



HAIS 2010, San Sebastián, Spain, June 22-25, 2010

Image Segmentation with a Hybrid Ensemble of One-Class Support Vector Machines

Bogusław Cyganek

AGH University of Science and Technology

Al. Mickiewicza 30, 30-059 Kraków, Poland

cyganek@agh.edu.pl









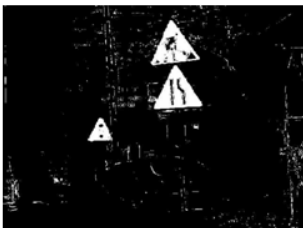
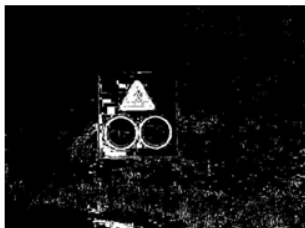

Outline:

- Image segmentation framework
- Problem statement.
- System architecture.
- Data space segmentation.
- One-class Support Vector Machines.
- Experimental results (implementation, other applications...).
- Conclusions.

What we tried to achieve?

Image segmentation belongs to the key problems in computer vision and image processing. Basically this process can be characterized as automatic partitioning of an image into the areas of interest (objects) and background. Thus segmentation relies on specific features of an image, such as color, texture, etc.

Therefore image segmentation finds broad applications, for instance...

Original road scenes			
Red objects			
Yellow objects			

What we tried to achieve?

Peer P., Kovac J., Solina F.: Human skin colour clustering for face detection. EUROCON 2003 – International Conference on Computer as a Tool, 2003

Fuzzy rules for sun lighting

	<p>“Range of skin color components in daily conditions found in experiments” R₁: IF $R > 95$ AND $G > 40$ AND $B > 20$ THEN $T_0 = \text{high}$;</p>
	<p>“Sufficient separation of the RGB components; Elimination of grey areas” R₂: IF $\max(R, G, B) - \min(R, G, B) > 15$ THEN $T_1 = \text{high}$;</p>
	<p>“R, G should not be close together” R₂: IF $R - G > 15$ THEN $T_2 = \text{high}$;</p>
	<p>“R must be the greatest component” R₃: IF $R > G$ AND $R > B$ THEN $T_3 = \text{high}$;</p>

Fuzzy rules for flash lighting

	<p>“Skin color values for flash illumination” R₄: IF $R > 220$ AND $G > 210$ AND $B > 170$ THEN $T_4 = \text{high}$;</p>
	<p>“R and G components should be close enough” R₅: IF $R - G \leq 15$ THEN $T_5 = \text{high}$;</p>
	<p>“B component has to be the smallest one” R₆: IF $B < R$ AND $B < G$ THEN $T_6 = \text{high}$;</p>

The combined (aggregated) fuzzy rule for human skin detection

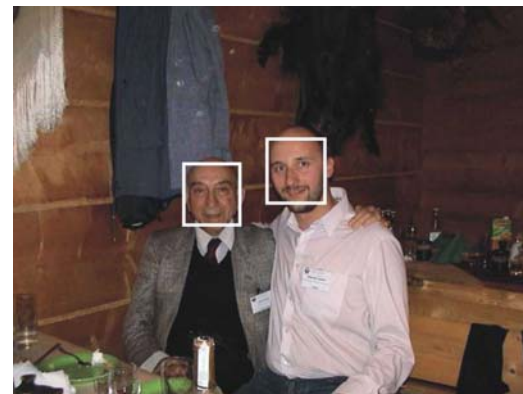
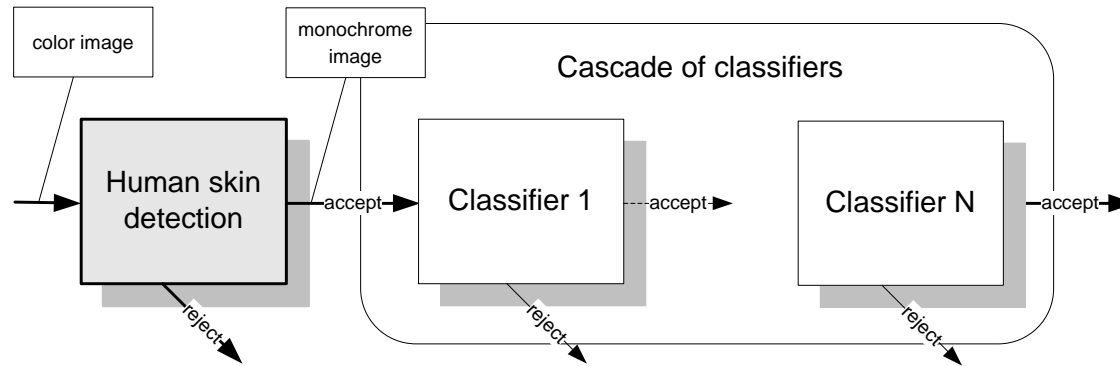
	<p>“Fuzzy rule for human skin color detection in RGB” R_{HS}: IF T_{0-3} are high OR T_{4-6} are high THEN $H = \text{high}$;</p>
--	--

Jayaram S., Schmutz S., Shin M.C., Tsap L.V.: Effect of Colorspace Transformation, the Illuminance Component, and Color Modeling on Skin Detection. IEEE Computer Society Conference on Computer Vision and Pattern Recognition CVPR’04, 2004

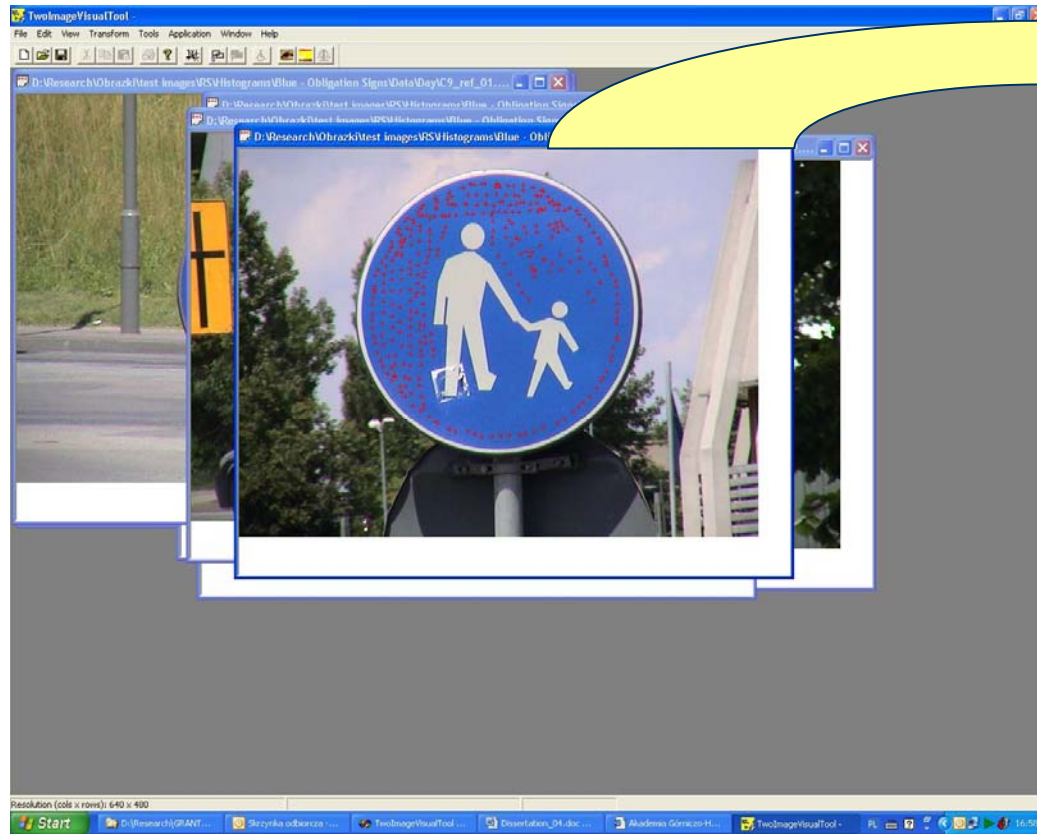
Phung S.L., Bouzerdoum A., Chai D.: Skin Segmentation Using Color Pixel Classification: Analysis and Comparison. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 27, No. 1, pp. 148-154, 2005

What we tried to achieve?

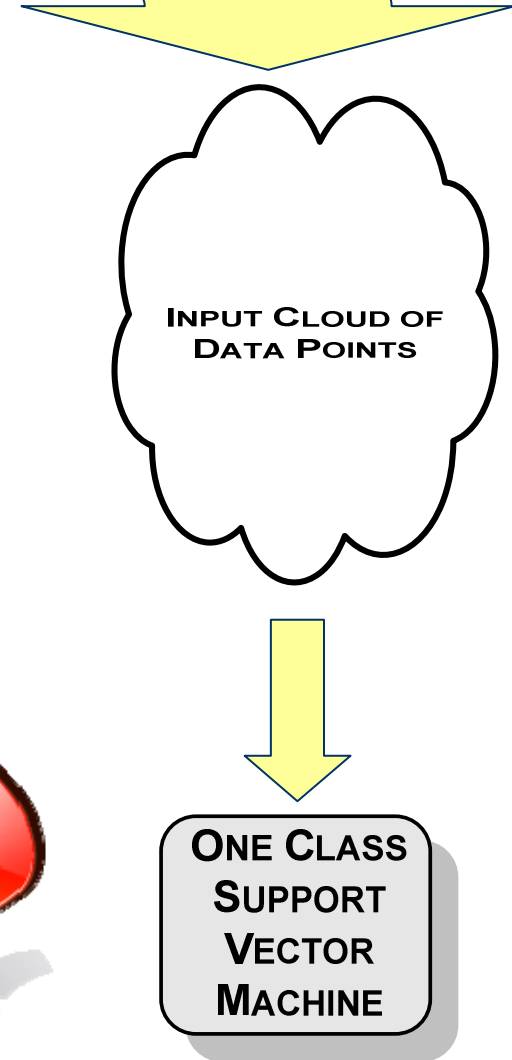
A cascade system for human face detection. The first classifier does dimensionality reduction selecting only pixels-of-interest based on a model of a color for human skin



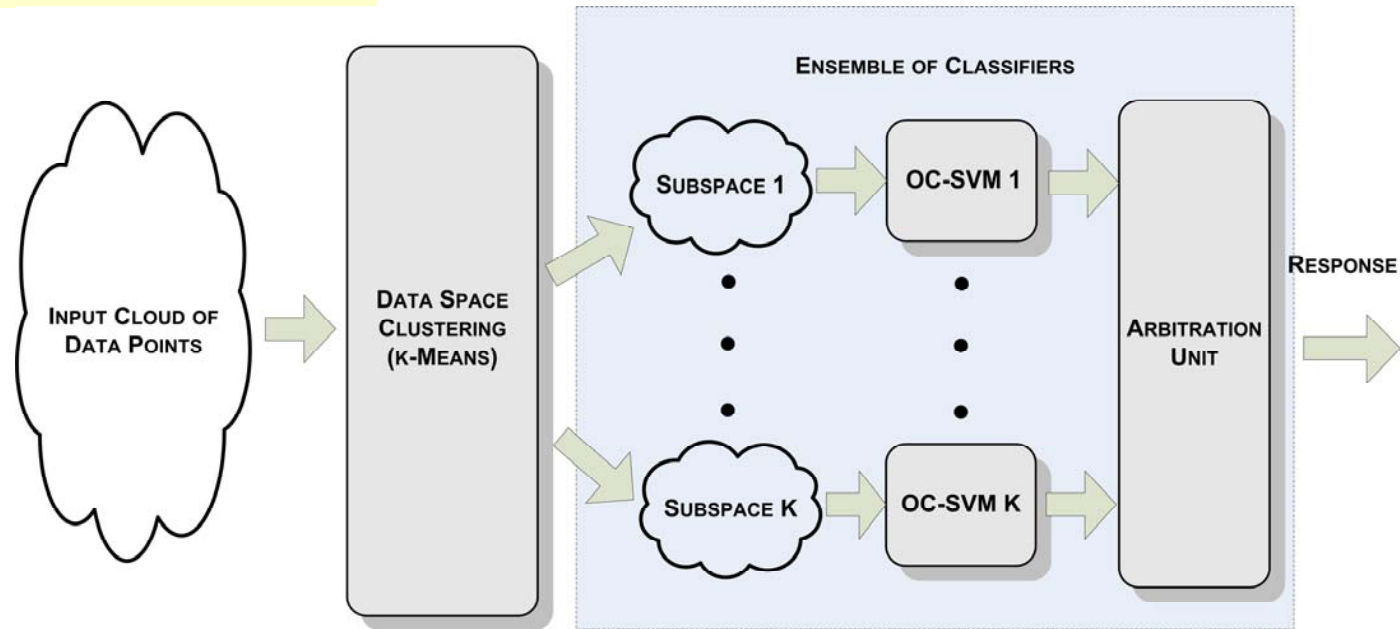
What we tried to achieve?



The color acquisition tool



Architecture of the System:



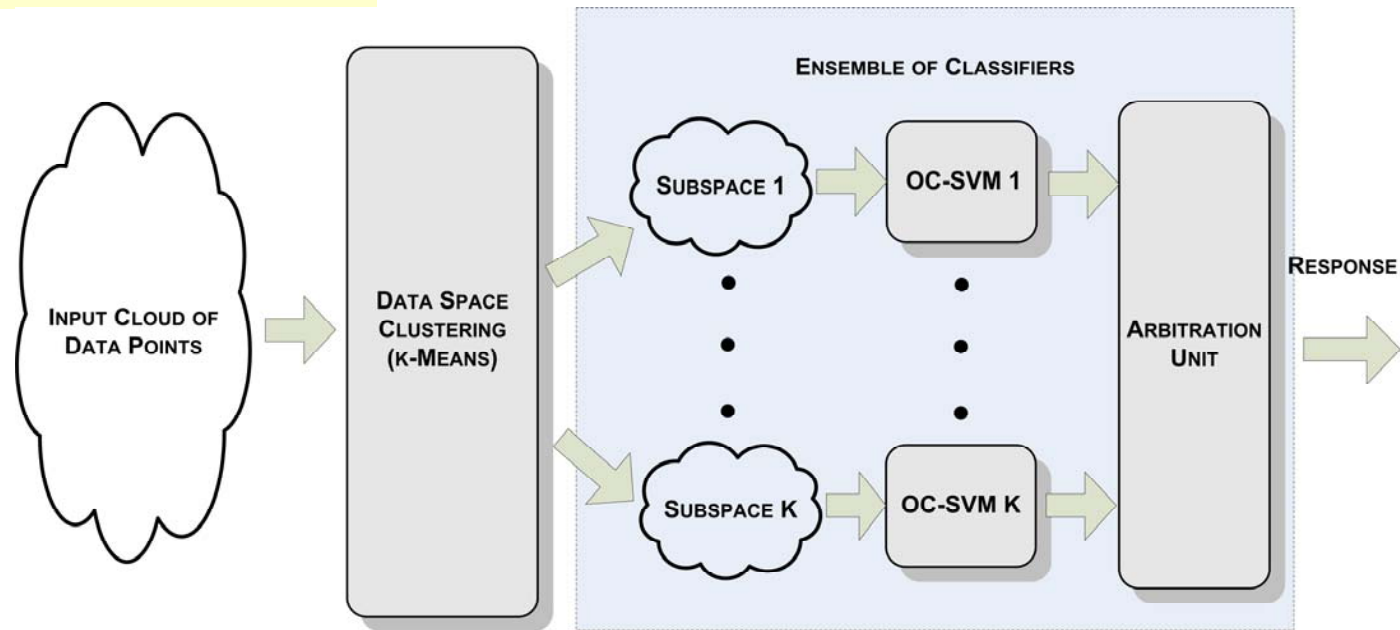
Quantitatively the data fit parameter can be measured

$$\rho_i = \frac{\#SV_i}{\#D_i}$$

$\#SV_i$ is a number of support vectors for a given data set D_i . In practice good classification results were obtained for $\rho_i < 0.1$

ρ_i depends on two parameters of the OC-SVM - the width σ of the Gaussian kernel and the training parameter ν , (\rightarrow grid search). However, for some data sets a satisfactory ρ_i cannot be obtained due to data complexity even in the feature space.

Architecture of the System:



The idea is to *split the initial data space* into separate and as compact as possible clusters and then train individual OC-SVM classifiers with each of the clusters.

$$\rho = \sum_{i=1}^M \rho_i = \sum_{i=1}^M \frac{\#SV_i}{\#D_i}$$

M is a number of members of the ensemble and each D_i is a subset of data. The input space D is partitioned into a set $\{D_i\}$ with an unsupervised clustering method. In effect there are $2M$ parameters to be found. However, the problem gets complicated since those depend on partitioning of data. Therefore optimization of the above imposes optimization of the data clustering process.

Input Space Segmentation:

Given a set of training points $\{\mathbf{x}_i\}$, the algorithm starts with choice of the initial number of clusters D_i and for each a mean value μ_i is selected.

After initialization the method proceeds iteratively by assigning each point \mathbf{x}_i to the closest mean μ_m in a cluster $D(i)$ in accordance with the following

$$D(i) = \arg \min_{1 \leq m \leq M} \|\mathbf{x}_i - \mu_m\|_L$$

Then each mean value is recomputed

$$\mu_m = \frac{1}{\# D_m} \sum_{\mathbf{x}_i \in D_m} \mathbf{x}_i$$

The above steps follow until convergence state is reached, i.e. there are no changes in means and in the point assignments to the clusters.

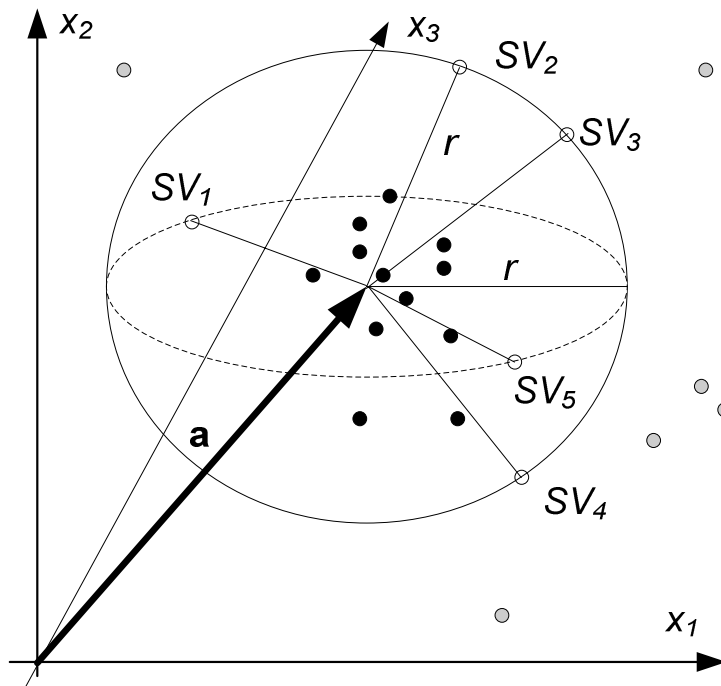
A qualitative insight can be gained by analyzing the total sums of distances

$$S_m = \sum_{\mathbf{x} \in D_m} \|\mathbf{x} - \mu_m\|^2 \qquad S_t = \sum_{m=1}^M S_m$$

The above should be as minimal as possible since k-means does not guarantee the globally optimal solution, though in practice it converges very fast.

One-class Support Vector Machines (OC-SVM):

Support vector machine (SVM) is a new type of a classifier originally proposed by Vapnik. The most characteristic is transformation of data into so called **feature space** in which the classification can be done with a linear hypersurface. The new space is of *higher* dimension than the original one. The transformation is done with a **kernel**, usually selected based on a type of data. However, direct computation of the high dimensional representation in the feature space is not necessary due to so called **kernel trick** which consists simply in computation always of a scalar product in all classification stages.



The idea of data clustering with OC-SVM consists in composing a closed boundary, an n -dimensional hypersphere, around the n -dimensional input data. This way a test point is classified as belonging into the class (an inlier) if it falls inside this hypersphere. Otherwise it is an outlier.

One-class Support Vector Machines (OC-SVM) – ctd.

This volume is proportional to r^n . However, minimization of r^n means also minimization with respect to r^2 which simplifies further derivations. Thus, the minimization functional

$$\Theta(\mathbf{a}, r) = r^2$$

with the constraint

$$\forall_i: \|\mathbf{x}_i - \mathbf{a}\| \leq r$$


where \mathbf{x}_i are data points, \mathbf{a} is a center of the hypersphere and r is its radius.

However, to introduce a possibility of some outliers in the training set we allow further distances than r but with some additional penalty. To accomplish this task the slack variables ξ_i are introduced (Vapnik):

$$\Theta(\mathbf{a}, r) = r^2 + C \sum_i \xi_i$$

with the constraints to assure that almost all objects are within the sphere

$$\forall_i: \|\mathbf{x}_i - \mathbf{a}\| \leq r + \xi_i \quad \xi_i \geq 0$$



Alternative:
Least-Squares

Given a set of training points $\{\mathbf{x}_i\}$, solution to the above equation can be obtained by means of the Lagrange multipliers (**nonnegative** α_i and β_i):

$$Q_L(r, \mathbf{a}, \alpha_i, \beta_i, \xi_i) = r^2 + C \sum_i \xi_i - \sum_i \alpha_i \left[r^2 + \xi_i - \|\mathbf{x}_i - \mathbf{a}\|^2 \right] - \sum_i \beta_i \xi_i$$

One-class Support Vector Machines (OC-SVM) – ctd.

The functional Q_L has to be minimized with respect to r , \mathbf{a} , and ξ_i , and simultaneously maximized with respect to α_i and β_i . Thus, the next step is computation of the partial derivatives of Q_L with respect to the mentioned variables, then equating them to zero.

$$\frac{\partial Q_L}{\partial r} = 0 \rightarrow \frac{\partial Q_L}{\partial r} = 2r - 2r \sum_i \alpha_i = 0$$

$$\frac{\partial Q_L}{\partial \mathbf{a}} = 0 \rightarrow \frac{\partial Q_L}{\partial \mathbf{a}} = -2 \sum_i \alpha_i \mathbf{x}_i + 2\mathbf{a} \sum_i \alpha_i = 0$$

$$\frac{\partial Q_L}{\partial \xi_i} = 0 \rightarrow \frac{\partial Q_L}{\partial \xi_i} = C - \alpha_i - \beta_i = 0$$

From these we conclude that

$$\sum_i \alpha_i = 1$$

$$\mathbf{a} = \sum_i \alpha_i \mathbf{x}_i$$

$$\alpha_i = C - \beta_i \rightarrow 0 \leq \alpha_i \leq C \quad \text{since } \alpha_i \text{ and } \beta_i \text{ are nonnegative}$$

Introducing the above (red frame) into Q_L the dual form is obtained.

One-class Support Vector Machines (OC-SVM) – ctd.

In the light of the Kuhn-Tucker optimality conditions at **the optimal point** it holds that

$$Q_L(r, \mathbf{a}, \alpha_i, \beta_i, \xi_i) = r^2 + C \sum_i \xi_i - \sum_i \alpha_i \left[r^2 + \xi_i - \|\mathbf{x}_i - \mathbf{a}\|^2 \right] - \sum_i \beta_i \xi_i$$
$$\forall_i: \alpha_i \left[r^2 + \xi_i - \|\mathbf{x}_i - \mathbf{a}\|^2 \right] = 0 \quad \forall_i: \xi_i \beta_i = 0$$

- A training point \mathbf{x}_i for which $\alpha_i > 0$ and $\xi_i > 0$ lies outside the hypersphere. Also for such points it holds that $\beta_i = 0$, so $\alpha_i = C$. These are *bounded support vectors* (BSV).
- If for a point \mathbf{x}_i the corresponding $\xi_i = 0$, then such a point falls inside the hypersphere or it lies exactly on its border. To differentiate this case let us notice that if further $0 < \alpha_i < C$, then \mathbf{x}_i lies exactly on the border of the hypersphere. Such points are support vectors (SV). All other points are inliers for which, $\alpha_i = 0$.

One-class Support Vector Machines (OC-SVM) – ctd.

Inliers:	$\ \mathbf{x}_i - \mathbf{a}\ ^2 < r^2$ then $\alpha_i=0$, $\beta_i=C$, and $\xi_i=0$
SVs:	$\ \mathbf{x}_i - \mathbf{a}\ ^2 = r^2$ then $0 < \alpha_i < C$, $\beta_i > 0$, and $\xi_i=0$,
BSVs:	$\ \mathbf{x}_i - \mathbf{a}\ ^2 > r^2$ then $\alpha_i=C$, $\beta_i=0$, and $\xi_i > 0$.

However, because at least two SVs are necessary to define a hypersphere (since they lie on its border, whereas the BSVs are excluded at the same time), if $C \geq 1$ then $C - \alpha_i - \beta_i = 0$ cannot be fulfilled for any i since it is also required that $\sum_i \alpha_i = 0$. Thus, in this case there are not any outliers, i.e. there are no bounded support vectors BSVs.

From Q_L the Wolfe dual form Q_W is obtained

$$Q_W = r^2 + C \sum_i \xi_i - r^2 \sum_i \alpha_i - \sum_i \alpha_i \xi_i + \sum_i \alpha_i \|\mathbf{x}_i\|^2 - 2\mathbf{a} \sum_i \alpha_i \mathbf{x}_i + \|\mathbf{a}\|^2 \sum_i \alpha_i + - \sum_i (C - \alpha_i) \xi_i$$

which reduces to

$$Q_W = \sum_i \alpha_i K(\mathbf{x}_i, \mathbf{x}_i) - \sum_j \alpha_j \sum_i \alpha_i K(\mathbf{x}_i, \mathbf{x}_j) \quad \longrightarrow \quad K(\mathbf{x}_i, \mathbf{x}_j) \text{ denotes a scalar product}$$

However, $K(\mathbf{x}_i, \mathbf{x}_j)$ can be a kernel function which transforms points into a higher dimensional feature space.

One-class Support Vector Machines (OC-SVM) – ctd.

Based on the above derivations, a close formula of a distance d from the center \mathbf{a} of the hypersphere to a test point \mathbf{x}_x . Then if $d \leq r$, i.e.

$$d^2(\mathbf{x}_x, \mathbf{a}) \leq r^2$$

we classify \mathbf{x}_x as belonging to the class. Otherwise, it is an outlier.

$$d^2(\mathbf{x}_x, \mathbf{a}) = \|\mathbf{x}_x - \mathbf{a}\|^2 = K(\mathbf{x}_x, \mathbf{x}_x) - 2K(\mathbf{x}_x, \mathbf{a}) + K(\mathbf{a}, \mathbf{a}) \quad \text{but} \quad \mathbf{a} = \sum_i \alpha_i \mathbf{x}_i$$

$$d^2(\mathbf{x}_x, \mathbf{a}) = K(\mathbf{x}_x, \mathbf{x}_x) - 2 \sum_{i \in \text{Idx}(SV)} \alpha_i K(\mathbf{x}_x, \mathbf{x}_i) + \sum_{j \in \text{Idx}(SV)} \alpha_j \sum_{i \in \text{Idx}(SV)} \alpha_i K(\mathbf{x}_i, \mathbf{x}_j)$$

where $\text{Idx}(SV)$ denotes a set of indices of all the support vectors found for this problem. The summation in the above takes on only such \mathbf{x}_i which are SVs, since for the inliers we have $\alpha_i = 0$, whereas BSVs do not fulfill the optimization criteria.

$$\forall_{x_s \in SV} r^2 = d^2(\mathbf{x}_s, \mathbf{a}) = K(\mathbf{x}_s, \mathbf{x}_s) - 2 \sum_{i \in \text{Idx}(SV)} \alpha_i K(\mathbf{x}_s, \mathbf{x}_i) + \sum_{j \in \text{Idx}(SV)} \alpha_j \sum_{i \in \text{Idx}(SV)} \alpha_i K(\mathbf{x}_i, \mathbf{x}_j),$$

one of the support vectors

OC-SVM classification rule:

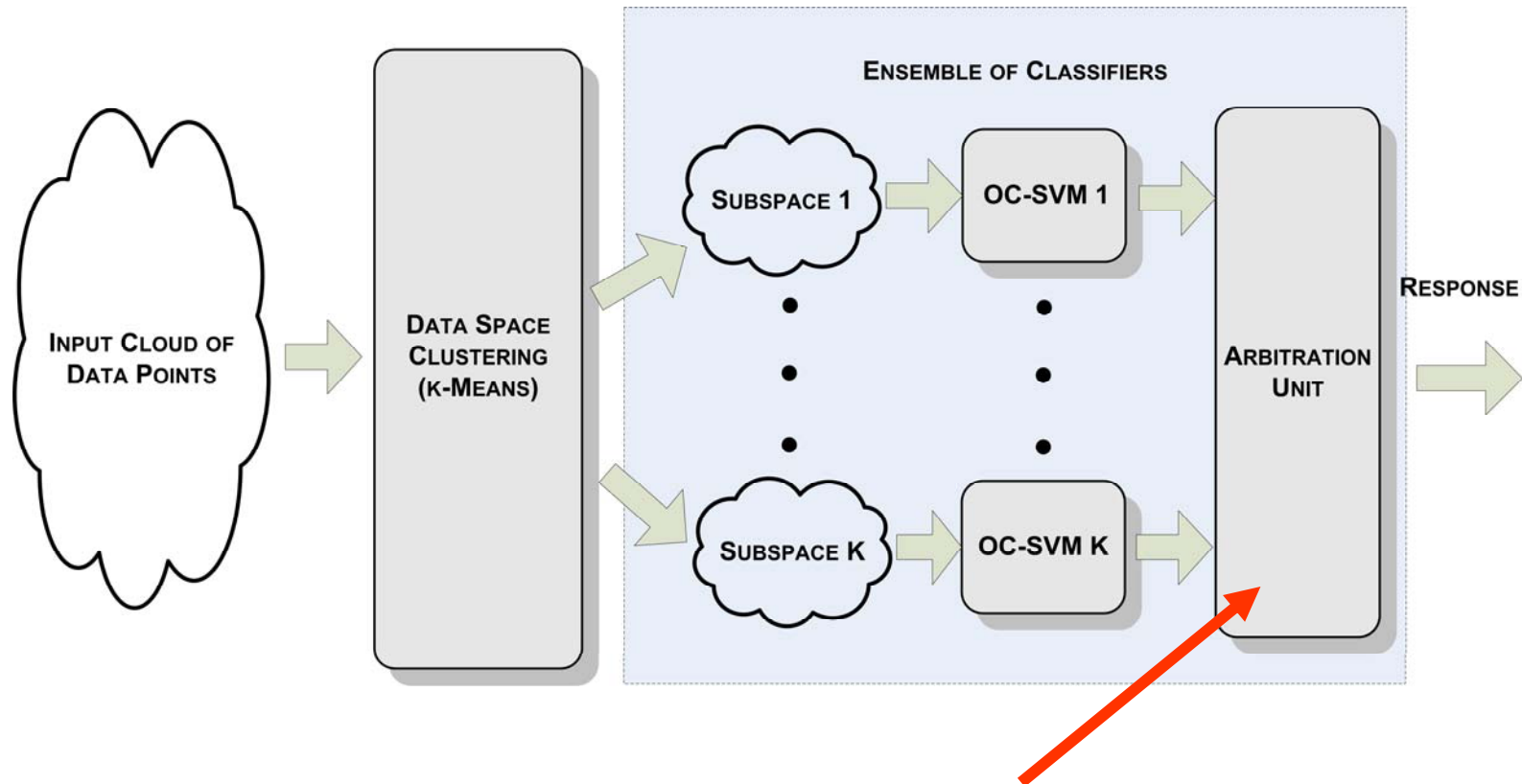
$$\sum_{i \in \text{Idx}(SV)} \alpha_i K(\mathbf{x}_x, \mathbf{x}_i) \geq \sum_{i \in \text{Idx}(SV)} \alpha_i K(\mathbf{x}_s, \mathbf{x}_i) = \tau$$

System Training:

The goal of the system training is to achieve the best performance measured as the **best accuracy** and **fast operation**. In practice however, the accuracy factor is difficult to measure, since we lack sufficient reference data. The only available ones are samples of the points of interest. To some extent performance can be measured with $\#SVs/\#D_i$. However, we found experimentally that if for some SVM this is greater than 0.1 then performance is usually inferior.

1. Select a number of clusters M , which determines number of expert OC-SVM classifiers. For practical reasons this was in the range 1-25.
Select number of trials T .
2. For each $1 \leq m \leq M$:
3. Do T times:
 4. Randomly select the centers μ_m and run the k-means procedure.
 5. For each partition D_i search the best parameters ν and γ to train its expert OC-SVM such as to maximally separate D_i . In experiments we used the grid search over $\nu \in \{0.005, 0.01, 0.05\}$ and $0.001 \leq \gamma \leq 48$ in steps of 0.02.
 6. Train each OC-SVM with the best parameters found in the previous step.
Compute its fitness measure (1).
 7. Compute the overall fitness measure (2) and if it is *minimal* store the whole configuration (all α and SVs).

System Run-Time Arbitration:



During point testing each expert OC-SVM provides its yes/no answer. The arbitration unit, is responsible for selecting the final answer of the ensemble. However, it can happen that a point is classified as a positive by more than one classifier. Therefore it is assumed that a test point is classified as a positive one iff it is classified as a positive one *exclusively by one of the experts* in the ensemble.

Experimental Results:

EXPERIMENTAL SETUP...

The entire system was implemented in C++ and run on the computer with the 2GB RAM and with the processor Pentium Core 2 T7600 @ 2.33GHz. Below we present results obtained in a set of 16 different images in which 11174 skin sample points were manually selected. Then all the points are k-means segmented into disjoint partitions. These partitions do not necessarily follow image boundaries. Then each partition is used to train a dedicated OC-SVM. More precisely, randomly selected 90% of a partition is used for training, whereas the rest 10% for validation. This process is repeated for iteratively changed parameters γ and ν . The best value in terms of correct answers and #SV is stored

Experimental Results:

Values of five partitions in our experiment: number of points in each cluster, cumulative distance in each cluster, number of support vectors necessary to bound a cluster

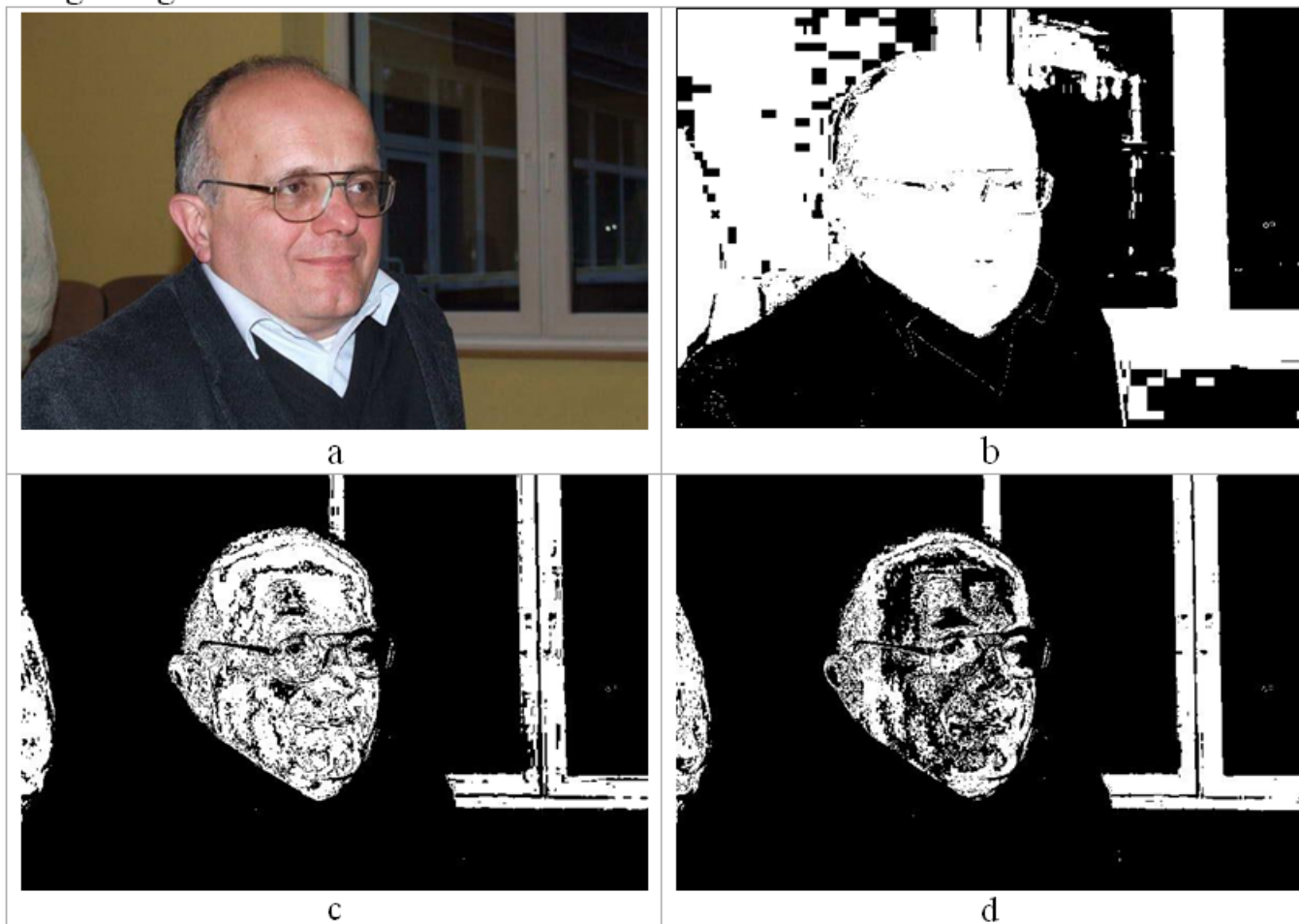
$m, \text{ for } M=5$	$\#D_m$	S_m	$\#SVs$
0	2956	1 428 400	9
1	2500	1 326 400	6
2	3894	1 303 600	7
3	831	700 021	8
4	993	518 922	7

Experimental Results:

Results of image segmentation for skin detection. An original 318x480 image (a). Segmentation for $k=1$, i.e. one expert OC-SVM (b), for $k=5$ (c), $k=10$ (d). In all variants the best training parameters γ and ν were found by cross validation.



Experimental Results:



Experimental Results:

The best partitioning of the initial space with the k-means algorithm was obtained in the RGB color space. However, it happened often that for the RBF OC-SVM slightly better results were obtained with the Farnsworth color space. In general this can depend on a data set and the kernel. Parameter M was in the range of 1 to 25, and T was set to 5-10. Since we assume there are only singular outliers in the training data set, the parameter ν was chosen from the set of three values $\{0.005, 0.01, 0.05\}$, whereas $0.05 \leq \gamma \leq 64$, in steps of 0.05.

Performance parameters for different number M of initial clusters

M	1	5	10
Av. #SVs	8	7	9
Av. answer time [s] per image (318x480)	1.9	2.2	3.6
Av. precision	0.77	0.89	0.94

Conclusions:

The method was applied and tested in the task of image segmentation based on manually selected color samples of the human skin. However, the areas of application can be much wider, such as e.g. medical or image annotation.

Pros:

1. The number of clusters M can start from 1 which means only one “classical” OC-SVM;
2. Different features can be used for segmentation and different for training of the OC-SVM thus introducing an *a priori* knowledge;
3. The method naturally builds a parallel structure (possible improvement of performance);
4. System can be used for novelty detection (e.g. tumors in medical images, malfunctions of a machinery, etc.), as well as other applications such as image annotations, etc.

Cons:

1. The best configuration can depend on a data set; The number of parameters to set is $2M+1$;
2. There are not strict rules of choosing some parameters (e.g. M , T , and γ).
3. If well balanced positive/negative data sets are available then classical binary SVM can lead to better results.

Thank you!