14/01/2022

DOCUMENTATION

Projet Anamorphoses

Table des matières

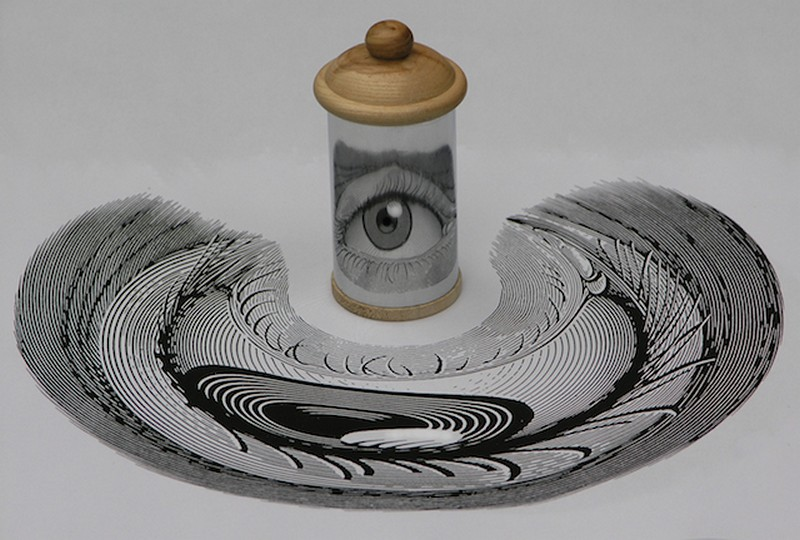
[Les calcules 2](#_Toc92752750)

[Le code 6](#_Toc92752751)

[Organisation du code 9](#_Toc92752752)

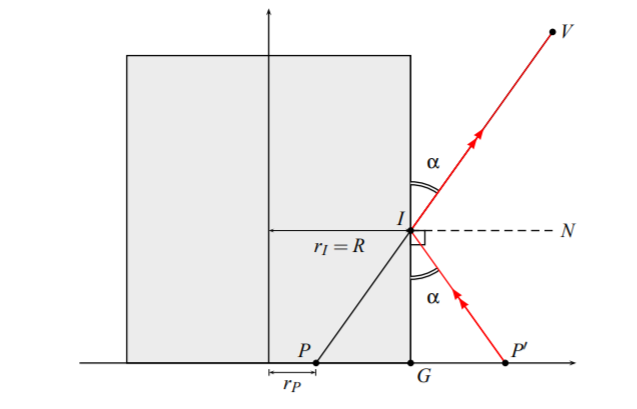
# Les calcules

Pour ce sujet nous sommes parties sur l’anamorphose cylindrique



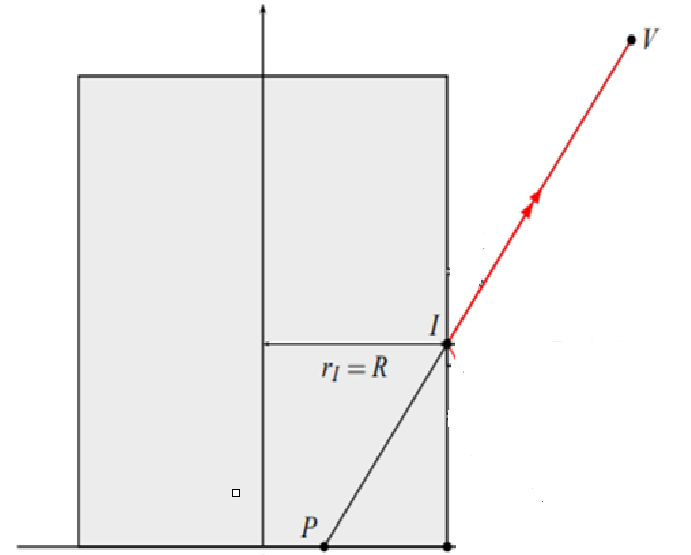
Pour pouvoir reproduire l’anamorphoses cylindrique, nous nous sommes rendu compte qu’il était extrêmement compliqué de calculer les points de l’image non déformer devant apparaitre dans le miroir cylindrique en partant de l’image déformer sur le plan.

Nous sommes donc parties d’un autre point de vue, nous allons partir du principe que l’image non déformer est déjà présente dans le miroir cylindrique puis calculer l’image déformer.



Dans cette image,

* Le point V correspond à la position de la caméra
* Le point P est un point de l’image non déformer qui apparait dans le cylindre.
* Le point P’ est un point de l’image déformer correspondant au point P.



En premier, nous cherchons à calculer le point I qui est l’intersection entre le cylindre et le vecteur PV

Comme appris en cours, IV = k\*(V-P) soit

Une image contenant texte

Description générée automatiquement

Ce qui donne aux finales

Une image contenant texte

Description générée automatiquement

Le point I appartenant au cylindre, ses coordonnées vérifie cette relation.

Avec R le rayon du cylindre.

En remplaçant Xi et Yi de la première relation par ceux de la deuxième et que l’on développe on obtient.

Une image contenant texte

Description générée automatiquement

Ce qui correspond à une équation du second degré. On peut donc calculer les solutions de ce polynôme afin de trouver « p » et donc de pouvoir résoudre l’équation Une image contenant texte

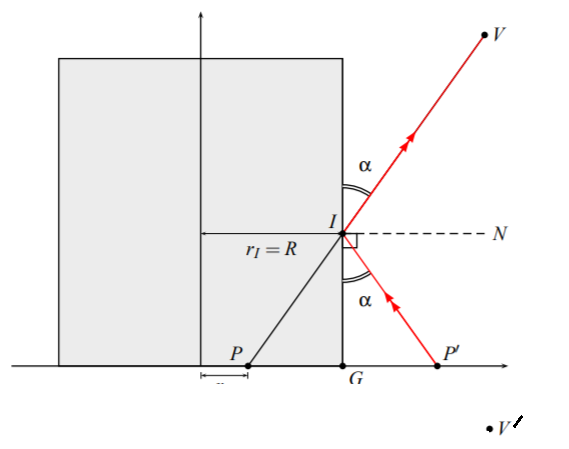
Description générée automatiquement

On obtient donc les solutions de bases d’un polynôme du second degré. On ne garde que la solution positive la plus petite puis on calcule les coordonnées de I.

Une image contenant texte, horloge, montre

Description générée automatiquement

Maintenant que nous avons I, nous pouvons commencer à essayer de trouver P’ le point de l’image déformer.



Dans la réalité, la lumière va d’abord rebondir sur l’image déformer puis toucher le miroir avant de rebondir dans l’œil du spectateur, ici le point V. Il va donc y avoir une réflexion spéculaire de la lumière au point I. Ainsi le vecteur IV possède un vecteur IV’ symétrique par rapport à la normale N en I. Si nous réussissons à calculer V’ nous pourrons utiliser les vecteur IV’ afin de calculer P’.

V’ étant la symétrique de V par rapport à la normale N il remplit deux conditions.

Une image contenant texte, objet, antenne, jauge

Description générée automatiquement

La normale IN ayant pour vecteur directeur IN(xI,yI,0), la première condition ce traduit par

Une image contenant texte

Description générée automatiquement

La deuxième par



En remplaçant xv’ et yv’ de la première expression on obtient

Une image contenant texte

Description générée automatiquement

On obtient alors les coordonnées de V’

Une image contenant texte

Description générée automatiquement

Maintenant, il ne reste plus que à trouver P’ l’intersection entre le vecteur IV’ et le plan Z=0

Comme on le sait IP’ = alpha\*(IV’) ce qui donne

Une image contenant texte

Description générée automatiquement

Or, comme on rechercher l’intersection avec le plan z=0 on obtient

Une image contenant texte

Description générée automatiquement

Ce qui donne pour P’ les coordonnées

Une image contenant texte

Description générée automatiquement

Il ne reste plus qu’a effectuer ses calculs pour l’entièreté des points de l’image non déformer afin d’obtenir l’ensemble des points de l’image déformer.

# Le code

Le code possède plusieurs fonctions principales permettant la génération et l’affichage des différentes figures.

***GénérerObjet()***

Cette fonction est différente en fonctions du fichier sélectionner, cette fonction a pour but de générer les points des de l’image non déformer qui apparaitre dans le miroir cylindrique

***GenererPlan()***

Cette fonction a pour but de générer le plan sur lequel sera poser le miroir cylindrique ainsi que l’image déformer.

La façon dont le plan est réalisé est que l’on génère 4 points différents puis on les créer une face les reliant ce qui créer le plan.

Une image contenant texte, reçu

Description générée automatiquement

***GenererCylindrique()***

Afin de créer le cylindrique, nous avons décidé de le réaliser manuellement en calculant l’ensemble des points du cercle du bas puis du haut puis de créer les différentes faces

Pour cela nous avons utilisé la formule de la courbe d’un cercle qui est :

X = R\*cos(theta)

Y = R \* sin(Theta)

Z = 0 ou 12 (en fonction du cercle du haut ou du bas)

***GenererFaceCylindreQuadra()***

Cette fonction a pour but de générer les faces quadratiques du cylindre en reliant les points des deux cercles. Pour cela, on définit les différents points de chacune des faces dans le bon ordre afin que l’ordinateur puisse afficher les faces correctement.

Une image contenant texte

Description générée automatiquement

***GenererImageDeforme()***

Cette fonction à pour but de générer les points et les faces de l’image déformer qui apparait sur le plan. La façon dont les points sont générés dans cette fonction sont lié au calculs différents calculs que nous avons réalisés. Cette fonction va donc générer autant de points que pour l’image non déformer en suivant les différents calculs réaliser.

Les faces seront ensuite créées de la même façon que pour l’image non déformer.

Par exemple, si l’image non déformer possède une face [1,2,4,3] alors l’image déformer possèdera aussi une face [1,2,4,3].

***Affichage ()***

La fonction Affichage a pour but d’afficher l’ensemble des points des différents figure qui apparaitront à l’écran tel que le cylindre, le plan et l’image déformer et non déformer.

***processSpecialKeys ()***

Cette fonction permet de déplacer la caméra en appuyant sur les fléche directionnelle.

Les flèche du haut et du bas permette d’avancer et de reculer respectivement.

Les flèche de gauche et de droit permette de se déplacer sur la droite ou sur la gauche.

Cependant comme la caméra fixe un point précis, si l’on se déplace sur la gauche ou la droite cela donne plus un effet de rotation que de déplacement mais la caméra bouge vraiment.

***Mouse() et mouseMotion()***

Ces deux fonctions travaillent ensemble afin de permettre de déplacer la caméra en appuyant sur le clique gauche et bougeant la souris.

***Clavier()***

Cette fonction permet de modifier l’affichage en appuyant sur les touches du clavier.

P => affichage des figures pleine

F => affichage en mode fil de fer

S => affichage des sommet seul

Q => Quitter le programme

***Main()***

La fonction main initialise et lance le programme. Elle va d’abord appeler toutes les fonctions permettant de calculer les points des différentes figures.

Une image contenant texte

Description générée automatiquement

Elle initialise ensuite Le programme OpenGLUne image contenant texte

Description générée automatiquement

Puis pour finir elle associe les différentes fonctions tel que Affichage() ou clavier() au programme OpenGL.

Une image contenant texte

Description générée automatiquement

# Organisation du code

Le projet est divisé en trois fichiers différents

* Figure1.cpp
* Figure2.cpp
* Figure3.cpp

La majorité du code de ces trois fichiers est similaire. La seul différence est la partie fonction genererObjet()