Assessment of acrosome state in boar spermatozoa heads using n-contours descriptor and RLVQ

E. Alegre[a], M. Biehl[b], N. Petkov[b], L. Sanchez[c]

[a]*Department of Electrical, Systems and Automatic Engineering, University of Leon, Spain*
[b]*Johann Bernoulli Institute for Mathematics and Computer Science, University of Groningen, The Netherlands*
[c]*Department of Mechanical, Computing and Aerospace Engineering, University of Leon, Spain*

Abstract

This paper proposes a method for assessing the acrosome state of boar spermatozoa heads using digital image processing. We use gray level images in which spermatozoa have been labelled as acrosome-intact or acrosome damaged using the information of a coupled fluorescent image. The heads are segmented obtaining the outer head contour. A set of "n" inner contours separated by a logarithmic distance function is calculated later. For each point of the, in this case, seven contours a number of local texture features are computed. We have compared the classification performance of Relevance Learning Vector Quantization, Class Conditional Means and KNN, employing cross-validation for the evaluation. Gradient magnitude data offer the best result with an overall test error of only 1%. This result outperforms previously applied methods and suggests this approach as an interesting automatized approach to this veterinarian problem.

Key words: Acrosome assessment, image description, texture descriptor, classification, RLVQ

Email addresses: enrique.alegre@unileon.es (E. Alegre), m.biehl@rug.nl (M. Biehl), n.petkov@rug.nl (N. Petkov), lidia.sanchez@unileon.es (L. Sanchez)

1. Introduction

Semen quality assessment is a significant challenge in medical and veterinarian research whose solution can be applied both to people with fertility problems and for breed improvement of some species like boars. In the last case, one of the main applications is in the porcine industry that focuses its efforts on obtaining better individuals on each generation for human consumption. This industry has demanded systems that automate semen evaluation and such tools are highly welcome in the field. For several years, the computer vision industry has been offering Computer Assisted Semen Analysis (CASA) systems that allow to infer valuable information about the reproduction and, at the same time, to assess the semen in a quicker and more reliable way. At the beginning, these computer applications were developed for human semen analysis [1, 2], but later they have been adapted to other species [3, 4].

CASA systems such as the Hamilton-Thorne IVOS, the Microptic Sperm-Class Analyzer (SCA) or the Minitube SpermVision have attracted considerable interest in veterinary research. Researchers use these tools to measure a variety of properties such as the effect of sperm quality and fertility of the thawing boar semen in presence of plasma [5], the effects of organic peroxides over the prostatic secretory granules in rabbit semen[6], the ability of Catalonian donkey sperm to penetrate zona-pellucida [7] or the effects of antioxidant butylated hydroxytoluene on the sperm membrane integrity in frozen dog semen [8].

The main drawback of the current CASA systems is that they are only able to return sperm concentration, morphometry, motility and, more recently, droplets measures. Using phase contrast images, these systems do not perform assessment of the acrosome state or vitality. Therefore, those operations still have to be performed using stained images, mostly using fluorescent stains, and have to be assessed either manually by veterinary experts or with a CASA system that requires an expensive fluorescence microscope. Manual assessment has several drawbacks such as its high cost in terms of time, its lack of objectivity, or the requirement of specialized veterinary staff and equipments. The assessment using fluorescent stains with a CASA system, as we have pointed out, is expensive. Therefore, it is very interesting, and very profitable to companies providing artificial insemination, to have available methods to perform the automatic and objective classification of the acrosomes as intact or damaged based on graylevel phase-contrast images.

Following this idea, a number of digital image processing methods have been suggested for the segmentation, description and classification of sperm cells. Examples of this are the publications of Beletti et al., where the authors describe the spermatozoa head's morphology [9], while in [10], they characterize the animal sperm shape using a multiscale curvature estimation. Other authors, such as Linneberg et al. [11] describe the human sperm head morphology using invariant Fourier descriptors grouped in energy bands classifying them by means of neural networks.

From a different perspective, Sanchez et al. [12] proposed several methods for the classification of boar spermatozoa heads based on their intracellular intensity distribution observed in microscopic images. The authors used images labelled by veterinary experts and, by means of a proposed model, they classified the cells in function of the intensity distribution of their cytoplasm densities. Other methods for the vitality assessment based on gray level images have been suggested in the literature. For example, the authors of [13] proposed a new textural descriptor named LTP and they compared its performance with some classical descriptors, such as Pattern Spectrum, Flusser and Hu, obtaining hit rates around 70%. In [14] two textural descriptors, named NCSR and NCSH were proposed and compared with Zernike, Haralick and wavelet statistical features descriptors, with a best hit rate around 77%.

A number of publications are concerned with the segmentation problems associated with these cells. Some authors, e.g. Gonzalez et al. [15], addressed the head segmentation applying an intelligent thresholding. The threshold changes its value when the binary image obtained does not fulfil some surface and eccentricity conditions, and a watershed transform is applied when the previous method fails. Bijar et al. [16] proposed a method to segment sperm's acrosome, nucleus and mid-piece based on a Bayesian classifier which utilizes expectation-maximization and Markov random field to upgrade a class conditional probability density function.

In relation with the membrane integrity validation, in the last years it is possible to find some works addressing this problem. An early one was the proposed by Petkov et al. [17], that uses the gradient magnitude along the outer contour of the spermatozoa's head as a features vector and classify this vectors with LVQ. The authors of [17] use images of boar spermatozoa obtained with an optical phase-contrast microscope and try to automatically classify single sperm cells as acrosome-intact (class 1) or acrosome-damaged (class 2). The same authors [18] improve their method obtaining a hit rate

of 93.2%. Following a new point of view based on several texture descriptors, an improved hit rate of 94.93% was reached in [19] characterizing the spermatozoa heads with several Haralick descriptors computed after applying a discrete wavelet transform. Finally, a recent work of Gonzalez et al [20], using a descriptor based in the Curvelet Transform outperformed the previous approaches reaching a hit rate of up to 97%. In the following, we present a modified approach with which we achieve a hit rate of 99%.

The rest of the article is organized as follows: in Section 2 we will explain the methods used to detect the outer and inner contours of the sperm head images and to extract the proposed sets of features. The classification methods employed are presented in Section 3. Experimental results and achieved conclusions are summarized in Sections 4 and 5, respectively.

2. Methods

Here we describe the suggested methods for the automated segmentation, extraction of features and classification of the graylevel images.

2.1. Pre-processing and Segmentation

An optical phase-contrast microscope Nikon Eclipse $E-400$ (Nikon, Tochigi, Japan) that has an infinity optical system CFI60 and a parfocal distance of 60mm, and with a magnification of 100× was used. The objective used was the phase contrast DL 100× oil, with a NA of 1.25, a W.D. of 0.23mm and the thickness of the cover glass is 0.17 with spring loaded.

The boar semen images were acquired with a digital camera Nikon Coolpix 5000 choosing a resolution of 2560×1920 pixels per image. As the mounted sensor is a 2/3" CCD (8.8×6.6 mm), the pixel size is $3.4 \times 3.4\mu$m. The average size of the spermatozoa's heads was 160×90 pixels.

To analyze the performance of the proposed method, two different images of the same sample are captured. The first one has fluorescence illumination so that sperm cell presents a red color if its acrosome is intact or green if it is damaged (see Fig. 1 first row). The second one does not have fluorescence illumination, so it is a graylevel image of the sample (Fig. 1 second row). The samples were stained with hypogaea (peanut) agglutinin (PNA) that is a labelling method where the labelling is restricted to the acrosome and is not influenced by the fixation procedure. As it is known [21], the FITC-PNA binding site is mainly limited to the outer acrosomal membrane of boar

sperm, therefore it is an accurate test for studying boar sperm acrosome reaction.

In this work, the heads were cropped manually because we created the dataset by selecting in the fluorescent image, for example, an intact acrosome (heads only with red color) and cropping the corresponding head in the gray-level image by hand. An automatic cropping of all the heads in a sample can be carried out in a very straightforward way, with an automatic thresholding, some morphological operations and a final filtering by shape and size as it is presented in [15].

During the cropping step, we label each image as acrosome-intact or acrosome-damaged according to its corresponding cell color in the fluorescence image. Next, on the gray level image, we apply a segmentation method to detect the boundaries of the sperm head by combining different histogram adjustments, morphological operations and thresholding [17]. We also find the point where the tail intersects the outer contour of the cell and use it as a reference point.

2.2. Contour extraction

After the segmentation stage, a set of points (x_p, y_p), where $p = 1, 2, ..., L$ corresponding to the boundary of the sperm head image is obtained. From these points, we compute six additional contours $c_N = 6$, which are inside of the cell boundaries. Along these seven contours, we extract different sets of gradient, graylevel and textural features to classify images as acrosome-damaged or acrosome-intact.

First, we compute the centroid of the region delimited by the obtained boundary (x_c, y_c) as:

$$x_c = \frac{1}{L}\sum_{p=1}^{L} x_p, \quad y_c = \frac{1}{L}\sum_{p=1}^{L} y_p \tag{1}$$

Then we calculate the Euclidean distance of each point of the boundary (x_p, y_p) from the centroid:

$$d((x_p, y_p), (x_c, y_c)) = \sqrt{(x_p - x_c)^2 + (y_p - y_c)^2} \tag{2}$$

For each pixel (x_p, y_p) of the outer boundary c_b, its corresponding point (x_{inner}, y_{inner}) in an inner contour c_i, $i = 1, 2, ...6$ is determined by computing the following distance:

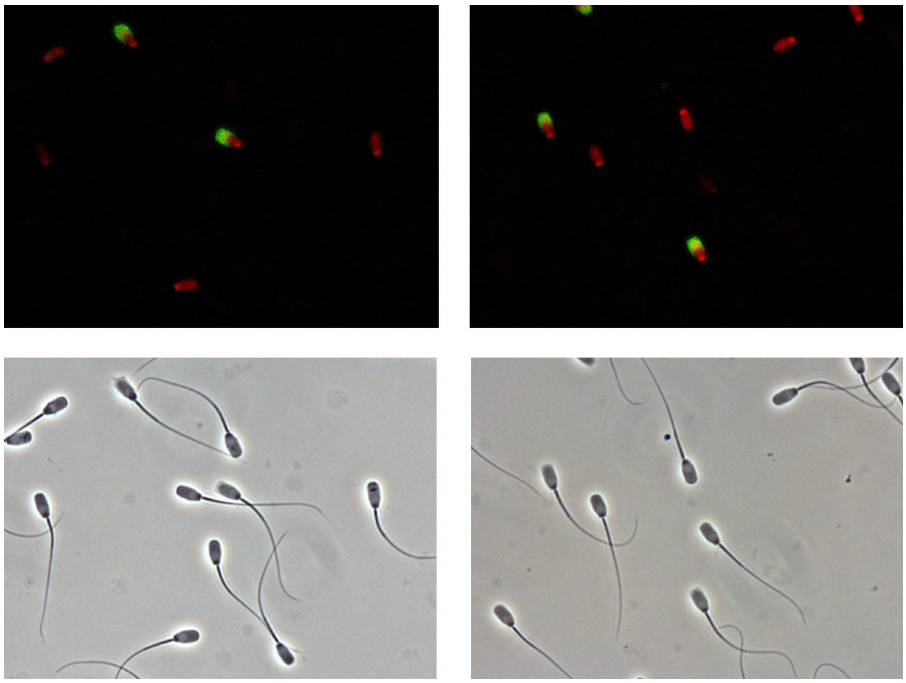

Figure 1: Examples of boar semen sample images acquired with (first row) and without (second row) fluorescence illumination. Stains mark spermatozoa whose acrosome is damaged (green) or intact (red).

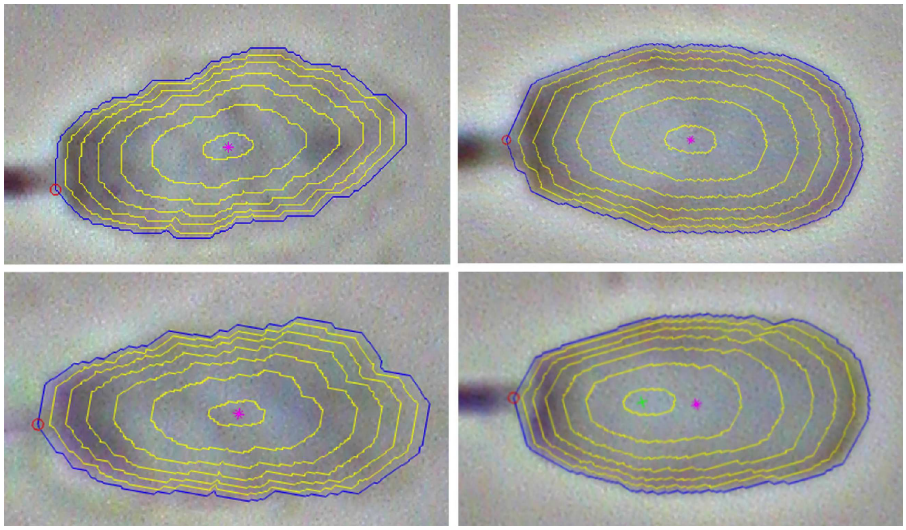

Figure 2: Once the boundary of the sperm head is obtained (blue contour), we compute six inner concentric contours (yellow contours) centred at the centroid (represented by a magenta asterisk). Contours start at the reference point represented by a red circle (which is the intersection between the boundary and the beginning of the tail). We do the same for both kind of images either labelled as acrosome-damaged (left column) or acrosome-intact (right column).

$$d((x_p, y_p), (x_{inner}, y_{inner})) = d((x, y), (x_c, y_c)) * \frac{c_i}{c_N + 1} * (1 - 0.7 * log_{10}(1 + c_N - c_i)) \quad (3)$$

We define the point (x_{inner}, y_{inner}) as the pixel which belongs to an imaginary line that links its corresponding point (x_p, y_p) of the outer contour c_b with the centroid (x_c, y_c) and it is $d((x_p, y_p), (x_{inner}, y_{inner}))$ pixels far away from the point (x_p, y_p).

The reason why we use a logarithmic distance is because we have observed that most of the texture differences, and therefore most of the information needed to differentiate the two classes of acrosomes is located inside the head, but in the closer to the border part of the spermatozoon head. In our experiments we also tested a linear distance obtaining worse results.

If we repeat this procedure for each point of the boundary and each contour c_i, we obtain c_N contours centred at the centroid (x_c, y_c) as it is illustrated in Figure 2. As we use a logarithmic distance, inner contours are closer when they are next to the outer contour.

The inner contours are computed obtaining first the coordinates for each point belonging to each inner contour. To do that, the distance obtained when applying the logarithmic distance function is rounded to the nearest pixel coordinate. Later, when each feature is computed, the gray level pixel values will be picked up from the appropriate image. For example, the gray level values will be picked up from the gray level image and the gradient value for each contour pixel will be picked up from the gradient image, that is the image resulting after applying a gradient transformation to the original one.

We also employ a shifted centroid (x_{sc}, y_{sc}) obtained after shifting the centroid a 30% of the distance to the reference point. We obtain another set of inner contours by considering the shifted centroid in the equations 2 and 3. Examples of the contours obtained using the shifted centroid are showed in Figure 3. We can see also the outer contour, the centroid point, the shifted centroid point and the reference point.

2.3. Feature extraction

Once we have obtained the seven contours centred at the centroid and the seven ones centred at the shifted centroid (x_{p_i}, y_{p_i}) where $p = 1, 2, ...L$ and $i = 1, 2, ...7$, we consider different sets of texture features for a neighborhood of each point of the computed contours:

(4)

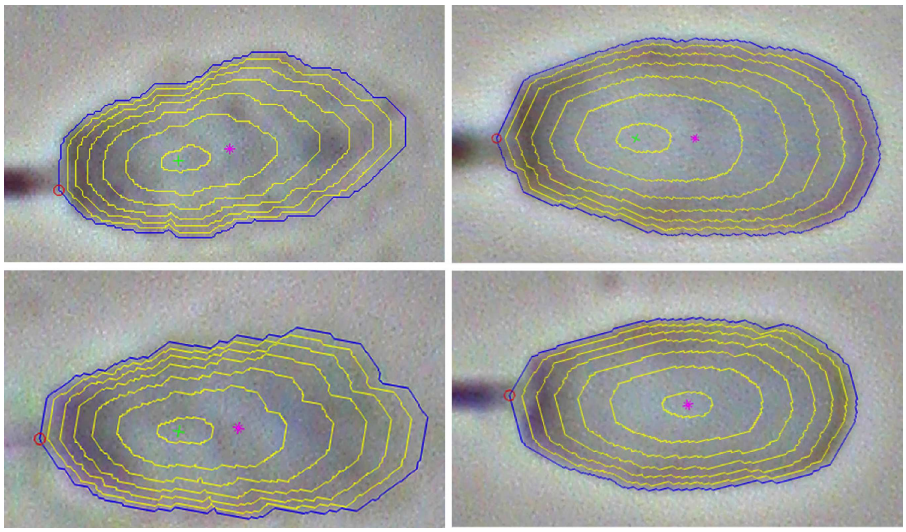

Figure 3: Examples of obtained contours for acrosome-damaged (left column) and acrosome-intact (right column) images when the shifted centroid (represented by a green plus sign) is considered. We also show the centroid (magenta asterisk) and the reference point (red circle).

- Local maximum gradient values.
- Local mean of the graylevel values.
- Local standard deviation.

2.3.1. Gradient values.

We compute the gradient of the image as [17]:

$$\bigtriangledown_{\sigma,x} f = f * \frac{\partial g_\sigma}{\partial x}, \quad \bigtriangledown_{\sigma,y} f = f * \frac{\partial g_\sigma}{\partial y} \tag{5}$$

To extract the features, we consider the magnitude $M_\sigma(x, y)$ of the scale-dependent gradient [17]:

$$M_\sigma(x, y) = \sqrt{\left(\bigtriangledown_{\sigma,x} f(x, y)\right)^2 + \left(\bigtriangledown_{\sigma,y} f(x, y)\right)^2} \tag{6}$$

Next, we consider a 5-by-5 neighborhood N around each pixel of each computed contour and we find the pixel that has the maximum gradient value, obtaining a feature v_i for each pixel:

$$v_i = \max_{(x,y)\in N} (M_\sigma(x, y)) \tag{7}$$

The starting point is always the reference point (or its corresponding point in the inner contours) and we go along the contours clockwise. We interpolate each feature vector extracted for the considered contours to a constant size of 40 elements. So, a feature vector of 280 elements (40 features * 7 contours) is formed.

2.3.2. Graylevel values.

We consider a 3-by-3 neighborhood N around each pixel of each contour and compute the mean of the graylevels of the pixels of the image that belong to such neighborhood:

$$v_i = \frac{1}{9} \sum_{(x,y)\in N} I(x, y) \tag{8}$$

So, we obtain a feature for each pixel of a contour. This feature vector is interpolated using a Nearest-neighbor interpolation to a constant size of 40 items. We carry out the same procedure for each contour so 7 feature vectors

of 40 elements are calculated. We form a vector of 280 features (40 features $*$ 7 contours) that is used for the classification.

2.3.3. Local standard deviation.

We computed the standard deviation of the gray levels of the pixels that belong to a 3×3 neighborhood, N, around a given point of a contour. The size of the neighbourhoods was selected experimentally, for the three different methods presented. The size was chosen based on the classification results, after computing neighbourhoods of sizes 3×3, 5×5, 7×7 and 9×9.

For each pixel belonging to the contour, a feature was obtained in the following way:

$$v_i = \sqrt{\frac{1}{n-1} \sum_{(x,y) \in N} (I(x,y) - \mu_I)^2} \tag{9}$$

where n is the size of the neighborhood N and μ_I is the mean of the graylevels of the pixels that belong to the neighborhood $\mu_I = \frac{1}{n} \sum_{(x,y) \in N} I(x,y)$.

As we obtain a feature for each point of the contour, we form a feature vector that we resize by interpolation to 40 elements. Calculating this feature vector for each contour, 7 vectors of 40 features are obtained. They formed a vector of 280 elements that is used for classification.

3. Classification

In the following we analyse the performance of distance based classification schemes which are obtained from the available example data. Our main interest is in prototype based classifiers which identify typical representatives of the classes by means of Learning Vector Quantization (LVQ). LVQ is easy to implement and so-called relevance learning schemes can be included which allow for an adaptive weighting of features. The aim is to improve the performance and, at the same time, potentially simplify the classifier by selection of the most relevant features. We also compare with the k-nearest-neighbor (KNN) approach, which makes use of all example data in the classification.

Representation of feature vectors:

In the following, the construction or training of classifiers is based on subsets of all available labeled data which we denote as $I\!D = \{\boldsymbol{\xi}^\mu, S_T^\mu\}_{\mu=1}^M$.

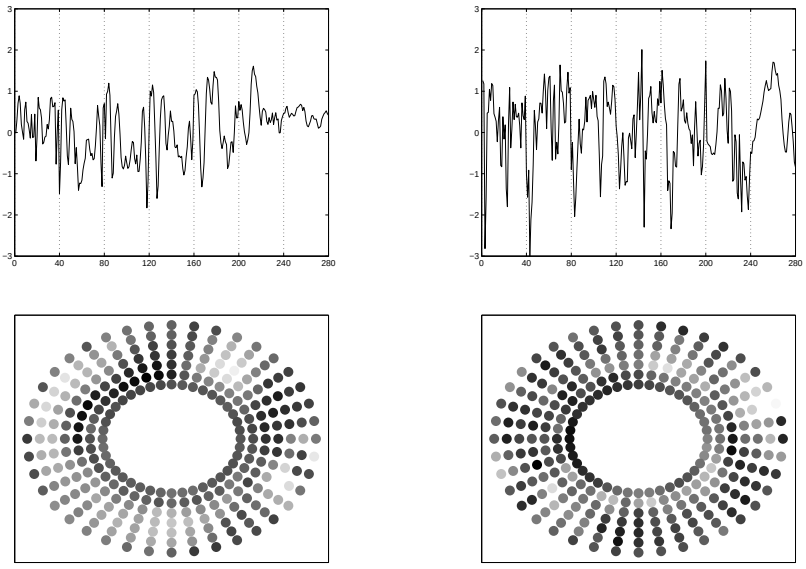

Figure 4: Examples of z-transformed graylevel profiles $\boldsymbol{\xi} \in I\!R^{280}$ from class 1 (acrosome-intact, left column) and class 2 (acrosome-reacted or reacting, right column) The discrete 1D functions (upper row) represent the vectors used for LVQ as ξ_i vs. index i. Vertical dotted lines separate the seven contours; the first values $\xi_1, \ldots \xi_{40}$ correspond to the outermost contour etc.
The same graylevel profiles are displayed in the lower row, schematically taking into account the topology of the contour grid. Black (white) dots correspond to the highest (lowest) graylevel in the respective 280-dim. vector.

Here, the class membership of training examples is denoted as $S_T^\mu \in \{1, 2\}$. Components of vectors $\boldsymbol{\xi}^\mu \in I\!R^N$ ($N = 280$) are obtained from the features v_i representing, for instance, graylevel data or local gradient magnitudes. To all data sets we apply a z-transformation, such that the transformed values display zero mean and unit variance over the set of available data:

$$\frac{1}{P}\sum_{\mu=1}^{P} \xi_i^\mu = 0 \quad \text{and} \quad \frac{1}{P}\sum_{\mu=1}^{P} (\xi_i^\mu)^2 = 1 \quad \text{for} \quad i = 1, 2, \ldots, N$$

The transformation influences the classification performances discussed in the following only very weakly. However, it facilitates a straightforward interpretation of the relevance factors which we define and consider below.

In total, $M = 360$ examples are available, 210 of which represent class 1 (acrosome-intact) while 150 belong to class 2 (acrosome-reacted or reacting). Fig. 4 (upper row) shows one example profile from each of the two classes, in this case displaying z-transformed graylevels ξ_j vs. index j. Alternatively, we display feature vectors in a manner which preserves the topology of the contour grid, see Fig. 4 (lower row).

Distance measure:

Throughout the following we will use a modified quadratic Euclidean distance measure when comparing two N-dimensional vectors $\boldsymbol{x}, \boldsymbol{y} \in I\!R^N$:

$$d_\lambda(\boldsymbol{x}, \boldsymbol{y}) \;=\; \sum_{j=1}^{N} \lambda_j \; (x_j - y_j)^2 \,. \tag{10}$$

Here, a relevance factor λ_j controls the role of feature component j in the evaluation of distances. The factors are non-negative, $\lambda_j \geq 0$, and obey the normalization $\sum_{j=1}^{N} \lambda_j = 1$.

In the simplest case $\lambda_j = 1/N$ for all j, the distance measure (10) corresponds to the generic quadratic Euclidean distance, which we denote as $d_o(\boldsymbol{x}, \boldsymbol{y}) = (\boldsymbol{x} - \boldsymbol{y})^2/N$. The introduction of relevance factors would be analogous for other measures, e.g. for

$$d_\lambda^{(q)}(\boldsymbol{x}, \boldsymbol{y}) = \sum_{j=1}^{N} \lambda_j \; |x_j - y_j|^q$$

including the special case of the Manhattan distance with $q = 1$. The classifiers analysed in the following exhibit almost identical performance for $q = 1$ and $q = 2$, while classification errors increase for larger values of q. Hence, we restrict the discussion to the use of modified Euclidean distances with $q = 2$ in the following.

3.1. Distance based classifiers

3.1.1. K-Nearest-Neighbor classification

In order to evaluate the performance of this standard approach we determine for a given feature vector $\boldsymbol{\xi}^\nu \in I\!D$ all Euclidean distances $d_o(\boldsymbol{\xi}^\mu, \boldsymbol{\xi}^\nu)$ with $\mu \neq \nu$. The majority of labels S_T^μ among the k closest vectors $\boldsymbol{\xi}^\mu$ then determines to which class $\boldsymbol{\xi}^\nu$ is assigned, denoted as $S_{knn}^\nu = \pm 1$.

The fraction of vectors $\boldsymbol{\xi}^\nu$ with $S^\nu_{knn} \neq S^\nu_T$ corresponds to a Leave-One-Out estimation of the over-all classification error and will be denoted as ε_{knn}. Analogously, the errors $\varepsilon^{(1)}_{knn}$ and $\varepsilon^{(2)}_{knn}$ quantify the performance with respect to data from class 1 or class 2 only, respectively.

3.1.2. Class conditional means

The above discussed KNN approach requires explicit storage of all example data and involves the evaluation of many distances for each classification. Hence, it is preferential to represent the example data by only a few *prototype* vectors. Novel data can then be labeled according to a nearest prototype classification (NPC) scheme with much lower computational effort.

The conceptually simplest set of prototypes is obtained from a training set containing P labeled examples $\{\boldsymbol{\xi}^\mu, S^\mu_T\}$ by evaluating the class-conditional means (CCM)

$$\boldsymbol{m}_1 = \frac{1}{P_1} \sum_{\mu=1}^{P} \boldsymbol{\xi}^\mu \, \delta(S^\mu_T, 1) \quad \text{and} \quad \boldsymbol{m}_2 = \frac{1}{P_2} \sum_{\mu=1}^{P} \boldsymbol{\xi}^\mu \, \delta(S^\mu_T, 2) \tag{11}$$

where $\delta(k,l)$ is the Kronecker-delta and $P_S = \sum_\mu \delta(S^\mu_T, S)$ is the number of examples from class $S = 1, 2$, respectively. The resulting classifier assigns a vector $\boldsymbol{\xi}$ to class 1 if $d_o(\boldsymbol{m}_1, \boldsymbol{\xi}) \leq d_o(\boldsymbol{m}_2, \boldsymbol{\xi})$ and to class 2 else.

3.1.3. Relevance Learning Vector Quantization

We apply Learning Vector Quantization (LVQ) for the identification of class prototypes. LVQ was originally proposed by Kohonen and has been used in a variety of problems due to its flexibility and conceptual clarity. Here we employ the original LVQ1 algorithm [22, 23], together with a heuristic relevance update which adapts the distance measure in the course of learning.

A set of prototypes $\left\{\boldsymbol{w}^1, \boldsymbol{w}^2, \ldots, \boldsymbol{w}^K\right\}$ with $\boldsymbol{w}^j \in I\!R^N$ is to represent the classes according to the associated labels $S^j \in \{1, 2\}$. These assignments as well as the number of prototypes have to be specified prior to learning.

At each time step t of the iterative training procedure, one example $\{\boldsymbol{\xi}^\mu, S^\mu_T\}$ is selected randomly from the training set. The modified distances $d_\lambda(j, \mu) = d_\lambda(\boldsymbol{\xi}^\mu, \boldsymbol{w}^j(t))$ from all prototype vectors $\boldsymbol{w}^j(t)$ are evaluated according to Eq. (10) with the current relevances $\lambda_j(t)$. Next we identify the minimal distance $d_\lambda(J, \mu)$ among all prototypes and the corresponding *winner*

$$\boldsymbol{w}^J(t) \quad \text{with} \quad d_\lambda(J, \mu) = \min_k \left\{ d_\lambda(k, \mu) \right\}. \tag{12}$$

In LVQ1, only this winner is updated according to

$$\boldsymbol{w}^J(t+1) = \boldsymbol{w}^J(t) + \eta_w \left[2\,\delta(S_T^\mu, S^J) - 1\right] \left(\boldsymbol{\xi}^\mu - \boldsymbol{w}^J(t)\right), \qquad (13)$$

where the update is towards (away from) the actual input $\boldsymbol{\xi}^\mu$ if the class labels of winner and example agree (disagree). Note that the factor [...] in (13) is $+1$ if $S_T^\mu = S^J$ and -1 else.

Our heuristic realization of relevance Learning Vector Quantization (RLVQ) follows closely the prescription of [24]. In parallel with the prototype update (13), relevances are adapted as follows:

$$\begin{aligned} \tilde{\lambda}_j(t+1) &= \lambda_j(t) - \eta_\lambda \left[2\,\delta(S_T^\mu, S^J) - 1\right] \left|\xi_j^\mu - w_j^J(t)\right| \\ \lambda_j(t+1) &= \max\left\{0, \tilde{\lambda}_j(t+1)\right\} \Big/ \sum_{k=1}^{N} \max\left\{0, \tilde{\lambda}_k(t+1)\right\}, \end{aligned} \qquad (14)$$

where the second step implements the non-negativity condition and the required normalization. This prescription decreases relevance factor λ_j if, for instance, the winning prototype $\boldsymbol{w}^J$ does represent the correct class, $S^J = S_T^\mu$, but the contribution $\left(\xi_j^\mu - w_j^J\right)^2$ of feature j to $d_\lambda(J, \mu)$ is relatively large. On the contrary, the relevance of features with relatively small $\left|\xi_j^\mu - w_j^J\right|$ will increase in such a case. Thus, after performing (14), the measured distance will be smaller upon presentation of the same or a similar feature vector, resulting in an even higher probability for correct classification.

In the following studies we employ typically only one prototype vector per class which is initialized in the corresponding class conditional mean.

The learning rates η_w, η_λ in Eqs. (13, 14) control the step width of the iteration. We will demonstrate in the following sections that excellent performance can be achieved already with constant rates $\eta_w = 10^{-2}$ and $\eta_\lambda = 10^{-6}$, for example. This choice reflects the plausible requirement that the distance measure should be changed less rapidly than the prototype positions themselves. A further optimization by employing fine-tuned time-dependent learning rates is beyond the scope of this publication and will be addressed in a forthcoming project.

3.1.4. Validation

In order to obtain estimates of the performance after training, we split the set of 360 available training data randomly into disjoint subsets of $P =$

240 examples for training and 120 test data. This procedure is repeated $n_{rep} = 50$ times, each time for a different random split of data. Performing the corresponding averages reduces random fluctuations and the influence of *lucky set* compositions.

In the following, ε_{train} denotes the fraction of misclassified example data, obtained after training and on average over the 50 runs. Correspondingly, the test error ε_{test} quantifies the averaged performance with respect to the test set. We furthermore evaluate the class-specific test errors $\varepsilon_{test}^{(1)}$ and $\varepsilon_{test}^{(2)}$ as well as the training errors $\varepsilon_{train}^{(1)}$ and $\varepsilon_{train}^{(2)}$ with respect to only class 1 or class 2 data, respectively.

The main purpose of this validation scheme is to evaluate the test errors of the LVQ schemes as a function of training time t/P, i.e. the number of sweeps through the data set. It will be used to identify the best performance achieved in the course of training. We also apply it to the classifiers constructed from the class-conditional means, while for the KNN classifier we resort to the above described leave-one-out estimate.

3.1.5. Receiver Operating Characteristics

An important quality measure of classification systems is the so-called Receiver Operating Characteristics (ROC). In the distance based CCM and LVQ systems with two prototypes we introduce an additional threshold γ and redefine the classification scheme as

$$S(\boldsymbol{\xi}) = \begin{cases} 1 & \text{if} \quad d_\lambda(\boldsymbol{\xi}, \boldsymbol{w}^1) \, - \, \gamma \; < \; d_\lambda(\boldsymbol{\xi}, \boldsymbol{w}^2) \\ 2 & \text{else.} \end{cases} \quad , \tag{15}$$

with $\boldsymbol{w}_i = \boldsymbol{m}_i$ in the case of the CCM classifier.

By choosing positive (negative) values of γ, the classification can be biased to label more inputs as class 1 (class 2), respectively. The original NPC is recovered for $\gamma = 0$. Corresponding pairs of class-specific test errors are plotted as $1 - \varepsilon_{test}^{(2)}(\gamma)$ (correct hit rate) vs. $\varepsilon_{test}^{(1)}(\gamma)$ (false alarm rate) in the ROC curve. The resulting deviation from the trivial behavior of a biased, random labeling with $(1 - \varepsilon_{test}^{(2)}) = \varepsilon_{test}^{(1)}$, is an indicator of the classification quality.

KNN classifier	ε_{knn}	$\varepsilon_{knn}^{(1)}$	$\varepsilon_{knn}^{(2)}$
graylevel data			
k=1	6.4	1.4	13.3
k=3	5.6	2.4	10.0
gradient magnitudes			
k=1	3.1	2.4	4.0
k=11	1.7	1.0	2.7
local std. deviations			
k=1	8.6	2.9	16.7
k=5	3.9	1.4	7.3

Table 1: Leave-One-Out error estimates (in %) of KNN classifiers for graylevel data, local gradient magnitudes, and local standard deviations. For each data set, the result for $k = 1$ and the corresponding optimal neighborhood are shown.

4. Results

4.1. KNN classification

Applying KNN classification schemes provides a first insight into the performance achievable by distance based classifiers. Table 1 shows the results for the nearest neighbor classifier with $k = 1$ and the best result obtained by optimal choice of k for each type of data.

Note that these results suggest to employ the gradient magnitude data for the classification as it gives the best over-all performance and relatively balanced class-conditional error rates.

4.2. Class conditional means

Next we consider the nearest prototype classifier based on class conditional means, cf. (11). The validation scheme described in Sec. 3.1.4 provides estimates of the training and test errors which are summarized in Table 2. Note that for all data sets the CCM classifier is comparable with or outperforms KNN schemes even with optimized values of k.

Here, the results are comparable when using gradient magnitudes and local standard deviations. Both types of data are similar in spirit and are to be peferred over the naive use of graylevel values.

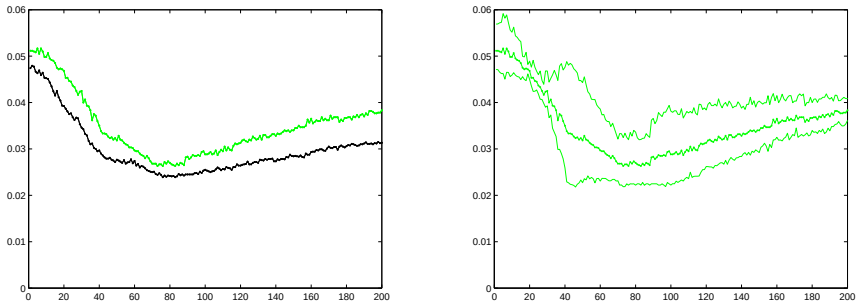

Figure 5: **Learning from graylevel data.** Left: test error (upper) and training error (lower curve) vs. the number of sweeps through the training set in RLVQ training. Right: total test error (center), class-specific errors $\varepsilon_{test}^{(1)}$ (bottom) and $\varepsilon_{test}^{(2)}$ (top). Results are averaged over 50 randomly shuffled data sets.

CCM classifier	ε	$\varepsilon^{(1)}$	$\varepsilon^{(2)}$
graylevel data			
training set	4.7	4.3	5.4
test set	5.1	4.7	5.7
gradient magnitudes			
training set	1.9	2.3	1.4
test set	2.2	2.0	2.4
local std. deviations			
training set	1.7	1.9	1.4
test set	2.1	2.7	1.1

Table 2: Error estimates of the CCM classifiers (in %) for graylevel data, gradient magnitudes, and local standard deviations.

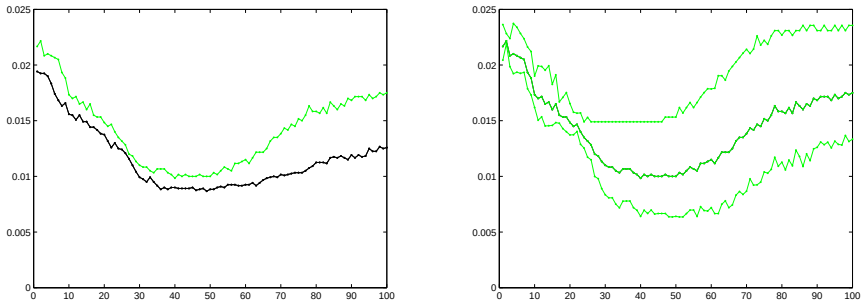

Figure 6: **Learning from gradient data,** see Fig. 5 for a detailed explanation.

4.3. RLVQ classification

Finally we study the classifiers obtained by RLVQ training. Here, we employ one prototype per class only which is placed in the corresponding CCM, initially. In the course of training, cf. (13,14), we monitor the performance and obtain so-called learning curves: training and test errors as a function of training time (randomized sweeps through the training set).

Qualitatively we observe the same behavior for all data sets, which we discuss first in the context of the graylevel data. Initially, prototypes are placed in the CCM and relevances are set equal, $\lambda_j = 1/N$. Hence, initial performances correspond to Table 2. We observe that, in the RLVQ training process, prototypes change only very little and remain very close to the CCM. However, significant improvement is achieved by adaptation of the relevances. Generically, a minimum of training and test error is reached after a number of sweeps which can be clearly identified by means of the validation scheme. The behavior is examplified in Figure 5 for training with graylevel data, where the best test and training errors are reached after about 80 sweeps through the training set. Figure 7 shows the prototype configurations in this stage of training. The corresponding, non-trivial relevance profile is displayed in Fig. 8 (left). Note that a significant number of features is effectively neglected as signaled by $\lambda_j \approx 0$; they could be removed from the dataset without affecting the classification performance.

If further training is performed, the relevance profile becomes more pronounced and RLVQ over-simplifies the classifier, yielding increased error rates, cf. Figure 5. Note that this behavior is not related to over-fitting effects because (a) the complexity of the system is reduced in the course of training and (b) it affects both training and test error. The right panel of Fig.

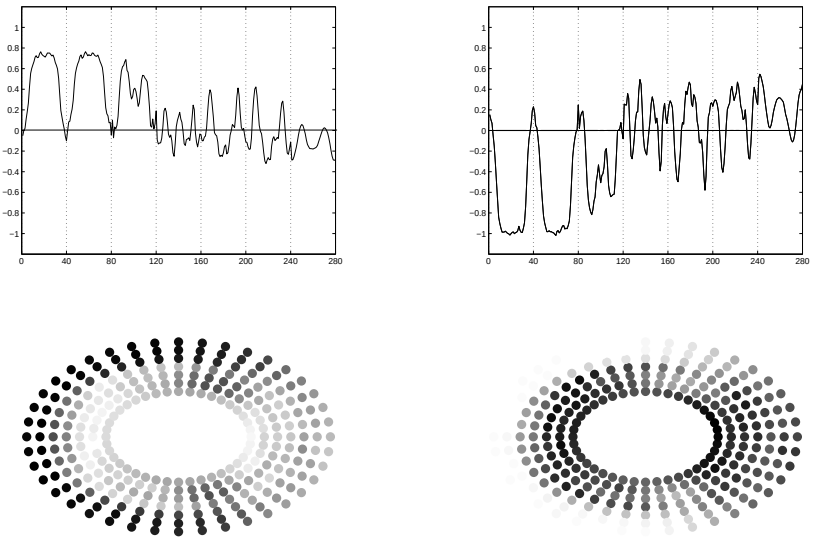

Figure 7: **Graylevel data:** Prototypes obtained by RLVQ after 80 sweeps through the example data, corresponding to the minimum of the learnign curves in Fig. 5. An average over 50 randomly shuffled data sets was performed. The class 1 prototype $\boldsymbol{w}^{(1)}$ is displayed in the left panels, the right panels show $\boldsymbol{w}^{(2)}$.

8 shows the relevance configuration after 200 sweeps through the training set, i.e. at the end point of learning curves in Fig. 5. Obviously, RLVQ assigns the greatest importance to graylevels in the outermost shells, in addition a few values in the center are taken into account by the system as well.

As it is not possible to aim at the direct minimization of errors, non-monotonic learning curves are well possible and frequently observed in classification problems. The above discussed behavior suggests to introduce regularization terms into the update rules (13,14) which would allow to control the non-uniformity of relevance profiles by means of an additional parameter. Here we resort to the simpler *early stopping* strategy and focus on the best behavior obtained in the minimum of the learning curves.

For data sets containing vectors of local standard deviation or gradient magnitudes, we observe the same qualitative behavior, see Figure 6 for the learning curves in the latter case. However, the achievable performance is better than for graylevel data and the location of the most relevant features

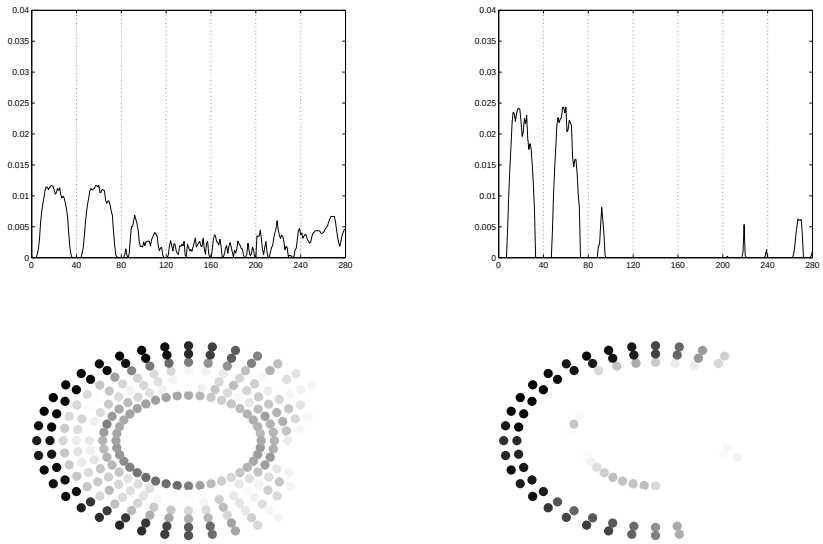

Figure 8: **Graylevel data:** Relevance profile in the minimum of the learning curve (left panels) and in the oversimplified configuration after 200 sweeps (right), cf. Fig. 5. An average over 50 random splits of the data was performed.

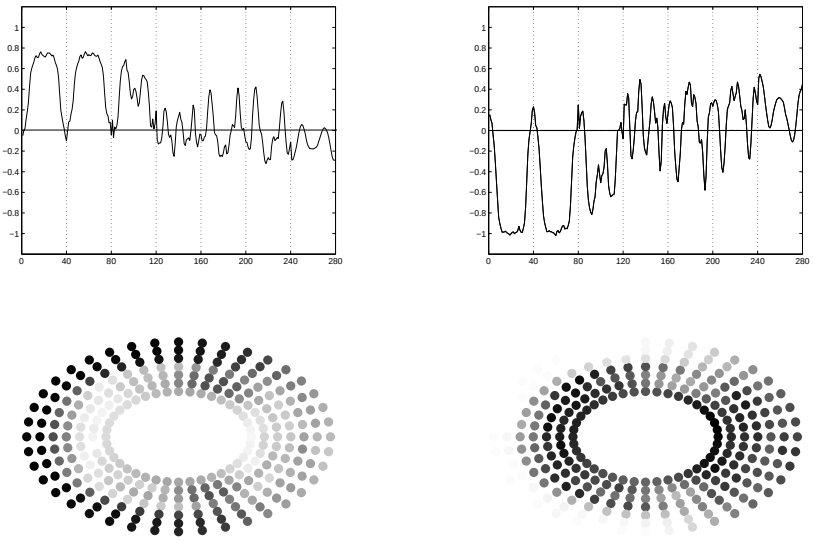

Figure 9: **Gradient data:** Prototypes obtained by RLVQ in the minimum of the learning curves, cf. Fig. 6, after 45 sweeps through the example data, on average over 50 randomly shuffled data sets. The class 1 prototype $\boldsymbol{w}^{(1)}$ is displayed in the left panels, the right panels show $\boldsymbol{w}^{(2)}$.

differs significantly. Figure 10, left panel, displays the relevance profile corresponding to the minimum of the learning curve, cf. Fig. 6. The right panel corresponds to an oversimplified system with only slightly larger errors after performing 100 sweeps through the training set. Here, the classification is essentially making use of the two outer shells of the grid, only.

In Table 3 we summarize the outcome of RLVQ training for the different data sets considered here. The listed values correspond to the best performance obtained in the respective minimum of the learning curves. Gradient magnitude data appears to be the most appropriate one for this classification problem and we obtain an over-all test error of about only 1%. Furthermore, the negative effect of over-simplification is least pronounced for the gradient magnitude data.

The Receiver Operator Characteristics provide further insight into the quality of the classification schemes. Figure 11 (left panel) displays the ROC curves corresponding to the three data sets. Also here, the classification

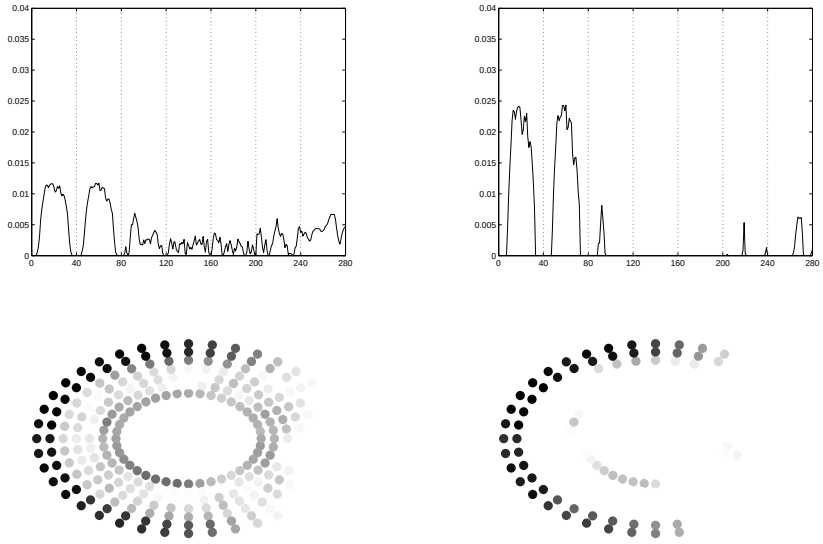

Figure 10: **Gradient data:** Relevance profile in the minimum of the learning curve (left panels) and in the oversimplified configuration after 100 sweeps (right), cf. Fig. 6. An average over 50 randomly shuffled data sets was performed.

RLVQ classifier	ε	$\varepsilon^{(1)}$	$\varepsilon^{(2)}$
graylevel data			
training set	2.4	2.3	2.7
test set	2.7	2.2	3.4
gradient magnitudes			
training set	0.9	0.6	1.3
test set	1.0	0.7	1.5
local std. deviations			
training set	1.4	1.4	1.4
test set	1.7	2.1	1.3

Table 3: Error estimates of the RLVQ classifiers (in %) for graylevel data, gradient magnitudes, and local standard deviations. Results correspond to the best achieved performance, i.e. in respective minima of the learning curves.

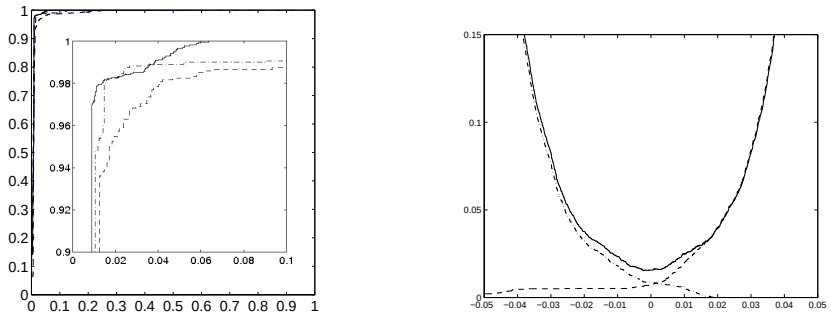

Figure 11: **Receiver Operating Characteristics. Left panel:** Comparison of the ROC curves obtained for gradient magnitude data (solid line), local standard deviations (dash-dotted), and graylevel data (dashed line). The inset shows the upper left corner of the ROC plot in detail. **Right panel:** Characteristics of the RLVQ classifier based on gradient magnitude data in the minimum of the learning curve. The solid line corresponds to the averaged test error ε_{test} as a function of the threshold γ, see Eq. (15). The dashed line marks $p_1\,\varepsilon^{(1)}_{test}$, the dashed-dotted line corresponds to $p_2\varepsilon^{(2)}_{test}$, respectively. Here, $p_1 = 210/360$ is the overall frequency of class 1 data and $p_2 = 1 - p_1$.

based on gradient magnitudes outperforms the other variants, apart from a very small region where the use of local standard deviations seems favorable.

The right panel of Fig. 11 shows the dependence of the total and class conditional errors for gradient magnitude data on the threshold γ in Eq. (15). Note that the best performance is achieved by the unmodified classifier with $\gamma = 0$, indeed.

5. Summary and Outlook

In this paper we propose three different sets of descriptors to determine the state of the head's acrosome indicative of fertilization capacity. For this purpose, the texture of each acrosome is represented by a 280 element vector. This vector consists of 40 features computed along seven concentric contours. The first contour corresponds to the perimeter of the head and the remainder are logarithm spaced inner contours. A feature of each contour takes as value the local maximum gradient, the local mean of the graylevels or the local standard deviation.

For classifying the three proposed sets of vectors we have considered a k-nearest-neighbour with Leave-One-Out cross-validation, a Class Conditional

Means approach and a Relevance Learning Vector Quantization with k-fold cross-validation and k=50. Each classifier has compared the 280 feature vectors obtained for the considered descriptors: gradient, gray level mean and local standard deviation.

KNN yields a hit rate of 98.3% with gradient magnitudes and k=11. Considering local standard deviation the hit rate is 96.1%, while for graylevel it is no higher than 94.4%. For Class Conditional Means and gradient magnitude the hit rate is 97.8%, increasing to a 97.9% when including local standard deviations. The graylevel features keeps obtaining the worst results with a 94.9% of hit rate. Finally, we obtain the best approach considering gradient magnitudes and RLVQ achieving a hit rate of 99%. For local standard deviation the hit rate reaches 98.3%.

The obtained results are promising and clearly better than previously published works [17, 18, 19, 20], but they should be confirmed through testing on independent samples. If that confirms our results, we will apply this method to the vitality assessment problem, trying to improve the hit rates obtained in previous works [13, 14].

Acknowledgments

This work has been partially supported by the research project DPI2009-08424 from the Spanish Ministry of Education.

The authors would like to thank CENTROTEC for providing us the semen samples and for their collaboration in the image acquisition.

References

[1] M. F. Vine, J. R. Woodrow Setzer, R. B. Everson, A. J. Wyrobek, Human sperm morphometry and smoking, caffeine, and alcohol consumption, Reproductive Toxicology 11 (1997) 179–184.

[2] L. Ramos, J. C. M. Hendriks, P. Peelen, D. D. M. Braat, A. M. M. Wetzels, Use of computerized karyometric image analysis for evaluation of human spermatozoa, Journal of Andrology 23 (6) (2002) 882–888.

[3] J. Verstegen, M. Iguer-Ouada, K. Onclin, Computer assisted semen analyzers in andrology research and veterinary practice, Theriogenology 57 (2002) 149–179.

[4] J. Auger, C. Lesaffre, A. Bazire, D. Schoevaert-Brossault, F. Eustache, High-resolution image cytometry of rat sperm nuclear shape, size and chromatin status. experimental validation with the reproductive toxicant vinclozolin, Reproductive Toxicology 18 (2004) 775–783.

[5] J. Garcia, J. Dominguez, F. Penas, B. Alegre, R. Gonzalez, M. Castro, G. Habing, R. Krkwood, Thawing boar semen in the presence of seminal plasma Effects on sperm quality and fertility, Animal Reproduction Science 119 (1-2) (2010) 160–165.

[6] E. Mourvaki, R. Cardinali, A. D. Bosco, C. Castellin, In vitro antioxidant activity of the prostatic secretory granules in rabbit semen after exposure to organic peroxides, Reproductive Biology and Endocrinology 8 (16) (2010) 8–16.

[7] E. Tabernera, R. Moratoa, T. Mogasaand, J. Miro, Ability of catalonian donkey sperm to penetrate zona pellucida-free bovine oocytes matured in vitro, Animal Reproduction Science 118 (2–4) (2010) 354–361.

[8] V. Neagua, B. M. Garcia, C. S. Sandoval, A. M. Rodriguez, C. O. Ferrusola, L. G. Fernandez, J. Tapiacan, F. Peña, Freezing dog semen in presence of the antioxidant butylated hydroxytoluene improves postthaw sperm membrane integrity, Theriogenology 73 (5) (2010) 645–650.

[9] M. E. Beletti, L. d. F. Costa, M. P. Viana, A comparison of morphometric characteristics of sperm from fertile bos taurus and bos indicus bulls in brazil, Animal Reproduction Science 85 (1-2) (2005) 105–116.

[10] M. Beletti, L. Costa, M. Viana, A spectral framework for sperm shape characterization, Computers in Biology and Medicine 35 (6) (2005) 463–473.

[11] C. Linneberg, P. Salamon, C. Svarer, L. Hansen, Towards semen quality assessment using neural networks, in: Proc. IEEE Neural Networks for Signal Processing IV, 1994, pp. 509–517.

[12] L. Sanchez, N. Petkov, Acrosome integrity classification of boar spermatozoon images using DWT and texture descriptors, in: Lecture Notes in Computer Science, 2009.

[13] E. Alegre, O. Garcia-Olalla, V. Gonzalez-Castro, S. Joshi, Boar spermatozoa classification using longitudinal and transversal profiles (LTP) descriptor in digital images, in: 14th International Workshop on Combinatorial Image Analysis (IWCIA), Madrid, España, 2011.

[14] E. Alegre, M. T. Garcia-Ordas, V. Gonzalez-Castro, S. Karthikeyan, Vitality assessment of boar sperm using NCSR texture descriptor in digital images, in: Iberian Conference on Pattern Recognition and Image Analysis (IbPRIA), La Palma de Gran Canaria, España, 2011.

[15] V. Gonzalez-Castro, E. Alegre, P. Morala-Arguello, S. A. Suarez, A combined and intelligent new segmentation method for boar semen based on thresholding and watershed transform, International Journal of Imaging 2 (70–80) (2009) S09.

[16] A. Bijar, A. Peñalver-Benavent, M. Mikaeili, R. Khayati, Fully automatic identification and discrimination of sperm parts in microscopic images of stained human semen smear, Journal of Biomedical Science and Engineering (2012) 384–395doi:10.4236/jbise.2012.57049.

[17] N. Petkov, E. Alegre, M. Biehl, L. Sanchez, LVQ acrosome integrity assessment of boar sperm cells, in: Proceedings of Computational Modelling of Objects Represented in Images: Fundamentals, Methods and Applications (CompIMAGE), 2006.

[18] E. Alegre, M. Biehl, N. Petkov, L. Sanchez, Automatic classification of the acrosome status of boar spermatozoa using digital image processing and LVQ, Computers in Biology and Medicine 38 (4) (2008) 461–468.

[19] E. Alegre, V. Gonzalez-Castro, R. Alaiz-Rodriguez, M. T. Garcia-Ordas, Texture and moments-based classification of the acrosome integrity of boar spermatozoa images, Computer Methods and Programs in Biomedicine 108 (2) (2012) 873–81.

[20] V. Gonzalez-Castro, E. Alegre, O. Garcia-Olalla, D. Garcia-Ordas, M. T. Garcia-Ordas, L. Fernandez-Robles, Curvelet-based texture description to classify intact and damaged boar spermatozoa., in: International Conference on Image Analysis and Recognition, Aveiro, 2012.

[21] A. Fazeli, W. . Hage, F.-P. Cheng, W. F.Voorhout, A. Marks, M. M. Bevers, B. Colenbrander, Acrosome-lntact boar spermatozoa initiate binding to the homologous zona pellucida invitro, Biology of reproduction 56 (1997) 430–438.

[22] T. Kohonen, Self-Organizing Maps, 2nd Edition, Springer, Berlin, Heidelberg, 1997.

[23] Biliography on the self-organizing map (SOM) and Learning Vector Quantization (LVQ), Neural Networks Research Centre, Helsinki University of Technology (2002).

[24] T. Bojer, B. Hammer, D. Schunk, K. Tluk von Toschanowitz, Relevance determination in Learning Vector Quantization, in: M. Verleysen (Ed.), European Symposium on Artificial Neural Networks 2001, d-side publications, 2001, pp. 271–276.