Contents lists available at ScienceDirect

Computers & Graphics

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

Combining Voxel and Normal Predictions for Multi-View 3D Sketching

Johanna Delanoy[a], David Coeurjolly[b], Jacques-Olivier Lachaud[c], Adrien Bousseau[d]

[a] *Inria, Université Côte d'Azur*
[b] *Université de Lyon, CNRS, LIRIS*
[c] *Laboratoire de Mathématiques (LAMA), UMR 5127 CNRS, Université Savoie Mont Blanc*
[d] *Inria, Université Côte d'Azur*

ARTICLE INFO

Article history:
Received May 2, 2019

Keywords: Sketch-Based Modeling, Deep Learning, Surface Regularization, Voxels, Normal Map

ABSTRACT

Recent works on data-driven sketch-based modeling use either voxel grids or normal/depth maps as geometric representations compatible with convolutional neural networks. While voxel grids can represent complete objects – including parts not visible in the sketches – their memory consumption restricts them to low-resolution predictions. In contrast, a single normal or depth map can capture fine details, but multiple maps from different viewpoints need to be predicted and fused to produce a closed surface. We propose to combine these two representations to address their respective shortcomings in the context of a multi-view sketch-based modeling system. Our method predicts a voxel grid common to all the input sketches, along with one normal map per sketch. We then use the voxel grid as a support for normal map fusion by optimizing its extracted surface such that it ~~best agrees~~is consistent with the re-projected normals, while being as piecewise-smooth as possible overall. We compare our method with a recent voxel prediction system, demonstrating improved recovery of sharp features over a variety of man-made objects.

1. Introduction

As many related fields, sketch-based modeling recently witnessed major progress thanks to deep learning. In particular, several authors demonstrated that generative convolutional networks can predict 3D shapes from one or several line drawings [1, 2, 3, 4]. A common challenge faced by these methods is the choice of a geometric representation that can both represent the important features of the shape while also being compatible with convolutional neural networks. Voxel grids form a natural 3D extension to images, and were used by Delanoy et al. [1] to predict a complete object from as little as one input drawing. This complete prediction allows users to rotate around the 3D shape before creating drawings from other viewpoints. However, the memory consumption of voxel grids limits their resolution, resulting in smooth surfaces that lack details. Alternatively, several methods adopt image-based representations, predicting depth and normal maps from one or several drawings [2, 3, 4]. While these maps can represent finer details than voxel grids, each map only shows part of the surface, and multiple maps from different viewpoints need to be fused to produce a closed object.

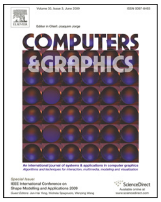
Motivated by the complementary strengths of voxel grids and normal maps, we propose to combine both representations within the same system. Our approach builds on the voxel prediction network of Delanoy et al. [1], which produces a volumetric prediction of a shape from one or several sketches. We complement this architecture with a normal prediction network similar to the one used by Su et al. [4], which we use to obtain a normal map for each input sketch. The voxel grid thus provides us with a complete, closed surface, while the normal maps allow us to recover details in the parts seen from the sketches.

Our originality is to not only use the voxel grid as a preliminary prediction to be shown to the user, but also as a support for normal map fusion. To do so, we first locate the voxels delin-

Contents lists available at ScienceDirect

Computers & Graphics

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



Combining Voxel and Normal Predictions for Multi-View 3D Sketching

ARTICLE INFO

Article history:
Received May 22, 2019

Keywords: Sketch-Based Modeling, Deep Learning, Surface Regularization, Voxels, Normal Map

ABSTRACT

Recent works on data-driven sketch-based modeling use either voxel grids or normal/depth maps as geometric representations compatible with convolutional neural networks. While voxel grids can represent complete objects – including parts not visible in the sketches – their memory consumption restricts them to low-resolution predictions. In contrast, a single normal or depth map can capture fine details, but multiple maps from different viewpoints need to be predicted and fused to produce a closed surface. We propose to combine these two representations to address their respective shortcomings in the context of a multi-view sketch-based modeling system. Our method predicts a voxel grid common to all the input sketches, along with one normal map per sketch. We then use the voxel grid as a support for normal map fusion by optimizing its extracted surface such that it is consistent with the re-projected normals, while being as piecewise-smooth as possible overall. We compare our method with a recent voxel prediction system, demonstrating improved recovery of sharp features over a variety of man-made objects.

1. Introduction

As many related fields, sketch-based modeling recently witnessed major progress thanks to deep learning. In particular, several authors demonstrated that generative convolutional networks can predict 3D shapes from one or several line drawings [1, 2, 3, 4]. A common challenge faced by these methods is the choice of a geometric representation that can both represent the important features of the shape while also being compatible with convolutional neural networks. Voxel grids form a natural 3D extension to images, and were used by Delanoy et al. [1] to predict a complete object from as little as one input drawing. This complete prediction allows users to rotate around the 3D shape before creating drawings from other viewpoints. However, the memory consumption of voxel grids limits their resolution, resulting in smooth surfaces that lack details. Alternatively, several methods adopt image-based representations, predicting depth and normal maps from one or several drawings [2, 3, 4]. While these maps can represent finer details than voxel grids, each map only shows part of the surface, and multiple maps from different viewpoints need to be fused to produce a closed object.

Motivated by the complementary strengths of voxel grids and normal maps, we propose to combine both representations within the same system. Our approach builds on the voxel prediction network of Delanoy et al. [1], which produces a volumetric prediction of a shape from one or several sketches. We complement this architecture with a normal prediction network similar to the one used by Su et al. [4], which we use to obtain a normal map for each input sketch. The voxel grid thus provides us with a complete, closed surface, while the normal maps allow us to recover details in the parts seen from the sketches.

Our originality is to not only use the voxel grid as a preliminary prediction to be shown to the user, but also as a support for normal map fusion. To do so, we first locate the voxels delineating the object's boundary, and re-project the normal maps on the resulting surface to obtain a distribution of candidate normals for each surface element. We then solve for the smoothest normal field that best agrees with these observations [5]. Finally, we optimize the surface elements to best align with this normal field [6]. We evaluate our approach on the dataset of Delanoy et al. [1], on which we recover smoother surfaces with sharper discontinuities.

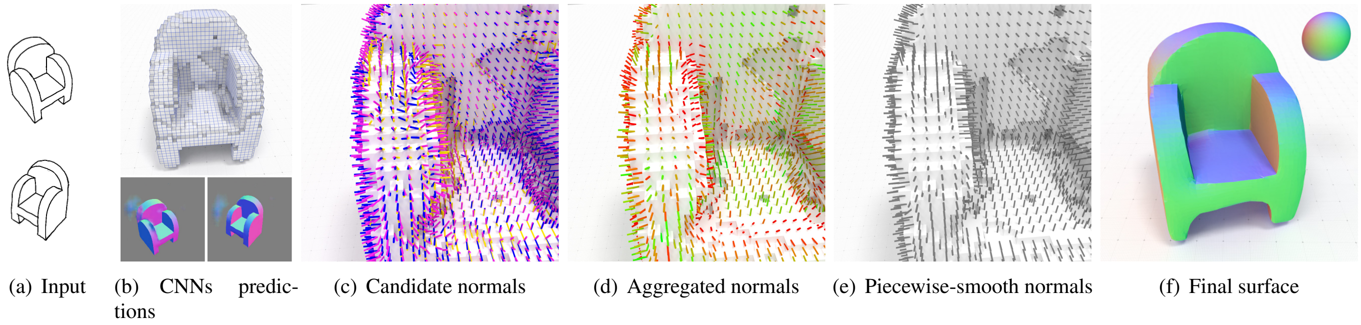

(a) Input (b) CNNs predictions (c) Candidate normals (d) Aggregated normals (e) Piecewise-smooth normals (f) Final surface

Fig. 1. Overview of our method. Our method takes as input multiple sketches of an object (a). We first apply existing deep neural networks to predict a volumetric reconstruction of the shape as well as one normal map per sketch (b). We re-project the normal maps on the voxel grid (c, blue needles for the first normal map, yellow needles for the second normal map), which complement the surface normal computed from the volumetric prediction (c, pink needles). We aggregate these different normals into a distribution represented by a mean vector and a standard deviation (d, colors denote low variance in green and high variance in red). We optimize this normal field to make it piecewise smooth (e) and use it to regularize the surface (f). The final surface preserves the overall shape of the predicted voxel grid as well as the sharp features of the predicted normal maps.

2. Related work

Reconstructing 3D shapes from line drawings has a long history in computer vision and computer graphics. A number of methods tackle this problem by geometric means, for instance by detecting and enforcing 3D relationships between lines, like parallelism and orthogonality [7, 8, 9, 10, 11]. However, computing these geometric constraints often require access to a clean, well-structured representation of the drawing, for instance in the form of a graph of vectorial curves. In addition, geometric methods often require user annotations to disambiguate multiple interpretations, or to deal with missing information.

Data-driven methods hold the promise to lift the above limitations by providing strong priors on the shapes that a drawing can represent. In particular, recent work exploit deep neural networks to predict 3D information from as little as a single bitmap line drawing. However, convolutional neural networks have been originally developed to work on images, and several alternative solutions have been proposed to adapt such architectures to produce 3D shapes.

A first family of methods focuses on parametric shapes such as buildings [12], trees [13], and faces [14], and train deep networks to regress their parameters. While these methods produce 3D shapes of very high quality, extending them to new classes of objects require designing novel parametric models by hand.

A second family of methods target arbitrary shapes and rely on encoder-decoder networks to convert the input drawing into 3D representations. Among them, Delanoy et al. [1] rely on a voxel grid to represent a complete object. Users of their system can thus visualize the 3D shape, including its hidden parts, as soon as they have completed a single drawing. Their system also supports additional drawings created from different viewpoints, which allow the network to refine its prediction. Nevertheless, their system is limited to voxel grids of resolution 64^3, which is too little to accurately capture sharp features. Alternatively, Su et al. [4] and Li et al. [3] propose encoder-decoder networks to predict normal and depth maps respectively. While these maps only represent the geometry visible in the input drawing, Li et al. allow users to draw the object from several viewpoints and fuse the resulting depth maps to obtain a complete object. A similar image-based representation has been proposed by Lun et al. [2], who designed a deep network to predict depth maps from 16 viewpoints, given one to three drawings as input. In both cases, fusing the multiple depth maps requires careful point set registration and optimization to compensate for misalignment. Our approach combines the strength of both voxel-based and image-based representations. On the one hand, per-sketch normal maps provide high-resolution details about the shape, while on the other hand, the voxel grid provides an estimate of the complete shape as well as a support surface for normal fusion. By casting normal fusion as the reconstruction of a piecewise-smooth normal field over the voxel surface, our method alleviates the need for precise alignment of the normal maps.

Line drawing interpretation is related to the problem of 3D reconstruction from photographs, for which numerous deep-learning solutions have been proposed by the computer vision community. While many approaches rely on voxel-based [15, 16] and image-based [17] representations as discussed above, other representations have been proposed to achieve finer reconstructions. Octrees have long been used to efficiently represent volumetric data, although their implementation in convolutional networks requires the definition of custom operations, such as convolutions on hash tables [18] or cropping of octants [19]. Point sets have also been considered as an alternative to voxel-based or image-based representations [20], and can be converted to surfaces in a post-process as done for depth map fusion. More recently, several methods attempted to directly predict surfaces. Pixel2Mesh [21] relies on graph convolutional networks [22] to predict deformations of a template mesh. However, this approach is limited to shapes that share the same topology as the template, an ellipsoid in their experiments. In contrast, Groueix et al. [23] can handle arbitrary topology by predicting multiple surface patches that cover the shape. Since these patches do not form a single, closed surface, their approach can also be used to generate a dense point set from which a surface can be computed as a post-process. In contrast to the above approaches, we chose to combine voxel-based and image-based representations because both can be implemented using standard convolutional networks on regular grids.

3. Overview

Our method takes as input several sketches of a shape drawn from different known viewpoints (Figure 1a). We first use existing deep neural networks [1, 24] to predict a volumetric reconstruction of the shape, along with one normal map per sketch (Figure 1b). We then project the normal maps on the surface of the volumetric reconstruction and combine this information with the initial surface normal to obtain a distribution of normals for each surface element (Figure 1c,d). While the normals coming from different sources are mostly consistent, some parts of the shape exhibit significant ambiguity due to erroneous predictions and misalignment between the input sketches and the volumetric reconstruction. Therefore in the next step of our approach we reconstruct a piecewise-smooth normal field by a variational method [5] that filters the distribution of normals and locates sharp surface discontinuities (Figure 1e). The reconstruction energy is weighted by the variance of the distribution of normal vectors within each surface element, which acts as a confidence estimate. Finally, we regularize the initial surface such that its quads and edges align with this normal field [6], resulting in a piecewise-smooth object that follows the overall shape of the volumetric prediction as well as the crisp features of the predicted normal maps (Figure 1f).

4. Volumetric and normal prediction

Our method builds on prior work to obtain its input volumetric and image-based predictions of the shape. Here we briefly describe these two types of prediction and refer the interested reader to the original papers for additional details.

4.1. Volumetric prediction

We obtain our volumetric prediction using the method of Delanoy et al. [1]. Their approach relies on two deep convolutional networks. First, the *single-view* network is in charge of predicting occupancy in a voxel grid given one drawing as input. Then, the *updater* network refines this prediction by taking another drawing as input. When multiple drawings are available, the updater network is applied iteratively over the sequence of drawings to achieve a multi-view coherent reconstruction. Both networks follow a standard U-Net architecture [25] where the drawing is processed by a series of convolution, non-linearity and down-scaling operations before being expanded back to a voxel grid, while skip-connections propagate information at multiple scales. This method produces a voxel grid of resolution 64^3 from drawings of resolution 256^2.

4.2. Normal prediction

We obtain our normal prediction using a U-Net similar to the one we use for volumetric prediction. The network takes as input a drawing of resolution 256^2 and predicts a normal map of the same resolution. Lun et al. [2] and Su et al. [4] have shown that this type of architecture performs well on the task of normal prediction from sketches. We base our implementation on Pix2Pix [24], from which we remove the discriminator network for simplicity.

5. Data fusion

The main novelty of our method is to combine a coarse volumetric prediction with per-view normal maps to recover sharp surface features. However, these different sources of information are often not perfectly aligned due to errors in the predictions as well as in the input line drawings. Prior work on multi-view prediction of depth maps [2, 3] tackle a similar challenge by aligning the corresponding point sets using costly iterative non-rigid registration. We instead implement this data fusion in two stages, each one being the solution of a different variational formulation that is fast to compute.

In the first stage, we project the normal predictions onto the surface of the volumetric prediction, and complement this information with normals estimated directly from the voxel grid. We then solve for the piecewise-smooth normal field that is most consistent with all these candidate normals, such that sharp surface discontinuities automatically emerge at their most likely locations [5]. In the second stage, we optimize the surface of the voxel grid such that it respects the normal field resulting from the first stage, while staying close to the initial predicted voxel geometry [6].

5.1. Generation of the candidate normal field

We begin by thresholding the volumetric prediction to obtain a binary voxel grid. The boundary of this collection of voxels forms a quadrangulated surface Q made of isothetic unit squares, which we call *surface elements* in the following. We then project the center of each surface element into each normal map where it appears to look up the corresponding predicted normal. We compute this projection using the camera matrix associated to each sketch, which we assume to be given as input to the method. Interactive sketching systems like the one described by Delanoy et al. [1] provide these matrices by construction. We use a simple depth test to detect if a given surface element is visible from the point of view of the normal map. We also compute the gradient of the volumetric prediction using finite differences, which we use as an additional estimate of the surface normal. We aggregate these various estimates into a spherical Gaussian distribution, with normalized mean $\bar{\mathbf{n}}$ and standard deviation $\sigma_{\mathbf{n}}$. For surface elements not visible in any normal map, we set $\bar{\mathbf{n}}$ to the estimate given by the volumetric prediction.

5.2. Reconstruction of a piecewise-smooth normal vector field

For each surface element, we now have a unique normal vector $\bar{\mathbf{n}}$ as well as an estimate of its standard deviation $\sigma_{\mathbf{n}}$. We obtain our final piecewise-smooth normal field $\mathbf{n}^*$ by minimizing a discrete variant of the Ambrosio-Tortorelli energy [5].

On a manifold Ω, the components $\{n_0^*, n_1^*, n_2^*\}$ of $\mathbf{n}^*$ and a scalar function v that captures discontinuities are optimized to minimize

$$AT_\varepsilon(\mathbf{n}^*, v) := \int_\Omega \alpha \sum_i |n_i^* - \bar{n}_i|^2 + \sum_i v^2 |\nabla n_i^*|^2 + \lambda\varepsilon|\nabla v|^2 + \frac{\lambda}{\varepsilon}\frac{(1-v)^2}{4} ds\,, \qquad (1)$$

for some parameters $\alpha, \lambda, \varepsilon \in \mathbb{R}$. Note that the scalar function v tends to be close to 0 along sharp features and close to 1 elsewhere.

The first term ensures that the output normal $\mathbf{n}^*$ is close to the input $\bar{\mathbf{n}}$. The second term encourages $\mathbf{n}^*$ to be smooth where there is no discontinuity. The last two terms control the smoothness of the discontinuity field v and encourage it to be close to 1 almost everywhere by penalizing its overall length. Note that fixing all the n_i^* (resp. v), the functional becomes quadratic and its gradient is linear in v (resp. all the n_i^*), leading to an efficient alternating minimization method to obtain the final $\mathbf{n}^*$ and v. Parameter α controls the balance between data fidelity and smoothness. A high value better preserves the input while a low value produces a smoother field away from discontinuities. Parameter λ controls the length of the discontinuities – the smaller it is, the more discontinuities will be allowed on the surface. We use the same value $\lambda = 0.05$ for all our results. The last parameter ε is related to the Γ-convergence of the functional and decreases during the optimization. We used the sequence $(4, 2, 1, 0.5)$ for all our results. Please refer to [5] for more details about the discretization of Equation (1) onto the digital surface Q and its minimization.

We further incorporate our knowledge about the distribution of normals at each surface element by defining α as a function of the standard deviation $\sigma_{\mathbf{n}}$. Intuitively, we parameterize α such that it takes on a low value over elements of high variance, effectively increasing the influence of the piecewise-smoothness term in those areas:

$$\alpha(s) := 0.2(1 - \sigma_{\mathbf{n}}(s))^4\,.$$

at a surface element $s \in Q$. This local weight allows the Ambrosio-Tortorelli energy to diffuse normals from reliable areas to ambiguous ones. We set $\alpha(s)$ to 0.8 for surface elements not visible in any normal map.

5.3. Surface reconstruction

Equipped with a piecewise-smooth normal field $\mathbf{n}^*$, we finally reconstruct a regularized surface whose quads are as close to orthogonal to the prescribed normals as possible. We achieve this goal using the variational model proposed in [6]. As illustrated in Figure 2, this surface reconstruction guided by our piecewise-smooth normal vector field effectively aligns quad edges with sharp surface discontinuities.

6. Evaluation

We first study the impact of the different components of our method, before comparing it against prior work. For all these results, we use the dataset provided by Delanoy et al. [1] to train the neural networks. This dataset is composed of abstract shapes assembled from cuboids and cylinders, along with line drawings rendered from front, side, top and 3/4 views. Note however that we only train and use the normal map predictor on 3/4 views because the other views are often highly ambiguous.

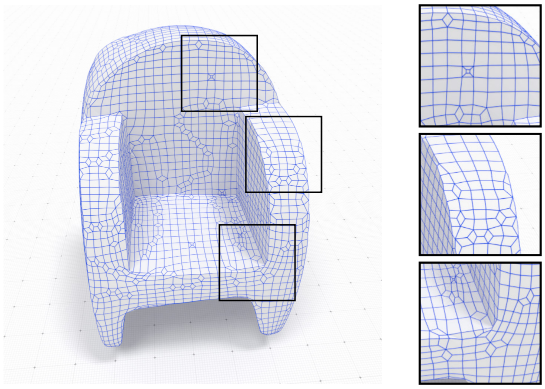


Fig. 2. Surface reconstruction obtained from the normal field regularized with our weighted Ambrosio-Tortorelli functional (see Fig.1b for the input voxel grid). The insets show how the quadrangulation perfectly recovers the surface singularities.

6.1. Ablation study

Figure 3 compares the surface reconstructions obtained with different sources of normal guidance, and different strategies of normal fusion. We color surfaces according to their orientations, as shown by the sphere in inset. As a baseline, we first extract the surface best aligned with the gradient of the volumetric prediction, similarly to prior work [1]. Because the volumetric prediction is noisy and of low resolution, this naive approach produces bumpy surfaces that lack sharp features (second column). Optimizing the normal field according to the Ambrosio-Tortorelli energy removes some of the bumps, but still produces rounded corners (third column). Aggregating the volumetric and image-based normals into a single normal field produces smoother surfaces, but yield bevels where the normal maps are misaligned (fourth and fifth column). We improve results by weighting the aggregated normal field according to its confidence, which gives the Ambrosio-Tortorelli energy greater freedom to locate surface discontinuities in ambiguous areas (last column).

We further evaluate the importance of our local weighting scheme in Figure 4. We first show surfaces obtained using a constant α in the Ambrosio-Tortorelli energy. A low α produces sharp creases and smooth surfaces but the final shape deviates from the input, as seen on the cylindrical lens of the camera that becomes conic (Figure 4b). On the other hand, a high α yields a surface that remain close to the input, but misses some sharp surface transitions (Figure 4d). By defining α as a function of the confidence of the normal field, our formulation produces a surface that is close to the input shape and locates well sharp transitions even in areas where the normal maps are misaligned (Figure 4e).

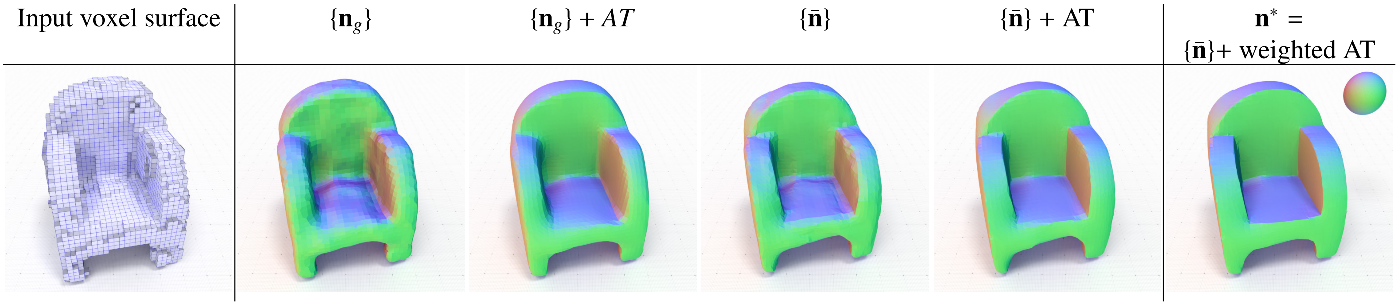
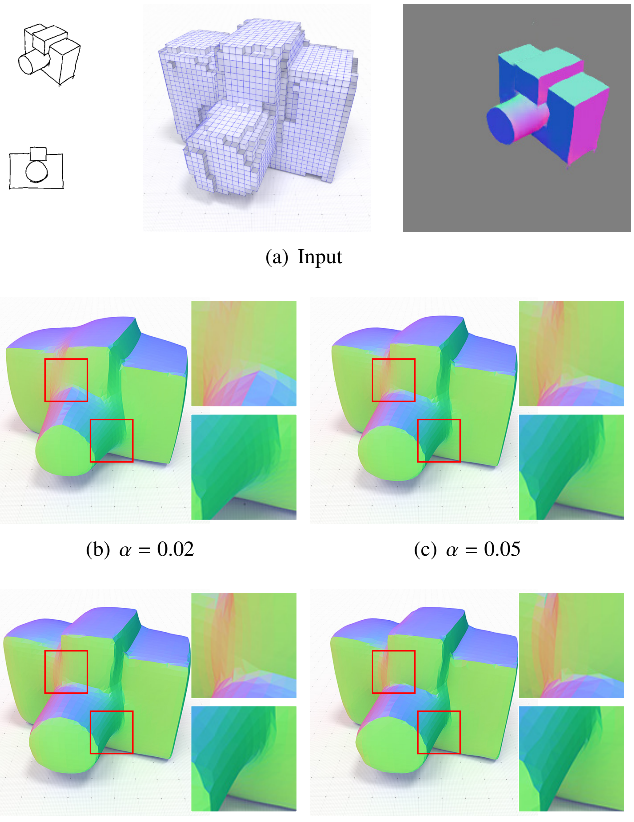


Fig. 3. Ablation study showing the surface obtained using various normal fields as guidance. The volumetric gradient $\mathbf{n}_g$ produces bumpy surfaces that lack sharp features (second column), even after being optimized according to the Ambrosio-Tortorelli energy (third column). Our aggregated normal field $\bar{\mathbf{n}}$ yields multiple surface discontinuities where the normal maps are misaligned, such as on the arms and the seat of the armchair (fourth and fifth column). We obtain the best results by reducing the influence of the aggregated normals in areas of low confidence (last column, $\mathbf{n}^*$).



(a) Input

(b) α = 0.02

(c) α = 0.05

(d) α = 0.1

(e) ours

Fig. 4. Ambrosio-Tortorelli with a fixed α deviates from the input shape (b) or misses sharp discontinuities (d). Our spatially-varying α allows the recovery of sharp features in areas where the aggregated normal field has a low confidence (e).

6.2. Performances

We implemented the deep networks in Caffe [26] and the normal field and surface optimizations in DGtal[1]. Both the prediction and optimization parts of our method take approximately the same time. The volumetric prediction takes between 150 and 350 milliseconds, depending on the number of input sketches [1]. The normal prediction takes around 15 milliseconds per sketch. In contrast, normal field optimization takes around 700 milliseconds and surface optimization takes around 30 milliseconds. Note that we measured our timings using GPU acceleration for the deep networks, while the normal field and surface optimizations were performed on the CPU.

Our approach is an order of magnitude faster than prior image-based approaches [3, 2], which need around ten seconds to perform non-rigid registration and fusion of multiple depth maps. However, our fast normal aggregation strategy is best suited to objects dominated by smooth surface patches delineated by few sharp discontinuities, while it is likely to average out information in the presence of misaligned repetitive details.

6.3. Comparisons

Figure 5 compares our surfaces with the ones obtained by Delanoy et al. [1], who apply a marching cube algorithm on the volumetric prediction. Our method produces much smoother surfaces while capturing sharper discontinuities. While our method benefits from the guidance of the predicted normal maps, it remains robust to inconsistencies between these maps and the voxel grid, as shown on the armchair (top right) where one of the normal maps suggests a non-flat back due to a missing line in the input drawing.

We also provide a comparison to feature-preserving denoising methods [28, 27] applied on the results of Delanoy et al. [1]. Without normal guidance, these methods either maintain low-frequency noise, remove important features, or introduce spurious discontinuities.

6.4. Robustness

Figure 6 evaluates the robustness of our method to noisy volumetric predictions, showing that our combination of normal map guidance and piecewise-smooth regularization yields stable results even in the presence of significant noise. We also designed our method to be robust to normal map misalignment, common in a sketching context. Figure 7 demonstrates that our method is stable in the presence of global and local misalignment. We simulate a global misalignment by shifting one of the normal maps by 5 pixels, and a local misalignment by replacing each normal by another normal, sampled in a local neighborhood.

[1] https://dgtal.org/

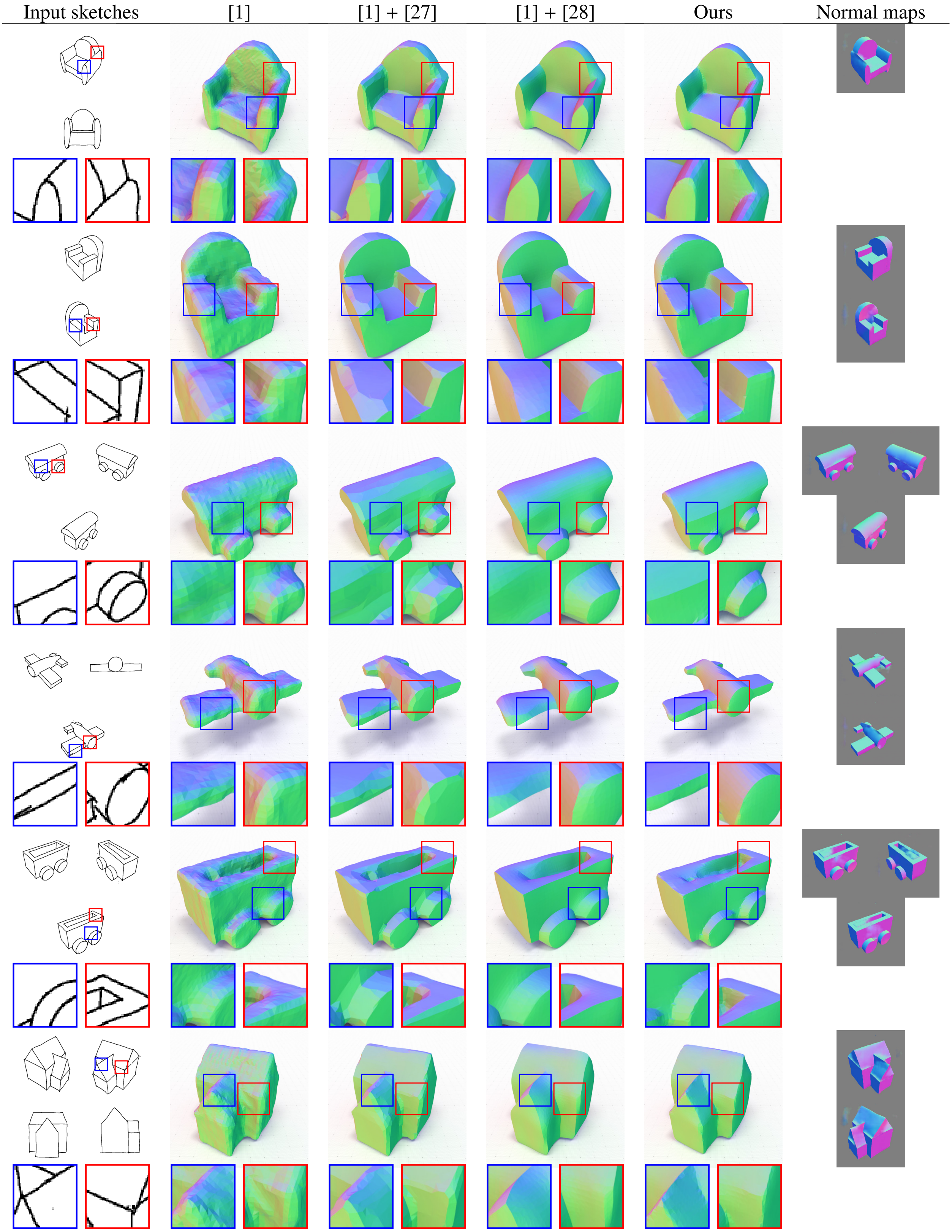


Fig. 5. Comparison to Delanoy et al. [1] on a variety of objects. Applying marching cube on the volumetric prediction results in noisy surfaces that lack sharp discontinuities (second column). Denoising these surfaces with $L0$ minimization [27] introduces spurious discontinuities as curved patches are approximated by planes (third column). Guided denoising [28] produces piecewise-smooth surfaces closer to ours (fourth column) but maintains low-frequency noise and tends to misplace discontinuities, like on the arm of the armchair (second row) or on the wings and front of the airplane (fourth row). Our formulation based on the Ambrosio-Tortorelli energy can be seen as a form of guided filtering that benefits from extra guidance from the predicted normal maps (fifth column). We included close-ups on the input drawings and output surfaces to show that our method better captures the intended shape.

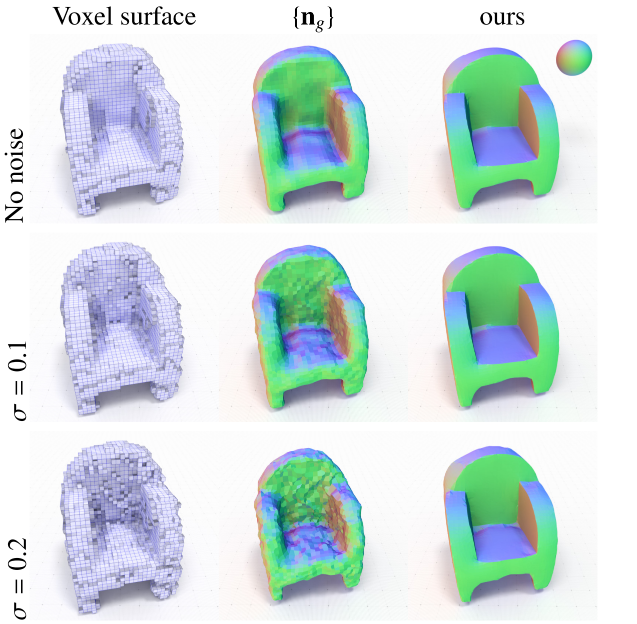


Fig. 6. Robustness to noisy volumetric prediction. Adding gaussian noise to the input volumetric prediction has little impact on the final result.

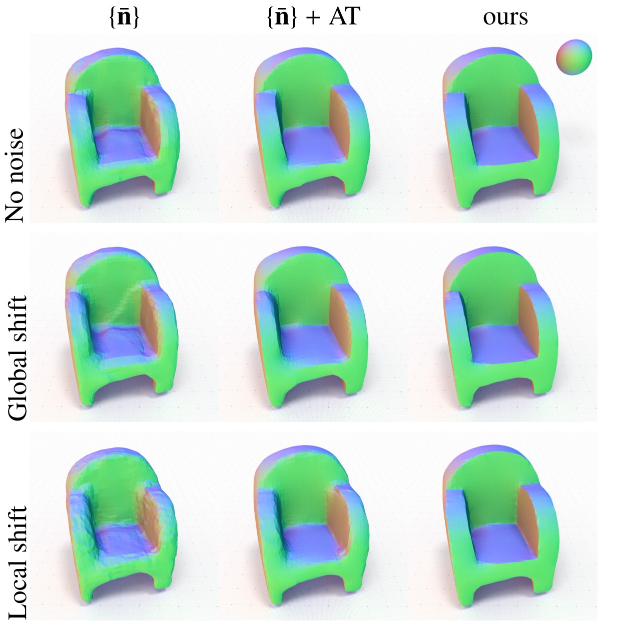


Fig. 7. Robustness to misaligned normal maps. Here we simulate global misalignment by shifting an entire normal map by the same amount (second row) or by shifting each normal by a random amount (third row). While these perturbations degrades the result of the baseline methods, our method remains stable.

6.5. *Limitations*

Figure 8 illustrates typical limitations of our approach. Since our method relies on normal maps to guide the surface reconstruction, it sometimes misses surface discontinuities between co-planar surfaces, as shown on the top of the locomotive. An additional drawing would be needed in this example to show the discontinuity from bellow. A side effect of the surface optimization energy is to induce a slight loss of volume, which is especially visible on thin structures like the wings of the airplane and the toothbrush. Possible solutions to this issue includes iterating between regularizing the surface and restoring volume by moving each vertex in its normal direction. Another limitation of our approach is that normal maps only help recovering fine details on visible surfaces, while hidden parts are solely reconstructed from the volumetric prediction, as shown on the back of the camera in Figure 9. Finally, because we favor piecewise-smooth surfaces, our approach is better suited to man-made objects than to organic shapes made of intricate details.

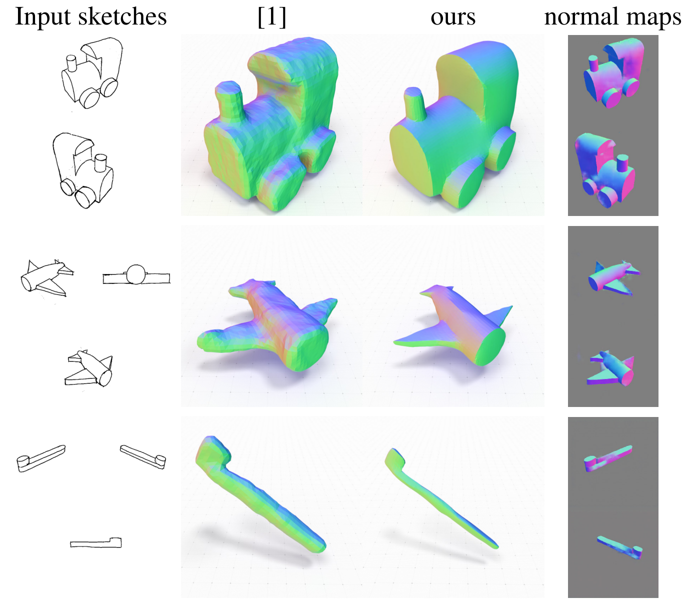


Fig. 8. Limitations of our method. Our method cannot recover surface discontinuities that are not captured by the normal maps, such as the top of the locomotive. The surface optimization energy tends to shrink the object, as seen on thin structures like the wings of the airplane and the toothbrush.

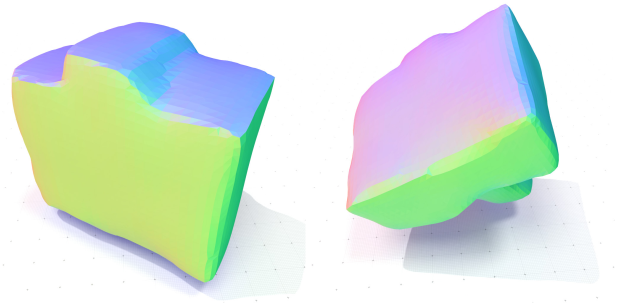


Fig. 9. Since normal maps only capture visible surfaces, the back and bottom of this camera is solely defined by the volumetric prediction. Nevertheless, the method reconstructs a smooth surface in such cases as it still benefits from the piecewise-smoothness of the Ambrosio-Tortorelli energy.

7. Conclusion

Recent work on sketch-based modeling using deep learning relied either on volumetric or image-based representations of 3D shapes. In this paper we showed how these two representations can be combined, using the volumetric representation to capture hidden parts and the image-based representation to capture sharp details. Furthermore, we showed how the volumetric representation can serve as a support for normal map fusion by solving for a piecewise-smooth normal field over the voxel surface. This method is especially well suited to man-made objects dominated by a few sharp discontinuities.

References

[1] Delanoy, J, Aubry, M, Isola, P, Efros, AA, Bousseau, A. 3d sketching using multi-view deep volumetric prediction. Proceedings of the ACM on Computer Graphics and Interactive Techniques 2018;1(1):21.

[2] Lun, Z, Gadelha, M, Kalogerakis, E, Maji, S, Wang, R. 3d shape reconstruction from sketches via multi-view convolutional networks. In: IEEE International Conference on 3D Vision (3DV). 2017, p. 67–77.

[3] Li, C, Pan, H, Liu, Y, Tong, X, Sheffer, A, Wang, W. Robust flow-guided neural prediction for sketch-based freeform surface modeling. In: ACM Transaction on Graphics (Proc. SIGGRAPH Asia). ACM; 2018, p. 238.

[4] Su, W, Du, D, Yang, X, Zhou, S, Fu, H. Interactive sketch-based normal map generation with deep neural networks. Proceedings of the ACM on Computer Graphics and Interactive Techniques 2018;1(1).

[5] Coeurjolly, D, Foare, M, Gueth, P, Lachaud, JO. Piecewise smooth reconstruction of normal vector field on digital data. In: Computer Graphics Forum; vol. 35. Wiley Online Library; 2016, p. 157–167.

[6] Coeurjolly, D, Gueth, P, Lachaud, JO. Digital surface regularization by normal vector field alignment. In: International Conference on Discrete Geometry for Computer Imagery. Springer; 2017, p. 197–209.

[7] Barrow, H, Tenenbaum, J. Interpreting line drawings as three-dimensional surfaces. Artificial Intelligence 1981;17.

[8] Xu, B, Chang, W, Sheffer, A, Bousseau, A, McCrae, J, Singh, K. True2form: 3d curve networks from 2d sketches via selective regularization. ACM Transactions on Graphics (Proc SIGGRAPH) 2014;33(4).

[9] Malik, J, Maydan, D. Recovering three-dimensional shape from a single image of curved objects. IEEE Pattern Analysis and Machine Intelligence (PAMI) 1989;11(6):555–566.

[10] Schmidt, R, Khan, A, Singh, K, Kurtenbach, G. Analytic drawing of 3d scaffolds. In: ACM Transactions on Graphics (Proc. SIGGRAPH Asia); vol. 28. ACM; 2009, p. 149.

[11] Lipson, H, Shpitalni, M. Optimization-based reconstruction of a 3d object from a single freehand line drawing. Computer-Aided Design 1996;28(8):651–663.

[12] Nishida, G, Garcia-Dorado, I, G. Aliaga, D, Benes, B, Bousseau, A. Interactive sketching of urban procedural models. ACM Transactions on Graphics (Proc SIGGRAPH) 2016;.

[13] Huang, H, Kalogerakis, E, Yumer, E, Mech, R. Shape synthesis from sketches via procedural models and convolutional networks. IEEE Transactions on Visualization and Computer Graphics (TVCG) 2016;22(10):1.

[14] Han, X, Gao, C, Yu, Y. Deepsketch2face: A deep learning based sketching system for 3d face and caricature modeling. ACM Transactions on Graphics (Proc SIGGRAPH) 2017;36(4).

[15] Choy, CB, Xu, D, Gwak, J, Chen, K, Savarese, S. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In: IEEE European Conference on Computer Vision (ECCV). 2016, p. 628–644.

[16] Ji, M, Gall, J, Zheng, H, Liu, Y, Fang, L. Surfacenet: An end-to-end 3d neural network for multiview stereopsis. In: IEEE International Conference on Computer Vision (ICCV). 2017,.

[17] Tatarchenko, M, Dosovitskiy, A, Brox, T. Multi-view 3d models from single images with a convolutional network. In: European Conference on Computer Vision (ECCV). 2016,.

[18] Tatarchenko, M, Dosovitskiy, A, Brox, T. Octree generating networks: Efficient convolutional architectures for high-resolution 3d outputs. In: IEEE International Conference on Computer Vision (ICCV). 2017, p. 2088–2096.

[19] Häne, C, Tulsiani, S, Malik, J. Hierarchical surface prediction for 3d object reconstruction. In: IEEE International Conference on 3D Vision (3DV). 2017, p. 412–420.

[20] Fan, H, Su, H, Guibas, L. A point set generation network for 3d object reconstruction from a single image. IEEE Computer Vision and Pattern Recognition (CVPR) 2017;.

[21] Wang, N, Zhang, Y, Li, Z, Fu, Y, Liu, W, Jiang, YG. Pixel2mesh: Generating 3d mesh models from single rgb images. In: European Conference on Computer Vision (ECCV). 2018,.

[22] Bronstein, MM, Bruna, J, LeCun, Y, Szlam, A, Vandergheynst, P. Geometric deep learning: Going beyond euclidean data. IEEE Signal Processing Magazine 2017;34(4):18–42.

[23] Groueix, T, Fisher, M, Kim, VG, Russell, BC, Aubry, M. Atlasnet: A papier-mâché approach to learning 3d surface generation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2018, p. 216–224.

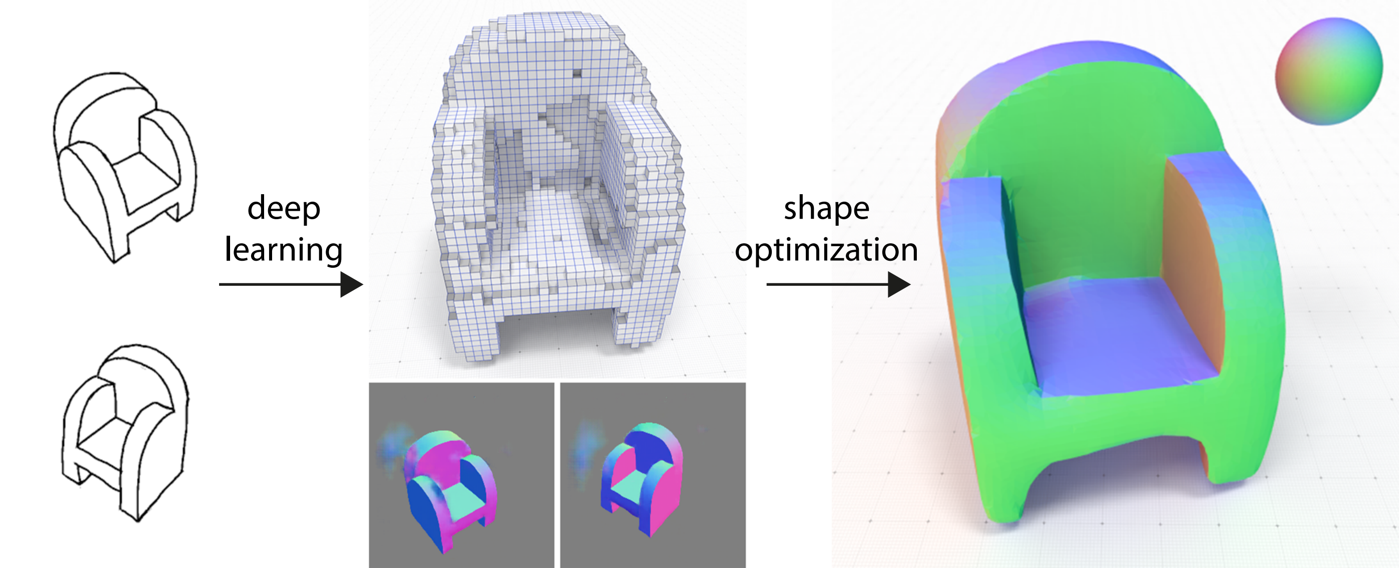
[24] Isola, P, Zhu, JY, Zhou, T, Efros, AA. Image-to-image translation with conditional adversarial networks. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2017;.

[25] Ronneberger, O, Fischer, P, Brox, T. U-net: Convolutional networks for biomedical image segmentation. In: Medical Image Computing and Computer-Assisted Intervention - MICCAI. 2015, p. 234–241.

[26] Jia, Y, Shelhamer, E, Donahue, J, Karayev, S, Long, J, Girshick, R, et al. Caffe: Convolutional architecture for fast feature embedding. arXiv preprint arXiv:14085093 2014;.

[27] He, L, Schaefer, S. Mesh denoising via l 0 minimization. ACM Transactions on Graphics (TOG) 2013;32(4):64.

[28] Zhang, W, Deng, B, Zhang, J, Bouaziz, S, Liu, L. Guided mesh normal filtering. In: Computer Graphics Forum; vol. 34. Wiley Online Library; 2015, p. 23–34.


deep learning
shape optimization