Kernel based Extreme Learning Machines for Image Classification

Stevica S. Cvetković¹, Miloš B. Stojanović², Saša V. Nikolić³, Goran Z. Stančić⁴

Abstract – This paper investigates possibilities for application of Kernel based Extreme Learning Machines (K-ELM) to the problem of multiclass image classification. It is combined with Local Binary Pattern (LBP) image descriptor, to reach highly accurate results. LBP is widely used global image descriptor characterized by compactness and robustness to illumination and resolution changes. Classification is done using recently introduced K-ELM method. Experimental evaluation on a standard benchmark dataset consisting of thousand images classified in ten categories, has shown high accuracy of results comparing to other benchmark models.

Keywords – Image classification, Neural networks, Kernel based extreme learning machines, Local binary patterns

I. INTRODUCTION

Image classification based on visual content is a crucial problem in computer vision research. The goal of an image classification system is to assign a category label with the most similar visual content, to the given query image. Visual similarity between images is commonly measured using robust and compact image descriptors (features).

There is a large set of visual descriptors available in the literature [1, 13]. The choice of the descriptor essentially affects the overall performance of the classification system. Local Binary Pattern (LBP) is one of the most widely used descriptor due to robustness to resolution and lighting changes, low computational complexity, and compact representation [2, 3, 4, 7]. The second crucial part of the system is machine learning technique to be applied for classification of descriptors. Support Vector Machine (SVM) is the most widely used machine learning technique for image classification purpose [5, 6].

In this study we investigate application of Kernel based Extreme Learning Machines (K-ELM) [8, 9, 10, 11, 12] for image classification, as an alternative to the commonly used SVM technique. The training of the SVM is based on solving quadric programming problem, which is usually time consuming when the number of training examples is large. Beside that SVMs are originally proposed for binary classification, while for the multi-class classification one-against-all (OAA) or one-against-one (OAO) approaches must be used in SVM implementation. On the other hand, K-ELM shows much better generalization performances for multiclass classification cases, and has better scalability and much faster training speed, compared with SVM [9].

In the rest of the paper we first give an overview of K-ELM classification method for multi-class image classification. Then we describe the process of LBP descriptor extraction. Finally, experimental evaluation and conclusion are presented.

II. KERNEL BASED EXTREME LEARNING MACHINES (K-ELM) FOR MULTICLASS CLASSIFICATION

Let us define \( N \) training examples as \((\mathbf{x}_i, y_i)\) where \( \mathbf{x}_i = [x_{i1}, x_{i2}, ..., x_{im}]^T \in \mathbb{R}^d \) denotes \( j \)-th training instance of dimension \( n \) and \( y_i = [y_{i1}, y_{i2}, ..., y_{im}]^T \in \mathbb{R}^m \) represents \( j \)-th training label of dimension \( m \), where \( m \) is the number of classes. LBP image descriptor, which will be described in the next section, will further be denoted as \( x_i \). As \( y_j \), we will denote \( m \) dimensional vector of binary class labels with value “1” denoting membership to the class. SLFN with activation function \( h(x) \) and \( L \) hidden neurons could be defined as:

\[
\sum_{i=1}^{L} \beta_i h(\mathbf{w}_i \cdot \mathbf{x} + b_i) = \mathbf{f}_j, \quad j = 1, \ldots, N
\]

(1)

where \( \mathbf{w}_i = [w_{i1}, w_{i2}, ..., w_{im}]^T \) denotes the vector of weights which connects the \( i \)-th hidden neuron and all input neurons, \( \beta_i = [\beta_{i1}, \beta_{i2}, ..., \beta_{im}]^T \) is the weight vector which connects \( i \)-th hidden neuron and all output neurons, and \( b_i \) is the bias of the \( i \)-th hidden neuron. By ELM theory [8], \( \mathbf{w}_i \) and \( b_i \) can be assigned in advance randomly and independently, without a priori knowledge of the input data. The ELM network structure is presented in Figure 1.

SLFN in (1) should satisfy \( \sum_{i=1}^{L} \| \mathbf{f}_j - \mathbf{y}_j \| = 0 \), i.e., there exist \( \beta_i \), \( \mathbf{w}_i \), and \( b_i \) such that:

\[
\sum_{i=1}^{L} \beta_i h(\mathbf{w}_i \cdot \mathbf{x}_j + b_i) = \mathbf{y}_j, \quad j = 1, \ldots, N
\]

(2)
The equivalent compact matrix form of (2) can be written as

$$H\beta = Y$$

(3)

where $H$ in (3) represents the hidden layer output matrix of the neural network; the $i^{th}$ column of $H$ represents the $i^{th}$ hidden neuron’s output vector in regard to inputs $x_1, x_2, \ldots, x_N$.

Although the output weights can be analytically determined by finding the unique smallest norm least-squares solution of the linear system (3) in order to improve the performance the constrained optimization problem can be formed for ELM multiclass classifier with multiple outputs, as shown in [9]:

$$Minimize \mid L_p = \frac{1}{2} \| \beta \|^2 + C \frac{1}{2} \sum_{j=1}^{N} \| \xi_j \|^2$$

(6)

Subject to $h(x_j)\beta = y_j^T - \xi_j^T$, $j = 1, \ldots, N$

where $\xi_j = [\xi_{j1}, \ldots, \xi_{jm}]^T$ is the training vector of the $m$ output nodes with respect to the training sample $x_j$, while $C$ represents tradeoff parameter between model complexity and allowed errors $\xi_j$ during training. Based on Karush - Kuhn - Tucker (KKT) theorem, the optimization problem defined in (6) is equivalent of solving the dual optimization problem:

$$L_D = \frac{1}{2} \| \beta \|^2 + C \frac{1}{2} \sum_{j=1}^{N} \| \xi_j \|^2$$

$$-\sum_{j=1}^{N} \sum_{i=1}^{m} \alpha_{ij} (h(x_j)\beta_i - y_j + \xi_j)$$

(7)

where $\alpha_{ij} = [\alpha_{j1}, \ldots, \alpha_{jm}]^T$ are Lagrange multipliers.

After solving (7) based on KKT conditions, which can be found in detail in [9], the following solution is obtained:

$$\beta = \left( \frac{1}{C} + H^T H \right)^{-1} H^T Y$$

(8)

and the decision function of ELM classifier is:

$$f(x) = h(x)\beta = h(x)H^T \left( \frac{1}{C} + H^T H \right)^{-1} Y$$

(9)

If feature mapping $h(x)$ is unknown, we can apply Mercer’s condition on ELM. We can define kernel matrix for ELM as:

$$\Omega_{ELM} = HH^T$$

(10)

$$\Omega_{ELM,i,j} = h(x_i)h(x_j) = K(x_i,x_j)$$

In KELM $H = \left[ h(x_1)^T \ldots \ldots \ h(x_N)^T \right]^T$ represents hidden layer output matrix which maps data $x_i$ from the input space to the hidden layer feature space and it is irrelevant to target values $y_i$ and number of output nodes $m$. The kernel matrix $\Omega_{ELM} = HH^T$ is related only to input data $x_i$ and number of training samples $N$, for regression, binary classification and multi class classification.

Then, the output function of ELM classifier (9) can be written compactly as:

$$f(x) = \left[ K(x,x_1)^T \ldots \ K(x,x_N)^T \right] \left( \frac{1}{C} + \Omega_{ELM} \right)^{-1} Y$$

(11)

In this case the feature mapping $h(x)$ does not need to be defined by users, as well as the dimensionality of feature mapping.

![Fig. 1. Structure of an ELM network.](image-url)
space \( L \) (number of hidden nodes), just its kernel \( K(u,v) \). In our experiments RBF kernel is used, defined as:

\[
K(u,v) = \exp(-\gamma \|u-v\|^2)
\]

(12)

where \( \gamma \) represents parameter of Gaussian kernel. It can be noted from (11) and (12) that optimal combination of parameters \( C \) and \( \gamma \) have to be obtained in order to achieve good generalization performance.

### III. LOCAL BINARY PATTERNS (LBP)

Local Binary Pattern (LBP) is a popular image descriptor that captures local appearance around a pixel. LBP descriptor of the complete image is formed as a histogram of quantized LBP values computed for every pixel of the image. It was introduced in [4] for the texture classification problem, and extended to general neighborhood sizes and rotation invariance in [2]. Since then, LBP has been extended and applied to variety of applications [3].

For a given image \( I \), the local LBP descriptor centered on pixel \( I(x, y) \) is an array of 8 bits, with one bit encoding each of the pixels in the 3×3 neighborhood (Fig 2). Each neighbor bit is set to 0 or 1, depending on whether the intensity of the corresponding pixel is greater than the intensity of the central pixel. To form the binary array, neighbors are scanned starting from the one to the right, at position \( I(x+1, y) \), in anti-clockwise order.

<table>
<thead>
<tr>
<th>a) Pixel intensities</th>
<th>b) Thresholded difference</th>
<th>c) LBP</th>
</tr>
</thead>
<tbody>
<tr>
<td>167 221 221</td>
<td>0 1 1</td>
<td>01100010</td>
</tr>
<tr>
<td>147 217 198</td>
<td>0 0</td>
<td></td>
</tr>
<tr>
<td>132 230 212</td>
<td>0 1 0</td>
<td></td>
</tr>
</tbody>
</table>

Fig. 2. Example of a LBP extraction process for central pixel of intensity 217.

<table>
<thead>
<tr>
<th>% of training images per class</th>
<th>80%</th>
<th>50%</th>
<th>20%</th>
</tr>
</thead>
<tbody>
<tr>
<td>K-ELM</td>
<td>90.86 (±1.32)</td>
<td>89.58 (±1.24)</td>
<td>85.49 (±1.07)</td>
</tr>
<tr>
<td>SVM (kernel) [14]</td>
<td>90.79 (±1.92)</td>
<td>88.26 (±1.62)</td>
<td>84.04 (±0.87)</td>
</tr>
<tr>
<td>SVM (linear) [14]</td>
<td>89.62 (±1.27)</td>
<td>87.65 (±1.03)</td>
<td>83.41 (±1.25)</td>
</tr>
</tbody>
</table>

We further measured average training and testing time of the method on an Intel Core i7 3.5GHz computer. Training time of the complete training set was less than 1 second, while classification of a test image is done instantly (< 0.1ms). These results demonstrate high performances in terms of training and test speed on the test dataset.

In order to compare results of the ELM with other common classification techniques, we measured accuracy of the Linear SVM and kernelized RBF SVM [14], on the same dataset. Linear SVM parameter \( C \) was set to value 0.1. RBF SVM optimal parameters were determined by grid-search and 5-fold cross-validation, where \( C \) was examined in range \([2^3, ..., 2^{10}]\) and \( \gamma \) in range \([2^{-10}, ..., 2^4]\). On the other hand, parameters of K-ELM were set to fixed values \( C = 10 \) and \( \gamma = 2^{10} \). It can be noted from experimental results that in terms of accuracy ELM constantly outperforms both Linear SVM and kernelized RBF SVM, without additional computational costs.
V. CONCLUSION

In this study we presented results of our research in the field of automatic image classification using Kernel based ELM classifier (K-ELM) combined with LBP image descriptor. K-ELM classifier could be used as an effective alternative to the commonly used SVM methods. We reached classification accuracy of over 90%, on a test dataset containing 1000 images in 10 categories, what is high quality result on this dataset. It can be concluded that combination of K-ELM classifier with the LBP image descriptor is reasonable choice for image classification applications. In the further research, we plan to investigate integration of other image descriptors combined with K-ELM classifier. Particularly we will focus our research on fusion of color and texture descriptors.

REFERENCES


New interactive interface for visualization of 3D meshes

Nicole Christoff\textsuperscript{1,2}, Hadrien Greef\textsuperscript{3}, Sébastien Germond-Mazet\textsuperscript{3}, Jean-Luc Mari\textsuperscript{2}, Laurent Jorda\textsuperscript{4} and Agata Manolova\textsuperscript{1}

Abstract – We propose a novel interface that visualizes a 3D mesh form the Polygon File Format (PLY), which represent a Martian terrain data. It juxtaposes the crater location, which coordinates of the center and diameter are specified in another input file. This program is designed to check and manually correct the results, obtained by the craters detection algorithm (CDA). This validation software is developed for use by astronomers. The focus is on the ergonomics of the interface and features.

Keywords – 3D mesh, interface, visualization, craters

I. INTRODUCTION

The collision of celestial bodies, such as asteroids or meteorites, gives as result craters on their surface. The study of different morphological features of craters, and counting their numbers, is the only method to estimate the age of the celestial body. Therefore, it is a problem of great scientific importance in astronomy. The complexity of the problem arises when the impact crater zones are very heterogeneous due to the distribution and size of the craters.

To be able to solve partially this issue, different methodologies of geometric image analysis are proposed. They are applied to 2D or 3D data for automatic crater detection. Indeed, the manual identification of craters on the planet surface is an effective and precise method. The volume of data that needs to be processed grows with the development of new imaging technologies. That pushes scientists to look for a fully automatic method.

The combination of the two techniques (manual and automated) is a good solution of the problem. It is based on a rapid automatic pre-detection of craters that serves as a base to the human validation. We provide an ergonomic interface for visualization of 3D craters, which can correct the automatic detection.

The rest of the paper is organized as follows: in Section II, the existing approaches related to crater detection and interface are briefly described. In Section III, we describe the environmental data, then we develop the core of our interface in Section IV. The Section V is dedicated to the theoretical solutions for the visualization of 3D meshes. We present our results in Section VI. Conclusions and direction of future research are given in Section VII.

II. RELATED WORKS

Many CDAs have been developed. They are based on 2D image data or 3D data, often in the form of Digital Elevation Model (DEM). The different approaches developed for 2D CDA are: the circle fitting [1], the exhibition of highlight and shadow region matching [2] or analysis textures [3]. Several approaches combine many of these methods [4].

The advantages of using detection from DEM data are the high resolution and the lack of hypothesis for the reflection of the surface, as well as the integration of additional 3D information. Previous works in this area are based on the detection of edges [5], identification of contours as a local circular depression [6] or the circle fitting Hough transform [7].

The creation of an ergonomic interface and 3D visualization of data, has been sparsely discussed in scientific works, often focusing on results and theoretical issues.

There are some works, which incorporate a graphical interface and they make the connection with CDA [8] and the existing functionality of Geographic Information System (GIS) software. For example, ArcGIS and QGis and various external searching modules [9], [10], based on the same software, each time in 2D. As far as we know, there is no reference regarding the 3D visualization of meshes, combined with the displaying of detecting craters by CDA.

III. ENVIRONMENT AND DATA

A. Environment

The C++ programming language is used for the visualization of the interface and the data structures, used for meshes: Polygon File Format (PLY) and Comma Separated Values (CSV). Qt 4.8.6 in its version of 2013 is a cross-platform application framework that is widely used for developing application software [13].

The Visualization ToolKit (VTK) is used for the acquisition of a 3D visualization library for manipulating meshes and.PLY files. It is a tool, directly linked with Qt and supports advanced graphics features. VTK is optimized to manage heavy amounts of data and complex displays. Its graphic operating resources are developed to run our software on fairly modest configurations, even with large files.

B. Data

The 3D mesh, in .PLY format, represents an area of Mars. It is considered for the detection, already triangulated from the...
DEM. In our case, it is a large data: over 5 million points (vertex) and 10 million faces. For comparison, the smaller ones are about 3 million points and 6 million faces. The area is a sample of 0.46 km / mesh. The input file is detected with a .CRAT file (CRATer extension, especially created for this work). This is a CSV file, with the following structure: coordinates of X, Y, Z of the center and the diameter of craters, detected by a CDA.

IV. DEVELOPMENT OF THE INTERFACE

A. Menus and windows

For the functional and ergonomic interface, we use the complex Qt classes - QMainWindow. The global window of the software contains a top menu, side toolbars, footer menu and the viewing window VTK. We made many inheriting of Qt classes to define our own widgets (Fig. 2). The main file generates a MainWindow, containing a CentralWindow, derived from VTKWidget class. The latter inherits the characteristics of QVTkWidget class. These heritages allow us to override certain functions of the original classes, while maintaining the interactions between all these objects. In the main window, we define the areas of the program: (1) a horizontal toolbar (Fig. 1. at the top of the window.), which contain the main interactions, opening files, saving, display options, etc. ; (2) a lateral toolbar (Fig. 3) irremovable for configuring the mouse modes of action (add a crater trackball mode, etc.) and display a basic information about uploaded files; (3) the central window, in which the mesh is displayed and where the user can click to move or edit 3D craters (Fig. 1 right).

All functionality of the menu and the toolbar are connected to signals and slots (a language construct introduced in Qt for communication between objects [14]), to pass them on the program's global data (the list of craters) or to change the Interactor mouse (see Section IV-c). We manage pop-ups: (1) for searching of files that will be used and (2) to link these windows to the main application. The theoretical solutions of all the features are explained in Section V.

B. Displaying the grid

The solution had to be considered for a fluid display of the mesh, which can be moved, enlarged and covered, in all directions, by the user. To load the file, a decimation algorithm was
used to produce a lower level of detail. It is sufficient to continue to distinguish the biggest reliefs. In fact, it is destined to be displayed instead of the detailed model, every time the camera moves and zooms. In this way, to move the entire mesh remains a fluid process. Once the model is positioned, it is the detailed version that appears. This technique, used in GIS software [10] does not utilize many graphics resources and makes visualizing in real-time the mesh of millions of polygons possible. The decimation technique is presented in Section IV. The number of polygons, obtained in the decimation step, cannot be known in advance. In our case, it depends on the desired refreshing rate, produced during the movement and is thus likely to vary, depending on system configuration. The performances do not vary. Fig. 4 illustrates both Level of Detail (LOD).

Fig. 4. Comparison between original 10 million polygons meshes (left) and decimated one of ~1000 polygons (right).

On Fig. 5 is presented a gray material, specular component of the dominant light. It is rather difficult to spot instinctively the potential craters. We implemented a staining method, which is called geographic. We have done a color mapping at each point, which depend on the value, ranges of distances from the center of Mars. More a point is far from the center of Mars (and therefore the high altitude), more the color should be cold (in a classic pattern red -> green -> blue). We get a very convenient local coloring, because of the abrupt altitude depressions, the result of a color change, as shown in Fig. 5.

Fig. 5. Comparison between the original (left) and the geographic staining (right).

This staining was obtained by the computing the radius of the planet center, located at X, Y, Z (0,0,0) in the landmark planet. It is a color scale between the minimal and the maximal height and the correct color range is assigned to each point. There is a significant improvement for the identification of small craters (a few kilometers) on the mesh. If we are very close to the surface, it was a difficult case to distinguish craters without coloring. That becomes an additional tool to facilitate the manual detection.

C. User interactions and the .CRAT file

We present an ergonomic operating interface, which seems to be intuitive to users, who are accustomed to software as QGis [10]. We implemented four classes. Each of them corresponding to one of the four modes: (1) TRACKBALL uses to move the mesh to rotate and zoom in / out, and resumes the VTK usual trackball by adding features, such as changing the associated focal point with a double click. (2) ADD CRATER allows clicking on the mesh to include therein a crater. It is represented as a sphere. Holding down the button, the size of the crater can be adjusted and it starts from the original size adjustment, with the slider lateral toolbar. (3) EDIT CRATER can "raise" a crater to its correct position and adjust its diameter. (4) REMOVE CRATER allows removing a crater. If the user makes a mistake or if he wants to invalidate a false positive, it is proposed a drag function. If we lose the click outside the bounding sphere of the crater, it is not deleted. During these actions, the overall craters list is updated in real time and is recordable, at any time, in .CRAT format. This file format is used as input or output information data for the CSV files. We can load the detection file, make all kinds of changes and save the modified file. This makes much faster the detection of craters, retaining the human critical look at the results. Finally, we illustrate the different modes and which colors are associated with the craters, as shown in Fig. 1 (central window). With green mesh are covered craters, detected by the CDA, orange mesh is added by hand and a red mesh indicates a crater ready to be removed.

V. THEORETICAL SOLUTIONS

A. Decimation of surfaces

To streamline the interface, we discussed the idea of a hierarchy of levels of detail. It is called by the current performance of the center window. Thus, we use a quadratic mesh decimation that preserves the overall topology of the form by drastically reducing the number of polygons. It is based on the classification of points [12] according to their complexity: (1) by calculating the distance to the tangent to the surface plane and (2) by comparing the decimated surfaces and the not decimated, with a quadratic computing (improved decimation simple). It is in conjunction with the class, which manages the different LODs and allows setting the desired performance.

B. Picking

Another important feature is the ability to click on a 3D point and to select the mesh, corresponding to the clicked place, or picking. The mouse coordinates are tied with the 2D (on the screen) to their 3D projection on the existing surface and recovers the point, which is the nearest place, considered as a mesh. The VTK has a picking manager, which works by ray tracing. Between the camera and the clicked point, we are launching a radius straight until the launching of the mesh or the bounding sphere of a crater. Then, we made a linear interpolation and we found a contact with the point, which recovers a "real" point. This method of picking is very effective and accurate. It allows
moving a crater, "sticky" to the surface below, rather than moving the camera in the plane or on an axis of the mark. It ensures that the action of moving of a crater remains faithful to other 3D craters.

C. Fitting / Orientation

Finally, two problems have been solved: (1) to correct an offset between the mesh and the real crater and (2) to fit the mesh covering the crater to the size of the real crater. The offset is a known error, due to the improper manual submission by geologists [11]. We propose a correction of inaccurate detections craters (see Section IV-C). To guide our disc in the space, we used the least squares optimization method. For surface fitting, we have the coordinates of the center of Mars, as information on the craters file. We calculate the direction of the Martian axe, passing through its center. We use it as a normal to orient our mesh in successive rotations. This method is more elegant in its simplicity.

VI. RESULTS

Once we have a .CRAT file, listing all the supposed detected craters by the CDA, the corresponding mesh is loaded into our visualization program. We can see the 3D detected craters. It is easy to identify the offset and to correct in some few clicks. In addition, we can add a forgotten crater.

Fig. 6. Visualization of craters, which are detected by a CDA.

Fig. 7. Visualization of craters, which are detected after manual corrections.

It is difficult to represent exhaustively the results, because the detection rate is important. We can present it as a significant contribution to the detection algorithms. We illustrate the offset correction on the file, between the output of an algorithm CDA and its validation by the user on a small area of Mars. Corrections are applied in a few seconds (Fig. 6 and Fig. 7).

After the saving a .CRAT file, we obtain a theoretically perfect detection (if it is performed by an expert). The simplicity of the validation provides high-performance detection in a short time, by combining a starting rate and an effective manual correction.

VII. CONCLUSION

We present an interface for validation, which allows users to browse intuitively 3D meshes. It induces a better perception of reliefs than 2D. It can move and adjust the craters. For this work, we were focused on the development phase on the fundamental points of the program, namely the picking and user interactions. For future work, we think to improve the performance, with the management of more than two levels of detail.

ACKNOWLEDGEMENT

This paper was supported by Contract No 162ПД0033-07 of Technical University-Sofia, Research Sector. Research project: «Segmentation and modeling of geometrical characteristics of 3D objects» - 2016.

REFERENCES

Comparison of Complex Hadamard Transform with other Unitary Transforms

Rumen P. Mironov

Abstract – A comparison of the developed complex Hadamard transform with the well-known unitary (orthogonal) transforms of Karhunen-Loeve, Fourier, Walsh-Hadamard and discrete cosine transform is presented. The comparison is made on the base of minimization of mean-squared error of reconstructed transform coefficients for two test images.

Keywords – Digital Image Processing, Complex Hadamard Transform, Orthogonal Transforms.

I. INTRODUCTION

The use of unitary (orthogonal) transformations is inextricably linked with the development of modern digital image processing methods. Discrete unitary transforms described in [1], [2] and [3], have found applications in many areas of N-dimensional signal processing, spectral analysis, pattern recognition, digital coding, computational mathematics and etc. Stated simply, these transform coefficients that are small may be excluded from processing operations, such as filtering, compression and etc., without much loss in processing accuracy.

In this article a comparison of the developed complex Hadamard transform [4] with most used unitary transforms of Karhunen-Loeve (KLT), Fourier (DFT), Walsh-Hadamard (WHT) and discrete cosine transform (DCT) is presented. There are various studies and comparisons of the orthogonal transforms [5], [6], [7], in which they discussed their properties, advantages and disadvantages from a statistical point of view. In this article a comparison of transformations in terms of the developed method [8] for optimal performance of the coefficients at the block coding for two test images was made.

The comparison is made through simulation of the developed algorithms for five unitary transforms – FFT, DCT, WHT, KLT and CHT on Matlab environment for two test images “Lena” and “Fruits” and the results are given in the experimental part.

II. MATHEMATICAL DESCRIPTION

The forward and the inverse 2D discrete unitary transform of sub-image g(x,y) of size N x N can be expressed as the following equations:

\[
S(u,v) = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} g(x,y) r(x,y,u,v)
\]

\[
g(x,y) = \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} S(u,v) t(x,y,u,v)
\]

In this equations g(x,y) is the input image with spatial variables (x,y), S(u,v) is forward transform with transform variables (u,v), r(x,y,u,v) and t(x,y,u,v) are called the forward and inverse transformation kernels, respectively. Because the inverse kernel t(x,y,u,v) in (1) depends only on the indices (x,y,u,v) and not on the values of g(x,y) and S(u,v), it can be viewed as defining a set of basis functions or basis images [3].

This interpretation becomes clearer if the equation is modified in matrix form:

\[
G_v = \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} S(u,v) T_{uv}
\]

where: \(G_v\) is N x N matrix containing the pixels of g(x,y), the matrices \(T_{uv}\) are the basis images and \(S(u,v)\) are the spectral coefficients.

The forward transformation kernel is said to be separable if:

\[
r(x,y,u,v) = r_1(x,u) r_1(y,v)
\]

and the kernel is said to be symmetric if \(r_2(x,u)\) functionally equal to \(r_2(y,v)\), so that:

\[
r(x,y,u,v) = r_1(x,u) r_2(y,v)
\]

Identical comments apply to the inverse kernel by replacing function \(r(x,y,u,v)\) with \(t(x,y,u,v)\) in the equations (1).

As a sample the 2D Fourier transform has the following forward and inverse kernels:

\[
r(x,y,u,v) = e^{-j2\pi(uM+vy/N)}
\]

\[
t(x,y,u,v) = \frac{1}{MN} e^{j2\pi(uM+vy/N)}
\]

where: \(j = \sqrt{-1}\), so these kernels are complex.

A computationally simpler Walsh-Hadamard transform is derived form the following functionally identical kernels [3]:

\[
r(x,y,u,v) = t(x,y,u,v) = \frac{1}{n} \sum_{l=0}^{n-1} \delta[b_0(x) b_1(y) \ldots b_n(z)]
\]

where \(n=2^n\). The summation in the exponent is performed in modulo 2 arithmetic and \(b(k)\) is the \(k\)th bit in the binary representation of \(z\). The \(\delta(l)\) is the conversion of \(b(l)\) with code of Grey.

---

1 Rumen P. Mironov is with the Faculty of Telecommunications, Technical University of Sofia, Boul. Kl. Ohridsky 8, Sofia 1000, Bulgaria, E-mail: rmironov@tu-sofia.bg
The most used discrete cosine transformation is obtained by following equal kernels:

\[ r(x,y,u,v) = f(x,y,u,v) = a(u)a(v) \cos \left( \frac{2x + 1}{2n} \pi \right) \cos \left( \frac{2y + 1}{2n} \pi \right), \quad (7) \]

where:

\[ a(u) = \begin{cases} \frac{1}{\sqrt{n}} & \text{for } u = 0, \\ \frac{2}{\sqrt{n}} & \text{for } u = 1,2,\ldots,n-1, \end{cases} \]

and similarly for \( a(v) \).

The identical description for the developed complex Hadamard transformation kernels are made in [9] and are used for comparison.

The kernels of optimal Karhunen-Loeve transformation are calculated for each test image by calculation the correlation function and the preparation of the eigenvalues and eigenvectors as is given in [1]. To simplify the calculations the input learning vectors are taken from the current test image consecutively according to image linear scanning.

From the equations (1) to (4) the compared transformations can be generalized for two-dimensional signals (images) in the following way:

\[
\begin{bmatrix} Y \end{bmatrix} = \begin{bmatrix} T_N \end{bmatrix} \begin{bmatrix} X \end{bmatrix} \begin{bmatrix} T_N \end{bmatrix}^T, \quad (8)
\]

where: \([X]\) is matrix of the input image with size \( N \times N \), \([T]\) is a matrix form of each transform kernel and the result is a spatial spectrum matrix \([Y]\) with the same size.

The symmetry of \([T]\) matrix coefficients allows 2D transforms to be accomplished in two steps. The first one is one-dimensional transform for every row of the image and the second one is one-dimensional transform for the columns. This difference of transformation makes easier the calculations and the symmetry guarantees that the correlations between image elements in horizontal and vertical direction will influence in the same way the determination of transformed elements. The same considerations can be made for two steps calculation of the inverse 2D transforms.

III. Experimental Results

The comparison is made on Matlab environment for five 2D unitary transforms – Karhunen-Loeve, Discrete Fourier Transform, Discrete Cosine Transform, Discrete Walsh-Hadamard Transform and Complex Hadamard Transform. The obtained experimental results for the test grayscale images “Lenna” and “Fruits” (512x512, 8 bits) with sub-image kernel 8x8 are given on Table 1 and Table 2 respectively.

<table>
<thead>
<tr>
<th>Reduction type</th>
<th>MSE</th>
<th>NMSE</th>
<th>SNR, dB</th>
<th>PSNR, dB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zero DFT</td>
<td>114.488</td>
<td>1.47 x 10^{-5}</td>
<td>48.3180</td>
<td>27.5772</td>
</tr>
<tr>
<td>Mean DFT</td>
<td>103.124</td>
<td>1.32 x 10^{-5}</td>
<td>48.7720</td>
<td>28.0312</td>
</tr>
<tr>
<td>Zero DCT</td>
<td>-0.3842</td>
<td>-4.9 x 10^{-8}</td>
<td>73.0599</td>
<td>52.3191</td>
</tr>
<tr>
<td>Mean DCT</td>
<td>-0.2903</td>
<td>-3.7 x 10^{-8}</td>
<td>74.2765</td>
<td>53.5357</td>
</tr>
<tr>
<td>Zero WHT</td>
<td>19.0409</td>
<td>2.45 x 10^{-6}</td>
<td>56.1087</td>
<td>35.3679</td>
</tr>
<tr>
<td>Mean WHT</td>
<td>16.1269</td>
<td>2.07 x 10^{-6}</td>
<td>56.8301</td>
<td>36.0893</td>
</tr>
<tr>
<td>Zero CHT</td>
<td>16.2735</td>
<td>2.09 x 10^{-6}</td>
<td>56.7908</td>
<td>36.0500</td>
</tr>
<tr>
<td>Mean CHT</td>
<td>15.0354</td>
<td>1.93 x 10^{-6}</td>
<td>57.1345</td>
<td>36.3936</td>
</tr>
<tr>
<td>Zero KLT</td>
<td>299.676</td>
<td>3.86 x 10^{-5}</td>
<td>44.9283</td>
<td>24.1875</td>
</tr>
<tr>
<td>Mean KLT</td>
<td>249.875</td>
<td>2.45 x 10^{-5}</td>
<td>44.9283</td>
<td>24.1875</td>
</tr>
<tr>
<td>55 Reduced Coefficients, 9 saved</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Zero DFT</td>
<td>126.300</td>
<td>1.62 x 10^{-5}</td>
<td>47.8915</td>
<td>27.1507</td>
</tr>
<tr>
<td>Mean DFT</td>
<td>113.403</td>
<td>1.45 x 10^{-5}</td>
<td>48.3593</td>
<td>27.6185</td>
</tr>
<tr>
<td>Zero DCT</td>
<td>-0.2014</td>
<td>-2.6 x 10^{-8}</td>
<td>75.8649</td>
<td>55.1241</td>
</tr>
<tr>
<td>Mean DCT</td>
<td>-0.0054</td>
<td>-7.0 x 10^{-10}</td>
<td>91.5437</td>
<td>70.8029</td>
</tr>
<tr>
<td>Zero WHT</td>
<td>48.4430</td>
<td>6.23 x 10^{-6}</td>
<td>52.0533</td>
<td>31.3255</td>
</tr>
<tr>
<td>Mean WHT</td>
<td>45.8332</td>
<td>5.89 x 10^{-6}</td>
<td>52.2938</td>
<td>31.5530</td>
</tr>
<tr>
<td>Zero CHT</td>
<td>46.2743</td>
<td>5.95 x 10^{-6}</td>
<td>52.2522</td>
<td>31.5114</td>
</tr>
<tr>
<td>Mean CHT</td>
<td>42.5687</td>
<td>5.48 x 10^{-6}</td>
<td>52.6147</td>
<td>31.8739</td>
</tr>
<tr>
<td>Zero KLT</td>
<td>313.557</td>
<td>4.03 x 10^{-5}</td>
<td>43.9424</td>
<td>23.2016</td>
</tr>
<tr>
<td>Mean KLT</td>
<td>260.008</td>
<td>3.34 x 10^{-5}</td>
<td>44.7557</td>
<td>24.0149</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Reduction type</th>
<th>MSE</th>
<th>NMSE</th>
<th>SNR, dB</th>
<th>PSNR, dB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zero DFT</td>
<td>20.2979</td>
<td>3.53 x 10^{-6}</td>
<td>54.5211</td>
<td>35.0903</td>
</tr>
<tr>
<td>Mean DFT</td>
<td>17.7615</td>
<td>3.08 x 10^{-6}</td>
<td>55.1008</td>
<td>35.6699</td>
</tr>
<tr>
<td>Zero DCT</td>
<td>-2.0464</td>
<td>-3.5 x 10^{-7}</td>
<td>64.8485</td>
<td>55.0457</td>
</tr>
<tr>
<td>Mean DCT</td>
<td>-2.0174</td>
<td>-3.5 x 10^{-7}</td>
<td>64.5477</td>
<td>55.1168</td>
</tr>
<tr>
<td>Zero WHT</td>
<td>3.7753</td>
<td>6.56 x 10^{-6}</td>
<td>61.8261</td>
<td>42.3952</td>
</tr>
<tr>
<td>Mean WHT</td>
<td>0.8382</td>
<td>1.45 x 10^{-6}</td>
<td>68.3620</td>
<td>48.9312</td>
</tr>
<tr>
<td>Zero CHT</td>
<td>0.8032</td>
<td>1.39 x 10^{-6}</td>
<td>68.5470</td>
<td>49.1162</td>
</tr>
<tr>
<td>Mean CHT</td>
<td>0.3326</td>
<td>5.78 x 10^{-8}</td>
<td>72.3756</td>
<td>52.9438</td>
</tr>
<tr>
<td>Zero KLT</td>
<td>61.9106</td>
<td>1.07 x 10^{-5}</td>
<td>49.6779</td>
<td>30.2472</td>
</tr>
<tr>
<td>Mean KLT</td>
<td>55.0704</td>
<td>9.58 x 10^{-6}</td>
<td>50.1864</td>
<td>30.7556</td>
</tr>
<tr>
<td>55 Reduced Coefficients, 9 saved</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Zero DFT</td>
<td>22.203</td>
<td>3.86 x 10^{-6}</td>
<td>54.1313</td>
<td>34.7005</td>
</tr>
<tr>
<td>Mean DFT</td>
<td>20.814</td>
<td>3.62 x 10^{-6}</td>
<td>54.4119</td>
<td>34.9811</td>
</tr>
<tr>
<td>Zero DCT</td>
<td>-3.8473</td>
<td>-6.7 x 10^{-7}</td>
<td>61.7440</td>
<td>42.3132</td>
</tr>
<tr>
<td>Mean DCT</td>
<td>-4.0750</td>
<td>-7.1 x 10^{-7}</td>
<td>61.4943</td>
<td>42.0635</td>
</tr>
<tr>
<td>Zero WHT</td>
<td>5.2066</td>
<td>9.06 x 10^{-6}</td>
<td>60.4300</td>
<td>40.9992</td>
</tr>
<tr>
<td>Mean WHT</td>
<td>3.7104</td>
<td>6.45 x 10^{-7}</td>
<td>68.3620</td>
<td>48.9312</td>
</tr>
<tr>
<td>Zero CHT</td>
<td>7.1197</td>
<td>1.24 x 10^{-6}</td>
<td>69.0710</td>
<td>49.6402</td>
</tr>
<tr>
<td>Mean CHT</td>
<td>6.1880</td>
<td>1.07 x 10^{-6}</td>
<td>69.6801</td>
<td>50.2493</td>
</tr>
<tr>
<td>Zero KLT</td>
<td>62.011</td>
<td>1.07 x 10^{-5}</td>
<td>49.6709</td>
<td>30.2400</td>
</tr>
<tr>
<td>Mean KLT</td>
<td>54.837</td>
<td>9.53 x 10^{-6}</td>
<td>50.2048</td>
<td>30.7740</td>
</tr>
</tbody>
</table>
In the first part of each table the results for 48 reduced coefficients (the upper-left block with size 4x4 are saved), approximated with zero and mean values are shown, and in the second part the same experiments for 55 reduced coefficients (the upper-left block with size 3x3 - saved) are shown. The calculated values for the mean-square error (MSE), normalized mean-square error (NMSE), signal to noise ratio (SNR) and peak signal to noise ratio (PSNR) are shown.

The general principles of comparison of 2D unitary transforms by the using of block transform coding of their coefficients of high order are given. The basic properties of CHT are discussed in previous publications. The obtained simulation results are practically identical for the CHT and real HT and show that both can be used in similar applications.

The best results are obtained for optimal Karhunen-Loeve transform and for the most used in compression standards discrete cosine transform. The results for the CHT are better then the DFT for small size of kernels.

The main advantages of the developed algorithm for CHT are:
- faster calculation compared to other transformations;
- similar results with integer valued HT;
- using the CHT instead most complicated Fourier transform and keep the possibilities for working with complex spectrum.

The developed Complex Hadamard Transform can be used in digital signal processing for spectral analysis, pattern recognition, digital watermarking, coding and transmission of one-dimensional and two-dimensional signals.

V. ACKNOWLEDGEMENT

The author thanks the National Fund for Scientific Research of the Bulgarian Ministry of Education and Science for the financial support by the contract I-02/1.

VI. REFERENCES

b) Output image “Fruits”  

a) Output image “Lenna”

Fig. 2. Output images after reduction with 48 coefficients for FFT, DCT, WHT, CHT and KLT.
Some Finger Vein Databases Evaluation
Ivo Draganov¹, Darko Brodić²

Abstract – In this paper a comparative evaluation is presented of some popular finger vein databases freely available for scientific research. The main aspects of the evaluation concern the quality of the images relating to their contrast, sharpness, present noise and distortions. Also quantitatively estimated values among images of one and the same database are presented. They are based on various measures which give an initial idea of the distribution of the visual representation of the veins for further recognition. Useful directions are made for image enhancement based on pre-processing prior to segmentation preceding the identification stage.

Keywords – Finger Vein Recognition, Database, Contrast, Sharpness, Noise

I. INTRODUCTION

Finger vein recognition becomes more popular in the recent years as additional modality available in biometric systems [1-7] that could be used separately or in combination with other modalities. Vein patterns captured in two-dimensional image representation by illuminating human finger either from above relying on transmissive principle or from below based on diffusive scattering and reflectance with near infrared light provide a number of advantages compared to other biometric features [2]. First of all they are considered unique for every human even between twins [3]. Since finger veins are located under the skin there is less probability for visual change over long period of time and in parallel to that the current outer condition of the finger is of no importance to the capturing process, e.g whether it’s dry, wet or contaminated [4]. The vein patterns are permanent with aging. Like other features they allow to check the liveness of the sample, i.e. it’s an effective measure against forgery [6]. The authentication is eased by the contactless way of acquisition. It is also thought to be zero-valued as for the Failure to enrolled rate (FER) and along with its other benefits qualified as highly secure and reliable.

Since a set of promising algorithms for finger vein recognition were proposed in the last decade [1-4] and still there are currently introduced enhancements to them an unified approach is needed for estimation of their overall performance and applicability to the practice. Considered as appropriate for incorporating in electronic passports and additionally assuring robust personal identification together with facial and fingerprint recognition [7] especially in the variable conditions in developing countries [5] a suitable database of finger veins needs to be selected as a starting point for the foundation of such unified testing framework. It should have a variety of abundant properties such as wide range of covered ages for the volunteers, equally spread from youngsters to elderly with equal presence between men and women. Different nationalities preferably from all continents are desirable to participate. Also more captured fingers from both hands would assure better selection of the proper ones for application and better evaluation of the tested algorithms will be made although the index and middle fingers are thought to be the most promising followed by the ring finger and much less attention is directed to the thumb and little finger. A few captures should be made of all selected fingers for testing with positional variations and partly such for the illumination conditions aiming to resemble real-case scenarios. Proper indexing preferably in the form of a related database and/or following specific xml schema is another requirement for efficient access while testing. All these features along with the objective measures for image quality are going to be evaluated below in this paper.

In Section II description is given for each of 5 evaluated databases, in Section III – quantitative analysis of the quality of the images from them and some considerations based on subjective inspection, and then in Section IV a conclusion is made.

II. EVALUATED DATABASES

SDUMLA-HMT database [8] is presented in 2010 by the Group of Machine Learning and Applications from Shandong University (SDUMLA) as a Homologous Multi-modal Traits (HMT) set. It consists of face images incorporating 7 view angles, finger vein images including 6 fingers per individual, iris images from a single sensor and fingerprint images obtained with the use of 5 different sensors. All this multimodal data is collected from 106 individuals. It is considered by its collectors as the first open database of this kind. The capturing device used for the finger vein images was developed by the Joint Lab for Intelligent Computing and Intelligent Systems of Wuhan University. Each participating individual in the collection process provided an image of his/her index, middle and ring fingers from both hands for totally 6 times. Thus 3816 images were gathered in bmp format with resolution of 320x240 pixels resulting in 0.85 GB database size.

The Hong Kong Polytechnic University Finger Image Database [9] has been gathered since April 2009 aiming large scale bundling of finger vein images publicly available for research. Along with the finger vein images finger surface texture images were also acquired from both male and female volunteers – a process which ended in March 2010 followed by database release in September 2010. A contactless capturing device was used producing 6264 images from 156 individuals in bmp format. Around 93% of the subjects were below 30 years of age at the time of the collection. It was...
done in 2 sessions separated from 2 to 6 months apart in each of which 6 images of the finger vein patterns of the index and middle fingers of the left hand and 6 finger texture images or 24 images totally per subject.

Idiap Research Institute in Martigny with the assistance of Haute EcoleSpécialisée de Suisse Occidentale in Sion, Switzerland, developed the VERA finger vein database [10]. It consists of 440 images from 110 subjects. An open finger vein sensor [11] was used for the purpose. Both index fingers were scanned of each subject in two sessions following in 5 minutes one after the other. Forty women and seventy men between 18 and 60 years old volunteered in the process. Images are in png format with resolution 665 x 250 pixels with an average size of 80 kB per file.

Another effort for free finger vein database for academic use was made by the Tsinghua University by putting together Finger Vein and Finger Dorsal Texture images (THU-FVFDT) [12]. All images were taken from 610 subjects in two sessions separated by a period from a few dozens of seconds up to a week where four finger vein and four finger dorsal texture images were captured. In the publicly available version of the database only one of these four images of each type is offered since the difference with the rest three is considered negligible. The resolution is 720 x 576 pixels and consequently a Region of Interest (ROI) was introduced with a size of 200 x 100 pixels covering the veins’ area for each picture.

Incorporating information for the finger veins along with finger geometry it becomes possible to verify unimodal or bimodal (fusion of vein data with outer shape of the fingers) biometrics systems as it is done in the Finger Vein UniversitiSains Malaysia (FV-USM) database [13]. A ROI was also introduced in it for the veins to be used in algorithms’ testing and comparing. Totally of 123 volunteers participated in the data collection of whom 83 were males and 40 females, mainly staff and students from the UniversitiSains Malaysia. Their age ranges from 20 to 52 years. Four fingers were captured – left index, left middle, right index and right middle which led to obtaining of 492 classes. Six snapshots were made for each finger in one session and after more than two weeks period the capturing was repeated and thus 5904 images were gathered as a result, with 2952 images from each session. Resolution is 640 x 480 pixels with 256 shades of gray per pixel.

III. DATABASE IMAGE PROPERTIES

In order to evaluate quantitatively the images from the databases described in Section II the approach described in [14] is employed where the following parameters are found on average for each one:

\[ \bar{I} = \frac{1}{P \cdot M \cdot N} \sum_{p=0}^{P-1} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} I(i, j), \] (1)

where \( \bar{I} \) is the average intensity of the images; \( P \) – their number in the current database; \( M \) and \( N \) – the dimensions of the images from it in number of pixels. The intensity \( I(i, j) \) for the current spatial position \((i, j)\) where \( i = 0, M - 1 \) and \( j = 0, N - 1 \) is changing between 0 and 255, i.e. in 8 bpp scale representing single tone variability.

Secondly, the average Root-Mean-Square (RMS) contrast \( \bar{C} \) for a whole database is found based on the definition:

\[ \bar{C} = \frac{1}{P} \sum_{p=0}^{P-1} \frac{1}{M \cdot N} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} [I(i, j) - \bar{I}]^2. \] (2)

Lastly, the average image entropy \( \bar{E} \), again for a whole database, is found using the following expression:

\[ \bar{E} = \frac{1}{P} \sum_{p=0}^{P-1} \left[ -\sum_{i=0}^{M-1} \sum_{j=0}^{N-1} h(I(i, j)) \log_2 (h(I(i, j))) \right], \] (3)

where \( h \) is the probability for particular intensity \( I \) to occur inside the image.

Sample images from the tested databases are given in Fig. 1.

![Sample finger vein images](image)

**Fig. 1.** Sample finger vein images from: a) SDUMLA-HMT, b) HKPUFID, c) VERA, d) THU-FVFDT, e) FV-USM databases

The results from calculating (1)-(3) for all 5 databases described above are given in Table I. The average intensity for the SDUMLA-HMT and HKPUFID are the same as those estimated in [14] but the average contrast and entropy differ since the calculations are performed for the images as a whole here rather than using mosaicking windows.

First of all an impression makes the significant difference for the mean intensity between FV-USM from below as minimum and HKPUFID as maximum more than 7 times. The other three databases are scattered in between at more than 2 and 4 times higher than the darkest set. Obviously the
intensity of the LEDs used and the sensitivity of the receiver influence considerably the quality of the images in general. Darker images not necessarily mean worse quality. As the average contrast shows no more than just 3 levels is the difference with the highest value for the SDUMLA-HMT in comparison to FV-USM and less with the others. From Fig. 1.e it is visible that there are no surrounding flashes from diffuse scattered light around the finger contour for FV-USM and the contrast itself is decent as a subjective estimate. Not that is the case for the images from Fig. 1.b, c, and d – considerable side illuminations are present which possibly could affect the recognition process further without additional pre-processing. Parts of the support equipment are also visible in the Fig. 1.a, b and d which is desirable to be removed before further processing possibly at the stage of finger contour detection.

Similar is the case for the average image entropy (Table I) – there is less than 2 levels deviation among evaluated databases – from 5.48 to 7.37. Nevertheless of the presence of uneven background due to scattered light, sustaining elements or other factors the homogeneity of the image structure is affected mostly by the blood vessels pattern and the smooth intensity change introduced by finger bones and tissue. It is a positive factor which indicates the possibility for higher recognition rates when combined with proper normalization procedures.

### Table I

<table>
<thead>
<tr>
<th>Parameter</th>
<th>SDUMLA-HMT</th>
<th>HKPUFID</th>
<th>VERA</th>
<th>THU-FVFDT</th>
<th>FV-USM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average Intensity</td>
<td>55.51</td>
<td>191.33</td>
<td>119.51</td>
<td>53.49</td>
<td>26.78</td>
</tr>
<tr>
<td>Average Contrast</td>
<td>10.72</td>
<td>10.06</td>
<td>9.61</td>
<td>8.09</td>
<td>7.67</td>
</tr>
<tr>
<td>Average Entropy</td>
<td>6.85</td>
<td>6.22</td>
<td>7.37</td>
<td>5.66</td>
<td>5.48</td>
</tr>
</tbody>
</table>

Detailed examination of the structure of the finger region inside tested images and the surrounding area pose additional considerations about the quality of veins’ structure (Fig. 2). For the first two cases – excerpts from the SDUMLA-HMT (Fig. 2.a) and HKPUFID (Fig. 2.b) there are significant portions of the finger area that possess almost even distribution of intensity for both darker and lighter areas. The presence of these isles as they could be called lead to smaller local contrast which in turn makes the veins’ pattern less distinctive over the tissue surroundings. This effect is considerably suppressed if not absent at all for the VERA, THU-FVFDT and FV-USM databases (Fig. 2.c, d and e). Although for the latter two the average entropy is smallest the intensity change around blood vessels has its repetitive nature underlying better their path. Various reasons may cause such differences among examined databases – from unsatisfactory set saturation levels (calibration) of sensitivity for the sensors accompanied by unwanted reflections from adjacent structures, through unsatisfactory initial image transformation, e.g. inappropriate intensity modification curve, or particular properties of the selected file format (mostly predetermined by the coder at use) for permanent storage as well as other factors. All of them should be carefully examined at the stage of design and implementation of the experimental setup for gathering the database and only as exception post-processing after the final files had been written would be recommended for further enhancement of image quality.

Image compression especially block-based such as the JPEG algorithm may additionally increase the effect of lowering the local contrast around veins. Since image dimensions are relatively small – of about 100x200 pixels or so for the Region of Interest (RoI) where the vessels are located and because of their repetitive structure with smaller number of intersections and small overall contrast it would be recommended to use some other compressor preserving more naturally continuous tones along separate directions such as the JPEG-2000 for example. Indexed-based formats, e.g. GIF, are also less perspective for such application. Of course, the optimal solution disregarding content size is to store data in a raw stream or in a popular uncompressed format such as the BMP as in some of the databases described here.

Noises are other aspect of the quality assessment of the finger vein database images. They are dealt with the magnitude fluctuation of the near-infrared light flow used for capturing the image and the level of thermogenerated currents inside the sensor. Due to the absence of raw data obtained with another type of illuminator and detector it is impossible to have quantitative estimation for their level and change over
time with modifications of the experimental setup. Visually a partial evaluation could be done, e.g. from Fig. 1 it is observed that with the lowering of the contrast (case d) and e)) the noise level is perceptibly higher (than that of the case for a) – c)). The selection of optical gathering system with appropriate filtering properties is another aspect to be considered when designing the complete finger vein sensor.

IV. CONCLUSION

In this paper evaluation is presented for 5 popular finger vein databases. It is based on objective comparison considering the average brightness, contrast and entropy of the images from each database as well as on visual inspection of the vein patterns captured, present noise, distortions and other factors concerning the overall perception of the content closely related to the recognition accuracy when identification stage takes place later. There is no particular parameter on which one can rely for simple selection of one database over the other for higher degree of confidence at testing newly developed recognition techniques. Moreover it is desirable to have images with implied imperfections such as higher noise level, distortions, diffused light, etc. in order to try unrepresentative capturing conditions which may be valid for some type of sensors from the practice. Useful directions for improving the capturing process are given in addition to the analysis of the factors leading to poor quality finger vein images which may be particularly effective for sensors design and their continuous exploitation.

ACKNOWLEDGEMENT

This work was supported by the National Scientific Fund at the Ministry of Education and Science, Republic of Bulgaria, within the project DFNI I02/1 "Intelligent man-machine interface for assistive medical systems in improving the independent living of motor disabled users".

REFERENCES


