

Deep Precomputed Radiance Transfer for Deformable Objects

YUE LI, University of Pennsylvania

PABLO WIEDEMANN, Edinburgh Napier University

KENNY MITCHELL, Edinburgh Napier University

We propose, *DeepPRT*, a deep convolutional neural network to compactly encapsulate the radiance transfer of a freely deformable object for rasterization in real-time.

With pre-computation of radiance transfer (PRT) we can store complex light interactions appropriate to the shape of a given object at each surface point for subsequent real-time rendering via fast linear algebra evaluation against the viewing direction and distant light environment. However, performing light transport projection into an efficient basis representation, such as Spherical Harmonics (SH), requires a numerical Monte Carlo integration computation, limiting usage to rigid only objects or highly constrained deformation sequences. The bottleneck, when considering freely deformable objects, is the heavy memory requirement to wield all pre-computations in rendering with global illumination results. We present a compact representation of PRT for deformable objects with fixed memory consumption, which solves diverse non-linear deformations and is shown to be effective beyond the input training set. Specifically, a *U-Net* is trained to predict the coefficients of the *transfer function* (SH coefficients in this case), for a given animation's shape query each frame in real-time.

We contribute deep learning of PRT within a parametric surface space representation via *geometry images* using harmonic mapping with a texture space filling energy minimization variant. This surface representation facilitates the learning procedure, removing irrelevant, deformation invariant information; and supports standard convolution operations. Finally, comparisons with ground truth and a recent linear morphable-model method is provided.

CCS Concepts: • Computing methodologies → Rendering; Ray tracing; Rasterization;

Additional Key Words and Phrases: real-time rendering, global illumination, pre-computed radiance transfer, spherical harmonics, deep learning

ACM Reference Format:

Yue Li, Pablo Wiedemann, and Kenny Mitchell. 2019. Deep Precomputed Radiance Transfer for Deformable Objects. *Proc. ACM Comput. Graph. Interact. Tech.* 2, 1, Article 3 (May 2019), 16 pages. <https://doi.org/10.1145/3320284>

1 INTRODUCTION

Rendering photo-realistic appearances entails solving the *rendering equation* for each point on an object's surface. This computation can be extremely demanding, especially considering global illumination effects where the problem becomes highly recursive. *Precomputed Radiance Transfer* (PRT) is a technique intended to overcome this by simplifying the rendering equation but still

Authors' addresses: Yue Li, University of Pennsylvania, yueli.cg@gmail.com; Pablo Wiedemann, Edinburgh Napier University, p.wiedemann@napier.ac.uk; Kenny Mitchell, Edinburgh Napier University, k.mitchell2@napier.ac.uk.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2019 Copyright held by the owner/author(s). Publication rights licensed to ACM.

2577-6193/2019/5-ART3 \$15.00

<https://doi.org/10.1145/3320284>

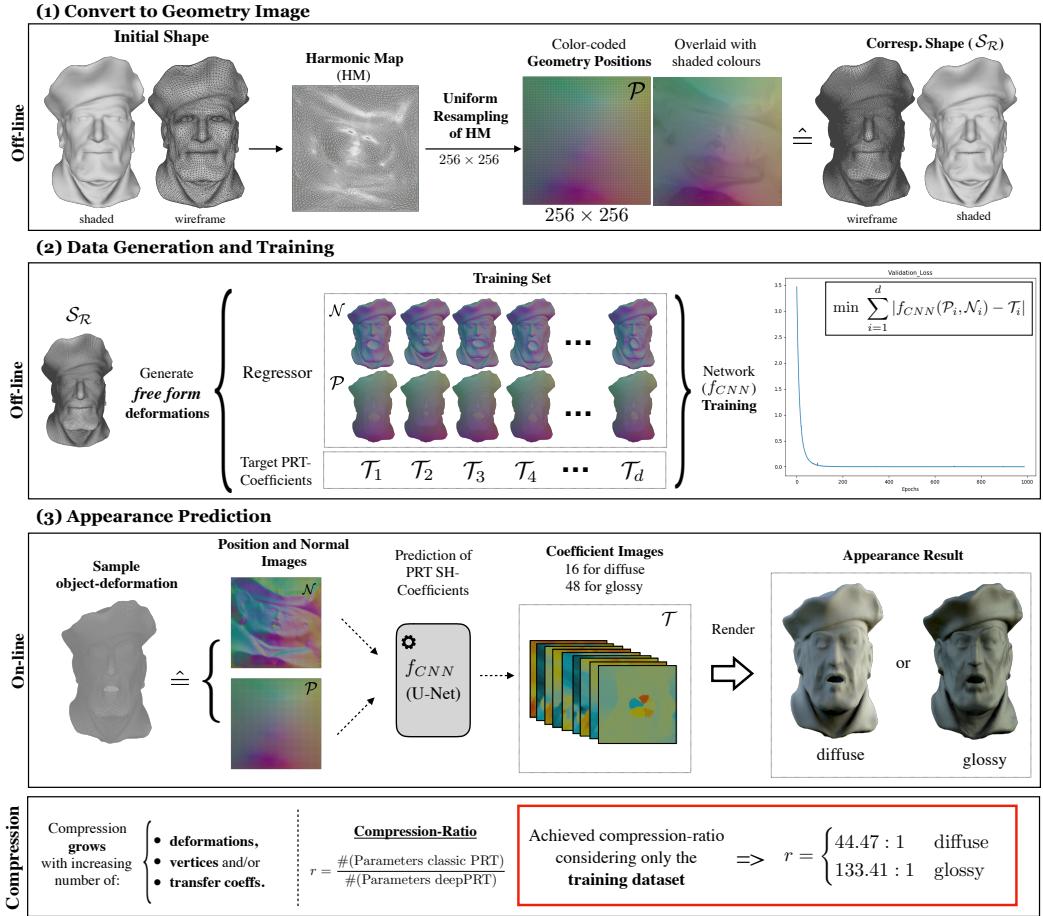


Fig. 1. Overview of our deep deformable PRT compression scheme. **1)** we map an object's surface into a *geometry image* [Gu et al. 2002] on the unit square, via Harmonic Mapping, and then resample uniformly. **2)** we generate training data by applying free form deformations to the reference surface pose \mathcal{S}_R . In addition to the position images we feed the network with surface normals represented in the same regular structure (normal images). Given those inputs, the network is trained to approximate the encoded transfer function \mathcal{T} . **3)** finally, for various deformations of \mathcal{S}_R , the network is able to accurately predict the transfer coefficients (or transfer radiance, for glossy surfaces). Our visual validation application's render time performance is noted in section 3.2.4. In the example above, with a fixed U-Net memory of 45MB compared to classic PRT storage, our method achieves a compression ratio of 44.47:1 sufficient for practical use.

enabling high-quality renderings for complex illuminations. The quintessence is to perform a single pre-computation step of the light-transport and only evaluate scene relative arithmetic at runtime to achieve real-time. Classic PRT algorithms function well for static scenes; however, these are destined to fail in dynamic and/or interactive environments, in which considered objects undergo significant deformations. Most significant is the component of the rendering equation called the *transfer function* which is fully dependent of the shape structure of the object's surface. That is, any object deformation implies a re-computation of the transfer function, often using many ray-traced

samples for each surface point. Hence, using classic PRT to render deformable objects would involve pre-computing large amounts of data, leading to immense storage consumption, rapidly becoming unwieldy for such applications.

Additionally, the time and memory consuming pre-computation of these transfer functions, presumes knowledge of all future deformations of the regarded object. Nevertheless, dynamic or interactive scenes may require on-the-fly adaptive, previously unknown, object deformations.

We propose a Deep Learning framework to overcome the limitations of traditional PRT algorithms described above to enable practical use for diversely deforming objects. In particular, we replace expensive ray-tracing algorithms by a deep Convolutional Neural Network (CNN) that, for a given deformation, infers the corresponding set of SH - coefficients that represent the transfer function. Thus, we enable a compact PRT representation that maintains a constant/fixed storage consumption, regardless of the number of deformations. Moreover, due to the inherent generalization capabilities of deep Neural Networks, our method is able to accurately predict appearances of previously unknown shapes. We call our approach *Deep Precomputed Radiance Transfer* (DeepPRT).

Finding an appropriate representation of shape, or manifold like, data to use in a CNN framework is a challenging task due to the non-Euclidean nature of the domain in which the data is defined on. In such domains, basic operations, such as the convolution, are not well defined. Nonetheless, more recently some authors have started to address non-Euclidean data proposing a variety of approaches [Bronstein et al. 2016; Maron et al. 2017; Masci et al. 2015] (for further reading: [Monti 2018]). In particular, we propose learning on a regular surface representation *geometry images* proposed by Gu et al. [2002]. Surfaces are mapped into squares, we perform harmonic mapping, and resampled into a regular grid. This regular surface space representation allows standard convolution operations and shows to be advantageous within the DL context due to its dieffemorphic property, as shown by Sinha et al. [2016].

The main contribution of our approach is:

- a method of appearance prediction for PRT enabling more general (*free form*) and adaptive deformations, whilst maintaining a fixed compressed representation.

2 RELATED WORK

2.1 Precomputed Radiance Transfer and Extensions

PRT was first proposed by [Sloan et al. 2002] to address global illumination effects on objects for real-time applications. This technique exploits the limitation of static objects by making a single pre-computation step of the transfer function, allowing fast computations at runtime.

PRT for dynamic or deformable objects would require pre-computing sequences of *transfer functions* to account for every pose, resulting in datasets that expand in proportion to the number of poses; hence, rapidly becoming unwieldy or inefficient. Memory IO operations can be 3 orders of magnitudes more energy demanding than floating point summations or multiplications [Horowitz 2014]. Our aim is to extend traditional PRT to deformable geometries while preserving a rather manageable and limited storage consumption.

One extension of PRT was introduced by Sloan et al. [2005] to enable transfer of local illumination effects, such as bumps and wrinkles, to arbitrary deformations. Nevertheless, this method cannot account for distant moving self-shadowing effects, such as cast shadows from a limb to the trunk from an articulated figure. Our intention is to enhance PRT to account for such self-shadowing effects within the modeled range of deformable objects.

Other approaches, rely on exploiting the information of a specific dataset to reduce the dimensionality of the problem and thus the storage consumption. For instance, Feng et al. [2007] introduce

a data-based compression scheme of precomputed radiance transfer matrices. Precomputed transfer matrices of surface samples, deformed by *skinning*, are clustered and compressed, such that de-compression and interpolation can be performed efficiently.

An appearance model, that approximates PRT lighting, is presented by James and Fatahalian [2003]. The model is based on a reduced state space of deformable shapes that allows only very limited variety of poses/shapes.

Similarly, Schneider et al. [2017] suggest a linear self-shadowing model to predict the coefficients of the transfer function from shape parameters of Morphable Models (MoMo) [Blanz and Vetter 1999]. Their proposed model shows good results while operating within the reduced shape space of MoMo; nevertheless, our aim is to provide a more powerful PRT-model, that endows good approximations for more arbitrary deformations living within a larger and more generic shape space. To that end, we rather propose a non-linear model with well known strong generalization properties, namely a deep Convolutional Neural Network [Karpathy et al. 2014; Krizhevsky et al. 2012; LeCun et al. 2015].

2.2 Deep Learning Appearance on Geometry Data

Deep Learning (DL) has been used for appearance predictions before, albeit mostly focusing on learning illumination effects from screen-space data. In Nalbach et al [2017], Chaitanya et al [2017] and Thomas and Forbes [2017] learning is conducted on image data gathered from the shading buffers to predict illumination effects in screen space. However, image-based approaches often suffer from significant information loss, depending on the visibility of the object, and do not leverage the underlying structure of the geometry. These factors make the learning procedure harder requiring large amounts of training data.

A very recent broad approach to apply deep learning to solving light transport given combined geometry, illumination and material [Hermosilla et al. 2018] is available, which solves via unstructured 3D point clouds into a latent representation for resulting image synthesis. Whilst interactive rates are achieved, the ability to vary viewpoint and light environment in a solved PRT representation is much faster and more suitable for real-time interactive 3D graphics applications where latency is important.

Alternatively, we propose directly learning on **geometric** data. However, learning on surfaces using CNNs is a rather challenging task. Due to the non-Euclidean nature of the domain basic operations such as the convolution are not well-defined, leading current research down different paths on the effort to adapt CNNs to such domains (we refer the readers to Bronstein et al [2016] and Masci et al [2016] for a more detailed overview).

One approach, is to circumvent this difficulty by representing the surface data as a probability distribution on a 3D grid and apply volumetric CNNs [Wu et al. 2015]. However, this extrinsic representation has many shortcomings when applied to deformable geometries: They are very sensitive to deformations, are computationally expensive and, equally to the screen-space strategies, require abounding training data. Conversely, strategies for intrinsic shape representations propose different adaptations of CNNs to such domains [Boscaini et al. 2016; Maron et al. 2017; Masci et al. 2015].

In our work, we chose a shape representation that, on the one hand, can endow the underlying shape structure, and on the other hand, supports standard 2D convolution operations. We adopt a parametric approach introduced by Gu et al. [2002], called *geometry images*, that transforms a discrete surface into a regularly sampled unit square. This approach has been extended by Praun and Hoppe [2003] to smooth out some critical limitations of the original work and later validated by Sinha et al [2016] as suitable framework for Deep Learning purposes.

To the extent of our knowledge, our work is the first to tackle the problem of PRT for deformable objects from a Deep Learning perspective and especially on manifold like data.

3 METHOD

Precomputed Radiance Transfer is a physically-based rendering method to accelerate on-line computations of the (simplified) rendering equation:

$$L(\omega_0) = \int_{\Omega} L_{\epsilon}(\omega_i) \underbrace{f(\omega_i, \omega_0)V(\omega_i)H_N(\omega_i)}_{T(\omega_i, \omega_0)} d\omega_i \quad (1)$$

where L_{ϵ} accounts for all incoming radiance over the hemisphere, f describes the surface reflectance properties f (BRDF), H_N is the Lambert's Law and V the visibility function describing geometric information of the scene. PRT precisely exploits the essence of static/non-deformable objects by uniquely determining the integrand $T(\omega_i, \omega_0)$ (called the **transfer function**), which contains the costly-to-compute visibility term,

$$V : \mathcal{S} \times \Omega \rightarrow \{0, 1\}$$

for each surface point $s \in \mathcal{S} \subset \mathbb{R}^3$ [Cohen et al. 1993].

Both functions L_{ϵ} and T are projected onto a suitable set of orthonormal basis functions for faster evaluation of the rendering equation 1. For m number of coefficients of the basis functions and l_i, t_i being the i -th coefficient of L_{ϵ} and T respectively, equation 1 reduces to, [Sloan et al. 2002]

$$L(\omega_0) \approx \sum_j^m l_j \cdot t_j \quad (2)$$

We chose a *Spherical Harmonics* (SH) basis to encode the transfer function T and the light environment L_{ϵ} .

As mentioned above, our aim is to extend the PRT method to malleable and dynamic objects, but avoiding costly precomputations and storage of every single transfer function T_i per shape query S_i (with $i \in [1, 2, \dots, d]$ and $d : \# \text{deformations}$).

With this in mind, we suggest a data-based model, a fully convolutional neural network, to infer the transfer function T_i for any given shape query S_i . This makes the costly ray-casting computations superfluous and solves the exorbitant memory requirements, only necessitating the storage of the network's parameters.

Moreover, by choosing to estimate the transfer function, instead of a direct prediction of L , our method is flexible to dynamic light environments (see figure 2).

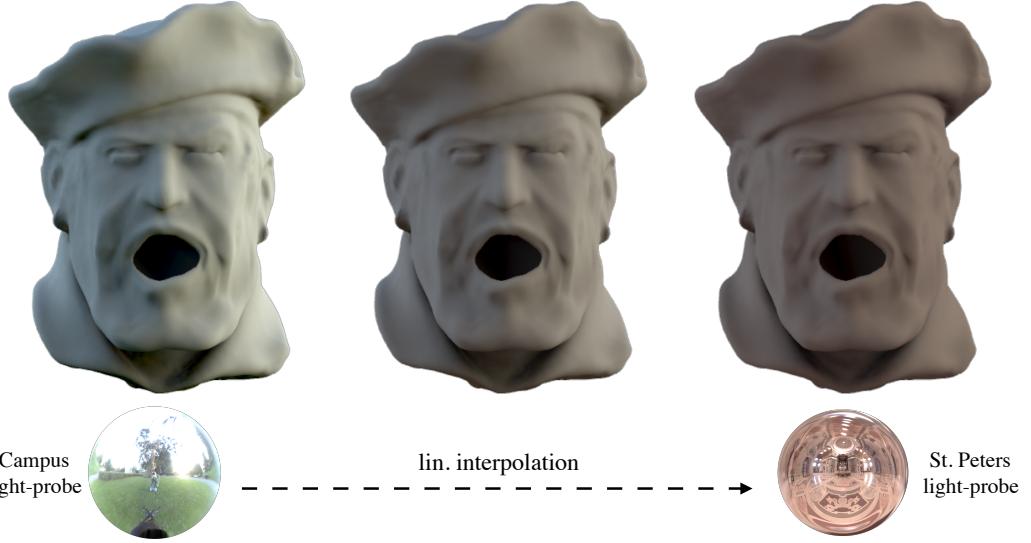


Fig. 2. Dynamic lighting. Rendered appearance of the *Pirate Head* under three different lighting conditions. Lighting is interpolated between two light-probes.

3.1 Data: Geometry Images

We propose learning directly **on** the object’s surface in order to leverage its underlying shape structure and further to be consistent with regular animated surface geometry representation for real-time rasterization. Geometry images present a planar and regular shape representation on which standard 2D CNNs can be applied [Gu et al. 2002; Sinha et al. 2016]. Surfaces with a single boundary (topological disks) are mapped onto a unit square and later discretized (or re-sampled) into a regular grid of $n \times n$ vertices. We chose harmonic map (HM) as implicit parametrization of the interior of the 2D domain [Ells and Sampson 1964; Gu 2018]. Further, our intent is that, for practically zero genus object surfaces, HMs provide the conditions for a homomorphism with diffeomorphic interior mapping such that the mapped differentiable manifold takes place smoothly in the image domain.

Importantly, we apply deformations only on the reconstructed object (shape \mathcal{S}_R in Figure (1)) in order to make our shape representation, geometry images, invariant to deformations. By doing so, we maintain a one-to-one pixel correspondence between deformations; hence, filtering out deformation invariant information of the representation; and therefore, facilitating the feature extraction of surface properties that are more closely related to self-shadowing and light transfer effects. Moreover, this also means that the conversion process of a surface into a geometry image is required only once and can be performed off-line, saving precious computation time.

In addition to position values \mathcal{P} , we use surface normals \mathcal{N} as regressor for the network, where

$$\mathcal{P} = [P_x, P_y, P_z]^T, \quad \mathcal{N} = [N_x, N_y, N_z]^T$$

and $P_i, N_i \in \mathbb{R}^{n \times n}$ being each coordinate image $i \in \{x, y, z\}$ of positions and normals respectively.

In result, our network predicts a corresponding sequence of images $\mathcal{T}_{\mathcal{D}, \mathcal{G}}$,

$$f_{CNN}(\mathcal{P}, \mathcal{N}) = \begin{cases} \mathcal{T}_{\mathcal{D}} & \text{diffuse} \\ \mathcal{T}_{\mathcal{G}} & \text{glossy} \end{cases}$$

which, for **diffuse materials**, consists of the SH-coefficients of the transfer function of the input shape, as introduced above (see equation 2):

$$\mathcal{T}_{\mathcal{D}} = [t_1, t_2, \dots, t_m]^T \in \mathbb{R}^{m \times n \times n}$$

that is, pixel i of image t_j represents the transfer coefficient j of vertex i of the input surface.

For **glossy materials**, our network predicts the transferred radiance $\mathcal{T}_{\mathcal{G}}$ consisting of three radiance channels,

$$\mathcal{T}_{\mathcal{G}} = [R_r, R_g, R_b]^T \in \mathbb{R}^{3 \times m \times n \times n}$$

resulting from the product between the transfer matrix M_T and the lighting coefficients L_i^{sh} [Sloan et al. 2002].

$$R_i = M_T \cdot L_i^{sh} \in \mathbb{R}^{m \times n \times n} \quad \text{for } i \in \{r, g, b\}.$$

3.2 Network and Performance

3.2.1 Architecture: Our deep convolutional network is configured as a *U-Net* [Ronneberger et al. 2015], consisting of an *encoder* and *decoder* with *skip-connections*. Both encoder and decoder consist of sequences of *ResNet* blocks [He et al. 2015] each comprising a series of *2D-convolutions*, *batch-normalisation* and *ReLU* activation layers (illustrated in Figure (3)). For the last layer of the decoder we use a *sigmoid* activation function. Instead of a *pooling-layer* we perform down-sampling by increasing the stride, by a factor of two, within a convolutional layer [Springenberg et al. 2014]. To avoid information loss, we make use of skip-connections passing information between outputs of encoding layers and corresponding inputs of the decoding layers.

3.2.2 Synthesis of Training Data: We generate the training data applying sequences of free form deformations onto a given object. Specifically, for our experiments, deformations are generated either by a physically driven simulation or by linear combinations of blendshapes (figure 7). We also refer readers to the supplementary video.

Each generated deformation sequence has a total length of 500 frames. At each frame $i \in [1, 2, \dots, 500]$ we store the position \mathcal{P}_i and normal \mathcal{N}_i images of the current deformation, and perform a full integration using ray samples to compute the corresponding ground truth $\mathcal{T}_{\mathcal{D},i}$ or $\mathcal{T}_{\mathcal{G},i}$. For our experiments we chose a SH-band number of 4, which corresponds to 16 coefficients. Moreover, we experimented with image resolutions of 256×256 and 512×512 .

Our implementation for the generation of the ground truth data $\mathcal{T}_{\mathcal{D},\mathcal{G}}$ requires a duration of 39.6s for diffuse and 49.7s for glossy surfaces, per frame.

3.2.3 Training: The data is split into a training and validation set consisting of 450 and 50 data samples respectively. As cost function, we minimize the pixel-wise absolute error between predicted output and the ground-truth (L_1 -loss), and the optimizer we use is ADAM [Kingma and Ba 2014].

Convergence varies from object to object, but in most cases 500 to 1000 epochs are sufficient, whereby we used a batch size of 5. The network is implemented in Keras with Tensorflow as backend [Chollet et al. 2015] and takes close to 16 hours to compute 800 epochs of training on a single NVIDIA GeForce GTX 2080 GPU.

3.2.4 Runtime: In order for *DeepPRT* to function in real-time, fast network inference is required. However, to achieve good approximations existing deep neural networks, such as ours, rely on very deep architectures with sufficient quantities of parameters. In particular, our network has approximately 11.8 million parameters, occupying 45Mbytes of CPU memory, and with an inference time of 45ms on a high-end GPU NVIDIA RTX 2080. In addition, for a network input size of

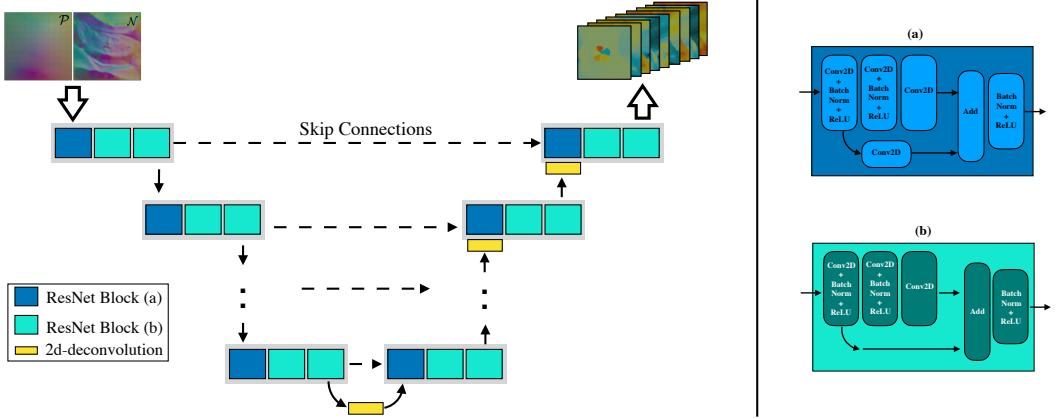


Fig. 3. *Left:* Illustration of the configuration of our U-shaped network. *Right:* Shows the two kind of operations performed within our ResNet blocks.

4.5Mbytes there is a further upload (loading input images, \mathcal{P} , and \mathcal{N} , from CPU into the GPU) cost of approximately 5ms.

After prediction, the performance for the rendering process differs notably between diffuse and glossy surfaces: For **diffuse** surfaces, the dot product between predicted transfer coefficients $\mathcal{T}_{\mathcal{D}}$ and the three channels of the lighting coefficients are calculated (eq. 2) and the result passed to the buffers for final rendering. For our sample geometries with 256^2 number of vertices the computation of the dot product takes 13ms on the GPU, and an upload time of 0.09s for the resulting 4.0Mbytes exiting radiance vectors. On the other hand, the output of the network for **glossy** surfaces $\mathcal{T}_{\mathcal{G}}$ is three times the size as for diffuse objects, thus for the same sample geometry as above, it occupies 12.0Mbytes. For further rendering, $\mathcal{T}_{\mathcal{G}}$ is convolved with a BRDF-kernel, based on Kautz et al. [2002], before passed to the shaders [Sloan et al. 2002]. We perform these steps in the CPU resulting in an additional duration of 1.29s.

We note that our implementation is far from being optimal. Both, the rendering and the prediction process can be accelerated significantly. It has been extensively shown that most neural network models are highly compressible and can be significantly accelerated, eventually making them deployable to devices with low memory resources and applicable in real-time [Cheng et al. 2017; Han et al. 2015]. Nevertheless, exploring network optimization methods reaches beyond the scope of this work. Hence, for our experiments we used the uncompressed U-Net introduced above. In section 4 we show that even neglecting network optimization our DeepPRT approach already achieves immense memory savings, compared to traditional PRT.

4 EXPERIMENTS AND ANALYSIS

We test our DeepPRT method against different animated objects: We animate a *Pirate Head* and a *Fish* object using a blendshape model and apply physically-based deformations to animate a *Cloth* object. The geometry image mesh resolution of the objects is 256×256 . However, for the Fish object we increase the resolution to 512×512 to avoid reconstruction artifacts such as under-sampling and the limitation of harmonic mapping to recover sharp features (figure 4)

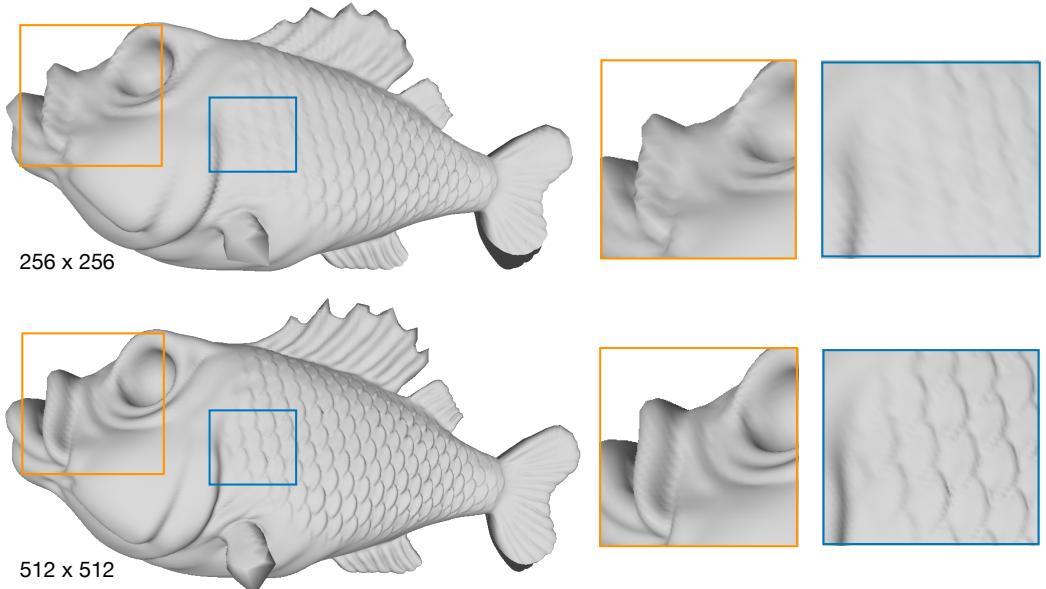


Fig. 4. Fish reconstructions after regular sampling of squared harmonic map. *Top*: Sampled by a 256×256 grid, shows detail loss and distortions. *Bottom*: Higher sampling 512×512 needed to preserve detail and minimize artifacts.

These distortions arise due to the uniform remeshing, while using a sub-optimal choice of 2D surface parametrization (Harmonic Map), leading to poorly sampled surface areas in regions of higher curvatures. While sufficient for our experiments, a more suited parametrization, that minimizes distortion after resampling, is the *geometric-stretch* parametrization [Gu et al. 2002].

Table 1. Network accuracy and SSIM are computed on training plus validation data, first and last columns respectively. Third and fourth columns show the training and validation losses of the network.

	Accuracy	Train-Loss	Val-Loss	SSIM
Pirate	0.9817	0.000397	0.000399	0.99386
Fish	0.9729	0.002104	0.002200	0.92451
Cloth	0.9818	0.000078	0.000098	0.98840

4.1 Quality and Memory Savings

For each of the test conditions, our train- and validation sets together comprise 500 distinct object deformations. Within that deformation set, our U-Net model is able to achieve accuracies up to 98% (table 1). Moreover, the resulting rendered appearances are close to indistinguishable from the ground truth. We further quantify the quality of the results using the perceptual metric SSIM visualization (figures 5 and 7). Based on this result, we imply that our trained network is able to faithfully approximate self-shadowing effects of 500 distinct shapes **at the very least**. Contrarily to classic PRT, which involves storing the transfer coefficients of each vertex for every single shape in the set, whilst our method only requires the storage of the network parameters. Such classic PRT data could be 10GB for a 10 second animation sequence. The weights of the neural network we produce on the other hand is only 143MB and as a fixed sized, independent of the

length animation, is more uniform for practical storage allocation usage. We are able to have close results both visually and numerically to ground truth shading with our experiments. Across the metrics errors visualizations, we observe higher error regions that coincide with originating sharp features of the test objects, such that primarily follows from a limitation of harmonic maps to retain these geometric events, which may be further addressed by alternative geometry image mapping approaches.

For our particular network of approx. 11.8 million parameters, the example objects with 256×256 and 512×512 number of vertices, and a choice of 16 transfer coefficients per vertex, this implies a compression ratio $r = (\# \text{ PRT parameters}) / (\# \text{ network parameters})$ of:

$$r_{\text{diffuse}} = \begin{cases} 44.47 : 1, & \text{for } 256^2 \text{ #vertices} \\ 177.86 : 1 & \text{for } 512^2 \text{ #vertices} \end{cases}$$

$$r_{\text{glossy}} = \begin{cases} 133.41 : 1, & \text{for } 256^2 \text{ #vertices} \\ 533.56 : 1 & \text{for } 512^2 \text{ #vertices} \end{cases}$$

for diffuse and glossy surfaces respectively. These numbers grow linearly with an increasing number of coefficients, deformations or number of vertices.

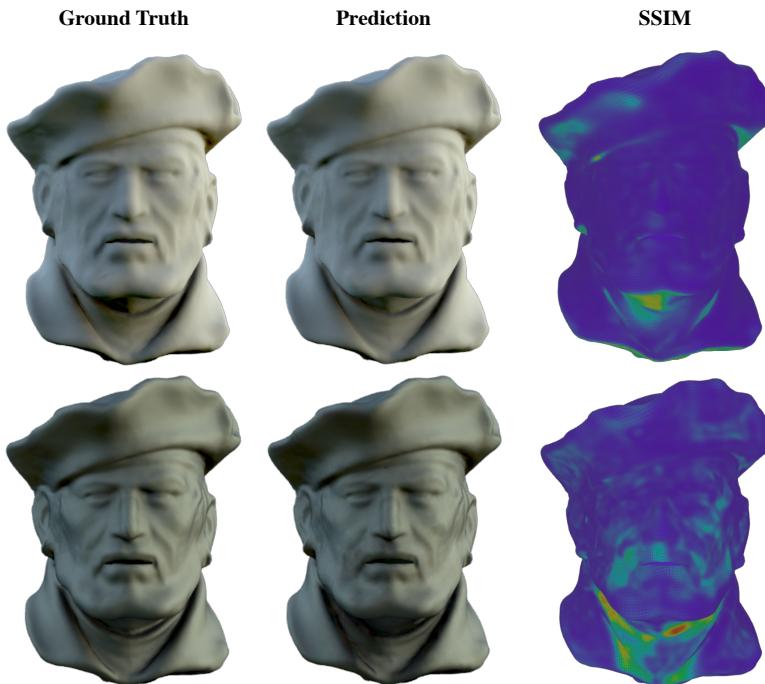


Fig. 5. Prediction of a test sample (unseen while training) for a diffuse (top) and glossy (bottom) surface.

The measures shown above express the compression ratio taking into account solely the training and validation sets. Nevertheless, our network shows good generalization properties, also enabling accurate and qualitatively precise appearance predictions of deformations outside the training set. Hence, taking this into account the compression ratio grows significantly with potential to cover the full range of natural/possible motions for a given object within the fixed network footprint.

As mentioned above, deep neural networks can generally be further optimized, hence making DeepPRT even more efficient memory, speed and energy-wise [Cheng et al. 2017].

4.2 Generality of DeepPRT and Comparison

4.2.1 Generalization Capability: We validate the generalization capabilities of our model by standard machine learning procedures. We base our parameter tuning on minimizing the validation loss and later analyze prediction quality using a test set.

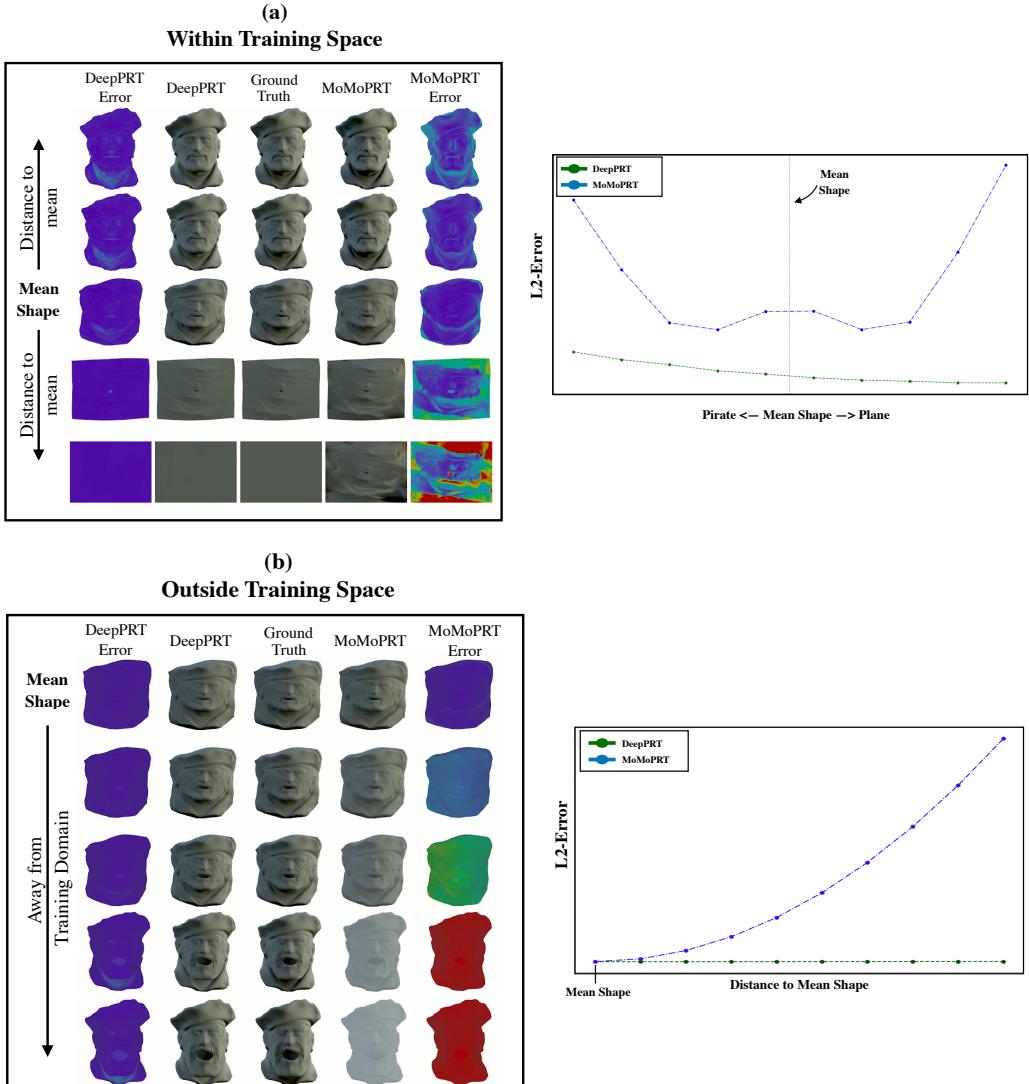


Fig. 6. DeepPRT (ours) vs. MoMoPRT, (a) within training space and (b) moving away from the training space. *Left:* Appearance and vertex-wise L_2 -RGB distance between ground truth and predictions. *Right:* Plot of the mean L_2 -RGB distance for each predicted shape. DeepPRT error is lower overall and close to constant, while MoMoPRT gets worse with increasing distance to mean shape.

4.2.2 Comparison with MoMoPRT: Furthermore, we compare our method with *MoMoPRT* [Schneider et al. 2017] and show that DeepPRT it is more accurate and can handle more general deformations. Here, Schneider et al [2017] proposed a linear model f_{lin} to predict transfer coefficients from shape parameters of a *morphable model*.

Clearly, MoMoPRT is limited to shape deformations that are contained within the space described by a linear-shape-model S_{lin} . Moreover, although a linear model may be enough to approximate self-shadowing effects of shapes that are close to the mean shape of the data distribution of the training set, the model lacks complexity to accurately approximate data samples existing farther away from the mean shape (under-fitting).

On the other hand, our more complex non-linear model (f_{CNN}) is able to capture the relationships between the dataset's features (shape) and the target variable (transfer coefficients), enabling accurate approximations for a much larger deformation domain.

For demonstration purposes, we generate a new training set consisting of 500 shapes resulting from linear combinations between visually more dissimilar basis shapes¹: 1) the *Pirate Head* on one side, and 2) a simple *Plane* on the other.

$$S_{lin} = \alpha S_{pirate} + (1 - \alpha) S_{plane}$$

We train both models, f_{CNN} and f_{lin} , and compute their predictions for a series of test-shapes that are evenly distributed. Figure 6a shows that the prediction accuracy of our f_{CNN} model is higher and remains almost constant over the entire domain; on the contrary, the prediction accuracy of the linear model f_{lin} drops significantly moving away from the mean shape (the *Pirate/Plane* hybrid), as expected.

Last but not least, we demonstrate that our model approximates data that is contained in a much larger domain than the one spanned by a linear-shape-model S_{lin} . The models, f_{CNN} and f_{lin} , are fed with a series of sample shapes, starting from the mean shape and increasingly deforming towards a pirate face expression that was excluded from the training set.

As can be seen in figure 6b, the linear model performs poorly away from S_{lin} , while our network's accuracy remains constant.

5 CONCLUSION AND FUTURE WORK

We present a compact representation of PRT for **free form** deformable objects, informing of a non-linear model, namely a deep convolutional neural network, with significant memory savings demonstrated. We demonstrate that DeepPRT is able to produce good quality appearance estimations for deforming objects by only requiring the storage of the network parameters. In contrast, classic PRT methods require storing every single light-transport-state for each deformation pose, which in practice becomes unfeasible.

The model is able to extract the features of the dataset relevant to the light transfer function and thus generates good visual approximations for a wide spectrum of deformations, with resulting appearances that are practically indistinguishable from the ground truth.

Moreover, our method shows much higher generalization properties than previous approaches, allowing deformations from much larger and less constrained deformation spaces.

¹More distinguishable between each other, than between each face expressions used in [Blanz and Vetter 1999].

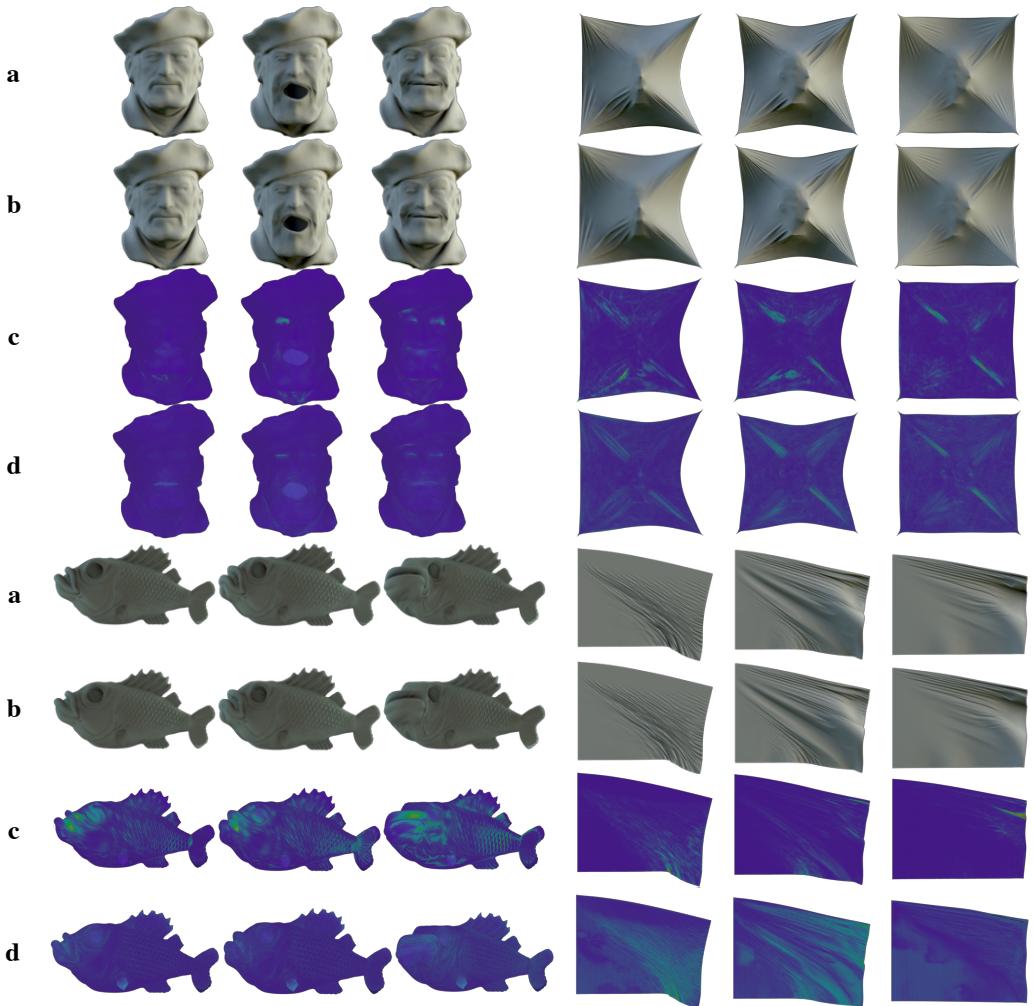


Fig. 7. Prediction Quality: a) ground truth appearance. b) predicted appearance. c) SSIM. d) L1-Error between ground truth and predicted transfer coefficients

Although we showed DeepPRT can be much more compact than traditional PRT, our network is far from being fully optimized. Exploiting network compression and acceleration methods could be highly beneficial [Cheng et al. 2017; Han et al. 2015]. The memory requirements of a network can be reduced by using these network compression algorithms, without loosing accuracy, further reducing the storage requirements and speeding-up the inference time eventually enabling PRT to be applicable on deformable objects for real-time applications.

Moreover, other network topologies and/or cost functions could be explored in order to achieve better approximations, for instance, as proposed by Man et al. [2018] to account for individual object deformations the use of deformable convolutions [Dai et al. 2017] could be investigated. Further to the transfer function representation, a direct learning of more bespoke and compact representations [Sloan and Silvennoinen 2018] could also be applied in this context, such as sparse SH vectors or a CPCA-like compressed basis.

The particular choice of our basis functions (Spherical Harmonics), currently restricts our method to low-frequency lighting. However, an extension to all-frequencies is possible by fitting the model to an alternative representation of the transfer function T , such as non-linear Wavelets [Ng et al. 2003].

The receptive field is a factor that can contribute to the performance of the CNN. Our network is adapted to the input image resolution with the intention to account for global feature extraction. Our particular network is deep enough to account for a receptive field that is almost the size of the input image of 256x256. By using other mappings (instead of the HM) such as the geometric-stretch parametrization, more flexible image resolutions can be used and possibly reduce the numbers of necessary layers. We assume and observe that geometry images, as experimented in this work, appropriately expose intrinsic properties of the surface deformations making the learning process more adept and due to the smoothness of the images more suitable for global feature extractions. However, a more explicit study follows regarding this assumption in evaluating performance of distant occluders and other distant or dislocated fully global illumination events.

The most significant further area of investigation for our method resides within the choice of our parametrization (harmonic map). In our tests, harmonic mapping performs well for modest curvature variations and is restricted to surfaces with one boundary (topological disks); however, using the geometric-stretch parametrization instead and further using the extension proposed by Praun and Hoppe [2003], the representation would be more accurate in the reconstruction of shape. The U-Net process in this context of sharp feature reconstruction and non-disk topologies would need additional measures. In particular, a treatment of the *diffeomorphic* behavior [Detlefsen et al. 2018] of these mappings appropriate to efficient deep learning of light transport for animating arbitrary surfaces would be interesting to further understand the representation trade-offs.

ACKNOWLEDGMENTS

We would like to thank the anonymous reviewers for the insightful comments that greatly helped improve this manuscript. Also, Peter-Pike Sloan and Llogari Casas Cambra for corrections and assistance in the runtime rendered results preparation. This project has received funding from the European Union Horizon 2020 research and innovation program under the Marie Skłodowska-Curie grant agreement No. 642841.

REFERENCES

- Volker Blanz and Thomas Vetter. 1999. A Morphable Model for the Synthesis of 3D Faces. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '99)*. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 187–194. <https://doi.org/10.1145/311535.311556>
- Davide Boscaini, Jonathan Masci, Emanuele Rodolà, and Michael M. Bronstein. 2016. Learning shape correspondence with anisotropic convolutional neural networks. *CoRR* abs/1605.06437 (2016). arXiv:1605.06437 <http://arxiv.org/abs/1605.06437>
- Michael M. Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. 2016. Geometric deep learning: going beyond Euclidean data. *CoRR* abs/1611.08097 (2016). arXiv:1611.08097 <http://arxiv.org/abs/1611.08097>
- Chakravarty R, Alla Chaitanya, Anton S. Kaplanyan, Christoph Schied, Marco Salvi, Aaron Lefohn, Derek Nowrouzezahrai, and Timo Aila. 2017. Interactive Reconstruction of Monte Carlo Image Sequences Using a Recurrent Denoising Autoencoder. *ACM Trans. Graph.* 36, 4, Article 98 (July 2017), 12 pages. <https://doi.org/10.1145/3072959.3073601>
- Yu Cheng, Duo Wang, Pan Zhou, and Tao Zhang. 2017. A Survey of Model Compression and Acceleration for Deep Neural Networks. *CoRR* abs/1710.09282 (2017). arXiv:1710.09282 <http://arxiv.org/abs/1710.09282>
- François Chollet et al. 2015. Keras. <https://keras.io>.
- Michael F. Cohen, John Wallace, and Pat Hanrahan. 1993. *Radiosity and Realistic Image Synthesis*. Academic Press Professional, Inc., San Diego, CA, USA.
- Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. 2017. Deformable Convolutional Networks. *CoRR* abs/1703.06211 (2017). arXiv:1703.06211 <http://arxiv.org/abs/1703.06211>
- Nicki Skafte Detlefsen, Oren Freifeld, and Søren Hauberg. 2018. Deep Diffeomorphic Transformer Networks. *Conference on Computer Vision and Pattern Recognition (CVPR)* (2018).

- James Eells and J. H. Sampson. 1964. *Harmonic mappings of Riemannian manifolds*. The Johns Hopkins University Press. 86:109–160 pages.
- Wei-Wen Feng, Liang Peng, Yuntao Jia, and Yizhou Yu. 2007. Large-scale Data Management for PRT-based Real-time Rendering of Dynamically Skinned Models. In *Proceedings of the 18th Eurographics Conference on Rendering Techniques (EGSR'07)*. Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 23–34. <https://doi.org/10.2312/EGWR/EGSR07/023-034>
- David Gu. 2018. Harmonic Map. <https://www3.cs.stonybrook.edu/~gu/tutorial/HarmonicMap.html> Accessed: 28-11-2018.
- Xianfeng Gu, Steven J Gortler, and Hugues Hoppe. 2002. Geometry images. *ACM Transactions on Graphics (TOG)* 21, 3 (2002), 355–361.
- Song Han, Huizi Mao, and William J. Dally. 2015. Deep Compression: Compressing Deep Neural Network with Pruning, Trained Quantization and Huffman Coding. *CoRR* abs/1510.00149 (2015). arXiv:1510.00149 <http://arxiv.org/abs/1510.00149>
- Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep Residual Learning for Image Recognition. *CoRR* abs/1512.03385 (2015). arXiv:1512.03385 <http://arxiv.org/abs/1512.03385>
- Pedro Hermosilla, Sebastian Maisch, Tobias Ritschel, and Timo Ropinski. 2018. Deep-learning the Latent Space of Light Transport. *arXiv* abs/1811.04756 (2018).
- M. Horowitz. 2014. 1.1 Computing’s energy problem (and what we can do about it). In *2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)* 10–14. <https://doi.org/10.1109/ISSCC.2014.6757323>
- Doug L. James and Kayvon Fatahalian. 2003. Precomputing Interactive Dynamic Deformable Scenes. *ACM Trans. Graph.* 22, 3 (July 2003), 879–887. <https://doi.org/10.1145/882262.882359>
- A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. 2014. Large-Scale Video Classification with Convolutional Neural Networks. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*. 1725–1732. <https://doi.org/10.1109/CVPR.2014.223>
- Jan Kautz, Peter-Pike Sloan, and John Snyder. 2002. Fast, Arbitrary BRDF Shading for Low-frequency Lighting Using Spherical Harmonics. In *Proceedings of the 13th Eurographics Workshop on Rendering (EGRW’02)*. Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 291–296. <http://dl.acm.org/citation.cfm?id=581896.581934>
- Diederik P. Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. *CoRR* abs/1412.6980 (2014). arXiv:1412.6980 <http://arxiv.org/abs/1412.6980>
- Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2012. ImageNet Classification with Deep Convolutional Neural Networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1 (NIPS’12)*. Curran Associates Inc., USA, 1097–1105. <http://dl.acm.org/citation.cfm?id=2999134.2999257>
- Yann LeCun, Yoshua Bengio, and Geoffrey E. Hinton. 2015. Deep learning. *Nature* 521, 7553 (2015), 436–444. <https://doi.org/10.1038/nature14539>
- Yunze Man, Yangsibo Huang, Junyi Feng, Xi Li, and Fei Wu. 2018. Deep Q Learning Driven CT Pancreas Segmentation with Geometry-Aware U-Net.
- Haggai Maron, Meirav Galun, Noam Aigerman, Miri Trope, Nadav Dym, Ersin Yumer, Vladimir G. Kim, and Yaron Lipman. 2017. Convolutional Neural Networks on Surfaces via Seamless Toric Covers. *ACM Trans. Graph.* 36, 4, Article 71 (July 2017), 10 pages. <https://doi.org/10.1145/3072959.3073616>
- Jonathan Masci, Davide Boscaini, Michael M. Bronstein, and Pierre Vandergheynst. 2015. ShapeNet: Convolutional Neural Networks on Non-Euclidean Manifolds. *CoRR* abs/1501.06297 (2015). arXiv:1501.06297 <http://arxiv.org/abs/1501.06297>
- Jonathan Masci, Emanuele Rodolà, Davide Boscaini, Michael M. Bronstein, and Hao Li. 2016. Geometric Deep Learning. In *SIGGRAPH ASIA 2016 Courses (SA ’16)*. ACM, New York, NY, USA, Article 1, 50 pages. <https://doi.org/10.1145/2988458.2988485>
- Federico Monti. 2018. Geometric Deep Learning. <http://geometricdeeplearning.com/> Accessed: 26-11-2018.
- Oliver Nalbach, Elena Arabadzhyska, Dushyant Mehta, Hans-Peter Seidel, and Tobias Ritschel. 2017. Deep Shading: Convolutional Neural Networks for Screen-Space Shading. 36, 4 (2017).
- Ren Ng, Ravi Ramamoorthi, and Pat Hanrahan. 2003. All-frequency Shadows Using Non-linear Wavelet Lighting Approximation. *ACM Trans. Graph.* 22, 3 (July 2003), 376–381. <https://doi.org/10.1145/882262.882280>
- Emil Praun and Hugues Hoppe. 2003. Spherical Parametrization and Remeshing. *ACM Trans. Graph.* 22, 3 (July 2003), 340–349. <https://doi.org/10.1145/882262.882274>
- Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. *CoRR* abs/1505.04597 (2015). arXiv:1505.04597 <http://arxiv.org/abs/1505.04597>
- A. Schneider, S. Schünborn, B. Egger, L. Froben, and T. Vetter. 2017. Efficient Global Illumination for Morphable Models. In *2017 IEEE International Conference on Computer Vision (ICCV)*. 3885–3893. <https://doi.org/10.1109/ICCV.2017.417>
- Ayan Sinha, Jing Bai, and Karthik Ramani. 2016. Deep learning 3D shape surfaces using geometry images. In *European Conference on Computer Vision*. Springer, 223–240.
- Peter-Pike Sloan, Jan Kautz, and John Snyder. 2002. Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments. In *ACM Transactions on Graphics (TOG)*, Vol. 21. ACM, 527–536.

- Peter-Pike Sloan, Ben Luna, and John Snyder. 2005. Local, Deformable Precomputed Radiance Transfer. ACM, 1216–1224. <https://www.microsoft.com/en-us/research/publication/local-deformable-precomputed-radiance-transfer/>
- Peter-Pike Sloan and Ari Silvennoinen. 2018. Directional Lightmap Encoding Insights. In *SIGGRAPH Asia 2018 Technical Briefs (SA '18)*. ACM, New York, NY, USA, Article 12, 3 pages. <https://doi.org/10.1145/3283254.3283281>
- Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin A. Riedmiller. 2014. Striving for Simplicity: The All Convolutional Net. *CoRR* abs/1412.6806 (2014). arXiv:1412.6806 <http://arxiv.org/abs/1412.6806>
- Manu Mathew Thomas and Angus Graeme Forbes. 2017. Deep Illumination: Approximating Dynamic Global Illumination with Generative Adversarial Network. *CoRR* abs/1710.09834 (2017). arXiv:1710.09834 <http://arxiv.org/abs/1710.09834>
- Zhirong Wu, S. Song, A. Khosla, Fisher Yu, Linguang Zhang, Xiaou Tang, and J. Xiao. 2015. 3D ShapeNets: A deep representation for volumetric shapes. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 1912–1920. <https://doi.org/10.1109/CVPR.2015.7298801>