Pacific Graphics 2020
E. Eisemann, A. Jacobson, and F.-L Zhang
(Guest Editors)

Volume 39 (2020), Number 7

A Deep Residual Network for Geometric Decontouring

Zhongping Ji[†1], Chengqin Zhou[1], Qiankan Zhang[1], Yu-Wei Zhang[2] and Wenping Wang[3]

[1]Institute of Graphics and Image, Hangzhou Dianzi University, Hangzhou, China
[2]School of Mechanical and Automotive Engineering, Qilu University of Technology, Jinan, China
[3]Department of Computer Science, the University of Hong Kong, Hong Kong, China

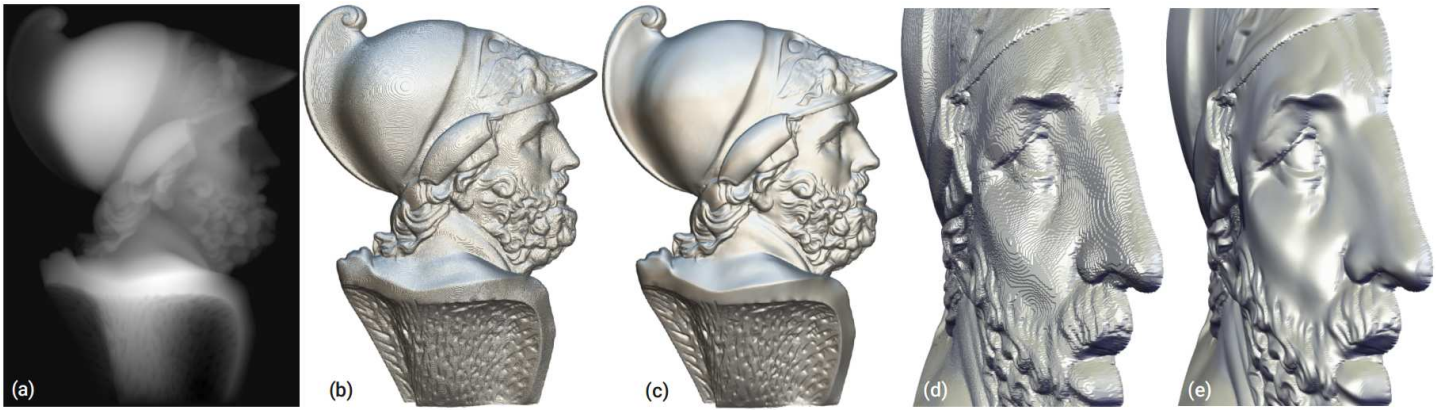


Figure 1: *Geometric decontouring for a grayscale image. (a) Input grayscale. (b) Convert (a) to a 3D mesh and render it. Note that the false contours arise. (c) Our network eliminates the false contours and preserves the fine details well. (d) Close-up of (b). (e) Close-up of (c).*

Abstract

Grayscale images are intensively used to construct or represent geometric details in field of computer graphics. In practice, displacement mapping technique often allows an 8-bit grayscale image input to manipulate the position of vertices. Human eyes are insensitive to the change of intensity between consecutive gray levels, so a grayscale image only provides 256 levels of luminances. However, when the luminances are converted into geometric elements, certain artifacts such as false contours become obvious. In this paper, we formulate the geometric decontouring as a constrained optimization problem from a geometric perspective. Instead of directly solving this optimization problem, we propose a data-driven method to learn a residual mapping function. We design a Geometric DeContouring Network (GDCNet) to eliminate the false contours effectively. To this end, we adopt a ResNet-based network structure and a normal-based loss function. Extensive experimental results demonstrate that accurate reconstructions can be achieved effectively. Our method can be used as a relief compressed representation and enhance the traditional displacement mapping technique to augment 3D models with high-quality geometric details using grayscale images efficiently.

CCS Concepts

• ***Computing methodologies*** → *Height map; False contour; Residual network; Geometric decontouring; Constrained optimization;*

1. Introduction

This paper focuses on removing the false contours from an 8-bit grayscale image which quantified from a height map, as shown in Figure. 1. A height map $h(u,v)$ is a raster image, where the value of each pixel corresponds to the height of a point on the surface. Height maps are used to encode texture maps (such as bump maps, displacement maps and parallax maps). Especially, displacement mapping is an technique transferring the height values to the actual geometric position of vertices over the textured surface are displaced, often along the local surface normal, it gives surfaces a great sense of depth and detail. This technique are intensively applied for use in 3D modeling and in 3D Games. For these

† jzp@hdu.edu.cn

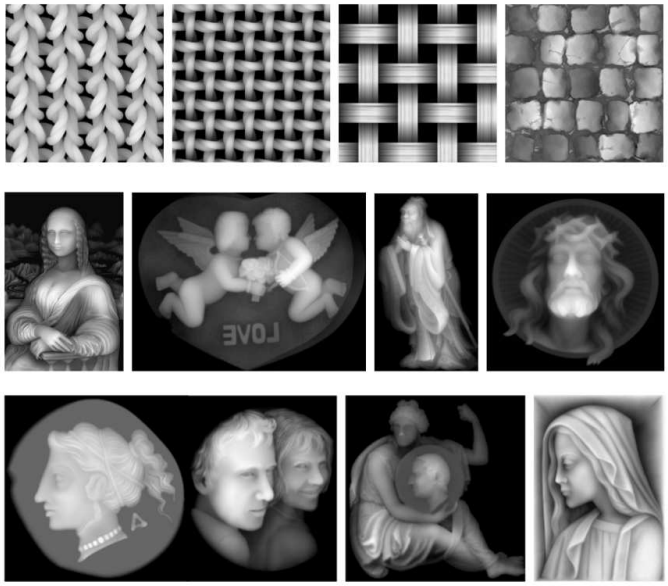


Figure 2: *Some geometric textures and relief artworks represented in grayscale images. There are many artworks on the Internet which are presented in the form of grayscale images.* ***Top row****: some geometric textures used to generate small-scale features in 3D modeling.* ***Middle and bottom rows****: relief artworks released as grayscale images.*

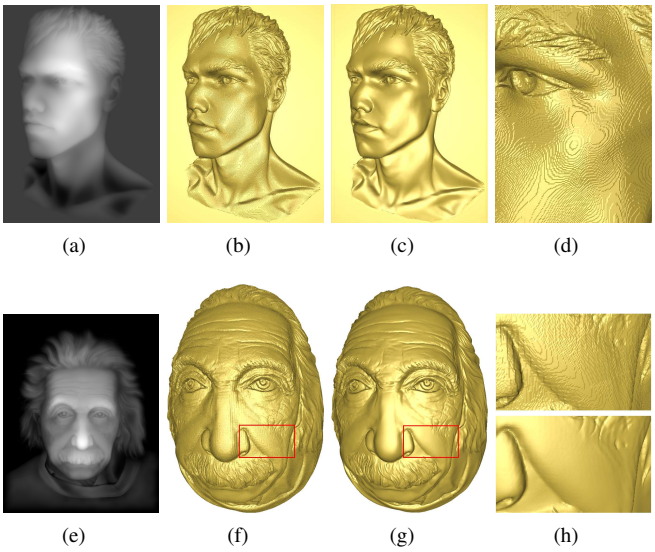


Figure 3: *Artifacts (false contours) introduced by quantization from a height map to an 8-bit grayscale image. When we convert a grayscale image to a 3D mesh and render the mesh under a lighting model, the heavy false contours arise. The goal of this paper is to eliminate the false contours and preserve the fine details by designing and training a deep neural network.*

applications in computer graphics field, height maps are mainly used to construct or to represent high-quality geometric details. These maps are usually represented as 8-bit grayscale images in practice, which only provide 256 levels of luminances. The 8-bit grayscale images do not cause loss of detail for visual display. However, when the luminances are converted into geometric elements, certain artifacts such as contouring (or false contouring) begin to appear under a common lighting model. These refer to artificial boundaries which become visible due to the large and abrupt intensity changes between consecutive gray levels. Using alternative 16-bit grayscales is an option, but this paper focuses on processing plenty of existing 8-bit grayscales available online. Given 8-bit grayscale images, eliminating the false contours to reveal high-quality detailed geometry will be an interesting and useful research topic.

In addition, digital reliefs are usually represented as 2.5D height maps that give each position (u, v) a single height value h above a flat background. As an artistic expression between 2D painting and 3D sculpture, the application of relief is very wide. The relief artworks are often preserved and disseminated in the form of 2D grayscale images. For example, some 8-bit grayscale images which represent relief artworks are shown in Figure. 2. There are a great number of these kinds of grayscale images online. Many height maps are exported from scuplting softwares (such as ArtCAM, JDPaint, etc.) and stored in 8-bit format images, then published and distributed online as free or paid resources.

Compared with 3D mesh model, grayscale image is more convenient for browsing, local storage and sharing online. However, artifacts arise when converting from grayscale image into mesh surface. The fact behind this phenomenon is that the human eye is not good at distinguishing the exact strength of a rapidly varying brightness variation (the high frequencies). Figure. 3(a) shows an example which demonstrates that the false contours in grayscale image is hardly visible for the human eyes. When the image are converted to a mesh surface (see Figure. 3(b)), the rounding errors become visually obvious under a certain lighting model (see close-up in Figure. 3(d)). Figure. 3(e) shows a grayscale with rich details, but some important features (such as the eyebrows) are obscure in the grayscale. Figure. 3(f) presents the noisy (see the top close-up in Fig. 3(h)) image rendered as a mesh surface. When a user downloads a height map and converts it to geometric meshes for some applications (such as 3D engraving, 3D printing and displacement mapping), since there is no better alternative, most users alleviate the false contours by image or mesh denoising methods. To recovered the faithful detailed geometry from a height map, this paper designs and trains a neural network with a normal-guided loss function. The restored result by our method is presented in Figure. 3(c), Figure. 3(g) and at the bottom of Figure. 3(h). By recovering the precision, we will be able to improve the reusability of existing artworks and extend the further editing space of these designs.

Our problem is that how to remove false contour artifacts and reveal detailed geometry, given only an 8-bit grayscale image. The key contributions of this paper are as follows:

- To the best of our knowledge, this is the first study about the geometric decontouring for 8-bit grayscale image with an end-to-end trained deep neural network. Our method can also be regarded as a 3D vectorization technique that converts a grayscale into a 3D mesh surface.
- Our trained network can generate high-quality detailed geometry very efficiently from an 8-bit grayscale image without requiring any parameter tuning, which shows great interest of the method in practice.
- A height map dataset is first constructed to train the proposed neural

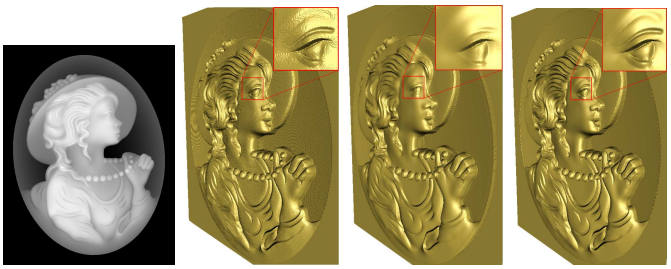
network. The height map in the dataset has a larger bit depth than the RGB color image, which is also potentially beneficial for the research on computer vision. We will release this dataset publicly for further research on related problems.

2. Related Work

Image dequantization Smooth areas of images and video frames should not contain color edges, but false contours are often visible in those areas after color bit-depth reduction or video codec encoding. Several false contour detection and decontouring methods [AK05, LLP*06, BLZ09, JGN11, HKT*18] have been proposed to address this problem. Most of these methods first locate the positions of the false contours and then applies dedicated operations to suppress them visually. However, the quantization artifacts in height maps are more sensitive when rendering under a common lighting model, and the dequantization of height image requires considering the geometric elements, such as normal and position of vertex.

Image denoising The false contours introduced by the quantization for a height map can be regarded as a kind of image degradation or noise. Therefore, we can use the image denoising techniques to restore the image content to an extent. Image denoising aims to recover the clean image from its degraded version. This problem has been extensively studied in literature, and most of the existing methods can be categorized into local structure based methods [TM98, FFLS08, HST13, ZSXJ14, LZZ*15], nonlocal self-similarity based methods [BCM05, DFKE07, GZZF14], sparsity based methods [MES08, WLB17, XZZ18], and low-rankness based methods [DSL13, GZZF14, WLB17]. These methods are also apt to attenuate the sharp features while removing the false contours. We will demonstate the performance of these types of methods on the presented problem. In addition, the convolutional neural networks (CNNs) have been successfully applied for image denoising [JS09, XXC12, ZZZ18, AB19, TXL*20]. Most of these methods are designed to handle Gaussian noise removal. And other neural network based methods which can address special noises are also proposed, such as Deep Image Prior [UVL18] and DoubleDIP [GSI19]. Unlike these methods, we formulate our problem as a geometric optimization and solve it geometrically.

Mesh smoothing A height map can be converted into a 2-manifold mesh directly. However, when the heights are converted into geometric elements, the artificial boundaries become visible due to the large and abrupt intensity changes between consecutive gray levels. The mesh smoothing techniques can be used to remove some false contours from a geometric point of view. These methods can be roughly categorized into two classes, including vertex based methods [FDCO03, JDD03, NISA06, LTJW07, HS13], and normal based methods [SRML07, ZFAT11, ZDZ*15, ZDH*18, LZFH18, YRP18]. The geometric quantities, such as vertices and normals, are taken into account, these methods seem better suited to the presented problem. However, they also do not effectively distinguish between false contours and feature details. Some mesh smoothing methods which achieve impressive results are investigated in our experiments.



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

Figure 4: *(a) The false contours are often invisible in a grayscale. (b) Obvious artifacts (false contours) emerge when a grayscale image converts into a mesh model. (c) The artifacts can be eliminated using traditional image denoising methods, but some important features are removed or attenuated as well (compare the sharp features around eye in (c) and (d)). (d) Our network makes these features well preserved while eliminating the false contours.*

3. Motivation and Our Approach

3.1. Motivation

As shown in Figure. 4, the reliefs are often represented and transmitted by grayscale images, but they are actually height maps. When we convert the height maps into a 2-manifold mesh directly, the rounding errors introduced by image quantization appears as a staircase-like artifact or false contouring (see the close-up in Figure. 4(b)). Unfortunately, quantization from a height map to a grayscale image is irreversible and results in loss of details, and the inversion of quantization is an ill-posed problem. If the artifacts are eliminated from the image or the mesh model using traditional denoising methods, many important details are removed or attenuated as well (compare the sharp features around the eye in Figure. 4(c) and Figure. 4(d)).

Unlike previous algorithms, we do not attempt to detect and remove the false contours, but restore the residual errors from the local orientations and structures directly. Specifically, the goal of this paper is to restore a height map from an 8-bit grayscale image by geometric decontouring. We create a dataset and design a deep network to learn a residual mapping function to predict the residual errors. As can be seen from Figure. 4(d), our network removes the false contours and reveals the fine details well.

3.2. Our Approach

Generally, human eyes are insensitive to a small change of luminance in a grayscale image, due to the light-sensitive nerves in human eyes can only distinguish limited gradations of luminance. However, the problem emerges when the surface mesh converted from the grayscale is rendered under a lighting model. In fact, normals play a central role in surface shading. Specifically, the orientation of a surface plays an important role in the amount of light it reflects. This orientation can be represented at any point p on the surface, by a normal vector $\mathbf{n}$ which is perpendicular to the surface at p.

In this paper, we formulate the geometric decontouring as a constrained optimization problem from a geometric perspective. Because

the surface shading is more sensitive to normal vectors than height values, we define the optimization objective based on the local orientation of a surface. Therefore, given a grayscale image $h_g(u,v)$, we intend to remove the false contours by minimizing the reconstruction error between the original normals $\mathbf{n}(h_o(u,v))$ and the reconstructed normals $\mathbf{n}(h_g(u,v)+r(u,v))$ under the constraints of rounding error bounds,

$$\begin{aligned} \min_{r(u,v)} \; & \|\mathbf{n}(h_g(u,v)+r(u,v)) - \mathbf{n}(h_o(u,v))\|_2^2 \\ s.t. \quad & r(u,v) \le 0.5. \\ & r(u,v) \ge -0.5. \end{aligned} \tag{1}$$

If the original normals $\mathbf{n}(h_o(u,v))$ are known, we can approximate the underlying height map by solving the above constrained optimization problem. Unfortunately, $\mathbf{n}(h_o(u,v))$ is unknown. Without prior knowledges about rounding errors, we seek an alternative solution based on machine learning. Given a training dataset (pairs of $h_o(u,v)$ and $h_g(u,v)$), we can design and train a neural network $\Phi(h_g(u,v))$ to predict the residual error $r(u,v)$, by minimizing the total reconstructed normal errors over a training set,

$$\min_{\Phi} \sum_i \left\|\mathbf{n}(h_g^i(u,v)+\Phi(h_g^i(u,v))) - \mathbf{n}(h_o^i(u,v))\right\|_2^2. \tag{2}$$

To this end, we design a neural network equiped with an activation function to make $\Phi(h_g(u,v))$ satisfy the constraints: $-0.5 \le \Phi(h_g(u,v)) \le 0.5$.

3.3. Architecture

To predict the rounding errors, our GDCNet naturally adopts a residual learning scheme. Specifically, our network is trained to learn a residual mapping function $\Phi(h_g(u,v))$, whose range is $[-0.5, 0.5]$, to approximate the rounding errors. To solve this problem, we have experimented with many network architectures and we finally design a simple but effective architecture. Figure. 5 shows the architecture of the proposed GDCNet. Following the suggestion in [LSK*17], we remove the batch normalization layers from the residual block. We set the size of convolutional filters to be 3×3 and remove all pooling layers. Therefore, the receptive field of GDCNet with d residual blocks is $(2d+2) \times (2d+2)$. Increasing receptive field size can make use of the context information in larger image region. For better tradeoff between performance and efficiency, we set a proper depth $d = 16$ which performs well in our experiments.

Our GDCNet takes as input an 8-bit grayscale image $h_g(u,v)$, and outputs a height map $\Phi(h_g(u,v))$. The $h_g(u,v)$ represents a quantified height map with discrete height values and the estimated rounding errors $\Phi(h_g(u,v))$ are scalars represented by floating point numbers. Then we reconstruct the height map $\hat{h}(u,v) = h_g(u,v) + \Phi(h_g(u,v))$ to approximate the underlying surface. Finally, due to the rounding errors belong to the interval $[-0.5, 0.5]$, we set the last activation function of our network as $\frac{1}{2}tanh(x)$ which guarantees the outputs of network $\Phi(h_g(u,v)) \in [-0.5, 0.5]$.

3.4. Dataset construction

As mentioned above, the goal of this paper is to remove the false contours from a quantified grayscale image in the geometric context. The target to be treated with is a height map, not a general high dymanic range image, so we create a dataset of height maps with fine geometric details. Naturally, we can construct the height maps from 3D objects.

We choose 100 3D objects of the common types from the public datasets (such as Princeton ModelNet [WSK*15], ShapeNet [CFG*15], AIM@SHAPE [Fal04], etc.) and construct 1000 simple 3D scenes which contain one to five objects. Given a 3D scene, we can capture a batch of normal vector images with a resolution of 1024×1024 from some different viewpoints. The normal images are converted into height maps using the bas-relief generation method [JMS14] automatically. What is noteworthy is that the proposed method is not tailored for bas-reliefs, it is applicable to all height maps stored in 8-bit greyscale image. We trained the network using bas-reliefs to make the network handle richer details by leveraging as much detail as possible within a limited height range. Our network is also able to remove false contours for general height maps. The height maps are then quantified into 8-bit grayscale images. We further augment the data by capturing each 3D scene along different directions. Finally, we obtain about 10000 pairs of height maps and grayscales which are split into two subsets: 8,000 pairs for training set, 2,000 for validation set. Figure. 6 shows some height maps (rendered as surfaces) and grayscale images in our training set. In addition, the test set includes grayscale images (without the 'ground truth' height values), and height maps generated from other 3D models and their quantified versions. We will publicly release our dataset in the future.

3.5. Loss function

The primary task of our network is to restore the lost details introduced by the rounding errors. An 8-bit grayscale image here is actually a height map which can be viewed as a 2-manifold surface, and the local orientation of a surface plays an important role in the amount of light it reflects. This orientation at any point p on the surface can be represented by a normal vector which is perpendicular to the surface at p. Specifically, for a height field $z = f(x,y)$, the normal at point $p = (x_0, y_0, z_0)$ can be expressed by

$$\mathbf{n}(f,p) = \frac{1}{\sqrt{1+f_{x_0}^2+f_{y_0}^2}} \left[-f_{x_0}, -f_{y_0}, 1\right]^T, \tag{3}$$

where, the derivatives are discretely approximated with the forward difference over the nearest neighboring pixels. Given a set of height pairs, we learn the parameters of network $\Phi(\cdot)$ by minimizing the reconstruction errors based on normal vectors. Specifically, the squared errors of normal vectors for a pair $(h_o(u,v), h_g(u,v))$ are defined as

$$\begin{aligned} \mathcal{L} &= \sum_{(u,v)\in\Omega} \left\|\mathbf{n}(h_o(u,v)) - \mathbf{n}(h_g(u,v)+\Phi(h_g(u,v)))\right\|_2^2 \\ &= \sum_{(u,v)\in\Omega} \left\|\mathbf{n}(h_o(u,v)) - \mathbf{n}(\phi(u,v))\right\|_2^2, \end{aligned} \tag{4}$$

where, Ω indicates the definition domain in $\mathbb{R}^2$, $\Phi(h_g)$ and $\phi = h_g + \Phi(h_g)$ are the prediction of our network and the reconstructed height map respectively.

Driven by this loss function, our network is trained to repair the normal vector through the local region of each pixel in the grayscale image, thereby indirectly repairing the height map.

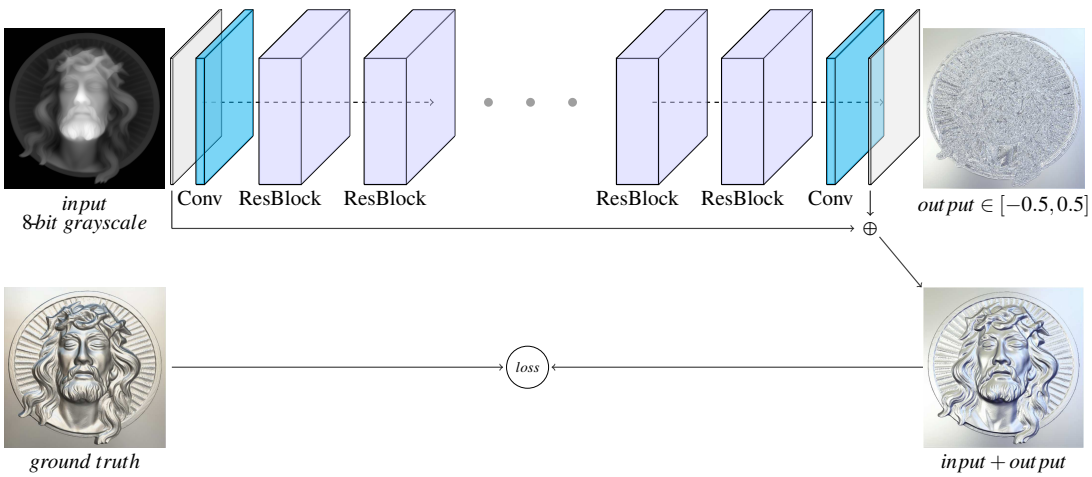


Figure 5: *Architecture of our network for geometric detail restoration from an 8-bit grayscale image. Our network consists of two convolutional layers and several residual blocks. The gray dots indicate repetition of the residual blocks. The output of our network is the predicted residual errors.*

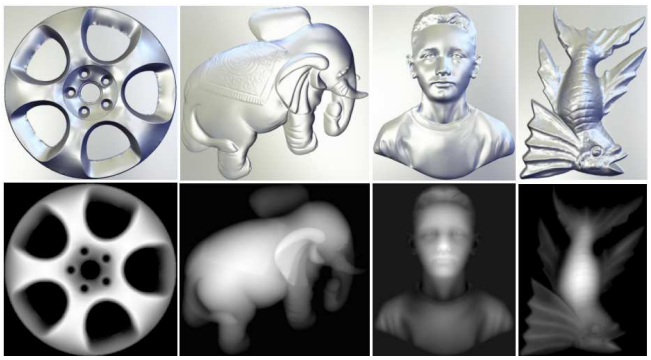


Figure 6: *Some samples of our height-and-grayscale dataset. The images shown in the top row are height maps, and the images in the bottom row are grayscales. A height map encodes the geometric details, so here we render it as a surface.*

In fact, the network can be trained using a loss function that penalizes deviation from the absolute height or the relative height difference encoded by the local geometric quantity. We tested several loss functions based on the height (Equation (5)), the Laplacian ((Equation (6))) of height map and the gradient ((Equation (7))). Here we have not used the activation function $tanh(x)$ which suffers from vanishing gradient problem. Therefore, the networks are free of rounding error constraints in Equation (1). The Laplacian Δ and gradient ∇ are discretely approximated by the finite-difference method. The Laplacian loss $\mathcal{L}_2$ focuses on preserving the local details, which results in bigger height and normal deviations than other losses. The performance of these losses evaluated on a dataset with 32 height maps are compared using the Peak Signal-to-Noise Ratio (PSNR) shown in Table I. The average normal based errors on the dataset are 3.253, 4.961, 3.362, 2.728 degrees for $\mathcal{L}_1$, $\mathcal{L}_2$, $\mathcal{L}_3$ and the normal-based loss respectively. Based on these evaluations, we finally use the normal-based loss function in our current architecture.

$$\mathcal{L}_1 = \sum_{(u,v)\in\Omega} \left\| r(u,v) - \Phi_1(h_g(u,v)) \right\|_2^2, \tag{5}$$

$$\mathcal{L}_2 = \sum_{(u,v)\in\Omega} \left\| \Delta h_o(u,v) - \Delta(h_g(u,v) + \Phi_2(h_g(u,v))) \right\|_2^2, \tag{6}$$

$$\mathcal{L}_3 = \sum_{(u,v)\in\Omega} \left\| \nabla h_o(u,v) - \nabla(h_g(u,v) + \Phi_3(h_g(u,v))) \right\|_2^2, \tag{7}$$

3.6. Training Procedure

We train our network using the Adam optimization algorithm [KB14] by setting $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\varepsilon = 10^{-8}$. The validation set is used to avoid overfitting by early stoping the training. It takes about 20 hours to train our GDCNet. In our experiment, it takes roughly 0.4 to 0.5 seconds to predict the rounding errors using the trained GDCNet. We implement our networks in Pytorch and train it on a PC with an Intel(R) Core(TM) i7-7800X @ 3.5GHz, 16GB of RAM and a GeForce GTX 1080 Ti GPU.

4. Results and Comparisons

In this section, we compare the proposed method with state-of-the-art image denoising algorithms, including BM3D [DFKE07], WLS [FFLS08], GIF [HST13], RNF [ZSXJ14], WNNM [GZZF14], WGIF [LZZ*15], Strollr2d [WLB17], and TWSC [XZZ18], and with some mesh denoising algorithms, including Fast-EFPMD [SRML07], BNF [ZFAT11], GMNF [ZDZ*15] and RoFi [YRP18]. In this paper, the image denoising algorithms take as input the grayscale image, and the mesh denoising algorithms take as input the geometric mesh converted from the grayscale image.

We demonstrate the performance of our GDCNet on a set of 32 3D models and their grayscale images as shown in Figure. 18 and Figure. 19. These grayscale images are the quantified results of the height maps sampled from 3D objects which are not contained in the training set. Therefore, they are unknown for our trained network. These reliefs are used as the ground truth of the grayscale images to evaluate different methods. We use the source codes of state-of-the-art methods to generate results and estimate the runtime on the same machine with 3.5 GHz Intel(R) Core(TM) i7-7800X (16G RAM).

We show the quantitative evaluation (in terms of PSNR) in Table I and Figure. 7. Our GDCNet performs favorably against existing image denoising methods. As mentioned above, normals play a central role in surface shading. In addition to PSNR, we also calculated the normal based error measures for all methods.

Error measures: Given two triangle meshes with the same topology, for each triangle t, $\mathbf{n}_{t,g}$ is the ground truth normal and $\mathbf{n}_{t,p}$ is the estimated one. Similar to [BM12, ZCL*13, LZC*15], we define the error measure for precision as follows,

$$\Theta = \frac{1}{|\mathcal{C}|} \sum_{t \in \mathcal{C}} A(\mathbf{n}_{t,g}, \mathbf{n}_{t,p}) \tag{8}$$

where

$$A(\mathbf{n}_{t,g}, \mathbf{n}_{t,p}) = \begin{cases} \widehat{\mathbf{n}_{t,g}, \mathbf{n}_{t,p}} & \text{if } \widehat{\mathbf{n}_{t,g}, \mathbf{n}_{t,p}} < 10 \\ 90 & \text{otherwise} \end{cases} \tag{9}$$

and $\widehat{\mathbf{n}_g, \mathbf{n}_p}$ is the angle between normal vectors $\mathbf{n}_g$ and $\mathbf{n}_p$. As proposed by [BM12, ZCL*13, LZC*15], we regard the points with the measure greater than 10 degrees as bad points.

We transformed the ground truth images and all results into triangle meshes, and then calculated error measures between the meshes. Our method achieves the best performance efficiently. We show the quantitative results in Table II. The average is calculated according to Equation (9). The maximum value refers to the maximum actual angle.

4.1. Comparison on high resolutions

In order to augment rich details on 3D models, high-resolution images are often required. We compare our method qualitatively with state-of-the-art methods on two high resolution cases, 1400×1800 (Figure. 16) and 3600×2500 (Figure. 17). These high-resolution images contain more high-frequency details which are used to evaluate the ability to restore fine details for different methods. For these two examples, we have no ground truth height map, only grayscale images are available. As can be seen in the figures, most traditional methods fail to restore high-quality details and generate over-smoothed outputs. Some sophisticated methods, such as Strollr2d [WLB17] and TWSC [XZZ18] can also restore some details well, but they are much more time-consuming. Furthermore, most previous methods relied on one or more critical parameters. Tuning the optimal parameter values for each image is a time-consuming task. It took us about dozens of minutes to several hours to find relatively fine parameter settings for these methods. To accelerate the modeling pipeline, this paper proposes an very efficient solution without requiring parameter tuning to restore high-quality geometric details from a height map. This is also an advantage of our method. Comparing with other methods, our network can infer the high-frequency details well very efficiently. The runtime of each method is shown in Table 3. As is shown in the figures and table, our method removes the false contours and achieves the best performance very efficiently.

5. Applications

5.1. Relief Compressed Representation

The digital relief artworks are often preserved and disseminated in the form of 2D grayscale images. This form facilitates viewing and transmission, but it also reduces the quality of the relief model. Therefore, converting the relief model into grayscale images is just the first step of compressed representation. An effective decompression step is also required to recover the high quality reliefs One of the purposes of our network is to achieve this step. Figure. 8 gives an example and shows the result of detail recovering from a grayscale image. The 3D model converted from the grayscale directly is shown in Figure. 8(b). Although the grayscale representation reduces the size of the relief file, it also introduces a large number of obvious false contours (see Figure. 8(b), Figure. 8(d) and Figure. 8(f)). The 3D model converted from the height map recovered using our GDCNet is shown in Figure. 8(c). Due to the constraints in Eq. (1), our method preserves the sharp features very well while removing the heavy false contours effectively. The normals of the relief model and our reconstructed model are shown in Figure. 8(h) and Figure. 8(i) respectively. Figure. 9 shows an example with small-scale features. As shown in Figure. 9(d), the small feature lines and the false contours are almost indistinguishable. Our network makes these feature lines well preserved as shown in Figure. 9(e).

It is worth noting that, the proposed method is not tailored for bas-reliefs, it is also applicable to general height maps stored in 8-bit greyscale image. An example is shown in Figure. 10. In addition, as shown in Figure. 11, our method can also remove false contours from a terrain map stored in a grayscale image.

Our method allows relief to be recovered from a grayscale image, which is useful when one can not access the original relief height maps. In addition to a single image, our network can also process a batch of images efficiently. For instance, given a GIF image which stores a series of grayscale images, a high quality relief animation can be created using our network. Figure. 12 shows a few frames extracted from a relief animation. Our network took about 3 seconds for a total of 200 grayscale images in this GIF, including the time to load the network model.

5.2. High-quality Displacement Mapping

Height maps often appear in the form of assistive tools in computer graphics. For instance, displacement mapping is an technique gives surfaces a great sense of depth and detail, and it is intensively applied for 3D modeling. As shown in Figure. 13, when a grayscale image is directly used as a displacement map, we obtain a rough model with many false contours (see Figure. 13(b) and Figure. 13(d)). Our GDCNet takes as input a grayscale image and reveals the fine details. Then the height values are transferred to the actual geometric position of vertices over a surface mesh along local surface normals. As can be seen in Figure. 13(c) and Figure. 13(e), we obtain a 3D model with fine details, using the restored height map of our GDCNet as a displacement map.

In addition, grayscale images can be easily edited in 2D space. An example of creating a model with rich geometric details by using several

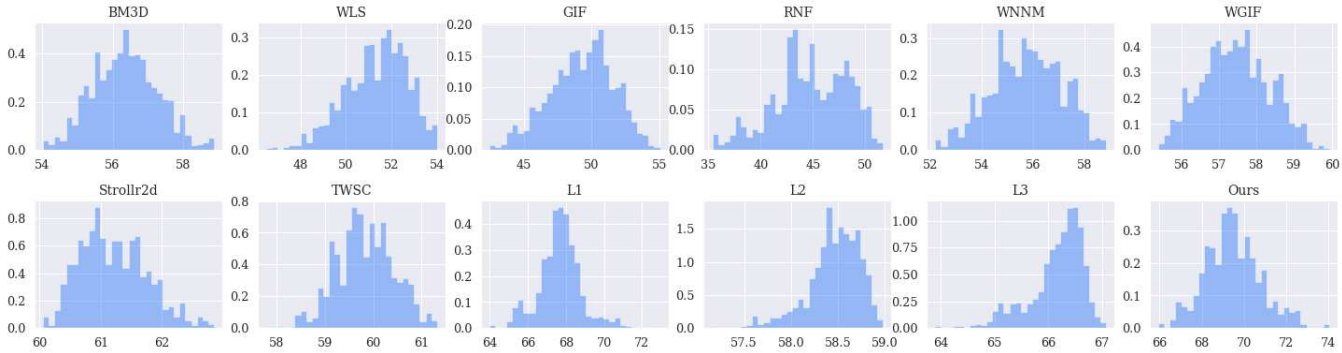


Figure 7: *The histogram of PSNR over test set of each method.*

Table 1: *PSNR of each method*

Image#	BM3D	WLS	GIF	RNF	WNNM	WGIF	Strollr2d	TWSC	$\mathcal{L}_1$	$\mathcal{L}_2$	$\mathcal{L}_3$	ours
1	54.01	47.50	48.96	46.59	54.41	55.89	60.06	57.78	63.97	57.81	64.03	**65.14**
2	55.56	52.40	45.36	42.49	56.67	57.89	61.06	59.06	67.13	58.77	66.15	**68.32**
3	54.79	50.71	47.41	45.19	55.91	58.62	61.06	60.07	68.33	58.85	66.61	**69.63**
4	56.49	50.90	51.12	48.53	56.66	57.13	60.61	59.03	68.18	58.68	66.43	**69.44**
5	57.18	53.62	54.49	51.77	56.69	56.82	60.19	58.45	67.19	58.57	66.31	**68.41**
6	55.81	52.95	49.26	47.62	56.12	58.18	60.81	59.38	68.59	58.90	66.49	**70.06**
7	56.64	50.34	48.78	46.82	57.97	59.18	62.63	61.46	66.96	58.09	65.76	**68.45**
8	57.09	51.54	44.32	37.82	57.43	56.99	60.32	59.09	68.38	58.81	66.53	**69.77**
9	56.71	53.90	50.09	48.12	57.62	59.41	60.61	60.07	72.94	58.97	67.08	**74.02**
10	58.43	53.98	55.04	52.13	57.96	58.28	61.16	59.83	69.15	58.76	66.56	**70.39**
11	54.07	44.70	42.34	37.76	56.23	57.53	60.86	59.24	63.97	57.16	63.76	**65.28**
12	56.15	51.58	48.16	46.36	57.85	57.78	62.07	60.93	67.79	58.51	66.34	**69.06**
13	56.33	52.63	50.57	48.80	56.61	58.21	61.25	59.92	68.56	58.63	66.60	**69.83**
14	54.65	48.24	45.84	43.36	56.68	57.02	60.70	59.25	67.27	58.35	66.56	**68.79**
15	55.79	54.02	50.13	48.55	55.85	57.32	60.48	58.95	66.40	58.68	65.36	**67.69**
16	55.54	51.31	50.46	48.21	55.51	55.77	60.81	58.81	66.34	58.41	65.43	**67.81**
17	54.78	49.04	44.12	41.12	57.24	55.84	60.35	59.27	67.02	58.45	66.00	**68.46**
18	58.74	53.59	46.61	38.49	27.33	59.34	61.23	60.30	68.11	58.98	66.22	**69.38**
19	55.31	47.35	43.22	39.46	57.95	57.81	60.63	59.73	67.05	57.99	66.03	**68.39**
20	55.27	46.97	44.10	39.02	56.47	55.39	60.62	59.23	66.36	58.07	65.53	**67.71**
21	58.76	52.65	42.59	33.93	29.15	59.65	61.04	60.60	66.49	58.10	64.75	**67.66**
22	58.10	51.40	44.42	38.19	58.84	59.01	61.24	60.20	68.88	58.60	66.63	**70.15**
23	56.28	53.40	48.94	47.21	57.93	59.28	61.76	60.72	69.19	58.75	66.82	**70.46**
24	56.19	49.76	48.61	46.70	56.73	56.28	60.76	59.92	69.74	58.86	67.04	**71.07**
25	57.12	48.11	43.84	36.81	58.08	57.23	61.26	60.25	67.48	58.17	66.39	**68.81**
26	57.40	51.24	50.66	48.28	57.44	59.38	62.00	60.46	68.83	58.65	66.84	**70.11**
27	56.57	52.70	51.43	49.33	56.35	59.93	62.67	60.15	66.53	58.38	66.43	**67.74**
28	55.48	50.56	44.58	41.28	58.52	59.23	62.02	60.80	66.63	58.46	66.01	**67.91**
29	58.86	53.02	53.73	51.30	58.84	59.43	62.85	61.18	67.84	58.69	66.09	**69.12**
30	57.07	52.72	50.99	49.11	57.59	59.20	62.36	61.01	68.56	58.63	66.74	**70.03**
31	55.85	47.00	44.13	37.45	56.65	55.45	60.37	59.18	67.58	58.07	66.62	**68.90**
32	56.76	49.65	50.01	47.66	57.18	57.94	60.89	59.27	67.49	58.21	65.92	**68.82**
Avg PSNR	56.37	50.92	46.55	44.55	55.17	57.89	59.25	59.80	67.65	58.43	66.13	**68.96**
Avg Time	18.43s	3.14s	**0.19s**	0.6s	437.55s	0.26s	669.31s	270.19s	0.444s	0.443s	0.445s	0.445s

Table 2: *Normal based errors of each method, maximum/average (degrees).*

Image#	BM3D	WLS	GIF	RNF	WNNM	WGIF	Strollr2d	TWSC	Fast-EFPMD	BNF	GMNF	RoFi	ours
1	156/30.7	147/39.8	151/38.8	153/39.2	154/31.7	111/27.6	151/13.5	155/22.5	157/44.1	158/41.9	157/40.5	178/44.9	**62/5.7**
2	151/24.1	113/30.2	152/28.6	167/28.3	128/27.9	104/22.5	166/11.3	129/21.1	150/38.9	151/37.2	150/35.8	168/40.7	**60/3.9**
3	150/10.5	125/17.4	146/18.2	145/19.1	149/11.3	102/10.0	158/5.2	145/6.4	139/23.6	140/21.7	141/20.2	173/25.7	**57/1.7**
4	162/12.6	165/19.8	151/17.4	152/19.0	162/14.1	162/12.8	163/8.2	164/10.5	163/24.9	165/22.7	155/21.6	178/27.8	**67/2.6**
5	148/17.5	133/22.4	141/20.2	146/20.4	148/20.1	119/18.1	134/10.7	143/15.8	131/26.6	130/24.7	132/23.2	166/29.5	**67/4.5**
6	159/11.5	157/16.4	149/18.9	153/19.0	157/11.8	148/12.3	154/8.0	146/9.1	167/20.6	156/19.0	156/17.5	169/22.4	**60/2.5**
7	164/14.6	144/26.8	157/25.2	160/25.8	152/15.5	107/16.9	156/7.1	153/10.3	167/31.3	155/29.2	154/27.5	176/33.0	**57/2.4**
8	128/13.9	116/22.2	149/20.0	156/20.3	124/15.7	116/15.0	155/8.4	123/11.8	140/58.1	141/57.6	143/56.2	177/57.2	**61/3.1**
9	122/3.4	120/5.8	121/5.9	141/6.4	123/3.6	111/3.1	139/2.5	123/2.4	129/9.8	128/8.6	129/7.7	158/11.3	**52/0.9**
10	141/11.1	118/16.7	139/13.7	139/13.7	141/13.1	118/11.5	137/5.3	140/8.5	131/20.8	129/19.0	126/17.5	166/23.1	**60/2.1**
11	157/18.0	150/33.0	164/33.8	171/36.2	157/17.0	96/17.6	169/9.2	141/11.8	177/48.3	171/45.2	177/44.1	177/48.7	**63/3.6**
12	144/16.9	121/27.4	140/26.0	152/26.0	148/17.9	109/19.4	138/8.4	139/12.0	152/33.1	148/30.9	150/29.1	168/34.6	**56/3.1**
13	152/15.2	132/22.7	139/21.2	145/21.3	146/17.3	127/15.1	137/7.2	138/11.2	156/28.0	178/26.2	145/25.0	160/30.3	**59/2.4**
14	161/14.1	143/30.1	161/23.9	164/24.7	150/17.1	113/14.8	164/5.9	148/9.6	155/39.2	159/35.6	177/34.1	176/42.8	**51/2.1**
15	160/21.5	142/26.9	151/27.6	154/27.1	159/24.4	121/21.8	157/12.0	146/17.4	134/32.9	138/31.7	137/30.8	160/34.6	**79/4.9**
16	152/24.4	139/35.5	150/35.7	154/35.4	150/26.9	117/29.0	147/10.1	147/16.0	135/39.2	137/37.5	136/36.3	176/40.5	**59/3.7**
17	162/17.7	140/31.0	159/29.8	169/29.2	144/17.0	121/21.4	142/11.7	144/13.6	163/33.9	159/32.2	158/29.3	173/34.4	**68/3.8**
18	168/10.5	168/15.0	168/18.1	166/14.6	168/11.7	168/12.2	108/7.7	168/9.2	148/17.3	148/16.0	146/14.9	171/19.2	**37/1.9**
19	154/10.4	142/21.0	151/20.3	169/20.6	154/10.5	115/12.0	156/6.6	152/7.7	177/31.7	159/28.7	168/26.7	175/34.2	**63/2.3**
20	155/18.6	152/41.7	157/36.0	155/39.0	155/20.8	113/24.9	167/7.5	152/11.0	170/52.4	170/50.2	174/48.5	171/52.4	**59/2.9**
21	179/13.3	179/20.6	178/29.2	177/21.3	179/13.9	179/17.5	152/9.5	179/11.9	165/18.5	163/17.4	175/15.9	176/19.6	**40/2.3**
22	153/9.5	128/16.3	153/15.5	158/16.1	141/10.4	120/10.3	141/6.1	142/7.8	148/55.1	152/54.6	175/51.8	161/52.5	**52/2.2**
23	129/9.5	124/16.5	128/15.4	132/15.7	130/10.5	104/9.5	133/5.5	130/7.2	117/22.8	119/20.9	120/19.4	145/25.3	**65/1.9**
24	161/9.0	116/21.0	154/17.8	155/17.8	159/9.8	112/12.7	126/5.9	139/6.7	141/22.5	149/20.4	139/18.3	174/24.4	**53/1.9**
25	155/16.5	137/33.41	162/28.4	169/29.2	144/18.2	118/20.0	170/8.2	145/11.3	162/64.6	159/64.1	169/62.4	175/61.6	**57/2.7**
26	143/11.3	140/20.9	143/17.6	148/18.1	141/14.5	85/11.7	155/4.8	139/8.7	145/27.2	141/24.7	121/23.7	171/30.4	**59/1.9**
27	157/25.1	143/35.2	152/29.2	153/29.7	158/32.6	83/20.0	151/6.2	155/17.8	144/42.8	147/39.8	159/38.9	162/46.0	**62/2.3**
28	162/15.3	138/23.2	158/23.5	169/24.7	137/15.8	102/14.1	163/8.9	138/11.3	157/33.1	156/31.0	156/29.1	169/35.0	**60/3.0**
29	150/15.3	127/25.2	146/23.0	147/22.3	151/16.8	112/17.1	138/7.3	149/10.9	138/29.2	137/27.1	134/25.3	177/31.4	**56/2.4**
30	149/13.3	120/23.2	141/21.2	142/20.8	149/14.8	105/13.8	149/7.4	142/9.5	152/27.8	152/25.3	156/23.4	175/30.7	**60/2.1**
31	155/16.2	150/36.2	153/29.4	164/31.1	154/18.4	115/21.1	151/6.0	152/9.2	163/62.4	149/61.3	174/59.4	173/60.3	**66/2.3**
32	155/12.4	142/21.1	154/20.5	156/20.7	155/12.9	122/13.8	151/6.8	155/9.8	170/24.6	139/22.3	143/20.5	173/26.9	**57/2.2**
Avg Time	18.43s	3.14s	**0.19s**	0.6s	437.55s	0.26s	669.31s	270.19s	58s	115s	4450s	345s	0.445s

Table 3: *Timing (seconds) of each method*

Image#	BM3D	RNF	WNNM	WGIF	Strollr2d	TWSC	ours
Fig. 16	40.2s	4.13s	927s	1.02s	1440s	1673s	**0.447s**
Fig. 17	-	21.3s	6027s	7.55 s	-	4988s	**0.449s**

grayscale images is shown in Figure. 14. More examples of displacement mapping are shown in Figure. 15. Our method can enhance the traditional displacement mapping technique to augment 3D models with high quality geometric details using grayscale images efficiently.

6. Conclusion

In this work, we have investigated the geometric decontouring for 8-bit grayscale images. To predict the rounding errors between the height maps and the 8-bit grayscale images, our geometric decontouring network (GDCNet) naturally adopts a residual learning scheme. Specifically, our ResNet-based network is trained to learn a residual mapping function to approximate the rounding errors.

Because the surface shading is sensitive to the normals, we introduce the normal-based loss function and error measure to estimate the reconstruction errors. Furthermore, based on our experiments, the presented network can perform better for details restoration than traditional image processing and mesh domain processing methods. In future, we plan to investigate the problem of geometric decontouring for grayscale lossy compressed images (such as JPG format).

Acknowledgements

We would like to thank the anonymous reviewers for their careful reviews and constructive comments. This work was supported in part by the National Natural Science Foundation of China (Grant No.61572161, No.61772293) and the Zhejiang Provincial Science and Technology Program in China (2018C01030).

References

[AB19] ANWAR S., BARNES N.: Real image denoising with feature attention. In *ICCV* (2019), IEEE, pp. 3155–3164. 3

[AK05] AHN W., KIM J.-S.: Flat-region detection and false contour removal in the digital tv display. In *ICME* (2005), IEEE Computer Society, pp. 1338–1341. 3

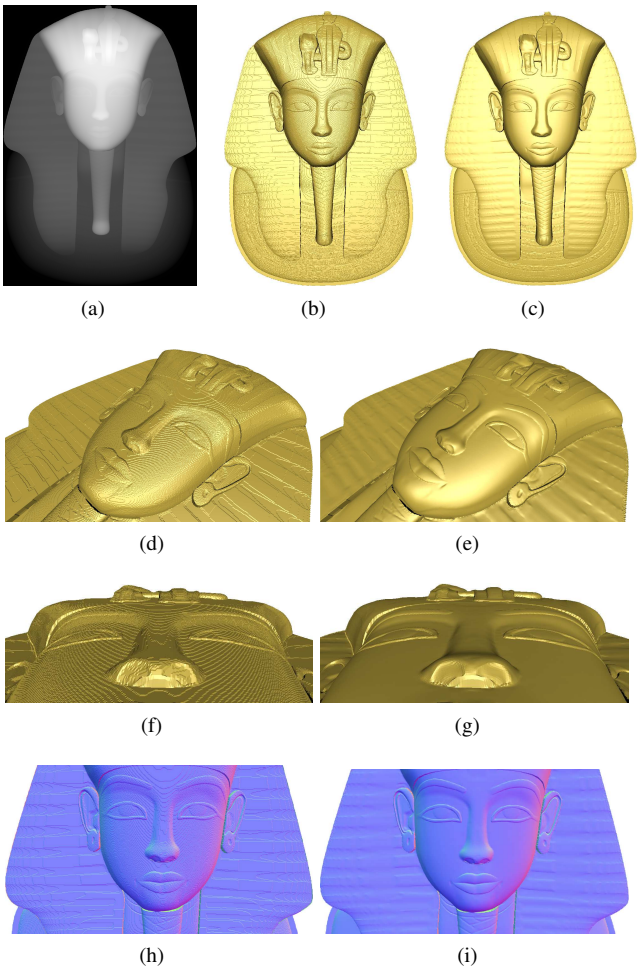


Figure 8: *Geometry and normal recovering from a grayscale image (a) using our GDCNet. (b) The mesh converted from grayscale directly presents obvious false contours. (c) The mesh converted from the height map recovered using our network. (d,f) Close-ups of (b) from different views. (e,g) Close-ups of (c) from different views. (h) The normal shading of (b). (i) The normal shading of (c). Our method preserves the normals and geometric features very well while eliminating the false contours.*

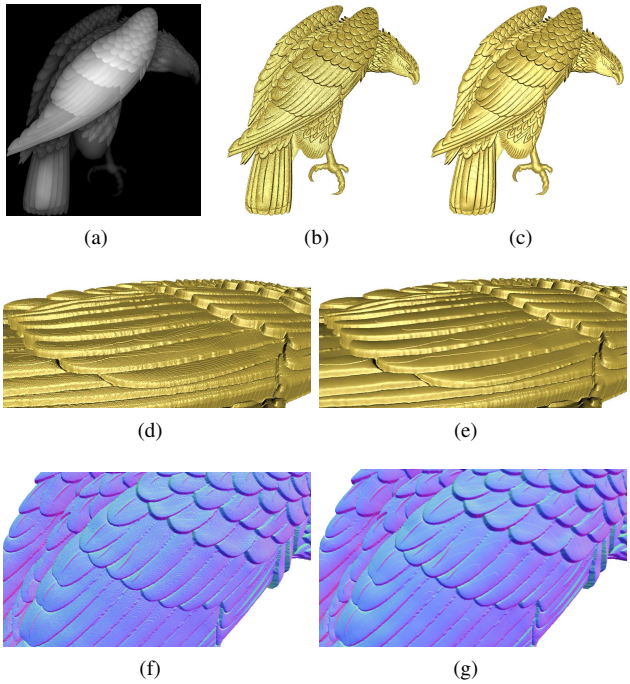


Figure 9: *(a) A grayscale image with small-scale features. (b) The mesh converted from (a) directly. (c) The mesh converted from the height map recovered by our network. (d) Close-up of (b). (e) Close-up of (c). (f) Close-up of the normal shading of (b). (g) Close-up of the normal shading of (c). The small curves on the feathers presented in (e) are small-scale features and our method preserves them very well while eliminating the false contours.*

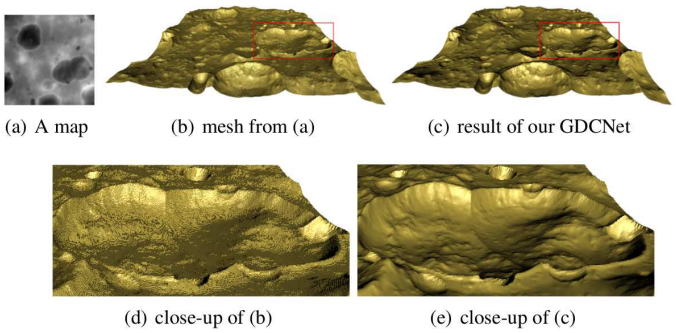
Figure 10: *Decontouring a general height map using GDCNet. (a) The mesh converted from a grayscale directly. (b) The mesh converted from the result of GDCNet. (c) The ground truth mesh.*

[BCM05] BUADES A., COLL B., MOREL J.-M.: A non-local algorithm for image denoising. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 2 - Volume 02* (Washington, DC, USA, 2005), CVPR '05, IEEE Computer Society, pp. 60–65. URL: http://dx.doi.org/10.1109/CVPR.2005.38, doi:10.1109/CVPR.2005.38. 3

[BLZ09] BHAGAVATHY S., LLACH J., ZHAI J. F.: Multiscale probabilistic dithering for suppressing contour artifacts in digital images. *IEEE Trans. Image Processing 18*, 9 (Sept. 2009), 1936–1945. URL: http://dx.doi.org/10.1109/TIP.2009.2022293. 3

[BM12] BOULCH A., MARLET R.: Fast and robust normal estimation for point clouds with sharp features. *Comput. Graph. Forum 31*, 5 (2012), 1765–1774. 6

[CFG*15] CHANG A. X., FUNKHOUSER T. A., GUIBAS L. J., HANRAHAN P., HUANG Q.-X., LI Z., SAVARESE S., SAVVA M., SONG S., SU H., XIAO



(a) A map (b) mesh from (a) (c) result of our GDCNet

(d) close-up of (b) (e) close-up of (c)

Figure 11: *Decontouring a terrain map using GDCNet.*

J., YI L., YU F.: Shapenet: An information-rich 3d model repository. *CoRR abs/1512.03012* (2015). 4

[DFKE07] DABOV K., FOI A., KATKOVNIK V., EGIAZARIAN K. O.: Image denoising by sparse 3-D transform-domain collaborative filtering. *IEEE Trans. Image Processing 16*, 8 (Aug. 2007), 2080–2095. URL: http://dx.doi.org/10.1109/TIP.2007.901238. 3, 5

[DSL13] DONG W., SHI G., LI X.: Nonlocal image restoration with bilateral variance estimation: A low-rank approach. *IEEE Trans. Image Processing 22*, 2 (2013), 700–711. 3

[Fal04] FALCIDIENO B.: Aim@shape project presentation. In *Shape Modeling International* (2004), IEEE Computer Society, p. 329. 4

[FDCO03] FLEISHMAN S., DRORI I., COHEN-OR D.: Bilateral mesh denoising. *ACM Trans. Graph. 22*, 3 (July 2003), 950–953. 3

[FFLS08] FARBMAN Z., FATTAL R., LISCHINSKI D., SZELISKI R.: Edge-preserving decompositions for multi-scale tone and detail manipulation. *ACM Trans. Graph 27*, 3 (2008), 67:1–67:10. 3, 5

[GSI19] GANDELSMAN Y., SHOCHER A., IRANI M.: "double-dip": Unsupervised image decomposition via coupled deep-image-priors. 3

[GZZF14] GU S., ZHANG L., ZUO W., FENG X.: Weighted nuclear norm minimization with application to image denoising. In *CVPR* (2014), IEEE Computer Society, pp. 2862–2869. URL: http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=6909096. 3, 5

[HKT*18] HUANG Q., KIM H. Y., TSAI W.-J., JEONG S., CHOI J. S., KUO C.-C. J.: Understanding and removal of false contour in hevc compressed images. *IEEE Trans. Circuits Syst. Video Techn 28*, 2 (2018), 378–391. 3

[HS13] HE L., SCHAEFER S.: Mesh denoising via l_0 minimization. *ACM Trans. Graph 32*, 4 (2013), 64:1–64:8. 3

[HST13] HE K., SUN J., TANG X.: Guided image filtering. *IEEE Trans. Pattern Anal. Mach. Intell 35*, 6 (2013), 1397–1409. URL: http://doi.ieeecomputersociety.org/10.1109/TPAMI.2012.213. 3, 5

[JDD03] JONES T. R., DURAND F., DESBRUN M.: Non-iterative, feature-preserving mesh smoothing. *ACM Trans. Graph. 22*, 3 (July 2003), 943–949. doi:10.1145/882262.882367. 3

[JGN11] JIN X., GOTO S., NGAN K. N.: Composite model-based dc dithering for suppressing contour artifacts in decompressed video. *IEEE Trans. Image Processing 20*, 8 (2011), 2110–2121. 3

[JMS14] JI Z., MA W., SUN X.: Bas-relief modeling from normal images with intuitive styles. *IEEE Transactions on Visualization and Computer Graphics 20*, 5 (2014), 675–685. 4

[JS09] JAIN V., SEUNG S.: Natural image denoising with convolutional networks. In *Advances in Neural Information Processing Systems 21*, Koller D., Schuurmans D., Bengio Y., Bottou L., (Eds.). Curran Associates, Inc., 2009, pp. 769–776. URL: http://papers.nips.cc/paper/3506-natural-image-denoising-with-convolutional-networks.pdf. 3

[KB14] KINGMA D. P., BA J.: Adam: A method for stochastic optimization. *CoRR abs/1412.6980* (2014). 5

[LLP*06] LEE J. W., LIM B. R., PARK R.-H., KIM J.-S., AHN W.: Two-stage false contour detection using directional contrast and its application to adaptive false contour reduction. *IEEE Trans. Consumer Electronics 52*, 1 (2006), 179–188. 3

[LSK*17] LIM B., SON S., KIM H., NAH S., LEE K. M.: Enhanced deep residual networks for single image super-resolution. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops* (July 2017). 4

[LTJW07] LIU L., TAI C.-L., JI Z., WANG G.: Non-iterative approach for global mesh optimization. *Comput. Aided Des. 39*, 9 (Sept. 2007), 772–782. URL: http://dx.doi.org/10.1016/j.cad.2007.03.004, doi:10.1016/j.cad.2007.03.004. 3

[LZC*15] LIU X., ZHANG J., CAO J., LI B., LIU L.: Quality point cloud normal estimation by guided least squares representation. *Computers & Graphics 51* (2015), 106–116. 6

[LZFH18] LI X., ZHU L., FU C.-W., HENG P.-A.: Non-local low-rank normal filtering for mesh denoising. *Comput. Graph. Forum 37*, 7 (2018), 155–166. 3

[LZZ*15] LI Z., ZHENG J., ZHU Z., YAO W., WU S.: Weighted guided image filtering. *IEEE Trans. Image Processing 24*, 1 (2015), 120–129. 3, 5

[MES08] MAIRAL J., ELAD M., SAPIRO G.: Sparse representation for color image restoration. *IEEE Trans. Image Processing 17*, 1 (Jan. 2008), 53–69. URL: http://dx.doi.org/10.1109/TIP.2007.911828. 3

[NISA06] NEALEN A., IGARASHI T., SORKINE O., ALEXA M.: Laplacian mesh optimization. In *Proceedings of the 4th International Conference on Computer Graphics and Interactive Techniques in Australasia and Southeast Asia* (New York, NY, USA, 2006), GRAPHITE '06, ACM, pp. 381–389. doi:10.1145/1174429.1174494. 3

[SRML07] SUN X., ROSIN P. L., MARTIN R. R., LANGBEIN F. C.: Fast and effective feature-preserving mesh denoising. *IEEE Trans. Vis. Comput. Graph 13*, 5 (2007), 925–938. URL: http://doi.ieeecomputersociety.org/10.1109/TVCG.2007.1065. 3, 5

[TM98] TOMASI C., MANDUCHI R.: Bilateral filtering for gray and color images. In *ICCV* (1998), pp. 839–846. URL: http://dx.doi.org/10.1109/ICCV.1998.710815. 3

[TXL*20] TIAN C., XU Y., LI Z., ZUO W., FEI L., LIU H.: Attention-guided cnn for image denoising. *Neural Networks 124* (2020), 117–129. 3

[UVL18] ULYANOV D., VEDALDI A., LEMPITSKY V. S.: Deep image prior. In *CVPR* (2018), IEEE Computer Society, pp. 9446–9454. 3

[WLB17] WEN B., LI Y., BRESLER Y.: When sparsity meets low-rankness: Transform learning with non-local low-rank constraint for image restoration. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2017), IEEE, pp. 2297–2301. 3, 5, 6

[WSK*15] WU Z., SONG S., KHOSLA A., YU F., ZHANG L., TANG X., XIAO J.: 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of 28th IEEE Conference on Computer Vision and Pattern Recognition* (2015), IEEE Computer Society. 4

[XXC12] XIE J., XU L., CHEN E.: Image denoising and inpainting with deep neural networks. In *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States* (2012), Bartlett P. L., Pereira F. C. N., Burges C. J. C., Bottou L., Weinberger K. Q., (Eds.), pp. 350–358. 3

[XZZ18] XU J., ZHANG L., ZHANG D.: A trilateral weighted sparse coding scheme for real-world image denoising. *ECCV* (2018). 3, 5, 6

[YRP18] YADAV S. K., REITEBUCH U., POLTHIER K.: Robust and high fidelity mesh denoising. *IEEE transactions on visualization and computer graphics 25*, 6 (2018), 2304–2310. 3, 5

[ZCL*13] ZHANG J., CAO J., LIU X., WANG J., LIU J., SHI X.: Point cloud normal estimation via low-rank subspace clustering. *Computers & Graphics 37*, 6 (2013), 697–706. 6

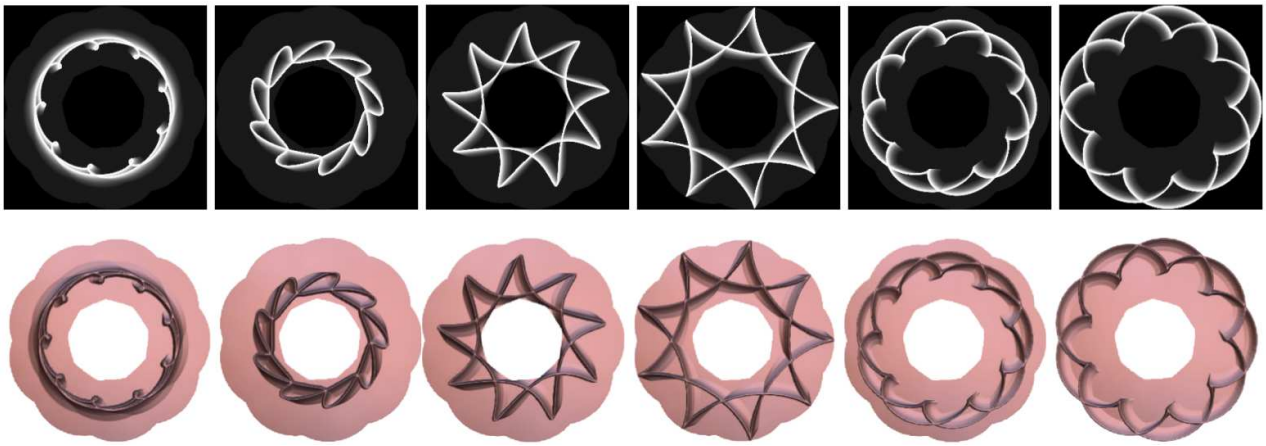


Figure 12: *Given a grayscale animation, our method can convert it into a relief animation. A grayscale animation with 200 frames is given here. Each frame of this animation is recovered by our GDCNet, which generates a relief animation. The 25th, 50th, 80th, 100th, 180th, 200th frames of the grayscale animation and relief animation are listed in the top row and bottom row respectively.*

Figure 13: *A grayscale image is added on a base mesh to create a bottle model with rich details. (b) To displace the position of vertex using the luminance of pixel will introduce the inevitable false contours (see d). (c) Our network takes as input a grayscale image and restores the floating point values which are then transferred to the actual geometric position of vertices along the local normals. Our method can enhance the traditional displacement mapping technique so that it allows grayscale images input to generate 3D models with high quality details (e).*

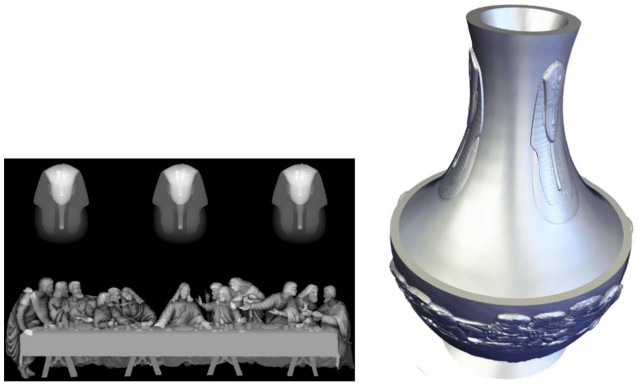


Figure 14: *Users can edit some grayscale images to create a texture map. Our network converts the texture map to a height map stored as floating point values. The height values are then transferred to the model using displacement mapping. Our method makes traditional displacement mapping technique generate 3D models with high quality details more efficiently.*

[ZDH*18] ZHANG J., DENG B., HONG Y., PENG Y., QIN W., LIU L.: Static/dynamic filtering for mesh geometry. *IEEE Transactions on Visualization and Computer Graphics 25* (2018), 1774–1787. 3

[ZDZ*15] ZHANG W., DENG B., ZHANG J., BOUAZIZ S., LIU L.: Guided mesh normal filtering. *Comput. Graph. Forum 34*, 7 (2015), 23–34. 3, 5

[ZFAT11] ZHENG Y., FU H., AU O. K.-C., TAI C.-L.: Bilateral normal filtering for mesh denoising. *IEEE Trans. Vis. Comput. Graph 17*, 10 (2011), 1521–1530. URL: http://doi.ieeecomputersociety.org/10.1109/TVCG.2010.264. 3, 5

[ZSXJ14] ZHANG Q., SHEN X., XU L., JIA J.: Rolling guidance filter. In *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part III* (2014), Fleet D. J., Pajdla T., Schiele B., Tuytelaars T., (Eds.), vol. 8691 of *Lecture Notes in Computer Science*, Springer, pp. 815–830. 3, 5

[ZZZ18] ZHANG K., ZUO W., ZHANG L.: Ffdnet: Toward a fast and flexible solution for cnn-based image denoising. *IEEE Trans. Image Process 27*, 9 (2018), 4608–4622. 3

Figure 15: *More 3D models created by augmenting rich details recovered from grayscale images using our method.*

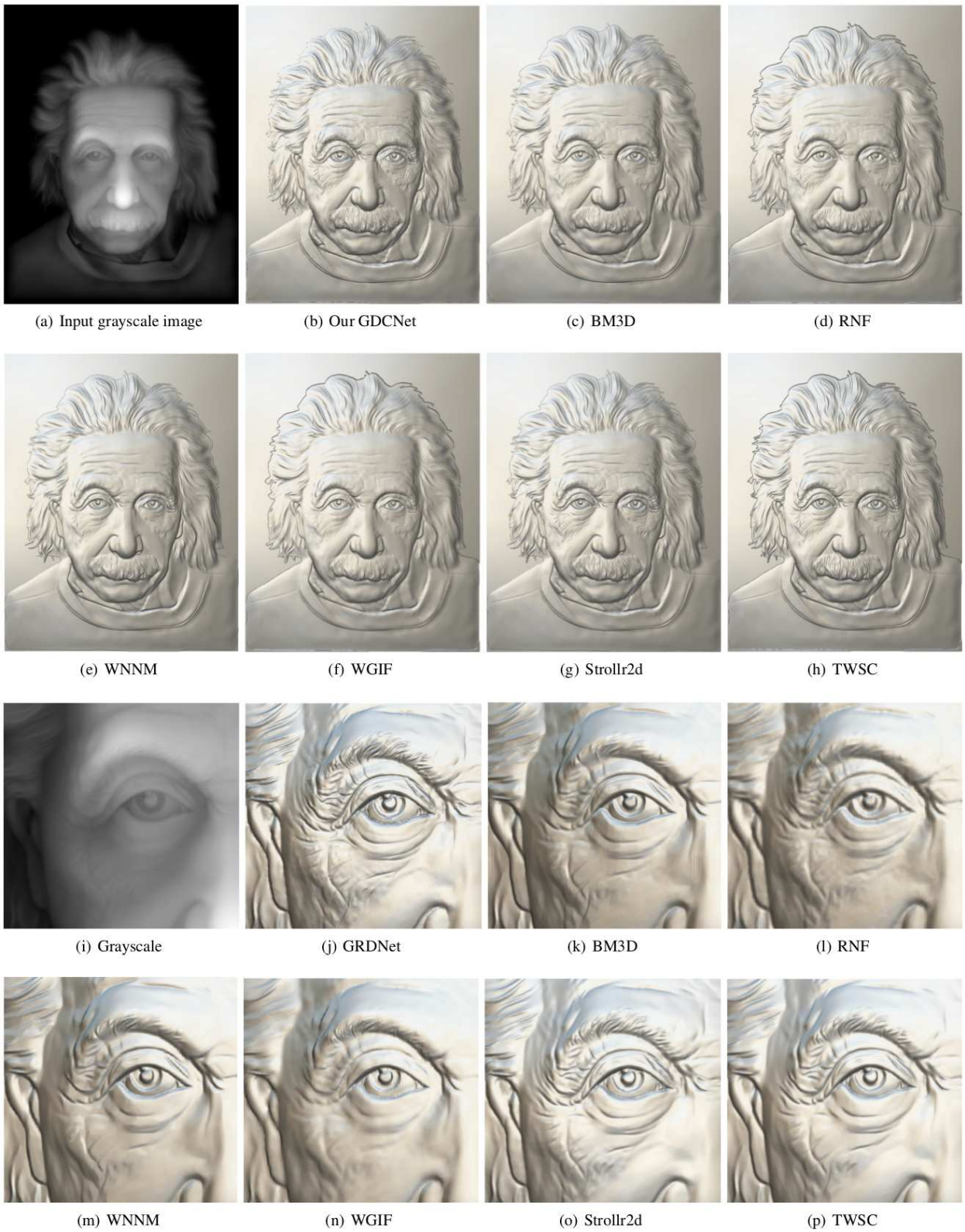


Figure 16: *Results for a portrait grayscale image (resolution:* 1400×1800*) using GDCNet, BM3D, RNF, WNNM, WGIF, Strollr2d and TWSC. The last two rows are close-ups of the first two ones respectively.*

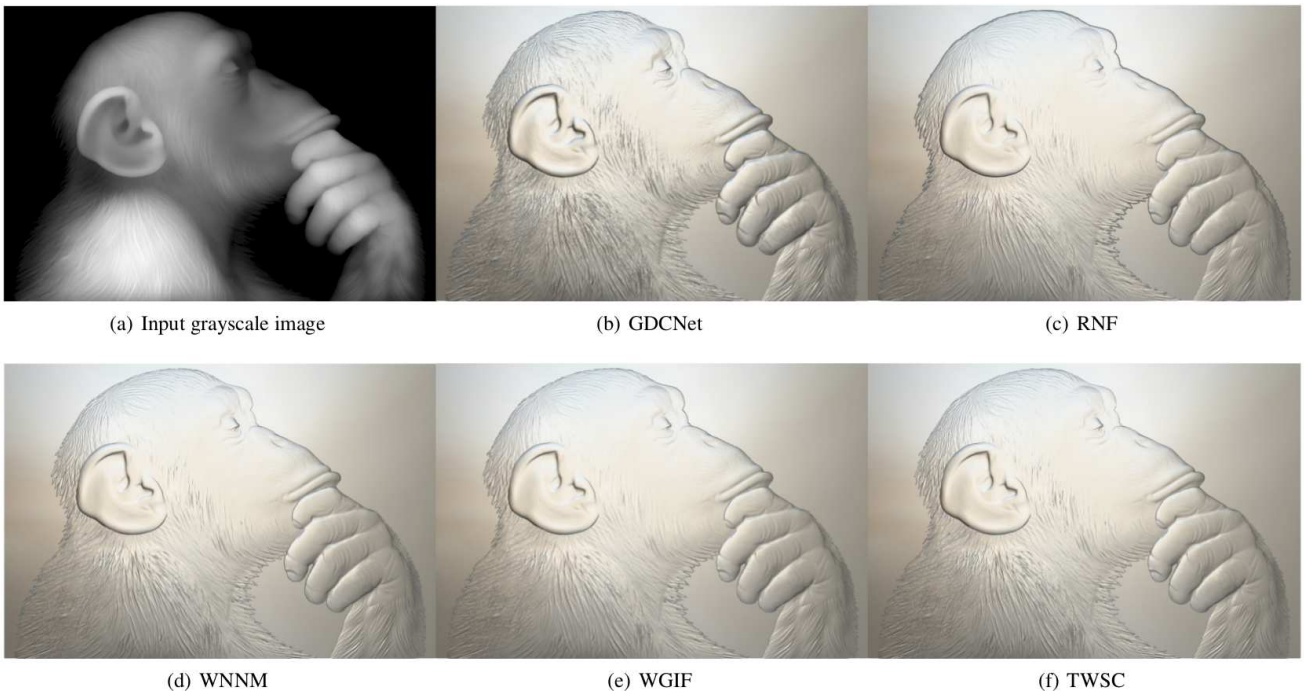


Figure 17: *Results for a gorilla grayscale image (resolution:* 3600×2500*) using GDCNet, RNF, WNNM, WGIF and TWSC. Compare the details on the mouth and the hair on the body and hands.*

Figure 18: *The test set used to estimate the performaces of different methods.*

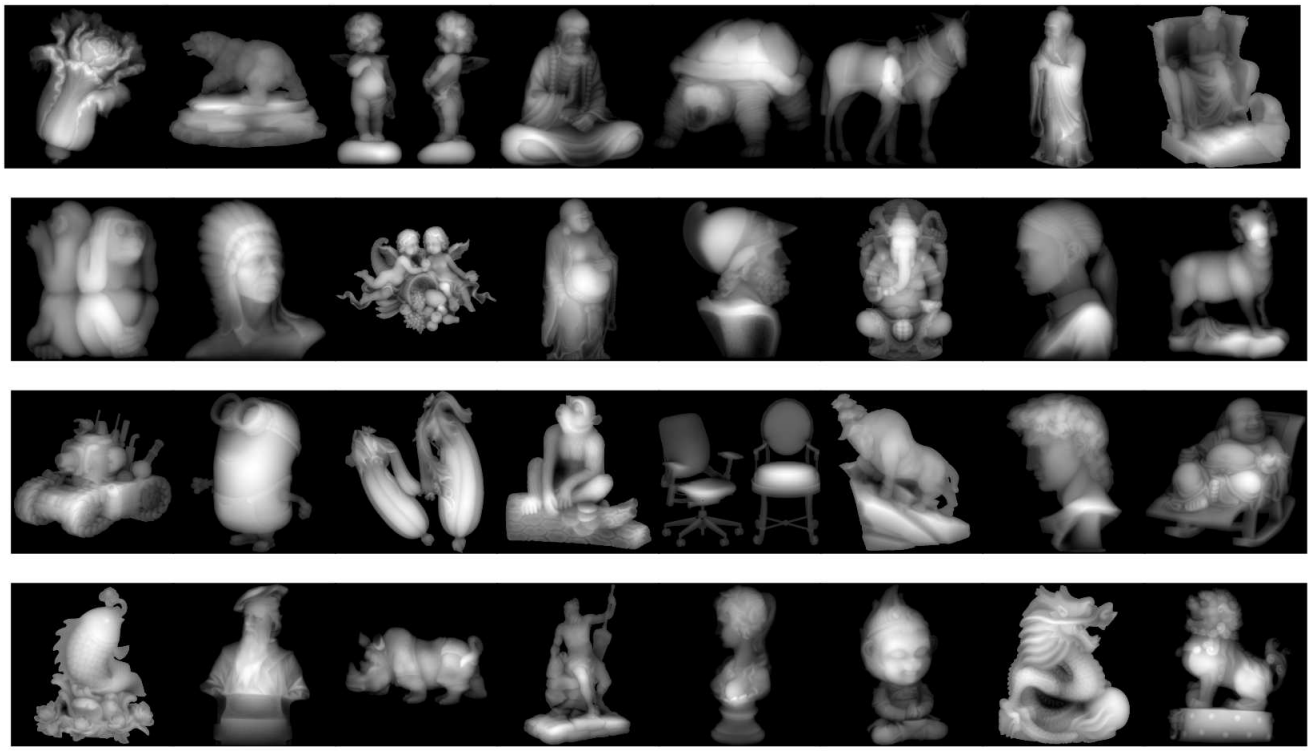


Figure 19: *The grayscale images in the test set.*