Real-World Textured Things:
A Repository of Textured Models
Generated with Modern Photo-Reconstruction Tools⋆

Andrea Maggiordomo[a,*], Federico Ponchio[b], Paolo Cignoni[b], Marco Tarini[a]

[a] University of Milan, Italy
[b] ISTI - CNR, Italy

Abstract

We are witnessing a proliferation of textured 3D models captured from the real world with automatic photo-reconstruction tools by people and professionals without a proper technical background in computer graphics. Digital 3D models of this class come with a unique set of characteristics and defects – especially concerning their parametrization – setting them starkly apart from 3D models originating from other, more traditional, sources. We study this class of 3D models by collecting a significant number of representatives and quantitatively evaluating their quality according to several metrics. These include a new *invariant* metric we carefully design to assess the amount of fragmentation of the UV map, which is one of the main weaknesses potentially hindering the usability of these models. Our results back the widely shared notion that models of this new class are still not fit for direct use in downstream applications (such as videogames), and require challenging processing steps. Regrettably, existing automatic geometry processing tools are not always up to the task: for example, we verify that the available tools for UV optimization often fail due to mesh inconsistencies, geometric and topological noise, excessive resolution, or other factors; moreover, even when an output is produced, it rarely represents a significant improvement over the input (according to the aforementioned measures). Therefore, we argue that further advancements are required by the computer graphics and geometry processing communities specifically targeted at this class of models. Towards this goal, we share the models we collected in this study as a new public repository, *Real-World Textured Things* (RWTT), intended as a benchmark to systematic field-test and compare future algorithms. RWTT consists of 568 carefully selected textured 3D models representative of the most popular photo-reconstruction tools currently available. We also provide a web interface to browse the dataset by the metadata we collected during our experiments and a tool, TexMetro, to compute the same set of measures on generic UV mapped datasets.

Keywords: Benchmarks, Photo-Reconstruction, 3D Acquisition, Quality Measures, Surface Parametrization, UV Maps, Textures, Real-World Models

1. Introduction

In the last decade, many research and implementation efforts have culminated in the development of algorithms, technologies, and suites that facilitate creating and publishing 3D content , attracting users not necessarily equipped with a technical background in computer graphics or geometry processing. Among the most influential and exciting advancements, photo-reconstruction techniques played a primary role,

impacting a variety of application fields such as artwork and architectural documentation, environmental monitoring, data acquisition for manufacturing, 3D assets creation. Commercial and free software solutions are now available to both professionals and hobbyists, and can be used to generate accurate 3D digital copies of real-world objects in an almost fully automatic way.

This democratization of 3D content creation determined the diffusion of textured 3D digital models sharing certain specific characteristics, differing both qualitatively and quantitatively from those of 3D representations typically targeted – and assumed – by the large majority of academic works on computer graphics and geometry processing. Consequently, there is now an increased demand for robust techniques designed to process and manipulate this class of 3D models, and further research efforts are required to fill this gap.

Characteristics of Textured Models Generated with Photo-Reconstruction Tools. Digital 3D models acquired with photo-reconstruction techniques share a combination of traits that cannot be found in any other class of 3D models (such as meshes digitally sculpted or manually designed by professional 3D modelers, range-scanned models, or procedurally generated models).

The **meshing** is typically high-resolution but low quality, triangle-based, and completely irregular. The frequent inaccuracies in the 3D reconstruction result in geometrical noise and meshing defects. While many solutions have been suggested to face these problems (Attene et al., 2013; Ju, 2004; Chu et al., 2019), the proposed approaches focus on the geometrical aspect and do not take into account the presence of an existing parametrization. Even more importantly, geometry is but one aspect of the description: photo-reconstructed meshes are enriched with one or multiple high-resolution textures derived from the same set of photographs used to reconstruct the shape; these are a central part of the description of the object, as well as its UV map, i.e. the parametrization defined to map the texture data onto the object surface.

The **UV maps** of the 3D models present the most distinctive traits, directly reflecting the way they have been constructed (see Section 1.1 below). Specifically, they are atlas-based parametrizations where the atlases are typically extremely fragmented and with irregularly shaped charts, to a much larger extent than other categories of models (see for example Fig. 1).

This drastically exacerbates the numerous defects and costs related to texture seams and UV map discontinuities, impacting downstream applications. The reader is referred to (Yuksel et al., 2019) for a more comprehensive review of these issues. In sum-

Figure 1: One model of the RWTT dataset, with its texture and UV map.

mary, they include poor GPU memory usage, due to a combination of packing inefficiency and the need to replicate texels on a small band around each seam to avoid noticeable texture "bleeding" artifacts; unavoidable minor rendering artifacts caused by the inconsistent bilinear interpolation on the seams, even in the presence of said texel duplication; inability to use MIP-mapping without incurring in additional bleeding artifacts (unless the width of the texel duplication bands is enlarged exponentially to the maximal MIP-map level); vertex duplications in the data structure encoding the mesh, which are necessary to be able to encode the seams in the first place, implying an additional cost both in memory and per-vertex workloads; consequent constraints on processing or preprocessing performed over the mesh data structure, for example hindering to some extent the compression, streaming, coarsening, construction of multi-resolution structures (such as Cignoni et al., 2005).
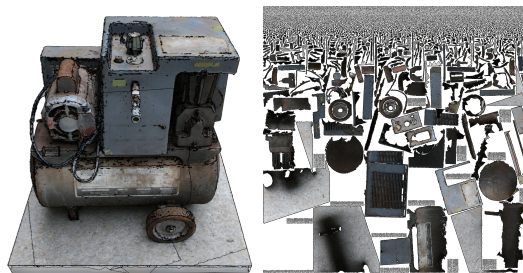
Contributions. With this in mind, in this paper we offer the following contributions. We present an **analysis** of the aforementioned class of 3D models and their characterizing features, by collecting a large sample of representative models and performing systematic measurements, according to several criteria that we identify as relevant for geometry processing algorithms and computer graphics applications in general. We focus on the UV maps, its presence being one defining characteristics of this class of models.

The analysis includes a **new quality measure**, the atlas *crumbliness*, designed to capture the severity of texture seams in a given UV map. This new measure is succinct, easy to compute, and has desirable properties such as being invariant to UV scaling and mesh subdivision.

Since our analysis shows that the quality of the parametrizations in our sample data is low, we **evaluate** a selection of commercial, open source and academic software on the task of recomputing UV maps for these 3D models, and show how existing algorithms and implementations typically either fail, are unable to improve the quality of the maps, or are not practical.

We make our **dataset**, Real-World Textured Things (RWTT), publicly available on the web. The dataset has been curated and organized to facilitate batch-processing, and is suited to test, benchmark, and challenge existing and future geometry processing algorithms. Overall, RWTT features 568 models generated by users of the aforementioned photo-reconstruction technologies and released with permissive licenses, striving to represent the wide variety of existing user categories, photo-reconstruction tools, application fields and styles. Our aim is to provide a standard, realistic, and challenging testbed for algorithms, able to indicate their usability in the real world. All models in RWTT come pre-assessed according to the measures we collected, and are browsable and searchable by them.

Finally, we provide the **measuring tool** we used to evaluate the models as a publicly available, standalone software application: TexMetro (Section 7). TexMetro produces a set of meshing and UV quality measures and statistics for user-provided models (identical to the ones we used to evaluate the RWTT dataset), and is intended as a tool to help researchers and practitioners assessing and comparing the effectiveness of future algorithms and techniques designed to target this class of models. In this, we are inspired by the impact that tools of this kind had in the geometry processing community (Section 2.1).

1.1. Background in 3D Photo-Reconstruction

Photo-reconstruction techniques (Hartley and Zisserman, 2004; Seitz et al., 2006; Remondino et al., 2014) generate a textured 3D mesh starting from a set of high-resolution 2D pictures shot at the subject.

There are many possible variations, but the general principle can be summarized as follows. First, Structure-From-Motion techniques are used to estimate camera parameters: feature points are identified on each input image, then the matching feature points are identified across the images using appropriate similarity measures. In a global adjustment phase, a set of camera positions is estimated, together with 3D positions of feature points, to maximize consistency, e.g. by minimizing the back-projection error, or by maximizing the color consistency across images. Once the camera parameters have been computed, Multi-View-Stereo algorithms are employed to generate a dense point cloud.

Finally, the point cloud is processed to remove noise and outliers, and fused into one polygonal mesh representation. The mesh representation is used to estimate the occlusion and visibility of each pixel in the picture. A subset of the visible pixels are then copied into one or more high-resolution textures, which provide the majority of the visual detail of the final result. Textures are sometimes post-processed to alleviate shadow artifacts, color jumps when switching from one source image to another, or can optionally store additional material and surface information (e.g. surface normals).

Each step of this chain constitutes a challenging problem, which has been studied in separation or within complete systems. The research on each sub-problem has recently matured enough so that a large number of off-the-shelf tools integrating implementations of all the used techniques are now available either as commercial or open source software (Capturing Reality, 2014–2019; Autodesk, Inc, 2019b; Agisoft LLC, 2010–2019; 3Dflow SRL, 2014–2020; Pix4D SA, 2011–2019; Moulon et al., 2016; Ranzuglia et al., 2013; Bentley Systems, Inc, 2014–2019; itSeez3D, 2019).

For our purposes, the most relevant part is the construction of the UV map. Parametrization construction techniques traditionally employed in other contexts are often not adequate to the task, in part because these methods often lack the full automatism required in this scenario. Moreover, many methods (automatic or user-assisted) are hindered by the characteristics of the meshing (i.e. its large resolution and various defects). At the same time, the presence of a set of images detailing every part of the object surface unlocks UV map construction strategies that are only available in this scenario. Namely, each mesh triangle can be assigned to one of the photos using heuristics based on estimated visibility or other factors, implicitly defining the components of the UV atlas which then just need to be packed into one or more textures.

Note also that that mapping distortions, traditionally of central importance in the effort of constructing a good quality UV map for a given mesh, are not particularly relevant in this context. Differently from most other scenarios, where a UV map is constructed before the texture is filled with the signal samples, here the texture content *precedes* the UV map construction. Because the source of the texture colors are limited to the images, there is no gain in diminishing the mapping distortions which can already present from these images to the 3D surface. Therefore, instead of reducing the distortion by optimizing a suitable map, the optimization strategy can only consist in choosing the least distorted input image to use as a source for a given triangle (for example, the image offering the most orthogonal view of that portion of the surface, barring other considerations such as occlusion, presence of highlights, optimization of island shapes, and others).

These factors determine that the UV maps commonly found in this important class of models feature unique and specific characteristics, motivating the present work.

2. Related Work

The practice of creating good, comprehensive, realistic, challenging benchmarks is well recognized, and there is no need to state their importance in promoting good practices and improving replicability, comparability, and validation of research. In this section we briefly summarize other dataset contributions made to the geometry processing community and public repositories of textured models. There are many geometric and computer vision datasets available that target a large variety of tasks.

Datasets for machine learning generally feature large collections of objects, and include metadata and annotations to facilitate the training and validation of learning-based algorithms. The ModelNet benchmark (Wu et al., 2015) is a collection of more than a hundred thousand CAD models, and has been successfully used to train and benchmark object recognition algorithms. ShapeNet (Chang et al., 2015) is a large collection of annotated 3D objects, providing geometric and volumetric data, screenshots, textual descriptions and categorizations. On a similar note, the ABC dataset (Koch et al., 2019) is a massive dataset of one million CAD models proposed to test the effectiveness of learning-based approaches to the solution of common geometric problems such as computing surface normals.

Scanning campaigns have produced popular model repositories, such as the Digital Michelangelo Project (Levoy et al., 2000) and the Stanford Scanning Repository (Levoy et al., 2005), whose models are known to the entire computer graphics community. None of these models, however, have textures, and they have been generated using 3D scanners; in contrast, the models featured in our dataset are created with photo-reconstruction algorithms.

The Princeton Shape Benchmark (Shilane et al., 2004) is a large collection of low-resolution models with very simple geometries. Turk and Mullins (2003) focus instead on high-resolution models. Both datasets are now rather outdated.

The wide availability of consumer-level hardware capable of capturing depth images to produce object scans has resulted in a proliferation of datasets based on RGB-D data. The review of those datasets is beyond our scope, and we refer the reader to the excellent survey by Firman (2016).

More related datasets are Thingi10K (Zhou and Jacobson, 2016), which features real-world objects and targets 3D printing shapes, and (Choi et al., 2016), a large dataset of more than ten thousand object scans and the associated RGB-D data that, like ours, features models obtained from 3D reconstruction, but again without textures.

SceneNN (Hua et al., 2016) and ScanNet (Dai et al., 2017) feature annotated scenes reconstructed from RGB-D video sequences. Among the existing datasets, these are, to our knowledge, the most similar to ours; however, they do not contain models reconstructed with photogrammetric techniques, feature mostly indoor scenes, and the 3D reconstructions are not particularly accurate.

As outlined above, geometric and computer vision datasets rarely include textures. Textured models can be found in asset stores such as the Unity Asset store (Unity technologies, 2019), the Unreal Marketplace (Epic Games, 2004 – 2019), and online repositories such as TurboSquid (TurboSquid, 2019). Models from these sources however are typically non-free, sculpted, and manually textured by 3D artists.

In contrast to all of the above, Real-World Textured Things is a collection of large, textured 3D models generated with a variety of photo-reconstruction software tools, and published with permissive licenses to allow use in research and other practical contexts.

2.1. Quality Measurement Tools

The fruitful idea of providing public tools intended to be used to evaluate specific quality measures of user-provided models has several predecessors. Among the most notable is *Metro* (Cignoni et al., 1998), which computes and reports the Hausdorff distance between pairs of surfaces provided as triangular meshes. Over the last two decades, *Metro* has helped to assess and compare vast amounts of algorithms, e.g. for surface simplification, remeshing, surface reconstruction, and other geometry processing tasks, as reported by thousands of research papers (Google Scholar, 2019). As an additional benefit, the adoption of *Metro* contributed to uniform the way quality measures are defined and implemented. Overall, *Metro* has proven to be a tremendous aid to the research community striving to invent new algorithms and solutions toward a given set of geometry processing problems.

3. Dataset

As mentioned in the introduction, one of the motivations behind the creation of the Real-World Textured Things dataset is to provide researchers and developers a challenging testbed of textured 3D models indicative of the actual models that can be encountered in practice. Rather than generating the models in-house, we have collected publicly available models created and released by professional practitioners and hobbyists. This strategy has been successfully employed by other authors (e.g. Zhou and Jacobson, 2016; Koch et al., 2019) and has the important advantage of allowing us to include models generated with various software packages, which is crucial to obtain a representative sample of the 3D data that can be found in the real world. In particular, it is reasonable to expect the emergence of some common traits characterizing the geometry and textures of the models in the dataset according to the implementation details and algorithmic choices of different reconstruction pipelines.

At the time of writing, the dataset features 568 models created by 129 unique authors using at least 18 different acquisition pipelines. Fig. 2 reports a taxonomy of the models featured in the RWTT dataset, categorized according to the reconstruction software used to generate them; this information was obtained from the textual descriptions and tags available when the models were retrieved.

We have collected models that were published across 6 years from 2013 to 2019 from three sources: Sketchfab (2011–2019), the Smithsonian 3D Program (Smithsonian Institution, 2013–2019) and the Zamani Project (Rüther et al., 2020). SketchFab is a popular online repository for publishing and sharing 3D content used by thousands of users and constantly growing; it features a large collection of 3D models reconstructed from photographs and conveniently exposes a web API to automatically retrieve metadata such as the model author, description and tags, among others. The Smithsonian 3D Program is a digitization project from the Smithsonian Institution which published on the web and made available a selection of 3D reconstructions of artifacts and relics. The Zamani Project is an ongoing effort to document important heritage and archaeological sites in Africa by providing, among other material, high-resolution textured 3D reconstructions.

We have been careful to almost exclusively include models that have been released with permissive licenses to allow anyone using the dataset to freely share their results: all of the models from SketchFab and the Zamani Project are released under various declinations of the Creative Commons 4.0 license; the models from the Smithsonian 3D Project follow the "Smithsonian Terms of Use", allowing fair use as long as the original source is cited. We plan to keep the dataset updated by adding new models in the future.

3.1. Dataset Curation

The dataset is organized as a collection of folders using a progressive naming scheme to facilitate the automatic processing of the 3D models. Each model in the dataset is assigned a progressive identifier ID and is stored in a folder named RWTID.
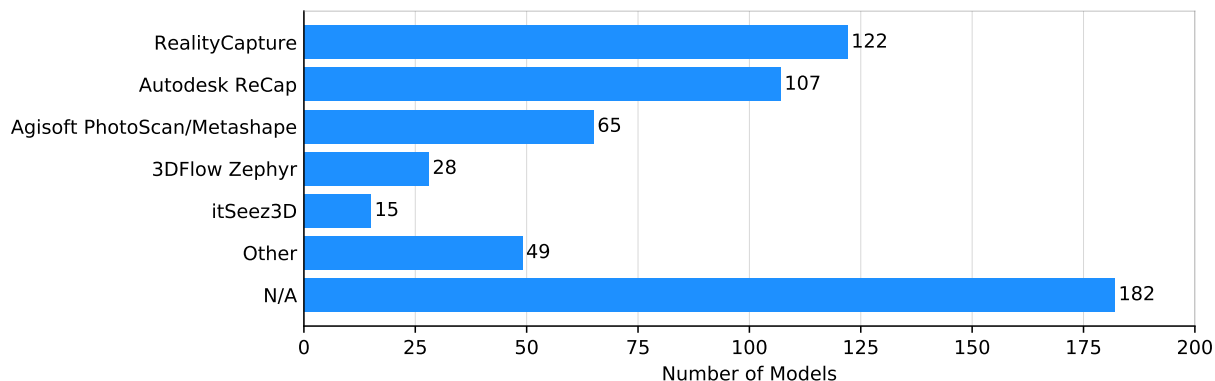
For each model, we provide the following data:

Figure 2: Overview of the different software tools used to create the models featured in Real-World Textured Things. There are at least 18 different reconstruction pipelines represented in the dataset; here for compactness we only report the most frequent occurrences.

- The 3D model in OBJ format with a corresponding MTL file defining the materials

- The texture images referenced by the materials

- A JSON file with the provenance metadata of the model

- A JSON file with geometric and parametrization measures

Although the original models and textures came in a variety of formats, we have converted all of them to the same format for uniformity and convenience; in several cases, manual fixing of the object files was required to restore correct material and texture image references. All the models are provided as OBJ files, and the textures as PNG images. Whenever necessary, the conversion only involved the file formats while the 3D and image data was left untouched; the models archived in their original formats are available separately. The textures we provide generally encode surface colors; however, several models also come with high-resolution textures storing other signals such as surface normals, roughness, specularity, and ambient occlusion.

Each 3D model comes with a corresponding *metadata.json* file reporting information such as the model name, the original author, the date on which the model was published and the license used, a semantic category we assigned to the models for classification purposes, and a unique resource identifier. The unique identifier points to the original data source: SketchFab models use the URI assigned by SketchFab itself, models published by the Smithsonian 3D Program use the web link to the object information page, while models published by the Zamani Project use their Digital Object Identifiers (DOIs). The metadata file optionally also reports, if available, the software used to generate the model, a textual description of the subject and a collection of tags that the original author may have provided.

Finally, we also include a *texmetro.json* file generated with the TexMetro tool. This JSON file contains information about the geometry, parameterization, and textures of the models; a detailed description of the TexMetro tool and its output is given in Section 7.

4. Quality Measures for Textured 3D Meshes

The quality of a textured mesh can be assessed by several quantifiable criteria, reflecting the resources that will be necessary to deploy the model (e.g. store it, stream it, or render it) and the quality of the rendered images, and thus the degree of usability of a given model in a specific context.

Models in RWTT are scored according to a suitable set of such measures, which is described in this Section. Different measures can impact downstream applications in different ways. Collectively, the set of

measures reveals the peculiarity of the class of meshes we are focusing on, setting it apart from meshes commonly encountered in contexts different from automated photo-reconstruction.

Our set of measures includes general criteria such as data size and surface genus, UV map measures concerning the parametrization, and defects (number of occurrences of typical inconsistencies).

4.1. General Measures

Geometric and topological measures include the model resolution in terms of faces, vertices, and edges, the surface area, the number and total length of the mesh boundaries and the surface genus. Topological information is relevant since many geometry processing algorithms require strict assumptions on the topology of the input data. Texture resolution and number of textures are also meaningful measures, as these are all quantities that would make this data impractical to be handled by algorithms that do not scale well with the input size.

4.2. UV Map Measures

The quality of the surface parametrizations and texture maps are the main focus of our analysis and in the following we provide a detailed description of the measures we have adopted.

4.2.1. UV Occupancy

We define UV occupancy as the ratio of the number of texels strictly inside any UV triangle over the total number of texels. A better UV occupancy results in a higher texel density over the surface for a fixed texture resolution, or a lower GPU memory footprint for the same rendering quality.

Occupancy is affected by the quality of the atlas packing: tightly packed atlases waste fewer texels between the charts. Atlases that are highly fragmented also have a worse occupancy, as they require comparatively more UV area to ensure that the charts are suitably distanced from each other to avoid mip-mapping artifacts.

4.2.2. Atlas Crumbliness and Solidity

As summarized in the introduction, the presence of texture seams is somewhat detrimental to the usability of the dataset, for several different reasons. While texture seams are, to some extent, necessary in any textured mesh, we focus on the presence of redundant texture seams that are due to the fragmentation of the UV atlas independent from the complexity of the reconstructed 3D shape (see Fig. 3). Therefore, we need to adopt a quality measure to assess the level of fragmentation of an existing texture atlas.

Because there is no standard, we devise a new measure for this. We consider several alternatives. We observe that the number of charts is not, alone, a good descriptor, for two reasons: first, charts can have varying sizes and boundaries of varying complexity; second, a chart may include "non-disconnecting" cuts, i.e. cuts that do not split it into disconnected components. The number of seam edges, or the ratio of seam edges to non-seam edges, are also not adequate: these measures would be affected by a regular subdivision of the mesh, which does not affect the level of fragmentation. Similarly, the total length of the seam edges would not be invariant to UV scaling.

Instead, we adopt a new measure, which we name *crumbliness*. Crumbliness is the ratio of the total length of the perimeter of the atlas charts, summed over all charts, over to the perimeter of an ideal circle having the same area as the summed area of all charts. Because the circle is the shape maximizing the area for a given perimeter, this measure has 1 as the lower bound. In formulas, if a_i is the area of chart i and p_i is its perimeter, the crumbliness C of the atlas is given by

$$C = \left(\sum_i p_i \right) \left(4\pi \sum_i a_i \right)^{-\frac{1}{2}}$$

Crumbliness is independent of the mesh resolution, invariant under element subdivision, and scale-independent. As such, it can be used to compare atlases of different models and for textures of varying sizes.

(a) Crumbliness = 7.4, Solidity = 0.135

(b) Crumbliness = 21.1, Solidity = 0.047

(c) Crumbliness = 94.7, Solidity = 0.010
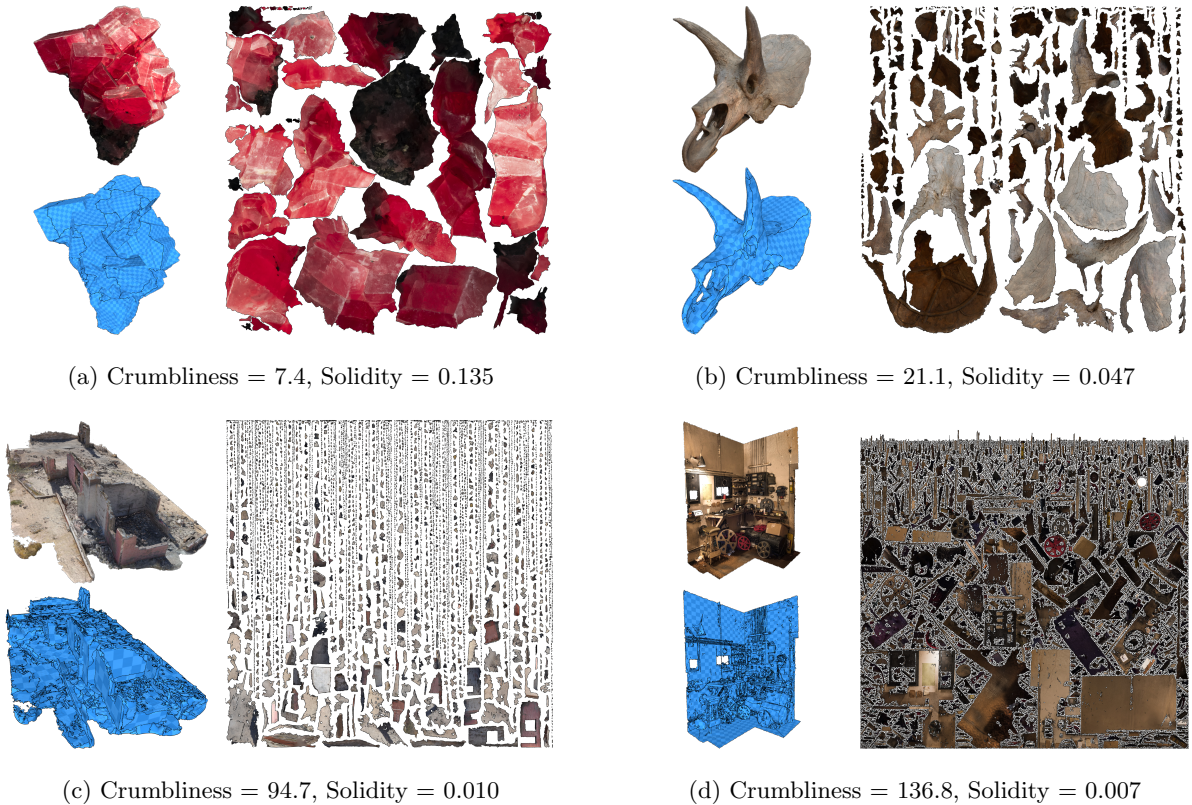
(d) Crumbliness = 136.8, Solidity = 0.007

Figure 3: Four models with their texture seams highlighted. A few of the texture maps featured in our dataset have a relatively low chart count (3a). Many others exhibit a surprisingly large number of texture seams, unjustified by their geometrical or topological complexity (3b, 3c, 3d). *Crumbliness* and *solidity* measures capture the fragmentation of the texture atlas.

We also define *atlas solidity* as the inverse of crumbliness. Compatibly with many other quality measures, atlas solidity has an upper bound of 1, and higher values indicate a better quality.

We observe that the square of the crumbliness gives, up to a constant factor, the quantity sometimes referred to, in geometry, as the "isoperimetric ratio" of a given 2D shape, which is the ratio between its squared perimeter and its area. We favor the use of "crumbliness" and "solidity", as their name is more descriptive and more intuitive to correlate with quality, their numerical quantities are easier interpreted (being normalized), and they empirically appear to distribute better among real-world datasets, compared to their squared counterparts (see Fig. 7).

4.2.3. Uniformity of Texture Sampling

A well-understood characteristic of any parametrization is the distortion it introduces. In the ideal case, the mapping between the surface and the texture preserves areas, angles, or lengths (implying both areas and angles); the failure in preserving these features is measured by various *distortion measures*. Conformal distortion reflects angles variation, area distortion reflects area variation, and isometric distortion reflects length variation. A multitude of different measures have been presented in the literature (for each case), usually to minimize them during automatic UV map construction (see Floater and Hormann, 2005 for a survey and, for more recent examples, Liu et al., 2008; Smith and Schaefer, 2015; Jiang et al., 2017; Poranne et al., 2017 among many others).

In our context, which is texture mapping, distortion is relevant because it directly determines how texture samples will be distributed on the surface. Area distortion translates to nonuniform sampling density, and conformal distortion translates to anisotropic sampling.
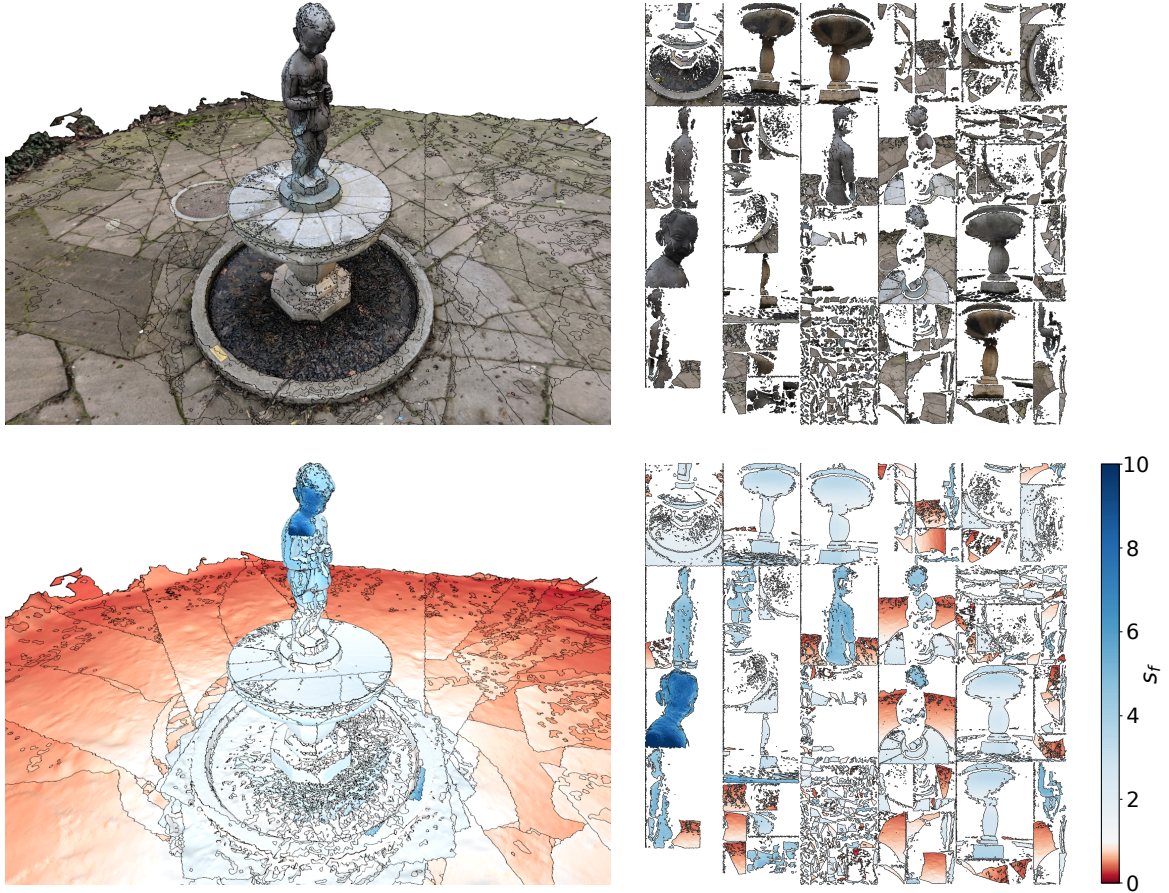
Figure 4: Visualization of the scalar field s_f on a model affected by uneven sampling frequency. Regions densely and sparsely sampled are colorized in blue and red respectively, reflecting the amount of color information encoded in the texture image. In this case, the model was clearly reconstructed using close-ups of the statue with the pavement left in the background.

Sampling density can be hugely varying in models of RWTT, due to the different relative resolution of the pictures used to reconstruct the models, and the different distance between the acquiring camera and the captured object. Instead of adopting any specific measure of area distortion, we choose to measure directly its effect by analyzing the density of texture samples over the mesh.

To this end, we define a per-element scalar quantity s_f that represents the texture sampling density as the ratio between the local (per-element) and global (per-mesh) change of scale from 3D to UV:

$$s_f = \frac{A_{\mathrm{UV}}(f)/A_{\mathrm{3D}}(f)}{\sum_f A_{\mathrm{UV}}(f)/\sum_f A_{\mathrm{3D}}(f)}.$$

Where A_{UV} and A_{3D} indicate element areas computed in UV and 3D space respectively. The local value of s_f is close to 1 for uniform sampling distributions, approaches 0 as the mapping resolution of f gets smaller relative to the global resolution of the map, and becomes greater than 1 otherwise. Therefore the scalar field s_f indicates, over a model, where the color signal is undersampled and oversampled w.r.t. to the average sampling density of one given model. Fig. 4 shows an example of how uneven sampling can impact the quality of a textured model when it is rendered.

As aggregate measures, we take the variance and percentile ranges (which may be less sensitive to outliers) of s_f over the mesh (note that the average is 1 by construction). Small values reflect more uniform sampling, and the minimal value 0 represents a perfectly uniform sampling. Note that this is not necessarily

desirable for models in our class; in fact, even a vastly nonuniform sampling density can correctly reflect the different semantic importance of certain parts of the model relative to others (such as an object in focus vs. background) or different quantities of available color data.

4.2.4. UV Conformal Distortion

UV maps that exhibit low conformal distortion are desirable because they imply a more isotropic texel sampling distribution over the surface, which positively affects the quality of the signal reconstruction and avoids texture stretching artifacts. Because this factor is not captured by the sampling density, we measure conformal distortion separately.

To this end, we adopt a measure based on the quasi-conformal distortion proposed by Kharevych et al. (2006), which is the ratio of the smallest to largest singular value of the Jacobian of the mapping. We define the UV atlas conformality as the inverse of the quasi-conformal distortion; we opt for this formulation so that the maximal of value of 1 corresponds to the best case (a perfectly conformal map), consistently with most of the other quality measures in our set.

4.2.5. Discrepancy in the Signal Reconstruction at Seam Edges

A specific defect linked to the presence of texture seams is the possibility of observing discrepancies in the signal reconstruction at each side of the seam, which can be due to either mismatching texel values, or to inconsistent bilinear interpolation between texture values. This does not depend on the UV map alone, but on the texture data as well; therefore, we include a separate measure to assess the extent of this problem for a given model.

To this end, we adopt a measure of seam discrepancy recently defined (for the purpose of minimizing it) in Liu et al. (2017), which, for completeness, we describe in the following.

Given an UV edge e, let $e(t)$ denote the linear combination of its endpoints, with the parameter t restricted to the $[0, 1]$ interval. A (manifold) edge e of the 3D mesh is a seam edge if corresponds to two non coinciding UV edges e_1, e_2. Let f be the function returning the texture lookup value for a given UV coordinate, using bilinear interpolation of the four closest texels. The reconstruction discrepancy at a seam edge e is then defined as:

$$D(e) = \int_0^1 |f(e_1(t)) - f(e_2(t))| \, dt.$$

As a global discrepancy measure, we report the average discrepancy of all the seam edges in a 3D model

$$D(S) = \frac{\sum_{e \in S} \|e\| \cdot D(e)}{\sum_{e \in S} \|e\|},$$

where S denotes the set of seam edges, and each seam edge contribution is weighted by its length $\|e\|$ to capture the visibility and severity of the resulting artifacts independently from the mesh subdivision. In our implementation, edge discrepancies are estimated using discrete samples.

4.3. Defects

Another defining characteristic of datasets of RWTT is the common occurrence of defects and inconsistencies of specific nature. These issues are not usually considered when geometry processing algorithms are developed on paper, which often assume to be receiving sanitized input. Different issues can be encountered in the geometric part, or the parametrization part.

Defects of geometry include extremely irregular triangulations, zero-area triangles, topologically degenerate triangles (referring the same vertex more than once), and unreferenced or duplicated primitives. Topological defects include errors such as local loss of two-manifoldness, holes, and incomplete meshes. Another potential source of issues is the presence of "topological noise", i.e., small handles unrelated to the ideal topology of the reconstructed 3D shape (typically generated during the surface reconstruction phase from noisy dense point clouds), which is revealed by incongruous genus number with respect to the overall model shape.
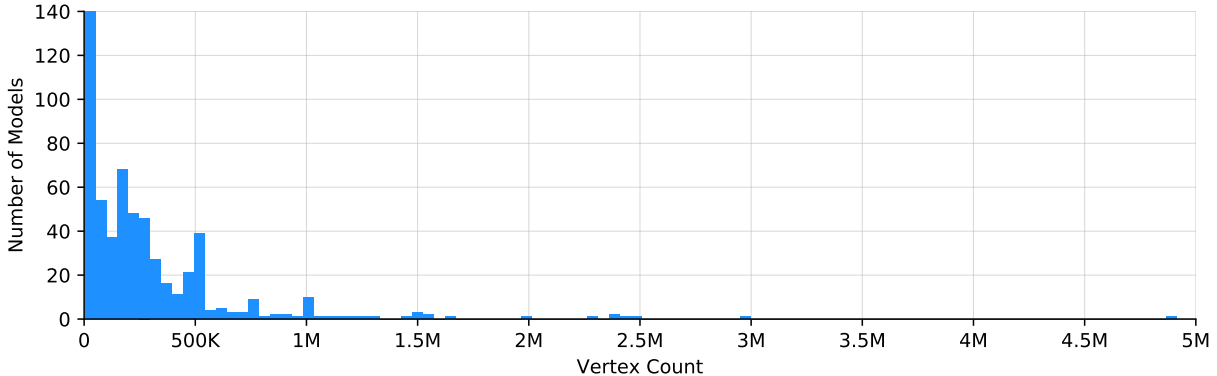
Figure 5: Geometric complexity of the 3D models found in our dataset. Models generated from 3D reconstruction technology are typically stored at high resolution to maximize accuracy and fidelity to their real-world counterpart.

Defects of the UV map include lack of injectivity caused by UV overlaps, which result in the same texture data being erroneously mapped to multiple parts of the mesh. Overlaps in the UV map can be either *local*, when triangles flip in UV space and change orientation relative to their neighbors, or *global*, when the boundary of one or several charts intersect in UV space. While they cause the same problem, it is useful to distinguish between local and global overlaps, because the strategies available to counter them are typically different. Additionally, several datasets feature only partial maps, that is, UV maps that do not map a model surface in its entirety, likely due to unavailable or insufficient photographic data.

5. Measurements of RWTT Models: an Analysis

We extracted our set of measures from each model of RWTT. In this section, we offer an analysis of the results, which serves as a picture of the merits and limitations of the currently available off-the-shelf photo-reconstruction software.

5.1. Measures on Geometry

Real-World Textured Things features models of many different sizes, ranging from low-poly models made of only a few thousand primitives to high-resolution ones with millions of vertices and faces. The histogram of the geometric complexity of the dataset, expressed as the number of vertices, is reported in Fig. 5. On average, our models tend to be fairly large compared to those found in similar datasets: as an example, 37% of the RWTT models have more than 250K vertices compared to less than 2% of the models found in Thingi10K (Zhou and Jacobson, 2016). This is expected, as objects are typically reconstructed at high resolution and then optimized for efficient use in downstream applications if needed.

5.2. Texture Maps and Parametrizations

Texture Size. Modern photo-reconstruction pipelines are able to generate accurate digital reproductions from hundreds or thousands of high-resolution images. Consequently, the majority of textures found in RWTT are high-resolution, ranging from 4K upwards. Moreover, several models feature multiple or non-square textures.

Texture Space Occupancy. In Fig. 6 we plot a histogram of texture occupancy values found in RWTT, clearly showing how atlas packing is often sub-par, leading to unnecessarily large memory requirements for texture signals sampled at relatively lower densities. This is particularly significant as GPU memory is often a critical bottleneck in graphics applications.
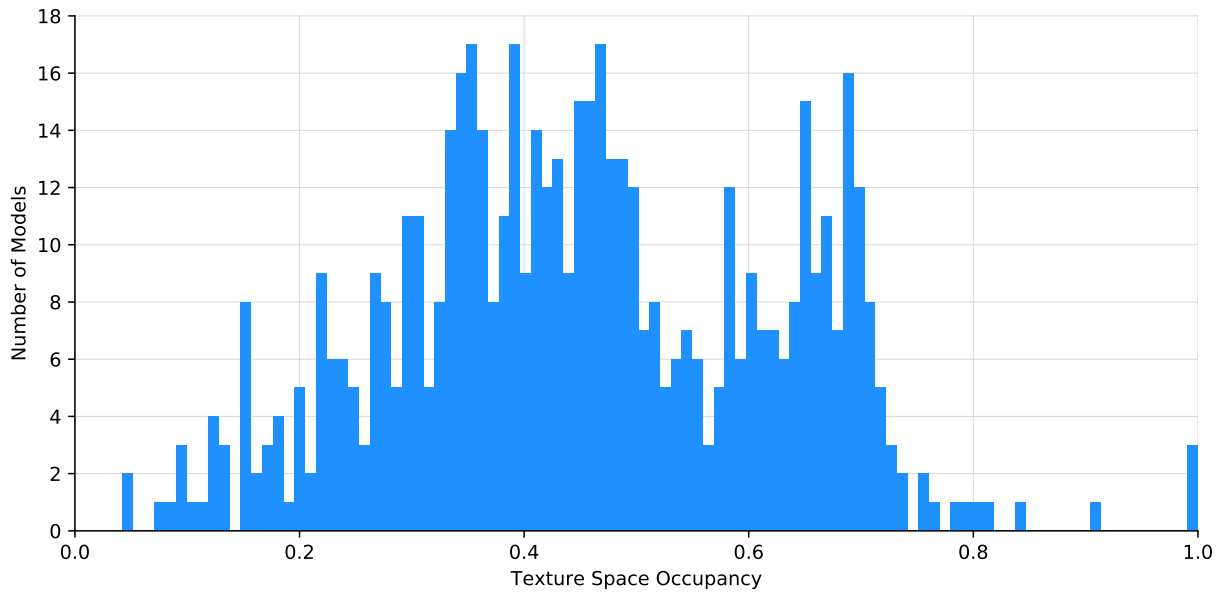
Figure 6: Distribution of the RWTT textures occupancy. Many textures are not efficiently packed, leading to a waste of GPU memory when the models are rendered in real-time applications.
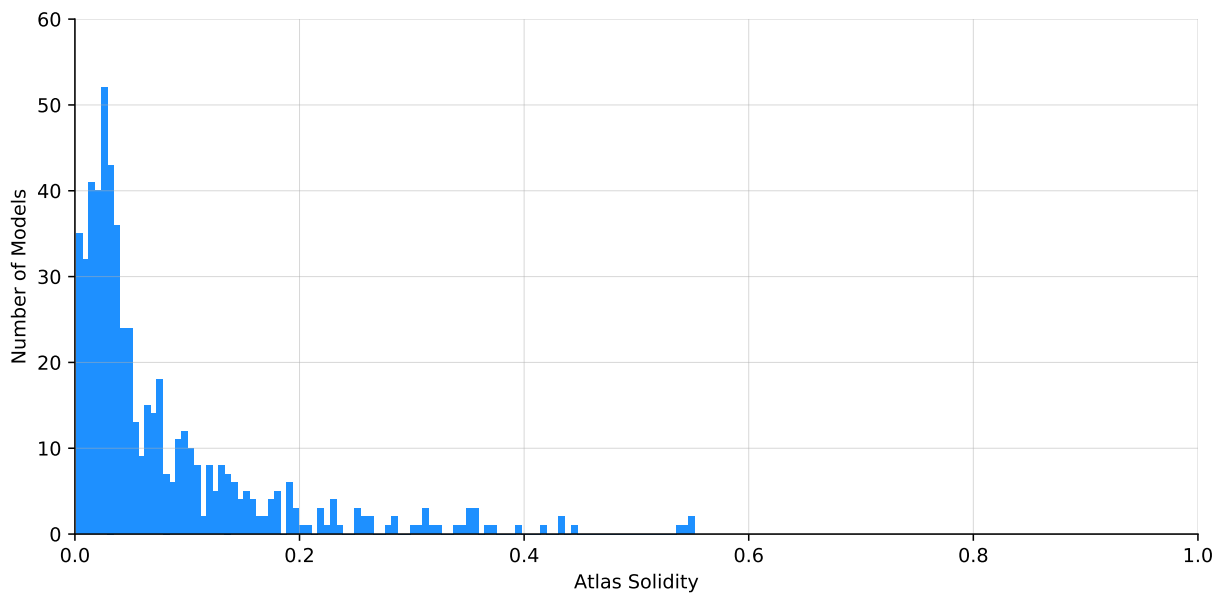

Figure 7: Distribution of atlas Solidity for models featured in the RWTT dataset. Many atlases exhibit high-to-extreme fragmentation: the model with the highest chart count has 555252 charts.
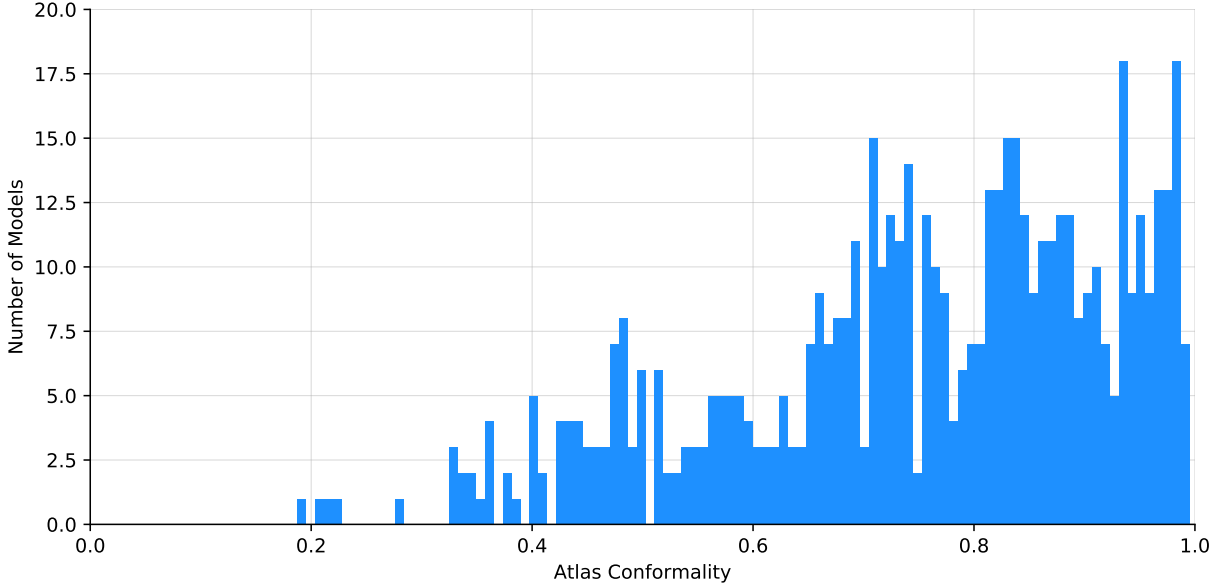
Figure 8: Distribution of atlas Conformality, where a lower value implies larger angle deviations. Despite many fine-grain chart decompositions, many parametrizations are still affected by non-negligible conformal distortion.

Atlas Solidity. We report the atlas solidity distribution over our dataset in Fig. 7. The plot highlights a bias towards low solidity values, indicating the frequent occurrence of texture maps affected by the presence of large amounts of seams.

The surprising fact is, however, that the degree of fragmentation witnessed in the majority of cases is almost never justified by the topological and geometrical complexity of the models we have collected (see Fig. 3). We conjecture that this happens because of two distinct causes. First, this happens because the textures are synthesized using large numbers of photographs, with reconstruction tools optimizing for color fidelity and attempting to store as much information as possible from the available images, disregarding the creation of many atlas charts. Second, this may also happen because of intrinsic limitations in existing automatic UV mapping technology, which we investigate in Section 6.

Conformality and Sampling Frequency. Fig. 8 reports the distribution of average quasi-conformal distortion values for our dataset. Unsurprisingly, most of the texture maps exhibit good angle preservation, which is to be expected given the high degree of fragmentation affecting many of the atlases.

A scatter plot of the 1st, 50th and 99th percentile values of the models UV scaling, sorted by range from P_1 to P_{99} is reported in Fig. 9, showing how a significant amount of models exhibit a non-negligible spread in the UV scaling values and, consequently, non-uniform texel density across the surface. As already observed in Section 4.2.3, this is quite peculiar but also not particularly surprising, as models with less uniform sampling frequency distributions were likely generated from photographs that captured surface details at different scales (see for example Fig. 4).

Seam Color Discrepancy. The distribution of average color discrepancies along texture seams is reported in Fig. 10. Overall, our dataset report fairly heterogeneous scores. There are two likely explanations for this. First, there can be differences in the uniformity of the object colors, whereby models with more varied colors are likely to exhibit more severe artifacts at texture seams. Second, there can be differences in the way texture seams are handled by the reconstruction tools during the texturing process: some tools may attempt to generate textures that are smoother across seams by optimizing for consistency in the texture data whereas others may prioritize fidelity to the photographic data.
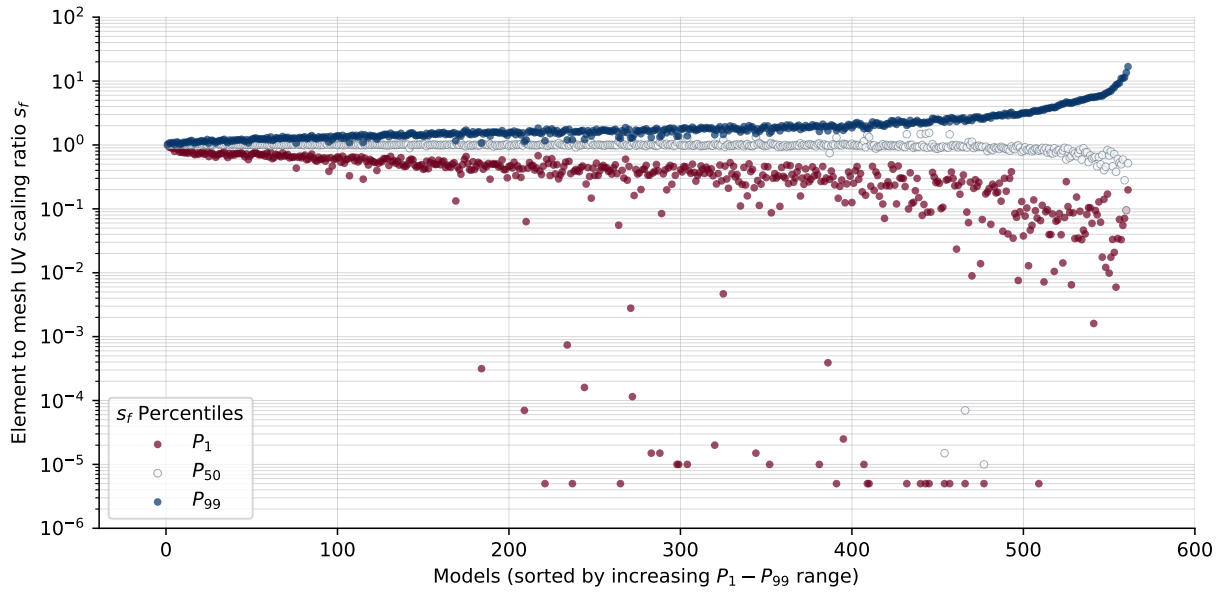
13

Figure 9: Visualization of the UV scaling ranges present in the dataset. We plot the 1st, 50th and 99th percentiles for the models sorted by UV scaling range width, highlighting how a significant portion of the dataset exhibits large variances in the mapping resolution.
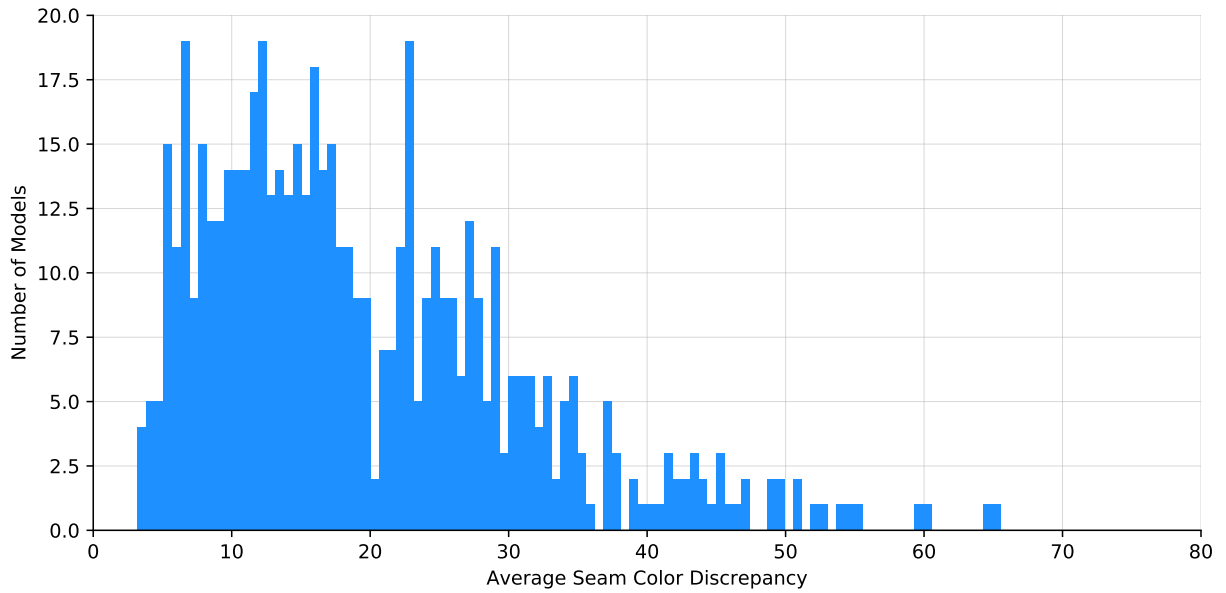

Figure 10: Distribution of average seam color discrepancy (measured in RGB space).
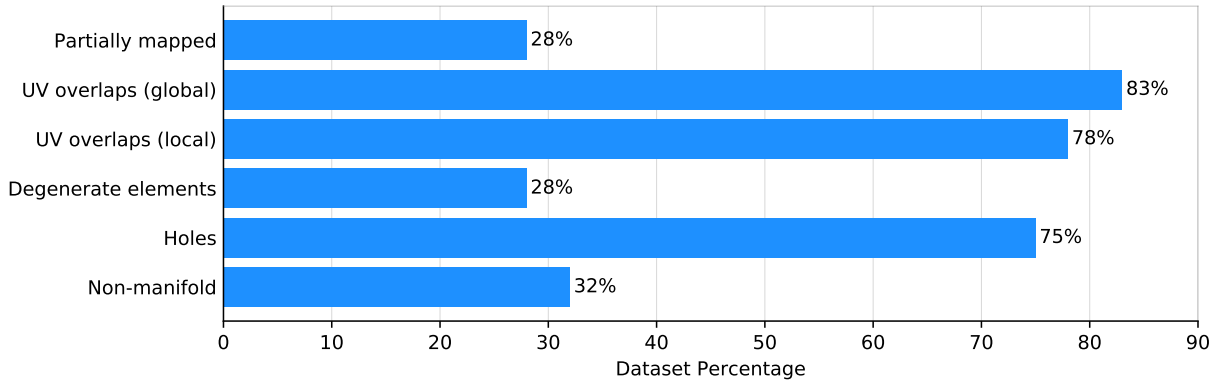
Figure 11: Common defects found in Real-World Textured Things models.

5.3. Defects

The ability to deal with arbitrary input meshes without requiring preemptive cleaning is a key property of any robust implementation, whose importance has been increasingly recognized as testified by the emergence of more and more "real-world" datasets, and algorithms specifically designed to deal with them. RWTT is a further exemplification of this, and features meshes with all kinds of meshing and geometric defects, as well as defects related to texture mapping, such as local and global overlap of UV coordinates, and models that are only partially textured. An overview of the situation is reported in Fig. 11.

6. Recomputing UV Maps

Given the number of issues these UV maps exhibit, an obvious question is whether or not they are peculiar to the tools that generated the textures and parametrizations as part of the 3D reconstruction process, or if there is an intrinsic limitation in the current technology regarding the automatic UV mapping of complex and high-resolution models. Therefore, we have decided to evaluate the performance of existing state of the art UV mapping tools and see if they can produce maps of better quality than the ones found in our dataset.

We have performed experiments with commercial, open source and academic automatic UV mapping solutions. Maya (Autodesk, Inc, 2019a) is a commercial 3D modeling software that provides a simple automatic UV unwrapping procedure by projecting 3D faces to a fixed set of orthogonal planes. Blender (Blender Foundation, 2019) also offers an automatic projection-based UV unwrapping functionality, but the set of projection planes is not fixed: instead, charts are computed by grouping together faces with similar normal vectors, and the projection plane of each chart is the one orthogonal to the average normal. Finally, we also experimented with OptCuts (Li et al., 2018), which is a recent academic work that automatically computes UV maps by alternating steps in which the distortion of the mapping is reduced, and steps in which seams can be added to the atlas charts to further reduce distortion, or removed if they are unnecessary.

For our experiments, we decided to set a time limit on the overall running time of 20 minutes. Since these tools cannot be conveniently automated to produce multiple textures, we have restricted the experiment only to models with single textures in order to ensure fair comparisons with the input data. We also restricted the experiments to models comprised of a unique connected component as the available implementation of OptCuts fails to initialize multi-component models. Moreover, to meet the topological requirements of OptCuts and ensure that the input mesh is two-manifold, we implemented a simple preliminary cleaning step that removes non-manifold vertices and edges, unreferenced vertices, and degenerate geometry. Note that, since OptCuts is an iterative algorithm, we saved the result obtained after 20 minutes of running time, rather than simply interrupting the task, if the algorithm was unable to reach numerical convergence within the given time frame.
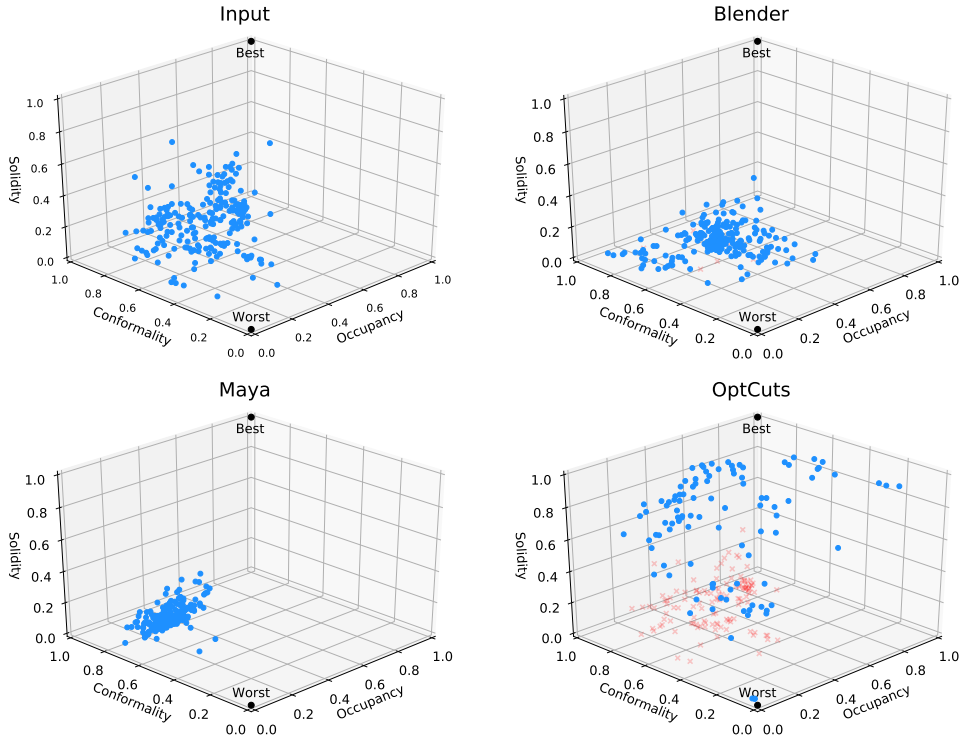
15

Figure 12: Performance of existing UV mapping tools on models generated with photo-reconstruction technology.

The results of this experiment are reported in Fig. 12 as a 3D scatter plot of the texture occupancy, mapping conformality, and atlas solidity of the input data and the outputs produced by the three aforementioned implementations, with failure cases marked in red. Ideally, the scatter plots of the outputs should move towards the top corner at coordinates (1, 1, 1). Blender performs worse in terms of atlas solidity compared to the input UV maps, without managing to improve under any of the remaining two metrics. Maya also generates heavily fragmented output atlases, but succeeds in significantly reducing the conformal distortion of the UV maps; note, however, that the results are affected by packing issues, as shown by the poor performance in terms of texture occupancy. OptCuts shows the best results despite being often interrupted before completing the atlas optimization, but is affected by a very high failure rate (54%). OptCuts failures are either due to numerical issues that prevent the correct initialization of the UV configuration to a bijective map, or to the inability to reduce the surface genus before starting. Finally, it is worth mentioning that, of the three implementations we tested, OptCuts is the only one that guarantees bijectivity of the output map; however, it is a much less efficient, practical and reliable solution.

Overall, our experimental evaluation shows that automatically computing UV maps for this category of 3D models is intrinsically challenging regardless of the method employed, and currently available methodologies are unable to consistently provide satisfactory results.

7. The TexMetro Tool

Alongside the dataset, we also release an open source command line tool – TexMetro – that can be used to compute all the measures described in Section 4 on arbitrary, user provided textured models. This tool can be used to assess and compare existing or future alternative photo-reconstruction tools (e.g. starting from the same set of images), as well as to assess the efficacy of existing or future methods aimed to improve the quality of 3D models.

The UV map, geometry, and connectivity of the input models are analyzed. The full set of statistics and measures are collected in a JSON file, while only a relevant subset of them is displayed on the standard output, consisting of salient information about the mesh geometry, its parameterization and texture data. A description of the JSON file structure can be found in the documentation from the TexMetro code repository.

A few of the measures are dependent on the texture resolution (in texels): to compute these, TexMetro retrieves the resolution of the texture images referenced by the input model or, simply, accepts a set of texture sizes passed as command-line arguments.

The TexMetro tool is open source and is implemented using C++, Qt for interfacing, and OpenGL to accelerate texture overlap detection (which is done by rasterizing UV triangles over off-screen buffers).

8. Conclusions

In this work, we introduced a dataset of textured 3D models generated with automatic software, carefully selected and gathered from publicly available sources, and enriched with relevant measures, including new ones tailored to highlight the defects of this increasingly important and still neglected class of 3D models; we also provided an open source tool to collect measures and statistics on the geometry, parametrization and texture maps of user-provided models.

The analysis of our dataset revealed interesting traits that characterize these models. In particular, it is extremely common to find heavily fragmented texture atlases; this could be attributed to the fact that these parametrizations are generated automatically and tailored to the available pictures. On the other hand, experimental evaluation of current state of the art automatic UV mapping tools, both commercial and academic, suggests that the issues that affect the UV maps of our dataset are not peculiar to the implementations of the software used to reconstruct the mesh; rather, this seems to be a limitation of current automatic UV mapping tools and algorithms that either fail to produce better results or are not practical and robust enough to reliably undertake the task.

Our aspiration is that this dataset and tool will help researchers and developers in their efforts to improve the quality of photo-reconstruction algorithms and implementations, as well as new tools targeted at post-processing the resulting models.

Acknowledgements

References

3Dflow SRL, 2014–2020. 3DF Zephyr: The complete photogrammetry solution. https://www.3dflow.net/3df-zephyr-photogrammetry-software/.

Agisoft LLC, 2010–2019. Metashape: photogrammetric processing of digital images and 3D spatial data generation. https://www.agisoft.com.

Attene, M., Campen, M., Kobbelt, L., 2013. Polygon mesh repairing: An application perspective. ACM Computing Surveys (CSUR) 45. doi:10.1145/2431211.2431214.

Autodesk, Inc, 2019a. Autodesk Maya 2019. https://www.autodesk.com/products/maya/overview.

Autodesk, Inc, 2019b. ReCap Pro: Reality capture & 3D scanning software. https://www.autodesk.com/products/recap/overview.

Bentley Systems, Inc, 2014–2019. ContextCapture: 3D reality modeling software. https://www.bentley.com/en/products/brands/contextcapture/.

Blender Foundation, 2019. Blender: Free and Open 3D Creation Software. URL: https://www.blender.org.

Capturing Reality, 2014–2019. RealityCapture: Mapping and 3D modeling photogrammetry software. https://www.capturingreality.com/Products/.

Chang, A.X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., Xiao, J., Yi, L., Yu, F., 2015. ShapeNet: An Information-Rich 3D Model Repository. Technical Report arXiv:1512.03012 [cs.GR]. Stanford University — Princeton University — Toyota Technological Institute at Chicago.

Choi, S., Zhou, Q.Y., Miller, S., Koltun, V., 2016. A large dataset of object scans. arXiv:1602.02481.

Chu, L., Pan, H., Liu, Y., Wang, W., 2019. Repairing man-made meshes via visual driven global optimization with minimum intrusion. ACM Transactions on Graphics (TOG) 38. doi:10.1145/3355089.3356507.

Cignoni, P., Ganovelli, F., Gobbetti, E., Marton, F., Ponchio, F., Scopigno, R., 2005. Batched multi triangulation, in: Proceedings IEEE Visualization, IEEE Computer Society Press, Conference held in Minneapolis, MI, USA. pp. 207–214. URL: http://vcg.isti.cnr.it/Publications/2005/CGGMPS05.

Cignoni, P., Rocchini, C., Scopigno, R., 1998. Metro: Measuring error on simplified surfaces. Computer Graphics Forum 17, 167–174. doi:10.1111/1467-8659.00236.

Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T., Nießner, M., 2017. ScanNet: Richly-annotated 3D reconstructions of indoor scenes, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

Epic Games, 2004 – 2019. Unreal engine 4 marketplace. https://www.unrealengine.com/marketplace/.

Firman, M., 2016. RGBD datasets: Past, present and future, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops.

Floater, M.S., Hormann, K., 2005. Surface parameterization: a tutorial and survey, in: Dodgson, N.A., Floater, M.S., Sabin, M.A. (Eds.), Advances in Multiresolution for Geometric Modelling, Springer, Berlin, Heidelberg. pp. 157–186. doi:10.1007/3-540-26808-1_9.

Google Scholar, 2019. "Metro: Measuring error on simplified surfaces" (Cignoni et al., 1998) citation page. https://scholar.google.it/scholar?cites=4029217030792744566.

Hartley, R., Zisserman, A., 2004. Multiple View Geometry in Computer Vision. 2 ed., Cambridge University Press. doi:10.1017/CBO9780511811685.

Hua, B., Pham, Q., Nguyen, D.T., Tran, M., Yu, L., Yeung, S., 2016. SceneNN: A scene meshes dataset with annotations, in: Fourth International Conference on 3D Vision (3DV), IEEE Computer Society, Conference held in Stanford, CA, USA. pp. 92–101. doi:10.1109/3DV.2016.18.

itSeez3D, 2019. itSeez3D: #1 mobile 3D scanner app for ipad. https://itseez3d.com/.

Jiang, Z., Schaefer, S., Panozzo, D., 2017. Simplicial complex augmentation framework for bijective maps. ACM Transactions on Graphics (TOG) 36. doi:10.1145/3130800.3130895.

Ju, T., 2004. Robust repair of polygonal models. ACM Transactions on Graphics (TOG) 23, 888–895. doi:10.1145/1015706.1015815.

Kharevych, L., Springborn, B., Schröder, P., 2006. Discrete conformal mappings via circle patterns. ACM Transactions on Graphics (TOG) 25, 412–438. doi:10.1145/1138450.1138461.

Koch, S., Matveev, A., Jiang, Z., Williams, F., Artemov, A., Burnaev, E., Alexa, M., Zorin, D., Panozzo, D., 2019. ABC: A big cad model dataset for geometric deep learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).

Levoy, M., Gerth, J., Curless, B., Pull, K., 2005. The stanford 3D scanning repository. https://graphics.stanford.edu/data/3Dscanrep/.

Levoy, M., Pulli, K., Curless, B., Rusinkiewicz, S., Koller, D., Pereira, L., Ginzton, M., Anderson, S., Davis, J., Ginsberg, J., Shade, J., Fulk, D., 2000. The digital Michelangelo project: 3D scanning of large statues, in: Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques, ACM Press/Addison-Wesley Publishing Co., USA. pp. 131–144. doi:10.1145/344779.344849.

Li, M., Kaufman, D.M., Kim, V.G., Solomon, J., Sheffer, A., 2018. OptCuts: Joint optimization of surface cuts and parameterization. ACM Transactions on Graphics (TOG) 37. doi:10.1145/3272127.3275042.

Liu, L., Zhang, L., Xu, Y., Gotsman, C., Gortler, S.J., 2008. A local/global approach to mesh parameterization. Computer Graphics Forum 27, 1495–1504. doi:10.1111/j.1467-8659.2008.01290.x.

Liu, S., Ferguson, Z., Jacobson, A., Gingold, Y., 2017. Seamless: Seam erasure and seam-aware decoupling of shape from mesh resolution. ACM Transactions on Graphics (TOG) 36. doi:10.1145/3130800.3130897.

Moulon, P., Monasse, P., Perrot, R., Marlet, R., 2016. OpenMVG: Open multiple view geometry, in: International Workshop on Reproducible Research in Pattern Recognition, Springer. pp. 60–74.

Pix4D SA, 2011–2019. Pix4Dmapper: Professional photogrammetry software for drone mapping. https://www.pix4d.com/product/pix4dmapper-photogrammetry-software.

Poranne, R., Tarini, M., Huber, S., Panozzo, D., Sorkine-Hornung, O., 2017. Autocuts: Simultaneous distortion and cut optimization for UV mapping. ACM Transactions on Graphics (TOG) 36. doi:10.1145/3130800.3130845.

Ranzuglia, G., Callieri, M., Dellepiane, M., Cignoni, P., Scopigno, R., 2013. Meshlab as a complete tool for the integration of photos and color with high resolution 3D geometry data, in: CAA 2012 Conference Proceedings, Pallas Publications - Amsterdam University Press (AUP). pp. 406–416. URL: http://vcg.isti.cnr.it/Publications/2013/RCDCS13.

Remondino, F., Spera, M.G., Nocerino, E., Menna, F., Nex, F., 2014. State of the art in high density image matching. The Photogrammetric Record 29, 144–166.

Rüther, H., Schröder, R., Bhurtha, R., Wessels, S., McDonald, B., 2020. Kua ruins heritage spatial documentation metadata dataset. URL: https://zivahub.uct.ac.za/articles/Kua_Ruins_Heritage_Spatial_Documentation_Metadata_Dataset/11770374, doi:10.25375/uct.11770374.v1.

Seitz, S.M., Curless, B., Diebel, J., Scharstein, D., Szeliski, R., 2006. A comparison and evaluation of multi-view stereo reconstruction algorithms, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE Computer Society, Washington, DC, USA. pp. 519–528. doi:10.1109/CVPR.2006.19.

Shilane, P., Min, P., Kazhdan, M., Funkhouser, T., 2004. The Princeton shape benchmark, in: Proceedings of the Shape Modeling International (SMI 2004), IEEE Computer Society, Washington, DC, USA. pp. 167–178.

Sketchfab, 2011–2019. Sketchfab: Your 3D content on web, mobile, AR, and VR. https://sketchfab.com/.

Smith, J., Schaefer, S., 2015. Bijective parameterization with free boundaries. ACM Transactions on Graphics (TOG) 34.

doi:10.1145/2766947.

Smithsonian Institution, 2013–2019. Smithsonian 3D Program. https://3d.si.edu/.

TurboSquid, 2019. TurboSquid: 3D models for professionals. https://www.turbosquid.com/.

Turk, G., Mullins, B., 2003. Large geometric models archive. URL: https://www.cc.gatech.edu/projects/large_models/.

Unity technologies, 2019. Unity asset store. https://assetstore.unity.com/.

Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., Xiao, J., 2015. 3D ShapeNets: A deep representation for volumetric shapes, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1912–1920.

Yuksel, C., Lefebvre, S., Tarini, M., 2019. Rethinking texture mapping. Computer Graphics Forum 38, 535–551. doi:10.1111/cgf.13656.

Zhou, Q., Jacobson, A., 2016. Thingi10k: A dataset of 10,000 3D-printing models. arXiv:1605.04797.