MID-CURVE COMPLETION USING CONVOLUTIONAL NEURAL NETWORK

Flavien Boussuge[1,2], Raphaël Marc[3]

[1]CEA, DAM, DIF, F-91297 Arpajon, France. flavien.boussuge@cea.fr
[2]Université Paris-Saclay, CEA, Laboratoire en Informatique Haute Performance pour le Calcul et la simulation, 91680, Bruyères le Chatel, France.
[3]EDF Lab Paris-Saclay, 91120 Palaiseau, France. raphael.marc@edf.fr

ABSTRACT

Although of great interest for Finite Element (FE) analysis, the automatic generation of a lower dimensional representation of shapes is not straightforward. One-dimensional (1D) mid-curve model contains entities such as lines, circles and more generally curves representing higher dimension entities like surfaces to be located in the middle of a thin-walled model following its shape. In this paper we present a novel approach to adapt a given medial axis of a polygon in order to generate a dimensionally reduced mid-curve model suitable for FE analysis. The novelty is to use a deep learning completion network[1] which is trained to automatically modify the perturbed regions of the medial axis (end and connection regions). The network takes as input a local image containing the polygon boundary, a mask of the regions to complete and the surrounding valid mid-curves. It returns a predicted mid-curve image which is inserted back into the global mid-curve model.

Keywords: dimensional reduction, mid-curve generation, machine learning, convolutional neural network

1. INTRODUCTION

In Finite Element Analysis (FEA), the model size influences directly the computation time. To represent thin shell regions or long and slender regions, specific Finite Elements (FE) (plate, shell and beam elements) are available in CAE software. These elements can significantly reduce the number of degrees of freedom leading to a shorter computation time while maintaining a high degree of simulation fidelity. In 3D, the initial CAD volume can be represented by its mid-surfaces where the thickness becomes a physical property associated to the finite element. Similarly, in 2D, although less sensitive to computation time, elongated shapes can be represented by equivalent mid-curves.

As explained in [2–4], although these dimensionally reduced finite elements are efficient when simulating large structural models, the process to generate them, called the dimensional reduction process or idealisation process, from an initial CAD model is not straightforward. Indeed, when a fully dimensionally-reduced model is required, the extraction of a skeleton from a 2D contour or a 3D volume using a geometrical algorithm such as the Medial Axis Transform[5,6], a reeb graph algorithm[7–9], thinning algorithms[10,11] does not produce the desired result in all the regions of the initial shape. Indeed, mid-surfaces and mid-curves must be located in the middle of a thin-walled model following its shape. For example, when using a Medial Axis Transformation (MAT) operator, local perturbations where the medial vertices have a degree superior to two (end regions, connections) appear and need to be transformed to obtain a fully dimensionally-reduced FEA model (see Figure 1 showing perturbation regions of the medial axis in the context of dimensional reduction model generation).

In practice, users often apply an automatic face-pairing operator [12], available in most CAE software, which identifies face pairs and extracts the local mid-surface. Then, the connections between the mid-surface/mid-curves are manually generated by extending/trimming the mid-surface/mid-curves. This task is not only time consuming but relies on the geometrical interpretations of the user on how to connect best the mid-surface/mid-curves (see Figure 1 showing two possible interpretations of a connection region). Although guidelines are often proposed in industry to ensure that the applied connection type is consistent across the FE models; they do not cover all the various

geometric configurations. Complex geometric configurations leave room for user interpretation with regard to the connection type to apply.

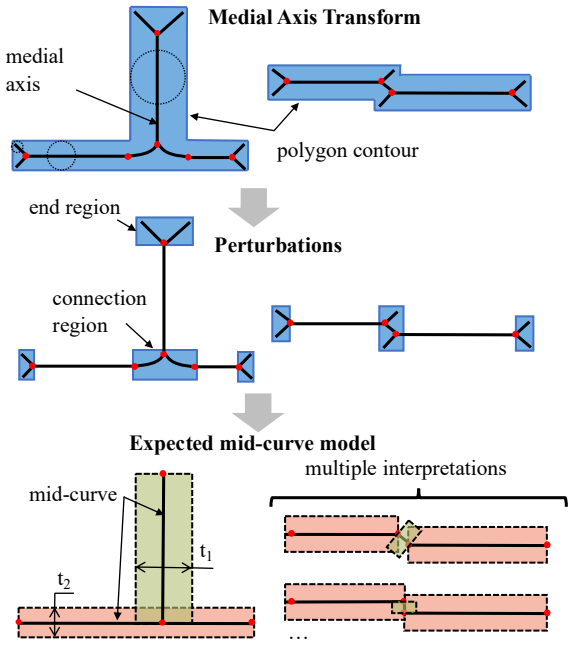

Figure 1 Dimensional reduction process of 2D contours for FE analysis using a medial axis transformation.

Although 3D mid-surface generation remains the main challenge for industrial application, the motivation of this work is to address first the 2D problem of mid-curves connection as an initial research work. To this end, we propose in this paper a new approach for mid-curve extraction based on training an image completion neural network to automatically fill masked regions corresponding to the connections and end regions of an initial medial axis of a thin contour). Here, our proposition is to replace the user decisions by neural network decisions locally for the connection regions. A medial axis is produced and kept in the thin regions (corresponding to the user expectations) but replaced in local connections and end regions by the result of a pixel-based neural network trained on images of connected mid-curves. The novelty of this approach is to combine the results of a robust geometric algorithm (the MAT), producing correct mid-curves in the thin regions of the initial contour, with the capability of neural networks to learn from a dataset of desired connection models and to artificially produce coherent completion of masked regions. In addition, by applying locally the neural network, we overcome the limitation of the pixel resolution and produce realistic local discrete connections that the mid-curves can then follow to produce a final dimensionally-reduced model for finite element analysis.

The paper is structured as follows. Section 2 reviews prior contributions. Section 3 gives an overview of the proposed approach. Section 4 details the neural network architecture, the dataset generation and the training strategy. Then, Section 5 shows the results on test contours and discusses the limitations of the approach.

2. PRIOR WORK

2.1 Dimensional reduction operators

The Medial Axis Transform: Geometrical methods have been proposed for the dimensional reduction of 2D or 3D shapes. Among them, Armstrong[2,6] uses the Medial Axis Transform (MAT). The MAT is defined by the centroid's location of the maximal inscribed circles in a 2D contour (see Figure 1) or, in 3D, by the maximal spheres inscribed in a solid. The MAT provides a skeleton of lower dimensionality; however, local perturbations (end and, connections regions) need to be further identified and transformed to obtain a fully dimensionally-reduced model for FEA. Armstrong[5] proposed to use an aspect ratio (ratio of the minimum length between the medial edge and its boundary edges to the inscribed maximum disk along this medial edge) and a taper criterion (maximum rate of diameter change with respect to medial edge length) to automatically identify the regions where medial edges should be extended or suppressed. In [13], Donaghy applies coupling constraints between mix-dimensional models keeping full dimension finite elements in thick or connection regions. Mix-dimensional modelling allows to increase the accuracy of the analysis in regions which cannot be accounted for with beam theory. Although, one can argue the dimensional reduction of connection region in regard to the accuracy of the FEA model, in the paper, our objective is to produce a fully dimensionally-reduced to increase the computational efficiency of the analysis. Geometric rules can automatically extrapolate the medial axis in end regions, however, the generalisation to all connections configurations is not trivial. Indeed, as shown in Figure 1, for a given connection, multiple solutions are possible. Often the analyst connects manually the mid-curves based on its own experience and its understanding of given modelling practices.

In 3D, the application of the MAT for dimensional reduction is still a research topic. Currently, the most efficient algorithms are derived from the generation of Voronoi diagrams [14] (see QuickHull [15] or CGAL [16] for a publicly implementation of Voronoi diagrams). The software CADFix[17] proposes a functionality to generate a 3D MAT made of B-splines surfaces from a CAD input model. Similarly, in 3D, the difficulty remains in processing further the end and connection regions of the MAT to produce the desired mid-surface. In 3D, the MAT generates complex configurations in connection areas. Manually extending, cutting and sewing the medial surfaces remains a tedious task on complex shapes.

Face pairing: Typically, commercially available CAE software use a face pairing approach (edge pairing in 2D) to generate mid-curves/mid-surfaces. The face/edge pairing process identifies opposite pairs of CAD edges/faces given

a user distance threshold[12]. For each edge/face pair, a mid-curve/mid-surface is produced as an interpolation of surfaces/curves associated to the pair of edges/surfaces. Following this approach, additional geometric treatments can be performed to the face pairs to identify robustly thin and slender regions[18,19]. The drawback of the face pairing approach, for dimensional reduction, remains in the final step where the mid-surfaces need to be connected together in connection areas. Recent efforts have been made by Kulkarni[3] to formalize the connections configurations based on a cellular modelling approach. However, current algorithms available in CAE software still encounter difficulty to compute automatically a well-connected mid-surface model for complex CAD geometries. Indeed, covering all configurations for complex model under a unified logic is not trivial for an algorithm as well as producing a robust geometric support helping locally the mid-surface connections (such as a cellular model used in [3] or the solid decomposition used in[20]).

As explained in the previous subsection, the main difficulty of mid-curve/mid-surface extraction remains in the end and connection regions. It remains difficult to design a geometric algorithm for extracting mid-curves where user decisions are required. The connections regions are dependent on the user best practices and not necessarily following clear and consistent geometric rules. As shown in the introduction section, the expected solution is difficult to define and is dependent on modeling practices, which can change from one company to another. Hence this specific connection task, which involves learning from the experience of high-skilled engineers, is a natural candidate for machine learning which can learn from previously generated models and reproduce the connection behaviour in unseen models.

2.2 Machine learning for FEA pre-processing

Machine learning has shown impressive results in shape classification, image completion/inpainting and segmentation, etc. However, few research works have applied these techniques to the pre-processing of simulation models. To identify features on CAD models, Danglade[21] analyses the use of neural networks, decision trees and support vector machines. Similarly, the featureNet of Zhang[22] uses a neural network on a voxelised representation of CAD model to identify machining features. Owen[23] recently proposed a vector-machine based approach to defeature CAD models for tetrahedral meshing. It uses mesh quality metrics as learning objectives calculated on automatically generated tetrahedral meshes from a small set of CAD models. This early-stage research has shown promising results on a limited set of CAD models. The remaining challenges to overcome for training algorithms properly remain:

- **Labelling the simulation models** depending on the simulation objectives. The meshes and CAD models used for training should be labelled according to a specific simulation objective. For example, the mesh quality metrics and size depend on the accuracy needed for a specific simulation. It is also dependent on the FE solver quality requirements. In addition, some small CAD features can be considered as details for a given simulation but may be important to keep for another simulation. In our context, the mid-curve model is an abstraction of a given 2D shape and some small details on the boundary of the 2D contours might not be represented on the dimensionally reduced model. Hence, in a supervised training scenario, it is important that the models are coherent in their representation of geometric details. This level of details should correspond to a given modelling rules defined by the user.
- **Defining the appropriate input structure** to train the network. Most neural networks use convolutional layers requiring a regular input structure to be trained on. The geometrical shape should initially be transformed into a regular structure such as a 2D grid (for example, a pixel image of the contour, or a quadtree structure). Similarly, in 3D, the CAD object should be transformed into a voxel, a points cloud or an octree[24]. These discrete representations are sensitive to the resolution used to compute them. Especially in 3D, to capture small geometric details, a high grid resolution is necessary, impacting the training time and memory. Hence, it is essential to determine the most appropriate structure depending on the application of the neural network, to identify the family of shapes to be trained on and to increase the quality of the training data using techniques such as instance filtering[25].
- **Generating a large dataset of realistic CAD / simulation models** to increase classification / segmentation performance. One of the main pillars of deep learning success is the access to a large amount of labelled data to learn from. Having a sufficiently large set of data is crucial to train efficiently the neural networks. Large datasets of 3D CAD models are already available such as ABC[26] (3D CAD) or SketchGraphs[27] (2D sketches with constraints). However, this dataset cannot directly be used for learning the extraction of mid-curves/mid-surface as they contain only the external CAD contours/surfaces of the objects. It is also difficult to obtain manually generated mid-curves/mid-surface models from industry: most of the models are proprietary and it is unlikely to represent a large amount of different shapes.

Regarding the state of the art, the closest work related to this paper is from Kulkarni[28]. Kulkarni proposed MidcurveNN, an encoder-decoder neural network to extract mid-curves from polygonal 2D shapes. The principle is to train the network with both a pixel image of the polygonal shape and of the final desired mid-curves. Although in an early stage of research, the network is able to produce reasonably well the mid-curves of simple L-shaped polygons. The limitations of this work remain in the noisiness of the produced results. It has not been tested on a large diversity of shapes and is performed on the full shape globally potentially requiring a high-resolution pixel grid for large models.

3. GLOBAL APPROACH

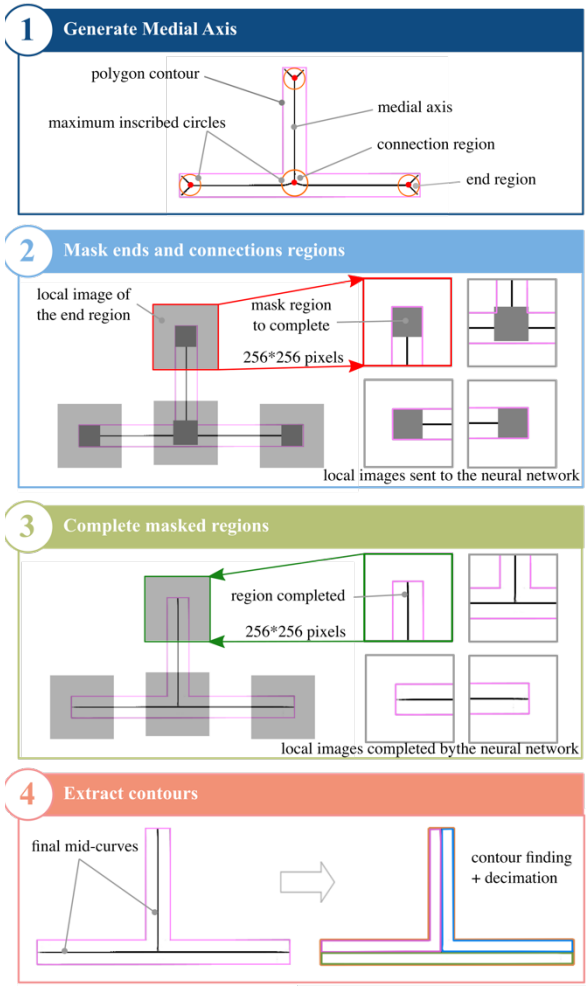

Figure 2. Process of the proposed approach

Figure 2 gives an overview of the proposed approach. The main steps of the approach are:

1. **Generate a medial axis**. The first stage generates an initial medial axis from an input contour. The MAT is a good basis to produce a 2D skeleton structure and provides geometric proximity between non-adjacent edges. In this paper we generate a discrete medial axis using a Voronoi diagram of the input polygon generated by the qhull [15] library. The end and connection regions are located at vertices v_i having a valence greater than two. A high-resolution global image (2048×2048-pixel) of the initial shape is generated bounding the input contour.

2. **Mask end and connection regions**. For each connection or end region, a local 256×256-pixel image is extracted from the global image. This image is centred at the medial axis vertex v_i. A mask is generated to hide the connection or end region. The rest of the image contains the valid surrounding mid-curves to connect. The objective is to hide only the region of the medial axis which does not correspond to a valid mid-curve model. For end regions, the mask's width and height are the maximum between the inscribed circle diameter at the medial vertex and the length of the leaf medial curves. For connection regions, the mask's width and height are equal to the inscribed circle diameter. As a first approach the mask is a rectangle bounding the inscribed circle.

3. **Complete masked region**. Each pair of image and mask is sent to the completion neural network. The neural network completes the masked part of the image with alternative content learned during its training. The content learned represents pixelized mid-curves connection models in our case. The completed local image is then put back to the initial global image to create a pixelized mid-curves model.

4. **Extract mid-curves**. The last phase to generate the geometry of the mid-curves from the pixelized representation. Here different strategies are available. One can extend the initial medial axis trying to follow the pixelized completed regions. In this paper, we use a contour finding algorithm based on the marching squares from scikit-image[29] finding constant valued contours in an image. The extracted contours contain series of points connected by line segments. The contours are bounding the inside regions generated by the mid-curves (corresponding to a segmentation of the initial shape). The polygon approximation algorithm from scikit-image is then used to simplify the extracted contours. The final mid-curves correspond to the intersection of the extracted contours. This strategy allows us to ensure that the artificially generated parts of the mid-curves (completed regions) are connected to themselves and with the input shape.

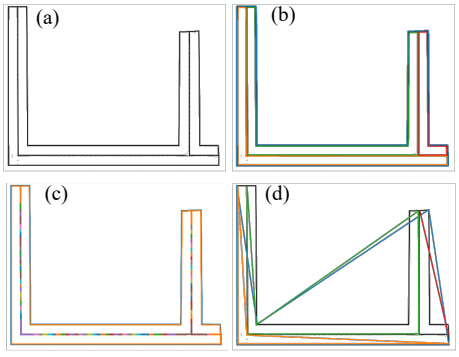

Figure 3 Application of the contour finding and polygon approximation algorithms of scikit-image[29]. (a) the input image in which to find contours, (b) the resulting contours with default gray pixel value and default polygonal approximation tolerance, (c) the resulting contours with a high gray pixel value, (d) the resulting contours with a high tolerance in the polygonal approximation.

Compared to previous approaches, our proposition lets a trained neural network take decisions on the connections of

mid-curves. Having a generic connection operator taking into account any geometry and connection model is complicated to design (see section 2). In our case, the neural network is trained on mid-curves already connected. Hence, it is up to the dataset to contain images of the expected connection in order to replicate it to unseen model. A company can specify a modelling connection rule by providing or generating a set of expected CAD shapes with their associated dimensionally-reduced models. Similarly, small details which are not represented in the final mid-curves can be learned by the neural network from the dataset of models. When a small geometric detail is consistently ignored in the final mid-curves of the training models, this removal behavior can also be replicated by the network on a new shape.

4. NETWORK ARCHITECTURE AND TRAINING STRUCTURE

4.1 Data acquisition

As explained in Section 2, no dataset is available containing 2D contours with associated mid-curve models. To test the validity of the proposed approach, we developed a simple tool to generate artificially mid-curve models. As illustrated in Figure 4, from a random polygon, we compute the straight skeleton using the CGAL library[16]. From this skeleton, we remove its external leaves and then inflate it, given a random thickness. Generating models with different thickness increases the diversity of the dataset. For practical reason, the random thickness is the same for a given skeleton to avoid overlaps artefacts in the connection of the inflated regions. The final shape represents an artificial thin contour with its associated mid-curve model from which a pixel image can be generated.

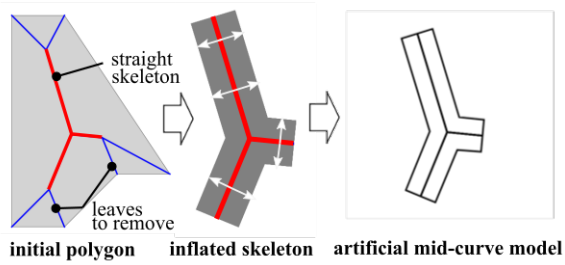

Figure 4 Process to generate an artificial mid-curve model

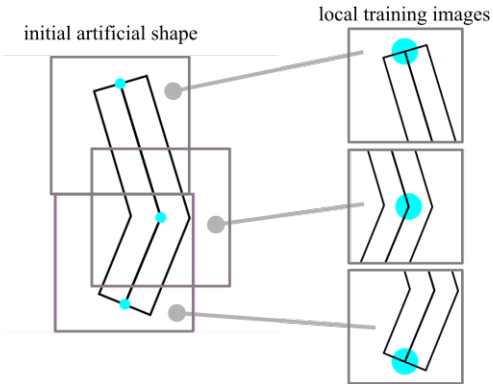

Figure 5 Extraction of training images from an artificial mid-curve model

The training images to pass to the neural network should contain connection and end regions. Hence from an artificial mid-curve model, we extract local images centred at the end and connection regions. As shown in Figure 5, at each connection vertex of the mid-curves we generate a 256×256-pixel image centred at the vertex position. For end vertices, we offset the local image in order to have a maximum of geometric information contained in the training image. We repeat this process on random polygons in order to generate a dataset of 15000 images containing end and connection regions with their expected mid-curves. The distribution of end and connection region images is 30% end regions and 70% connection regions. Figure 6 shows some examples of training images.

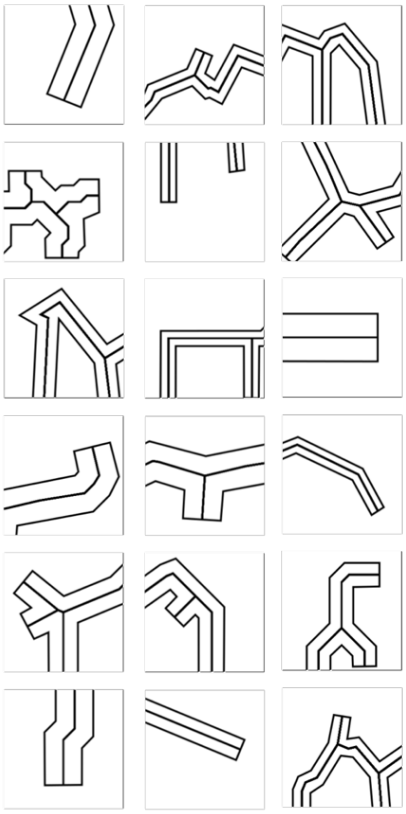

Figure 6 Examples of training images

4.2 Image completion neural network model

The neural network objective is to fill the masked end and connection regions M_c based on the surrounding context present in the extracted image (polygon boundary and valid medial axis regions). To complete the masked regions of the local extracted image of the medial axis (stage 2 and 3), we use the Globally and Locally Consistent Image Completion (GLCIC) deep neural network architecture proposed by Iizuka et al.[1]. The advantages of this learning-based inpainting approach are:

- Can generate novel image fragment which are not present in the input image. Unlike patch-based methods (such as PatchMatch of Barnes et al.[30]) which assumes that similar image patches can be found outside the missing region to fill in the missing regions in the same image , Iizuka et al.[1] approach is able to generate novel image fragment learned during the training phase. In our case, the connection and end regions are not present in the extracted image and should be learned from the dataset of midcurve images.
- Can consider both the local and global information of an image to enforce locally and globally consistency. The network architecture of Iizuka et al.[1] is able to look at the entire image for understanding its context, then filling in the missing parts based on its context. In our case, the polygon boundary as well as the valid surrounding medial axis regions can be used by the network to understand the image context.

As illustrated in Figure 7, for learning the image completion the GLCIC architecture follows a Generative Adversarial Network (GAN) [31] framework. It is composed of a completion network which is trained to complete the masked region and two discriminators networks (local discriminator and global discriminator) which are used in the training phase as auxiliary networks for learning. During the training phase, the objective of the generator is to complete the masked regions in order to fool the two discriminators while the discriminators are responsible for distinguishing completed images from real images. The advantage of combining two discriminators is that the global discriminator looks at the whole image in a global sense (ensuring the validity of the image as a whole) while the local discriminator looks at the sub-image around the filled region in a local sense (verifying the local validity of the completed region).

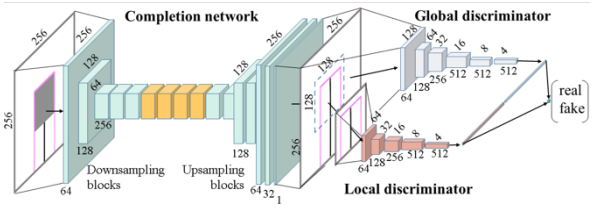

Figure 7 Architecture of the GLCIC neural network of Iizuka [1] for image completion/inpainting.

As the network architecture can deal with arbitrary image size and masks, we keep the architecture initially proposed by Iizuka. The completion network noted $C(x,M_c)$ takes as input the original image x and fills the masked regions M_c. The binary mask M_c takes the value 1 inside regions to be filled-in and 0 elsewhere. The completion network follows an encoder-decoder architecture composed of a series of convolution and deconvolution layers. Unlike Iizuka's paper[1] on completing an RGB image, we use a 256×256-pixel grayscale image (one-channel image) as input. A random hole in the 48×48-pixel to 96×96-pixel range is generated for the mask. The combined discriminators noted $D(x, M_d)$ are composed of a series of convolution layers. The input of the global discriminator is a 256×256 pixel image (corresponding to the entire global image) and for the local context discriminator the input is a 128×128 pixel patch centred around the mask to complete. When used with unseen images, the completion network returns a 256×256 pixel image with only the region corresponding to the mask modified.

As mentioned in Iizuka et al.[1], in order to train the network to complete the input image realistically, two loss functions are jointly used: a weighted Mean Squared Error (MSE) loss for training stability, and a GAN[31] loss to improve the realism of the results. The MSE loss considering the completion region mask is defined by:

$$L(x, M_c) = \|M_c \odot (C(x, M_c) - x)\|^2,$$

where $\odot$ is the pixelwise multiplication and $\| \cdot \|$ is the Euclidean norm. The MSE loss is computed within the masked region, the pixels outside the masked regions (marked as 0 in M_c) are directly replaced by the valid pixels.

The GAN loss refers to a minimax optimisation problem where in which at each iteration the discriminator networks are jointly updated with the completion network.

$$min_C max_D \, E[log(D(x, M_d) + log(1 - D(C(x, M_c), M_c)],$$

where M_d is a random mask (to randomly select an image mask for local discriminator), M_c is the input mask, and the expectation value is the average over the training images x. The goal of this GAN loss is to train the completion network to generate realistic images that the discriminators cannot distinguish between completed images and real images. By combining the two loss functions, the joint loss function to train the network optimization becomes[1]:

$$min_C max_D \, E[L(x, M_c) + \alpha \, log(D(x, M_d) + \alpha \, log(1 - D(C(x, M_c), M_c)],$$

where alpha is a weighting hyper-parameter to balance the MSE loss and the GAN loss.

For more details on the network architecture, the justification of the layers number and size, the training strategy as well as the limitations on the mask size, we let the reader refer to Iizuka's paper[1].

4.3 Training

The GLCIC model is implemented in Python using TensorFlow[32]. The model is trained on the 15,000 images collected from artificially generated mid-curve models (see Section 4.1). The dataset is split into training and test subsets and trained. The training set contains 95% of images of the input dataset. The learning rate is set to 1e-3. The completion network is initially trained for 200 iterations with the mean squared error (MSE) loss. Then both the completion network and discriminators are trained for 200 additional iterations. The entire training procedure is executed on a NVIDIA V100 GPU and takes approximatively 8 hours to reach the 400 iterations.

5. RESULTS AND DISCUSSIONS

5.1 Results

In this section, we present the application of the proposed approach to transform a medial axis model into a mid-curve model suitable for FE analysis. Figure 8 shows the application of the completion neural network on the end and connection regions of a given medial axis. Initially a global image (2048×2048 pixel) of the given contour with its associated medial axis is generated. For each end and connection region of the medial axis, a local 256×256 pixel image is extracted. A mask is placed at the intersection medial vertex. The local image and its mask are given as input to the completion network. The output completed image is then placed back to the global image. Finally, a contour extraction algorithm (see Section 3) is applied to the global image in order to extract geometric entities from the image and generate a geometrical mid-curve model. Figure 9 illustrates examples of mid-curve model generated from medial axis models, generated with the process explained in Section 4.1.

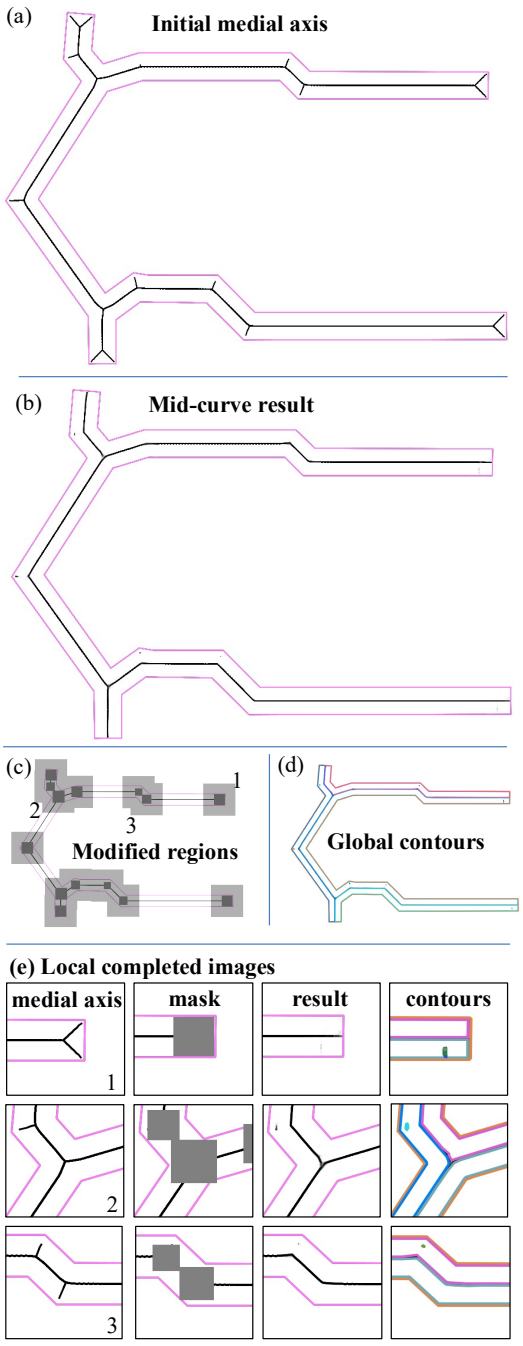

Figure 8 Result of the completion network on an input contour: (a) initial contour and its medial axis, (b) the mid-curve model result, (c) location of the extracted local images; (d) application of the contour finding algorithm to extract geometric entities, (e) example of the completed images corresponding to the 1,2,3 regions of (c).

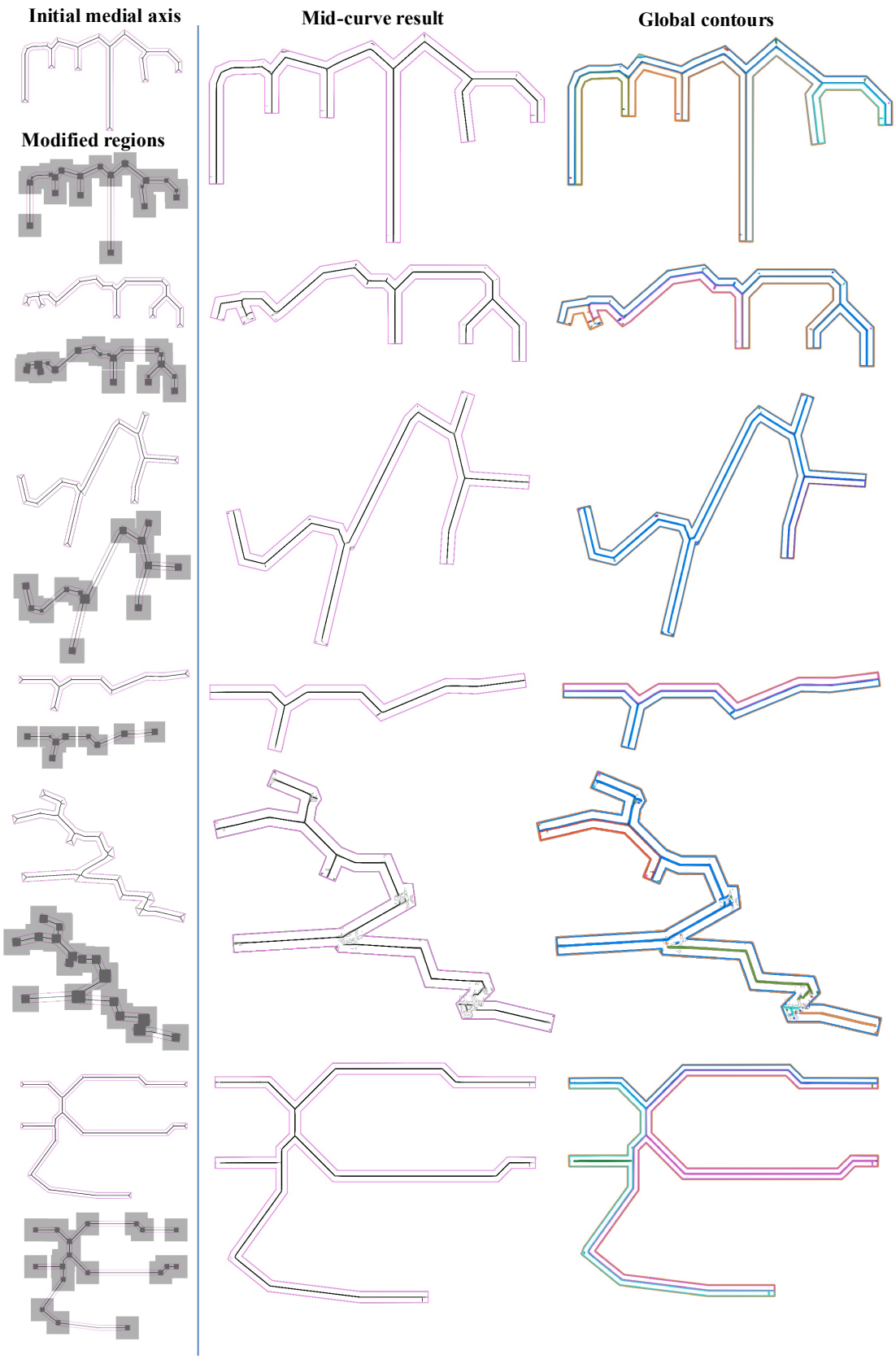

Figure 9 Examples of mid-curve models automatically generated from an initial medial axis model.

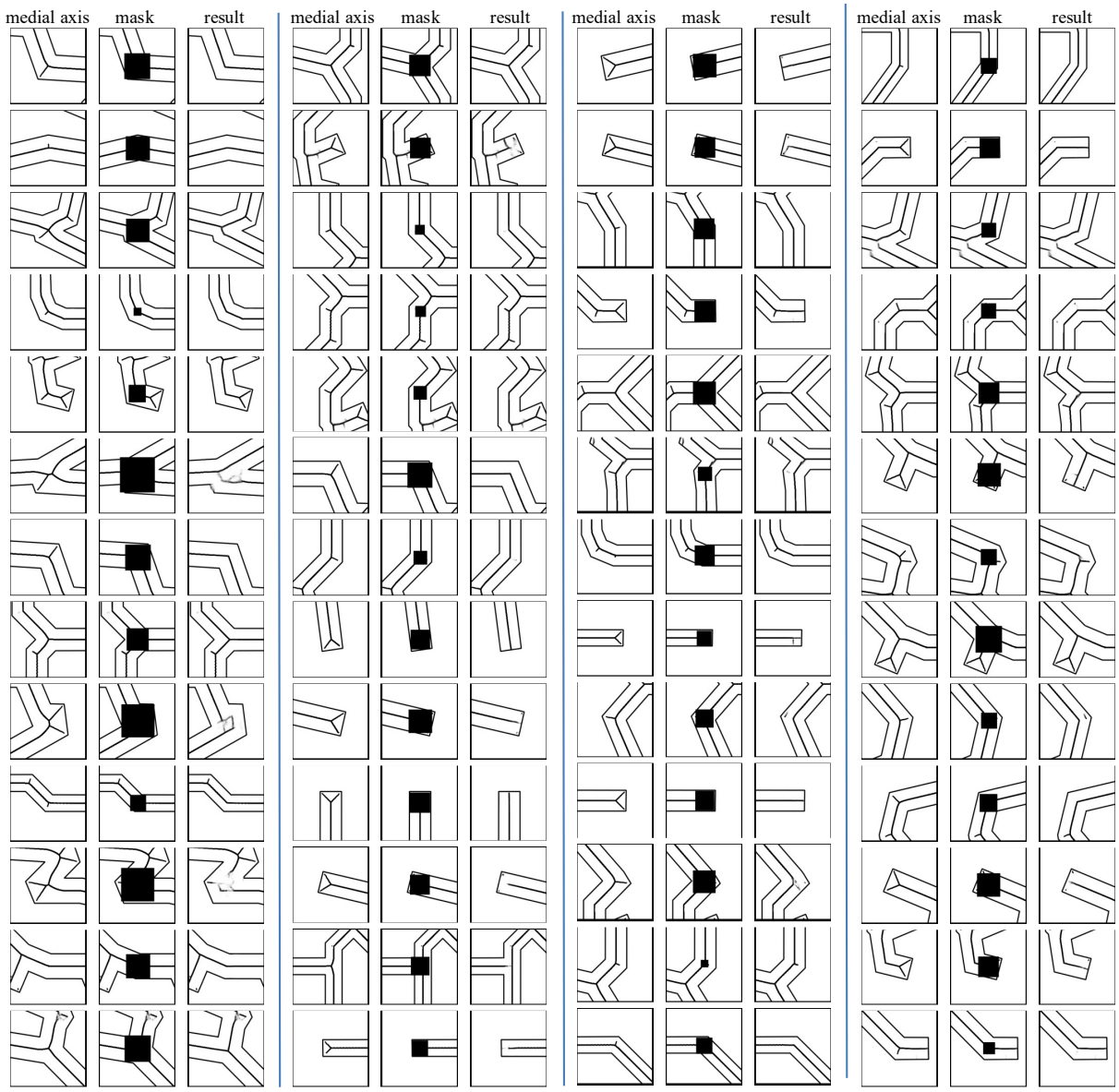

Figure 10 Results produced by the completion network. The masked region (in black in the image) corresponds to the end or connection region of the medial axis to modify.

5.2 Discussion and future work

Currently, the proposed approach faces the following limitations:

- **Mask orientation:** In the current implementation, the mask is aligned with the global XY coordinates. However, as illustrated in Figure 11 (a), when the end region is not aligned with the global axis, the mask is not covering the full region to modify, giving an incomplete result. A better location (manual) of the mask covering the region to modify produces the expected mid-curve.
- **Mask position:** For end region, the network has been trained with offset images containing a maximum of medial axis geometric information. As mentioned in the data acquisition section 4.1, for end vertices, the local extracted image initially centred at the medial vertex v_i position is offset along the medial axis in order to place the medial vertex v_i next to the image border. This allows us to reduce the empty space in the training images. However, in stage 2, the extracted local image is centred at the medial vertex v_i position and the image

offset has not been implemented. Hence, as shown in Figure 11 (b), the centred image leads to slightly worse result than if the image had been offset.

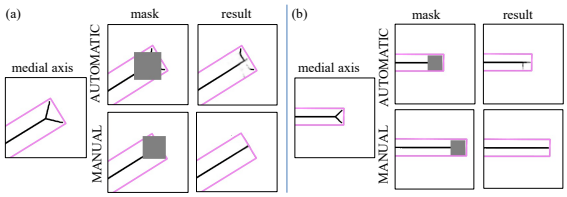

Figure 11 Limitations in local window and/or mask locations.

- **Mask size and shape:** The network has been trained with rectangle masks of size in the range 48 to 96-pixels. As shown in Figure 12, the network does not produce the expected mid-curve when the mask has a triangular oriented shape covering the region surrounding the medial vertex v_i (e.g., by building a triangle from the external vertices of the medial axis and the inscribed circle). A better approach would be to train the network with these minimal mask shapes instead of rectangle. This has been left for future work.

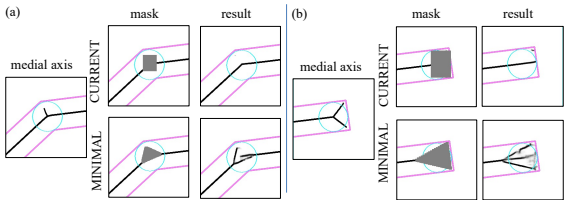

Figure 12 Constraint in using the mask shape the network has been trained for.

- **Artefacts generated by the image completion network** can appear in the result image as shown in Figure 13. These small "blurry lines" can be interpreted either as part of other polygon boundary (with a different thickness) or as alternative midcurves which the network starts to predict. In this case, the training should be improved by providing additional training images to enhance the prediction of the network. Alternatively, a simple approach to remove the isolated artefact is to filter the contours extracted by the contours finding algorithm based on the contour size.

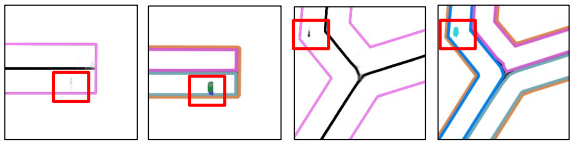

Figure 13 Artefacts generated by the image completion network

- **Multiple network predictions:** Currently, only one image is returned by the neural network. However, for some complex connection configurations, multiple connection models might be valid. Applying a network, such as the pluralist network of Zheng et al.[33], multiple completed images could be proposed from which the user could choose the most appropriate connection.
- **Long and slender geometry:** Another limitation deals with the initial shape of the input contour. In this work, we concentrated on long and slender shapes where the hypothesis of replacing these regions by dimensionally reduced model is valid. This hypothesis allows us to train the network on local 256×256 pixel images where the medial axis as well as the initial contour is present and hence saving training time. When applying this method to regions where the diameter of the inscribed circle is large will require to a larger image resolution to keep the same level of detail. The training time will be impacted and new approaches[34,35] handling ultra-resolution images should be considered.

In future work we want to consider the extension of this 2D approach to the 3D dimensional reduction process. Clearly, the real industrial challenge is to generate automatically equivalent mid-surface models of thin structural components. Dealing with 3D models implies new issues to overcome. For example, regarding the size of input structure: the extra dimension has new information to consider which might impact considerably the training time. Our training strategy should be reconsidered: for example by removing duplicated models; by considering data augmentation to handle multiple orientations of a given 3D model, etc.

6. CONCLUSIONS

In this paper, we presented a new approach to build a 2D dimensionally reduced model of a given thin polygon. The approach is based on the application of an image completion neural network. The neural network automatically completes the masked regions of a given 2D medial axis. By applying the completion network to the perturbed regions of the medial axis (i.e. end and connection regions) we are able to produce a mid-curve model where the mid-curves are located in the middle of a thin-walled model. The benefit of this approach is to let the neural network take the decision on the mid-curve connection models to apply from the given learning dataset. Future work concerns the optimisation of the mask geometry, the consideration of non-thin geometry and the extension of the approach to the 3D dimensional reduction process.

REFERENCES

[1] S. Iizuka, E. Simo-Serra, H. Ishikawa, Globally and locally consistent image completion, ACM Trans. Graph. 36 (2017) 1–14.

[2] C.G. Armstrong, Modelling requirements for finite-element analysis, Comput. Des. 26 (1994) 573–578.

[3] Y.H. Kulkarni, A. Sahasrabudhe, M. Kale, Leveraging feature generalization and decomposition to compute a well-connected

midsurface, Eng. Comput. 33 (2017) 159–170.
[4] F. Boussuge, J.-C. Léon, S. Hahmann, L. Fine, An analysis of DMU transformation requirements for structural assembly simulations, in: Int. Conf. ECT2012, Proc. Eighth Int. Conf. Eng. Comput. Technol. Dubrovnik, Croat. 4-7 Sept. 2012, 2012.
[5] C.G. Armstrong, D.J. Robinson, R.M. McKeag, T.S. Li, S.J. Bridgett, R.J. Donaghy, C.A. McGleenan, Medials for meshing and more, in: 4th Int. Meshing Roundtable, Citeseer, 1995: pp. 277–288.
[6] C.G. Armstrong, D.J. Robinson, R.M. McKeag, T.S. Li, S.J. Bridgett, R.J. Donaghy, Applications of the medial axis transform in analysis modelling, in: NAFEMS, Proc. 5th Int. Conf. Reliab. FEM Eng. Appl., 1995: pp. 415–426.
[7] G. Reeb, Sur les points singuliers d'une forme de pfaff completement integrable ou d'une fonction numerique [on the singular points of a completely integrable pfaff form or of a numerical function], Comptes Rendus Acad. Sci. Paris. 222 (1946) 847–849.
[8] D. Bespalov, W.C. Regli, A. Shokoufandeh, Reeb graph based shape retrieval for CAD, in: ASME 2003 Int. Des. Eng. Tech. Conf. Comput. Inf. Eng. Conf., American Society of Mechanical Engineers, 2003: pp. 229–238.
[9] M. Hilaga, Y. Shinagawa, T. Kohmura, T.L. Kunii, Topology matching for fully automatic similarity estimation of 3D shapes, in: Proc. 28th Annu. Conf. Comput. Graph. Interact. Tech., 2001: pp. 203–212.
[10] T.Y. Zhang, C.Y. Suen, A fast parallel algorithm for thinning digital patterns, Commun. ACM. 27 (1984) 236–239.
[11] L. Lam, S.-W. Lee, C.Y. Suen, Thinning methodologies-a comprehensive survey, IEEE Trans. Pattern Anal. Mach. Intell. 14 (1992) 869–885.
[12] M. Rezayat, Midsurface abstraction from 3D solid models: general theory and applications, Comput. Des. 28 (1996) 905–915.
[13] R.J. Donaghy, C.G. Armstrong, M.A. Price, Dimensional reduction of surface models for analysis, Eng. Comput. 16 (2000) 24–35.
[14] D.-M. Yan, W. Wang, B. Lévy, Y. Liu, Efficient computation of clipped Voronoi diagram for mesh generation, Comput. Des. 45 (2013) 843–852.
[15] C.B. Barber, D.P. Dobkin, H. Huhdanpaa, The quickhull algorithm for convex hulls, ACM Trans. Math. Softw. 22 (1996) 469–483.
[16] A. Fabri, S. Pion, CGAL: The computational geometry algorithms library, in: Proc. 17th ACM SIGSPATIAL Int. Conf. Adv. Geogr. Inf. Syst., 2009: pp. 538–539.
[17] CAD Exchange, Geometry repair and Model simplification with CADfix, (n.d.).
[18] M. Ramanathan, B. Gurumoorthy, Generating the Mid-Surface of a Solid using 2D MAT of its Faces, Comput. Aided. Des. Appl. 1 (2004) 665–674.
[19] L. Sun, C.M. Tierney, C.G. Armstrong, T.T. Robinson, Decomposing complex thin-walled CAD models for hexahedral-dominant meshing, Comput. Des. (2017). doi:https://doi.org/10.1016/j.cad.2017.11.004.
[20] F. Boussuge, J.-C. Léon, S. Hahmann, L. Fine, Idealized models for FEA derived from generative modeling processes based on extrusion primitives, Eng. Comput. 31 (2015) 513–527.
[21] F. Danglade, J.-P. Pernot, P. Véron, On the use of machine learning to defeature CAD models for simulation, Comput. Aided. Des. Appl. 11 (2014) 358–368.
[22] Z. Zhang, P. Jaiswal, R. Rai, Featurenet: Machining feature recognition based on 3d convolution neural network, Comput. Des. 101 (2018) 12–22.
[23] S.J. Owen, T. Shead, S. Martin, CAD Defeaturing using Machine Learning., Sandia National Lab.(SNL-NM), Albuquerque, NM (United States), 2019.
[24] W. Cao, T. Robinson, Y. Hua, F. Boussuge, A.R. Colligan, W. Pan, Graph Representation of 3D CAD Models for Machining Feature Recognition With Deep Learning, in: ASME 2020 Int. Des. Eng. Tech. Conf. Comput. Inf. Eng. Conf., American Society of Mechanical Engineers Digital Collection, 2020.
[25] M.R. Smith, T. Martinez, Improving classification accuracy by identifying and removing instances that should be misclassified, in: 2011 Int. Jt. Conf. Neural Networks, 2011: pp. 2690–2697.
[26] S. Koch, A. Matveev, Z. Jiang, F. Williams, A. Artemov, E. Burnaev, M. Alexa, D. Zorin, D. Panozzo, ABC: A big CAD model dataset for geometric deep learning, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2019: pp. 9601–9611.
[27] A. Seff, Y. Ovadia, W. Zhou, R.P. Adams, SketchGraphs: A Large-Scale Dataset for Modeling Relational Geometry in Computer-Aided Design, ArXiv Prepr. ArXiv2007.08506. (2020).
[28] Y.H. Kulkarni, MIDCURVENN: Encoder-decoder neural network for computing midcurve of a thin polygon, in: Open Data Sci. Conf., 2019.
[29] S. van der Walt, J.L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J.D. Warner, N. Yager, E. Gouillart, T. Yu, the scikit-image contributors, scikit-image: image processing in {P}ython, PeerJ. 2 (2014) e453. doi:10.7717/peerj.453.
[30] C. Barnes, E. Shechtman, A. Finkelstein, D.B. Goldman, PatchMatch: A randomized correspondence algorithm for structural image editing, ACM Trans. Graph. 28 (2009) 24.
[31] I.J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial networks, ArXiv Prepr. ArXiv1406.2661. (2014).
[32] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G.S. Corrado, A. Davis, J. Dean,

M. Devin, TensorFlow: Large-scale machine learning on heterogeneous systems, (2015).
[33] C. Zheng, T.-J. Cham, J. Cai, Pluralistic image completion, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2019: pp. 1438–1447.
[34] S. Wu, M. Zhang, G. Chen, K. Chen, A new approach to compute cnns for extremely large images, in: Proc. 2017 ACM Conf. Inf. Knowl. Manag., 2017: pp. 39–48.
[35] L. Hou, Y. Cheng, N. Shazeer, N. Parmar, Y. Li, P. Korfiatis, T.M. Drucker, D.J. Blezek, X. Song, High resolution medical image analysis with spatial partitioning, ArXiv Prepr. ArXiv1909.03108. (2019).