

Deep Precomputed Radiance Transfer for Deformable Objects

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

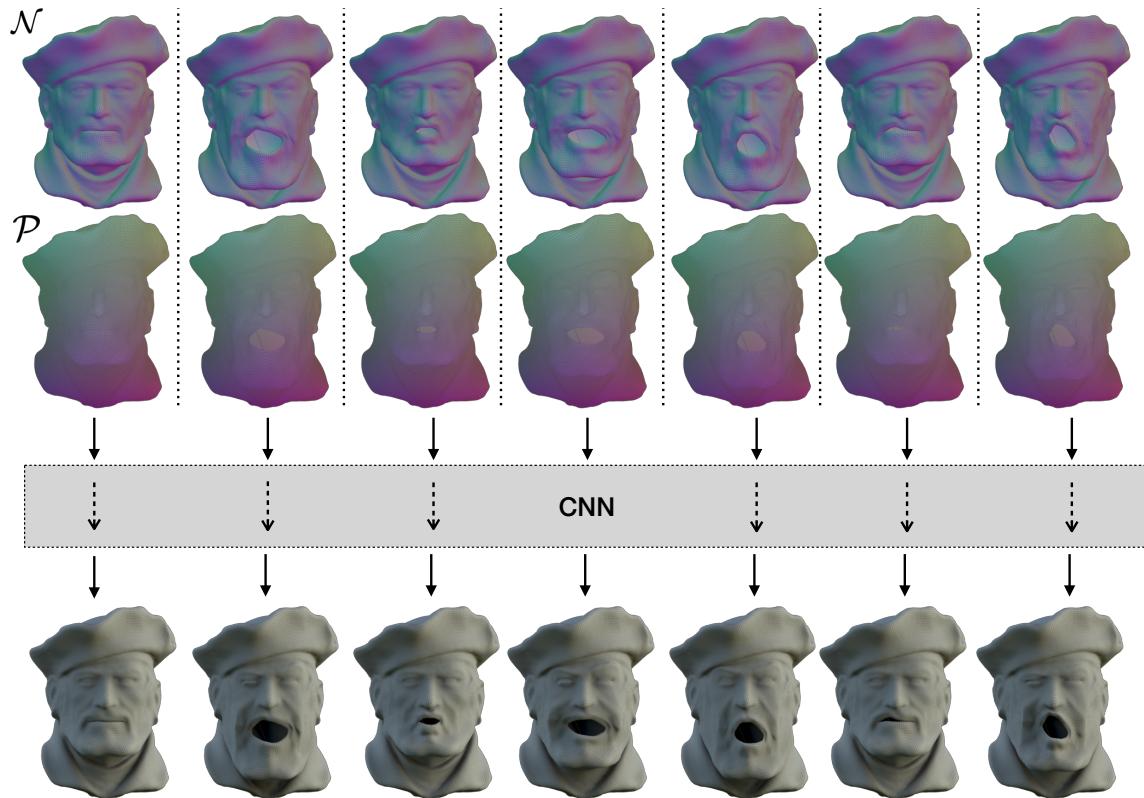


Figure 1: TODO

ABSTRACT

TODO:

(Written by Yue Li)

Traditional Precomputed Radiance Transfer methods require a lot of memory to save the precomputed data for real time rendering. For an animation sequence such data would be gigantic. We proposed a deep learning precomputed radiance transfer framework (DPRT) for deforming object, saving only the weights of the network, which lowers the memory cost in orders of magnitudes. Object is first parameterized

via harmonic mapping and reconstructed to form geometry and normal image as the inputs of a carefully designed fully convolutional network.

CCS CONCEPTS

- Computing methodologies → Rendering: Ray tracing;

KEYWORDS

ray tracing, global illumination, octrees, quadtrees

ACM Reference Format:

Yue Li, Pablo Wiedemann, and Kenny Mitchell. 2018. Deep Precomputed Radiance Transfer for Deformable Objects. In *Proceedings of Interactive 3D Graphics and Games*. ACM, New York, NY, USA, 8 pages. <https://doi.org/10.1145/888888.777777>

Permission to make digital or hard copies of part or all 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 third-party components of this work must be honored. For all other uses, contact the owner/author(s).

Interactive 3D Graphics and Games, 2019, Montreal

© 2018 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-1234-5/17/07.

<https://doi.org/10.1145/888888.777777>

1 INTRODUCTION

Rendering photo-realistic appearances entails solving the *rendering equation* for each point on an objects 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 addressed to overcome this computational overkill, simplifying the rendering equation but still enabling high-quality renderings for complex illuminations. The quintessence is to perform a single pre-computation step of the light-transport information and only evaluate the equation at runtime.

Classic PRT algorithms function well for static scenes; however, these are destined to fail eyeing dynamic and/or interactive environments, in which considered objects undergo significant deformations. Responsible is a term in the rendering equation called the *transfer function* which is fully dependent on the shape of the surface. That is, any object deformation implies a re-computation of the *transfer function*, requiring expensive ray-casting. Hence, using classic PRT to render deformable objects would involve pre-computing large amounts of data, leading to immense storage consumptions. Hence, rapidly becoming unwieldy and inefficient since memory IO's operations can be 3 orders of magnitudes energy demanding than floating point summations or multiplications [Horowitz 2014]. On top of that, the costly 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. Examples of such are: interactive physically based deformations for cloth or soft-bodies [find references] ; or more recent developments in the field of automatic character animations involving on-the-fly pose adaptation [Bütepage et al. 2017; Holden et al. 2017; Zhang et al. 2018].

We propose a Deep Learning framework addressed to overcome the limitations of traditional PRT algorithms described above. In particular, we replace expensive ray-casting 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, enabling a compact PRT representation that maintains a constant/fix storage consumption, regardless of the number of deformations. Moreover, due to the inherent generalisation capabilities of DNN's, our method is able to accurately predict appearances of previously unknown shapes. We call this compact self-shadowing technique *Deep Precomputed Radiance Transfer* (DPRT).

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. Here, basic operations, such as the convolution, are not well defined being a major impediment for Deep Learning (DL) to fully flourish in this particular field. Nonetheless, more recently some authors have started to address the paradigm of DL on non-Euclidean data proposing a variety of approaches [Bronstein et al. 2016; Maron et al. 2017; ?] (for further reading: [Monti [n. d.]]).

In particular, we propose learning on *geometry images*, a parametrisation proposed by [Gu et al. 2002] and further explored within the DL context by [?].

The main contributions of our approach, is an extension of PRT that:

- enables accurate appearance predictions of more general and adaptive deformations than previous approaches,
- while maintaining a much smaller and compact representation.

2 RELATED WORK

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 data sets that expand in proportion to the number of poses; hence, rapidly becoming unwieldy for such applications.

Our aim is to extend traditional PRT to arbitrary 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 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 global self-shadowing effects. 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 decompression and interpolation can be performed efficiently.

An appearance model, that approximates PRT lighting, is presented in [James and Fatahalian 2003]. The model is based on a reduced state space of deformable shapes that allows only very limited kind 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 show 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 generalisation properties, namely a deep Convolutional Neural Network [Karpathy et al. 2014; Krizhevsky et al. 2012; LeCun et al. 2015].

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] 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. Alternatively, we propose directly learning on geometric data. However, learning on surfaces using CNN's 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 CNN's 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 CNN's [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 CNN's 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 Image*, 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 extend 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 DPRT METHOD

Classic PRT is a physically-based rendering method to accelerate on-line computations of the (simplified) *Rendering Equation*:

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

where L_e 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.

It 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_e 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_e 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) bases to encode the Transfer Function T and the light environment L_e .

As mentioned above, our aim is to extend the PRT method to malleable and dynamic objects, but avoiding costly pre-computations 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 , more precisely the coefficients of its SH-encoding t_j 's, for any given shape query S_i . This makes the costly ray-casting computations superfluous and solves the abusive memory requirements, only necessitating the storage of the network's parameters.

3.1 Data: Geometry Images

We propose learning directly on the object's surface in order to leverage its underlying shape structure. *Geometry Images* present an planar 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 resampled) into a regular grid of $n \times n$ vertices. For simplicity, but without loss of generality of our method, we chose a *Harmonic Map*, based on [Gu [n. d.]], for the parametrisation of the interior of the 2D-grid.

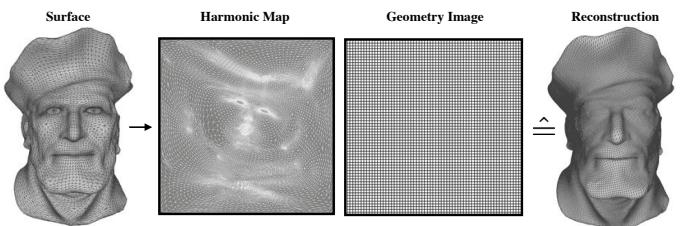


Figure 2: GeoImage

It is to note that we apply deformations only on the reconstructed object (right shape of Figure (2)) in order to make our shape representation, *Geometry Images*, invariant to deformations. By doing so, we maintain a one-to-one pixel correspondence; hence, filtering out deformation invariant information of the surface; and therefore, facilitating the feature extraction of surface properties that are more correlated to self-shadowing.

The surface information we transform into *Geometry Images* to use as regressor for the CNN are: vertex positions \mathcal{P} and normals \mathcal{N} .

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

with $P_i, N_i \in \mathbb{R}^{n \times n}$ being the position and normal images, respectively, for each coordinate $i \in \{x, y, z\}$.

Resulting, our CNN model predicts a corresponding sequence of *Geometry Images* \mathcal{T} ,

$$f_{CNN}(\mathcal{P}, \mathcal{N}) = \mathcal{T}$$

consisting of the SH-coefficients of the *Transfer Function* of the input shape, as introduced above (see eq. 2):

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

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

Figure 3 illustrates the basic procedure of the method.

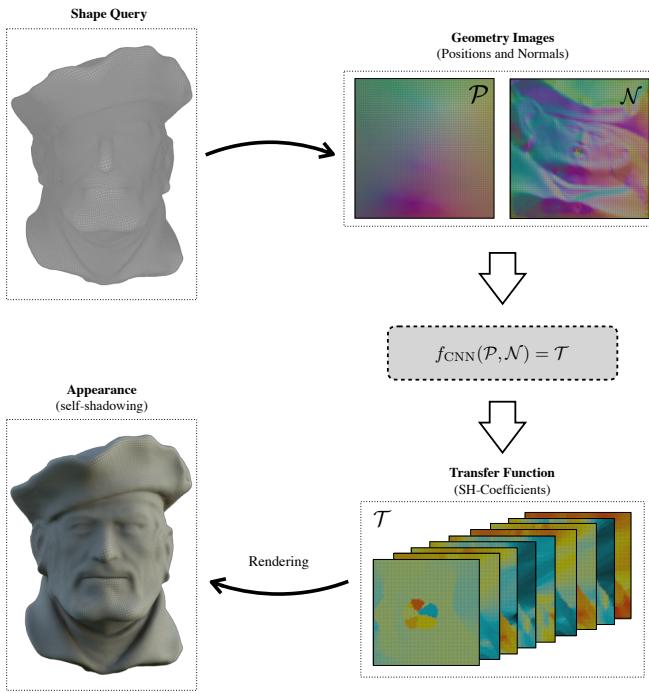


Figure 3: Method Overview

3.2 Network Architecture and Training

In order to DPRT to function for real-time rendering scenarios, it requires very fast network inference times. However, to achieve high accuracies, existing deep convolutional network models rely on very deep architectures with millions of parameters, consequently being computationally expensive and memory intensive, and therefore becoming impractical for real-time application.

However, 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 to real-time environments [Cheng et al. 2017; Han et al. 2015].

However, the topic of *network optimisation* reaches beyond the scope of this work. Here, we will make use of a classic deep CNN

architecture to demonstrate the main principles of our method. In section [??] we show that even neglecting network compression our DPRT-approach already achieves immense memory savings.

Architecture:

The topology of our deep convolutional network consists of an encoder and decoder with skip connections based on [Ronneberger et al. 2015]. Both encoder and decoder consist of sequences of ResNet blocks [He et al. 2015] each comprising a series of 2D-Convolution, Batch Normalisation, Down-sampling and ReLU - Activation layers (illustrated in Figure (4)). 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, which passes outputs of encoding layers to the respective inputs of the decoding layers. The network has an approximate amount of $1,1 \cdot 10^7$ parameters.

Synthesis of Training Data :

For a given object, we generate the training data by applying sequences of smooth deformations, obtained by a physically based or blendshape based animation (see Section ?? for examples), each of a total length of 500 frames.

For each frame $i \in [1, 2, \dots, 500]$ we store the position \mathcal{P}_i and normal \mathcal{N}_i images, and perform a full self-shadowing integration using ray-casting to compute and store the corresponding coefficients of the *Transfer Function* \mathcal{T}_i (ground truth).

For most objects we chose an image resolution of 256×256 .

Training:

The network is trained on 450 samples, each consisting of six image channels of size 256×256 . 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, using a batch size of 5. The network is implemented in Keras [Chollet et al. 2015] with Tensorflow as Backend. On a high-end GPU (NVIDIA GeForce GTX 2080) this takes around... (TO CHECK!)

4 EXPERIMENTS AND RESULTS

We test our DPRT method against different animated objects, see Figure (5). We animate a *Pirate Head* and a *Fish* object using linear blend-shapes and apply physics-based deformations to animate the *Cloth* object.

For the *Pirate Head* and the *Cloth* objects we chose a reconstruction resolution of 256×256 and for the *Fish* object of 512×512 .

Memory Savings

For each of the test objects, our training sets comprises 450 distinct object deformations.

Now, for demonstration purposes, lets assume the shapes of this training set are precisely the target shapes which appearances are requested to be computed for a given application.

Within that training set, our CNN model is able to achieve accuracies up to 98% (table 1) and its rendered appearances become

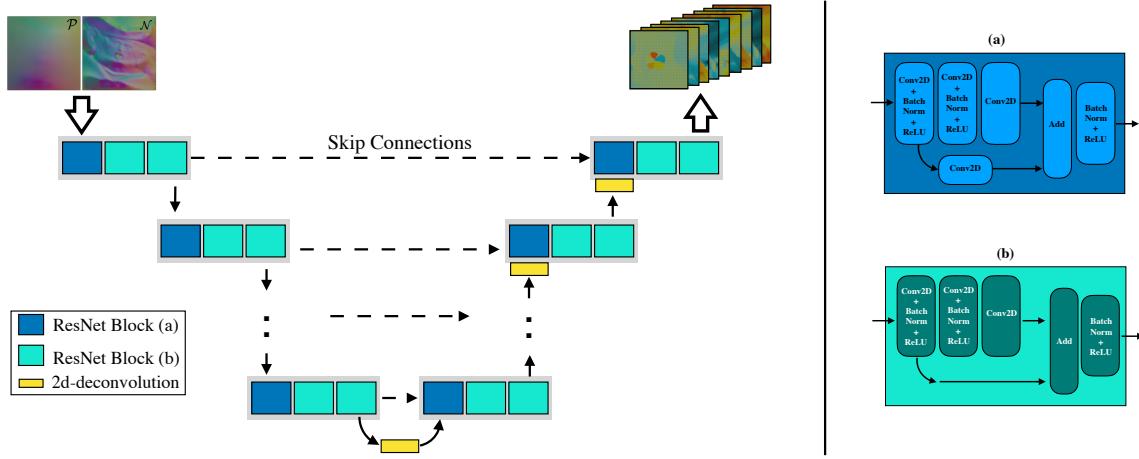


Figure 4: Network Topology

visually indistinguishable from the ground truth (Figure (5)). This means, that our trained network is able to faithfully generate self-shadowing effects of 450 distinct shapes **at the very least**, only requiring the storage of the network parameters.

On the contrary classic PRT would imply storing the transfer coefficients of each vertex for every single shape of the training set. 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 of:

$$r = \frac{\# \text{ PRT-params}}{\# \text{ CNN-params}} = \begin{cases} 40,022, & \text{for } 256^2 \# \text{vertices} \\ 160,088 & \text{for } 512^2 \# \text{vertices} \end{cases}$$

This number grows linearly with increasing number of coefficients, number of deformations or number vertices.

	Accuracy	Loss	Val-Loss
Pirate Head	0.9817	0.000397	0.000399
Fish	0.9729	0.002104	0.002200
Cloth	0.9818	0.000786	0.000907

Table 1: Network Accuracy

Remark: the numbers shown above express the compression ratio taking into account solely the training set. On top of that, our network shows good generalisation properties, also enabling accurate and qualitatively precise appearance predictions of deformations outside the training set. Thus, by taking this into account the compression ratio grows to immeasurable values.

Thus, we show that DPRT, as it is, is capable of drastically diminishing the storage consumption of PRT algorithms for deforming objects.

As mentioned above, commonly deep convolutional networks itself can be highly optimised, hence making DPRT even much more efficient, energy, speed and memory wise [Cheng et al. 2017].

Generality of DPRT and Comparison

We validate the generalisation capabilities of our model by standard machine learning procedures. We base our parameter tuning on minimising the validation loss and later analyse prediction quality of using a test set.

The results show small errors and are in most cases visually indistinguishable or close to the ground truth (Figure [??]).



Figure 6: Example: glossy pirate

Comparison with MoMoPRT: Furthermore, we compare our method with [Schneider et al. 2017] (MoMoPRT) and show that DPRT it is more accurate and can handle more general deformations.

The authors of [Schneider et al. 2017] proposed a linear model f_{lin} to predict transfer coefficients within a "linear shape space", namely the space spanned by a Morphable Model.

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

On the other hand, our more complex non-linear CNN 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, randomly sampled, linear combinations between visually more dissimilar basis shapes¹: 1) the *Pirate Head* on one side, and 2) a simple *Plane* on the other. We train both models, f_{CNN} and f_{lin} , and compute their predictions for a series of test-shapes that are evenly distributed over the linear shape space. Figure (7) 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.

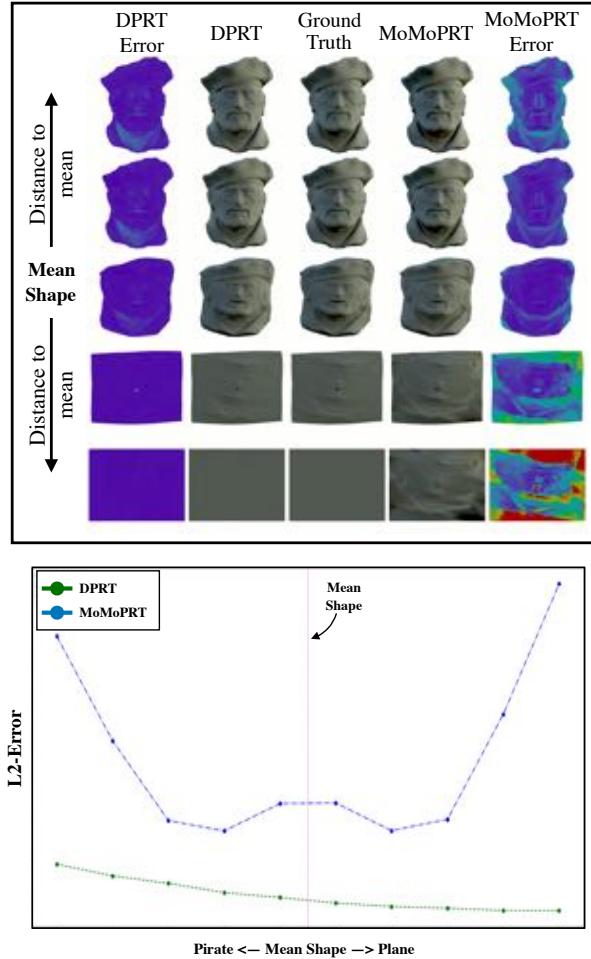


Figure 7: Example: DPRT vs MoMoPRT

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 (8).

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

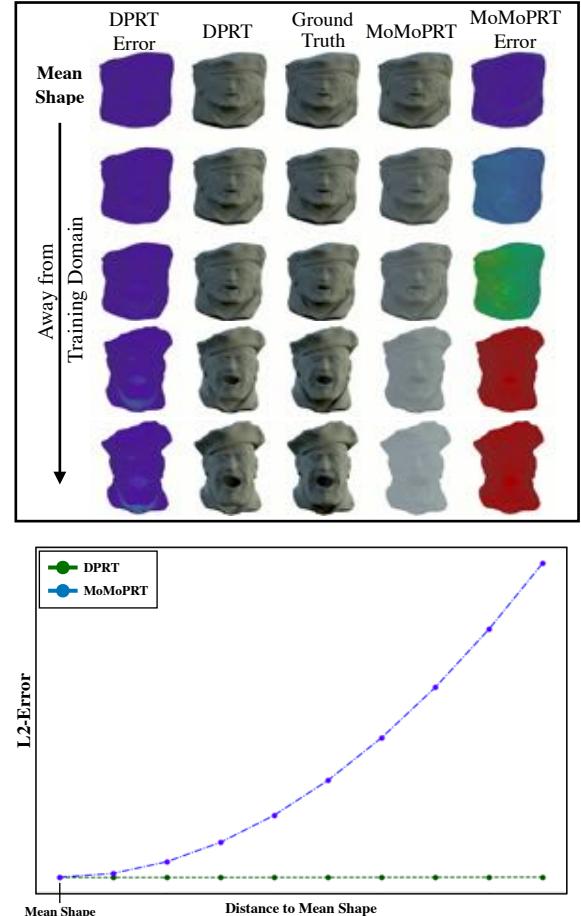


Figure 8: Example: DPRT vs MoMoPRT

5 CONCLUSION AND FUTURE WORK

We present a compact representation of PRT for deformable objects by introducing a non-linear function, a Convolutional Neural Network (CNN), to approximate the *Transfer Function* for a given shape query. As a result, our proposed CNN is able to make accurate approximations generating appearances that are visually undistinguishable from the ground truth. Moreover, our method shows much higher generalisation properties than previous approaches allowing deformations from a much larger and less constrained deformation space.

The particular choice of our basis functions (*Spherical Harmonics*), currently restricts our method to low-frequency lightings. However, an extension to all-frequencies is straight forward and can be made by fitting the model to an alternative representation of T , such as non-linear Wavelets [Ng et al. 2003].

The most significant restrictions of our method reside within the natural flaws of *Geometry Images*. Currently, our algorithm can only operate on surfaces containing one boundary and performs well for modest curvature variations. In future, alternative surface representations could be explored to overcome this restrictions.

- Geo.image resolution limited to powers of 2 (256 , 512, ...)
- Harmonic map -> stretch min. map

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>
- Judith Bütepage, Michael J. Black, Danica Kragic, and Hedvig Kjellström. 2017. Deep representation learning for human motion prediction and classification. *CoRR* abs/1702.07486 (2017). arXiv:1702.07486 <http://arxiv.org/abs/1702.07486>
- 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.
- 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. [n. d.]. 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>
- Daniel Holden, Taku Komura, and Jun Saito. 2017. Phase-functioned Neural Networks for Character Control. *ACM Trans. Graph.* 36, 4, Article 42 (July 2017), 13 pages. <https://doi.org/10.1145/3072959.3073663>
- 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>
- 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>
- 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. [n. d.]. 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. Schäufelin, B. Egger, L. Frobenius, 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/>
- 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, Xiaoou 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>
- He Zhang, Sebastian Starke, Taku Komura, and Jun Saito. 2018. Mode-adaptive Neural Networks for Quadruped Motion Control. *ACM Trans. Graph.* 37, 4, Article 145 (July 2018), 11 pages. <https://doi.org/10.1145/3197517.3201366>

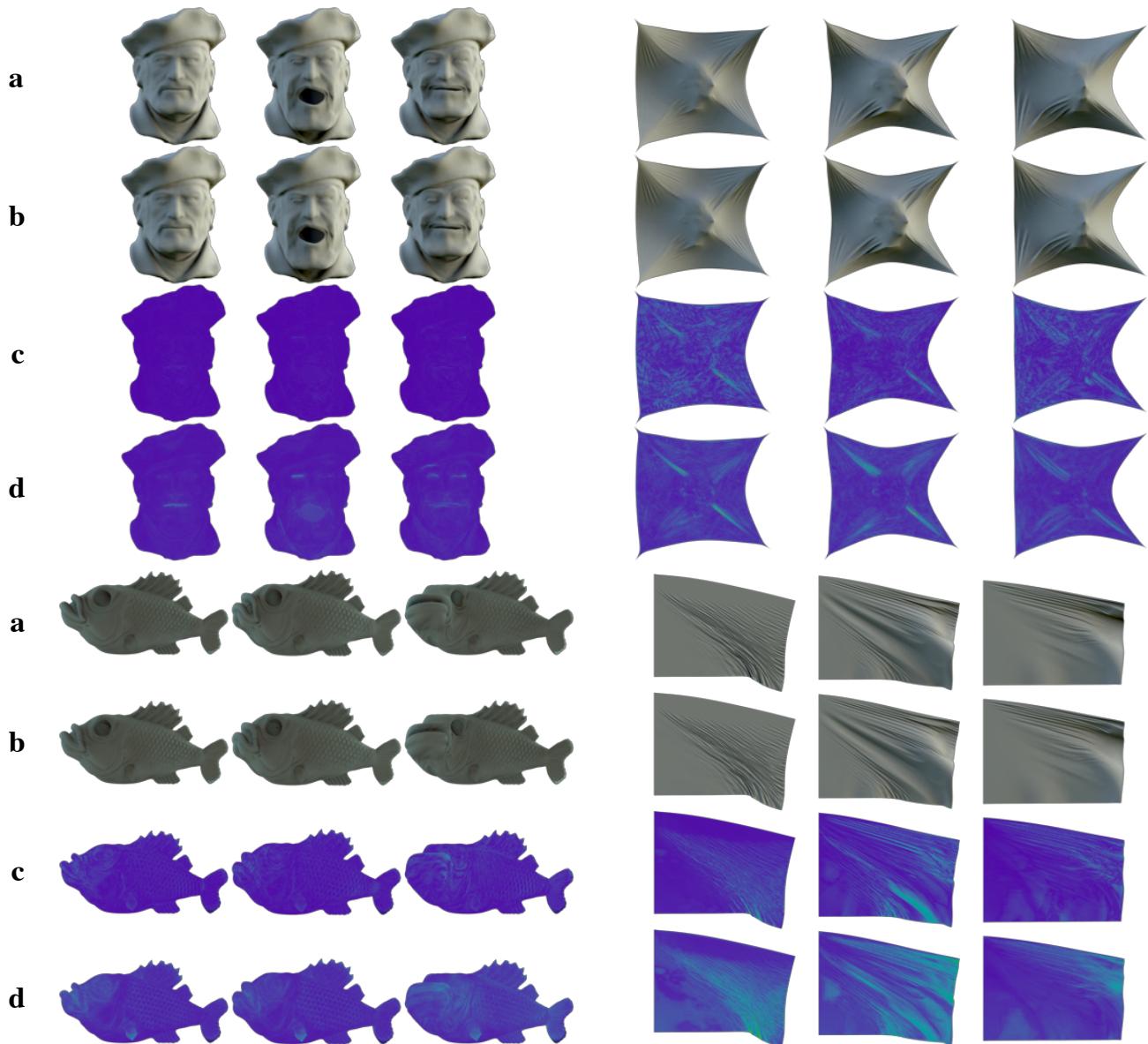


Figure 5: Visual Quality: a : Ground truth appearance. b: CNN appearance prediction. c: L2-Error between ground truth RGB-colors and prediction RGB-colors. d: L1-Error between ground truth and predicted transfer coefficients