Projet POO – 2018_S1

QUALITES ATTENDUES DU PROJET POO

Fiabilité : Il doit donner les résultats corrects attendus : Initialiser les variables même si Python le fait dans certains cas, ce qui évite certaines surprises et des débogages laborieux…

Robustesse : Il doit gérer les erreurs évidentes de manipulation des utilisateurs.

Convivialité : Il doit être agréable à utiliser (souris, icônes, menus…), et facile à prendre en main (le scenario d'utilisation semble « logique » à l'utilisateur).

Efficacité : Il doit donner des réponses rapides et claires.

Compacité : Il doit occuper une place modérée en mémoire et sur le disque !

Lisibilité : Il doit être structuré en classes, fonctions et procédures toutes commentées et présentées clairement. Vos variables doivent être explicites. N'hésitez pas à développer vos propres classes pour plus de lisibilité !

Portabilité : Il doit être aisément transférable sur une machine d'un autre type (attention aux widgets ou aux librairies spécifiques, aux chemins d'accès codés spécifiquement pour votre machine…).

Flexibilité : Une partie de votre travail doit pouvoir être aisément utilisable par d'autres applications. Pour cela nous vous préconisons de dissocier fortement vos interfaces de vos traitements et de généraliser vos fonctions et procédures.

Et Enfin : **Privilégiez le bon fonctionnement du programme à son aspect !** Le résultat n'est pas le seul point évalué : votre autonomie, votre travail personnel, l'originalité et la bonne exécution de vos solutions, le respect des consignes… sont d'autant d'éléments qui participent à votre évaluation !

SCANNER 3D PAR TRIANGULATION

☆ Interface
☆☆ Algorithmes de traitement
☆☆ Concepts / Outils non vus en cours

Présentation du problème et objectifs

On souhaite concevoir et implémenter un outil permettant de reconstruire un modèle 3D d'un objet existant à des fins de contrôle dimensionnel sans contact… Dans le cadre de ce projet POO, c'est la stratégie par triangulation qui est retenue parmi les nombreuses solutions disponibles, comme par exemple les approches : par profil, par lumière structurée (solution utilisée par la tête Atos de l'entreprise GOM disponible sur le « petit » robot dans la halle de déformations plastiques), par immersion dans un fluide (http://milkscanner.moviesandbox.net/), stéréoscopiques...

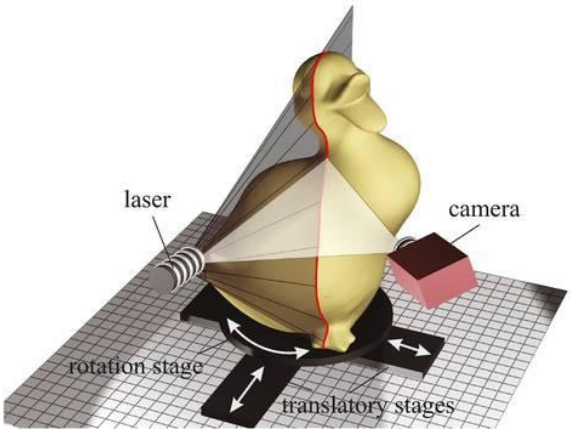

Figure 1 : Illustration du principe de fonctionnement de la triangulation

Le principe de fonctionnement de cette approche est illustré par la Figure 1, ainsi que par la vidéo Youtube suivante : https://www.youtube.com/watch?v=RVgyyIlQydg. Cette approche est intéressante car elle peut, à faible coût, être mise en œuvre, mais à des niveaux de précisions relativement faibles.

Prise de mesures : Un objet, posé sur une table rotative indexée, est frappé par un laser ligne fixe. Une caméra, fixe également, enregistre l'image (ou la vidéo) ainsi reçue de l'objet éclairé par la ligne laser. Il est ainsi possible de retrouver le profil de cette intersection laser/objet. La table est alors mise en rotation et plusieurs photographies (ou un film) sont alors effectuées et enregistrées.

C'est l'étape de traitement de ces photos (ou de photos extraites du film) qui aboutit à la définition numérique 3D du volume de l'objet ainsi numérisé. Cette étape de traitement est le but de ce projet de l'UEF POO.

Reconstruction du modèle 3D : Les différentes photos de l'objet doivent être analysées pour identifier des points de la surface de l'objet illuminé par le laser. Une fois ces points identifiés, il est nécessaire, en considérant l'angle de la table, la position et l'orientation de la caméra et de ses caractéristiques, d'effectuer un changement de repère et une projection pour passer du repère caméra à celui de l'objet. Une fois ces opérations effectuées, une « tranche » de l'objet est

numérisée. La dernière étape consiste à fusionner ce morceau avec l'ensemble des autres tranches provenant des autres vues de ce même objet.

Cas test : Pour tester le bon fonctionnement de votre outil, plusieurs cas d'étude sont proposés. Ils ont été réalisés au sein du laboratoire LCFC à l'aide de ses ressources et de ses expertises (Merci Alexandre et Daniel ☺). Vous trouverez ainsi, en tant que ressources sur le portail informatique Info/Math (pour rappel : http://sps.ensam.eu/sites/TRA-COL/InfoMathV2/) plusieurs fichiers : certains contenant deux flux photographiques à exploiter (un pour chaque appareil) et plusieurs vidéos (deux pour chaque pièce à traiter). Le montage utilisé est illustré par la Figure 2. L'ensemble des paramètres considérés importants pour la reconstruction (angles, distances, focales et positions…) y est identifié et illustré. Les valeurs précises pour chaque expérience sont précisées sur le site dans leur répertoire.

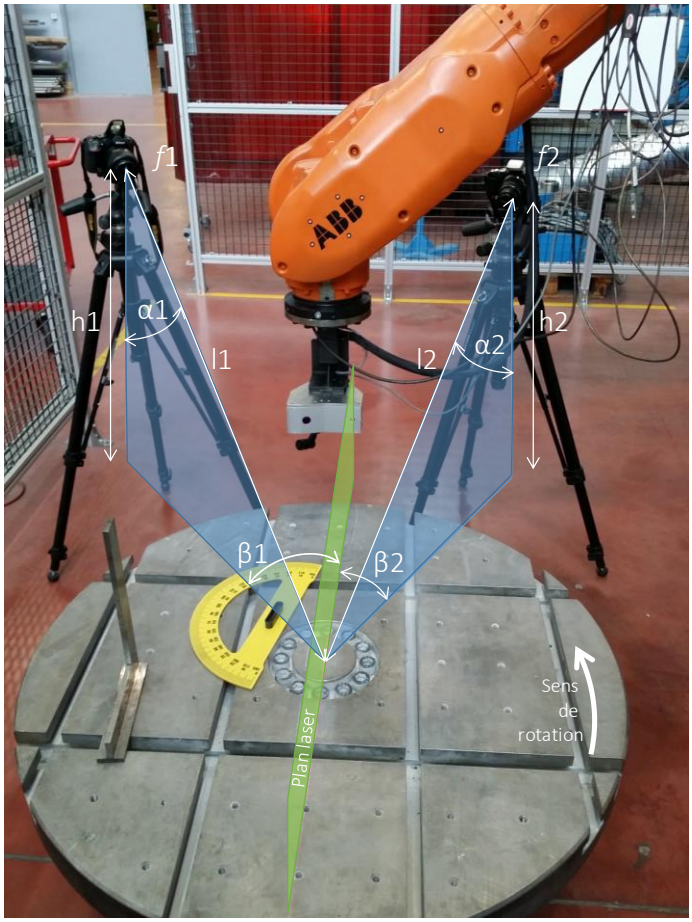

Figure 2 : Montage expérimental et ses identification des paramètres clefs

La suite de cet énoncé propose une décomposition du travail à réaliser en différentes étapes. Certaines étapes sont indépendantes. Les mots écrits en gras dans cet énoncé doivent apparaitre dans votre programme sous la forme d'une classe. Vous pouvez ajouter d'autres classes si vous en avez besoin ou si vous les jugez nécessaires.

La première classe à implémenter, la classe **pièce**, est celle qui contient toutes les informations liées à une expérience (les caractéristiques des points de vue, les liens vers les photos ou les vidéos, les caractéristiques d'extraction des photo…). Elle sera enrichie au fur et à mesure de l'avancement dans ce projet (ajout des lignes, enregistrement des données calculées…)

Etape 0 : Extraction des images à partir de vidéos

Pour la majorité des pièces, et afin de permettre d'augmenter la résolution de la numérisation (le nombre de tranches 2D du volume à recréer et ainsi la qualité de la surface reconstruite), les expériences n'ont pas été photographiées, mais filmées. La première étape, revient à réaliser une procédure permettant d'extraire un nombre d'images désiré par l'utilisateur depuis les vidéos fournies. Vous pouvez pour cela utiliser les librairies Python bien connues telle que OpenCV par exemple.

Au cas où vous n'arriveriez pas à réaliser cette étape :

- deux lots (un lot par appareil photo utilisé) de photos vous sont proposés pour un seul cas d'étude.
- il est également possible de passer par un logiciel tiers pour effectuer cette extraction. Plusieurs logiciels gratuits (VLC, VirtualDub…) sont capables de générer des images Jpeg à partir d'un flux vidéo.

Etape 1 : Traitement des images

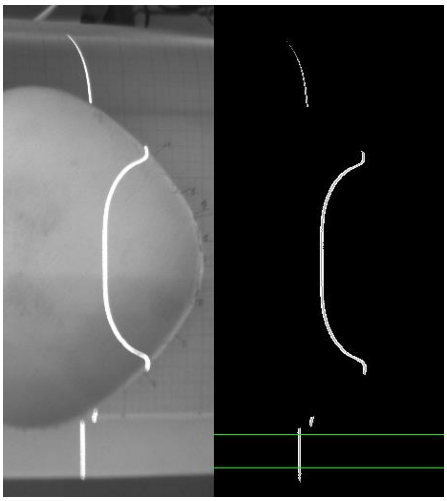

Figure 3 : Traitement d'une image : Isolement de la ligne formée par le laser

L'objectif de cette première étape est d'isoler la **ligne** formée par le laser afin d'obtenir la liste des points qui la compose, dont les coordonnées sont exprimées dans le repère de l'image, en pixel.

Pour ce faire, il est nécessaire de considérer comme blanc, tous les pixels dont la luminosité est maximale et comme noir les autres pixels que l'on souhaite filtrer. Afin de déterminer cette valeur seuil, il est nécessaire d'analyser l'image complète et la luminosité de chaque pixel.

Votre fonction doit alors retourner un objet **ligne**, composé de plusieurs points, du nom de l'image analysée, voire également de la position angulaire de la table lors de la prise de la photo. On peut également améliorer la détection de la position de ces points en utilisant la méthode dite du Sub-Pixel.

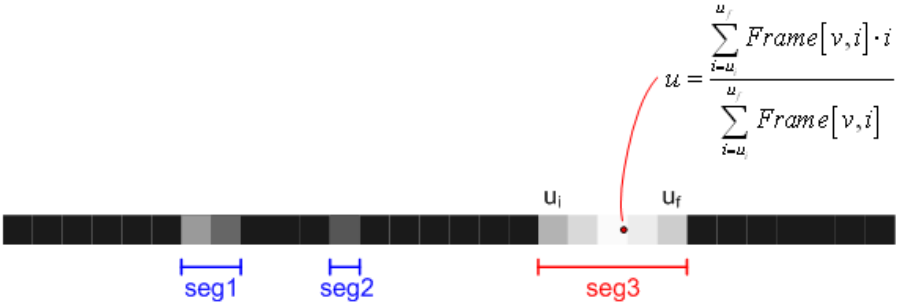

Figure 4 : Détermination de la position du point laser

Détermination de coordonnées de sous pixel (Sub-pixel) : Pour déterminer plus précisément la position du point de l'objet frappé par le laser, on se propose non plus de travailler que dans les

coordonnées entières des pixels, mais dans le domaine réel (comme par exemple : 118,82 pixels). Pour ce faire, afin de déterminer plus précisément la position de chaque point, on se propose de déterminer le centre de gravité lumineux, comme le montre la Figure 4. Améliorer votre algorithme de détection de la ligne lumineuse, en ayant recours à l'approche Sub-Pixel.

Etant donné que la pièce est vue par deux yeux (deux appareils photo), on souhaite que ces deux vues soient gérées dans un unique objet **ligne**. Attention à cette étape, ces informations ne sont plus codées dans des images (sinon la taille en mémoire explose).

Etape 2 : Transformations mathématiques

Le but de cette étape est de passer du repère de l'image et de son unité qu'est le pixel, à celui du monde réel et de son unité le mètre (ou le millimètre dans notre cas).

Changement de repère image => caméra

Afin de permettre d'effectuer la transformation pixels/distances en système métrique et donc de passer du repère image au repère caméra (Attention aux origines qui changent !), une photographie d'un objet dont les dimensions sont connues vous est proposée. On considère dans ce projet que le passage du repère caméra au repère image n'est le fruit que de translations (aucune rotation n'intervient ici).

Réaliser la méthode de la classe **ligne,** *tranformation_vers_plan_camera*, qui prend en argument les valeurs numériques des translations nécessaires et des facteurs d'échelle, et qui génère et retourne une liste de points 3D exprimés dans le repère caméra.

Projection – Modèle de sténopé

Ces transformations mathématiques sont principalement dues à la position de la caméra par rapport au plan du laser ligne, aux défauts et fonctionnement même d'une caméra qui capture une scène tridimensionnelle sur le plan image, comme l'illustre la Figure 5:

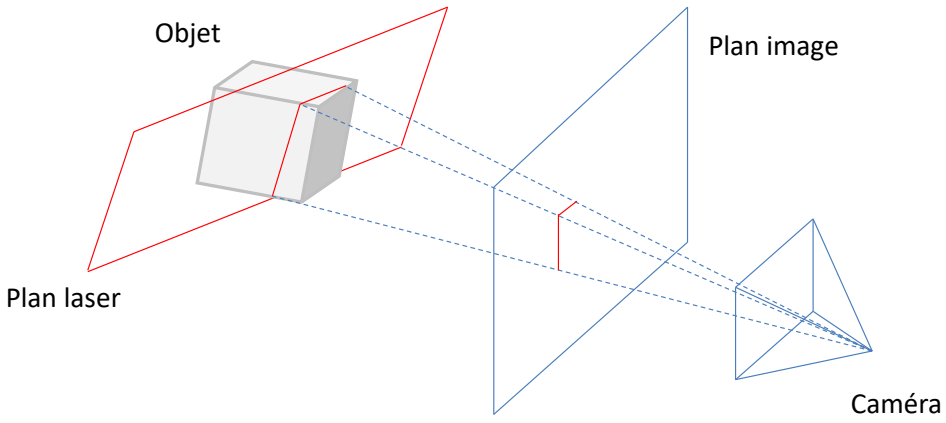

Figure 5 : Projection de l'objet dans le plan image d'une caméra

Le plan image est celui que vous avez traité dans l'étape précédente. L'objectif de cette étape est donc de définir une fonction permettant de déterminer la position des points sur la pièce, et donc dans le plan fixe du laser. Il existe plusieurs modèles qui permettent d'effectuer cette projection inverse : les modèles de sténopé. Après avoir fait quelques recherches simples sur ces modèles, et

sélectionné le plus adapté en fonction des données fournies et de votre objectif, concevoir, puis implémenter une fonction qui, en prenant comme argument les paramètres de l'expérience, génère les deux lignes vues par les deux cameras.

Dans un second temps, proposer une méthode de la classe **ligne** qui retourne une seule liste de points 3D à partir des deux lignes calculées précédemment. Vous pouvez définir votre propre méthode permettant de déterminer cette unique ligne dans le plan laser à partir des deux lignes vues par les deux yeux de notre système.

Etape 3 : Construction du modèle 3D

Afin de pouvoir construire le volume de la **pièce**, il est nécessaire d'associer à chaque objet **ligne** sa position angulaire (angle de la table indexée).

Réaliser la méthode de la classe **pièce** qui, à partir des lignes exprimées dans le plan laser et consultables à partir des objets de la classe **ligne**, génère la liste des points 3D exprimés cette fois-ci dans le repère tridimensionnel de la pièce.

Etape 4 : Affichage du modèle 3D dans votre programme

Le but est de cette étape est d'avoir un visuel de l'ensemble des points 3D repositionnés dans le repère pièce. Il n'est pas demandé dans cette question d'utiliser des moteurs 3D puissants : le but est d'avoir un retour visuel rapide, pas nécessairement de manipuler le modèle 3D de façon fluide (ou avec des jeux de lumière ou de texture).

Etape 5 : Export en STL

Le format de fichier STL est fortement utilisé dans le monde de la fabrication additive (dont les imprimantes 3D en sont les représentantes les plus connues du grand public). Afin d'effectuer une impression de l'objet scanné et numérisé, on souhaite exporter ce dernier dans le format STL.

Ce dernier est un fichier texte, simple et structuré autour du concept de facette. Une facette (`facet`) est une surface triangulaire définie par sa normale (`normal`) et trois points en délimitant les frontières (`vertex`). Ces points sont des réels dont la séparation unités/décimales est le point :

```
facet normal ni nj nk
        outer loop
                vertex v1x v1y v1z
                vertex v2x v2y v2z
                vertex v3x v3y v3z
        endloop
endfacet
```

L'ensemble des facettes de l'objet sont ainsi décrites les unes après les autres. Afin de toutes les contenir, celles-ci sont encadrées entre deux lignes de texte définissant les limites de définition du solide :

```
solid Nomdelobjet
```
Liste de toutes les facettes
```
endsolid Nomdelobjet
```

Un exemple de fichier STL est proposé sur le serveur Info/Math. Etant donné que c'est un fichier textuel, vous pouvez l'ouvrir et le lire directement depuis un éditeur de texte.

Réaliser les méthodes de la classe **pièce** qui réalise :

- le regroupement de 3 points mesurés en une facette (en utilisant la méthode que vous considérez intéressante/efficace)
- l'enregistrement, qui prend en argument le chemin d'accès du fichier et le nom de ce dernier, et enregistre votre modèle 3D de l'objet sous le format STL.

Etape 6 : Tests et validations

La dernière étape de ce projet consiste à vérifier que l'ensemble des étapes précédentes se sont bien déroulées et qu'il en résulte un modèle numérique de l'objet réel de qualité (faible distorsion, peu d'écart géométrique…). Pour ce faire, on se propose d'ouvrir dans un outil commercial (CATIA ou un logiciel de commande d'une imprimante 3D, voire sur le site https://www.viewstl.com/ qui permet d'ouvrir un fichier de ce type sans aucune installation nécessaire) le fichier STL obtenu précédemment. Vérifier que ce dernier est bien lu et analyser le modèle 3D ainsi obtenu (qualité, formes…).

Si le temps et les ressources le permettent, nous verrons s'il est possible d'imprimer le modèle numérique ainsi obtenu sur une des machines disponibles à l'Ecole.

Améliorations

Une fois le programme fonctionnel donnant de bons résultats, les voies d'amélioration sont multiples :

- **Interface utilisateur** : faire que via l'interface l'ensemble des étapes soient paramétrables et que l'on puisse suivre le bon déroulement du processus complet (allant de l'extraction des images jusqu'à la génération du volume final codé en STL)
- **Algorithmes :** Améliorer les méthode d'agrégation des images vues par les deux points de vue, ou la méthode de sélection des points utilisés pour construire les facettes du format STL
- **Rapidité :** Identifier les moyens d'accélérer les traitements afin d'améliorer le rapport qualité de la surface 3D sur temps nécessaire à sa génération.

Ressources bibliographiques

Plusieurs sources documentaires peuvent être utiles pour la bonne compréhension de ce qui vous est demandé dans ce projet de support à la conception :

- Librairie OpenCV : https://opencv.org/
- Modèle de sténopé d'une caméra : https://fr.wikipedia.org/wiki/Calibration_de_cam%C3%A9ra
- GOM – Mesures par Méthodes Optiques : https://www.gom.com/fr.html
- MOREIRA, Ana S. Ferreira1 A. Paulo et COSTA, Paulo G. Low-Cost System for Object Positioning Through Laser-Camera Triangulation. EPIA 2007, 2007 (disponible sur le portail).