A comparison of two machine learning approaches for Photometric Solids Compression

Samuel Delepoulle[1] François Rousselle[1] Christophe Renaud[1]
Philippe Preux[2]

[1]{delepoulle,rousselle,renaud}@lisic.univ-littoral.fr
Université Lille- Nord de France
ULCO
LISIC - BP 719 - 62228 Calais cedex
[2]philippe.preux@inria.fr
Université de Lille
Centre de Recherche INRIA-Lille Nord Europe
& Laboratoire d'Informatique Fondamentale de Lille (LIFL, UMR CNRS 8022)
Parc Scientifique de la Haute Borne 40, avenue Halley, 59650 Villeneuve d'Ascq

Abstract

The use of photometric solids into both real time and photorealistic rendering allows designers and computer artists to enhance easily the quality of their images. Lots of such data are available from lighting societies since they allow these societies to easily present the luminance distribution of their often complex ligthing systems. When accuracy is required the amount of discretized luminance directions and the number of photometric solids that have to be used increase considerably the memory requirements and reduce the algorithm efficiency. In this paper we describe and compare two machine learning approaches used for approximating any photometric solid: an artificial neural network and ECON (Equi-Correlated Network Algorithm). By applying these two approaches on a large set of real photometric distribution data, we were able to show that one of them provides generally a better approximation of the original distribution.

keywords: approximation, image render, machine learning, photometric solids, light distribution.

1 Introduction

Computing realistic lighting effects for image synthesis requires to model accurately the light propagation and the interaction of light with the object materials. Another important feature is an accurate model of the source light distribution since it determines all the lighting simulation. This can be done by modeling in depth the numerous parts of any lighting system and then to use this model in the lighting simulation. The main drawback of such an approach is that modeling is often complex and simulation times increase according to the details the source is modeled with. this approach is obviously not really usable for any real-time rendering software.

One way to approximate the main features of the light distribution of any lighting system is to use a discretized representation of the photometric distribution of this system: measurements are performed for a set of directions around the center of the lighting system which is approximated as a point light source. Assuming a large distance between any illuminated point and the source, interpolating between discrete photometric data can then provide a useful approximation of the source light distribution. This light distribution is known as a photometric solid (PS) (see figure 1 for examples) and is generally stored in a standardized representation known as the IES file format [IES95].

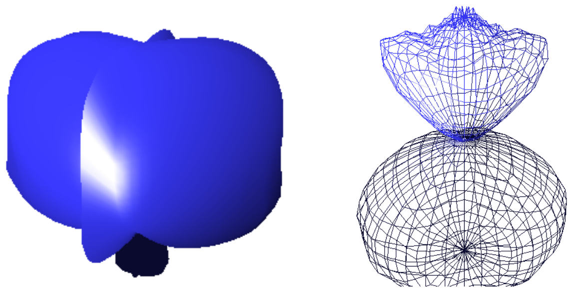


Figure 1: A sketch of two photometric solids built from data extracted from the Metalumen photometric data files library [Met].

When accuracy is required in a rendering software, a large number of measurements have to be performed and the size of the IES file increases considerably. Then when large light sources are approximated or for close to source illuminated points, the point light source approximation becomes false and results are far from being accurate. A workaround is to split the source area into small sub-sources, each one having its own PS [Den02]. Obviously, this increases both memory requirements and computation times. Using a large number of accurate photometric solids thus involves finding a way of reducing their memory consumption and their computation time overhead. Up to our knowledge, compressing PS has not yet been much studied. Lots of works have been done in some related problems such as compressing Bidirectionnal Reflectance Distribution functions (BRDF) [SAWG91][CPB06][NP00], and for approximating 3D shapes out of large point clouds [WSC06]. In the latter, the goal is not the same as ours since the purpose is to allow the shape to be rendered through its new representation using potentially cost effective algorithms. For the former, Deniel [Den02] showed that the classical approaches for its resolution are not well-suited for photometric solids compression. He then proposed a hierarchical adaptive compression model that efficiently compress the PS and provides a fast access to the directionnal luminance data. In previous works, we studied the use of machine learning methods [DRP09][LPDR09] for PS compression. The goal was to use these methods in order to learn the "shape" of any PS and then to be able to aproximate it with very few basis function. The outcome of this work was to compress efficiently the PS and to be able to efficiently generate the luminance emitted from the source along any direction. In this paper, we are interested in comparing our two previous approaches on a large data set of PS. This allows us to provide a better analysis of their advantages and drawbacks.

In the next section, we provide a brief overview of the machine learning techniques that are suitable to deal with the problem at hand, and describe more precisely the two methods we compare. Our methodology is detailed in section 3, and our results are presented in section 4.

2 Machine learning approach

The general idea is to use supervised learning to produce a regression model. The regression problem may be summarized as follows: we have a set of N examples (x_i, y_i) with $i \in \{1, \dots, N\}, x_i \in \mathscr{D} \subset \mathbb{R}^P, y_i \in \mathbb{R}$; y_i is assumed to be a noisy realization of an unknown function y, that is: $y_i \equiv y(x_i) + \epsilon$ where ϵ denotes some noise. The goal is to learn, or estimate, a function $\hat{y}$ that approximates as well as possible y, and so that $\hat{y}$ may be further used to predict the value associated to any data $x \in \mathscr{D}$. $\hat{y}$ may also be considered as a model of the data, a model that is learned in order that the function y may subsequently be estimated for any data x in the domain. The model is obtained by induction, that is, from data, one derives a general rule. The regression problem has been studied for at least two centuries, and tens of thousands publications, and hundreds techniques and algorithms are known. The study of this problem is still a very active research area, with very significant challenges still pending. There has been very significant advances since the 1980's, with the development of multi-layer perceptrons, and more recently, the development of statistical learning which is deeply rooted in statistics, the theory of function approximation, and functional analysis.

2.1 Artificial neural networks

In this work, to learn a $\hat{y}$, we use a multi-layer perceptron (MLP), and we restrict ourselves to 1 hidden layer perceptrons [Hay08]. In this case, $\hat{y}$ has the form:

$$\hat{y}(x_.) = S_o(\sum_{k=1}^{k=K} w_k S_h(a_{j,k} x_{.,j})) \tag{1}$$

where S_o (resp. S_h) is the so-called activation function of the output (resp. hidden) unit, the w_k and $a_{j,k}$ are real weights, $x_{.,j}$ denotes the j^{th} component of the data $x_.$, and K is the number of hidden units. Here, both S_o, and S_h are sigmoid functions of the form $S_o(x) \equiv \frac{1}{(1+e^{-x})}$. Training an MLP means finding the "best" value for the K weights w_k and the $P \times K$ weights $a_{j,k}$. Please note that K is fixed *a priori*.

To learn these weights, we initially set them to some value (arbitrary or not), so that $\hat{y}$ may be computed for any data $x \in \mathscr{D}$. Basically, iteratively, one computes $\hat{y}(x_i)$ for some example x_i, and modifies the value of the weights according to the discrepancy between the expected value y_i, and the predicted value $\hat{y}(x_i)$. Various methods exist to achieve this, the most well-known being the backpropagation of error, which is a simple gradient descent algorithm.

2.2 ECON

Introduced in [LPDR09], the Equi-Correlated Network Algorithm (ECON for short) is considering models of the form:

$$\hat{y}(x) \equiv w_0 + \sum_{k=1}^{k=K} w_k \phi_k(x). \tag{2}$$

For each k, $\phi_k : \mathscr{D} \rightarrow \mathbb{R}$ is a feature function that maps a data to a real value; it may be any such function. Here again, the w_0 and the set of w_k are real weights. There are two points that come in sharp contrast with the MLP approach introduced above:

- K is not fixed: it is learned by ECON,
- generally, each feature function ϕ_k has some parameters; these parameters are also learned by ECON. For the sake of illustration, one may consider that ϕ_k is drawn from the family of multivariate Gaussian functions $\phi_k(x) = e^{-(x-\mu_k)^T C_k (x-\mu_k)}$, and the parameters for such a function are the center $\mu_k \in \mathscr{D}$, and the covariance matrix $C_k \in \mathbb{R}^{P \times P}$.

These two points add a great flexibility to ECON with regards to previous algorithms.

Thus, ECON learns a lot of parameters: the very number of parameters itself (K), $K + 1$ weights, and the parameters of each feature function. As such this is an ill-posed problem (in Hadamard's sense). To solve it, we adopt a principled way based on a parsimonious approach, which tends to make K as small as possible; particularly, we look for the optimal parameters for the smallest value of K. Probably quite surprisingly, there is an exact algorithm to obtain the solution of this problem, and this algorithm is very efficient, and effective. To this end, ECON is a generalization of the LARS algorithm introduced in [OPT00], and named by [EHJT04]. The idea is to iteratively compute the optimal $\hat{y}$ for each value of K, starting with $K = 0$, and until some stopping criterion is met. For each value of K, the weights are computed exactly, and the feature function parameters are obtained either analytically if the form of the feature function has such an analytical solution, or via numerical optimization in the general case.

2.3 ECON vs. MLP

MLP and ECON both learn an approximation of a real function given a set of samples of this function. The result is a weighted sum of feature functions. But some differences arise that should provide interesting advantages to ECON:

- a MLP has a fixed number of units whereas ECON learns the number of units that should be used;
- all the parameters of a MLP unit are the same for all units whereas ECON selects the parameters of each single feature function;
- ECON seeks a parsimonious representation, which means that it compresses the information.

In the next sections of the paper, we thus investigate the use of ECON for photometric solids approximation.

3 Methodology

3.1 The learning stage

In order to compare the results provided by the two learning methods (MLP and ECON), we apply a methodology that is sketched in figure 2. Each IES file used in our experiments is first of all submitted to both learning methods.

Both methods use a subset of the input data: we use the rest of the data to control the accuracy of the model. This way, we avoid overfitting, that is, the model is too fit to training data, and no longer good to predict accurately other data in the domain. For practical purposes 80% of the input data are used for learning and 20% for evaluating the resulting approximator. This split is made at random.

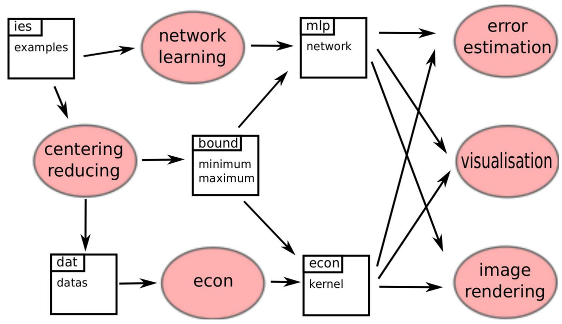


Figure 2: The methodology of the approach used in this paper to compare MLP and ECON.

After each learning algorithm has been run, its output (the model it induced) is written into a new file according to a specific file format. MLP outputs the weights, while ECON outputs the weights and the function parameters. According to the input data, the learning computation time can range from a few seconds to a few minutes. Note however that learning is performed only once and that once learned, the use of the model is very fast.

3.2 Using the resulting approximators

Once learning has been performed, MLP and ECON provide an approximator of a photometric solid. These approximators can then be inserted into any global illumination or realtime rendering method using a simple interface where an emission direction is provided to the approximator and a luminance value is returned.

The main use of the output of the approximators in this paper is the analysis of the discrepancy between the IES data and the approximated values. These outputs are thus used either in for error analysis (see section 3.3), or in a 3D visualization tool that can help us in locating the directions where large errors occur.

3.3 Computing the error

In any rendering method, evaluating the accuracy of the new representation during its use requires the definition of a relevant error criterion. From a rigorous point of view, we should be able to compare the luminance emitted by the photometric solid along any direction to a measure of the luminance received along the same direction for the corresponding real lighting system. But in addition to the complexity of such measures, it is obviously difficult to access the numerous lighting systems for which photometric solids have been analyzed. We thus have to deal with this problem only with the IES available samples. One way to measure the error could be to compute a visual distance between two images rendered respectively with the original IES data and the new representation. But two problems arise with such an approach. The first one is concerned with the fact that images are generally computed with a classical camera model that allows only a part of the scene to be rendered. Some kind of fish eye camera could be used but it requires dealing with distorted images. The second problem is concerned with the choice of the interpolation method that has to be used with the IES data, since samples represent a small subset of the emission directions set. Because interpolation is used, it can introduce artefacts into the rendered images that would not be present in real images (see for example figure 3). It is thus difficult to get a relevant measure of the error in this case.

Another way to measure the error is to consider the numerical distance between the available data and the corresponding values provided by the new approximation models.

Being unable to make a global comparison of two images, we have to use a local comparison based on a set of sampled directions, for which we compare the luminance obtained from the IES file, and from the MLP, or ECON. From this local error, we obtain a global error measure using the normalized root mean square error, and the normalized mean absolute error.

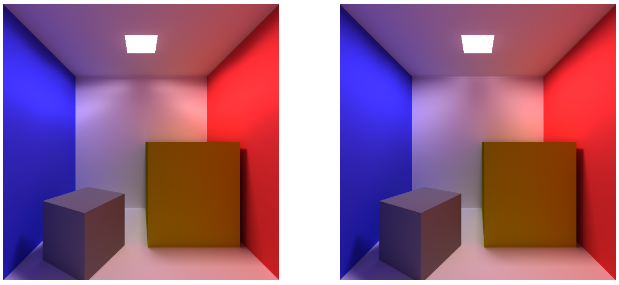


Figure 3: Global illumination with IES data using linear interpolation (left) and with an MLP model (right). Lighting discontinuities due do the linear interpolation are clearly visible onto the back wall while they do not exist when using the MLP model.

We used the normalized root mean squared error (RMSE) which is given by equation 3, where n is the number of directions used for the error computation, y the measured value of luminance along any direction and $\hat{y}$ the value provided by the model along the same direction. Because IES file provides only discrete values, a linear interpolation is computed to find y.

$$RMSE = \sqrt{\frac{1}{n} . \sum_{i=1}^{n} (y - \hat{y})^2} \tag{3}$$

Normalization is performed using equation (4), and provides the normalized root mean squared error (NRMSE). $max(y)$ (resp. $min(y)$) is the maximum value (resp. minimum value) of y over the n sampled directions.

$$NRMSE = \frac{RMSE}{max(y) - min(y)} \tag{4}$$

Mean square error has the disadvantage to over-represent the large error terms in the result: since differences are squared, the contribution of large error is exaggerated. That is why a Mean Absolute Error (MAE) and a Normalised Mean Absolute Error (NMAE) are computed (equations (5) and (6)).

$$MAE = \frac{1}{n} . |\hat{y} - y| \tag{5}$$

$$NMAE = \frac{MAE}{max(y) - min(y)} \quad (6)$$

Another way to normalize the error taht has been suggested in [Den02] is to normalise the error with respect to the statistical variance of y. We thus can define an error function according to equation (8).

$$\begin{aligned} ErrFunct &= \sqrt{\frac{MSE}{\sigma_y^2}} && (7) \\ &= \sqrt{\frac{\frac{\sum(\hat{y}-y)^2}{n}}{\sum\frac{y^2}{n} - \left(\frac{\sum y}{n}\right)^2}} && (8) \end{aligned}$$

4 Results

4.1 Experimentation data

To overcome the unavailability of a formal proof that one method outperforms the other, we compare MLP and ECON on a large set of IES data files. For this purpose we download the photometric data files libraries of two lighting societies: Ledalite [Led] (613 data files) and Metalumen [Met] (1310 data files). This provides us with 1.923 IES files that have all been used during our experimentations [1].

Regarding MLP and ECON, we have implemented them both in C. The MLP used here is a 1 hidden layer perceptron, with 10 units in the hidden layer (that is $K = 10$ in eq. (1)). ECON uses multivariate Gaussian feature functions with a diagonal covariance matrix.

Data are provided both to ECON and to an MLP. The median learning time is 28s for ECON and 16s for the MLP[2]. Let us recall that learning is performed only once during the IES data compression.

[1] Statistical analysis and graphical representation are made with the R [R D09].

[2] Learning is performed on an Intel®Core™2 Duo CPU T8300@2.40GHz running Ubuntu 9.10; programs are build using GNU g++ 4.4.1 compiler with -03 optimization flags.

4.2 Error

Table 1 presents statistical results for the two photometric data libraries that have been used during our experimentations. It highlights that ECON provides generally better results that the MLP approach for the two libraries. With ECON the median NRMSE is distinctly smaller: 0.06615 for Ledalite and 0.07119 for Metalumen (as compared to 0.1537 and 0.1718 for the MLP).

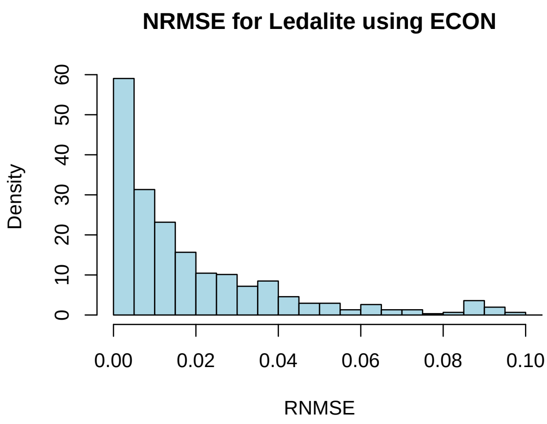
In figure 4, for each photometric database and for each approximator, we report the distribution of the number of IES file concerned with some specific NRMSE error. The diagrams show clearly that most of the approximators provided by ECON are more accurate than those provided by the MLP.

Even if the NRMSE is relatively low in the two cases, the luminance distribution can be visually different. This is highlighted in the images of figure 5 where these distributions are visualized on a wall: the MLP approximator provides clearly a less accurate distribution especially in the grazing directions (additionnal comparisons are given in figure 8 for complex photometric solids and in figure 7).

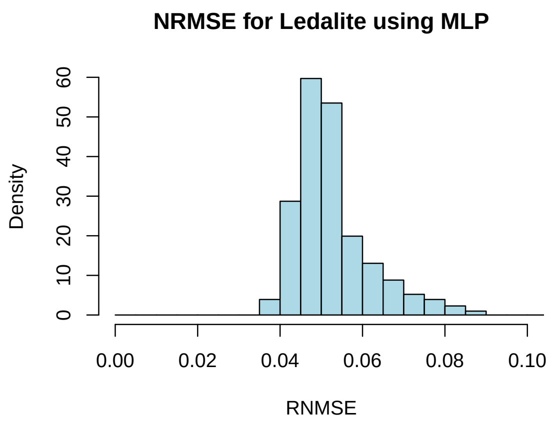
Figure 5: We illustrate the visual impact of using the different methods on the Ledalite $9414D1H254$ IES file. The 3 subfigures are the distribution of the luminance obtained by linear interpolation of the original IES file (center), the one obtained from the MPL approximator (left) and the one obtained from the ECON approximation (right). RNMSE(MLP) = 0.204, RNMSE(ECON) = 0.0369 for ECON.

	Ledalite				Metalumen			
	ECON		MLP		ECON		MLP	
error mesure	mean	median	mean	median	mean	median	mean	median
RMSE	0.1357	0.1065	0.2283	0.2245	0.1631	0.09554	0.2234	0.2207
NRMSE	0.1164	0.06614	0.1791	0.1537	0.1639	0.07119	0.2073	0.1718
MAE	0.09481	0.07387	0.1786	0.1807	0.1181	0.06511	0.1732	0.1733
NMAE	0.08107	0.04472	0.1388	0.1217	0.1167	0.05020	0.1596	0.1317
ErrFunct	0.1540	0.1172	0.1540	0.1172	0.2491	0.2253	0.1736	0.1056

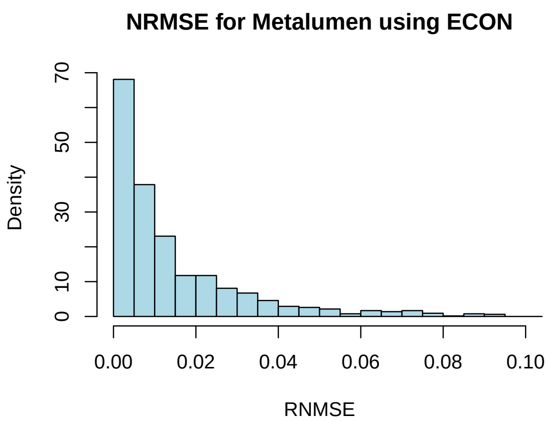
Table 1: Statistical results for the approximators provided by ECON and a Multi-layer Perceptron applied on the Ledalite and Metalumen photometric data libraries. For each, mean and median are shown. The median is less sensitive to abnormal values than the mean.



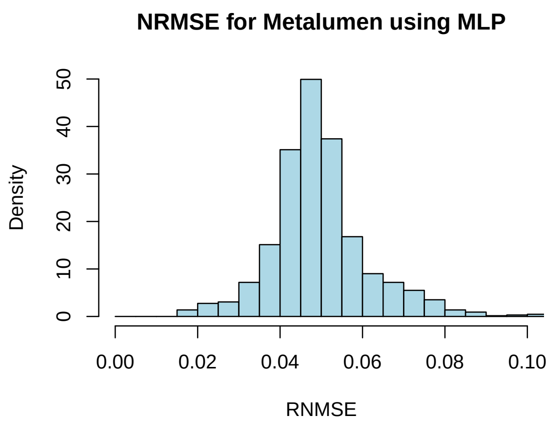
(a)



(b)



(c)



(d)

Figure 4: Histograms of the NRMSE of the approximations provided by ECON and the MLP, measured on the set of 1923 IES files (Ledalite database on the left, Metalumen on the right). The shape of these distributions does not really depend on the library, but the dependence on the algorithm is striking.

4.3 Compression

Another important way of evaluating the approximators is to measure the size of the data they generate as compared to the original data size. For this purpose we compute a compression ratio C_r that is defined as the ratio between the number n_{IA} of floating point values required for storing the data of the approximator and n_{IES}, the one of the corresponding IES file. Those values represent the weights for the MLP, the weights and feature parameters for ECON, and the angle and luminance values for the IES files.

$$C_r = \frac{n_{IA}}{n_{IES}}$$

According to this criterion, the MLP has generally better compression performances than ECON. The C_r median value for MLP is 0.0895 for the Ledalite library, and 0.1752 for the Metalumen library, whereas these values are respectively 0.2249 and 0.4342 for ECON. Here again, we can note that the dispersion of results is much more important with ECON (see figure 6). Furthermore we found that the compression ratio is greater than 1 for a few IES files using ECON. This is really surprising and must be investigated in future works.

5 Conclusion

In this paper we studied and compared the use of two machine learning approaches for photometric solids compression. By using a large number of photometric data from two lighting societies we were able to show that this kind of approach could be of great value for photometric approximation. In most cases the ECON algorithm appears to give the best approximator but in some few cases the MPL approach surpass its results. Reciprocally the compression ratio is generally better when using a MLP. These two features should be investigated deeply in a future work. Nonetheless one way to take advantage of the two approaches could be to use them simultaneously inside a multi-modeling approach: both approaches could be run onto the input file and their results would be compared in order to choose which one should be finally used according to a given criterion (error, compresssion rate, ...). This kind of approach should then be extended by studying the use of other feature functions in ECON. Finally, we will investigate the few IES files where either a large error, or a bad compression rate, has been found in order to be able to generalize our approach.

References

[CPB06] L. Claustres, M. Paulin, and Y. Boucher. A Wavelet-Based Framework for Acquired Radiometric Quantity Representation and Accurate Physical Rendering . *The Visual Computer*, 22(4):221–237, April 2006.

[Den02] J.M. Deniel. *Modélisation des luminaires et des BRDF : réalisation, mesure et compression.* PhD thesis, Université de Rennes 1,, 2002.

[DRP09] S. Delepoulle, Ch. Renaud, and Ph. Preux. Light source storage and interpolation for global illumination: a neural solution. In D. Plemenos and G. Miaoulis, editors, *Artificial Intelligence Techniques for Computer Graphics*, volume 159 of *Studies in Computational Intelligence*, pages 87–104. Springer, 2009.

[EHJT04] B. Efron, T. Hastie, I. M. Johnstone, and R. Tibshirani. Least angle regression. *Annals of Statistics*, 32(2):407–499, April 2004.

[Hay08] S. Haykin. *Neural Networks: A Comprehensive Foundation.* Prentice Hall, 3rd edition, 2008.

[IES95] IESNA. *IESNA Standard File Format for Electronic Transfer*

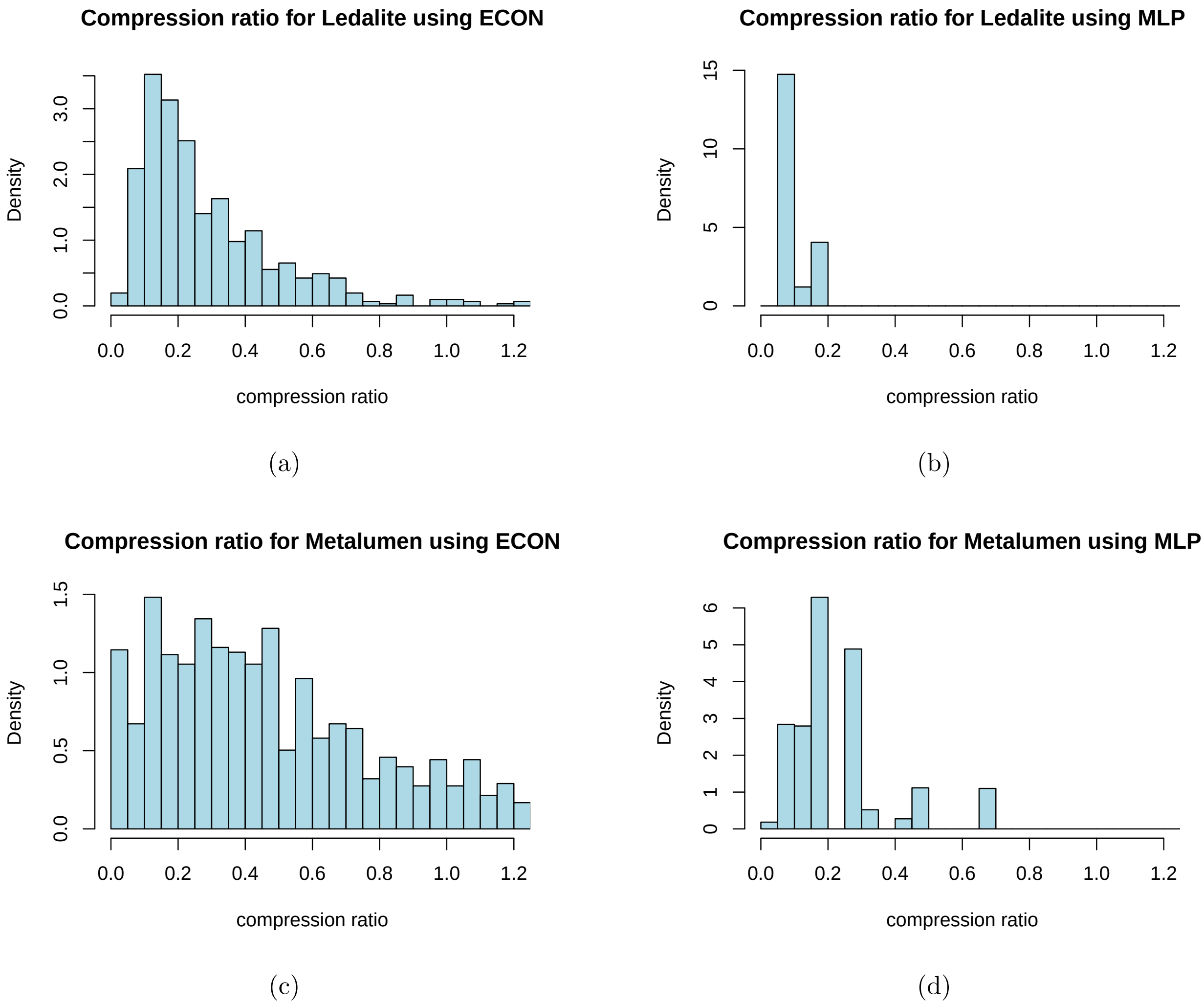


Figure 6: Histograms of the compression rate C_r for ECON and the MLP, obtained on the set of 1923 IES files (Ledalite database on the upper row, Metalumen on the lower row). The shape of these distributions slightly depends on the library, while the dependence on the algorithm is striking.

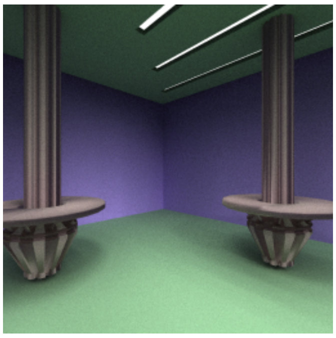
of Photometric Data. Illuminating Engineering Society of North America, lm-63-1995 edition, 1995.

[Led] Ledalite. Electronic online photometric data files library. url: www.ledalite.com/.

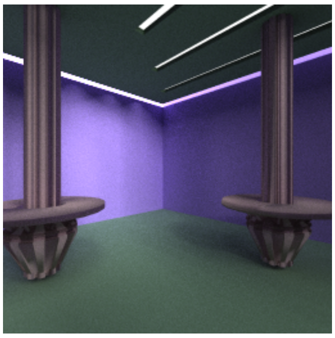
[LPDR09] L. Loth, Ph. Preux, S. Delepoulle, and Ch. Renaud. ECON: a Kernel Basis Pursuit Algorithm with Automatic Feature Parameter Tuning, and its Application to Photometric Solids Approximation. In M.A. Wani, M. Kantardzic, V. Palade, L. Kurgan, and Y. Qi, editors, *Proc. International Conference on Machine Learning and Applications (ICML-A)*, pages 162–169. IEEE Press, December 2009.

[Met] Metalumen. Electronic online photometric data files library. url: www.metalumen.com.

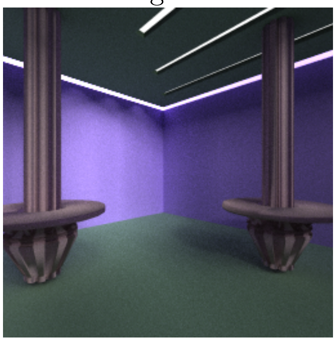
[NP00] N. Noé and B. Péroche. Hierarchical reconstruction of brdfs using locally supported functions. *Com-*



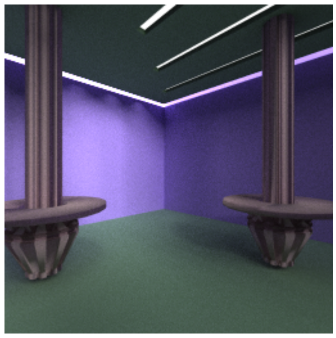
Rendering with isotropic lights



Rendering using IES



Rendering using MLP



Rendering using ECON

Figure 7: Example of rendering using MLP or ECON as approximator. The same scene is used to compare with a linear interpolation of IES file and with isotropic light sources.

puter Graphics Forum, 19(2):173–183, 2000.

[OPT00] M.R. Osborne, B. Presnell, and B.A. Turlach. A new approach to variable selection in least squares problems. *IMA Journal of Numerical Analysis*, 20(3):389–403, 2000.

[R D09] R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2009. ISBN 3-900051-07-0.

[SAWG91] F. Sillion, J. Arvo, S. Westin, and D. Greenberg. A global illumination solution for general reflectance distributions. *Computers Graphics*, 25(4):187–196, July 1991.

[WSC06] C. Walder, B. Schölkopf, and O. Chapelle. Implicit surface modelling with a globally regularised basis of compact support. *Computer Graphics Forum*, 25(3):635–644, 2006.

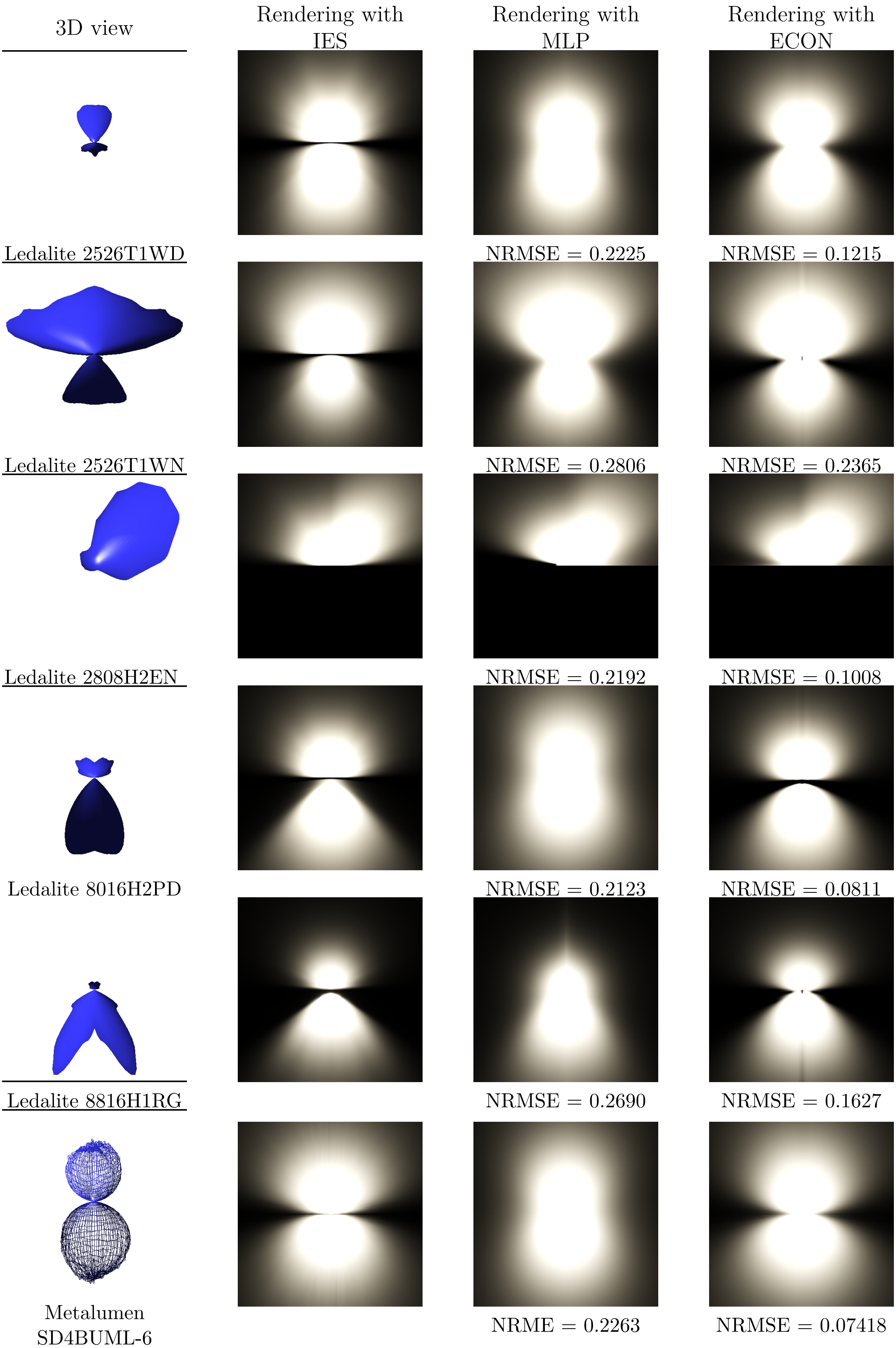


Figure 8: Examples of rendering using MLP or ECON as approximator.