ISSN 1330-3651 (Print), ISSN 1848-6339 (Online)
https://doi.org/10.17559/TV-20180328175336
Original scientific paper

Feature Sensitive Three-Dimensional Point Cloud Simplification using Support Vector Regression

Veljko MARKOVIC, Zivana JAKOVLJEVIC, Zoran MILJKOVIC

Abstract: Contemporary three-dimensional (3D) scanning devices are characterized by high speed and resolution. They provide dense point clouds that contain abundant data about scanned objects and require computationally intensive and time consuming processing. On the other hand, point clouds usually contain a large amount of redundant data that carry little or no additional information about scanned object geometry. To facilitate further analysis and extraction of relevant information from point cloud, as well as faster transfer of data between different computational devices, it is rational to carry out its simplification at an early stage of the processing. However, the reduction of data during simplification has to ensure high level of information contents preservation; simplification has to be feature sensitive. In this paper we propose a method for feature sensitive simplification of 3D point clouds that is based on ε insensitive support vector regression (ε-SVR). The proposed method is intended for structured point clouds. It exploits the flatness property of ε-SVR for effective recognition of points in high curvature areas of scanned lines. The points from these areas are kept in simplified point cloud along with a reduced number of points from flat areas. In addition, the proposed method effectively detects the points in the vicinity of sharp edges without additional processing. Proposed simplification method is experimentally verified using three real world case studies. To estimate the quality of the simplification, we employ non-uniform rational b-splines fitting to initial and reduced scan lines.

Keywords: 3D scanning; 3D data acquisition; point cloud simplification; support vector regression

1 INTRODUCTION

Three dimensional (3D) scanning devices and 3D data acquisition techniques are nowadays frequently employed, not only in reverse engineering [1], but also in a number of different industrial applications [2, 3]. The state of art 3D scanners [4], especially optical ones, are characterized by high resolution and speed. They provide dense point clouds which carry profuse information about scanned object's surfaces [5] and, as a rule, contain a large number of points that make the processing of scanned data (e.g., feature extraction or CAD model generation) difficult, computationally expensive, and time consuming, thus requiring significant computational resources [6]. Besides, a large amount of point cloud data brings about high data transfer latencies and additional difficulties in cloud services and transmission of 3D data to Web clients.

To address these issues, one of the main objectives during point cloud data pre-processing is the reduction of number of points. This reduction has to ensure retention of information about initial surface shape; it has to be feature sensitive. Namely, dense point clouds are convenient for reconstruction of features that are represented by high curvature surfaces. On the other hand, when flat surfaces dominate in the part, dense clouds will contain a large number of points that carry little additional information about part's geometry. During point cloud simplification, ideal situation would be to remove redundant points on flat segments and to keep the most of the points from high curvature areas. In this way, the reconstruction of scanned object using simplified point cloud would lead to almost the same and likewise accurate results as the reconstruction based on original point cloud.

Simplification of point clouds has been in the focus of a number of research studies. The existing methods for point clouds simplification can be classified into two main categories: (1) mesh-based simplification methods, and (2) direct point cloud simplification methods. Mesh-based simplification methods start from a polygonal (usually triangular) mesh generated over point cloud and remove points from cloud using different criteria. These methods require implementation of meshing and re-meshing procedures, they are computationally intensive and consequently time consuming [7]. The review of mesh-based simplification methods can be found in [8].

Recently proposed methods carry out direct point cloud simplification without mesh generation and can be divided into two sub-categories: (1) methods for simplification of unstructured, and (2) methods for simplification of structured/organized point clouds. Direct simplification of unstructured point clouds is, as a rule, based on clustering methods. Divisive hierarchical clustering methods, such as octree decomposition using standard deviation of point normals as subdivision criterion [9], or binary partition based on points' deviation from tangent plane of cluster centroid, were proposed [7]. In [10], the authors propose recursive binary subdivision of clusters using k means algorithm and normal vector deviation. Research work from [11] presents an agglomerative hierarchical clustering approach that is based on adaptive mean-shift clustering scheme using points positions and normal vectors as features. In [12], the authors propose an agglomerative algorithm for merging 3D object's faces by applying 2D Boolean operations on orthogonal pseudo-polyhedra consecutive cuts. Fuzzy c-means clustering algorithm has been also employed [13].

Iterative re-clustering procedure based on Voronoi cells from [14] extracts the points with minimal distance to tangent planes of other points in cluster and declares them as new cluster representatives until stopping criterion (minimal geometric deviation of clusters) is met. Since this technique is prone to smoothing out sharp edges, the authors in [6] propose the method that first extracts the edge points, and afterwards removes less important points from clusters in smooth areas. Another method uses similar strategy for iterative removing the points from point cloud based on anticipated error between principle curvature vectors at point before and after its removal [15].

The other group of methods for direct point cloud simplification carries out the simplification of structured/organized point clouds. Namely, in most cases 3D scanners provide structured point clouds as low-level output. In these point clouds, points are structured into cross sectional planar scanned lines - scanned contours. Nevertheless, to get the information about the whole part, the part is usually scanned from multiple views that are registered into one coordinate system and integrated. This processing (registration and integration) deteriorates the structure of point cloud, and some devices provide unstructured point clouds at output, depending on the level of data processing carried out at the scanning device. However, in many applications, single-view scan data is sufficient. In addition, point cloud simplification can be carried out before registration and integration. Besides, there are a number of procedures that can be employed for point cloud structuring [16, 17]. Starting from this rationale, a number of techniques that are based on scan lines simplification were proposed.

Most frequently employed methods for scan line points reduction are uniform sampling, chordal distance method, spatial method, straightness method, and height decision method [16, 18]. To improve the accuracy of these methods, fuzzy inference mechanisms (FIMs) for decision on point elimination in chordal distance method [16], straightness method [19], and spatial method [20] were introduced.

An approach presented in [21, 22] removes points from scan line based on distance from Akima spline interpolated through remaining points in planar [21], and bi-Akima spline in spatial scan lines [22]. A research on application of continuous wavelet transform in feature sensitive simplification of scan lines is presented in [23].

Analysis of the related methods shows that in clustering based methods, selected points in simplified point cloud are cluster centroids [10,11] or representative points from cluster [6,9]. Thus selected points lead to smoothing out the edges, and make the point clouds unsuitable for reverse engineering, product quality control [4] and similar tasks. To deal with this issue, extraction of points on edges [6, 10] or filtering of the original point cloud [13] before simplification were proposed. However, this kind of pre-processing brings about additional computational efforts and time consumption. On the other hand, the main shortcoming of the scan line based reduction methods is that, as a rule, they assume that the points are connected by straight lines, and do not take into account the deviation that removal of certain point introduces to simplified scan line [20, 22].

In this paper we propose a method for simplification of structured point clouds that employs ε-insensitive support vector regression (ε-SVR) with *spline* and *b-spline* kernels [24] for finding the significant points from high curvature areas in scanned lines. Due to the properties or ε-insensitive SVR, the method inherently extracts the points from high curvature areas in G1 continuous lines. Elaboration of the initial idea that considered only *b-spline* kernel [25], and the comparison of *b-spline* and *spline* kernels that we perform in this paper, show that *spline* kernel leads to better overall performances. In addition, as we present in the paper, the method recognizes the points in the vicinity of edges that are represented as abrupt changes in scan lines with G0 continuity. It is not prone to smoothing out sharp edges and it does not require additional processing for edges detection. These are the major benefits of the proposed method with respect to alternative methods. In our approach, the reduced point cloud retains a high level of initial information about scanned object's shape and structure, as will be illustrated in the presented case studies.

It should be noted that support vector machines (SVM) were frequently employed for classification of objects based on features obtained from 3D point clouds [26, 27]. In addition, procedures for 3D point cloud smoothing, de-noising and hole filling based on ε-SVR using compactly supported Wu function kernel [28] and for surface reconstruction based on ε-SVR with Gaussian kernel [29] were proposed. However, the potentials of SVM in point cloud simplification have not been exploited yet.

The reminder of the paper is organized as follows. Section 2 contains basic background about ε-insensitive SVR (ε-SVR) which is essential for the proposed method for point cloud simplification. In Section 3 we present proposed method in details and elaborate on the influence of ε-SVR parameters on simplification results. In addition, in this section we provide three case studies to illustrate basic properties of the method. ε-SVR is performed using SVM toolbox from [30]. Section 4 is dedicated to the simplification quality estimation, while in Section 5 we give concluding remarks and future work guidelines.

2 ε-INSENSITIVE SUPPORT VECTOR REGRESSION

SVR is a part of a wider concept of machine learning - SVM [31, 32]. SVM represents supervised machine learning technique that is primarily developed for classification. The main goal of ε-SVR is to find regression function $f(\boldsymbol{x})$ based on the given training data $\{(\boldsymbol{x}_1, y_1), \dots, (\boldsymbol{x}_l, y_l)\}$. Distance between the regression function $f(\boldsymbol{x})$ and all y_i, $i = 1, 2, \dots, l$ has to be lower than (or equal to) pre-determined value ε, i.e. all training data values y_i should be within ε tube formed around function $f(\boldsymbol{x})$.This means that regression error can be neglected if the function deviation from each y_i is less than ε, but deviations that are higher than ε are not allowed. ε-SVR presents regression function using a subset of training points referred to as support vectors (as will be shown in the sequel). One of the main goals of the ε-SVR is to generate approximation function of low complexity which is at the same time characterized by appropriate quality of approximation; in such way generalization properties of the model are improved [24, 31]. Thus, the objective of ε-SVR is to reduce the complexity of the function $f(\boldsymbol{x})$, i.e. to make it as flat as possible.

For linear regression, function $f(\boldsymbol{x})$ has the following form:

$$f(x) = \langle \boldsymbol{w}, \boldsymbol{x} \rangle + b, \tag{1}$$

and it will be flat for small $\boldsymbol{w}$. Minimization of the norm $1/2\|\boldsymbol{w}\|^2$ subject to the condition that all deviations should be lower than or equal to ε will ensure function flatness. This can be presented as a convex optimization problem:

$$\begin{aligned} &\text{minimize} && \frac{1}{2}\|\boldsymbol{w}\|^2 \\ &\text{subject to} && \left| y_i - \langle \boldsymbol{w}, \boldsymbol{x}_i \rangle - b \right| \le \varepsilon. \end{aligned} \tag{2}$$

However, in practice, situations in which all data y_i, lie within ε tube around function $f(\boldsymbol{x})$ are very rarely met, and optimization problem (2) cannot be solved. In order to find function $f(\boldsymbol{x})$ anyway, violation of the condition from problem (2) has to be allowed. For this reason, slack variables ξ_i and ξ_i^*(ξ_i, $\xi_i^* \geq 0$) are introduced and optimization problem (2) is reformulated [31]:

$$\begin{aligned}&\text{minimize} && \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{l}\left(\xi_i + \xi_i^*\right)\\ &\text{subject to} && \begin{cases} y_i - \langle \boldsymbol{w}, \boldsymbol{x_i}\rangle - b \leq \varepsilon + \xi_i \\ \langle \boldsymbol{w}, \boldsymbol{x_i}\rangle + b - y_i \leq \varepsilon + \xi_i^* \end{cases}.\end{aligned} \quad (3)$$

Slack variables allow mobility of borders during given optimization problem solving. The number of points outside ε tube are controlled by parameter $C > 0$, which determines the trade-off between the flatness of $f(\boldsymbol{x})$ and the amount of deviations larger than ε. Solution of optimization problem (3) will lead to regression function which ensures that the most of data points are placed inside ε tube, and a small number of points are outside that area. The vectors (points) outside ε tube are called support vectors. Number of support vectors can be adjusted using trade-off parameter C.

Optimization problem (3) can be represented in dual form [24]:

$$\begin{aligned}&\text{maximize} && \begin{cases} -\frac{1}{2}\sum_{i,j=1}^{l}\left(\alpha_i - \alpha_i^*\right)\left(\alpha_j - \alpha_j^*\right)\langle \boldsymbol{x_i}, \boldsymbol{x_j}\rangle \\ -\varepsilon\sum_{i=1}^{l}\left(\alpha_i - \alpha_i^*\right) + \sum_{i=1}^{l} y_i\left(\alpha_i - \alpha_i^*\right) \end{cases},\\ &\text{subject to} && \sum_{i=1}^{l}\left(\alpha_i - \alpha_i^*\right) = 0 \text{ and } \alpha_i, \alpha_i^* \in [0, C]\end{aligned} \quad (4)$$

where α_i, α_i^* represent Lagrange multipliers. For support vectors ($|f(\boldsymbol{x_i})-y_i| \geq \varepsilon$) Lagrange multipliers are nonzero, while for vectors (points) inside ε-insensitive tube multipliers vanish. In addition, $\boldsymbol{w}$ can be represented as a combination of given training data $\boldsymbol{x_i}$, $i = 1, 2, \ldots, l$ [24]:

$$\boldsymbol{w} = \sum_{i=1}^{l}\left(\alpha_i - \alpha_i^*\right)\cdot\boldsymbol{x_i}. \quad (5)$$

Regression function defined by relation (1) is linear. However, in practice, nonlinear regression problems are more frequently met. Presented methodology can be employed for nonlinear regression by mapping the data from original feature space into high dimensional space in which linear regression can be utilized. From relation (4) it can be observed that training data $\boldsymbol{x_i}$ are only present within inner product $\langle \boldsymbol{x_i}, \boldsymbol{x_j}\rangle$. Consequently, it is not necessary [31] to define high dimensional feature space explicitly – it is enough to know the inner product in this space introduced by using kernel function $K(\boldsymbol{x_i}, \boldsymbol{x_j})$. Therefore, in optimization problem from (4) $\langle \boldsymbol{x_i}, \boldsymbol{x_j}\rangle$ can be replaced by $K(\boldsymbol{x_i}, \boldsymbol{x_j})$ and regression function can be defined as:

$$f(x) = \sum_{i=1}^{l}\left(\alpha_i - \alpha_i^*\right)K\left(\boldsymbol{x_i}, \boldsymbol{x}\right) + b. \quad (6)$$

Kernel can be any function that satisfies the condition of Mercer's theorem, i.e. $K(u, v)$ represents an inner product in some feature space if:

$$\iint K(u, v)\, g(u)\, g(v)\, \mathrm{d}u\mathrm{d}v > 0, \quad (7)$$

is valid for all $g \neq 0$ for which: $\iint g^2(u)\,\mathrm{d}u < \infty$ [31].

There are many different kernel functions: polynomial kernels, Gaussian kernel, sigmoid kernel, *spline* kernel, *b-spline* kernel etc. It should be noted that new kernel can be defined by summing or multiplication of simpler kernels. The choice of kernel depends on training data and desired regression results. Since freeform surfaces, as most complex geometric primitives in 3D objects, are generated using splines, for the point cloud simplification process presented in this paper, *spline* and *b-spline* kernels are of the most importance. In this research we considered *spline* and *b-spline* kernel of order 3. Namely, in computer graphics this order of spline is employed as a rule, since splines of lower order cannot adequately represent high curvature areas, and splines of higher order are prone to oscillations at flat areas.

3 POINT CLOUD SIMPLIFICATION BASED ON ε-*SVR*

The method for point cloud simplification that we present in this paper is based on ε-SVR of scanned contours, and it is applicable to point clouds that are structured into planar scan lines. It exploits the flatness property of ε-SVR. ε-SVR regression line will be as flat as possible and it will approximate well all flat areas of the scanned line. On the other hand, high curvature areas will not be well approximated and points from these areas will be recognized as support vectors (Fig. 1).

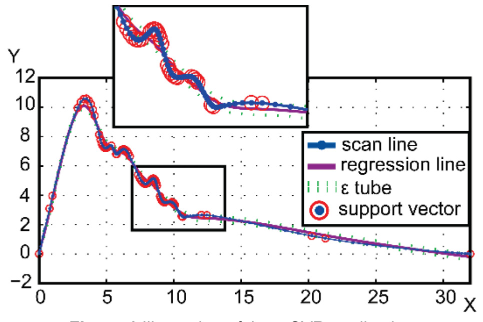


Figure 1 Illustration of the ε-SVR application

Keeping the support vectors, i.e., points from high curvature areas, and only some of the points (e.g. uniformly sampled) from flat areas will lead to feature sensitive simplification of scan line. Consequently, the simplified point cloud will retain all the points that carry most information about object's shape, while the redundant points with respect to information contents will be excluded from point cloud.

The number of support vectors, i.e. the number of retained points from high curvature areas, can be controlled by ε-SVR parameters ε and C. The higher is the value of ε, the ε-tube will be wider, and the number of support vectors will be smaller. In addition, the higher is the error cost parameter C, the number of support vectors will be lower. Another ε-SVR parameter that has to be set is kernel function, in this case *spline* or *b-spline*.

To illustrate the influence of ε-SVR parameters (ε, C, and *kernel*) on the performances of proposed method and to present method's properties, we will use three case studies. First case study refers to a part of Stanford dragon [33] that contains repetitive high curvature areas with relatively small flat regions. Second case study considers turbine blade that has a large flat segment, and small high curvature areas. Objects from case studies 1 and 2 do not contain sharp edges. In the third case study, we present the performances of the method when sharp edges are present in the object.

3.1 Case Study 1: Stanford Dragon

First case study refers to point cloud containing high curvature surface - a part of Stanford dragon [33] that is marked in Fig. 2. CAD model of the dragon in STL (Stereolithography) file format is obtained using a high-resolution 3D scanner and subsequent standard point cloud processing (polygonal meshing).

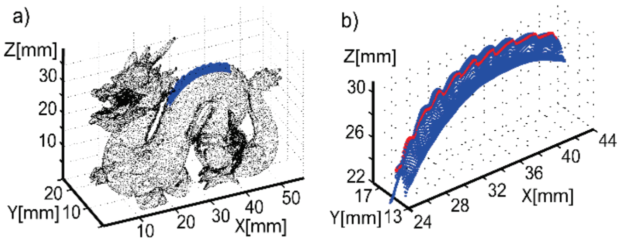


Figure 2 Stanford dragon – a) point cloud; b) structured point cloud of interest

The vertices of the polygons in the STL mesh represent the points of the unstructured point cloud (Fig. 2a). For point cloud structuring we have employed modified z-buffer algorithm [34]. Using this algorithm, points of contours (scanned lines) within structured point cloud are obtained as intersection points between mesh triangles and observation vectors that pass through uniformly distributed points of image raster. In this case study observation vector is set to $-\boldsymbol{k}(0, 0, -1)$ and image raster is a set of equidistant rows with 0,1mm increment.

Fig. 2b shows structured point cloud separated from dragon's back. For the clarity of presentation we will first illustrate the effect of ε-SVR parameters (ε, C, and kernel) using one scanned contour that is marked red in Fig. 2b. Afterwards, the described method will be applied to the whole structured point cloud from Fig. 2b.

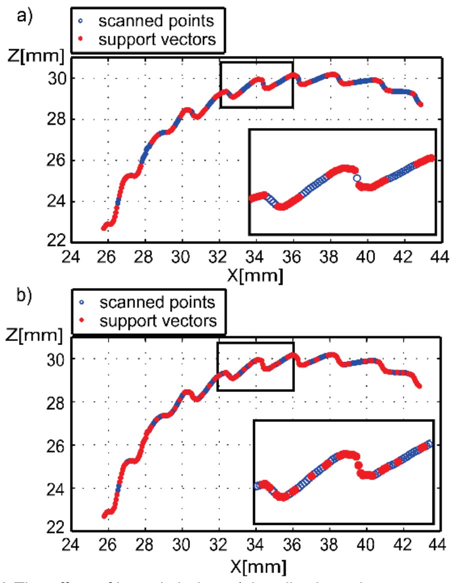


Figure 3 The effect of kernel choice: a) *b-spline* kernel: ε = 0,15, C = 10, reduction ratio 52,8%; b) *spline* kernel: ε = 0,01, C = 10, reduction ratio 49,0%

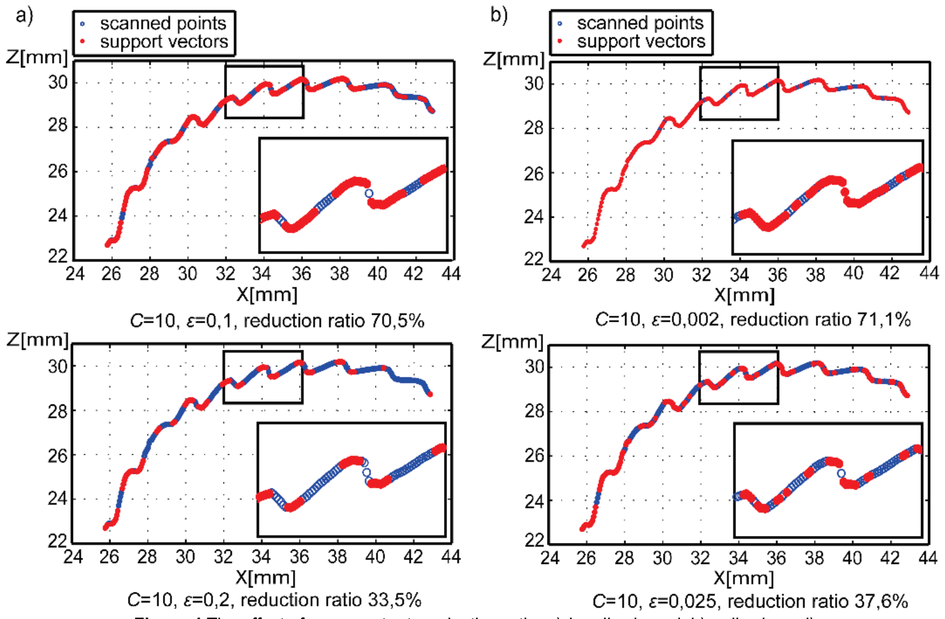


Figure 4 The effect of ε parameter to reduction ratio: a) *b-spline* kernel; b) *spline* kernel)

First, we will illustrate the effects of utilized kernel (*spline* and *b-spline* kernel) on the method performances. During analysis of the effect of kernel, error cost parameter C was set to 10 for both kernels, while for *b-spline* kernel ε was set to 0,15, and for *spline* kernel to 0,01. For these values of ε and C, the percent of reduced (retained) points was similar for both kernels: 52,8% for *b-spline*, and 49,0% for *spline* kernel. Such a close reduction rate is convenient for comparison of the kernel influence. Graphical representation of regression results for *b-spline* kernel is presented in Fig. 3a, and for *spline* kernel in Fig. 3b. From these Figures it can be observed that in both cases the method was able to recognize the points in high curvature areas. However, for *spline* kernel, the method retained some of the points from low curvature area, as well.

The influence of ε parameter on the method performances for considered kernels is illustrated in Fig. 4. If parameter ε has small value, ε-SVR will not be able to fit input data well, and regression function will hardly follow scanned line in predefined tight borders $[-\varepsilon, \varepsilon]$. Accordingly, the influence of error margin ε on reduction rate can be stated as follows. The smaller the parameter ε, the number of support vectors will be higher, and vice versa. Besides, for similar reduction ratio, *b-spline* kernel needs for an order of magnitude larger value of ε than *spline* kernel.

The third parameter that influences the performances of the method is error cost C. This SVR parameter, as already mentioned, represents the trade-off between function flatness and deviation from input data. Low value of parameter C will lead to flat regression function, and the number of detected support vectors will be high, especially in high curvature areas, and spots of sharp geometry changes. On the other hand, the number of support vectors will be low for high C, but regression function will not be flat enough. It should be noted that it is possible to set C to infinite value which means that regression process will find function with deviation from input data set that is as low as possible. The effect of parameter C on the performances of the method is presented in Fig. 5.

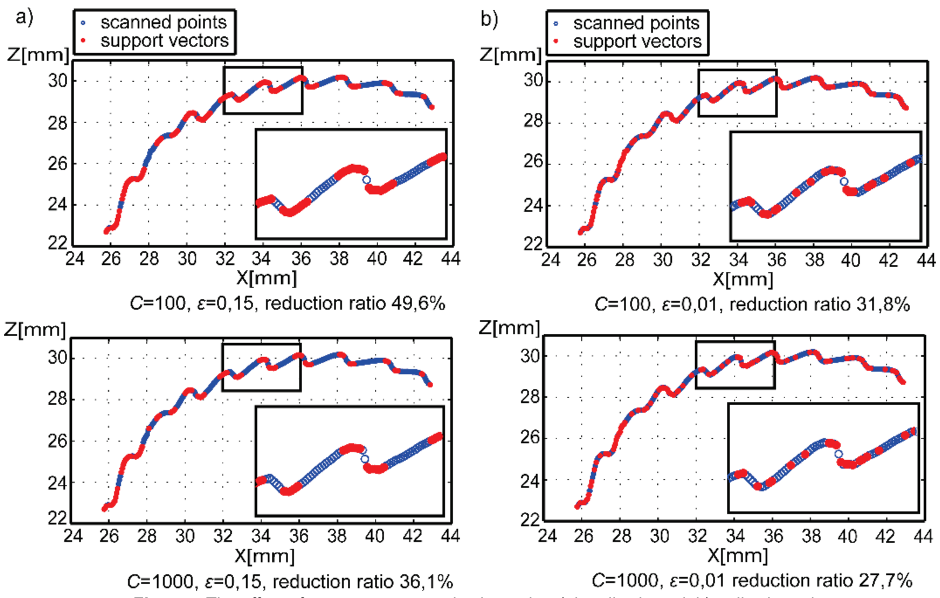


Figure 5 The effect of C parameter to reduction ratio: a) *b-spline* kernel; b) *spline* kernel

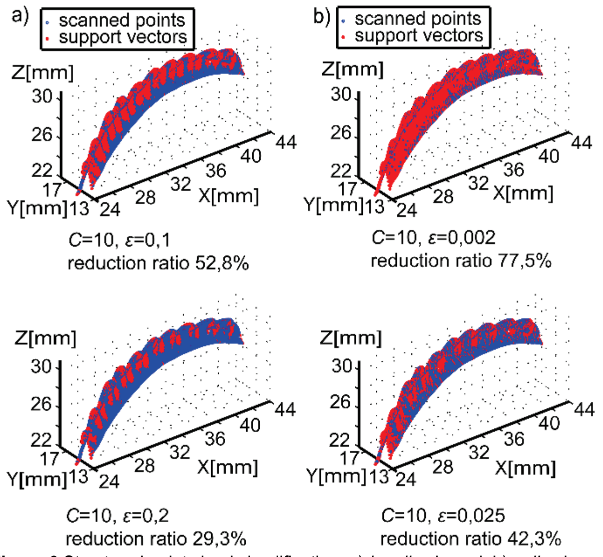
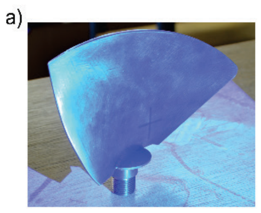


Figure 6 Structured point cloud simplification: a) *b-spline* kernel; b) *spline* kernel

Fig. 6 presents the results of the application of the proposed method to the whole structured point cloud from Fig. 2b. The procedure was applied to scan lines along X axis.

3.2 Case Study 2: Turbine Blade

The second case study refers to the reduction of real world point cloud that is obtained by turbine blade scanning (Fig. 7a). This part contains a large flat segment as well as high curvature areas, and it is convenient for illustration of the method performance on low curvature free form surfaces. Scanning is carried out using ATOS Compact Scan by GOM mbh [35]. Scanner provides unstructured point cloud at output, and in order to apply the proposed method, we have carried out point cloud structuring using the same procedure as in the first case study. The obtained structured point cloud contains 5210 points (3 mm raster along X and 1 mm along Y axis), and it is presented in Fig. 7b.



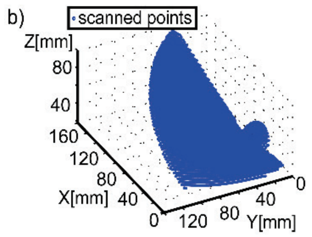


Figure 7 a) turbine blade; b) structured turbine blade point cloud

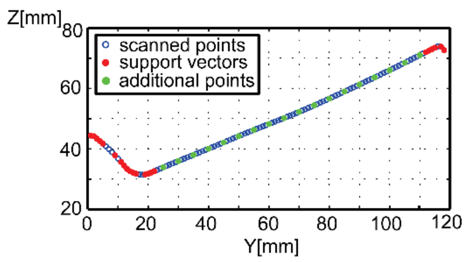


Figure 8 ε-SVR of single contour from Fig. 7: $-\varepsilon = 0{,}01$, $C = 1000$, *spline* kernel - support vectors and additional points $x_{\max} = 5$ mm

Fig. 8 presents the results of implementation of ε-SVR with ε=0,01, C=1000, and *spline* kernel to one contour of the blade. As expected, no support vectors were identified in flat area. On the other hand, method extracted a large number of support vectors in high curvature areas thus providing high degree of contour shape preservation.

Nevertheless, removing all the points from flat segments is not appropriate, especially if subsequent object reconstruction is needed. Lower value of ε or higher value of C would lead to increase of the number of support vectors in flat areas, but simultaneously it would increase the number of support vectors in high curvature areas, thus deteriorating the reduction performances of the method. For these reasons, we have opted to use alternative approach to increase the number of points in flat areas; the approach is based on uniform distance up-sampling. Namely, if the distance along abscissa between two adjacent support vectors is lower than predefined value - $x_{\max}$, the algorithm will retain the points with approximately $x_{\max}$ distance between these support vectors. In this way we ensure the regularity of points' arrangement and adequate quality of flat areas information preservation. During points up-sampling for contour from Fig. 8 and $x_{\max}$=5 mm the percentage of reduced points increased from 18,64% to 33,90%.

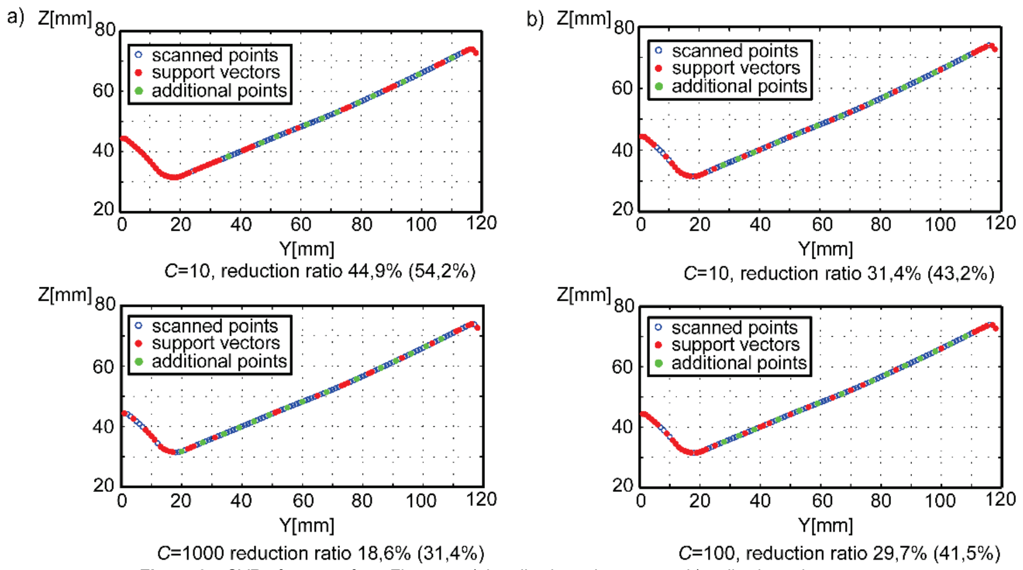


Figure 9 ε-SVR of contour from Figure 7: a) *b-spline* kernel, $\varepsilon = 0{,}15$; b) *spline* kernel, $\varepsilon = 0{,}01$

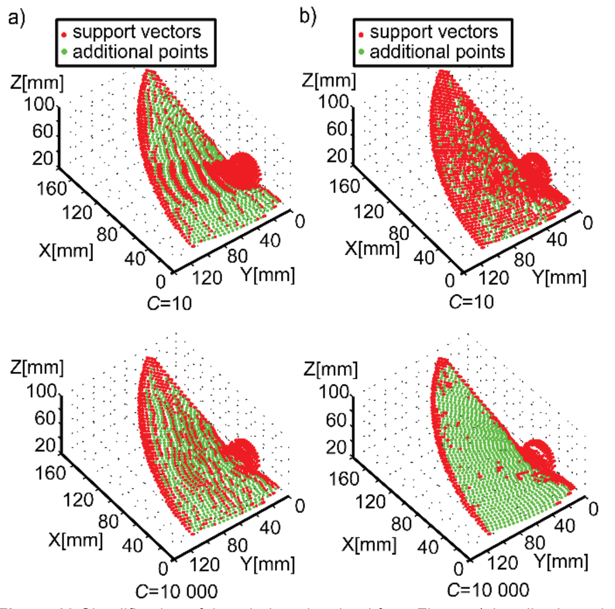


Figure 10 Simplification of the whole point cloud from Fig. 7: a) *b-spline* kernel, $\varepsilon = 0{,}05$; b) *spline* kernel, $\varepsilon = 0{,}002$

Fig. 9 presents additional examples of single contour reduction results for different values of ε-SVR parameters. From this figure it can be observed that for low values of parameter C (for both kernels) the periodic groups of support vectors emerge in the flat area. This effect is higher for *b-spline* kernel, and for its mitigation the appropriate choice of parameters C and ε is crucial.

The results of the application of the proposed procedure to the whole point cloud are presented in Fig. 10, as well as in Tab. 1. From Fig. 10 it can be observed that *spline* kernel is less prone to detection of support vectors in the flat areas in contours that simultaneously contain low and high curvature segments.

Table 1 Simplification of real structured point cloud

Kernel	ε	C	Support vectors (%)	Total preserved points (%)
b-spline	0,05	10	29,0	41,2
b-spline	0,05	10000	22,8	33,1
spline	0,002	10	41,6	46,7
spline	0,002	10000	16,4	32,2

3.3 Case Study 3: An Object with Sharp Edges

Case study 3 illustrates the performances of the presented procedure in the presence of sharp edges, and it considers a real-world object from Fig. 11a. This is a benchmark test part that is used in a number of research studies in reverse engineering [17, 36, 37]. Here, we consider a single-view scan data that is marked in Fig. 11b. The part was scanned using the same device as in the case study 2 (ATOS Compact Scan by GOM mbh), and the same point cloud structuring procedure was applied. Structured point cloud raster along X (scan lines) and Y directions was 0,5 mm (Fig. 11c).

Fig. 12a presents the results of application of SVR based simplification procedure on one scan line that is marked in Fig. 11c, while simplification of the entire view is presented in Fig. 12b and 12c. We employed the following parameters: *spline* kernel, $C = 100$, $\varepsilon = 0{,}05$, and $x_{\max} = 5$ mm (10 samples). Obtained overall reduction ratio was 29,02%.

From Fig. 12, it can be observed that the proposed method was able to adequately recognize the areas in the vicinity of sharp edges. The frequency of preserved points in these areas is significantly higher than in the planar or slightly curved areas. This holds for both sides of edge. In addition, the frequency of detected support vectors in planar regions is smaller in comparison to curved regions.

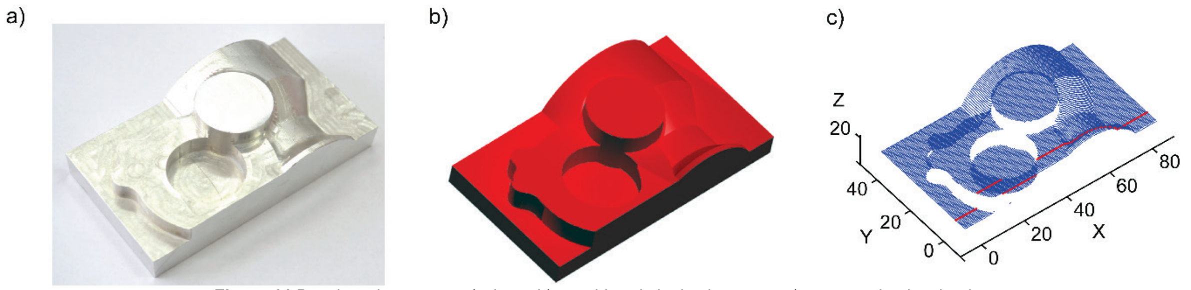


Figure 11 Benchmark test part: a) photo; b) considered single view scan; c) structured point cloud

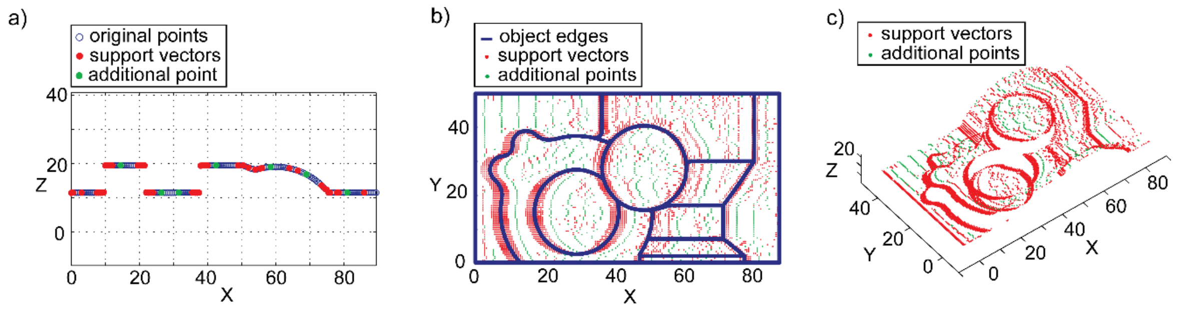


Figure 12 Results of application of proposed simplification method: a) single contour; b) whole point cloud – top view compared to object's edges; c) whole point cloud – isometric view

4 SIMPLIFICATION PROCESS QUALITY ESTIMATION

After presenting the main properties of the proposed method, in this section we estimate the quality of the point cloud simplification. The simplification can be observed as of high quality, if the high amount of information about geometrical properties of scanned objects is preserved after point cloud simplification. Since the method is based on contour by contour simplification, the quality estimation will be carried out using contours, as well. The method for simplification quality estimation that we propose in this paper employs NURBS (Non-uniform rational b-spline) curves. We create two NURBS: (1) for input contour (contour before simplification) – NURBS_1, and (2) for simplified contour – NURBS_2. The quality of simplification process is estimated by comparing NURBS_1 and NURBS_2. High matching level implies that high level of contour's shape information is preserved after simplification process and that lower number of points contains similar geometry information about scanned object.

NURBS_1 and NURBS_2 are compared using the following procedure. The points on NURBS_1 and NURBS_2 are calculated for equidistant values r_j, $j = 1, \ldots, m$ (equidistant raster) and two series $\text{NURBS}_1(r_j)$ and $\text{NURBS}_2(r_j)$ are obtained. Raster between r_j can be defined optionally, but it is convenient to use the same raster as in the initial (scanned) contour.

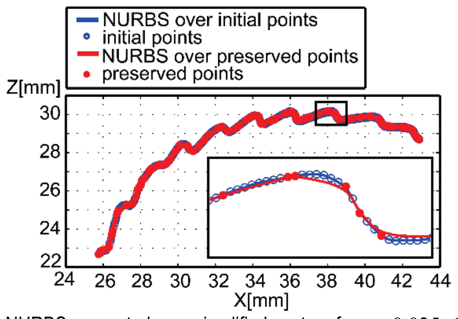


Figure 13 NURBS generated over simplified contour for $\varepsilon = 0{,}025$, $C = 1000$, *spline* kernel along with NURBS generated over initial scanned contour from Fig. 3

For estimation of simplification quality, we use the discrepancy between $\text{NURBS}_1(r_j)$ and $\text{NURBS}_2(r_j)$ and we employ three metrics: (i) maximum absolute deviation dev_m, (ii) total square deviation - dev_t (8), (iii) average absolute deviation – dev_a (9):

$$dev_t = \sum_{j=1}^{m} \left[NURBS_1\left(r_j\right) - NURBS_2\left(r_j\right) \right]^2, \tag{8}$$

$$dev_a = \frac{\sum_{j=1}^{m} \left| NURBS_1\left(r_j\right) - NURBS_2\left(r_j\right) \right|}{m}. \tag{9}$$

During NURBS generation, a set of control points, knot vector, and a set of weights for each control point have to be defined. For NURBS control points we have chosen points from contour – scanned points for $NURBS_1$ and simplified points for $NURBS_2$. Knot vector contains n+3 elements where n is the number of control points; the values of elements are defined in equidistant increasing order in interval [0,1], except for the first three elements that are set to 0, and the last three that are set to 1 to ensure that NURBS passes through the first and the last point on the contour. Weights for all control points are set to 1 (each point has the same influence). Figs. 13 and 14 present NURBS curves that are designed using initial scanned contours from Fig. 3 and 9, respectively.

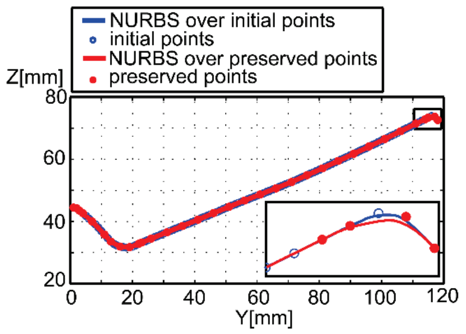


Figure 14 NURBS generated over simplified contour for ε = 0,05, C = 10000, *spline* kernel along with NURBS generated over initial scanned contour from Fig. 9

Table 2 Quality estimation contour from Fig. 3

Kernel	ε	C	SVs (%)	Total preserved points (%)	Maximum deviation (mm)		Total square deviation (mm)		Average deviation (mm)	
					All points	SVs only	All points	SVs only	All points	SVs only
b-spline	0,1	10	70,6	71,4	0,0613	0,0608	0,0217	0,0227	0,0028	0,0032
b-spline	0,15	10	52,8	55,7	0,1190	0,1901	0,0891	0,2527	0,0065	0,0113
b-spline	0,15	100	49,6	52,8	0,1256	0,2076	0,1043	0,3060	0,0073	0,0129
b-spline	0,15	1000	36,2	40,2	0,1061	0,1061	0,1389	0,2599	0,0105	0,0179
b-spline	0,2	10	33,5	38,2	0,1777	0,3185	0,3960	1,9488	0,0165	0,0391
b-spline	0,15	10000	18,1	23,3	0,1492	0,3040	0,3023	1,7207	0,0185	0,0448
spline	0,0020	10	71,1	71,7	0,0154	0,0154	0,0012	0,0013	0,0007	0,0008
spline	0,0100	10	49,0	49,9	0,0328	0,0327	0,0161	0,0185	0,0038	0,0043
spline	0,0250	10	37,6	40,2	0,0630	0,0901	0,0382	0,1069	0,0062	0,0096
spline	0,0100	100	31,8	32,9	0,0549	0,0550	0,0571	0,0631	0,0086	0,0093
spline	0,0100	1000	27,7	29,4	0,0751	0,0751	0,0894	0,1256	0,0107	0,0131
spline	0,0250	1000	18,9	22,2	0,1040	0,1040	0,2158	0,3105	0,0173	0,0222

Table 3 Quality estimation for contour from Fig. 9

Kernel	ε	C	SVs (%)	Total preserved points (%)	Maximum deviation (mm)		Total square deviation (mm)		Average deviation (mm)	
					All points	SVs only	All points	SVs only	All points	SVs only
b-spline	0,05	10	60,2	66,9	0,0144	0,0740	0,0017	0,0496	0,0024	0,0020
b-spline	0,05	100	48,3	55,9	0,0161	0,1270	0,0022	0,1669	0,0025	0,0026
b-spline	0,05	1000	44,1	53,4	0,0542	0,0540	0,0046	0,0148	0,0038	0,0027
b-spline	0,05	10000	28,8	41,5	0,1555	0,4325	0,0694	0,6634	0,0184	0,0085
spline	0,002	10	50,8	52,5	0,2165	0,2164	0,0699	0,0701	0,0057	0,0060
spline	0,002	100	47,4	50,0	0,2165	0,2164	0,0744	0,0756	0,0064	0,0073
spline	0,002	1000	38,9	47,5	0,2164	0,2164	0,0759	0,4366	0,0073	0,0355
spline	0,002	10000	22,0	35,6	0,2164	0,7859	0,0895	18,135	0,0112	0,2575
spline	0,05	10	18,6	32,2	0,2164	0,2497	0,2801	1,8849	0,0204	0,0899
spline	0,05	100	15,2	29,7	0,2164	0,3761	0,2994	4,7092	0,0223	0,1477
spline	0,05	1000	11,9	28,0	0,2164	1,4806	0,2998	79,505	0,0228	0,6490
spline	0,05	10000	11,9	28,0	0,2164	1,4806	0,2998	79,505	0,0228	0,6490

The results of simplification process for contour from Fig. 3 for different values of SVR parameters are presented in Tab. 2. During simplification, minimum distance along abscissa between two adjacent preserved points was set to 10 samples (0,5 mm), thus increasing the total number of preserved points with respect to number of support vectors. From Tab. 2 it can be observed that all deviation measures (dev_m, dev_t, and dev_a) have higher values for *b-spline* kernel than for *spline* kernel for similar reduction ratios.

NURBS generated over points with the highest reduction ratio from Tab. 2, along with NURBS generated over initial contour is presented in Fig. 13.

The results of SVR based simplification for contour from Fig. 9 (case study 2) are given in Tab. 3. In this case study, according to all three deviation measures, SVR based simplification using *b-spline* kernel has shown lower discrepancy from original signal than simplification using *spline* kernel. Fig. 14 presents NURBS generated over points with highest reduction ratio from Tab. 3, along with NURBS generated over initial contour.

5 CONCLUSION

In this paper we have presented a method for feature sensitive simplification of structured point clouds that is based on ε-SVR of scanned lines. During ε-SVR, the points in high curvature areas are recognized as support vectors. Inherently, these points carry the most information about object's high curvature features and they are retained in reduced point cloud. To deal with the possible over-simplification of point cloud in flat areas, we have proposed the additional low-resolution uniform distance up-sampling. This kind of up-sampling is in accordance with the employed ε-SVR kernels (*spline* and *b-spline*) and the nature of free form curves/surfaces that are, as a rule, represented by splines.

The performances of the presented method are dependent on the choice of ε-SVR parameters (ε, C, and kernel). ε and C have high influence on the reduction, and higher values of ε and C will lead to higher reduction ratio. Due to the nature of these parameters, it is convenient to set ε depending on the parts tolerance, and to utilize C for the reduction ratio tuning. Besides ε and C, the choice of ε-SVR kernel has significant influence on the simplification results. It can be observed that, in large flat areas that are combined with high curvature areas, *b-spline* kernel is prone to oscillation, as illustrated in case study 2. In addition, *b-spline* kernel shows lower performances with respect to information contents quality in the case of high curvature surfaces as illustrated in case study 1.

Very important characteristic of the presented method is its ability to recognize and keep in reduced point cloud the points in the vicinity of sharp edges. This property is illustrated in case study 3. Opposite to previously reported methods, the proposed method does not require recognition of edges prior to implementation.

The proposed method is applicable for simplification of point clouds which are structured into planar scan lines. To utilize the method for unstructured point clouds, we have employed modified z-buffer algorithm for point cloud structuring. Our future work will address the extension of the proposed algorithm to direct simplification of unstructured point clouds. In addition, we will direct our research efforts to the downstream processes in 3D point cloud analysis, in particular in the area of automatic recognition of geometric primitives.

Acknowledgements

This research was partially supported by Serbian Ministry of Education, Science and Technological Development, under research grants TR35004 and TR35020. The authors express the gratitude to the company Topomatika Zagreb for scanning service.

6 REFERENCES

[1] Mandić, V. & Ćosic, P. (2011). Integrated Product and Process Development in Collaborative Virtual Engineering Environment. *Tehnički Vjesnik, 18*(3), 369-378.

[2] Fang, Z., Xu, D., & Tan, M. (2011). A Vision-Based Self-Tuning Fuzzy Controller for Fillet Weld Seam Tracking. *IEEE/ASME Transactions on Mechatronics*, *16*(3), 540-550. https://doi.org/10.1109/TMECH.2010.2045766

[3] Fu, G., Corradi, P., Menciassi, A., & Dario, P. (2011). An Integrated Triangulation Laser Scanner for Obstacle Detection of Miniature Mobile Robots in Indoor Environment. *IEEE/ASME Transactions on Mechatronics*, *16*(4), 778-783. https://doi.org/10.1109/TMECH.2010.2084582

[4] Groš, J., Medić, S., & Šimunić, N. (2015). Quality control of a shaft using the Steinbichler T-Scan laser scanning. *Tehnički Glasnik, 9*(3), 273-278.

[5] Savio, E., De Chiffre, L., & Schmitt, R. (2007). Metrology of freeform shaped parts. *Annals of the CIRP*, *56*(2), 810-834. https://doi.org/10.1016/j.cirp.2007.10.008

[6] Song, H. & Feng, H. Y. (2009). A progressive point cloud simplification algorithm with preserved sharp edge data. *International Journal of Advanced Manufacturing Technology*, *45*(5-6), 583-592. https://doi.org/10.1007/s00170-009-1980-4

[7] Pauly, M., Gross, M., & Kobbelt, L.P. (2002). Efficient simplification of point-sampled surfaces. *Proceedings of the IEEE Visualization Conference* (pp. 163-170). Boston: IEEE.

[8] Cignoni, P., Montani, C., & Scopigno, R. (1998). A comparison of mesh simplification algorithms. *Computers & Graphics*, *22*(2), 37-54. https://doi.org/10.1016/S0097-8493(97)00082-4

[9] Lee, K.H., Woo, H., & Suk, T. (2001). Point data reduction using 3D grids. *International Journal of Advanced Manufacturing Technology, 18*(3), 201-210. https://doi.org/10.1007/s001700170075

[10] Shi, B. Q., Liang, J., & Liu, Q. (2011). Adaptive simplification of point cloud using k-means clustering. *Computer Aided Design*, *43*(8), 910-922. https://doi.org/10.1016/j.cad.2011.04.001

[11] Miao, Y., Pajarola, R., & Feng, J. (2009). Curvature-aware adaptive re-sampling for point-sampled geometry. *Computer Aided Design*, *41*(6), 395-403. https://doi.org/10.1016/j.cad.2009.01.006

[12] Cruz-Matias, I. & Ayala, D. (2014). A new lossless orthogonal simplification method for 3D objects based on bounding structures. *Graphical Models, 76*(4)*,* 181-201. https://doi.org/10.1016/j.gmod.2014.01.002

[13] Kang, E.-C., Kim, D.-B., & Lee, K.H. (2008). Balanced feature-sensitive point sampling for 3D model generation. *International Journal of Advanced Manufacturing Technology*, *38*(1-2), 130-142. https://doi.org/10.1007/s00170-007-1073-1

[14] Song, H. & Feng, H.-Y. (2008). A global clustering approach to point cloud simplification with a specified data reduction ratio. *Computer Aided Design*, *40*(3), 281-292. https://doi.org/10.1016/j.cad.2007.10.013

[15] Ma, X. & Cripps, R.J. (2011). Shape preserving data reduction for 3D surface points. *Computer Aided Design*, *43*, 902-909. https://doi.org/10.1016/j.cad.2011.03.006

[16] Budak, I. Soković, M., & Barisić, B. (2011). Accuracy improvement of point data reduction with sampling-based methods by Fuzzy logic-based decision making. *Measurement*, *44*(6), 1188-1200. https://doi.org/10.1016/j.measurement.2011.03.026

[17] Jakovljević, Ž., Puzović, R., & Pajić, M. (2015). Recognition of Planar Segments in Point Cloud based on Wavelet Transform. *IEEE Transactions on Industrial Informatics*, *11*(2), 342-352. https://doi.org/10.1109/TII.2015.2389195

[18] Lee, K. H., Woo, H., & Suk, T. (2001). Data reduction methods for reverse engineering. *International Journal of Advanced Manufacturing Technology*, *17*(10), 735-743. https://doi.org/10.1007/s001700170119

[19] Budak, I., Vukelić, D., Bračun, D., Hodolič, J., & Soković, M. (2012). Pre-processing of point-data from contact and optical 3D digitization sensors. *Sensors, 12*(1), 1100-1126. https://doi.org/10.3390/s120101100

[20] Budak, I., Soković, M., Kopač, J., & Hodolič, J. (2009). Point data pre-processing based on fuzzy logic for reverse engineering modelling. *Strojniski Vestnik/Journal of Mechanical Engineering, 55*(12), 755-765.

[21] Wang, Y.-Q., Tao, Y., Zhang, H.-J., & Sun, S.-S. (2013). A simple point cloud data reduction method based on Akima spline interpolation for digital copying manufacture. *International Journal of Advanced Manufacturing Technology*, *69*(9-12), 2149-2159. https://doi.org/10.1007/s00170-013-5195-3

[22] Tao, Y., Li, Y., Wang, Y.-Q., & Ma, Y.-Y. (2016). On-line point cloud data extraction algorithm for spatial scanning measurement of irregular surface in copying manufacture. *International Journal of Advanced Manufacturing Technology, 87*(5-8), 1891-1905. https://doi.org/10.1007/s00170-016-8592-6

[23] Xu, G., Cheng, X., Wang, L., Tan, K., & Lou, Q. (2014). An adaptive compression algorithm of scattered point cloud based on wavelet technology. *Journal of Information and Computational Science, 11*(6), 1917-1928. https://doi.org/10.12733/jics20103257

[24] Smola, A. J. & Schölkopf, B. (2004). A tutorial on support vector regression. *Statistics and Computing, 14*(3), 199-222. https://doi.org/10.1023/B:STCO.0000035301.49549.88

[25] Jakovljević, Ž. (2012). Point cloud reduction using support vector machines. *Proceedings of 11th International Scientific Conference MMA 2012* (pp. 121-124), Novi Sad.

[26] Yu, Q. & Wang, K. (2013). 3D vision based quality inspection with computational intelligence. *Assembly Automation, 33*(3), 240-246. https://doi.org/10.1108/AA-12-2013-065

[27] Serna, A. & Marcotegui, B. (2014). Detection, segmentation and classification of 3D urban objects using mathematical morphology and supervised learning. *ISPRS Journal of Photogrammetry and Remote Sensing, 93*, 243-255. https://doi.org/10.1016/j.isprsjprs.2014.03.015

[28] Steinke, F., Schölkopf, B., & Blanz, V. (2005). Support Vector Machines for 3D shape processing. *Euroraphics*, *24*(3), 285-294. https://doi.org/10.1111/j.1467-8659.2005.00853.x

[29] Zhang, L., Wang, W., Li, Y., Liu, X., Shi, M., & He, H. (2009). A Method for Surface Reconstruction Based on Support vector machine. *Journal of Computers, 4*(9), 806-812. https://doi.org/10.4304/jcp.4.9.806-812

[30] Gunn, S. (1998). *Support Vector Machines for Classification and Regression – Matlab toolbox*. Retrieved from http://www.isis.ecs.soton.ac.uk/resources/svminfo/

[31] Vapnik, V. (2000). *The Nature of Statistical Learning Theory*. New York, NY: Springer-Verlag. https://doi.org/10.1007/978-1-4757-3264-1

[32] Vapnik, V. (2006). *The Estimation of Dependences Based on Empirical Data, Reprint of 1982 Edition*, New York, NY: Springer-Verlag.

[33] Stanford Computer Graphics Laboratory. The Stanford 3D scanning repository. (2015, May). Retrieved from http://graphics.stanford.edu/data/3Dscanrep/

[34] Catmull, E. (1974). *A Subdivision Algorithm for Computer Display of Curved Surface* (Doctoral Dissertation). Retrieved from http://www.dtic.mil/dtic/tr/fulltext/u2/a004968.pdf

[35] (2014). GOM GmbH, ATOS Compact Scan - The compact class of scanning. http://www.gom.com/metrology-systems/system-overview/atos-compact-scan.html

[36] Sacchi, R., Poliakoff, J. F., Thomas, P. D., & Hafele, K.–H. (2000). Improved extraction of planar segments for scanned surfaces. *Proceedings of IEEE International Conference on Computer Visualization and Graphics* (325-330), London: IEEE.

[37] Kjellander, J. A. P. & Rahayem, M. (2009). Planar segmentation of data from a laser profile scanner mounted on an industrial robot. *International Journal of Advanced Manufacturing Technology*, *45*(1-2), 181-190. https://doi.org/10.1007/s00170-009-1951-9

Contact information:

Veljko MARKOVIC, PhD Student
University of Belgrade, Faculty of Mechanical Engineering,
Kraljice Marije 16, 11000 Belgrade, Serbia
markovicveljko@yahoo.com

Zivana JAKOVLJEVIC, Associate Professor
University of Belgrade, Faculty of Mechanical Engineering,
Kraljice Marije 16, 11000 Belgrade, Serbia
zjakovljevic@mas.bg.ac.rs

Zoran MILJKOVIC, Full Professor
University of Belgrade, Faculty of Mechanical Engineering,
Kraljice Marije 16, 11000 Belgrade, Serbia
zmiljkovic@mas.bg.ac.rs