the blurred image I_{LPF}, which is shown in Figure 4.9b. The difference image $I_{diff} = I - I_{LPF}$ is computed, which evens out the intensity levels of the initial image and also enhances the edges, if present, as shown in Figure 4.9c.



(a) (b) (c)

Figure 4.9: Edginess calculation. The difference between the original image a) and the blurred image b) exhibits large image energy for the difference image c).

In the end, the edginess feature is obtained as the ratio between the energy (sum of all pixels) of the modulus of the I_{diff} image and the energy of the initial image. A high value of this ratio corresponds to a high number of edges, an edge being defined as a sudden and steep change in the grey levels of neighboring pixels. The energy of an image is a measure of information. For a texture with a high number of edges, even after applying the low-pass filter and computing the difference of the two images, the information content is still going to be high. Since this is still smaller than in the original image, large value of the ratio of the two energies will be obtained. For textures with a few edges, the information which is lost after applying the low-pass filter and performing the subtraction is considerable, yielding a low value for the previously mentioned ratio. The edginess feature (IE) is computed as:

$$IE = \frac{\sum_i \sum_j |I_{diff}(i,j)|}{\sum_i \sum_j I(i,j)} = \frac{\sum_i \sum_j |I - I_{LPF}(i,j)|}{\sum_i \sum_j I(i,j)} \tag{4.10}$$

4.2.2 Color Distance

Even if color is not a tactile feature, it becomes necessary in terms of surface classification while differentiating among different types of glossy surfaces, for example metals. In fact, the human tactile system is not able to distinguish between aluminum, brass and gold by touch only, but by the dominant color during a visual inspection.

To determine the dominant color of the texture, the RGB space is converted to the CIE L*c*h* (Luma Chroma Hue) colorspace. The reason for this is that the RGB space is not "perceptually uniform", which means that a distance metric in this colorspace will not match the distance perceived by humans between colors. In the next step, a cluster of points is formed with all the chroma values on the X axis and all the hue values on the Y axis, as it is shown in Figure 4.10b. The luminance component is disregarded, because

of its high sensibility to illumination conditions, which need not alter the dominant color decision (a blue texture is be classified as blue regardless if the image is taken in a dark room or in a well-lit one). Afterwards, k-means clustering with $k = 5$ clusters is performed, the best arrangement out of five initializations being chosen and also the centroid of each cluster is identified, each one corresponding to the five most dominant colors within the image, as shown in Figure 4.10c. Five clusters are chosen because this yielded the right amount of variance among the dominant colors which resulted that is perceptible by the human subjects, a clustering with $k = 3$ clusters yielding almost identical colors. Once the clusters are determined, the five dominant colors are ordered decreasingly by the number of pixels which are associated to the corresponding cluster, as it is shown in Figure 4.10d.

The k-means clustering is an unsupervised machine learning algorithm that aims to partition m observations into k clusters in which each observation belongs to the cluster with the closest mean. The goal of unsupervised learning is to try to find hidden structure in unlabeled data. To do this, k points from the data are selected and a loop is performed for every point in the set, computing its distance to each one of the k clusters. For every point the nearest cluster is found in terms of the defined metric and that point is associated with the corresponding cluster. At this point, the center for each cluster is recomputed by averaging all distances associated with the points belonging to them. The algorithm is repeated until the centers stop moving according to a specified tolerance.

In the next step, each color is taken as a reference and the difference between it and all the other four colors is measured. The difference between two color samples is often expressed as ΔE and is computed as described in [SWD05]. The resulting number will show how far apart from one another the two colors are in the color sphere described by the L*c*h* colorspace. ΔE expresses the difference between two colors as a single number, a ΔE value of 4 and over being normally visible to an untrained user. Because there are 5 clusters, there are 5 possible colors which can be chosen as a reference and the one that yields the smallest variance of the color distance within the considered group is chosen. Afterwards, by considering only the three most dominant colors, it is decided, based on the distances which were previously determined and an empirically based threshold, whether the texture has one or two dominant colors. If the three most dominant colors have all a distance among them which is less than 41, then there is only one dominant color whose RGB components are determined by averaging the RGB values of the three colors. If not all colors have distances among them which are less than 41, then the RGB values of the ones that obey this rule are averaged, obtaining in this way the first dominant color, and the RGB components of the other colors which have the distance greater than 41 are also averaged, determining the second dominant color.

Very few naturally occurring textures exist which exhibit two radically different dominant colors exist. However, this algorithm is still capable of identifying these artificial textures. Sometimes one of the colors are either black or white, which suggests that bichromaticity can also be due to illumination.

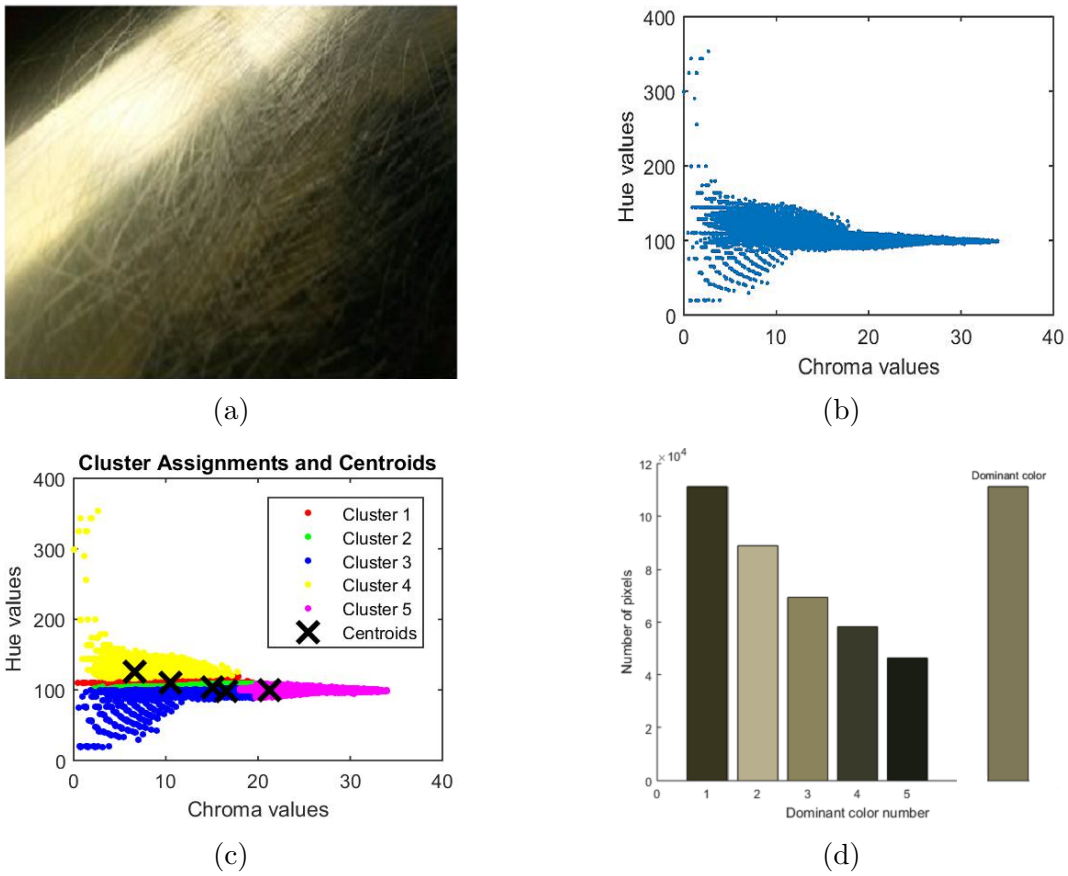


Figure 4.10: Dominant color example: given the texture a), k-means clustering is performed c) to determine the dominant color, which is shown in the last column in d).

However, the decision whether there are one or two dominant colors is not relevant for the surface classification task considered in this work. Therefore, the RGB components of the two dominant colors, provided it is the case, are averaged to obtain a single dominant color, as shown in Figure 4.10d. A single value, even as a mixture of two dominant colors, uniquely describes each image in terms of the visual appearance.

The center of the LCH color sphere, which is shown in Figure 4.11, is chosen as the origin of axes and the dominant color which was previously obtained is represented by a point within this sphere. The image color distance feature (ICD) is represented by the red line in Figure 4.11. It models the perceptual distance between the dominant color and the reference one, which

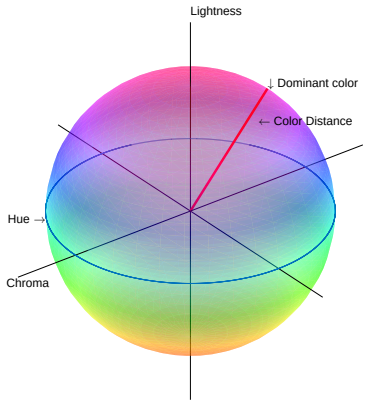


Figure 4.11: Color sphere of the LCH colorspace.

is represented as pure black, and is computed as:

$$ICD = \Delta(C_D, C_R) \tag{4.11}$$

where C_D represents the dominant color, C_R the reference color and Δ the distance between the two colors in the LCH colorspace.

4.2.3 Roughness

When natural textures are inspected, people are able to classify them in terms of roughness. The textures can be cataloged as either rough or smooth, as it is shown in Figure 4.12. The question to be answered is whether this subjective judgment that is performed subconsciously is due to the total energy changes in gray levels or due to the tactile sense that is emulated in the human brain, recreating the feeling which is perceived after touching the textures.

(a) (b)

Figure 4.12: Different roughness levels. a) Rough texture. b) Smooth texture.

Human beings have the amazing ability of transferring knowledge from one perceptual impression to another through the means of intermodal coordination, as described in [HSG03], being able to predict the haptic features of an object with a seemingly high degree of accuracy only by visualizing it. Even if usually a surface needs to be touched to sense how rough (or smooth) it is, the human visual system is able to perceive these properties even without a haptic input. For example, tiny hairs on fabrics, smooth surfaces in glass objects, crinkles in leather and grains in paper can be detected only by visual inspection.

The proposed regression-approximated roughness feature is based on $GLCMs$ of the input image I, which describe a set of visual properties and on the outcome of a subjective experiment, as described in Chapter 5. In [HSD73] Haralick et al. first presented $GLCM$ for texture descriptions in the 1970s and this method is popular until today, due to its good performance and reliability. The $GLCM$ can be either symmetric, when $P(i,j) = P(j,i)$, or nonsymmetric, depending on whether the pair of pixels is counted once (only in the forward direction) or twice (also in the forward and also in the backward direction).

Research shows that texture features which are extracted from the $GLCM$ can be used as effective descriptors for the image properties. Since the $GLCM$ measures the spatial relationship among gray levels, it can explore the nature of textures seemingly well.

Initially, the images have 256 gray levels, implying that the computed $GLCMs$, as in [HSD73] and in Section 2.3.4, will be sparse matrices of size 256×256. To reduce the number of zero elements of these matrices and to make the algorithm more computationally effective, the number of gray levels of the image is reduced to 32.

In the next step, the $GLCM$ values are normalized in order for its elements to represent probabilities instead of numbers. The normalization process implies the division of the number of counts by the total number of counted pixel pairs, as detailed in [HSD73]. In the horizontal direction, for $d = 1$, there will be $2 \cdot (N_x - 1)$ neighboring resolution cells on each row, times N_y rows, yielding a total of $2 \cdot N_y \cdot (N_x - 1)$ nearest horizontal neighbor pairs. For the nearest right diagonal neighbor ($d = 1$, $\theta = 45°$) there will be $2 \cdot (N_x - 1)$ 45° resolution cell pairs for each row except the first, for which none exists, and there are N_y rows, yielding $2 \cdot (N_x - 1) \cdot (N_y - 1)$ nearest right diagonal neighbor pairs. Similarly, there will be $2 \cdot (N_x - 1) \cdot (N_y - 1)$ nearest left diagonal neighbor pairs and $2 \cdot N_x \cdot (N_y - 1)$ nearest vertical neighbor pairs. The matrix is normalized by dividing each element in the matrix by the corresponding number of neighboring pairs.

The $GLCMs$ are computed considering a certain distance and angle between the pixels. The angular dependency makes the $GLCMs$, and also the subsequent features computed based on them, rotation dependent, which is not desirable. To illustrate this, assume image A has the four $GLCMs$ a,b,c,d for angles 0°, 45°,90°, 135° respectively and image B is identical with A, being only rotated by 90°. Thus, B will have the $GLCMs$ c,d,a,b for angles 0°, 45°, 90°, 135° respectively. Thus, to produce the same features when considering the two images, in [HSD73] it is suggested that the angularity dependent features not be used directly, but instead two functions of a,b,c,d, their average and range (which are invariant under rotation) be used as the inputs to the classifier. To mitigate this problem, the parameters using the $GLCMs$ computed for the four directions are averaged through the four angular relations, making thus the parameters rotation invariant.

After the experiment described in Chapter 5, the tactile-relevant mean ordering results are used to define roughness approximation. All the 14 Haralick parameters, as detailed in [HSD73], are computed for each image in the database. Even though these features contain information about the textural characteristics of the the image under analysis, it is hard to pinpoint which specific textural property is represented by each of the computational features previously suggested. Most of these features have a visual meaning. For example, contrast W_2 emphasizes the sharpness of the image and the amount of local variations that occur. The darker the texture grooves, the higher the value of the contrast feature and the clearer appears the image to the user. On the other hand, the lighter the texture grooves, the smaller the value of the contrast and the less sharp is the visual effect. Thus, the more likely the pairs of pixels with high contrast, the larger the value of W_2. Inverse Difference Moment W_5 measures the local variation of the image texture, denoting how

uniform the entries in the $GLCM$ are. If the value of W_5 is large, that means that small changes between different regions of the image occur, suggesting thus uniformity. Entropy W_9 is a measure of the degree of spatial disorder of the image. When the $GLCM$ elements have a higher dispersion, the value of the entropy is larger. Thus, the more complex the texture is, the higher the value of the entropy, whereas the more uniform a texture is, the larger the difference between the $GLCM$ elements and the smaller the value of the entropy. Therefore, the entropy measures the degree of heterogeneity and complexity of the image texture, whereas energy is a measure of its local homogeneity.

Multivariate normal regression is performed to compute the coefficients column vector $\mathbf{b}$ that would yield a ordering similar to the one obtained after the experiment, denoted by the column vector $\mathbf{y}$. The Haralick coefficients matrix is denoted by $\mathbf{H} = [\mathbf{w}_1\ \mathbf{w}_2\ ...\ \mathbf{w}_{14}]$, where $\mathbf{w}_k$, for $k \in \{1,\ ...\ , 14\}$ are the column vectors of the Haralick coefficients corresponding to all images in the database. The problem that needs to be solved can be written as $\mathbf{y} = \mathbf{Hb}$.

Both symmetric and nonsymmetric $GLCM$ configurations were tried out when considering all possible Haralick parameter combinations, namely $2^{14} - 1 = 16383$. The highest matching accuracy to the subjective test, which is computed as described in Chapter 5, was obtained when the $GLCMs$ are nonsymmetric, yielding a maximum accuracy $a = 0.3623$ with the following 6 Haralick coefficients: W_3 (Correlation), W_5 (Inverse Difference Moment), W_6 (Sum Average), W_8 (Sum Entropy), W_{12} and W_{13} (Information Measures of Correlation). With symmetric $GLCMs$ the highest accuracy which was obtained is $a = 0.3478$ using only 4 Haralick coefficients. In terms of the GFC feature, the results are also better for the nonsymmetric $GLCMs$ than for the symmetric ones, suggesting that the former are a better option from a machine learning point of view. Since the goal is to obtain a feature which corresponds as much as possible to the the human perception, the results obtained after considering the nonsymmetric $GLCMs$ are chosen for the computation of the roughness feature (IR), as follows:

$$\begin{aligned} IR = {} & 183.25W_3 + 8.03W_5 - 0.66W_6 \\ & -66.85W_8 - 510.32W_{12} - 51.01W_{13} \end{aligned} \tag{4.12}$$

4.2.4 Glossiness

The appearance of surfaces results from an intricate combination of illumination, reflection and surface geometry, which are very difficult to separate from one another. Therefore, the gloss of a surface is directly connected to its capability of reflecting light and thus its microstructure. Two components, related to diffuse and specular lighting reflections, are considered as important for the surface classification task at hand.

In [MNSA07] it is shown that the skewness of the luminance histogram is correlated with surface gloss. Also, useful evidence is found that human observers use skewness, or a similar value of histogram asymmetry in judging different properties of surfaces. When the

image of a surface has positively skewed statistics, it tends to appear as being darker and glossier than a similar surface with a smaller skewness.

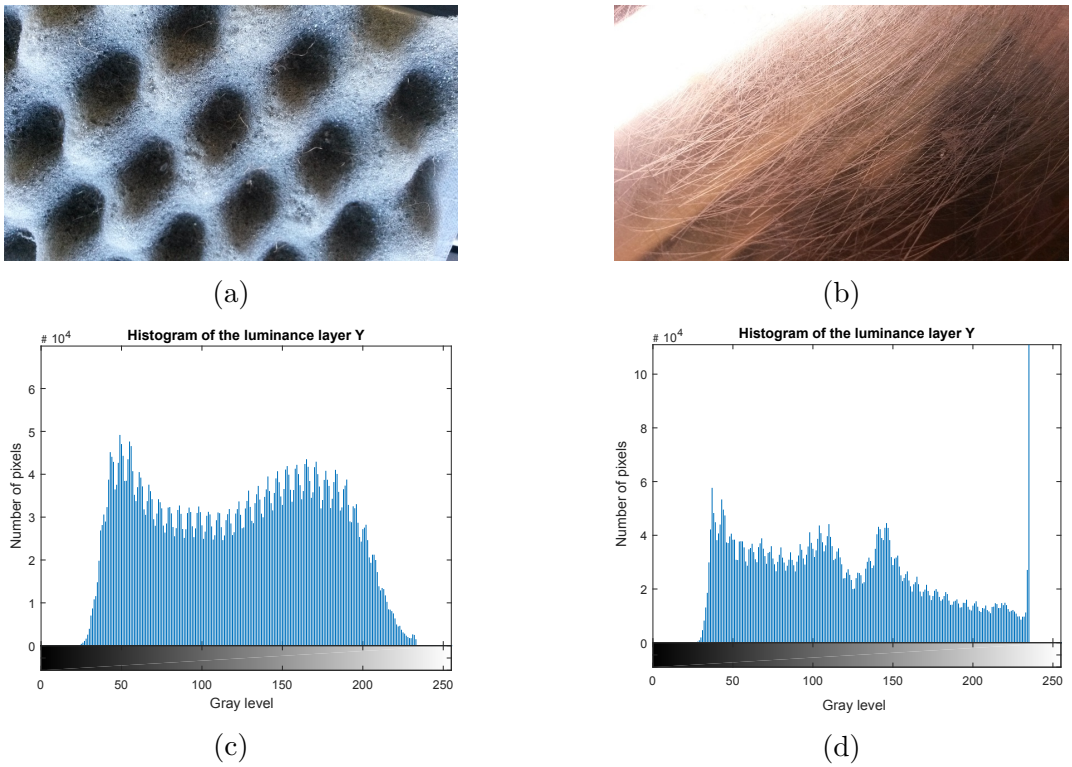


Figure 4.13: For the profiled foam texture shown in a), the skewness of the luminance histogram H_Y, illustrated in c), is $s = -0.5021$ and $r = 0.0115$. For copper, shown in b), the skewness of the luminance histogram H_Y, represented in d), is $s = 12.5790$ and $r = 0.0619$.

To obtain the glossiness feature, the RGB image is converted into the YCbCr colorspace, which is perceptually uniform. Only the Y layer is kept because of its human perceptually-relevant luminance motivation. The intensity values of image I are mapped to the new values in $I_{adjusted}$ to modify its contrast so that 1% of the data is saturated at low and high intensities of I. Afterwards, the histogram H_I of $I_{adjusted}$ and also the histogram H_Y of the initial image luminance layer are computed. Subsequently, the skewness s of H_Y is determined, which is correlated to the surface gloss, as previously mentioned. If the luminance histogram H_Y is negatively skewed, the corresponding surface is perceived in general as less reflective and has lower diffuse and ambient light reflection capabilities, whereas when H_Y is positively skewed, the surface is perceived as more reflective and thus glossier. As for the specular reflection, the area under the histogram H_I from gray level 250 to 255 is computed, obtaining A_s and this value is divided by the area under the entire histogram A_t, obtaining thus the area ratio denoted by r. Experimentally, it follows that high values in the intensity levels correspond to specular reflections and the existence of spotlights. This component also allows to even differentiate between glossy surfaces and

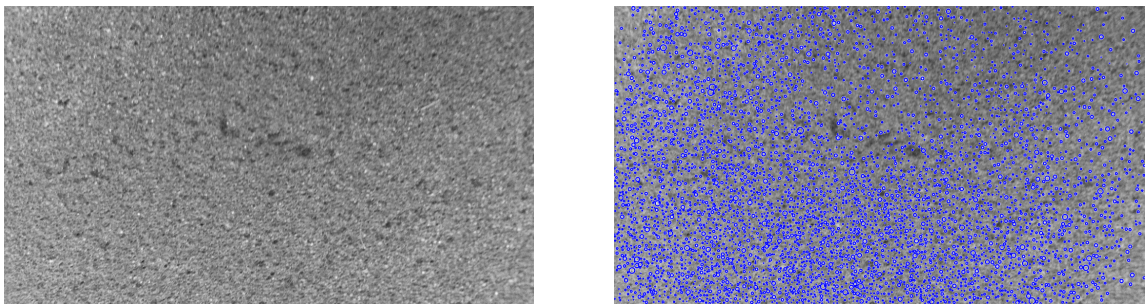
surfaces with large white areas. Characteristically high values for A_s can only be achieved, when a surface has a strong tendency for specular reflections, whereas a simply white surface has most of its high intensity values below the previously mentioned threshold. The image glossiness feature (IG) is defined as:

$$IG = \frac{A_s}{A_t} + s \; = \; r + s \tag{4.13}$$

In Figure 4.13, a matte surface is presented against a glossy surface which has a long, positive tail. The luminance histogram has a negative skew for the left image ($s = -0.6353$) and is positively skewed for the right image ($s = 1.0770$). In statistics, skewness measures the asymmetry of a probability distribution function, indicating the balance between its positive and negative tails. It can be positive or negative.

4.2.5 Softness

Softness is an important touch-based feature that humans recognize when faced with a new texture. However, it is believed that the haptic perception that the humans are able to infer only by looking at a texture is due to the unconscious process of recognizing the texture from previous experiences and retrieving from memory how it feels like. To extract the softness information only by processing images, without having any prior knowledge of the textures under consideration, two assumptions are made to implement the algorithm. In the first place, it is assumed that a texture which exhibits a considerable number of hairs (filaments) is highly likely to be soft. The second assumption which is made concerns the very structure of foams, namely that they have gaps in their structure, which has most of the times a circular form. Therefore, a custom image analysis algorithm is developed which is able to perform filament and circle detection, to infer the softness of textures.



(a) (b)

Figure 4.14: In a) the grayscale image I of a foam texture is shown. In b) the superposition of the circles detected by the CHT algorithm and image I is presented.

In the first stage, given the grayscale image I, circle detection is performed, searching for radii between 2 and 20 pixels, using the circular Hough transform (CHT). This is a very

robust algorithm in the presence of noise, occlusion and different illumination conditions. The CHT is not a rigorously specified algorithm, a multitude of approaches being available for its implementation. However, the following steps are common for all implementations. First of all, the gradient of the image is computed, by convolving the image with a filter, for example the Sobel filter. Afterwards, the pixels of high gradient vote in pattern around them that forms a circle of a fixed radius and are accumulated in an accumulator array. This way, the edge points that lie along the contour of a circle of a given radius tend to accumulate at the accumulator array bin which corresponds to the center of the circle. In the third step, to perform center estimation, peak detection in the accumulator array is performed. Once a peak is detected, which means that a circle is found at a particular point, nearby points which lie within one half of the original radius are excluded from the candidate set of circle centers to avoid to detect the same circular feature more times. After the algorithm is complete, a vector is obtained, which contains the position, as well as the estimated radii for all circles which are found in the image, which fall within the specified radius range.

In the next step, the sum of areas of the n circles which are found is computed, obtaining the total sum of circles areas A_c. Afterwards, R_c is determined, which represents the ratio of circles in the image, which is computed as:

$$R_c = \frac{A_c}{h \cdot w} = \frac{\sum_{i=1}^{n} \pi r_i^2}{h \cdot w} \tag{4.14}$$

where h and w represent the height and the width of the image, respectively. To illustrate this, in Figure 4.14b the superposition of the grayscale image I, corresponding to a foam texture, and the circles in the range $[2, 20]$ pixels returned by the CHT algorithm, which are marked with blue, is shown. The area of the individual circles is not negligible, in this particular case a ratio $R_c = 8.63$ is obtained. In this way, a large value of the ratio will be describe textures which have a considerable amount of circular patterns in their structure, as is the case of foams, suggesting that these surfaces are soft, which is perceptually accurate.

In the next stage, the filament detection part of the algorithm is implemented. Hairs, or filaments as they are commonly referred to, have a very high aspect ratio, which means that they appear as long and narrow objects within an image. Starting with the grayscale image I, histogram equalization followed by 3×3 median filtering is performed to increase the contrast, reduce noise and preserve structure edges. Even after these steps, when histogram analysis is performed, if the pixels of the object are in the low gray level domain and are distributed over a wide range, the contrast between the object and the background is low, as shown in Figure 4.15a. Thus, it is difficult to perform image segmentation to be able to extract the region of interest. To dramatically improve the contrast, a piecewise nonlinear transform model is used. A two-piece nonlinear model is used to segment the filament-like objects out of the background. It is considered that the original gray level in image I at position (i, j) is $O_{GL}(i, j)$ and the corresponding gray level after enhancement is $E_{GL}(i, j)$. Let the initial gray level range of the objects be $[0, L_h]$. As described in

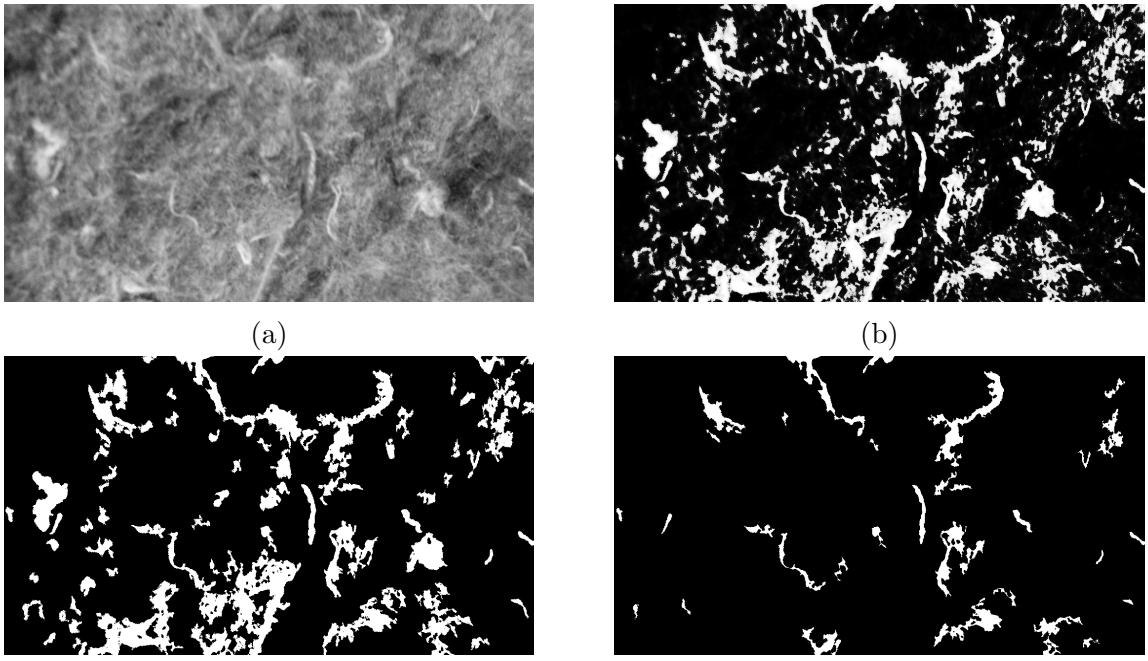


Figure 4.15: Illustration of the hair segmentation algorithm. a) Grayscale image of a hairy texture. b) Grayscale image after contrast enhancement. c) Area filtered binary image. d) Binary image obtained at the end of the algorithm, which contains the detected hairs.

[YLZ+09], the two-piece image enhancement nonlinear transform model is described by:

$$E_{GL}(i,j) = \begin{cases} \frac{L_h}{\sqrt{L_h - O_{GL}(i,j)+1}} - \frac{L_h}{\sqrt{L_h}+1}, & O_{GL}(i,j) \leq L_h \\ O_{GL}(i,j), & O_{GL}(i,j) > L_h \end{cases} \tag{4.15}$$

After conducting a series of tests with different values for the threshold value, $L_h = 210$ offered the best contrast enhancement. The outcome is shown in Figure 4.15b. In the next step, the contrast-enhanced intensity image is converted to a binary image by performing gray level thresholding using Otsu's method, which calculates the optimal threshold that minimizes intraclass variance of the black and white pixels. After thresholding, it is very likely that the new binary image will contain small objects that are of no real interest for the algorithm. This is why area filtering is performed on the binary image which was previously obtained to extract all connected components (objects) which have an area (number of white pixels) between $[2000, 150000]$ pixels. In this way, a new binary image as shown in Figure 4.15c is obtained. There are also circular-like objects which are also of no interest for the segmentation algorithm. Therefore, any object in the binary image which has an extent less that 0.55 is eliminated. The extent is a number which represents the ratio between the number of pixels in the region and the total number of pixels in the bounding box. To disconnect as much as possible the objects, without altering too strongly their shape and area, the binary image from the previous step is eroded with a diamond-shaped

structuring element with a size of 8 pixels. The morphological operation of erosion of the binary image B by the structuring element E is the mathematical operation that returns the set of pixel locations z, where the structuring element which is translated to location z overlaps only with the foreground pixels in B. In the last step, for each connected region in the image the maximum radius r_{max} of a circle which can fully contain the object whose area is A_{obj} is determined and its corresponding circularity C is computed:

$$C = \frac{A_{obj}}{\pi * r_{max}^2} \tag{4.16}$$

All objects which have a circularity value greater than 0.08 and an area which is not within the interval $[1000, 50000]$ are disconsidered, because they either correspond to objects which are not filament-like or to specular reflections. The binary image which is obtained after the segmentation process is done is shown in Figure 4.15d. Only the linelike objects are preserved from the initial image, with a seemingly high degree of accuracy which is also perceptually accurate. Then, the area of the filament-like objects A_f is computed and normalized by the total number of pixels in the image, obtaining the ratio of filament objects in the image R_f as follows:

$$R_f = \frac{A_f}{h \cdot w} = \frac{\sum_{i=1}^{h} \sum_{j=1}^{w} BW_{filament}}{h \cdot w} \tag{4.17}$$

The softness computational feature (IS) is defined as:

$$IS = R_c + R_f \tag{4.18}$$

This equation models the observation that a texture is very likely to be soft if it has predominantly filament-like or circular-like objects, or both. Thus, a high value of this feature corresponds to a soft surface, whereas a low value would indicate the opposite.

Chapter 5

Subjective Experiment

A subjective test with 20 subjects, out of which 4 were female and 16 male, was performed to obtain the roughness feature and to find the regression coefficients. The experiment consisted of two parts and both parts were conducted on different days. The first one required the subjects to order the textures in the database from the roughest to the least rough one only by looking at images of them, and in the second part a similar ordering had to be performed, this time the subjects inspecting the surfaces with their fingers to determine the roughness level of each texture. Before the visual part of the experiment, the subjects were asked to base their decisions only on what they perceive in terms of roughness when looking at the database images and to avoid recognizing what the textures really are, excluding thus the recreation of the feeling of the texture when performing the ordering. For this part of the experiment, two monitors were used, where the subjects could order the surface image files on the desktop according to the visually perceived roughness level. Prior to the touch-based experiment, the subjects were asked to keep the inspection speed and applied pressure as consistent as possible throughout the whole set of textured surfaces, because these factors can influence how the humans perceive roughness. The textured surfaces were sliced in stripes and two tables were used to allow the subjects to freely perform the ordering procedure. The average time spent performing the image and touch-based part of the experiment was 26.5 minutes and 28.25 minutes, respectively.

In the visual part of the experiment, subjects had the tendency to classify textures like brass and copper as very rough. This can happen because of the visible scratches these structures exhibit, which are falsely interpreted by the subjects as hairs/fibers, making them believe they are rougher than they actually are. This observation is in accordance with the linelikeness computational feature, which for these two textures yields very high values. In the touch-based part of the experiment, the subjects have encountered difficulties when trying to establish a hierarchy in terms of roughness when smooth surfaces had to be ordered. This indicates that the human touch is more sensible for rough surfaces and for smooth surfaces this sensibility is decreased. A number of subjects also reported difficulties when trying to order the textures that have a medium level of roughness, suggesting that

the sensibility level can decrease considerably as a texture becomes less rough.

As a second observation of the subjective experiment, the average image and touch-based orderings, which are illustrated in Figure 5.2, are directly compared. To model the observation that the human perception of roughness has a decreased sensibility as the roughness level of the inspected surfaces decreases, a method was developed that considers some tolerance d while matching two orderings from two disjoint rank lists, since it is extremely unlikely that two rankings will completely match. The meaning of the tolerance d is the following: if an image gets rank i in one of the ranked lists and in the other gets the rank $i \pm d$, it is assumed that there is an agreement between the two ranking lists for that image and this is considered as a match. The accuracy a of matching the two ranked lists is computed as the ratio between the number of matches divided by the total number of images. A slight mismatch is desired when ranking the top roughest images, for the middle range of roughness there can be a considerable mismatch between the two rankings and for the images which are least rough, the ordering is not important. To use a position dependent d that would yield small tolerances for matching highly ranked images and increase the tolerance value up to a certain threshold and then maintain the obtained value for the low ranked images, an exponential equation was used for the tolerance. As it is suggested in [IZL08], $d(r)$ is implemented as:

$$d(r) = 3 + 4.5(1 - e^{-2r/\mu}) \tag{5.1}$$

where μ is the mean of ranks from 1 to 69, and $d(r)$ is the tolerance for rank r. The integer part of the $d(r)$ values is used in the computation of the agreement accuracy, as shown in Figure 5.1. By using this metric to measure how similar the two average rankings across the subjects are, a value of $a = 0.4638$ is obtained, which indicates, intuitively, that a positive correlation between the visual and tactile orderings exists, including also several mismatches.

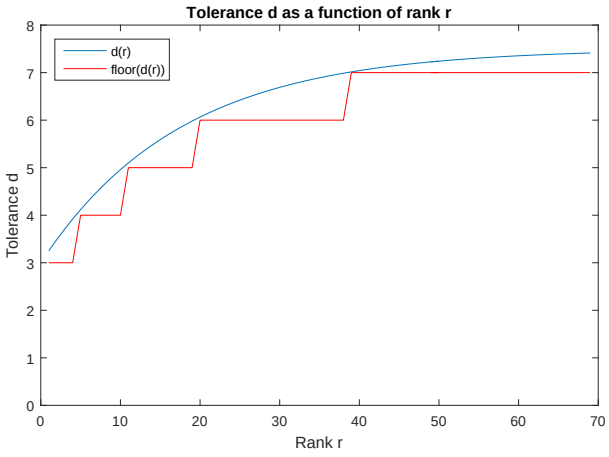
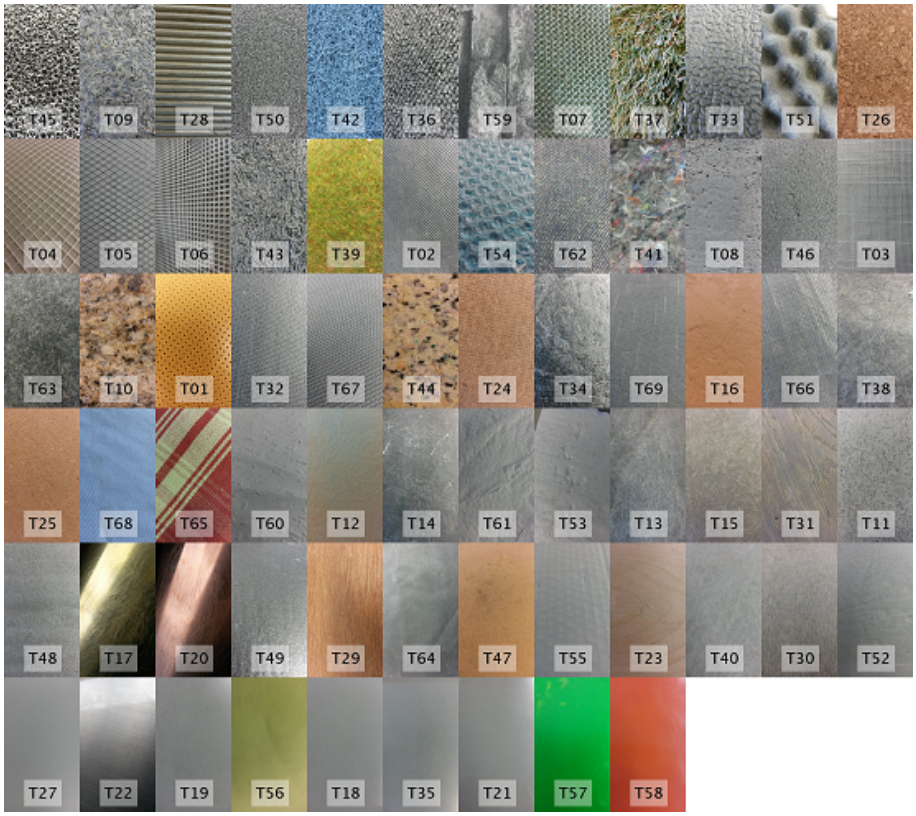
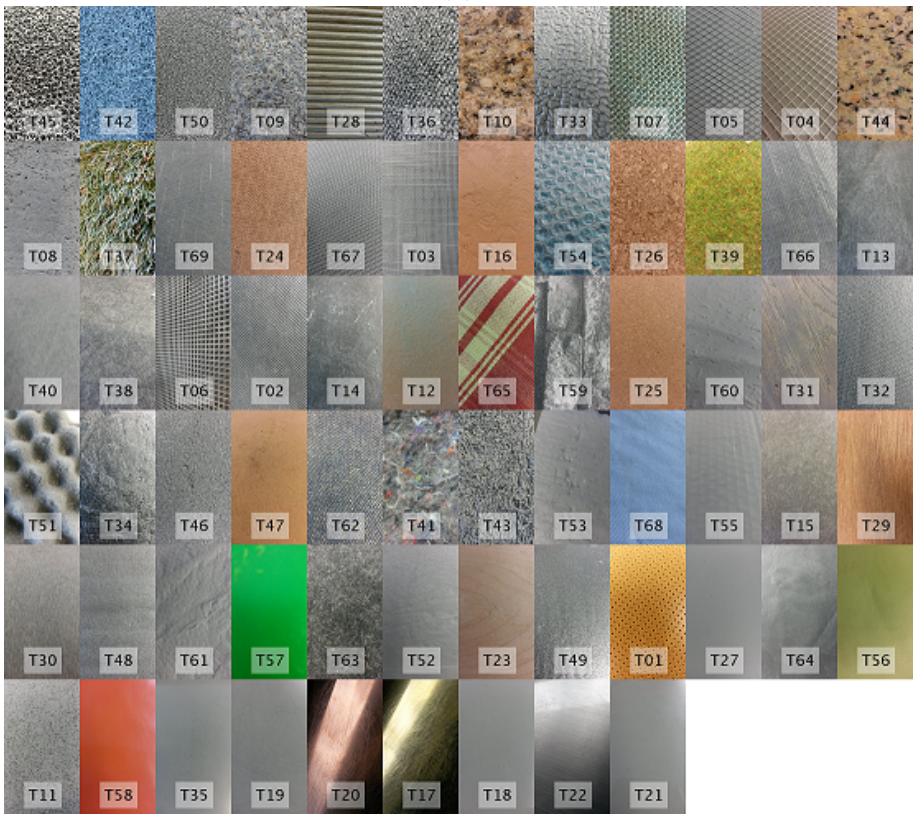


Figure 5.1: Rank-dependent tolerance.



(a)



(b)

Figure 5.2: The two average orderings obtained at the end of the subjective experiment. a) Image-based ordering. b) Touch-based ordering.

Chapter 6

Results

6.1 Statistical Evaluation of Features

6.1.1 Feature Scaling and Cross Validation

To achieve good performance for many machine learning algorithms, normalization is an important pre-processing step that needs to be made. Feature scaling is a method which is used to standardize the range of data features. The motivation for normalizing the range of all features which are considered for a machine learning approach is that most algorithms use for the objective function some kind of metric (e.g. Euclidian distance) and if one feature has a broader range than another, then the bias would be towards the first feature. Therefore, through normalization each feature contributes in the same manner to the final metric.

The rescaling of the features to the fixed range [0,1] was employed as a pre-processing step. The formula for describing this is:

$$\mathbf{X}_{norm} = \frac{\mathbf{X} - X_{min}}{X_{max} - X_{min}} \tag{6.1}$$

where $\mathbf{X}$ is the vector of original values and $\mathbf{X}_{norm}$ is the vector of normalized values. The drawback of having this bounded range is that after normalization, the standard deviations are going to be smaller than initially, which can suppress the effects of outliers.

As described in [Gei93], cross-validation is a very useful model validation method which is used for evaluating how the results of a statistical analysis will map to an independent data set, which was not seen before by the model. The success rate after cross-validation is a good estimation of how the predictive model will behave in practice. To perform prediction, a part of the data in the original database is not considered. The model is trained with the rest of the data, which represents the training dataset. Afterwards it is tested against

the data which was initially removed, which represents the validation dataset. In k-fold cross-validation, the original dataset is randomly partitioned into k equally sized subsets. From these subsets, only one sample is considered as the validation data, which is used for testing the model, whereas the remaining $k-1$ samples are used as training data. To reduce variability, this cross-validation process is repeated k times, each of the k samples belonging only once to the validation dataset and $k-1$ times to the training dataset. Then, the average error across all k trials is computed. The major advantage of this method is that it is not sensible to how the data is divided. The variance of the resulting estimate is reduced as k is increased. The major disadvantage of this approach is that the training algorithm has to be rerun from the very beginning k times, which yields a k-fold increase in the time required for obtaining an error estimation.

An extreme case of k-fold cross validation is the "leave-one-out cross-validation", where $k = n$, the number of data points in the set. This is very expensive from a computational point of view. On the other hand, the simplest type of cross-validation is the holdout method which simply separates the data set into a training and test part, without any other restrictions. Even if this is the fastest among all flavors of cross-validation, the evaluation can show a high variance, because it is heavily dependent on how the division in data sets is done. The goal of cross-validation is to mitigate situations like overfitting and also estimate how the model will behave on a dataset of unseen data.

Overfitting is a modeling error which usually occurs when a model is too complex with respect to the training set size. This can be thought of as a model who is more likely to memorize the training data rather than generalize it. This usually occurs when attempting to fit the model too closely to slight inaccuracies in the data, which usually arise from some degree of error or random noise. In this way, considerable predictive errors occur which translate to a reduced predictive power.

6.1.2 Statistical Analysis

As described in [ES10], the null hypothesis, which is denoted by H_0, refers to the statement that there is no difference between two groups. The alternative hypothesis, denoted by H_1, represents the opposite of H_0, which is the hypothesis that is to be investigated and is expected to be true if H_0 is false. In the particular case considered in this work, the null-hypothesis states that groups A and B have the same mean. However, it is not possible to prove that H_1 is true. What can be proven is that the alternative is much more plausible than the null hypothesis, given the data under consideration, through probabilities (p-values) which reflect the strength of the evidence against H_0, favoring thus H_1.

To measure the level of statistical significance of a test, the so called p-values are used. Depending on the statistical test which is taken under consideration, the probability (expressed through the p-value) of observing the sample data (or a more extreme test statistic in the direction of the alternative hypothesis) when the null hypothesis is true is calculated.

As it is mentioned in [ES10], the p-values are commonly misinterpreted as the probability of the null-hypothesis or that it models the probability of the data which has arisen by chance. The significance level is the probability level at which the null hypothesis will be rejected, modeling thus the probability of making a type I error. It is denoted by α and it commonly set to 0.01 or 0.05. If the p-value $p \leq \alpha$, H_0 is rejected in favor of H_1. If $p > \alpha$, the H_0 is not rejected. Given the sample data, it is worth investigating whether this is consistent with H_0 or whether it can be accepted that the samples offer reasonable evidence to support H_1. To address this question, the value of the test statistic needs to be computed. This is done by performing a significance test, which is a statistical procedure that when applied to the set of observations, it yields p-value relative to H_0. A Student's t-test is performed, which is a significance test which asses hypotheses concerning population means. Since the samples in the defined features are independent, the test statistic is:

$$t = \frac{\bar{x_1} - \bar{x_2}}{s\sqrt{\frac{1}{n_1} + \frac{1}{n_2}}} \tag{6.2}$$

where $\bar{x_1}$ and $\bar{x_2}$ are the means of samples of size n_1 and n_2 taken from each population, whereas s^2 is an estimate of the assumed common variance given by:

$$s^2 = \frac{(n_1 - 1)s_1^2 + (n_2 - 1)s_2^2}{n_1 + n_2 - 2} \tag{6.3}$$

where s_1 and s_2 are the variances of $\bar{x_1}$ and $\bar{x_2}$, respectively.

If the null hypothesis of the equality of the two population means is true, t has a Student's t-distribution with $n_1 + n_2 - 2$ degrees of freedom, allowing p-values to be calculated. This is also called the null distribution and the distribution of the variable t is computed as:

$$t = \frac{\sqrt{n}(\bar{x} - \mu)}{s} \tag{6.4}$$

where $\bar{x}$ is the arithmetic mean of n observations from a normal distribution with mean μ and s is the sample standard deviation. As in [Sta12], the number of degrees of freedom of a distribution is equal to the number of observations divided by the measurements which were performed in the experiment.

Given the value of the test statistic (which is a number), and the null distribution (which is a probability density function) of the test statistic, of great importance is to investigate where the test statistic is placed within the distribution. If t is in the middle of the distribution, then it means that the observations are consistent with the null hypothesis, whereas when t is placed in a tail of a distribution, the alternative hypothesis tends to be more plausible the further apart in the tail t is to be found. When a right tailed test needs to be performed, only the right hand tail of the distribution is considered. In a similar manner a left tailed test is conducted. Also both tails can be considered when a two tailed test needs to be performed. All these depend on how the test statistic and alternative hypothesis are defined.

The p-value is a number between [0,1] and a value $p < 0.001$ shows very strong evidence against the null hypothesis in favor for the alternative one, making the samples under consideration to be statistically highly significant. If $p < 0.05$ the samples are statistically significant, whereas a value $p > 0.05$ indicates weak evidence against the null hypothesis, favoring thus the alternative one.

The Kruskal-Wallis test is a distribution-free method that is analogous to the analysis of variance of a one-way design. As described in [Mur12], a nonparametric test does not make any assumption about the probability distributions of the variables which are taken into consideration, unlike parametric tests which assume that the population under consideration is normally distributed with the mean μ and variance σ. This is very useful when the data is not normally distributed, as it is the case of the results presented in this work. However, the data for nonparametric tests are an independent random sample, which means that the samples are not be correlated.

The omnibus Kruskal-Wallis method tests whether the groups to be compared have the same population median to determine if the samples come from the same population (or, equivalently, from different populations with the same distribution). The Kruskal-Wallis test uses ranks of the data, rather than numeric values, to compute the test statistics. As described in [ES10], it finds ranks by ordering the data from smallest to largest across all groups, and taking the numeric index of this ordering. The rank for a tied observation is equal to the average rank of all observations tied with it. The test statistic is obtained as:

$$H = \frac{12 \cdot \sum_{i=1}^{k} n_i(\bar{R}_i - \bar{R})^2}{N(N-1)} \tag{6.5}$$

where n_i is the number of observations in group i, R_i is the mean of their ranks, R is the average of all the ranks, given explicitly by $R = (N+1)/2$. When the null hypothesis is true the test statistic has a chi-squared distribution with $k-1$ degrees of freedom.

The Kruskal-Wallis test only tells that at least two groups are different from one another, but does not yield also the information about which of them are statistically different from each other. To determine how many groups are statistically different from one another, a follow-up post-hoc test is performed.

Post-hoc comparisons are analyses which are not explicitly planned at the start of a study but are needed to examine the data. Such comparisons are generally performed only after obtaining a significant overall F value. In this work, the F-statistic used in classical one-way ANOVA is replaced by a chi-square statistic, which is obtained after performing the Kruskal-Wallis test.

The post-hoc comparisons imply multiple comparison tests, which are procedures for the detailed examination of the differences between a set of means, usually after a general hypothesis that they are all equal was rejected. No single technique performs equally well in all situations and a major distinction between techniques is how they control the possible inflation of the type I error. To illustrate this, consider for example that when performing

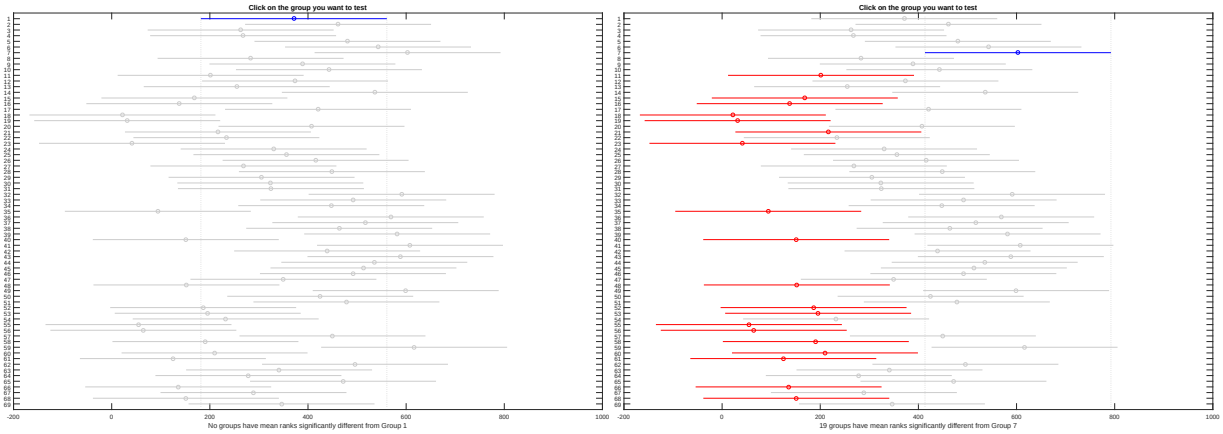
a single test, the significance level for which the null hypothesis H_0 is rejected is $\alpha = 0.05$. If 10 tests are performed, the likelihood that the tests correctly fail to reject H_0, which is true in each case, is $p_1 = 0.95^{10} = 0.5987$, and the probability that H_0 is incorrectly rejected by the null hypothesis is $p_2 = 1 - 0.5987 = 0.4013$, which is much higher than 0.05. Thus, when the number of tests increases, p_1 tends to 0, whereas p_2 tends to 1. This problem is known as a Type I Error, which represents the incorrect rejection of a true null hypothesis, also known as a "false positive". A Type II error is the error that results when the null hypothesis is falsely accepted.

The Bonferroni correction is a procedure which guards against an increase in the probability of a type I error when performing multiple significance tests with the same significance level. To maintain the probability of a type I error at some selected value α, each of the m tests to be performed is judged against a significance level of $\alpha_{bonferroni} = \alpha/m$. For a small number of simultaneous tests (up to five) this method provides a simple and acceptable answer to the problem of multiple testing. It is however highly conservative and not recommended if large numbers of tests are to be applied, when one of the many other multiple comparison procedures available are generally preferable.

6.1.3 Feature Quality Identification

The different features which were described so far are evaluated by using a tenfold cross-validation and standard machine learning approaches, implemented in Matlab. However, before sequential feature selection is used on all features, the relationship between different parameter configurations (e.g. thresholds, different distances d for which the $GLCMs$ are computed) of each feature and the corresponding feature quality is identified. Thus, statistic tests are used on every feature to determine whether it is a well or poorly defined one. Each feature vector, which contains 690 entries (10 data traces for each of the 69 textured surfaces) is subjected to a nonparametric one sample Kolmogorov-Smirnov test, which tests for the null hypothesis that the data in each feature comes from a standard normal distribution (with zero mean and unit variance). The null hypothesis is rejected at the 5% significance level for all features. Hence, a nonparametric Kruskal-Wallis test is applied to determine if there is at least one group (in this context a texture) that is significantly different to the other groups. After this analysis is performed, to determine how many groups are statistically significantly different from one another, a post-hoc statistical analysis using Bonferroni correction is conducted. For each feature, multiple pairwise comparisons of the 69 groups means are performed. Thus, each hypothesis is tested separately, using the significance level $\alpha = 0.05$. In Figure 6.1 a visual representation of the outcome of all possible pairwise comparisons is shown. In Figure 6.1a, all the p-values for the first group, which is represented by the blue bar, are $p \in \{0.4824, 0.6253, 0.8756, 1\}$. These high values indicate that there are no significantly different groups among the 69 considered. This is graphically represented through that all bars corresponding to the 69 groups overlap and all other bars, with the exception of the one corresponding to the current group,

are gray. For the seventh group, there are 19 p-values which are smaller than the initial significance level of $\alpha = 0.05$, obtaining thus 19 out of 69 statistically different groups which are marked with the red bars in Figure 6.1b. The statistical independence of the 19 groups with respect to the second one is represented through the lack of overlapping between the blue and red bars. The relationship between the distance which is used when graphically representing the significantly different groups and the corresponding p-values obeys a logarithmic rule. For example, for a p-value of 0.04 the bars corresponding to the two groups which are compared are very close to one another, whereas for a p-value of 10^{-5} they are further apart.



(a) (b)

Figure 6.1: Comparison between high p-values a) and low p-values b).

The number of comparisons n which is made is given by the number of combinations of 69 items taken two at a time, e.g. $n = C_{68}^{2} = \frac{69!}{2!(68-2)!} = 69 \times 34 = 2346$. Considering an $\alpha = 0.05$, the corrected alpha threshold $\alpha_{bonferroni} = 2.1313 \times 10^{-5}$ is obtained and the number of comparisons which are statistically significantly different for all p-Values are counted and included in the vector $\mathbf{p}$. The ratio of this sum to all possible comparisons C defines the goodness of features criterion GFC:

$$GFC = \frac{\sum_{i=1,j=1}^{69,69} \mathbb{1}_{(\mathbf{p}_{i,j} < \alpha_{bonferroni})}(\mathbf{p}_{i,j})}{C} \text{ , where } i \neq j \tag{6.6}$$

where $\mathbb{1}$ is the indicator function. Since for some features a series of parameters which can be set exists. Therefore, this criterion is applied to find the best parameter settings for each feature and to reduce the initial set for the feature selection procedure, where all features are combined with a machine learning approach.

6.2 Feature Selection for High-Dimensional Data Classification

In the last years, a considerable increase in data has been encountered, with a corresponding growth of the information which is contained within. Thus, the redundancy of data is continuously increasing, and this redundancy seems to negatively impact the performance of associated classification methods. There are two general approaches for performing dimensionality reduction and reduce data redundancy. The first one is feature extraction, which involves the transformation of the existing features into a lower dimensional space, without guaranteeing that the newly extracted features will have a physical interpretation or meaning (e.g. Linear Discriminant Analysis, which is a supervised method, Principal Component Analysis, which is an unsupervised method). The second is feature subset selection, which selects a subset of the existing features with any transformation. In the presence of categorical features, where numerical transformations are inappropriate, feature selection becomes the primary means of dimension reduction. The major motivation for the need of a small number of discriminative features is to avoid the "curse of dimensionality" and to reduce the computational burden.

To solve a classification problem in machine learning, an input x needs to be mapped to one of the finite set of class labels C. To achieve this, the classification algorithm is trained on the set of labeled training samples $I = (x_1, c_1), (x_2, c_2), \ ... \ , (x_n, c_n)$, whereas the quality of the algorithm is determined by the generalization error on a separate test set. As suggested in [ROvdS11], the features are considered as disjoint scalars (or vectors) of the input x, with the feature labels $f_i \in F$ and the feature values $f_i(x)$. One of the best methods that can be used to obtain good classification results is feature subset selection, which disconsiders irrelevant, noisy or redundant features. This approach is therefore a combinatorial optimization problem that tries to identify the subset of features that will minimize the generalization error by reducing the amount of redundant data to process.

The feature subset selection algorithm operates on a given feature set $F = \{f_i | i = 1, \ ... \ , n\}$ and finds a subset $F_M = \{f_{i_1}, f_{i_2}, \ ... \ , f_{i_M}\}$, with $M < N$ that minimizes an objective function, also called a criterion, denoted by $J(F)$ over all possible feature subsets. Commonly chosen criteria that are chosen to be minimized are the mean squared error (for regression models) and misclassification rate (for classification models). Mathematically, this is expressed as:

$$\{f_{i_1}, f_{i_2}, \ ... \ , f_{i_M}\} = \underset{M, i_M}{\operatorname{argmax}} \ F = \{f_i | i = 1, \ ... \ , n\} \tag{6.7}$$

By reducing the number of features, an improvement of the generalization capabilities is obtained and a reduced complexity is achieved, which also translates to a decreased run time.

To perform feature selection, a search strategy to select the candidate subsets, by adding or removing features, and an objective function for their evaluation have to be chosen. Since

an exhaustive comparison of the criterion value for all 2^n subsets of an n-feature data set is typically not feasible (depending on how large n is and the cost of each individual call), sequential searches move in only one direction, always growing or always shrinking the candidate set.

The objective function evaluates the candidate subsets and returns a measure of their quality. This measure can be used as a feedback signal which is used by the search strategy to select new candidates. Objective functions can be filters and wrappers. In the case of filter objective functions, feature subsets are evaluated based on their information content, which is usually measured through interclass distance (Euclidean, Mahalanobis, etc.), statistical dependence or information measures (correlation, mutual information). Wrapper objective functions are pattern classifiers, which evaluate feature subsets by considering their prediction accuracy, making use of statistical resampling or cross-validation.

One major advantage of the filter objective functions is their fast execution speed, which is due to the noniterative computations which are performed on the dataset, which usually execute much faster than training a classifier. These functions achieve also a high degree of generality, since they fundamentally evaluate the intrinsic properties of data and not their interactions with a particular classifier. On the other hand, they exhibit the tendency to select large subsets of features as the optimal solution, since most of the filter objective functions are monotonic. Thus, the user is constrained to use an arbitrary limit for the number of features to be considered.

Wrappers achieve in general higher prediction accuracies than filters, since they are fine-tuned to the interactions between the particularities of the classifier and of the dataset. Wrappers also have the ability to generalize the obtained results because of different mechanisms used to prevent overfitting, as is the case of cross-validation. The major disadvantage of wrappers is their slow execution, which is caused by the need to train a classifier for each feature subset, or several classifiers if cross-validation is used. They also lack generality, in the sense that they are interwoven with the bias of the classifier which is used in the evaluation function, the optimal feature subset which is obtained being thus specific to the classifier which is considered.

Sequential forward selection (SFS) is one of the simplest greedy search algorithms. It starts with the empty set $F_0 = \varnothing$ and sequentially adds the feature:

$$f_i^+ = \underset{f_i \notin F_k}{\operatorname{argmax}} J(F_k \cup f_i) \tag{6.8}$$

that results in the largest objective function $J(F_k \cup f_i)$, when compared with the features F_k which were already selected. After the new feature set $F_k = F_k \cup f_i^+$ is updated, the algorithm is repeated until the addition of further features does not decrease the criterion. Sequential forward selection performs best when the optimal subset has a small number of features. When the search is a few states after the empty set, a large number of states can be evaluated, whereas when the full set of features is reached, the choice is much narrower, since most of the features were already selected. The major drawback of this approach is

that once a feature is retained, it cannot be discarded, exhibiting thus a so called nesting problem.

Sequential backward selection (SBS) works in an opposite manner with respect to SFS. It starts from the full set and sequentially removes the feature:

$$f_i^- = \underset{f_i \in F_k}{\operatorname{argmax}} J(F_k \cap f_i) \tag{6.9}$$

that results in the smallest decrease in the value of the objective function $J(F_k \cap f_i)$. After updating the set of features $F_{k+1} = F_k - f_i$, these steps are repeated until the removal of further features increase the criterion. This algorithm works best when the optimal feature subset has a large number of features, because there is a great variety to choose from in terms of the next feature which is going to be eliminated. This algorithm requires more computations than SBS and it cannot reevaluate the usefulness of a feature after it was discarded.

Both these methods do not examine all possible subsets, therefore there is no guarantee that the optimal subset will be found.

As a pre-processing step, the linear correlation coefficient for all pairs of features is computed. The obtained results are shown in Table 6.1. The mapping between the shorthand form and the complete names of the features is shown in Table A.2 in Appendix A. Features F_1 and F_7 have the highest correlation among all possible combinations, with a value $c = 0.9623$, which suggests that they are redundant, meaning that no additional information would be obtained by considering both of them simultaneously. Feature F_1 is also highly correlated with F_4, F_6, F_{10}, whereas F_6 is highly correlated with F_4, F_7, F_{10}. The lowest correlation values are obtained for F_1 and F_{11}, and F_7 and F_{11}, both having a correlation coefficient of $c = -0.53$, which indicates some degree of anticorrelation.

F_1	F_2	Corr	F_1	F_2	Corr	F_1	F_2	Corr	F_1	F_2	Corr	F_1	F_2	Corr
1	2	0.055	2	4	0.069	3	7	-0.092	4	11	-0.513	6	11	-0.190
1	3	-0.132	2	5	0.029	3	8	0.129	5	6	-0.052	7	8	-0.037
1	4	0.591	2	6	0.151	3	9	-0.051	5	7	-0.428	7	9	-0.062
1	5	-0.398	2	7	0.071	3	10	0.093	5	8	-0.098	7	10	0.547
1	6	0.582	2	8	-0.024	3	11	-0.139	5	9	-0.068	7	11	-0.532
1	7	0.962	2	9	-0.061	4	5	-0.113	5	10	-0.461	8	9	0.082
1	8	-0.028	2	10	-0.019	4	6	0.572	5	11	0.588	8	10	0.025
1	9	-0.074	2	11	0.006	4	7	0.573	6	7	0.648	8	11	0.013
1	10	0.467	3	4	0.236	4	8	-0.053	6	8	-0.161	9	10	0.128
1	11	-0.526	3	5	0.108	4	9	-0.050	6	9	-0.139	9	11	-0.017
2	3	0.012	3	6	-0.003	4	10	0.231	6	10	0.449	10	11	-0.378

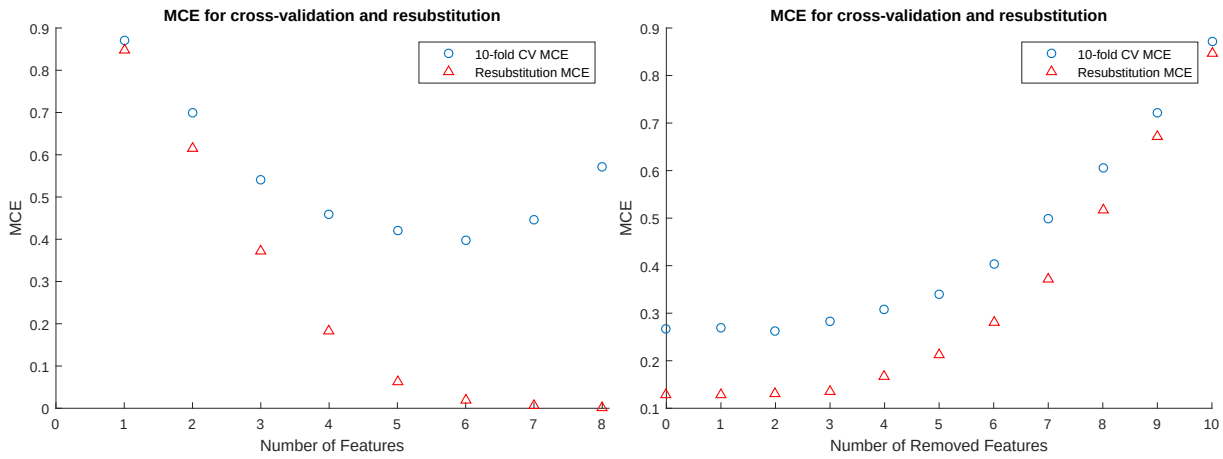
Table 6.1: Correlation coefficient for all pairs of extracted features.

In the first step SFS in a wrapper fashion is implemented to find the meaningful features. As for the classification algorithm, the Quadratic Discriminant Analysis (QDA) is chosen. Since the typical goal of classification is to minimize the misclassification error (MCE), the feature selection procedure performs a sequential search using the MCE of the learning algorithm on each candidate feature subset as the performance indicator for that subset. The training set is used to select the features and to fit the QDA model, whereas the test set is used to evaluate the performance of the final subset of features. During the feature selection procedure, to evaluate and to compare the performance of each candidate feature subset, a stratified tenfold cross-validation (CV) is applied to the training set.

The SFS algorithm stops when the first local minimum of the cross-validation MCE is found. The selected features are: F_2, F_3, F_6, F_7, F_8, F_9 and the corresponding $MCE_{CV} = 0.3971$. It is possible that the algorithm stopped prematurely. In some cases, a smaller MCE is obtainable by looking for the minimum of the cross-validation MCE over a wider range of features. This is why the entire set of 11 features was considered. However, after adding 8 features, leaving out only F_1, F_5 and F_{10}, the algorithm cannot proceed any further because the covariance matrix of each group in the training set must be positive definite. These features were removed one-by-one and in pairs and the algorithm was rerun with one or two feature less, but the problem still persisted. This can be because QDA works under the assumption that the data is represented by a multivariate probability distribution, which always has a positive definite covariance matrix unless one or more variables are exact linear combinations of the others. Thus, all these three features seem to have within their structures values which are linear combinations of others and cause the covariance matrix to be singular. The other possibility could be because of trying to fit a model with too many predictor variables and too few observations. However, this is not the case, since a set 690 observations and 11 features are considered. This is why the cross-validation MCE is plotted as a function of the number of features, up to a maximum of 8, as shown in Figure 6.2a.

The cross-validation MCE reaches the minimum value when 6 features are used and it increases when more than 7 features are used, which means overfitting occurs here. Thus, the minimum of the cross-validated MCE which was initially obtained is also the single minimum of the objective function.

It is also interesting to look at the plot of the resubstitution MCE values on the training set (i.e., without performing cross-validation during the feature selection procedure) as a function of the number of features, as shown marked with the red triangles in Figure 6.2a. Resubstitution means that the original data is passed to the QDA classifier as both the training and test data to compute the criterion. This violates the principle that the test set has the same format as the training set. If the test set is reused as a subset of the training set, then a model that simply memorized its input, without learning how to generalize to new examples, would receive misleadingly high classification accuracies. This is highlighted through the overoptimistic MCE values which are obtained. They are smaller than the cross-validation MCE values, and the MCE goes to zero when 8 features are used. This



(a) (b)

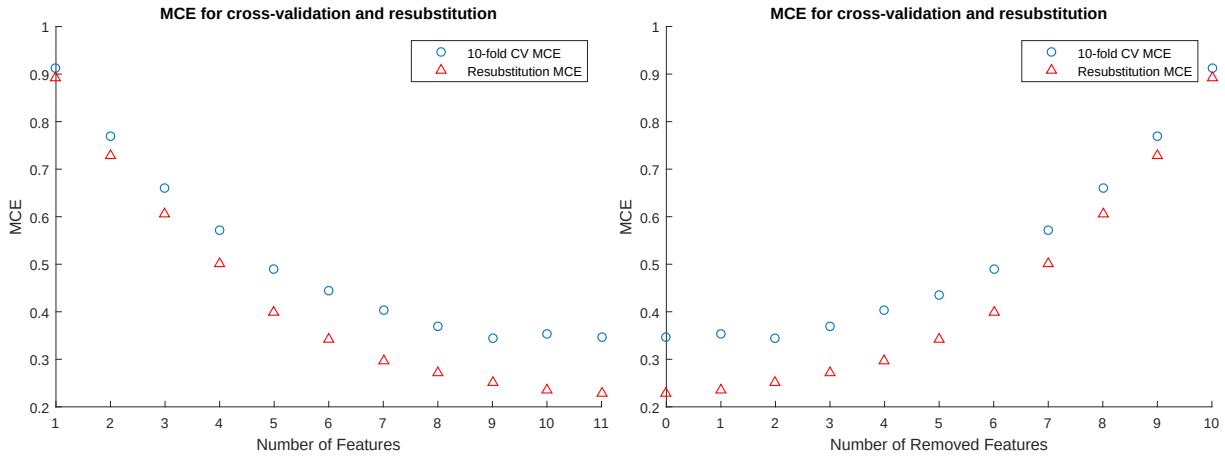
Figure 6.2: MCE with cross-validation and resubstitution. a) Sequential forward selection using the QDA classifier. b) Sequential backward selection using the $DQDA$ classifier.

indicates that the resubstitution error generally is not a good performance estimate for evaluating and selecting features. Therefore, the resubstitution error needs to be avoided to be used, not only during the final evaluation step, but also during the feature selection procedure.

To be able to perform SBS, considering that not all covariance matrices are nonsingular for all features, the Diagonal Quadratic Discriminant Analysis ($DQDA$) classifier is used. This belongs to the family of Naive Bayes classifiers which considers multivariate normal distributions for the variables within each class. As described in [DFS02], the $DQDA$ classifier is a modification of the QDA classifier, where the off-diagonal elements of the covariance matrix of each class are assumed to be zero. In other words, the features are assumed to be uncorrelated. This, together with the assumption that each feature has a multivariate normal distribution implies the assumption that the features are independent. The last assumption is a notable attribute of the family of Naive Bayes classifiers, which are fast and have fewer parameters to estimate, especially when the number of features is quite large.

When the sequential backward selection algorithm is run, considering initially the whole set of features, the algorithm stops after one iteration. Only F_1 is removed, which does not cause any decrease in the MCE value, which has a value of $MCE = 0.2681$. The main reason for this is that the correlation coefficient between F_1 and F_7 is close to unity, which means that the two features are redundant, so the elimination of one of them from the feature set does not degrade the MCE. To avoid overfitting, the SBS $DQDA$ algorithm is run until 6 features remain in the subset, this being F_2, F_3, F_4, F_7, F_8, F_9, for which a $MCE = 0.34$ is obtained, which is lower that the one obtained with SFS. The SBS algorithm with resubstitution was also considered, which did not eliminate any features

before stopping, yielding a MCE value of $MCE = 0.1275$. This is as expected a much smaller value than in the cross-validation case because the whole set of images is used for training, the model having thus much more data to learn from. The test set is a subset of the training set, this being the second reason why the MCE is overoptimistic, representing actually the lowest limit for the MCE error that can be obtained when giving the classifier an unknown input. To evaluate how the MCE error varies as a function of the number of removed features, the SBS algorithm was rerun, removing one feature at a time until only one remains. As shown in Figure 6.2b, the red curve, which corresponds to the resubstitution MCE is always below the blue one, which corresponds to the tenfold cross validated MCE. The two are far from one another at the beginning, when a few features are removed, and they almost superimpose when 9 and 10 features are removed from the initial set. The only feature which remains in the cross-validated SBS approach is F_7 with a corresponding $MCE = 0.871$, which is also the only one which remains in the final set for the the resubstitution approach, where a $MCE = 0.846$ is achieved.



(a) (b)

Figure 6.3: MCE with cross-validation and resubstitution. a) Sequential forward selection using the LDA classifier. b) Sequential backward selection using the LDA classifier.

Linear Discriminant Analysis (LDA) is another commonly used dimensionality reduction technique which is usually used as a pre-processing step for pattern recognition and machine learning applications. Its goal is to find a projection of the starting feature space (which is a dataset of n-dimensional samples) onto a lower-dimensional subspace (with $k \leq n - 1$ dimensions) which achieves a good class-separability, minimizing thus the error in parameter estimation. In this way, the "curse of dimensionality" problem, which concerns the model overfitting, is mitigated.

LDA is similar to Principal Component Analysis (PCA), which is also a linear transformation technique that is commonly used for dimensionality reduction. PCA is an unsupervised algorithm, because it ignores the class labels, its main goal being to find the directions (principal components) that maximize the variance in a dataset. Unlike PCA,

LDA is a supervised algorithm which computes the directions (linear discriminants) that will maximize the separation between multiple classes. LDA finds a linear combination of features that best separates the classes in the input dataset. The fundamental assumptions which the LDA method makes is that the measurements of each class are statistically independent, normally distributed and that each class has identical covariance matrices.

As shown in Figure 6.3a, for the $SFS\ LDA$ classifier a minimum $MCE = 0.3449$ is obtained, considering all features except F_{11} and F_5. Their addition to the feature set causes the error to increase to $MCE = 0.3463$. The LDA classifier with resubstitution has a purely monotonic descending behavior, the minimum $MCE = 0.2289$ being obtained when all features are considered. In Figure 6.3b the results obtained with the cross-validated and resubstitution $SBS\ LDA$ algorithm are plotted. The two curves, corresponding to the two methods of validation, start with a discrepancy of about 12% in terms of MCE in the favor of the LDA classifier with resubstitution, which achieves a $MCE = 0.229$. As more features are removed from the feature set the two curves start getting closer to one another, reaching almost the same MCE with only one feature. Considering a subset of only of 6 features (it has was previously concluded that from 7 features onwards overfitting occurs), the $SBS\ LDA$ classifier obtains a $MCE = 0.4362$ with features F_3, F_4, F_7, F_8, F_9, F_{10}, whereas the $SFS\ LDA$ classifier obtains a $MCE = 0.4435$, with almost the same subset as the SBS version, where feature F_7 takes the place of feature F_1. This resubstitution is due to the high degree of correlation which exists between two features.

Comparing the MCE for both SBS and SFS using either LDA or QDA, it follows that $MCE_{QDA} < MCE_{LDA}$. This is expected, since the quadratic classifier is a more general version of the linear classifier, separating the measurements of two or more classes of objects or events through a more complex surface, namely a quadric one, which is a n-dimensional hypersurface in a n+1 dimensional space. Because this classifier does not assume the classes to be independent and normally distributed (for the LDA classifier these assumptions are violated), the lower MCE values are obtained. These are not valid assumptions for the dataset considered in this work, as it is indicated by the feature correlation coefficients and the Kolmogorov-Smirnov test.

6.3 Classification Accuracy

The different features which were discussed so far are evaluated using tenfold cross-validation and a standard machine learning approach, which are implemented in Matlab.

A naive Bayes classifier is used for the classification task. Naive Bayes classifiers are supervised learning linear classifiers, whose underlying probabilistic model is based on Bayes' theorem, which states:

$$P(\omega_j|\mathbf{x}_i) = \frac{P(\mathbf{x}_i|\omega_j)P(\omega_j)}{P(\mathbf{x}_i)} \tag{6.10}$$

where ω_j is the class j label, $j \in \{1, 2, \ ... \ , m\}$, $P(\omega_j)$ is the prior probability, $P(\omega_j|\mathbf{x}_i)$ is the posterior probability and $P(\mathbf{x}_i|\omega_j)$ is the conditional probability of observing sample $\mathbf{x}_i$ given that it belongs to class ω_j. The objective function in the naive Bayes classifier is to maximize the posterior probability given the training data to formulate the decision rule:

$$\omega_j = \underset{j=1, \ ... \ ,M}{\operatorname{argmax}} P(\omega_j|\mathbf{x}_i) \tag{6.11}$$

One assumption of this classifiers is that the features in a dataset are independent and identically distributed, which means that they are random variables that are mutually independent and have the same probability distribution. The adjective *naive* is due to the second assumption, which is represented by the conditional independence of features. As detailed in [Ras14], this means that the class conditional probabilities of the samples can be estimated from the training data. Thus, for an n-dimensional feature vector $\mathbf{x}$, the class-conditional probability is represented by:

$$P(\mathbf{x}|\omega_j) = P(x_1|\omega_j)P(x_2|\omega_j) \ ... \ P(x_n|\omega_j) = \prod_{i=1}^{n} P(x_i|\omega_j) \tag{6.12}$$

As it is mentioned in [Ris01], even if the conditional independence assumption is very often violated in practice, naive Bayes classifiers still tend to perform very well under these circumstances. However, when the independence assumptions are strongly violated and when nonlinear classification problems need to be approached, naive Bayes classifiers can perform poorly. The classification model to be chosen is dependent on the type of data and the problem which needs to be solved. This is the main reason why it is always recommended to compare different classification models on the dataset and consider the prediction performance and computational efficiency for each considered approach.

There are two commonly used approaches in machine learning. On one hand is supervised learning, where a function is inferred from labeled training data. The training data is represented by the pair formed from the feature vector and the desired output value, which is then analyzed by the learning algorithm which infers a function, which can be used for labeling new, unseen data. On the other hand is unsupervised learning, which tries to find hidden structures in unlabeled data. For this approach, no error measurement exists, since the input to the learning algorithm is unlabeled.

The tenfold cross-validated naive Bayes classifier from [Fla12] is used on the instance space, where all data is numerical, supervised (the true texture names serve as labels) and normalized in the interval [0,1]. Since the feature space has 11 dimensions, a fully exhaustive search of the optimum instance configuration in terms of the MCE is still feasible, the number of cases to be tried for being $2^{11} - 1 = 2047$. When a selection of features is used as an input feature space, a confusion matrix is obtained. This method is often used in supervised learning to describe both visually and numerically the performance of a classifier on a set of test data, for which the true values (or labels) are known. Each column of the matrix represents the instances in the actual class while each row represents

the instances in a predicted class. All correct guesses are located on the diagonal of the matrix, being thus easy to visually inspect the table for errors, as they are represented by off-diagonal elements. Therefore, the definition of the classification accuracy (CA) is based on the resulting confusion matrix $\mathbf{M}_{ConfMat}$, as follows:

$$CA = \frac{\sum_{k=1}^{69} \mathbf{M}_{ConfMat}(k,k)}{N} \text{ , where } N = 69 \cdot 10 \tag{6.13}$$

This equation sums up the number correctly classified surfaces on the main diagonal of $\mathbf{M}_{ConfMat}$ as a fraction of all the entries in $\mathbf{M}_{ConfMat}$ and measures the performance for all the selected input feature set on the considered database.

6.3.1 Original Database

The highest accuracy values which are obtained, as a function of the number of elements of the feature set, are shown in Table 6.2.

Size of feature set	Selected features	Classification accuracy [%]
1	F_7	12.8986
2	F_3,F_7	27.8261
3	F_3,F_7,F_8	39.4203
4	F_3,F_7,F_8,F_9	50.1449
5	F_3,F_4,F_7,F_8,F_9	59.7101
6	F_2,F_3,F_4,F_7,F_8,F_9	65.9420
7	F_2,F_3,F_4,F_6,F_7,F_8,F_9	69.2754
8	F_2,F_3,F_4,F_6,F_7,F_8,F_9,F_{10}	71.7391
9	F_2,F_3,F_4,F_5,F_6,F_7,F_8,F_9,F_{10}	73.7681
10	F_2,F_3,F_4,F_5,F_6,F_7,F_8,F_9,F_{10},F_{11}	73.0435
11	F_1,F_2,F_3,F_4,F_5,F_6,F_7,F_8,F_9,F_{10},F_{11}	73.1884

Table 6.2: Classification accuracy for different feature set sizes, considering the original image database.

The data in Table 6.2 show that major increases in terms of classification accuracy are achieved when considering an increasing number of features, up to a maximum number of six. This is in perfect accordance to the previous observation which was made when SFS analysis showed that with more than 6 features, overfitting occurs. Another interesting aspect which is worth mentioning is that the feature set which is selected by the fully exhaustive search coincides with the one obtained by the SBS $DQDA$ algorithm. Thus, this algorithm manages to select the optimum combination of 6 features to obtain the maximum classification accuracy with a minimal computational time, since it is not a fully exhaustive search technique.

When a single feature is used, the highest classification accuracy is achieved for the edginess feature, which has the highest value of the GFC metric among all other features, as it is shown in Table 6.2 and in Table A.1 from Annex A. However, when two or more features are combined, the highest classification accuracy does not necessarily occur when combining the best features in terms of the GFC metric. For example, when two features are considered, the highest classification accuracy is obtained for coarseness and edginess (F_3 and F_7) and not for complexity and edginess (F_1 and F_7), which have the two highest GFC values. This suggests that synergies among features occur, the best classification accuracy being thus not obtained when combining only features which have individually high discriminating power.

The elements in the confusion matrix shown in Figure 6.4 are represented both numerically and visually, with different gray levels. The elements on the main diagonal have predominantly dark shades of gray, without large discontinuities in terms of successfully classified textures, which means that the classifier works reasonably well for all inputs. All off-diagonal elements represent the misclassified textures. On the X and Y axis the true

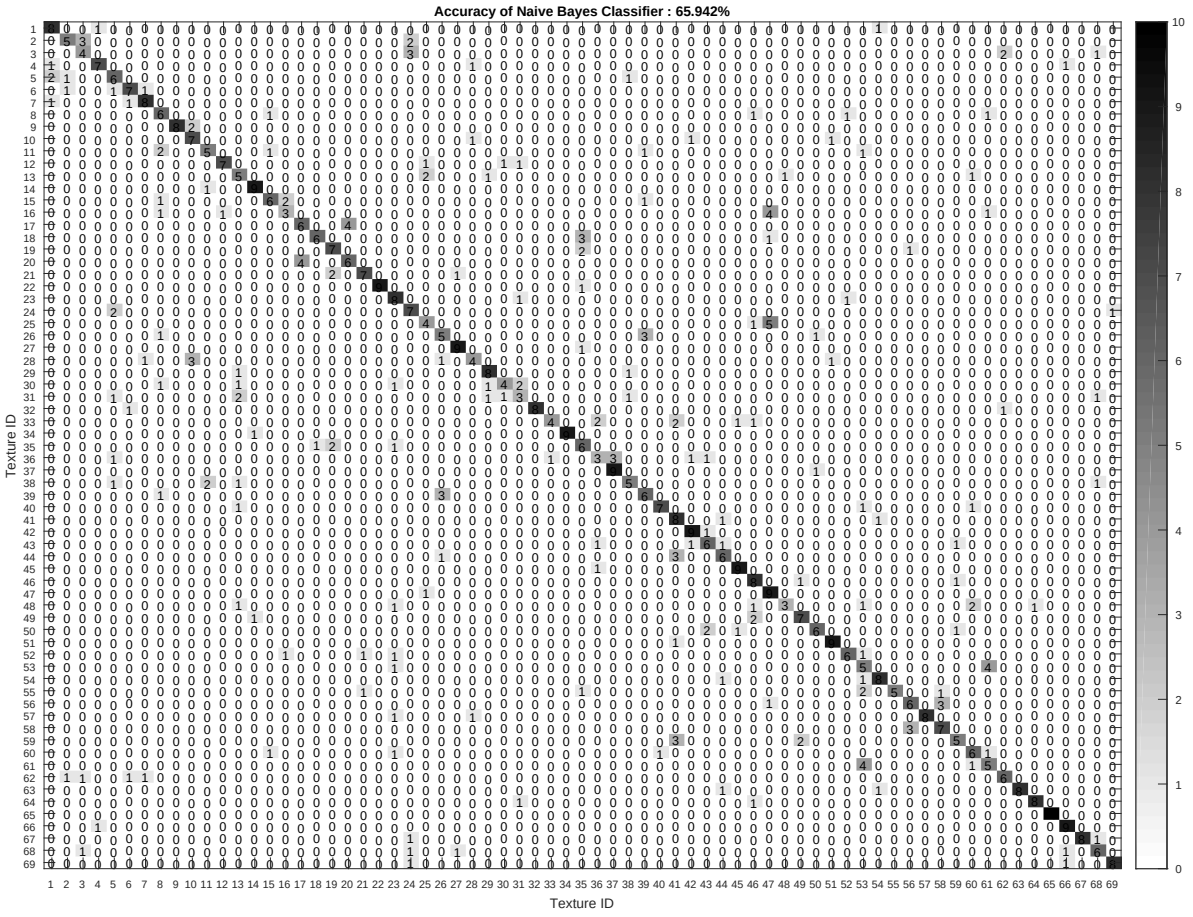
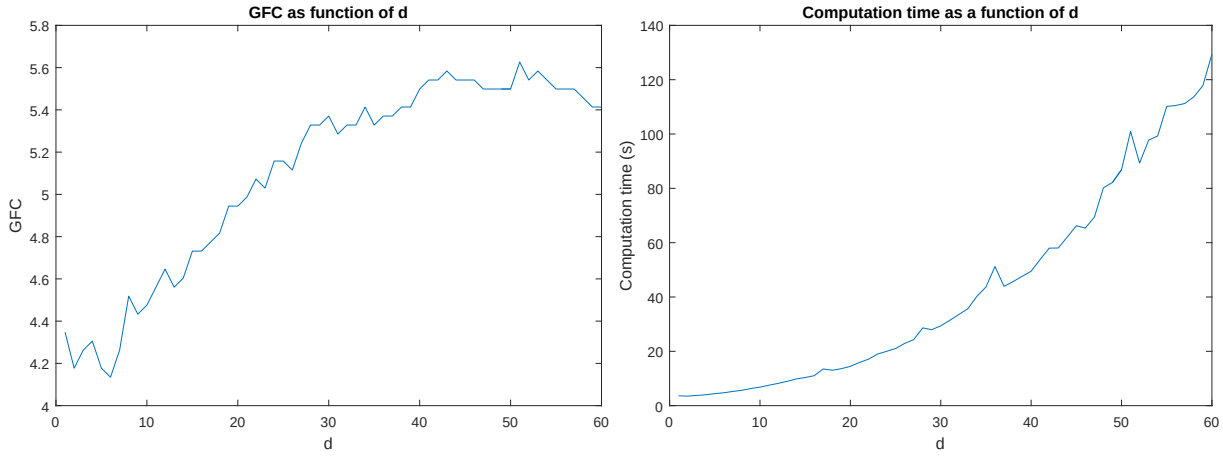


Figure 6.4: Confusion matrix for the 6 feature set, considering the original image database.

texture number and the predicted texture number are shown. The off-diagonal elements which have nonzero values are randomly distributed, which shows that the classifier does not consistently misclassify certain textures. This is always the desired behavior, since the classifier has to be as generic as possible and be able to successfully detect a wider range of inputs, and not be fine-tuned to only certain ones.

6.3.2 Magnified Database

For the magnified database of images the same sequence of steps is followed as previously described. All feature settings, e.g. thresholds, search areas, which are used for the initial set are also considered in this case, with the exception of the complexity feature. This feature heavily relies on the average gray level of the surrounding neighborhood of the current pixel. In the case of magnified texture images, the information content is spread over a wider spatial range than in the original database and its redundancy is lower, due to the the smaller distance the images are taken from. The same range is considered for d as for the original database, namely $d \leq 60$.



(a) (b)

Figure 6.5: a) GFC as a function of the range d. b) Computation time per image as a function of the range d for the magnified database.

As shown in Figure 6.5, the GFC metric values increase as the search range is increased, and from $d = 58$ onwards it starts to decrease. To be consistent with the complexity feature for the normal database of images, the trade-off between computation time and the GFC value was also considered in this case, taking as an additional constraint the computation time, which needs to be similar for the two databases. For the magnified database, the largest two GFC values occur for for $d = 43$ and $d = 51$, for which the corresponding computational time is $t_{43} = 58s$ and $t_{51} = 101s$. Since the GFC values corresponding to the two ranges are very similar, whereas the computation time almost doubles, $d = 43$ was

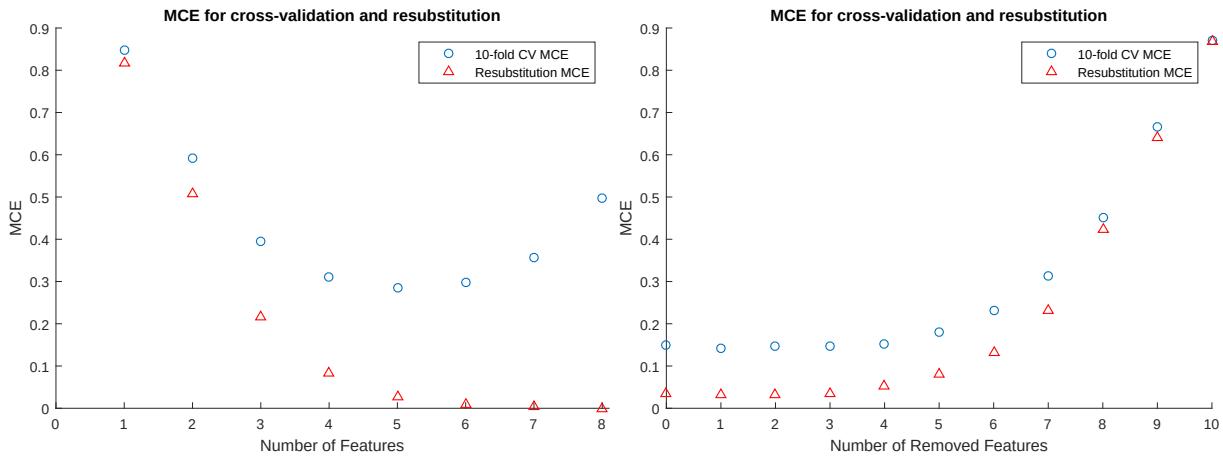
chosen for the complexity feature. In this way, the computation time is also consistent for the two image databases.

The linear correlation coefficient for all pairs of features is computed. The obtained results are shown in Table 6.3. The mapping between the shorthand form and the complete names of the features is shown in Table A.2 in Appendix A. The highest correlation among all possible feature pairs is obtained for features F_1 and F_7 with a correlation coefficient $c = 0.9784$, which means that the two features are redundant with respect to one another. Feature F_5 is also highly correlated with F_6 and F_{11}. There is no correlation between features F_6, F_7, F_8 and F_9, whereas a high degree of anticorrelation occurs for features F_1, F_4, F_7 and F_{11}, all with a correlation coefficient $c \leq -0.60$.

F_1	F_2	Corr	F_1	F_2	Corr	F_1	F_2	Corr	F_1	F_2	Corr	F_1	F_2	Corr
1	2	-0.219	2	4	-0.143	3	7	-0.259	4	11	-0.693	6	11	0.248
1	3	-0.187	2	5	-0.098	3	8	-0.087	5	6	-0.869	7	8	0.250
1	4	0.481	2	6	-0.086	3	9	-0.103	5	7	-0.524	7	9	0.008
1	5	-0.511	2	7	-0.218	3	10	-0.087	5	8	-0.145	7	10	0.310
1	6	-0.253	2	8	-0.047	3	11	0.001	5	9	-0.026	7	11	-0.585
1	7	0.978	2	9	-0.098	4	5	-0.318	5	10	-0.167	8	9	0.009
1	8	0.249	2	10	-0.065	4	6	-0.068	5	11	0.551	8	10	0.161
1	9	0.031	2	11	0.044	4	7	0.470	6	7	-0.287	8	11	-0.145
1	10	0.339	3	4	0.208	4	8	-0.025	6	8	-0.089	9	10	-0.108
1	11	-0.624	3	5	0.411	4	9	-0.180	6	9	0.009	9	11	0.062
2	3	-0.152	3	6	0.429	4	10	0.023	6	10	-0.006	10	11	-0.219

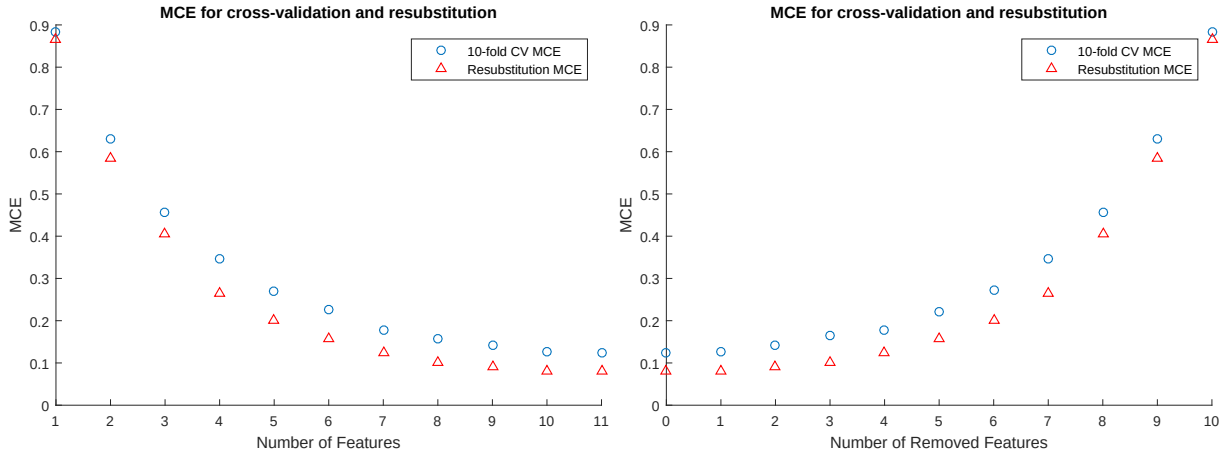
Table 6.3: Correlation coefficient for all pairs of extracted features for the magnified database.

Sequential feature selection is performed to select the most relevant features among the 11 which are considered. Firstly, $SFS\ QDA$ is performed until a set 8 features is obtained, any additional feature which could be added representing a linear combination of the already existing ones. In Figure 6.6a, the MCE as a function of the number of selected features is shown. The cross-validated MCE decreases until 6 features are present in the set and afterwards it starts to increase, indicating thus that overfitting occurs when more than 6 features are considered. The six features which are selected by the $SFS\ QDA$ algorithm are F_2, F_3, F_4, F_6, F_7, F_9 and the corresponding tenfold cross validated classification error is $MCE = 0.2971$. In Figure 6.6b, the results obtained by performing $SBS\ DQDA$ classification algorithm are shown. $DQDA$ is used because this algorithm does not require the covariance matrix of each feature to be positive definite. With a set of 6 features, namely F_1, F_2, F_3, F_6, F_8, F_9, a tenfold cross-validated error of $MCE = 0.17971$ is obtained. Both values are much smaller than the corresponding ones for the initial image database, indicating that the classification accuracy increases for the magnified image database, when the same number of features is considered.



(a) (b)

Figure 6.6: MCE with cross-validation and resubstitution for the magnified database. a) Sequential forward selection using the QDA classifier. b) Sequential backward selection using the $DQDA$ classifier.



(a) (b)

Figure 6.7: MCE with cross-validation and resubstitution for the magnified database. a) Sequential forward selection using the QDA classifier. b) Sequential backward selection using the $DQDA$ classifier.

Similarly, $SFS\ LDA$ algorithm is considered and the results which have been obtained are shown in Figure 6.7. A $MCE = 0.2275$ is obtained when 6 features are chosen, namely F_2, F_3, F_7, F_8, F_9, F_{10}. Afterwards, $SBS\ LDA$ is considered and a $MCE = 0.2217$ is obtained using features F_2, F_3, F_4, F_7, F_9, F_{10}. The obtained misclassification errors are two times smaller than the ones corresponding to the initial image database.

A tenfold cross validated naive Bayes classifier is implemented as in the original database

scenario, since also 10 images per texture are taken. All possible feature combinations are tested and the corresponding classification accuracy is computed. The best classification accuracies for different feature sizes are shown in Table 6.4.

Size of feature set	Selected features	Classification accuracy [%]
1	F_7	15.3623
2	F_3,F_7	37.8261
3	F_3,F_7,F_9	56.6667
4	F_1,F_2,F_3,F_9	68.6957
5	F_2,F_3,F_4,F_7,F_9	77.1014
6	F_2,F_3,F_4,F_7,F_8,F_9	82.4638
7	F_2,F_3,F_4,F_6,F_7,F_8,F_9	85.0725
8	F_2,F_3,F_4,F_6,F_7,F_8,F_9,F_{10}	85.2174
9	F_1,F_2,F_3,F_6,F_7,F_8,F_9,F_{10},F_{11}	84.9275
10	F_1,F_2,F_3,F_4,F_6,F_7,F_8,F_9,F_{10},F_{11}	85.7971
11	F_1,F_2,F_3,F_4,F_5,F_6,F_7,F_8,F_9,F_{10},F_{11}	85.0725

Table 6.4: Classification accuracy for different feature set sizes, considering the magnified image database.

All the classification accuracy values which are obtained in this case are considerably higher than the corresponding ones for the original database. One interesting observation that can be done is that features F_2, F_3, F_4, F_7, F_8, F_9 achieve the highest classification accuracy among all possible sets of six features in both considered cases, a classification accuracy of 82.46% is obtained when considering the magnified database, whereas a classification accuracy of only 65.94% is obtained when considering the original one. The increased classification accuracy achieved in the case of the features extracted from the magnified database of images is due to the increased values of the GFC metric for most of the features, as it is shown in Table A.1. The additional information which is present in the magnified images increases also the discriminating power of features, allowing for better distinctions to be made among textures. The elements in the confusion matrix shown in Figure 6.8 are represented both numerically and visually, with different gray levels which correspond to different levels of correctly identified textures. A low number is visually represented by light shades, whereas large numbers are represented by dark shades. The sum of the values on each row and column equals to 10, which represents the number of images per texture. In the ideal case, the main diagonal will be populated only with values of 10, which means a classification accuracy of 100%. The elements on the main diagonal have predominantly dark shades of gray and the values are consistent, without any discontinuities, which means that the classifier works well for all inputs. All off-diagonal elements represent the misclassified textures. The off-diagonal elements which have nonzero values are randomly distributed, which shows that the classifier does not consistently misclassify certain textures.

Comparing the confusion matrix shown in Figure 6.8 with the one shown in Figure 6.4,

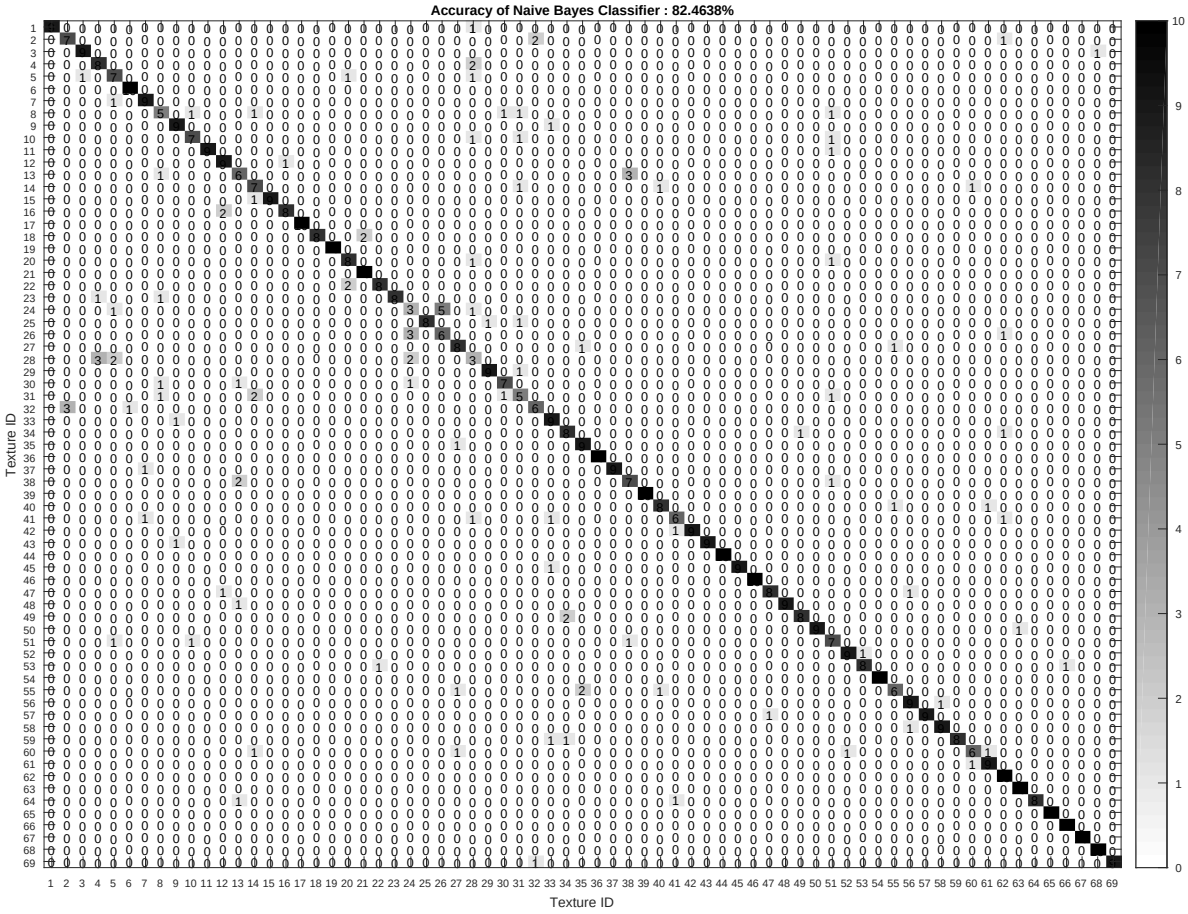


Figure 6.8: Confusion matrix for the 6 feature set, considering the magnified image database.

it follows that the discontinuities which occur on the main diagonal, which are visually represented by significantly different shades of gray for neighboring values, are much more frequent in the second case than in the first one. This is confirmed by the classification accuracy of the naive Bayes classifier, which is greater when using as inputs the features extracted from the magnified image database than from the original one.

6.4 Discussion

As it is shown in Figure 6.8, the off-diagonal elements correspond to the textures which are mistakenly classified. For example, textures 4, 5 and 28, which correspond to *Plastic Mesh*, *Rhomb Aluminium Mesh* and *Profiled Wood* are confused with one another in 50% of the cases. This is because all these textures are coarse and exhibit a similar degree of roughness. Textures 24 and 26, which correspond to *Compressed Wood*

and *Cork*, are also confused in 50% of the cases, since both have the same dominant color and similar values in terms of contrast and roughness, which matches the visual perception. Textures 13 and 38, corresponding to *Stone Tile* and *Fibers* are mistakenly classified in 30% of the cases, because they have the same dominant color and very similar values in terms of the regularity feature. Textures 2 and 32, which are the IDs for *Isolating Foil Mesh* and *Fine Rubber* textures, are also confused in 30% of the cases, the reason for this being the similar values which they exhibit in terms of color, contrast and roughness.

Considering these observations, it can be concluded that even the confused textures are similar to one another to some extent. The similarity of these textures is represented by at least 3 features which have similar values for the textures under observation. Taking into account that only 6 features have been considered to obtain the confusion matrix shown in Figure 6.8, it follows that the textures which are mistakenly classified for others have, on average, very similar values for 50% of the considered features.

Chapter 7

Conclusion

A set of eleven touch-relevant features, which are extracted from images, is discussed. The prediction accuracy of a cross-validated naive Bayes classifier which takes as input this set of features is presented.

A series of haptically-relevant parameters, e.g. coarseness, contrast, directionality, line-likeness, are implemented as suggested. Moreover, their perceptual relevance and behavior is investigated from a machine learning point of view. Their correlation with the human perception is not very high and the discriminating power which they exhibit is reduced.

To obtain features which can successfully distinguish among different textures and be also perceptually accurate, improvements to the original features are suggested and implemented and a series of new features are defined, e.g. edginess, roughness, glossiness, softness and color distance. A subjective experiment for the roughness feature is conducted, which consists of two parts, namely an image and a touch based one. Its outcome is taken as a reference for the roughness feature, which tries to obtain a similar average ordering as the human subjects by considering the images of the textures which have been used in the experiment. A second noteworthy observation of the subjective experiment is that the two average orderings obtained after considering the outcomes of each part of the experiment have a matching accuracy of $a = 0.4638$. This indicates that that a positive correlation between the visual and tactile ordering exists. However, a considerable number of mismatches occur, which shows that the information extracted from images is not sufficient to successfully recognize the textures. Therefore, such that this kind of limitations are avoided, more sources of information need to be used.

Two image databases have been created to test the relevance of the defined features. Each database consists of 690 images, where 10 images have been taken for each one of the 69 considered textures. All images have been captured with a common smartphone, with and without a magnifying glass attached to its camera, obtaining thus the two image databases. Once all features are computed by using these databases, feature selection is performed to reduce dimensionality, which is a necessary pre-processing step for any machine-learning

approach. Both forward and backward sequential feature selection with quadratic and linear discriminant analysis classifiers are considered and compared.

Finally, a cross-validated naive Bayes classifier is created based on the existing feature set and the obtained prediction accuracy is discussed. For the original image database a maximum classification accuracy of 73.18% is achieved when all defined features are considered. To avoid overfitting, only the best six features are selected, a classification accuracy of 65.94% being thus obtained. The suggested features perform better on the magnified version of the images, since more information is available with less redundancy. Therefore, a maximum classification accuracy of 85.79% is obtained when 10 features are selected, and when only 6 are considered, a classification accuracy of 82.46% is achieved.

Even if the results presented in this work have demonstrated the effectiveness of the defined features in correctly identifying different textures based only on their images, additional developments can be considered. As it was already mentioned, the amount of information which is present in the magnified image database is greater than in the original one, which is translated in an increased value for the GFC metric for most of the defined features. The first exception is the regularity feature, whose definition is heavily dependent on periodicity. When images of textures are taken from a smaller distance, the periodicity of the texture elements, if given, is emphasized to a lesser extent. This is why the ability of the defined regularity feature to distinguish among textures slightly decreases, which is also shown by the smaller value of the GFC metric for the magnified database than in the original case. One improvement that can be done in this direction is to define a method of detecting regularity which is not dependent on the amount of redundant information which is present within an image. The softness feature also has an unexpected behavior when it comes to the magnified database. The first reason for this is that the foam detection part of the algorithm searches for circles with radii which fall within a specified range. When the distance the images are taken from decreases, as is the case for the magnified image database, the circles appear to be larger and therefore their radius increases as well. Similarly, the hair detection algorithm performs different stages of area filtering to isolate the fibers. Since the hairs appear to be larger in size, and therefore with a larger area than in the initial case, the algorithm misleadingly classifies different objects as fibers. To counteract this behavior, an improved version of the algorithm needs to be developed which bases its decisions on adaptive thresholds, which are determined based on the distance the images have been captured from with respect to the object surface.

Appendix A

Additional Plots and Tables

Feature	Original dataset GFC [%]	Magnified dataset GFC [%]
Coarseness	2.7280	3.4953
Color difference	1.3214	4.0921
Complexity	4.3478	5.6266
Contrast	3.4953	4.1773
Directionality	0.8951	4.0921
Edginess	5.0725	5.7118
Glossiness	2.4723	3.3248
Linelikeness	3.2822	3.6658
Regularity	3.6658	3.3674
Roughness	1.7477	4.7315
Softness	3.0691	1.8329

Table A.1: The GFC metric values for all defined features for the original and magnified dataset.

F1	F_{com}	F5	ID	F9	IR
F2	ICD	F6	IL	F10	IS
F3	F_{crs}	F7	IE	F11	IG
F4	F_{con}	F8	$IREG$		

Table A.2: Mapping between shorthand and full feature names. The features whose names start with F are computed exactly as suggested in related work, whereas the ones which start with I are either modified versions of the original features or are newly-defined.

Texture ID	Texture Name
1	'G1EpoxyRasterPlate'
2	'G1IsolatingFoilMesh'
3	'G1PlasticMeshVersion1'
4	'G1PlasticMeshVersion2'
5	'G1RhombAluminumMesh'
6	'G1SquaredAluminumMesh'
7	'G1TextileMeshVersion1'
8	'G2Brick'
9	'G2CrushedRock'
10	'G2GraniteTypeVeneziano'
11	'G2Marble'
12	'G2RoofTile'
13	'G2StoneTileVersion1'
14	'G2StoneTileVersion2'
15	'G2StoneTileVersion3'
16	'G2StoneTypeTerraCotta'
17	'G3Brass'
18	'G3CeramicPlate'
19	'G3CeramicTile'
20	'G3Copper'
21	'G3Glass'
22	'G3StainlessSteel'
23	'G4Beech'
24	'G4CompressedWoodVersion1'
25	'G4CompressedWoodVersion2'
26	'G4Cork'
27	'G4LaminatedWood'
28	'G4ProfiledWoodPlate'
29	'G4Teak'
30	'G4WoodTypeSilverOak'
31	'G4WoodenPaper'
32	'G5FineRubber'
33	'G5ProfiledRubberPlate'
34	'G5SolidRubberPlateVersion1'
35	'G5SolidRubberPlateVersion2'
36	'G6Carpet'
37	'G6CoarseArtificialGrassFibers'
38	'G6Fibers'
39	'G6FineArtificialGrassFibers'
40	'G6IsolatingFoilVersion1'
41	'G6IsolatingFoilVersion2'
42	'G6SteelWool'
43	'G6Towel'
44	'G7BumpyFoam'
45	'G7CoarseFoam'
46	'G7FineFoamVersion1'
47	'G7FineFoamVersion2'
48	'G7FoamFoilVersion1'
49	'G7FoamPlate'
50	'G7MediumFoam'
51	'G7ProfiledFoam'
52	'G7StyrofoamVersion1'
53	'G7StyrofoamVersion2'
54	'G8BubbleFoil'
55	'G8Cardboard'
56	'G8Paper'
57	'G8PlasticFoilVersion1'
58	'G8PlasticFoilVersion2'
59	'G8WallpaperVersion1'
60	'G8WallpaperVersion2'
61	'G9FloorCloth'
62	'G9Jeans'
63	'G9Kashmir'
64	'G9Leather'
65	'G9TableClothVersion1'
66	'G9TableClothVersion2'
67	'G9TextileVersion1'
68	'G9TextileVersion2'
69	'G9TextileVersion3'

Table A.3: List of names for all 69 surfaces, corresponding to their texture ID in the textured surface database (http://www.lmt.ei.tum.de/texture/).

List of Figures

List of Tables

Bibliography

[AK89] Moses Amadasun and Robert King. Textural features corresponding to textural properties. *Systems, Man and Cybernetics, IEEE Transactions on*, 19(5):1264–1274, September 1989.

[BM88] Kevin P Balanda and HL MacGillivray. Kurtosis: a critical review. *The American Statistician*, 42(2):111–119, May 1988.

[Bro66] Phil Brodatz. *Textures: a photographic album for artists and designers*. Dover Pubns, August 1966.

[DFS02] Sandrine Dudoit, Jane Fridlyand, and Terence P Speed. Comparison of discrimination methods for the classification of tumors using gene expression data. *Journal of the American statistical association*, 97(457):77–87, March 2002.

[ES10] Brian S Everitt and Anders Skrondal. The cambridge dictionary of statistics. *Cambridge: Cambridge*, October 2010.

[Fla12] Peter Flach. *Machine learning: the art and science of algorithms that make sense of data*. Cambridge University Press, November 2012.

[Gei93] Seymour Geisser. *Predictive inference*, volume 55. CRC Press, June 1993.

[Har79] Robert M Haralick. Statistical and structural approaches to texture. *Proceedings of the IEEE*, 67(5):786–804, May 1979.

[HSD73] Robert M Haralick, Karthikeyan Shanmugam, and Its' Hak Dinstein. Textural features for image classification. *Systems, Man and Cybernetics, IEEE Transactions on*, (6):610–621, November 1973.

[HSG03] Yvette Hatwell, Arlette Streri, and Edouard Gentaz. *Touching for knowing: cognitive psychology of haptic manual perception*, volume 53. John Benjamins Publishing, September 2003.

[HZLL05] Xing-Jian He, Yue Zhang, Tat-Ming Lok, and Michael R Lyu. A new feature of uniformity of image texture directions coinciding with the human eyes perception. In *Fuzzy Systems and Knowledge Discovery*, pages 727–730. Springer, August 2005.

[IZL08] Md Monirul Islam, Dengsheng Zhang, and Guojun Lu. A geometric method to compute directionality features for texture images. In *Multimedia and Expo, 2008 IEEE International Conference on*, pages 1521–1524. IEEE, April 2008.

[Jul62] Bela Julesz. Visual pattern discrimination. *Information Theory, IRE Transactions on*, 8(2):84–92, February 1962.

[LC+02] Kuen-Long Lee, Ling-Hwei Chen, et al. A new method for coarse classification of textures and class weight estimation for texture retrieval. *PATTERN RECOGNITION AND IMAGE ANALYSIS C/C OF RASPOZNAVANIYE OBRAZOV I ANALIZ IZOBRAZHENII*, 12(4):400–410, June 2002.

[LC05] Kuen-Long Lee and Ling-Hwei Chen. An efficient computation method for the texture browsing descriptor of mpeg-7. *Image and Vision Computing*, 23(5):479–489, December 2005.

[LP96] Fang Liu and Rosalind W Picard. Periodicity, directionality, and randomness: Wold features for image modeling and retrieval. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 18(7):722–733, July 1996.

[LSAR10] Ce Liu, Lavanya Sharan, Edward H Adelson, and Ruth Rosenholtz. Exploring features in a bayesian framework for material recognition. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 239–246. IEEE, June 2010.

[MMM73] S Mori, Y Monden, and T Mori. Edge representation in gradient space. *Computer Graphics and Image Processing*, 2(3):321–325, July 1973.

[MNSA07] Isamu Motoyoshi, Shin'ya Nishida, Lavanya Sharan, and Edward H Adelson. Image statistics and the perception of surface qualities. *Nature*, 447(7141):206–209, May 2007.

[MS12] Tomáš Majtner and David Svoboda. Extension of tamura texture features for 3d fluorescence microscopy. In *3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), 2012 Second International Conference on*, pages 301–307. IEEE, October 2012.

[Mur12] Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, August 2012.

[Pre70] Judith MS Prewitt. Object enhancement and extraction. *Picture processing and Psychopictorics*, 10(1):15–19, January 1970.

[Ras14] Sebastian Raschka. Naive bayes and text classification i-introduction and theory. *arXiv preprint arXiv:1410.5329*, October 2014.

[Ris01] Irina Rish. An empirical study of the naive bayes classifier. In *IJCAI 2001 workshop on empirical methods in artificial intelligence*, volume 3, pages 41–46. IBM New York, November 2001.

[RL+93] Ravishankar Rao, Gerald L Lohse, et al. Towards a texture naming system: identifying relevant dimensions of texture. In *Visualization, 1993. Visualization'93, Proceedings., IEEE Conference on*, pages 220–227. IEEE, October 1993.

[ROvdS11] Thomas Rückstieß, Christian Osendorfer, and Patrick van der Smagt. Sequential feature selection for classification. In *AI 2011: Advances in Artificial Intelligence*, pages 132–141. Springer, December 2011.

[Sta12] DH Stamatis. *Essential statistical concepts for the quality professional*. CRC Press, May 2012.

[SWD05] Gaurav Sharma, Wencheng Wu, and Edul N Dalal. The ciede2000 color-difference formula: Implementation notes, supplementary test data, and mathematical observations. *Color research and application*, 30(1):21–30, February 2005.

[TMY78] Hideyuki Tamura, Shunji Mori, and Takashi Yamawaki. Textural features corresponding to visual perception. *Systems, Man and Cybernetics, IEEE Transactions on*, 8(6):460–473, June 1978.

[YLZ+09] Wenzhu Yang, Daoliang Li, Liang Zhu, Yuguo Kang, and Futang Li. A new approach for image processing in foreign fiber detection. *Computers and Electronics in Agriculture*, 68(1):68–77, April 2009.