

VAULT MODELING WITH NEURAL NETWORKS

L. Barazzetti, M. Previtali

Politecnico di Milano, Department of Architecture, Built Environment and Construction Engineering,
Via Ponzio 31, 20133 Milano, Italy
(mattia.previtali, luigi.barazzetti)@polimi.it

Commission II

KEY WORDS: Automation, Machine learning, Modeling, Neural Network, Point cloud, Vault

ABSTRACT:

Nowadays, the digital reconstruction of vaults is carried out using photogrammetric and laser scanning techniques able to capture the visible surface with dense point clouds. Then, different modeling strategies allow the generation of 3D models in various formats, such as meshes that interpolates the acquired point cloud, NURBS-based reconstructions based on manual, semi-automated, or automated procedures, and parametric objects for Building Information Modeling. This paper proposes a novel method that reconstructs the visible surface of a vault using neural networks. It is based on the assumption that vaults are not irregular free-form objects, but they can be reconstructed by mathematical functions calculated from the acquired point clouds. The proposed approach uses the point cloud to train a neural network that approximates vault surface. The achieved solution is not only able to consider the basic geometry of the vault, but also its irregularities that cannot be neglected in the case of accurate and detailed modeling projects of historical vaults. Considerations on the approximation capabilities of neural networks are illustrated and discussed along with the advantages of creating a mathematical representation encapsulated into a function.

1. INTRODUCTION

Laser scanning and photogrammetric point clouds can capture the visible surface of vaults (intrados) with a high level of detail. This method reveals the geometric irregularities of the surface, making geometric survey a powerful non-destructive inspection and analysis tool in conservation and restoration projects.

3D models generated from point clouds can be delivered in different formats. Photogrammetric software usually provides a mesh interpolated from the original point clouds, which is then textured to produce 3D models with color information and digital orthophotos. Similar results can be obtained with laser scanning point clouds, notwithstanding the quality of color information is not comparable with the one achievable from images.

Meshes generated with the automatic interpolation of point clouds have some disadvantages, especially when they have to be used for conservation purposes. The first issue is related to the size of the model, which can be made up of several million triangles, resulting in heavy models. An additional (probably more important) issue concerns the interpolation algorithms that do not consider the different elements of the vault. This gives a single surface that does not follow the logic of construction of the vault.

An alternative solution for vault modeling consists of the use of software for modeling with NURBS surfaces. Examples are Rhinoceros and Autodesk Maya. Commercial software allows the user to import point clouds and generate 3D surfaces with manual, semi-automatic, and automatic algorithms. The user plays a fundamental role in the modeling process. He/she can identify the different structural elements and separate the different surfaces, obtaining a model that is not only a "pure" geometric representation, but it also includes the interpretation carried out to understand better how the vault is built. After defining the generative geometry and modeling the discontinuities between constructive elements, modeling can become a more automatic procedure using both edges (generally

NURBS curves) and point clouds. A surface is fitted to the point cloud using curves as boundaries to obtain a perfect element-to-element correspondence in the final model.

Meshes and NURBS surfaces are powerful tools to reconstruct the visible surface of the vault. However, what happens for those elements that cannot be surveyed with point clouds? This could be the case of the internal structure of the vault as well as the extrados that (in some cases) it is not visible. Such problems are relevant in the case of BIM projects, which cannot be limited to the visible surfaces. BIM requires 3D objects with the opportunity to add information about the internal structure (layers) of the considered architectural elements. In other words, a vault for BIM must become a sort of "solid" object with a thickness. The advantage of BIM consists in parametrization procedures that allows one to modify the shape of the modeled elements without redrawing. In the case of a vault with unknown thickness, the extrados can be generated using an offset-based parametrization of the intrados. This is not a static solution since changing the value of the thickness in the BIM database modifies the 3D representation.

The solution proposed in this paper relies on the reconstruction of the vault intrados with a neural network, which is trained using point clouds. The network is able to learn and approximate the geometry of the vault. It is obvious that geometry plays a fundamental role when dealing with vaults. Vaults can be intended as a "combination" of basic geometric primitives (cylinders, planes, spheres, etc.) with an additional aspect that cannot be neglected in the case of historic vaults: the irregular shape, i.e. the deviation from the ideal geometry made up of geometric primitives.

Laser scanning and photogrammetric point clouds are able to capture the visible surfaces with millimeter-level accuracy, so that they can reveal the actual shape. Let us suppose that the surface of a "perfect" vault (i.e., the vault is made up of just a combination of basic primitives) is described by a function $p = p(\mathbf{x})$, where \mathbf{x} is a vector. We can assume that the geometric irregularities can be modeled by a second function $d = d(\mathbf{x})$.

The function describing the entire vault will be the sum $f = p + d$.

It is not difficult to find a closed-form expression form for p . On the other hand, the deviation term d is not a constant for different vaults, since every vault has its own deformations.

We propose a solution able to reconstruct the entire vault surface f (without separating the contribution of p and d) using the point cloud as representative of discrete values computed for f .

The advantage of the proposed method consists in the representation of the vault with a mathematical function that can be evaluated on a domain to reconstruct the vault. The approach includes modeling without drawing, since the vault is expressed using a closed-form expression. Sections can also be extracted by evaluating the estimated function along particular directions, for instance using a 2D line $x_2 = mx_1 + q$ in the horizontal plane. The paper is structured as follows. Section 2 presents an overview of the developed method addressing in detail the concept of surface approximation with neural networks. Section 3 is focused on testing the viability of the proposed method by presenting some preliminary tests on synthetic datasets. Those tests were mainly devoted to determining the influence of the number of hidden layers of the neural network and the number of neurons to be used in each layer. Section 4 presents results for some real datasets. In particular, the reconstruction of some sets of real vaults with different shape is presented and discussed. The reconstructed vaults show a close correspondence with the input point cloud demonstrating the effectiveness of the proposed approach. The final section is devoted to draw some conclusions and to address future developments.

2. OVERVIEW OF THE PROPOSED METHOD

This work is an extension of what presented in Barazzetti (2018), here applied to the specific case of historical vaults. As mentioned, we assume that the surface of a vault can be approximated by a function $f: D \rightarrow C \subseteq \mathbb{R}$. 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 a task is not a trivial problem, especially in the case of irregular vaults like those of historic buildings, which have geometric irregularities and anomalies.

The mathematical problem behind the definition of the function f is not a simple task. Let us suppose that the idea is to implement an algorithm able to recover a suitable mapping for a specific vault. This is a challenging task not only for the intrinsic difficulties in finding a closed-form expressions with sufficient metric accuracy. The intrinsic deformations of historic vaults make the approach not replicable for other case vaults, which would require specific mapping functions. In other words, working with different vaults requires a continuous change of the mapping function.

The use of photogrammetric or laser scanning point clouds allows one to collect metric information about the unknown mapping. A set of measured values (x_{1m}, x_{2m}, x_{3m}) measured (index m) can be used to recover $f = p + d$ without separating the contribution of p and d . The aim is to find a mapping function that provides a set of correspondences $(x_{1m}, x_{2m}) \rightarrow (x_{3m} + e)$, i.e. an approximate mapping with a "small" error e , which is acceptable and comparable with the expected metric accuracy of the final output or the chosen scale for the model.

The mapping problem may be cast in the form $f: D \subseteq \mathbb{R}^2 \rightarrow \mathbb{R}$. The function may also be expressed in the more common form

$x_3 = f(x_1, x_2)$, which assumes that there is a direct mapping between the spatial coordinates x_1, x_2 and the vertical one x_3 .

The proposed (automatic) solution that realizes this process without manual intervention is based on a neural network $N: D \subseteq \mathbb{R}^2 \rightarrow \mathbb{R}$ 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 .

For an in-depth study of neural network, the reader is addressed to Haykin (2008). Here, only the main concepts will be briefly reviewed. To describe neural networks, we can start considering the most straightforward possible neural network, the one that comprises a single "neuron" (Figure 1).

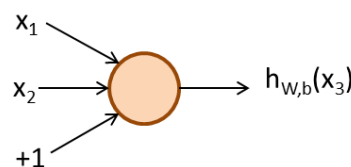


Figure 1. A diagram denoting a neural network with a single neuron.

This "neuron" is a computational unit that takes as input x_1, x_2 (and a $+1$ intercept term), and outputs $h_{w,b}(x_3) = A(W^T x) = f(\sum_{i=1}^2 W_i x_i + b)$, where $A: \mathbb{R} \rightarrow \mathbb{R}$ is called the activation function. In these paper we have used the sigmoid as activation function:

$$A(t) = \frac{1}{1 + \exp(-t)} \quad (1)$$

A neural network is obtained by hooking together many simple "neurons," so that the output of a neuron can be the input of another. In the neural network presented in Figure 2 the leftmost layer of the network is called the input layer ($x_1, x_2, +1$), and the rightmost layer the output layer (which, in our examples, has only one node, x_3). The middle layer of nodes is termed hidden layer.

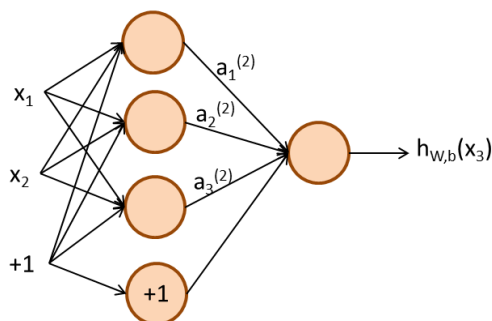


Figure 2. Diagram denoting a neural network with 2 input units (x_1 and x_2 not counting the bias unit), an hidden layer (composed of 3 hidden "neurons") and 1 output unit (x_3).

$a_i^{(l)}$ is the output value (also called activation) of the neuron i in layer l and given a set of the parameters W, b the neural network defines a hypothesis $h_{w,b}(x)$ that outputs a real number. Specifically, the computation of the neural network in Figure 2 is given by:

$$\begin{aligned}
 a_1^{(1)} &= f(W_{11}^1 x_1 + W_{12}^1 x_2 + b_1^{(1)}) \\
 a_2^{(2)} &= f(W_{21}^1 x_1 + W_{22}^1 x_2 + b_2^{(1)}) \\
 a_3^{(2)} &= f(W_{31}^1 x_1 + W_{32}^1 x_2 + b_3^{(1)}) \\
 h_{w,b}(x) &= f(W_{12}^2 a_1^{(1)} + W_{12}^2 a_2^{(2)} + W_{13}^2 a_3^{(2)} + b_1^{(2)})
 \end{aligned}
 \tag{2}$$

A vector consisting of $W_{(i,k)}$ is called weight vector. These weights are what makes each neuron unique. They are fixed during testing. During training such values are the numbers that the algorithms will change to ‘tune’ the network. Neural networks may present different architectures (i.e., patterns of connectivity between neurons). In particular, we may have neural networks with multiple hidden layers. The most common choice is a n_l-layered network where each layer l_i is densely connected to layer l_{i+1}. In this setting, to compute the output of the network, we can successively compute all the activations in layer l₂, then layer l₃, and so on, up to layer l_n.

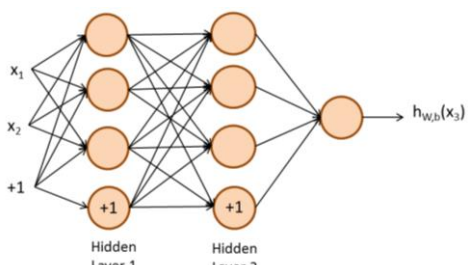


Figure 3. A diagram denoting a neural network with 2 hidden layers.

An example of the presented method based on neural network is shown in Figure 4. The vault was scanned with a Faro Focus 3D, obtaining point coordinates with an expected metric accuracy of about ±2 mm. A feed-forward neural network with a single hidden layer and 40 neurons (Figure 5) was trained using a subset of the original point clouds decimated using curvature information. Regression information on training, test and validation (70%, 15%, 15% of original dataset, respectively, see Figure 6) and a mean squared error (MSE) of $4.02 \cdot 10^{-3}$, $4.12 \cdot 10^{-3}$, and $4.06 \cdot 10^{-3}$ (the relationship between MSE and the more common root mean squares is $RMS = (2 \cdot MSE)^{1/2}$) reveal that the network can approximate vault geometry with millimeter-level precision.



Figure 4. The vault used to explain the procedure in section 2.

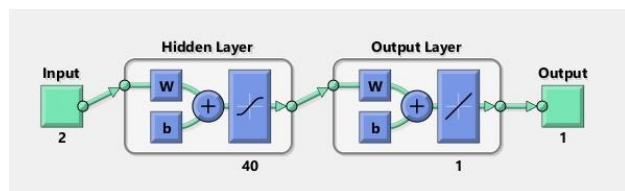


Figure 5. The chosen network architecture for the vault in Figure 1.

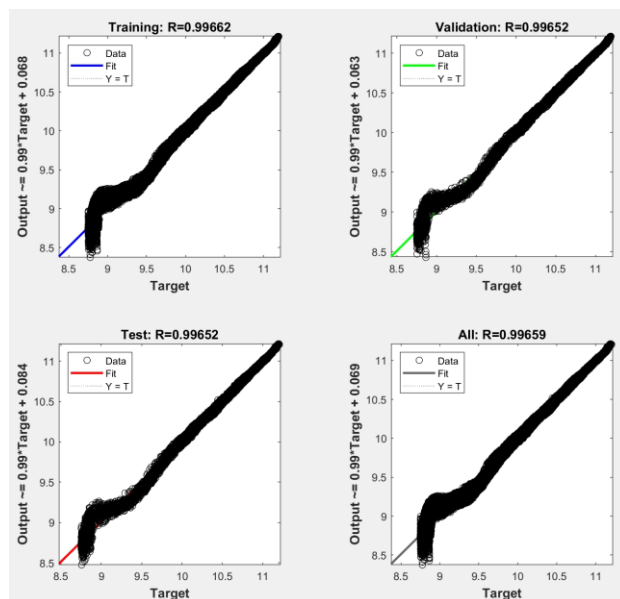


Figure 6. Regression lines for training, validation, and test datasets.

3. TEST WITH SIMULATED DATA

As presented in the previous section the main element constituting a neural network are the ‘neurons’ and the architecture of the neural network itself (i.e., the number of layers in which the ‘neurons’ are organized). This section aimed at investigating the viability of the proposed method for vault fitting and, in particular, to investigate the influence of the number of hidden layers net and the number on neurons in each of them.

To perform this analysis, a set of synthetic data was generated. The aim of using simulated data sets was to discard the influence of noise in the data and allow an easier comparison between the output of the neural network and the original point cloud.

In particular, two synthetic datasets were generated:

- ‘Barrel vault’ data set; and
- ‘Cloister vault’ data set

Some details about the three datasets are provided into Table 1.

	Number of points	Dimensions [m x m]	Point cloud density [points / m ²]
‘Barrel vault’	4.000.000	3 x 5	~ 85.000
‘Cloister vault’	4.000.000	5 x 5	~ 66.000

Table 1. Details about the syntactic datasets used in this work.

Such datasets were chosen for two main reasons. Firstly, barrel and cloister vaults represent a relevant number of real vaults. Secondly, the selected datasets presents a quite large variety of possible criticalities for neural network fitting. Indeed, even if a barrel vault is here represented by a simple shape (i.e., as a semi-cylinder), at the boundary of the barrel vault there are vertical tangency areas that could be critical. Cloister vaults, i.e. those obtained with the intersection of two semi-cylinders, are more complex than barrel vaults, also because the breaklines in correspondence of the intersection between the cylinders.

The following procedure was used. First, the vault is approximated with a synthetic point cloud, which is then down-sampled to 10% of the original point cloud. Using this

subsampled data set a feed-forward neural network was trained splitting the points into a subset for training, test and validation (70%, 15%, 15% of the original dataset, respectively). The trained network was used for recreating the original, not subsampled, point cloud. At this stage, the two point clouds (estimated and original) were compared.

Several tests were performed with different neural network architectures (number of layers and number of neurons). Results seem to highlight two critical aspects of neuron network fitting applied to vaults:

- sharp boundaries seem more problematic than regular surfaces (Figure 7 and Figure 9);
- points characterized by vertical tangent planes are critical areas for the neural network (Figure 7).

Results related to the number of hidden layers in the network along with the neurons into each layer show that:

- for neural network with one layers only, increasing the number of neurons improves the accuracy of the final model (Figure 7 and Figure 8b-c). However, the estimated model seems to produce results affected by noise when the number of neurons is enlarged. The final data presents some bump artifacts (Figure 8d);
- increasing the number of hidden layers allows sharp boundaries and areas with vertical tangent to be reconstructed more precisely (Figure 7, Figure 8, Figure 9 and Figure 10).

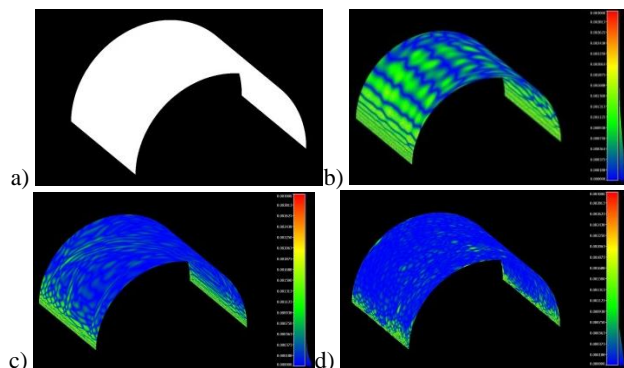


Figure 7. Results for a synthetic barrel vault with a unique layer: (a) original point cloud; (b) result with 20 neurons; (c) results with 100 neurons; and (d) results with 200 neurons (color scale is the same for images b-c-d).

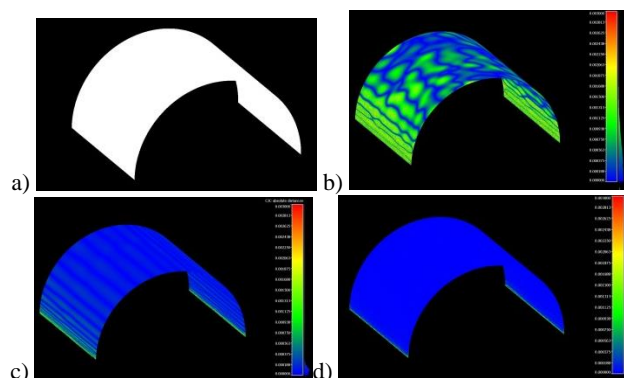


Fig. 8. Results for a synthetic barrel vault with a 10 neuron into each layer: (a) original point cloud; (b) result with 3 layers; (c) results with 5 layers; and (d) results with 7 layers (color scale is the same for images b-c-d).

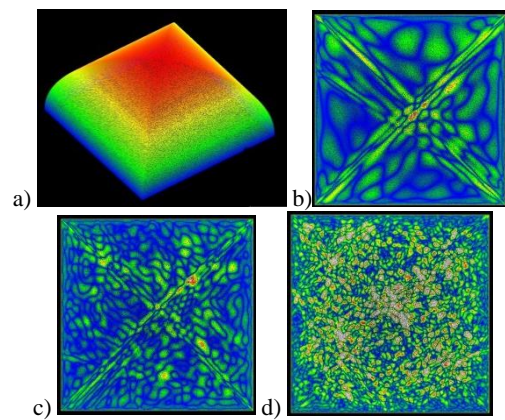


Figure 9. Results for a synthetic cloister vault with a unique layer: (a) original point cloud; (b) result with 50 neurons; (c) results with 200 neurons; and (d) results with 1000 neurons (color scale is the same for images b-c-d).

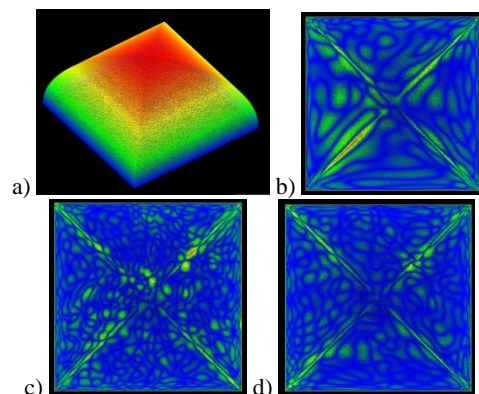


Figure 10. Results for a synthetic cloister vault: (a) original point cloud; (b) result with 3 layers (10 neurons in each); (c) results with 3 layers (30 neurons in each); and (d) results with 3 layers (50 neurons in each) (color scale is the same for images b-c-d).

Barrel vault		
Neural Network Architecture	Percentage of points < 0.005m	Percentage of points < 0.001m
1 Layer 20 neurons	100%	80.61%
1 Layer 100 neurons	100%	85.34%
1 Layer 200 neurons	100%	88.16%
3 Layers 10 neurons each	100%	93.21%
5 Layers 10 neurons each	100%	95.77%
7 Layers 10 neurons each	100%	98.85%
Cloister vault		
Neural Network Architecture	Percentage of points < 0.005m	Percentage of points < 0.001m
1 Layer 50 neurons	100%	85.23%
1 Layer 200 neurons	100%	90.95%
1 Layer 1000 neurons	95.32%	53.11%
3 Layers 10 neurons each	100%	89.30%
3 Layers 30 neurons each	100%	93.46%
3 Layers 50 neurons each	100%	95.21%

Table 2. "Barrel Vault" and "Cloister vault" data set results.

A numerical comparison of the results achieved with the different neural networks was also performed. In particular, the percentage of points at absolute distances less than 0.005 m and 0.001 m is estimated. Results of this comparison are reported in Table 2. These data confirm the previous considerations and the viability of the proposed method.

4. TEST WITH REAL DATA

Different tests were carried out with laser scanning datasets acquired with a Faro Focus 3D. Shown in Figure 11a are the results for an irregular barrel vault 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.

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 projection of the vault were evaluated through a regular grid of 0.01 m x 0.01 m (Figure 11b). A solution to check the quality of the results for real case studies is to compare the results of the network with the original point cloud. Results with CloudCompare provided a discrepancy of 0.006 ± 0.003 m (Figure 11c), which means that the neural network is a good approximation of the original point cloud.

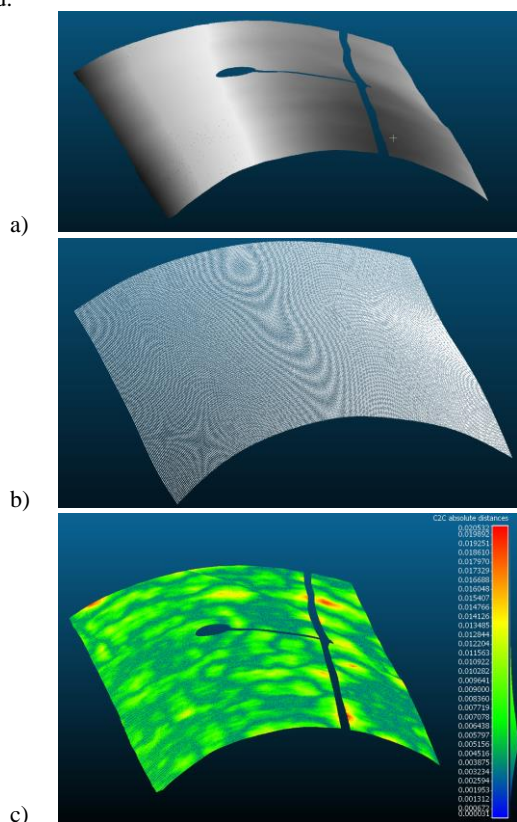


Figure 11. Results for a historic barrel vault: (a) provides a visualization of the original point cloud and the occlusion caused by a wire and a chandelier; (b) illustrates the recovered point cloud evaluated on a 0.01 m x 0.01 m grid; (c) provides a comparison between original and recovered point clouds, obtaining a final discrepancy of 0.006 ± 0.003 m.

The second case study is a cloister vault. Different scans were acquired to capture the vault but some occlusions were found after inspecting scans in the office, as shown in Figure 12a. 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) were used to create a new point cloud on a regular grid of 0.01 m x 0.01 m (Figure 12b). The final discrepancy estimated with CloudCompare between the averaged results and the original point cloud is 0.006 ± 0.003 m.

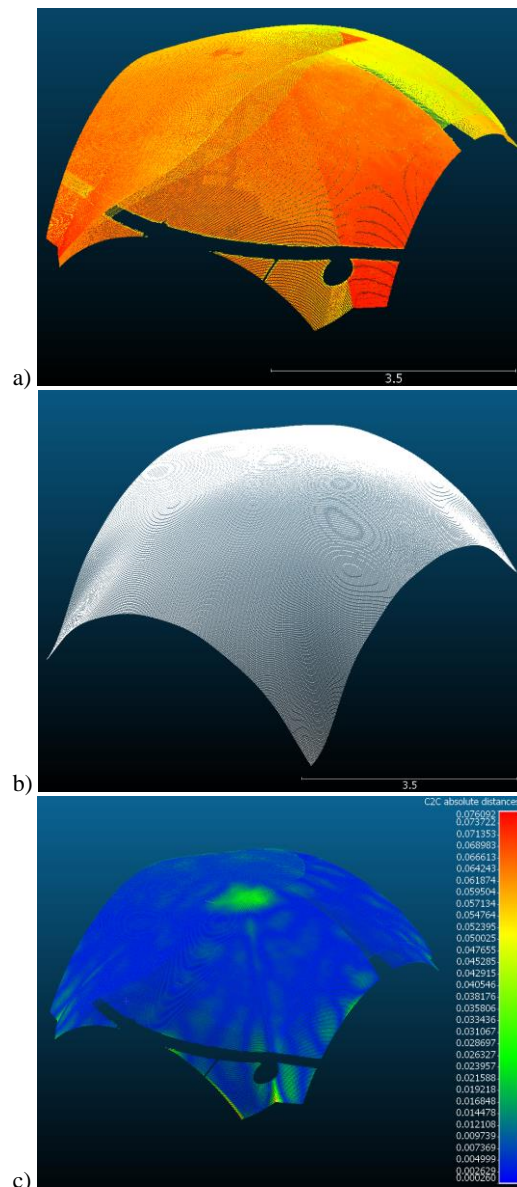


Figure 12. Results for a historic cloister vault: (a) provides a visualization of the original point cloud; (b) illustrates the recovered point cloud evaluated on a 0.01 m x 0.01 m grid; (c) provides a comparison between original and recovered point clouds, obtaining a final discrepancy of 0.006 ± 0.003 m.

The last case study is a pavilion vault in the “Centro Culturale Chiesa Vecchia” a former church in Calusco d’Adda (Italy). The point cloud of the vault is composed by more than 4 million points. In this case, different configurations of the neural network architecture were tested (Figure 13 and Table 3). Results seem to confirm the conclusions that were discussed about the simulated data sets.

Pavilion vault		
Neural Network Architecture	Percentage of points < 0.005m	Percentage of points < 0.003m
1 Layer 50 neurons	98.26%	81.66%
1 Layer 100 neurons	98.28%	84.01%
1 Layer 1000 neurons	90.16%	83.16%
3 Layers 20 neurons each	98.03%	85.12%
5 Layers 30 neurons each	98.26%	86.69%

Table 3. "Pavilion Vault" data set results.

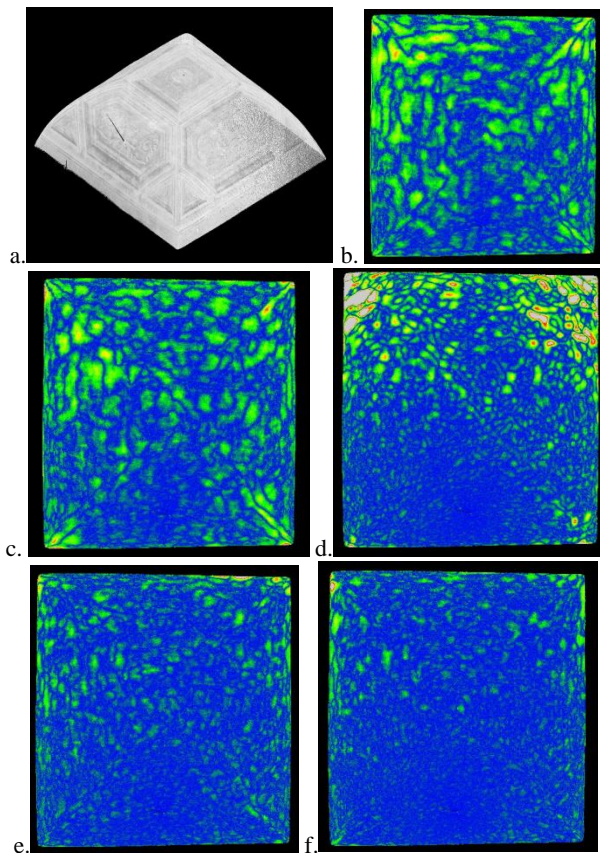


Figure 13. Results for a historic pavilion vault: (a) provides a visualization of the original point cloud; (b) comparison results result with 1 layers 50 neurons; (c) comparison results result with 1 layers 100 neurons; (d) comparison results result with 1 layers 1000 neurons; (e) comparison results result with 3 layers 20 neurons each; and (f) comparison results result with 3 layers 30 neurons each.

5. CONCLUSION

The paper presented a novel procedure for the reconstruction of vaults. The process could be intended as a strategy for "modeling without modeling" since the vault is mathematically reconstructed with a closed-form expression encapsulated in the numerical coefficients of a neural network.

The method has advantages regarding memory occupation. The model of the vault is available after evaluating the computed network on a spatial domain. The parameters that must be stored are the code behind the network and the computed numerical coefficients. This is a simple textfile with very limited memory occupation.

Network architecture has an essential role not only in the final metric quality of the method but also in the time required to estimate network coefficients. Other issues are related to the convergence of the solution, the number of hidden layers, the number of neurons on each layer, and the decimation procedure for the original point cloud. Additional work will focus on these issues, addressing the case of other input parameters. Indeed, more information captured by the laser scanner or digital cameras (e.g., intensity, RGB, ...) can be used to train the network, extending the output to other parameters rather than just using z coordinates.

6. REFERENCES

- Barazzetti, L., Banfi, F., Brumana, R., Previtali, M., 2015. Creation of Parametric BIM Objects from Point Clouds Using NURBS. *The Photogrammetric Record*, 30(152), pp. 339–362.
- Barazzetti, L., 2016. Parametric as-built model generation of complex shapes from point clouds. *Advanced Engineering Informatics*, 30, pp. 298–311.
- Barazzetti, L., 2018. Point cloud occlusion recovery with shallow feedforward neural networks. *Advanced Engineering Informatics* 38, pp. 605–619.
- Fassi, F., Achille, C. and Fregonese, L., 2011. Surveying and modeling the main spire of Milan Cathedral using multiple data sources. *Photogrammetric Record*, 26(136): 462–487.
- Haykin, S. (2008). *Neural Networks and Learning Machines*. Third Edition. Pearson, Prentice Hall, 936 pages.
- 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.
- Kraus, K., 2008. *Photogrammetry: Geometry from Images and Laser Scans*. Second edition. Walter de Gruyter. 459 pages.
- 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).
- Vosselman, G., Maas, H.-G., 2010. *Airborne and Terrestrial Laser Scanning*. Whittles Publishing, Dunbeath, Caithness, Scotland, United Kingdom.