3D Modeling with Support Vector Machines

Montilla G1, Bosnjak A1*, Paluszny M2 and Villegas H1

1The Image Processing Center Universidad de Carabobo, Valencia, Venezuela
2Graphical Computer and Applied Geometry Center, Universidad Central de Venezuela, Caracas, Venezuela

Abstract

We developed a new approach based on Support Vector Machines (SVM) to model solids defined by a set of points on their surface. In problems of classification, regression and support of distribution, SVM are focused in the hyper-planes of maximum margin in the feature space. However, the forms that could be described by these surfaces when they return to the input space have not been studied in depth. In this paper, these surfaces are used to model complex objects, connected or non-connected, with a great amount of points, in order of tens of thousands, and with various topologies (hollow, branches, etc.). Two constrains were kept: 1) The use of traditional algorithms of SVM theory; and 2) The design of the appropriate training sets from the object. This combination produced a novel tool and the results obtained illustrated the potential of the proposed method. Therefore, this new application of SVM of Vapnik is capable of creating surfaces of decision and estimation functions, which are well adjusted to objects of complex topology.

Keywords: Computer graphics; Surface modeling; Machine learning; 3D modeling; Support vector machines

Introduction

Modeling can be defined as the process of finding a mathematical function that best represents a set of given data from an experiment, with both precision and good generalization. In computer graphics, modeling allows to extend the knowledge of an object represented by a cloud of points. For example, an implicit model can be used to evaluate in a simple way the inside and the outside of an object. An implicit model also can be used to visualize the object by ray casting without an intermediate representation. The implicit model makes easier the detection of proximity and collisions. The implicit model can also compress the information; it provides estimation of parameters, such as curvature; and by using logical operations, the implicit model produce more complex objects.

A great diversity of approaches for modeling of clouds of points exists in computer graphics. One of these advanced methods is the variational approach [1]. In this method, radial basis functions are adjusted to the surface of the object using a system of equations, in the N^2 order, where N is the number of points of the object. The radial basis functions |x|^2 and |x|^2 log|x| are proposed in [1]. This approach solves complicated topologies with great safety. Another approach is to use a learning machine to guide a parametric model. For example a Genetic Algorithm (GA) can be used to fit the parameters of an implicit function based on a quality criteria or a GA adaptation function. This method has been used to model organs of the human anatomy with simple surfaces, for example: the super quadrics [2,3]. The objective of this approach is to describe the organ with very few parameters. The approach used in this research is similar to the variational method for its data-driven characteristic and is similar to the GA approach for using a learning machine as search engine.

The SVMs are frequently used in tasks of pattern recognition, like face recognition [4], characters recognition [5], disease diagnosis [6], and genetic classification [7]. Their application had been limited in Computer Graphics. Faloutos et al. [8] provided an example for applying the SVM to characters animation. The use of the SVM machines on surface modeling has been proposed by Garcia and Moreno [9]. They used the One Class SVM for the modeling of 3D objects. In this work, we analyzed the usefulness and limitations of this approach.

An important characteristic of the SVM machines is the data mapping to another space, called feature space, where the problem is solved with simple geometries. For example, Garcia and Moreno [9] moved the data points inside of one sphere in the feature space. Schölkopf et al. [10] reformulated the SVM machine in order to project the points of the object between two parallel planes in the feature space, where the training set is the whole object. The proximity of these planes represents the accuracy of the model. In this work, we used a single plane geometry in the feature space. Our proposal is different from Schölkopf’s in several aspects: (1) In the Schölkopf’s method the Support Vectors (SV) is located on the two planes, and the model oscillates among those two limits. In our method the points of the object move away from the surfaces where the support vectors reside and the model interpolates the object points. (2) In our method, the accuracy is fixed beforehand. (3) Our proposal uses traditional machines. The objectives of the present work are to: (1) model with the traditional SVM, in order to use the more efficient algorithm available, (2) compare different strategies for building the training sets, and (3) analyze the modeling quality using criteria of accuracy and compression.

This work is organized as follows. One class svm, Binary classifiers, Regression machine -svr and -svr present the five traditional SVM machines, under the optics of surface modeling. Solid Modeling presents the modeling of single bi-dimensional objects using “One Class SVM”, the binary classifier and the regression machine. In this section we propose several methods to design the training sets for the binary classifier and the regression machine. Four criteria of quality are used in this section: compression, accuracy, classification error, and computation time. Section 6 introduces a multi-kernel-multi-scale proposal. Multikernel-Multiscale Machine is dedicated to the modeling of 3D objects [11] with several topologies, with thousands of points to...
tens of thousands of points, and the compression is analyzed. Section 8 presents the conclusions.

One Class Svm

The surface of an object can be represented by a set of connected or disconnected points in the bi-dimensional or three-dimensional space. Two problems can be proposed: (1) to find a surface that contains the object, (2) to find a surface that is contained by the object. Sometimes it is possible to get a surface sufficiently near to the object to consider that it represents the object. These concepts will orient the study of the “One Class SVM” to model a solid surface.

Formulation of the one class svm

The SVM theory proposes to project a set of Points \( \{x_i\} \subset \mathbb{R}^n, i = 1, \ldots, l \) to a space of different dimension (feature space) using a transformation \( \Phi : \mathbb{R}^n \rightarrow \mathbb{R}^{m(\Phi)} \). In the feature space the points can be restricted to simple geometries like the interior of a sphere or behind a plane. When the plane or the sphere returns to the input space they will contain the object in its interior. This proposal is presented in Figure 1 and the following equations describe the optimization problem.

\[
\begin{align*}
\min_{w, \xi, \rho} & \quad \frac{1}{2} ||w||^2 + \frac{1}{2} \sum_i \xi_i - \rho \\
\text{subject to} & \quad w^T \Phi(x_i) \geq \rho - \xi_i, \\
& \quad \xi_i \geq 0, \quad i = 1, \ldots, l.
\end{align*}
\]

(1)

(2)

(3)

The left term of equation (2) represents a planar function that fills the feature space, this function is oriented by the vector \( w \) (see Figure 1) and is evaluated at the point \( \Phi(x) \). We call this planar function the “Distance Function”. The points that produce the equal sign in equation (2), with \( \xi \) null, fall on the plane located to a distance function \( \rho \) from the origin. Let’s consider that \( \rho \) represents the distance from the plane to the origin, although the true distance of the plane to the origin is \( \rho' = ||w|| \). From equation (2) several points are projected behind the plane and others fall in front of the plane, depending on the value of the parameter \( \xi \). Equation (1) contributes to diminish the magnitude of the variable \( w \) to obtain a smooth distance function; this condition is propagated to the input space. An ideal situation for modeling is that the points are located very close to the plane when the machine is optimized. The One Class SVM proposed by equations (1)-(3), cannot reach an optimal situation of modeling, except for simple objects.

However, it can produce models which contain the object, which is important in applications of collision analysis.

Equation (1) combines three variables to be optimized: the smoothness of the function, the error and the distance from the plane to the origin. The parameter \( \nu \) controls the error minimization. We will discuss later this parameter. We use the denominations interior points, border points and outliers for the points behind the plane, over the plane and in front the plane.

To solve equations (1)-(3) the method of the Lagrange multipliers is used. The Lagrangian of equation (4) is re-written in the equation (5).

\[
\begin{align*}
L(w, \xi, \rho, \alpha, \beta) &= \frac{1}{2} ||w||^2 + \frac{1}{2} \sum_i \xi_i - \rho - \sum_i \alpha_i (w^T \Phi(x_i) - \rho + \xi_i) - \sum_i \beta_i \alpha_i \beta_i \cdot \rho' \\
& \quad \text{subject to} \quad \alpha_i \geq 0, \quad \beta_i \geq 0, \quad i = 1, \ldots, l.
\end{align*}
\]

(4)

(5)

Equations (6)-(9) result from the differentiation procedure.

\[
\begin{align*}
\sum_i \alpha_i (w^T \Phi(x_i) - \rho + \xi_i) &= \frac{1}{2} \sum_i \alpha_i \beta_i \Phi(x_i) \\
\sum_i \alpha_i = 1 \\
L(w) &= -\frac{1}{2} ||w'||^2
\end{align*}
\]

(6)

(7)

(8)

(9)

Combining equations (6) and (9) we obtain equation (10), which has a minimum value. \( \mathcal{L}(\alpha) = \frac{1}{2} \sum \alpha_i \beta_i \Phi(x_i)^T \Phi(x_j) \)

(10)

The "dual problem" in the \( \alpha \)-space is presented in equations (11)-(13), which we obtain from equations (7), (8) and (10). The inner product in equation (10) between vectors \( \tilde{x}_i \) and \( \tilde{x}_j \) in the feature space is replaced by the term \( K(\tilde{x}_i, \tilde{x}_j) \) in equation (11).

\[
\begin{align*}
\min_{\alpha} & \quad \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j K(x_i, x_j) \\
\sum_i \alpha_i &= 1 \\
0 & \leq \alpha_i \leq \frac{1}{l}, \quad i = 1, \ldots, l
\end{align*}
\]

(11)

(12)

(13)

To solve the optimization problem it is sufficient to know the inner product \( K(\tilde{x}_i, \tilde{x}_j) \), which is known as the kernel of the transformation between the input space and the feature space. The Mercer’s condition establishes that a valid kernel is one whose matrix \( K(\tilde{x}_i, \tilde{x}_j) \) is positive defined. There exists a great amount of possible kernels, such as the Gaussian kernel of equation (14) and the polynomial kernel of equation (15).

\[
\begin{align*}
K(x_i, x_j) &= \exp(-||x_i - x_j||^2) \\
K(x_i, x_j) &= (x_i^T x_j + c)^p
\end{align*}
\]

(14)

(15)

Comparing two geometrical approaches

As we mentioned early in Formulation of the One Class SVM, the points in the feature space can be restricted to simple geometries like the interior of a sphere or behind a plane. In this section we verify that the hyper-sphere of minimum radius and the plane further away from the origin give similar solutions when a Gaussian kernel is considered.

With a Gaussian kernel the points in the feature space are projected on the surface of a sphere of radius 1, as it is shown in Figure
In the distance function \( \rho \), the non-null components called the support vectors are considered. The support vectors contribute to the sum of equation (20) while the interior vectors do not contribute to it. The support vectors are split in border vectors which satisfy equation (20) and the outliers which satisfy the condition \( \xi > 0 \) from equation (21). Table 1 presents a summary of this classification.

\[
\begin{align*}
\alpha_i &= 0 & \xi &= 0 & w^T \Phi(x_i) > \rho \\
0 < \alpha_i < \frac{1}{\nu l} & \xi &= 0 & w^T \Phi(x_i) - \rho = 0 \\
\alpha_i &= \frac{1}{\nu l} & \xi &= 0 & w^T \Phi(x_i) - \rho < 0
\end{align*}
\]  

Equations (1)-(3) contain a single input parameter. In this section we analyze the effect of \( \nu \) on the number of support vectors and the outliers (allowed error). This analysis is based on equation (12). The sum in this equation contains \( n_w \) non-null components called the support vectors, from which there are \( n_{sv} \) support vectors considered as outliers. In a first step we apply this sum to equations (20) and (21) to obtain equation (22). Further, in a second step we apply the sum

\[
D(x) = \sum_{i=1}^{l} \alpha_i y_i K(x_i, x) - \rho
\]  

\[
\text{Table 1: One Class SVM Vectors Classification.}
\]

2. Independently of the kernel, there is a minimum radius hypersphere which contains the points in the feature space. Figure 2 shows that there is a spherical cap containing these points, and this cap is reduced to a minimum size when the condition of minimum hypersphere radius or maximum plane distance is achieved. Therefore the two actions (maximize the distance \( \rho \) or minimize the sphere radius) have the same effect. The inner points keep closer to the plane than to the hyper-sphere of minimum radius, which from a modeling point of view signifies that the plane returns to the input space a better model than the hyper-sphere, nearest to the points cloud. In our experiments we have verified the correspondence between the obtained values of maximum distance of the plane and minimum radius of the sphere. In other words, we have verified that it is possible to obtain one value from the other, following two different algorithms.

The solution to the modeling problem

The distance function is represented by the term \( w^T \Phi(x) \) in equation (2) and therefore the term \( w^T \Phi(x) - \rho \) is the distance function with respect to the plane. By application of equation (6) to the distance function we obtain equation (16), which is the distance function with respect to the plane.

\[
D(x) = \sum_{i=1}^{l} \alpha_i y_i K(x_i, x) - \rho
\]  

Equation (16) distinguishes between interior and exterior, it is positive behind the plane, negative in front the plane and null over the plane. Furthermore, it represents the planar function in both spaces. In the input space, the surface of zero level of this function is the solution of the modeling problem.

Support vectors for the one class SVM

The Karush-Kuhn-Tucker’s conditions (KKT) establish that the product between the dual variables \( (\alpha_i, \rho) \) and the constraints, see equations (2)-(4), is null when the optimization is achieved. This is expressed in equations (17)-(18), obtained with the help of equation (7). These equations are valid only for the values which solve the optimization problem expressed in equations (11)-(13).

\[
\begin{align*}
\alpha_i (w^T \Phi(x_i) - \rho + \xi) &= 0 \\
\left( \frac{1}{\nu l} - \alpha_i \right) \xi &= 0
\end{align*}
\]  

From these equations, the vectors \( x_i \) are classified in two groups. The first one satisfies equations (19), which are called interior vectors. The second group, those with \( \alpha_i \neq 0 \) called support vectors. The support vectors contribute to the sum of equation (16) while the interior vectors do not contribute to it. The support vectors are split in border vectors which satisfy equation (20) and the outliers which satisfy the condition \( \xi > 0 \) from equation (21). Table 1 presents a summary of this classification.

\[
\begin{align*}
\alpha_i &= 0 & \xi &= 0 & w^T \Phi(x_i) > \rho \\
0 < \alpha_i < \frac{1}{\nu l} & \xi &= 0 & w^T \Phi(x_i) - \rho = 0 \\
\alpha_i &= \frac{1}{\nu l} & \xi &= 0 & w^T \Phi(x_i) - \rho < 0
\end{align*}
\]  

Tuning the input parameters of the one class SVM

Equations (1)-(3) contain a single input parameter. In this section we analyze the effect of \( \nu \) on the number of support vectors and the outliers (allowed error). This analysis is based on equation (12). The sum in this equation contains \( n_w \) non-null components called the support vectors, from which there are \( n_{sv} \) support vectors considered as outliers. In a first step we apply this sum to equations (20) and (21) to obtain equation (22). Further, in a second step we apply the sum
only to equation (21) to obtain equation (23). Equations (22) and (23) produce equation (24). Equations (25) and (26) are obtained from equation (24). Finally, equation (27) is obtained from equation (25).

\[ \sum_{i=1}^{n} \alpha_i = 1 - \frac{n_{\text{nu}}}{v I} \]  
\[ \sum_{i=1}^{n} \alpha_i n_{\text{nu}} v I \leq 1 \]  
\[ \frac{n_{\text{nu}} v I}{v I} < 1 \]  
\[ \frac{n_{\text{nu}} v I}{v I} > \frac{v}{v} \]  
\[ 0 < v < 1 \]

According to equation (25) \( v \) fixes an upper limit to the fraction of error vectors (the outliers) and a lower limit to the fraction of support vectors. A high value of \( v \) produced many support vectors and outliers and a low value of \( v \) had the opposite effect; it is possible to reduce to zero the number of outliers.

These results are used in Figures 3(a) and 3(b). In Figure 3(a) Gaussian kernel is used and \( v \) was adjusted to a low value (\( v=0.001 \)) to obtain zero outliers. In this case the plane comes back to the input space containing the object. If the factor \( v \) is changed to 0.999 the plane comes back to the input space contained by the object. For a polynomial kernel the conditions already discussed are inverted. Figure 3(b) shows the result for a low value of \( v \). This indicates a partial loss of control, since the points behind the plane in Figure 1, (called interior points), may come back to the interior or to the exterior of the model in the input space. The parameter \( v \) can be used to reverse this condition but at the expense of a large number of support vectors.

### Binary Classifiers

The SVM have been constructed by Vapnik [12] as learning machines which minimize the classification error, finding the hyperplane of maximum margin that separates two classes in the feature space. The problem of binary classification is discussed in the following sections.

#### Single binary classifier (svc)

Given a set of points in input space \( \{x_i\} \subset \mathbb{R}^n i=1,\ldots,l \) and a function \( \psi: \mathbb{R}^n \rightarrow \mathbb{R} \) which assigns to the points one of two possible values, Vapnik [11] proposed to map the problem to another space (feature space) using a transformation \( \psi: \mathbb{R}^n \rightarrow \mathbb{R}^m \). In the feature space the classes are linearly separable by a hyper plane of maximum margin. This proposal is presented in Figure 1, and the optimization problem is defined by the following equations:

\[ \min \frac{1}{2} \|\theta\|^2 + C \sum \xi \]  
\[ y_i (\psi(x_i) + b) \geq 1 - \xi \]  
\[ \xi \geq 0 i=1,\ldots,l \]

As before, we consider a planar function which represents a “Distance Function” in the feature space. This function is null over the hyper-plane of maximum separation (See Figure 4). Equation (29) indicates that all the points are projected behind the planes with distance function equals 1, except the border vectors that lie over the \( \pm 1 \) planes and outliers with a distance function \( 1 - \xi \). The variable \( w \) (gradient of the distance function) adjusts the smoothness of the function: a minimum value of \( w \) gives the maximum smoothness and a maximum separation between the two classes, since the real distance between the two planes of distance function 1 and -1 is \( 2 \|\theta\| \). Equation (28) presents a minimization problem with several objectives which includes the magnitude \( w \) and the sum of the lacking variables.

Equations (31)-(33) provide the dual problem obtained from the Lagrangian. As before it provides the kernel that induces the transformation between the two spaces \( K(\mathbf{x}, \mathbf{x}) \), which represents the scalar product in the feature space. Equation (34) represents the decision function of the classifier and the zero level surface of this function will be used to solve the modeling problem.

\[ \min \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) - \sum_{i=1}^{l} \alpha_i \]  
\[ 0 \leq \alpha_i \leq C i=1,\ldots,l \]  
\[ \sum_{i=1}^{l} \alpha_i y_i = 0 \]  
\[ D(\mathbf{x}) = \sum_{i=1}^{l} \alpha_i y_i K(\mathbf{x}_i) + b \]

#### v-svc classifier

The v-SVC classifier differs from the simple binary classifier in that the planes located to a distance 1. In Figure 4 are located to a distance 1/2. This indicates a partial loss of control, since the real distance between the two classes, since the real distance between the two planes of distance function 1 and -1 is \( 2 \|\theta\| \). Equation (35) presents a minimization problem with several objectives which includes the magnitude \( w \) and the sum of the lacking variables.

\[ \min \frac{1}{2} \|\theta\|^2 + \frac{1}{2} \sum \xi \]  
\[ y_i (\psi(\mathbf{x}_i) + b) \geq \rho - \xi \]  
\[ \xi \geq 0 i=1,\ldots,l \]  
\[ \rho \geq 0 \]

Equations (36)-(42) are obtained using the Lagrangian method. In these equations \( \alpha_i \), \( \rho \), \( \gamma \) are the dual variables associated to the restrictions defined by equations (36) and (37).

\[ w = \sum_{i=1}^{l} \alpha_i y_i \psi(\mathbf{x}_i) \]  
\[ \sum_{i=1}^{l} \alpha_i = \nu + \gamma \]  
\[ \nu \geq 0 \]  
\[ \gamma \geq 0 \]

Equations (43)-(46) provide the dual problem; the decision function is given by equation (34).

Figure 4: Points and hyper-plane in the feature space.
\[
\min \frac{1}{2} \sum_{i,j} a_i a_j y_i y_j K(x_i, x_j) + \sum_{i} c_i \tag{43}
\]

\[
0 \leq a_i \leq \frac{1}{l}, \quad i = 1 \ldots l
\]

\[
\sum_{i} a_i y_i = 0
\]

\[
\sum_{i} a_i \geq v
\]

\[
\text{Support vectors for the } \nu\text{-svc classifier}
\]

The Karush-Kuhn-Tucker (KKT)’s conditions establish that the product between the dual variables \((\alpha, \beta, \gamma)\) and the restrictions given in equations (36) and (37) is null when optimization is reached, with the help of equation (40) we obtain equations (47)-(49).

\[
\alpha \left[ y_i \left(\mathbf{w}^T \Phi(x_i) + b\right) - \rho + \xi_i \right] = 0
\]

\[
\bigg(1 - \frac{1}{l}\bigg) \xi_i = 0
\]

\[
\gamma \rho = 0 \tag{49}
\]

Following these equations the vectors \(x_i\) can be classified in two groups. A first group of vectors which do not contribute to the decision function, called the interior vectors, which satisfy equation (50).

\[
\alpha_i = 0, \quad \xi_i = 0, \quad y_i \left(\mathbf{w}^T \Phi(x_i) + b\right) > \rho
\]

A second group of vectors, the support vectors, those contribute to the decision function. The support vectors are split in border vectors which satisfy equation (51) and the outliers defined by equation (52). Table 2 presents a classification summary of these vectors.

\[
0 < \alpha_i < \frac{1}{l}, \quad 0 \leq \xi_i < l, \quad y_i \left(\mathbf{w}^T \Phi(x_i) + b\right) = \rho
\]

Equation (49) constrains the values of \(\rho\) and \(\gamma\) when arriving to optimization. Equation (37) establishes a margin \(\rho \geq 0\), but the objective is to maximize the margin and in consequence, only the classifiers with \(\rho > 0\) are the important ones. With these criteria we conclude that \(\gamma = 0\) when optimization is achieved, then equations (41) and (46) are transformed in equation (53).

\[
\sum_{i} a_i = \nu \
\quad \rho > 0 \tag{53}
\]

\[
\text{Tuning the input parameters of the } \nu\text{-svc classifier}
\]

We analyzed again the effect of the parameter \(\nu\) on the number of support vectors. The analysis is based on equation (53). The sum in this equation contains \(\nu n_o\) components different from zero or support vectors, including \(n_o\) outliers. In a first step we apply equation (53) to equations (51) and (52) to obtain the equation (54). In a second step, we apply equation (53) to equation (52) to obtain equation (55). These two equations are combined to obtain equation (56) and, finally, we obtain equations (57) and (58). The lower limit of equation (58) corresponds to zero error vectors \((a_i = 0)\), and the upper limit corresponds to the trivial case when all points of the training set are support vectors, namely \(n_o = l\).

Looking to equation (56) \(\nu\) sets an upper limit to the outliers and a lower limit to the support vectors. This behavior is similar to the one described in equations (25)-(27) for the One Class SVM.

\[
\sum_{i} a_i = \nu < \frac{n_o}{l} \tag{54}
\]

\[
\sum_{i} a_i = \frac{n_o}{l} < \nu \tag{55}
\]

\[
\frac{n_o}{l} < \nu < \nu
\]

\[
0 < \nu < 1
\]

\[
\sum_{i} a_i = \nu \quad \rho > 0 \tag{53}
\]

Table 2: Classification of the Classifier Vectors N-Svc.
\[
\begin{align*}
\min_{\alpha, \xi} & \frac{1}{2} \sum_{i=1}^{l} (\alpha_i - \alpha_i^*)^2 + C \left( \sum_{i=1}^{l} \xi_i \right) \\
& - \sum_{i=1}^{l} \alpha_i \left( y_i \phi(x_i) \right) + b = 0 \\
& \sum_{i=1}^{l} \left( \alpha_i - \alpha_i^* \right) = 0 \\
& 0 \leq \alpha_i, \alpha_i^* \leq \frac{C}{l} \quad i = 1, \ldots, l
\end{align*}
\] (59)

Equations (60)-(65) provide the dual problem, \( \alpha_i \) and \( \alpha_i^* \), are the Lagrange multipliers which establish the constraints given by equations (60) and (61).

The regression machine \( \nu \)-SVR differs from the machine \( \epsilon \)-SVR because it incorporates a priori parameter \( \nu \), which affects the band of zero error defined by \( \epsilon \), as can be seen in equation (66).

\[
\begin{align*}
\min_{\alpha, \xi} & \frac{1}{2} \sum_{i=1}^{l} (\alpha_i - \alpha_i^*)^2 + C \left( \sum_{i=1}^{l} \xi_i \right) \\
& - \sum_{i=1}^{l} \alpha_i \left( y_i \phi(x_i) \right) + b = 0 \\
& \sum_{i=1}^{l} \left( \alpha_i - \alpha_i^* \right) = 0 \\
& 0 \leq \alpha_i, \alpha_i^* \leq \frac{C}{l} \quad i = 1, \ldots, l \\
& 0 \leq \xi_i \leq \frac{C}{l} \quad i = 1, \ldots, l
\end{align*}
\] (66)

Equations (67) to (70) provide the dual problem. It is simple to demonstrate that if the band of zero error in equation (69) is restricted to the condition \( \epsilon > 0 \), equation (72) becomes equality.

\[
\begin{align*}
\min_{\alpha, \xi} & \frac{1}{2} \sum_{i=1}^{l} (\alpha_i - \alpha_i^*)^2 + \frac{C}{l} \left( \sum_{i=1}^{l} \xi_i \right) \\
& - \sum_{i=1}^{l} \alpha_i \left( y_i \phi(x_i) \right) + b = 0 \\
& \sum_{i=1}^{l} \left( \alpha_i - \alpha_i^* \right) = 0 \\
& 0 \leq \alpha_i, \alpha_i^* \leq \frac{C}{l} \quad i = 1, \ldots, l \\
& 0 \leq \xi_i \leq \frac{C}{l} \quad i = 1, \ldots, l
\end{align*}
\] (69)

The estimation function for both machines is given in equation (74). When we compare this equation with equation (34) for the binary classifier we observe that both machines have the same modeling capacity. On the other hand, when we compare with equation (16) for the One Class SVM we observe that the last one admits only positive coefficients. This restriction limits its modeling capacity. Later on, we will make a more extensive comparison between the binary classifier and the regression machine for modeling geometric solids.

\[
f(x) = \sum_{i=1}^{l} (\alpha_i^* - \alpha_i) \phi(x_i, x) + b
\] (74)

The inner vectors between the two planes for the two regression machines do not contribute to the estimation function. The support vectors are represented by the vectors lying outside the planes or over the planes. As before we can prove that \( \nu \) controls the proportion of outliers and support vectors, and equations (25)-(27) and (56)-(58) continue to be applicable.

**Solid Modeling**

In the following sections several approaches are proposed to model connected and disconnected objects using the three SVM algorithms proposed by Vanik. Attention will be focused in the design of the SVM training set.

**Modeling with “one class svm”**

In this Section the behavior of the “One Class SVM” is analyzed following to the focus of solid modeling. In Figure 6(a) a simple object is modeled using a Gaussian kernel. All the points have been applied in the training set. Attention will be focused in the design of the SVM training set.

![Figure 6](image6.png)

**Figure 6:** (a) The model interpolates all the points. We used a Gaussian kernel with \( \gamma = 40 \). The value of \( \nu \) is taken to be 0.1. (b) To add points in the interior of the object do not affect the model. (c) The model separates the components of a non-connected object (\( \gamma = 60 \) \( \nu = 0.01 \)).

![Figure 7](image7.png)

**Figure 7:** (a) Cloud of points which represent the edges of a hollow object. (b) Model obtained using the One Class SVM with Gaussian kernel with \( \gamma = 200 \), \( \nu = 0.05 \). (c) Model obtained with the method of section 5.2.

![Figure 8](image8.png)

**Figure 8:** (a) Artificially created classes and regions produced by the binary classifier (\( C = 90 \), \( \nu = 100 \)). The height of the object is 0.25 and \( \epsilon = 0.01 \) (4%). (b) Curves of levels 1 \( \gamma = 1 \), together with the support vectors (c) Model over the original object points. (d) Model and support vectors.
interpolated by the zero level contour of the distance function to the plane, see equation (16). In the feature space shown in Figure 2, the points of the object are projected to the intersection of the hyper-plane, the sphere of radius 1 and the sphere of minimum radius. To be sure that the model touches all the points of the object, these points have to be border vectors. However, there is not an input parameter which controls the modeling error, when the points are not border vectors. Only the parameter $\nu$ affects the error vectors as is shown in equation (25).

When inside points are added as it is show in Figure 6(b) the same model persist with the same support vectors. This follows from the fact that the inner vectors do not produce support vectors. The kernel parameters also affect the quality of the model, for example, the value of $\gamma$ for a Gaussian kernel determines the smoothness of the interpolated contour.

The “One Class SVM” can model non-connected objects as it is shown in Figure 6(c). This is a very important property that can be used to detect cell populations in a histological image.

The “One Class SVM” also can be used to detect the contours of an object with a hollow topology as it is shown in the Figures 7(a) and 7(b). In this case the model does not fit exactly to the convexities of the object as it is shown by the arrows in Figure 7(b). In this case it is impossible to reach a result as the one of Figure 7(c), which is obtained with a procedure to be explained in “Modeling with the Binary Classifier”.

**Modeling with the binary classifier**

In this Section we use the decision surface of the binary classifier to model objects. A binary classifier requires two classes and these can only be reproduced in an artificial manner from the surface object points. For this reason, we duplicate each point of the surface (Figure 8 (a)) and we move them in small quantities $+\varepsilon$ and $-\varepsilon$ from their initial position, in a sense normal to the surface of the object. In this way, we obtain a dipole with the tags $+1$ and $-1$, for each point of the object. The result is two layers of points with tags, which represent two versions of the original object. What our method states is that if $\varepsilon \to 0$ and if a binary classifier is able to separate the classes with a zero classification error, independent of the object topology, then this surface models the object with an error less than $\varepsilon$.

In Figure 8(a), this method is apply to an object of size 0.25 using $\varepsilon = 0.01$ (4% the size of the object). We have chosen a high value of $\varepsilon$ to obtain a visual perception of this method. In this example we obtain accuracy less than 4% when we used the decision surface to model the given object.

Dipoles of Gaussian functions participate in the method, when we use the Gaussian kernel. In principle, for a symmetric dipole, the zero level surface goes thru its centre and an asymmetry between the two dipoles produces a displacement of the surface. However, most of dipoles disappear as it is observed in Figure 8(d) and only survive few dipoles in regions of higher curvature. The curves of level +1 and -1 (Figure 8(b)) define limit for the model position. An important characteristic of this method is that the points on the surface of the object are not included in the training set and therefore they are not part of the support vectors. In this way the model does not visit the curves of level +1 and -1, and, in consequence, we obtain a better approach to the object.

The SVM input parameter can be adjusted according to two criteria: compression and precision. In Figure 8 the maximum error (separation object-model) was 0.004 (median 0.0019), for a prefixed error value of $\varepsilon = 0.01$. The number of points of the object is 24, the training set is 48, and the support vectors number is 20, therefore we
have a low compression 24/20. When we adjust the parameters $C$ and $\gamma$ to optimize the SVM machine as function of precision; the maximum error was reduced to 0.00089 (median 0.00045).

The proposed method also models non-connected objects as it is shown in Figure 9. In this case the classifier was adjusted to obtain maximum compression. The object contains 84 points, the training set is composed of 168 points and the support vectors are 45 (see Figure 9(d)). The compression reached is 1.87 with respect to the object and 3.83 with respect to the training set. Figure 7(c) proves that the method is also useful to model objects of hollow topology. It will be interesting to check the behavior of this method for 3D objects.

**Modeling with the regression machine**

For the regression machine, we designed several training sets and its elements was labeled with one three possible values, namely, $1$, $0$ and $-1$. We considered the training set as a sample of a continuous function which covers the data space. The task of the regression machine is to estimate this function with an error previously defined. The surface of zero level of the estimation function is interpolating the object points passing at a distance which depends of the a priori error.

Figures 10(a) and 10(d) show two types of symmetric training sets. In Figure 10(a) the training set contains three layers of points, where the central layer is the object. The layers are separated by a prefixed $\varepsilon$ value. We assigned the values $1$, $0$ and $-1$ to these three layers. The surface calculated by the regression machine (equation (74)) crossed the plane where the data reside and its intersection with this plane defines a contour of zero level which interpolates the object points. This concept can be easily extended to 3D. Figure 10(g) corresponds to the model analyzed in Modeling with the Binary Classifier, for the binary classifier, which, we include to compare the behavior of two machines, classification and regression, with the same object.

Figures 11(a) and 11(d) depict two novel training set proposals. In Figure 11(a) the external layer goes away from the object and in Figure 11(d) both layers go away from the object. Again we assigned the values $1$, $0$ and $-1$ to these three layers. For the five proposals of Figure 10 and 11, the left column represents the model with maximum compression superimposed to the training set, the central column represents the support vectors superimposed to the model of maximum compression and the third column represents the model with optimal precision superimposed to the original points of the object.

**Analysis of results**

Tables 3 and 4 show the results of the experiments. Learning is guided by the criteria of optimum compression and optimum precision (minimum error). Columns 3-6 of these tables present the machine type and its parameters. The next three columns quantify the size of the training set, the number of support vectors and the number of outliers. The next column quantifies the execution time. The last two columns present the modeling error, evaluated from a first order approximation given by the equation $D = f(x) f'(x)$ where $f'(x)$ is the derivative of the prediction function. This equation is valid when the points are really closed to the model. It loses precision as long as the points get farther away. However, the resulting value indicates the behavior of the error.

From tables 3 and 4, we perform the following analysis.

<table>
<thead>
<tr>
<th>Figures</th>
<th>Optimizes</th>
<th>SVM</th>
<th>$C$</th>
<th>$\varepsilon$</th>
<th>$\gamma$</th>
<th>$n$</th>
<th>$n_{sv}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>10(a-c)</td>
<td>Compression</td>
<td>$\varepsilon$ SVR</td>
<td>400</td>
<td>50</td>
<td>0.7</td>
<td>99</td>
<td>12</td>
</tr>
<tr>
<td>10(d-f)</td>
<td>$\varepsilon$ SVR</td>
<td>400</td>
<td>54</td>
<td>0.77</td>
<td>66</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>10(g-i)</td>
<td>C-SVR</td>
<td>400</td>
<td>70</td>
<td>-</td>
<td>66</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>10(a-c)</td>
<td>Precision</td>
<td>$\varepsilon$ SVR</td>
<td>400</td>
<td>420</td>
<td>0.07</td>
<td>99</td>
<td>56</td>
</tr>
<tr>
<td>10(d-f)</td>
<td>$\varepsilon$ SVR</td>
<td>400</td>
<td>440</td>
<td>0.1</td>
<td>66</td>
<td>45</td>
<td></td>
</tr>
<tr>
<td>10(g-i)</td>
<td>C-SVR</td>
<td>400</td>
<td>640</td>
<td>-</td>
<td>66</td>
<td>44</td>
<td></td>
</tr>
</tbody>
</table>

**Table 3: Experiments of figure 10.**

<table>
<thead>
<tr>
<th>Figures</th>
<th>Optimizes</th>
<th>SVM</th>
<th>$C$</th>
<th>$\varepsilon$</th>
<th>$\gamma$</th>
<th>$n$</th>
</tr>
</thead>
<tbody>
<tr>
<td>11(a-c)</td>
<td>Compress-ion</td>
<td>$\varepsilon$ SVR</td>
<td>400</td>
<td>190</td>
<td>0.3</td>
<td>76</td>
</tr>
<tr>
<td>11(d-f)</td>
<td>$\varepsilon$ SVR</td>
<td>400</td>
<td>200</td>
<td>0.122</td>
<td>47</td>
<td></td>
</tr>
<tr>
<td>11(a-c)</td>
<td>Precision</td>
<td>$\varepsilon$ SVR</td>
<td>400</td>
<td>400</td>
<td>0.055</td>
<td>76</td>
</tr>
<tr>
<td>11(d-f)</td>
<td>$\varepsilon$ SVR</td>
<td>400</td>
<td>400</td>
<td>0.004</td>
<td>47</td>
<td></td>
</tr>
<tr>
<td>Figures</td>
<td>Optimizes</td>
<td>SVM</td>
<td>$nE$</td>
<td>msec</td>
<td>$E_{\infty} 10^{3}$</td>
<td>$E_{\infty} 10^{3}$</td>
</tr>
<tr>
<td>---------</td>
<td>-----------</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-------------</td>
<td>--------</td>
</tr>
<tr>
<td>11(a-c)</td>
<td>Compress-ion</td>
<td>$\varepsilon$ SVR</td>
<td>0</td>
<td>31</td>
<td>7.92</td>
<td>4.22</td>
</tr>
<tr>
<td>11(d-f)</td>
<td>$\varepsilon$ SVR</td>
<td>0</td>
<td>10</td>
<td>7.91</td>
<td>4.05</td>
<td></td>
</tr>
<tr>
<td>11(a-c)</td>
<td>Precision</td>
<td>$\varepsilon$ SVR</td>
<td>0</td>
<td>1200</td>
<td>0.64</td>
<td>0.49</td>
</tr>
<tr>
<td>11(d-f)</td>
<td>$\varepsilon$ SVR</td>
<td>0</td>
<td>47</td>
<td>0.97</td>
<td>0.43</td>
<td></td>
</tr>
</tbody>
</table>

**Table 4: Experiments of Figure 11.**

![Figure 11](image1.png)

**Figure 11:** Two asymmetric training sets. First column: training set and the optimum compression model. Second column: support vectors with model of optimum compression. Third column: model of optimum precision superimposed to the object.

![Figure 12](image2.png)

**Figure 12:** (a) Outcome of figure 9. (b) Improvement applying regression to the residue.
Compression: The symmetric training sets produce the maximum compression, regardless of the symmetric model of three layers triple the size of the training set. This superiority is due to the “data-driven”, (characteristic of the SVM). In consequence, increasing the number of vectors nearby the object expands the universe of solutions. On the other hand, vectors far away from the object in the asymmetric models do not contribute to a better determination of the object.

Precision: The training sets of three layers produce a better precision. However, other experiments prove that if we reduce the displacement in the training set of the binary classifier we obtain the same precision than the one obtained for training sets of three layers.

Computation time: The classification machine had more stable computation time when we change the parameters between the conditions of compression and optimun precision. The regression machine with training sets of three layers increase sensitively the computation time as function of precision. This is an important consideration to be taken because the used object is 33 points size, if we intent to model a one hundred bigger object and, in addition, the learning algorithm had a N² behavior, therefore the computation time for the classifier goes from 16 ms to 160 seconds (2 minutes and 40 seconds) and for the regression machine using the training set of three asymmetric layers the computation time increases from 1.2 s to 12000 s (3 hours and 20 minutes).

Multikernel-Multiscale Machine

A novel proposal consists in applying a regression method over to the residue. A regression machine of the type ε-SVR or ν-SVR is trained using the minus error produced by the first SVM machine with some of the training sets already studied. As prediction function we use the sum of prediction functions of both machines. Figure 12 shows how this scheme reduces very much the position errors by a factor of 10 or bigger. This is achieved by adding the support vectors of both machines. The main objective of this technique is to supply the prediction function with kernels which had a bigger spatial spectrum. In the example of Figure 12 we increase the value of γ for the Gaussian kernel of the second machine by a factor of 10 with respect to the first machine. In this way the regression machine used the residue to refine the details of the model. The result is a multikernel – Multiscale approach.

3D Modeling

In this section we analyze the modeling of 3D objects. In Figure 3(a), we used the One Class SVM to covering an object. The obtained function can be used to evaluate in a simple style the closeness to the object, when detecting collisions. In this section we used two Vapnik machines for 3D modeling. In all experiments we adjusted the dimension and the position of the object to get the interior of a sphere of radius 1, which its center of mass is in the sphere center. In all experiments we use Gaussian kernel. The experiments were performed with a Pentium IV processor of 1.7 GHz.

Table 5 shows results of a series of experiments with a classifier ν-SVC, when modeling the surface of a version of 1488 points of the rabbit of Stanford University. We used a training set of two classes shifted 0.001 along the normal to the surface of the object. Instead of using the criteria of optimization already studied (precision and compression), we used the classification error as criterion of optimization. The forth column of Table 5 represents the number of
ill-classified vectors. A zero classification error guaranties that the decision surface of the classifier is located between the two layers of the training set. Figure 13(b) shows the obtained surface for zero classification error.

Precision and error of classification are deeply related, and the constraint for a priori separation between classes fixes a bound for the precision. But the lower limit of this a priori value depends on the capacity of the SVM machine to separate the two classes. For the Vapnik’s classifier the classification error is minimized for some value of parameters of the machine (\(\nu\), \(\gamma\) for the machine \(\nu\)-SVC, and \(C\), \(\gamma\) for the simple SVC). However, there is no warranty that the classification error reaches zero. If this is the case it is necessary to increase the given a priori separation between classes. This defines the limit to model with a single machine. Beyond that it is necessary to enter into the field of the multikernel-multiscale proposal studied in multikernel-multiscale machine.

As the SVM machine works far away from its limit of capacity to model, therefore it had the capacity to compress. This is verified in the experiment presented in Figure 14. In this figure we model a version of the Stanford University rabbit of 34,835 points. The training set is composed of 69,670 points and the classes are separated by 0.01. The obtained model (Figure 14(b)) contains 3,129 support points. This means a compression of 21.6 with respect to the training set or 10.8 means a compression of 21.6 with respect to the training set or 10.8 with respect to the object. In Figure 14(a) the surface of the object was colored with the position error affected by a non-linear fitting defined by a gamma correction of 0.1, to identify the low levels values of errors.

Among the training sets analyzed in Modeling with the Regression Machine, to model with the regression machine, we use the asymmetric training set of three layers, (see Figure 15). The zero value layers are the object. The inner layer (-1) is obtained by a displacement of the object points along the normal (0.01 in this experiment). The external layer (+1) is obtained throwing points randomly over the surface of a sphere of radius 1.5, taking care the separation between the points overcomes a minimal value. The object is contained inside of a sphere of radius 1.

Figure 16 shows two experiments for a more complex topology. The left object is created by four surfaces, two of them intersecting the teapot, and there is no continuity in the mesh. At left we used a \(\nu\)-SVC classifier, with a symmetric training set with a displacement of 0.001. This value is selected because the object had a region of small thickness. The greater error is observed in the inferior part of the figure, where we had two discontinue crossing mesh.

At right to the Figure 16, the package of the teapot is modeled, using a \(\varepsilon\)-SVR regression machine, applied on a symmetrical training set of three layers, with a displacement of 0.001. The machine generates a unique surface that is wrapping in it to adhere to the external and internal surfaces of the object. This is very high constraint because the thickness of the object is zero. A better result can be obtained when we removed this constraint and the object is closed. The high time of polygonization in Figure 16 (b) is due to the size of the cell of polygonization. Before we have been using a cell of size 0.01, and for figure 16(b) it was reduced to 0.005. This supports the convenience of using a non-uniform polygonization model. Another alternative is to perform a ray casting until the first point of crossing by zero value of the function, and applying the Phong model at the point found.

### Conclusion

In this paper we demonstrated the performance of the SVM machines to obtain mathematical models of the surface of objects. We verified that is possible to use the traditional SVM algorithms with the appropriate selection of the training set. Therefore, our method guarantees that the most efficient algorithm available can be selected. It was also demonstrated that the Vapnik machines are able to create surfaces from their functions of prediction that adjusted to objects of complex topology. The developed method is robust to variations in parameters of the SVM machines. Important values of compression and precision have been achieved with the proposed methodology. The method had been an acid test for the SVM classifiers and regression machines proposed by Vapnik [10]. It would be interesting to study if other learning machines can reach the same benefits.

### Acknowledgment

This research was funded by FONACIT-Venezuela, under project G-97000651 “Nuevas Tecnologías en Computación” (New Technologies in Computation). The experiments were developed using the VVM “Virtual Vision Machine” API [15]. VVM contains an SVM interface to the library developed by Chang & Lin [16,17] called LIBSVM.

### References

Reconstruction and representation of 3D objects with radial basis functions. SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques, 67—76.


