Contents lists available at ScienceDirect

Computers & Graphics

journal homepage: www.elsevier.com/locate/cag

Special Section on CEIG 2023

Deep learning of curvature features for shape completion

Marina Hernández-Bautista [a,c,*], Francisco Javier Melero [b,c]

[a] Department of Computer Science and Artificial Intelligence, University of Granada, 18071 Granada, Spain
[b] Department of Software Engineering, University of Granada, 18071 Granada, Spain
[c] Andalusian Research Institute in Data Science and Computational Intelligence (DaSCI), Spain

ARTICLE INFO

Article history:
Received 20 May 2023
Received in revised form 27 June 2023
Accepted 5 July 2023
Available online 16 July 2023

Keywords:
Shape completion
Curvature representation
Parameterization
Inpainting

ABSTRACT

The paper presents a novel solution to the issue of incomplete regions in 3D meshes obtained through digitization. Traditional methods for estimating the surface of missing geometry and topology often yield unrealistic outcomes for intricate surfaces. To overcome this limitation, the paper proposes a neural network-based approach that generates points in areas where geometric information is lacking. The method employs 2D inpainting techniques on color images obtained from the original mesh parameterization and curvature values. The network used in this approach can reconstruct the curvature image, which then serves as a reference for generating a polygonal surface that closely resembles the predicted one. The paper's experiments show that the proposed method effectively fills complex holes in 3D surfaces with a high degree of naturalness and detail. This paper improves the previous work in terms of a more in-depth explanation of the different stages of the approach as well as an extended results section with exhaustive experiments.

1. Introduction

3D digitizers, either active (scanners) or passive (photogrammetry), have become increasingly popular for professionals in cultural heritage documentation, 3D printing, or the virtual reality and video game industries, as they allow for the creation of models that accurately depict reality.

However, even with the use of advanced technology, point clouds obtained from 3D digitizers are sometimes insufficient for creating complete models. This can occur due to various factors, such as low data quality or the absence of points in specific areas. These issues can arise for several reasons, including incorrect scanning angles, low or high surface reflectance, occlusion of parts of the surface, or even lost parts of the original model. All these defects result in visual inaccuracies and holes in the obtained surface model, which must be repaired to obtain a closed model suitable for 3D printing or for calculating geometric properties.

To overcome these challenges, researchers have developed various methods to repair incomplete surface models, such as manually filling the holes, applying surface reconstruction techniques, or using neural network-based approaches like the one proposed in this paper. These methods aim to generate missing geometry and topology by estimating the surface and filling in the gaps, resulting in realistic and detailed models that accurately represent the original object.

Classical work in the field of Computer Graphics addresses the coverage of holes with volumetric data [1–3] or surface extension models [4–6] to generate new points within the missing part. These methods usually generate surfaces with a low level of detail in complex models and are inadequate for surfaces with a high number of polygons.

Advanced approaches include the use of deep learning to learn strong priors [7–9] and have been shown to repair complex holes in large surfaces. These methods require a huge amount of data to train the models, and hence they need a lot of computational time and resources. However, in approaches such as [10], a CNN neural network automatically learns a prior from an input point cloud without using a dataset or training time, but the results lack the accuracy needed for complex shapes.

Therefore, we propose a solution that adapts our shape completion problem to any inpainting process done by neural networks that have already been trained, such as DALL·E 2 [11] or Stable Diffusion [12].

Our work, graphically depicted in Fig. 1 consists of a coarse-to-fine reconstruction using images representing our shape. Given a normalized mesh, we apply a coarse repair method following a classical approach [4]. After that, a heuristic segmentation of the surface is performed to produce the parameterization [13] of the model in patches that preserve the area of the hole to be filled. Once parameterized, the generated image is colored with

[*] Corresponding author at: Department of Computer Science and Artificial Intelligence, University of Granada, 18071 Granada, Spain.
E-mail addresses: marinahbau@ugr.es (M. Hernández-Bautista), fjmelero@ugr.es (F.J. Melero).

https://doi.org/10.1016/j.cag.2023.07.007

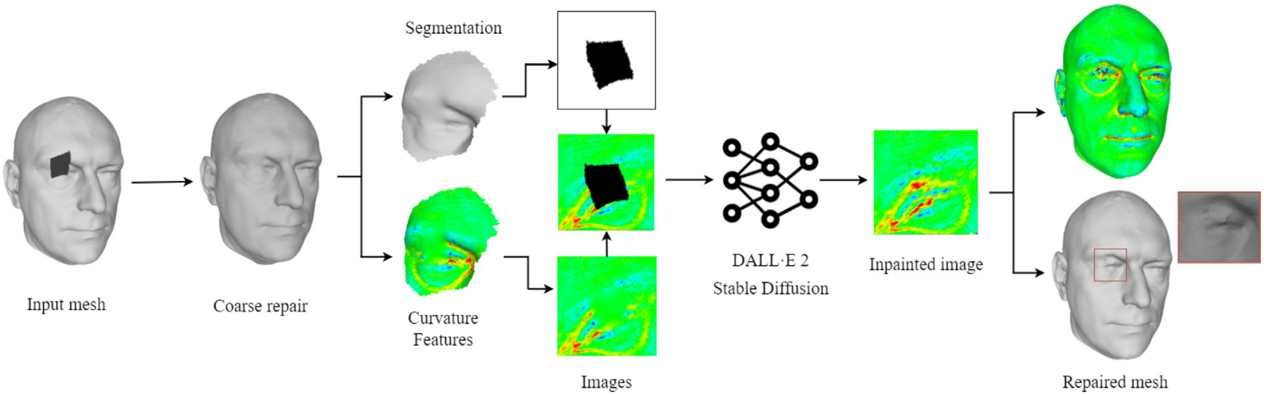


Fig. 1. Pipeline of the whole process of deep learning of curvature features for shape completion.

the values of the curvature [14]. These images are then proposed to DALL·E 2 or Stable Diffusion, general-purpose neural networks that complete the images using inpainting techniques, among other tasks. Finally, the local meshes are submitted to a fairing process. The geometry of the hole is progressively modified until its curvature values match those predicted by the artificial intelligence.

This shape completion neural network-based approach does not require any specific neural network design and implementation, training dataset, or training time. We demonstrate that DALL·E 2 and Stable Diffusion are able to receive our curvature images and inpaint them so that we can properly infer new geometries and repair holes in simple and complex surfaces.

In conclusion, this paper presents an extended and improved version of the work in [15]. Our main contributions are:

- A novel surface completion technique based on neural networks that uses images and inpainting processes to complete geometries without requiring a training dataset.
- A novel representation of a 3D surface geometry as a 2D image based on its curvature value and color for use with an inpainting neural network.
- The demonstration that a general-purpose artificial intelligences such as DALL·E 2 or Stable Diffusion can be used to inpaint images representing curvature values, thereby eradicating the process of designing an *ad-hoc* neural network model and training it. This significantly decreases computational time and resources.

The structure of this paper is divided into three main sections. Firstly, in Section 2, we will review the existing literature related to the techniques proposed in this paper. Following this, in Section 3, we will provide a detailed and comprehensive explanation of our approach, focusing on its individual components. Finally, we will compare our approach with state-of-the-art classical and learning-based methods in Section 4 to demonstrate its effectiveness and potential advantages over existing techniques.

2. Related work

Our approach is situated at the intersection of two major research areas in Computer Graphics. The first area involves using inpainting techniques to fill in large missing parts of images. The second area concerns using these images to generate new geometries to produce a complete 3D shape.

Both of these research areas have received considerable attention in recent years, and as such, we will provide a thorough literature review of shape completion and inpainting approaches in the following subsections.

2.1. Shape completion

Extensive research has been conducted to address the issue of filling holes or artifacts in simple and complex 3D objects. Classical approaches can be broadly classified into two categories. The first category is concerned with volumetric information, where the data is stored as a grid and often involves a complete remeshing of the mesh. These methods work well for complex holes and have been applied in several studies. For instance, [1] represented the surface using a grid of signed distance function values and propagated these values through a diffusion process with alternating steps of blurring and compositing to fill the holes. Similarly, [3] used diffusion to propagate a voxel grid of quadruplets, which represents distance, direction, weight, and changeability. Another volumetric approach involves completing incomplete point sets, as demonstrated by [2], who used octree discretization to find the most similar surface patch using its digital curvature signature, and then merged it into the hole region by solving a Poisson equation. This approach was further improved by [16] through the application of positional constraints, resulting in better outcomes.

The second group of methods focuses on using the surface information of the mesh to repair holes. These methods typically use the information from the vicinity of a hole to repair it without altering the remaining mesh. However, these methods may not be suitable for larger or more complex holes. For instance, the approach proposed by [4] triangulates the hole and minimizes the resulting area using a weighted function. Then, it refines the surface of the reconstructed hole to match the density of the surrounding triangles. Similarly, [5] inserts new vertices to triangulate the hole and match the density, while minimizing normal variation. Another framework by [17] combines the work of [4] with [18] to achieve a fast and precise repair. To preserve curvature, [6] unfolds the hole and minimizes the area of the triangulated patch. In these works, the Delaunay triangulation algorithm [19] is widely used, as in other works such as [20,21].

Another category of hole filling methods is context-based, which uses existing information or patterns in the mesh to fill gaps. One example is the work of [22], which goes beyond simply filling the hole and also incorporates coherence, ensuring that the completed model's local neighborhoods are similar to those of the original model. The method proposed by [23] first fills the hole smoothly and then transfers the Laplace coordinates of the surrounding region to the reconstructed area.

New hole-filling methods based on neural networks have emerged with the advancement of artificial intelligence. These methods are able to extract features from a mesh to infer new geometries and close holes. Some use an encoder–decoder architecture to learn different prior knowledge and create a latent space of shapes that they map to their partial inputs [7]. The use of a signed distance function to represent noisy input data and recover shapes and missing regions is prevalent in many approaches [8,9,24]. The approaches of [9,25] use variational autoencoder (VAE) in their training and have demonstrated their accuracy on 3D shapes from the KITTI [26] and ShapeNet [27] datasets. In addition, [28] uses a 2.5D sketch estimator, a 3D shape completion network, and a naturalness network model to repair models from a single depth image.

In both [7,29], a coarse-to-fine approximation is employed. The former proposes the use of a 3D shape classification network to provide additional information, although this approach requires training data and has limited applicability. The latter abstracts a point cloud into a feature vector and achieves a coarse-to-fine-grained result. GAN architectures have also been used for 3D shape completion, where a generator and a discriminator work together to learn features of a 3D model, similar to their use in image generation. These approaches have shown promise in the field of 3D shape completion. For instance, in [30], voxelized input is used to train two GANs that share the same generator. One GAN generates the repaired region locally, while the other GAN assembles the repaired region with the original model globally. This enables both GANs to learn global and local features. Another work, [31], uses an inverted process in GAN where the output is degraded, resulting in a partial shape. This allows the output to be generated even if the network has not been trained for a new category of models. Another method for surface reconstruction [32] uses a set of deep ReLU networks to obtain local charts or parametrizations. Reinforcement learning is also employed in 3D shape completion. For example, [33] uses a RL agent to adapt the seed of the generator of the GAN, so that the GAN generates output that is optimal for an autoencoder to recover full shapes.

2.2. Inpainting

In our research, we employ an image inpainting process to fill in the missing parts of an image that represents the mean curvature values of a 3D model. Image inpainting is a well-established technique for restoring missing portions of an image. Conventional approaches, such as those described in [34,35], utilize patch-based synthesis to reconstruct images and are effective for large textures. In contrast, [36] is effective at filling in missing regions by propagating isophote lines in the image, but is not well-suited for handling textures.

In addition to shape completion, recent studies have utilized neural networks to obtain unsupervised and enhanced outcomes [37–39]. The work of [37] shares a similar idea with our approach. The damaged regions in their work are isolated by an offset and converted into a texture image that undergoes an inpainting process. The difference between their method and ours is the use of an offset in the mesh segmentation.

In our work, we use DALL·E 2 [11] as an external neural network to perform the inpainting process on an image representing the mean curvature values of a 3D model. DALL·E 2 is a recent neural network architecture designed for generating high-quality images from textual descriptions. The architecture includes an encoder and a decoder, with the text label being passed through a CLIP transformer to generate a vector representation called a text embedding, which is a compressed representation of the text label information. This embedding is then used to produce an image embedding through an autoregressive or diffusive prior, which is then passed to a diffusion decoder to generate a final image. The architecture is capable of generating a variety of images that retain the semantic information of the input label.

In addition, we have made the decision to incorporate Stable Diffusion [12], providing an open source alternative for the inpainting process. This further illustrates the independence of our approach from the specific neural network used, eliminating the need for additional training time while still yielding accurate results. Notably, the inpainting process demonstrates robust generalization capabilities for our curvature images.

The Stable Diffusion architecture retains certain components from the DALL·E 2 architecture but implements a distinct procedure. It employs an Autoencoder architecture to generate a condensed representation of the images, referred to as the lower latent representation. This representation is then utilized to train various diffusion models for different tasks, including text-to-image generation, inpainting, super-resolution, and more.

3. Shape completion by curvature inpainting

Our approach restores the surface of a complex object with multiple holes by depicting the curvature of the object. The curvature feature image is then processed using inpainting techniques to fill in the missing parts. Iterative deformations are used to infer the missing geometry over the mesh surface from the resulting image, until the surface curvature map matches the inpainted curvature image. This is how our method successfully fixes intricate surfaces.

The process is divided into several stages, which will be explained in the following subsections. First, we introduce a coarse repair method in Section 3.1. Next, we explain a segmentation and parameterization technique for generating an image representation in Sections 3.2–3.5. Then, we analyze the DALL·E 2 and Stable Diffusion inpainting methods and their results in Sections 3.6–3.7. Finally, we perform a mesh deformation process to create a fine-tuned repair in Section 3.8.

3.1. Coarse reparation process

We first preprocess the mesh and normalize it by scaling to a unit bounding box in order to prevent distortions in the curvature values, which depend on the length of the edges. By doing this, we ensure that curvature values are consistently contained within a predetermined range.

The first step, given a normalized mesh, is to fill the holes using a traditional method. We repeatedly iterate through the mesh's half-edges and examine their incident faces in order to find every hole. If a half-edge only has one incident face, it is a boundary half-edge, and its vertex also qualifies as boundary vertex.

After locating the half-edge loops that define the holes, we can now fill such holes using the classical method formulated by [4]. The process, shown in Fig. 2, works as follows: the algorithm first triangulates the holes, using a weighted function to minimize the area of the resulting triangles. The resulting triangulation is then refined to fit the surrounding density of triangles while maintaining the Delaunay triangulation conditions. This refinement process aims to maximize the minimum angle of all triangles in the mesh. To further improve the mesh quality, the algorithm computes the edge lengths and diffuses these values across the surface mesh. This step subdivides the triangles to obtain an approximate value of the edge length. This ensures that the mesh has an even distribution of edges, which leads to better visual quality and makes it more suitable for numerical simulations. The algorithm repeats this process for every hole in the mesh, producing what we call a coarse reconstruction surface. The objective of our approach consists of fine-tuning these coarse patches in the last phase of the process to improve the overall mesh quality.

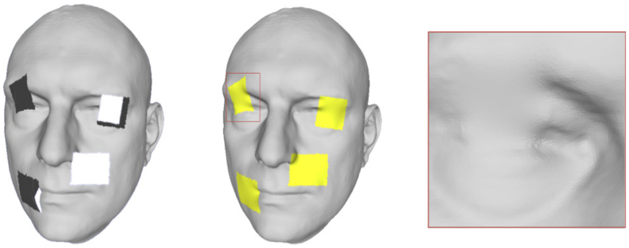


Fig. 2. Model with holes (left). Coarse repair of the model (middle). Zoom of the repair (right).

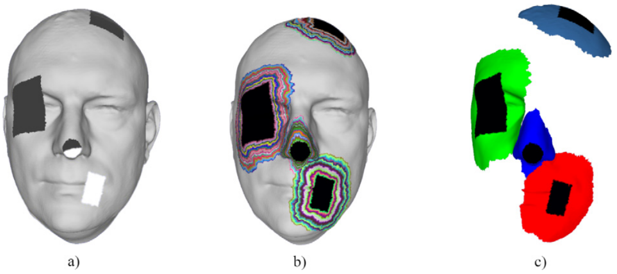


Fig. 3. (a) Model with holes, (b) Expansion process, each iteration in each hole is painted in a different color. (c) Final local patches created in the segmentation process. These meshes are then parameterized and an image representing its surface curvature is generated.

3.2. Segmentation of the mesh

Our method processes 3D data using a 2D technique (picture inpainting). For that reason, it is necessary to parametrize the mesh and obtain a planar representation of the surface. However, if we conduct a global parameterization and generate a single image per mesh, the triangles will be severely deformed, and it is highly likely that small holes will occupy only a few pixels in the image, making it impossible to generate a reliable inpainting of the image and thus a precise reconstruction of the mesh. To reduce triangle deformation (and more specifically, hole deformation), the mesh is segmented using a hole-centric method. Our objective is to concentrate on our areas of interest, i.e., to generate local meshes representing each hole and its surroundings, so that the parameterization and the generated image have a reasonably wide contextual information that the inpainting neural network is able to use in order to reproduce symmetries, lines, or recognize certain shapes, while the dimensions and contour of the hole remain undeformed.

We perform such segmentation through an iterative expansion process illustrated in Fig. 3. We initialize patches using the hole-boundary vertices and the new vertices generated during the coarse repair process. These patches are then expanded by adding immediately neighboring vertices to create extended patches. Iteratively, all vertices within the external boundary of the patch are expanded. During patch expansion, we ensure that each patch boundary stops just before crossing the boundaries of other patches, or after a given number of iterations, or when the patch is entirely surrounded by other patches. This way, we avoid overlapping patches and ensure that each patch covers a distinct region. After the patch expansion stage, we obtain a set of vertices and faces labeled according to the patch to which they belong.

This information is used to create local meshes representing each hole and its surrounding contextual information. These patches are parameterized in the next step of our method, enabling us to generate planar representations of each patch that can be used for further processing.

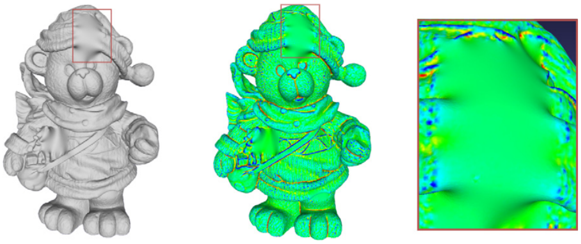


Fig. 4. Coarse-repaired model with curvature colors per vertex. The right Figure shows the simple repaired surface of the hole, which is very smooth. It contrasts with the curvature colors in the real original surface.

3.3. Curvature calculation and color

We want to create an image that defines the surface of our mesh so that an AI is able to inpaint this feature with colors. We chose the mean curvature property. Mean curvature is a measure that characterizes the curvature of a surface and, by extension, the geometrical shape of surfaces. The mean curvature at a point on a surface is the average of its two principal curvatures. (i.e., the maximum and minimum curvature in the two principal directions on the surface that pass through that point). The formula for calculating the mean curvature is as follows:

$$H = \frac{1}{2}(k1 + k2)$$

where $k1$ and $k2$ are the principal curvatures at the point of interest.

We calculate the mean curvature at each point of the repaired mesh following the APSS approach of [14]. APSS is a method for representing a surface as a set of algebraic points. For the calculation of the mean curvature by means of APSS, a sphere fitting method is suggested. This method consists of selecting a small neighborhood and finding the best-fitting sphere. The radius of this fitted sphere can then be used as an estimate of the curvature at the point of interest. This approach is proposed to be relatively easy to compute, rather than the expensive evaluation of a shape matrix proposed in traditional methods.

Once we have the normalized mean curvature estimated for each vertex, we map such values to RGB tuples according to a color map. In this case, we use a rainbow map where blue represents low (convex) values, medium (flat) values are colored as green, and high (concave) values are displayed as red, as depicted in Fig. 4. Once the color is applied, we transfer it to the vertex of each patch, so that we can produce a 2D image representing the curvature features of each patch in the next stage.

3.4. Parameterization

The next step is to convert the 3D surface of each hole patch into a 2D representation (i.e., a parameterization) so that the neural network can identify them and inpaint the content.

A parametric surface is a 2D plane where a function g, called the parameterization function, is defined in the domain $[0, 1]^2$. This function allows the transformation of a point $p(x, y, z)$ to a point on the parametric surface $p'(s, t)$, called the parametric coordinates of point p.

$$p = g(s, t)$$

Thanks to this function, we can unfold 3D surfaces into a 2D domain and thus create an image representing the surface.

To perform the parameterization of the patches, we follow the work of [13] because its main advantages are the shape preserving feature and the high speed of computation. In this approach, we need to compute the mean value coordinates for every vertex. These coordinates are defined as the weighted sum of the angles subtended by the edges of the neighboring triangles at the vertex. This means that neighboring vertices with a larger angle are more influential in determining the position of the target vertex in 2D space. The coordinates of each vertex are then calculated by taking an average of the coordinates of its neighboring vertices weighted by the calculation of its barycentric coordinates. The final coordinates may not exactly match the original mesh, but they should preserve its overall shape as much as possible.

3.5. Curvature image and holes masking

We use an image-based representation of the surface to create a smooth curvature map that is essential for the neural network to better understand the image content and produce a smoother reconstruction of the missing part. To create the curvature map, we map each vertex color to its corresponding pixel using its parametric coordinates. To interpolate the colors of the intermediate pixels, we use a bilinear interpolation algorithm that takes the weighted average of the colors of the four nearest texels to the pixel being rendered. This produces a smoother image that helps the neural network better understand the shape of the surface. In addition to the curvature map, we create a binary mask that represents the hole to be reconstructed using black and white colors. The black color represents the missing part of the surface, and the white color represents the rest of the surface. The binary mask helps the neural network to focus on the hole and generate an accurate reconstruction.

Finally, we combine the curvature map and the binary mask to produce the final image that will be used by the neural network. This image has transparent pixels in the areas belonging to the hole, indicating to the neural network that these are the areas that need to be reconstructed. The resulting image provides the neural network with the information it needs to generate a high-quality reconstruction of the missing part of the surface.

3.6. Inpainting with DALL·E 2

In contrast to other deep learning-based approaches, we delegate the inpainting procedure to the well-known neural network DALL·E 2 [11]. This approach has a computational advantage because we do not need to train a model. We demonstrate that DALL·E 2 is already capable of inpainting the generated curvature value images. Therefore, we do not need a dataset, nor do we need to invest time in training and refining a model in order to obtain accurate reconstructions.

The architecture of DALL·E 2 is simple, as illustrated in Fig. 5. They introduce an encoder–decoder architecture that is divided into two main parts. The first part consists of training a CLIP model [40]. The CLIP model contains two encoders, one for text input and the other for image input. The CLIP model itself is a transformer, and it is able to create a latent space of what they call image and text embeddings (i.e., encoded feature vector representations of images and text), so the CLIP model is able to tell when a text embedding or an image embedding match well together. The second goal is to unCLIP the previously obtained result. For this purpose, there is a prior model and an image embedding decoder. First, a text embedding produced in CLIP is fed into an autoregressive or diffusion prior to generating an image embedding. Then, a diffusion decoder takes the image and decodes it into the final reconstructed image.

To perform the curvature image inpainting process, DALL·E 2 is fed with our generated curvature images with the holes marked as transparent pixels. We tried different prompts on DALL·E 2 in order to find one that satisfies the requirement of completing the missing parts. The command chosen for DALL·E 2 is simply *"fill in the blanks"*. Some other possible prompts have been proposed, such as *"fill with the same colors"* or *"remove the transparent region out of the image"*, but the results do not fulfill the objective of the inpainting process. The result is a reconstructed image that is consistent with the surroundings of the hole. Despite providing DALL·E 2 images that have nothing in common with nature images (i.e. they have been synthetically produced and are unintelligible to a human), Fig. 7 shows that DALL·E 2 completes the gaps of the image with the same color gamma provided and succeeds in completing lines and inferring similar colors and shapes. Moreover, it is also able to detect and reproduce symmetries.

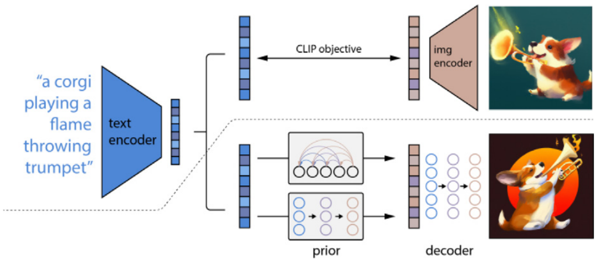


Fig. 5. DALL·E 2 unCLIP architecture [11].

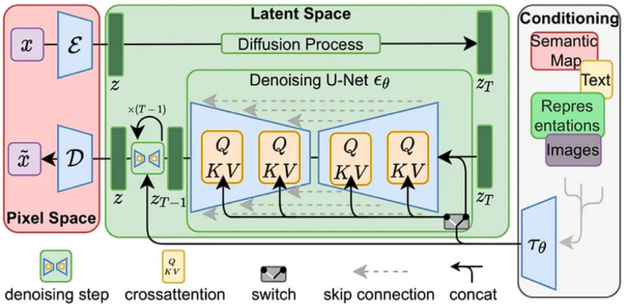


Fig. 6. Stable Diffusion architecture [12].

3.7. Inpainting with stable diffusion

Stable Diffusion has emerged as the leading open-source alternative to DALL·E 2. We have successfully employed this method for inpainting curvature images and achieved impressive generalization results.

The underlying principle of this approach, depicted in Fig. 6, is rooted in the understanding that a substantial portion of image data consists of imperceptible and insignificant details. These unnecessary details only serve to increase the training time and computational requirements when using adversarial or diffusion networks for inpainting tasks. The architecture of Stable Diffusion effectively mitigates these challenges by training an autoencoder network capable of both perceptual compression and semantic compression. By doing so, this network creates a latent space representation that possesses a lower dimension while preserving the perceptual equivalence of the original images.

The autoencoding network is trained as a one-time process and serves as the foundation for training various diffusion models. Diffusion models are capable of modeling conditional distributions by utilizing different conditional encoders. This approach enables us to leverage different input types to achieve favorable outcomes in tasks like text-to-image generation or image-to-image generation. By utilizing the latent space representation learned by the autoencoder, we can effectively generate

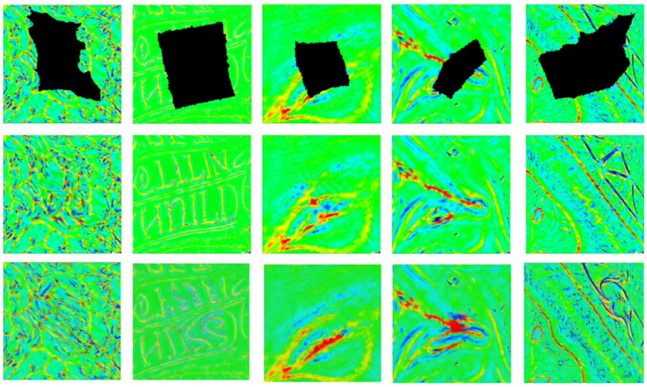


Fig. 7. Generated images representing holes (top). Inpainted DALL·E 2 images (center). Inpainted Stable Diffusion images (bottom).

diverse and high-quality results across a range of conditional input scenarios.

In the Stable Diffusion approach, two images are provided as input: the curvature image and a separate black and white image serving as a mask to represent the hole. However, compared to DALL·E 2, Stable Diffusion requires more careful prompt engineering. We have observed that Stable Diffusion encounters difficulties when prompts involve specific orders, such as *"fill in the blanks"*, resulting in suboptimal outcomes. Instead, a more effective approach involves describing to Stable Diffusion the desired picture and the specific details it should recover from the image.

After several attempts, such as prompts like *"Color map representing curvatures made by green, red, and blue zones"* or *"Smooth color map representing curvatures made by green, red, and blue gradient zones, matching the surroundings"*, it was determined that the most successful self-explanatory prompt was *"Color map representing curvatures made by green, red, and blue gradient zones, match the surroundings, diffuse colors"*. We chose this prompt because it yielded the best qualitative results in our experiments.

In Fig. 7 we can observe a visual comparison of the inpainting results produced by the two neural networks, DALL·E 2 and Stable Diffusion. Despite not being specifically trained on this type of image, both networks generate high-quality results. However, there are notable differences between the outputs.

One significant difference is the color loss exhibited by Stable Diffusion. This can be attributed to the resizing of images to a lower resolution of 512 × 512, which is necessary to meet the model requirements of Stable Diffusion. In contrast, DALL·E 2 is capable of inpainting images with a resolution of 1024 × 1024, allowing for higher-resolution and more consistent outputs.

In conclusion, DALL·E 2 excels in its ability to comprehend text prompts effectively and deliver outputs with higher resolution and consistency, requiring minimal prompt engineering. Despite these advantages, after employing prompt engineering techniques, Stable Diffusion produces results with minimal differences in terms of quality and consistency. Moreover, Stable Diffusion's architecture requires less computational time and resources compared to DALL·E 2.

3.8. Adapt surface to estimated curvature

The final step of our proposal is to unfold these new color values, transform them into curvature values, and reconstruct the missing surface. For that purpose, we use an iterative reconstruction algorithm that takes into account the difference between the colors of the curvature of the coarse repair and the colors of the pixels inpainted by DALL·E 2 or Stable Diffusion.

To achieve this, we deform the vertices that present a certain color difference. This similitude parameter is set as an average value so that the reparation of the mesh is optimal in any case. For this approach, we have set a similitude parameter to a difference of 30 units in any color channel. For example, if a blue color represented by RGB coordinates (0, 254, 228) is shown in the image as a darker blue pixel with RGB coordinates (28, 77, 250), it presents a color difference bigger than the similitude parameter in the green channel, so the vertex needs to be repaired. Choosing a lower or higher similitude parameter results in noise generation or lower precision reconstructions. The color difference determines the direction that must be used to displace the vertices. For example, in the coarse repair surface, the source vertex is usually green, representing a smooth value. If the corresponding inpainted pixel is red, it means that its curvature should be a high (concave) value. Therefore, we have to displace the vertex in the opposite direction of its normal vector so that it reaches the optimal curvature value. If the pixel value is blue, the direction is the opposite: we have to move the vertex a certain amount of distance in the normal direction so that it reaches a lower (more convex) value of curvature. Since we do not know the exact offset we have to move the vertex to generate the predicted curvature, this offset parameter (named the transformation constant) can be modified for each mesh to achieve a more precise or smoother result.

Such fine-grained displacement of vertices introduces noise in the mesh surface when performed as described above. Instead, to reduce the noise introduced by the displacement of the vertices, we use a mesh deformation approach similar to [41], which provides a simple and intuitive way to simulate the elastic deformation of an object based on the forces acting on its surface (Fig. 8b). As illustrated in Fig. 8, by updating the vertex position in response to these forces, the algorithm can generate realistic-looking deformations. We define an area of interest (the patch), and in that area, we select the vertices with the largest color difference as control vertices to perform the deformation according to their new target positions.

4. Results

Shape completion poses a significant challenge in the domains of computer vision and graphics, as it involves inferring the complete shape of an object from partial and potentially noisy input data. Over the past few years, researchers have put forth various methods to tackle this problem, including data-driven approaches that leverage the power of machine learning from example data, as well as geometry-based methods that utilize prior knowledge about object shapes.

In this paper, we aim to enhance the existing research on shape completion by providing an extensive evaluation of state-of-the-art approaches. In Section 4.1, we will meticulously review these approaches and assess their respective strengths and weaknesses based on qualitative results. Our analysis will primarily focus on the methods' efficacy in reconstructing missing parts of objects and their ability to generate plausible and visually appealing shapes. Moreover, we propose to evaluate different reconstruction parameters outlined in the methodology section. This evaluation aims to provide readers with a deeper understanding of the shape completion process and the specific parameters' impact on the experimental results.

Additionally, we will present quantitative results in Section 4.2 to enable a more rigorous evaluation of these approaches. Moreover, we will offer interpretations of these results and discuss their implications for the broader field of shape completion.

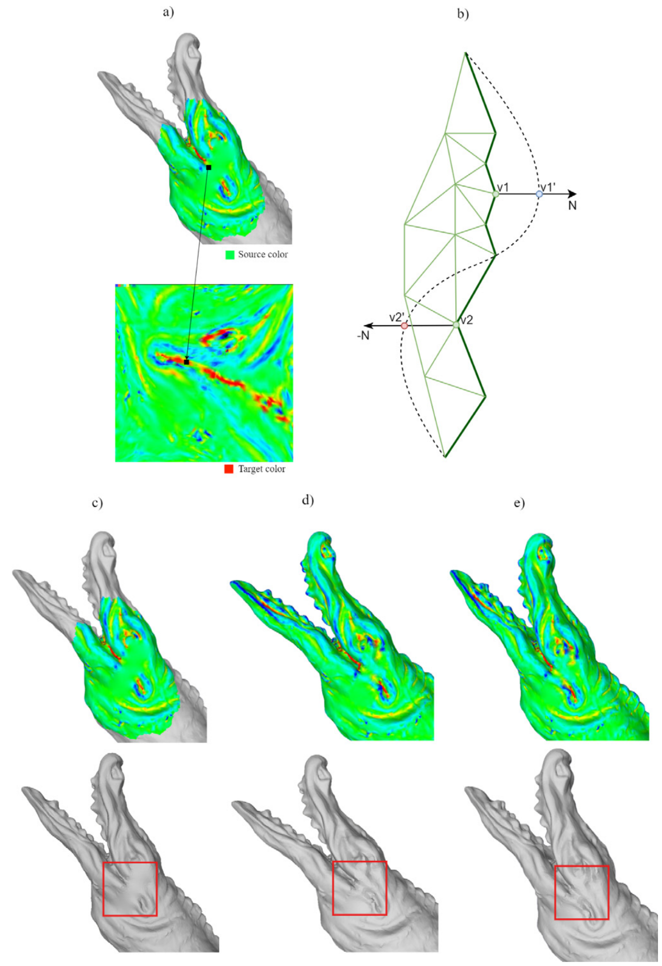


Fig. 8. (a) Coarse reconstruction vs predicted curvature map of the reconstruction. (b) Scheme of the deformation of the vertices depending on the color difference. (c) the coarse repair; (d) first deformation step; (e) final reconstructed mesh. Reconstruction made using DALL·E 2 inpainting.

4.1. Qualitative results

Curvature completion results comparison. In our study, we compared our proposed approach with two of the most widely used methods for surface completion, learning-based approaches, and classical approximations.

For the learning-based approaches, we selected a recent method by [25], which uses a variational auto-encoder to learn shapes from raw point input datasets. However, this method requires a large amount of training data and time to achieve good results. While the reconstructed shapes produced by this method are smooth, the authors themselves noted in their paper that it struggles with repairing small and thin structures, as is evident from the bear example shown in Fig. 9. Therefore, there are limitations to this approach in terms of accurately reconstructing highly detailed shapes.

Another method that we compared our proposed approach with is the Point2Mesh approach, as introduced by [10]. In contrast to the previously mentioned learning-based approach, Point2Mesh reconstructs a mesh by optimizing the weights of a convolutional neural network (CNN) to learn a prior for deforming an initial mesh to wrap the given input point cloud. The advantage of this method is that it does not require a dataset or training time to produce results. However, it can be computationally expensive in terms of time and memory if we aim to obtain smoother reconstructions on surfaces with more complex curvatures.

We also evaluated three classical approaches for surface completion. The first approach, MeshFix, as proposed by [5], can produce a triangulated surface with similar triangle density, but it is not effective in dealing with larger surfaces that have significant curvature variations. The second approach, called Ramesh [17], works well for small pieces, but it fails to reconstruct larger holes and often produces unappealing results due to the creation of big triangles. The third approach we considered is the screened Poisson surface reconstruction method proposed by [16]. This approach produces smooth surfaces, but it is unable to accurately represent the curvature complexities of the input surface. Overall, each of these classical approaches has its own limitations and may not be suitable for complex surface completion tasks.

The comparison chart in Fig. 9 illustrates the effectiveness of our proposed approach. An important advantage of our approach is that it does not require the design and training of a neural network, nor does it rely on a dataset for training. This makes our approach more accessible and applicable for surface completion purposes. The chart illustrates that our approach has the ability to reconstruct both small and larger holes while maintaining the coherence of curvature values, regardless of the complexity or size of the holes to be repaired.

For instance, in the first example featuring a face model, our approach, both with DALL·E 2 or Stable Diffusion, can detect symmetries and successfully inpaint the image with the missing eye, resulting in a reconstructed surface that closely matches the original. Moreover, our approach is capable of continuing lines and shapes, as demonstrated in the Christmas bear example, where the shape of the hat is maintained during the completion process.

The last two examples further demonstrate the efficacy of our method in completing holes by detecting surrounding patterns and accurately completing the surface with very close to ground truth results, resulting in an appealing and realistic reconstruction. Altogether, our proposed approach outperforms the other compared methods, demonstrating its potential for surface completion applications in different contexts.

Evaluation of the approach with a benchmark. In line with the suggestions put forward by Williams et al. in their work [32], we conducted an evaluation of our approach using a widely recognized surface reconstruction benchmark proposed by Berger et al. [42]. This benchmark comprises five models that exhibit intricate features, providing a robust evaluation platform for point cloud completion methods. To adapt these models for our evaluation, we employed the Screened Poisson algorithm [16] to triangulate them, transforming them into surface meshes with holes that require repair.

Our experimentation yielded promising qualitative results when compared to the approach presented in [32], effectively fulfilling our objective of completing complex surface geometries. As depicted in Fig. 10, it is evident that the compared approach struggles to fill the designated holes, while our proposed approach succeeds in generating visually appealing completions. However, it is important to note that our approach does have certain limitations arising from the coarse nature of the repair process. For instance, in the anchor model, although every hole is successfully repaired, the resulting Figure may not appear realistic because some of the holes are naturally present in the original model. This occurs due to the insufficiency of the partial scan provided as input, which does not provide enough context to prevent our approach from filling unnecessary holes.

Nonetheless, our approach demonstrates excellent performance when applied to meshes with genus 0 or those with more organic forms such as faces or animals, as exemplified by the

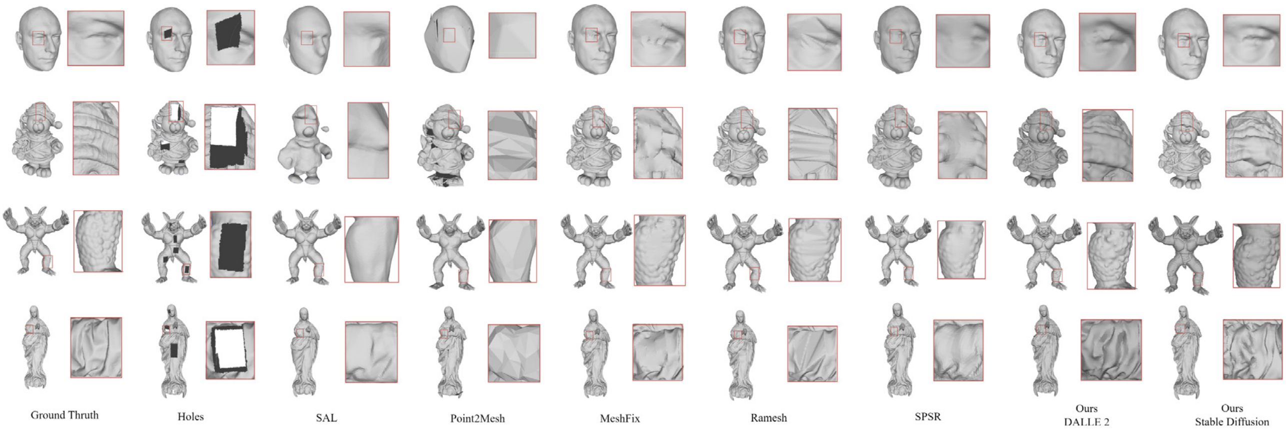


Fig. 9. Comparison between reconstructed models using deep learning based methods (SAL [25], Point2Mesh [10]), classical methods (MeshFix [5], Ramesh [17], SPSR [16]) and our approach.

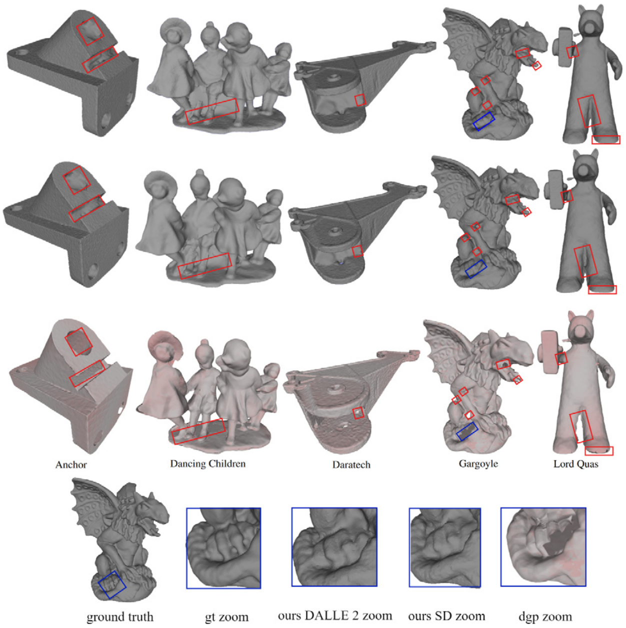


Fig. 10. Comparison between reconstructed benchmark models using our approach with Stable Diffusion (top row), with DALL·E 2 (second row) and deep-geometric-prior [32] approach (third row). Bottom row represents a zoom of a hole reconstruction comparing our approach to deep geometric-prior-approach.

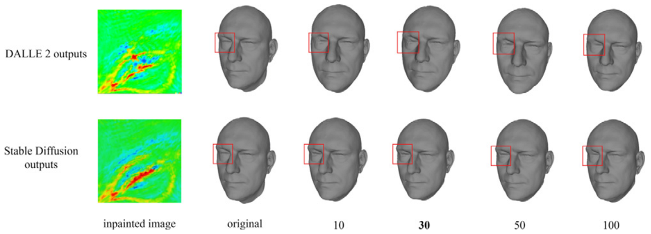


Fig. 11. Different reconstructions depending on the similitude parameter value.

Gargoyle and Lord Quas models. These cases showcase the effectiveness of our approach in handling surfaces with intricate geometries, yielding satisfactory completions.

In the quantitative evaluation of the benchmark results, we have encountered negative outcomes for our approach, despite successfully repairing every hole in the mesh. This discrepancy arises from the complete deformation technique employed to infer the new geometry on the surface. As a result, the traditional quantitative distance measures used in the evaluation may not accurately reflect the performance of our approach. We consider that these quantitative distance measures are not relevant and have not been added to the section. However, in Section 4.2 we provide extended distance measures and a discussion of the results.

Estimating parameters qualitatively. Our proposed method involves the utilization of several parameters that play a crucial role in achieving varying levels of precision in the surface reconstruction process. These parameters include the “similitude parameter” and the “transformation constant”, both of which significantly impact the final quality of the reconstructed surface.

The similitude parameter serves as a threshold that determines when a vertex in the mesh needs to be modified to ensure a fair reconstruction. In order to determine the optimal value for this parameter, we conducted a qualitative assessment, as depicted in Fig. 11. Through this evaluation, we aimed to identify the similitude parameter value that produces the most comprehensive reconstruction. Our qualitative analysis indicates that values of 10 and 30 exhibit optimal results, showcasing a natural-looking outcome. It is important to note that higher values of the similitude parameter result in smoother reconstructions, thus reducing accuracy.

In addition to the similitude parameter, the transformation constant is another crucial parameter in our method that significantly impacts the quality of the reconstructed surface. This parameter determines the displacement applied to each vertex in a normalized mesh to achieve the desired surface curvature.

To determine the optimal value for the transformation constant, we also performed a qualitative analysis, as depicted in Fig. 12. The purpose of this evaluation was to identify the transformation constant value that produces the best qualitative results in terms of the reconstructed surface. Through this analysis, we observed that a value of 0.00005 yielded the most favorable outcomes.

When the transformation constant has larger values, it results in a more pronounced displacement of the vertices, which can lead to a sharper and less smooth reconstructed surface. This loss of smoothness can adversely affect the overall quality and visual appeal of the reconstruction. Hence, it is crucial to choose an appropriate transformation constant to strike a balance between achieving the desired surface curvature and maintaining a smooth and visually pleasing result.

Results with different DALL·E 2 variations. Our approach has successfully reconstructed complex surfaces, producing natural

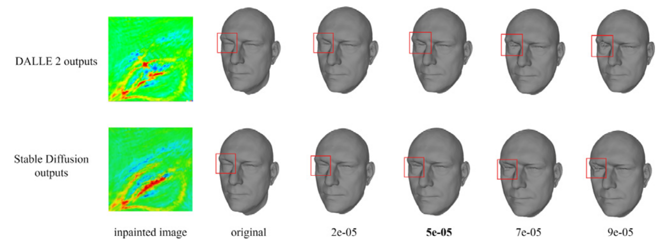
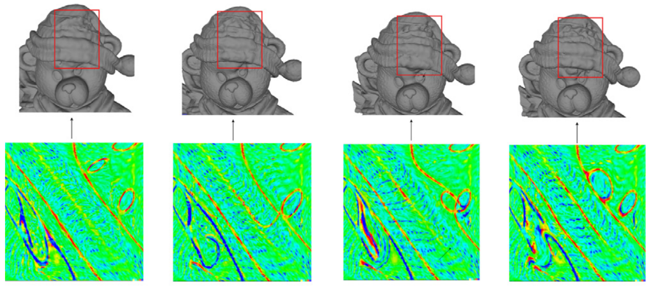
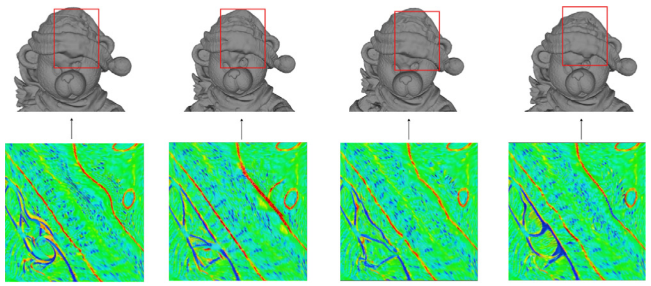


Fig. 12. Different reconstructions depending on the transformation constant value.



prompt: *"fill in the blanks"*

Fig. 13. Four different image variations produced by DALL·E 2 and its reconstructions.



prompt: *"Color map representing curvatures made by green, red, and blue gradient zones, match the surroundings, diffuse colors."*

Fig. 14. Four different image variations produced by Stable Diffusion and its reconstructions.

and visually appealing results. The key to achieving such outcomes lies in the careful selection of the main parameters, as well as leveraging the capabilities of DALL·E 2. As discussed previously, these parameters include the prompt used for inpainting and the choice of a suitable variation of the inpainted image to infer new geometries on the surface.

The selection of the variation of the inpainted image plays a crucial role in the reconstruction process. An inpainting network such as DALL·E 2 or Stable Diffusion can generate variations of the same image, each with its own advantages and disadvantages. These variations may differ in color choices, forms that deviate from the original image, or the presence of artifacts that can impact the final reconstruction. Therefore, it is essential to carefully choose an image that appears coherent with the input image. This selection process is not random and requires the identification of the optimal choice among the infinite possible variations.

The importance of choosing the right variation of the inpainted image is exemplified in Figs. 13 and 14, where the difference in reconstruction outcomes is depicted. By selecting a suitable image variation, we can enhance the quality and fidelity of the reconstructed surface.

In our future research, we have an ambitious plan to advance the shape completion process by making it unsupervised and eliminating the dependency on image variations. Our goal is to train a dedicated inpainting network using curvature images, which will enable us to generate inpainted reconstructions that fulfill the necessary requirements for inferring proper new geometry over the mesh. By developing a specialized inpainting network tailored to our shape completion task, we anticipate achieving improved final reconstruction results.

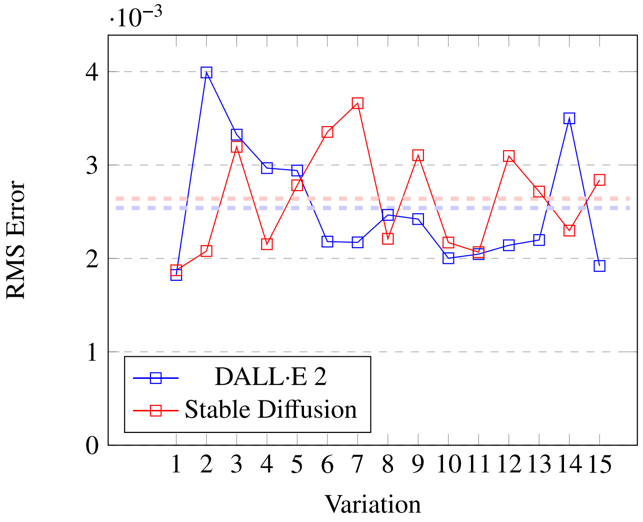


Fig. 15. Measure distances depending on random variations for DALL·E 2 and Stable Diffusion. Mean values represented as dashed lines of their corresponding color.

By adopting an unsupervised approach, we aim to reduce the reliance on manual supervision and intervention, making the shape completion process more efficient and autonomous. This will allow our method to generalize better to different datasets and scenarios, eliminating the need for manually selecting specific image variations.

However, to assess the robustness of the inpainted image and reconstruction results, we conducted an experiment to measure the root mean square (RMS) error. In Fig. 15 we present the results of this experiment, which involved comparing the original mesh with the reconstructed versions generated from 15 randomly obtained inpainting results using DALL·E 2 and Stable Diffusion on the face mesh. The measurements revealed that the RMS error exhibited minimal variation across the different inpainting results, and it consistently approached the mean value. The standard deviation for DALL·E 2 was calculated as *0.0006536202261*, while for Stable Diffusion it was found to be *0.0005571187956*. These values further support the notion that both neural networks introduce negligible errors in the reconstruction process, as they are very close to zero.

4.2. Quantitative results

Distance measures on curvature completion results. To provide a comprehensive evaluation of our approach, we compared the Hausdorff distance and the Root Mean Square (RMS) error between the original mesh and the repaired mesh using the approaches that exhibited the best qualitative results. The results of this comparison are presented in Table 1. Our distance measures are evaluated in the models only using Stable Diffusion to ensure reproducibility.

In terms of the Hausdorff distance, our method demonstrates a significantly smaller maximum distance compared to the other approaches. This indicates that our approach has fewer large errors or discrepancies when compared to the original mesh.

Table 1
Distance measures on different meshes and approaches. Our measures use Stable Diffusion inpainting.

Face mesh				Armadillo mesh			
	max	mean	RMS		max	mean	RMS
SAL	2.23e−01	3.43e−02	4.57e−02	SAL	2.10e−02	1.53e−03	1.96e−03
Ramesh	2.07e−01	4.23e−03	2.17e−02	Ramesh	1.78e−02	6.32e−04	**1.10e−03**
SPSR	2.33e−01	**6.49e−04**	6.89e−03	SPSR	5.76e−02	**6.21e−04**	3.72e−03
MeshFix	2.07e−01	5.42e−03	2.50e−02	MeshFix	2.11e−02	6.87e−04	1.20e−03
Ours SD	**1.41e−02**	1.57e−03	**2.47e−03**	Ours SD	**1.52e−02**	1.63e−03	2.07e−03
Christmas Bear mesh				Inmaculada mesh			
	max	mean	RMS		max	mean	RMS
SAL	8.96e−02	1.20e−02	1.61e−02	SAL	1.10e−01	5.15e−03	1.25e−02
Ramesh	9.32e−02	1.44e−03	7.89e−03	Ramesh	1.05e−01	1.19e−03	7.45e−03
SPSR	7.01e−02	**4.79e−04**	2.71e−03	SPSR	9.91e−02	**5.34e−04**	2.88e−03
MeshFix	9.31e−02	1.83e−03	8.16e−03	MeshFix	1.05e−01	1.76e−03	9.45e−03
Ours SD	**2.22e−02**	1.60e−03	**2.42e−03**	Ours SD	**1.45e−02**	1.35e−03	**2.00e−03**

Although our method may not have the smallest mean Hausdorff distance, primarily due to the mesh deformation process introduced to reduce noisy results, the difference between our approach and the others is not substantial. This suggests that most approaches exhibit similar overall errors or discrepancies. However, it is worth noting that the SAL method [25], which involves complete remeshing of the repaired mesh, introduces higher discrepancies compared to other approaches.

Furthermore, when considering the RMS error after the application of the Screened Poisson Surface Reconstruction (SPSR) method [16], our approach ranks among the best. The SPSR method introduces minimal vertex movement, indicating that our approach achieves smaller average distances between corresponding points of the repaired meshes and their original surfaces without holes. It is noteworthy that even in this case, our approach produces more natural and complex results, as evident in the qualitative evaluation depicted in Fig. 9.

By considering both the Hausdorff distance and the RMS error, our approach demonstrates competitive performance compared to other state-of-the-art methods. The smaller Hausdorff distance suggests that our approach successfully addresses the larger errors or discrepancies, while the smaller RMS error indicates that our approach maintains better overall correspondence between the repaired meshes and their original surfaces. These results further validate the effectiveness of our method in achieving accurate and visually appealing shape completions.

Estimating parameters quantitatively. As we did in the previous section, a quantitative analysis of the different reconstruction parameters is provided. We examined the results of distance measures to evaluate the impact of varying these parameters on the quality of the reconstructions.

In the first plot (Fig. 16), we illustrate the relationship between the similitude parameter values and the distance measures. We observe that smaller similitude values generally lead to better distance measures. This indicates that the reconstructions generated with smaller similitude parameters exhibit closer correspondence to the original surfaces without holes.

Interestingly, we also notice some lower distance values when the similitude parameter is very large. This can be attributed to the behavior of the similitude parameter as a threshold during surface reconstruction. When a large similitude parameter is chosen, fewer vertices are selected for modification, resulting in minimal mesh deformation and minimal changes to the positions of the vertices. As a result, the reconstructed surface remains coarse and less accurate. While this may yield lower distance values in terms of quantity, the reconstruction quality is compromised.

Based on the average distance measures and the qualitative reconstructions provided, we made the final selection of a similitude parameter value of 30. This value strikes a balance between achieving favorable distance measures and producing visually appealing and accurate reconstructions. It represents a threshold that allows for sufficient modification of the vertices to achieve improved surface completion while avoiding excessive mesh deformation.

These measures demonstrate how the precision of the reconstruction decreases as the transformation constant increases. This behavior is attributed to the significant displacement introduced to the vertices in the normalized mesh.

As the transformation constant grows (see Fig. 17), the vertices experience larger displacements, resulting in more pronounced deformations of the mesh. This leads to a smoother surface without sharp features, as observed in the qualitative results presented in Fig. 12, where the face becomes more deformed in various positions as the transformation constant increases.

While a larger transformation constant may produce smoother reconstructions, it also leads to a loss of precision. The excessive displacement of the vertices can result in inaccuracies and distortions in the reconstructed surface.

Considering the balance between distance measures and quality reconstructions, we have selected a transformation constant value of 0.00005. This value strikes a good compromise, as it produces reconstructions with a reasonable level of smoothness while still preserving important details and maintaining a high level of precision. This optimal combination of distance measures and reconstruction quality makes it the preferred choice for the transformation constant in our approach.

5. Conclusions and future work

This paper introduces a novel approach for shape completion using deep learning techniques that address the limitations associated with existing methods. Deep learning-based shape completion techniques require the design of a neural network and the collection of extensive datasets for training. However, our approach employs DALL·E 2 or Stable Diffusion inpainting process to recover missing parts of images that represent the curvature features of a mesh. This enables us to infer new geometries based on the inpainted curvature values, eliminating the need for a dedicated neural network and large amounts of data for training. We highlight the numerous advantages of using an external neural network, including improved accuracy and flexibility.

To evaluate the effectiveness of our method, we conducted qualitative and quantitative assessments of complex holes in various 3D models. Our approach was able to complete both sharp and curving shapes, preserve symmetries, and fill patterns accurately. We believe that our approach could pave the way for further advancements in shape completion, particularly in scenarios where limited data is available or where there is a need for fast completion of complex shapes.

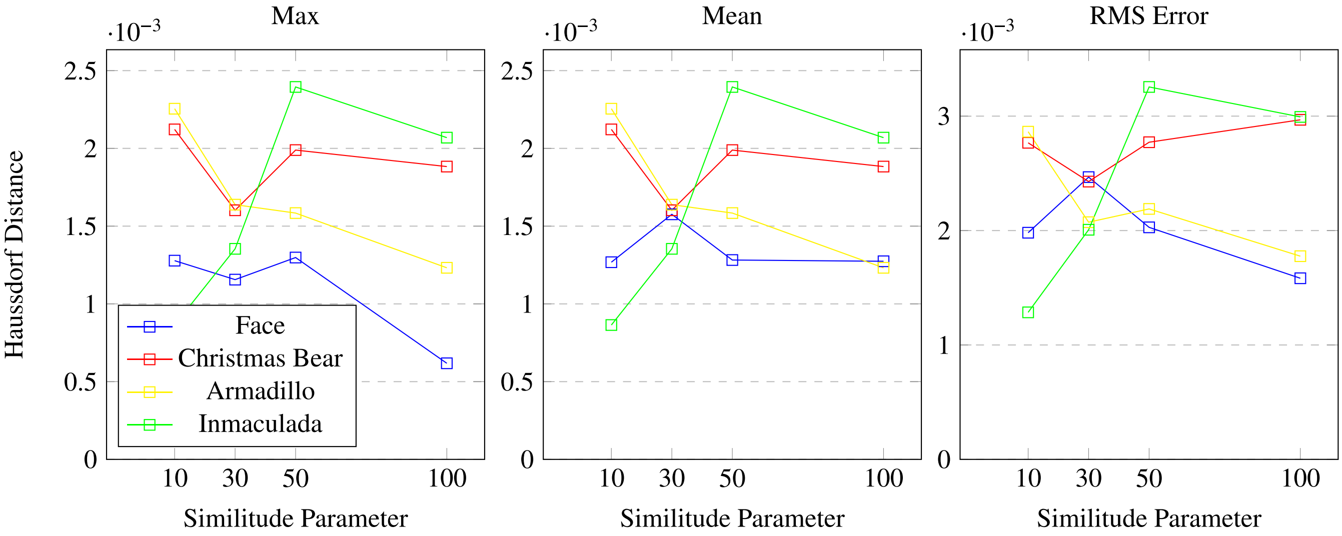


Fig. 16. Measure distances depending on the similitude parameter value. These reconstruction distance measures use Stable Diffusion inpainting.

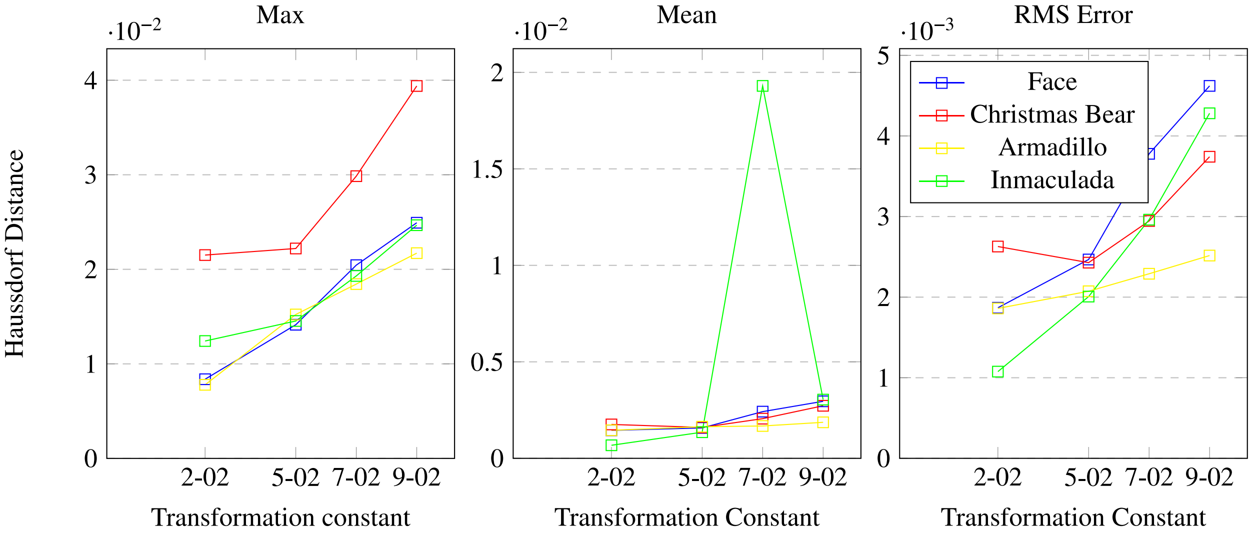


Fig. 17. Measure distances depending on the transformation constant value. These reconstruction distance measures use Stable Diffusion inpainting.

In the methodology section, we made significant improvements to provide a more accurate explanation of each step of our approach. Additionally, we expanded the experimentation section to include new models for evaluation. By incorporating more complex and challenging 3D models, we were able to assess the performance and effectiveness of our approach across a wider range of scenarios. Furthermore, we introduced the estimation of different parameters in our approach, such as the similitude parameter and the transformation constant. By evaluating the effects of these parameters on the reconstruction results, we provided quantitative assessments that shed light on the optimal choices for achieving comprehensive and accurate shape completions. This analysis allowed us to offer guidance and recommendations for selecting suitable parameter values for future applications.

One significant drawback of our current approach is the limited control over the output generated by models like DALL·E 2 or Stable Diffusion. Since these models are pretrained, we do not have complete control over their design and training process. While we can engage in prompt engineering and tweak prompts to achieve more accurate inpainting results, we face a challenge in incorporating specific cultural, historical, or contextual information into the final image.

In the near future, our plan is to develop an ad-hoc neural-network architecture trained with a database of complex 3D models, with the aim of transferring information from one 3D model to others and inpaint the surfaces with a common-domain source of data, so we acquire control over the possible results of the inpainting. On the other hand, we plan to test the behavior of the algorithm with images with coarser curvature values, avoiding such fine-grained spots of curvature values.

CRediT authorship contribution statement

Marina Hernández-Bautista: Conceptualization, Methodology, Software, Investigation, Writing – original draft. **Francisco Javier Melero:** Conceptualization, Writing – review & editing, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Acknowledgments

This work was supported by the Spanish Ministry of Science and Technology under projects PID2020-119478GB-I00 and TED2021-132702B-C21 financed by MCIN/AEI/10.13039/501100011033 and European Regional Development Fund (ERDF).

References

[1] Davis J, Marschner SR, Garr M, Levoy M. Filling holes in complex surfaces using volumetric diffusion. In: Proceedings. First international symposium on 3d data processing visualization and transmission. IEEE; 2002, p. 428–41.
[2] Park S, Guo X, Shin H, Qin H. Surface completion for shape and appearance. Vis Comput 2006;22:168–80.
[3] Guo TQ, Li JJ, Weng JG, Zhuang YT. Filling holes in complex surfaces using oriented voxel diffusion. In: 2006 international conference on machine learning and cybernetics. IEEE; 2006, p. 4370–5.
[4] Liepa P. Filling holes in meshes. In: Proceedings of the 2003 eurographics/ACM SIGGRAPH symposium on geometry processing. 2003 p. 200–5.
[5] Attene M. A lightweight approach to repairing digitized polygon meshes. Vis Comput 2010;26:1393–406.
[6] Brunton A, Wuhrer S, Shu C, Bose P, Demaine ED. Filling holes in triangular meshes by curve unfolding. In: 2009 IEEE international conference on shape modeling and applications. IEEE; 2009, p. 66–72.
[7] Yuan W, Khot T, Held D, Mertz C, Hebert M. Pcn: Point completion network. In: 2018 international conference on 3D vision. IEEE; 2018, p. 728–37.
[8] Chu L, Pan H, Wang W. Unsupervised shape completion via deep prior in the neural tangent kernel perspective. ACM Trans Graph 2021;40(3):1–17.
[9] Stutz D, Geiger A. Learning 3D shape completion from laser scan data with weak supervision. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2018, p. 1955–64.
[10] Hanocka R, Metzer G, Giryes R, Cohen-Or D. Point2mesh: A self-prior for deformable meshes. 2020, arXiv preprint arXiv:2005.11084.
[11] Ramesh A, Dhariwal P, Nichol A, Chu C, Chen M. Hierarchical text-conditional image generation with clip latents. 2022, arXiv preprint arXiv:2204.06125.
[12] Rombach R, Blattmann A, Lorenz D, Esser P, Ommer B. High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2022, URL https://github.com/CompVis/latent-diffusionhttps://arxiv.org/abs/2112.10752.
[13] Floater MS. Mean value coordinates. Comput Aided Geom Design 2003;20(1):19–27.
[14] Guennebaud G, Gross M. Algebraic point set surfaces. In: ACM Siggraph 2007 papers. 2007, p. 23–es.
[15] Hernandez-Bautista M, Melero FJ. Deep learning surface completion based on curvature features. In: XXXII Spanish computer graphics conference. 2023.
[16] Kazhdan M, Hoppe H. Screened poisson surface reconstruction. ACM Trans Graph 2013;32(3):1–13.
[17] Centin M, Signoroni A. RameshCleaner: Conservative fixing of triangular meshes. The Eurographics Association; 2015.
[18] Botsch M, Kobbelt L. A remeshing approach to multiresolution modeling. In: Proceedings of the 2004 eurographics/ACM SIGGRAPH symposium on geometry processing. 2004, p. 185–92.
[19] De Floriani L, Puppo E. An on-line algorithm for constrained delaunay triangulation. CVGIP, Graph Models Image Process 1992;54(4):290–300.
[20] Qiang H, Shusheng Z, Xiaoliang B, Xin Z. Hole filling based on local surface approximation. In: 2010 international conference on computer application and system modeling, vol. 3. IEEE; 2010, p. V3–242.
[21] Fang TP, Piegl LA. Delaunay triangulation in three dimensions. IEEE Comput Graph Appl 1995;15(5):62–9.
[22] Harary G, Tal A, Grinspun E. Context-based coherent surface completion. ACM Trans Graph 2014;33(1):1–12.
[23] Vichitvejpaisal P, Kanongchaiyos P. Surface completion using Laplacian transform. Eng J 2014;18(1):129–44.
[24] Park JJ, Florence P, Straub J, Newcombe R, Lovegrove S. Deepsdf: Learning continuous signed distance functions for shape representation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019, p. 165–74.
[25] Atzmon M, Lipman Y. Sal: Sign agnostic learning of shapes from raw data. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020, p. 2565–74.
[26] Geiger A, Lenz P, Urtasun R. Are we ready for autonomous driving? the kitti vision benchmark suite. In: 2012 IEEE conference on computer vision and pattern recognition. IEEE; 2012, p. 3354–61.
[27] Chang AX, Funkhouser T, Guibas L, Hanrahan P, Huang Q, Li Z, et al. Shapenet: An information-rich 3d model repository. 2015, arXiv preprint arXiv:1512.03012.
[28] Wu J, Zhang C, Zhang X, Zhang Z, Freeman WT, Tenenbaum JB. Learning shape priors for single-view 3d completion and reconstruction. In: Proceedings of the European conference on computer vision. 2018 p. 646–62.
[29] Dai A, Ruizhongtai Qi C, Nießner M. Shape completion using 3D-encoder-predictor cnns and shape synthesis. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2017, p. 5868–77.
[30] Wang X, Xu D, Gu F. 3D model inpainting based on 3D deep convolutional generative adversarial network. IEEE Access 2020;8:170355–63.
[31] Zhang J, Chen X, Cai Z, Pan L, Zhao H, Yi S, et al. Unsupervised 3D shape completion through gan inversion. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2021 p. 1768–77.
[32] Williams F, Schneider T, Silva C, Zorin D, Bruna J, Panozzo D. Deep geometric prior for surface reconstruction. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019 p. 10130–9.
[33] Sarmad M, Lee HJ, Kim YM. Rl-gan-net: A reinforcement learning agent controlled gan network for real-time point cloud shape completion. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019, p. 5898–907.
[34] Efros AA, Freeman WT. Image quilting for texture synthesis and transfer. In: Proceedings of the 28th annual conference on computer graphics and interactive techniques. 2001, p. 341–6.
[35] Liang L, Liu C, Xu YQ, Guo B, Shum HY. Real-time texture synthesis by patch-based sampling. ACM Trans Graph 2001;20(3):127–50.
[36] Bertalmio M, Sapiro G, Caselles V, Ballester C. Image inpainting. In: Proceedings of the 27th annual conference on computer graphics and interactive techniques. 2000, p. 417–24.
[37] Maggiordomo A, Cignoni P, Tarini M. Texture inpainting for photogrammetric models. In: Computer graphics forum. Wiley Online Library; 2023.
[38] Zeng Y, Lin Z, Lu H, Patel VM. Cr-fill: Generative image inpainting with auxiliary contextual reconstruction. In: Proceedings of the IEEE/CVF international conference on computer vision. 2021, p. 14164–73.
[39] Yu J, Lin Z, Yang J, Shen X, Lu X, Huang TS. Generative image inpainting with contextual attention. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2018, p. 5505–14.
[40] Radford A, Kim JW, Hallacy C, Ramesh A, Goh G, Agarwal S, et al. Learning transferable visual models from natural language supervision. In: International conference on machine learning. PMLR; 2021, p. 8748–63.
[41] Chao I, Pinkall U, Sanan P, Schröder P. A simple geometric model for elastic deformations. ACM Trans Graph 2010;29(4):1–6.
[42] Berger M, Levine JA, Nonato LG, Taubin G, Silva CT. A benchmark for surface reconstruction. ACM Trans Graph 2013;32(2):1–17.