Exploring Clustering Algorithms in the Appearance Modeling Problem

Fernando Melo Nascimento
Departamento de Computação
Universidade Federal de Sergipe
São Cristóvão – SE, Brazil
Email: nascimentofm@ufs.br

André Britto de Carvalho
Departamento de Computação
Universidade Federal de Sergipe
São Cristóvão – SE, Brazil
Email: andre@ufs.br

Beatriz Trinchão Andrade
Departamento de Computação
Universidade Federal de Sergipe
São Cristóvão – SE, Brazil
Email: beatriz@ufs.br

***Abstract*—Modeling the appearance of a given material is a complex task with many approaches in the literature. One is to measure the BRDF of the material, which is time and resource consuming. One alternative is to use a linear combination of an existent BRDF-basis to approximate the material's appearance. However, the renderization of many BRDF-basis is also consuming, which could be eased by selecting a subset of the materials and use them as basis. Clustering algorithms can perform this task, by grouping BRDFs in clusters, where just one BRDF can represent one entire cluster. We use two classical and one evolutionary clustering algorithms to reduce the number of terms in the linear combination and a NNLS procedure to estimate the contributions of each BRDF-base in the reproduction of a desired material.**

***Keywords*-BRDF; clustering; appearance modeling; genetic algorithm; NNLS, linear combination.**

I. Introduction

In Computer Graphics, appearance modeling involves the simulation of the reflectance properties of different materials. This is a complex phenomenon to reproduce, since several unique characteristics define how a material absorbs or reflects the incident light. In an attempt to define these features analytically, many reflectance models have been created [1].

A BRDF (Bidirectional Reflectance Distribution Function) is a function that aims to model the reflectance of dielectric materials. It has four parameters, corresponding to the direction of the light that arrives at a point (θ_i, ϕ_i) and the direction of the light that comes out of that point (θ_o, ϕ_o). A BRDF calculates the portion of the incident light that is reflected by the material in the direction defined by (θ_o, ϕ_o) [1], [2].

One approach in the literature is to represent a new BRDF with another known BRDFs in a linear combination [2], [3], [4]. This method involves the selection of a set of BRDFs and an optimization process to find the coefficients in this combination. To this end, a BRDF-basis must be defined and a numeric method or an optimization algorithm could be used.

There are some well-known basis of BRDFs in the literature. One of the most significant and widely used in computer graphics is the MERL basis (Mitsubishi Electric Research Laboratories), collected by Matusik et al. [5] and composed of one hundred isotropic materials. This basis could be used to approximate another BRDFs. However, an one hundred terms linear combination is hard to solve and the renderization of 100 BRDFs is time consuming [2], [3]. For this reason, some representative BRDFs could be chosen to reduce the number of terms required in the linear combination.

One way to define a set of representative BRDFs is to use clustering algorithms [2], [4]. This method consists in divide the BRDFs into groups, whose elements inside each group (cluster) have similar reflectance appearance. Doing so, instead of combine 100 materials, only the representative member of each cluster is used in the linear combination.

This work explores clustering algorithms in the Appearance Modeling Problem. Aiming this, we used two classical partitional clustering algorithms (k-means and k-medoids) and one evolutionary clustering algorithm: Genetic Clustering for Unknown K (GCUK). The representative members of each cluster were used as terms of a linear combination to approximate other materials. We validated the solutions using the Davies-Boldin index and minimized the coefficients of the linear combination through a numeric optimization (NNLS). These coefficients indicate how much each representative material contribute in the approximation of other materials.

The rest of the paper is organized as follows: Section II contains the related works, followed by a technical background in the Section III. Section IV describes our approach. The experiments and results are shown in the Sections V and VI. Finally, our conclusions and suggestions for future works are presented in Section VII.

II. Related Work

BRDFs Acquisition: The classical way to measure reflectance properties of a material is through an equipment known as gonioreflectometer [2]. This method reaches high-precision results, but it is highly time and resource consuming [2], [6], [7]. For this reason, alternative methods have been developed, using an image-based approach. Matusik et al. used this approach to collect the BRDFs of 100 isotropic materials, including plastics, metals and fabrics [5]. They used a small sphere of each material to capture 330 HDR pictures of half of the sphere. This process took about 3 hours for each material [5]. Ghosh et al. managed to speed-up the raw data acquisition process, reducing it to few minutes [6]. This time reduction was possible due to an alternative way of measuring the data [6]. However, the results did not reach the same

quality, once the main goal was to develop a faster BRDF measuring.

Reflectance Models: There are several analytical reflectance models with distinct characteristic and goals. Montes and Ureña [7] classified 20 of these reflectance models by its characteristics. The graphical tree classification of the BRDFs in the paper is a way to compare and analyze which model fits better for each application. Another strong point is the explanation of the characteristics presented (or not) by the models.

Other approach is the one used by Brady et al. [8] to find new analytical reflectance models. Instead of trying to derivate them analytically, they used a genetic algorithm – genBRDF – capable of generate reflectance models based on a grammar. The goal is not to replace the human effort of deriving analytical models, as in many cases the output formulas have no meaning without a human interpretation of the terms [8]. Instead, genBRDF could be seen as a support tool in the development of analytic models [8].

Coefficients Minimization: Ngan et al. [9] used the MERL basis to fit the parameters of seven analytical reflectance models through a least squares optimization [10]. They found that the model proposed by Ashikhmin and Shirley [11] reached smaller errors than the other models in many materials. All models were compared using the same parameters, which is somewhat unfair, as the goals and degrees of freedom of the models are very different [9]. A supplemental document [10] presents the formulations used for each model and the minimized parameters for each one of the 100 rendered materials with 1 specular lobe.

Bilgili et al. [12] conducted a similar experiment (using the same basis) with a few different models, including an original one. Their supplemental document [13] contains the parameters of the 100 rendered materials with 3 specular lobes and the peak signal-to-noise ratio (PSNR) – a measure of the quality – for each material.

Clustering BRDFs: Häußler [4] and Carvalho [2] used k-means to generate a subset of expressive BRDFs from MERL basis. The main goal of using clustering in that context is very similar to ours: if all 100 MERL BRDFs were used as basis, the BRDF estimation and rendering would result in longer computation times.

III. Technical Background

A. Appearance Modeling

Materials are generally classified into three distinct categories: isotropic, anisotropic and retro-reflectors [1], [7]. Isotropic materials do not change their appearance if the incidence and outgoing rays of light are rotated by the same angle to the normal vector of a point on the surface [1], [2]. Anisotropic materials, like velvet and brushed metal, have the opposite behavior of the previous condition. Finally, retro-reflective materials are those that reflect light more strongly in the opposite direction of incidence [1].

There are three essential properties for a reflectance model be physically plausible: be non-negative, i. e., light emission should be greater than or equal to zero; conserve energy, which means that the amount of reflected light should be less than or equal to the incident light; and meet the principle of Helmholtz reciprocity: changing the angles of incoming and outgoing light, the result should remain the same [1], [2], [12].

B. Clustering

Clustering is an unsupervised data-mining task that aims to divide an input dataset into clusters with the following property: members of the same cluster have high similarity, whereas the clusters are very separated from each other [14], [15]. Several algorithms have been developed to perform clustering, some of which specific for particular sets of data.

Clustering techniques are divided in two categories: hard clustering, where an element belong to exactly one cluster, and fuzzy clustering, where elements have a level of relevance to each cluster [14], [16]. To determine to which cluster (or clusters) an element belongs, similarity measures (fitness functions) are used. They calculate how close each element is to each cluster [14]. Usually, for simplicity, this function is the Euclidean distance, but there are many others like Manhattan and Mahalanobis distance. Some of them are listed in [14], [16].

One problem with classical clustering algorithms is the obligation of choosing into how many clusters the data set must be divided [15]. One way to overcome this is through a genetic clustering algorithm, capable of finding the optimal value for K – the number of clusters [15]. This kind of algorithm is more appropriate to cluster a BRDF-basis, since we do not know a priori how many terms the linear combination must have to represent well others materials.

IV. Clustering Algorithms Applied to Appearance Modeling

In this work, it was explored three clustering algorithms: k-means, k-medoids and GCUK [17]. The first two are classical algorithms, while the last is a genetic one.

A. Datasets

The two datasets used in this work consist of the minimized parameters by Ngan et al. and Bilgili et al. The eight parameters for the Ashikhmin and Shirley model by Ngan et al. are: $d_r, d_g, d_b, s_r, s_g, s_b, F_0, n$. The first three represent the diffuse RGB values. The next three are the specular RGB values, followed by the Fresnel factor and the exponent n.

Twelve parameters were minimized by Bilgili et al. for the Ashikhmin and Shirley model: $k_{dr}, k_{dg}, k_{db}, k_{sr}, k_{sg}, k_{sb}$, $F_{01}, n_1, F_{02}, n_2, F_{03}, n_3$. The meaning of these parameters are similar to the ones of Ngan et al., differing only because there are three Fresnel factors and exponents.

For both datasets, we only used the Fresnel factors and exponents for similarity analysis, since our clustering intended to group materials whose light reflection were similar, despite of their colors.

B. k-means and k-medoids

The main difference between k-means and k-medoids is the representative member of each cluster: the first uses centroid – the mean of all elements of the cluster – while the second uses medoid, the closest element to the centroid of the cluster. Both algorithms usually use Euclidean distance as fitness function, so both tend to generate hyper spherical clusters [16]. k-medoids, however, has a significant benefit over k-means: the representative member of the cluster is a real element of the input dataset, not a mean, which, normally, does not exists [14], [16]. Algorithms 1 and 2 show the pseudo-code of our implementations of k-means and k-medoids, respectively.

Data: Dataset, K.
Result: Clustering with a validity measure DBindex.
begin
  Generate centroids for each cluster randomly;
  repeat
    Assign each element of the dataset to the nearest cluster using the fitness function;
    Recalculate the centroids;
  until *no cluster changes*;
  Generate DBindex of the clustering;
end

Algorithm 1: k-means

Data: Dataset, K.
Result: Clustering with a validity measure DBindex.
begin
  Generate medoids for each cluster randomly;
  repeat
    foreach *element of the dataset* **do**
      if *this element is not a medoid* **then**
        Trade one medoid with this element;
        Assign each element of the dataset to the nearest cluster using the fitness function;
      end
      Calculate the total cost of this configuration;
    end
    Save the smallest cost;
  until *the smallest cost does not change*;
  Generate DBindex of the clustering;
end

Algorithm 2: k-medoids

The main drawback of both k-means and k-medoids is the obligation of giving the value of K as input. In many datasets, like the ones used in this work, the number of cluster is not known a priori. That means that we need a strategy to use these algorithms. One is to variate the value of K and check the results. Another approach is to use genetic algorithms [15].

C. Genetic Clustering for Unknown K

GCUK is a genetic algorithm proposed by Bandyopadhyay and Maulik [17]. It has as main input the dataset, the size of the population and the number of iterations (generations). It uses the Davies-Boldin index to measure the fitness of each individual solution, being capable to select the best chromosomes and, hence, the best value for K.

GCUK's input is described in the Table I and algorithm 3 contains the pseudo-code of our implementation of GCUK.

TABLE I
INPUT FOR GCUK

Input	Meaning
Dataset	Minimized parameters by Ngan et al. or Bilgili et al.
K_{min}	Minimum number of clusters
K_{max}	Maximum number of clusters
P	Size of the population
G_{max}	Maximum number of generations
$Prob_c$	Probability of crossover
$Prob_m$	Probability of mutation

Data: Dataset, K_{min}, K_{max}, P, G_{max}, $Prob_c$, $Prob_m$.
Result: Clustering with a validity measure DBindex.
begin
  foreach *chromosome in the population* **do**
    Generate a K_i in the range [K_{min}, K_{max}];
    Choose K_i points from the dataset;
    Distribute these points randomly in the chromosome;
  end
  for $i = 0$ **to** G_{max} **do**
    foreach *chromosome in the population* **do**
      Assign each element of the dataset to the nearest cluster using the fitness function;
      Compute DBindex;
      Set fitness as 1/DBindex;
    end
    for $i = 0$ **to** $P/2$ **do**
      Generate two sons by performing crossover with $Prob_c$ in two fathers selected by roulette;
      Perform mutation with $Prob_m$ in both sons;
    end
    Join the original and descendant population;
    Select the best P chromosomes of the new 2P population by roulette;
  end
  Generate DBindex of the clustering;
end

Algorithm 3: GCUK

Each chromosome in GCUK is represented by a vector of K_{max} positions. Each position contains a invalid value (#) or a centroid, represented by F_0 and n – Fresnel factor and exponent, respectively. For simplicity we assume here just one specular lobe (Ngan et al. basis) but the same idea holds for more lobes. For instance, let K_{max} = 10 and K_i = 4. One chromosome in this configuration could be:

C_0 = # (F_0 n) # # (F_0 n) (F_0 n) # # (F_0 n) #

where each (F_0 n) is from one different material of the dataset. Each chromosome represents one possible solution and the number of clusters is defined by K_i.

To define how good one chromosome is, its DBindex is calculated. The fitness is defined as 1/DBindex, i. e., the lower the DBindex, the better the chromosome.

After calculate the fitness of each chromosome in the original population P, the algorithm generate P sons. The first step is choose two fathers in the original population and apply crossover with probability $Prob_c$ to generate two sons. If the crossover does not occur, two new fathers are chosen.

The mutation is applied to each value of valid positions v (F_0 and n). A $\delta \in [0, 1]$ is generate for each v. The mutation is defined by:

$$v = v \times (1 \pm 2\delta), \quad v \neq 0$$
$$v = \pm 2\delta, \quad v = 0$$

Generated P sons, the algorithm select the best P chromosomes in the fathers + sons population by roulette. This individuals survives for the next iteration. After G_{max} iterations, the best chromosome in the final population is the solution.

V. Experiments

We chose the analytical model proposed by Ashikhmin and Shirley [11] to be used in our experiments. It presents many desirable features like: it is widely known, accounts for the Fresnel effect and has a half angle parametrization. Also, this model had its parameters minimized by Ngan et al. [10] and Bilgili et al. [13], reaching smaller errors than most of the models evaluated in both works. We adopted here the formulation used by Ngan et al. [10] for the specular term, presented in Equation 1.

$$\rho_s(L, V, F_0, n) = \frac{(n+1) \times (N \cdot H)^n \times f(F_0, V, H)}{8\pi \times (V \cdot H) \times max((N \cdot L), (N \cdot V))}. \quad (1)$$

This formulation uses Schlick's approximation for the Fresnel term [18], where F_0 represents the reflectance of the material at normal incidence:

$$f(F_0, V, H) = F_0 + (1 - F_0) \times (1 - (V \cdot H))^5. \quad (2)$$

As diffuse term, we used the Lambertian BRDF constant:

$$\rho_d(L, V) = \frac{1}{\pi}. \quad (3)$$

The choice of K: To overcome the drawback of choosing a fixed value for K as input of k-means and k-medoids, we have done an experiment based on the work of Carvalho [2] and Häußler [4] where K assumed the values 5, 10, 20, 30, 40, 50, 60 and 70. Here, we only varied K among 2, 5, 10, 15 and 20, since the errors found by Carvalho stood practically the same for higher values of K [2].

Executions: We executed k-means and k-medoids one thousand times for each value of K ($K \in \{2, 5, 10, 15, 20\}$) for both basis. For GCUK algorithm we defined the size of population P = 50, the number of generations G_{max} = 100 and the crossover and mutation rates 90% and 1%, respectively. We performed two thousand executions for each basis: half with K_{min} = 2 and K_{max} = 50 and the other thousand with K_{min} = 5 and K_{max} = 50.

Linear Combination: We performed a non negative least squares (NNLS) optimization to solve the linear combination given as:

$$a_1 M_1 + a_2 M_2 + ... + a_K M_K = B. \quad (4)$$

where a_i are the coefficients we want to optimize, M_i are the representative materials obtained in the clustering process and B is the material we want to approximate. The number of terms K vary accordingly with the quantity of clusters found by clustering the basis. M_i and B values follows the BRDF representation R:

$$R = d_c \times \rho_d(L, V) + \sum_{j=1}^{m} (s_c \times \rho_s(L, V, F_j, n_j)). \quad (5)$$

where m is the number of specular lobes and c is the color channel ($c \in \{r, g, b\}$).

The values d_c, s_c, F_j and n_j are taken from Ngan et al. and Bilgili et al. basis. The diffuse and specular terms are, respectively, Lambertian constant (Equation 3) and the Ashkhimin and Shirley model (Equation 1).

VI. Results and Discussion

A. Numerical Clustering Results

Figures 1 and 2 show box-plots of all DBindex values for each thousand execution of k-means, k-medoids and GCUK for Ngan et al. and Bilgili et al. basis, respectively.

We can observe in Figures 1 and 2 that DBindex is generally lower when small values of K are used in k-means and k-medoids. Strictly analyzing these values, one could assume that the best value for K is 2, since a lower DBindex implies, theoretically, a better clustering. However, some points must be considered:

1) A linear combination with just 2 terms may not have the needed degrees of freedom to approximate any material we want.
2) The difference of DBindex median from K = 2 to K = 5 is no bigger than 0.5 for k-means and less than 1.25 for k-medoids.
3) A small value of DBindex do not necessarily implies in a good clustering.
4) The minimum DBindex of each value of K in both algorithms and basis are very close in most cases.

For GCUK algorithm, DBindex values are lower than those for k-means and k-medoids. Even if we consider that all solutions of GCUK have only 2 clusters as output this result of GCUK is much better. The number of clusters found by GCUK is shown in Table II.

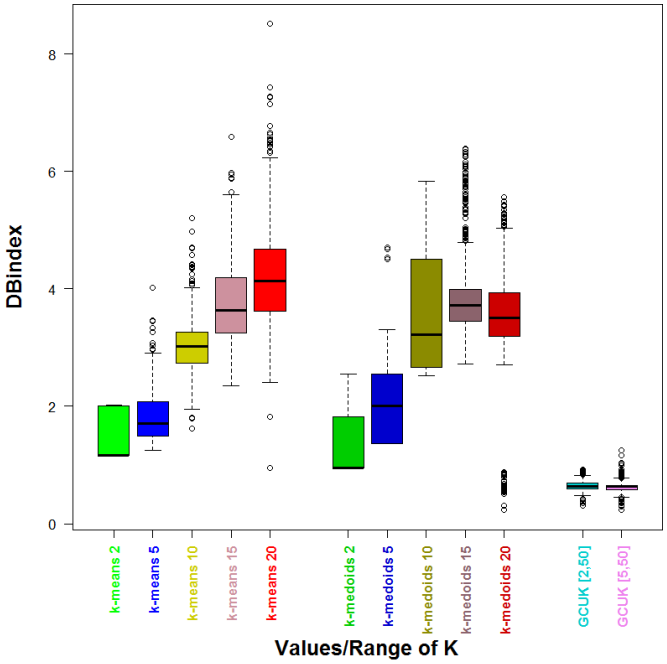

Fig. 1. Box-plot of DBindex values for Ngan et al. basis

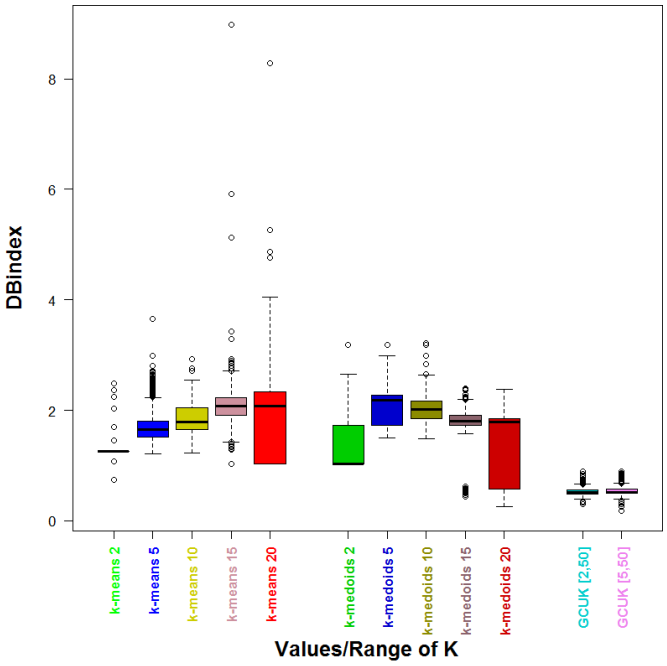

Fig. 2. Box-plot of DBindex values for Bilgili et al. basis

Based on the these results, we could assume that the ideal number of clusters vary between 2 and 12. However, with just two terms the linear combination may not have the freedom needed to represent a large variety of materials. For this reason, we opted for 5 as an ideal number of clusters.

B. Coefficients Optimization with NNLS

We chose four different materials from Ngan et al. basis to be represented by a linear combination (Equation 4) of our basis: Blue Rubber, Fabric Beige, Hematite and Orange BBall. Their parameters are shown in Table VIII, while the parameters of our basis are in the Table VII. This basis was generated by GCUK algorithm with K $\in [5, 50]$ and has DBindex = 0.831692.

TABLE II
MEAN OF VALUES OF K FOUNDED BY GCUK ALGORITHM

Basis	K range	Mean of K	Maximum K	Minimum K
Ngan	$[2, 50]$	2.646	5	2
Ngan	$[5, 50]$	2.863	12	2
Bilgili	$[2, 50]$	2.210597826	6	2*
Bilgili	$[5, 50]$	2.904827586	12	2*

Tables III to VI present the coefficients estimated with NNLS for the chosen materials. They indicate how much each material in our basis contribute to approximate each material. Materials that did not contribute in any channel were omitted.

TABLE III
CONTRIBUTIONS OF EACH MATERIAL FOR BLUE RUBBER

Material	Red	Green	Blue
M_4	0.000201	0.000233	0.000194
M_5	1.342572	1.740014	2.518673

TABLE IV
CONTRIBUTIONS OF EACH MATERIAL FOR FABRIC BEIGE

Material	Red	Green	Blue
M_5	1.392117	1.037895	0.921093

TABLE V
CONTRIBUTIONS OF EACH MATERIAL FOR HEMATITE

Material	Red	Green	Blue
M_2	1.286881	1.950771	1.816894
M_4	0.000011	0.000025	0.000010

TABLE VI
CONTRIBUTIONS OF EACH MATERIAL FOR ORANGE BBALL

Material	Red	Green	Blue
M_1	0.090207	0.084581	0.000610
M_2	0	0	0.027084
M_4	0.076629	0.075486	0.117023
M_5	2.593467	0.229267	0

We can observe that Fabric Beige could be approximate with just one basis material, while Orange BBall had contributions of four materials. This scenario shows the importance of having a basis with more than 2 materials.

C. Visual Clustering Results

We can observe in Figure 3 how the basis in the Table VII was clustered by GCUK algorithm. The highlighted materials are the ones closest to their cluster centroid.

Materials were grouped by their specular lobes, and not by their colors. However, in cluster 4 many elements with different specular properties were clustered together. Additionally, clusters 1 and 2 have only one element, while cluster 4 has 70. We justify this by the use of Euclidean Distance as fitness function, which may not be the best approach to our problem. To adopt a similarity measure capable of modeling reflectance properties, instead of one based on a geometric distance, could improve the clustering process.

VII. Conclusion

In this paper, we show the application of clustering algorithms in the Appearance Modeling problem. We divided a BRDF-basis into clusters using classical clustering algorithms and one genetic algorithm, aiming to reduce the numbers of terms needed in a linear combination of BRDFs.

We found that 5 is a good value for K, since in all three algorithms, the DBindex for this value is low. Other point to consider is that a linear combination with 5 different materials have a good degree of freedom, being capable to represent well a variety of other materials.

This way, as future work, we suggest trying different fitness functions and validity measures in the clustering algorithms and using a genetic algorithm like PSO for minimization of the linear combination. Also, a renderization of the combination of our BRDF-basis could help measuring its expressiveness.

Acknowledgment

The authors would like to thank PIBIC-COPES for the financial support for this research.

TABLE VII
GCUK-basis with 5 materials

Attributes	M_1	M_2	M_3	M_4	M_5
d_r	0.30800	0.0401	0.0356	0.075900	0.000072
d_g	0.24700	0.0269	0.0308	0.040700	0.000086
d_b	0.15000	0.0143	0.0249	0.028600	0.000196
s_r	0.01550	0.0549	0.0490	1.040000	0.032500
s_g	0.00952	0.0315	0.0361	0.609000	0.026600
s_b	0.00885	0.0235	0.0178	0.266000	0.018000
F_0	0.99900	0.9990	0.9990	0.080732	0.999000
n	78200	66300	39800	19842.66211	22.643148

TABLE VIII
Coefficients of the chosen Materials

Attributes	Blue Rubber	Fabric Beige	Hematite	Orange BBall
d_r	0.035800	0.199000	0.036800	0.323000
d_g	0.064900	0.112000	0.032000	0.045700
d_b	0.092700	0.066400	0.022200	0.003970
s_r	0.341000	0.197000	0.059900	0.039800
s_g	0.281000	0.136000	0.052100	0.022900
s_b	0.187000	0.082200	0.036200	0.015500
F_0	0.053200	0.973000	0.999000	0.140000
n	20	0.5	78200	42600

Fig. 3. GCUK Clustering – 5 clusters

References

[1] T. Weyrich, J. Lawrence, H. P. A. Lensch, S. Rusinkiewicz, and T. Zickler, "Principles of appearance acquisition and representation," *Found. Trends. Comput. Graph. Vis.*, vol. 4, no. 2, pp. 75–191, 2009.
[2] B. T. A. de Carvalho, "Image-based appearance preservation," Ph.D. dissertation, Universade Federal do Paraná, 2013.
[3] H. P. A. Lensch, "Efficient, image-based appearance acquisition of real-world objects," Ph.D. dissertation, 2003.
[4] T. Häußler, "Mobile reflectance estimation," Master's thesis, Universität Ulm, Fakultät für Ingenieurwissenschaften und Informatik, 2012.
[5] W. Matusik, H. Pfister, M. Brand, and L. McMillan, "A data-driven reflectance model," *ACM Trans. Graph.*, Jul. 2003.
[6] A. Ghosh, S. Achutha, W. Heidrich, and M. O'Toole, "Brdf acquisition with basis illumination," pp. 1–8, 2007.
[7] R. Montes and C. Ureña, "An overview of BRDF models," 2012.
[8] A. Brady, J. Lawrence, P. Peers, and W. Weimer, "genBRDF: Discovering new analytic BRDFs with genetic programming," *ACM Trans. Graph.*, vol. 33, no. 4, pp. 114:1–114:11, Jul. 2014.
[9] A. Ngan, F. Durand, and W. Matusik, "Experimental analysis of brdf models," in *Proceedings of the Sixteenth Eurographics Conference on Rendering Techniques*, ser. EGSR '05, 2005, pp. 117–126.
[10] ——, "Experimental analysis of brdf models - supplemental," 2005.
[11] M. Ashikhmin and P. Shirley, "An anisotropic phong brdf model," *J. Graph. Tools*, vol. 5, no. 2, pp. 25–32, Feb. 2000.
[12] A. Bilgili, A. Öztürk, and M. Kurt, "A general BRDF representation based on tensor decomposition," *Comput. Graph. Forum*, vol. 30, no. 8, pp. 2427–2439, 2011.
[13] ——, "A general BRDF representation based on tensor decomposition: Supplemental material," 2011.
[14] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: A review," *ACM Comput. Surv.*, vol. 31, no. 3, pp. 264–323, Sep. 1999.
[15] E. R. Hruschka, R. J. G. B. Campello, A. A. Freitas, and A. C. P. L. F. De Carvalho, "A survey of evolutionary algorithms for clustering," *Trans. Sys. Man Cyber Part C*, vol. 39, no. 2, pp. 133–155, Mar. 2009.
[16] R. Xu and D. Wunsch, II, "Survey of clustering algorithms," *Trans. Neur. Netw.*, vol. 16, no. 3, pp. 645–678, May 2005.
[17] S. Bandyopadhyay and U. Maulik, "Genetic clustering for automatic evolution of clusters and application to image classification," *Pattern Recognition*, vol. 35, no. 6, pp. 1197–1208, 2002.
[18] C. Schlick, "An inexpensive BRDF model for physically-based rendering," *Computer Graphics Forum*, vol. 13, pp. 233–246, 1994.