www.cad-journal.net
Computer-Aided Design
AND APPLICATIONS™

Automatic 3D Modeling Process for Predefined Geometrical Categories Based on Convolutional Neural Network and Computer-Vision Analysis of Orthographic Images

Laura Loredana Micoli[1], Gabriele Guidi[2] and Giandomenico Caruso[3]

[1] Politecnico di Milano, Italy, laura.micoli@polimi.it
[2] Indiana University, Bloomington, USA, gabguidi@iu.edu
[3] Politecnico di Milano, Italy, giandomenico.caruso@polimi.it

Corresponding author: Giandomenico Caruso, giandomenico.caruso@polimi.it

Abstract. Implementing an alternative reality for Metaverse implies modeling optimized digital content to guarantee real-time interaction and high-quality rendering. Even if 3D reconstruction based on 3D scanning techniques provides a good replica of real objects, the output files are challenging to use in this application. On the other hand, manually developed optimized 3D models require much time and effort. This aspect becomes crucial in scenarios including thousands of 3D models with which humans should interact. This paper proposes a method to automate the 3D modeling process of items whose shapes can be classified according to predefined geometrical categories. The dataset for this study relates to products, which present a wide variety of shapes but are attributable to just a few formal archetypes. In the proposed pipeline, metric orthographic images of the object to be digitally reproduced are analyzed by Convolutional Neural Networks (CNN)s. Subsequently, the same images are analyzed with Computer-Vision (CV) algorithms to extrapolate the characteristic dimensions related to the assigned archetypes. The method has been tested on different items, and the results proved the effectiveness of the whole approach in terms of correct archetypes recognition, parameter extraction, and creation of the 3D model, which are comparable with digitized 3D models with high-quality scanning tools but much lighter in model size.

Keywords: Artificial Intelligence, Convolutional Neural Network, Computer Vision, Automatic 3D Modeling, Process Optimization, 3D Object Database
DOI: https://doi.org/10.14733/cadaps.2024.677-692

1 INTRODUCTION

According to the Metaverse paradigms, humans should interact with digital content replicating real objects in a world that provides the illusion of an alternative reality [8]. The need for real-time response and high-quality rendering environments implies strict constraints on the 3D models that can be used. This aspect becomes crucial in scenarios, including thousands of 3D models, such as a replica of large commercial areas (e.g., supermarkets) involving a massive set of different products. In this context, having a high-quality digital replica of consumer goods is fundamental for marketing and management design. The 3D models replicating consumer goods must have the same level of detail and fidelity as their physical counterparts. Such visually sophisticated models involve a certain complexity and must be frequently updated according to the marketing needs. These needs make reducing the development time for a single virtual replica even more critical [7].

3D scanning techniques allow for high-resolution sampling of objects, but the resulting high-density meshes often limit their application in real-time rendering environments. In addition, they could include some imperfections due to possible acquisition errors generated by the poor optical response of high-reflective, monochromatic, and transparent surfaces that might be present in serially produced items. Even if optimizing techniques allow for reducing the mesh complexity by preserving the initial quality of the 3D model, manually developed meshes are still more efficient and preferable, imposing possible symmetries by design, optimizing the mesh density, and rationalizing the mesh organization in the case of UV mapping. However, the manual modeling of numerous items is time-consuming and includes significant issues, especially if the 3D models must replicate real objects in photorealistic quality.

This work proposes a method to support the automatic 3D modeling process for objects whose shapes can be classified according to predefined geometrical categories (archetypes) [6]. The method uses orthographic images of an object to be evaluated by Convolutional Neural Networks (CNN)s to assign it to a specific archetypal category [20]. Based on this assignment, Computer-Vision (CV) algorithms analyze the same images to extrapolate the characteristic dimensions related to the assigned archetypes. Once the category and the parameters are defined, a virtual replica is automatically generated through a scripted procedure running on 3D modeling software.

2 RELATED WORKS

Further research showed the potential of applying Artificial Intelligence (AI) and specifically CNNs in the modeling process to obtain a procedure that can automatically reconstruct 3D models starting from 2D images. Among them, the methods that reconstruct a 3D model are divided into two categories: supervised methods that only use a single view of the object and supervised methods that use a multi-view of the object. The Single View methods use a single 2D image with relative annotations as input.

One of the first pioneering works in this group was done by Kar et al. [12], who developed a Deep Neural Network that estimates a class-specific 3D model using 2D images. Sinha et al. [22] extended their previous work [21] based on geometry images as an intermediate representation for 3D surfaces. Later, they refined this work to add a CNN that also predicted the viewpoint jointly [26]. Recently, a new technique developed that has helped in this task, differentiable rendering. Kato et al. [13] proposed a Neural 3D Mesh Renderer (N3MR) and showed the potential to reconstruct a 3D model from a single image. To differentiate the rendering process, they propose to approximate the color transition across pixels, which is typically discrete, with a smooth transition function with improved differentiability. Also, Liu et al. proposed a differentiable rendering called SoftRasterizer [14] with an application to reconstruct a 3D model. The main difference here is that they split the rendering process by modeling the contribution of each triangle to a pixel’s color as a probability function.

Most multi-view methods use an encoder-decoder architecture to map 2D images into 3D volumes. They require larger datasets than the single-view methods, but having different views for the same object facilitates the estimation of the objective measurements. Works of Kato et al. [13] and Kanazawa et al. [11] use recurrent networks to combine information from multi-view images and recreate the 3D model but are computationally expensive and permutation-variant. Pix2Vox [27] and its refined version Pix2Vox++ [28] of Xie et al. use a deep neural network to obtain a rough 3D model from each view. Then, they use a context-aware fusion module that combines the previous model to generate a single refining version. These methods show the incredible advancements made by CNN in reconstructing a 3D model and making it look realistic. However, they all have a common drawback, creating an approximation of the real object. These works can only reconstruct the target object with the correct and precise measurements. Also, only some of the models produced are suitable since they use a lot of faces or voxels.

Other works use AI to reconstruct the 3D model and to facilitate the modeling procedure, especially in procedural modeling. An example is the work of Huang et al. [10], which developed a Convolutional Neural Network that inputs a sketch of an object and generates a 3D model. In this case, CNN is used to extrapolate discrete and continuous features as parameters for procedurally modeling the 3D surface. Nash et al. instead created ShapeVAE [18], a variational auto-encoder that learns a generative model of 3D shape based on the input images. With this 3D model, they can synthesize new samples and complete partial objects. Yumer et al. [29] used an autoencoder to facilitate the exploration of the high-dimensional space formed by the possible combination of parameters in procedural modeling. In this way, they also create new possible models at high speed.

In all these works, AI is used to help the modeling process in different ways. However, they generate new 3D models not associated with real objects. Therefore, a reliable method that can reconstruct a 3D model of a real object while maintaining its measures still needs to be solved. This research aims at overcoming these limitations by implementing a multistep automatic 3D modeling process, including different CNNs and CV algorithms, which are enabled according to specific and codified objects.

3 THE METHOD FOR AUTOMATIC 3D MODELING

The method has been developed considering a dataset related to a specific consumer goods category: “Personal Care” products (e.g., bottles, tubes, boxes, jars, etc.). This dataset apparently includes a wide variety of shapes but can be attributable to just a few archetypes. If a dataset of different 3D models varies in their formal macro-characteristics, the hypothesis is to use an automatic procedural modeling approach based on codified features and semi-automatic processes [4], [5], [16], [16], [19], [23], [24]. The method was implemented through an in-depth preliminary and iterative study to define the following steps:

- definition of an archetypes catalog;
- 3D digitization and generation of the orthographic views;
- identification of the archetype category through CNNs;
- CV analysis and extraction of the object’s profile features;
- automatic 3D modeling according to the archetype category and the metric parameters extracted by the profiles.

3.1 Definition of the Archetypes Catalog

The proposed automatic 3D modeling process is based on recognizing pre-codified objects. Consequently, a list of the expected model options is required. A morphological macro-analysis was carried out on about 1600 objects to codify the standard geometrical features of heterogeneous products with archetypes that were then collected in a catalog. The “Personal care” consumer goods category was selected because they present an excellent geometrical variety, helpful in studying the applicability of the proposed automatic 3D modeling process. Packages can be straightforward and devoid of details, such as boxes, or very complex, like the ones defined by free-form geometries, minute details, and asymmetrical features. Looking at the counter products, many items share evident morphological characteristics, even between different brands and product subcategories, diverse in size, shape, and texture but attributable to formal families that require the same 3D modeling steps [2][3], [15].

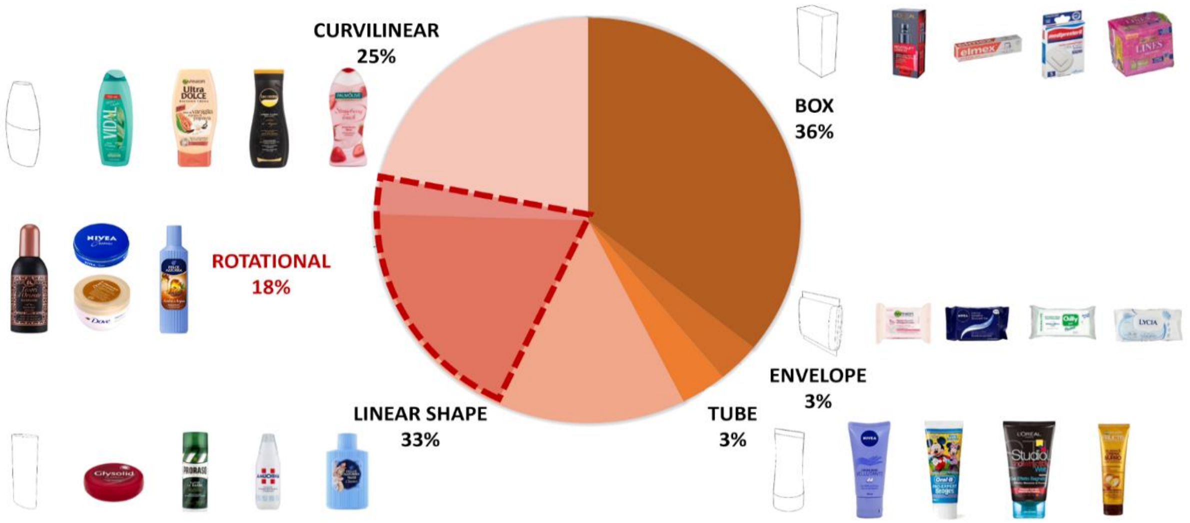


Figure 1: Percentage cataloging chart of typical shape products divided into five macro-families.

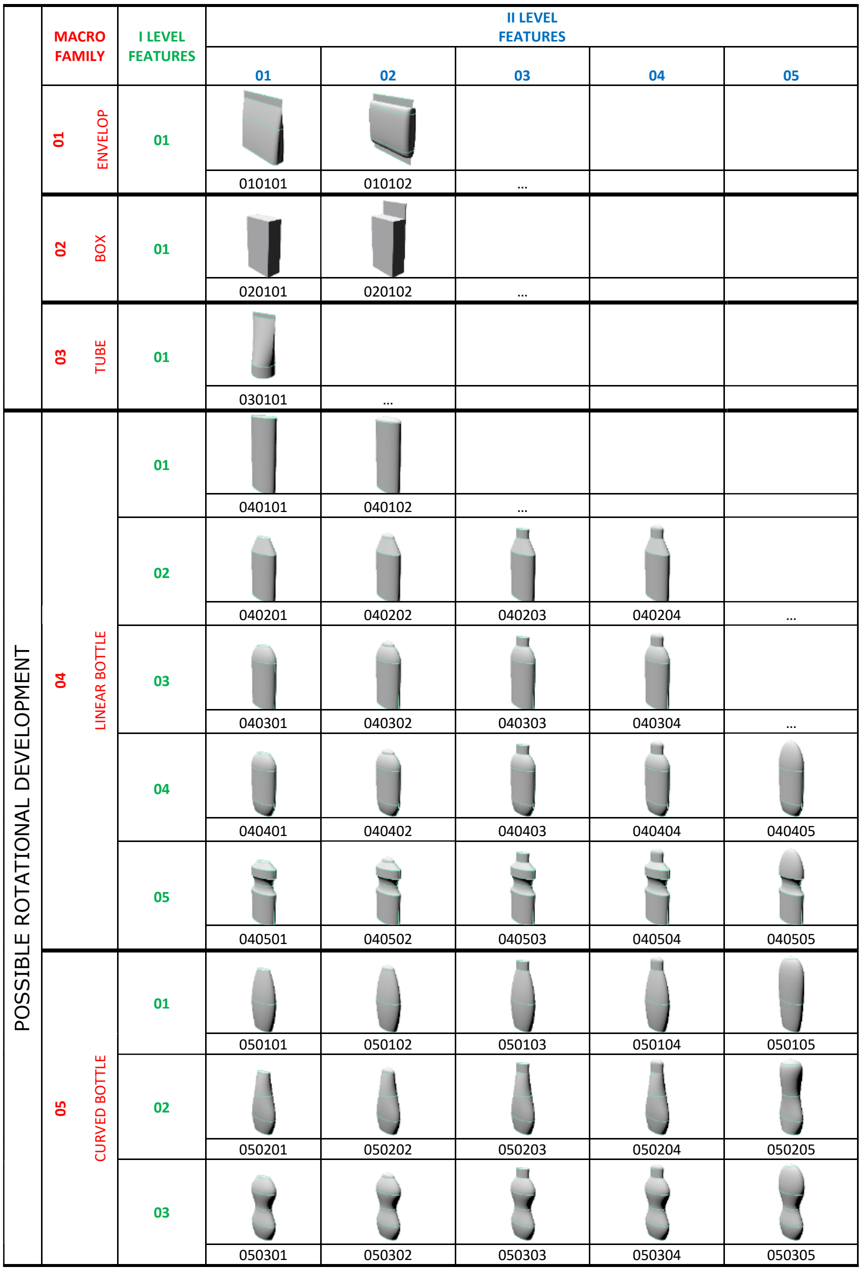
The morphological analysis on the 1600 items taken into consideration highlighted that most of them (86%) could be modeled with double symmetry surfaces according to the principal vertical planes (i.e., the two planes containing the object's vertical axis, orthogonal to the horizontal plane, and identifying the fontal and the lateral views), while 14% had none, being characterized by sculptural surfaces. According to this analysis, three main macro-families, covering 94% of the typical shape objects, have been identified: box, linear bottle, and curvilinear bottle. The remaining 6% includes shapes that can be equally distributed between soft envelopes and tubes. Furthermore, about 18% of the typical products, especially linear ones, show a rotational development around the vertical axis and can be treated. Figure 1 shows the pie chart of typical shape products divided into five macro-families.

These five macro-families have been subsequently codified within a catalog used as a reference to implement dedicated CNNs for shape recognition of new products (Figure 2). The catalog was implemented according to the geometrical features of different macro-families: i) the type of fundamental curves/points on which the modeling will be based and ii) how these geometries are connected. Moreover, each macro family was hierarchically subdivided according to the features used for modeling the different shapes of the same macro family. The discriminant for the first-level features is the number of closed profiles distributed along the vertical axis and how they are connected. In contrast, the second-level features identify the shape of the apical part of the object. It is worth noting that the catalog organization depends on the strategies chosen by the human operator to model the different objects of the dataset. Subsequently, a unique identification code has been defined for each archetype, consisting of six digits, considering three pairs identifying the salient formal features: macro family, first-level features, and second-level features.

The catalog is scalable and can be widened by including further families and archetypes. Thanks to this approach, starting from the same archetype, it is possible to generalize the digital replica of apparently very different products that share the same formal scheme. Figure 3 shows how archetype 050103 can model three different products sharing formal similarities. These involve the same modeling process based on four ellipses properly sized and used as generators of the object surfaces.

3.2 3D Digitization and Generation of the Orthographic Views

Once a catalog of shapes is defined, the following step of the generative process relies on automatically identifying a random object and assigning it to the correct category according to the criteria specified in the previous section. Such a process is crucial to let a computer-based procedure decide the proper sequence of modeling steps to generate the random object using standard surface-modeling functions.



	MACRO FAMILY	I LEVEL FEATURES	II LEVEL FEATURES 01	02	03	04	05
	01 ENVELOP	01	010101	010102	...		
	02 BOX	01	020101	020102	...		
	03 TUBE	01	030101	...			
POSSIBLE ROTATIONAL DEVELOPMENT	04 LINEAR BOTTLE	01	040101	040102	...		
		02	040201	040202	040203	040204	...
		03	040301	040302	040303	040304	...
		04	040401	040402	040403	040404	040405
		05	040501	040502	040503	040504	040505
	05 CURVED BOTTLE	01	050101	050102	050103	050104	050105
		02	050201	050202	050203	050204	050205
		03	050301	050302	050303	050304	050305

Figure 2: Graphical representation of the archetypes catalog. In the red text, the five macro-families with their two-digit code. In the green text, the two-digit code indicates the first-level features. In the blue text, the two-digit code of the second-level features. The sequence of these three codes is the six-digits identification code of a single archetype.

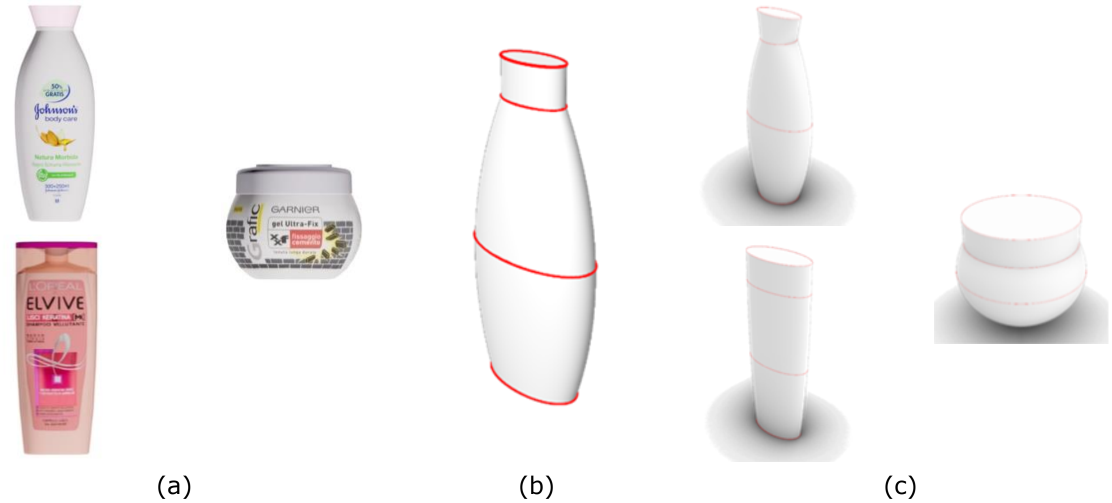


Figure 3: A example of how a catalog archetype can replicate real products. (a) pictures of three different products, (b) the reference archetype 050103, and (c) a digital replica of the products modeled by applying the main dimensions of the products to the archetype.

This 3D shape identification can be implemented automatically with an AI-based method using different approaches. The approach chosen here aims to minimize the neural network's computational complexity, and for this reason, it uses the orthographic views of the unknown object as starting point.

Orthographic views of the object can be generated by directly shooting the same object with a standard camera from conventional directions and then elaborating the image to transform a perspective view into an orthographic one. This process assumes that the camera positioning and shooting directions should observe a precise pattern that is not trivial to implement automatically.

For this reason, a 3D digitization experimental process was implemented based on a "Structure From Motion/Image Matching" (SFM/IM) process to speed up the orthographic data capture. The photogrammetric shooting set is based on multiple cameras and controlled lights. Furthermore, a set of codified targets were allocated on the shooting set, and their relative distances were preliminarily measured, allowing the 3D digitization results to be scaled to the actual size. As a result, an approximated mesh model of the object is generated and used as the first processing stage for two purposes: i) generate a metrically accurate 3D replica of the real object; ii) extract the six metrically accurate orthographic views of it (front, rear, left, right, top, bottom), needed for the following steps of the process. This methodology can be used effectively when the products are optically cooperative with the photogrammetric acquisition; for example: without transparent, excessively reflective, or completely monochromatic parts.

According to the catalog and the different 3D modeling strategies for each archetype, a plan has been implemented to elaborate and treat the orthographic images. Figure 4 shows the main steps of the process, starting from the CNN and CV analyses to the final 3D modeling.

3.3 Identification of the Archetype Category Through CNNs

The development of the catalog led to the implementation of different CNNs for the automatic recognition of a specific archetype, starting from orthographic images of the real product. The CNNs have been implemented with MobileNet [9], which allows the implementation of efficient CNNs widely used for image recognition. The results obtained with MobileNet were satisfactory even if compared with sophisticated CNNs like VGG16 [20], which also required more time for training. Our CNNs were trained with heterogeneous datasets, one for each archetype category, including real and synthetic orthographic images, i.e., generated by parametric 3D models. The training approach adopted in this research was based on the work of Su et al. [25], which demonstrates in a similar task that synthetic images can be profitably used in place of real images.

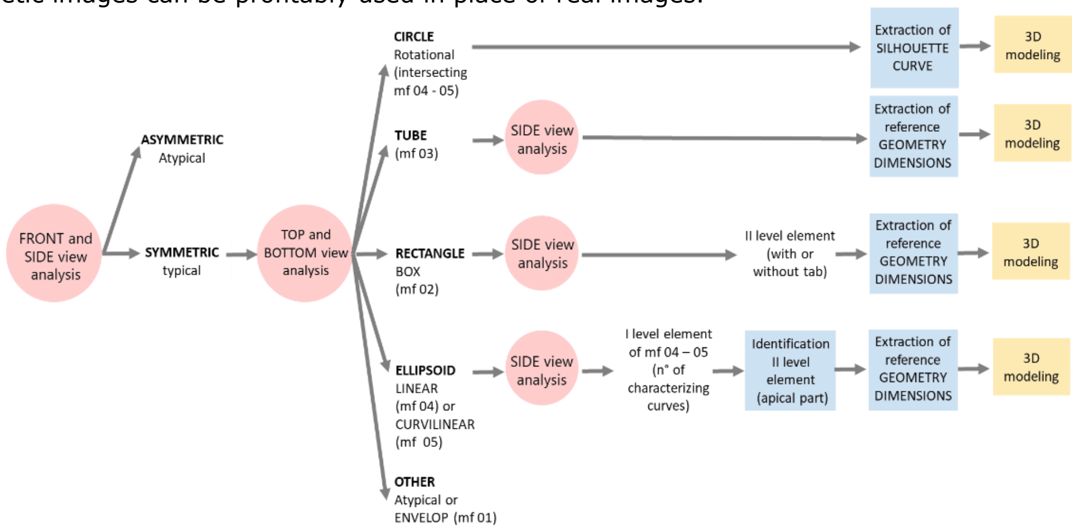
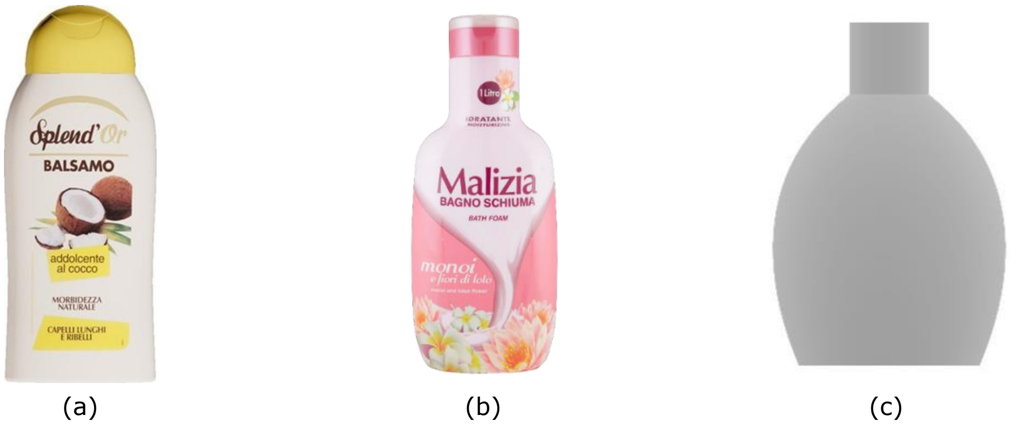

CIRCLE
Rotational (intersecting mf 04 - 05)
Extraction of SILHOUETTE CURVE
3D modeling
ASYMMETRIC
Atypical
TUBE
(mf 03)
SIDE view analysis
Extraction of reference GEOMETRY DIMENSIONS
3D modeling
FRONT and SIDE view analysis
SYMMETRIC
typical
TOP and BOTTOM view analysis
RECTANGLE
BOX
(mf 02)
SIDE view analysis
II level element (with or without tab)
Extraction of reference GEOMETRY DIMENSIONS
3D modeling
ELLIPSOID
LINEAR (mf 04) or CURVILINEAR (mf 05)
SIDE view analysis
I level element of mf 04 – 05 (n° of characterizing curves)
Identification II level element (apical part)
Extraction of reference GEOMETRY DIMENSIONS
3D modeling
OTHER
Atypical or ENVELOP (mf 01)

Figure 4: Workflow diagram of the proposed automatic 3D modeling process. The pink circles represent the analyses performed through CNNs. The blue squares represent the elaboration of CV algorithms. The yellow rectangles are the modeling operations.

Images from different databases were used for CNN training (Figure 5): i) orthographic images of real products obtained from the SFM/IM process; ii) photographic images of real products of known dimensions, not orthographic but with general characteristics suitable for the purpose; iii) images of the orthogonal views of 3D models that respect the characteristics of the single archetypes, automatically generated with random dimensions, within pre-assigned intervals, through the implemented automatic modeling process. Images relating to approximately 4300 real and hypothetical products were used.

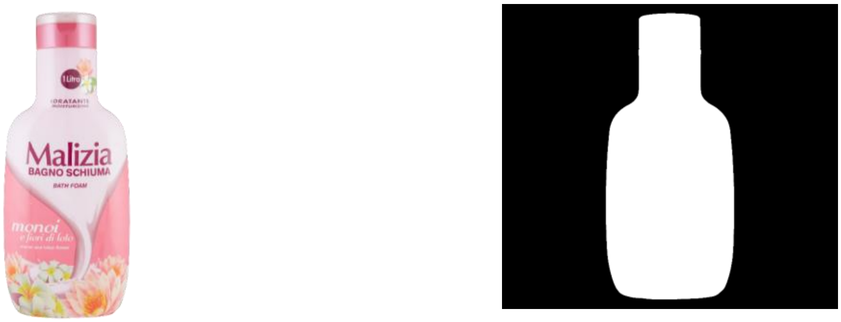


(a) (b) (c)

Figure 5: Example of frontal views used as the basis for CNN training, coming from different datasets: (a) orthophotos from SFM/IM process; (b) front image of the real product; (c) rendering of the orthographic front view of a hypothetical product of the category 050103.

To optimize the use of different types of images in 2D CNN networks, a pre-processing task was carried out to standardize images and make the whole process more efficient (Figure 6). Specifically, the following steps were taken: i) conversion from color to a black-and-white image to improve and speed up the process of analyzing the silhouette of the object (white); ii) addition of black pixels at the sides of the image to allow the random rotation of the images, which simulates real photographic images, without cutting the shapes of the objects; iii) conversion to 1:1 format, as a requirement of the MobileNet network, to avoid possible distortions; iv) resize the image to 224x224 pixels, as a requirement of the MobileNet network.

Thanks to the reduced size of MobileNet and the development of different CNNs, one for each archetype category, we could train each CNN without requiring comprehensive datasets. In all the networks used in the project, an attempt was made to maintain a constant proportion of the number of images present in each of the three sets: training at around 70%, validate 20%, and testing at 10%.



(a)

(b)

Figure 6: Image standardization: (a) original orthographic image of the product; (b) elaborated images feeding the CNNs.

The attribution of the images starting from the dataset to these three groups occurs randomly. The training of the networks provided highly positive results, as seen in the confusion matrix in Figure 7. Almost four-hundred products equally distributed among the eight categories of first-level features of the macro-families 04 and 05 have been correctly identified. Only six were attributed to the wrong category.

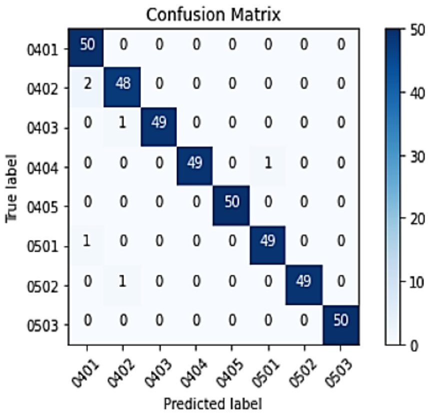


Figure 7: CNN Confusion Matrix related to classifying four-hundred products equally distributed among the eight categories of first-level features of the macro-families 04 and 05.

The main steps of the method substantially answer three questions that allow, with an ever-increasing degree of detail, to identify the correct cataloging of an object. The first CNN defines whether the object is catalogable, analyzing the object's orthographic front and side views and verifying the required double symmetry of the object. The second CNN performed for the objects defined as catalogable discriminates the main archetype category by analyzing the object's orthographic top and bottom views. In addition, the second CNN allows identifying other atypical products whose analysis of the top and bottom side does not provide results compatible with the catalog, even if both front and side views were symmetric. It is worth noticing that a deeper analysis of these atypical products could lead to new categories of products that would extend the number of items within the catalog. According to the specific category, other CNNs were implemented to analyze the side view for identifying the second-level features again.

The steps the CNNs perform are supported by standard CV algorithms and others specifically developed for implementing the proposed method. For example, the first CNN's work for verifying the object's symmetry is based on the analysis of the principal components of the image carried out with CV algorithms.

3.4 CV Analysis and Extraction of the Object's Profile Features

As discussed in section 3.3, according to the archetype, the last CNN identifies the second-level features by analyzing the lateral view of the product again. Then the front and lateral views are then elaborated with a CV algorithm to extract the pixels constituting their silhouette curves. This algorithm includes functions of OpenCV (https://opencv.org/), a cross-platform library for real-time CV applications. In particular, the OpenCV functions findContours and drawContours have been used to extract the pixels. Due to the irregular positioning of the pixels along the edges, a 1D median filter has been applied. Figure 8 shows the results of these elaborations.

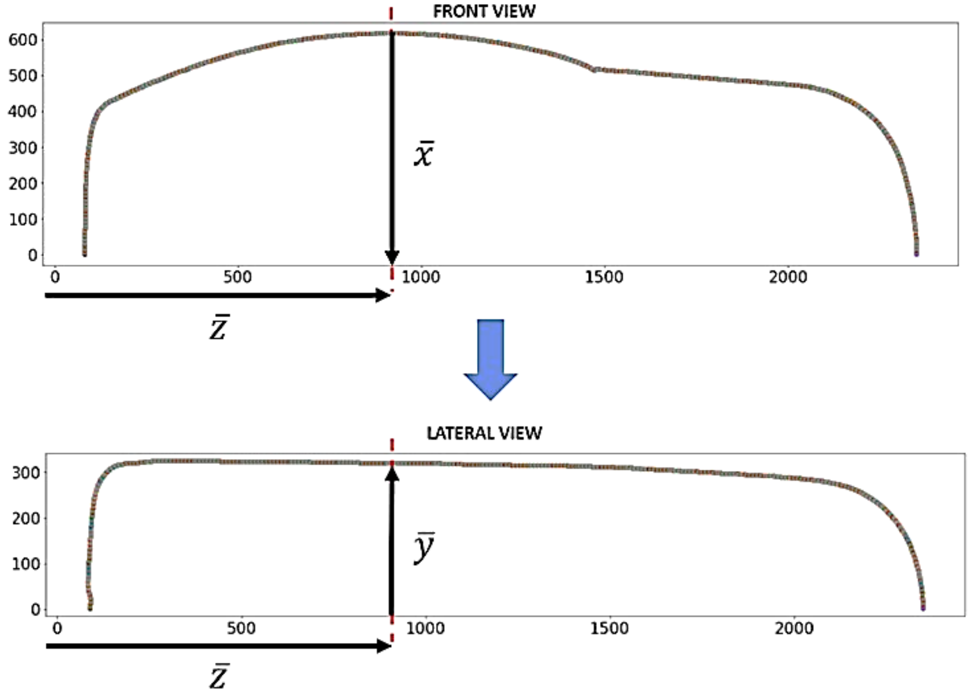


Figure 8: Derivative analysis of the silhouette curves extracted with the CV algorithm of the half-front and half-lateral views.

After these image elaborations and according to the identified category, a dedicated derivative analysis of the silhouette extracts the sizes and positions of the reference geometries used for the automatic 3D modeling. The analysis of the object's front view starts from the object's base, and it seeks slope changes. The calculation of the derivative is approximated with the local slope of the function. Since calculating the slope value between two successive pixels is often insignificant, the algorithm analyzes the slope over a range of pixels (step). The step changes according to the complexity of the shaper defined by the category and ranges between 5 and 10 pixels in our implementation. This operation identifies the position (Z coordinate) of the geometric primitives corresponding to the specific archetype. Once this phase has been completed, it is possible to extract the object’s dimensions (X coordinate) in the points corresponding to the primitives. The other dimension of the same primitive is elaborated from the lateral view (Y coordinate) at the same position (Z coordinate).

The analysis described in section 3.1 shows that approximately 18% of the considered products across the macro-families 04 and 05 present a rotational symmetry. A different 3D recognition and modeling strategy have been implemented for these products to extract more formal details faster. No other CNN analysis is required if the product’s top and bottom views identify circular shapes. The CV algorithm described above extracts the points (pixels) of the silhouette curve from the front view of the object, and they are directly used for the subsequent modeling operations. Figure 9 shows how the points extracted by the CV algorithm can generate high-detailed curves to be directly used for modeling objects with rotational symmetry.

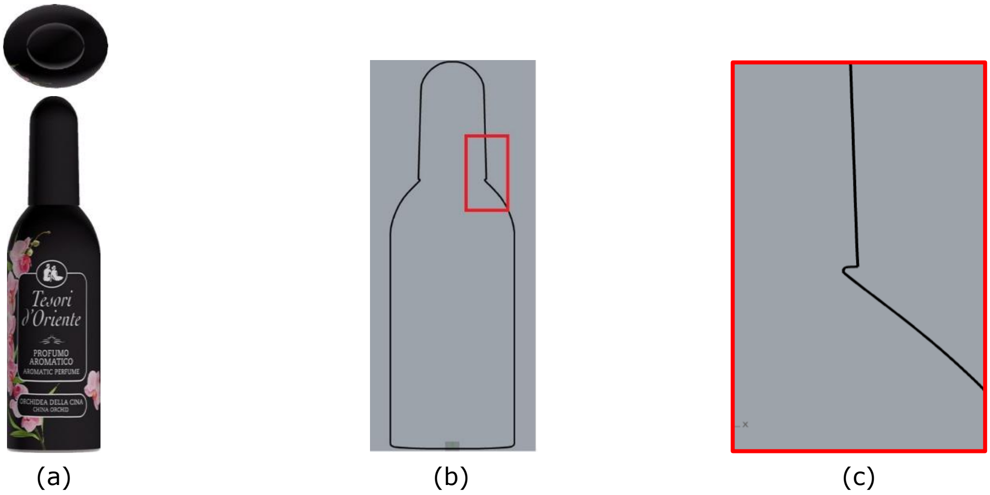


Figure 9: Example of the method applied to a rotational object: a) reference images of the product used for the CNN analysis; b) silhouette curve elaborated from the points extracted by the CV algorithm; c) magnification of the silhouette to show the obtainable level of detail with this product.

3.5 Automatic 3D Modeling According to the Archetype Category

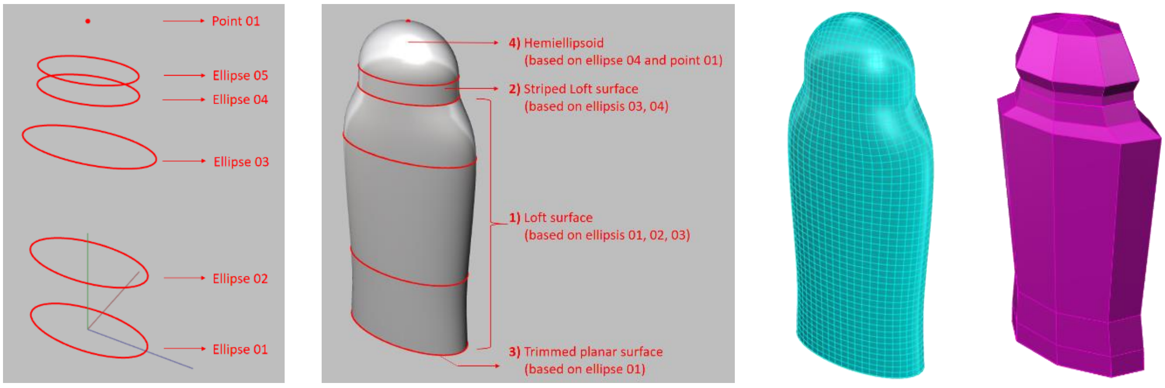
Usually, geometric modeling for real-time rendering is performed with polygonal modeling techniques. However, according to the proposed method, objects are modeled starting from curves and surfaces, and then they are tessellated with a proper polygonal density. Even with the limited number of parameters gathered from the CNN and CV analysis, better shape control led to obtaining high-quality 3D polygonal models. For this reason, it was necessary to identify a software application capable of managing automatized operations to generate suitable geometries. The software used for testing the method was Rhinoceros (https://www.rhino3d.com/), which allows: i) modeling curves and surfaces and the subsequent Sub-D conversion and check; ii) automated data input and modeling through a scripting language [1], [4], [5] [16] [16], [19].

The code language used for scripting is Phyton (https://www.python.org/), a cross-platform language used to execute a series of commands as a script or to make links between different applications; also used in this study for image-based shape analysis. The processing pipeline to be carried out to realize each archetype was outlined using the chosen modeling software and scripting language to identify the sequence of actions suitable for optimizing the quality of the result and the computational efficiency. This phase was carried out considering the subsequent conversion of the operations into scripts; the choices made aim at the generalization of the parameters and the repeatability of the approach.

The procedure for most of the archetypes, except for boxes and objects with rotational symmetry, was developed with the following step: a) definition of dimensional parameters; b) drawing of reference curves (ellipses, lines, arcs, etc.), set up so that they all have a coherent and functional orientation for the construction of the surfaces; c) modeling of vertically developed surfaces based on the drawn curves (extrusion, striped loft, normal loft, uniform loft, sweep2rail); d) bottom part modeling (planar surface); e) top part modeling (planar, ellipsoid, lofted end point surface); f) Surface normal check. Once the surfaces are completed, the model is converted to polygonal geometry as QUAD and SubD surfaces.

QUAD surfaces are optimized for i) preserving the shape of curved surfaces, the representation of edges, and the correct transition between surfaces; ii) obtaining a visually satisfactory smoothed model; iii) limiting the number of polygons. This solution is ideal for items that do not need more details and are ready to be texturized. SubD surfaces are extracted to obtain a polygonal model composed of a minimal number of faces coherent with the subdivision of the previously defined surface patches. This solution provides a polygonal reference mesh optimal for adding shape details using the typical Box Modeling technique (i.e., sculpting the shape from an elementary 3D primitive).

Once the optimal modeling path of the archetypes had been defined, this was translated into a scripting language using specific functions implemented in the modeling software. To generate the 3D model of a product, it is necessary to know the archetype to be used and, based on this, to have the related dimensional parameters, which typically are taken in correspondence with the peaks of the surface curvature gradients. The execution of a single script requires about 5-10 seconds, ensuring high efficiency compared to a manual modeling approach. Figure 10 presents the representative images for the main steps of the proposed method.

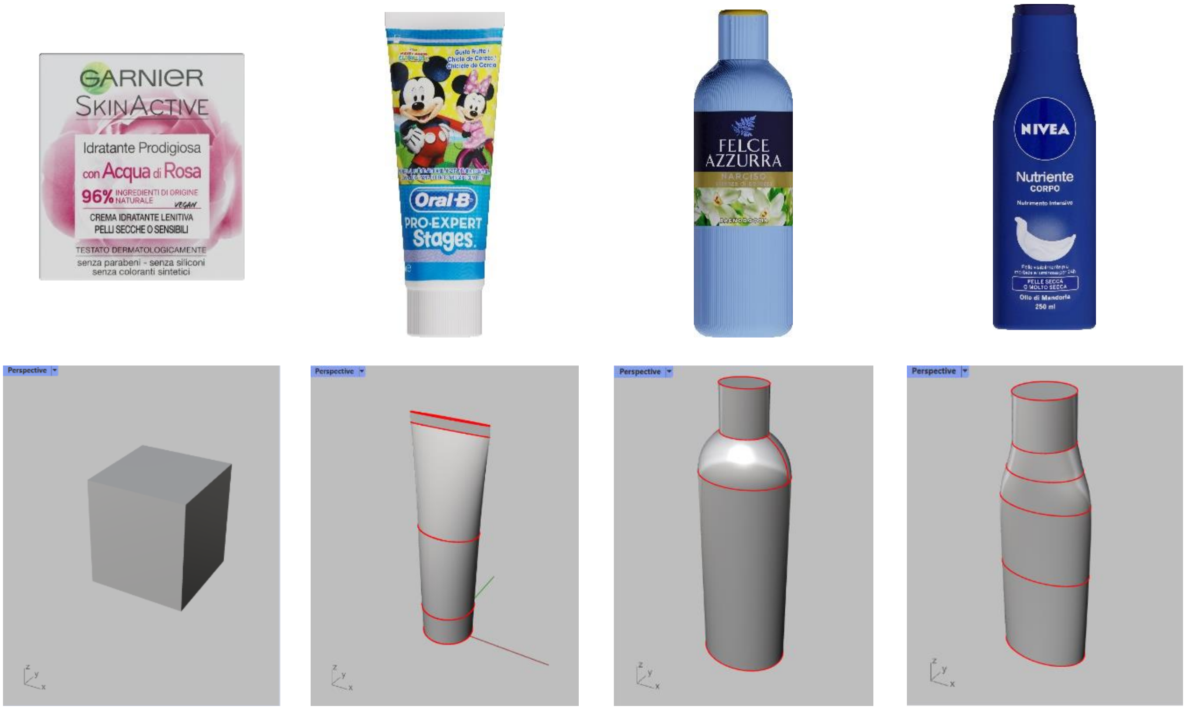


(a) (b) (c) (d)

Figure 10: Main 3D modeling steps of the proposed method, based on the identified archetype and the extrapolated dimension. Example on a product of archetype 050204: (a) main curves and point of the products, extracted from the orthographic images thanks to the CNN and CV analysis, the curves are built in a number consistent with the identified archetype and the starting point always lies on the XZ plane, so that the orientations are consistent; (b) 3D surface model obtained: sequence and type of surfaces are related to the identified archetype, at the end of the construction a check is

performed on normals orientation; (c) model conversion to Quad Mesh (d); model conversion to SubD surfaces.

Figure 11 shows some results obtained by applying the proposed process; starting from the reference images loaded in the system, the 3D models for surfaces were obtained, subsequently converted into mesh and sub-D, and ready for any modeling operations suitable for adding detail finishes.



(a) (b) (c) (d)

Figure 11: Examples of reference images and 3D models obtained through the proposed process related to different categories of archetypes: (a) macro-family box, archetype 020101; (b) macro-family tube, archetype 030101; (c) macro-family linear bottle, archetype 040303; (d) macro-family curved bottle, archetype 050303.

4 VALIDATION OF THE METHOD

The proposed method was evaluated from qualitative results and execution times. The results of the CNN and CV analyses for attributing a product to a specific archetype category were largely positive; over 90% of the attributions were correct. Two possible reasons for the remaining 10% of discrepant models: 1) the images introduced into the process were not perfectly orthographic when coming from the dataset of photographic images of real products; this highlights the crucial role of the real product acquisition for the proper implementation of the method; 2) the manual classification made by the human operator was sometimes inaccurate; here, the method demonstrates its potential for category detection by overcoming human uncertainty. It is interesting to note that the analysis of CNN uncertainty on a decision, if studied critically, can lead to improve and extend the catalog categories.

The 3D models created with the proposed method were also subjected to qualitative verification, comparing the results with reference models obtained through three-dimensional scanning. The digitizing system used to generate the reference models is the eviXscan 3D Optima Heavy Duty device manufactured by Evatronix (https://evixscan3d.com). This sensor is based on the structured light principle and consists of a blue light pattern projector in the center of the unit and two 5 MPixel side cameras. Each shot generates a 3D image (range map) of 5 million points, eliminating possible occlusions thanks to the double camera. In the test object acquisitions, the maximum standard deviation of error found in areas of overlap between 3D images was 32 μm.

The surfaces obtained from the proposed process were visually and computationally compared with the reference model. The analysis carried out on a visual level was performed by comparing the superimposed models and observing them from various perspectives. This qualitative comparison allows the observer to grasp the differences and to critically evaluate whether these are to be considered significant within the process or attributable, as found, to detail features not considered in the defined modeling flow (Figure 12).

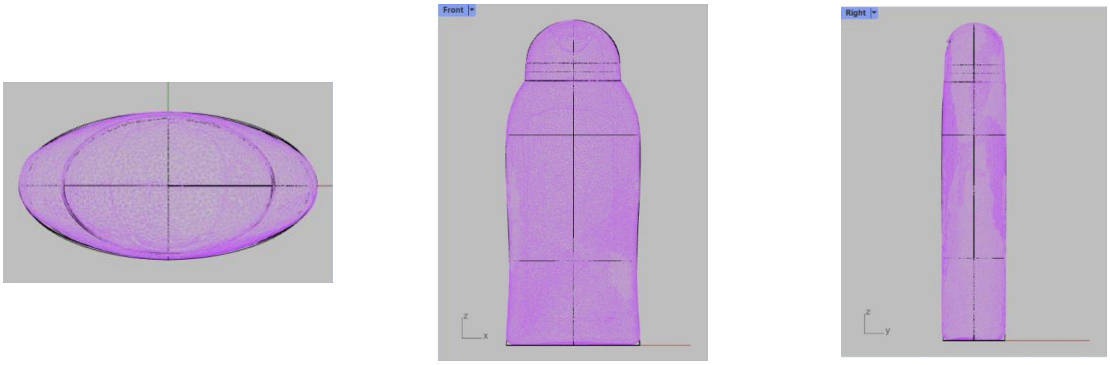
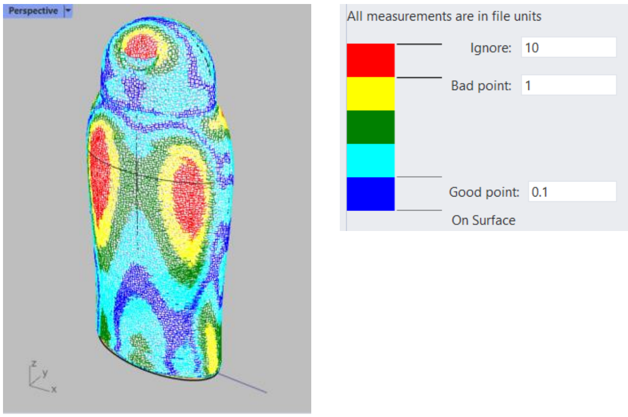


Figure 12: Qualitative comparison of a 3D model of a product archetype 050204.

For the quantitative comparison of the two models, a function was used to calculate the deviation of the nodes of the reference mesh model (obtained from the 3D survey) concerning the surfaces of the model generated with the process described here. The median and average distance values obtained in the tests were always less than one millimeter (Figure 13). The comparison was executed for objects of all the identified categories, and all qualitative and qualitative results validated the proposed process. Finally, from the point of view of temporal profitability, it should be emphasized that the execution time of the entire process, from identification to the 3D model in its various forms, for a single product is approximately 20 seconds, of which about 5-10 seconds for the modeling phase.



All measurements are in file units

Ignore: 10

Bad point: 1

Good point: 0.1

On Surface

Point test statistics:
Total points: 25231
Close point count: 25211
Average distance: 0.4432053
Median distance: 0.3376578
Standard deviation: 0.387879
Maximum distance: 2.976924
Minimum distance: 2.654054E-05
(Measures are in mm)

Figure 13: Quantitative comparison of a 3D model of a product archetype 050204.

5 CONCLUSION

This work presents a method to automate the modeling of 3D models of a massive set of products sharing the same shape structure. The method is based on the definition of a catalog collecting the archetypes of products. This catalog allows the grouping of different products according to the 3D modeling strategies elaborated by the human operator. The case study's catalog includes objects with at least double symmetry. However, objects without these specific symmetrical features can be included by adding further dedicated CNNs, provided a strict modeling strategy can define their geometry. It is worth noticing that objects with no catalogable geometry cannot be included. As emerged during the implementation, CNNs could lead to identifying new archetype groups, and it can support the human operator in developing alternative modeling strategies for these new products. According to this assumption, the method has great potential even if applied in other fields with different products with a well-defined and recognizable geometry.

The different CNNs, implemented according to the catalog, are used in sequence to speed up the analysis and increase the reliability of the results. The confusion matrix analysis proves that the CNNs can effectively identify different products from the catalog of archetypes by using orthogonal views of the real product. The measurement technique developed with CV algorithms allows extracting the right product dimensions or the rotational shape profile to build accurate and optimized 3D models.

The comparison with referenced 3D models shows that the errors are very low and often are located in areas of models containing details that are knowingly neglected. The comparison made with reference models reveals an accurate reconstruction of the object. In conclusion, the developed method shows promising perspectives for future works by optimizing the orthophotos analysis, increasing the variability within the catalog, creating new catalogs, or developing and adding new pipelines working with point clouds of 3D scanned objects.

Laura Loredana Micoli, https://orcid.org/0000-0001-6091-3125
Gabriele Guidi, https://orcid.org/0000-0002-8857-0096
Giandomenico Caruso, https://orcid.org/0000-0003-2654-093X

ACKNOWLEDGMENTS

The authors sincerely thank INVRSION S.r.l. (https://www.invrsion.com/), which supported the research by providing data and highlighting the real issues for reproducing large scenarios, and Matteo Agostini for his valuable contribution.

REFERENCES

[1] Aliaga, D.G.; Rosen, P.A.; Bekins, D.R.: Style Grammars for Interactive Visualization of Architecture, IEEE Transactions on Visualization and Computer Graphics, 13(4), 2007, 786–797. http://doi.org/10.1109/TVCG.2007.1024.

[2] De Luca, L.; Vron, P.; Florenzano, M.: A generic formalism for the semantic modeling and representation of architectural elements, The Visual Computer, 23(3), 2007, 181–205. http://doi.org/10.1007/s00371-006-0092-5.

[3] Demir, İ.; Aliaga, D.G.: Guided proceduralization: Optimizing geometry processing and grammar ex-traction for architectural models, Computers & Graphics, 74, 2018, 257–267. http://doi.org/10.1016/j.cag.2018.05.013.

[4] Ebert, D.S.; Kenton Musgrave, F.; Peachey, D.; Perlin, K.; Worley, S.: Texturing and Modeling: A Procedural Approach. Elsevier, 2003. https://linkinghub.elsevier.com/retrieve/pii/B9781558608481500292.

[5] Guidi, G.; Frischer, B.; Lucenti, I.: Rome Reborn - Virtualizing the Ancient Imperial Rome, In IS- PRS Archives Volume XXXVI-5/W47. ETH Zurich, Switzerland, 2007.

https://www.isprs.org/ proceedings/XXXVI/5-W47/pdf/guidi_etal.pdf.

[6] Guidi, G.; Micoli, L.L.: A Semi-Automatic Modeling System for Quick Generation of Large Virtual Reality Models, In ASME 2011 World Conference on Innovative Virtual Reality, 2011, 1–8. ASMEDC, Milan, Italy. http://doi.org/10.1115/WINVR2011-5512.

[7] Guidi, G.; Micoli, L.L.; Casagrande, C.; Ghezzi, L.: Virtual reality for retail, In 2010 16th International Conference on Virtual Systems and Multimedia, 2010, 285–288. IEEE, Seoul, Korea (South). http://doi.org/10.1109/VSMM.2010.5665949.

[8] Han, D.I.D.; Bergs, Y.; Moorhouse, N.: Virtual reality consumer experience escapes: preparing for the metaverse, Virtual Reality, 26(4), 2022, 1443–1458. http://doi.org/10.1007/s10055-022-00641-7.

[9] Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H.: MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications, 2017. http://doi.org/10.48550/ARXIV.1704.04861.

[10] 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, 23(8), 2017, 2003–2013. http://doi.org/10.1109/TVCG.2016.2597830.

[11] Kanazawa, A.; Tulsiani, S.; Efros, A.A.; Malik, J.: Learning Category-Specific Mesh Reconstruction from Image Collections, 2018. http://doi.org/10.48550/ARXIV.1803.07549.

[12] Kar, A.; Tulsiani, S.; Carreira, J.; Malik, J.: Category-specific object reconstruction from a single image, In 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, 1966–1974. http://doi.org/10.1109/CVPR.2015.7298807.

[13] Kato, H.; Ushiku, Y.; Harada, T.: Neural 3D Mesh Renderer, 2017. http://arxiv.org/abs/1711. 07566.

[14] Liu, S.; Li, T.; Chen, W.; Li, H.: Soft Rasterizer: A Differentiable Renderer for Image-based 3D Reasoning, 2019. http://arxiv.org/abs/1904.01786.

[15] Martin, I.; Patow, G.: Ruleset-rewriting for procedural modeling of buildings. Computers & Graphics, 84, 2019, 93–102. http://doi.org/10.1016/j.cag.2019.08.003.

[16] Mas, A.; Martin, I.; Patow, G.: Simulating the Evolution of Ancient Fortified Cities. Computer Graphics Forum, 39(1), 2020, 650–671. http://doi.org/10.1111/cgf.13897.

[17] Müller, P.; Wonka, P.; Haegler, S.; Ulmer, A.; Van Gool, L.: Procedural modeling of buildings. In ACM SIGGRAPH 2006 Papers on - SIGGRAPH '06, 614. ACM Press, Boston, Massachusetts, 2006. http://doi.org/10.1145/1179352.1141931.

[18] Nash, C.; Williams, C.K.I.: The shape variational autoencoder: A deep generative model of part-segmented 3D objects, Computer Graphics Forum, 36(5), 2017, 1–12. http://doi.org/10.1111/cgf.13240.

[19] Parish, Y.I.H.; Mller, P.: Procedural modeling of cities, In Proceedings of the 28th annual conference on Computer graphics and interactive techniques, 2001, 301–308. http://doi.org/10.1145/383259.383292.

[20] Simonyan, K.; Zisserman, A.: Very Deep Convolutional Networks for Large-Scale Image Recognition, 2014. http://doi.org/10.48550/ARXIV.1409.1556.

[21] Sinha, A.; Bai, J.; Ramani, K.: Deep Learning 3D Shape Surfaces Using Geometry Images. In B. Leibe;J. Matas; N. Sebe; M. Welling, eds., Computer Vision ECCV 2016, 910, 2016, 223–240. http://link. springer.com/10.1007/978-3-319-46466-4_14.

[22] Sinha, A.; Unmesh, A.; Huang, Q.; Ramani, K.: SurfNet: Generating 3D shape surfaces using deep residual networks, 2017. http://arxiv.org/abs/1703.04079.

[23] Stiny, G.; James Gips: Shape Grammars and the Generative Specification of Painting and Sculpture, In IFIP Congress, 1971, 1460 – 1465.

[24] Stiny, G.; Mitchell, W.J.: The Palladian grammar, Environment and Planning B: Planning

and Design, 5(1), 1978, 5–18. http://doi.org/10.1068/b050005.

[25] Su, H.; Qi, C.R.; Li, Y.; Guibas, L.J.: Render for CNN: Viewpoint Estimation in Images Using CNNs Trained with Rendered 3D Model Views, In 2015 IEEE International Conference on Computer Vision (ICCV), 2015, 2686–2694. http://doi.org/10.1109/ICCV.2015.308.

[26] Tulsiani, S.; Kar, A.; Carreira, J.; Malik, J.: Learning Category-Specific Deformable 3D Models for Object Reconstruction, IEEE Transactions on Pattern Analysis and Machine Intelligence, 39(4), 2017, 719–731. http://doi.org/10.1109/TPAMI.2016.2574713.

[27] Xie, H.; Yao, H.; Sun, X.; Zhou, S.; Zhang, S.: Pix2Vox: Context-aware 3D Reconstruction from Single and Multi-view Images, In 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 2019, 2690–2698. http://doi.org/10.1109/ICCV.2019.00278.

[28] Xie, H.; Yao, H.; Zhang, S.; Zhou, S.; Sun, W.: Pix2Vox++: Multi-scale Context-aware 3D Object Reconstruction from Single and Multiple Images, International Journal of Computer Vision, 128(12), 2020, 2919–2935. http://doi.org/10.1007/s11263-020-01347-6.

[29] Yumer, M.E.; Asente, P.; Mech, R.; Kara, L.B.: Procedural Modeling Using Autoencoder Networks. In Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology, 2015, 109–118, ACM, Charlotte NC USA. http://doi.org/10.1145/2807442.2807448.