

# Application mobile : Visualisation 3D de la cornée

Alexandre Nicaise

21 août 2015

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Département Informatique et Recherche Opérationnelle . . . .	3
1.2	Contexte Biologique . . . . .	3
1.2.1	L'œil . . . . .	3
1.2.2	La cornée . . . . .	4
1.2.3	Topographie de la cornée . . . . .	5
1.3	Problématique du stage . . . . .	7
<b>2</b>	<b>Matériels et méthodes</b>	<b>9</b>
2.1	Données disponibles . . . . .	9
2.2	VTK (Visualisation Toll Kit) . . . . .	9
2.3	Langage de programmation : C++ . . . . .	11
2.4	Programmation mobile . . . . .	12
<b>3</b>	<b>Résultat</b>	<b>13</b>
3.1	Organisation de l'application . . . . .	13
3.2	Utilisation de VTK . . . . .	13
3.3	rendu . . . . .	13
<b>4</b>	<b>Discussion</b>	<b>14</b>
4.1	Organisation de l'application . . . . .	14
4.2	Difficulté de la programmation mobile . . . . .	14
<b>5</b>	<b>Conclusion</b>	<b>15</b>

# 1 Introduction

## 1.1 Département Informatique et Recherche Opérationnelle

Fondé en 1966, le Département Informatique et Recherche Opérationnelle (DIRO) fut le premier département d'informatique créé au Québec et le troisième au Canada. Le DIRO est situé dans le pavillon André Aisenstadt qui fait partie de l'Université de Montréal (UdeM). L'UdeM a été fondée en 1978 et a été la première Université francophone de Montréal. Elle fait partie des quatre établissements supérieurs de Montréal au Québec. Selon la firme QS (Quacquarelli Symonds), l'université de Montréal se classe au 33<sup>e</sup> rang des meilleures universités du monde en recherche opérationnelle. Elle fait également belle figure en informatique, prenant place parmi le groupe de tête constitué de 150 universités d'excellence.

Durant ce stage, j'ai pu intégrer l'équipe du DIRO dirigé par Jean Meunier. Il s'intéresse à l'analyse et au traitement numérique d'images et de vidéos dans un contexte médical. Ici nous allons plutôt nous intéresser à l'œil et plus précisément à la cornée.

## 1.2 Contexte Biologique

### 1.2.1 L'œil

L'œil est l'organe de la vision, c'est à dire qu'il capte la lumière afin que le cerveau puisse l'analyser et ainsi pouvoir interagir avec son environnement. Il possède trois membranes opaques (sclérotique, choroïde et rétine) et quatre milieux transparents (cornée, humeur aqueuse, cristallin et humeur vitrée). Il est aussi protégé par plusieurs structures annexes comme les paupières, les cils et les sourcils.

L'œil est souvent comparé à un appareil photo. Il <sup>*est*</sup> constitué d'un système optique (un objectif) et d'une surface sensible à la lumière. Le système optique est constitué de la cornée et du cristallin. Il permet de capter les rayons lumineux provenant d'un objet source et de le dévier afin qu'ils convergent en un même point, idéalement situé dans le plan de la surface



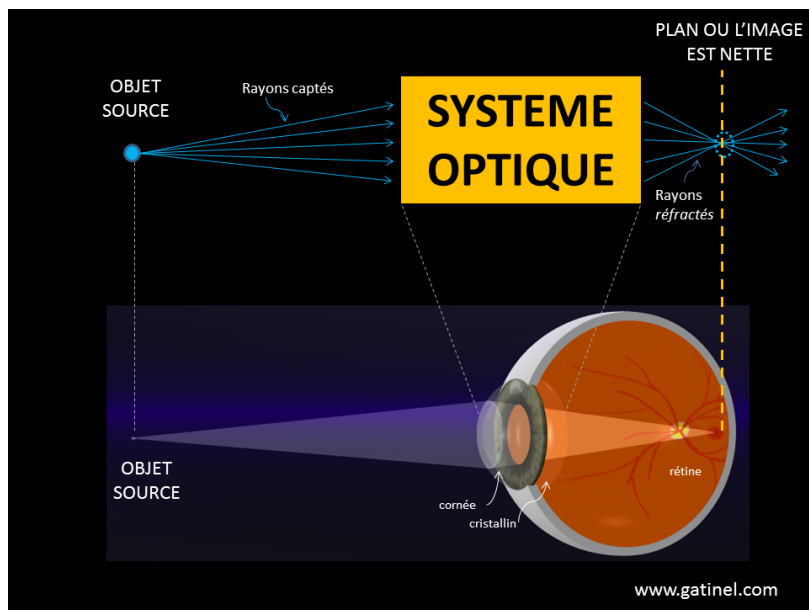


FIGURE 1 – *Présentation du mécanisme de la vision*

photosensible. Cette surface est *située* au sein de la rétine et est recouverte de cellules appelées photorécepteurs (cônes et bâtonnets). Ensuite nous quittons le domaine de l'appareil photo et passons dans le traitement de l'information. Les photorécepteurs *créent* un influx nerveux en réaction à leur exposition aux rayons lumineux. Il passe dans le nerf optique pour atteindre l'aire visuelle du cerveau. Le cerveau se charge ensuite de déchiffrer l'information contenue dans l'influx nerveux afin de permettre l'interaction avec l'environnement.

Ici nous nous intéresserons plus à un des éléments du système optique : la cornée.

### 1.2.2 La cornée

La cornée est un composant oculaire essentiel au fonctionnement de la vision : elle est la première structure que rencontre la lumière qui pénètre l'œil. Son rôle principale est de faire converger les rayons lumineux incidents qui se dirigent ensuite vers le cristallin, avant de rencontrer la rétine et d'enclencher la cascade visuelle. La cornée assure les deux tiers du pouvoir optique des structures oculaires, le tiers restant étant dévolu au cristallin. Ce pouvoir

optique découle de deux caractéristiques :

- une courbure plus importante que celle du cristallin (non accommodé)
- le contact avec l'air ambiant, offrant la plus grande différence d'indice de réfraction aux rayons lumineux incidents

La cornée est un tissu non vascularisé et transparent, de géométrie courbe. Elle est semblable à une coupole hémisphérique, et est souvent comparée à un hublot situé à l'avant de l'œil, au contact direct de l'air. Elle couvre environ un cinquième de la surface de l'œil et à un diamètre d'environ 11 mm. La cornée est légèrement plus épaisse en périphérie (0.6 mm) par rapport à son centre (0.5 mm). Pour qualifier l'épaisseur, j'utiliserais le terme de pachymétrie. Le rayon de courbure de la surface antérieure varie de 7 à 9 mm, et est d'environ 6,5 mm pour la surface postérieure. Pour permettre à l'information lumineuse d'être transmise à la rétine sans déperdition qualitative et quantitative trop préjudiciable, la cornée doit demeurer transparente, et conserver une certaine régularité de courbure. Seulement 4% à 6% de la lumière incidente est réfléchi par la surface cornéenne. Cette propriété de réflexion est mise à profit pour la réalisation de l'examen de topographie cornéenne.

### 1.2.3 Topographie de la cornée

La topographie permet de recueillir des informations relatives à la courbure ou au relief (élévation) de la cornée, grâce à la projection et l'analyse du reflet d'un motif lumineux éclairant ou balayant la cornée. Les images recueillies sont analysées de façon automatisée par un logiciel, et des cartes en couleurs sont fournies au praticien pour interprétation. Elle est courante lors d'un examen ophtalmologique, pour réaliser un diagnostic ou un suivi. En effet, des déformations peuvent être liées par exemple, à des pathologies, des traumatismes, une chirurgie, ou simplement à l'âge. Dès lors, la possibilité d'estimer et de caractériser cette déformation permet d'apporter une information pertinente au médecin pour aider à son diagnostic.

Lors du stage, j'ai utilisé des données obtenues via l'ObScan II (Figure 2),

un appareil qui permet les mesures d'élévation de la cornée. Il est capable de mesurer la partie antérieure ainsi que la partie postérieure de la cornée avec une marge d'erreur de l'ordre du micron. Les données peuvent se présenter sous la forme d'une grille 101x101 contenant les valeurs d'élévation et qui ont été prise toute les 0.1 mm. A partir de cette matrice d'élévation, il est possible de construire des maillages qui seront utilisés par la suite pour la visualisation en 3 dimensions de la cornée.

La cornée étant presque sphérique, un moyen simple et efficace de visualiser l'aspect de sa surface est d'utiliser une référence sphérique afin d'étudier ses différences par rapport à une sphère.

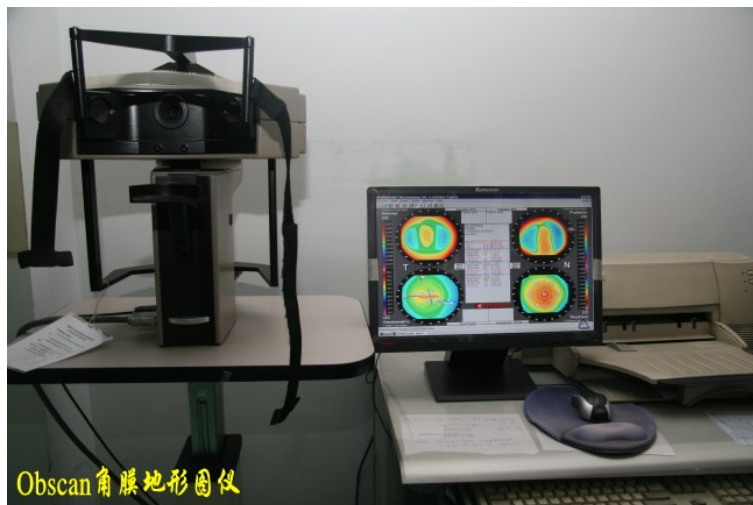


FIGURE 2 – Obscan II

Afin d'affiner leur diagnostic les praticiens ont besoin de mieux connaître l'aspect de la surface de la cornée. Pour cela, on utilise le fait qu'elle soit presque sphérique afin de la comparer avec une sphère en faisant la différences des deux. Pour cela, On calcule donc la "Best Fit Sphere" (BFS) qui est la sphère qui correspond le mieux à la matrice d'élévation, puis on effectue la différence entre les deux surfaces. Cette différence va permettre de construire une carte semblables à celles utilisées par les géologues pour représenter les relief de la terre. En effet, afin de mettre en évidence ces "reliefs" ces différences sont représentées selon un jeu de couleur standard. Les différences positives

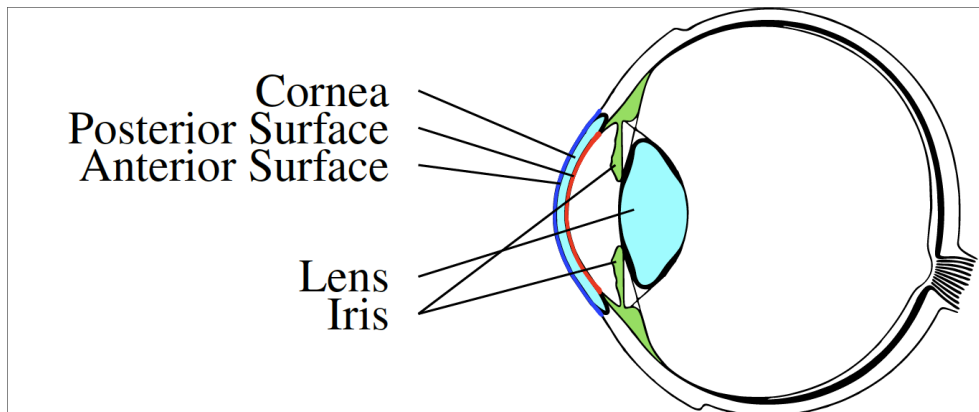


FIGURE 3 – Section d'un œil : la surface antérieure de la cornée est représentée en bleu foncé, la face postérieure en rouge et la pachymétrie en bleu clair

seront associées des couleurs chaudes (points à l'extérieur de la BFS) et les différences négatives (points à l'intérieur de la BFS) à des couleurs froides. Il est nécessaire de calculer deux BFS différentes, une pour la face antérieure et une pour la face postérieure.

La BFS est calculée en minimisant la somme des distances au carré entre la sphère et chaque point de la cornée, ce qui donne l'expression suivante à minimiser :

$$f(c, R) = \sum_{k=1}^n \sqrt{(x_k - x_c)^2 + (y_k - y_c)^2 + (z_k - z_c)^2} - R^2$$

Avec  $k$  un point de la cornée,  $c$  le centre de la BFS,  $R$  son rayon et  $n$  le nombre de point de la cornée.

L'association d'image représentée par la figure 4 permet la visualisation des différentes étapes de construction de la carte topographique de la cornée.

### 1.3 Problématique du stage

Le but de ce stage a été de construire une application mobile permettant de visualiser une cornée en trois dimensions. Ce projet a été accueilli à bras ouvert par les praticiens de l'hôpital. En effet, cela faciliterait l'accès aux différentes cartes représentant les cornées de leur patient.

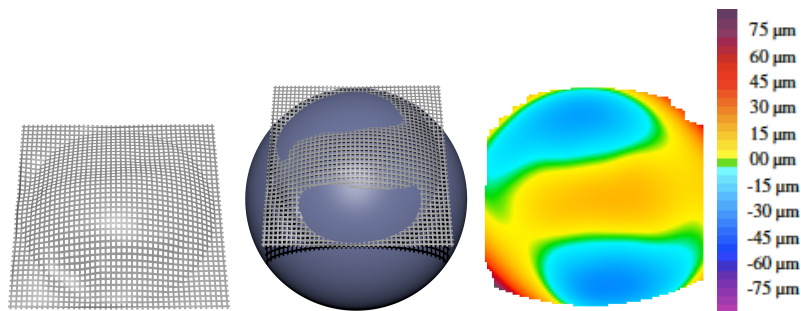


FIGURE 4 – ~~Présentation de la construction de la carte d'élévation d'une cornée.~~ *Non, c'est la construction de la carte de couleurs* A gauche nous pouvons voir la matrice 101x101 obtenue par l'obscan. L'image suivante est la superposition de la BFS (Best Fit Sphere) et de cette matrice. La troisième image est l'ajout des couleurs afin de visualiser les reliefs de la cornée selon une palette de couleurs ~~située~~ *située* à côté. On peut voir que les couleurs froides sont situées ~~en-dessous de la BFS~~ *en-dessous de la BFS* et les couleurs chaudes ~~au-dessus~~ *au-dessus*. Le vert symbolise ~~l'égalité~~ *le passage par 0* entre la BFS et la cornée.

Avec un simple cliquer il serait <sup>t</sup>toucher du doigt <sup>?</sup>il leur serait <sup>t</sup>possible d'accéder au dossier du patient et donc à l'affichage de leur cornée. Le nombre de photocopie serait <sup>t</sup>donc limiter <sup>é</sup>et ils pourraient la transporter n'importe où. Ce projet a été débuté l'année dernière par Ouri Saban un étudiant du master CCI. Mon but a été d'utiliser une bibliothèque appelée ~~Visual Tool Kit~~ *Visual Tool Kit*, qu'on nommera VTK par la suite, afin de visualiser la cornée sur des plateformes <sup>s</sup>mobile. Tout d'abord, je présenterais ~~le matériel~~ *le matériel* et les méthodes que j'ai utilisé tout au long de mon stage, puis je continuerais ~~par les résultats~~ *par les résultats* et je finirais ~~par la discussion~~ *par la discussion* et la conclusion.

*à revoir*



## 2 Matériels et méthodes

### 2.1 Données disponibles

Pour mes tests, je me suis servi<sup>e</sup> d'un document texte fournis par Arnaud Polette. Il contient toutes les informations de la cornée obtenues par le biais de l'obscan. Elles sont sous la forme de plusieurs matrices<sub>s</sub> 101x101 contenant les valeurs d'élévations et dont les valeurs de X et de Y sont situées de -5 à +5 mm séparé<sup>es</sup> par des pas de 0.1 mm.

Ces matrices permettent de visualiser :

- la face postérieur
- la face antérieur
- la différence entre la postérieur et la BFS
- la différence entre la antérieur et la BFS
- la pachymetry
- la tangentiel de la face antérieur
- la tangentiel de la face postérieur

Je ne me suis servi<sup>e</sup> que des matrices représentant la face antérieur, postérieur, la BFS, la pachymetry. En plus de ces données, ces fichiers informent sur le rayons et le centre des sphères utilisées comme BFS. Nous avons donc tout les éléments nécessaires à la visualisation en trois dimensions de la cornée.

### 2.2 VTK (Visualisation Toll Kit)

Pour effectuer la visualisation en trois dimensions, j'ai décidé d'utiliser un logiciel très utilisé par les chercheurs : VTK (Visual Tool Kit) créé par Kitware. C'est un logiciel open-source permettant l'infographie 3D, le traitement d'image et la visualisation. Il est constitué d'une librairie de classes en C++ et plusieurs couches de surfaces interprétées incluant le Tcl/Tk, Java, et Python. En effet, nous avons été séduits par une application open-source utilisant VTK appelé KIWIVIEWER fonctionnant sur android et ios. Pour

*répétition*

le portage sur android, j'ai aussi utilisé les exemples mis~~es~~ à disposition sur git-hub. Pour apprendre à utiliser VTK, un ensemble d'exemple~~s~~ est mis à disposition ainsi qu'un wiki bien structuré.

Pour la visualisation j'ai utilisé les éléments de vtk suivants :

- vtkPoints → stocke les coordonnées obtenue~~s~~ par le biais des matrices d'élévations
- vtkFloatArray → affecte un "scalar" à une coordonnée. Un scalar est une valeur qui permettra la représentation sous forme de couleurs. Ici nous utiliserons les valeurs d'élévations.
- vtkCellArray → construit le maillage. Le maillage est la représentation d'une surface (ici la cornée) en la subdivisant en un ensemble de polygones. Il est composé de sommets, connectés les uns aux autres par des faces ou facettes de forme polygonale. Lorsque toutes les faces sont des triangles, on parle de maillage triangulaire (trimesh), ou de triangulation selon les domaines. Les maillages par quadrilatères sont aussi très courants. En 3 dimensions, il est aussi possible d'utiliser des maillages volumiques, qui relient les sommets par des tétraèdres, des hexaèdres et des prismes. Ici, j'ai utilisé le maillage triangulaire.
- vtkPolyData → regroupe les objets vtkPoints, vtkFloatArray et vtkCellArray.
- vtkPolyMapper → permet la transformation de la structure de données contenue dans vtkPolyData en primitives OpenGL. Les primitives OpenGL sont des points, des lignes ou des triangles. Il permet aussi l'association de couleurs qui peut être contrôl<sup>é</sup> en spécifiant une table de couleurs (lookup table) et un intervalle.
- vtkActor → sert à positionner et orienter l'objet, à le colorer et choisir des propriété~~s~~ graphique~~s~~
- vtkRenderer → lance les algorithmes nécessaire~~s~~ au rendu de l'objet
- vtkRenderWindow → spécifie une fenêtre qui affiche le rendu
- vtkRenderWindowInteractor → permet l'interaction avec l'objet en faisant bouger la caméra.

La figure 5 permet de visualiser la construction du mapper à partir de la matrice d'élévation.

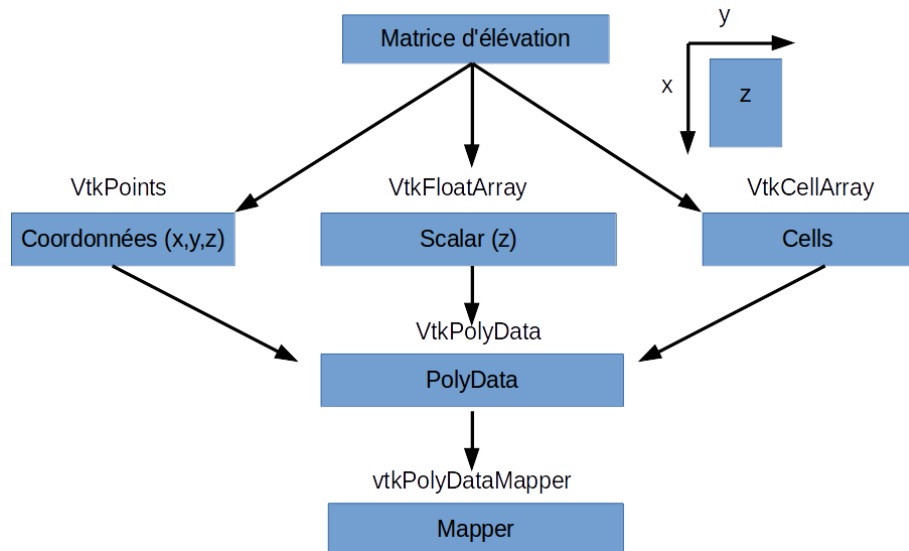


FIGURE 5 – Construction de l'objet `polydataMapper` à partir d'une matrice d'élévation. Un schéma de la matrice est représenté en haut à gauche de la figure. A partir de cette matrice 3 éléments sont extraits : les coordonnées stockés dans un `vtkPoints`, les valeurs qui seront associées à une couleur stockées dans un `vtkFloatArray` et les cellules représentant le maillage dans un `vtkCellArray`. Ces 3 éléments sont ensuite associés dans un `vtkPolyData` qui servira ensuite à construire le mapper.

Pour utiliser VTK, nous pouvons utiliser plusieurs langage tel que le java, le python, le C, le C# ou encore le C++. Étant donné que les exemples mis à disposition pour android sont codés en C++, j'ai donc utilisé ce langage afin de créer mon propre programme.

## 2.3 Langage de programmation : C++

Le C++, descendant du C, est un langage de programmation compilé apparu en 1983 et ayant subi plusieurs révisions depuis, dont la dernière était en 2014. Avec lui, nous pouvons programmer sous les paradigmes :

- procédurale → permet de décomposer le code en sous-fonctions pouvant être réutilisées plusieurs fois dans le programme voir s'appeler lui-même (récursivité) **elles (les fcts)**

à refaire

- orienté objet → permet la représentation de principes réels et de ces ?  
? d'interactions via des objets informatique et ces méthodes.
- générique → permet la définition d'algorithme identique opérant sur des données de types différents.

Pour coder sous ce langage plusieurs choix étaient à ma disposition : un environnement de développement (IDE), un éditeur de texte et CMAKE (cross platform make).

Dans un premier temps, j'ai utilisé un IDE (code : :block) qui est un ensemble d'outils permettant de faciliter le travail du développeur. Il possède donc un éditeur de texte destiné à la programmation, des fonctions permettant de démarrer le compilateur ou l'éditeur de liens par simple pression sur un bouton et un débogueur en ligne. Le débogueur en ligne permet d'exécuter le programme en construction ligne par ligne.

Dans un deuxième temps, j'ai donc utilisé un éditeur de texte (Sublime text) couplé à CMAKE. Sublime text est un éditeur de texte permettant une auto-compilation ainsi qu'une palette de couleur pour plusieurs langages de programmation tel que le C++. CMAKE est un logiciel de compilation multiplateforme. CMAKE va utiliser des fichiers appelés "CMAKELists.txt", communs pour toutes les plateformes, afin de générer des makefiles. Les makefiles sont utilisés par le programme make pour exécuter un ensemble d'actions, comme la compilation d'un projet, l'archivage de document, la mise à jour de site, etc ... Ici on l'utilise pour la compilation du projet en C++.

## 2.4 Programmation mobile

## **3    Résultat**

### **3.1    Les classes**

### **3.2    visualisation des faces antérieur et postérieur**

### **3.3    Le volume**

### **3.4    La pachymétrie**

## 4 Discussion

### 4.1 Organisation de l'application

### 4.2 Difficulté de la programmation mobile

??

Séduit par cette application, je l'ai étudié dans l'espoir de pouvoir y basé ma propre application. Malheureusement, cette application n'était plus mise à jours régulièrement. En effet, VTK a décidé d'arrêter son développement afin d'inclure la partie android et ios dans VTK en lui même. Donc une grande partie de mon stage a été de trouver comment porter VTK dans une application android. Peu de documentation peuvent être trouvé sur ce sujet.

à reformuler

### 4.3 Environnement utilisé pour coder

J'ai préféré utiliser l'association entre Sublime text et CMAKE. En effet, les exemples de VTK utilisais déjà CMAKE donc leur "CMAKELists.txt" m'ont pu servir de base à la construction de celui de mon projet. De plus, étant débutant en C++ j'ai trouvé que cette manière de faire me permettait de mieux comprendre la construction d'un projet en C++.

## 5 Conclusion