

COMPUTER GRAPHICS MEETS IMAGE FUSION: THE POWER OF TEXTURE BAKING TO SIMULTANEOUSLY VISUALISE 3D SURFACE FEATURES AND COLOUR

Geert J. Verhoeven ^a

^a Ludwig Boltzmann Institute for Archaeological Prospection & Virtual Archaeology, Franz-Klein-Gasse 1, 1190 Vienna, Austria
Geert.Verhoeven@archpro.lbg.ac.at

Commission II

KEY WORDS: 3D surface model, Image-based modelling, Image fusion, Normal map, Surface morphology, Texture baking

ABSTRACT:

Since a few years, structure-from-motion and multi-view stereo pipelines have become omnipresent in the cultural heritage domain. The fact that such Image-Based Modelling (IBM) approaches are capable of providing a photo-realistic texture along the three-dimensional (3D) digital surface geometry is often considered a unique selling point, certainly for those cases that aim for a visually pleasing result. However, this texture can very often also obscure the underlying geometrical details of the surface, making it very hard to assess the morphological features of the digitised artefact or scene. Instead of constantly switching between the textured and untextured version of the 3D surface model, this paper presents a new method to generate a morphology-enhanced colour texture for the 3D polymesh. The presented approach tries to overcome this switching between objects visualisations by fusing the original colour texture data with a specific depiction of the surface normals. Whether applied to the original 3D surface model or a low-resolution derivative, this newly generated texture does not solely convey the colours in a proper way but also enhances the small- and large-scale spatial and morphological features that are hard or impossible to perceive in the original textured model. In addition, the technique is very useful for low-end 3D viewers, since no additional memory and computing capacity are needed to convey relief details properly. Apart from simple visualisation purposes, the textured 3D models are now also better suited for on-surface interpretative mapping and the generation of line drawings.

1. INTRODUCTION

Since about a decade, Image-Based Modelling (IBM) workflows that rely on the computer vision algorithms Structure-from-Motion (SfM) and dense Multi-View Stereo (MVS) have become omnipresent in all possible fields of research: from medical sciences to the whole variety of geosciences. Also in the cultural heritage domain, many users consider such IBM-pipelines as an ideal means to yield visually-pleasing, photo-realistic – and in some cases accurate – digital three-dimensional (3D) meshed models of their artefacts, excavations, buildings and even entire landscapes. The ability to provide a photorealistic texture (also called ‘diffuse texture map’ or ‘diffuse texture’) along the 3D surface geometry is very often also one of the main reasons to favour IBM over an active approach such as laser scanning.

In many instances, however, this texture can obscure the underlying geometrical details, making it hard to assess the morphological features of the 3D surface. Therefore, several applications require switching between the untextured and textured version of the 3D model, depending on the purposes of the interpretational process. To generate line drawings or study geometrical details, an object’s digital 3D model is commonly visualised without any colour to convey the overall shape and individual morphological details better. Compare, for example, Figure 1A and 1B and notice how the texture obscures morphological details like the stamped marks on the pot’s shoulder or the clear undulations on the inner part of its base. When a visually pleasing depiction or the object’s visible reflectance is of importance (e.g. when studying the various building materials of a structure or the different layers in an archaeological profile), the colour texture is again enabled.

This paper presents a new method to generate a detailed, morphology-relevant diffuse texture of a 3D model (Figure 1). The presented approach tries to overcome this switching between object visualisations by fusing the original diffuse colour data with a particular representation of the object’s morphology to generate a final texture that conveys the object’s geometrical features along with its visible spectral reflectance (i.e. colour). The texture is also very suited for conveying spatial detail in low-end viewers (e.g. 3D PDFs) because these programs do not offer advanced rendering capabilities nor can they handle densely meshed geometries.



Figure 1. A triangular mesh with (A) and without (B) its default diffuse texture. The new texture (C) conveys the pot’s colour along its morphological features in a quasi-photorealistic way.

2. METHOD

In general, the proposed method principally follows the default Computer Graphics (CG) workflow for creating photorealistic game or movie assets. A highly detailed mesh/polymesh featuring millions of polygonal facets (therefore generally denoted ‘high-poly’ model) is retopologised into a so-called ‘low-poly’ mesh. The latter is then UV unwrapped so that a ‘texture baking’ process can transfer the details of the high-poly model to the low-poly mesh (Chopine, 2011). The next sections will detail all these steps and the main considerations that are involved. Since the generation of the final texture necessitates extra steps and modifications to the CG pipeline mentioned before, they will be highlighted as well. The flowchart in Figure 2 depicts all the individual steps that are part of this workflow. Finally, it remains important to mention that the processing pipeline was conceived with cost-effectiveness, automation and software availability in mind. Although every step described in this article was executed within a specific software package (based on the criteria mentioned above), various alternative software solutions will be mentioned along the way to provide the necessary freedom in adopting this workflow. Bibliographic overkill is omitted by only referencing free software that results from academic research.

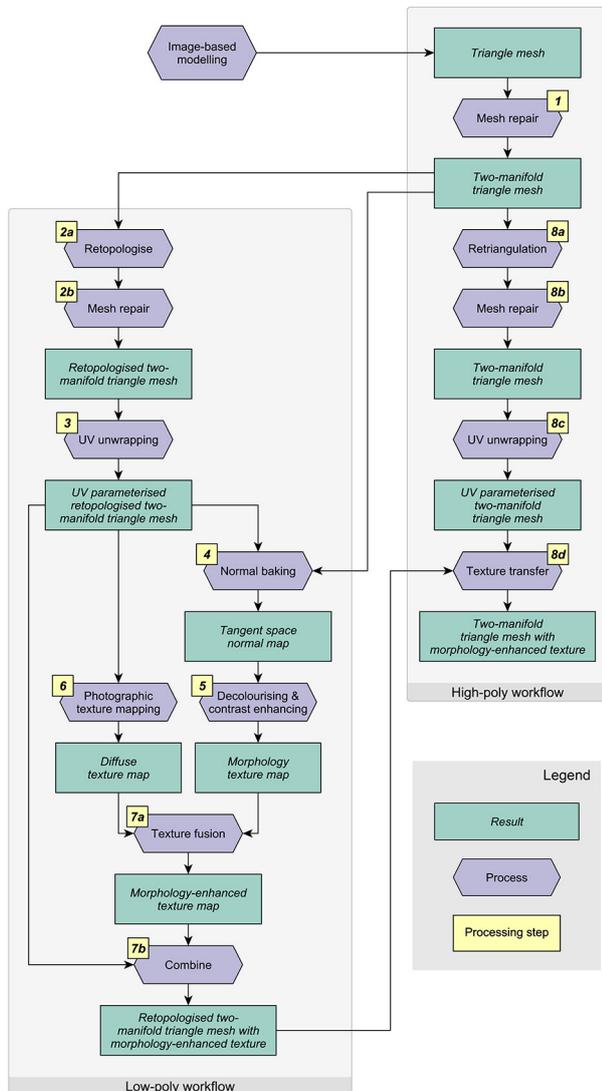


Figure 2. The complete processing workflow detailed in this paper. In the text, one finds more information about every step.

2.1 From initial IBM result to two-manifold polymesh

To start, IBM is used to compute a triangle-based polymesh surface model of an object. To date, several robust SfM-MVS-based packages can be employed to obtain an image-based 3D surface model in a (semi-) automated way. For this paper, Agisoft LLC PhotoScan Professional Edition (version 1.2.4 or higher) was used. Similar to PhotoScan, most IBM packages generate a continuous triangular polymesh (or simply mesh) from the unstructured dense point cloud that results from the dense image matching.

The quality of this 3D meshed surface plays, however, an important role in the efficiency and accuracy by which subsequent processing steps and visualisation can take place. Not only should it accurately represent the small morphological features of the real-world object, but the mesh must also satisfy particular geometric constraints. In most cases, this means that the mesh should be artefact-free, hole-free (called watertight when dealing with a closed mesh), have a uniform or adaptive face distribution and be two-manifold. Since meshes generated by IBM software typically suffer from various topological and geometrical errors such as holes, islands of facets, singular vertices, very complex edges, overlapping and intersection facets (Attene et al., 2013), some form of repair is essential.

Free packages such as MeshLab (Cignoni et al., 2008) and Blender, medium-cost software like Pixologic’s ZBrush and Pilgway’s 3D-Coat or very pricey solutions such as 3D Systems’ Geomagic Wrap can yield a hole-free two-manifold triangle mesh. However, two lesser-known programs should receive particular attention here. First, the free version of ReMake, a complete end-to-end mesh processing solution from software giant Autodesk, has the capability to handle and clean really large meshes. Second, the rather unknown but powerful open-source geometry processing software Graphite (ALICE-INRIA, 2015). Being a research platform for various 3D modelling and CG topics, Graphite has state-of-the-art mesh repair, remeshing and reconstruction algorithms.

2.2 Creating a low-poly mesh with clean topology

Although being artefact-free, watertight and two-manifold, the triangular polymesh is at this stage still characterised by a rather poor layout of all its individual facets. In addition, the mesh also features too many polygons for the remaining workflow steps. It is thus of the utmost importance to reduce the complexity of the mesh while maximising the mesh topology. In essence, two main approaches can address this issue: decimation and retopologising.

Mesh decimation is a procedure to simplify the 3D meshed surface through a repetitive, error-controlled reduction of the number of faces, edges and vertices present in the mesh (Gotsman et al., 2002; Botsch et al., 2010). In essence, decimation tools should preserve the 3D model’s essential morphology and keep texture coordinates (see later) or material boundaries intact. Almost all software programmes that allow for mesh decimation enable the user to specify the amount of reduction, via either facet count or a percentage. Atango Balancer nPro is one of the exceptions to this approach. This small and inexpensive application does not provide only an interactive decimation approach that is driven by the visual appearance of features in the meshed surface, Balancer nPro also allows for a metric tolerance-based simplification that works very accurately (Verhoeven, 2016).

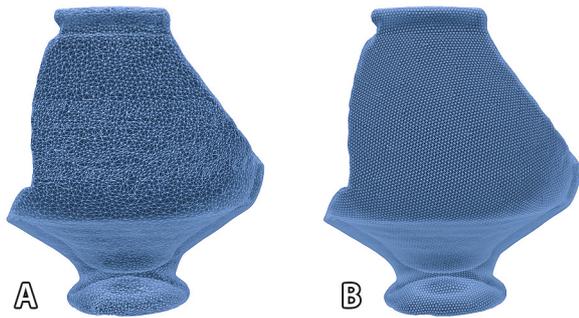


Figure 3. The same 3D surface model, decimated by PhotoScan (A) and retopologised by Instant Meshes (B) to 40k triangles.

In the CG industry, however, artists often resort to the approach of retopologising, or *retopo* for short. Topology refers to the manner in which the components that form a 3D model, be it NURBS or mesh facets, are connected and how their edges ‘flow’. In contrast to simply reducing the polygon count of the initial mesh as decimation does, retopologising creates an entirely new 3D polymesh surface from scratch. Although this new (i.e. retopologised) surface conforms to the original mesh, it has a much lower poly count but, more importantly, highly optimised, usually quad-dominant topology so that it lends itself much better to posing, animating and texturing. While mesh decimation and retopologising can thus both be considered mesh reduction approaches, retopologising tools have one major benefit over mesh decimation tools in that they produce an orderly, generally quadrilateral (‘quads’) -based polymesh. In contrast, most decimators will end up with sub-optimal topology and sometimes even messy, triangulated surfaces (Figure 3).

When retopologising a mesh, the aim is to convey the overall model’s morphology solely and not bother with the minute details from the original mesh, since the latter are transferred to the new, retopologised mesh by a process called ‘texture baking’ or simple ‘baking’. In essence, baking creates a normal, displacement or bump map that stores all the surface details of the initial, high-poly mesh. This map (or a combination of maps) is then applied to the retopologised, low-poly meshed model to create the illusion of high-resolution surface detail even though it is physically not there. In section 2.4, it will become clear how one aspect of this texture-baking approach – namely normal baking – will be exploited.

Although retopologising is mainly used to create usable assets for the film or game industries and a mere decimation might thus be sufficient for simple documentation or still rendering purposes, one should never underestimate the power of proper topology. Every preceding part of any 3D model pipeline (such as UV unwrapping and baking – see sections 2.3 and 2.4) will benefit from decent topology.

Automatic and purely manual retopologising tools exist. Although automatic retopologising is still inferior to purely time-consuming manual approaches, these (semi-) automatic approaches can be a real lifesaver when time and skill are an issue. Moreover, many automated tools enable the user to draw guiding curves on the surface to control the edge flow of the final mesh more accurately. Examples of excellent applications that can achieve this are ZBrush, 3D-Coat and The Foundry’s Modo. However, a real paradigm shift in auto-retopologising came about with Instant Meshes.

Introduced at SIGGRAPH ASIA in 2015, this free semi-automatic retopology program is nothing but revolutionary and currently unsurpassed when it comes to creating high-quality isotropic meshes out-of-the-box. It uniquely approaches remeshing by a unified local smoothing operator that optimises both the edge orientations and vertex positions in the output mesh (Jakob et al., 2015). In the software, this means that one has to compute first an orientation field to guide the placement of the faces and control their edge alignment. During a second phase, a position field is optimised so that it uniformly distributes the facets of the new polymesh while also snapping edges to sharp features. In addition, the software also offers edge flow tools (to alter both orientation and position of the output mesh facets manually) and the capability to work with triangular and quad-dominant polymeshes, point clouds and range maps. The end-result is always a high-quality triangulated or quadrangulated polymesh with a clean topology. The facets are naturally aligned and follow the surface curvature while snapping their edges to sharp model features. Moreover, Instant Meshes leverages the power of multiple processing cores, while it robustly scales to dense meshes featuring several hundred millions of polygons. In its current form, Instant meshes makes clean isotropic meshes that cannot have differing polygon densities, but this feature is planned for the near future. Finally, the user interface is very simple and tidy, which makes the software straightforward to use.

Figure 3 displays two triangular meshes of the same object, obtained by PhotoScan’s decimation and Instant Meshes’ retopologising, respectively. Notice the highly optimised polygonal flow of the latter. Since this workflow will leverage the power of normal map baking (explained in section 2.4), the low-poly retopologised mesh should adhere to a few rules. First, the polygon count should not be too low, as the low-poly mesh should have a silhouette that very closely matches the high-poly one. However, the model should also not feature too many polygons, as this will make the normal map less expressive (see Figure 7).

Since every mesh is morphologically different, it is not feasible to provide a rule-of-thumb. Usually, a poly-count of approximately 40 k is targeted, but thanks to the speed of Instant meshes, the final amount of polygons can be driven by the visual appearance of the mesh features. Although Instant Meshes can create pure quad-based meshes and these might have advantages in UV unwrapping (next section), it is advised to have Instant Meshes compute a triangular mesh for two reasons. First, this will omit possible inconsistent shading across various applications in a later stage (because different rendering engines triangulate meshes differently). Second, PhotoScan works only with triangular facets. Finally, and before going to the next step, it pays off to go through one more round of mesh repair because some small erroneous polygons might still be present in the retopologised mesh.

2.3 UV unwrapping

Before executing the important step of normal map baking (section 2.4), the mesh needs to be ‘UV unwrapped’. Not only will the low-poly, topology-optimised mesh be very suited for unwrapping, but this step is always needed since Instant Meshes destroys any existing UV parameterisation.

To understand UV unwrapping, one must realise that a textured 3D meshed model necessitates every vertex of that 3D surface

to have texture coordinates u and v associated. These so-called UV texture coordinates (abbreviated “UVs” in CG lingo) map every vertex of the 3D model to a specific location in the texture map. Since the latter is a 2D image, two coordinates suffice to describe every position in that image. Given that X , Y , and Z are already used to indicate the Cartesian axes of the 3D object in model space, the letters U and V were chosen. The process in which polygon meshes have their UV coordinates assigned is known as surface parameterisation or UV mapping (Paquette, 2013; Ganovelli et al., 2015).

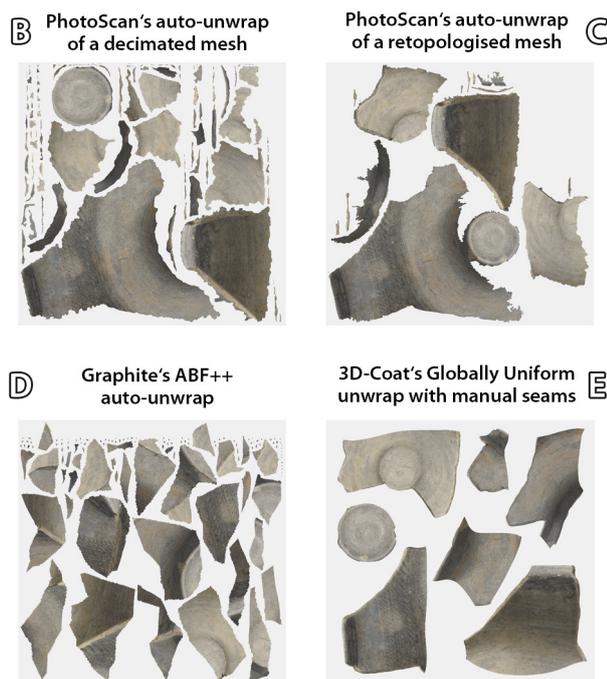
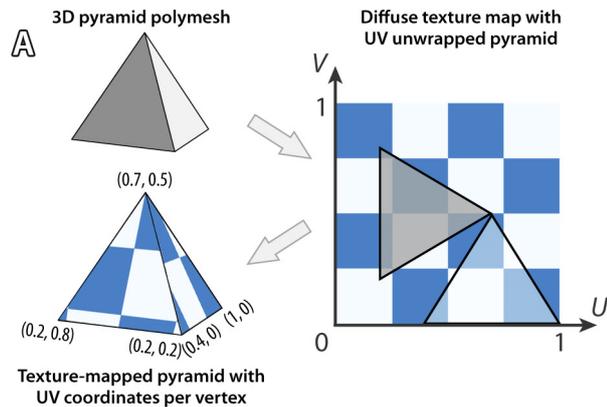


Figure 4. (A) illustrates the process of UV mapping a pyramid. (B) and (C) are the texture maps that result from PhotoScan’s ‘Generic’ parameterisation of the mesh in Figure 2A and 2B, respectively. (D) displays the texture atlas obtained in Graphite using the ABF++ unfolding algorithm (Sheffer et al., 2005), while (E) shows the result of 3D-Coat’s proprietary Globally Uniform unwrapping after manually placing a few seams. Both (D) and (E) were executed on the retopologised mesh.

In essence, UV mapping takes every discrete triangular polygon facet of the mesh and lays them all (cleverly or not-so-cleverly) out on the 2D texture map (Figure 4A). Analogous to

(un)wrapping a present, this process is commonly denoted unfolding or (UV) unwrapping. Obtaining clean and efficient automatic mesh UV parameterisation is a kind of Holy Grail in CG. The literature offers many approaches (consult Floater and Hormann (2005) or Sheffer et al. (2006)) for in-depth surveys), but there is still no silver bullet. Professional modellers will therefore typically choose a manual unwrapping solution, carefully splitting the geometry at well-chosen seams to keep certain polygons intelligently grouped together and end up with sensible UVs (Figure 4E). More specifically, the position of these seams is critical. Ideally, they are invisible. CG artists typically position them on an edge or cut of the model, such as in the joint between two flat surfaces or on the seam of a pair of trousers, or in locations where a viewer will never turn his/her attention to (like in a person’s armpit or under a hat).

Moreover, manual unwrapping is capable of achieving a more uniform sampling in texture space and wasting less UV map texels (i.e. texture pixels) compared to any automated method (compare Figure 4E with Figures 4B-D). All this makes sense for various reasons. First, clean and intuitive UV layouts are critical when having to paint on them in an image editor such as Adobe’s Photoshop. Second, optimal texture layouts are crucial to minimise any stretching, incorrect scaling or overlap effects that might reduce the final visual quality of the textured model (Angel and Shreiner, 2015).

To obtain these results, a variety of applications that specialise in UV mapping exist. Although Blender, Meshlab and Graphite offer unwrapping functionality, the most dedicated tools are payware: UVMapper Professional, Modo, ZBrush and Autodesk’s 3ds Max and Maya (all differing in cost, capabilities and ease of operation). During the last decade, these packages have incorporated more automated UV-generating approaches, while still offering various levels of manual control to place the seams, adequately control the UV islands and manage the stretching or pinching of the individual facets.

Nevertheless, some CG pros already rely on fully automated solutions, even though the seams are often ending up in less-than-optimal (i.e. visible) places. The reason being that unwrapping a polymesh is not enjoyable for most people. Despite the added automation, obtaining excellent results can still be very tedious and thus time-consuming. Depending on the complexity of the model, it is a long journey of maintaining a minimally distorted texture map and acceptably hide the 3D model’s texture seams. Moreover, even the best unfolding will not hide any seam indefinitely.

Although the same arguments may hold for the less-seasoned 3D artist (such as scholars in geomatics or archaeology), the latter typically resort to the automatic solutions that are embedded in most IBM software because of the lack of expert knowledge on this topic. However, IBM packages typically yield rather poor results (Cipriani et al., 2014). In most cases, they produce shredded UVs with too many ‘islands’ in the (u, v) parameter space. Although a package such as Agisoft PhotoScan offers several texture “Mapping modes” in an attempt to optimally UV unwrap a surface, most parameterisations are still rather messy (Figures 4B and 4C). These sub-optimal UV maps notwithstanding, it still does the trick of mapping the texture onto the mesh. This workflow relies on 3D-Coat’s UV toolset for unwrapping. Besides being easy to use, affordable and fast, the combination of its proprietary Globally Uniform unfolding and manual seam placing outperforms many other approaches (Figure 4E).

2.4 Normal baking

CG artists working for movies and games often face a difficult balance. On the one hand, the 3D content must look as realistically as possible, while on the other hand, the amount of geometrical data should be as small as possible to enable easy handling and processing of the data (e.g. for animation purposes and rendering). In other words, the polygon count of a triangular or quad mesh should be kept as low as possible but still display all the small and big features of the highly detailed mesh that represents the object or scene. Texture baking is a technique that transfers ('bakes' in CG lingo) all the fine details from the highly detailed polymesh to a mesh featuring a much lower number of polygons (i.e. the low-poly model).

Baking tools support multiple map types. Their colour and shade information instruct the rendering engine how to properly render all the fine geometrical details and create the illusion of a very detailed surface, despite the use of a low-poly model. By creating the illusion of a highly realistic 3D content for a fraction of the computing power needed for ultra-detailed geometry, texture maps are thus masters of deception.

For this workflow, both the original high-poly and retopologised low-poly mesh are used to create a normal map. Similar to bump maps, 8-, 16- or 32-bit 2D normal maps are often used to create the illusion of depth and raised detail. However, whereas bump maps simply encode height information using greyscale values, normal maps use full-colour Red-Green-Blue (RGB) values to signify the orientation of the surface normals (i.e. the directions perpendicular to the polygonal face) (Chopine, 2011). As such, they represent three dimensions of depth: the red shades depict the height information for the X-axis, while green and blue shades describe the high and low values for the Y- and Z-axis respectively (Figure 5).

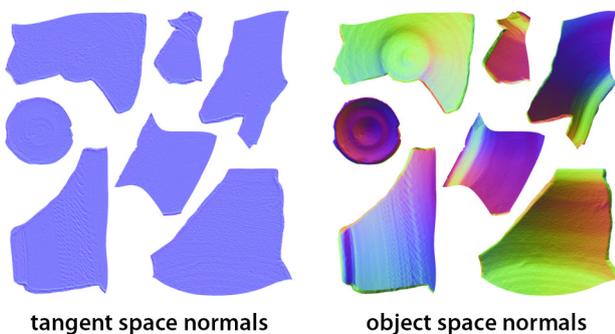


Figure 5. Tangent and object space normal maps.

When combining these surface normal data with the angle of the light source in the virtual environment, the light interaction on a given surface patch can be computed. Although the orientation of the faces in the low-poly mesh might be identical, the interaction of the light with the 3D orientation of the normal (extracted from every high-poly face) gives the low-poly model the illusion of morphological detail, as such enabling a highly realistic shading of the relief.

Although the free xNormal, which was long considered the industry standard when it came to baking, could be used for this purpose, the normal map baking step in this workflow is executed using Knald Technologies' Knald (Figure 6). This standalone and GPU (Graphics Processing Unit)-powered

software is specifically made for generating high-quality textures at incredible speeds. Besides, Knald also stands out from other packages by the possibilities it offers. The software can generate both conventional and bent tangent and object space normal maps with a maximum dimension of 8192 pixels by 8192 pixels, while various options can control the occlude weighting (cosine, uniform clamped and uniform unclamped). Moreover, Knald supports high-poly meshes up to 350 million triangles.

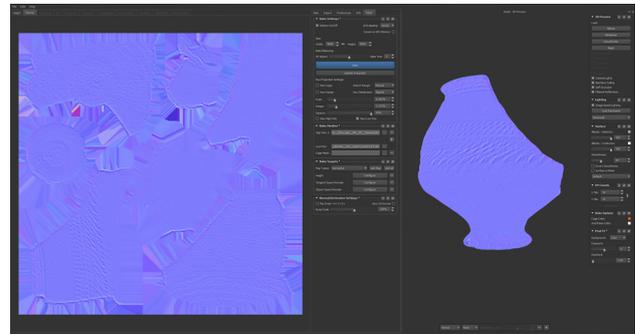


Figure 6. Tangent space normal map baking in Knald.

To obtain the final texture that encodes the shape and structural details of the object, Knald computes a tangent space normal map. Although tangent space normal maps shade the same as object space ones, they work (and hence look) dissimilar because they express the perturbation of the normal differently (Ganovelli et al., 2015). Object space normal maps are more rainbow-like (Figure 5B) and provide a slightly improved performance over their tangent space counterparts. However, since they store the object's normals using the same 3D coordinate reference system as the object's frame of reference, one cannot easily convert the resulting colour values to a greyscale map for use in the subsequent texture fusion step.

Tangent space normal maps define the normals relative to the surface of the low-poly mesh (and do not provide absolute normals like an object space normal map). So the normals encoded in a tangent space map compensate for the behaviour of the low-poly normals. This has two major consequences. First, conversion of their bluish-purplish colour (Figures 5-7) to a greyscale image that credibly represents the morphological details of the digital object is possible. Second, the amount of detail baked into the normal map depends on the spatial details of the low-poly mesh. A less detailed low-poly mesh will yield a larger deviation between its surface and the high-poly model's normals. For that reason, visual appearance should guide the retopologising step, since it will directly influence the amount of detail baked out in the normal map (Figure 7). Luckily, Knald also allows 'pushing' the amount of baked normal detail.

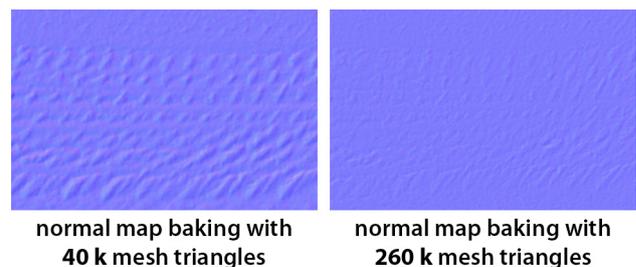


Figure 7. Tangent space normal map baking from low-poly meshes with different amounts of triangular facets.

2.5 Normal map decolourisation

Since the encoded RGB triplets of the normal map do not represent the real colour but the normal vector, the normal map has to be decolourised into a greyscale variant. However, using a standard luma Y_{709} decolourisation does not truly convey the morphological details captured in the original tangent space normal map.

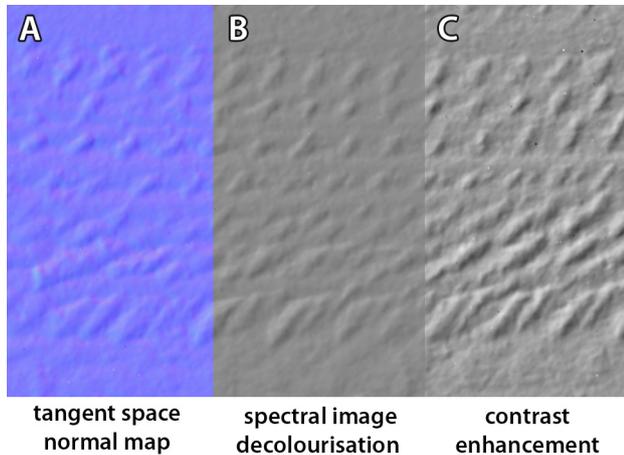


Figure 8. Encoding the object's shape and structural details (C) by enhancing the contrast of the decolourised (B) tangent space normal map (A).

Over the past decades, several simple and more complex colour-to-greyscale algorithms have been developed to derive the best possible decolourised version of a colour input image. To find out the optimal greyscale conversion algorithm, a collection of 57 different decolourisation methods which had previously been implemented in MATLAB (Verhoeven et al., 2015) were applied to various normal maps. After assessing all the results, it became evident that the spectral image decolourisation method proposed by Zhao and Tamimi (2010) was most capable of conveying the morphology captured in the bluish-purplish tangent space normal map (Figure 8B). After an additional contrast enhancement step, for which the specific amount depends on the spatial details one seeks to convey and the amount of normal 'pushing' performed by Knald, this image represents all the morphological details of the 3D surface model (Figure 8C).

2.6 Creating a textured 3D surface model

Before creating the final optimised texture by image fusion, it is essential to compute the standard photographic texture (often called 'diffuse map'). This is accomplished by importing the retopoligised mesh with its UVs into PhotoScan and choosing "Keep uv" as the texture mapping mode. Doing so prevents PhotoScan from re-unwrapping the model and generates a photographic texture map from the selected source photographs using the existing UV parameterisation.

2.7 Texture fusion

Finally, the fusion between the standard colour texture map and the greyscale-converted normal map can take place. Image fusion is a process in which the data from two or more images are combined in a specific way so that a single composite output image is generated that holds specific or substantial information from all input images.

In this case, the fused output should ideally be very close to the colour of the original texture, while also conveying the relief as the greyscale normal map visualises it. To find the fusion algorithm(s) that could best achieve this result, a previously programmed MATLAB toolbox TAIFU (Toolbox for Archaeological Image FUSion) was used (Verhoeven et al., 2016). After testing a wide variety of well-established blending modes and more recently developed multimodal image fusion algorithms, it became apparent that the 'Overlay' blending mode works relatively well when using the morphology texture as the top layer (Figure 9B). Depending on the dark tones in the diffuse texture and the amount of contrast enhancement applied to the decolourised normal map, changing the opacities/alpha values of the base and top layer can improve the effect needed. There is also no need for TAIFU, since any decent image processing software (such as GIMP, Pixelmator or Adobe's Photoshop) offers this blending option.

That said, slightly enhanced results are often provided when the morphology and diffuse textures are first fused using the Multi-Scale Transform (more specifically the Laplacian pyramid transform) and Sparse Representation (MST-SR) image fusion approach (Liu et al., 2015). In a second stage, this intermediate greyscale (Figure 9C1) output gets fused with the diffuse texture using the Overlay mode (Figure 9C2). As before, changing the opacities/alpha values of the input images might optimise the result (this is certainly the case when there are very dark tones in the diffuse texture such as in Figure 9A).

When compared to the product of the single 'overlay' blending, this two-stage fused image usually features more of the details present in the diffuse texture. As such, it is very efficient at obtaining the result sought after (Figure 9C2 and Figure 1C). Although TAIFU is not yet publically released, this approach is easy to mimic since the source code of the MST-SR fusion algorithm can be retrieved from Liu's website.

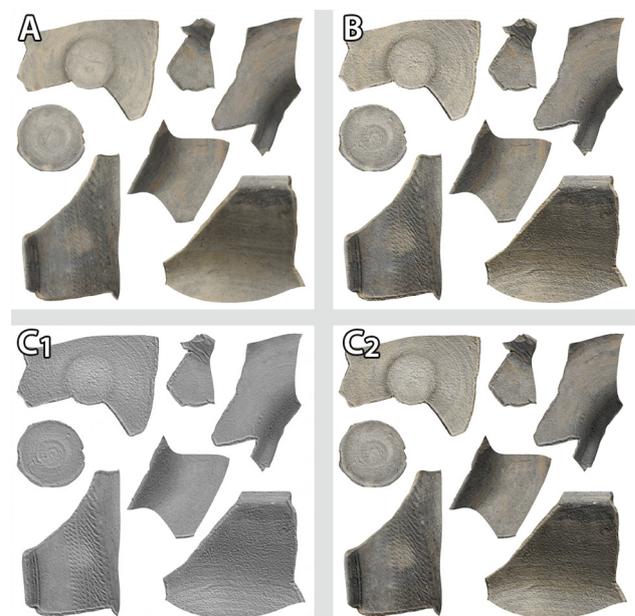


Figure 9. The default diffuse photographic texture (A) is turned into a morphology-enhanced version using the Overlay blending method (B) or using a combination of the Multi-Scale Laplacian Pyramid Transform and Sparse Representation image fusion (C1) followed by an Overlay blending step (C2).

2.8 From low-poly to high-poly

At this stage, this newly created, morphology-optimised texture is applied to the low-poly model inside PhotoScan or any CG application (Figure 1C). However, if one would like to have it displayed on the initial high-poly mesh, it is crucial to apply one last step since the UV parameterisation between both meshes differs. Given that most parameterisation applications cannot manage very dense polymeshes, the UV mapping of the high-poly mesh is best left to PhotoScan. Even though its parameterisation might be sub-optimal, it is less of a concern at this stage, because there is no more need to create a normal map (which is the step in which ill-placed seams are highlighted most). Nevertheless, it pays off to globally retriangulate the high-poly mesh to produce a more uniform mesh tessellation for PhotoScan to parameterise (compare Figure 4B to Figure 4C). Although xNormal along with major CG packages (such as The Foundry's Mari, Autodesk's Mudbox and Allegorithmic's Substance Painter) are capable of baking textures between entirely different UV sets, 3D-Coat also proved here to be a reliable and cost-effective option to transfer the new texture from the low- to the high-poly model.

3. RESULTS

Figure 1C shows the result of the proposed approach. As can be seen, this method fuses the normal with the original texture data to obtain an improved texture that conveys both colour and morphological features. When viewed in 3D, the textured model does not convey simply the colours in a proper way, but also enhances the small-and large-scale morphological features that are hard or impossible to perceive in the original textured model (Figure 1A).

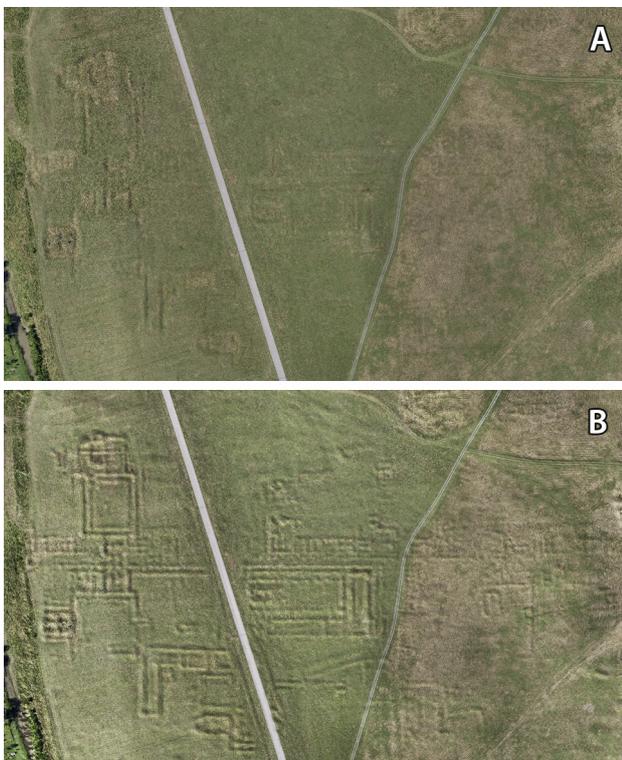


Figure 10. (A) shows the diffuse texture of a landscape surface model representing a part of *Bassiana*, a Roman city in present-day Serbia. Building traces (created by dug out foundations) are only clearly revealed in the morphology-enhanced texture (B).

Moreover, this approach is not applicable to artefacts alone. Figure 10 illustrates how even landscape-scale scenes can benefit from injecting spatial details into the diffuse texture. This scene represents a part of the Roman town of *Bassiana(e)*, located in present-day Serbia. Although one would expect the surface to be flat, it is rather undulating due to an intricate network of robber trenches spread over the site. These ditches originally contained the building foundations, but these stones were dug out for reuse. Although such earthworks can show up in aerial images when thrown into relief by low slanting sunlight (creating so-called shadow marks, some of which are visible on the left side of Figure 10A), the grass and the position of the sun at the time of photography masked the majority of them. The proposed processing pipeline was, however, capable of extracting the morphological landscape details from the meshed 3D surface model and encoding them into a new texture (Figure 10B).

4. DISCUSSION

Although other authors have previously used image fusion to merge both normal and texture information (Pan *et al.*, 2017), their method was limited to 2D views because they rendered a normal and an orthographic texture image from a chosen view direction (which means that the third dimension is lost). Furthermore, their normal map generation was very basic while the final, gradient domain fused output was based on the pseudo-coloured normal map with the texture details blended in (i.e. the result is non-photorealistic).

In this paper, the final texture uses the colour of the original diffuse texture map with the morphological features of the normal map blended in. This provides a more photorealistic result and does not rely on pseudo-colours that are entirely unrelated to the original texture. Also, the method outlined here uses the full, 3D surface geometry, while much flexibility is offered when generating the normal map data. To provide other researchers with the highest possible flexibility in adopting this workflow, every single step of this pipeline also came with a variety of dedicated software options.

Apart from simple visualisation purposes, the newly textured models are now also better suited for on-surface interpretative mapping by 3D vectorisation (a procedure which was explained in detail by Verhoeven (2016)). Since archaeology, or cultural heritage in general, still rely on line drawings to illustrate the general shape and detailed morphological features of an artefact, these newly fused textures can also be used as the basis for such (manual or automated) line drawings.

Although this workflow tried to be as generally applicable as possible, it is impossible to find a universal approach for every mesh and diffuse texture. Every 3D surface model is different and might require altered or sometimes additional steps. Mesh smoothing before normal generation or semantic mesh segmentation coupled with a tedious manual UV unwrapping are some of the steps that first come to mind. That said, the presented workflow is also not intended to be applied to all 3D content. Very often, the texture does not hide the morphological features of interest, or the colour of the diffuse texture is so dark that additional morphological shading remains unnoticeable. In other cases, the texture might be irrelevant, for instance when the polymesh exclusively serves to obtain metrical object information. In all of those cases, generating a morphology-enhanced diffuse texture would be a waste of time.

5. IMPROVEMENTS

Obviously, this workflow is not perfect and improvements are always possible. First, the normal map baking might generate artefacts at object edges. Although a higher number of facets in the low-poly model and better-placed UV seams might mitigate these effects, they often will remain visible to a certain extent simply because of the way in which tangent normal map baking works. Therefore, future research could focus on possibilities to incorporate the object space normal map.

Second, the accuracy of the diffuse colour data depends on many factors: the camera's spectral sensitivities, the white balance and exposure of the photographs, the illumination, the post-RAW colour processing steps as well as the anisotropic reflectance behaviour of most surfaces. In addition, the diffuse texture will also encode direct shadows and ambient occlusions. For longer outdoor image acquisitions, both shadowing effects might even vary. As such, it makes sense to focus on robust approaches for de-shadowing the diffuse texture map. Virtually recreating the lighting situation during image acquisition and, in combination with the meshed surface, use these data to render shadows that can be subtracted from the source photographs (Cipriani and Fantini, 2017) is an approach worth pursuing.

6. CONCLUSION

This paper presented a novel approach to enhance the default diffuse texture map of an (image-based) 3D model by fusing the object's photographic colour values with data that conveys its morphological features. Although it is possible to conceptualise future improvements, the currently achievable result is a quasi-photorealistic texture that simultaneously displays the object's geometrical surface features. Moreover, the fused texture is equally well suited for both high- and low-poly versions of the modelled object/scene. Finally, all suggested software packages are cost-effective and easy to learn, making this method flexible enough to implement in any existing mixed software pipeline.

REFERENCES

- ALICE-INRIA, 2015. Graphite, <http://alice.loria.fr/index.php/software/3-platform/22-graphite.html> (29-May-17).
- Angel, E., Shreiner, D., 2015. *Interactive computer graphics. A top-down approach with WebGL*, 7th edition. Pearson, Boston.
- Attene, M., Campen, M., Kobbelt, L., 2013. Polygon mesh repairing. *ACM Computing Surveys* 45 (2), article 15.
- Botsch, M., Kobbelt, L., Pauly, M., Alliez, P., Lévy, B., 2010. *Polygon Mesh Processing*. A K Peters, Natick.
- Chopine, A., 2011. *3D art essentials. The fundamentals of 3D modeling, texturing, and animation*. Focal Press, Burlington.
- Cignoni, P., Callieri, M., Corsini, M., Dellepiane, M., Ganovelli, F., Ranzuglia, G., 2008. MeshLab: an Open-Source Mesh Processing Tool. In: Scarano, V., Chiara, R. de, Erra, U. (Eds.), *Proceedings of the Sixth Eurographics Italian Chapter Conference*. Eurographics, Aire-la-Ville, pp. 129–137.
- Cipriani, L., Fantini, F., 2017. Digitalization culture vs archaeological visualization. Integration of pipelines and open issues. In: *The International Archives of the Photogrammetry, Remote Sensing and Spatial Inform. Sciences*, Nafplio, Greece, Vol. XLII-2/W3, pp. 195–202.
- Cipriani, L., Fantini, F., Bertacchi, S., 2014. 3D models mapping optimization through an integrated parameterization approach. Cases studies from Ravenna. In: *The Intern. Archives of the Photogrammetry, Remote Sensing and Spatial Inform. Sciences*, Riva del Garda, Italy, Vol. XL-5, pp. 173–180.
- 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.
- Ganovelli, F., Corsini, M., Pattanaik, S., Di Benedetto, M., 2015. *Introduction to computer graphics. A practical learning approach*. CRC Press, Boca Raton.
- Gotsman, C., Gumhold, S., Kobbelt, L., 2002. Simplification and Compression of 3D Meshes. In: Iske, A., Quak, E., Floater, M.S. (Eds.), *Tutorials on Multiresolution in Geometric Modelling. Summer School Lecture Notes*. Springer, Berlin, pp. 319–361.
- Jakob, W., Tarini, M., Panozzo, D., Sorkine-Hornung, O., 2015. Instant field-aligned meshes. *ACM Transactions on Graphics* 34 (6), article 189.
- Liu, Y., Liu, S., Wang, Z., 2015. A general framework for image fusion based on multi-scale transform and sparse representation. *Information Fusion*, 24, pp. 147–164.
- Pan, R., Tang, Z., Xu, S., Da, W., 2017. Normals and texture fusion for enhancing orthogonal projections of 3D models. *Journal of Cultural Heritage*, 23, pp. 33–39.
- Paquette, A., 2013. *An introduction computer graphics for artists*, 2nd edition. Springer, New York.
- Sheffer, A., Lévy, B., Mogilnitsky, M., Bogomyakov, A., 2005. ABF++. Fast and robust angle based flattening. *ACM Transactions on Graphics* 24 (2), pp. 311–330.
- Sheffer, A., Praun, E., Rose, K., 2006. Mesh Parameterization Methods and Their Applications. *Foundations and Trends® in Computer Graphics and Vision* 2 (2), pp. 105–171.
- Verhoeven, G.J., 2016b. Mesh Is More—Using All Geometric Dimensions for the Archaeological Analysis and Interpretative Mapping of 3D Surfaces. *Journal of Arch. Method and Theory*.
- Verhoeven, G.J., Karel, W., Štuhec, S., Doneus, M., Trinks, I., Pfeifer, N., 2015. Mind your grey tones – examining the influence of decolorization methods on interest point extraction and matching for architectural image-based modelling. In: *The International Archives of the Photogrammetry, Remote Sensing and Spatial Inform. Sciences*, Avila, Spain, Vol. XL-5/W4, pp. 307–314.
- Verhoeven, G.J., Nowak, M., Nowak, R., 2016. Pixel-level image fusion for archaeological interpretative mapping. In: Lerma, J.L., Cabrelles, M. (Eds.), *Proceedings of ARQUEOLÓGICA 2.0*. Valencia, Spain, 5-7 September 2016, Edit. Univ. Politècnica de València, Valencia, pp. 404–407.
- Zhao, Y., Tamimi, Z., 2010. Spectral image decolorization. In: Bebis, G., Boyle, R., Parvin, B., Koracin, D., Chung, R. (Eds.), *Proceedings of the 6th International Conference on Advances in Visual Computing (ISVC'10). Part II*. Springer, Berlin, pp. 747–756.