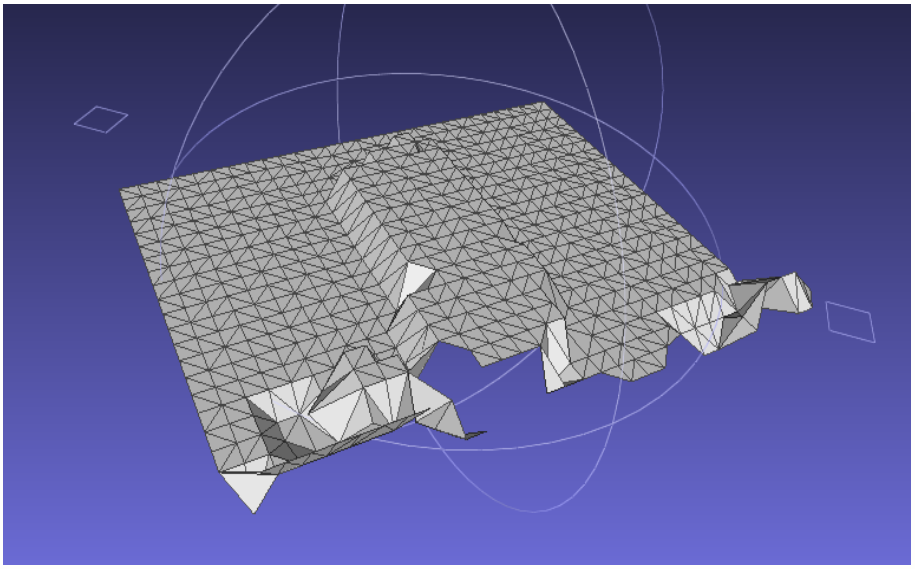
Projet de Physique P6-3

STPI/P6-3/2011 – 040

Reconstitution de la 3D à partir d'images 2D par projection de stries



Étudiants :

Ken DEGUERNEL **Clotilde LEBRUN**

Jérémy ELOY **Léo LEFEBVRE**

Elodie GUERY **Blandine SAINTEMARIE**

Ya HUANG **Fanny VERRIER**

Enseignant-responsable du projet :

Jérôme YON

Date de remise du rapport : ***18/06/11***

Référence du projet : ***STPI/P6-3/2011 – 040***

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

Type de projet :

- ***expérimental***
- ***informatique et mathématique***

Objectifs du projet :

Notre projet est la continuation d'une étude qui a commencé il y a deux ans. Celle-ci avait pour objectif de reconstituer un objet photographié (visage humain ou une main) en 3 dimensions sur l'ordinateur. Néanmoins, ceux-ci ne sont visibles que d'un seul côté. Cette année, notre but est de représenter le maximum de faces de l'objet et de pouvoir le faire tourner sur l'ordinateur.

Mots-clefs du projet :

- ***2D***
- ***reconstitution de surface 3D***
- ***corrélation d'images***
- ***rotation de l'objet***

Table des matières

1. INTRODUCTION

Notre projet de P6-3 a pour principal objectif d'améliorer les deux projets antérieurs consistant à reconstituer un objet en 3D à partir d'images 2D (photographies).

En effet, il y a deux ans, les étudiants avaient formulé une fonction mathématique permettant de calculer la profondeur z en fonction de x et de y. Nous reprendrons cet aspect mathématique dans notre projet. Ainsi, grâce à la corrélation de deux photographies : une sans l'objet (un visage), et l'autre avec, ils sont parvenus à le reconstituer en 3D.

Puis, l'année dernière, le projet a été repris et l'extraction de l'information a été automatisée. En effet, il est beaucoup plus rapide de repérer les décalages de points entre les deux photographies grâce à un programme informatique plutôt que manuellement.

Cette année, nous reprendrons cette méthode de corrélation d'images. Néanmoins, nous prendrons plusieurs clichés d'un objet qui tourne autour de son axe principal. Ainsi, toutes les faces cachées pourront être visualisées et l'objet reconstitué sera plus complet. En outre, l'objectif de notre projet est de reconstruire l'objet en 3D le plus réellement possible. D'autre part, nous rendrons portable le protocole et le traitement d'image : nous établirons un tutoriel et un protocole expérimental afin de faciliter la reprise en main du projet.

Nous donnerons donc, dans un premier temps, quelques exemples de méthodes existantes pour passer de la 2D à la 3D. Puis, nous expliquerons l'organisation du travail et la répartition des tâches. Enfin, nous décrirons notre expérience avant de nous attarder sur la méthode de traitement des images, puis la description et l'analyse de nos résultats.

2. MÉTHODOLOGIE / ORGANISATION DU TRAVAIL

Au tout début, nous nous sommes répartis en quatre groupes : Léo et Ken ont pris en main les logiciels Open CV et le script (Scilab) et Jérémy et Clotilde se sont occupés de la réalisation de la surface grâce à Blender et Meshlab. Enfin, les deux derniers groupes sont des sous-groupes de la partie expérimentale : Élodie et Ya devaient réfléchir à une expérience avec une caméra qui tourne alors que Fanny et Blandine s'intéressaient à un objet qui tourne et une caméra fixe (voire 4.2).

Les premières séances, nous nous sommes familiarisés avec les projets antérieurs, pour bien comprendre la méthode utilisée. Puis, dès la troisième séance, nous sommes passés à l'aspect pratique, pour avoir des images à traiter. Ainsi, Léo et Ken ont pu commencer le traitement des images : les redimensionner, les compresser et extraire les coordonnées des points. Le groupe chargé du traitement graphique a lui aussi eut un premier aperçu du travail attendu.

Enfin, pour communiquer sur l'avancement du projet, nous avons dès le départ créé un groupe sur google où tous les membres du projet étaient inscrits. De plus, toutes les semaines, chaque sous-groupes précédemment cités devait rédiger un compte-rendu de la séance afin d'avoir un suivi de ce qui avait été réalisé les séances antérieures.

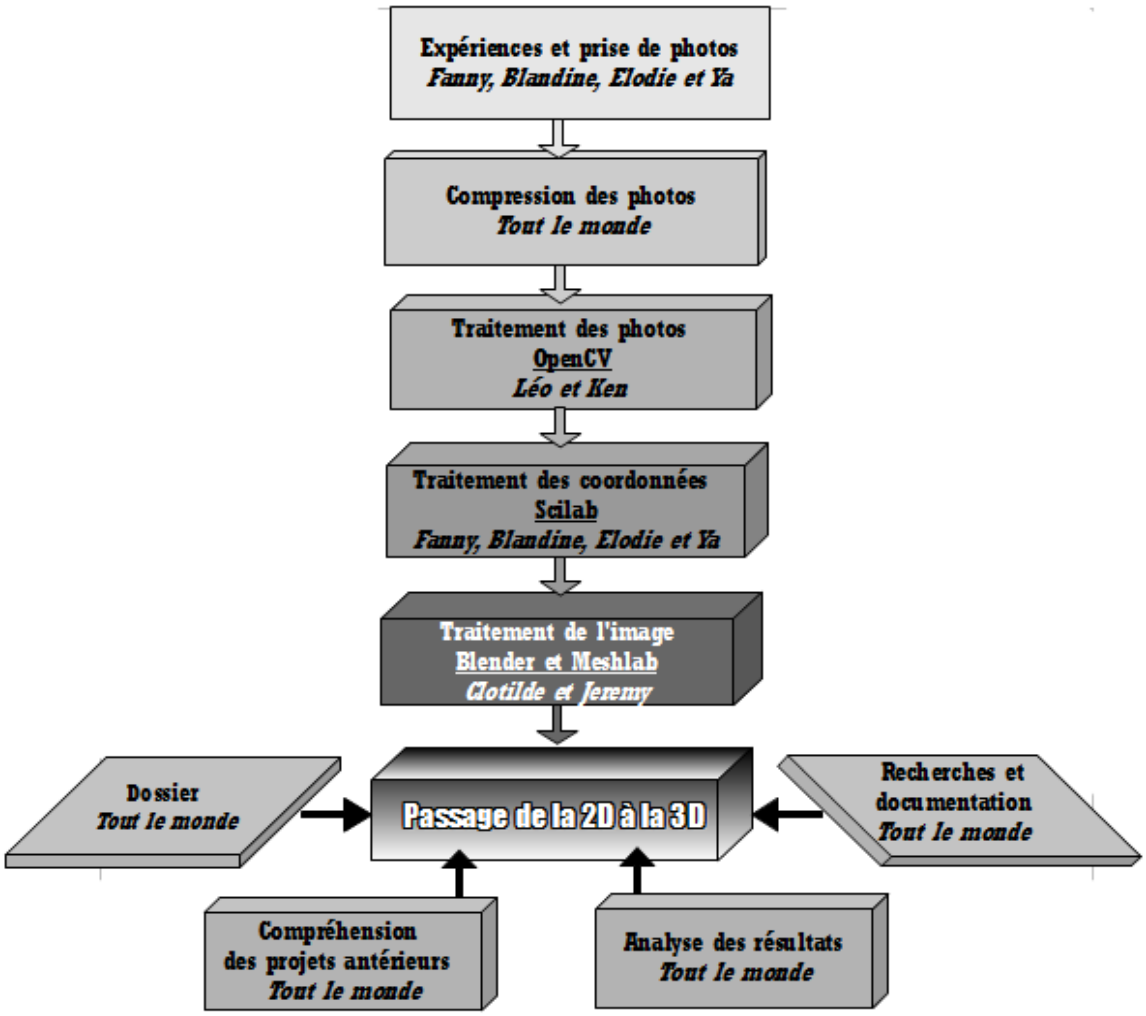


Illustration 1: Diagramme représentant la répartition du travail

3. MÉTHODES EXISTANTES POUR LA RECONSTRUCTION D'IMAGES 3D

Les groupes des années précédentes avaient présentés déjà plusieurs méthodes de reconstitution 3D, telles que les scanners 3D, la méthode des ombres ou morphing 3D, se basant sur les similarités des visages pour les reconstruire. Enfin nous avons eu un aperçu de l'auto-calibrage à partir d'images bidimensionnelles. Cette année, nous avons choisi de vous présenter de nouvelles méthodes, telles que la tomographie 3D, la résonance magnétique, la méthode des ultrasons et la projection en relief stéréoscopique. Nous finirons par exposer le principe de notre méthode : la corrélation d'image.

3.1. La tomographie

La tomographie est une technique de reconstruction 3D qui se base sur de nombreuses mesures appliquées à l'extérieur de l'objet. Ces mesures donnent des informations sur la nature et les propriétés de l'objet en profondeur, et sont très utilisées en imagerie médicale pour reconstituer par exemple la forme d'un organe supposé malade, mais aussi en géophysique pour reconstituer une structure géologique.

La tomographie utilise principalement les rayons X pour recéler les informations sur l'objet, mais peut être adaptée à tous types de rayonnement pour obtenir des informations complémentaires.

3.1.1. La tomographie par absorption des rayons X

Cette technique repose sur l'analyse de l'interaction du rayon X avec la matière. Ces rayons sont captés après avoir traversé l'objet par des détecteurs qui les enregistrent dans toutes les directions de propagation. Ces données sont ensuite traitées mathématiquement pour donner une image numérique. Celle-ci fait apparaître des niveaux de gris dont chacun traduit l'atténuation du faisceau traversant l'objet, donc les variations d'absorption radiologiques, et ainsi les différences de composition de l'objet. En outre, cela permet de mettre en évidence toute hétérogénéité, singularité, vide ou inclusion de l'objet.

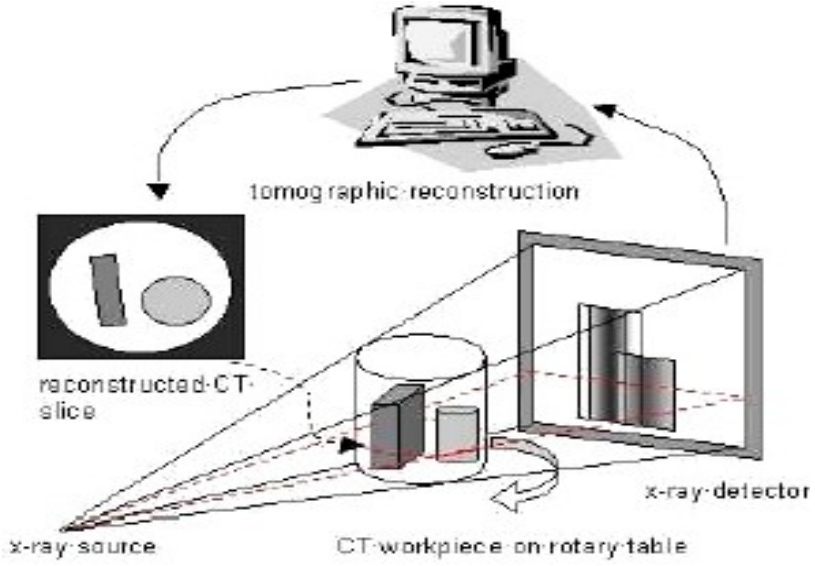


Illustration 2: Schéma de la tomographie par rayon X

3.1.2. La tomographie par émission de positons (TEP)

La tomographie par émission de positons est une méthode d'imagerie médicale permettant de mesurer en trois dimensions l'activité métabolique d'un organe.

La technique est ici d'utiliser un traceur dont la connaissance de ses propriétés biologiques nous indique l'organe dans lequel il ira. Ce traceur est marqué par un atome radioactif émettant des positons dont l'annihilation produit elle-même deux photons. La trajectoire de ces photons peut ensuite être détectée par la caméra TEP, localisant ainsi le lieu de leur émission, et donc la concentration du traceur en chaque point de l'organe. A l'aide de ces données, une image numérique est calculée représentant les zones de forte concentration du traceur par un dégradé de couleur.

3.1.3. La tomographie optique ou tomographie 3D par fluorescence

La tomographie optique utilise la lumière infrarouge pour reconstruire en 3D les zones fluorescentes. Cette nouvelle et prometteuse technique permet de visualiser la structure biologique sous la peau par l'injection de marqueurs fluorescents. On cherche ici à mesurer les variations de fluorescence transmises par les organes et les tissus selon leur nature. La lumière infrarouge excite la matière et fait ressortir les zones plus ou moins opaques que l'on enregistrera : on dresse alors à partir de ces résultats des calculs pour reconstruire numériquement les zones émettant une fluorescence.

3.2. Résonance magnétique

Prenons l'exemple de l'imagerie par résonance magnétique (IRM). Cette technique d'imagerie médicale permet d'obtenir des vues 2D ou 3D de l'intérieur du corps. Son principe est le suivant : grâce à un champ magnétique (produit par un aimant supraconducteur à retirer), les tissus sont magnétisés par alignement des moments magnétiques de spin. Des champs magnétiques oscillants plus faibles sont appliqués et modifient légèrement cet alignement, ce qui produit un phénomène de précession. Celui-ci donne lieu à un signal électromagnétique mesurable. En localisant l'origine de ce signal dans l'espace, il est alors possible de reconstruire une image en 2 dimensions puis en 3 dimensions de la composition chimique et donc de la nature des tissus biologiques explorés.

3.3. La méthode des ultrasons

Les ultrasons peuvent également servir à reconstruire des surfaces, notamment en médecine avec l'échographie 3D.

3.3.1. Principe

Tout d'abord, une sonde est utilisée afin d'envoyer des ultrasons dans le corps du patient. Cependant, il existe une fine couche d'air entre la peau et la sonde. C'est pourquoi, au préalable, il faut appliquer sur la peau un gel spécifique qui permet la conduction des ultrasons sans réfraction, ni réflexion au niveau de la peau.

3.3.2. *Apparition des différents tissus de l'organisme*

L'image qui apparaît sur l'écran se fait en niveaux de gris selon l'intensité de l'écho en retour des ultrasons.

Les liquides simples (sans particules en suspension) laissent passer le son. Ils ne se signalent donc pas par des échos et apparaissent noir sur l'écran.

Les liquides avec particules (sang, mucus) renvoient de petits échos. Ils sont donc gris sur l'écran.

Enfin, les structures solides (os, par exemple) renvoient mieux les échos. On voit donc une forme blanche avec une ombre derrière.

Pour réaliser une forme tridimensionnelle, il faut utiliser une sonde volumique qui réalise automatiquement un balayage de la zone à examiner. La succession des plans de coupes ainsi acquis a pour but de construire une matrice volumique. Schématiquement, celle-ci se présente sous la forme d'un parallélépipède dont l'image d'acquisition élémentaire constitue les axes X et Y et dont l'empilement des images d'acquisition constitue l'axe Z.

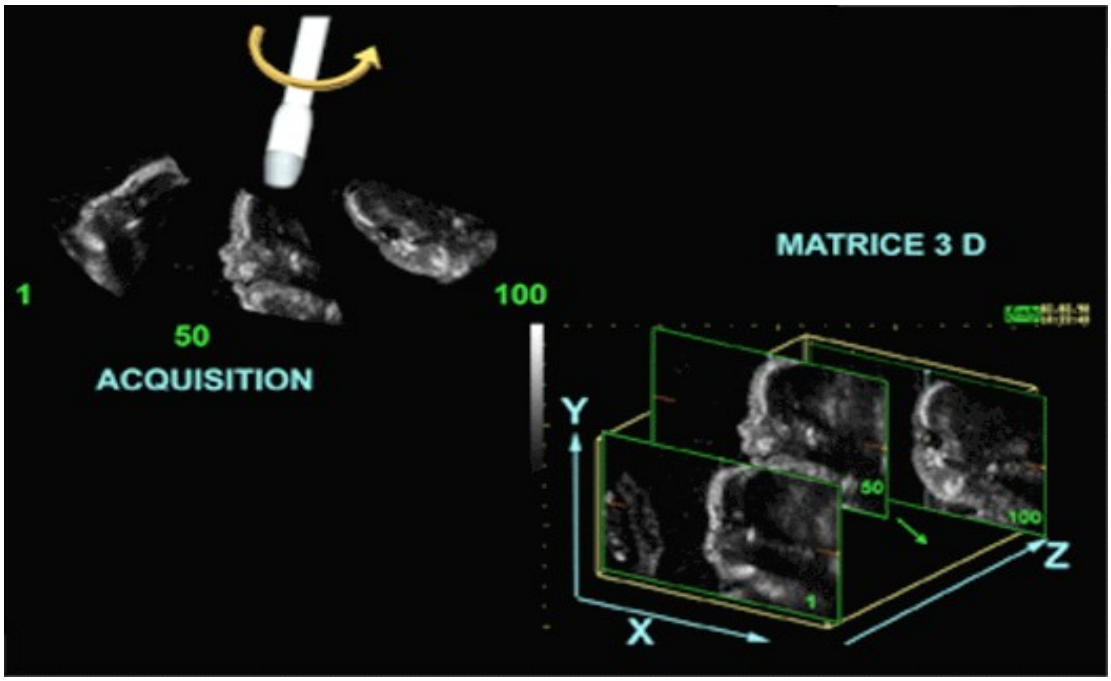


Illustration 3: Schéma représentant le fonctionnement d'une échographie 3D

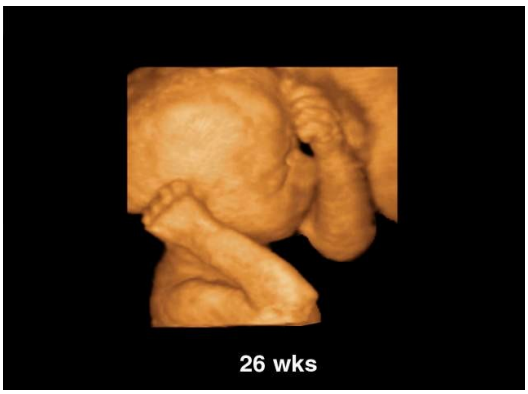


Illustration 4: Résultat d'une échographie en 3D

3.4. La stéréoscopie

La stéréoscopie est l'ensemble des techniques mises en œuvre pour reproduire une perception du relief à partir de deux images planes, l'une vue par l'œil gauche et l'autre par l'œil droit. Le relief est produit à partir des différences entre les images vues par chacun des deux yeux.

Cependant, lorsque les deux images sont devant nous, nous regardons l'image vue de gauche avec nos deux yeux, idem pour l'image vue de droite. Au final, nous perdons la sensation de relief ! Il existe ainsi plusieurs procédés pour reconstituer une vision stéréoscopique, c'est-à-dire s'arranger pour que l'œil droit voit "l'image vue de droite" pendant que l'œil gauche voit "l'image vue de gauche". On peut citer la stéréoscopie par vue croisée, vue parallèle, avec filtre de couleur … À noter que l'informatique apporte de nouvelles possibilités. Par exemple, il existe des lunettes qui obscurcissent alternativement l'œil droit et l'œil gauche, synchronisées avec un écran d'ordinateur envoyant alternativement une image pour l'œil gauche et une image pour l'œil droit.

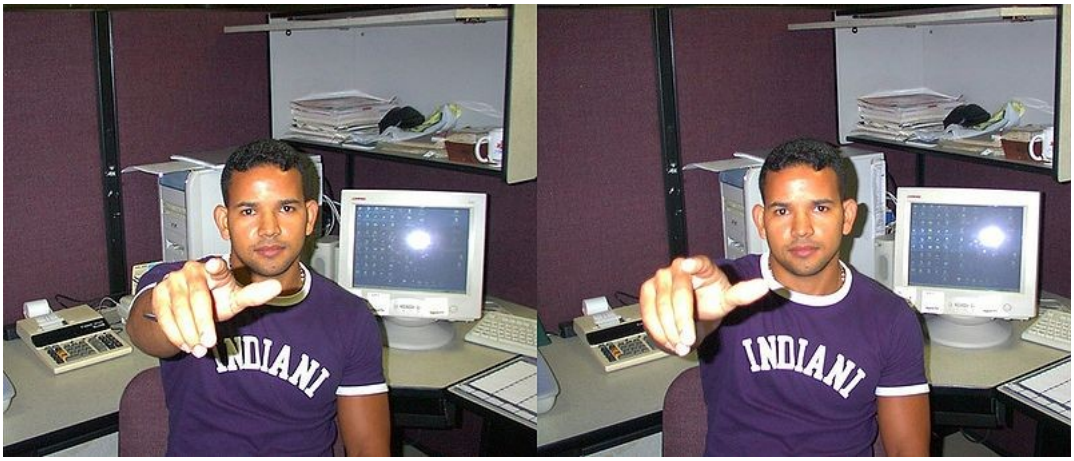


Illustration 5: Un couple stéréoscopique : images vues par l'œil gauche et l'œil droit

3.5. Corrélation d'images

Afin de reconstruire en 3D notre objet à partir des photos prises, nous avons utilisé la méthode de corrélation d'images. Elle permet de mesurer les déplacements, les transformations subies entre deux images.

Dans notre cas, nous prenons une photo sans objet. L'image est alors découpée en de nombreuses petites parties, le but étant de retrouver chacune d'entre elles dans la photo prise avec l'objet. Une fois cette opération effectuée, il suffit de calculer le vecteur déplacement, qui nous permet de déterminer l'épaisseur de chaque fragment d'objet contenue dans chaque partie. Au final, nous obtenons le relief entier de l'objet. Cette procédure s'effectue informatiquement à l'aide d'un programme C++ et de bibliothèques OpenCV.

4. LA THÉORIE

4.1. Pré-requis

Afin de reconstituer notre objet en 3D, nous avons besoin d'éléments mathématiques. D'après la méthode de corrélation d'images, nous devons définir la profondeur de chaque fragment de l'objet dans chaque partie de l'image, selon son déplacement vertical et latéral. Ces résultats ont déjà été établis les années précédentes.

En fait, nous recherchons Xp (coordonnée suivant le vecteur P , profondeur) en fonction de Xh (coordonnée suivant le vecteur H, horizontal) et Xv (coordonnée suivant le vecteur V, vertical). Soit Xp = f(Xh,Xv).

On notera donc:

$$\mathbf{O'P' = X_h * H + X_v * V + f(X_h,X_v) * P}$$

On obtient, après calculs :

$$\mathbf{f(X_h,X_v)=(x_0-x)/\sin(\theta)}$$

$$\mathbf{f(X_h,X_v)=(y-y_0)/(\sin(\varphi)*\cos(\theta))}$$

Avec :

f(Xh,Xv) = Xp : coordonnée suivant le vecteur P (profondeur)

Xh : coordonnée suivant le vecteur H (horizontal)

Xv : coordonnée suivant le vecteur V (vertical)

(x0,y0) : coordonnées du point dans l'image sans objet

thêta : Angle de rotation

Phi : Angle formé entre l'horizontale et l'axe de la caméra

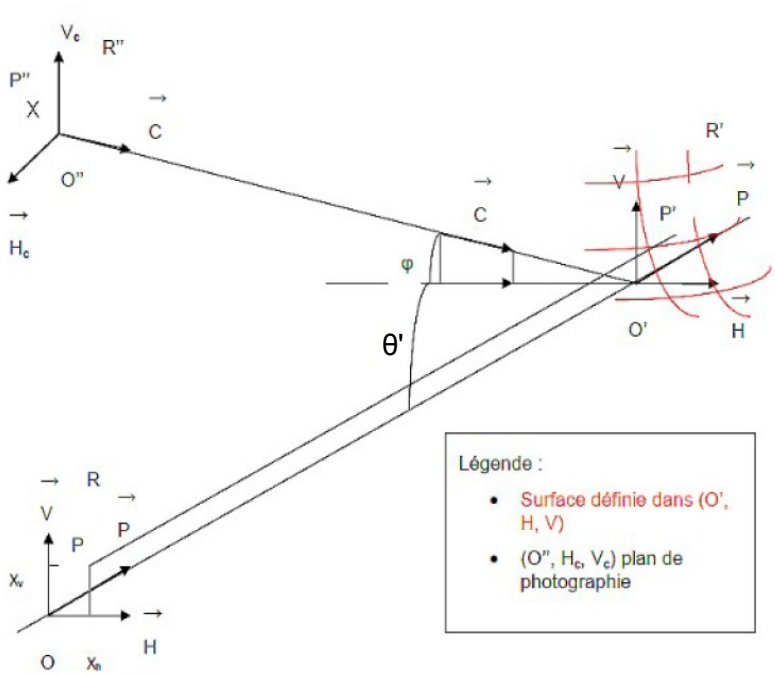


Illustration 6: Schéma de projection

4.2. Notre méthode

Au tout début, nous avions penser à réaliser deux expériences : une avec la rotation de la caméra et l'autre avec l'objet en rotation. Mais, très vite, nous nous sommes rendus compte qu'il serait trop long de mener les deux à bout. Par conséquent, nous nous sommes focalisés sur celle avec l'objet en rotation, qui nous permettrait de révéler une face supplémentaire (l'arrière). En effet, le dessus et le dessous ne peuvent pas être découverts puisque la caméra ne bouge pas et que l'objet est posé sur une table *(voir le protocole expérimental 9.1 en annexe)*.

Puis, nous avons tourné l'objet selon plusieurs angles : π /6, π/4, π /3, π/2, 3π /4... La caméra est restée fixe donc à φ et θ' constants *(voir le schéma du protocole expérimental 9.1 en annexe)*.

Notre but est en fait de reconstruire l'objet dans son intégralité (ou presque) en superposant toutes les photographies que nous avons prises.

Le logiciel Scilab va nous permettre d'atteindre cet objectif. En premier lieu, nous allons ramener tous les points dans un seul repère : celui du mur. Effectivement, l'objet est décalé par rapport au mur d'une distance d = 7,3cm.

Il faut donc, translater le repère (x,y,z) du mur sur les photos en enlevant la distance d. En effet, toutes les photos sont prises par rapport au mur mais la rotation de l'objet se fait selon son axe principal et donc selon le repère (x',y',z') de l'objet.

Puis, une fois la première matrice translation appliquée aux points, nous allons utiliser une matrice de rotation inverse à trois dimensions. Comme toutes les photographies ont été prises en tournant l'objet d'un certain angle, nous devons remettre toutes les images sur le même plan afin de pouvoir les superposer. Ainsi, nous pouvons reconstruire l'image en 3D *(voir le programme Scilab 9.2 en annexe).*

$$R_{\mathbf{y}}(\theta) = \begin{pmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{pmatrix},$$

Illustration 7: Matrice de rotation inverse autour de l'axe y

5. LES EXPÉRIENCES ET LES RÉSULTATS

5.1. Traitement des images

Pour réaliser le traitement des images, un programme codé en C++ a été mis en place l'année dernière afin d'automatiser cette tâche : Pattern_Matching_2.exe. Nous avons donc réutilisé ce logiciel. Cependant, celui-ci nécessite pour son fonctionnement un ensemble de librairies de C++ contenues dans la bibliothèque openCV.

Nous avons donc téléchargé puis installé plusieurs versions d'openCV avant de trouver la bonne car de nouvelles sont parues depuis l'écriture du programme. Après quelques tentatives nous avons enfin réussi à l'exécuter.

Cependant, cette démarche d'installation ne nous a pas semblé des plus judicieuses dans l'optique d'un projet qui se transmet d'une année sur l'autre. Nous avons donc cherché à simplifier cette installation.

Nous sommes alors parvenus à isoler uniquement les bibliothèques nécessaires au fonctionnement du Pattern_Matching_2.exe en ajoutant une à une les bibliothèques demandées au lancement du logiciel dans l'invite commande et en enregistrant le log dans un fichier texte.

Patern_Matching_2.exe DSC_0414.JPG DSC_0398.JPG 10 10 50 > log.txt

En plaçant ces bibliothèques dans le même dossier que celui du programme, il n'est plus indispensable d'installer openCV. Ainsi, les images peuvent être traitées facilement sur plusieurs ordinateurs, car en plus d'être automatisée, cette tâche est dorénavant portable (dossier d'environ 50Mo).

Afin de rendre encore plus aisé le traitement de ces images pour la personne qui n'est trop à l'aise avec l'invite de commande, nous avons créé un script de commande qu'il suffit de modifier comme un fichier texte. Nous avons alors rédigé un tutoriel rapide *(voir 9.4 en annexe)* qui explique la manière de modifier ce fichier (scriptPhoto.bat) ainsi que le reste de la méthode pour traiter les images (où placer les images,comment régler les paramètres, comment filtrer les résultats...).

5.2. Post-traitement des données

L'année dernière, une feuille de calcul Scilab avait été rédigée pour filtrer les points (x,y,z) obtenus et supprimer les valeurs aberrantes. Nous avons donc réutilisé ce fichier. Cependant, il nous a fallu modifier la sortie des résultats.

En effet, avec ce fichier on obtenait une matrice rectangulaire contenant les valeurs de z, avec les lignes et les colonnes correspondantes aux valeurs de x et de y. Or pour réaliser la rotation, nous avons besoin d'une matrice colonne de vecteurs à 3 coordonnées.

Soit x en 1ère colonne, y en 2ème colonne, z en 3ème, et autant de lignes que de points. Nous avons donc utilisé le calcul suivant :

```
[a b]=size(CartoZ)
        for i=1:a
                for j=1:b
                        fprintf(f,"%i\t%i\t%f\n",xTab(i*b-modulo(j,b)),yTab((i*b)-b+j),CartoZ(i,j))
                end
        end
```

Nous avons également intégrer une boucle au fichier afin de pouvoir filtrer toutes les photos en une seule fois, ce qui évite de devoir modifier trop le fichier. En effet, il suffit de rentrer le numéro de photo du fichier de début (ex : 404 pour PatternMatching_2_results404.txt issu de la photo DSC_0404.JPG) et le numéro de fin.

Le logiciel Scilab nous a également permis d'appliquer les matrices translation-rotation inverse-translation comme nous l'avons expliqué au 4.2.(*voir programme en annexe 9.3).*

5.3. Re-positionnement des images

Une fois les images traitées, nous obtenons une liste de point dans un fichier .txt sous la forme :

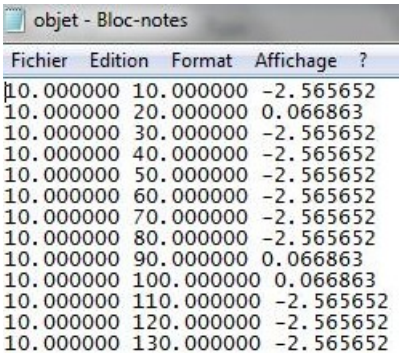


Illustration 8: Liste de points représentant les trois colonnes

- Première colonne : position en x
- Deuxième colonne : position en y
- Troisième colonne : position en z

Nous avons ensuite cherché à modéliser cette liste de points sous la forme d'un nuage de points en 3 dimensions. Pour cela nous avons utilisé la méthode de l'année dernière, c'est-à-dire utiliser le logiciel de modélisation Blender, qui comprend de très nombreuses fonctionnalités liées au traitement d'image en 3D. Mais dans le cadre de notre projet, nous nous en sommes servis uniquement pour la programmation Python.

En effet le groupe de l'année précédente avait utilisé un programme *(voir annexe 9.6)* permettant de modéliser un nuage de points à partir d'une liste de points. Une fois importées, les coordonnées des points sont analysées puis affichées sous forme de nuage de points dans la fenêtre graphique. Cela nous permet d'avoir un premier aperçu de l'image 3D finale *(voir annexe 9.5).*

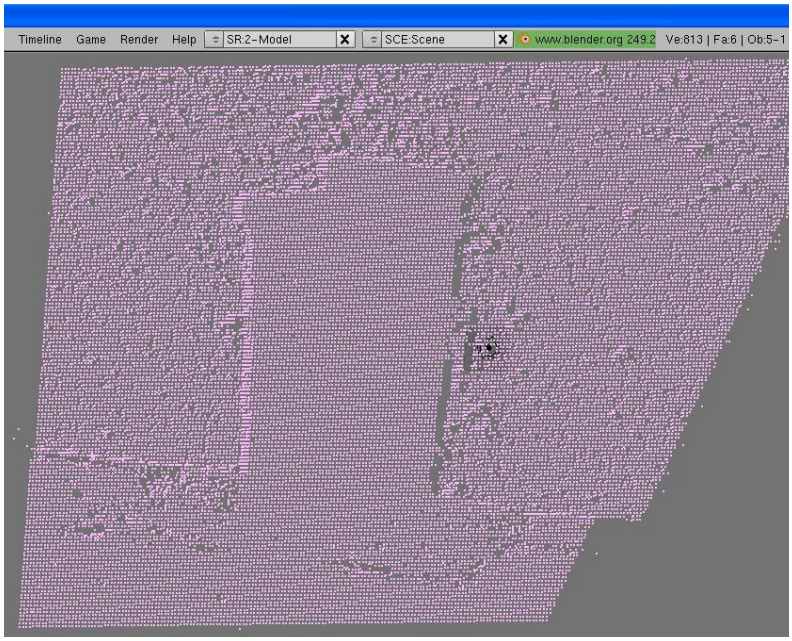


Illustration 9: Nuage de points sous Blender

Une fois le nuage de point de la face modélisé, nous l'avons exporté au format .obj de Blender pour toutes les images traitées de la brique.

5.4. Observation du champ global

Ensuite nous avons cherché à obtenir un rendu 3D de la brique à partir des nuages de points des images traitées auparavant. Pour cela, nous avons d'abord cherché à récupérer un script de Blender qui nous aurait permis de modéliser des faces à partir des points du nuage de points. Mais faute de le trouver, nous avons décidé d'utiliser le même logiciel que l'année précédente, Meshlab.

Meshlab est comme Blender un logiciel de modélisation 3D moins complet mais permettant de modéliser des faces à partir d'un nuage de points grâce à sa version de base, sans avoir à y installer un script.

Il nous a fallu dans un premier temps corriger les nuages de points obtenus sous Blender. En effet, un certain nombre de points parasites, qui n'avaient pas été supprimés lors du filtrage des points aberrants, étaient encore présent dans le nuage de points. Nous avons dû les supprimer « manuellement » afin qu'il ne fausse pas le résultat final.

Ensuite nous avons modélisé les différents nuages de points, afin de reconstituer les faces de la brique. Pour cela, nous avons utilisé la même méthode que l'année précédente, c'est-à-dire appliquer trois commandes du logiciel Meshlab afin de transformer les points en une surface 3D.

La première de ces commandes, « Ball pivoting », consiste à relier les points entre eux pour former des faces plus ou moins basiques en fonction des réglages. Le but étant d'obtenir une première visualisation des faces de la brique.

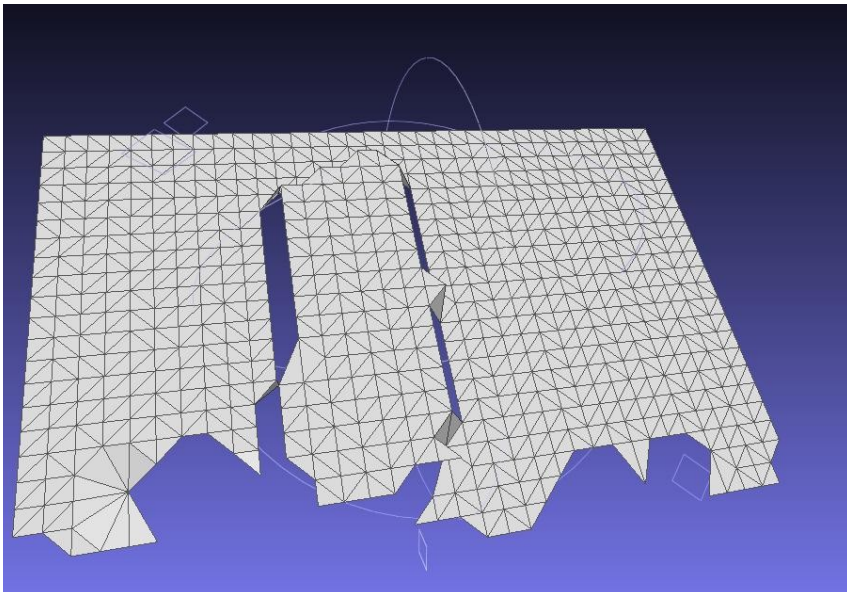


Illustration 10: Image après « Ball pivoting »

La seconde commande, « Laplacian smooth », a surtout servie à lisser le résultat obtenu après « Ball pivoting », afin d'arrondir les angles trop rigides. Il nous a fallu faire attention avec cette commande, car une fois l'effet utilisé, il est impossible de revenir en arrière sur Meshlab, contrairement au « Ball pivoting » qui peut supprimer les faces initiales.

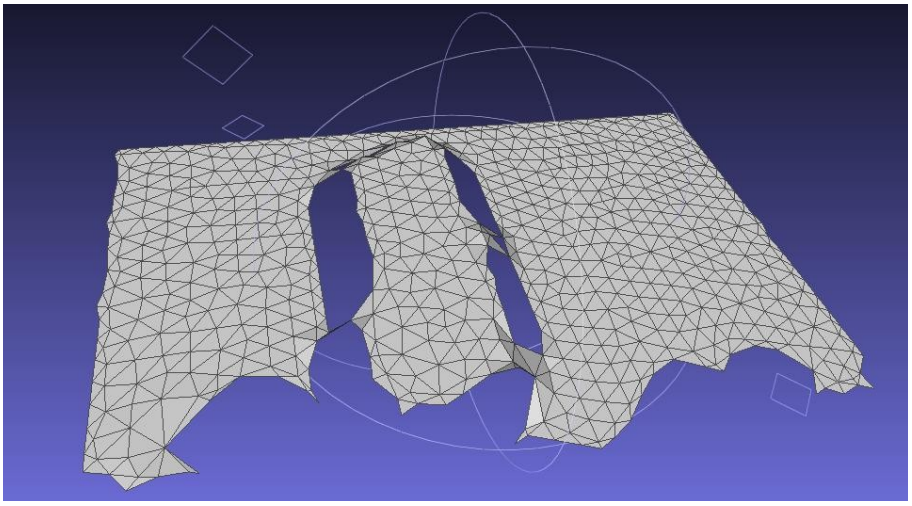


Illustration 11: Image après « Laplacian smooth »

Et si les réglages sont mauvais, on peut se retrouver avec une face qui ne ressemble plus à l'image initialement voulue.

Enfin, la dernière commande que nous avons utilisée est la commande « Poisson ». Elle consiste à créer une nouvelle forme 3D à partir de la précédente, mais en comblant les trous, causés par la suppression des points, et en re-lissant le tout pour obtenir un résultat homogène.

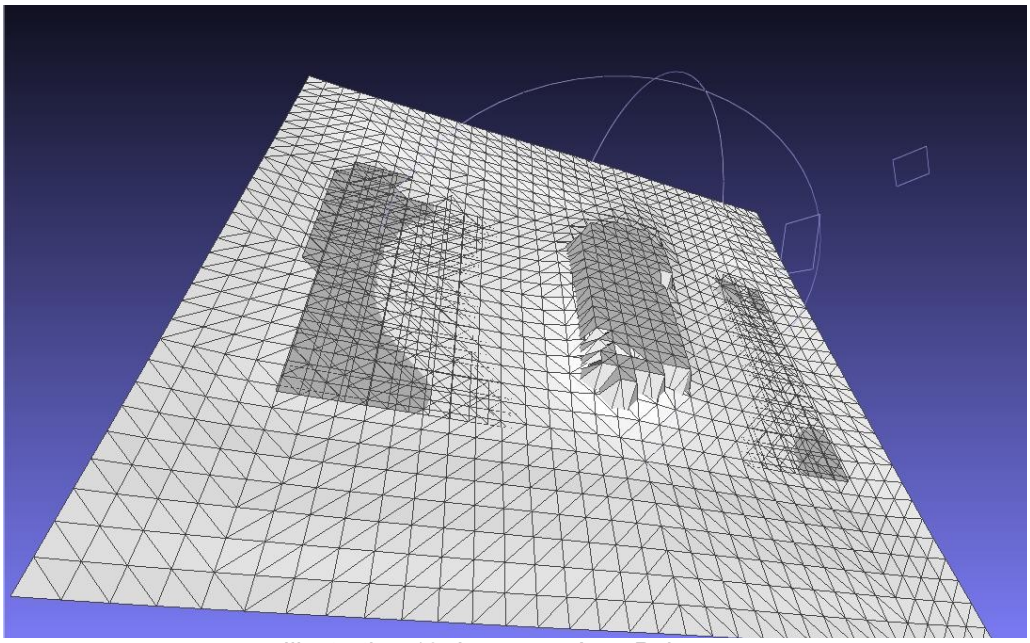


Illustration 12: Image après « Poisson »

Une fois toutes les images modélisées en 3D, il nous a fallu reconstituer les différentes images afin d'obtenir une unique image : la brique en 3D. Pour cela, nous avons décidé d'utiliser une méthode trouvée sur Internet dans un tutoriel réalisé par des élèves de Telecom Lille (lien dans la bibliographie). Il s'agit d'une méthode de recalage des modèles qui permet de recaler plusieurs objets d'un même plan afin de les fusionner et d'en faire un seul objet. Malheureusement, suites aux problèmes rencontrés, nous n'avons pas pu tester cette dernière méthode.

5.5. Compte-rendu de l'expérience

5.5.1. Résultat de l'expérience

L'expérience avait pour but d'obtenir un objet visible en 3 dimensions qu'il était possible de faire tourner. Nous avons pu obtenir en partie cette vu en 3D.

En effet, chaque face prise en photo a pu être modélisée grâce à Blender et Meshlab. Un résultat pour chaque image est visible ci-dessous. L'expérience, le traitement des images puis la mise en 3D fonctionne donc correctement.

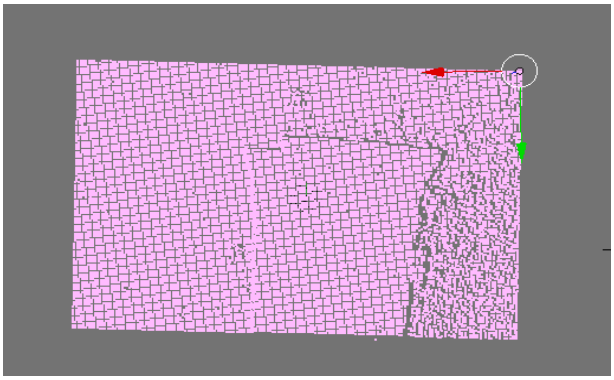


Illustration 13: Image traitée avec Blender

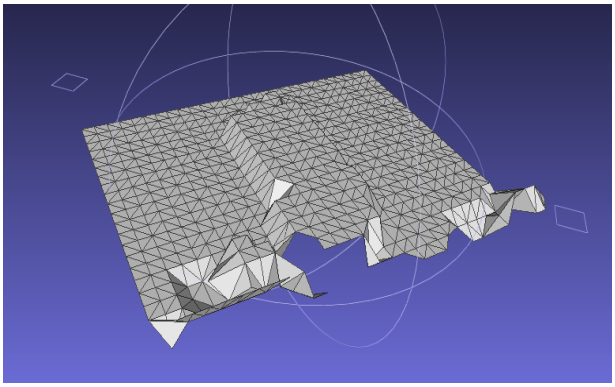


Illustration 14: Image traitée avec Meshlab

De plus, nous avons réussi à faire tourner les images grâce à la matrice de rotation. Le programme fonctionne parfaitement. Les nuages de points ont ainsi pu être replacés à leur place d'origine par rapport à l'objet en général.

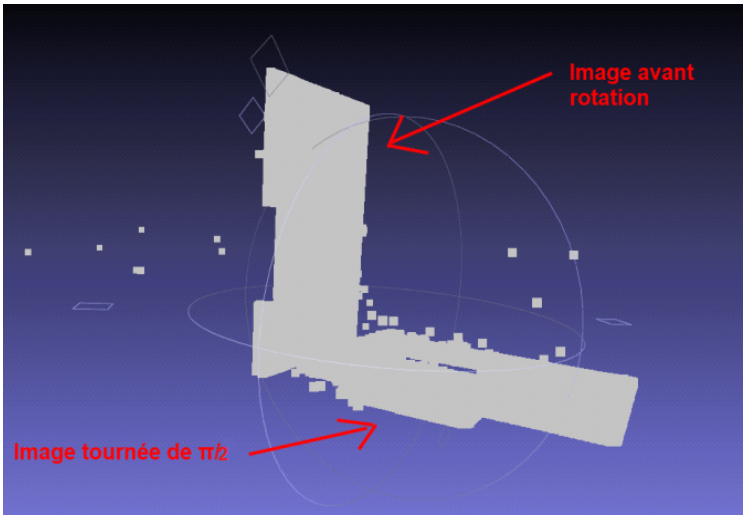


Illustration 15: Images avec et sans rotation sous Meshlab

En revanche, les résultats visuels finaux ne correspondent pas tout à fait aux volumes réels. En outre, les silhouettes obtenues sont bonnes, mais le volume se détachant du mur est plat. Il nous est donc impossible d'assembler les faces pour ne donner qu'un seul objet.

5.5.2. Analyse des résultats

Les raisons pour lesquelles les résultats ne correspondent pas tout à fait à nos attentes peuvent être les suivantes.

Tout d'abord, le projet de l'année dernière avait traité un objet proche du mur, et relativement plat. Contrairement, nous avons tenté notre expérience sur un objet nettement moins plat. Cela a provoqué un décalage du nuage de point projeté, comme souhaité et une déformation de ce dernier. Or, le programme qui traite les images repère les différences d'emplacements entre points sans l'objet et avec l'objet. Il recherche donc les points identiques des deux images. Finalement, les points étant déformés, le programme ne les retrouve pas tous.

Cette déformation des points est aussi due à l'éloignement de l'objet par rapport au mur. En effet, le protocole expérimental demande de faire tourner l'objet. Ce dernier doit donc être à une certaine distance du mur pour pouvoir le faire tourner sans bouger son axe central. Ce point obligatoire peut être arrangé si on décide de travailler sur un objet plus petit, ou du moins ayant une petite largeur et épaisseur.

En conclusion, les résultats sur Blender nous ont permis de constater que l'objet obtenu possédait la bonne silhouette mais pas le volume souhaité. Malgré de nombreuses manipulation sur le logiciel Blender ou avec le programme de traitement d'images, rien n'a pu être amélioré. C'est alors que nous avons constaté que le même problème était apparu l'année précédente. Ce problème n'était pas aussi visible que pour nous, puisque les étudiants ont modélisé un objet presque plat. Le problème viendrait donc soit du nuage de points projeté, soit du programme traitant les photos.

Malgré ces quelques problèmes notamment sur le choix de l'objet, le traitement d'images du côté informatique (OpenCV, Scilab, Blender, Meshlab) se réalise très bien. Les résultats obtenus visuellement grâce à Scilab ne sont pas déformés par la suite. De plus, les images finales correspondent, au niveau de la silhouette, aux photos initialement prises.

Les résultats nous permettent donc de conclure qu'une reproduction en 3D par cette méthode est possible, mais il faut peut-être re-penser quelques aspects un peu différemment (l'objet choisi, la distance au mur..).

5.5.3. Les erreurs commises

Effectivement, tout au long de notre projet, nous avons commis des erreurs que ce soit pour la partie expérimentale que la partie programmation. Nous allons alors décrire brièvement ces erreurs afin que les étudiants des années suivantes ne perdent pas de temps.

5.5.3.1 Erreurs expérimentales

Tout d'abord, avant d'arriver à des résultats convaincants, nous avons réalisé trois expériences différentes. En effet, à chaque fois, une erreur de manipulation faussait les traitements des images.

Pour la première expérience, nous nous étions concertés avant la séance pour définir l'objet à modéliser et les difficultés que nous pouvions rencontrer. Nous avions pensé à des objets originaux tels des lunettes de soleil, une chaussure.. Cependant, le jour de la séance de projet, le professeur nous a rappelé que l'objet devait être clair, pas en toile et ne pas refléter. De plus, il nous a fortement conseillé de commencer par un objet facile. Alors, nous avons opté pour une boîte en carton rectangulaire.

Néanmoins, bien qu'ayant préparé la séance, nous avons rencontré des problèmes pour maintenir l'objet. Nous l'avons donc collé au bout d'un bâton de bois pour le faire tourner et ne pas voir la personne qui le tenait sur le cliché. Cette méthode n'était évidement pas précise.

D'autre part, nous avons tracé des angles au sol avec une ficelle : après avoir fait un cercle, nous avons appliqué la règle des médianes pour l'angle π/4. Pour l'angle π/6 et π/3, nous savons que les sinus et cosinus (respectivement) valent ½. Bref, ce tracé manquait aussi de précision.

Ces erreurs nous ont permis d'améliorer l'expérience. En outre, pour la deuxième expérience, nous avions apporté un trépied de caméra pour y placer l'objet. Ce dernier était d'ailleurs plus complexe : nous avions accroché un cylindre sur une face. Puis, nous avions tracé par les mêmes méthodes les différents angles. La rotation de l'objet était donc plus facile.

Cependant, cette expérience n'a pas été concluante puisque la luminosité avait changé entre les photographies. D'autre part, l'appareil photo avait bougé (même si c'est de quelques centimètres même millimètres seulement) cela a eu des répercussions sur les traitements de l'image. Mais l'erreur principale reste l'oubli de la prise d'une photographie sans objet avant de retirer tout le matériel. Bien que nous ayons essayé de remettre tous les éléments à la même place, la photographie de référence était totalement décalée et le traitement des images inutiles.

La troisième et dernière expérience semble donc être la bonne. Nous avons utilisé une boîte de jus de fruit blanche, fixée sur un carton avec un bâton qui modélise l'axe de rotation. Une feuille avec les différents angles est placée en dessous de l'objet. *(voir 9.7 en annexe le protocole expérimental).* Malgré les précautions, le rétroprojecteur a bougé de un ou deux millimètres provoquant l'apparition de deux plans différents lors du traitement de l'image. Finalement, nous avons corrélé les images et, bien que quelques modifications ont dues être effectuées, nous obtenons un résultat, même si le relief est peu visible.

5.5.3.2 Erreurs de programmation

D'autre part, quelques erreurs ont été commises lors de la programmation.

Premièrement, nous avions lancé plusieurs traitements à la fois, ce qui empêche l'enregistrement des traitements. En effet, lePattern_Matching_2 donne un fichier résultat qui n'est renommé qu'ensuite. Si plusieurs traitements sont lancés simultanément, les résultats sont écrasés à chaque fois.

De même, le lancement d'une corrélation avec des images non redimensionnées provoque un traitement très long (12heures pour seulement 7% du traitement). Il faut donc trouver le bon équilibre entre qualité de la corrélation et temps de calcul.

Bref, pour plus de rapidité, il faut faire attention à ne pas renouveler ces erreurs.

6. CONCLUSIONS ET PERSPECTIVES

6.1. Conclusions sur le travail réalisé

Notre projet a été très intéressant puisqu'il nous a permis, de continuer et d'améliorer un projet déjà commencé. Néanmoins, le résultat n'est pas vraiment satisfaisant puisqu'on ne remarque pas le relief et nous n'avons pas pu rassembler toutes les photos ensembles. Alors, nous aurions pu reconstituer en 3D l'objet presque en entier (sans la face de dessous et de dessus).

Malgré cela, nous avons tout de même réussi à rendre portable le projet, c'est à dire que les programmes, et notamment celui pour corréler les images, peuvent-être utilisés sur n'importe quel ordinateur. Ainsi, la partie traitement des images est séparée entre les membres du groupe, et cela est plus rapide.

Mais, le fait de reprendre des projets déjà existants n'est pas si facile. Effectivement , il faut tout d'abord comprendre les programmes et les équations mathématiques antérieures pour pouvoir les appliquer à notre méthode. De plus, pour créer une nouvelle expérience, il faut être très méticuleux et réfléchir à tous les aspects pratiques et théoriques, sans se rattacher aux expériences des années précédentes.

6.2. Conclusions sur l'apport personnel

D'un point de vue apport personnel, ce projet nous a permis de nous organiser. En outre, nous avons appris à travailler avec des personnes que nous ne connaissions pas, dans un temps assez court. Alors, rapidement, nous nous sommes répartis les tâches de travail pour pouvoir avancer le plus vite et efficacement possible.

Par ailleurs, le projet a été enrichissant puisque nous avons appris à nous servir de nouveaux logiciels (OpenCV, Blender..). De plus, nous avons mis concrètement en application des notions vues en cours (par exemple en mathématique, nous apprenons à nous servir des matrices de rotations/translations...).

6.3. Perspectives

Pour la poursuite de ce projet, il serait essentiel de penser à la manière de faire tourner l'objet en le rapprochant au maximum du mur. Ainsi, la corrélation fonctionnerait mieux et les images, après avoir été replacées dans le même plan, pourront être rassemblées. On obtiendrait donc un objet final, presque réel.

En effet, les faces de dessous et de dessus ne peuvent pas être vues puisque l'objet est posé sur la table et la caméra ne bouge pas. Il serait donc intéressant d'approfondir ce point, afin de reconstituer toutes les faces de l'objet.

Finalement, peut-être que prendre plusieurs photos autour de l'objet fixe serait plus facile. Alors, il suffirait de reprendre les méthodes des deux derniers projets, mais au lieu de prendre qu'un seul cliché avec l'objet, en prendre plusieurs en changeant d'angle θ .

Si cette technique est plus concluante, il deviendrait donc très intéressant de faire varier l'angle de la caméra φ . Le rendu serait encore plus réaliste.

7. BIBLIOGRAPHIE

Échographie :http://procubeteam.online.fr/LaurentM/echographie3D/ (valide à la date du06/06/2011)
http://fr.wikipedia.org/wiki/Échographie (valide à la date du 06/06/2011)

Stéréoscopie :http://fr.wikipedia.org/wiki/Stéréoscopie (valide à la date du 06/06/2011)

Tomographie: http://www.univ-tlemcen.dz/manifest/CISTEMA2003/cistema2003/articles/Articles%20-%20GBM/GBM18.pdf (valide à la date du 07/06/2011)

http://www.cea-technologies.com/articles/article/563/fr (valide à la date du 07/06/2011)

http://www.techniques-ingenieur.fr/base-documentaire/mesures-analyses-th1/techniques-d-analyse-par-imagerie-42387210/tomographie-a-rayons-x-p950/ (valide à la date du 07/06/2011)

http://www.azonano.com/news.aspx?newsID=18441&lang=fr(valide à la date du 07/06/2011)

Blender: http://jmsoler.free.fr/didacticiel/blender/tutor/indexpython.htm (valide à la date du 13/06/2011)

Meshlab : http://www.telecom-lille1.eu/people/benamor/TP_MeshLab.pdf (valide à la date du 13/06/2011).

OpenCVWiki : http://opencv.willowgarage.com/wiki/ (valide à la date du 07/06/2011)

SourceForge : http://sourceforge.net/projects/opencvlibrary/ (valide à la date du 07/06/2011)

TopDLL : http://fr.topdll.com/ (valide à la date du 07/06/2011)

8. INDEX DES ILLUSTRATIONS

9. ANNEXES

9.1. Protocole expérimental et schéma de l'expérience

- **Matériel** :

Voici une liste des outils que l'on a utilisé pour prendre des photos :

- Objet (boîte blanche)
- carton pour poser l'objet
- feuille quadrillée pour tracer les angles
- bâton au centre de l'objet modélisant l'axe
- appareil photo
- vidéo projecteur
- ordinateur
- mètre
- feuille blanche à mettre sur le mur
- crayon à papier

- **Déroulement de l'expérience** :

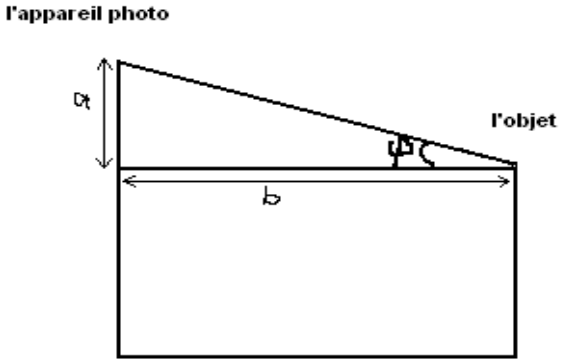
1) Allumer l'ordinateur et le rétroprojecteur.
2) Mettre la feuille blanche sur la mur en face du rétroprojecteur.
3) Placer l'appareil photo à un ou deux mètre du rétroprojecteur.
4) Fixer l'objet sur le carton grâce au bâton modélisant son axe.
5) Mettre une feuille en dessous de l'objet pour tracer les angles : 0, $\Pi/6$, $\Pi/4$, $\Pi/3$, $\Pi/2$, $2\Pi/3$, $3\Pi/4$ … jusqu'à $11\Pi/12$.
6) La table et l'objet doivent être le plus proche possible du mur. Faire un repère là où le carton est placé sur la table.
7) Mesurer les distances :
   - rétroprojecteur-appareil photo
   - rétroprojecteur-mur
   - rétroprojecteur-objet
   - mur-objet
   - hauteur de l'appareil photo
8) Mesurer l'angle φ : inclinaison de l'appareil photo : on mesure d'abord la hauteur de l'objet , la distance entre l'objet et l'appareil photo , et on utilise la hauteur de l'appareil photo mesurée avant .

Ensuite nous calculons la différence entre ces deux hauteurs :

a=(la hauteur d'appareil photo)-(la hauteur d'objet)

b=la distance entre l'objet et l'appareil photo .

Puis on cherche l'angle φ avec la fonction φ=arctan(a/b) .



9) Prendre toutes les photos avec les différentes rotations de l'objet. Attention, la luminosité dans la salle doit être la même, et **rien ne doit bouger (appareil photo, rétroprojecteur, objet..)**

10) Repérer chaque numéro de photos avec l'angle de rotation correspondant.

11) **Une fois toutes les photos terminées, prendre une photo sans objet !**

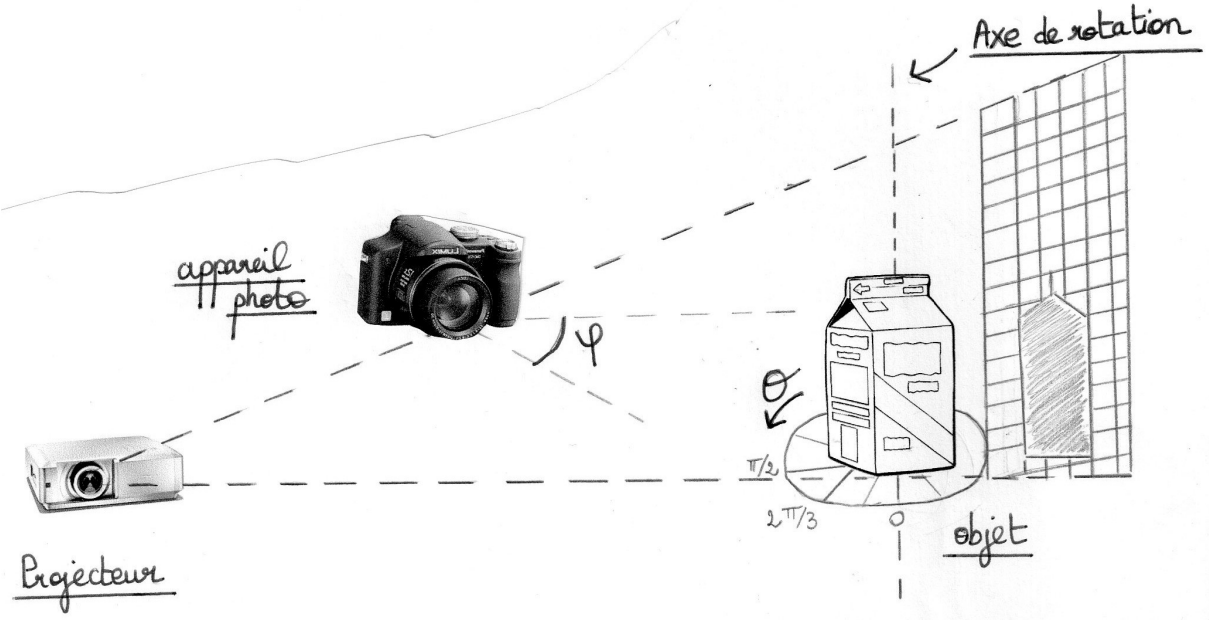


Schéma de l'expérience

9.2. Listings des programmes réalisés

ScriptPhoto.bat : fichier de commande pour simplifier le traitement.

Calculprofondeur_2.sce : fichier scilab pour filtrer les points

Traitement.zip : Archive contenant les 2 fichiers précedents et les bibliothèques C++ nécessaires

P6-3.zip : Archive contenant Traitement.zip, les dossiers, les programmes et les photographies des 3 années de projet sur le reconstitution 3D.

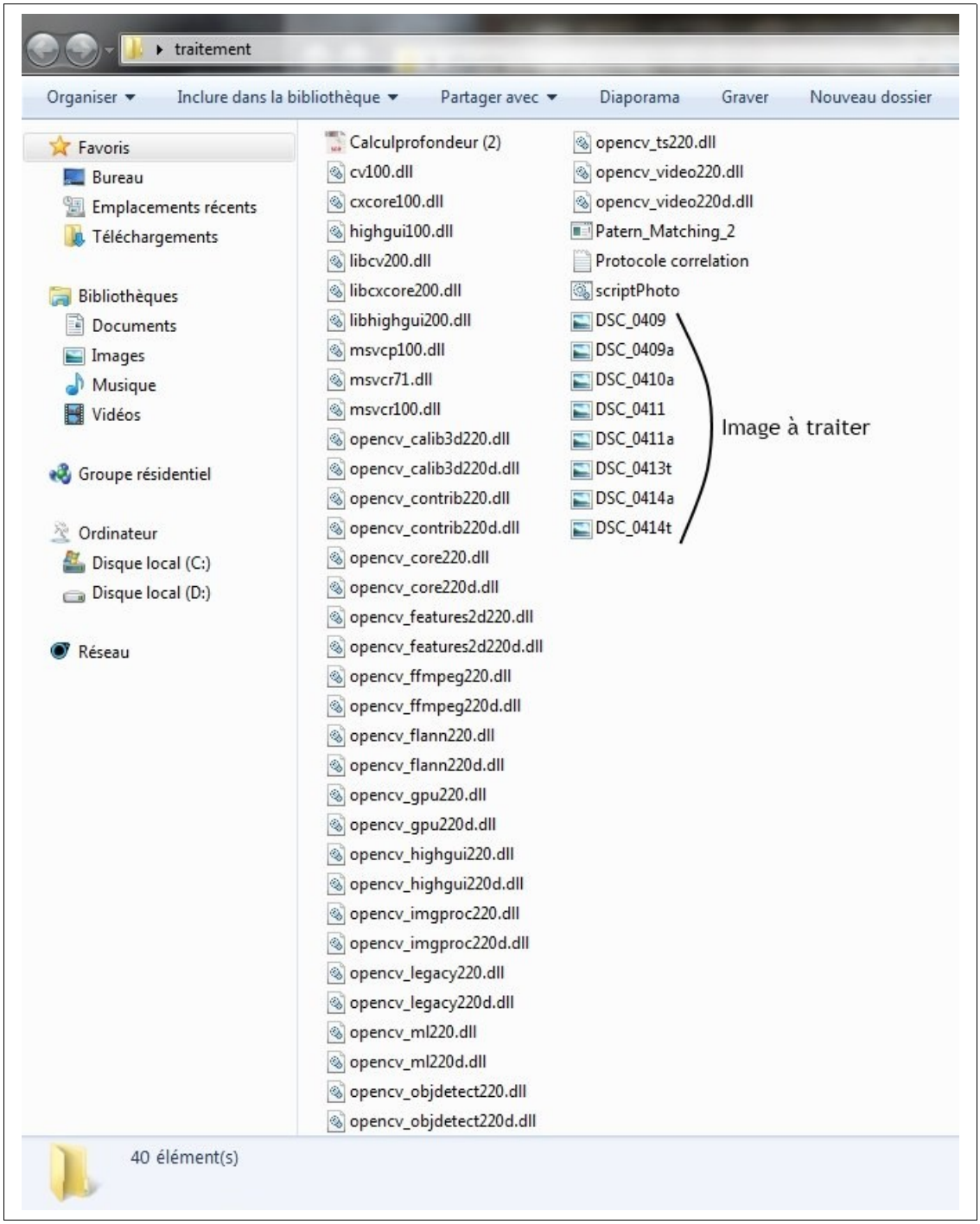
9.3. Programme SCILAB

Ce programme permet d'appliquer aux points une matrice translation, puis une matrice rotation-inverse et enfin une la matrice translation inverse de la première matice.

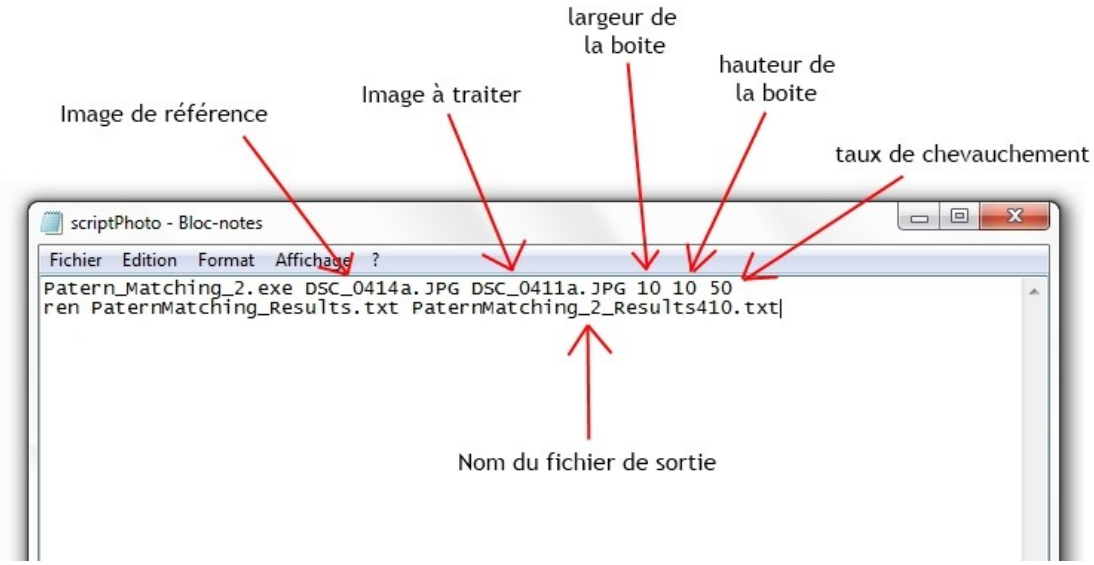
```
NomFichier='PaternMatching_Results-filtre-test411.txt'; // import du fichier avec les points
NomFichierSortie='PaternMatching_Results-filtre411-rotate.txt'; // enregistrement du nouveau nuage de points avec les trois matrices appliquées
d=7,3; //distance plan-objet
DATA=fscanfMat(NomFichier);

DATA(:,3)=DATA(:,3)-d; // première matrice translation
teta=%pi/6; // angle de rotation
invM=[cos(teta) 0 sin(teta);0 1 0;-sin(teta) 0 cos(teta)]; // matrice rotation inverse

[imax,b]=size(DATA);
for i=1:imax
  V=DATA(i,:)';
  newV=invM*V;
    DATA(i,:)=newV'; // application de la matrice rotation à l'ensemble des points par incrémentation
end
fprintfMat(NomFichierSortie,DATA);
```

9.4. Tutoriel de traitement des images

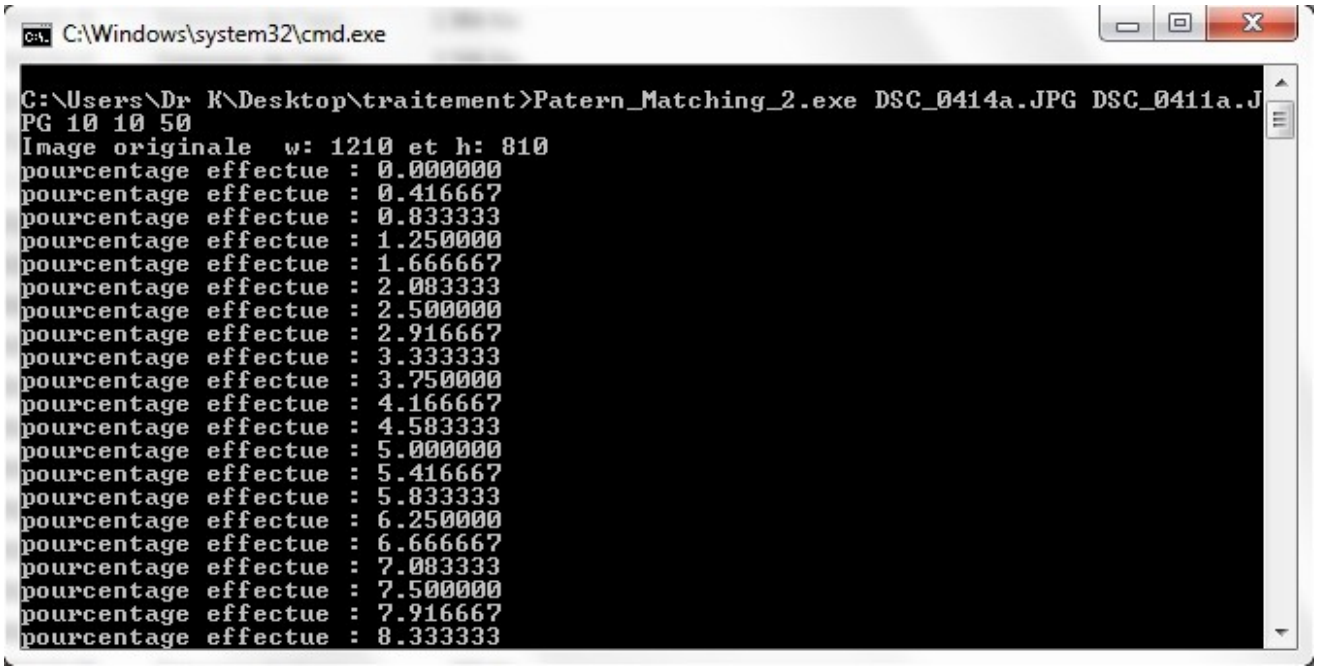
1. Mettre les photos à traiter dans le dossier traitement contenant le programme Patern_Matching_2, les bibliothèques nécessaires à l'exécution de ce programme, le script de lancement, ainsi que le fichier de commande SciLab permettant de filtrer les points.
   Attention ! Il est peut-être nécessaire de redimensionner les photos. Plus l'image à une résolution élevée, plus le temps du traitement est long.



2. Il faut désormais modifier le script (scriptPhoto) afin de préciser quelle image est a analyser, quelle est l'image d'origine, les paramètres du traitement, ainsi que le nom du fichier à retourner. Une fois modifié, enregistrer le script.



3. Lancer le script pour effectuer le traitement de l'image. Le traitement peut prendre plusieurs heures. (Si c'est trop long, redimensionner l'image ou changer la taille des blocs).
<u>Répéter les opérations 2 et 3 pour toutes les images à traiter.</u>

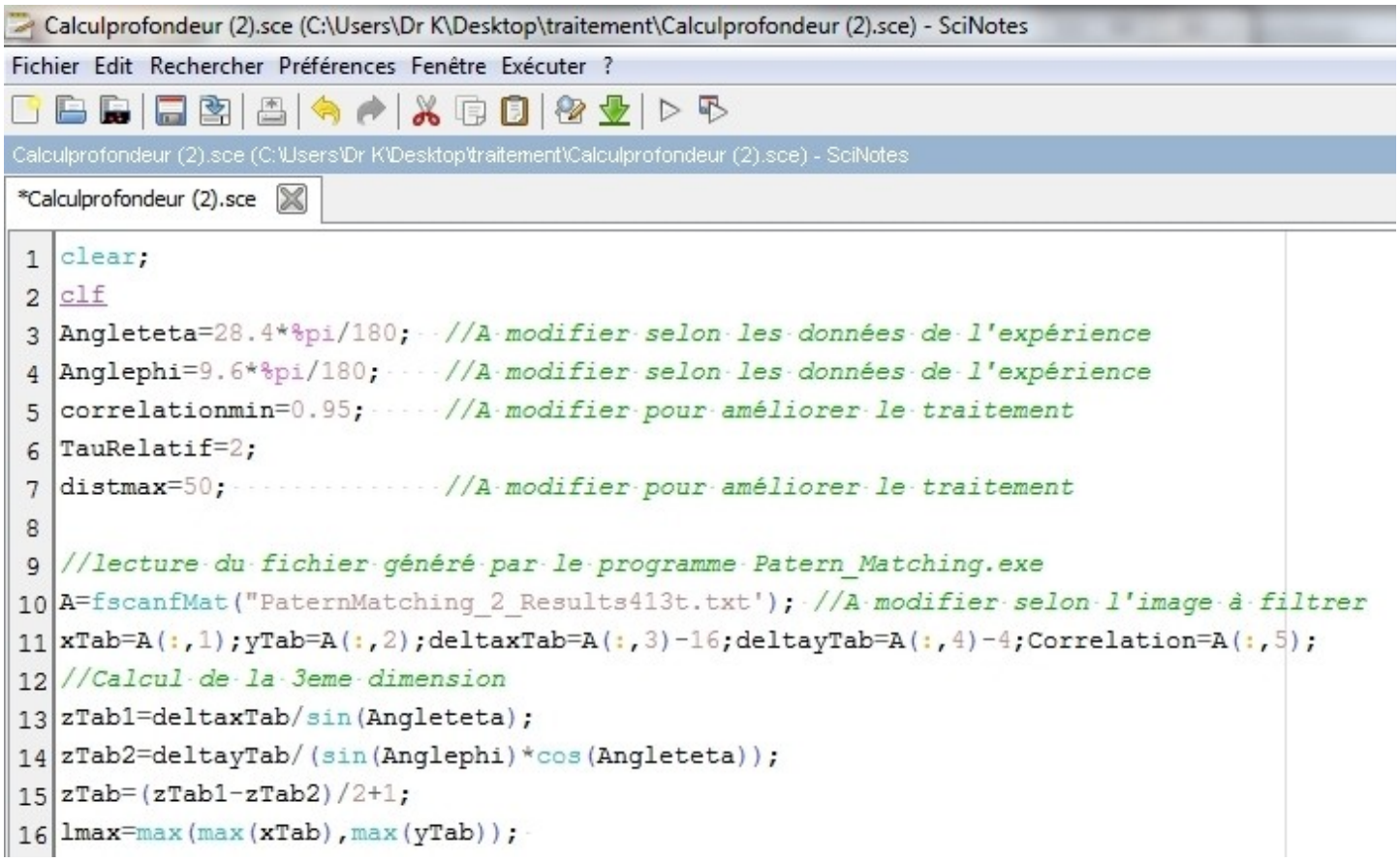


4. Vous avez désormais des fichiers textes représentant le traitement de chaque image dans le dossier. Il faut maintenant utiliser le programme SciLab pour filtrer les points. Ouvrir le fichier Calculprofondeur(2) dans Scilab.

   Régler les paramètres selon les mesures de l'expérience :

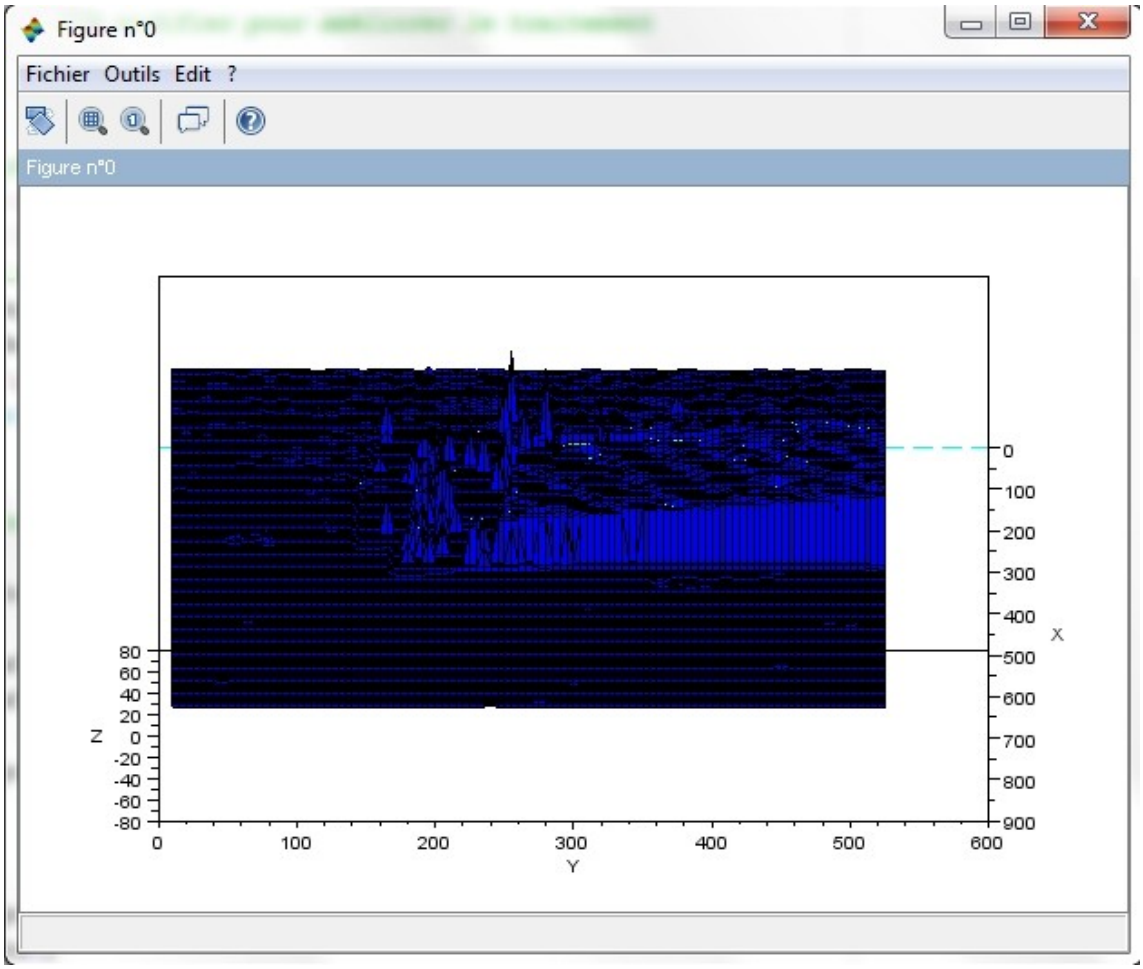
- Angleteta: correspond à l'angle entre le projecteur et l'appareil photo horizontalement.
- Anglephi: correspond à l'angle entre le projecteur et l'appareil photo verticalement.

Choisir le fichier texte à filtrer (ligne 10), et modifier le nom du fichier texte résultat (ligne 62).



5. Exécuter le programme. Vous obtenez une image test. Vous pouvez désormais essayer de modifier les paramètres correlationmin et distancemax afin d'obtenir des résultats plus précis.

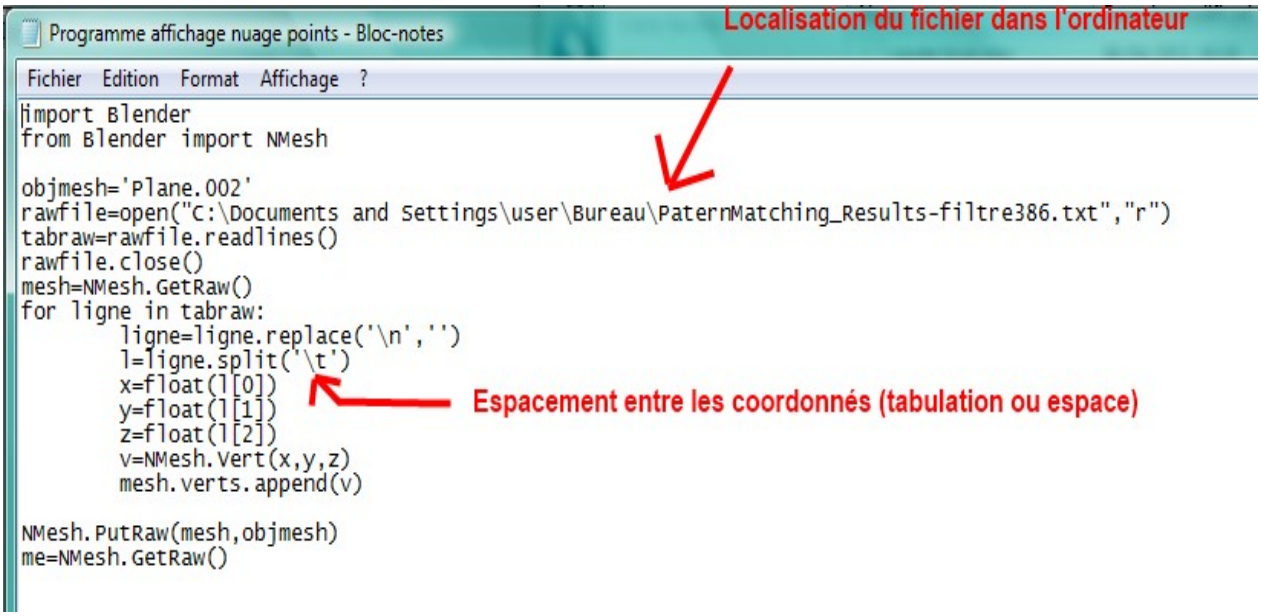
- Plus correlationmin est proche de 1, plus le filtrage sera précis (mais moins de point au final)
- Plus distancemax est petite, plus le filtrage sera précis (attention, prendre une distance trop petite peut-faire disparaître l'objet).
  Répéter les opérations 4 et 5 pour toutes les fichier à filtrer.



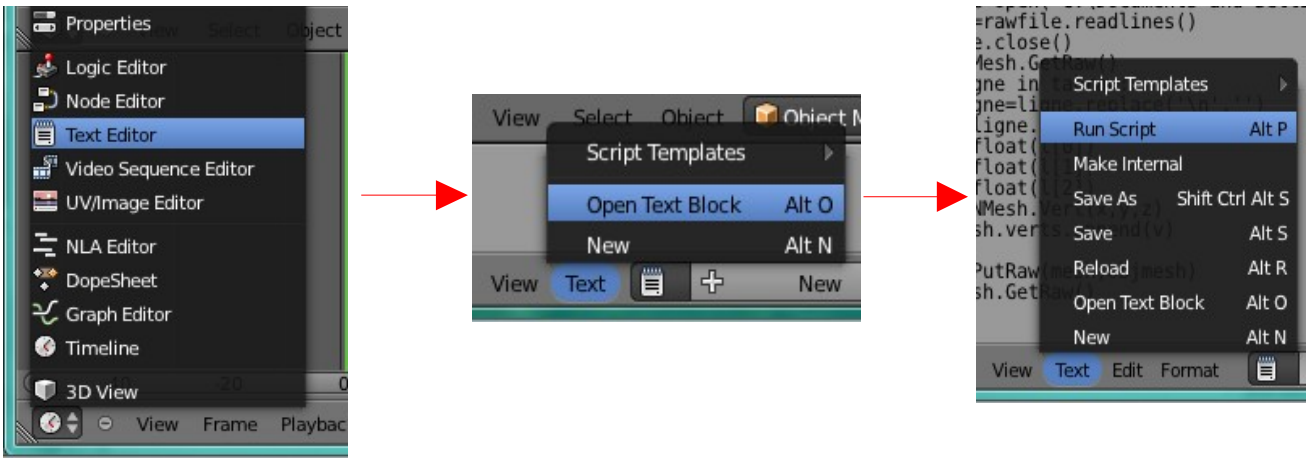
Vous avez désormais des fichiers textes représentant le traitement d'image filtré.

9.5. Traitement de la 3D sur Blender et Meshlab

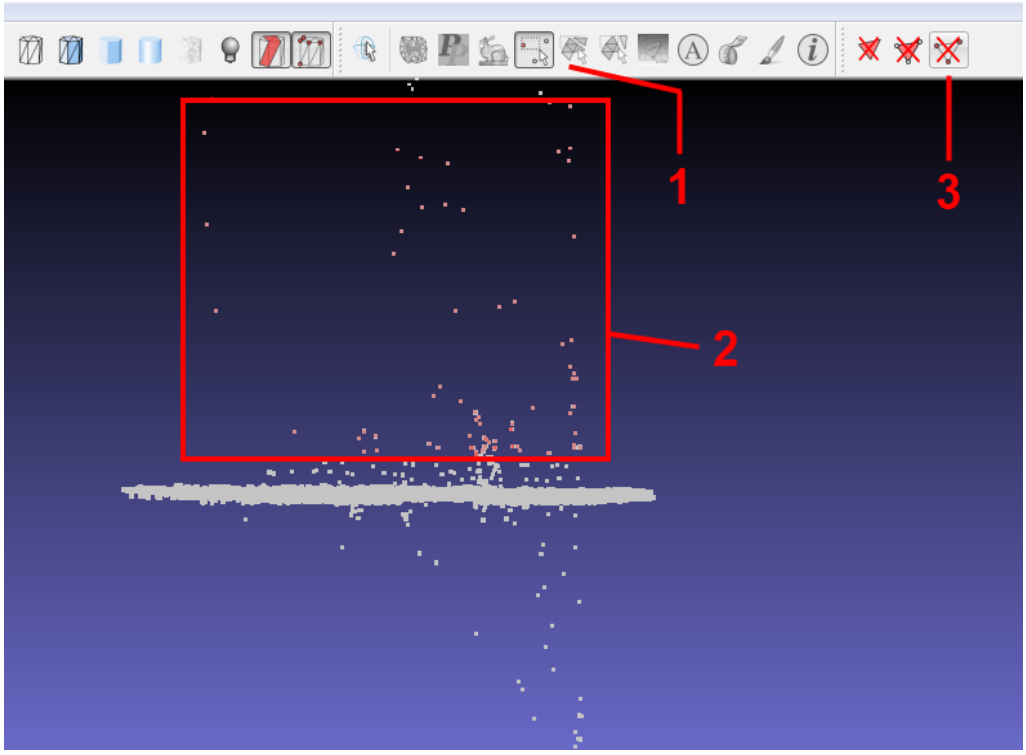
1. Ouvrir le fichier « Programme affichage nuage de points » Changer l'emplacement et le nom du fichier a traiter. Si il y a un espace entre chaque coordonnée, remplacer « \t » (tabulation) par « » (espace).



2. Ouvrir Blender, afficher la fenêtre « Text Editor ». Importer le fichier « Programme affichage nuage de points », puis compiler le programme grâce a la fonction Run Script (ou run python scipt en fonction de la version de Blender)



3. Exporter le fichier au format .OBJ (Wavefront).
4. Ouvrir le fichier format .obj dans Meshlab. Il est possible de supprimer les points incohérents qui n'ont pas été supprimés par le programme sous Scilab manuellement.



5. Traiter les point pour obtenir des faces grâce à Filters > Remeshing, Simplification & Reconstruction > Ball Pivoting en réglant les paramètres en fonction du résultat voulu. Faire ensuite un lissage grâce à Filters > Smoothing, Fairing & Deformation > Laplacian Smooth puis utiliser Filters > Remeshing, Simplification & Reconstruction > Poisson. Il est possible de tenter d'autres traitements pour améliorer l'image finale.

6. Les différentes images peuvent être assemblées pour en faire une seule. Charger les images à assembler dans la même fenêtre.

7. Démarrer l'outil d'alignement (recalage). Choisir un des deux modèles et fixer le à sa place à l'aide du bouton « Glue Mesh Here » (une étoile apparaît devant son nom)

8. Choisir le modèle restant et cliquer sur le bouton Point based alignment. Une nouvelle fenêtre apparaît contenant les deux modèles. Choisir au moins quatre points correspondants sur les deux modèles par double clic sur les endroits souhaités. Cliquez sur « Start process ».

9.6. Programme d'importation des points sous Blender

```
import Blender
from Blender import NMesh

objmesh='Plane.002'
rawfile=open("C:\Documents and Settings\user\Bureau\PaternMatching_Results-filtre386.txt","r")
tabraw=rawfile.readlines()
rawfile.close()
mesh=NMesh.GetRaw()
for ligne in tabraw:
        ligne=ligne.replace('\n','')
        l=ligne.split('\t')
        x=float(l[0])
        y=float(l[1])
        z=float(l[2])
        v=NMesh.Vert(x,y,z)
        mesh.verts.append(v)

NMesh.PutRaw(mesh,objmesh)
me=NMesh.GetRaw()
```

9.7. Photographie du groupe

De gauche à droite : Ya, Ken, Jérémy, Fanny, Léo, Clotilde, Élodie et Blandine