A comparative Study on Texture and Surface Descriptors for Ear Biometrics

Anika Pflug, Pascal Nicklas Paul, Christoph Busch
da/sec - Biometrics and Internet Security Research Group
Hochschule Darmstadt, Germany
{anika.pflug, pascal.paul, chrostoph.busch}@cased.de

***Abstract*—Recent research in texture-based ear recognition indicates that ear detection and texture-based ear recognition are robust against signal degradation and encoding artefacts. Based on these findings, we further investigate and compare the performance of texture descriptors for ear recognition and seek to explore possibilities to complement texture descriptors with depth information.**

On the basis of ear samples from visible light images and depth maps, we extract texture and surface descriptors from full profile poses (perspectives). We compare the recognition performance of selected methods for describing texture and surface structure, which are Local Binary Patterns, Local Phase Quantization, Histograms of Oriented Gradients, Binarized Statistical Image Features, Shape Context and Curvedness.

Secondly we propose a novel histogram-based descriptor that performs feature level fusion by combining two information channels to a new feature vector. Our concept can either be applied for fusing two different texture or two different surface descriptors or to combine texture and depth information. Based on the results of the previous experiment, we select the best performing configuration settings for texture and surface representation and use them as an input for our fused feature vectors. We report the performance of different variations of the fused descriptor and compare the behaviour of the fused feature vectors with single channel from the first series of experiments.

I. Introduction

As a consequence of increasing availability of high resolution surveillance cameras, the outer ear with is unique shape has moved into the focus of many forensic experts. These systems deliver images that reflect the detailed and unique structure of the outer ear and hence are suitable for automated personal identification. Moreover, cameras that capture video and depth information simultaneously have reached a state that make them applicable in semi-controlled scenarios, such as surveillance at ATMs, vending machines or border control. In situations, where the head is intentionally covered by a hat, the ear has been the only remaining characteristic to allow recognition. Recent research in texture-based ear recognition also indicates that ear detection and texture-based ear recognition are robust against signal degradation and encoding artifacts, which implies that we can achieve a good recognition performance, even from a distance of several meters from the camera. Based on these findings, we further investigate and compare the performance of texture descriptors for ear recognition and seek to explore possibilities to complement texture descriptors with depth information.

First proposed in 1989 by Iannarelli [1], ear recognition using texture and depth images has gained much attentions from the biometrics and forensics community during the last years. Based on the available data, the outer ear is widely regarded as a unique, permanent and easy to capture characteristic. The rich surface structure is especially valued for forensic image analysis, where the outer ear is more frequently visible than the face. Video footage can even provide material showing both ears and the face, which pushes the interest in automated ear recognition of criminal police agencies. Due to the increasing attentions for ear recognition, researches have shown that different techniques which have originally been proposed for face recognition can also be applied for ear recognition purposes. Among these features are approaches for landmark extraction, such as Active Shape Models [2] and SIFT [3] [4]. Key point detection and matching was also successfully applied to 3D images [5] and combined with a global image descriptor in [6]. Landmark and key point detection however, can be complex, time consuming and incorporate implicit assumptions about the capture scenario and object representation. Moreover the process of landmark or key point extraction adds a potential source for errors to the pipeline, which stacks with the probability of a wrong segmentation. In other words, even if the ear has been successfully detected, we can still get wrong landmarks or misplaced key points from the feature extraction stage. We hence focus on appearance features in this work.

In the field of appearance feature extraction from 2D images, Damer and Fuhrer obtained good results with HOG [7] and in [8] the authors use LBP in combination with different techniques for feature subspace projection. In an extensive survey of ear recognition algorithms, their performance and the databases the reported performance were obtained with can be found in [9].

The aforementioned selection of different approaches to ear recognition have been obtained from different datasets and hence are hard to compare with each other. The goal of this work is to benchmark different texture and surface description techniques and to give recommendations for the optimal settings under a given scenario, which is represented by a particular database. Based on the results on Local Binary Patterns (LBP) and Histograms of Oriented Gradients (HOG) in previous work, Local Phase Quantization (LPQ) and Binarized Statistical Image Features (BSIF) are likely to give good results for ear recognition as well. However, LPQ and BSIF have not been tested and compared to previous approaches for appearance based ear recognition before. Moreover, we compare the recognition performance of the texture image and

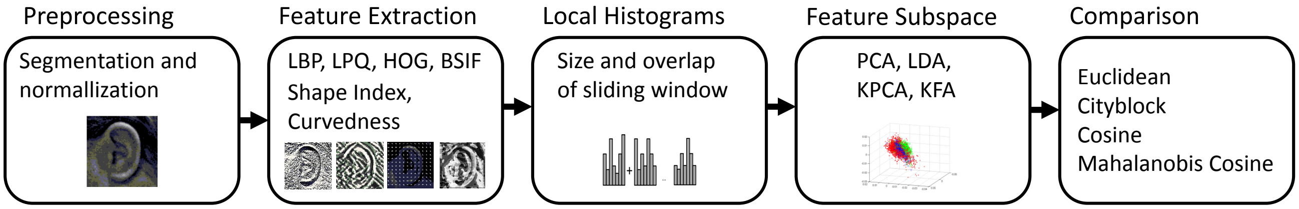

Fig. 1. Illustration of the experimental setup for our evaluations with all possible combinations between feature extraction, local histogram concatenation, feature subspace creation and comparison.

the depth image in order to see, which representation of the ear contains the most distinctive information.

We also compare different projection methods for feature subspace creation. Our experiments are conducted on three different datasets, which are UND-J2 [10], AMI [11] and IITK [12].

Our second contribution is a combined histogram descriptor, which combines 2D and 3D data. The combined histogram descriptor is based on the work of Zhou et al. [13] and [14], where two sources of data are combined to assign a given magnitude from one source to a bin that is determined by another source. The evaluation results for this descriptor are based on the UND-J2 dataset.

II. System Overview

As a first step, all images are transformed to gray scale and the contrast is adjusted using the CLAHE method [15]. Subsequently, variations in rotation and scale are removed with cascaded pose regression (CPR) [16]. With CPR, we estimate an ellipse that encloses the ear. Based on the length of the major and minor axis and the skewedness of the ellipse, we rotate and scale the ear image in a way that the major axis of the ellipse is vertical and corresponding to a predefined length. This is dome for the 2D images and for the depth images, such that the normalized images are still co-registered. Missing values in the 3D-Images are interpolated with a linear function for computing potential energy surfaces and a least squares approach [17] [1]. Finally, all images are resized to 100×100 pixels.

After preprocessing, we extract appearance or surface descriptors from overlapping grid cells and create fixed length histogram descriptors. The histogram descriptors are either computed on the basis of 2D images, depth images or both. Depending on the number of grid cells, the number of dimensions for each feature vector varies. For LBP, LPQ and BSIF we used local windows of the size 33×33, 20×20, 10×10 and 7×7 respectively with different overlaps that are dependent on the window size. The dataset is now split into a training and a testing set. The training set is used to estimate a projection matrix for a feature subspace. Based on the projection matrix, the remaining testing images are projected into the feature subspace. We compare the recognition performance of feature subspaces that were created with PCA, LDA as linear projection techniques and KPCA [18] and KFA [19] as non-linear projection techniques [2] Recognition is performed by a nearest neighbour classifier using a selection of different distance metrics which are the euclidean distance (euc), cityblock distance (ctb), cosine distance (cos) and mahalanobis cosine distance (mahcos). Parts of the source code for this experiment are based on the PhD Face recognition Toolbox. The whole data processing process, including all intermediate steps is illustrated in Figure 1.

Based on these experimental settings, we obtain biometric performance measures for different combinations between the feature extraction techniques, the size of the local windows, the overlap between local windows and the projection technique and the distance metric. In total we obtain more than 6000 different configuration settings, which we analyzed for this study. Each of the possible combinations is tested with 10-fold cross validation with a random selection for the training and testing set for the computation of the feature subspace.

III. Feature Extraction

A. Texture and Surface descriptors

Local Binary Pattern (LBP): LBP [20] encodes local texture information on a pixel level by comparing the grey level values of a pixel to the grey level values in its neighbourhood. The size of neighbourhood is defined by a radius around the pixel, which is at least 1 (for a neighbourhood having 8 pixels). Every pixel within the radius that has a larger grey level value than the center pixel is assigned the binary value 1, whereas all pixels with a smaller grey level value are assigned the binary value 0.

The binary values in the neighbourhood pixels are concatenated to form a binary string corresponding to the center pixel. Only those binary strings which have at most two bitwise transitions from 0 to 1 (or vice-versa) are considered - there are 58 such strings. This binary string is then mapped to a value between 0 and 255.

The LBP-based ear descriptor is computed by first sliding a window of a predefined size and overlap (step size in pixels) in the horizontal and vertical direction over the LBP image. From each sub window a local histogram with 256 bins is extracted.

We compare the performance values of this descriptor with various window sizes and overlap between neighbouring win-

[1] Because CPPR accurately segments the image, we can resize it to any size. Texture descriptors do not encode explicit shape information, hence the aspect ratio does not have to be maintained.

[2] We also evaluated the system without any projection technique. The EER in that case is significantly worse and varies between 10 and 20 %, when comparing the feature vectors directly, without prior subspace projection.

dows. We also compare the performance of LBP using a radius of 1 (n-8 neighbourhood) and radius 2 (n-16 neighbourhood).

Local Phase Quantization (LPQ): The concept behind LPQ [21] is to transform the image into the fourier domain and to only use the phase information in the subsequent steps. Given that a blurred image can be viewed as a convolution of the image and a centrally symmetric point spread function, the phase of a transformed image becomes invariant to blur. For each pixel in the image, we compute the phase within a predefined local radius and quantize the image by observing the sign of both, the real and the imaginary part of the local phase. Similar to LBP, the quantized neighbourhood of each pixel is reported as an eight digit binary string.

Given an image, the LPQ value is first computed for every pixel. Next, local histograms with 265 bins are computed within a sliding window. We compute the concatenated histogram descriptor for varying window sizes and with different radii for the neighbourhood of each pixel. In our experiments, we compare LPQ with different radii, windows sizes and overlap.

Binarized Statistical Images Features (BSIF): Inspired by LBP and LPQ, BSIF [22] also computes a binary string for each pixel in an image to represent the local structure of an image. The values of each bit within the BSIF descriptor is computed by quantizing the response of a linear filer. Each bit in the string is associated to a particular filter and the number of bits determines the number of filters used. As with LBP and LBP the binary code word is then mapped to a real value between 0 and 2^x for x different filters. Finally we create a histogram from the mapped values in the BSIF image for describing the local properties of the image texture.

In our experiments, we use the standard filters, which represent eight different orientations of edges. As before, we extract a local descriptor for different window sizes, overlap between neighbouring windows and different filter sizes and concatenate each local histogram to a global histogram representation. For all experiments, we use 8-bit code words and the 5×5, 1×1 and 17×17 filters

Histogram of Oriented Gradients (HOG): Computation of the HOG [14] descriptor involves five steps, which are the gradient computation, orientation binning, histogram computation, histogram normalization and concatenation of local histograms. The algorithm starts by computing the local gradient by convolving a 3×3 region (HOG cells) with two one-dimensional filters (-101) and $(-101)^T$. The local orientation at the center of each HOG cell is the weighted sum of the filter responses of each pixel.

The local orientations within a larger sub-window, denoted as block, are then quantized into bins in the $[0, 2\pi]$ interval. Subsequently, the image is divided into blocks of equal size and a local histogram of quantized orientations is extracted. Subsequently, the local histogram from each block is normalized with the L2-norm. Finally, all local histograms are concatenated to form the HOG descriptor for the image. We evaluate the recognition performance of HOG using 8×8 HOG cells and 9 bin histograms, as well as 16×16 HOG cells and 12 bin histograms.

Surface Descriptors: For describing three-dimensional structures in depth images, we use the shape index and curvedness. Both descriptors are based on the principal curvature of a shape and divide different surface structures into discrete categories, which are represented by values between 0 and 1. According to [23], the shape index for the maximum principal curvature k_{max} at a given position p in the image and the minimum principal curvature k_{min}, respectively, is defined as

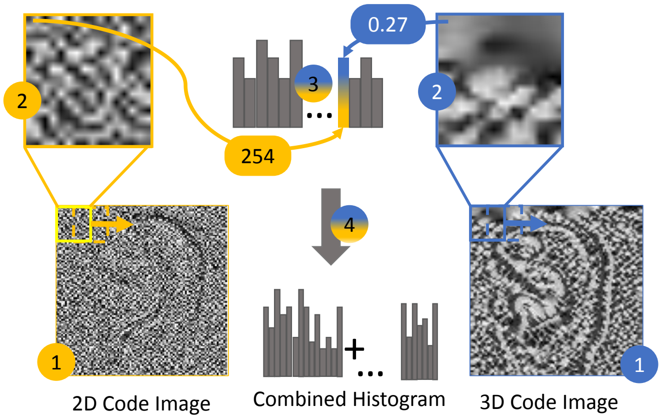

Fig. 2. Example for feature level fusion for the creation of the combined histogram descriptor.

$$S(p) = \frac{1}{2} - \frac{1}{\pi} \arctan \left(\frac{k_{max}(p) + k_{min}(p)}{k_{max}(p) - k_{min}(p)} \right) \tag{1}$$

Accordingly, the curvedness for a given image position can be written as

$$C(p) = \sqrt{\frac{k_{max}^2(p) + k_{min}^2(p)}{2}}. \tag{2}$$

B. Combined Descriptor

IV. System Overview

Our descriptor combines 3-dimensional and 2-dimensional data and compute local histograms from an image region that is defined by a sliding window with a given size and a given overlap between neighbouring windows. For each local window, we extract the code word images from both, the texture images (2D) and the depth image (3D). The number of bins can either be the total number of values that can occur in the code word image, or any other number that divides the value range of the descriptor into n equally sized bins between the minimum and the maximum possible value code word image.

Subsequently, we create a local histogram from these features by using the code word (feature) from position $I_{2D}(p)$ from the 2D-image for determining the bin. The code word at position $I_{3D}(p)$ from the 3D-image determines the value that is added to the bin size. This means that for each position p local window i, we create a local histogram $Hist_i$ with a given number of bins in the following way:

$$Hist_i(I_{2D}(p)) = Hist(I_{2D}(p)) + I_{3D}(p). \tag{3}$$

All local histograms are normalized using the L_2 norm and them concatenated to a global histogram descriptor. An example for the computation of a combined feature vector using LPQ and the Shape Index is depicted in Fig 2. In steps 1 and 2 we compute a code word image using LPQ for the texture image and the Shape Index for the depth image. In the upper right corner of the Shape Index image, we can see an interpolated region that clearly differs from the noisy signal of the depth sensor. Information from both channels is then fused in step 3, where the code word from the texture image determined the bin and the shape index from the depth is used for adding a weight to this bin. Finally, the combined histogram from LPQ and Shape Index data is normalized and then concatenated.

Obviously, this fusion scheme offers many different possibilities for combining texture images and co-registered depth images. We can either use the texture images for determining the bin and the depth images for adding a weight to this bin or vice versa. We will evaluate the performance of both options. Moreover, we can vary the number of bins to get longer of shorter descriptors. Finally, this scheme also allows us to combine two different feature vectors from texture images or depth images. In our experiments we will explore different combinations between the best performing configurations in texture- and depth images, as well as different combinations between texture descriptors in texture images and as the shape index or curvedness for depth images.

TABLE I. EQUAL ERROR RATE (EER) AND RANK-1 PERFORMANCE (RANK-1) (IN %) FOR SELECTED CONFIGURATIONS ON DATASETS UND-J2, AMI AND IITK. THE BEST CONFIGURATION FOR EACH DATASET IS MAKED IN BOLD.

Algorithm	UND-J2		AMI		IITK	
	EER	Rank-1	EER	Rank-1	EER	Rank-1
LPQ-3-20-10-LDA-cos	1.35	96.27	0.75	96.50	0.21	99.31
LPQ-5-33-15-LDA-cos	1.14	97.47	**0.00**	**100.00**	0.71	97.78
LPQ-11-20-10-LDA-cos	**0.67**	**98.73**	0.01	99.00	0.50	98.89
LPQ-11-33-15-LDA-cos	0.72	98.54	**0.00**	**100.0**	**0.17**	**99.03**
BSIF-5-20-10-LDA-cos	0.98	97.78	1.40	93.00	0.03	99.44
BSIF-5-20-15-LDA-cos	**0.39**	**98.67**	2.00	97.00	0.59	98.33
BSIF-5-20-15-PCA-cos	12.13	82.53	30.26	51.00	16.57	72.92
BSIF-11-20-5-LDA-mahcos	0.83	97.91	0.01	100.0	**0.19**	**99.44**
BSIF-11-20-10-LDA-cos	0.64	97.85	**0.00**	**100.0**	0.36	99.44
BSIF-11-20-15-LDA-cos	0.72	97.97	0.75	95.00	0.31	98.75
BSIF-17-20-15-LDA-cos	0.89	97.78	0.26	97.5	0.54	99.17
LBP-18-10-5-LDA-cos	0.98	97.34	0.68	97.50	0.69	98.47
LBP-18-20-5-LDA-cos	1.04	96.77	**0.55**	**97.20**	1.53	98.47
LBP-18-20-10-LDA-cos	1.73	96.77	0.66	97.50	1.94	97.92
LBP-18-20-5-LDA-cos	**0.94**	**97.22**	1.04	96.20	**0.48**	**98.89**
LBP-18-33-15-LDA-cos	7.88	69.24	5.14	82.70	0.68	97.87
HOG-8-18-LDA-cos	**1.27**	**97.84**	2.20	95.00	1.11	98.33
HOG-8-12-LDA-cos	1.44	96.77	2.61	93.50	1.66	97.64
HOG-16-18-LDA-cos	3.11	93.23	**0.13**	**100.0**	1.67	96.81
HOG-16-4-LDA-cos	12.11,	71.15	3.00	93	**0.52**	**98.61**
HOG-32-18-LDA-mahcos	14.72	66.52	16.25	48.50	3.30	93.47

TABLE II. EQUAL ERROR RATE (EER) AND RANK-1 PERFORMANCE (RANK-1) (IN %) FOR TEXTURE AND DEPTH IMAGES IN DATASET UND-J2 AND SELECTED CONFIGURATIONS. THE BEST CONFIGURATION FOR EACH DATASET IS MARKED IN BOLD.

Algorithm	texture image (2D)		depth image (3D)	
	EER	Rank-1	EER	Rank-1
BSIF-5-33-5-LDA-cos	1.01	96.96	2.09	95.31
BSIF-5-20-15-LDA-cos	0.39	98.67	1.81	95.25
BSIF-11-20-15-LDA-cos	0.72	97.97	2.58	93.23
BSIF-17-20-15-LDA-cos	0.89	97.78	5.43	85.39
LPQ-3-20-10-LDA-cos	1.35	96.27	2.81	95.24
LPQ-5-33-15-LDA-cos	1.14	97.47	3.59	90.32
LPQ-11-20-10-LDA-cos	0.67	97.97	2.34	93.80
LPQ-11-33-15-LDA-cos	0.72	97.59	0.61	98.86
LBP-18-20-15-LDA-cos	0.94	97.22	2.50	95.25
LBP-18-10-15-LDA-cos	0.98	97.34	3.51	91.51
LBP-18-33-15-LDA-cos	7.88	69.24	5.76	79.56
HOG-8-18-LDA-cos	1.27	97.85	3,58	91.96
HOG-16-18-LDA-cos	3.12	93.23	3.29	91.46

V. EXPERIMENTAL RESULTS

For our experiments, we select all subjects with at least 5 samples from each of the databases. For each selected subject, we then randomly pick 4 samples for training the projection matrix and for serving as reference images. The remaining samples are used for testing. For UND-J2 we select 158 different subjects with 790 images in total. For IITK, we use 288 images from 72 different subjects and for AMI the experiments are based on 500 images from 100 subjects.

The settings for the different algorithm configuration depends on the algorithm. A configuration also indicates the projection technique and the distance metric and is encoded as follows:

LBP: LBP - <radius; number of pixels> - <window size> - <overlap> - <projection technique> - <distance metric>

LPQ: LPQ - <radius> - <window size> - <overlap> - <projection technique> - <distance metric¿

BSIF: BSIF - <filter size> - <window size> - <overlap> - <projection technique> - <distance metric>

HOG: HOG - <block size> - <number of bins> - <projection technique> - <distance metric>

Surface: < SI | C> (SI = Shape Index; C = Curvedness)

The configuration settings for combined descriptors are encoded as follows: <bin configuration> - <number of bins> + <magnitude configuration> + <window size> - <overlap> - <projection technique> - <distance metric>

A. 2D and 3D Appearance Features

In Table I we have collected a selection of results from our evaluations. In general, we observe that the overlap between neighbouring windows does not have a major influence on the performance on any of the databases. The general expectation is, that smaller sizes of the local window yield a better performance than configurations with a larger local window. However, smaller local windows also imply that the information in the local histogram is bound to a smaller portion of the image, but also implies that the number of dimensions in the feature increases. The relation between these two factors should be balanced carefully. Moreover, the local window size should be chosen in a way that the texture descriptor is still able to extract a sufficient amount of information. For instance, LPQ with a radius of 7 only extract 9 values from a 10 × 10 local window, which is not sufficient for a good recognition performance. In general, we recommend to observe the sparsity of a local histogram for balancing the feature extraction parameters and the local window size. In our experiments, the best performance could be achieved, if at least 25% of the bins in each local histogram are different from zero. Sparsity is also an indicator for a clear image patch. Texture descriptors with about 50% of the bins equal to zero performed well in our evaluations.

Concerning the selection and parameters for the texture descriptors, we achieved excellent performance indicators with BSIF and LPQ. Some configurations for HOG and LBP result in good performance values too, but are in general inferior to the performance of BSIF and LPQ. The best configuration for the grid size is, in many of the cases, the 20×20 window with an overlap of 5 pixels between neighbouring windows.

Concerning the selection of a good window size, HOG plays a special role, compared to BSIF, LPQ and LBP. The local and non-overlapping windows in HOG (also referred to as HOG cells) are used directly for encoding local information, whereas all other algorithms extract local information from a sliding window. It is obvious that the window size has a strong impact on the performance of HOG. Based on our evaluations, we recommend window sizes between 8×8 and 16×16 pixels. Configurations with window sizes that are larger than 16×16 pixels did not perform well. To our surprise, the number of histogram bins plays a minor role for the effectiveness of the HOG descriptor.

The best combination between feature subspace projection technique and the distance measure is LDA in conjunction with cosine distance. The performance of LDA with mahalanobis cosine distance is close and in many cases within the confidence level. We hence prefer the simple solution and recommend the cosine distance measure together with LDA for ear recognition using texture features. Other distance measures resulted in significantly less recognition accuracy than cosine and mahalanobis cosine distance and can hence be discarded. Moreover, we observe that kernel methods for feature subspace projection (KFA and KPCA) take longer time for computation and yield less accuracy for recognition. The performance of configurations using PCA and BSIF, LPQ or LBP is between two and 10 times worse than LDA, depending on the database and feature extraction algorithm. For configurations using HOG, the performance of PCA is similar to LDA.

In Table II we show the performance of 2D and 3D images using selected configurations from the previous texture recognition configurations. The performance of depth images is always lower than for texture images, which can easily be explained by the fact that we are using feature vectors that were originally designed for describing texture features, not surface structures. Moreover, the surface structures can be noisy. Still we can conclude that texture descriptors are also capable to represent individual surface structures in depth images. We also observe that a good performance for a particular configuration of a texture descriptor also indicates a good performance for the same configuration in the depth image. We conclude from this that texture descriptors can be used to describe surface structure in depth images as well and that the behaviour of the feature vectors for texture and depth images is similar.

B. Combined Histogram Features

A selection of evaluation results from the second series of experiments is summarized in Table V-B. Compared to the results on texture images in the previous experiment, the recognition performance could not be improved by fusion texture and depth images. The best combination in our experiments was to use the texture information for defining the bin and the depth information for computing the magnitude. In many cases the results of these configurations were similar to the behaviour of the texture based pipelines. This means that LDA with cosine distance still appeared to be the best combination for most of the combined feature descriptors and that the overlap between neighbouring 20×20 windows should be 15 pixels.

An exception to this is LBP, where we observed low performance rates for all combined descriptors. Moreover we also found that the number of bins should be the same as for the texture descriptors only (namely 256). All configurations with a smaller number of bins performed significantly worse, which is due to a loss of information when several bin descriptors are mapped to the same bin. The fact that the combined descriptors do generally perform worse than the texture descriptors may be explained by the fact that the depth channel in noise, which can be seen in the example image on the right hand side in Fig. 2. Smoothing however, does not improve the performance, because smoothing also removes details from the ear structure. Artefacts from the interpolation process may also affect the recognition performance.

Using texture descriptors in the texture and the depth channel appears to be infeasible. Additionally, the usage of the noise depth channel for bin selection also yields low recognition rates and can hence not be recommended. The fact that the number of bins for combined histograms using the depth image for bin selection and the texture image for the bin magnitude suggests, that the histogram for these feature vectors is very sparse. Apparently, the loss of information from merging neighbouring bins does not affect the performance.

VI. Conclusion

Texture descriptors that were originally proposed for face recognition or other computer vision tasks can also be applied for ear recognition in texture and depth images. Our extensive evaluations show that the best performance for three different datasets is achieved with the LPQ and the BSIF descriptor in conjunction with LDA as dimensionality reduction methods and the cosine distance for a nearest neighbour classifier. The size and overlap of the local window should be balanced with the parameters for the feature extraction approach. In our experiments, we found that smaller local windows with more spatially bound descriptors do not improve the performance, because smaller radii for the local descriptors are more vulnerable to noise and the number of dimensions in the concatenated histogram becomes overly long.

We also proposed an approach, where texture and depth data is fused on the feature level and evaluated the performance. However, the performance of the combined descriptor turned out to be inferior to comparable configurations, where only the texture data is used. The usage of surface descriptors does not allow for a linear assignment of a histogram bin,

TABLE III. Equal Error Rate (EER) and Rank-1 performance (Rank-1) (in %) for selected combined descriptors.

Algorithm	Performance	
	EER	Rank-1
2D bins, 3D magnitude		
BSIF-5-256-SI-20-15-LDA-cos	**0.95**	**97.53**
BSIF-5-256-C-20-15-LDA-cos	1.32	96.08
LPQ-3-256-SI-20-10-LDA-cos	1.96	93.67
LPQ-3-256-C-20-10-LDA-cos	3.47	88,16
LBP-3-256-SI-20-15-LDA-cos	18.53	36.65
LBP-3-256-C-20-15-LDA-cos	19.60	30.76
3D bins, 2D magnitude		
SI-64-LBP-18-20-15-LDA-cos	26.06	19.24
SI-256-LBP-18-20-15-LDA-cos	27.35	18.54
SI-256-BSIF-5-20-15-LDA-cos	10.98	60.89
SI-8-BSIF-5-20-15-LDA-cos	12.27	61.58
SI-64-BSIF-5-20-15-LDA-cos	9.15	69.94
SI-256-LPQ-3-20-15-LDA-cos	19.40	37.66
SI-32-LPQ-3-20-15-LDA-cos	**8.89**	**74.74**
C-256-LBP-18-20-15-LDA-cos	33.43	9.11
C-64-LBP-18-20-15-LDA-cos	44.46	2.91
C-256-BSIF-5-20-15-LDA-cos	8.89	74.75
C-8-BSIF-5-20-15-LDA-cos	31.14	17.48
C-64-BSIF-5-20-15-LDA-cos	10.47	64.57
C-256-LPQ-3-20-15-LDA-cos	16.43	46.14
C-32-LPQ-3-20-15-LDA-cos	32.48	15.76
Texture descriptors (2D, 3D)		
BSIF-5-256-LPQ-3-20-15-LDA-cos	**4.21**	**85.25**
BSIF-5-256-LBP-18-20-15-LDA-cos	15.18	53.54
LPQ-3-256-LBP-18-20-15-LDA-cos	26.61	18.04
LPQ-3-256-BSIF-5-20-15-LDA-cos	2.67	94.75
LBP-18-256-LPQ-3-20-15-LDA-cos	14.13	56.14
LBP-18-256-BSIF-5-20-15-LDA-cos	14.15	54.62

because the resulting histograms are sparsely populated with large magnitudes in a small number of bins.

Even though the use of histogram-based texture descriptors provides a method for generating compact feature vectors, we observe hat the local histograms are sparsely populated. We plan to work towards a binary representation of the local feature vectors with the goal of providing a fast and accurate ear recognition system.

Acknowledgement

This project is funded by the Federal Ministry of Education and Research (BMBF) of Germany in the context of the research programme for public safety and security.

References

[1] A. V. Iannarelli, *’Ear identification’*. Paramont Publishing Company, 1989.

[2] L. Lu, Z. Xiaoxun, Z. Youdong, and J. Yunde, “Ear recognition based on statistical shape model,” in *First International Conference on Innovative Computing, Information and Control*, 2006, pp. 353–356.

[3] D. Kisku, H. Mehrotra, P. Gupta, and J. Sing, “Sift-based ear recognition by fusion of detected keypoints from color similarity slice regions,” in *International Conference on Advances in Computational Tools for Engineering Applications*, July 2009, pp. 380 –385.

[4] J. Zhou, S. Cadavid, and M. Abdel-Mottaleb, “Exploiting color sift features for 2d ear recognition,” in *International Conference on Image Processing*, 2011, pp. 553 –556.

[5] H. Zeng, J.-Y. Dong, Z.-C. Mu, and Y. Guo, “Ear recognition based on 3d keypoint matching,” in *International Conference on Signal Processing*, 2010.

[6] J. Zhou, S. Cadavid, and M. Abdel-Mottaleb, “A computationally efficient approach to 3d ear recognition employing local and holistic features,” in *Computer Vision and Pattern Recognition Workshops*, june 2011, pp. 98 –105.

[7] N. Damer and B. Fuhrer, “Ear recognition using multi-scale histogram of oriented gradients,” in *Conference on Intelligent Information Hiding and Multimedia Signal Processing*, July 2012, pp. 21–24.

[8] Y. Guo and Z. Xu, “Ear recognition using a new local matching approach,” in *International Conference on Image Processing*, October 2008, pp. 289 –292.

[9] A. Pflug and C. Busch, “Ear biometrics: a survey of detection, feature extraction and recognition methods,” *Biometrics, IET*, vol. 1, no. 2, pp. 114–129, June 2012.

[10] P. Yan and K. Bowyer, “’Biometric Recognition Using 3D Ear Shape’,” *Pattern Analysis and Machine Intelligence*, vol. 29, pp. 1297 – 1308, August 2007.

[11] E. Gonzalez, L. Alvarez, and L. Mazorra, “Normalization and feature extraction on ear images,” in *Security Technology (ICCST), 2012 IEEE International Carnahan Conference on*, 2012, pp. 97–104.

[12] A. Kumar and C. Wu, “’Automated human identification using ear imaging’,” *Pattern Recogn.*, vol. 45, no. 3, pp. 956–968, March 2012.

[13] J. Zhou, S. Cadavid, and M. . Abdel-Mottaleb, “’Histograms of Categorized Shapes for 3D ear detection’,” in *International Conference on Biometrics: Theory Applications and Systems*, November 2010.

[14] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, June 2005, pp. 886–893 vol. 1.

[15] K. Zuiderveld, *Graphics Gems IV*. Academic Press, 1994, ch. Contrast Limited Adaptive Histogram Equalization, pp. 474–485.

[16] A. Pflug and C. Busch, “Segmentation and normalization of human ears using cascaded pose regression,” in *19th Nordic Conference on Secure IT Systems*, 2014.

[17] P. Lancaster and K. Salkauskas, “Surfaces generated by moving least squares,” *Mathematics of Computation*, vol. 37, pp. 1–18, 1981.

[18] B. Scholkopf, A. Smola, and K.-R. Mller, “Kernel principal component analysis,” in *Advances in Kernel Methods - Support Vector Learning*. MIT Press, 1999, pp. 327–352.

[19] S. Mika, G. Ratsch, J. Weston, B. Scholkopf, and K. Muller, “Fisher discriminant analysis with kernels,” in *Neural Networks for Signal Processing IX, 1999. Proceedings of the 1999 IEEE Signal Processing Society Workshop.*, 1999, pp. 41–48.

[20] T. Ojala, M. Pietikainen, and T. Maenpaa, “Multiresolution gray-scale and rotation invariant texture classification with local binary patterns,” *IEEE Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 971 –987, July 2002.

[21] T. Ahonen, E. Rahtu, V. Ojansivu, and J. Heikkila, “Recognition of blurred faces using local phase quantization,” in *19th International Conference on Pattern Recognition*, Dec 2008, pp. 1–4.

[22] J. Kannala and E. Rahtu, “Bsif: Binarized statistical image features.” in *IEEE International Conference on Pattern Recognition*, 2012, pp. 1363–1366.

[23] P. J. Besl, *’Surfaces in Range Image Understanding’*, R. Jain, Ed. Spinger, 1988.