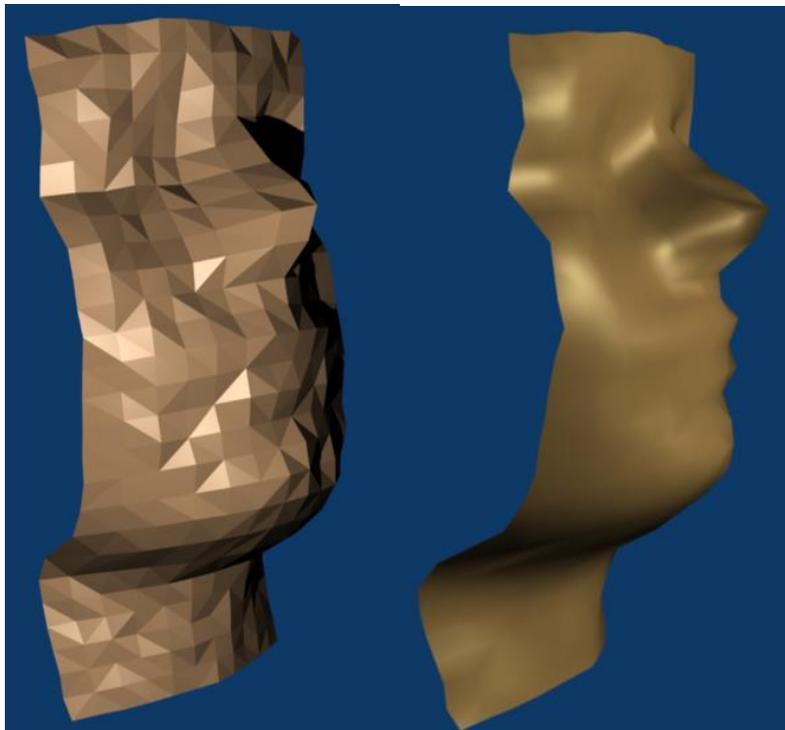


RECONSTITUTION DE LA 3D A PARTIR D'IMAGES 2D PAR PROJECTION DE STRIES



Etudiants :

Clémence PLEY

Simon BASSET

Stéphane GINIER

Amandine REYMONDIER

Sébastien NG CHUN HING

Enseignant-responsable du projet :

Jérôme YON

Date de remise du rapport : 22/06/09

Référence du projet : STPI/P6-3/2009 – 004

Intitulé du projet : **Reconstitution de la 3D à partir d'images 2D par projection de stries.**

Type de projet : **expérimental et informatique.**

Objectifs du projet:

Notre projet a pour but d'évaluer la faisabilité de la reconstruction de reliefs 3D à partir de différentes photographies.

- **Recherche des méthodes de reconstruction 3D existantes.**
- **Etude des lois de projection (de 2D vers 3D).**
- **Mise en œuvre expérimentale de la méthode de projection d'un quadrillage appliquée à des cas simples.**
- **Interprétation des résultats.**
- **Analyse et reproduction des volumes à l'aide d'un logiciel informatique.**

.

TABLE DES MATIERES

1.	Introduction.....	6
2.	Méthodologie / Organisation du travail.....	6
3.	Travail réalisé et résultats	7
3.1.	Les techniques existantes : les méthodes de scanner 3D.....	7
3.1.1.	Principe.....	7
3.1.2.	Techniques de collecte de données.....	8
3.1.2.1.	Scanner avec contact.....	8
3.1.2.2.	Scanner sans contact actif	8
3.1.2.3.	Scanner sans contact passif.....	9
3.2.	Notre méthode d'analyse	9
3.2.1.	Formule mathématique	9
3.2.2.	Dispositif expérimental.....	11
3.2.3.	Résultats.....	12
3.2.3.1.	Traitement des photos et extraction de l'information 3D	12
3.2.3.1.1.	Traitement des photos	12
3.2.3.1.2.	Extraction de l'information 3D	15
3.2.3.2.	Affichage de l'objet en 3D.....	17
3.2.3.2.1.	Lecture et affichage d'une liste de points	17
3.2.3.2.2.	Création des facettes (maillage)	17
3.3.	Comparaison des différentes méthodes.....	19
4.	Conclusions et perspectives	20
5.	Bibliographie.....	23
6.	Annexes	24
6.1.	Listings des programmes réalisés.....	24
6.2.	Schémas d'études	25

NOTATIONS, ACRONYMES

2D : deux dimensions.

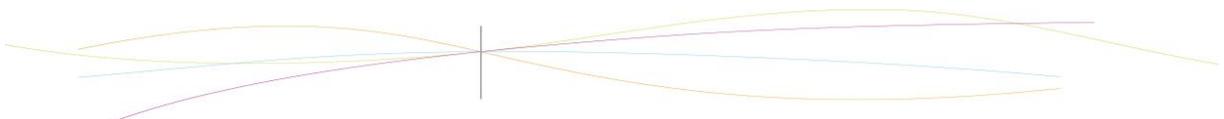
3D : trois dimensions.

α : angle entre le projecteur et la normale.

β : angle entre la normale et la caméra.

$\gamma = \alpha + \beta$: angle entre le projecteur et la caméra.

MMT : machine à mesurer tridimensionnelle.



1. INTRODUCTION

Notre projet de physique porte sur la reconstitution de la 3D à partir d'images 2D par projection de stries.

La troisième dimension est actuellement très utilisée dans différents domaines tels que la médecine, le cinéma, l'architecture... [1]. Par exemple, on l'utilise pour les échographies pour permettre aux parents de voir des images de leur bébé faciles à interpréter. Cependant les méthodes utilisées n'ont jamais été réellement explicitées et s'appuient souvent sur l'utilisation de logiciels très complexes permettant d'effectuer des calculs lourds.



figure 1 : Echographie et design 3D.

Nous avons voulu étudier la faisabilité d'une méthode de reconstruction 3D, jamais explorée à notre connaissance. Notre méthode consiste à extraire des coordonnées de points à partir de photos afin de reproduire des volumes. Tandis que des méthodes comme la photogrammétrie [2] et la stéréoscopie [3] ne produisent qu'un effet visuel en 3D et non pas un réel traitement des images. Nous nous sommes intéressés à d'autres méthodes se rapprochant de notre but: le scanner laser 3D.

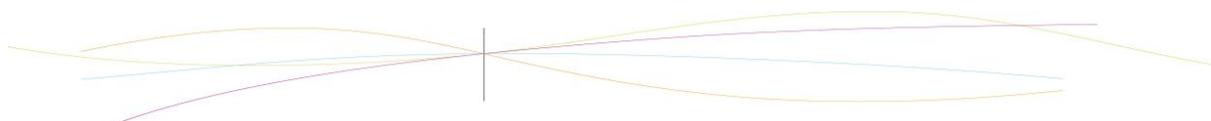
Nous allons dans un premier temps, à travers ce dossier, vous présenter notre organisation pour mener à bien ce projet. Puis, les différentes méthodes de reconstruction vous seront exposées. Nous nous attacherons à expliquer notre méthode utilisée : avec celle-ci nous avons essayé de reproduire un visage. Enfin nous la comparerons avec les autres techniques.

2. METHODOLOGIE / ORGANISATION DU TRAVAIL

Ce sujet nous a été attribué après un sondage de préférence. Le groupe a donc été formé ne tenant pas compte des affinités. Nous avons du apprendre, tout d'abord, à nous connaître puis, à travailler ensemble.

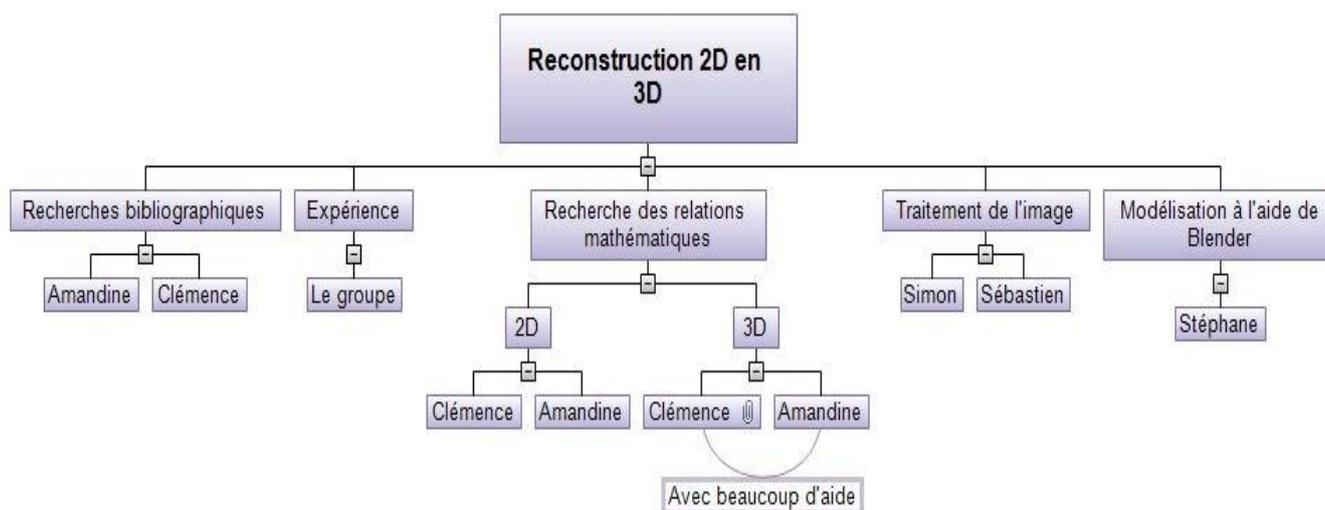
Concernant l'organisation de notre travail, nous avons mis un certain temps avant de bien comprendre le sujet qui nous était donné. En effet, comme dit précédemment, ce dernier n'avait jamais été traité les années précédentes et donc aucune pistes ne nous ont été proposées.

Nous allons donc vous expliquer la démarche que nous avons suivie. Au début, nous avons cherché des idées de réalisation grâce aux méthodes déjà existantes. Ensuite, nous nous sommes rendu compte que sans relations mathématiques, notre projet ne pourrait pas avancer. Ainsi, pendant qu'une partie du groupe cherchait la théorie mathématique, l'autre partie apprivoisait les différents logiciels de modélisation. Durant cette partie du projet, notre



motivation n'était plus à son comble, du fait de la complexité des calculs tridimensionnels et des recherches bibliographiques infructueuses. Pour palier à cette baisse d'entrain, nous avons décidé de commencer les expériences afin d'avoir une vision plus concrète. Suite à cette approche pratique, un nouveau souffle nous a été donné, de nombreuses questions ont enfin trouvé leurs réponses. Une deuxième session pratique a été nécessaire en adoptant les conditions requises pour les formules et plus de précision. Enfin, toute la partie extraction d'informations des images et modélisation a pris tout son sens.

Voici l'organigramme résumant notre répartition des tâches :



3. TRAVAIL REALISE ET RESULTATS

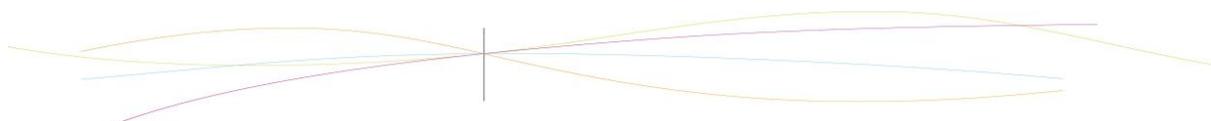
3.1. Les techniques existantes : les méthodes de scanner 3D.

Un scanner 3D est un appareil de numérisation et d'acquisition 3D qui analyse les objets ou leur environnement proche pour recueillir des informations précises sur la forme et éventuellement sur l'apparence (couleur, texture, ...) de ceux-ci. Les données ainsi collectées peuvent être utilisées pour construire des images de synthèse en trois dimensions (objets numériques) à des fins diverses : cinéma, jeux vidéos, industries etc... [4], [5], [6], [7].

3.1.1. Principe

Un scanner 3D mesure généralement le positionnement d'un échantillonnage de points dans un système de coordonnées - un nuage de points - de la surface d'un sujet pour ensuite en extrapoler la forme à partir de leur répartition. Si la couleur de chacun des points est analysée, alors celle de la surface peut également être reconstituée.

L'image produite est basée sur une série de données composées des coordonnées positionnant chacun des points échantillonnés par rapport au scanner 3D. Si un système de coordonnées sphériques est utilisé et que le scanner en est l'origine, chaque point peut alors être identifié par des coordonnées (r, φ, θ) . Ces coordonnées sphériques permettent de situer dans l'espace chacun des points par rapport au scanner, travail préalable et nécessaire à la modélisation numérique de l'image en trois dimensions de l'objet.



Généralement, les données (coordonnées des points) recueillies avec une seule passe ne sont pas suffisantes pour modéliser entièrement un sujet. Le travail doit être effectué de nombreuses fois, voir des centaines de fois, à partir de points de vue différents. Toutes les données recueillies doivent être réinterprétées et situées dans un système de coordonnées unique et regroupées.

3.1.2. *Techniques de collecte de données*

Les scanners 3D se divisent en deux familles : celles nécessitant un contact avec le sujet et les autres. Les sans-contacts peuvent également être subdivisés en deux principales catégories, les scanners actifs et passifs.

3.1.2.1. *Scanner avec contact*

Les scanners avec contact sondent le sujet grâce à un contact physique et ont une bonne précision. Cependant, leur usage ne peut-être généralisé à tous les types de numérisation.

En effet, leur principe même, basé sur un contact physique, peut détruire ou altérer des objets fragiles. Ils font courir un risque pour une utilisation sur des objets uniques ou de valeur comme, par exemple, des objets historiques. Un autre inconvénient de cette technologie est sa relative lenteur par rapports aux autres méthodes. Le lent déplacement du bras sur lequel le palpeur est monté fait que les mesures sont réalisées à une faible fréquence, à quelques 0,1 kilohertz. Par comparaison, un système utilisant un scanner optique effectue ses mesures entre 10 et 500 kilohertz. Ce type de scanner est utilisé dans l'industrie pour leur précision, dans l'industrie mécanique, par exemple, les machines à mesurer tridimensionnelles. Un autre exemple, dans l'industrie d'animation cinématographique, des modèles sculptés dans de la terre glaise sont ensuite numérisés en trois dimensions.

3.1.2.2. *Scanner sans contact actif*

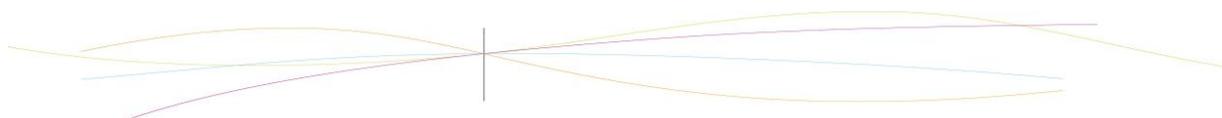
Les scanners actifs émettent un rayonnement et détectent sa réflexion afin de sonder un objet ou un environnement. Différents types de source de rayonnement sont utilisés : lumière, ultrason ou rayon X.

Scanner par temps de vol :

Ce scanner Temps de vol-LIDAR peut être utilisé pour scanner des bâtiments, des formations géologiques, etc. afin de produire une modélisation en trois dimensions. Son rayon est orientable sur un très large horizon : grâce à la rotation horizontale de sa tête, un miroir le dirige verticalement. Le rayon laser sert à mesurer la distance avec le premier objet coupant le faisceau.

Au cœur de ce type de scanner se trouve un télémètre laser permettant de calculer la distance avec la surface de l'objet étudié en comptant le temps nécessaire au trajet aller-retour de l'impulsion du faisceau laser réfléchi.

Puisque la vitesse de la lumière c est connue, le temps de retour permet de déterminer la distance parcourue par la lumière, qui est deux fois la distance entre le scanner et la surface.



Le télémètre laser détecte seulement un point à la fois dans la direction sur laquelle il est pointé. Pour cela, l'appareil scan la totalité de son champ de vision point par point et doit changer sa direction de vue à chaque mesure. Elle peut être changée par la rotation de l'appareil lui-même ou par l'utilisation d'un système de miroirs rotatifs. Cette dernière méthode est la plus couramment utilisée parce que les miroirs sont plus légers et peuvent changer de direction plus rapidement avec une plus grande précision. Les scanners 3D par temps de vol peuvent mesurer la distance de 10 000 à 100 000 points par seconde.

Scanner par triangulation :

Le scanner laser par triangulation est un scanner actif qui utilise également la lumière laser pour sonder son environnement. Il pointe sur le sujet avec un faisceau comme pour celui par temps de vol et utilise un appareil photo pour situer le point. En fonction de la distance jusqu'à une surface, le point apparaît à un endroit différent dans le champ de vision de l'appareil.

Cette technique est appelée triangulation parce que le point laser, l'appareil photo et l'émetteur laser forment un triangle. La longueur d'un côté du triangle, la distance entre l'appareil photo et l'émetteur laser est connue. L'angle du côté de l'émetteur laser est également connu. L'angle du côté de l'appareil photo peut-être déterminé en regardant l'emplacement du point laser dans le champ de vision de l'appareil photo. Ces trois données déterminent la forme et les dimensions du triangle et donnent la position du point laser. Dans la plupart des cas, une bande laser, plutôt qu'un point, balaie l'objet pour accélérer le processus d'acquisition.

3.1.2.3. Scanner sans contact passif

Les scanners sans contact passifs, n'étant émetteurs d'aucun type de rayonnement, se basent sur la détection de rayonnement ambiant réfléchi. La plupart des scanners de ce type détectent la lumière visible car elle est immédiatement disponible. D'autres types de rayonnement, comme les infrarouges peuvent également être utilisés. Les méthodes passives peuvent être bon marché, du fait que dans la majorité des cas elles ne nécessitent pas d'appareil d'émission spécifique.

Scanners stéréoscopiques :

Les systèmes stéréoscopiques utilisent généralement deux caméras vidéo, légèrement espacées, pointant vers la même scène. En analysant les légères différences entre les images des deux appareils, il est possible de déterminer la distance de chaque point de l'image. Cette méthode est basée sur la vision stéréoscopique humaine.

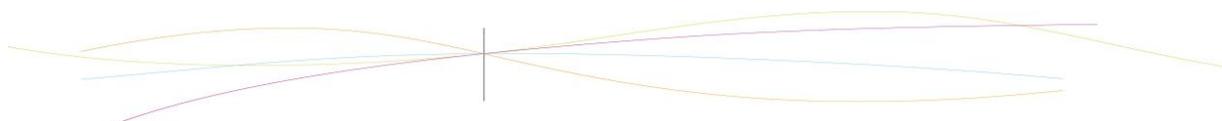
Scanners à silhouette :

Ces types de scanners 3D utilisent les contours créés à partir d'une séquence de photos prises autour d'un objet en trois dimensions devant un arrière plan contrasté. Ces silhouettes sont détachées de leur fond et assemblées les unes aux autres à l'endroit de l'axe de rotation de la caméra pour former une "coque visuelle" une approximation de l'objet. Avec ce type de techniques toutes sortes de concavité de l'objet - comme l'intérieur d'un bol - ne sont pas détectées.

3.2. Notre méthode d'analyse

3.2.1. Formule mathématique

Au début de notre recherche, ne savant pas dans quelle direction cherchée, nous avons, tout d'abord, commencé notre étude en deux dimensions. Nous nous sommes mis dans la



situation de projection de stries horizontale d'épaisseur e sur un écran incliné. Le schéma d'étude et les formules se trouvent en annexe.

Après avoir réalisé que nos formules en 2D ne correspondaient pas à notre projet, nous avons abordé le problème sous trois dimensions. Nous avons décidé de projeter un repère et un point. Voici notre nouveau schéma d'approche :

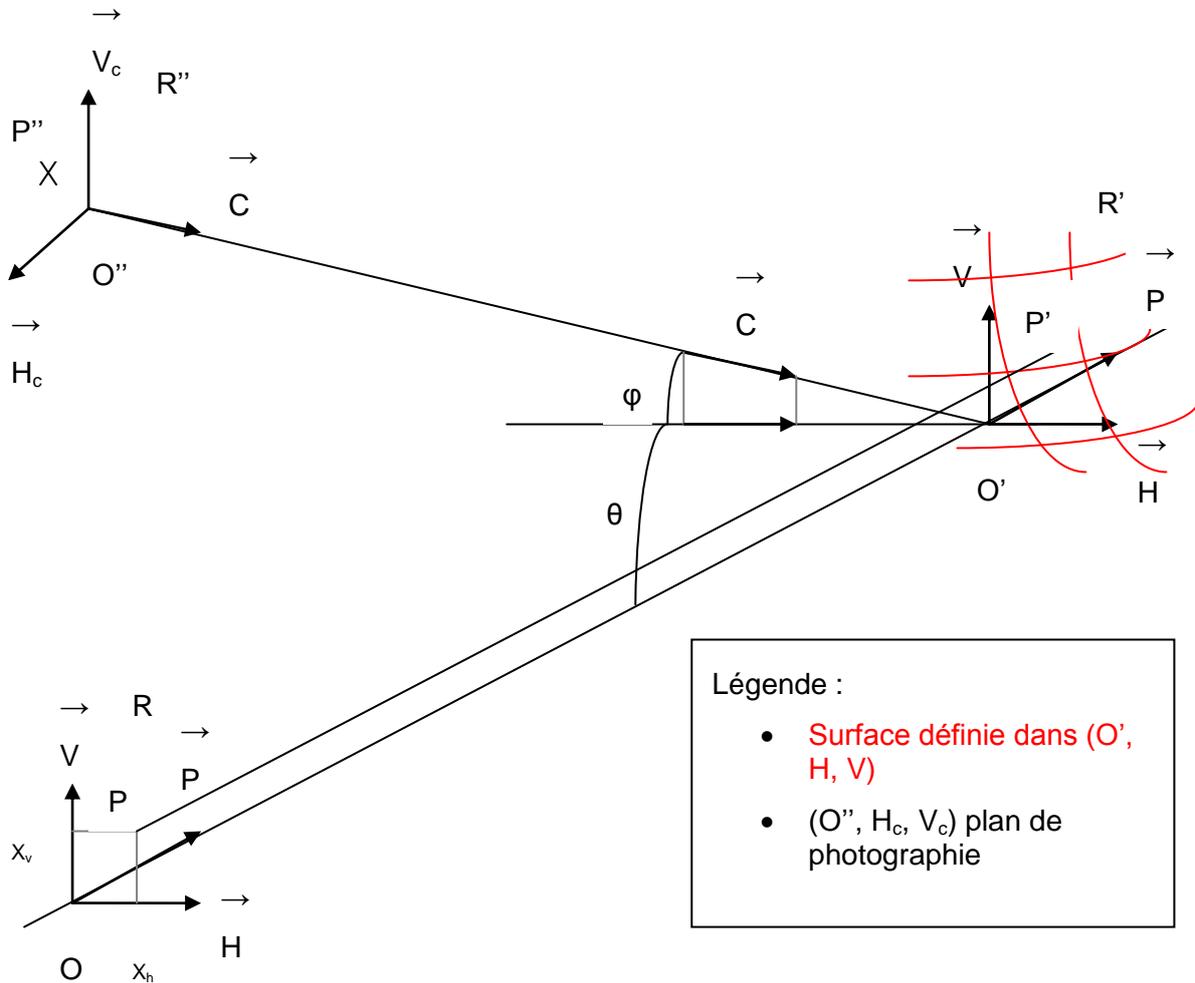


figure 2 : Schéma d'étude pour les formules 3D.

Le projeté du point P dans R' devient P' et à pour coordonnées dans R' :

eq 1. $O'P' = x_h * H + x_v * V + f(x_h, x_v) * P$

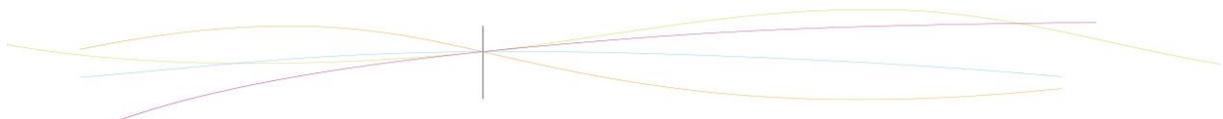
La caméra est positionnée par rapport a R' par les angles θ et φ (cf. figure 2). Le référentiel lié à la caméra est R'' (O'', C', Vc, Hc).

On a donc:

$$C = \begin{pmatrix} \cos(\varphi) * \cos(\theta) \\ \sin(\varphi) \\ \cos(\varphi) * \sin(\theta) \end{pmatrix} \quad H_c = \begin{pmatrix} -\sin(\theta) \\ 0 \\ \cos(\theta) \end{pmatrix}$$

(P, V, H)

Et comme (C, Vc, Hc) est un trièdre direct:



$$V_c = \begin{pmatrix} -\sin(\theta) \\ 0 \\ \cos(\theta) \end{pmatrix} \wedge \begin{pmatrix} \cos(\varphi) * \cos(\theta) \\ -\sin(\varphi) \\ \cos(\varphi) * \sin(\theta) \end{pmatrix} = \begin{pmatrix} \sin(\varphi) * \cos(\theta) \\ \cos(\varphi) \\ \sin(\theta) * \sin(\varphi) \end{pmatrix}$$

La projection suivant C de P' dans le plan de la caméra correspond à P'' tel que :

eq 2. $O''P'' = O''O' + O'P' + P'P''$

De plus, O''O' et P'P'' sont proportionnels à C et C est perpendiculaire à H_c. On obtient alors : $x_{hc} = O''P'' \cdot H_c = O'P' \cdot H_c$ et $x_{vc} = O'P' \cdot V_c$. Finalement, nous obtenons ces deux équations.

eq 3. $x_{hc} = -\sin(\theta) * f(x_h, x_v) + x_h * \cos(\theta)$

eq 4. $x_{vc} = \sin(\varphi) * \cos(\theta) * f(x_h, x_v) + \cos(\varphi) * x_v + \sin(\theta) * \sin(\varphi) * x_h$

Après avoir trouvé ces formules générales, il a fallu les adapter à nos mesures.

- On fixe les deux angles φ et θ car on les mesure.
- On fait une photographie de référence I₀ (image 0) en plaçant en O' un plan parallèle à (H, V).
- On nomme, dans l'image, x (précédemment x_{hc}) et y (précédemment y_{hc}), les coordonnées du point.

Pour la photo de référence I₀, f(x_h, x_v) = 0 car il s'agit de plans parallèles. Nous obtenons alors deux nouvelles équations.

eq 5. $x_h = x_0 / \cos(\theta)$

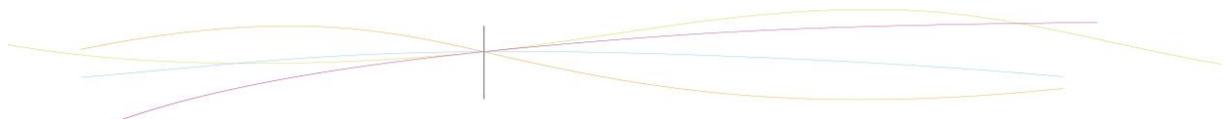
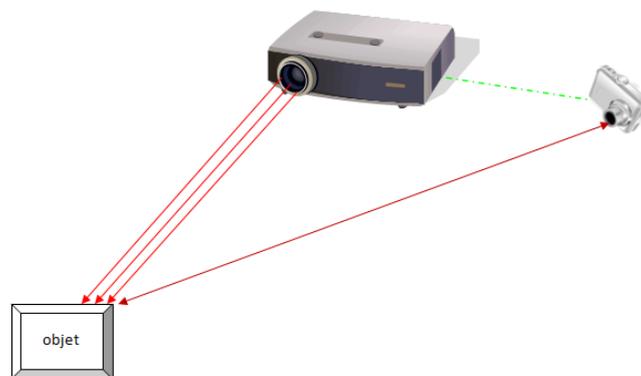
eq 6. $x_v = y_0 / \cos(\varphi) - \tan(\theta) * \tan(\varphi) * x_0$

Ensuite, on combine les quatre dernières équations et on obtient l'équation finale.

eq 7. $f(x_h, x_v) = (x_0 - x) / \sin(\theta) = (y - y_0) / (\sin(\varphi) * \cos(\theta))$

Remarque: on note que pour φ = θ = 0, aucune mesure n'est possible.

3.2.2. Dispositif expérimental



Matériel :

- Projecteur
- Appareil photo sur trépied
- Objet expérimental
- Mètre

Protocole :

1. Projeter la grille sur une surface plane (le mur) et faire la mise au point du projecteur afin d'obtenir une image nette sur le mur.
2. Placer l'appareil photo en décalage par rapport au projecteur afin d'obtenir un certain angle entre l'axe de l'appareil et l'axe de projection.
3. Prendre la photo référence.
4. Placer l'objet sur un support et faire la mise au point du projecteur.
5. Prendre la photo de l'objet.
6. Répéter l'opération en décalant l'appareil photo avec des angles différents. Prendre à chaque fois une photo de référence sans objet et une photo de l'objet.

3.2.3. Résultats

3.2.3.1. Traitement des photos et extraction de l'information 3D

Dans cette partie, nous nous intéressons à la procédure pour extraire l'information des photos obtenues au stade expérimentale. L'objectif est de transformer l'information indistincte, contenue dans les photos, en une information brute, contenue dans un fichier texte prêt à être utilisé pour un affichage 3D.

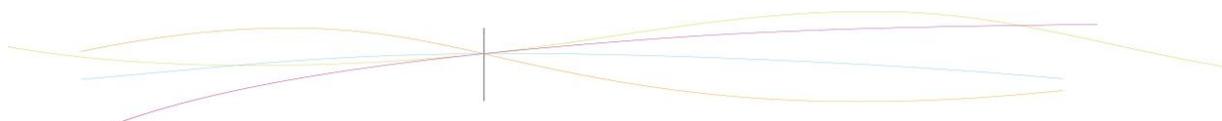


figure 3 : Schéma représentant la fonction, les objets qu'elle reçoit et celui qu'elle renvoie.

Pour réaliser le traitement des photos et l'extraction de l'information 3D, nous allons demander au stade expérimental de nous fournir deux photos, une avec l'objet que l'on souhaite reconstruire, et une autre sans, le tout photographié dans les mêmes conditions de prise de vue. Nous demanderons également les mesures des angles de prise de vue, entre le projecteur et l'appareil photo, horizontalement et verticalement. Enfin, notre fonction fournira un fichier texte qui respectera un cahier des charges imposé par la fonction d'affichage 3D.

3.2.3.1.1. Traitement des photos

Si on s'intéresse à la qualité et la netteté des photos fournies par le stade expérimental, on remarque rapidement qu'un traitement doit être appliqué à la photo avant de pouvoir l'utiliser.



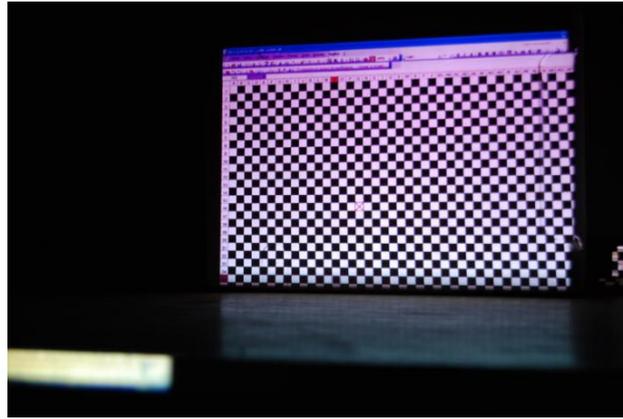


figure 4 : Photo de qualité acceptable.

A première vue, cette photo ne comporte pas ou peu de défaut dans la zone qui nous intéresse (autour de la cible rouge). En effet, elle est d'assez bonne qualité relativement aux autres photos.

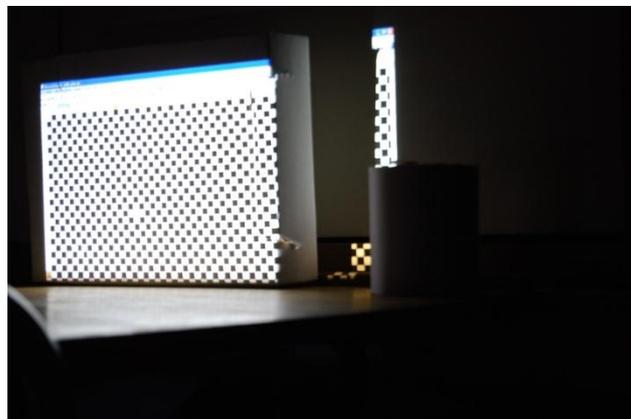


figure 5 : Photo de mauvaise qualité.

Au contraire, cette nouvelle photo semble avoir un problème de luminosité, on n'arrive même pas à distinguer la croix rouge repère, de plus les carrés noirs sont clairement disjoints. On va donc rejeter cette photo dans un souci de précision et de qualité de l'extraction des informations, d'autant plus que sa photo appariée (celle avec l'objet) ne souffre pas du problème.

L'extraction de l'information consiste en l'analyse de la déformation d'une projection (ici, un damier) lorsque l'on place l'objet devant le fond sur lequel on projette. On compare donc la position d'un point du damier sur la photo sans objet avec sa nouvelle position sur la photo avec objet. Il faut donc être en mesure de pouvoir repérer un point en particulier sur ces photos. Dans la suite, nous allons utiliser la photo exemple considérée de bonne qualité et vue plus haut.

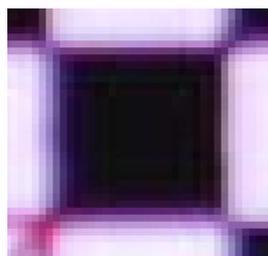
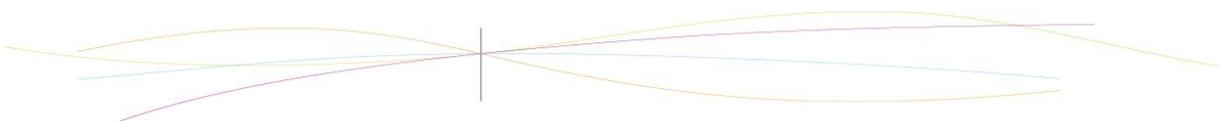


figure 6 : Zoom sur une case du damier.



En zoomant sur une zone près de la croix rouge, on s'aperçoit que les cases du damier ne sont pas si nettes que ce que l'on pensait. Si on avait par exemple prévu de choisir le coin inférieur droit d'une case comme point particulier, savoir lequel de ces pixels il faut choisir est un vrai problème. Il faut donc traiter cette image pour faciliter la partie extraction de l'information. On veut augmenter la netteté et le contraste des photos, une grande quantité de logiciel s'offre alors à nous pour retoucher les photos, mais dans notre recherche de gratuité et de liberté d'utilisation, The GIMP (GNU Image Manipulation Program) s'est imposé naturellement. En premier, on rogne la photo pour ne garder que la zone qui nous intéresse (inutile de garder les zones non atteintes par la projection, etc.). On utilise l'outil découpage  pour sélectionner la zone à conserver, il faut penser à découper à partir d'un point particulier précis, au pixel près, pour reproduire le même découpage sur la photo appariée, ce qui corrigera le décalage de quelques pixels qu'il y a entre les deux photos.

Ensuite on va utiliser des scripts pour améliorer les images, ils sont situés dans le menu déroulant Filtres puis Amélioration, on va utiliser Augmenter les contrastes... et renforcer la netteté... même sans rechercher les paramètres optimaux on obtient de très bon résultats.

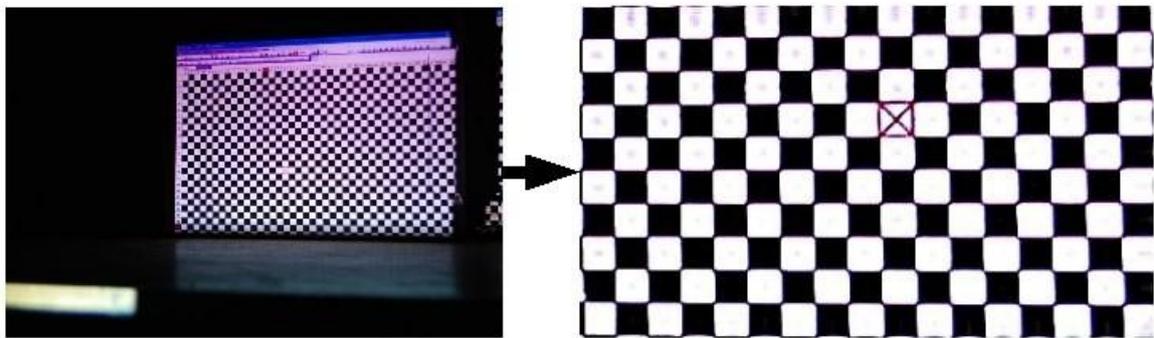


figure 7 : Résultat du traitement de la photo.

Et lorsque nous zoomons sur une case du damier, on note une bien meilleure netteté des cases.

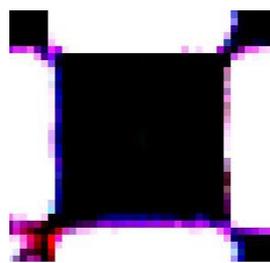
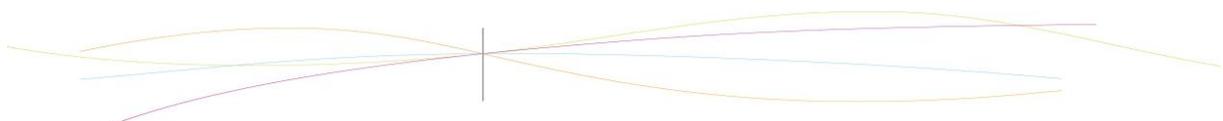


figure 8 : Zoom sur une case du damier après retouche.

Malheureusement, l'image n'est pas encore assez nette pour espérer la passer dans un algorithme très simple de recherche de points, par exemple : la recherche du pixel noir dont le pixel directement en dessous et celui à sa gauche soient blancs, de façon à obtenir, le point inférieur droit de la case. Après avoir utilisé un script Scilab (annexe) sur la photo, qui la convertit en niveau de gris puis parcourt tout les pixels de l'image et compare leur valeur de niveau de gris à un seuil pour affecter soit la couleur blanche soit la couleur noire, on voit que l'algorithme n'est pas applicable.



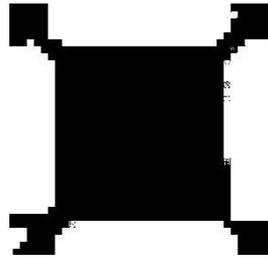


figure 9 : Après passage dans le script Scilab

Sur ce zoom, seulement 2 points correspondent à des coins inférieurs droits de cases noires. Or l'algorithme va trouver 6 points, selon ses critères de recherche, mais aucun d'entre eux ne correspond aux 2 points que nous souhaitons obtenir.

3.2.3.1.2. Extraction de l'information 3D

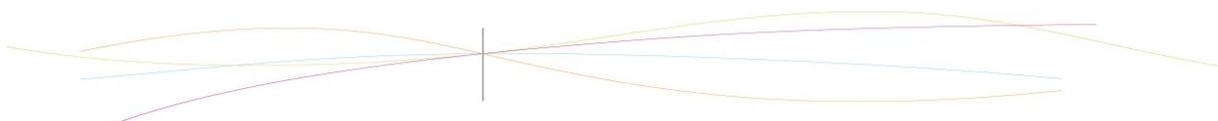
En supposant que nous avons traité deux photos appariées et que nous disposons des mesures expérimentales des angles de leur prise de vue, nous allons pouvoir commencer à en extraire l'information 3D. Pour cela, on va se servir de The GIMP et de Calc (tableur de la suite bureautique OpenOffice.org). Nous avons préparé, au préalable, un classeur avec des feuilles de calcul remplies avec les formules 3D.

	A	B	C	D	E	F	G	H	I	J	K
1		angle Θ	21,78								
2		angle Φ	7,75								
3											
4											
5	x0	y0	x	y			X	Y	Z1	Z2	Z
6	45	292	45	292			48.46	292.24	0	0	0
7	94	292	94	292			101.23	289.58	0	0	0
8	193	292	108	321			207.84	284.2	229.08	231.58	230.33
9	243	292	141	326			261.68	281.48	274.9	271.51	273.21
10	293	292	183	329			315.52	278.76	296.46	295.47	295.97
11	344	292	229	329			370.44	275.99	309.94	295.47	302.7
12	394	290	280	329			424.29	271.25	307.24	311.44	309.34
13	446	291	337	326			480.28	269.43	293.77	279.5	286.63
14	498	290	407	320			536.28	265.59	245.25	239.57	242.41
15	550	290	550	290			592.28	262.77	0	0	0
16	603	290	603	290			649.35	259.88	0	0	0
17	656	290	656	290			706.43	257	0	0	0

figure 10 : Exemple d'utilisation des feuilles de calcul préparées.

Il suffit de remplir les cases avec les bonnes informations, on commence par renseigner les deux angles. Ensuite il faut cibler un point sur la photo sans l'objet, grâce à The GIMP, pour en avoir les coordonnées en abscisse et en ordonnée puis on remplit les colonnes x0 et y0. Enfin on cible le même point sur la photo avec l'objet pour obtenir ses coordonnées et les mettre dans les colonnes x et y. Les colonnes X et Y représentent les coordonnées du point dans le plan normal à l'axe de projection. La colonne Z1, le relief calculé par la formule utilisant l'angle Θ . La colonne Z2, le relief calculé par la formule utilisant l'angle Θ et Φ . Enfin Z est la moyenne de Z1 et de Z2.

Une deuxième difficulté rencontrée, autre que la netteté de l'image, est le lien entre les 2 positions différentes d'un même point sur les photos. En effet, l'affichage 3D impose au fichier texte de suivre une certaine logique pour ranger les points dans le fichier. On va donc suivre des lignes de points.



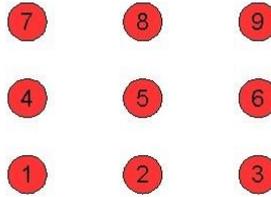


figure 11 : *Ordre dans lequel on va traiter les points.*

Or si l'on regarde la photo suivante, on a tendance à suivre la ligne rouge comme ligne de points :

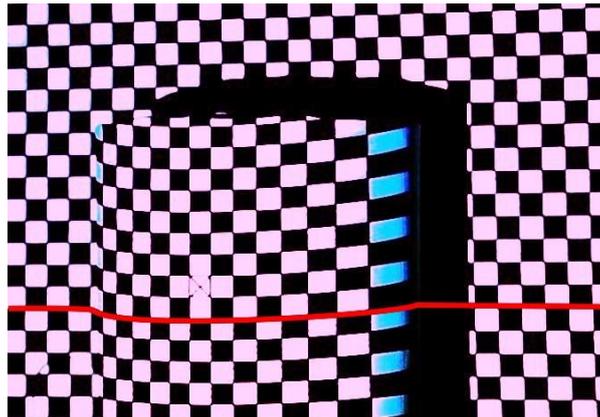


figure 12 : *Mauvaise ligne de points*

Alors qu'en réalité, la ligne de points qu'il faut prendre a cette forme :

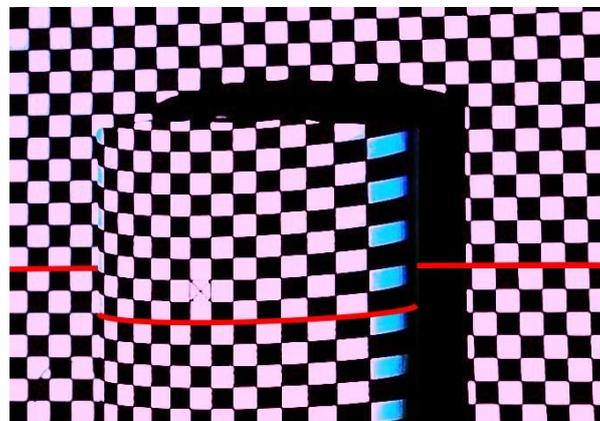
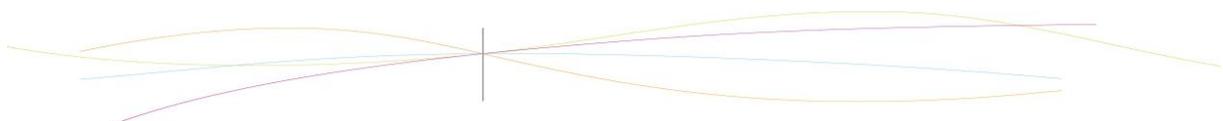


figure 13 : *Bonne ligne de points.*

Le repérage dans l'espace complique fortement l'élaboration d'un algorithme de recherche de points.

Pourtant il faut avouer que l'extraction manuelle de l'information 3D est une tâche longue et fastidieuse et donc qu'une recherche automatique des points devrait être un point intéressant à développer. Par exemple, en utilisant la photo de la page précédente, pour modéliser la partie de la boîte de conserve visible, avec la meilleure résolution qu'offre le damier, il faudrait : 338 points, 676 coordonnées, 1 heure 30... (en passant 15 secondes par point).

Une fois l'extraction de l'information 3D terminée et donc que notre feuille de calcul est bien remplie, il nous reste à préparer le fichier texte pour que l'affichage 3D puisse se dérouler dans de bonnes conditions. Blender est un logiciel anglophone, il ne comprend



donc pas la virgule décimale. Pour résoudre le problème il suffit de sélectionner les colonnes de notre feuille de calcul puis d'aller dans le menu déroulant Format puis Cellules... et changer la langue des cellules en anglais. Il suffit alors de copier-coller les colonnes X, Y et Z (ou Z1 selon si Φ est nul ou non).

3.2.3.2. Affichage de l'objet en 3D

Nous allons maintenant nous intéresser à la méthode utilisée pour afficher l'objet étudié en 3 dimensions sur l'ordinateur. Nous nous sommes tout d'abord orientés vers le logiciel Scilab, qui permet d'afficher des courbes et des graphiques en 3D, mais nous avons finalement choisis d'utiliser un logiciel de modélisation 3D. La plupart de ces logiciels étant payants (ou bien gratuit durant un temps limité), nous avons naturellement choisis Blender, qui est une suite libre et gratuite d'animation, modélisation et rendu 3D; ou gravite notamment une communauté très active.

Ainsi, pour créer l'objet en question, on peut séparer la procédure suivie en deux étapes distinctes. Premièrement, il faut que Blender lise la liste des coordonnées des points à afficher. Puis ensuite, il faut créer la surface qui reposera sur ces points (le maillage).

3.2.3.2.1. Lecture et affichage d'une liste de points

L'utilisation de Blender étant principalement axée sur la modélisation et l'animation 3D, la simple modélisation d'un objet 3D à partir d'une liste de coordonnées de points placée dans un fichier texte n'était pas accessible directement dans les options proposées. Toutefois, Blender offre la possibilité de créer des scripts basés sur le langage Python.

Nous avons ainsi utilisé un premier script (trouvé sur <http://jmsoler.free.fr>) permettant de lire la liste de coordonnées de points pour ainsi les afficher.

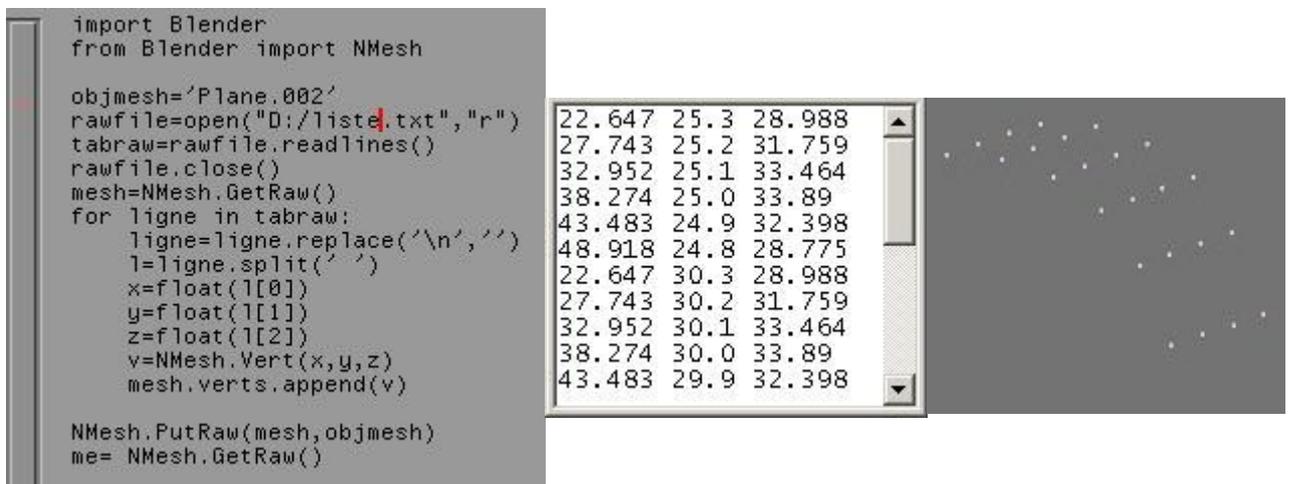
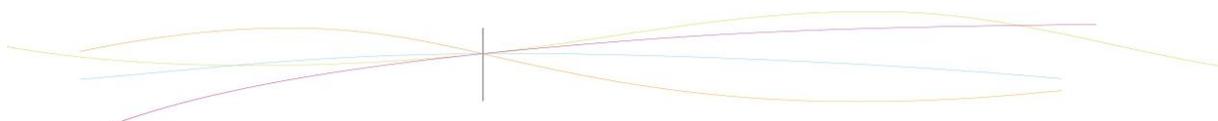


figure 14 : Le script, la liste de points et le rendu de l'affichage des points.

Ainsi, comme le montre le script, chaque ligne du fichier texte correspond aux coordonnées d'un point (chaque colonne représentant donc respectivement les coordonnées en X, Y et Z séparées par un espace).

3.2.3.2.2. Création des facettes (maillage)

Maintenant que les points sont créés, il faut donc établir une surface reposant sur ces points à l'aide de polygones. Bien qu'il soit possible de mettre en place manuellement les facettes en choisissant les points faisant rôle de sommet, le nombre de facettes à mettre en place devient rapidement trop élevé. Malheureusement, n'ayant pas trouvé d'options



permettant de créer automatiquement des facettes à partir de points situés dans l'espace, nous avons dû faire un script qui créera les facettes entre ces points. A vrai dire, nous avons trouvé un logiciel (shareware, donc utilisable en période limitée), qui permettait de créer un maillage automatiquement à partir d'une liste de points, toutefois cela nécessitait une importation/exportation des données entre Blender et ce logiciel.

Le maillage automatique d'un nuage de points peut être un problème très complexe lorsque les points ont une disposition chaotique. On pourrait penser qu'il suffit de calculer les distances entre chaque point pour ensuite créer les polygones entre les points les plus proches mais cela peut entraîner dans certains cas un résultat absurde.

Prenons par exemple le cas d'un tuyau (ou d'un tunnel):

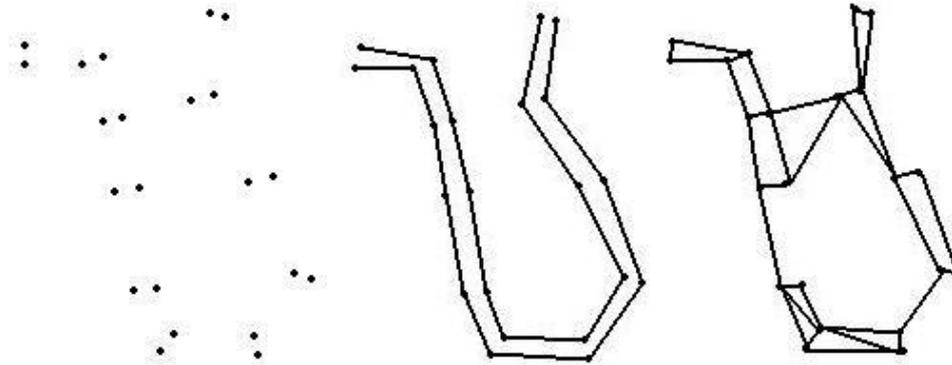


figure 15 : Exemples de maillages.

Ainsi, comme le montre la 3ème image, la reconstruction de surface à partir d'un maillage automatique suivant la logique des points les plus proches donneraient un résultat aberrant. Toutefois, dans le cas de notre projet, nous avons affaire à des points étant disposés dans un agencement ordonné. En effet, étant donné qu'on projette une grille, tout les points sont agencés les uns par rapport aux autres d'une manière très ordonnée. De plus, les points situés dans le fichier texte sont déjà rangés correctement, et lors de l'exécution du script précédant, les points ont été numérotés suivant leur ligne dans le fichier texte. Il est ainsi simple d'automatiser la création de facettes.

Il suffit donc de créer des polygones de la façon suivante:

```

import Blender
from Blender import NMesh

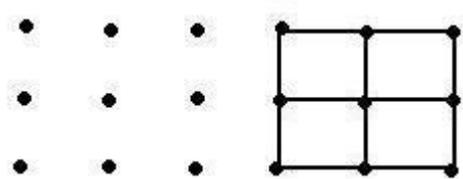
#a et b représentent le nombre de points
#sur chaque côté du rectangle
a=4
b=6

#Repère les points préalablement placés
lobjet = Blender.Object.Get('Mesh')
me=lobjet.getData()

#Créer les facettes
for i in range(0,a*b-b-1,1):
    if not i%b==b-1:
        f=NMesh.Face()
        f.v.append(me.verts[i])
        f.v.append(me.verts[i+1])
        f.v.append(me.verts[b+i+1])
        f.v.append(me.verts[b+i])
        me.faces.append(f)

NMesh.PutRaw(me, "plane", 1)
Blender.Redraw()

```



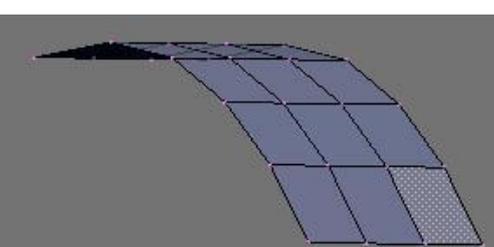
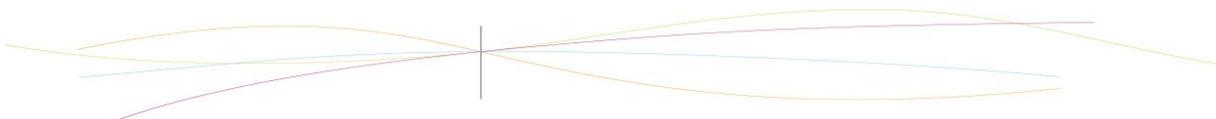


figure 16 : Création de polygones quadrangulaires (parallélogramme).



Mais on peut également créer les surfaces à partir de polygones triangulaires.

```

import Blender
from Blender import NMesh

#a et b représentent le nombre de points
#sur chaque côté du rectangle
a=4
b=6

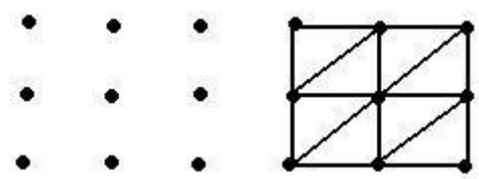
#Repère les points préalablement placés
lobjet = Blender.Object.Get('Mesh')
me=lobjet.getData()

#Créer les facettes
for i in range(0,a*b-b-1,1):
    if not i%b==b-1:
        f=NMesh.Face()
        f.v.append(me.verts[i])
        f.v.append(me.verts[i+1])
        f.v.append(me.verts[b+i])
        me.faces.append(f)

        f=NMesh.Face()
        f.v.append(me.verts[i+1])
        f.v.append(me.verts[b+i])
        f.v.append(me.verts[b+i+1])
        me.faces.append(f)

NMesh.PutRaw(me,"plane",1)
Blender.Redraw()

```



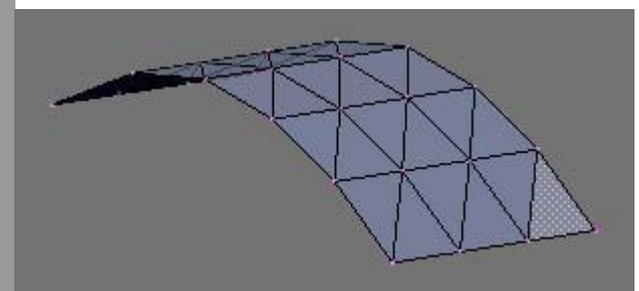


figure 17 : Création de polygones triangulaires.

Ainsi, on peut noter que Blender permet de créer des polygones quadrangulaires; par contre, si les 4 sommets du polygone ne se retrouvent plus dans le même plan, alors le parallélogramme est automatiquement scindé en 2 triangles. L'une ou l'autre méthode mène donc sensiblement au même résultat, toutefois, la solution des polygones quadrangulaires semble être préférable.

Le résultat final représente bien la forme d'une boîte de conserve :

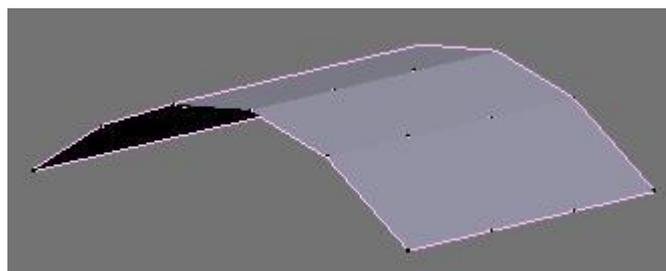
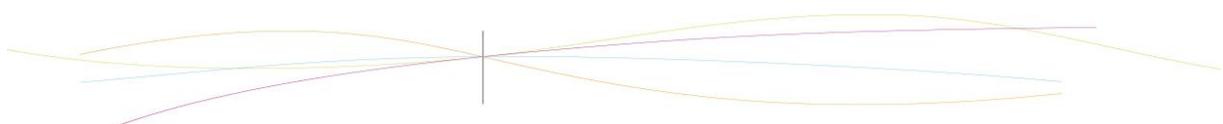


figure 18 : Résultat final

3.3. Comparaison des différentes méthodes

En approfondissant notre projet, nous avons pu détecter des différences et des ressemblances entre la méthode de projection d'un quadrillage et les autres méthodes. Comme dit précédemment, la grosse différence entre notre méthode et celles de stéréoscopie et photogrammétrie réside dans le fait que les résultats de ces dernières ont un effet visuel et non une restitution de volumes. Nous nous sommes intéressés surtout aux différentes méthodes de scanner 3D qui rejoignent le plus notre démarche. Grâce à nos recherches, nous nous sommes aperçus que la reconstruction 3D par projection d'une grille



est un domaine en cours de recherche et qu'elle porte le nom de **scanner à lumière structurée**.

La projection d'une grille sur une surface s'apparente avec les scanners sans contact actif (décrits au 3.1.2.2.). En effet, par analogie avec la triangulation, nos calculs de projection utilisent les distances émetteur-appareil photo et les angles du côté de l'appareil photo.

Cependant notre projet comporte des limites qu'il est judicieux de souligner. Tout d'abord dans l'aspect théorique, les calculs de projection ont été faits avec l'hypothèse de rayons parallèles provenant du projecteur, alors que dans la pratique ils sont émis dans toutes les directions. Ensuite, les objets étudiés sont de petite dimension (ex : une boîte de conserve). Effectivement, au-delà d'une certaine taille, il est difficile de recouvrir entièrement l'objet, prenons ici l'exemple d'une voiture, et obtenir une netteté d'image suffisante. Enfin l'acquisition des coordonnées des points est longue et contraignante : il s'agit de faire pour chaque point, un relevé manuel des coordonnées, ce qui s'avère être très vite contraignant (pour le visage d'Amandine, il a fallu prendre un par un les coordonnées de 100 points).

4. CONCLUSIONS ET PERSPECTIVES

Les résultats obtenues en fin de projet étaient inespérés, la qualité de la modélisation du visage est surprenante comparée aux moyens utilisés pour l'obtenir. Il est dommage que nous ayons eu trop peu de temps pour approfondir l'automatisation de l'extraction des données. En effet, il faut reconnaître que nous avons perdons beaucoup de temps au début du projet pour trouver les formules mathématiques. Nous nous sommes orientés sur la piste des reliefs relatifs, sans prendre le recul nécessaire pour trouver la solution avec les coordonnées absolues, bien que cela ait été proposé. Une fois les formules et les expérimentations finies, les formules vérifiées, testées et des bons résultats obtenus, il n'a pas resté beaucoup de temps pour peaufiner le projet et l'améliorer.

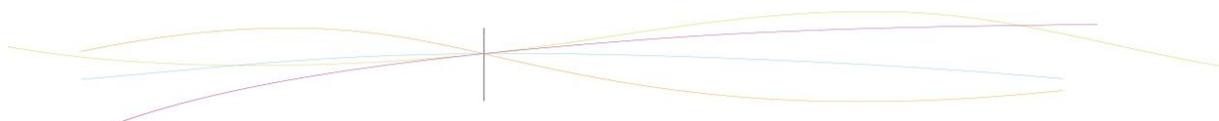
4.1. Conclusions sur l'apport personnel de cette U.V. projet

Cette UV nous a permis de travailler en groupe de façon relativement autonome, de construire un projet nous même, du début jusqu'à la fin, ce qui fut une très bonne expérience pour nous tous. Le projet nous a donné une autre approche de l'expérience de physique. Alors que depuis le collège, l'expérience sert à donner un exemple d'application d'une loi ou un mécanisme pour la mettre en évidence, ici, l'expérience s'inscrit dans un processus de recherche, elle sert à soumettre la théorie à un test, rajoute des contraintes mais donne aussi du recul face à la théorie, et finalement, permet d'obtenir un résultat concret.

De plus, le sujet nous a intéressé et nous a fait découvrir aussi bien les méthodes de restitutions et d'acquisition 3D que des logiciels gratuits, libres et efficaces pour réaliser notre projet. Enfin, nous avons eu l'occasion de discuter par courrier électronique avec des professionnels du domaine qui ont bien voulu nous aider.

4.2. Perspectives pour la poursuite de ce projet

Concernant la poursuite du projet, il faut voir que de nombreuses améliorations sont possibles. La partie d'extraction de l'information 3D est clairement la partie qui a le plus besoin d'être améliorée, comme on l'a dit l'extraction manuelle est une tâche longue et ennuyeuse. Nous avons donc cherché à créer un script Scilab permettant de cliquer sur les



deux photos pour obtenir les coordonnées des points. Malheureusement un problème avec la taille de la pile des variables Scilab (stacksize) nous a empêchés d'utiliser les photos. Même en réduisant leurs dimensions, l'affichage des photos se lance, mais crashe rapidement Scilab. Le problème peut venir du logiciel Scilab, de la toolbox SIP ou d'une mauvaise utilisation des fonctions graphiques.

L'idéal pour l'extraction de l'information est de pouvoir l'automatiser. Toutefois, on est vite confronté à deux difficultés qui sont : la détection et la réduction d'un point en un seul pixel précis et l'identification des positions d'un même point selon les photos.

On l'a vu précédemment, la projection d'un damier n'est pas adaptée à la recherche d'un pixel désignant un point particulier. On pourrait plutôt projeter un nuage de points, on écrira alors un algorithme qui cherche, par exemple, le pixel le plus bas d'un point. On peut ensuite développer un programme qui affiche les photos et propose d'identifier les différentes positions des points, il suffirait alors de cliquer sur les photos, sans s'embêter à zoomer/dézoomer, reporter les coordonnées dans une feuille de calcul, etc. Dans la pratique, les contraintes expérimentales (qualité/netteté des photos) pourraient nuire à la bonne détection des pixels.

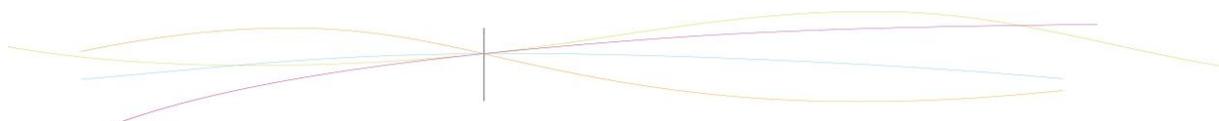
Maintenant, si l'on veut automatiser complètement l'extraction des coordonnées, le problème est plus compliqué. Admettons que l'on ait résolu le problème de la détection des points, on souhaite maintenant trouver automatiquement le couple de points (point sans l'objet, point avec l'objet). Comme on l'a vu dans la partie consacrée à l'extraction de l'information, la solution qui consiste à suivre une ligne de point ne s'avère pas aussi simple qu'elle n'y paraît. Quand la tâche est réalisée manuellement, on s'aide d'un point de référence centré sur l'objet et on commence par lier ses coordonnées. On peut donc imaginer un programme capable de faire la même opération. Il faudrait qu'il soit en mesure de détecter les bords de l'objet et il faudrait aussi lui fournir plus de points de référence lors de la projection. Le programme pourrait alors anticiper la forme de l'objet et donc prendre les bons points, malheureusement un tel programme est très compliqué à développer et ne peut fonctionner qu'avec des formes relativement simples.

Une autre solution serait de rajouter de l'information dans les points de la projection. Prenons un exemple simple, on projette 3 points, l'un de couleur rouge, l'autre bleu et le dernier vert. Une simple analyse RVB des photos permettra de dissocier les 3 points. Mais si la solution marche bien pour 3 points, pour des centaines de points, sachant que les photos présentent des différences de couleurs, une solution basée sur les couleurs n'est peut-être pas applicable.

Enfin, on pourrait projeter le nuage point par point, en prenant des photos. La recherche des points serait dans ce cas extrêmement simple. Mais on voit apparaître de nouvelles contraintes : la capacité de stockage des photos et la durée de l'expérience. Pour reconstruire un objet avec 100 points, il faut alors 200 photos. En combinant judicieusement cette méthode avec celle des couleurs, on peut espérer, par exemple, prendre 20 photos pour 100 points. Bien sûr cela fait beaucoup de photos mais c'est de loin la solution la plus simple et la plus efficace.

Une autre amélioration possible peut être la restitution des informations connus dans plusieurs prises de vue différentes. Par exemple, si on souhaite reconstruire un visage, en prenant qu'une seule photo, on ne peut reconstruire qu'un seul profil du visage. En effet, les formes du visage (le nez, etc.) cachent les points situés sur l'autre profil. Il suffit d'un ou deux points de référence sur les deux photos pour rectifier les décalages en X, Y et Z.

En ce qui concerne la partie affichage de l'objet en 3D, le script qui permet de faire un maillage automatique est très nettement améliorable. En effet celui-ci pose plusieurs contraintes, deux contraintes sur la distribution des points qui doit être uniforme et rectangulaire, et une contraire sur l'ordre de l'extraction des points, qui doit se faire suivant les X puis les Y croissants (par exemple). Il serait alors intéressant d'approfondir

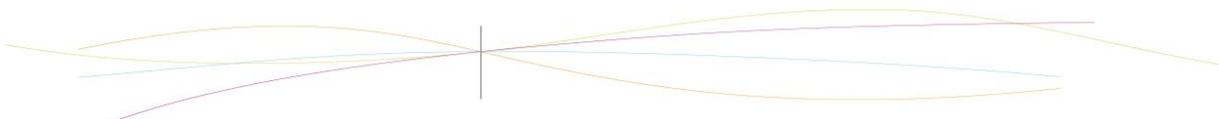


l'algorithme de maillage pour en supprimer ces contraintes et améliorer le rendu final de la restitution 3D, si possible.

On pourrait aussi orienter les recherches pour améliorer l'affichage du point de vue texturing de l'objet. Obtenir la texture, lorsque l'on prend la photo, pour ensuite l'appliquer sur l'objet, lors du rendu sous Blender, n'est pas facile, il faut manipuler l'image et la déformer pour qu'elle convienne à la surface 3D.

On peut aussi s'intéresser à la qualité de la restitution, c'est-à-dire développer des expériences pour s'assurer de la fiabilité des formules mathématiques, il faut savoir que les méthodes de restitution 3D trouvent principalement leurs applications dans le contrôle dimensionnel de pièces usinées, d'où l'importance de la fiabilité et de la précision. Comme nous l'avons dit précédemment nous ne sommes pas sûrs de l'impact que peut avoir le cône de projection sur les résultats, sachant que nous faisons l'approximation des rayons parallèles. La prise de photos présente aussi un problème de champ de profondeur et donc de netteté d'image, il faudrait chercher les techniques photographiques ou cinématographiques qui permettent d'augmenter les champs de profondeurs.

Une fois ces améliorations réalisées, nous aurons alors un dispositif fiable et efficace, et on pourra se tourner vers des perspectives plus ambitieuses, comme l'analyse d'une vidéo pour faire un rendu d'animation, ou encore la restitution de l'intégralité de l'objet plutôt que de l'une de ses faces.



5. BIBLIOGRAPHIE

[1] lien internet : <http://www.vocanson.fr> (valide à la date du 20/04/2009).

[2] lien internet : <http://fr.wikipedia.org/wiki/Photogramm%C3%A9trie> (valide à la date du 15/04/2009).

[3] lien internet : <http://fr.wikipedia.org/wiki/St%C3%A9r%C3%A9oscopie> (valide à la date du 15/04/2009).

[4] lien internet : <http://www.insa-strasbourg.fr/fr/actualites/andlau-juillet.html> (valide à la date du 10/06/2009).

[5] lien internet : http://www.urbica.net/fr/processus/processus_industrie.php (valide à la date du 10/06/2009).

[6] lien internet : <http://www.tplm-3d.fr/lasergrammetrie.htm> (valide à la date du 23/04/2009).

[7] lien internet : <http://www.techno-science.net/?onglet=news&news=5344> (valide à la date du 18/02/2009).

6. ANNEXES

6.1. Listings des programmes réalisés

```
// Modélisation 3D à partir de stries 2D
// Nécessite siptoolbox, un dossier Entrée et un dossier Sortie dans le dossier d'exécution du script
```

```
// *--Initialisation--*
```

```
//Redimensionnement du tampon :
```

```
stacksize(10000000);
```

```
//Définition du seuil :
```

```
seuil=0.55;
```

```
//Repérage du dossier d'exécution du script & changement de dossier courant :
```

```
cheminRacine=get_file_path('Analyse image.sci');
```

```
chdir(cheminRacine);
```

```
//Chargement de l'image :
```

```
image=imread('Entrée/photo.jpg');
```

```
// *--Traitement de l'image--*
```

```
//Normalisation de la matrice image :
```

```
image=normal(image);
```

```
//Niveau de gris d'image :
```

```
niveauGris=im2gray(image);
```

```
//Récupération de la dimension de niveauGris :
```

```
[largeur,hauteur]=size(niveauGris);
```

```
//On corrige le niveau de gris ( image "purement noire & blanche" ) :
```

```
for i=1:largeur
```

```
    for j=1:hauteur
```

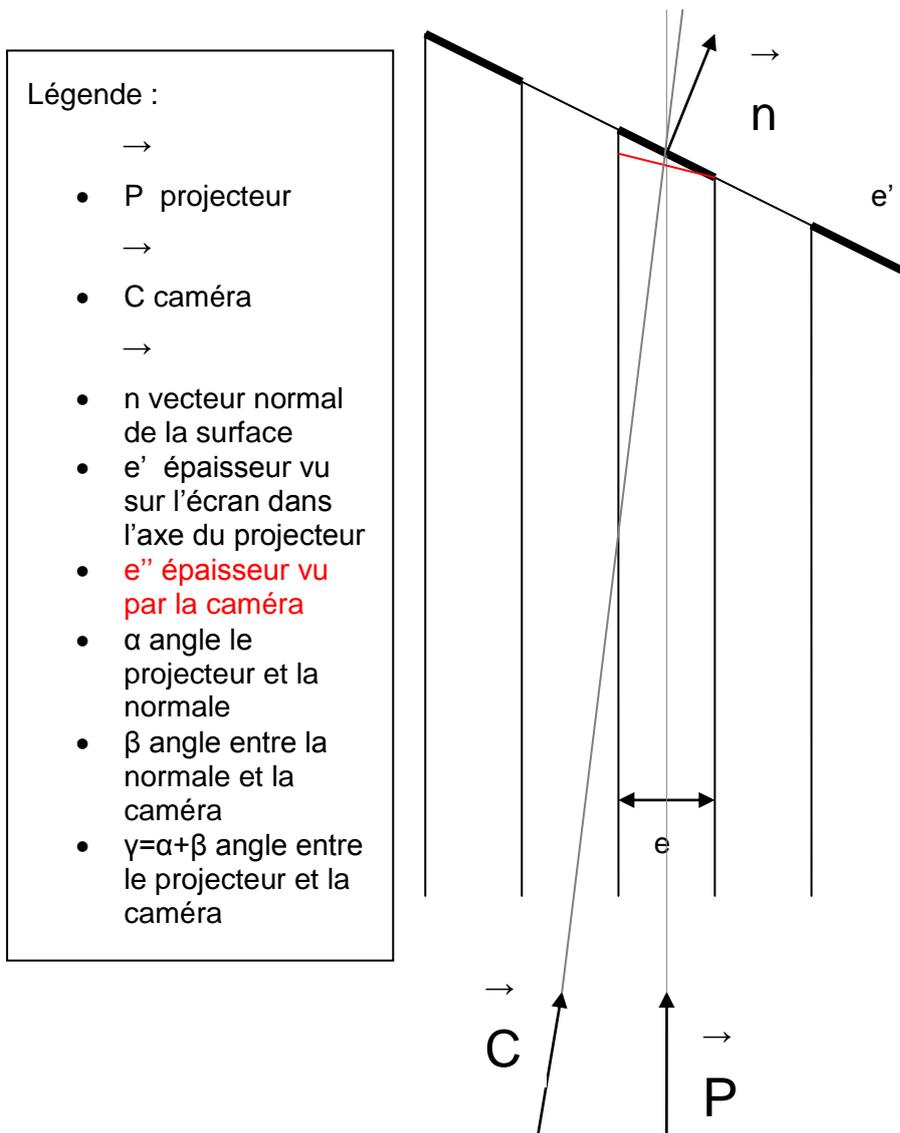
```
        if niveauGris(i,j)<seuil then
```

```

niveauGris(i,j)=0 ;
else
niveauGris(i,j)=1 ;
end
end
end
imwrite(niveauGris,'Sortie/noirEtBlanc.jpg');

```

6.2. Schémas d'études



eq 8. $e' = e / \cos(\alpha)$

eq 9. $e'' = \cos(\beta) * e'$

eq 10. $e''/e = \cos(\gamma - \alpha) / \cos(\alpha)$

