Pixel-Based Object Detection and Tracking with Ensemble of Support Vector Machines and Extended Structural Tensor

Bogusław Cyganek[1] and Michał Woźniak[2]

[1] AGH University of Science and Technology, Al. Mickiewicza 30, 30-059 Kraków, Poland
cyganek@agh.edu.pl
[2] Wroclaw University of Technology, Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Poland
Michal.Wozniak@pwr.wroc.pl

Abstract. In this paper we propose a system for visual object detection and tracking based on the extended structural tensor and the ensemble of one-class support vector machines. First, the input color image is transformed with the anisotropic process into the extended structural tensor. Then the tensor space is clustered into the number of partitions which are used to train a corresponding number of one-class support vector machines composing an ensemble of classifiers. In run-time the ensemble classifies the input video stream into an object and background. Thanks to high discriminative properties of the extended structural tensor and to the diversity of the ensemble of classifiers the method shows very good properties which were shown by experiments on real video sequences.

1 Introduction

Object detection and tracking belong to the fundamental tasks of computer vision. However, these depend greatly on chosen definition of an object and the method of signal analysis. An object can be defined providing its template, sparse representation or statistical model [9]. Nevertheless, detection of an object in real images is still difficult due to geometric transformations, lighting conditions, occlusions and noise. In many cases, a suitable definition of an object or a group of objects is available only on pixel bases. In such cases chosen group of pixels constitute a model of an object for detection or tracking.

In this paper we propose a system of pixel based object detection and tracking. The two novelties come from connection of the extended structural tensor (EST) for feature detection, proposed in paper [14], with the ensemble of one-class support vector machines (OC-SVM) for classification, presented in our previous work [7]. Additionally, different tensors comparison measures are investigated. Experimental results show high discriminative power of EST and ensemble of OC-SVMs.

The paper is organized as follows. Section 2 describes architecture and overall operation of the proposed system. In section 3 we present details of the EST. The methods of their comparisons are discussed in section 4. The paper ends with experimental results in section 5 and conclusions in section 6.

N.-T. Nguyen et al. (Eds.): ICCCI 2012, Part I, LNAI 7653, pp. 104–113, 2012.

2 Object Tracking with an Ensemble of OC-SVM

Fig. 1 depicts architecture of the proposed system for object detection and tracking in video streams. Basically it is a combination of an improved tracking system proposed in [6] and the framework for pattern classification by an ensemble of OC-SVMs, proposed in [7]. The system has two basic paths of processing. The first one constitutes a training module which goal is to prepare the ensemble of classifiers used during system run-time. For each classifier the OC-SVM was chosen [22][23][24].

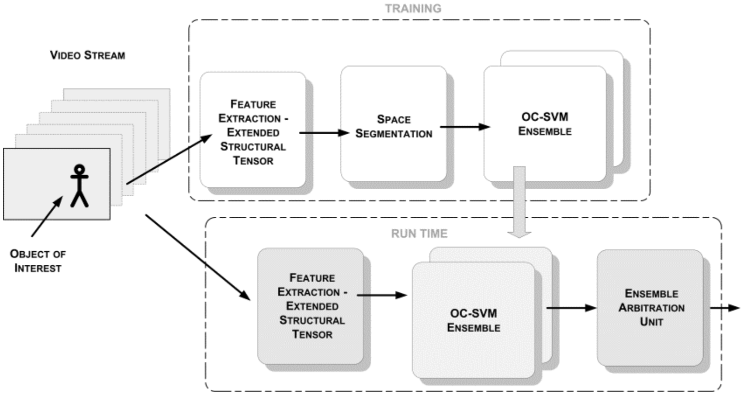

TRAINING
VIDEO STREAM
FEATURE EXTRACTION - EXTENDED STRUCTURAL TENSOR
SPACE SEGMENTATION
OC-SVM ENSEMBLE
OBJECT OF INTEREST
RUN TIME
FEATURE EXTRACTION - EXTENDED STRUCTURAL TENSOR
OC-SVM ENSEMBLE
ENSEMBLE ARBITRATION UNIT

Fig. 1. Architecture of the front-end of the tracking system with the ensemble of OC-SVMs

Training is instantiated by selection of the features characteristic to the objects of interest [6]. This makes relatively large spectrum of applications of the proposed method. Features which define the class of objects of interest usually come from an extracted template object from a reference frame. In the proposed system the features are computed with the EST, as discussed in the next section.

In the run-time the input consists of the video stream from which the same type of features needs to be computed before classification. Then each classifier in ensemble provides its output. All such responses are resolved in the arbitration unit [12].

During classification by OC-SVM, a test pattern $\mathbf{x}_x$ is assigned to a class represented by that classifier if the following inequality is fulfilled [7]

$$\sum_{i\in Idx(SV)} \alpha_i K\left(\mathbf{x}_x,\mathbf{x}_i\right) \geq \sum_{i\in Idx(SV)} \alpha_i K\left(\mathbf{x}_s,\mathbf{x}_i\right) = \tau, \tag{1}$$

where *Idx(SV)* represents a set of support vectors $\mathbf{x}_s$ found for this problem, while α_i are their associated scalar parameters. In computation of (1) the Gaussian kernel with the tensor distance (discussed in the next section) is used. It is given as follows

$$K_{RBF}\left(\mathbf{A},\mathbf{B}\right) = e^{-\gamma d^2_{(\cdot)}(\mathbf{A},\mathbf{B})}, \tag{2}$$

where $d^2_{(\cdot)}(\mathbf{A},\mathbf{B})$ denotes a distance between tensors **A** and **B**, while the parameter γ controls kernel spread. Details on choosing **A**, **B**, and a distance *d* are discussed in the section (4) of this paper. On the other hand, details on construction process and operation of the ensemble of OC-SVM are presented in [7].

3 Feature Collection with the Extended Structural Tensor

Determining right number and type of features in a signal for pattern recognition is the first step influencing performance of the recognition process. Too few, or not characteristic features, can lead to poor object discrimination. On the other hand, too many or ad hoc chosen features usually result in poor generalization properties and excessive processing time of a classifier. In images, the most obvious features are color and textures, although there exists dozens of different approaches, such as image jets, wavelets [15], statistical moments [10][11], or non-parametric measures [26], to name a few. However, even for the color and texture there are many ways of their computation and processing. For instance, there are many color spaces which show different characteristics for various tasks [13]. On the other hand, texture can be computed e.g. with wavelets . Last but not least is a question on the computational complexity of feature detection, as well as their fitness to the classifier. In this respect, good results can be obtained incorporating tensor algebra into feature detection.

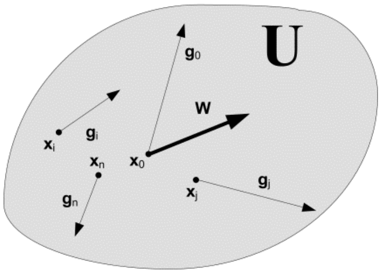


Fig. 2. Computation of the dominating orientation vector **w** in a pixel neighborhood **U** based on signal gradients $\mathbf{g}_i$

The 2D structural tensor is a symmetric matrix which eigenvector **w**, corresponding to the largest eigenvalue, points in a direction which is the most concordant with directions of all gradient vectors in a certain region ***U***. This is shown in Fig. 2. The 2D structural tensor is defined as follows [11]

$$\mathbf{T}=\begin{bmatrix} \int\limits_U \frac{\partial I(x,y)}{\partial x}\frac{\partial I(x,y)}{\partial x}d\mathbf{u} & \int\limits_U \frac{\partial I(x,y)}{\partial x}\frac{\partial I(x,y)}{\partial y}d\mathbf{u} \\ \int\limits_U \frac{\partial I(x,y)}{\partial y}\frac{\partial I(x,y)}{\partial x}d\mathbf{u} & \int\limits_U \frac{\partial I(x,y)}{\partial y}\frac{\partial I(x,y)}{\partial y}d\mathbf{u} \end{bmatrix} = F\left(\begin{bmatrix} I_x \\ I_y \end{bmatrix}\begin{bmatrix} I_x & I_y \end{bmatrix}\right) = F\left(\mathbf{G}\mathbf{G}^T\right), \tag{3}$$

where I stands for image intensity signal, $\mathbf{G}=[I_x\ I_y]^T$ is a gradient vector in which $I_x=\partial I/\partial x$, $I_y=\partial I/\partial y$, F denotes a smoothing operator, i.e. integration in the continuous and summation in discrete cases, respectively.

Structural tensor provides valuable information on structure of local regions in the input image. However, in many applications it is desirable to use intensity or color values and texture information together. For instance, such compound information was proposed in the previous version of the system [6]. A slightly different idea was proposed by Luis-García *et al.* for image segmentation [14]. In this version mixed products of gradients and intensity signal are created. This way the nonlinear extended structural tensor for monochrome image is obtained, as follows

$$\mathbf{T}_M = F\left(\begin{bmatrix} I_x \\ I_y \\ I \end{bmatrix}\begin{bmatrix} I_x & I_y & I \end{bmatrix}\right) = F\left(\begin{bmatrix} I_x^2 & I_x I_y & I_x I \\ I_y I_x & I_y^2 & I_y I \\ I I_x & I I_y & I^2 \end{bmatrix}\right) = F\left(\mathbf{G}_M \mathbf{G}_M^T\right), \tag{4}$$

where again F denotes the averaging operator, I_x and I_y denote directional derivatives of the intensity signal I in the x and y directions, respectively. Tensor $\mathbf{T}_M$ in (4) is symmetric. Hence, $\mathbf{T}_M$ provides $n(n+1)/2=6$ independent features.

In the case of color images, composed of three channels $[I_R\ I_G\ I_B]^T$, $\mathbf{G}_E$ is further extended to $\mathbf{G}_C$, which is now of size $n=5$, as follows

$$\mathbf{G}_C = \begin{bmatrix} I'_x & I'_y & I_R & I_G & I_B \end{bmatrix}^T . \tag{5}$$

In the above

$$I' = \tfrac{1}{3}\left(I_R + I_G + I_B\right) \tag{6}$$

is simply an averaged intensity signal (i.e. a monochrome intensity signal). This simply leads to the extended structural tensor for color images

$$\mathbf{T}_C = F\left(\mathbf{G}_C \mathbf{G}_C^T\right), \tag{7}$$

which due to the symmetry provides 15 independent components.

For discrete signals, the operator F can be implemented as one of the smoothing operators, such as Gaussian or binomial filters [11]. However, smoothing with these filters results in dislocation of edges. To remedy this problem Brox *et al.* proposed to employ the nonlinear diffusion into the smoothing process [2]. This is accomplished with the anisotropic diffusion.

Anisotropic diffusion was proposed by Perona *et al.* [18] as a modification to image smoothing with the heat equation which preserves sharp object boundaries. The main idea is to use a control function g in the computation of the Laplacian that stops smoothing if an edge is encountered. This procedure can be written in the PDE form

$$\partial_t f(x,y,t) = div\left(g\left(\left|\nabla f(x,y,t)\right|\right)\cdot \nabla f(x,y,t)\right), \tag{8}$$

where f denotes a signal to be smoothed. In our case we also use the above equation for discrete monochrome signal I' from (6). For the control function g Perona *et al.* propose to use $g(x)=(1+x^2/k^2)^{-1}$, where k is a positive constant [18]. However, as shown by Shapiro, the Tukey biweight function, given as follows

$$g\left(x\right)=\begin{cases}\frac{1}{2}\left(1-x^2/k_r^2\right)^2, & \left|x\right|\le k_r\\ 0, & \textit{otherwise}\end{cases}. \tag{9}$$

shows its superiority in leaving untouched strong signal variations [21]. For the parameter *k* in (9) the so called robust scale

$$k_r=1.4826\cdot med\left(\left\|\nabla I-med\left(\left\|\nabla I\right\|\right)\right\|\right) \tag{10}$$

can be used. It is computed from the gradient ∇*I* of the monochrome version of the original image, while *med* denotes a median function [21].

To compute I_x and I_y in (3), (4) and (8) the derivative filters for discrete signals are used. However, instead of the popular finite difference method [19], the discrete signal to be differentiated is first approximated with the continuous polynomial for which then derivatives are computed which finally are back sampled to the discrete domain. In effect computation of PDE (8) is stable and efficient since the filters are also separable. Further details on this method with references can be found in [5].

4 Distance Measures for Tensor Data

Pattern recognition in tensor space requires definition of a tensor distance. There are many ways to accomplish this task, however. The simplest comparison can be done with a version of the Frobenius norm, given as follows [16]:

$$d_{(F)}\left(\mathbf{A},\mathbf{B}\right)=\sqrt{Tr\left[\left(\mathbf{A}-\mathbf{B}\right)^2\right]}, \tag{11}$$

where **A** and **B** are two tensors and *Tr* denotes the trace. A statistical approach to tensor comparisons follows the Kullback-Leibler distance among probability distributions. Its symmetric version, called *J*-divergence, was proposed for tensors by Wang *et al.* as follows [25]:

$$d_{(J)}\left(\mathbf{A},\mathbf{B}\right)=\frac{1}{2}\sqrt{Tr\left(\mathbf{A}^{-1}\mathbf{B}-\mathbf{B}^{-1}\mathbf{A}\right)-2n}, \tag{12}$$

where *n* stands for tensor dimensionality (*n*=3 for $\mathbf{T}_M$ and *n*=5 for $\mathbf{T}_C$, respectively).

An interesting measure for comparison of the diffusion tensors arising in MRI was proposed by Pennec *et al.* [17]. It is given as follows:

$$d_{(R)}\left(\mathbf{A},\mathbf{B}\right)=\sqrt{Tr\left[\log^2\left(\mathbf{A}^{-\frac{1}{2}}\mathbf{B}\mathbf{A}^{-\frac{1}{2}}\right)\right]}. \tag{13}$$

in which for symmetric tensors **T** it holds that $\mathbf{T}^2=\mathbf{TT}=\mathbf{TT}^T$. However, computation of (13) can be time consuming and for this purpose Arsigny *et al.* proposed a simplified Log-Euclidean distance, as follows:

$$d_{(LF)}\left(\mathbf{A},\mathbf{B}\right)=d_{(F)}\left(\log\left(\mathbf{A}\right),\log\left(\mathbf{B}\right)\right)=\sqrt{Tr\left[\left(\log\left(\mathbf{A}\right)-\log\left(\mathbf{B}\right)\right)^2\right]}, \tag{14}$$

For the sake of computational complexity the most appropriate measures are $d_{(F)}$ and $d_{(LF)}$. They were also verified experimentally by Rittner *et al.* as the ones which show the best performance for comparisons of color images [20]. It is also interesting to notice, that the extended tensors, $\mathbf{T}_M$ in (4) and $\mathbf{T}_C$ in (7), due to multiplications of internal components show high variability of their values. Therefore, to assure well balanced comparisons, they can be normalized. However, as mentioned in the paper by Luis-García *et al.* [14], a simple normalization is not recommended since it leads to noise amplification in channels containing no information. Instead, they propose to replace the values by their square roots. In this paper we tested the $d_{(F)}$ and $d_{(LF)}$ measure, given in (11) and (14), respectively. The latter follows geodesics in the manifold of symmetric positive defined matrices [1]. Moreover, it can be seen as a simple Frobenius norm on log preprocessed input signals. This property also simplifies implementation, since all classifiers operating with the Frobenius norm do not need to be changed - only the input signals need to be preprocessed with the logarithm function.

Computation of the natural logarithm function in (14) is possible only for positive definite matrices. This simplifies further for symmetric matrices which have only real eigenvalues. In this case a symmetric matrix **A** can be first eigen-decomposed

$$\mathbf{A}=\mathbf{R}\boldsymbol{\Lambda}\mathbf{R}^T, \tag{15}$$

where **R** is an orthogonal matrix with eigenvectors, Λ is a diagonal matrix with eigenvalues. The above allows simple computation of the logarithm, as follows [15]

$$\log\left(\mathbf{A}\right)=\mathbf{R}\log\left(\boldsymbol{\Lambda}\right)\mathbf{R}^T. \tag{16}$$

However, to apply the measures (13) and (14) we need to be sure that the compared tensors are positive definite. From (3), (4) and (7) we easily notice that these tensors can be represented as an outer product of a vector. Thus, such a tensor can be represented as $\mathbf{A}=\mathbf{aa}^T$. Now, checking the positive definite condition the following is obtained

$$\mathbf{x}^T\mathbf{Ax}=\mathbf{x}^T\left(\mathbf{aa}^T\right)\mathbf{x}=\left(\mathbf{x}^T\mathbf{a}\right)\left(\mathbf{x}^T\mathbf{a}\right)^T\geq 0. \tag{17}$$

The last equation indicates the tensors of this type are positive semidefinite. Thus, to compute their logarithms we need to check a special case of eigenvalues equal to 0. In our implementation we set a threshold of 1e-13 below which a log value -30.0 is put.

5 Experimental Results

The presented method was implemented in C++ using the HIL library [5], as well as the LIBSVM [3]. Experiments were run on the computer with 8 GB RAM and Pentium® Quad Core Q 820 (clock 1.73 GHz). However, the run-time module was then ported to GPU with help of the CUDA environment. This allowed speed up of up to two orders of magnitude compared to the serial implementation, depending on the size of the input images.

Fig. 3 depicts fifteen components of the EST for the color image, also shown in Fig. 4a. All are in a form of products of two signals, as defined in (5)-(7).

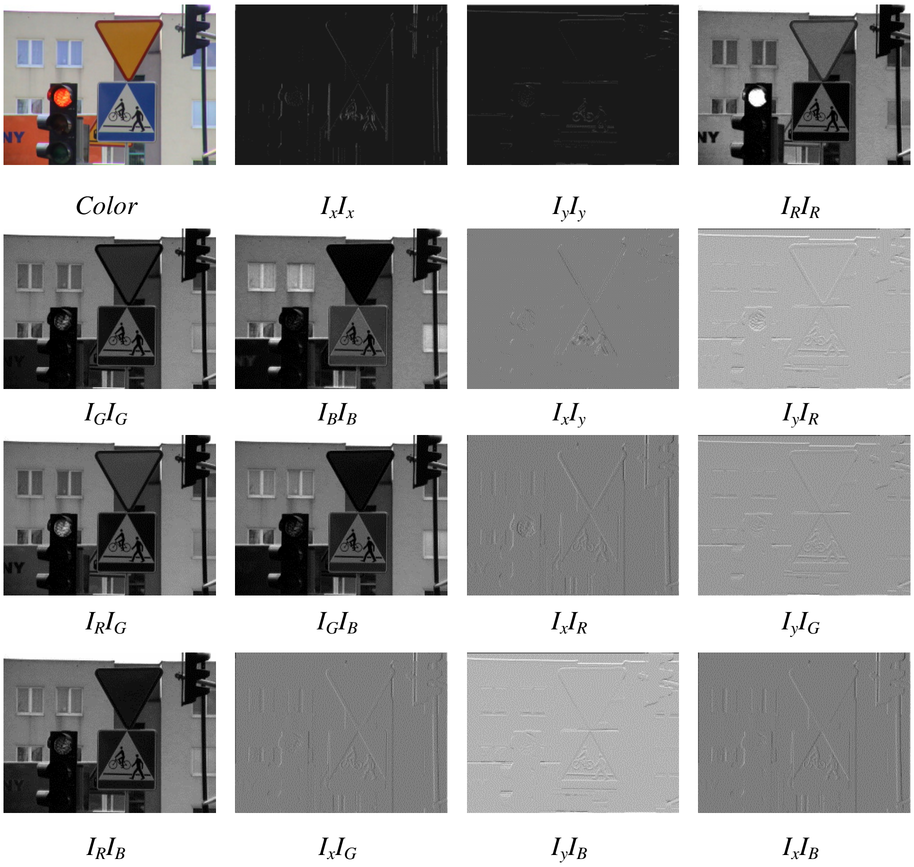
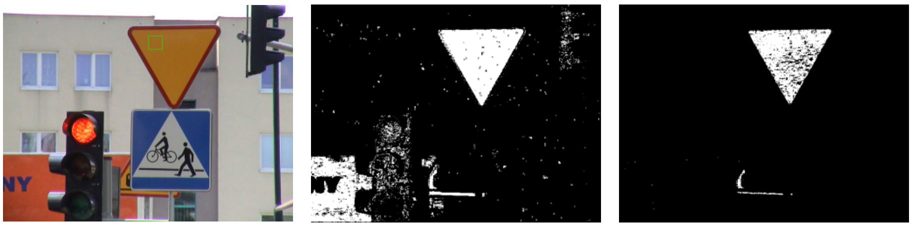


Fig. 3. Components of the Extended Structural Tensor (EST) of the upper-left color image



a b c

Fig. 4. Road sign detection based on its internal pixels (a). Detection with OC-SVM ensemble with 2 classifiers (b). Detection with 5 classifiers in the ensemble (c). The $d_{(LF)}$ used.

Fig. 4a depicts results of real object detection based on tensor definition of the pixels in the green rectangle (a road sign). Detection results with the ensemble of 2 and 5 OC-SVMs are shown in Fig. 4b and Fig. 4c, respectively. In the former, an excessive number of false positives is present. This is improved with larger ensemble.

Table 1 contains parameters of the OC-SVM classifiers in the ensemble of used to obtain results presented in Fig. 4c. Further details are in [6].

Table 1. Parameters of the ensemble of OC-SVM used to obtain Fig. 4c

	#SVs	γ	ν	ρ
1	11	0.0441257	0.206667	3.2132
2	7	0.0535715	0.161429	2.91624
3	9	0.0383056	0.0017331	0.739651
4	9	0.0081370	0.0640351	3.24109
5	5	0.0129474	0.0049751	0.870604

Fig. 5 shows results of tracking of a person head in a driver's monitoring system. Detected pixels with an ensemble of three OC-SVMs are shown in the top row of Fig. 5. Head orientation from the mean shift method is shown in the bottom row [4]. Parameters of the classifiers are presented in Table 2, whereas details of the mean shift in this context are presented in [6][4].

Results presented in Fig. 5 were obtained with the $d_{(LF)}$ measure. However, comparison of $d_{(LF)}$ with $d_{(F)}$ shows no winner in our experiments. Considering much larger computational demands of the former, this makes $d_{(F)}$ more favorable. Nevertheless, this comparison needs further research and in the context of other vision tasks.

The drawback of the presented method is processing time necessary for iterative computation of the $\mathbf{T}_C$ in (7), as well as the distance $d_{(LF)}$ in (14) if used. However, structure of the proposed ensemble of classifiers allows their independent and parallel operation during classification (see Fig. 1). Recently this module was ported to the GPU using the CUDA framework [8]. As a result, in a classification stage a speed-up ratio of two orders of magnitude was achieved.

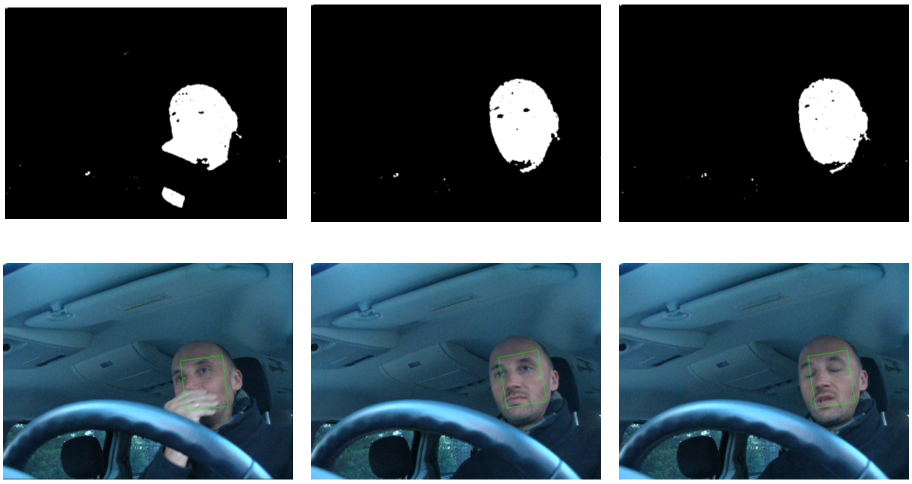


Fig. 5. Results of person head tracking in a driver's monitoring system. Detected pixels with an ensemble of three OC-SVMs (top row). Head orientation from the mean shift method (bottom).

Table 2. Parameters of the ensemble of OC-SVM used to obtain Fig. 5

	#SVs	γ	ν	ρ
1	4	0.0279517	0.00446429	0.80108
2	5	0.0219527	0.00406504	0.870835
3	4	0.0305445	0.00232558	0.794549

6 Conclusions

In the paper a system for object detection and tracking based on pixel analysis is proposed which extends the method presented in [6]. The novelty of the paper is connection of the extended structural tensor, used for feature detection, and an ensemble of one-class support vector machines used for pixel classification. The ensemble is build upon prior data clusterization with the k-means algorithm. Then, the best parameters of each OC-SVM are computed in an n-fold fashion. Application of the ensemble of classifiers and the extended structural tensor allow highly discriminative properties of classification. Also, to compare tensors the Frobenius and the Log-Frobenius norms were tested. However, the former offers similar results in the tested application with much smaller computational demands. The presented experimental results show high accuracy of the presented method for systems with pixel-based object definitions. Finally, a highly parallel structure of the ensemble of classifiers makes it appropriate for concurrent implementations.

Acknowledgements. The work was supported from the funds of the Polish National Science Centre NCN in the year 2012, contract no. DEC-2011/01/B/ST6/01994.

References

1. Arsigny, V., Fillard, P., Pennec, X., Ayache, N.: Log-Euclidean Metrics for Fast and Simple Calculus on Diffusion Tensors. Magnetic Resonance in Medicine 56(2), 411–421 (2006)
2. Brox, T., Rousson, M., Derich, R., Weickert, J.: Unsupervised Segmentation Incorporating Colour, Texture, and Motion. INRIA Technical Report No 4760 (2003)
3. Chang, C.-C., Lin, C.-J.: LIBSVM, a library for support vector machines (2001), `http://www.csie.ntu.edu.tw/~cjlin/libsvm`
4. Comaniciu, D., Meer, P.: Mean Shift: A Robust Approach Toward Feature Space Analysis. IEEE Transactions on Pattern Analysis And Machine Intelligence 24(5), 603–619 (2002)
5. Cyganek, B., Siebert, J.P.: An Introduction to 3D Computer Vision Techniques and Algorithms. Wiley (2009)
6. Cyganek, B.: Framework for Object Tracking with Support Vector Machines, Structural Tensor and the Mean Shift Method. In: Leung, C.S., Lee, M., Chan, J.H. (eds.) ICONIP 2009, Part I. LNCS, vol. 5863, pp. 399–408. Springer, Heidelberg (2009)
7. Cyganek, B.: One-Class Support Vector Ensembles for Image Segmentation and Classification. J. of Math. Imaging & Vision 42(2-3), 103–117 (2012)
8. `http://developer.nvidia.com/`
9. Duda, Hart, Stork: Pattern Classification. Wiley (2001)
10. Forsyth, D.A., Ponce, J.: Computer Vision. A Modern Approach. Prentice-Hall (2003)
11. Jähne, B.: Digital Image Processing. Springer (2005)
12. Kuncheva, L.: Combining Pattern Classifiers. Methods and Algorithms. Wiley (2004)
13. Lee, H.-C.: Introduction to Color Imaging Science. Cambridge University Press (2005)
14. de Luis-García, R., Deriche, R., Rousson, M., Alberola-López, C.: Tensor Processing for Texture and Colour Segmentation. In: Kalviainen, H., Parkkinen, J., Kaarna, A. (eds.) SCIA 2005. LNCS, vol. 3540, pp. 1117–1127. Springer, Heidelberg (2005)
15. Moon, T.K., Stirling, W.C.: Mathematical Methods and Algorithms for Signal Processing. Prentice-Hall (2000)
16. Peeters, T., Rodrigues, P., Vilanova, A., ter Haar Romeny, B.: Analysis of distance/similarity measures for diffusion tensor imaging. In: Visualization and Processing of Tensor Fields: Advances and Perspectives, pp. 113–136. Springer, Berlin (2008)
17. Pennec, X., Fillard, P., Ayache, N.: A Riemannian framework for tensor computing. International Journal of Computer Vision 66(1), 41–66 (2006)
18. Perona, P., Malik, J.: Scale-Space and Edge Detection Using Anisotropic Diffusion. IEEE Trans. on Pattern Analysis and Machine Intelligence 12(7), 629–639 (1990)
19. Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P.: Numerical Recipes in C. The Art of Scientific Computing, 2nd edn. Cambridge University Press (1999)
20. Rittner, L., Flores, F.C., Lotufo, R.A.: A tensorial framework for color images. Pattern Recognition Letters 31(4), 277–296 (2010)
21. Sapiro, G.: Geometric Partial Differential Equations and Image Analysis. Cambridge (2001)
22. Schölkopf, B., Smola, A.J.: Learning with Kernels. MIT Press (2002)
23. Shawe-Taylor, J., Cristianini, N.: Kernel Methods for Pattern Analysis. Cambridge University Press (2004)
24. Tax, D.M.J., Duin, R.P.W.: Support Vector Data Description. Machine Learning 54, 45–66 (2004)
25. Wang, Z., Vemuri, B.C.: DTI segmentation using an information theoretic tensor dissimilarity measure. IEEE Transactions on Medical Imaging 24(10), 1267–1277 (2005)
26. Zabih, R., Woodfill, J.: Non-parametric Local Transforms for Computing Visual Correspondence. Computer Science Department, Cornell University, Ithaca (1998)