Multimedia Tools and Applications (2021) 80:15093–15134
https://doi.org/10.1007/s11042-020-10120-3

Filling cavities in point clouds representing human body surface using Bezier patches

Marta Nowak[1] · Jakub Michoński[1] · Robert Sitnik[1]

Received: 7 July 2019 / Revised: 30 August 2020 / Accepted: 19 October 2020 /
Published online: 1 February 2021

Abstract

In this paper we introduce a cavity reconstructing algorithm for 3D surface scans (CRASS) developed for filling cavities in point clouds representing human body surfaces. The presented method uses Bezier patches to reconstruct missing data. The source of input data for the algorithm was an 8-directional structured light scanning system for the human body. Typical 3D scan representing human body consists of about 1 million points with average sampling density of 1 mm. The paper describes the complete scan processing pipeline: data pre-processing, boundary selection, cavity extraction and reconstruction, and a post-processing step to smooth and resample resulting geometry. The developed algorithm was tested on simulated and scanned 3D input data. Quality assessment was made based on simulated cavities, reconstructed using presented method and compared to original 3D geometry. Additionally, comparison to the state-of-the-art screened Poisson method is presented. Values' ranges of parameters influencing result of described method were estimated for sample scans and comprehensively discussed. The results of the quantitative assessment of the reconstruction were lower than 0,5 of average sampling density.

Keywords Point cloud · Full body scanning · Parametric surfaces · Bezier patches · Hole filling · Cavity filling

Abbreviations

CRASS	Cavity Reconstructing Algorithm for 3D Surface Scans
AMD	average minimum distance (between points)

✉ Marta Nowak
m.nowak@mchtr.pw.edu.pl

Jakub Michoński
j.michonski@mchtr.pw.edu.pl

Robert Sitnik
r.sitnik@mchtr.pw.edu.pl

[1] Institute of Micromechanics and Photonics, Virtual Reality Techniques Division, Faculty of Mechatronics, Warsaw University of Technology, ul. Św. Andrzeja Boboli 8, 02-525 Warsaw, Poland

1 Introduction

1.1 Motivation

3D scanning is becoming an increasingly important element of reality reconstruction, replacing manual modeling. Especially significant applications of this technology are systems that use models of the human body, e.g., in medical, clothing and gaming industries. It is difficult to model such shapes realistically, not to mention obtaining a metrologically correct representation.

Existing methods of capturing the surface of the human body include volumetric methods used primarily in medicine [19, 36], such as magnetic resonance imaging (MRI) or computer tomography (CT), which allow the extraction of the whole surface from a single measurement. These methods are costly, invasive, and generate excessive data; hence, these methods are rarely used for surface capture. Methods that provide direct information about the surface, such as structured light illumination (SLI) [12, 23], laser triangulation [28], or Structure from Motion (SfM) [9], are used much more frequently. These methods use visible or near-to-visible light and are non-invasive, which has made them more popular in medicine, especially for examining children and pregnant women [27], where exposure to ionizing radiation should be strictly avoided. However, most of these techniques deliver data from a single point of view and are prone to self-occlusion. This makes creating a complete model time consuming, which in the case of in vivo measurements is undesirable and can lead to serious shape deformations. In visualization systems, such as virtual fitting rooms [6, 31], human perception is easily drawn to unnatural discontinuities in the models, and the effect of presenting a realistic object is lost. In such cases, the visual effect is a major factor in determining the usability of a particular technique.

Factors that cause discontinuity in scans of a continuous surface include self-occlusion of the scanned object, object movement during frame capture, or a limited number of directions from which the object is captured [23]. These can all result in cavities of various shapes appearing in the scanned surfaces (Figs. 1, 2, and 3).

The aim of the work presented in this paper was to develop an algorithm for filling cavities in surface 3D scans. The algorithm was developed as part of an 8-directional SLI scanning system

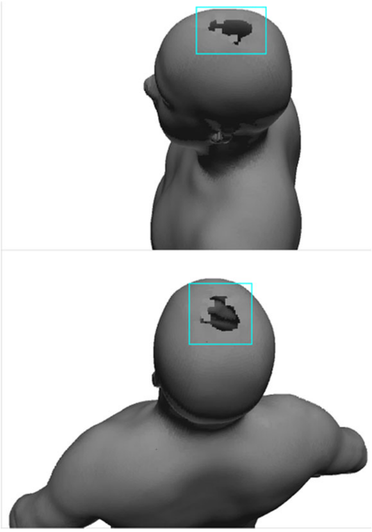

Fig. 1 Sample cavity in scan in area of head

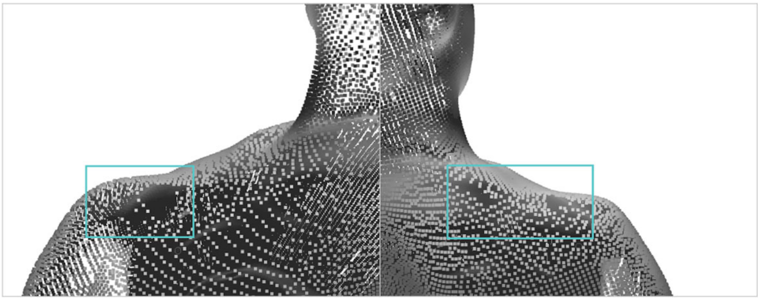

Fig. 2 Sample cavities on arms

designed at the Institute of Micromechanics and Photonics in Warsaw, Poland [26]. The aim of this system is to gather measurements of army officers for the manufacture of personalized uniforms. Although the algorithm is generic, its primary application is filling cavities in point cloud-based representations of the surface of the human body.

The number of measurement directions in the system was chosen to provide a good compromise between the area covered by the measurement volumes and the time taken for measurement. Because of the automated data-processing pipeline in the system, the algorithm for filling cavities in the model has to work directly on point clouds, preferably without any user interaction.

Cavities in SLI scans typically feature noise on the boundaries of the cavities [23]. Displacements on edges are caused by the method's systematic error on surfaces that are parallel to the measurement direction. Similar cavities can also appear due to changes in the local reflection factor [23]. Hence, it is important to be able to reconstruct missing data based on boundary points as well as on points surrounding the cavity boundary.

A cavity-filling algorithm for the introduced scanning system must satisfy the following requirements. It should:

- work on point clouds consisting of XYZ point coordinates and normal vectors,

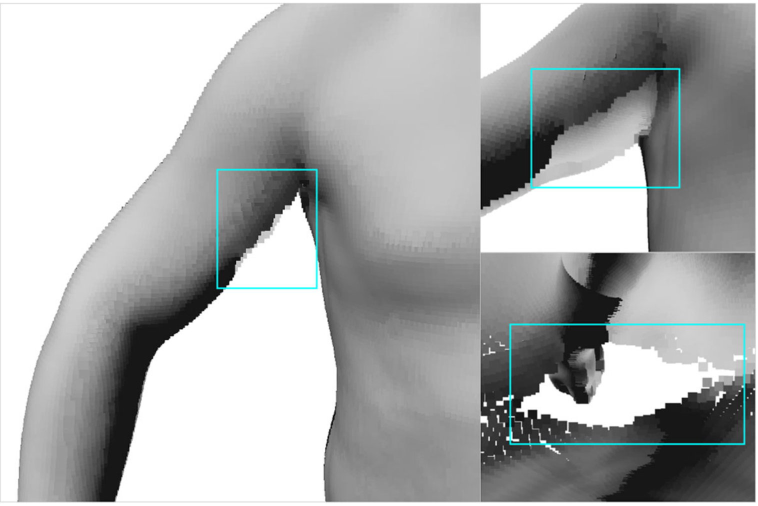

Fig. 3 Sample cavities in scan in area of an axilla

- be robust to substantial errors on bounds of the measurement area,
- reconstruct the shape of the surface including factors of degree two or three,
- not require additional information about the scans except surface data (e.g., measurement system characteristics and position in space),
- be automatic – any parameter can be estimated before running the algorithm, and user interaction for every cavity is not required,
- provide optional smooth transition between the original and reconstructed data (for presentation purposes),
- allow the parameters to be adjusted to the characteristics of the individual cavity,
- work on 2.5D surfaces (for possible future uses in the reconstruction of partial view scans).

2 State of technology review

Some previous attempts have been made in the field of cavity filling in point clouds [14, 29]. Currently, the developed algorithms can be divided into two groups: meshless and mesh-based methods, depending on whether the raw point cloud is used as input or a triangle mesh is created to facilitate analysis. Employing a triangle mesh provides a straightforward definition of a cavity boundary: a list of connected edges that have only one half-edge.

For the mesh-based group, the method described in [32] should be mentioned. It consists of two major steps, referred to as “make convex” and “add vertices”, performed iteratively. The first step creates a triangle by adding an additional edge to any pair of edges from the boundary with angle smaller than a defined parameter φ. This step eliminates sharp turns of the boundary. The second step adds a vertex for every boundary edge in the direction perpendicular to the edge, on the plane calculated using the Moving Least Squares (MSL) procedure. The user parameterizes the distance to the edge. A new triangle is created from each newly created vertex and the edge from which it originates. The two steps are repeated until the cavity is defined by only 3 edges, which form the final triangle. Another interesting approach was described in [18] where a learning set is built based on scanned human body surfaces and a template mesh is aligned and deformed to match a new scan. This is an extended SCAPE method proposed in [1] and performs well with solving the geometry folding problem appearing after template mesh deformation to registered scan. However, in SCAPE instance meshes are resampled to about 50,000 triangles before passing to the algorithm, what indicates meaningful data degradation compared to scans from our system (1 million of points per scan). Mesh deformation approach was also used in [30] for filling holes in scans of various engineering objects.

Meshless methods were proposed by Davis et al. [7] and by Chalmoviansky and Jüttler [3]. The first method [7] is based on defining a signed distance function, which describes the surface of the point cloud. An iterative algorithm is used to extend the function in hollow areas. After completing this step, missing points are generated using the marching cubes (MC) algorithm. The described method works well even for complicated shapes, but it requires information about the orientation of the detectors used during scanning to estimate in which direction the function should extend the shape. Besides, research presented in [14] shows that this method encounters problems filling bigger cavities e.g. in the Stanford bunny model. Some other research [33, 34] proposes the Moving Least Squares technique to solve a height function approximation and fill the cavity in a local coordinate system defined by a best-fit plane. The approach described in [24] bases on the property of some scanning methods that cause the point cloud to be stored as horizontal scan lines sorted according to height. This enabled the authors to approximate discontinuities in scan lines using cubic spline curve approximation. Apart from MLS and spline fitting methods, Poisson reconstruction methods have received some attention [2, 21, 22]. The basic idea behind this group of methods consists in using an input set of oriented points (points with normals) to construct a corresponding isosurface. The isosurface is extracted from an indicator function formulated using Radial Basis Functions and an adaptive octree. Later on, the output mesh is obtained using a triangulation method of choice. Poisson surface reconstruction implementations can be found in open-source software, e.g. MeshLab [4]. Zhou et al. [37] propose a surface reconstruction method incorporating a new visibility model for each line of sight along with introduction of likelihood term in binary labeling problem of Delaunay tetrahedra. The method works well with multi-view stereo dataset but requires information recording from which view every point in the point cloud was seen. Another work [25] describes an iterative algorithm for construction of signed distance field with l0 gradient minimization. The method uses half-quadratic splitting method and consists of alternating smoothing and edge-preserving steps. Authors point out that their algorithm has the advantage over Poisson method described earlier in the fact that it does not require accurate directions of normal vectors to produce correct results. Some recent works try to employ neural networks for the task of generating mesh representation of a 3D scanned point cloud [5, 11, 15–17, 35]. Latest of those, Point2Mesh [17] uses Convolutional Neural Network (CNN) to iteratively mold the initial mesh to input point cloud, shrink-wrapping the mesh until it reaches final shape. During the iterations, initial mesh is upsampled to reconstruct details from input point cloud. However, authors own that their method is more computationally expensive than state-of-the-art methods and is less robust to noise and outliers with irregular geometry. Those disadvantages are less important for point cloud to mesh conversion, but become blocking issue in hole filling, where relatively big cavities with irregular and noisy edge may appear.

The method described in [3] fills cavities using algebraic surface patches that are adapted using distance, and normal and tension terms calculated for the cavity boundary and its neighborhood. Firstly, normal vector estimation is performed. After that, boundary points are extracted using two conditions based on distance and angle, which when combined result in fast processing and correct output. The unorganized set of boundary points that is obtained is divided into cavities using an iterative cavity-polyline growth algorithm. After that, each cavity is filled with an algebraic surface according to an objective function that involves several terms. Calculations are performed in barycentric coordinates, and therefore require the definition of a simplex containing all given points. Depending on the weight factors for each part of the objective function, the method reproduces planes, spheres, or cylinders, which are sufficient for most engineering objects.

The algorithms described above do not ensure that our requirements are fulfilled but they do form a good basis for development. For the method described in [7], some authors highlight problems with artifacts caused by aliasing issues and by constructing unwanted geometry [32]. Moreover, “the system needs to know what portion of the space corresponds to the exterior of the surface and what portion corresponds to its interior, which may not always be readily available” [33]. Spine curve fitting proposed in [24] could be prone to noise in point translation that appears on cavity boundary in SLI scans. The method proposed in [3] constitutes a good basis but the reasons for using barycentric coordinates were not clarified in the paper. The only requirement was to create a simplex that contains all the given points, although choice of the simplex affects the tension terms [3]. Moreover, the algorithm is sensitive to sharp turns of the cavity boundary, which is acceptable in laser scans of engineering objects but not in our case. We regard methods based on Poisson reconstruction as current state-of-the-art for solving the cavity-filling problem in models such as the human body. The methods provide good quality of extrapolated data, and efficient implementations exist. However, because of their global nature it is difficult to approach cavities separately using different strategies, best fitted to the local character of each cavity. They also require the model to be 3D-complete – on 2.5D surfaces, e.g. face scans, large artifacts will be created outside of the defined surface. These properties make it difficult to develop these methods in some of the directions we would like to follow in the future. Nevertheless, our algorithm should achieve similar reconstruction accuracy.

Mesh-based methods are beyond our region of interest, as they require transforming the point cloud into a topologically correct triangular mesh. This operation is time consuming and usually requires human interaction to identify and correct possible artifacts.

In conclusion, some cavity filling methods for point clouds have already been developed but they are either specialized for particular classes of objects or have considerable computational cost and/or low noise resistance. To the best of our knowledge, there is no algorithm intended for filling cavities directly in point clouds that could be used with human body scans (Table 1). However, the algorithm described in [3] provides a good starting point because of its utilization of parametric surfaces. In the CRASS algorithm presented here, we base our method on boundary-point segmentation and cavity extraction methods from [3]. We decided to modify the approximation step to make it work with SLI scans, especially in the case of the human body.

The paper is organized as follows: in Section 3 the reader can find basic definitions and detailed description of CRASS algorithm, Section 4 contains reconstruction quality and performance tests along with comparison to Screened Poisson method. Section 5 includes discussion of results presented in Section 4, estimation of values for hyperparameters used in CRASS and limitations of the method presented. The paper ends with conclusions and future work described in Section 6.

3 Materials and methods

3.1 Basic definitions

All the input data used in this paper is in the form of a point cloud. This is defined as a set of points representing the surface of an object. Each point consists of position (XYZ) and a normal vector (N_x,

Table 1 Comparison of existing algorithms with respect to requirements

Method/feature	Works on point clouds	Robustness to error on measured area bounds	Reconstructing local shape	Automatable	No need for additional information except of model point cloud	Allows the parameters to be adjusted to the cavity characteristics	Works on 2.5D surfaces
Hole filling algorithm for triangular meshes [32]	–	–	+	+ / -	+	+	+
Hole filling using algebraic surfaces [3]	+	+ / -	+	–	+	+	+
Reconstruction using Volumetric Diffusion [7]	+	+ / -	+	+	–	–	+
Screened Poisson Reconstruction [21]	+	+	+	+	+	–	–

N_y, N_z). In the ideal case, a point cloud is complete, i.e., it does not contain any cavities. In most cases, scans are not complete, and then such a point cloud can be divided into groups of two types: the regular part and the hollow part. We define the regular part as a subset of points for which the cloud is locally complete. Points in the regular part will be called regular points. The hollow part contains one or more cavities, which are understood as areas of discontinuity in the scanned surface that feature either zero or small sampling density. We define a cavity as an area limited by a cavity boundary, namely a sequence of boundary points. Boundary points are those points from which the distance to a neighborhood's geometric center is significantly larger than the similar quantity for regular points (see Fig. 4). In order to approximate this value for the entire cloud, we introduce the average minimum distance (AMD). The average minimum distance for a point cloud is calculated as the average distance to the nearest point.

3.2 Prerequisites and case specific discussion

The aim of this section is to specify some geometrical dependencies in ideal input data affected by measurement precision only, which will be a basis for estimation of parameters for presented algorithm in further part of the article.

In SLI scans, the difference between point groups (regular and hollow) is fluid due to sampling density. Space between points in the regular part of the cloud can be treated as a micro-cavity. The factor enabling the distinction between the cavity and the space between points is sampling density (Fig. 4). Closer to the edges, the cloud becomes sparser. We assume that sampling density in the regular part is uniform, i.e., any change in sampling density does not exceed the measurement uncertainty.

We assume that the measurement precision is equal to half the AMD (3.1) in our scanning system.

$$\Delta \mathrm{d} = \frac{\mathrm{d}}{2} \tag{3.1}$$

where:

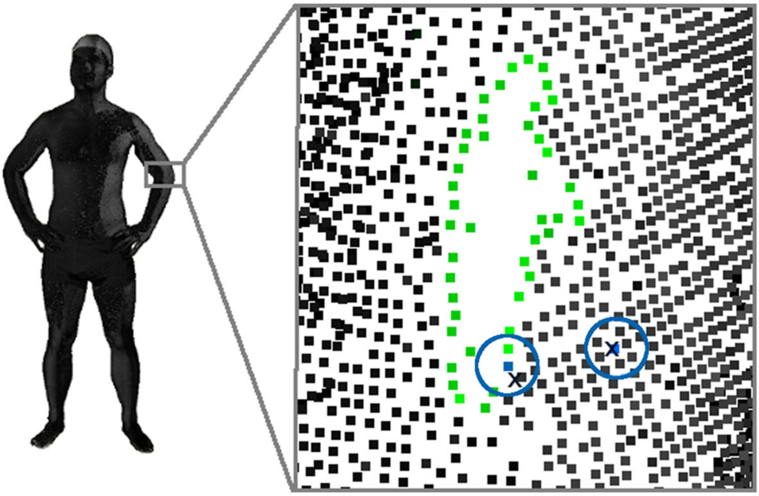

Fig. 4 Distance to center of neighborhood for boundary point (left blue) and regular point (right blue). A cross symbolizes center of a neighborhood, green points are boundary points of a cavity. Blue circles mark neighborhood search area

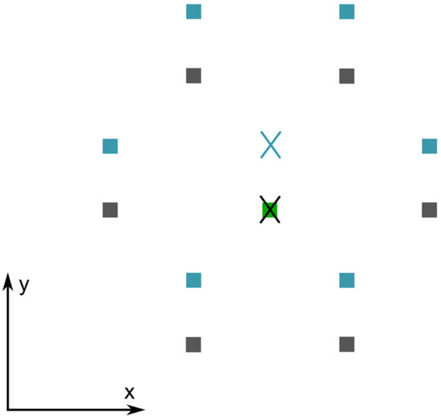

Fig. 5 Sample point (green) surrounded by its regular hexagonal neighborhood (grey). A grey cross corresponds to geometric center of neighborhood. Blue points are neighborhood shifted by maximum flow value towards y-axis

Δd measurement precision,
d average minimum distance (AMD) between points.

Consider a regular point and its neighborhood. As assumed before, the sampling density in the regular part of the scan is nearly uniform and in an ideal case distance between each two points would be equal and the neighborhood points would form a hexagon around the considered point (Fig. 5).

In the ideal case, the coordinates (described as vector v with corresponding subscript) of both the considered point p and the center of its neighborhood (of size 6) is described by the following eq. (3.2):

$$v_{center} = \frac{v_{p0} + v_{p1} + \ \ldots + v_{p5}}{6} = v_p \tag{3.2}$$

where:

v_{center} position of center of neighborhood of point p,
v_p position of point p in ideal case.

Due to measurement errors, points exhibit some variability around their theoretical position. This variability is bounded by the measurement precision, which in the target system equates to half of the AMD (3.1) In an extreme case, all points would be moved towards one and the same axis (3.3), for example the y-axis (see Fig. 5). The resulting hexagon would be shifted towards this axis by the value of the measurement precision (3.4).

$$y'_{pi} = y_{pi} + \Delta d \tag{3.3}$$

$$\mathrm{v}'_{\mathrm{center}} = \frac{\mathrm{v}'_{\mathrm{p0}} + \mathrm{v}'_{\mathrm{p1}} + \ldots + \mathrm{v}'_{\mathrm{p5}}}{6} = \mathrm{v_p} + [0, \Delta \mathrm{d}]^T \tag{3.4}$$

where:

y'_{pi} y cooordinate of i-th neighbor of point p in extreme case scenario,

y_{pi} y coordinate of i-th neighbor of point p in ideal case,
v'_{center} position of center of neighborhood of point p in extreme case scenario,
v'_{pi} position of i-th neighbor of point p in extreme case scenario.

This is due to the fact that the maximum distance between a point and the geometric center of its neighborhood in the regular part will not exceed the measurement precision. Additionally, gross errors must be considered. As a result, a boundary point can be characterized by a shift of its neighborhood center of mass that is larger than the measurement precision.

Similarly, for a corresponding angular entity, two points from the neighborhood and the middle point can span an angle that is greater than in the ideal case (Figs. 6 and 7). Assuming that a regular point is surrounded by 6 neighbors, this angle in the ideal case is equal to (3.5):

$$\alpha_{reg} = \frac{2\pi}{6} = \frac{\pi}{3} \tag{3.5}$$

where:

α_{reg} angle spanned by regular point and two neighbors in regular (ideal) case.

In the previously mentioned extreme case, we obtain (3.6):

With the measurement precision of our SLI scanner (3.1), this leads to (3.7),(3.8),(3.9):

$$tg\left(\frac{\alpha_{regFlow}}{2}\right) = \frac{\frac{d}{2}}{\frac{d\sqrt{3}}{2} - \Delta d} \tag{3.6}$$

$$tg\left(\frac{\alpha_{regFlow}}{2}\right) = \frac{\frac{d}{2}}{\frac{d\sqrt{3}}{2} - \frac{d}{2}} \tag{3.7}$$

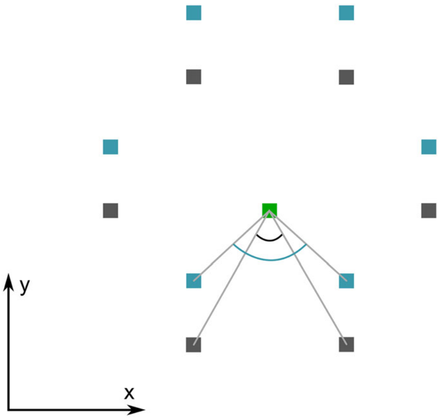

Fig. 6 Sample point (green) surrounded by its regular hexagonal neighborhood (grey). A grey arch corresponds to angle in regular case. Blue arch is an angle spanned by neighborhood shifted by maximum flow value towards y axis

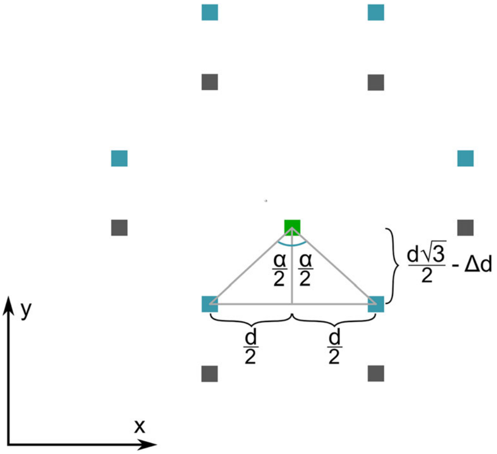
$\frac{\alpha}{2}$ $\frac{\alpha}{2}$
$\frac{d\sqrt{3}}{2}$ - Δd
$\frac{d}{2}$ $\frac{d}{2}$
y
x

Fig. 7 Sample point (green) surrounded by its regular hexagonal neighborhood (grey). Blue arch corresponds to angle spanned by neighborhood shifted by maximum flow value towards y axis

$$\frac{\alpha_{\text{regFlow}}}{2} = \operatorname{atan}\left(\frac{\frac{d}{2}}{\frac{d\sqrt{3}}{2}-\frac{d}{2}}\right)\approx 0.3\pi \tag{3.8}$$

$$\alpha_{\text{regFlow}}\approx 0.6\ \pi \tag{3.9}$$

where:

α_{regFlow} angle spanned by point p and two neighbors in extreme case,
d average minimum distance (AMD) between points.

Another interesting aspect is the characteristic of the input data from SLI scanners. In the target system, the measurement is multidirectional, which means that raw input data from scanning consists of a set of layers – parts of the model captured from single direction – which need to be matched to each other in order to receive the final cloud. Merging of the component clouds should be performed with sufficient accuracy to create a single surface.

Input data for the developed algorithm must meet following conditions. It must have:

- close to uniform sampling,
- sufficient merging accuracy,
- no artifacts.

3.3 Overview

The method begins with pre-processing, which is an optional step introduced to unify sampling and minimize noise. Next, boundary points are extracted. The resulting data form an unordered set of boundary points. Afterwards, boundary points are grouped into contours of particular holes. For each extracted cavity, a single Bezier patch is fitted. After sampling and merging, the cloud may be optionally post-processed. The output of the process is a cloud with filled holes (Fig. 8).

The CRASS algorithm was tested on both real scanned data and computer-generated examples of basic shapes: a computer-generated sphere (0.5 million points, generated based on parametric equations with radius equal to 100 and sampling resolution equal to 500), a scan of a female mannequin (1 million points), and a scan of a standing man (0.85 million points). Real scans were captured using an 8-directional SLI scanner [26]. As mentioned in Section 3.2, the measurement precision in the target system was estimated as 0.5 of the AMD.

3.4 Pre-processing

Data from SLI scanners is burdened with disturbances caused by thermal noise in the detector matrices, and by the measurement method itself [23]. To achieve the prerequisites specified in Section 3.2, additional processing steps have to be introduced.

In the first step, the data were smoothed using a plane-fitting algorithm. For each point, a spherical neighborhood of size $r_{PreSmooth}$ was found and used to approximate a local best-fitting plane. The processed point was projected onto the plane and its position updated.

Next, the data was simplified using a custom simplification algorithm, which for given minimum distance d iterates over cloud's points and deletes points nearer then d. This simplification was used with parameter $d_{preSimpl}$ to unify density in areas where sub-clouds from particular scanning directions overlap (Fig. 9).

This step requires setting the parameters $r_{PreSmooth}$ and $d_{preSimpl}$.

3.5 Boundary point extraction

Boundary points were extracted from the input cloud using a two-step algorithm [3]. The first step involves checking the condition that describes the minimum distance between a point and the center of its spherical neighborhood of radius r_{dist}. In the standard case, when a point is located in the regular part of the surface, the distance

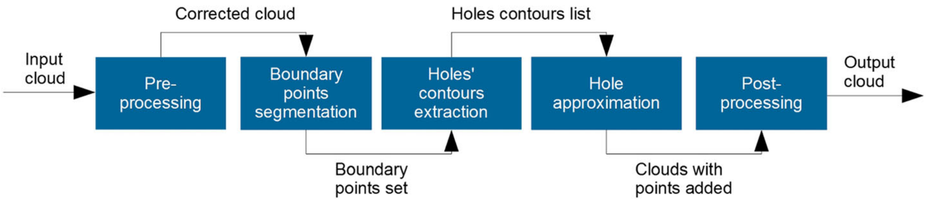

Fig. 8 Algorithm scheme

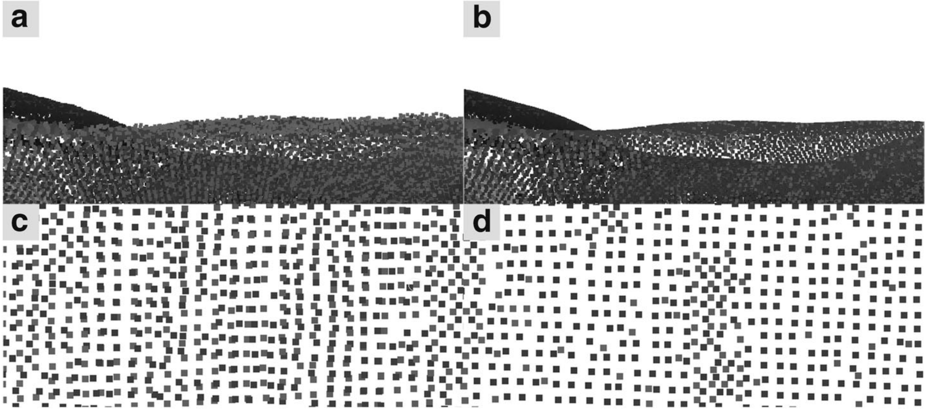

Fig. 9 Input cloud before (right) and after (left) pre-processing. Views **a** and **b** show scanned surface; **c** and **d** present zoom on point sampling in surface. In view (**a**), noise on the scanned surface can be observed. In view **c**, sampling density is irregular, many points either overlap or are slightly shifted from each other. After the pre-processing algorithm, sampling density is unified and noise reduced (**b**, **d**)

to the center is relatively small. The more the point "moves" towards the edge, the more the distance is increased (Fig. 10).

Point p_i passes the distance test if the following condition is met:

$$||p_i - \mathrm{a_i}|| \geq \epsilon_{\mathrm{dist}} \tag{3.10}$$

where:

p_i position of point p,

a_i the center of the spherical neighborhood of radius r,

ε_{dist} is the distance threshold set by the user.

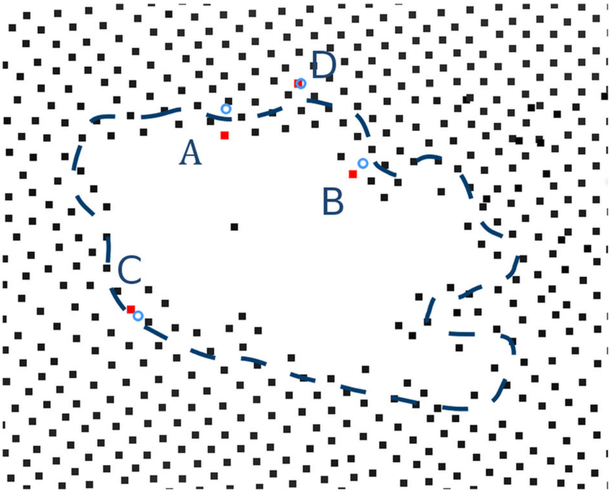

Fig. 10 Distance between center of neighborhood (blue circle) and point (red rectangle) for points on the edge of a cavity (A, B, C) and regular point (D). Dashed curve specifies cavity area

As a result, probable boundary points are selected. In the second step, final boundary points are chosen from the set of points found in the first step of the procedure.

Point p_i is a boundary point if:

$$\alpha_{max} \geq \epsilon_{\mathrm{bangle}} \tag{3.11}$$

where:

$\varepsilon_{\mathrm{bangle}}$ the angular threshold set by the user,
α_{max} maximum of wedge angles for point p.

α_{max} is calculated as follows (Figs. 11 and 12):

1 For the considered point p_i, take the spherical neighborhood of radius r_{ang} (name it as set N of point n_j).
2 Calculate the best-fit plane P_{pi+N} for N and p_i, and project all points onto P_{pi+N} (name the projected points p_i' and n_j' in set N′).
3 Create a local coordinate system with origin in p_i' and x-axis vector determined by the first point from N′.
4 Sort the points from N′ with respect to p_i' according to their polar coordinates (by angle to the x-axis).
5 For each two consecutive points from the sorted N′, calculate the angle α spanned by those points with apex in point p_i'.

The quantity α_{max} is the maximum value of all angles α in the sorted set N′.

This step requires setting the parameters $\varepsilon_{\mathrm{dist}}$, $\varepsilon_{\mathrm{bangle}}$, r_{dist} and r_{ang}.

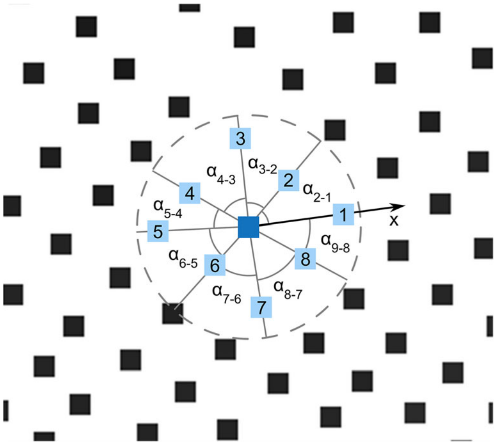

Fig. 11 Wedge angle α calculation for sample point p_i' (dark blue) after projection on plane P_{pi+N}. Dashed circle corresponds to spherical neighborhood of radius r, light blue points are neighbor points n_j' of sample point. Point no. 1 is used to set x-axis vector, after which all neighbor points are sorted with respect to α (numbers on points imply order after sorting). For each two consecutive points, angle α spanned by those points with apex in point p_i' is calculated. Maximum value from α is named α_{max}

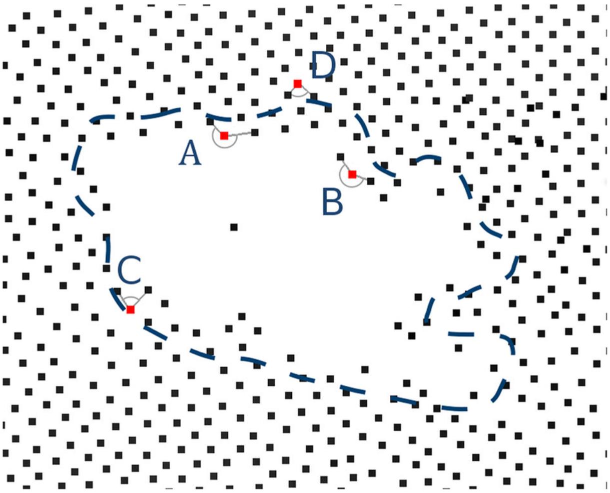

Fig. 12 Values of α_{max} for sample points. Examples on the edge of a cavity (A, B, C) and within regular surface (D). Dashed curve specifies cavity area

3.6 Cavity extraction

The aim of this step of the algorithm was to group boundary points into lists of points, each representing a separate cavity. The algorithm builds polylines representing cavities based on two initial points (any point from the set and its closest neighbor) serving as a seed, and adds the next points iteratively if the condition for the allowed expansion direction is met (see Figs. 13 and 14).

Let B be the set of boundary points extracted in the previous step. Firstly, a free (i.e., not assigned to any cavity) (boundary) point from B is chosen and its nearest neighbor (from B) is found. If the distance between them is smaller than the defined maximum extending distance $d_{maxExtnd}$, the mentioned pair of points is used to form a new cavity polyline. Next, more points are added to the list based on the allowed extension test. Such points, if found, are added to the proper end of the list. This action is repeated iteratively until the polyline represented by list C cannot be expanded anymore. After that, if B is not empty, lists for the next cavity are created and expanded, until there are no points left in B. In result of this step, we obtain a set of lists of points, each representing a separate cavity.

The allowed extension condition is determined by two tests. Firstly, the point to extend the polyline cannot be further from the current end of the polyline by more than $d_{maxExtnd}$. This

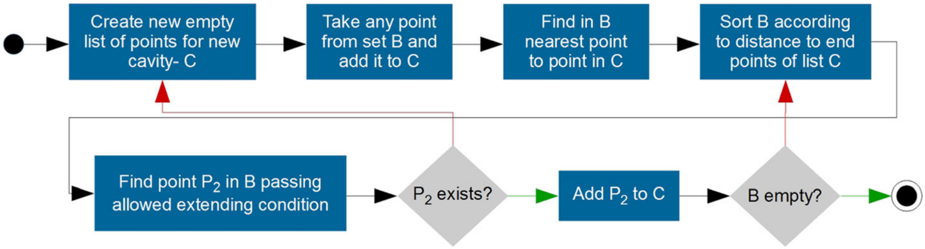
Create new empty list of points for new cavity- C
Take any point from set B and add it to C
Find in B nearest point to point in C
Sort B according to distance to end points of list C
Find point P2 in B passing allowed extending condition
P2 exists?
Add P2 to C
B empty?

Fig. 13 Extraction algorithm diagram

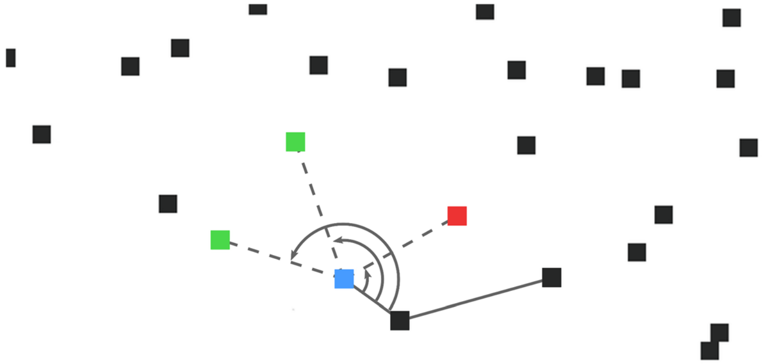

Fig. 14 Cavity extraction method. Solid line specifies current boundary polyline; blue point is current end point of boundary polyline. Green and red points are considered extensions of the polyline. Arcs show angular criteria for each considered point, based on which red point is rejected for extending polyline. From the remaining two green points, the closer one will be chosen

prevents the polyline from including points from other, nearby cavities. Secondly, the angle spanned by the last two points of the current polyline and the considered extension point (see Fig. 14) is calculated. The point is an allowed extension if the mentioned angle is larger than the user-defined parameter $\phi_{\mathrm{forbidDir}}$. This prevents the polyline from curving too sharply. This step requires setting parameters d_{maxExtnd} and $\phi_{\mathrm{forbidDir}}$.

3.7 Calculating Bezier patches

The approximation step fits a Bezier patch to a cavity defined by an ordered list of boundary points. We propose the choice of additional points, called surrounding points, which along with boundary points are considered in the patch approximation formula.

Our algorithm firstly estimates a local coordinate system ($X_{\mathrm{cavL}}Y_{\mathrm{cavL}}Z_{\mathrm{cavL}}$) for each cavity by calculating the best-fitting plane for all its boundary points. The XY plane is the best-fit plane, and the z-axis is the plane's normal vector. The origin is chosen as the center of mass of the boundary points projected onto the best-fit plane. Boundary points are then transformed to the new coordinate system. A cavity-bounding rectangle is determined in ($X_{\mathrm{cavL}}Y_{\mathrm{cavL}}$) and its borders are sampled with a density proportional to the fitted patch degree. Points extracted from the rectangle form the set R. For each point from R, the nearest point in the cloud is found, in the sense of ($x_{\mathrm{cavL}}y_{\mathrm{cavL}}z_{\mathrm{cavL}}$) local coordinates (Fig. 15) with privilege for the z_{cavL} coordinate. Distances along the z-axis and in XY plane are calculated separately. For two sorted points, if the Z distance is $\sqrt{5}$ times larger than the XY distance, comparison is performed using only the Z distance. Otherwise, comparison is made in the XY plane. Each nearest point that is found is added to the set S of surrounding points. Distinction between Z and XY distance causes that the choice of surrounding points is based more on distance to projection onto plane than on distance between surrounding point and point from R. Secondly, it helps avoiding misassignment of points from back side of the scan as surrounding point, because for such a far-away points comparison is done only in Z axis and points from front side of the cloud will be advantaged. Choice of value of the threshold in this case is experimental.

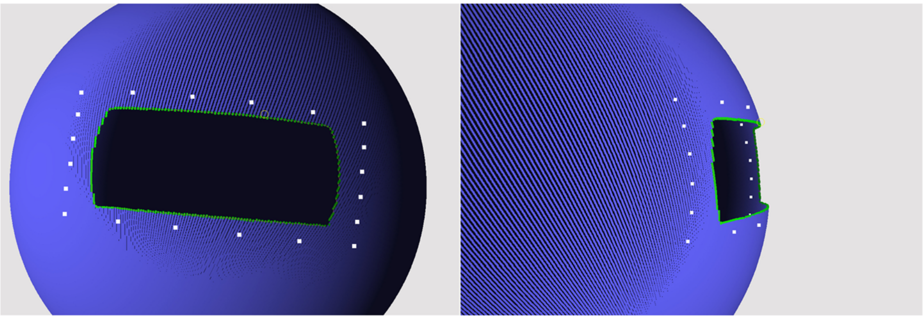

Fig. 15 Projected bounding rectangle points in sphere cloud (white points). Green points correspond to hole boundary points

Next, both the boundary and surrounding points are used as input for the fitting process, which is performed in the local coordinate system of the cavity. The formula consists in distance term defined as follows.

Consider an approximating Bezier patch defined by control points p = [p_{00}, p_{01}, …, p_{10}, p_{11}, …, p_{n*m}]^T (3.12):

$$\mathbf{f}(u,v)=\begin{bmatrix} x(u,v)\\ y(u,v)\\ z(u,v)\end{bmatrix}=\sum_{i=0}^{N}\sum_{j=0}^{M}\left(\mathrm{B}_i^n(u)\cdot\mathrm{B}_j^m(v)\cdot\begin{bmatrix} x_{p_{ij}}\\ y_{p_{ij}}\\ z_{p_{ij}}\end{bmatrix}\right) \tag{3.12}$$

where:

f(u,v)	Bezier patch function,
$x_{p_{ij}}, y_{p_{ij}}, z_{p_{ij}}$	x, y, z coordinates respectively of control point p_{ij},
N, M	degrees of the patch in the u and v directions.

For all boundary points (set Q) and surrounding points (set S), the building set T of adapting points and objective function are given by (3.13):

$$\begin{aligned} \mathrm{Q}= &\ \{\mathbf{Q_0},\mathbf{Q_1},\dots,\mathbf{Q_L}\},\mathrm{S}\ =\ \{\mathbf{P_0},\mathbf{P_1},\dots,\mathbf{P_{N*M}}\},\\ \mathrm{T}= &\ \mathrm{QuS}\ =\ \{\mathbf{T_0},\mathbf{T_1},..,\mathbf{T_{L+M*N+1}}\}\\ \mathbf{F}(\mathbf{p})= &\sum_{k=0}^{L+M\cdot N+1}\mu_k\left(\mathbf{f}(u_{T_k},v_{T_k})-\mathbf{T_k}\right)^2 \end{aligned} \tag{3.13}$$

where:

F(p)	objective function,
Q	set of all boundary points,
S	set of surrounding points,
μ_k	the weights of the boundary and surrounding points. These are either μ_b for boundary or μ_r for surrounding points. Manipulating them will modify the calculated surface to better fit either the boundary or surrounding points.

As the function F is quadratic and positive, finding its minimum value reduces to solving the Eq. (3.14):

$$\nabla \boldsymbol{F}(\boldsymbol{p}) = 0 \tag{3.14}$$

The above equation can be treated as a set of Eq. (3.15)

$$\frac{\partial \boldsymbol{F}(\boldsymbol{p})}{\partial \boldsymbol{p}_{ij}} = \sum_{k=0}^{r} 2\mu_k (\boldsymbol{f}(\mathrm{u}_{\mathrm{T_k}}, \mathrm{v}_{\mathrm{T_k}}) - \boldsymbol{T_k}) \cdot \mathrm{B}_{\mathrm{i}}^{\mathrm{n}}(\mathrm{u}_{\mathrm{T_k}}, \mathrm{v}_{\mathrm{T_k}}) \cdot \mathrm{B}_{\mathrm{j}}^{\mathrm{m}}(\mathrm{u}_{\mathrm{T_k}}, \mathrm{v}_{\mathrm{T_k}}) \tag{3.15}$$

with each equation expressing the derivative of a control point $\mathbf{p}_{\mathrm{ij}}$.

Using both the boundary and surrounding points (which are obtained using projection on the bounding rectangle on the cavity's best-fit plane) is the key to making the resulting patch approximate the local shape around the cavity using a simple set of equations.

For each adapting point, (u,v) coordinates were calculated using planar mapping of that point's projection onto best-fit plane calculated previously fot the search of the surrounding points. The bounding rectangle of boundary points, projected onto the plane and scaled up by 10%, defined a [0,1]x[0,1] (u,v) area on the plane.

The presented set of Eq. (3.15) was solved using the Least-Squares Method provided by the Eigen library [13]. The result of the calculation is a set of control points for the Bezier patch. The obtained parametric surface is defined in the local (i.e., that of the cavity) coordinate system.

In order to sample the patch only in the area where it covers the cavity, sampling inside the convex hull of the cavity [8] was introduced. The contour of the cavity was projected onto the best-fit plane, and the convex hull was calculated for the result. While sampling the Bezier surface, each point's coordinates in ($X_{\mathrm{cavL}}Y_{\mathrm{cavL}}Z_{\mathrm{cavL}}$) were tested against the Point-in-Polygon condition with respect to the hull. Only points inside the convex hull were added to the final cloud. Sampling of the patch was performed in parameter (i.e., UV) space. The densities in the u and v directions are proportional to the AMD. This step requires setting the parameters μ_b, μ_r, N and M.

3.8 Post-processing

After the Bezier patches were added to the original cloud, a post-processing step similar to the pre-processing operation was used in order to improve the result visually. Because the Bezier patch was sampled inside the cavity's convex hull, simplification was needed to keep the point density semi-constant over the entire area. Post-processing alters the original cloud, but the parameters were adjusted to minimize losses in the data and to obtain an improved visual effect.

Figure 16 shows a discontinuity on the edges of the patch, where the original and reconstructed surfaces are supposed to connect smoothly (Fig. 17). Therefore, smoothing is needed to join both surfaces. Smoothing was accomplished using a plane-fitting algorithm with a radius $r_{\mathrm{postSmooth}}$ as the spherical neighborhood size for the calculating planes. Sampling of the patch was performed in parameter space, which causes some density differences compared to the input cloud. In order to keep the sampling density similar, homogeneous simplification with parameter $d_{\mathrm{postSimpl}}$ was used. The value of the simplification parameter should be less than the AMD in order to avoid removing any original data. This step requires setting the parameters $r_{\mathrm{postSmooth}}$ and $d_{\mathrm{postSimpl}}$.

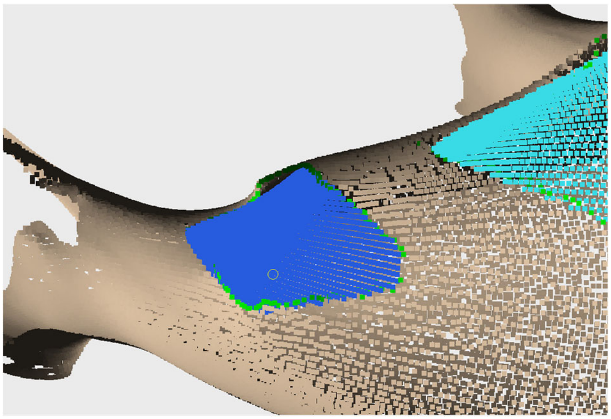

Fig. 16 Fitted Bezier patches before post-processing

4 Results

4.1 Algorithm assessment method

The CRASS algorithm was developed for two target uses: further calculations (e.g., for automatic dimensioning for the clothing industry) and visualization purposes. Therefore, it should be assessed not only in a visual way, but also as a numerical tool. In this paper, the testing method is based on comparison between the original data and an artificially created cavity reconstructed using the algorithm. To provide ground-truth information, original scan was hollowed manually and the cavity was reconstructed using algorithm under assessment. The difference between original part of scan and reconstructed surface indicates how well the algorithm performs.

In order to compare the original cloud with the reconstructed one, a separate algorithm was developed (see Fig. 18). As post-processing was used only to provide a good visual effect, it was omitted for the purpose of assessment.

In both clouds (i.e., original and processed), similar areas of points were selected (we name these sets P_o for the original and P_r for the reconstructed cloud). At each point p_r of the reconstructed cloud for these defined ranges, the corresponding point in P_o is searched for. A new (local) coordinate

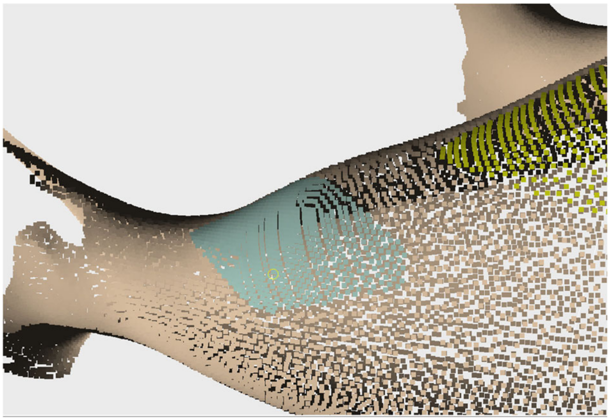

Fig. 17 Fitted Bezier patches after post-processing

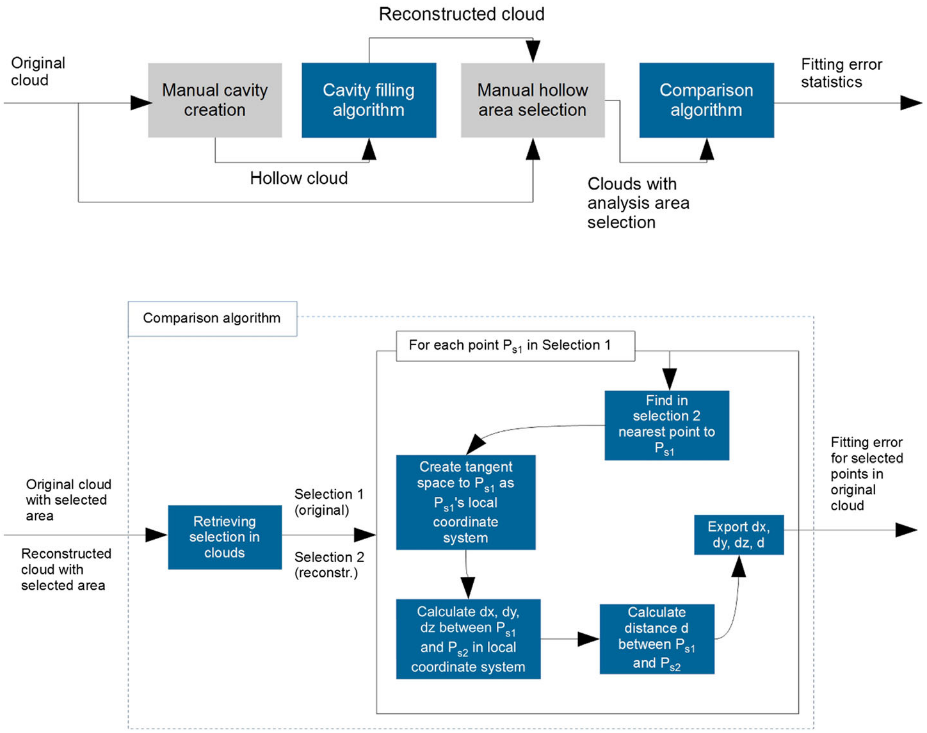
Reconstructed cloud
Original cloud
Manual cavity creation
Cavity filling algorithm
Manual hollow area selection
Comparison algorithm
Fitting error statistics
Hollow cloud
Clouds with analysis area selection
Comparison algorithm
For each point P_{s1} in Selection 1
Find in selection 2 nearest point to P_{s1}
Create tangent space to P_{s1} as P_{s1}'s local coordinate system
Original cloud with selected area
Reconstructed cloud with selected area
Retrieving selection in clouds
Selection 1 (original)
Selection 2 (reconstr.)
Export dx, dy, dz, d
Fitting error for selected points in original cloud
Calculate dx, dy, dz between P_{s1} and P_{s2} in local coordinate system
Calculate distance d between P_{s1} and P_{s2}

Fig. 18 Assessment method scheme

system was formed with point p_r as the origin, the tangent (to p_r) plane spanning the X-Y space, and the normal vector of p_r serving as the unit vector in the z-direction. The corresponding point was found in the local coordinate system as the nearest with privilege in the local XY coordinates, similar to the method of finding the surrounding points of the cavity (see 3.7 and Fig. 15), with the difference being that the plane was calculated separately for each point p_r as the plane constructed with its origin in p_r and its normal vector equal to that of p_r. In this way, the corresponding point p_{on} is found from the original set. The distance between p_r and p_{on} is taken as the absolute fitting error for the point p_r. The relative fitting error was defined as the absolute fitting error divided by the AMD. The resulting value was used for histograms and charts for error analysis. All the values were assigned the xy-coordinates of p_r in the local ($X_{cavL}Y_{cavL}Z_{cavL}$) coordinate system of the cavity.

The search for corresponding points was made in the local coordinate system in order to compare points correctly from surfaces of different levels of variability. If the original surface was more complicated than the applied Bezier patch, a standard nearest-point search would select not the corresponding, but the closest point (see Fig. 19). In order to avoid this, a search in local coordinates with privilege for local XY coordinates was introduced. As the XYZ-UV mapping was made using the bounding rectangle on the best-fit plane, it was necessary to use a similar strategy in the corresponding point search.

4.2 Experiment environment setting

The algorithm's efficiency was tested using a PC with an Intel Xeon CPU dual core (2.93 GHz), 16 GB RAM and an HDD drive.

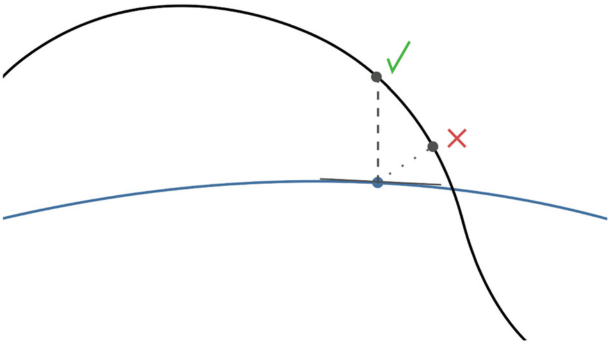

Fig. 19 Nearest point search for comparison algorithm. Blue curve symbolizes reconstructing Bezier patch, wavy one original data. Dotted line shows distance to globally nearest point, dashed line shows distance to appropriate point in searched in local coordinates. Tangent line symbolized point's tangent plane

The developed algorithm was implemented using the FRAMES library (a new generation software based on 3DMADMAC [20]), which provides an estimated average minimum distance between points. For the sake of efficiency, this quantity is calculated on a randomly chosen subgroup of 5000 points.

4.3 Summary

Tests of the algorithm were performed on a parametrically generated sphere, and on scans from the target system (Fig. 20). Table 2 summarizes errors for each of tested point clouds. Figures 21, 22, 23 and 24 depict results on the sphere. Figures 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38 and 39 show results on artificially created cavities in scans from the target system. Visual examples of entire method (incl. post-processing) on real cavities from scans were presented on Figs. 17, 40 and 41.

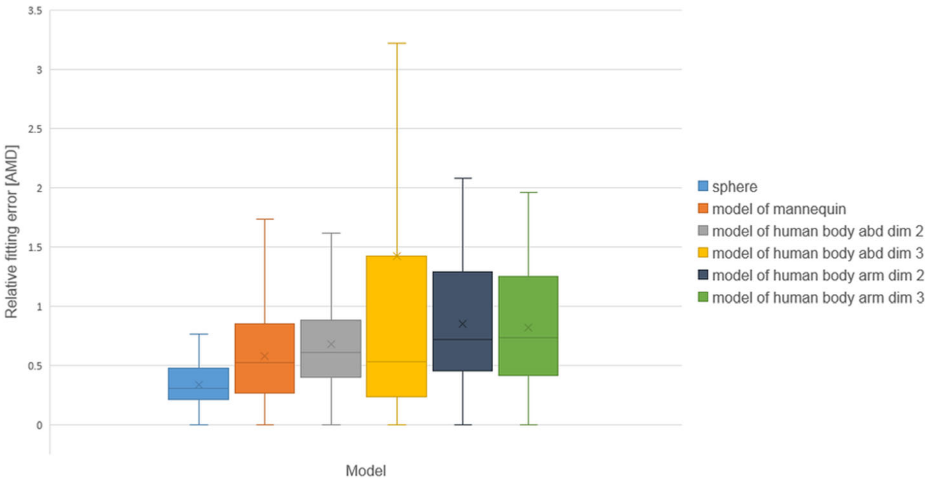

Fig. 20 Relative fitting-error box plot for the models that were used

4.4 Algorithm assessment using synthetic data

Tests performed on the generated sphere (Figs. 21, 22, 23, and 24) gave relatively good results for the following parameter values: $\varepsilon_{\mathrm{dist}} = 0.5$ x AMD, $\varepsilon_{\mathrm{bangle}} = 1.5$ rad, $\phi_{\mathrm{forbidDir}} = 0.7$ rad, and $d_{\mathrm{maxExtend}} =$ 20 mm. The cavity was prepared manually. A homogenous sampling density in the tested point cloud allowed the use of a wide range of values of the angle parameters.

4.5 Algorithm assessment using real data

Tests were then performed on a measurement of a mannequin of a standing woman (Figs. 25, 26, 27, and 28). A single cavity was cut manually in the abdomen area, and was filled computationally using the developed method. The relative-error histogram shows that values of the error were mostly concentrated in the range [0, 0.8]. Beyond this range, the frequency of an error's occurrence decreases exponentially.

For model no. 2, similar processing was carried out using individually adapted parameters (Figs. 29, 30, 31, 32, 33, and 34). For a cavity of about 6500 points in abdominal area, the following results were obtained:

Histograms and relative-error charts depending on XY coordinates were generated for the same cavity filled with a patch of degree (3,3). A large maximum value was considered to be that connected with outliers (9 samples). A further analysis was made for points with error in the range [0, 32] (amounting to over 6000 samples). The distribution of errors shows an increase for values of x in the range (525, 550). Outside this range, the error decreases to form a semi-regular distribution. In addition, Fig. 33 shows that 6118 probes (88.5%) are in the range (0, 3.2). The data form a wide range of values (9.6–32) with a small frequency, therefore a separate histogram for values near to the first peak is provided (see Fig. 34).

In forearm area on the same model another cavity was prepared (Figs. 35, 36, 37, 38 and 39). Forearm case served as high curvature example, which analyzed together with abdominal case allows making deductions about algorithm's robustness to surrounding shape curvature.

4.6 Performance

In addition to the artificially created sphere, the performance of the algorithm was measured for the original full models, i.e., all cavities that were originally found in the measurement data were analyzed. The performance was measured for the algorithm without pre- and post-processing (Table 3).

4.7 Comparison to poisson method

Proposed algorithm was compared to screened Poisson method. With this end in view two kinds of artificial cavities in model of human body were reconstructed using Screened Poisson method implemented in MeshLab software [4]. The reason for choosing Poisson as comparison method is first its high popularity in another comparison studies, what enables the researchers to compare independent works against the same basis technic. Secondly, Screened Poisson has freely available implementation in Meshlab, making it available to a larger group of scientists.

First, different values for input parameters were tested to achieve best result possible. The most promising input parameters setups were used to generate reconstructions for comparison with proposed method.

Table 2 Fitting error statistics for test point clouds. The relative fitting error was defined as the absolute fitting error divided by the AMD. Fitting error was defined as distance between corresponding points in original cloud and our reconstruction of manually created cavity in original cloud

Relative Error	Min	Max	Average	Median
Sphere	0.000	0.764	0.339	0.307
Model of mannequin	0.000	2.662	0.588	0.523
Model of human body patch (2,2)	0.000	4.527	0.681	0.610
Model of human body patch (3,3)	0.000	51.738	1.423	0.535

Two examples were used for the comparison: an artificial cavity in the abdominal area (Fig. 29) and an artificial cavity on the forearm (Fig. 35), both on the human body model. First example serves as a relatively planar case whereas the second one serves as a higher curvature case.

The Poisson reconstruction was run with a number of parameter values to ensure reliable results. Details of used parameters can be found in the Appendix.

Numerical accuracy results from Poisson reconstruction are presented in Tables 4 and 5, Figs. 42, and 43.

5 Discussion

5.1 Estimation of parameters for automated processing

In the CRASS algorithm, the parameters determine the correctness of the results. For this reason, some attempt to estimate the correct value ranges has been made. In order to automate the algorithm, we also define values of the required parameters according to cloud characteristics so that additional knowledge is not required about the model. Cloud characteristics like Average Minimum Distance and measurement precision are known for every scan, no matter if is a laser scan or structured light scan, what makes this selection application independent. The presented values may be treated as a general reference for setting parameters in a real-world scenario. Generally, the estimated values of the parameters will be called reference values.

5.1.1 Boundary-point segmentation

Two radial parameters, r_{dist} and r_{ang}, must be defined for neighborhood analysis in the step described in Section 3.5. Both radiuses should be large enough to include neighboring points

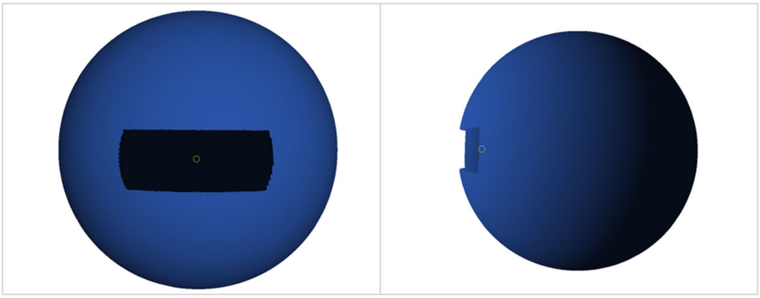

Fig. 21 Testing hollow sphere

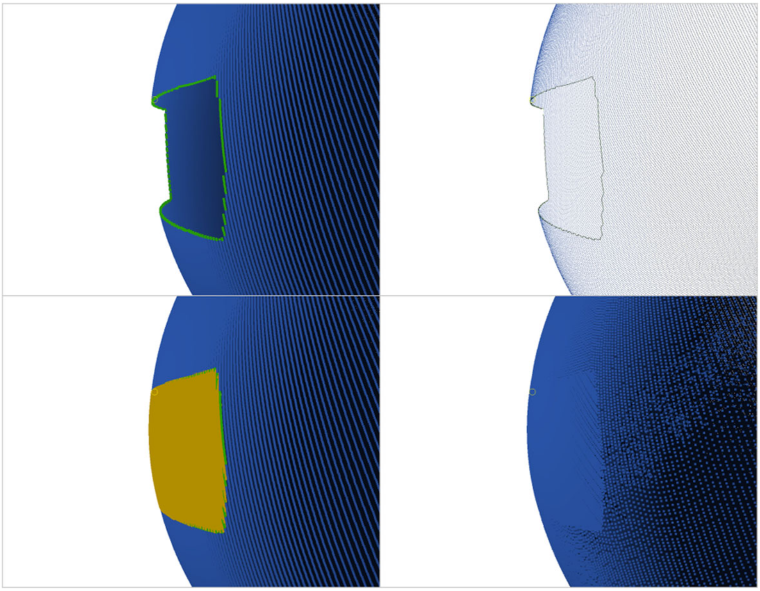

Fig. 22 Results of particular processing steps of developed method (upper left- boundary select, upper right- cavity poly, bottom left – filling with Bezier patch, bottom right- post processing

in the strip around the processed point. Therefore, this quantity should be (5.1):

$$r_{dist} \geq d + \frac{\Delta d}{2} \tag{5.1}$$

where:

d average minimum distance (AMD) between points in input point cloud.
Δd measurement precision.

Increasing r_{dist} will decrease the difference between the distances to neighborhood centers for the regular and boundary points. The bigger the size of neighborhood, the less the absence

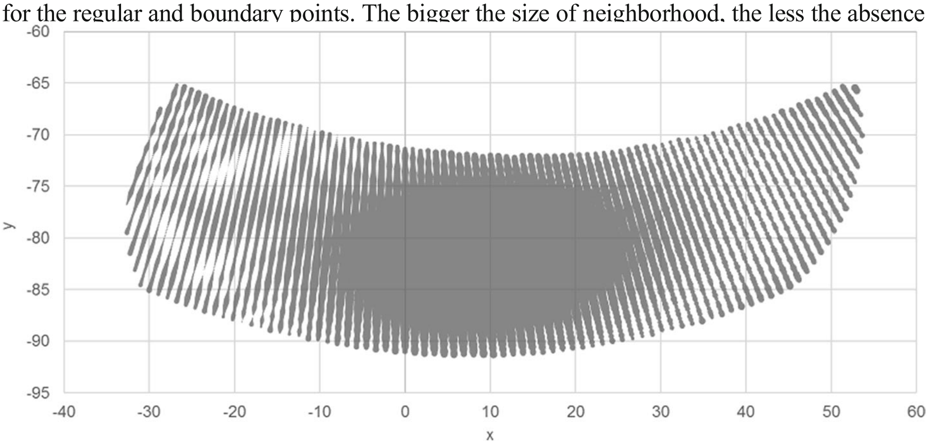

Fig. 23 Relative fitting-error depending on xy coordinates for sphere and patch of degree (2,2). Width of blob corresponds to value of error, scaled to range [0,10]

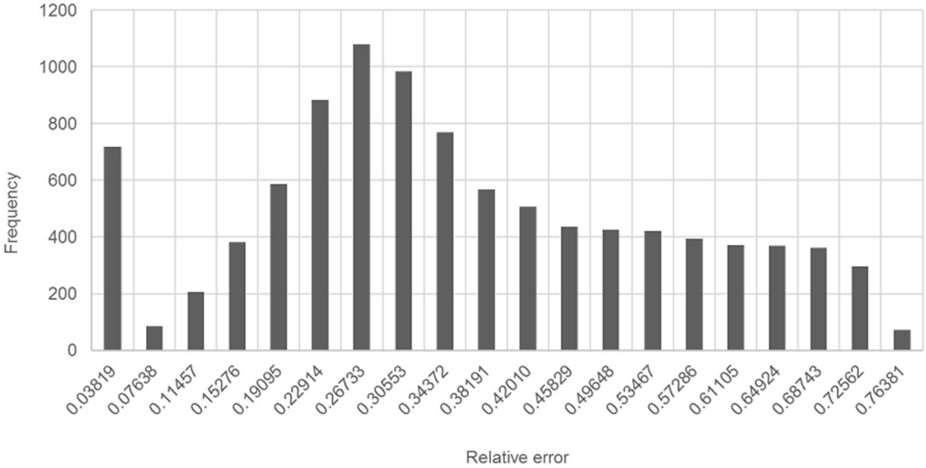

Fig. 24 Relative fitting-error histogram for sphere reconstructed using patch of degree (2,2)

of a few points affects the position of a neighborhood's center; what decreases is the precision of boundary-point selection. The optimal case occurs when only a single strip of nearest points is analyzed. In order to exclude points from the second strip, which are shifted from the corresponding points from the first strip by AMD, r_{dist} should not exceed:

$$r_{dist} \leq 2 \cdot d - \frac{\Delta d}{2} \tag{5.2}$$

The next parameter is the boundary threshold ε_{dist} described in Section 3.5. The difference in neighborhood center for various points was shown in Fig. 10.

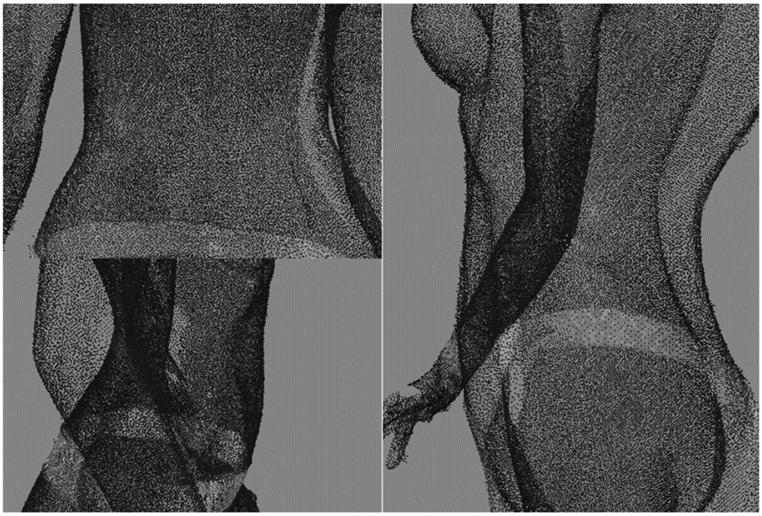

Fig. 25 Input data of woman posture

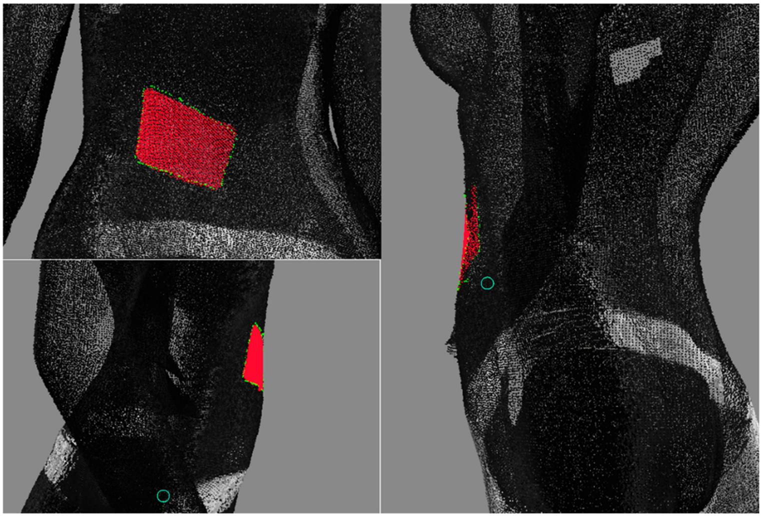

Fig. 26 Effects of developed algorithm for women posture case

After testing various values of this parameter, it was concluded that increasing $\varepsilon_{\mathrm{dist}}$ causes the selection of thinner strips of points around the cavity for the next threshold. As shown in Fig. 10, setting too strict a higher bound for this parameter would discard boundary points similar to point C. On the other hand, too lax a lower bound would cause the boundary to be wider than necessary (e.g., to include point D).

In order to estimate a proper value, tests were made on the generated sphere. The boundary-point segmentation step was calculated for values of $\varepsilon_{\mathrm{dist}}$ in the range [0.1 x AMD, 1.0 x AMD] (see Fig. 44). For values greater than 0.6 x AMD, some points on the boundary fail the test. For values less than 0.2 x AMD, points in the regular part of the surface were selected. A reasonable reference for the distance threshold parameter was estimated to be the range (0.2, 0.6) of AMD.

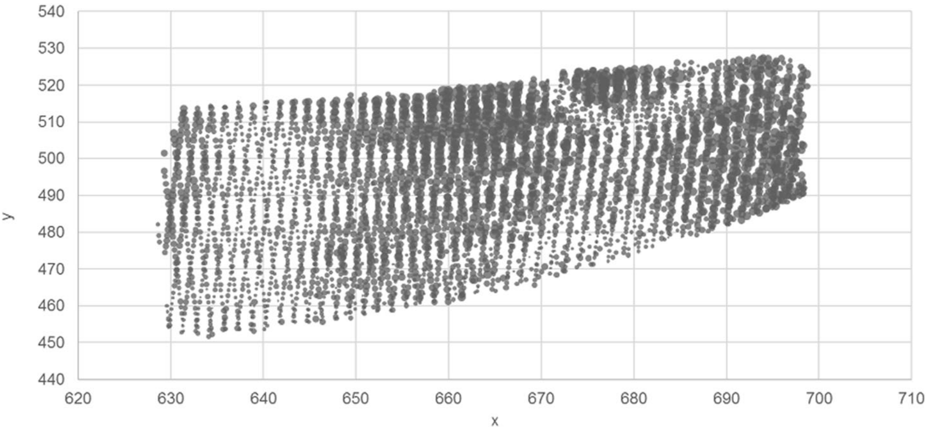

Fig. 27 Relative fitting-error for model of mannequin and patch of degree (2,2). Axes denote local xy coordinates. Size of blobs was scaled to range [0,10]

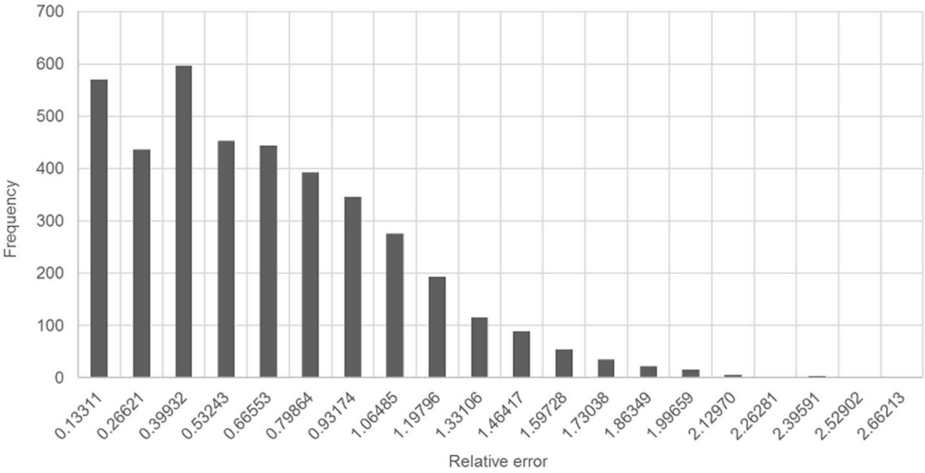

Fig. 28 Relative fitting-error histogram for model of mannequin and patch of degree (2,2)

The second threshold parameter is the wedge angle ($\varepsilon_{\mathrm{bangle}}$), which determines the threshold based on angular criteria, as described in Section 3.5 This condition is meant to select points on the edge of the strip, excluding points that are further away. Too high a value of the parameter results in skipping points on sharp edges. On the other hand, too low a value leaves wide strips on the edge. The lower bound can be defined by calculating the average angle between a regular point and its neighborhood points. This was done in Section 3.1, where the angle for a regular hexagonal formation of points on the surface was estimated to be in the range (0.33π, 0.6π).

In order to give a more reliable approximation, a test similar to the one used to estimate $\varepsilon_{\mathrm{dist}}$ was carried out. For the sample object, the wedge angle was calculated as described in 3.5. Points with a wedge angle greater than the given value were selected. Figures for several values of $\varepsilon_{\mathrm{bangle}}$ were generated to visualize the result. Calculations were made for values in the range (0.2π, $1.1\ \pi$) (see Fig. 45). The reference range for $\varepsilon_{\mathrm{bangle}}$ was estimated as ($0.4\ \pi$, $0.9\ \pi$). For smaller values, regular points pass the test, which is undesirable. For values larger than the upper bound of this range, many points on sharp edges are skipped, which would cause problems in cavity extraction. The optimal value of this parameter was estimated to be about

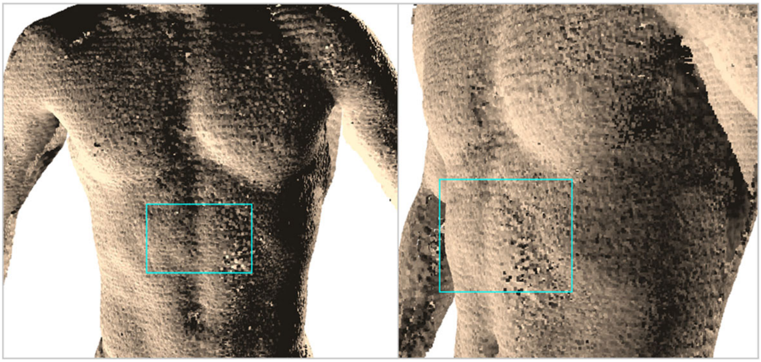

Fig. 29 Input data for tests. Rectangle shows area, where cavity was cut out manually

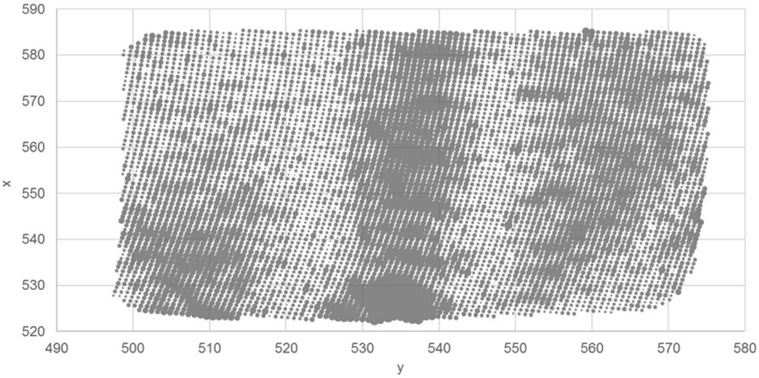

Fig. 30 Relative fitting-error for abdominal in human model and patch of degree (2,2). Axes denote xy coordinates. Area of blob corresponds to value of error, clamped to [0,10]

0.5π. This value allows the rejection of points from the regular part of the surface, but the acceptance of points on the boundary, including those located in the corners.

In the case of significant variability in sampling density, the distance parameter ($\varepsilon_{\text{dist}}$) should be increased. The angular parameter ($\varepsilon_{\text{bangle}}$) also partly depends on sampling density because the AMD between points affects the number of nearest surrounding points selected after a spherical-neighborhood search, which is parameterized with the AMD.

5.1.2 Cavity extraction parameters

The cavity-extraction step requires setting values of two parameters, d_{maxExtnd} and $\phi_{\text{forbidDir.}}$ The forbidden direction angle ($\phi_{\text{forbidDir}}$) is partly determined by the maximum wedge angle for a cavity in the cloud ($\varepsilon_{\text{bangle}}$). The wedge angle for a boundary point, when summed with the angle spanned by two subsequent points with an apex at this boundary point, forms a round angle. Therefore, the upper bound of $\phi_{\text{forbidDir}}$ is approximately 2π-α_{max}. Secondly, the forbidden direction angle should not block polyline creation between neighboring points in the case of ideal point distribution. This

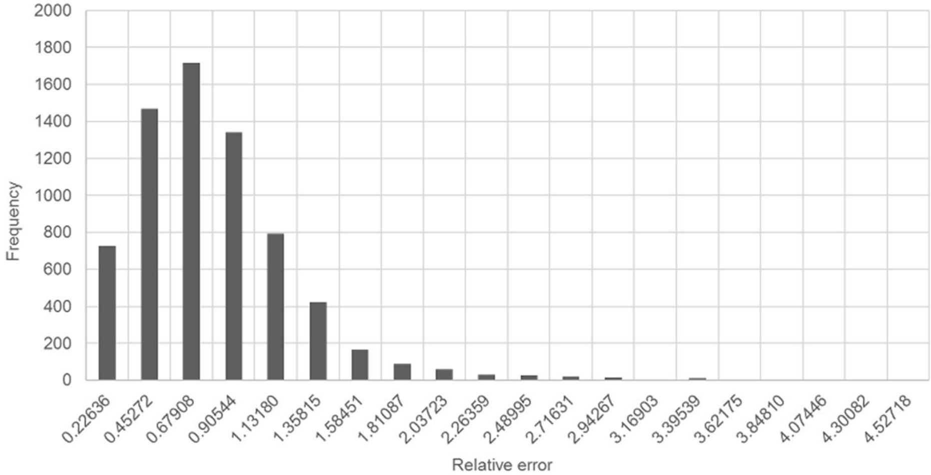

Fig. 31 Relative fitting-error histogram for abdominal in human model and patch of degree (2,2)

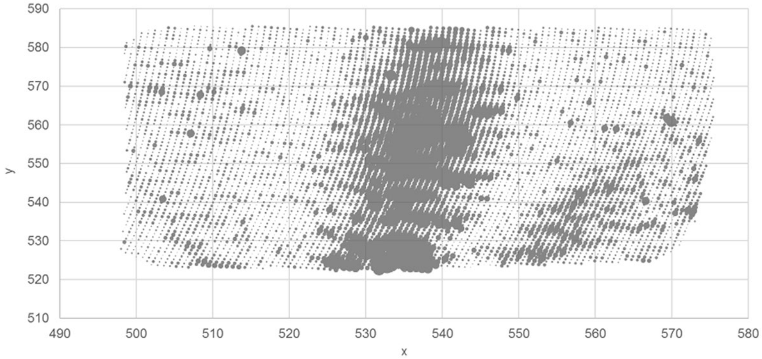

Fig. 32 Relative fitting-error for abdominal in human model and patch of degree (3,3). Axes denote xy coordinates. Area of blob corresponds to value of error, scaled to [0,20]

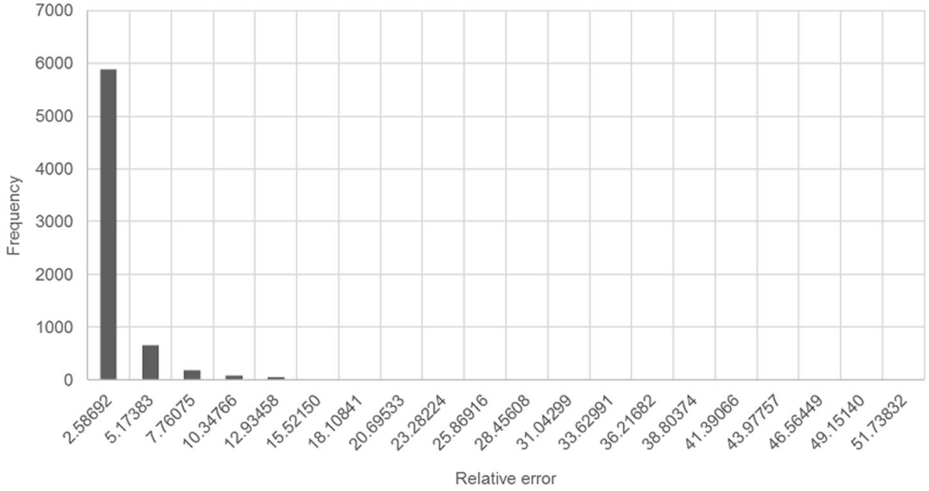

Fig. 33 Relative fitting-error histogram for abdominal in human model and patch of degree (3,3)

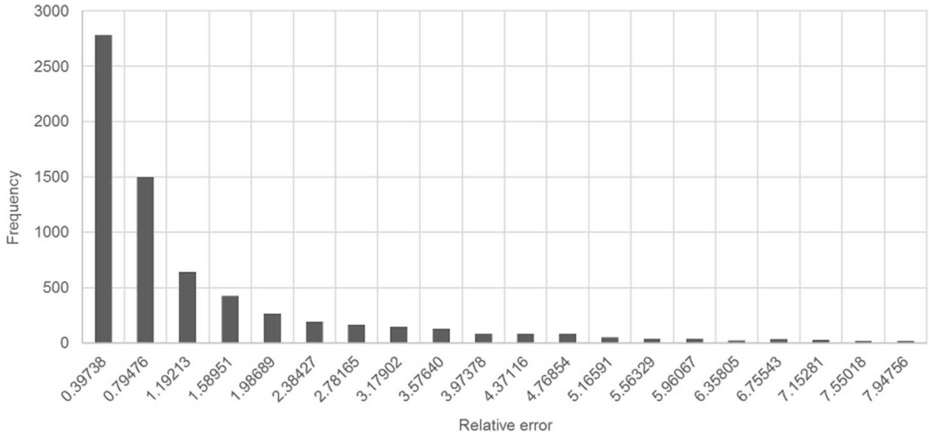

Fig. 34 Relative fitting-error histogram for abdominal in human model and patch of degree (3,3). Bin values were limited to the range [0,8] (remaining bins were skipped)

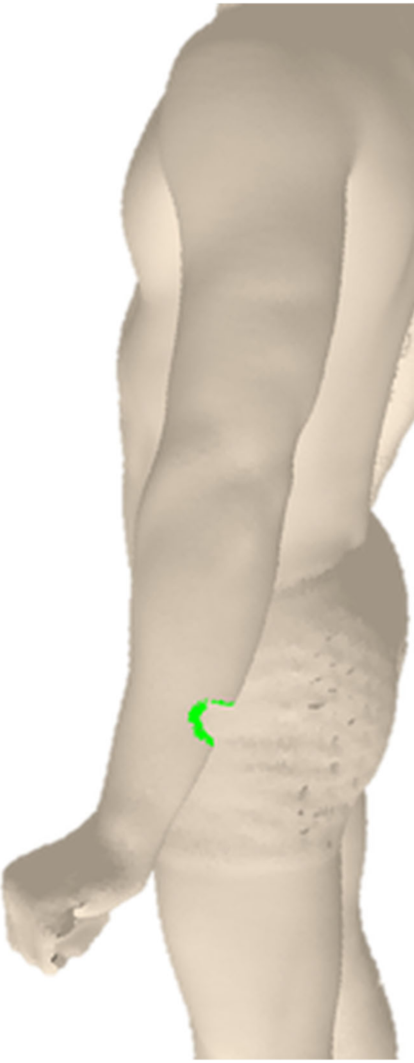

Fig. 35 Input cavity for forearm case of model of human body

implies that if the sampling density is unchanged on the bounds of the cavity, $\phi_{\mathrm{forbidDir}}$ should be small enough to allow a connection between neighboring points. Based on the discussion in Section 3.1, we assume that the value of this angle would be π/6.

Values of the wedge angle obtained from measurement for points in a sample cloud are shown in Fig. 46. The value of $\varepsilon_{\mathrm{bangle}}$ was mapped onto the color channel. The color of the scanned surface in the regular part is relatively constant, with some high frequency noise. This feature corresponds to the thermal noise mentioned previously. However, value pitch can be observed on the edges of holes. The maximum value of the wedge for Fig. 46 C was approximately 4.8 rad (about 1.53 π radians), which implies that the forbidden-direction angle is less than 0.47 π radians. Despite the noise in the regular parts of the surface, the difference on the edges allows extraction of boundary

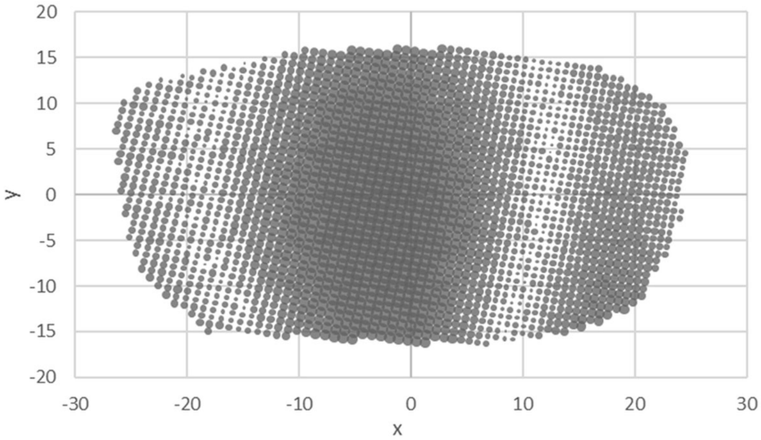

Fig. 36 Relative fitting-error for forearm in human model and patch of degree (2,2). Axes denote local xy coordinates of the cavity

points. Based on the presented results, the reference value for $\phi_{\mathrm{forbidDir}}$ was estimated to be in the range (0, 0.47π)∩(0, π/6) = (0, 0.47π) radians.

Setting the value of the second parameter d_{maxExtnd} allows a reduction in the influence of $\phi_{\mathrm{forbidDir}}$ on the correctness of cavity extraction. The lower bound for this parameter depends firstly on the AMD between points, secondly on the sampling density on the edges of the cavity, and thirdly on the parameters controlling boundary point segmentation (see Section 3.5). The two latter factors depend on the former one, so approximating them is a way to obtain the reference range for d_{maxExtnd}. Consider the ideal situation presented in Section 3.1. In the case of maximum position flow caused by measurement errors (Fig. 47), the distance between the considered point and the furthest point in the neighborhood equates to (5.3):

$$x=\sqrt{\left(\frac{d\sqrt{3}}{2}+\Delta d\right)^2+\left(\frac{d}{2}\right)^2}=\sqrt{d^2+d\sqrt{3}\Delta d+\Delta d^2} \tag{5.3}$$

where:

d average minimum distance (AMD) between points.
Δd measurement precision.

Table 3 Performance summary for sphere, model of mannequin, model of human body

Model	Number of points	Number of cavities	Total processing time [sec]	Processing time [sec/cavity]
Sphere	500,000	1	9	9
Model of mannequin	1,050,000	253	410	1.62
Model of human body	850,000	271	320	1.18

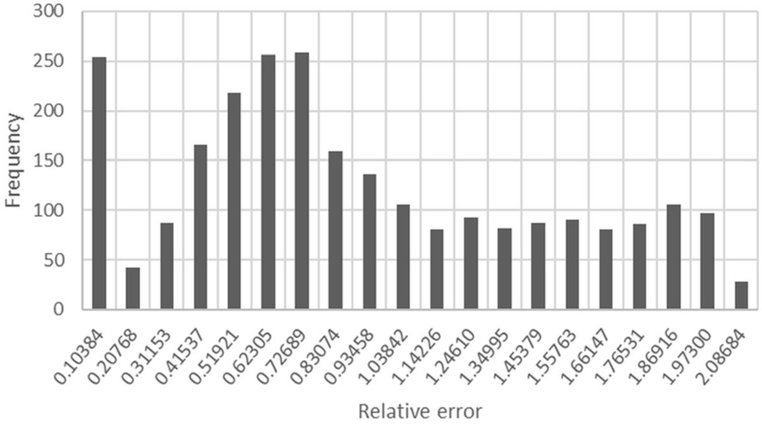

Fig. 37 Relative fitting-error histogram for forearm in human model and patch of degree (2,2)

In the case of our measurement system with $\Delta d = \frac{d}{2}$, this implies (5.4):

$$x = d\sqrt{\frac{5}{4} + \frac{\sqrt{3}}{2}} \approx 1.45\ d \tag{5.4}$$

Now consider the processing of a scan with uniform sampling density, even on the boundaries of the cavities. As a result of boundary-point segmentation, a single extraordinarily shifted boundary point may not qualify as a boundary point. This way, the distance between subsequent neighboring boundary points would be up to twice as large as normal. It should be assumed that in order to ensure proper polyline extraction, $d_{maxExtnd}$ should be larger than the distance between points that are neighbors of the skipped point. As the maximum distance between points in a uniformly sampled scan is equal to 1.45 of AMD, $d_{maxExtnd}$ should be larger than 1.45 × 2 = 2.95 AMD.

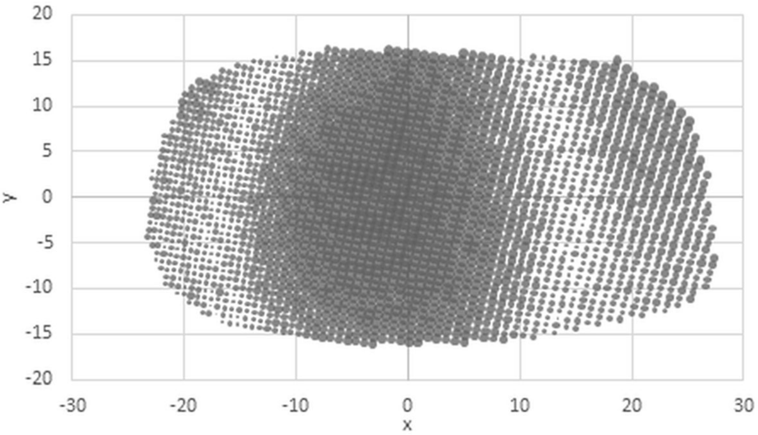

Fig. 38 Relative fitting-error for forearm in human model and patch of degree (3,3). Axes denote local xy coordinates of the cavity

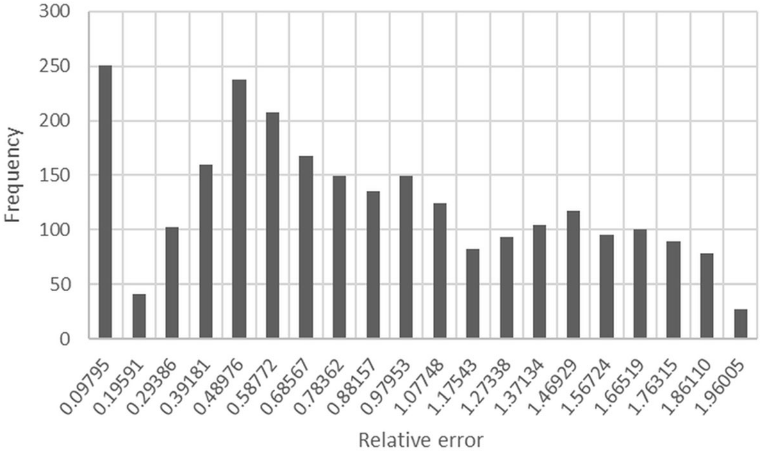

Fig. 39 Relative error histogram for forearm in human model and patch of degree (3,3)

5.1.3 Parameter estimation summary

Table 6 shows the parameters that were used for processing models during algorithm assessment. The presented values may be treated as reference values for setting the parameters for a custom case.

Parameters used for generating patches strongly depend on the characteristics of the scanned object. Because of this, the reference range was not estimated and only the sample values used for the model of a human body are given.

In order to even more facilitate reproducibility of results by other researchers, we listed parameters values used to evaluate our method in Table 6. We used reference ranges as staring point and fine-tuned the hyperparameters to achieve the best result.

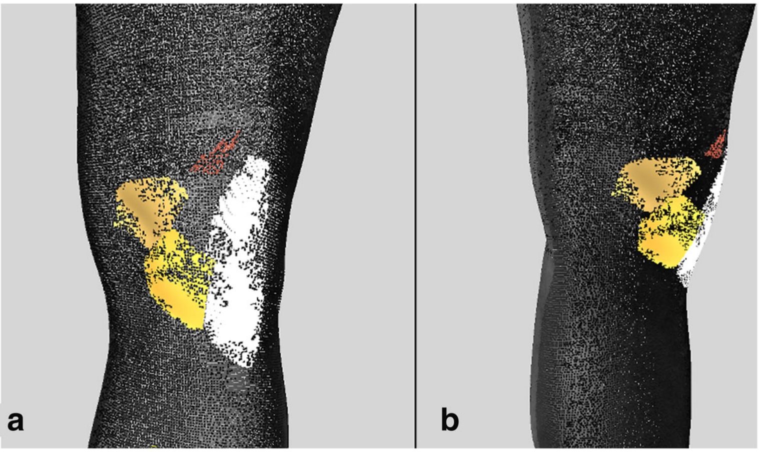

Fig. 40 Result of presented method (surface degree 5 × 5 with post-processing) on scanned data in region of knee, **a**- front view, **b**- side view

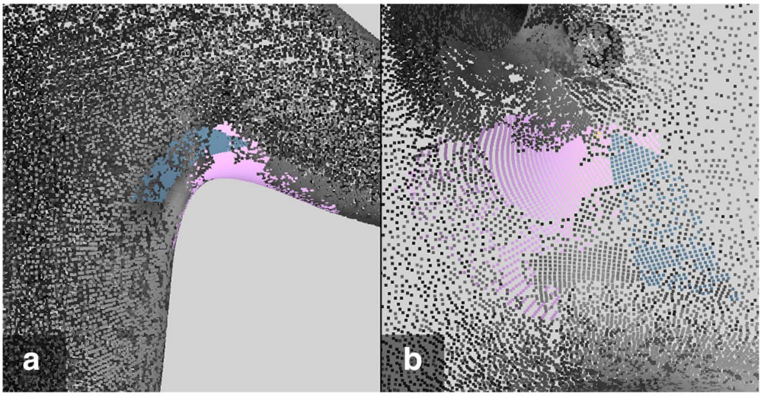

Fig. 41 Result of presented method (surface of degree 3 × 3 with post-processing) on scanned data in region of axilla, **a**- side view, **b**- top view from inside the surface

5.2 Fitting quality

The quality of the generated patches was assessed using the algorithm presented in Section 4.

5.2.1 Sphere

The results presented in Fig. 24 showed that the average fitting error for a patch of degree (2,2) is approximately equal to 0.34 AMD. The spatial distribution of the error (see Fig. 23) shows that the highest relative error was found in the middle part of the patch. Further away from the center, the error decreases and fluctuates around smaller values. Some regularity of the error distribution may be caused by the Bezier curves themselves, which do not reproduce conic sections ideally, but instead approximate them [10]. A sphere cannot be perfectly reconstructed with Bezier patches.

5.2.2 Model of mannequin

The fitting error presented in Figs. 27 and 28 is non-uniform due to a disturbance in the area at about (x ϵ [670, 690], y ϵ [500, 520]). This area corresponds to the navel, which introduces local complication of shape. However, the error histogram shows that the value of the error stays mostly in the range [0, 0.932], with a peak in the range [0, 0.399]. For 20% of the samples, the error exceeded the value of AMD, which probably corresponds to the area of the navel. A median value

Table 4 Comparison of reconstruction accuracy between our method and Poisson method for abdominal case of model of human body. Poisson #n denotes different parameter setup for the Screened Poisson method. Details of parameters used are listed in the Appendix

Relative Error (Model of human body, abdominal)	Min	Max	Average	Median
CRASS patch (2,2)	0.000	4.527	0.681	0.610
CRASS patch (3,3)	0.000	51.738	1.423	0.535
Poisson #00	0.047	2.815	0.432	0.476
Poisson #01	0.036	2.825	0.420	0.472
Poisson #02	0.048	2.851	0.428	0.479
Poisson #03	0.036	1.818	0.417	0.458
Poisson #04	0.052	1.901	0.459	0.493

Table 5 Comparison of reconstruction accuracy between our method and Poisson method for forearm case of model of human body. Poisson #n denotes different parameter setup for the Screened Poisson method. Details of parameters used are listed in the Appendix

Relative Error (Model of human body, forearm)	Min	Max	Average	Median
CRASS patch 2 × 2	0.000	2.077	0.849	0.718
CRASS patch 3 × 3	0.000	1.959	0.821	0.737
Poisson #00	0.090	8.788	0.950	0.589
Poisson #01	0.055	8.789	0.752	0.494
Poisson #02	0.054	7.832	0.742	0.492
Poisson #03	0.054	7.442	0.681	0.480
Poisson #04	0.117	14.610	1.200	0.653

of 0.58 AMD, compared to a measurement precision equal to 0.5 AMD, indicates that almost half of all the reconstructed points were reproduced with the desired quality.

5.2.3 Model of human body

The chosen part of the abdominal area forms a shape that is approximately parabolic, but with some higher-frequency features due to visible abdominal muscles. The error distribution for the model of a human body is semi-uniform, with higher values for x in the ranges [500, 520], [530, 540], [550, 570]. The middle strip corresponds to the shape of the abdominal muscles. The average relative error of 0.68 AMD is larger than desired. In this case, the visible abdominal muscles were not properly reproduced by our method, which was caused by the degree of the patch used. As shown in Fig. 31, most of the error was in the range [0, 1.352]. Bins with larger values of error correspond to the strip at x ϵ (530,540). Apart from errors in the middle part of the patch, the obtained result could be assessed as being acceptable, as the average error for this dataset is less than 0.5 AMD.

Changing the degree of the patch to (3,3) did not eliminate the problem that occurred in the middle area of the sample. The reason for this could be the method of choosing points for approximation. Firstly, boundary points are chosen, and then points from the extended bounding rectangle are added. Boundary points reproduce the change of shape in the middle strip, but boundary rectangle points do not, as the rectangle's edges are sampled with small

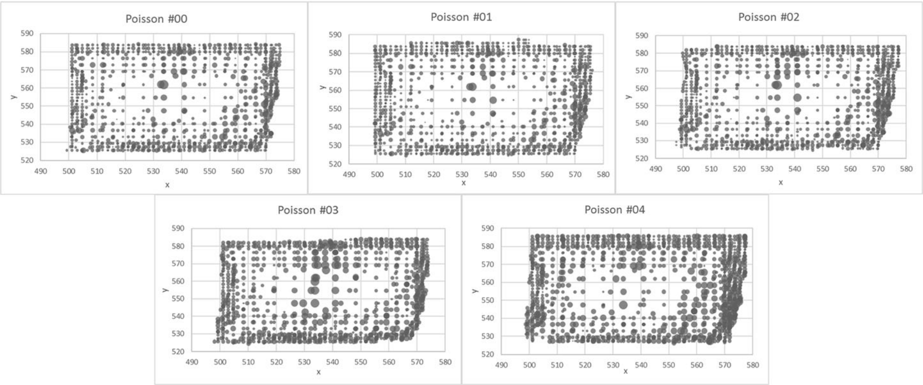
Poisson #00
Poisson #01
Poisson #02
Poisson #03
Poisson #04

Fig. 42 Relative error for Poisson on abdominal case (reconstruction depth = 10) depending on xy coordinates. Area of blob corresponding to error scaled to [0, 20]

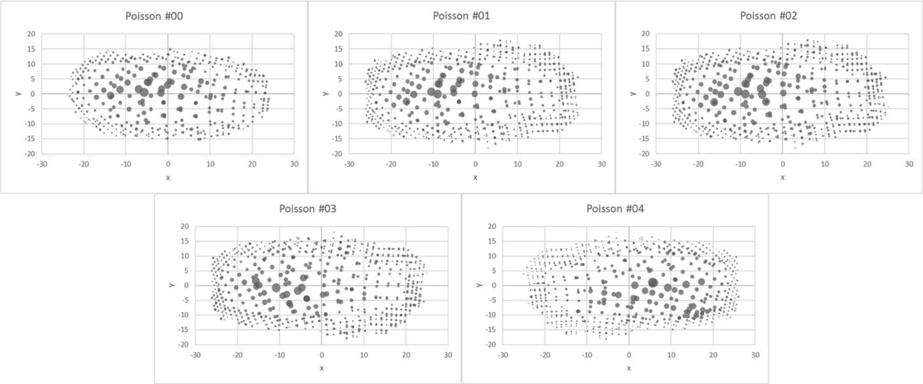

Fig. 43 Relative error for Poisson on forearm case depending on xy coordinates. Area of blobs corresponding to error was scaled to [0,20]

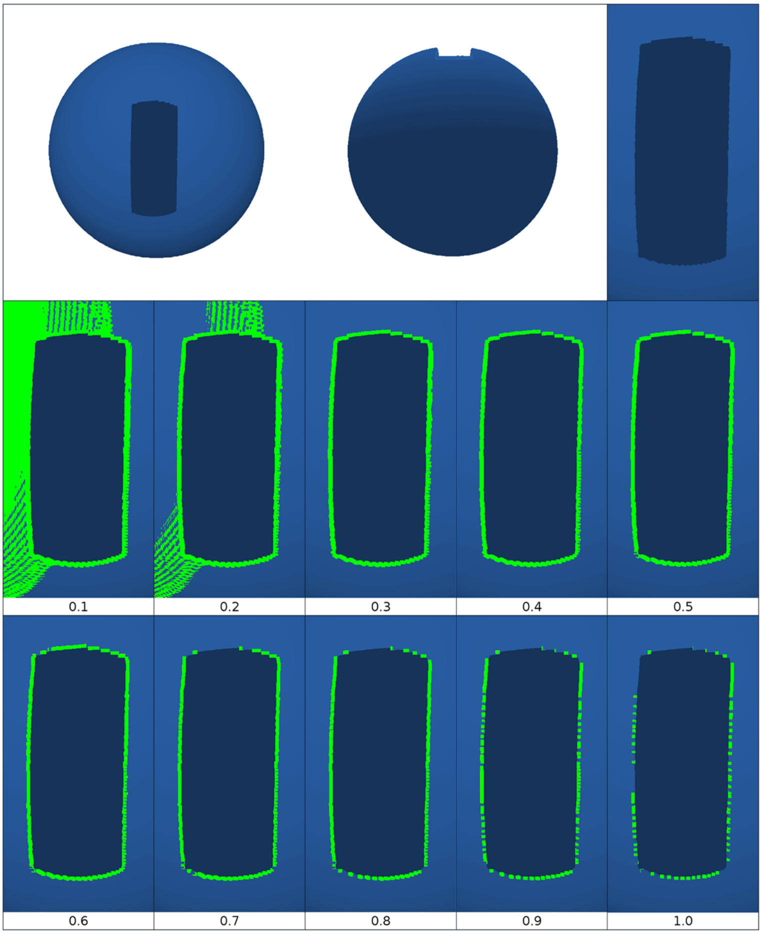

Fig. 44 Results of boundary-point segmentation depending on boundary threshold $\varepsilon_{dist.}$ Corresponding values of parameter (relative to average minimum distance between points) are shown below each view

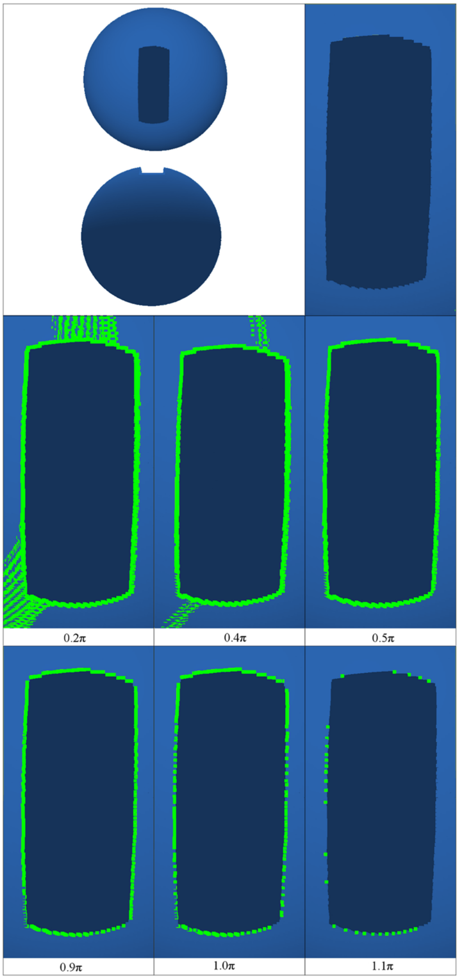

Fig. 45 Boundary-point thresholding depending on angle test. Values under particular cases imply $\varepsilon_{\mathrm{bangle}}$ value in radians

density (see Fig. 15). As the weighting factors for the rectangle and boundary points are equal 1.0 and 1.5 respectively, the influence proportions between those points are 0.4 against 0.6 for boundary points. This makes the patch reproduce the shape. However, due to low sampling on the edges of the rectangle, the method is less sensitive to high-frequency shape distortions. Nevertheless, the general shape was reconstructed correctly.

5.2.4 Comparison to poisson method

In the example of abdominal area cavity, fitting a (2,2) patch with the CRASS method resulted in average reconstruction error varying from 150 to 165% of average error of best Poisson method configuration. In case of a (3,3) patch the reconstruction error ratio raised to 310–339%. In the forearm case the results differed less, some of Poisson reconstruction setups performed better than the proposed algorithm but some worse, the average error of a (2,2) CRASS patch varied from 71% to 115% of average error of Poisson method and for a (3,3) CRASS patch the error varied from 68% to 111% of the Poisson algorithm error. However, Poisson reconstruction featured non-uniform

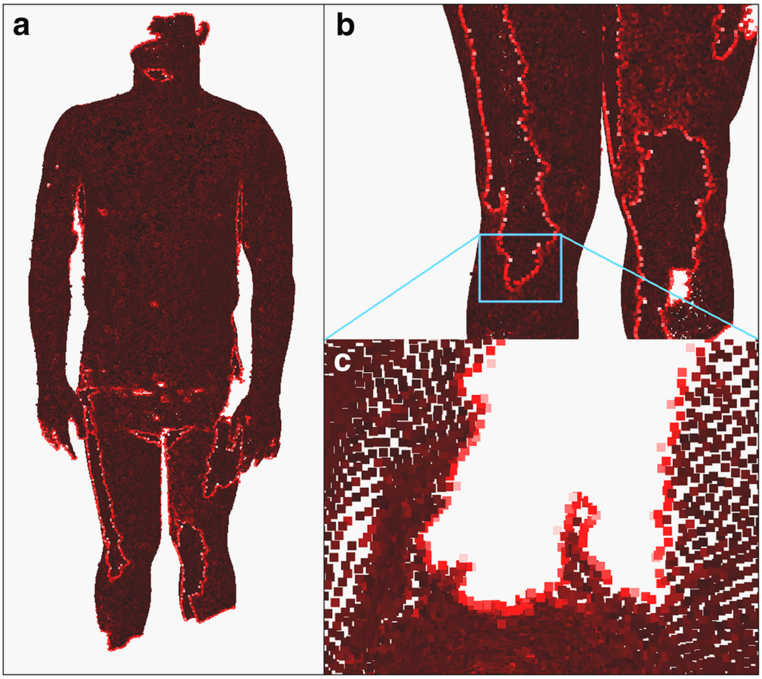

Fig. 46 Maximum spanned angle for points in sample cloud. The color map black-red-white was used, where black corresponds to minimum value, white to maximal one. Values are scaled from minimum to maximum value into lightness channel. Contrast and lightness were decreased by 35% and 60%

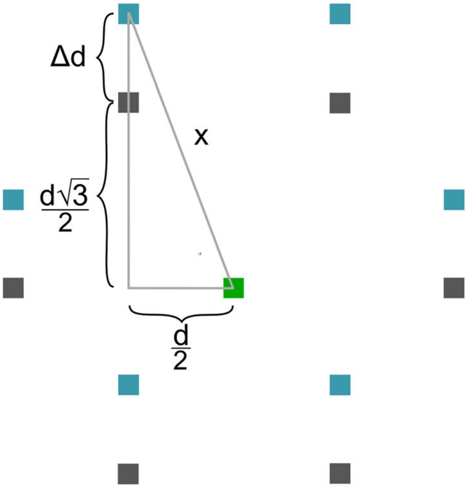

Fig. 47 Distance between points in the case of maximum sampling density flow. Green point is the central point, grey points are neighbor points in the ideal case, blue points are points in the worst case. “d” is the distance between points in the ideal case, Δd is the measurement precision

Table 6 Reference values for parameters and values used in model of human body

Module	Parameter	Reference range	Values used for the model of human body
Boundary points extraction	ε_{dist}	(0.2, 0.6) x AMD	0.5 x AMD
	ε_{bangle}	(0.4 π, 0.9 π) radians	0.48 π
	r_{dist}	(1.25, 1.75) x AMD	1.5 x AMD
	r_{ang}	(1.25, 1.75) x AMD	1.5 x AMD
Cavity bound extraction	$d_{maxExtnd}$	>2.95 x AMD	10 x AMD
	$\phi_{forbidDir}$	(0, 0.17π)	0.048 π
Patch fitting	μ_b	–	1.5
	μ_r	–	1.0
	N	–	2 or 3
	M	–	2 or 3

sampling density, what can be observed in Fig. 42. Sampling density on the strict boundary area was similar to the source cloud, but the further away the more it decreased, regardless of different input parameter values. This causes problems in numerical comparison of both algorithms. One could apply decimation to ensure uniform sampling, but this could influence reconstruction error. Number of generated points differed significantly, in the forearm case about 400 samples for the first Poisson setup against over 2500 samples for the CRASS method. This property influences the visual aspect of reconstruction, where possibly seamless integration of the patch with the original model is required. In such an application, higher numerical error can be accepted when paired with proper sampling density.

5.3 Summary

The CRASS algorithm was implemented and tested on computer-generated data and on models provided by the target measurement system. Usage of pre- and post-processing included in the description of the algorithm is optional, depending on the input-data quality and the user’s goals (visual effect or further analysis). Pre-processing is useful to ensure the meeting of the input data requirements, but it modifies the original data. Post-processing was introduced to improve the visual effect by unifying sampling density and ensuring a smooth blending between the generated patch and the original surface. Therefore, it causes some modifications to the original data close to the cavities.

In the case of running the method without pre- and post-processing, the average fitting error for computer-generated data was approximately 0.35 AMD. In the case of real data, the error was higher, at approximately 0.6 AMD. Factors causing this higher error value were discussed. Pre- and post-processing did not introduce significant reconstruction error, however the final visual result was much better. The final fitting quality was considered to be satisfactory.

Obtained performance was also satisfactory; it did not exceed 7 min for scan sizes of 1 million points on a regular PC-class computer.

5.3.1 Limitations

The developed algorithm produces good results only on point clouds with relatively smooth cavity bounds and homogenous point density. A possible solution was introduced in the pre-processing step, where the sampling density is made more uniform and the surface is smoothed. Another problem emerges when the cavity is located in an area with shape variability of degree higher than 3 but with a low amplitude. Such a small disturbance cannot be reconstructed with Bezier patches of small degree. Increasing the dimension of the patch only causes waviness of the fitted patch as it tries to approximate the fitting points (boundary and surrounding points). The cause of this problem is the property of the Bezier surface itself.

The proposed method cannot perfectly reproduce shapes based on conical curves (e.g., circles), due to the nature of Bezier patches. This problem might be solved by using rational Bezier patches or splines [10].

6 Conclusions and future work

A method for filling cavities in point clouds has been developed. The CRASS algorithm was tested on both computer-generated data and on real scanned data. The real data represented a human body, and consisted of 8-directional measurements obtained using SLI scanners. The results were satisfactory for the presented models; the relative fitting error did not exceed 0.5 AMD except in one unusual case (as discussed in Section 5.2.3).

One of the major issues to solve in the future is to reduce the fitting error for the reconstruction of surfaces with shapes featuring a variability of degree higher than 3 but with a low amplitude. This case partly appeared in the model of a human body (i.e., the abdominal muscle shape). One way to solve this would be to divide the cavity into two separate parts that may be filled with B-spline surfaces, keeping continuity on the single shared edge.

Another area for further development may be altering the approximation principle so that it respects the character of the local surface by using normal vectors from the original surface to find the Bezier patch. This task requires solving nonlinear equations, and is considered an interesting direction for future work.

Some improvement could be also achieved by alternation of UV mapping to make the UV distribution in nearby points on highly curved surface more distinct what should let the solver adapt the patch more accurately in such an area.

Using another type of approximating surface to increase the reconstruction quality could also be considered. Rational Bezier patches or B-splines could enable finer filling of cavities located on spherical surfaces. In the case of the human body, many parts of the surface represent deformed conical sections, so usage of different surfaces could provide better results.

Last but not least, we plan to test our algorithm on a bigger amount of scans from publicly available scanning datasets.

Acknowledgements The work described in this article was part of the project PBS3/B9/43/2015, funded by the National Centre for Research and Development.

Authors' contributions Marta Nowak – 55%.
Jakub Michoński – 25%.
Robert Sitnik – 20%.

Funding This study is partially supported by the project PBS3/B9/43/2015 funded by the National Centre for Research and Development.

Data availability Data will not be available.
Source code will not be available.

Compliance with ethical standards

Competing interests No competing interests.

Appendix

Following script was used to generate Poisson reconstruction results in MeshLab.

```
<!DOCTYPE FilterScript>
<FilterScript>
 <xmlfilter name="Screened Poisson Surface Reconstruction">
  <xmlparam value="0" name="cgDepth"/>
  <xmlparam value="false" name="confidence"/>
  <xmlparam value="10" name="depth"/>
  <xmlparam value="5" name="fullDepth"/>
  <xmlparam value="8" name="iters"/>
  <xmlparam value="4" name="pointWeight"/>
  <xmlparam value="false" name="preClean"/>
  <xmlparam value="1.5" name="samplesPerNode"/>
  <xmlparam value="1.1" name="scale"/>
  <xmlparam value="false" name="visibleLayer"/>
</xmlfilter>
<xmlfilter name="Screened Poisson Surface Reconstruction">
 <xmlparam value="0" name="cgDepth"/>
 <xmlparam value="false" name="confidence"/>
 <xmlparam value="10" name="depth"/>
 <xmlparam value="5" name="fullDepth"/>
 <xmlparam value="8" name="iters"/>
 <xmlparam value="3" name="pointWeight"/>
 <xmlparam value="false" name="preClean"/>
 <xmlparam value="1.5" name="samplesPerNode"/>
 <xmlparam value="1.1" name="scale"/>
 <xmlparam value="false" name="visibleLayer"/>
</xmlfilter>
<xmlfilter name="Screened Poisson Surface Reconstruction">
 <xmlparam value="0" name="cgDepth"/>
 <xmlparam value="false" name="confidence"/>
 <xmlparam value="10" name="depth"/>
 <xmlparam value="5" name="fullDepth"/>
 <xmlparam value="8" name="iters"/>
 <xmlparam value="2" name="pointWeight"/>
 <xmlparam value="false" name="preClean"/>
 <xmlparam value="1.5" name="samplesPerNode"/>
 <xmlparam value="1.1" name="scale"/>
 <xmlparam value="false" name="visibleLayer"/>
</xmlfilter>
<xmlfilter name="Screened Poisson Surface Reconstruction">
 <xmlparam value="0" name="cgDepth"/>
 <xmlparam value="false" name="confidence"/>
 <xmlparam value="10" name="depth"/>
 <xmlparam value="5" name="fullDepth"/>
 <xmlparam value="8" name="iters"/>
 <xmlparam value="1" name="pointWeight"/>
 <xmlparam value="false" name="preClean"/>
 <xmlparam value="1.5" name="samplesPerNode"/>
 <xmlparam value="1.1" name="scale"/>
 <xmlparam value="false" name="visibleLayer"/>
</xmlfilter>
<xmlfilter name="Screened Poisson Surface Reconstruction">
 <xmlparam value="0" name="cgDepth"/>
 <xmlparam value="false" name="confidence"/>
 <xmlparam value="10" name="depth"/>
 <xmlparam value="5" name="fullDepth"/>
 <xmlparam value="8" name="iters"/>
 <xmlparam value="0" name="pointWeight"/>
 <xmlparam value="false" name="preClean"/>
 <xmlparam value="1.5" name="samplesPerNode"/>
 <xmlparam value="1.1" name="scale"/>
 <xmlparam value="false" name="visibleLayer"/>
</xmlfilter>

</FilterScript>
```

References

1. Anguelov D, Srinivasan P, Koller D, Thrun S, Rodgers J, Davis J (2005) SCAPE: shape completion and animation of people. ACM Trans Graph 24:408–416. https://doi.org/10.1145/1073204.1073207
2. Centin M, Pezzotti N, Signoroni A (2015) Poisson-driven seamless completion of triangular meshes. Comput Aided Geom Des 35–36:42–55. https://doi.org/10.1016/j.cagd.2015.03.006
3. Chalmovianský P, Jüttler B (2003) Filling holes in point clouds. In: Wilson MJ, Martin RR (eds) Mathematics of surfaces: 10th IMA international conference, Leeds, UK, September 15–17, 2003. Proceedings. pp 196–212. https://doi.org/10.1007/978-3-540-39422-8_14
4. Cignoni P, Corsini M, Ranzuglia G (2008) Meshlab: an open-source 3d mesh processing system. ERCIM News 73:45–46. https://doi.org/10.2312/LocalChapterEvents/ItalChap/ItalianChapConf2008/129-136
5. Dai A, Nießner M (2019) Scan2Mesh: from unstructured range scans to 3D meshes. In: Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition. https://doi.org/10.1109/CVPR.2019.00572
6. Daněřek R, Dibra E, Öztireli C, Ziegler R, Gross M (2017) DeepGarment : 3D garment shape estimation from a single image. Comput Graph Forum 36:269–280. https://doi.org/10.1111/cgf.13125
7. Davis J, Marschner SR, Garr M, Levoy M (2002) Filling holes in complex surfaces using volumetric diffusion. In: Proceedings. First international Symposium on 3D Data Processing Visualization and Transmission. IEEE Comput. Soc, pp 428–861. https://doi.org/10.1109/TDPVT.2002.1024098
8. De Berg M, Cheong O, van Kreveld M, Overmars M (2008) Computational geometry: algorithms and applications. Springer-Verlag, Berlin Heidelberg
9. Dellaert F, Seitz SM, Thorpe CE, Thrun S (2000) Structure-from-motion without correspondence. In: Proceedings IEEE Computer Vision and Pattern Recognition 2:557–564. https://doi.org/10.1109/CVPR.2000.854916
10. Gallier J (2000) Curves and surfaces in geometric modeling : theory and algorithms, 1st edn. Morgan Kaufmann, San Francisco
11. Gao L, Yang J, Wu T, Yuan YJ, Fu H, Lai YK, Zhang H (2019) SDM-NET: deep generative network for structured deformable mesh. ACM Trans Graph 38:1–15. https://doi.org/10.1145/3355089.3356488
12. Glinkowski W, Sitnik R, Witkowski M, Kocoń H, Bolewicki P, Górecki A (2009) Method of pectus excavatum measurement based on structured light technique. J Biomed Opt 14:044041. https://doi.org/10.1117/1.3210782
13. Guennebaud G, Jacob B (2010) Eigen v3. http://eigen.tuxfamily.org. Accessed 20 June 2018
14. Guo X, Xiao J, Wang Y (2016) A survey on algorithms of hole filling in 3D surface reconstruction. Vis Comput 34:93–103. https://doi.org/10.1007/s00371-016-1316-y
15. Hanocka R, Fish N, Wang Z, Giryes R, Fleishman S, Cohen-Or D (2018) AligNet: partial-shape agnostic alignment via unsupervised learning. ACM Trans Graph 38:1–14. https://doi.org/10.1145/3267347
16. Hanocka R, Hertz A, Fish N, Giryes R, Fleishman S, Cohen-Or D (2019) MeshCNN: a network with an edge. ACM Trans Graph 38:1–12. https://doi.org/10.1145/3306346.3322959
17. Hanocka R, Metzer G, Giryes R, Cohen-Or D (2020) Point2Mesh: a self-prior for deformable meshes. ACM Trans Graph. https://doi.org/10.1145/3386569.3392415
18. Hirshberg DA, Loper M, Rachlin E, Black MJ (2012) Coregistration: Simultaneous alignment and modeling of articulated 3D shape. In: ECCV 2012. Lect Notes Comput Sci 7577:242–255. https://doi.org/10.1007/978-3-642-33783-3_18
19. Kambadakone AR, Sahani DV (2009) Body perfusion CT: technique, clinical applications, and advances. Radiol Clin N Am 47:161–178. https://doi.org/10.1016/j.rcl.2008.11.003
20. Karaszewski M, Sitnik R, Bunsch E (2012) On-line, collision-free positioning of a scanner during fully automated three-dimensional measurement of cultural heritage objects. Rob Auton Syst 60:1205–1219. https://doi.org/10.1016/j.robot.2012.05.005

21. Kazhdan M, Hoppe H (2013) Screened poisson surface reconstruction. ACM Trans Graph 32:1–13. https://doi.org/10.1145/2487228.2487237
22. Kazhdan M, Bolitho M, Hoppe H (2006) Poisson Surface Reconstruction. In: Proc Symp Geom Process pp 61–70. https://dl.acm.org/doi/10.5555/1281957.1281965
23. Lenar J, Witkowski M, Carbone V, Kolk S, Adamczyk M, Sitnik R, van der Krogt M, Verdonschot N (2013) Lower body kinematics evaluation based on a multidirectional four-dimensional structured light measurement. J Biomed Opt 18:56014. https://doi.org/10.1117/1.JBO.18.5.056014
24. Li X, Li X (2010) Filling the holes of 3D body scan line point cloud. In: Proceedings - 2nd IEEE International Conference on Advanced Computer Control, ICACC 2010. 4:334–338. https://doi.org/10.1109/ICACC.2010.5486910
25. Li H, Li Y, Yu R, Sun J, Kim J (2018) Surface reconstruction from unorganized points with l0 gradient minimization. Comput Vis Image Underst 169:108–118. https://doi.org/10.1016/j.cviu.2018.01.009
26. Markiewicz Ł, Witkowski M, Sitnik R, Mielicka E (2017) 3D anthropometric algorithms for the estimation of measurements required for specialized garment design. Expert Syst Appl 85:366–385. https://doi.org/10.1016/j.eswa.2017.04.052
27. Michoński J, Walesiak K, Pakuła A, Glinkowski W, Sitnik R (2016) Monitoring of spine curvatures and posture during pregnancy using surface topography - case study and suggestion of method. Scoliosis Spinal Disord 11:31. https://doi.org/10.1186/s13013-016-0099-2
28. Mostafa Abdel-Bary E (2015) 3D laser scanners' techniques overview. Int J Sci Res 4:323–331
29. Pérez E, Salamanca S, Merchán P, Adán A (2016) A comparison of hole-filling methods in 3D. Int J Appl Math Comput Sci 26:885–903. https://doi.org/10.1515/amcs-2016-0063
30. Quinsat Y, Lartigue C (2015) Filling holes in digitized point cloud using a morphing-based approach to preserve volume characteristics. Int J Adv Manuf Technol 81:411–421. https://doi.org/10.1007/s00170-015-7185-0
31. Saakes D, Yeo H-S, Noh S-T, Han G, Woo W (2016) Mirror mirror: an on-body T-shirt design system. In: Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems. pp 6058–6063. https://doi.org/10.1145/2858036.2858282
32. Tekumalla LS, Cohen E (2004) A hole-filling algorithm for triangular meshes. Sch Comput Univ Utah, UUCS-04-019, UT, USA_ 2.
33. Wang J, Oliveira MM (2003) A hole-filling strategy for reconstruction of smooth surfaces in range images. In: 16th Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI 2003). IEEE Comput. Soc, pp 11–18. https://doi.org/10.1109/SIBGRA.2003.1240986
34. Wang J, Oliveira MM (2007) Filling holes on locally smooth surfaces reconstructed from point clouds. Image Vis Comput 25:103–113. https://doi.org/10.1016/j.imavis.2005.12.006
35. Williams F, Schneider T, Silva C, Zorin D, Bruna J, Panozzo D (2019) Deep geometric prior for surface reconstruction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. https://doi.org/10.1109/CVPR.2019.01037
36. Zaidi H, Ojha N, Morich M, Griesmer J, Hu Z, Maniawski P, Ratib O, Izquierdo-Garcia D, Fayad ZA, Shao L (2011) Design and performance evaluation of a whole-body ingenuity TF PET-MRI system. Phys Med Biol 56:3091–3106. https://doi.org/10.1088/0031-9155/56/10/013
37. Zhou Y, Shen S, Hu Z (2019) Detail preserved surface reconstruction from point cloud. Sensors 19:1–21. https://doi.org/10.3390/s19061278