Data Fitting Report

Jurriaan Heuberger, Wouter Pasman

Introduction

This document will discuss the fitting of measured reflectance data to various reflectance models. This investigation was done in the context of a masters-project with as goal to measure and model the reflectance of paintings and other two-dimensional artwork. The models used in this investigation are those models that are candidate models for the representation of the reflection of paintings. A large number of candidates **[1]** have been reduced by certain criteria to two different models, the Lafortune Generalized Cosine Lobe model, and the Ashikmhin model. This investigation will examine aspects of these models and how well they lend themselves to data fitting.

By examining the process of fitting of measured data to these models and determining the goodness of fit, we can answer a number of different important questions. First of all we have a better understanding of how the models will behave when using our own measured data. We are also able to design a scheme for sampling the BRDF. By comparing the results of the fitting to measurements, it is possible to design a sampling scheme that, when used for the fitting, will result in a good representation of the actual reflective behavior of the surface.

First this document will discuss the theoretical aspects of the data fitting process and our application of these principles. This will be followed by an explanation of the complications and adjustments made specifically for this problem. Finally a comparison of the results will allow us to decide which model is more suitable for our purpose, and what sampling scheme gives an optimal balance between sampling density and fitting accuracy.

The data fitting problem

There are many different methods that can be used to fit measured data to a model. Since we are dealing with mathematical models, we look for a mathematical approach by which to do this. These methods are cumulatively named optimization algorithms. In essence we are dealing with a system of equations which we are trying to solve.

$$F(\alpha, \beta) \approx \vec{y}(\beta)$$

Where α is a vector of model parameters, β is measurement specific parameters such as the incoming and outgoing direction, F is the mathematical function of the reflection model, and $\vec{y}$ is a vector of the measured reflectance, which is of course also a function of β. The purpose of performing the data fitting is to find the set of parameters α that makes the model correspond to the measurements best.

There are several classes of optimization algorithms: line search algorithm, gradient search algorithms and other methods such as genetic algorithms. In this case the non-linear nature of the models means a non-linear method is needed to perform this fitting.

The original work done by Lafortune suggests the Levenberg-Marquardt algorithm, which is well suited to the problem and gives good results according to the paper.

The Levenberg-Marquardt Algorithm

The Levenberg-Marquardt algorithm is a non-linear optimization algorithm in the class of gradient search methods. It is based on the Gauss-Newton method of optimization. The Gauss-Newton method approximates the solution of a non-linear least squares problem by replacing it by a series of linear least squares problems.

The basis for the problem is the residual or error function $r(x)$, which describes the difference between the measured value y and the function value $F(x)$. It is this residual function that we try to minimize. So we try to find a set of parameters α such that $\vec{r}(\alpha) = \vec{y}(\beta) - \vec{F}(\alpha, \beta) \approx 0$. The Newton method of optimization attempts to find the minimum of an arbitrary function $f(x)$ by finding a point where the derivative of the function (or the Jacobian J in the case of a vector) is 0. This is done iteratively by calculating consecutive values $x_{k+1} = x_k + H_f^{-1}(x_k)\nabla f(x_k)$, using the Hessian matrix H, a matrix containing the second partial derivatives of function f for all values of x. These steps are repeated until $f'(x_k)$ is close enough to 0. In practice inverting the Hessian is very computation intensive so a linear system of equations $H_f(x_k)s_k = -\nabla f(x_k)$ is solved instead to calculate $x_{k+1} = x_k + s_k$.
For non-linear least squares optimization of a residual function $r(x)$ for a discrete number of chosen values of x, we minimize the function $g(x)$.

$$\vec{g}(x) = \tfrac{1}{2}\left|\vec{r}(x)\right|^2 = \tfrac{1}{2}\vec{r}^{\,T}(x)\vec{r}(x) \qquad \textbf{(1)}$$

This means we need to find the gradient vector and Hessian of $\vec{g}(x)$, which are

$$\nabla g(x) = J^T(x)r(x) \qquad \textbf{(2)}$$

functions of the Jacobian of $\vec{r}(x)$, and the Hessian matrix of $r_i(x)$, which is the i^{th} component of the residual function $\vec{r}(x)$.
Next, we calculate each Newton step to get an approximate solution.
The Gauss-Newton method for optimization of non-linear systems of equations

$$H_g(x) = J^T(x)J(x) + \sum_{i=1}^{m} r_i(x)H_i(x) \qquad \textbf{(3)}$$

$$[J^T(x_k)J(x_k) + \sum_{i=1}^{m} r_i(x_k)H_i(x_k)]s_k = -J^T(x_k)r(x_k) \qquad \textbf{(4)}$$

replaces this iteration step calculation by a simpler calculation $J^T(x_k)J(x_k)s_k = -J^T(x_k)r(x_k)$ because the Hessian is computationally inconvenient and expensive to compute. Also the Hessian is multiplied by the residual which should be quite small for well fitting systems. Like most Newton derived methods, the Gauss-Newton method can fail if it is started too far from the minimum. Also if the residual is not negligible, the method may not converge or converge very slowly.
The Levenberg-Marquardt method uses a similar strategy as the Gauss-Newton method, but uses the linear system $[J^T(x_k)J(x_k) + \mu_k I]s_k = -J^T(x_k)r(x_k)$ instead. In this system the parameter μ_k is chosen at every iteration using some suitable strategy. The Levenberg-Marquardt algorithm is essentially a weighted combination of the Gauss-Newton method and the steepest decent method. This allows the algorithm to be very robust in practice and it is therefore a very common algorithm used.

Other possibilities

There is a vast array of possible optimization algorithms available and while the Levenberg-Marquardt algorithm was a straightforward choice based on the literature,

there are many possible alternatives which could offer faster convergence or other advantages in our specific case. For our purposes the Levenberg-Marquardt algorithms sufficed but during the course of the research we came across an analysis of several different optimization methods done by J. Backer at the University of British Columbia **[2]**. This analysis compared the speed and accuracy of several optimization methods in fitting the 2-lobe Lafortune model to various sets of generated data altered by Gaussian noise. While relatively incomplete, this study may form a basis on which further research may be based.

Another approach, less based in mathematics, is fitting of the data using a genetic algorithm. This approach simply iteratively creates a large number of "children" from a starting point and selects the best children for each iteration step. Each child spawns more offspring which are filtered and this continues until a minimum is reached. The genetic algorithm is rather more insensitive to local minima than any of the gradient search methods because it does not take the gradient of the surface into account. This algorithm requires less computation per iteration but depending on the problem may require more steps to reach a minimum.

The data sets

Data fitting requires two things, a data fitting algorithm which was discussed in the previous section and data which to fit the model to. The data set contains a number of important aspects. First of all the source of the data is relevant as the type of material determines the shape and behavior of the actual reflectance function. Second the sampling of the data set influences the data fitting by emphasizing parts as well as influence the fitting time. This section will discuss the various aspects of the data sets that were used by us for the fitting.

The BRDF is a function of 2 parameters, the incoming and outgoing light direction, $BRDF(\omega_i, \omega_o)$. The ideal data set therefore, is a complete measurement, meaning all the incoming and outgoing direction, ω_i and ω_o, of the BRDF of certain very specific surfaces. Each direction can be written as a pair of angles, θ and φ, as shown in figure 1. These selected surfaces should be of a material similar to what we expect to be measuring. Unfortunately data sets such as these are not readily available. The data measured during our initial experimental investigation **[3]** was too inaccurate and too limited to be used for data fitting purposes. Instead the main portion of the fitting was based on the MIT/MERL dataset, part of which is available on the web **[4]**. These sets of densely measured, isotropic data allowed sufficient samples to perform a good fitting while allowing us the freedom to pick and choose the sample positions based on our own assessment of the important data positions. The initial test portion was done using data generated from the models themselves.

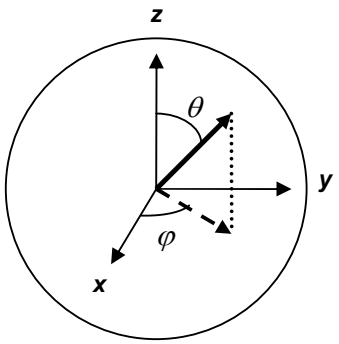

Figure 1: Spherical Coordinates

Because it costs time to measure the entire reflectance behavior of a surface, and not all viewing and lighting directions are needed for a good fit of the model, we attempt to eliminate directions that are deemed less relevant. A comparison of the signal-to-noise ratio of the resulting fittings can provide insight into which sub-sampling scheme is better. We sub-sampled the data according to four rather different

approaches and for a number of different sample densities. These are indicated in the following table.

Sampling	# of sets per material
Regular	5
Output Limited	6
Matusik Mapped	6
Custom	5

Table 1: The number and types of sub-sampling sets

The exact purpose of each of these different sampling approaches will be discussed in the following sections, but the general reasoning is the following. Sampling the reflectance is a very expensive process. Our setup would require the taking of photographs for each of the directions that are deemed relevant in the sampling. In the case of a regular sampling of 1 degree in every direction this would mean 90x360x90x360 = 1,049,760,000 pictures. Aside from this amount being impossibly large to perform in practice, using this number of samples would slow the fitting down to a significantly degree that it is practically impossible to perform. Any kind of reduction in the number of samples required to perform a fitting would therefore be useful. The different sample sets allow the examination of the fitting under different circumstances to measure the influence each different reduction of the sampling has on the resulting fitting.

Since the MIT/MERL data is isotropic, there is no use for the parameter φ_i since the data does not vary for differing values. This parameter is therefore limited to 3 regularly distributed values for all data sets that were used.

Generated Data

The very first data set used was a set of data generated specifically for testing purposes. The data set was generated by taking plausible model parameters and calculating the model values for a set of incoming and outgoing directions. A fitting to this dataset should converge to the exact parameters that were used to generate the data set. By performing a fitting to this generated data, we could test the fitting algorithm speed and accuracy based on different starting values. This gave us an idea of the sensitivity of the algorithm to starting position. When we were satisfied with the algorithm we continued to perform the fitting on the Matusik data with which we were able to test the different sampling schemes.

Regular Sampling

The first sampling scheme used was simply a regularly sampled dataset sampled at every x degrees over the 3 free parameters. This was done for x = 2, 3, 5, 10, and 15 degree spacing. Unfortunately the fitting takes a substantial amount of time and any dataset larger than the 5 degree samples set (70,000 samples) takes more than an hour to complete. Therefore the sets larger than this were dropped relatively quickly. (Matusik states that using more than 10,000 samples is infeasible, but we have found that even up to 1,000,000 samples is possible using a dual processor machine within a matter of a couple of hours) This regular sampling was done equally in all directions in order to form a basis with which the other sampling schemes can be compared.

Limited Output

Since frontal viewing of the object has more relevance than the side for most people viewing a painting, we can exclude a number of outgoing angles from the hemisphere of viewing directions. Therefore we can limit the outgoing sample positions to those

that fall within those ranges that are deemed relevant and useful. We use two different ranges to limit the number of outgoing directions used. First we limit the outgoing directions to the range [0 60] degrees for the zenith angle of the outgoing direction (θ_o), which results in a 33% reduction in the amount of data. Using a range of [0 45] degrees for the same parameter results in a reduction of 50%. Comparing the results of the fitting for reduced dataset to the regular sampled data might give insight into the general effect of limiting the data in this way for other sampling schemes, though the amount of data reduction will not be the same.

Matusik Exponential Mapping

The MIT/MERL data uses a mapping on the zenith angle of the halfway vector (θ_h), where the halfway vector is the vector bisecting the vector of the incoming and outgoing direction**,** to get a higher sample density around the specular highlight direction. This places more emphasis on the specular peak and should have an effect on the fitting. The parameter is split into 90 bins positioned at intervals based on a $\theta_h = 0.2 * 1.07^{bin}$ exponential mapping **[5]**. We make use of the same parameter mapping; however we adapted it slightly to $\theta_o = \theta_r + 0.2 * 1.07^{bin} - 0.2$ to make sure there are samples in the ideal reflection direction (θ_r). We also want to control the number of bins used, which means we need to alter the value 0.2 into a value A for each different sub-sampling density. We create distributions using the same total number of samples per direction as were used during the regular sampling. For example a 5 degree sampling results in a distribution of 90/5=18 samples for θ and 360/5=72 samples for φ. We keep the exponent parameter 1.07 and modify the initial parameter A such that the bins are spread out over the entire interval [0 θ_{range}].

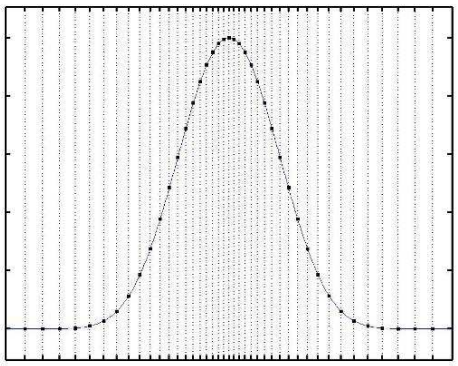

Figure 2: Exponential mapping of sample points (black dots) on a hypothetical highlight curve.

$$A * 1.07^{bin} - A = A(1.07^{bin} - 1) \qquad \textbf{(6)}$$

$$A = \frac{\theta_{range}}{(1.07^{nr\_bins-1} - 1)} \qquad \textbf{(5)}$$

In our first set we sample only θ_o in this way. In the second set both θ_o and φ_o are subject to the exponential mapping. This results in a higher density of sample points around the highlight which should correspond to a better fitting of the curve or an equivalent fitting using a lower number of data points.

Custom Mapping

The 7th set of data is created by manually supplying a number of incoming and outgoing directions based on what we think is important. First we can reduce the number of incoming directions by excluding those that we consider less important. It is difficult to say which lighting directions are irrelevant however as light could conceivably come from most directions. It is therefore smart to spread these incoming directions fairly evenly around the surface.

Fortunately it is easier to exclude angles in the outgoing direction. We can limit the amount of measurement points in the diffuse directions because we want a good fitting in the specular direction and the diffuse values show less variation than the highlight positions. This reduction is done by specifying the distance in degrees of the measured data points from the ideal reflectance angle, $\theta_o = \theta_r + \nabla\theta$. In general we use 3 different data lists to perform this reduction. They are listed in the table below and are sorted by which set they belong to I, II, III, IV, or V.

Parameter $\nabla\theta$	
I	[0, 1, 2, 3, 5, 7, 10, 15, 25, 35, 45, 60]
II	[0, 1, 2, 5, 10, 17, 25, 40, 60]
III	[0, 1, 3, 7, 15, 25, 45]
IV	[0, 1, 3, 7, 15, 25, 45]
V	[0, 3, 7, 15, 30, 50]
Parameter $\nabla\varphi$	
I	[0, 1, 2, 3, 5, 7, 10, 15, 25, 35, 45, 60, 90, 115, 140, 170]
II	[0, 1, 2, 5, 10, 17, 25, 40, 60, 90, 125, 170]
III	[0, 1, 3, 7, 15, 25, 45, 75, 110, 155]
IV	[0, 1, 3, 7, 15, 25, 45, 75, 110, 155]
V	[0, 3, 7, 15, 30, 50, 90, 135]

Table 2: Angle distribution for manual specified data sets

The incoming directions are, [0, 10, 20, 30, 40, 50, 60 70, 80] and [0, 90] for parameters θ_i and φ_i respectively for the first three sets. They are [0, 20, 45] for θ_i, and [0, 155, 270] for φ_i, for the last sets IV and V.

The θ_o values are limited depending on the actual angle of θ_i as any values out side the [0 90] range are excluded. This is not true for the φ_o parameter which has a possible range of [0 360] and which is circular all values will always be included.

Problems

In the process of performing the data fitting we ran across a number of issues that had to be resolved before we could complete the data fitting. This section will discuss the problems that arose during the fitting process, the solutions used to solve these problems and any future considerations that might need to be taken into account.

Error space manipulation

The very first runs of the Levenberg-Marquardt fitting of the generated data to the model brought up a very important problem. The algorithm was very dependent on starting positions in particularly for the exponent parameter n in the Lafortune model. While the diffuse parameter would rapidly scale to the desired value, the specular lobes would converge very slowly if they converged at all. If the exponent value was close to the desired value, the optimization would give the desired result, but in all other cases,

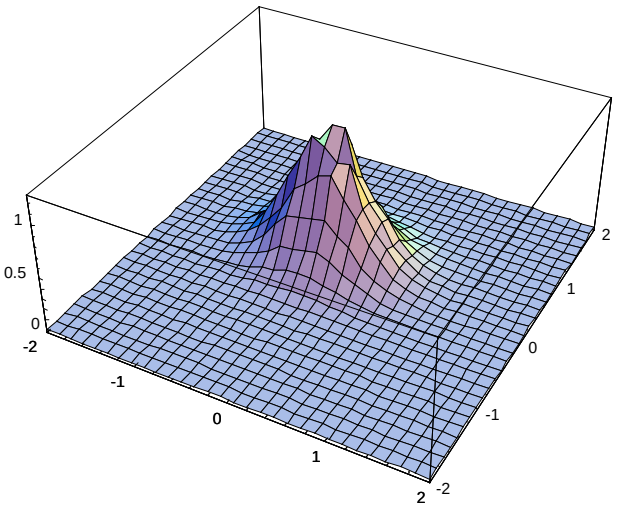

Figure 3: Noisy Gauss lobe

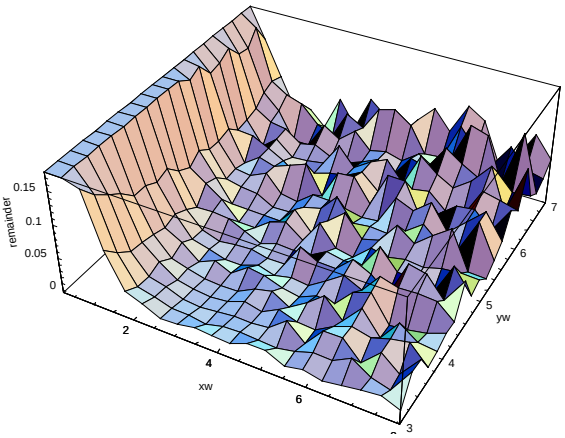

Figure 4: MSE as a function of lobe width and height

there was no convergence.

In order to determine what the problem was and how it would be possible to solve it, we made an analysis of the function space of the mean squared error in terms of the exponent parameter n.

First of all we examined the error space in terms of a broader or narrower lobe. To quickly simulate a reflectance lobe we use a 2D Gaussian bell curve. Also used by Ward **[6]** to model reflection, this simple lobe model allows us to quickly and easily examine what happens to the MSE as a function of the lobe width in both directions. We start with a Gaussian lobe with a certain width W_x and W_y in both directions with some noise added to simulate a lobe measurement (figure 2). Then we calculate the MSE between this data set and a group of other lobes with varying lobe widths. These residuals are then plotted against W_x and W_y to examine the behavior of the residual. As can be seen in figure 3, the space is noisy and riddled with local minima. An analysis of the error space between non-noisy gauss lobes results in figure 4. It shows a very flat space with a minimum which is only made very clear by pseudo-coloring. This is an exact fitting of an exact lobe, so the expected error space behavior corresponds more closely to figure 3.

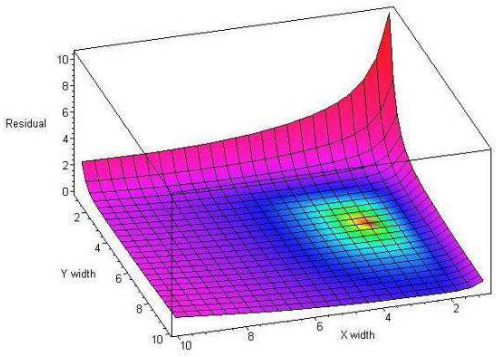

Figure 5: Non-noisy error space

Error Minimization

The second issue that arose was what error was going to be minimized. In a regularly sampled dataset the number of diffuse positions far outweighs the number in highlight directions. A simple minimization of the squared difference between the measured and the fitted data, as in equation **(1)**, therefore results in a slow fitting process and may have rather unexpected and unwanted results in terms of local minima. A good fitting of the specular peak is of particular importance. Certainly an accurate fitting of the size and shape of the specular lobe outweighs the need for a perfect fitting of the diffuse areas of the painting. The presence of noise has greater relative influence on the diffuse values than it does on the specular values, and the values in the diffuse vary less than in the specular. It is therefore important to find an error function that takes into account our objective to fit especially well to the highlight. The solution to this is scaling the error by a function of the distance of the angle from the ideal reflection direction. This is done by using the angle between the halfway vector and the surface normal (θ_h). This places a higher emphasis on the specular peaks and results in a better fit of the lobes. We replace the original error function $g(x)$ in equation **(1)** with the equation **(7)**.

$$g'(x) = \tfrac{1}{2}\left|r(x)\cos(\theta_h)\right| \qquad \textbf{(8)}$$

Fortunately this approach also led to a solution for the previously described problem of the error space. Experimentation using the error function resulted in a better fitting of the lobe width by placing special emphasis on the reflection direction. This was done by scaling the error by an exponent n of the cosine of the angle with the ideal reflection direction.

$$g''(x) = \tfrac{1}{2}\left|r(x)\cos(\theta_h)^n\right| \qquad \textbf{(7)}$$

Once a reasonable fitting has been found using error function **(8)**, the result is then used as a starting position for a fitting using the error function **(7)**. By using exponent values of 2 and 4, this approach results in a good fitting of the width of the specular lobe. Testing the method on the generated data sets also showed the resulting method of dual fitting was insensitive to the distance of user supplied starting point to the minimum.

Ashikhmin Model

The Ashikhmin model is relatively simple to calculate and requires a very limited number of parameters. Research by Addy Ngan **[7]** has shown that it also performs significantly better than the Lafortune model because of its ability to better model reflection at the extreme reflection angles.

Unfortunately we encountered some problems while trying to fit the Ashikhmin model to the data we used. This led to an in depth investigation into the mathematical workings of the model, which in turn gave some surprising results.

The original Ashikhmin model is based on the following formula.

$$p(k_i,k_o) = \frac{\sqrt{(n_u+1)(n_v+1)}}{8\pi}\frac{(\vec{N}\cdot h)^{n_u\cos^2\varphi_h + n_v\sin^2\varphi_h}}{(k\cdot h)\max(\vec{N}\cdot k_i, \vec{N}\cdot k_o)}F(k\cdot h) \qquad \textbf{(9)}$$

Where $\vec{N}$ is the surface normal, h is the halfway vector, k_i and k_o are the incoming and outgoing directions, φ_h is the azimuth angle of h, and n_u and n_v are the exponent parameters. $F(k\cdot h)$ is a fresnel function, for which Ashikhmin suggests Schlick's function $F(k\cdot h) = R_s + (1-R_s)(1-(k\cdot h))^5$. This leaves us with 3 free parameters R_s, n_u and n_v, which can be used to format the reflectance function.

The problem lies in the multiplication by $\frac{1}{(k\cdot h)\max(\vec{N}\cdot k_i, \vec{N}\cdot k_o)}$, which describes the increase of the specular lobe as the incoming angle increases towards the grazing angles. Since there is no free parameter here this relation is fixed. This behavior contrasts quite sharply with the behavior we observed in certain materials. Some of these have a much smaller growth than others, and the Ashikhmin model is unable to cope.

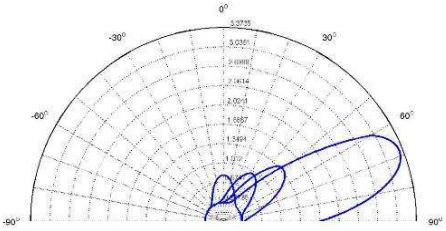

Figure 6: Original Ashikhmin Lobes

After examining the problem, we decided it was necessary to adapt the model slightly in order to make it useful to us. In doing so, it was also necessary to maintain the properties that made the original Ashikhmin model useful. The solution came in the

form of an additional exponent parameter n_s, which takes on values within the range of [0,1]. Observing that $\frac{1}{\cos(\phi)} \geq \frac{1}{\cos(\phi)^{n_s}}$ for all values of n_s within the specified range, we apply the exponent to the factor $\frac{1}{(k \cdot h)\max(\vec{N} \cdot k_i, \vec{N} \cdot k_o)}$ leading to the following new formula for the modified Ashikhmin model.

$$p(k_i, k_o) = \frac{\sqrt{(n_u+1)(n_v+1)}}{8\pi} \frac{(\vec{N} \cdot h)^{n_u \cos^2 \varphi_h + n_v \sin^2 \varphi_h}}{((k \cdot h)\max(\vec{N} \cdot k_i, \vec{N} \cdot k_o))^{n_s}} F(k \cdot h) \qquad \textbf{(10)}$$

This new parameter gives an additional degree of freedom and allows the model to be fit to materials which have a different growth rate towards the grazing angles. Resulting fittings also showed a much better fit of the model to the data.

Highly specular materials

Another issue that arose is that highly specular materials in the MIT/MERL database have a property which makes it very difficult to perform the data fitting. These materials have the property that, as they are supposed to, the specular peak values at near grazing angles become very large. The following images illustrate the difference in the peak size for various different incoming directions. The left image shows the peaks at incoming zenith angles of 0, 30 45 and 60 degrees. The right picture shows the same peaks but now with an additional peak at 80 degrees. This peak is a factor 100+ larger than the highest peak in the picture on the left.

The models are simply not capable of dealing with these extreme relations and the

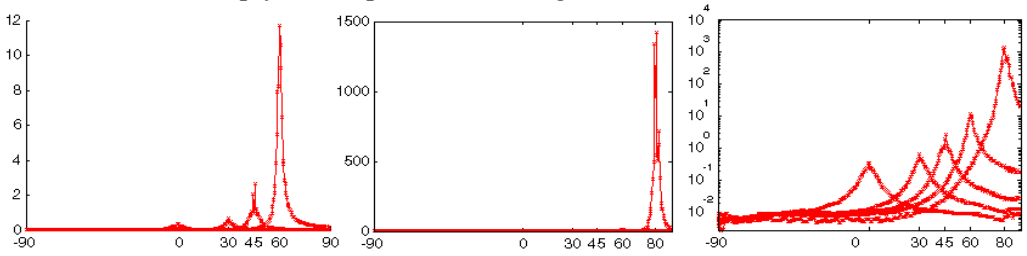

Figure 7: Height difference between 80° lobe and the other lobes in plastic data

data fitting results in models that fit well at the grazing angles since any error there is magnified 100 fold by the size of the peak itself.

There are a number of possible solutions to this problem. First of all it is possible to perform the fitting on the log of the measured data instead of the actual values. This means any error in the fitting at high values becomes much more costly than when using the current solution, but since these generally occur at extreme angles, they are less relevant to the viewing. However this also means the shape of the lobe will probably not be much like a cosine lobe as the models predict, but a $\log(\cos)$ lobe. This would obviously influence the ability of the models to fit to the data.

Another solution is to perform an additional scaling of the error function to place additional emphasis on the angles that are not near grazing. This has a similar result in creating a better fitting at other angles at the cost of large errors around the grazing angles. A third solution is to discard the near grazing angles altogether. Naturally this means a loss of information and it will result in bad fitting around the grazing angles. But because of the ease of implementation and consistency, it is this third option that is used.

Results

The following section will describe the results obtained using the different data sets. The section will focus on the results from the fitting of the blue metallic paint data, as this corresponds most closely to what we expect the paint to behave as. The rest of the fittings will be included as an appendix to this document. We start off with a plot of a few cuts of the Matusik data sets used to fit to the models. This gives a general idea of the shape of actual reflection that was used. Two-dimensional representations will be used as a backdrop for the plots of the fitted models, as a means to compare the models to actual reflectance data.

A series of 3d plots of the data for various incoming angles will give a general idea of the shape of the measured reflection.

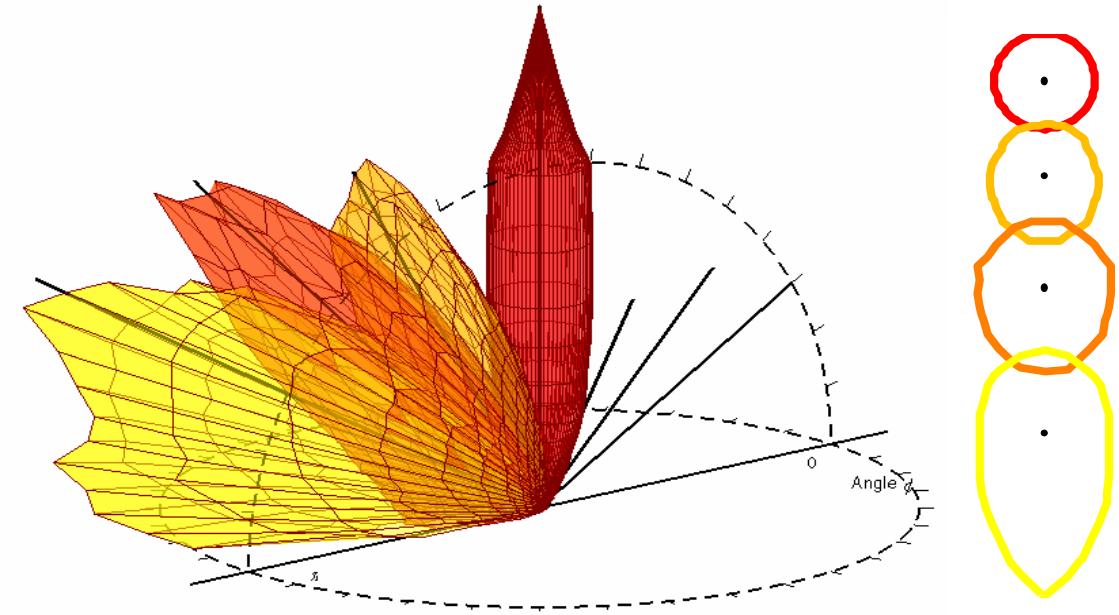

Figure 8: A 3D plot of measured data for Blue Metallic Paint for different incoming directions of light. On the right is a cross-section of the different lobes.

While the peak around the surface normal is symmetrical, the peaks at different angles become gradually higher and oval in nature, as is shown by the cross-section. They also become elongated when viewing from extreme angles, with an especially with incoming light from the grazing angles. This behavior is also present in the other materials, but they have a narrower peak, and a greater difference between the peak heights.

Regular Sampled Fitting

First of all we can look at the parameters that were the result of the regularly sampled fitting. While these do not give a clear view of the shape of the model, it is useful to note the difference in parameters between the different sampling densities.

		1 lobe Lafortune			
Blue Metallic Paint	Nr of samples	ρ_d	Cx	Cz	n
5 deg	69,984	0.0928	-1.0107	0.9694	21.4036
10 deg	8,748	0.1195	-1.0201	0.9819	30.3379
15 deg	2,592	0.1162	-1.0204	0.9847	30.7859

Table 3: Parameter values for 1 Lafortune lobe for regularly sampled data

		Ashikhmin				
Blue Metallic Paint	Nr of samples	ρ_d	n_s	R_s	n_u	n_v
5 deg	69,984	0.0537	0.2525	0.2431	50.0369	44.2850
10 deg	8,748	0.0566	0.3169	0.2318	55.7669	48.9106
15 deg	2,592	0.0844	0.1503	0.1839	85.5375	77.9953

Table 4: Parameter values for Ashikhmin lobe for regularly sampled data

		2 Lobe Lafortune						
Blue Metallic Paint	Nr of samples	ρ_d	Cx	Cz	n	Cx	Cz	n
5 deg	69,984	0.0766	-1.0044	0.9658	328.9611	-0.9956	0.9612	17.4077
10 deg	8,748	0.0810	-1.0097	0.8930	201.2291	-1.0014	0.9693	19.4582
15 deg	2,592	0.0570	-1.0012	0.9962	352.5315	-0.9761	0.9275	12.5118

Table 5: Parameter values for 2 Lafortune lobes for regularly sampled data

These numbers do not give any insight into how well the dat fit so a number of comparative images were made using the measured data (at a resolution on 1° sampling) as a background. The red line is the measured data with each marker representing a datapoint. For 5° sampling (cyan) this means every 5th marker was used, for 10° sampling (green) every 10th and for 15° sampling (blue) every 15th marker was used for the fitting. The following plots indicate the fitting of the different models to the data.

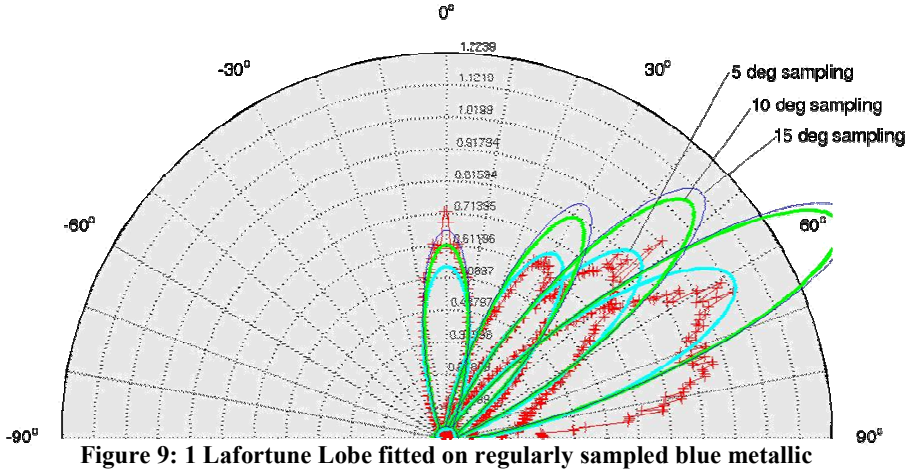

Figure 9: 1 Lafortune Lobe fitted on regularly sampled blue metallic paint data for angles of 0°, 30°, 45° and 60°

Note the fact that the lobe height grows much faster with increasing angle than the data for the 10° and 15° sampled lobes. This is due to the aforementioned property

that the peaks at the grazing angles are much higher than the other peaks. At 80° the fitting is much better for these models as seen in figure 12.

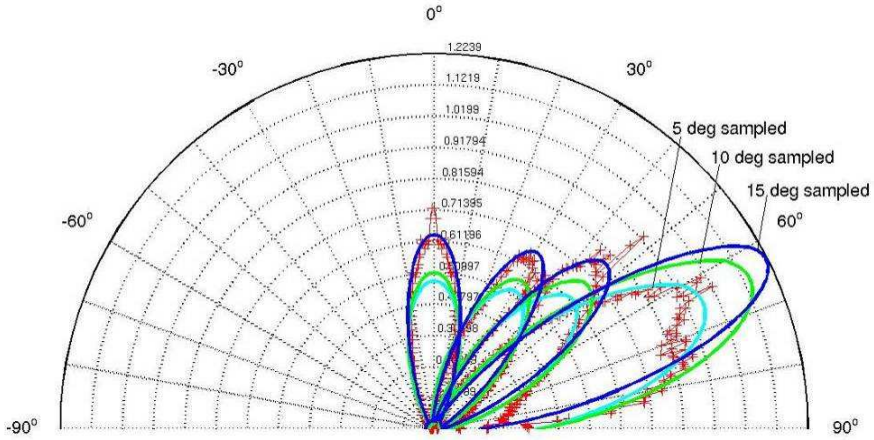

Figure 10: Ashikhmin Lobe fitted on regularly sampled blue metallic paint data for angles of 0°, 30°, 45° and 60°

It is interesting to note that Ashikhmin model is better able to cope with the bulging of the lobe towards the extreme angles. The model has a far better fitting of the measured data at these angles than the 1 lobe Lafortune does.

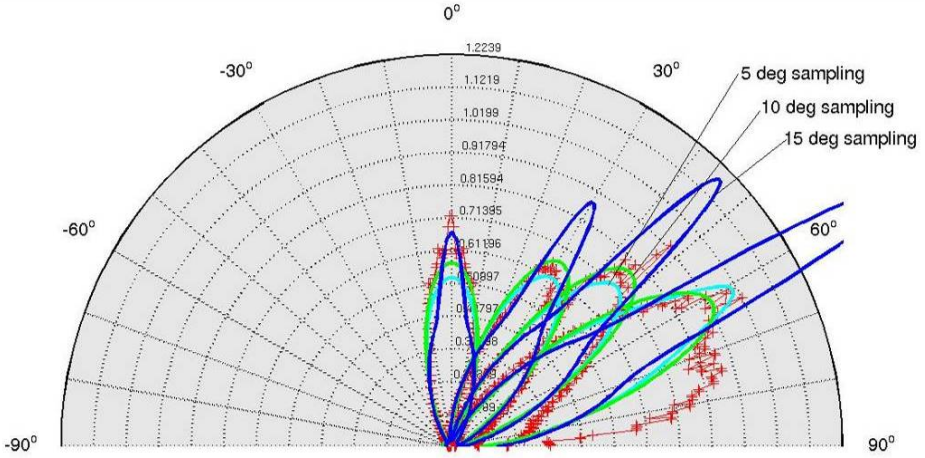

Figure 11: 2 Lafortune Lobes fitted on regularly sampled blue metallic paint data for angles of 0°, 30°, 45° and 60°

The 2 lobe Lafortune model is better able to model the peaks of the reflection because of the flexibility of the two lobes. While one lobe fits well to the bottom part of the specular lobe, the other forms towards the rest of the peak. Because of the lack of data points, however, the model will try and reach the extreme peaks at 80° while not having anything to adhere to in between for the 15° sampling, which results in the funny bottleneck looking peak.

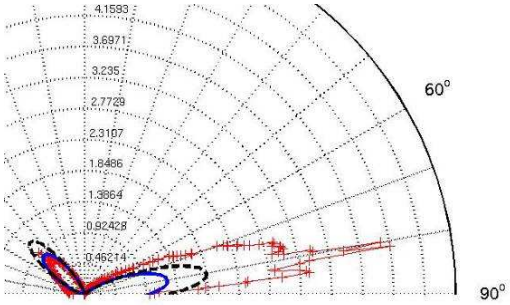

Figure 12: Comparison of an 80° lobe (right) and a -45° lobe (left) for measured data, and fitted lobes for Ashikhmin (dashed) and Lafortune (blue)

Limited Output Range fitting

The fitting in the previous section were still based on a rather large number of data points. We are going to try to reduce this number of data points in order to examine the effect this has on the fitting. By reducing the range of outgoing directions from the full range of [0°, 90°] to [0°, 60°], we reduce the number of data points used. The reduced set should result in a fitting that has fewer problems with extreme peaks at the grazing angles.

		1 lobe Lafortune			
Blue Metallic Paint	Nr of samples	ρ_d	*Cx*	*Cz*	*n*
5 deg [0 60]	46,656	0.0779	-0.9953	0.9617	17.5658
10 deg [0 60]	5,832	0.0861	-0.9983	0.9711	20.1451
15 deg [0 60]	1,728	0.1023	-1.0020	0.9813	25.0176

Table 6: Parameter values for 1 Lafortune lobe for output range limited, regularly sampled data

		Ashikhmin				
Blue Metallic Paint	Nr of samples	ρ_d	n_s	R_s	n_u	n_v
5 deg [0 60]	46,656	0.0627	0.5140	0.2059	59.3289	56.6947
10 deg [0 60]	5,832	0.0699	0.4542	0.1985	68.1899	64.5312
15 deg [0 60]	1,728	0.0892	0.3216	0.1758	89.8085	83.1329

Table 7: Parameter values for Ashikhmin lobe for output range limited, regularly sampled data

		2 Lobe Lafortune						
Blue Metallic Paint	Nr of samples	ρ_d	*Cx*	*Cz*	*n*	*Cx*	*Cz*	*n*
5 deg [0 60]	46,656	0.0447	-0.9693	0.9809	48.4114	-0.9582	0.8374	8.4967
10 deg [0 60]	5,832	0.0524	-0.9666	0.9862	65.5352	-0.9828	0.8822	10.3932
15 deg [0 60]	1,728	0.0588	-0.9911	0.9959	265.1631	-0.9907	0.9205	12.4124

Table 8: Parameter values for 2 Lafortune lobes for output range limited, regularly sampled data

What the numbers don't show, but the pictures do show, is that for these fittings, the size increase of the lobes is not much of an issue anymore. The models are fitted far more closely to the peaks since they don't have to stretch to reach the peak at 80°.

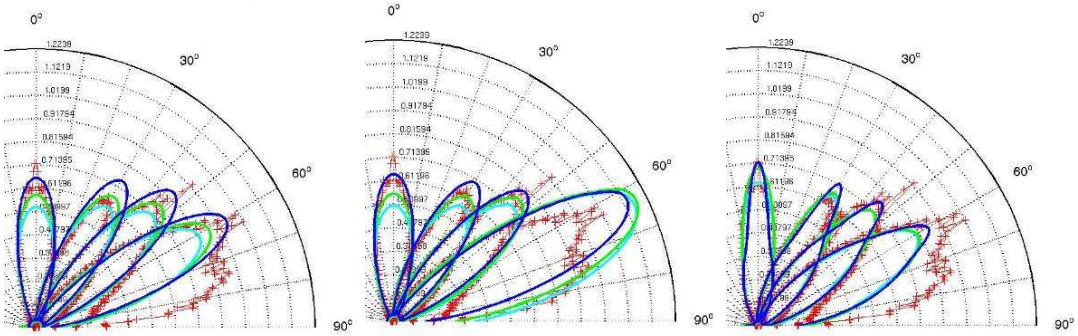

Figure 14: Data fitting for output limited to maximum 60°, blue metallic paint data for angles of 0°, 30°, 45° and 60°. From left to right: 1 lafortune lobe, ashikhmin, 2 lafortune lobes

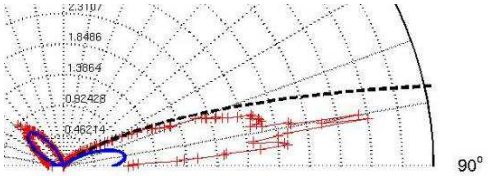

Figure 13: Comparison of an 80° lobe (right) and a -45° lobe (left) for measured data, and fitted lobes for Ashikhmin (dashed) and Lafortune (blue)

Next we reduce the set even further by excluding more of the outgoing angles. By limiting this to [0°, 45°], the data fit fail to reach the peak at 60°. This is exactly the behavior that we would expect. The exception in both cases is the Ashikhmin model which has a natural tendency to grow regardless of the parameters.

		1 lobe Lafortune			
Blue Metallic Paint	Nr of samples	ρ_d	Cx	Cz	n
5 deg [0 45]	34992	0.0871	-0.9927	0.9667	19.2725
10 deg [0 45]	4374	0.0997	-0.9894	0.9761	22.7135
15 deg [0 45]	1296	0.1145	-0.9951	0.9836	27.1433

Table 9: Parameter values for 1 Lafortune lobe for output range limited, regularly sampled data

		Ashikhmin				
Blue Metallic Paint	Nr of samples	ρ_d	n_s	R_s	n_u	n_v
5 deg [0 45]	34992	0.0724	0.6339	0.1800	69.8588	69.9069
10 deg [0 45]	4374	0.0844	0.4645	0.1679	83.8567	84.0973
15 deg [0 45]	1296	0.1003	0.3985	0.1556	102.7602	99.6616

Table 10: Parameter values for Ashikhmin lobe for output range limited, regularly sampled data

		2 Lobe Lafortune						
Blue Metallic Paint	Nr of samples	ρ_d	Cx	Cz	n	Cx	Cz	n
5 deg [0 45]	34992	0.0400	-0.9641	0.9803	43.2426	-0.9514	0.7972	7.3250
10 deg [0 45]	4374	0.0697	-0.6432	0.9881	97.2434	-1.0087	0.9370	14.5867
15 deg [0 45]	1296	0.0583	-0.9879	0.9955	237.3779	-0.9898	0.9180	12.1178

Table 11: Parameter values for 2 Lafortune lobes for output range limited, regularly sampled data

Reducing the included outgoing directions has a direct effect on the fitting of the models to the data. For this specific dataset, the grazing angles, which are problematic to the models, are excluded resulting in a more accurate fitting to the other lobes. The height difference between the peaks at 45° and 60° is not as great so the difference between figures 13 and 15 is less obvious.

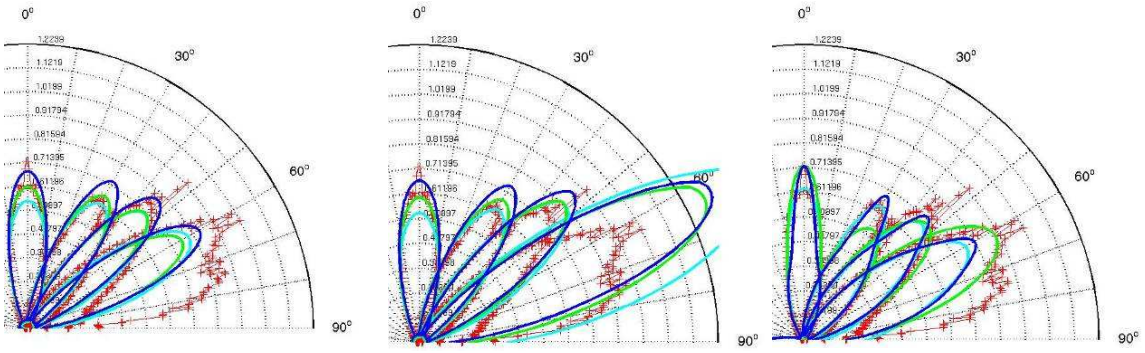

Figure 15: Data fitting for output limited to maximum 45°, blue metallic paint data for angles of 0°, 30°, 45° and 60°. From left to right: 1 lafortune lobe, ashikhmin, 2 lafortune lobes

Matusik Exponential Mapping

Another way to reduce the data set somewhat is to use the Matusik exponential mapping over the full data ranges. More importantly this gives a chance to examine how the fitting responds to placing the data points at different positions. By examining the difference with the regular sampled data it might be possible to get the same quality of fitting using fewer samples.

First we exponentially map the θ_o parameter over the range [0°, 90°] such that the highest density of data points is around the ideal reflected angle θ_r. The other parameters are regularly sampled according to the same density.

		1 lobe Lafortune			
Blue Metallic Paint	Nr of samples	ρ_d	*Cx*	*Cz*	*n*
18 bins θ_o	59,400	0.0990	-1.0102	0.9715	22.5173
9 bins θ_o	7,344	0.1273	-1.0201	0.9830	32.5072
6 bins θ_o	1,728	0.1181	-1.0205	0.9825	27.2888

Table 12: Parameter values for 1 Lafortune lobe for exponentially mapped θ_o sampled data

		Ashikhmin				
Blue Metallic Paint	Nr of samples	ρ_d	n_s	R_s	n_u	n_v
18 bins θ_o	59,400	0.0615	0.1767	0.2387	55.0499	46.5807
9 bins θ_o	7,344	0.0576	0.3217	0.2394	54.5388	47.7477
6 bins θ_o	1,728	0.0721	0.1908	0.2257	68.8862	70.1549

Table 13: Parameter values for Ashikhmin lobe for exponentially mapped θ_o sampled data

		2 Lobe Lafortune						
Blue Metallic Paint	Nr of samples	ρ_d	*Cx*	*Cz*	*n*	*Cx*	*Cz*	*n*
18 bins θ_o	59,400	0.0823	-1.0039	0.9701	380.6574	-0.9960	0.9645	18.4905
9 bins θ_o	7,344	0.0824	-1.0094	0.8952	208.9375	-1.0013	0.9700	19.5069
6 bins θ_o	1,728	0.0797	-1.0005	0.9959	2652.3	-0.9833	0.9770	18.7999

Table 14: Parameter values for 2 Lafortune lobes for exponentially mapped θ_o sampled data

In actual fact the behavior we see in these lobes is very similar to the behavior of the regularly sampled data sets. There is still a large overshoot of the peaks in the 1 lobe Lafortune model, and the strange bottleneck behavior still shows up for the 2 Lafortune lobes when using 6 bins.

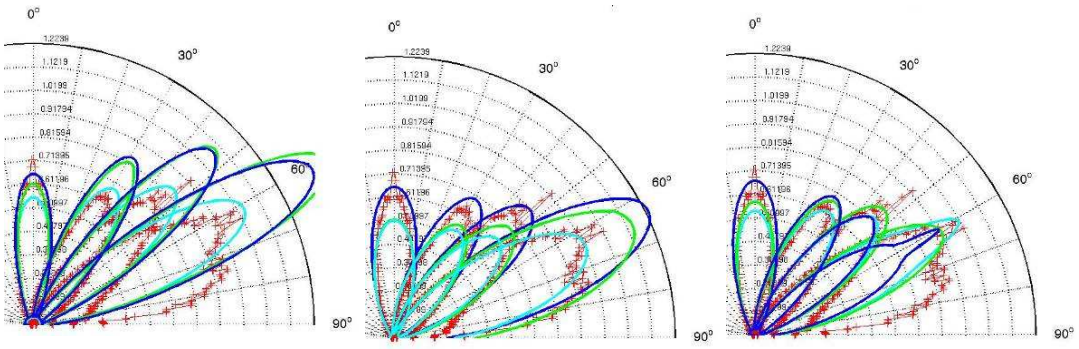

Figure 16: Data fitting exponentially mapped θ, blue metallic paint data for angles of 0°, 30°, 45° and 60°. From left to right: 1 lafortune lobe, ashikhmin, 2 lafortune lobes

The next step is to also use the exponential mapping on the other outgoing parameter φ_o by mapping it over the range [0°, 360°] so that the highest density is around the reflected direction φ_r.

		1 lobe Lafortune			
Blue Metallic Paint	Nr of samples	ρ_d	Cx	Cz	n
18 bin θ_o 72 bin φ_o	56,925	0.1520	-1.0113	0.9677	23.2669
9 bin θ_o 36 bin φ_o	6,732	0.1356	-1.0148	0.9748	25.7507
6 bin θ_o 24 bin φ_o	1,512	0.1153	-1.0136	0.9775	22.6943

Table 15: Parameter values for 1 Lafortune lobe for exponentially mapped θ_o and φ_o sampled data

		Ashikhmin				
Blue Metallic Paint	Nr of samples	ρ_d	n_s	R_s	n_u	n_v
18 bin θ_o 72 bin φ_o	56,925	0.1041	0.1957	0.2415	52.2858	45.4716
9 bin θ_o 36 bin φ_o	6,732	0.0726	0.3405	0.2430	48.5912	44.3160
6 bin θ_o 24 bin φ_o	1,512	0.0834	0.2396	0.2416	62.8981	57.8512

Table 16: Parameter values for 1 Lafortune lobe for exponentially mapped θ_o and φ_o sampled data

		2 Lobe Lafortune						
Blue Metallic Paint	Nr of samples	ρ_d	Cx	Cz	n	Cx	Cz	n
18 bin θ_o 72 bin φ_o	56,925	0.1108	-1.0027	0.9750	332.8595	-0.9977	0.9572	17.2941
9 bin θ_o 36 bin φ_o	6,732	0.1029	-1.0028	0.9652	724.9096	-1.0023	0.9661	19.2253
6 bin θ_o 24 bin φ_o	1,512	0.0898	-1.0008	0.9940	1873.7	-0.9844	0.9736	17.9174

Table 17: Parameter values for 2 Lafortune lobes for exponentially mapped θ_o and φ_o sampled data

There is a marked difference in the overshoot of the lobes for the 1 lobe Lafortune model. Aside from that there is little significant perceptual difference between the fittings including and excluding the exponential mapping of φ.

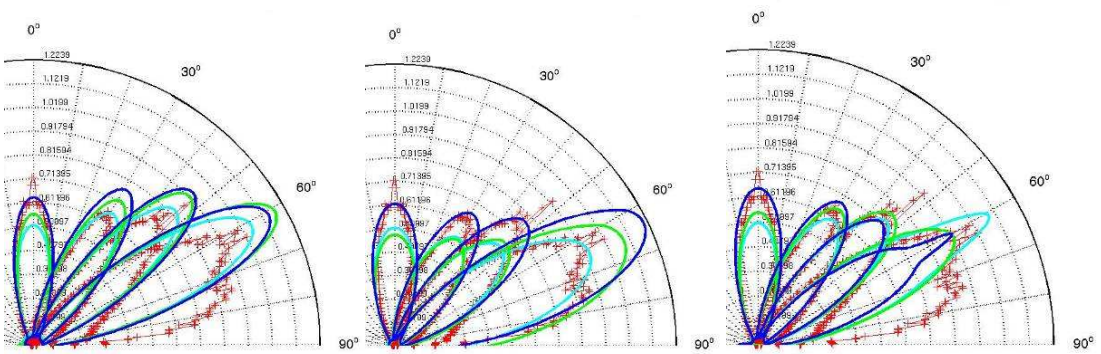

Figure 17: Data fitting exponentially mapped θ and φ, blue metallic paint data for angles of 0°, 30°, 45° and 60°. From left to right: 1 lafortune lobe, ashikhmin, 2 lafortune lobes

Custom Sampling Distribution

The final group of data sets is based on a custom selection of data positions. This allows an exact choice in what input and output angles are to be included in the data set. The 5 different specified datasets were indicated by table 2. Of these sets the first 3 are comparable in sample count to 10° regularly sampled data, ranging from 10,000 to 5,500 samples. The last to sets are relatively sparse in comparison.

		1 lobe Lafortune			
Blue Metallic Paint	Nr of samples	ρ_d	Cx	Cz	n
I	10,292	0.3809	-1.0123	0.9808	54.6710
II	8,487	0.4422	-1.0136	0.9788	58.7150
III	5,592	0.5086	-1.0123	0.9799	67.3642

Table 18: Parameter values for 1 Lafortune lobe for custom selected θ_o and φ_o

		Ashikhmin				
Blue Metallic Paint	Nr of samples	ρ_d	n_s	R_s	n_u	n_v
I	10,292	0.0885	0.3028	0.2772	47.4451	47.4471
II	8,487	0.1548	0.2586	0.2087	72.1862	58.3486
III	5,592	0.1026	0.3124	0.2541	54.6981	48.6877
IV	1,767	0.1313	0.4430	0.1771	76.5359	72.0680
V	1,080	0.1411	0.4467	0.1646	83.7116	78.4152

Table 19: Parameter values for Ashikhmin lobe for custom selected θ_o and φ_o

		2 Lobe Lafortune						
Blue Metallic Paint	Nr of samples	ρ_d	Cx	Cz	n	Cx	Cz	n
I	10,292	0.1334	-1.0046	0.9504	358.6305	-1.0003	0.9636	19.4978
II	8,487	0.1311	-1.0051	0.9425	329.4051	-1.0007	0.9638	19.4422
III	5,592	0.1297	-1.0043	0.9497	419.0863	-1.0034	0.9637	19.5609
IV	1,767	0.1321	-0.8903	0.9829	76.6958	-1.0127	0.9416	17.2885
V	1,080	0.1044	-0.9213	0.9867	87.1096	-1.0122	0.9329	15.4614

Table 20: Parameter values for 2 Lafortune lobes for custom selected θ_o and φ_o

What immediately becomes obvious is the 1 lobe Lafortune model which results in very awkward fitting for each of the first 3 data sets. This rather odd behavior, was confirmed after several runs of the fitting algorithm with different starting positions, and seems to indicate the presence of a rather problematic local minimum.

The other thing that is noticeable is the good fitting of the plotted lobes. This is because of the exclusion of many extreme angles in the specified sets. The behavior of the fitting looks a lot like that of the output range limited sets.

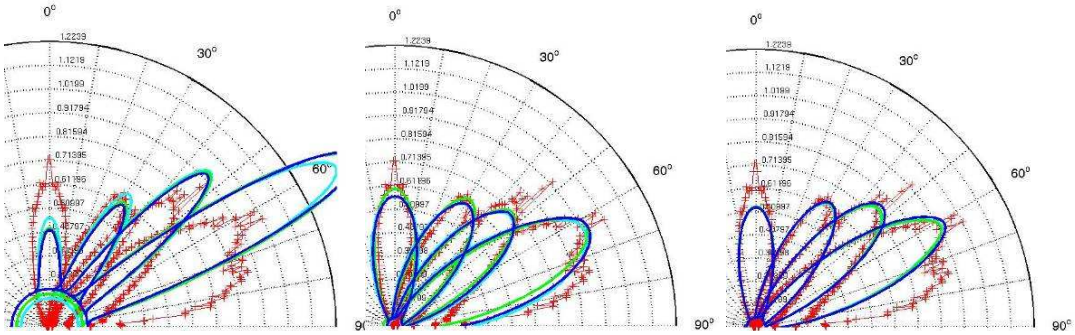

Figure 18: Data fitting for custom specified, blue metallic paint data (sets I, II and III) for angles of 0°, 30°, 45° and 60°. From left to right: 1 lafortune lobe, ashikhmin, 2 lafortune lobes

The sparse data sets was fitted and plotted for the Ashikhmin and 2 lobe Lafortune model only. What is noticeable is that the fitting seems to perform a lot better than expected. Though the number of data points is reduced significantly from the other sets, the fitting varies relatively little perceptually from the other sets.

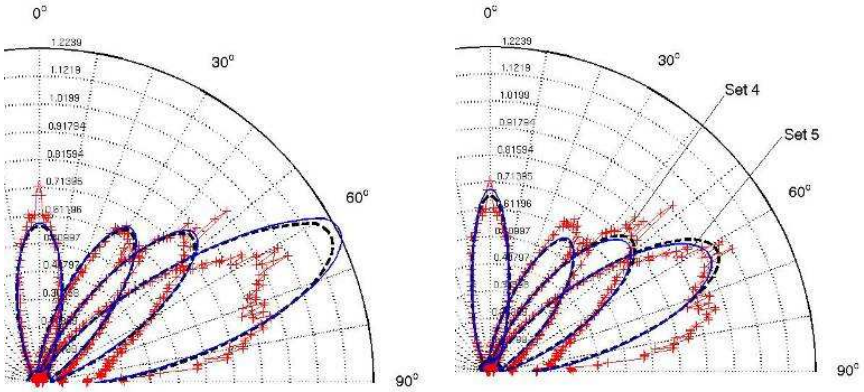

Figure 19: Data fitting of sparse custom data sets (IV and V) of blue metallic paint data for angles of 0°, 30°, 45° and 60°. From left to right: ashikhmin, 2 lafortune lobes

Comparison of results

While it is nice to have a visual representation of the data fitting results as shown above, it is not enough just to do so. Not only does this method of comparison not give enough information, but it also only visualizes a select part of the dataset. In order to judge the results appropriately and to determine the difference between the different fittings it is necessary to make a qualitative comparison of the resulting fittings to each other. The purpose and manner of making this qualitative comparison will be discussed in this section.

Qualitative comparison

A measure of how closely the model fits to the data should give us a general idea on how well the fitting succeeded. In order to determine a meaningful measure we need to compare the model to a set of data that is larger than the set to which the model was fitted to. The most obvious choice would be to calculate the average deviation of the model to the data for every position. Unfortunately this is not a good measurement of the accuracy of the fitting because it fails to take into account the variation within the reflection. We would like our error measure to be unbiased in this regard.

A common method of comparison is to use the Signal to Noise ratio as a comparison. The Signal to Noise ratio (SNR) is an estimation of the ratio of important data (signal) to unimportant or deviant data (noise). Consequently, the higher the SNR value is the better the model fit. There are various different metrics to calculate the SNR. The one used, which is commonly applied in graphics and computer vision to compare images, is calculated as follows.

$$MSE = \frac{1}{N}\sum_{i}^{N}(m_i - s_i)^2$$

$$VAR = \frac{1}{N}\sum_{i}^{N}(s_i - \bar{s})^2$$

$$SNR = 10\log_{10}(\frac{VAR}{MSE})$$

Here s_i is the measured (signal) value, m_i is the model predicted value, $\bar{s}$ is the mean of the signal and N is the number of samples in the set. We performed a comparison of the results using the SNR as a measure of accuracy, which will be discussed in the following section.

In order to get a good match of the width and shape of the lobes, a scaling of the squared mean difference was applied during the fitting process. In order to take into account the distance of the outgoing direction from the ideal reflection direction, the fitting was done towards the mean squared error scaled by $\cos\theta_{h,i}$, the angle between the halfway vector and the surface normal for measurement *i*. Unfortunately the original SNR comparison does not take this scaling into account, and we would therefore like to examine the ratio of this scaled error instead of the normal MSE. By simply scaling the MSE by the cosine of the angle θ_h, we unbalance the SNR which would require integrating the function for this angle over the hemisphere of possible angles to normalize it again. However, we can ignore the normalization if we don't compare the original SNR to the new SNR.

$$Error = \frac{1}{N}\sum_{i}^{N}((m_i - s_i)\cos(\theta_{h,i}))^2$$

$$SNR' = 10\log_{10}(\frac{VAR}{Error})$$

We know that the Ashikhmin model is better able to model the reflection behavior at grazing angles, however these are deemed less important to us than the angles in the range [0 60], for reasons specified before. A second set of ratios is therefore calculated using not the entire hemisphere of outgoing directions, but a reduced set in which only the values of θ_o that fall within this range are included. This should give an idea on which sub-sampling strategy and model combination works best in those situations.

Signal to Noise Ratio

The following SNR values were found for the fittings.

	SNR			SNR reduced set			SNR'			SNR' reduced set		
	1lobe	ashik	2lobe	1lobe	ashik	2lobe	1lobe	ashik	2lobe	1lobe	ashik	2lobe
5 deg	7.81	8.86	7.96	12.24	11.05	13.52	8.65	9.57	9.00	12.96	11.38	14.15
10 deg	6.05	7.61	4.50	8.93	11.56	12.72	7.21	8.16	5.65	9.53	11.95	13.33
15 deg	5.60	7.73	8.84	8.37	9.78	13.62	6.91	8.58	9.36	8.87	10.24	13.97
5 deg [0 60]	7.94	-2.70	8.18	13.48	9.55	17.20	8.50	-2.60	8.82	14.13	9.94	17.68
10 deg [0 60]	7.24	-0.48	7.86	12.58	8.98	16.10	8.40	-0.35	8.74	13.26	9.36	16.60
15 deg [0 60]	6.99	4.05	7.90	10.70	8.71	14.42	8.02	4.41	8.63	11.34	9.70	14.85
5 deg [0 45]	7.58	-10.78	7.41	13.01	9.46	17.04	8.25	-10.75	8.63	13.75	9.97	17.57
10 deg [0 45]	7.06	-2.78	7.84	11.55	10.50	14.90	7.58	-2.69	8.79	12.29	11.54	15.57
15 deg [0 45]	6.68	-0.80	6.79	10.11	10.10	14.51	7.37	-0.65	8.54	10.84	11.23	14.99
18 bin θ	7.84	8.49	8.20	11.96	12.34	13.29	8.62	9.21	9.01	12.74	12.82	14.00
9 bin θ	6.12	7.55	4.44	8.51	11.72	12.56	7.01	8.09	5.70	9.14	12.23	13.17
6 bin θ	5.91	7.88	7.06	8.69	9.96	10.03	7.09	8.63	7.61	9.22	10.47	10.37
18 bin θ 72 bin φ	6.83	8.35	8.11	10.46	11.67	12.70	8.31	9.34	9.53	11.85	12.19	13.76
9 bin θ 36 bin φ	6.56	7.39	6.83	10.36	10.67	12.65	8.18	7.93	8.15	11.40	11.22	13.57
6 bin θ 24 bin φ	6.85	8.07	7.01	10.01	9.25	10.30	7.97	8.84	7.96	10.66	9.78	10.72
I	1.77	7.61	7.11	2.08	10.75	11.79	3.83	8.30	8.25	3.77	11.11	13.17
II	0.53	6.99	6.91	0.72	11.43	11.87	2.70	8.37	8.04	2.43	11.81	13.21
III	-0.56	7.20	6.68	-0.54	9.24	11.88	1.67	7.93	7.90	1.21	9.57	13.19
IV	6.57	-0.74	6.95	10.35	9.91	12.43	7.79	-0.55	8.55	11.87	10.49	13.98
V	6.61	-1.59	7.59	9.99	11.18	13.78	7.62	-1.41	8.67	11.49	11.60	14.97

Table 21: SNR values for all the fitted data sets for blue metallic paint data

A plot of these values offers more insight into the relationship between them. Figures 20 and 22 show the unsorted SNR values for the unscaled and scaled error respectively. While the actual values should not be compared, the relationship between them is still valid. This shows that for the full comparison set, the differences are rather small. What is very noticeable is that the Ashikhmin model performs very badly for the output limited sets and the manual data sets. This is easily explained since these sets placed no emphasis on the values at the extreme regions while this is exactly where the large errors occur. A somewhat surprising observation is that the single lobe Lafortune model performs better when the output is limited to 45° than when the output is limited to 60°. Another surprising fact is that the 1 lobe Lafortune model does not have a significantly lower SNR on manual data set I, although the plot seemed to indicate a local minimum was interfering.

Figures 21 and 23 show the same values but then sorted towards best fit of the Ashikhmin model, which performs best of all models at 5° sampled data. The graphs show the Ashikhmin model outperforms the other two models for all but one of the sampling strategies. Also noticeable is the fact that the 2 lobe Lafortune model

outperforms the 1 lobe model in most cases, though the difference is not very large. The only difference in ranking, between the SNR and SNR' measures, occurs in manual set II, which is ranked significantly higher under the SNR' measure than under the SNR measure.
A more interesting comparison, however, should be the SNR for the reduced range of data, since this represents the visual area we are interested in. Figures 24 and 26 display the unsorted SNR and SNR' values, while figures 25 and 27 show the values sorted to the Ashikhmin model. Using the reduced set of directions, the difference between the Lafortune model and Ashikhmin model becomes more obvious. Where the Ashikhmin model outperformed the Lafortune model over the full range of viewing directions, the 2 lobe Lafortune model outperforms the Ashikhmin model for all sampling sets, and the single lobe model also gets the better of it for most of the exponentially mapped data sets, and all of the output range limited sets..
The performance of the Ashikhmin model has become more stable, as the sorted figures show. There are no severe dips in the SNR values for the 2 lobe Lafortune model either, but the single lobe model fails for the manual data sets I, II and III, as the fitting image in figure 18 had led us to believe.
Overall the conclusions that can be drawn are that the Ashikhmin model is very sensitive to the values around the grazing angles. It will outperform the Lafortune model if they are included in the fitting, indicating a better fitting of the reflection around these angles. Absence of these angles during the fitting process, however, leads to low values for the SNR. The Lafortune model, on the other hand is much less sensitive to the extreme angles, and is surprisingly consistent for even low sample density. If calculated using the reduced set of measurements, the SNR values of the 2 lobe model are consistently higher than the values of the Ashikhmin model. It also consistently outperforms the single lobe model.

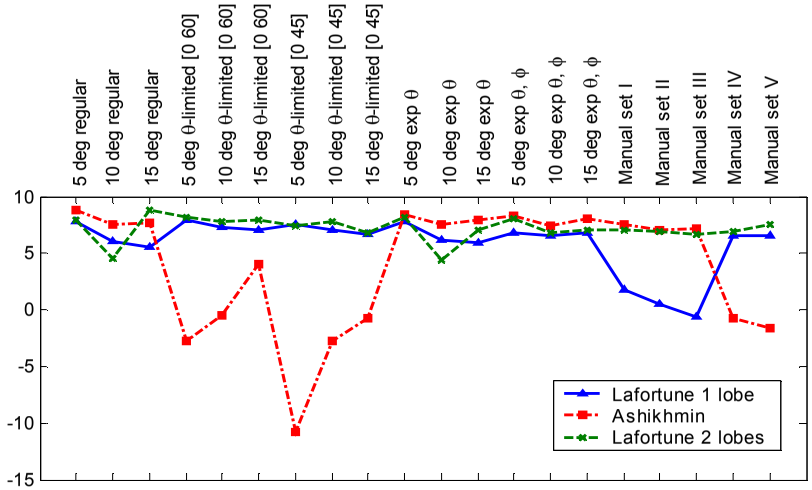

Figure 20: Unsorted full set SNR values for the different sampling strategies for blue metallic paint data

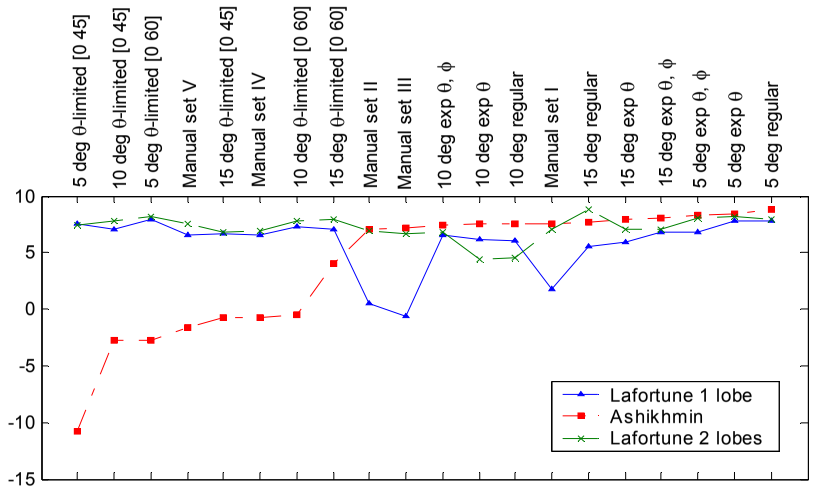

Figure 21: Full set SNR values of blue metallic paint sorted by SNR of Ashikhmin fit

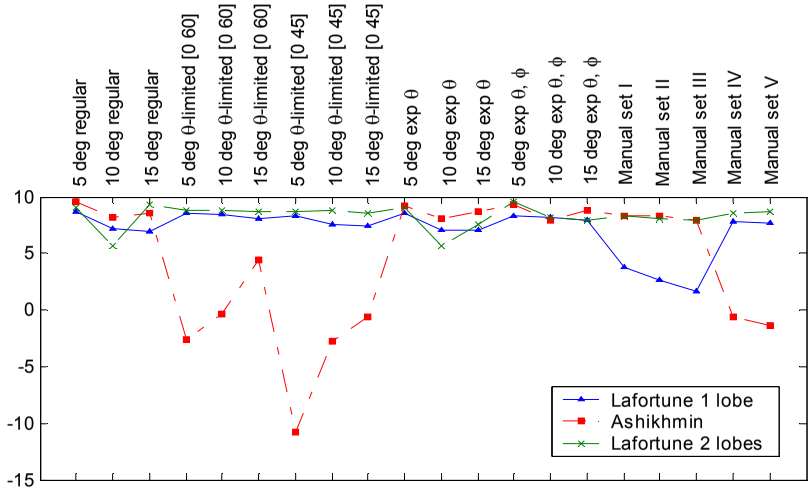

Figure 22: Unsorted full set SNR' values for the different sampling strategies for blue metallic paint data

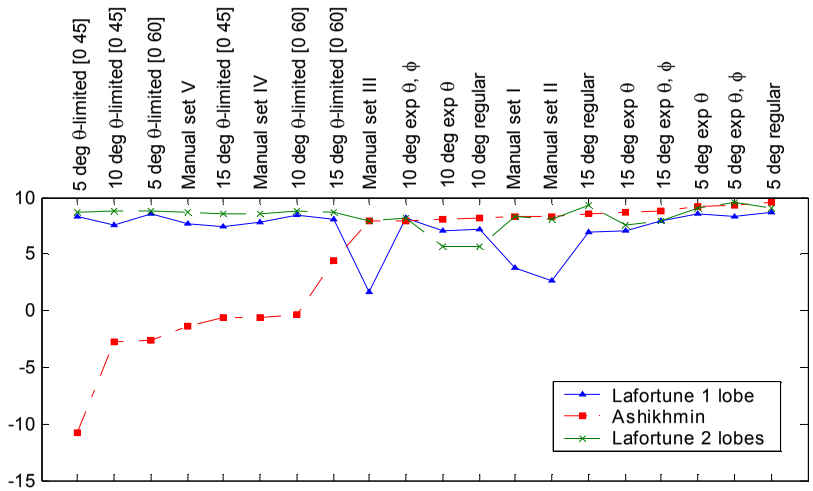

Figure 23: Full set SNR' values of blue metallic paint sorted by SNR' of Ashikhmin fit

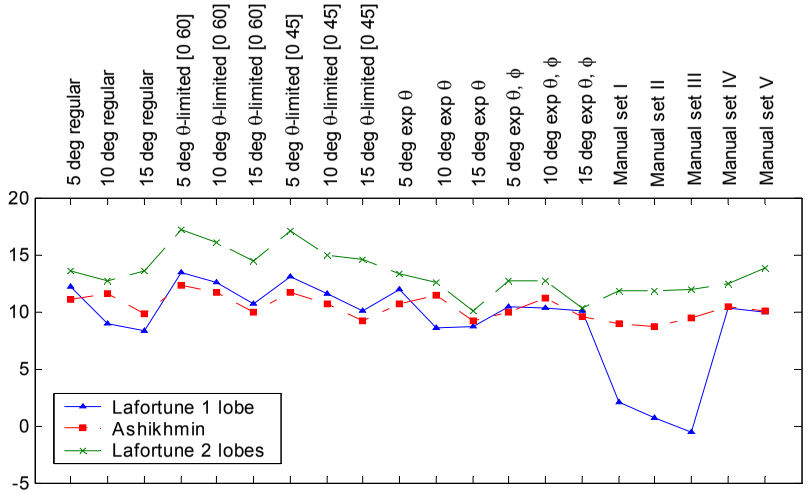

Figure 24: Unsorted reduced set SNR values for the different sampling strategies for blue metallic paint data

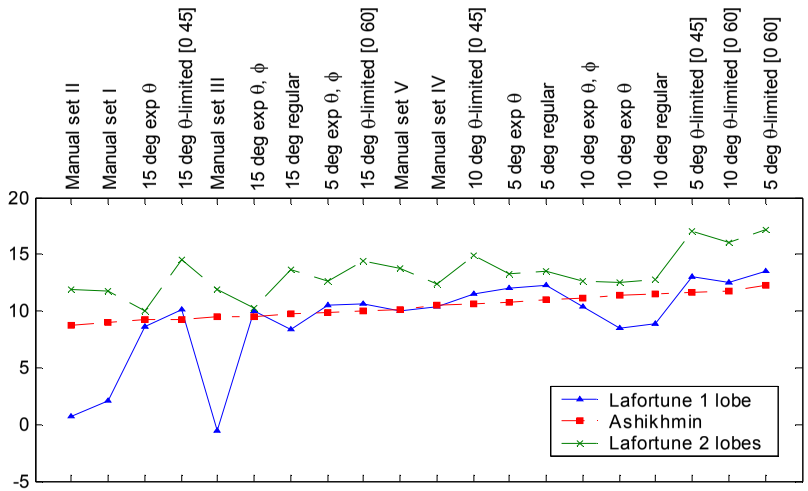

Figure 25: Reduced set SNR values of blue metallic paint sorted by SNR of Ashikhmin fit

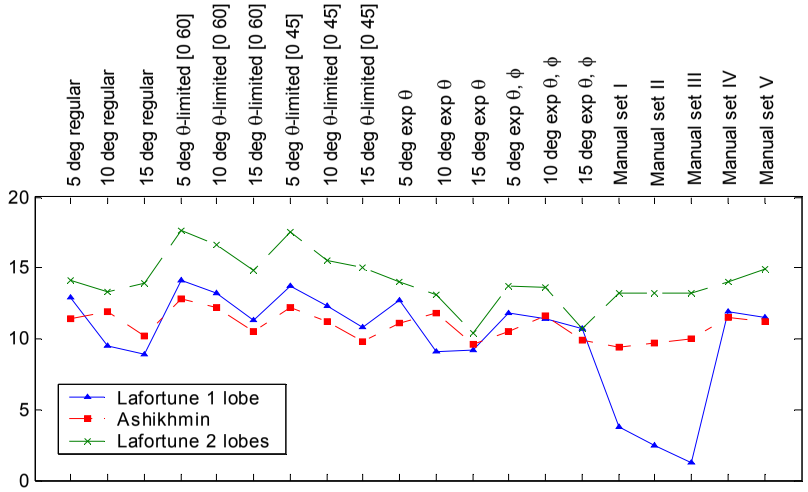

Figure 26: Unsorted reduced set SNR' values for the different sampling strategies for blue metallic paint data

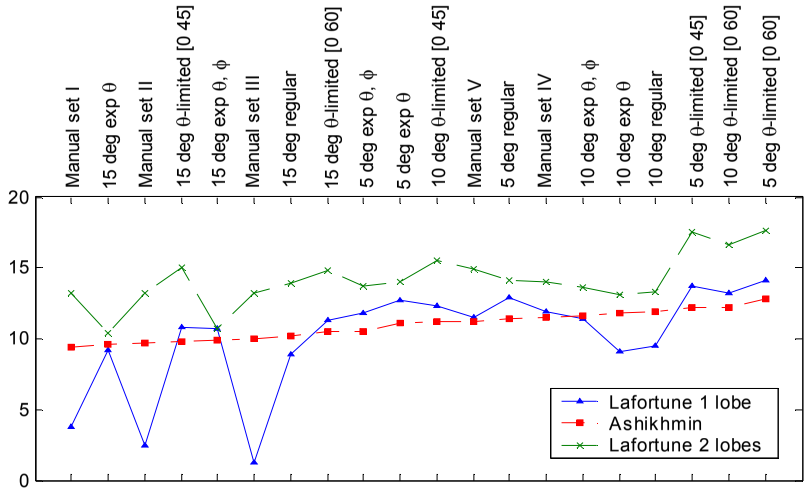

Figure 27: Reduced set SNR' values of blue metallic paint sorted by SNR' of Ashikhmin fit

Conclusions

As a result of the experiments performed we can answer the questions that prompted this investigation. These were that we wanted to know which model was better suited to data fitting and what sampling scheme should be used to gather samples for our measurement. These questions will be answered in this section.

Models

Though the issues of model choice and sampling scheme are directly linked, there are some compelling reasons that came up during the course of this investigation, that lead to a significant advantage of one model over the other. The Ashikhmin model has some issues related to it that severely hinder its use for the purpose of fitting. It had to be altered to be able to get decent results in fitting it to the data. Also it performed significantly worse than the Lafortune model for data sets that were limited around the grazing angles and specifically the very sparse data sets. These factors combined with the fact that the model is more computationally intensive (our modification added to this fact more) far outweigh the benefits the Ashikhmin model had, which was that it better modeled reflection at the grazing angles. Since we have stated these angles are of lesser significance than the frontal viewing angles, the choice falls on the Lafortune model.

The Lafortune generalized cosine lobes are more flexible in dealing with the oddities in the data sets, and fit well to the frontal viewing lobes. In terms of the number of lobes, there is no contest. Using 2 lobes instead of 1 offers significant improvement of the result, which is often in the order of 3-4 dB in the SNR of the fitting. The cost of the increased number of lobes is not very high for lower numbers of data points.

Sampling Scheme

The choice of the Lafortune model was partially based on the fact that it copes better with relatively sparse data sets. In fact the SNR qualitative comparison shows that the difference in quality between fitting of the 5° regular sampled data set, which contains close to 70,000 samples and the sparse manual data sets is surprisingly small. Since the manual data sets allow for customization and simple angle positions, this has an advantage over the exponentially mapped sets.

The output range limited data sets have significantly higher SNR and SNR' values for the reduced comparison, especially for the high resolution data sets. This indicates that not excluding angles has a strong negative impact on the fitting results in the directions up to 60°. It is therefore likely that excluding the extreme angles for the fitting will improve our results in all of the sampling strategies, since the extreme angles will no longer influence the results.

Taking into account the fact that limiting the output directions will improve the fitting results, we discard the values of the output range limited data sets. Comparisons between the other sampling schemes show that the difference between the sets is in the region of 1 dB, with a few lower outliers. The difference in sample density can be several thousand or tens of thousands sampling points. The cost in terms of time of the acquisition of such a large number of data points would be significant.

Since the SNR values are relatively stable for the sampling strategies and the exclusion of extreme viewing angles improves the fitting values of directions that fall within the region of interest, the suggested sampling scheme should be relatively sparse and exclude the extreme angles. Since the manual data sets are easy to

customize and manage they have our preference. It should be noted that some important aspects that will be taken into account when designing the set, are the exclusion of angles greater than 60°, the inclusion of 1° measures to capture lobes with a breadth less than 3°, and a total number of data points smaller than 3000.

References

[1] Heuberger J.S. *Model Investigations*, Delft University of Technology (2004)

[2] Backer J. *BRDF Fitting*. Project Report.University of British Columbia Department of Computer Science. (2002) http://www.cs.ubc.ca/spider/backer/

[3] Heuberger J.S. *Analysis Report*, Delft University of Technology (2005)

[4] Matusik,W, Pfister, H, McMillan, L., Brand, M.: MIT/Merl BRDF Database (2003) http://graphics.csail.mit.edu/~wojciech/BRDF

[5] Matusik, W., Pfister, H., Brand, M., McMillan, L.: A Data-Driven Reflectance Model. Proc SIGGRAPH, 22, 3 (2003), 759-769.

[6] Ward, Gregory J. Measuring and Modeling Anisotropic Reflection; Proceedings of SIGGRAPH 1992

[7] Ngan, A, Durand, F. Matusik, W.: *Experimental Analysis of BRDF Models*, Eurographics Symposium on Rendering 2005. http://people.csail.mit.edu/addy/research/brdf/index.html