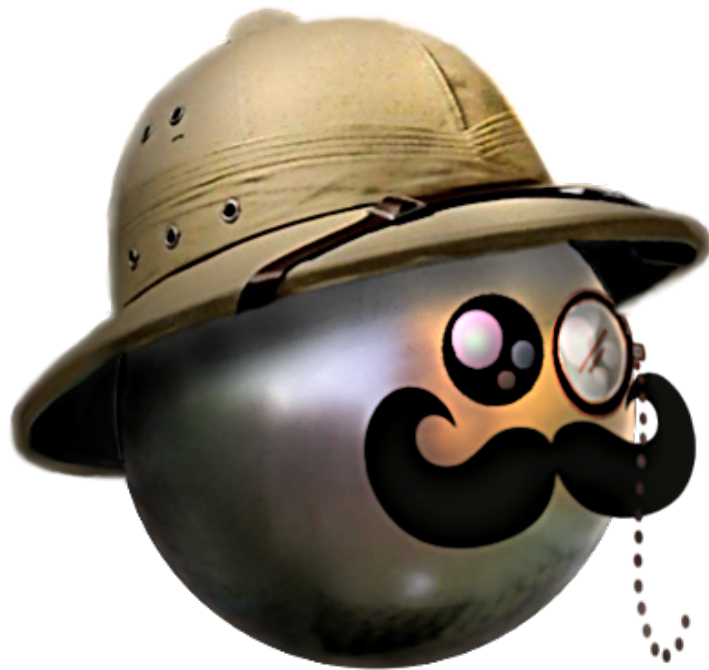


Exploration de l'espace des BRDFs pour la modélisation de matériaux

Rapport de Recette

.



Adrien Grosjean
Ludovic Burg
Karim Salama
Adèle Saint-Denis

Superviseur : Mathias Paulin

20 février 2019

Introduction

Dans le cadre du Master 2, nous avons choisi de réaliser notre chef d'oeuvre sur le sujet basé sur le papier *A Versatile Parameterization for Measured Material Manifolds* de [Soler et al., 2018]. Celui-ci propose une solution au problème suivant :

Lors du processus de design de modèles et de scènes 3D, les graphistes doivent être capable de choisir le matériau qui correspond à leur attentes. Pour ce faire ils ont besoin d'outils leur permettant de visualiser et de modifier ces matériaux. Le logiciel *BRDF Explorer* de Disney répond à ces besoins en proposant une visualisation d'une centaine de matériaux dont les données proviennent de la [base MERL](#).

Le problème réside dans le fait que ces représentations manquent d'ergonomie : d'une part la navigation dans la liste des 100 matériaux est pénible, et d'autre part les paramètres fournis pour les modifier sont insuffisants. L'idée du papier cité plus haut pour résoudre ces deux problèmes est de créer un espace de dimension q inférieur permettant d'explorer intuitivement l'espace des matériaux. Le choix de q dépend de la qualité souhaitée et de la manière dont nous souhaitons naviguer dans l'espace. Un espace 2D à l'avantage de pouvoir être représenté sous la forme d'une carte et d'être simple à visualiser.

Notre projet s'est déroulé en 2 temps : Dans un premier temps, nous avons implémenté l'algorithme permettant de fournir une paramétrisation optimale de l'espace des données mesurées par le MERL vers un espace à q dimensions : une variété topologique à q dimensions de l'espace des BRDFs.

En choisissant q égal à 2 nous pouvons générer le modèle de l'espace correspondant ainsi qu'une image représentant cet espace 2D. Ce faisant, nous obtenons une carte que les graphistes pourront explorer. Notre modèle lui, permet de reconstruire des données semblables à celles du MERL, c'est à dire produire un nouveau matériau qui n'existait pas dans la base à partir d'un point de la carte, le tout implémenté dans *BRDF Explorer*.

En mettant des points de référence sur la carte (les points de la variété topologique correspondant aux données d'origine), le graphiste peut alors aisément choisir un matériau entre l'or et l'aluminium par exemple.

BRDF3000

BRDF3000 est le logiciel permettant la réduction de dimensions de l'espace des BRDFs en une variété topologique de dimension q .

I.1 Présentation du logiciel et de son fonctionnement

BRDF3000 permet de produire un certain nombre de données, dont la paramétrisation de l'espace calculée à partir de l'optimisation faite sur la fonction de coût présentée dans [Soler et al., 2018], le papier qui est à l'origine de notre chef d'oeuvre. Les données qui définissent la paramétrisation sont écrites dans un fichier que nous avons détaillé dans le rapport de spécifications.

De plus, ce logiciel produit aussi une image au format *.bmp* lorsque la dimension de la variété topologique est de 2. Cette image représente l'espace latent des BRDFs dont les coordonnées appartiennent au carré de côté 8 centré sur 0 : au-delà, la paramétrisation converge vers la BRDF moyenne. Enfin, chaque pixel de l'image résultante représente l'albédo de la BRDF reconstruite à partir des coordonnées du pixel dans le repère.

Le programme dans son ensemble est hautement parallélisé et n'effectue aucune copie mémoire.

Le main fait appel aux différents modules pour, dans l'ordre :

- Lire la base de donnée et produire les matrices nécessaires pour l'optimisation
- Lancer l'optimisation du manifold et récupérer les données de la paramétrisation
- Créer un objet permettant de reconstruire des BRDFs et évaluer l'erreur de reconstruction
- Ecrire une BRDF reconstruite correspondant à une donnée d'origine
- Ecrire les données de la paramétrisation
- Calculer et écrire la carte

Tout ce processus est ponctué d'affichage le quel donne les temps d'exécution des tâches et un pourcentage d'avancement pour le calcul de la carte.

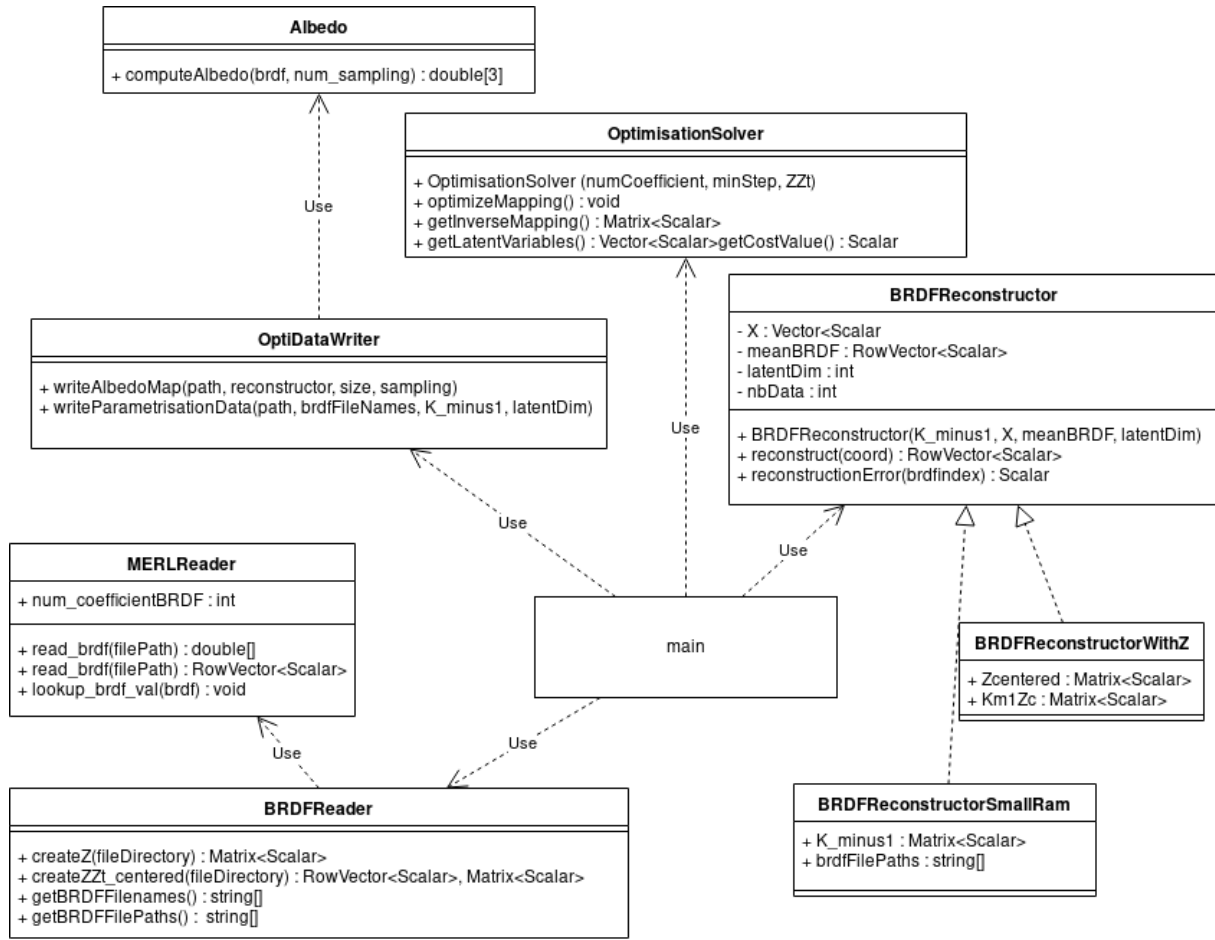


Figure I.1 – Diagramme de classe de *BRDF3000*.

Le programme offre deux approches pour les calculs matriciels :

- Une gourmande en mémoire (7.5 GB) mais rapide à l'exécution. On charge toutes les données (Z) dans la mémoire vive et on précalcule $K^{-1} * Z^c$ pour la reconstruction.
- Une plus économe mais plus lente. On fait les calculs impliquant Z dans des boucles lors desquels les BRDFs sont chargées depuis le disque puis remplacées par les suivantes.

NB : L'approche économe était à l'origine censée être implémentée via la librairie *stxxl* couplée à la classe *Map* de *Eigen* mais cela ne s'est pas avéré être possible car *stxxl* ne donne pas accès à un pointeur vers les données.

I.2 Manuel d'utilisation

Ce logiciel ne possède pas d'interface graphique. À l'exécution du logiciel dans un terminal, l'utilisateur peut spécifier certains paramètres.

I.2.1 Comportement par défaut

Lorsqu'aucun paramètre n'est spécifié au programme, *BRDF3000* effectue l'optimisation sur toutes les BRDFs de la base de données de MERL, contenues dans le dossier *data* du projet. De plus, les calculs seront effectués sur des *long double*. Enfin, *q* sera égale à 2. Avec cette dimension de l'espace latent, une image sera calculée sur 200x200 pixels.

I.2.2 Les paramètres

-h | -help

Cette option affiche l'aide et l'explication des options.

-d < entier non signé >

Cette option permet de spécifier la dimension de l'espace latent sur lequel on calcule l'optimisation. Par défaut la dimension de l'espace latent est de 2.

-m < entier non signé >

Cette option permet de spécifier la taille de la carte des matériaux calculée. La carte est forcément carrée. Par défaut, sa taille est 200 pour calculer une carte de 200x200 pixels

-b <Le chemin d'un dossier où se trouvent les BRDFs de référence>

Cette option sert à spécifier le chemin du dossier dans lequel se trouvent les BRDFs de référence. Ce dossier ne doit contenir que des BRDFs au format *MERL*. Par défaut on charge les BRDFs du dossier *../data*.

-smallRam

Cette option permet de faire en sorte que l'optimisation et la création de la carte des matériaux soit faite en utilisant peu de mémoire vive.

Sans cette option, avec les 100 BRDFs de la base de données MERL comme BRDFs de référence, le programme requiert au moins 7.5 GB de mémoire vive. Le chargement de *Z* et le calcul de la carte des matériaux sera cependant plus rapide.

I.3 Les résultats du programme

I.3.1 Optimisation de la paramétrisation et reconstruction de matériaux

Les résultats et la durée du processus d'optimisation dépendent :

- Du nombre de données (nombre de BRDFs)
- De la précision choisie pour les nombres flottants

Ceux-ci sont évalués via l'erreur de reconstruction à partir des variables latentes. Le calcul d'erreur choisit est celui de l'erreur quadratique moyenne au carré.

Le fichier de paramétrisation qui a été retenu pour BRDFExplorer3000 a été produit avec des flottants sur 128 bits grâce à la librairie *boost multiprecision*

Résultat de l'optimisation du set MERL entier				
Précision	Temps d'optimisation* (sec)	Erreur de reconstruction	Charge mémoire avec Z	Charge mémoire en mode <i>small ram</i>
double	0.19	0.027	~6.7 GB	~400Mo
long double	1.8	1E-11	~13 GB	~700Mo
float128	52	4E-41	~14 GB	X
* sans compter le précalcul de ZZt				

Figure I.2 – Résultats de *BRDF3000*

I.3.2 Calcul de la carte

Nous avons produit plusieurs cartes pour déterminer l'intervalle latent à représenter et dimensionner l'échantillonnage de l'hémisphère ainsi que la résolution spatiale. La carte ci-dessous a été produite au terme de 12h de calcul sur une machine personnelle, avec un échantillonnage de l'hémisphère de la surface des matériaux : 16 valeurs pour θ_i et θ_o et 64 valeurs pour ϕ_i et ϕ_o . Chaque pixel représente l'albédo calculé grâce au ré-échantillonnage des BRDFs aux coordonnées correspondantes.

I.4 Les résultats des tests

I.4.1 Le module Paramétrisation

Nous avons testé la fonction qui calcule la covariance entre deux variables latentes. Nous avons également testé le centrage de la matrice Z . Les résultats des fonctions sont les mêmes que les résultats escomptés (écrits dans des fichiers de référence et comparés à l'exécution)

I.4.2 Le module Optimisation

Nous avons testé le calcul de l'inverse de K et de son déterminant avec la formule de Sherman-Morison et du Lemme du déterminant de la matrice. Pour cela, nous avons généré des jeux de tests avec le logiciel Octave. La matrice K inversée et le calcul de son déterminant avec Octave ont été comparés avec les résultats de *BRDF3000*. Nous obtenons des erreurs très faibles, de l'ordre de 10^{-12} . Il se pourrait que cette erreur soit causée par Octave, qui manquerait de précision. *BRDF3000* peut aussi être responsable de ces erreurs. Les deux peuvent être responsables. Cependant, comme l'erreur est très faible, cela nous assure que notre programme fonctionne.

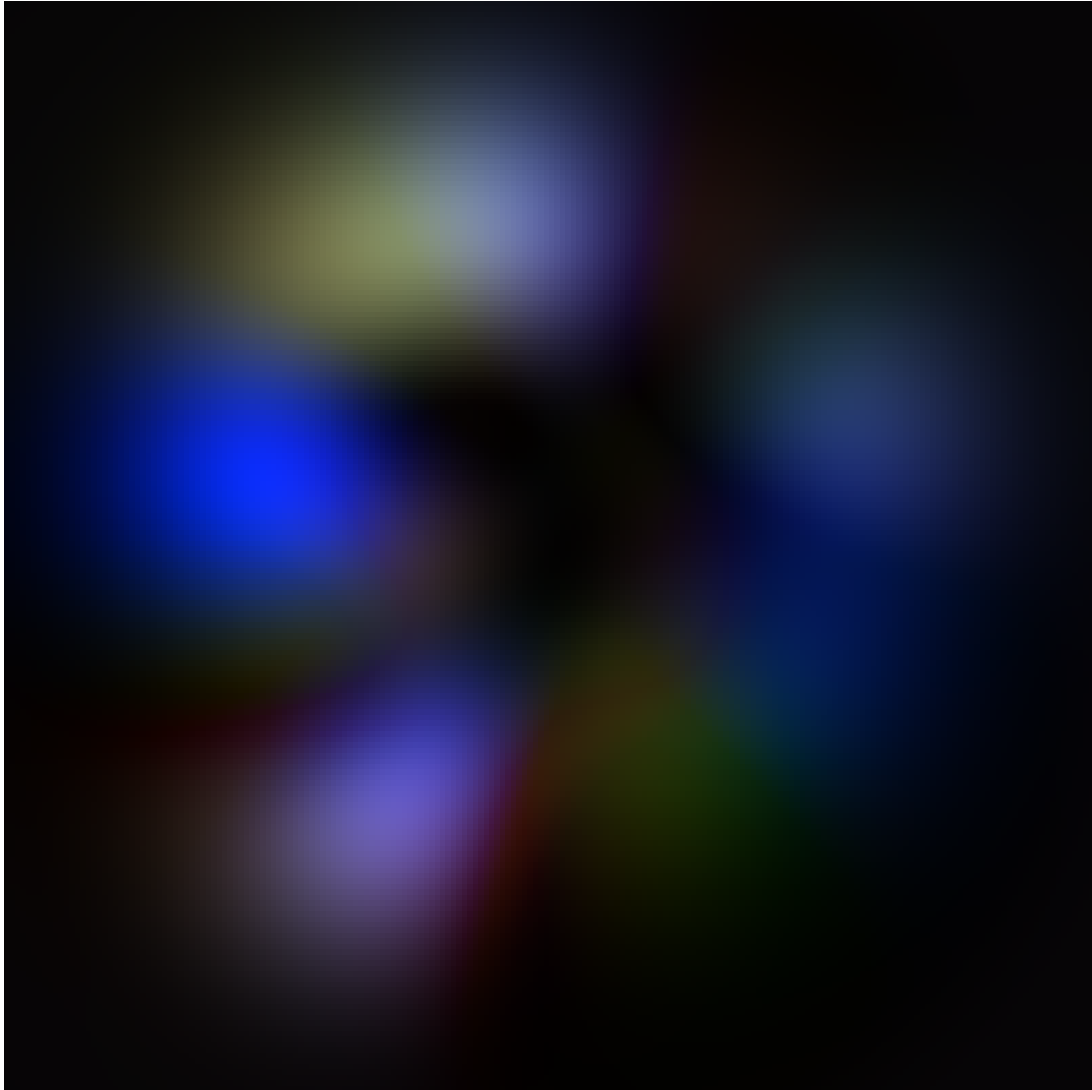


Figure I.3 – Carte des matériaux générée par *BRDF3000*. Les matériaux de référence sont les cent BRDFs de la base de données MERL

I.4.3 La lecture des BRDFs

Pour générer les résultats escomptés, nous avons utilisé le code fourni par MERL.

La fonction *Create ZZt centered* construit $Z \times Z^t$ centrée en utilisant peu de mémoire vive. Nous avons comparé son résultat avec la construction de $Z \times Z^t$ centré en utilisant la méthode qui charge Z en mémoire. Nous voyons qu'ils n'ont pas les mêmes résultats. Cependant, les coefficients de la matrice $Z \times Z^t$ sont très grands. De plus, l'erreur de reconstruction des BRDFs en chargeant Z en mémoire ou non est comparable.

I.4.4 Erreur de reconstruction

Les figures I.5 et I.6 montrent une erreur de reconstruction faible avec les deux méthodes que nous avons implémentées.

```

Testing the Parametrisation module:
| Covariancel: [OK]
| Covariance2: [OK]
| Center1: [OK]
| Center2: [OK]
| [OK] 4 tests out of 4 have been successfully executed
Testing the Optimisation module:
| Sherman Morisson update 1: 3.66374e-15[KO] Test failed
| Sherman Morisson update 2: 1.50191e-12[KO] Test failed
| Cost 1: -4.44089e-16[KO] Test failed
| Cost 2: -4.26326e-14[KO] Test failed
| Cost 3: [OK]
| [KO] 1 tests out of 5 have been successfully executed
Testing the BRDFReader module:
| Read BRDF: [OK]
| Create Z: [OK]
| Create ZZt centered: 512[KO] Test failed
| [KO] 2 tests out of 3 have been successfully executed

```

Figure I.4 – Le résultats des tests.

```

Reconstruction error for cherry-235.binary : 3.68765e-22
Reconstruction error for pink-fabric.binary : 1.54694e-21
Reconstruction error for silver-metallic-paint2.binary : 1.79986e-21
Reconstruction error for natural-209.binary : 1.19356e-22
Reconstruction error for chrome.binary : 2.98138e-22

```

Figure I.5 – Le résultats des tests avec la méthode qui charge Z en mémoire.

```

Reconstruction error for cherry-235.binary : 2.47582e-22
Reconstruction error for pink-fabric.binary : 8.94782e-22
Reconstruction error for silver-metallic-paint2.binary : 1.08776e-21
Reconstruction error for natural-209.binary : 6.3959e-23
Reconstruction error for chrome.binary : 1.96027e-22

```

Figure I.6 – Le résultats des tests avec la méthode qui ne charge pas Z en mémoire.

I.4.5 Tests de validation

Le chargement dans BRDFExplorer d'une BRDF reconstruite à partir de variables latentes et la comparaison avec le fichier d'origine a eu valeur de test de validation pour la paramétrisation produite : le calcul de l'erreur de reconstruction pouvait lui-même être faux.

I.5 Les améliorations possibles

Laisser le choix de la précision des calculs à l'utilisateur

Actuellement, la précision des calculs est définie dans le code sous la forme de "*using Scalar = type*" avec *type* un *long double*. Nous pourrions améliorer *BRDF3000* en laissant le choix de ce type à l'utilisateur.

Implémenter le calcul de la carte des matériaux avec Cuda

La parallélisation du calcul de la carte des matériaux est actuellement faite avec *OpenMP*. Hélas, le calcul d'une carte 200x200 avec cette version prend environ 12 heures à calculer. Afin d'accélérer ce calcul, nous pourrions effectuer cette parallélisation avec Cuda.

Proposer une meilleure représentation de la carte des matériaux

Le problème de la carte [I.3](#), c'est que la plupart des BRDFs sont regroupées dans la zone sombre au centre de la carte : on manque d'information sur leur couleur. Cela est dû au fait que la grande majorité des matériaux ne sont pas très spéculaires : leur pouvoir réfléchissant est faible et l'échantillonnage limité ne permet pas de faire ressortir la couleur diffuse. Cela nuit à la bonne compréhension de l'espace et donc à son exploration. Un possible vecteur d'amélioration serait donc soit de trouver un meilleur ré-échantillonnage permettant d'obtenir un albédo plus représentatif, soit de produire une image avec un échantillonnage spatial proportionnel à la concentration en variable latentes, ou encore d'utiliser une représentation de cet espace légèrement différente de l'albédo.

BRDFExplorer3000

BRDFExplorer3000 est une version augmentée de *BRDFExplorer*, le logiciel de *Disney* permettant de visualiser des BRDFs. Nous y avons implémenté une interface permettant de naviguer sur la carte des matériaux en se servant de la paramétrisation : deux éléments produits par *BRDF3000*. De fait, nous ne décrivons dans cette section que l'interface et le code que nous avons ajouté au logiciel initial.

II.1 Présentation du logiciel et de son fonctionnement

La spécificité de *BRDFExplorer3000* par rapport à *BRDFExplorer* est qu'il propose le choix d'une BRDF sur la carte des matériaux, la reconstruction de la BRDF choisie et la visualisation de ce matériaux en utilisant le système de *BRDFExplorer*.

Lors du premier appel de l'ouverture de la carte pour faire un choix de BRDF, la fenêtre de la figure II.1 s'ouvre pour proposer à l'utilisateur deux méthodes pour charger les données et reconstruire les BRDFs.

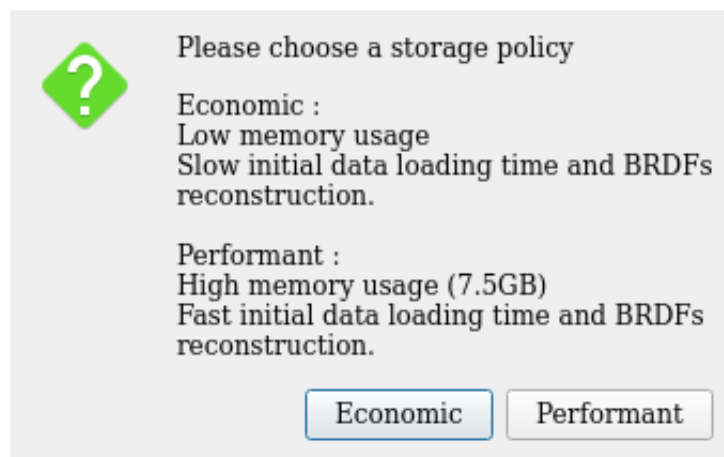


Figure II.1 – L'utilisateur se voit proposé deux options : utiliser peu de mémoire vive ou avoir un faible de temps de calcul.

La première méthode utilise peu de mémoire vive. Cependant, elle prendra plus de temps de calcul. Ainsi, avec les cent matériaux de la base de données *MERL* comme

matériaux de référence, la reconstruction prendra 15 secondes. En échange, seulement 33 Mo de mémoire vive seront requis.

La deuxième méthode, au contraire, utilisera beaucoup de mémoire vive mais sera plus rapide. Avec les cent matériaux de la base de donnée *MERL* comme matériaux de référence, 7.5 GB de mémoire vive seront requise. La reconstruction prendra moins de deux secondes avec un ordinateur qui possède quatre coeurs.

Les données du fichier de paramétrisation ainsi que la carte produite par *BRDF3000* sont chargés. Une barre de progression indique à l'utilisateur la progression du chargement des données dans le système, comme sur la figure II.2.

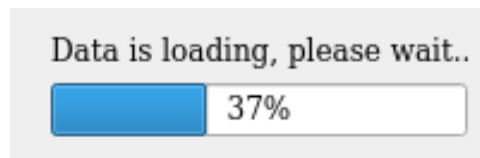


Figure II.2 – Durant le chargement de la carte des matériaux, cette barre de progression s'affiche.

La carte est une version améliorée de l'originale produite par *BRDF3000* : après l'avoir agrandie artificiellement, nous avons tracé les isolignes de la luminance grâce à un shader puis avons rehaussé les couleurs pour augmenter la visibilité. Les isolignes donnent une information sur la courbure du manifold et nous pensons qu'elles améliorent l'expérience utilisateur.

La validation d'une BRDF sur la carte entraîne la reconstruction de cette dernière, un gif animé indique à l'utilisateur que le système est en train de reconstruire la BRDF II.3.

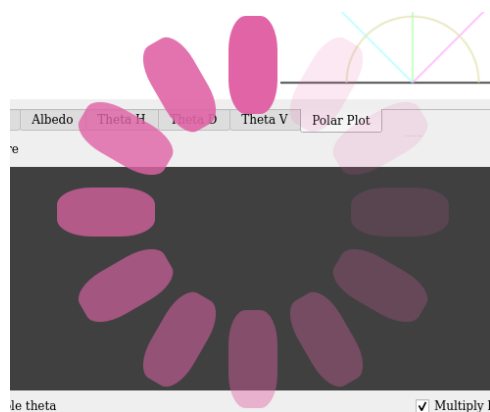


Figure II.3 – Durant la reconstruction d'une BRDF, une animation à lieu pour indiquer à l'utilisateur que le logiciel est en train de calculer.

II.2 Manuel d'utilisation

II.2.1 Fonctionnement de la carte de matériaux

Accéder à la carte

Dans la barre de menu en haut de la fenêtre se trouve un item *Choose BRDF from Map* qui ouvre la carte.

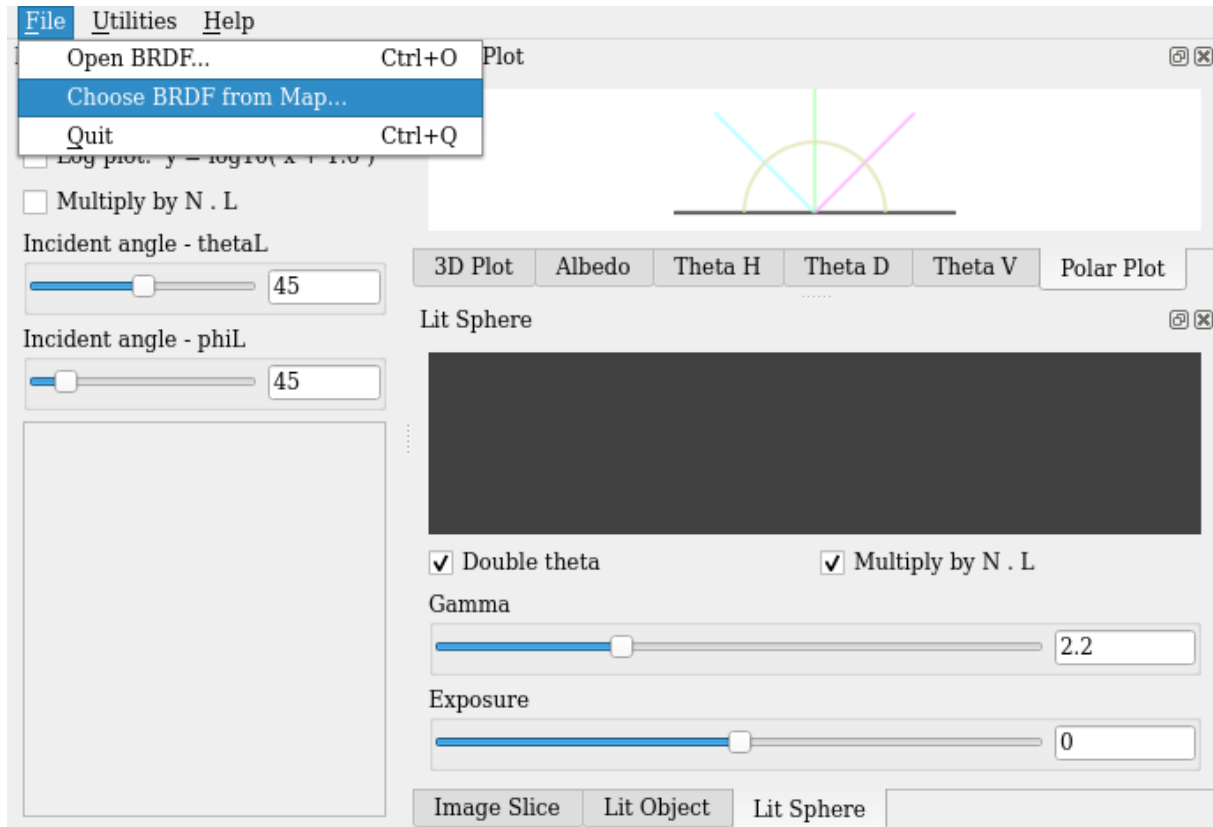


Figure II.4 – Interface pour ouvrir la carte des matériaux.

Zoomer sur la carte

Une fois que la carte est ouverte, il est possible de zoomer sur cette dernière en utilisant la molette de la souris

Afficher le nom des BRDFs de référence sur la carte

Pour s'orienter dans la carte par rapport aux BRDFs de référence, il y a l'emplacement des BRDFs de référence. Pour éviter de surcharger l'interface, les noms des BRDFs ne sont pas affichés. Il est possible d'afficher le nom de toutes les BRDFs de référence en restant appuyé sur *Ctrl*, comme sur la figure II.5. Il est aussi possible d'afficher le nom d'une BRDF de référence en passant la souris sur son point de référence sur la carte, comme on peut le voir sur la figure II.6.



Figure II.5 – Les noms des BRDFs sont affichés sur la carte lorsque le bouton *Ctrl* est appuyé.

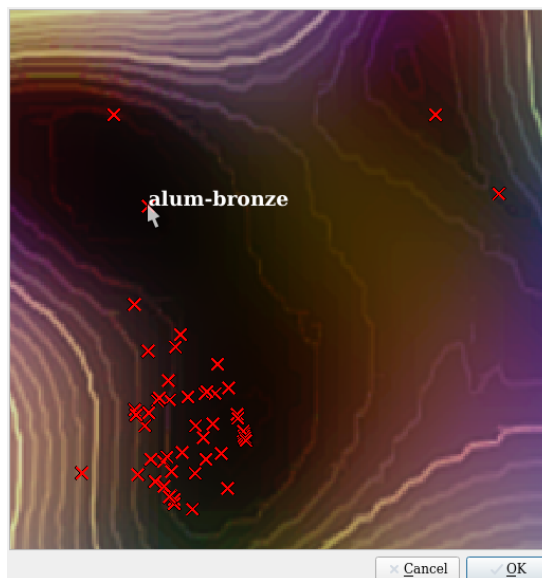


Figure II.6 – Interface pour ouvrir la carte des matériaux.

Sélectionner une BRDF à partir de l'espace latent

Afin de choisir une BRDF à reconstruire, il est possible d'effectuer un clic sur la carte. Cela pose un point sur la carte, comme sur la figure II.7.

Reconstruction d'une BRDF sélectionnée

Après avoir effectué un clic sur la carte pour sélectionner une BRDF à reconstruire, le bouton "ok" s'active, permettant de valider le choix de la BRDF à reconstruire et ainsi l'afficher dans l'interface de *BRDF Explorer 3000*.

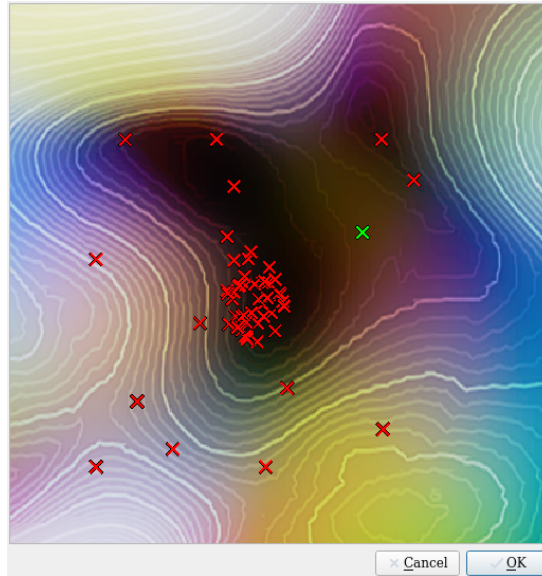


Figure II.7 – Un marqueur vert est placé sur la position du clic de l'utilisateur.

II.3 Résultats des tests

Après avoir mis à jour le diagramme UML de cas d'utilisation pour l'interface de BRDFExplorer3000 II.9, nous avons testé ce diagramme et validé chaque étape au fur et à mesure du développement de l'application.

Nous avons également recalculé l'erreur de reconstruction car dans BRDFExplorer3000 les calculs sont faits (pour des raisons de rapidité) avec des longs doubles alors que la paramétrisation est produite avec des flottants 128 bits à l'origine. Résultat : les erreurs sont de l'ordre de 10^{-20} là où elles avoisinaient 10^{-41} côté BRDF3000. Les erreurs de cet ordre là ne sont pas perceptibles.

II.4 Les améliorations possibles

Calcul dynamique des isolignes

Les isolignes que nous avons tracées sont parties de l'image de la carte elle-même. De ce fait, lorsque l'on zoome, on agrandit aussi les isolignes. Pour palier à ce problème, nous pourrions les calculer dynamiquement dans un shader intégré à l'interface : elles auraient une épaisseur indépendante du zoom.

Ajouter la possibilité à l'utilisateur de sauvegarder les BRDFs reconstruites

Dans le cas où l'utilisateur a reconstruit une BRDF qui l'intéresse, il aurait la possibilité de l'enregistrer au format MERL.

Donner la possibilité à l'utilisateur de poser des "flags" sur la carte

Dans le cas où l'utilisateur a trouvé une BRDF qui l'intéresse et souhaite simplement se souvenir de sa position sur la carte de matériaux, il aurait la possibilité d'y

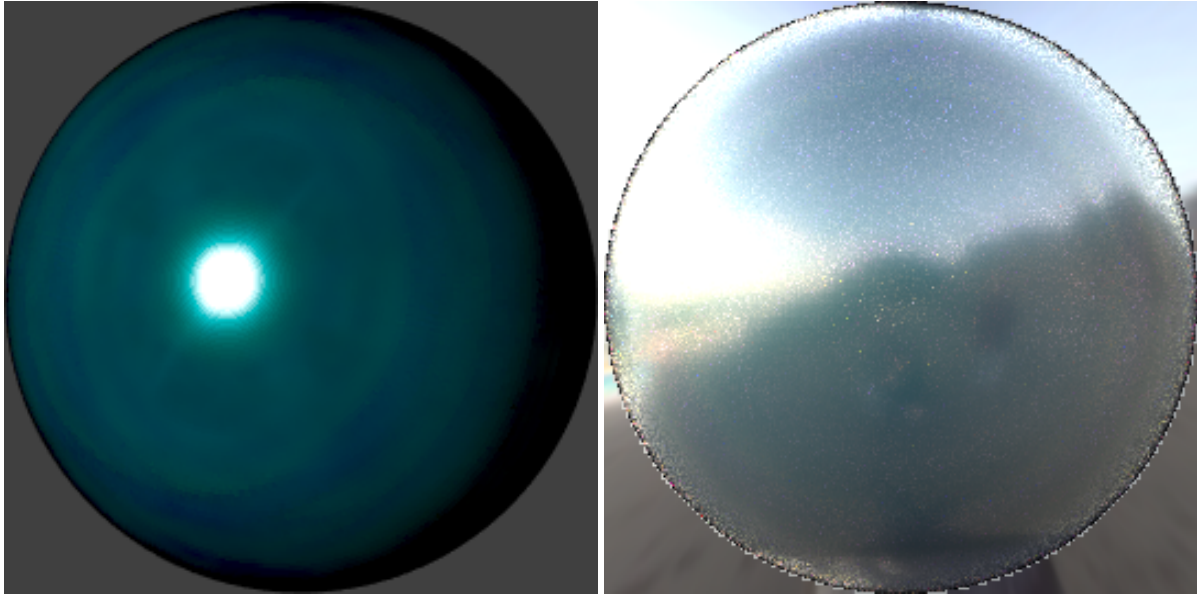


Figure II.8 – BRDF Reconstituée à partir de l'espace latent.

poser un drapeau lui indiquant l'emplacement de cette BRDF lors de la prochaine ouverture de la carte (n'y voir aucune forme de colonialisme mathématique).

Meilleure Résolution de la carte des matériaux

La carte des matériaux que nous avons générée avec *BRDF3000* a une dimension de 200x200 avec un échantillonnage de 16 en θ et 64 en ϕ . Une mise à l'échelle ne distord pas l'image donc la résolution spatiale ne pose pas tant de problème. La résolution pour l'échantillonnage de l'albédo elle, pourrait être augmentée pour améliorer l'information visuelle et capturer les subtilités des matériaux.

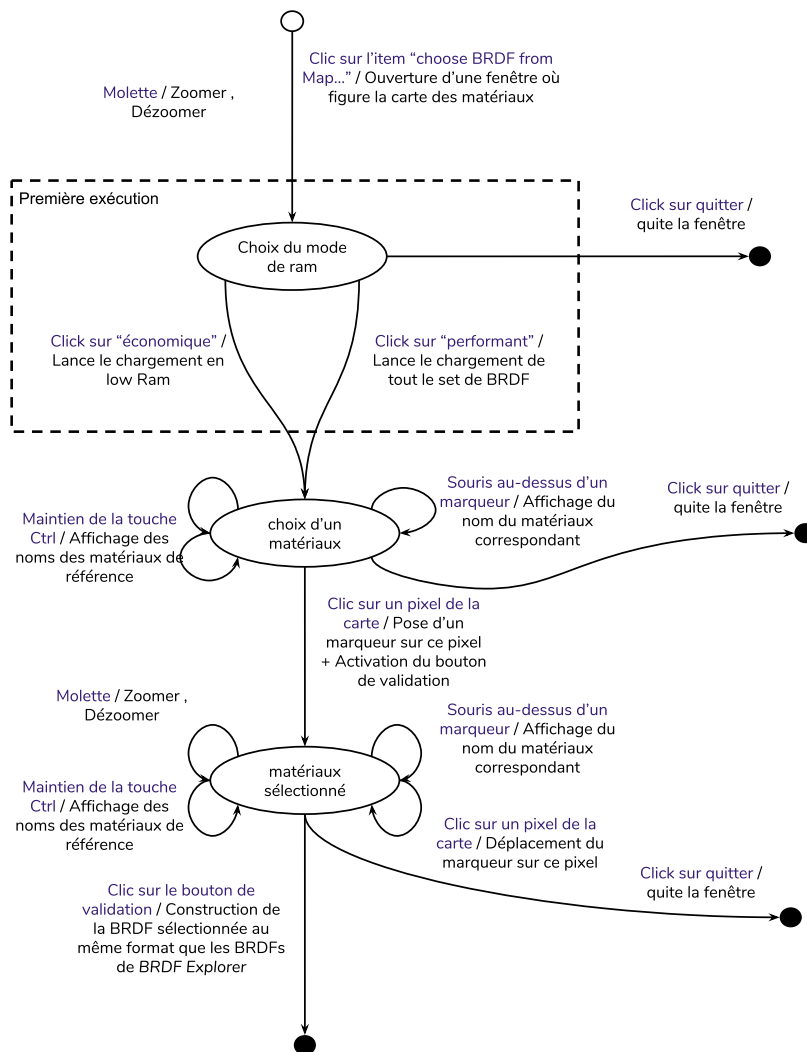


Figure II.9 – Diagramme UML d'activité de l'interface de BRDFExplorer3000

Conclusion

Les objectifs de ce projet de Chef d'Oeuvre ont pleinement été atteint.

Sur le plan technique nous avons su implémenter les méthodes prévues, avec des résultats conformes aux attentes et des temps de calculs rapide par rapport à l'ampleur des tâches numériques.

Sur le plan de la gestion de projet, nous avons réussi à respecter notre planning sans écart ni retard, tout en gardant la cohésion du groupe. Aillant commencé à coder tôt dans le processus de développement, nous n'avons pas particulièrement souffert de manque de temps. La conception s'est faite en partie de manière incrémentale, les grandes lignes ayant bien entendu été dessiné très tôt. Cela a pu être source de confusion à quelques reprises mais nous avons su régler les problèmes dès qu'ils se furent présentés.

Bibliographie

Cyril Soler, Kartic Subr, and Derek Nowrouzezahrai. A Versatile Parameterization for Measured Material Manifolds. *Computer Graphics Forum*, 37(2) :1–10, April 2018.
URL <https://hal.inria.fr/hal-01702722>.