Point-cloud occlusion recovery using feedforward neural networks

Abstract

3D scenes reconstructed from point clouds, acquired by either laser scanning or photogrammetry, are subject to data voids generated by foreground occluding objects. Modeling with missing data is usually a manual process where human interpretation plays an essential role. This paper presents an artificial neural network (ANN) learning algorithm capable of recovering point cloud occlusions for objects that can be approximated with injective 3D functions. Starting from the point clouds around the occluded parts, a set of single-layer feedforward networks with variable number of neurons (in a predefined range) is trained and validated with a subset of the original cloud decimated using local curvature. The averaged result of the best neural networks is evaluated on a spatial domain which contains the 2D projection of the void, obtaining a 3D point cloud for the occluded volume. Criteria for choosing the number of neurons and the activation function for both hidden and output layers are illustrated and discussed. Results are presented for both simulated (i.e. removing points to create a void) and real occlusions.

Keywords: approximation**,** neural network, occlusion, point cloud

1. Introduction

In the latest two decades, laser scanning technology has become a popular measuring tool in several disciplines such as civil engineering, archeology, architecture, cultural heritage documentation, mechanics, forensics, geology, among the others (Grussenmeyer et al., 2008; Vosselman and Maas, 2010). Nowadays, dense point clouds are used to generate accurate digital reconstructions in different formats such as 2D CAD project drawings (e.g., plans, cross sections, elevations), digital elevation models (DEMs), 3D models based on mesh or NURBS surfaces, and Building Information Modeling. On the other hand, users of laser scanning technology are aware that acquiring dense point clouds with metric integrity is not sufficient to guarantee the completeness of final output. For instance, users of point clouds that scenes with different object textures and materials provide a variable quality of the measured return pulses, specular reflective objects (e.g. mirrors) cannot be captured through point clouds, and transparent materials could result in refracted pulses.

The aim of this paper is to consider one of the most important practical issues during scan acquisition, i.e. occlusions and the consequent lack of data. Laser scanning pulses cannot penetrate opaque objects and 3D coordinates are (usually) measured only for the first reflective surface along the emission direction. Acquiring a single point cloud that reveals the whole scene is not possible in most practical applications. The acquisition of several scans from different station points that are then registered in a common reference system allows users to fill parts of the occlusions, notwithstanding complete scene acquisition (i.e. the whole scene) is rarely achievable. In addition, multiple scans of the same scene increase time for data acquisition and provide redundant data.

Point clouds can also be generated with photogrammetric techniques (Kraus, 2008; Luhmann et al., 2006), which however does not solve the problems related to occlusions (Zhu et al., 2015). Nowadays, the high level of automation of commercial software allows the creation of dense point clouds from large image datasets (Heinrichs et al., 2006; Vu et al., 2009; Pierrot-Deseilligny and Clery, 2011). Data density and metric accuracy are comparable to laser scanning point clouds, although images requires a processing workflow that could be more time consuming than laser scanning, in which point clouds are directly acquired on the field and then registered in the office (Stamos et al., 2008). Complex and detailed surveys often require the combined use of photogrammetry and laser scanning (Guidi et al., 2009; Grussenmeyer et al., 2010).

As described in Chi and Bisheng (2016), previous research work in point cloud occlusion recovery has been carried out by Becker et al. (2009), Cai et al. (2015), Doria and Radke (2012), Friedman and Stamos (2012), and Lozes et al. (2014), among others. The idea was to recover (i.e. to fill) the occluded areas with a synthetic point cloud, i.e. a point cloud that completes the original laser scanning dataset. In this case, recovering means to provide a dense set of 3D points (according to the laser scanner used) for areas that appear as shadowed in a point cloud because of occluding elements.

The approach proposed in this work is an alternative solution that still fills occluded areas through the generation of a synthetic point cloud for the occluded part. The method has been developed through an automatic learning procedure to be able to deal with different scenes and objects. The proposed solution cannot be applied to any kind of occluded objects, but it tries to replicate the work of a human operator who is manually tracing (modeling) from point clouds. An example is showsn in in Fig. 1a, where the vault was captured with multiple laser scans. However, a complete data acquisition of the vault surface is not possible because the object is partially occluded by a metal stiffener. Point cloud recovery becomes a two steps procedure where (i) 3D points of occluding objects are removed and (ii) the corresponding void is filled with new points of the occluded object. In other words, the final result should be a point cloud as close as possible to that achievable by scanning the vault after removing the stiffener.

After acquiring a set of laser scanning in the field, the effective input for digital reconstruction is a set of registered point clouds which captured the metal reinforcement installed on the vault. If the goal is the reconstruction of the entire vault, an expert operator has to identify the occluding objects, delete the corresponding laser points, and complete the reconstruction only with the points measured on the vault. This is similar to an image inpainting problem (Bevilacqua et al., 2017) applied to laser scans rather than an image.

Such approach requires manual (interactive) operations carried out by a human operator expert in 3D modelling, who has to:

- inspect the geometry of the vault;
- identify the rib and understand that is a continuous constructive element occluded by the stiffener;
- trace lines and surfaces taking into consideration such constructive aspects, i.e. the logic of construction of the vault (how the vault has been built).

Fig. 1b shows the results obtained with manual measurements and NURBS lines and surfaces using Rhinoceros. Here, the expert operator was able to understand the geometry of the object and generate a surface model without the stiffener. An alternative more automated solution is the use of software for 3D modeling which provide a mesh from the point cloud. Several software for mesh interpolation and editing are available on the commercial market and could be used to partially automate this task. On the other hand, automation based only on geometric aspects can be the source of gross errors. Indeed, most software for point cloud editing have automated algorithms for hole identification and filling, but they were mainly developed to deal with relatively small holes when compared to the whole object. Fig. 1c shows the results achieved with Geomagic Studio and the aforementioned approach: after generating the mesh the hole can be filled in a fully automated way. However, the achieved result is surely wrong because it does not take into consideration the constructive logic of the vault.

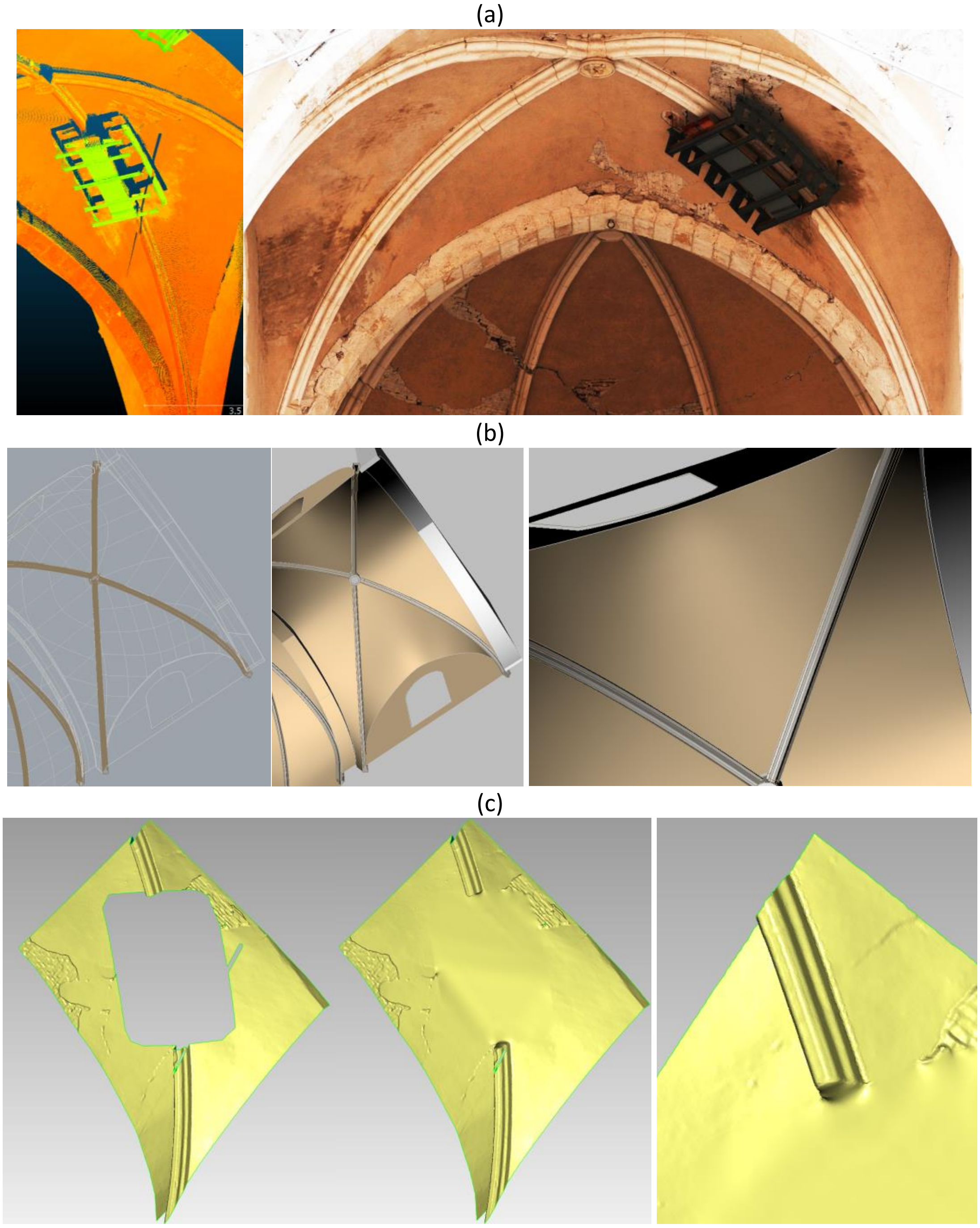


Fig. 1. (a) Original point cloud of the vault with the occluding object (the stiffener), (b) manual modeling carried out by an expert operators, who correctly interpreted the continuity of the rib, (c) mesh generated with Geomagic Studio after removing the occluding object and filling the holes with automated tools for mesh editing.

The proposed approach starts from a simple consideration: the human operator was able to provide a “good” reconstruct of the object because particular attention was paid to the full object. After a visual inspection, the solution was found by combining geometry and (human) interpretation. Can we implement an “intelligent” algorithm that can replicate human interpretation for this particular case study? Can the method be extended to deal with other categories of objects in an automated way?

2. Overview of the proposed approach

Let us suppose that the whole surface of the vault in Fig. 1 (without reinforcement) can be approximated by a function $f: D \to C \subseteq \mathbb{R}$, in which D is the domain and C the codomain. The traditional way to define f is to study the shape of the vault and implement a (static) computer program that generates the required mapping. Such task is not a trivial problem, especially in the case of irregular vaults like those of historic buildings, which have geometric irregularities and anomalies.

We may define the domain as a 2-dimensional space $D = \{(x_{1D}, x_{2D}) \in \mathbb{R}^2 \}$, notwithstanding the method can be extended to incorporate additional data such as laser intensity, point cloud colors, or point normals. In this second case, the *n* input variables have unknown statistical importance which requires sensitivity analysis. The case of multiple input variables is not considered in this paper.

The solution proposed in this work is based on a simple and intuitive consideration: defining the mapping function f and implementing a computer program that generates the mapping algorithm is not only difficult, but also not replicable for other case studies which require specific mapping functions. In other words, dealing with different objects implicate a change of the mapping function. Information on the unknown mapping is available only through punctual values (x_{1m}, x_{2m}, x_{3m}) measured (index *m*) by laser scanning or photogrammetry, for which we are looking for a mapping function that provide a set of correspondences $(x_{1m}, x_{2m}) \rightarrow (x_{3m} + e)$, i.e. an approximate mapping with a "small" error e, comparable with the expected metric accuracy of the final output.

The mapping problem may be cast in the form $f: (D - H) \subseteq \mathbb{R}^2 \rightarrow \mathbb{R}$, where $H = \{(x_{1H}, x_{2H}) \in \mathbb{R}^2\}$ is the set of planar coordinates corresponding to the hole (*H*) generated by occluding elements, and the codomain is related to the x_{3H} coordinate. The function may also be expressed in the more common form $x_{3(D-H)} = f\left(x_{1(D-H)}, x_{2(D-H)}\right)$, which assumes that there is a correlation between the spatial coordinates and $x_{3(D-H)}$. The mapping function replicates the work of the human operator, who studied and modelled the entire vault starting from the values measured only in $D - H$. This means that the information encapsulated in $D - H$ is sufficient to define the mapping f and to fill the hole by simply evaluating the function in H. An important consideration is that the proposed method cannot be used for objects for which an approximated mapping function is not available, such as irregular free-form objects like rock walls. This special case is not only a limitation for the proposed method, but also for traditional manual modeling.

The proposed (automatic) solution that realizes this process without manual intervention is based on a neural network (Haykin, 2008) $N: D \subseteq \mathbb{R}^2 \rightarrow \mathbb{R}$ that has the capability to learn and define the mapping from the measured punctual values (x_{1m}, x_{2m}, x_{3m}). In other words, the algorithm has to learn on its own without the need to explicitly write the code, and the neural network has to approximate the function with an "acceptable" error $||N - f|| < \epsilon$, so that we can say that $N(x_1, x_2)$ is an ϵ-approximated interpolant of the function f, that defines the shape of the object. On the other hand, how can we guarantee that such neural network exists?

Let us consider the problem presented in this paper. A generic neural network with 2 inputs (x_1, x_2) neurons and a single hidden layer is defined as:

$$x_3 = N(x_1, x_2) = \sum_{i=1}^{2} w_i \varphi \left(\sum_{j=1}^{m} a_{ij} x_i + b_i \right)$$

where φ is the activation function, a_{ij} is the weight of the synapses that connects input x_i and the j^{th} hidden neuron, b_j a bias, and w_i is the weight that goes from the j^{th} hidden neuron to the (linear) output unit.

The architecture of the network is a touple $(2, j, 1, W)$ where 2 are input units, j is the number of neurons, 1 the output unit for x_3, and W the set of weights that connect the different units. Every neuron is connected to all neurons of next layer (Fig. 2).

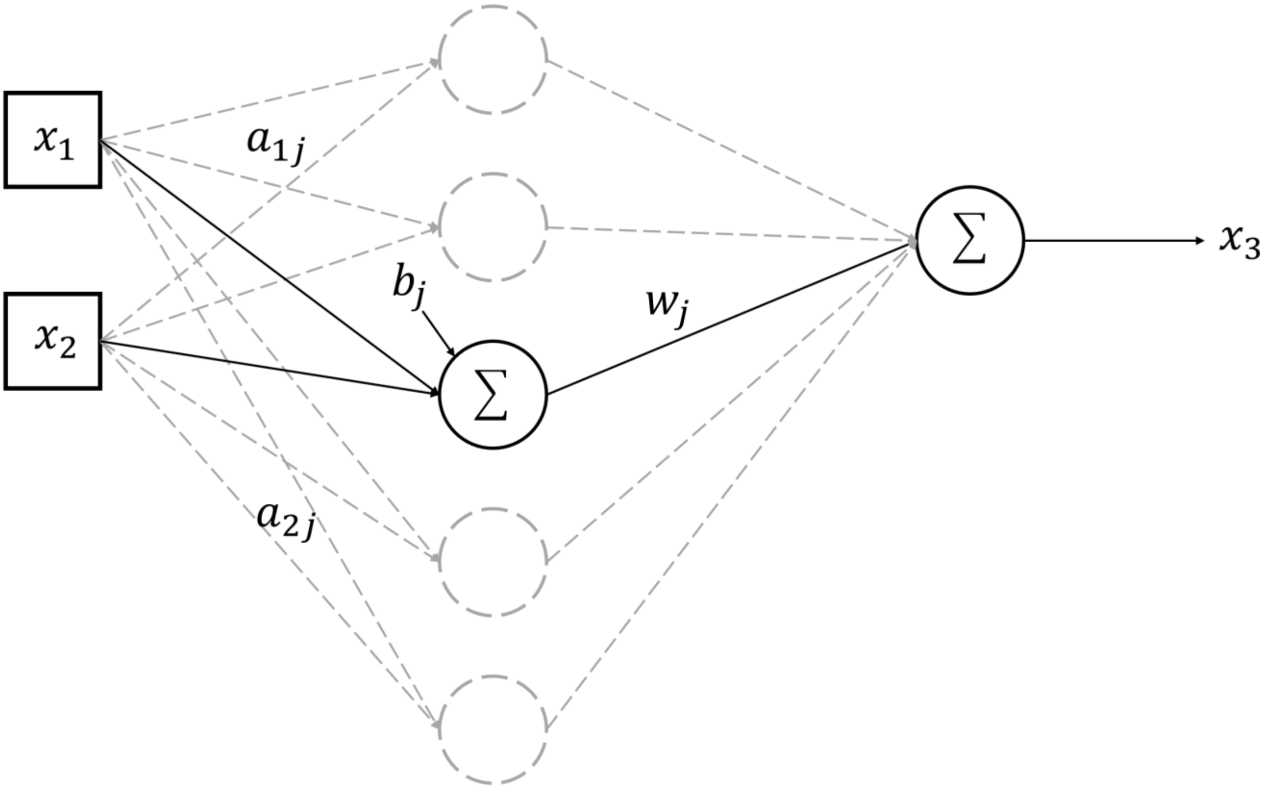


Fig. 2. The scheme of the feedforward network implemented in this work.

The universal approximation theorem (Hornik et al., 1989) provides an important result for the approximation problem. It states that a neural network can approximate an arbitrary continuous function using a single layer with a finite number of neurons and with arbitrary activation functions. For more details the reader is referred to Cybenko (1989), Hornik (1991), and Huang et al. (2006).

An important consideration deserves to to be mentioned: the theorem provides a very attractive result for the approximation of functions with single layer neural networks. On the other hand, the theorem does not provide a solution to estimate the optimal number of neurons in the hidden unit, except that the number is finite. In addition, it does not provide criteria about the algorithmic learnability of those parameters (Tikk et al., 2003). This makes the problem more complicated with serious implications in practical applications.

After setting up the size of the network (number of neurons), samples $(x_{1m}, x_{2m}) \leftrightarrow x_{3m}$ are used to train it. The implementation of the training algorithm is not a trivial choice. For instance, the solution could be a neural network that perfectly fits the training set. On the other hand, this model provides very bad results on the test data with unrealistic solutions. In other words, exact interpolants may result in overfitting.

In the next section the criteria to design the network are presented as well as the methods to quantify how the neural network approximates the measured point cloud. Once the mapping is numerically defined, it can be applied to a set of coordinates $(x_1, x_2) \in H$ to reconstruct the missing part of the occluded part. A flexible learning solution was developed to guarantee the applicability of the same method in other problems, so that there is an internal replicability and adaptability. The method was implemented in the generic form $x_3 = N(x_1, x_2)$ to remove the constraint of spatial (horizontal) and vertical components. In the rest of the paper object such as vaults are used to explain the work carried out, so that the more traditional form $z = N(x, y)$ could be used. However, since mapping between the laser points (x, y, z) and another reference system (x_1, x_2, x_3) can be achieved with a rigid transformation that does not modifies the shape of the object, a 3D roto-translation allows the user to deal with vertical walls or other objects for which the required one-to-one mapping can be guaranteed.

3. Implementation of the learning algorithm

The neural network $N(x_1, x_2)$ is a "black box" that produces an output x_3 from an input vector (x_1, x_2) . Laser scanning points are the teaching input since they provide a set of correspondences $(x_{1m}, x_{2m}) \leftrightarrow x_{3m}$. The error in approximating the measured point can be evaluated applying the neural network to the measured 2D coordinates, obtaining the difference $N(x_{1m}, x_{2m}) - x_{3m}$. In the case of p training points the overall error function during training is defined as $E = \frac{1}{2}\sum_{i=1}^{p}(N(x_{1m}, x_{2m}) - x_{3m})^2$. The more familiar root mean square error is directly related to the error function since $RMS = \sqrt{2E}$ and will be used to quantify the quality of results in the rest of the paper.

The learning process implemented is based only on progressive modifications of the weights, which includes also deleting connections by setting weights to zero. The activation function is set beforehand and no change of the function is performed during the training process.

The numerical estimation of $N(x_1, x_2)$ requires the following operations:

1. identification of the training dataset;
2. selection of the activation functions (for hidden and output layers);
3. selection of the number of neurons in the hidden layer;
4. iterative estimation of the weights.

The identification of the training dataset is apparently a simple task, since the user has to select laser points $(x_{1m}, x_{2m}) \leftrightarrow x_{3m}$, in which we suppose that $(x_m, y_m) \in (D - H)$.

Laser scanning technology provides a huge number of points, for which the computational cost of processing could exceed memory limitations. It is therefore mandatory to identify a new subset of points $(x'_{1m}, x'_{2m}) \leftrightarrow x'_{3m}$ which provide enough information to approximate the original points. Decimation has to be carried out to match the CPU requirements of the implemented machine. For instance, experiments carried out with 5,000 points and an Intel® Core™ i7-5500U with 8 GB RAM required a CPU time less than 2 minutes for training a network with less than 100 neurons. The CPU time became some minutes for 10,000 to 15,000 points with the same computer.

It is quite clear that the proposed number of points requires a strong decimation of the original point cloud, which could be made up of several million points (the average decimation factor was about 1-5% in the proposed case studies). As described in the previous section, one of the requirements for the proposed method is the existence of a function f that defines the surface of the object with an acceptable approximation. Although the function is not known, it is clear that the considered objects should be quite regular, possibly continuous notwithstanding this is not mandatory. Under this assumption, a subset of points is extracted analyzing the curvature of the point cloud.

The training set is additionally divided into a set really used to train (70%), a set for verification (15%), and a set for verification after successful training (15%), which is useful to provide an early stopping criterion when the network approximates the training dataset with an acceptable error. More specifically, we want to minimize the error function $E = \frac{1}{2}\sum_{i=1}^{p}(N(x'_{1m}, x'_{2m}) - x'_{3m})^2$ in an iterative way by initializing the network with random weights that are progressively adjusted to reach the minimum for *E*. The Quasi-Newton method is used in the current implementation, since it provides very fast convergence notwithstanding high memory requirements due to storing the Hessian matrix. It is not the author's intention to provide the mathematical bases for the minimization problem. For more details the reader is referred to Battiti (1992).

As the estimation of neural network parameters is carried out using laser points, defining the optimal configuration with a progressive adjustment of the weights could lead to results that are not consistent with the human perception of the object. After defining the subset of points $(x'_{1m}, x'_{2m}) \leftrightarrow x'_{3m}$ for training, we can generate a neural network that is an exact interpolant $N(x'_{1m}, x'_{2m}) = x'_{3m}$. As mentioned, minimizing the training error could lead to unrealistic solutions. Generating an exact interpolation through a neural network would be a mistake since the minimization does not take into account the noise of laser measurements. Exact approximation is possible by increasing the number of neurons (De Jesús and M.T. Hagan, 2001; Horn et al., 2009), whereas limiting the number of neurons is correlated to the plasticity of the network, i.e. the degrees of freedom, i.e. the number of weights. Increasing the number of weights allows one to reduce the error, and vice versa. It is well-known that there is no universal rule to define a priori the structure of the network. In the next section the used criteria and experiments to establish the number of neurons and the "best" activation function are illustrated and discussed.

4. Defining activation functions and number of neurons

Some experiments were carried out to test the performance of the developed algorithm on real and simulated case studies. This is also motivated by the intrinsic difficulties in defining network's architecture, for which there is no solution and the architecture is always decided a-priori by the user. Experiments were carried out with different datasets, but only one is illustrated in this paper because the proposed dataset is considered quite representative of the different results obtained.

The dataset is a historic umbrella vault with a size of 4,5 m x 4,2 m x 0,4 m (Fig. 3, top), which was scanned with a Faro Focus 3D laser scanner. The vault is not regular, i.e. the different surfaces forming the vault are different. For this test, no occlusion was generated since the goal was to test the performances of the neural network in approximating the points of an irregular continuous surface. The original dataset was then decimated according to the curvature obtaining 4,062 training samples $(x'_{1m}, x'_{2m}) \leftrightarrow x'_{3m}$. Then, 500 neural networks with different activation functions were trained.

The activation functions tested for both hidden and output layers are identity, sigmoid, hyperbolic tangent, exponential and sine functions, which were combined so that it is possible to have different activation functions for hidden and output layers. The developed strategy fixes the activation functions in a random way, whereas the number of neurons was randomly chosen between 60 and 70 in the different experiments. The reason of this choice will be clarified at the end of this section.

After the simulation, the estimated networks were sorted out using the estimated RMS on the points used as validation, which varied from ±0.007 mm to ±0.154 mm. The networks with worst results were those in which an exponential activation function was used for the output unit. Inaccurate results were also found when the activation function of the hidden layer was the identity function. To understand which function combination provided the best results, different combinations of activation functions were grouped and only the first 50 networks were kept. Neural networks were sorted out using the RMS value, which was better than ±0.011 mm for the first 50 networks. A graph with the results for the different combinations is shown in Fig. 3.

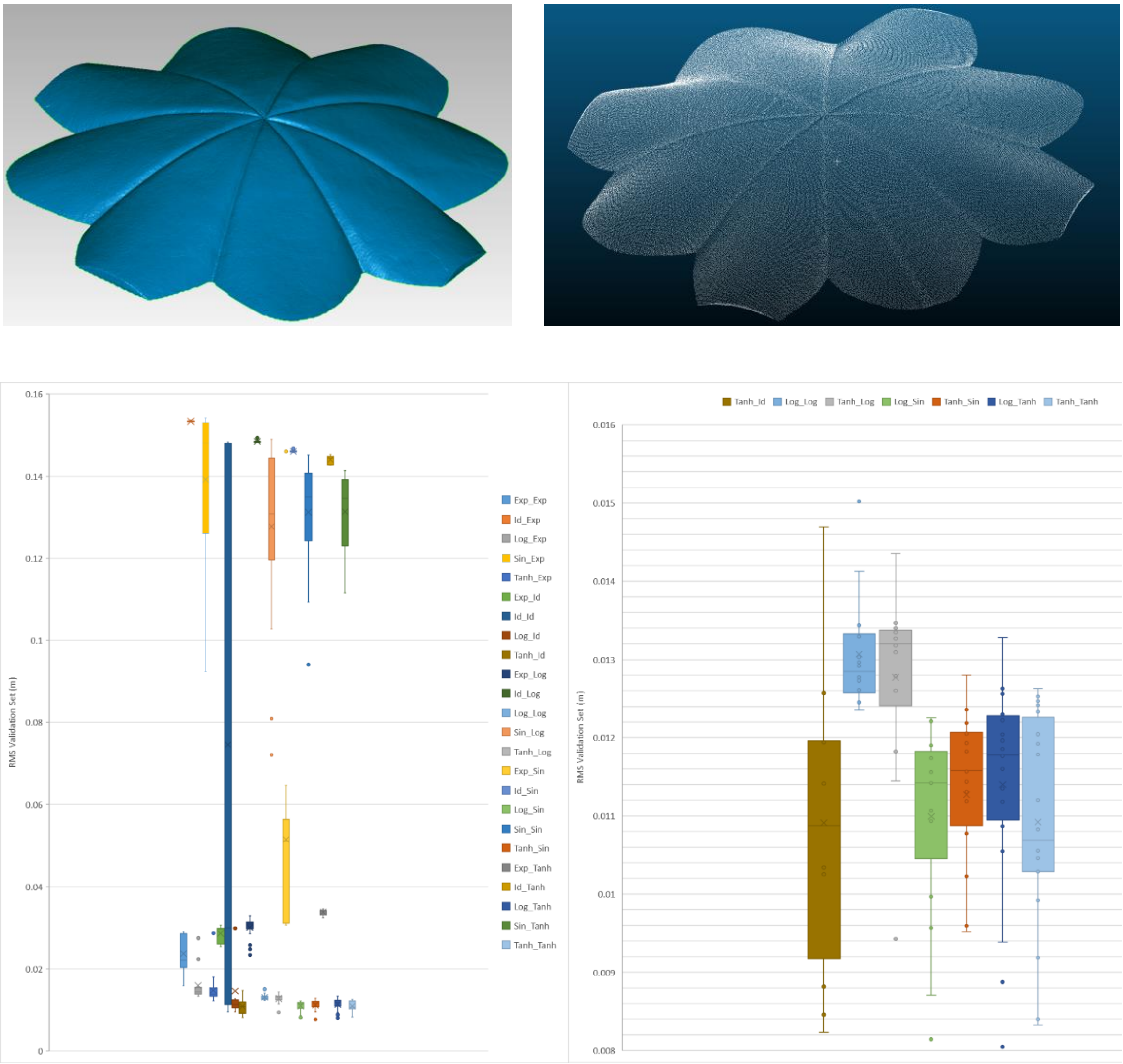


Fig. 3. Mesh from the original point cloud and decimated dataset used as input for training 500 neural networks (top). Results with different combinations functions (left) and a detail for the best combinations for input and output layers (right).

The best result was achieved when hyperbolic tangent $\varphi(z) = (e^z - e^{-z})/(e^z + e^{-z})$ is used for both hidden and output layers, for which 11 networks were found among the first 50. Results with the hyperbolic tangent for both layers provided good results also in other experiments with different point clouds. For this reason, hyperbolic tangent will be the activation function used in the rest of the paper.

The second test was carried out with the same dataset (and the hyperbolic tangent) but varying the number of neurons, from 10 to 100. In this case, 1000 neural networks were computed and sorted out as a function of the RMS of validation set. Fig. 4 shows the obtained results. As can be seen, neural networks with 10-20 neurons have large errors, whereas after 50 neurons an increment of network size does not give a significant improvement in terms of metric accuracy. As mentioned, it is not simple to define a universal criterion for the number of neurons not only for the proposed application, but also when dealing with neural networks in general. The choice used in the rest of the paper is to choose a number of neurons in the range $M_{min} = round\left(\sqrt{p}\right) - 3$ and $M_{max} = round\left(\sqrt{p}\right) + 2$. Then, 20 networks are automatically trained with a random number of neurons in this range and only the best 5 are kept. The estimate is given by the average of the best 5 neural networks in terms of RMS.

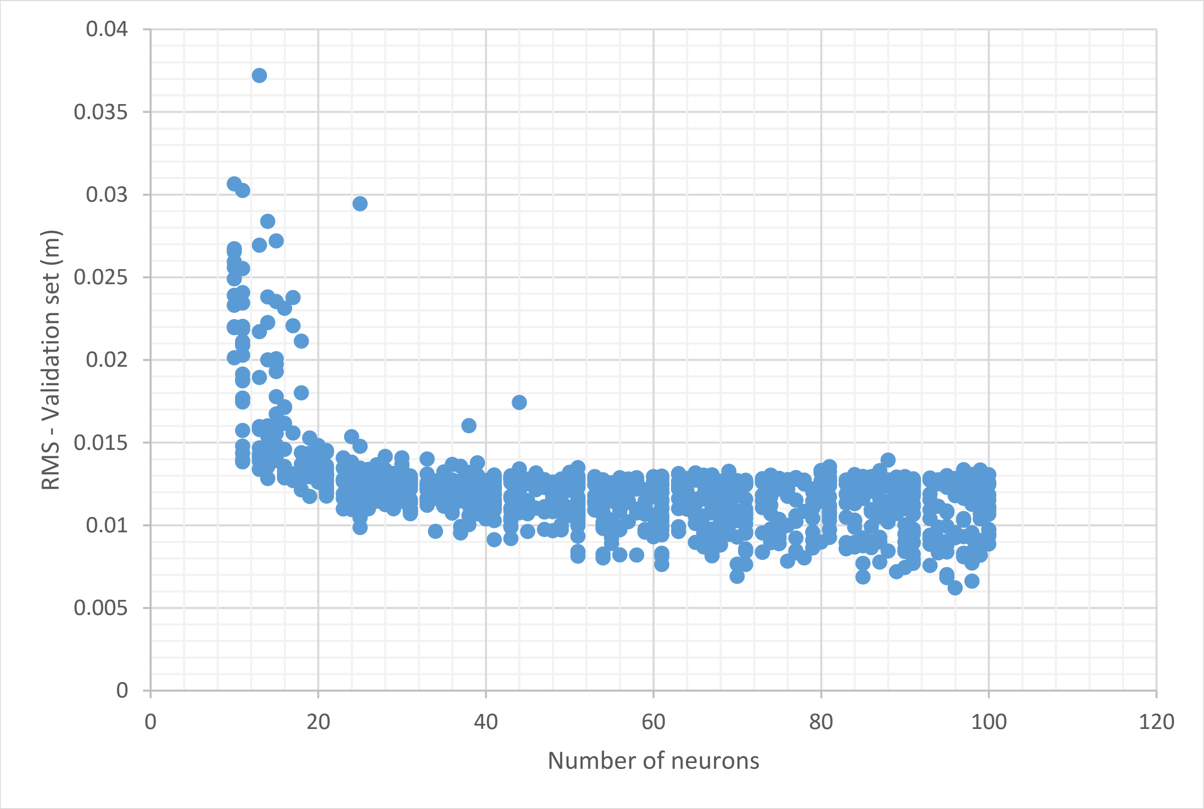


Fig. 4. The RMS of validation sets for 1000 neural networks with hyperbolic tangent as activation functions and a variable number of neurons.

Fig. 5 shows the results of the 5 averaged networks with hyperbolic tangent as activation function compared to the original point cloud. The network was evaluated on the discrete domain D which contains the projection of the vault, in which point spacing was set to 0.01 m to generate a (horizontal) grid for both x and y directions. The comparison with CloudCompare is therefore based on the point-to-point distance between the original point clouds and the synthetic one generated through the neural network. The software provided a mean distance of 0.007 mm and a standard deviation of 0.006 mm. Such results are consistent with the validation error in terms of RMS after the training phase (less than 0.01 cm).

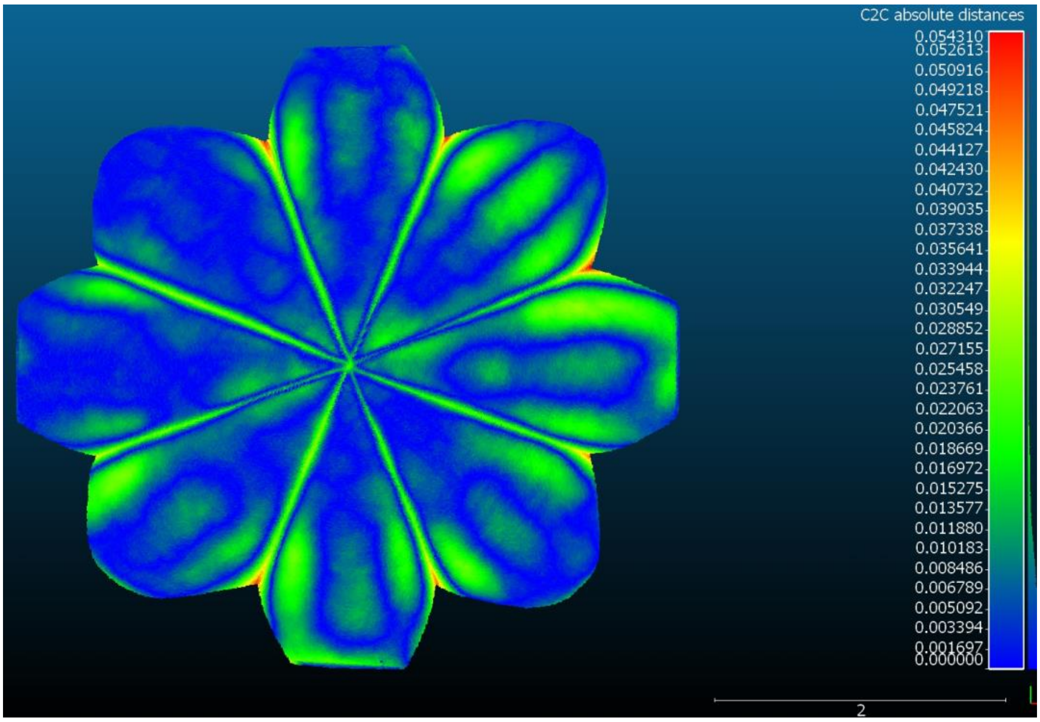
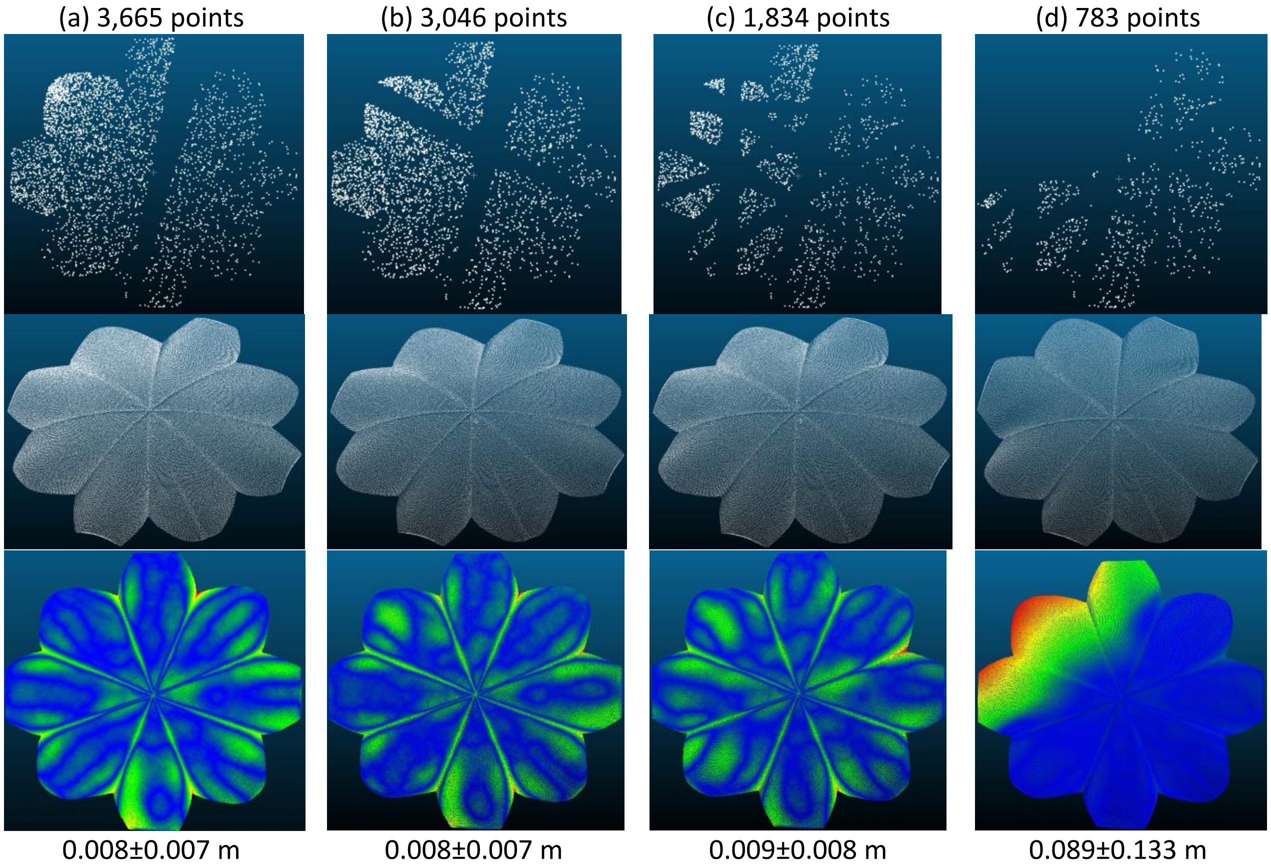


Fig. 5. Comparison between the original point cloud and the average of the 5 best networks.

5 How well the averaged neural networks approximate real objects?

The previous example allowed us to choose the activation function and establish a criterion for the number of neurons in the hidden unit. Results were quite satisfactory in terms of metric approximation, with an accuracy of about 0.01 m. On the other hand, the training dataset is made up of points decimated according to the curvature of the vault, therefore the distribution is optimal for the proposed algorithm, for which no occlusion was introduced. What happens when some parts are occluded and no information for the holes is available during the training phase?

Tests were carried out after removing some points from the training set to simulate different occlusions. In the first case (Fig. 6a) a central part of the vault was completely removed. A progressive point removal was carried out with a second cut (Fig. 6b) and a random decimation in specify areas, significantly reducing the number of points (Fig. 6c). In the last case (Fig. 6d) a significant part of the vault was completely removed. The second row of the figure shows the reconstruction obtained for the different cases. It was decided to create a domain that follows the planar (xy) projection of the vault, in which a set of points was created using a density of 0.01 m x 0.01 m. The neural network was then evaluated for these points and compared to the original point cloud from laser scanning. The comparison was carried out with CloudCompare. A visualization and the corresponding statistics are shown in the two last rows of the figure. As can be seen, the neural network was able to reconstruct cases (a), (b) and (c) with a very good metric accuracy, notwithstanding the strong simplification and cuts carried out. In the last case (d), the network was not able to approximate with a sufficient metric accuracy the part that was removed. It is interesting to see that the network was able to learn the geometry of the umbrella vault assuming a quite symmetric behavior. On the other hand, the vault is not regular (symmetric) and this resulted in a large error. This is quite similar to the error that a human operator would make when he tries to reconstruct the vault without sufficient metric information.



(a) 3,665 points

(b) 3,046 points

(c) 1,834 points

(d) 783 points

0.008±0.007 m

0.008±0.007 m

0.009±0.008 m

0.089±0.133 m

Fig. 6. Comparison between the original point cloud and the vault with some simulated occlusions.

6. Experiments with real occlusions

Different tests were carried out with laser scanning datasets acquired with a Faro Focus 3D. The case studies are vaults were occlusions where found after scan acquisition. Shown in Fig. 7a and 7b are the results for an irregular barrel vault. The vault is located in Castel Masegra, a medieval castle in Italy. As mentioned, the vault is quite irregular and is characterized by the uniqueness of its shape, for which basic primitives (e.g. a cylinder) cannot be used to approximate the surface when accurate reconstructions are required.

Occlusions during scan acquisition where generated by a cable and a chandelier, resulting in a hole formed by two lines and an elliptical occlusions. The point cloud was decimated to 3,595 points to allow rapid estimation of network parameters. 20 neural networks with a variable number of neurons in the hidden layer (from 57 to 62) were estimated and the best 5 networks were selected. Results on a spatial domain that contains the planar (xy) projection of the vault were evaluated through a regular grid of 0.01 m x 0.01 m. The calculation was carried out using the average of the 5 best networks. The result is the point cloud visible in Fig. 7c, in which occlusions were filled with a synthetic point cloud.

An evaluation of metric accuracy is not possible for the occluded areas because other scans were not available. A criteria to check the quality of the results for real case studies is to compare the results of the network with the original (not decimated) point cloud, only for the parts that are not occluded. Results with CloudCompare provided a discrepancy of 0.006±0.003 m (Fig. 7d), which means that the neural network is a good approximation of the original point cloud.

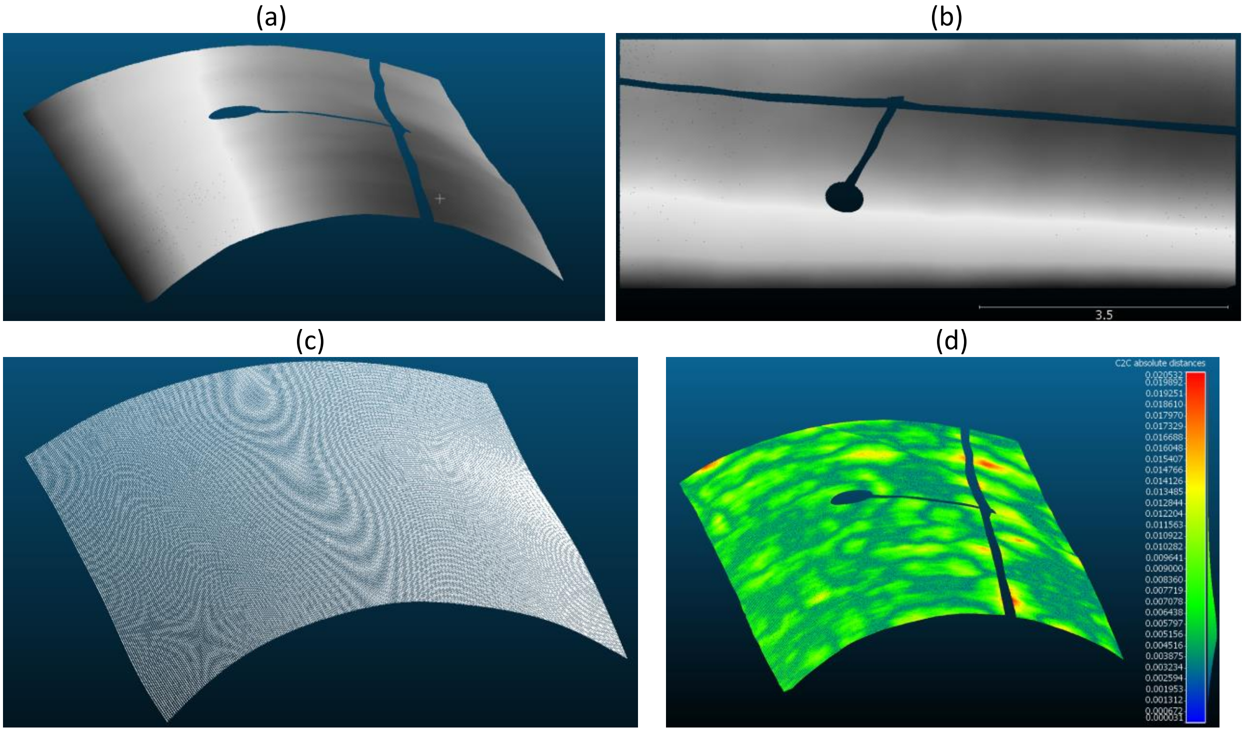


Fig. 7. Results for a historic barrel vault: (a) and (b) provide a visualization of the original point cloud and the occlusion caused by a wire and a chandelier; (c) illustrates the recovered point cloud evaluated on a 0.01 m x 0.01 m grid; (d) provides a comparison between original and recovered point clouds, obtaining a final discrepancy of 0.006±0.003 m.

The second case study is a cross vault in the Basilica di Sant'Ambrogio in Milan (Italy). Different scans were acquired to capture the vault but some occlusions were found after inspecting scans in the office, as shown in Fig. 8a and 8b. Occlusions were mainly generated by other architectural elements, such as the columns in the cloister of the Basilica. 20 neural networks were trained like in the previous case study, using 4,561 points and a variable number of neurons (from 65 to 69). Data processing took less than 3 minutes and the results of the best 5 neural networks (average value) where used to create a new point cloud on a spatial regular grid of 0.01 m x 0.01 m (Fig. 8c). The final discrepancy estimated with CloudCompare between the averaged results and the original point cloud is 0.006±0.003 m (Fig. 8d).

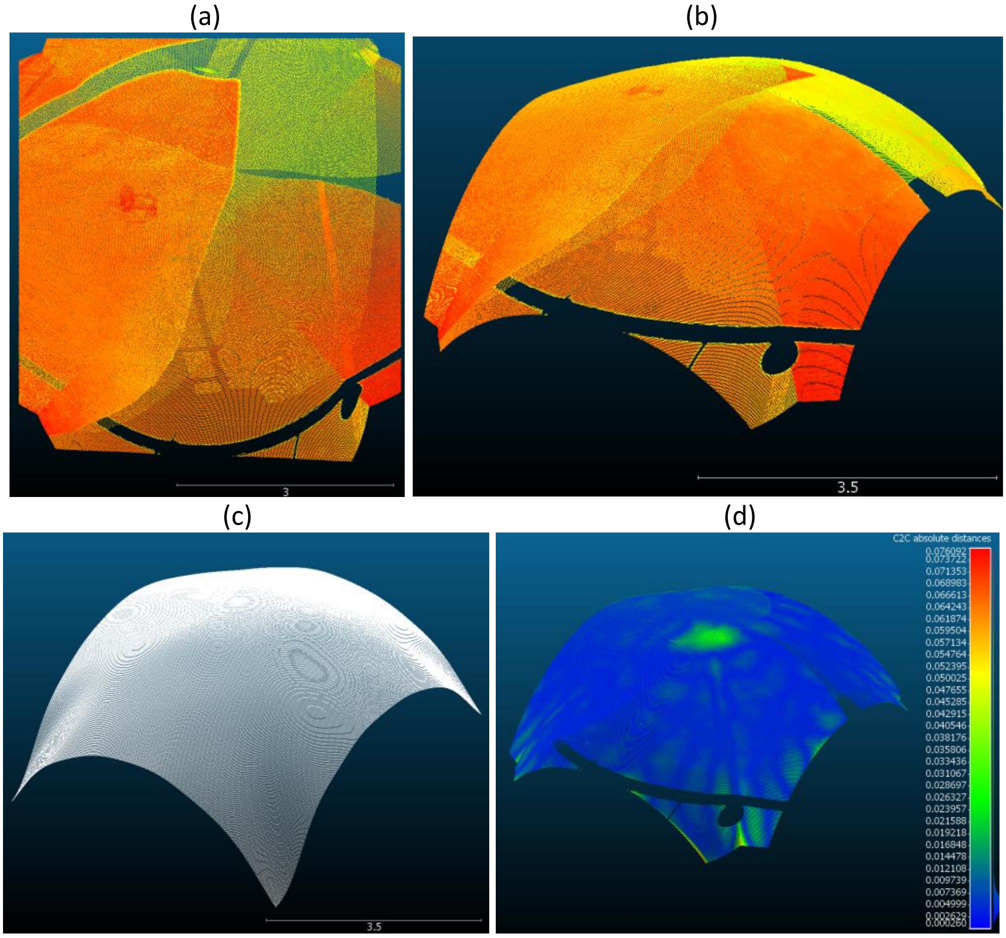


Fig. 8. Results for a historic cross vault: (a) and (b) provide a visualization of the original point cloud; (c) illustrates the recovered point cloud evaluated on a 0.01 m x 0.01 m grid; (d) provides a comparison between original and recovered point clouds, obtaining a final discrepancy of 0.006±0.003 m.

The final example is the vault in the Basilica di Collemaggio presented at the beginning of the paper (Fig. 9a). After deleting the point on the metal stiffener (Fig. 9b), the dataset consists in a point cloud made up of 182,083 points which were decimated to 9,104 points to leverage data processing. Network parameter computation required less than 5 minutes and the results of the best 5 neural networks (average value from 20 networks) were used to create a new point cloud on a spatial regular grid of 0.01 m x 0.01 m (Fig. 9c). Fig. 9d shows a mesh obtained with the synthetic point cloud. A visual comparison with the mesh obtained in Fig.

1 demonstrates how the network was able to capture the geometry of the vault, including the rib. The final discrepancy estimated with CloudCompare between the averaged results and the original point cloud is -0.0004±0.006 m (Fig. 9d). This result has to be compared with the manual reconstruction obtained in Fig. 1 and the automatic one via mesh-based interpolation and hole filling.

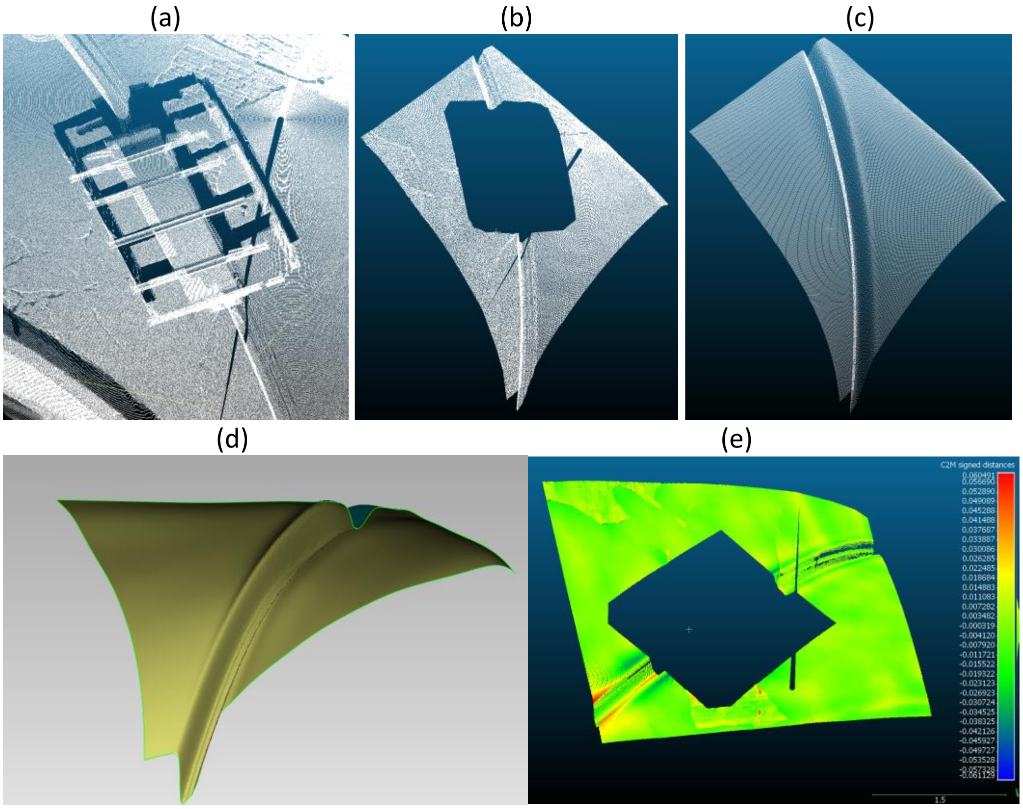


Fig. 9. (a) The original point cloud of the vault in Fig. 1, (b) the same point cloud without the metal stiffener, (c) the recovered point cloud with the proposed method, (d) the mesh generated from the recovered point cloud, and (e) a comparison between recovered and original point clouds, which provided a discrepancy of -0.0004±0.006 m.

Conclusions

This paper presented an artificial neural network (ANN) learning algorithm able to recover occlusions in point clouds. The method was developed to deal with objects that can be approximated with an injective function, which is approximated with a set of single-hidden layer feedforward neural networks trained and validated with a subset of the original point cloud decimated according to local curvature. The method proposed in this paper is based on computer algorithms that have the capability to learn, so that human interpretation in 3D modeling is (partially) substituted by artificial intelligence. The best neural networks are then evaluated and averaged on the spatial domain corresponding to the projection of the occlusion, obtaining a synthetic point cloud that fills the hole.

The chosen network architecture has a single layer with a number of neurons similar to $\sqrt{p}$, i.e. related to the square root of the number of points of the decimated dataset. Hyperbolic tangent was chosen as activation function for both hidden and output layers because of the better results achieved during tests with different activation functions. Tests carried out with simulated and real datasets demonstrated that this allows the neural network to approximate the original object surface without requiring a significant CPU time, which was of a few minutes for the proposed examples. Results highlighted that the neural network can learn the geometry of the object better than existing algorithms for hole filling in a mesh.

The aim of the work was to provide a synthetic point cloud also for occluded areas using a procedure that can be extended to different objects. This replicates the work of a human operator that tries to model the object starting from the points captured in a domain with an internal hole. The proposed examples are different kind of vaults of historic buildings, so that geometric irregularities and anomalies are expected. Future work will be carried with other objects.

Limitations of the proposed method consists in the need of objects that can be reasonably approximated with functions, so that fully 3D bodies cannot be analyzed as a whole, unless they are preliminary split into smaller portions which fulfil the previous criterion. The proposed examples were always vaults of historic buildings, so that they have geometric properties but at the same time they are quite irregular. The comparison between the original point cloud and the recovered one is the quantitative parameter used to assess the quality of the model. Such evaluation was performed for the real objects surveyed with laser scanning (or photogrammetry) only for the parts without occlusions. This is the same approach used by a human operator who reconstructs the surface with manual operations for 3D modeling.

Results with synthetic occlusions were instead used to test the method for sets of points in which the real shape was entirely captured in the point cloud. Here, some occlusions were manually created by deleting points, and then comparing neural network results to the point excluded from data processing.

Future work consists in the application of the proposed approach to other categories of object and in the extension of the methods to full 3D bodies, which cannot be reconstructed with the proposed approach at the moment.

References

Battiti, R., 1992. First and second order methods for learning: Between steepest descent and Newton's method. Neural Computation, Vol. 4, No. 2, pp. 141–166.

Becker, J., Stewart, C., Radke, R.J., 2009. LiDAR inpainting from a single image. In: IEEE International Conference on Computer Vision Workshops (ICCV Workshops), Kyoto, Japan, pp. 1441–1448.

Bevilacqua, M., Aujol, J.-F., Biasutti, P., Brédif, M., Bugeau, A., 2017. Joint inpainting of depth and reflectance with visibility estimation. ISPRS Journal of Photogrammetry and Remote Sensing, 125, pp. 16-32.

Cybenko, G., 1989. Approximation by Superposition of a Sigmoidal Function, Mathematics Control Signals Systems, Vol. 2, pp. 303-314.

Cai, Z., Wang, C., Wen, C., Li, J., 2015. 3D-PatchMatch: an optimization algorithm for point cloud completion. In: 2015 2nd IEEE International Conference on Spatial Data Mining and Geographical Knowledge Services (ICSDM), Fuzhou, China, pp. 157–161.

Chi, C., Bisheng, Y., 2016. Dynamic occlusion detection and inpainting of in situ captured terrestrial laser scanning point clouds sequence. ISPRS Journal of Photogrammetry and Remote Sensing, 119, pp. 90–107.

De Jesús, O., Hagan, M.T., Backpropagation Through Time for a General Class of Recurrent Network. Proceedings of the International Joint Conference on Neural Networks, Washington, DC, July 15–19, pp. 2638–2642.

Doria, D., Radke, R.J., 2012. Filling large holes in LiDAR data by inpainting depth gradients. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 Providence, RI, USA, pp. 65–72.

Friedman, S., Stamos, I., 2012. Online facade reconstruction from dominant frequencies in structured point clouds. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW) Providence, RI, USA, pp. 1–8.

Grussenmeyer, P., Landes, T., Voegtle, T. And Ringle, K., 2008. Comparison methods of terrestrial laser scanning, photogrammetry and tacheometry data for recording of cultural heritage buildings. International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, 37, Part B5: 213-218.

Grussenmeyer, P., Hanke, K., 2010. Cultural Heritage Applications. In: G. Vosselman, H.-G. Maas (eds.), Airborne and Terrestrial Laser Scanning. Whittles Publishing, Dunbeath, Caithness (Scotland-UK), pp. 271-290.

Guidi, G., Remondino, F., Russo, M., Menna, F., Rizzi, A., Ercoli, S, 2009. A multi-resolution methodology for the 3d modelling of large and complex archaeological areas. International Journal of Architectural Computing, 7(1), 40-55.

Haykin, S. (2008). Neural Networks and Learning Machines. Third Edition. Pearson, Prentice Hall, 936 pages.

Heinrichs, M., Rodehorst, V., Hellwich, O., 2007. Efficient Semi-global matching for trinocular stereo. International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences 36 (Part 3/W49A), 185-190.

Horn, J.M., De Jesús, O., Hagan, M.T., 2009. Spurious Valleys in the Error Surface of Recurrent Networks - Analysis and Avoidance. IEEE Transactions on Neural Networks, Vol. 20, No. 4, pp. 686-700.

Hornik, K., 1991. Approximation capabilities of multilayer feedforward networks. Neural Network, vol. 4, pp. 251–257.

Hornik, K., Stinchcombe, M., White, H., 1989. Multilayer feedforward networks are universal approximators, Neural Networks, Volume 2, Issue 5, pp. 359-366.

Huang, G.B., Chen, L., Siew, C.K. (2006). Universal approximation using incremental constructive feedforward networks with random hidden nodes. IEEE Transactions on Neural Networks, 17(4), pp. 879-892.

Kraus, K., 2008. Photogrammetry: Geometry from Images and Laser Scans. Second edition .Walter de Gruyter. 459 pages.

Lozes, F., Elmoataz, A., Lezoray, O., 2014. Partial difference operators on weighted graphs for image processing on surfaces and point clouds. IEEE Trans. Image Process. 23 (9), 3896–3909.

Luhmann, T., Robson, S., Kyle, S., Harley, I., 2006. Close Range Photogrammetry: Principles, Techniques and Applications. John Wiley & Sons, New York, 510 pages.

Pierrot-Deseilligny, M., Clery, I., 2011. APERO, an open source bundle adjustment software for automatic calibration and orientation of set of images. International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences 38 (Part 5/W16), pp. 8 (on CDROM).

Stamos, I., Liu, L., Chen, C., Wolberg, G., Yu, G., Zokai, S., 2008. Integrating automated range registration with multiview geometry for the photorealistic modelling of large-scale scenes. International Journal of Computer Vision 78(2-3): 237-260.

Tikk, D., Kóczy, L.T., Gedeon, T.D. (2003). A survey on universal approximation and its limits in soft computing techniques, International Journal of Approximate Reasoning 33(2), pp. 185-202.

Vosselman, G., Maas, H.-G., 2010. Airborne and Terrestrial Laser Scanning. Whittles Publishing, Dunbeath, Caithness, Scotland, United Kingdom.

Vu., H. H., Keriven, R., Labatut, P., Pons, J.-P., 2009. Towards high-resolution large-scale multi-view stereo. Proc. IEEE Conf. CVPR'09, pp. 1430-1437.

Zhu Z., Stamatopoulos, C., Fraser, C.S., 2015. Accurate and occlusion-robust multi-view stereo. ISPRS Journal of Photogrammetry and Remote Sensing, 109, pp. 47-61.