

Rapport semestriel

Projet DS3DLive

26 avril 2021

Maxime TEIL (LGP2 - Grenoble INP)
maxime.teil@lgp2.grenoble-inp.fr
04 76 82 69 43



Partenaires



Projet financé par la région Auvergne-Rhône-Alpes



Table des matières

1	État d'avancement	4
2	Montage du dispositif en statique	5
2.1	Système optique	5
2.2	Assemblage structurel et fonctionnel	6
3	Travaux numériques réalisés	7
3.1	Librairie Python ds3dlive	7
3.2	Calibration de caméra / APN	9
3.3	Système de stéréovision	10
3.4	Simulation 3D	11
3.5	Caractérisation de plis par Alicona Infinite Focus	13
4	Gestion du projet	14
A	Calibration des systèmes optiques - Théorie	17
B	Détection automatique de la présence de plis	20

Ce rapport semestriel est le premier depuis le lancement du projet DS3DLive le 1^{er} novembre 2020. Il a pour vocation de faire un rappel sur les objectifs qui ont été fixés lors de la réunion de lancement avec l'ensemble des partenaires, d'informer sur l'état d'avancement depuis cette date, de détailler l'ensemble des tâches accomplies et de faire un point sur la gestion actuelle du projet.

Une première partie se contentera de rappeler le planning envisagé lors du lancement et en fera une comparaison avec le planning réel. Il sera ensuite fait une prévision sur le planning des mois qui viennent.

Une présentation du montage d'essais en statique sera ensuite menée. Ce montage a pour objectif de répondre à l'ensemble des problématiques liées à la caractérisation par une technique de stéréovision en statique.

Le dispositif étant constitué de capteurs numériques, il est alors nécessaire de générer des codes de programmation pour mettre en œuvre les routines liées au projet. Pour cela, une librairie développée en Python a été créée et sera présentée.

Enfin, quelques points de discussion concernant la gestion du projet seront proposés.

1 État d'avancement

Un rappel est fait ici concernant l'organisation du travail dans ce projet. Quatre tâches principales ont été déterminées :

WP1 : Mise en place et développement technologique du dispositif d'imagerie par stéréovision ;

WP2 : Développement de routines d'analyse d'images et de stéréocorrélation ;

WP3 : Applications en statique ;

WP4 : Applications en dynamique sur pilote de laboratoire ou machine semi-industrielle.

La figure 1 met en avant les interactions entre ces différentes tâches. WP1 et WP2 sont les tâches prioritaires car elles sont toutes deux nécessaires à la réalisation des autres tâches. WP3 doit être réalisée avant WP4 afin d'assurer une réponse aux problématiques de manière progressive, et donc certainement plus efficace.

Planning prévisionnel du début de projet et planning réel

Un planning prévisionnel a été présenté en début de projet lors de la réunion de lancement. Le diagramme associé est présenté sur la figure 2. Sur ce dernier est mis en avant la date actuelle grâce à la présence d'une ligne verticale pointillée. À ce jour, une grande partie du travail de mise en place du dispositif est censé être réalisé et le travail numérique doit être bien entamé. Des premiers essais en statique doivent avoir été menés afin de faire progresser le développement expérimental et numérique (WP1 et WP2).

Qu'en est-il du planning réellement réalisé jusqu'à aujourd'hui ?

Le diagramme peut rester tel quel à la condition que la tâche WP2 soit légèrement modifiée, ce qui précise encore plus l'organisation du travail.

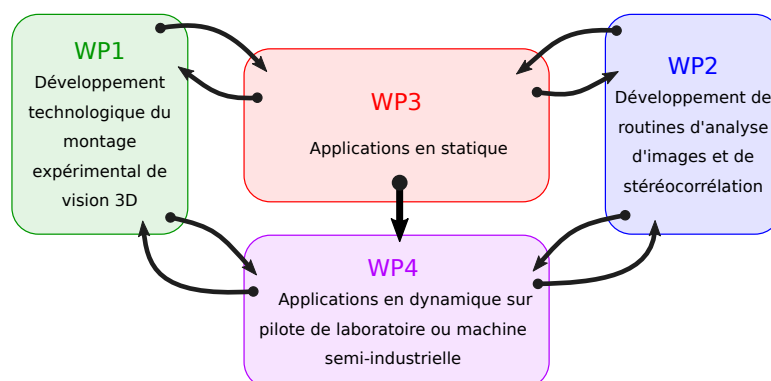


FIGURE 1 – Organisation du projet sous forme de tâches principales

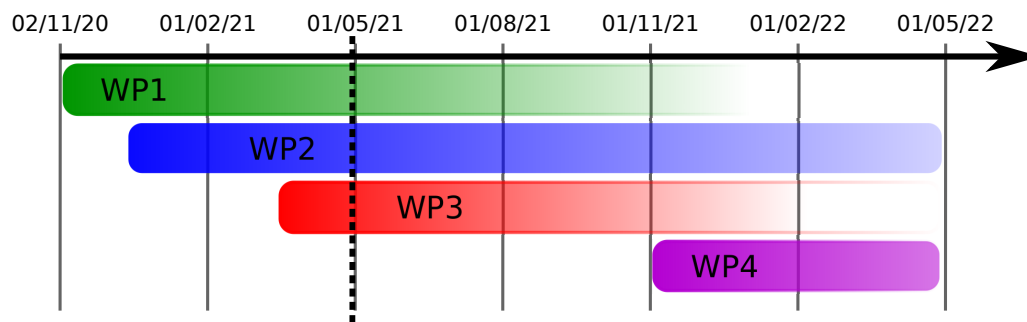


FIGURE 2 – Planning prévisionnel

WP2 : Développement de routines *de calibration des systèmes optiques*, d'analyse d'images et de stéréocorrélation.

En effet, il est primordial d'étalonner le système optique avant de procéder au traitement des images, et cela fait partie du travail numérique. Modification faite, le planning prévisionnel est respecté. Les prochains paragraphes ont pour objectif de présenter dans les détails les avancées concernant les tâches WP1, WP2 et WP3. De manière plus globale, le montage est opérationnel et peut être monté sur la machine pilote à tout moment : WP1 est donc très bien avancé. Une librairie Python permet de calibrer les caméras et d'étalonner le système de stéréovision afin de permettre certaines mesures de position dans l'espace : WP2 s'est donc bien déroulé jusqu'à présent. Enfin, des premiers essais de caractérisation 3D ont été menés afin de s'assurer de la fonctionnalité du dispositif expérimental et de la calibration du système : les caractérisations en statique sont donc possibles malgré certains efforts d'optimisation à fournir pour la suite.

2 Montage du dispositif en statique

L'objectif du projet est d'être en capacité de monter un dispositif de caractérisation par stéréovision sur une machine de production. Une première étape consiste à ne traiter que des caractérisations hors production, c'est-à-dire sur des papiers qui sont statiques. Cela permet, dans un premier temps, de ne se focaliser que sur les aspects liés au montage et à l'architecture algorithmique du projet. Lorsque l'ensemble de ces tâches primordiales seront correctement traitées, il sera nécessaire de considérer l'aspect dynamique pour permettre de rendre le dispositif fonctionnel à l'échelle de la production semi-industrielle. Le dispositif présenté dans cette partie est la version adaptée au cas d'études en statique. Il est envisagé d'y apporter certaines corrections ou modifications dans sa version d'études en dynamique, notamment en ce qui concerne le système optique.

2.1 Système optique

Puisque dans cette partie le sujet ne bouge pas, il n'est pas nécessaire de considérer une caméra haute-performance capable de délivrer un flux important d'images à traiter. Il est cependant nécessaire de considérer des capteurs d'images dont la définition est celle attendue dans ce projet. Afin de ne pas générer trop de transformations liées à la distorsion lors de la prise de vue, il est choisi d'utiliser des objectifs de type appareil photographique numérique (APN) réflex dont la distance focale ne soit pas inférieure à 24 mm. Pour ces raisons il a été fait le choix d'un ensemble de deux APN réflex de la marque Canon et du modèle EOS 1200D et de deux objectifs de la marque Canon et du modèle EF-S 24 mm f/2.8 STM dont les caractéristiques sont données, respectivement, dans les figures 3 et 4.

Le capteur ainsi choisi permet d'obtenir une taille de pixel de l'ordre de 70 µm en considérant un champ horizontal de approximativement 35 cm. D'après le résultat présenté au paragraphe 3.5, cette résolution permettrait de définir un pli par l'intermédiaire d'une dizaine de pixels.

Afin de garantir une relativement bonne répétabilité lors des essais, chaque APN est associé à un objectif et l'ensemble est étiqueté "gauche" ou "droit". Les erreurs expérimentales liées aux capteurs et objectifs sont donc répétées pour chaque prise de mesures sur les images. D'un point de vue pratique,

Format	APS-C (22.3 mm × 14.9 mm)
Pixels effectifs	18 MPx
Ratio de format	3 : 2
Filtre passe bas	intégré



(a)

(b)

FIGURE 3 – (a) Caractéristiques du capteur de l'APN EOS 1200D de chez Canon et (b) photographie de l'appareil. Source : www.canon.fr.

Focale équivalente (36 mm × 24 mm)	38 mm
Distance minimale mise au point	160 mm
Autofocus motorisé	oui
Angle de champ horizontal	59°10'
Angle de champ vertical	50°35'



(a)

(b)

FIGURE 4 – (a) Caractéristiques de l'objectif EF-S 24 mm f/2.8 STM de chez Canon et (b) photographie de l'objectif. Source : www.canon.fr.

la calibration est faite pour chaque couple APN-objectif, ce qui permet d'obtenir des matrices de calibration et coefficients de distorsion qui sont propres à chacun de ses couples. Si les appareils ou les objectifs ne sont pas échangés alors ces propriétés peuvent être sauvegardées puis rechargées.

L'enregistrement des images se fait au format JPEG haute qualité (18 MPx) sur carte SD insérée dans l'APN. Les traitements et post-traitements sont réalisés après la séance de prises de vue. L'ensemble des campagnes de prises de vue sont donc détaillées dans le cahier de laboratoire. Il est fait mention de la date, l'heure, le sujet de la prise de vue et les liaisons entre sujets et noms des images.

2.2 Assemblage structurel et fonctionnel

Les systèmes optiques sont censés pouvoir se fixer de manière permanente ou temporaire sur une coucheuse semi-industrielle. Dans le cadre de ce projet, et dans un premier temps, le choix se porte sur la machine pilote présente au laboratoire LGP2. Une structure a donc été imaginée afin de pouvoir fixer les systèmes optiques sur cette machine, tout en gardant la possibilité de régler le système de stéréovision. L'assemblage est présenté sur la figure 5 : il est constitué de poutres métalliques de section circulaire creuse de 30 mm de diamètre extérieur. Différentes liaisons sont présentes afin de permettre les réglages suivants :

- Distance entre les deux caméras (flèche rouge sur la figure 5-(a)) ;
- Hauteur des caméras sur l'axe de production (flèches vertes sur la figure 5-(a)) ;
- Distance entre les caméras et le papier (flèche rouge sur la figure 5-(b)) ;
- Angle de vue définissant le champ vertical (flèche verte sur la figure 5-(b)) ;
- Angles de vue, pour chaque caméra, définissant le champ horizontal (flèches bleues sur la figure 5-(b)).

La précision des mesures 3D par stéréovision va dépendre de chacun de ces réglages. La distance entre les caméras et le papier définit la résolution des mesures. En effet, le grossissement associé au rapprochement des appareils génère une taille de pixel plus petite. En contre partie, le champ d'analyse

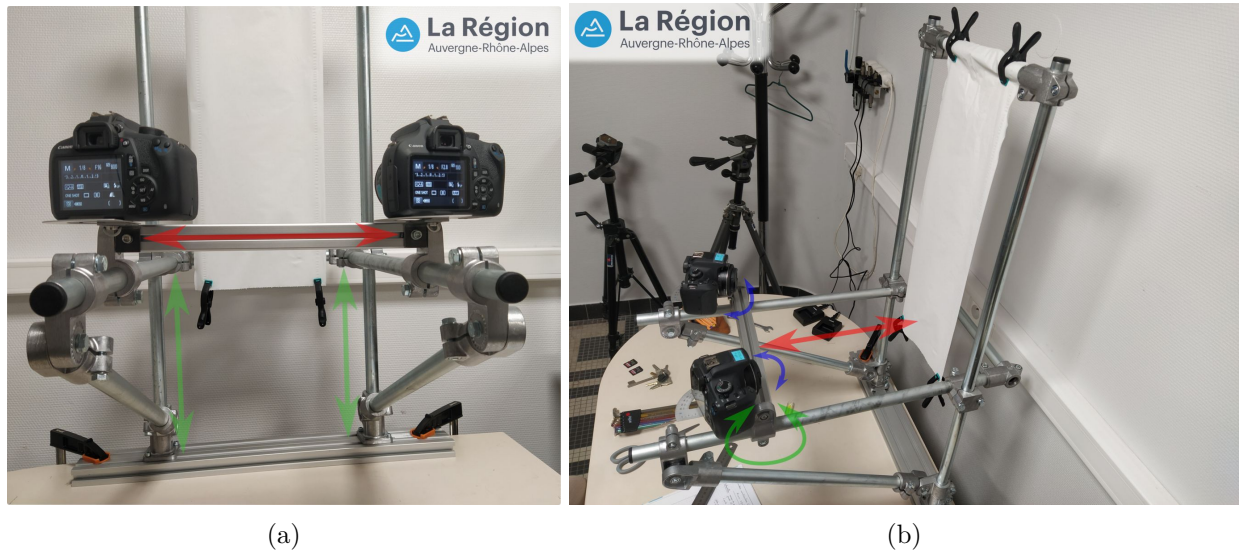


FIGURE 5 – Dispositif de caractérisation par stéréovision pour des applications en statique. Les degrés de liberté du dispositif sont représentés par les flèches de couleurs et présentés dans le paragraphe associé.

diminue. De la même manière, la distance entre les caméras et l'orientation de chacune d'entre elle va jouer un rôle sur la résolution en profondeur (axe normal au papier) mais occasionne, pour une meilleure résolution en Z , des zones d'ombres impossibles à caractériser.

Actuellement, il n'est pas évident d'être précis dans la connaissance de ces réglages et certaines améliorations pourraient voir le jour dans les prochains mois. La méthode actuelle consiste à mesurer à l'aide d'un régleur les différentes longueurs. Des rapporteurs sont également fixés aux supports des appareils photographiques afin d'estimer de manière relativement correcte les angles de chacune des caméras définissant les champs horizontaux. Il est à noter qu'il est possible de connaître de manière assez précise la position et l'orientation de chacune des caméras par une méthode numérique. Il suffit pour cela de prendre un cliché d'une mire dont la position est parfaitement connue dans l'espace : si les deux caméras sont calibrées alors il est possible de remonter à leurs positions respectives. Dans cette condition, il est important de connaître parfaitement la position de la mire sur le dispositif expérimental.

3 Travaux numériques réalisés

Bien que le montage soit fonctionnel, le projet se destine essentiellement à du traitement numérique. Le choix du langage de programmation Python a été fait, puisqu'il s'agit d'un langage très courant, de haut niveau et donc relativement facile à utiliser avec énormément de ressources. De plus, la librairie IP SDK développée par Reactiv'IP sera utilisée pour le traitement d'image et possède une implémentation en Python.

Le travail numérique consiste en plusieurs tâches qui peuvent être elles-mêmes regroupées en différentes catégories selon les objectifs. Pour cette raison, une librairie associée au projet a été créée. Cette librairie présente différents modules permettant de répondre à différentes problématiques. Une présentation plus détaillée est faite dans les prochains paragraphes.

3.1 Librairie Python ds3dlive

La librairie développée en Python dans ce projet est simplement nommée par le nom du projet : ds3dlive. Cette librairie est en cours de développement et n'est pas encore distribuée. Le travail de développement se fait par contre par l'intermédiaire de l'outil très populaire Git grâce à un workflow de type "gitflow". Afin de détailler cette dernière phrase : plusieurs versions du programme existent. Une version stable et fonctionnelle est toujours disponible. En parallèle, une branche du même programme

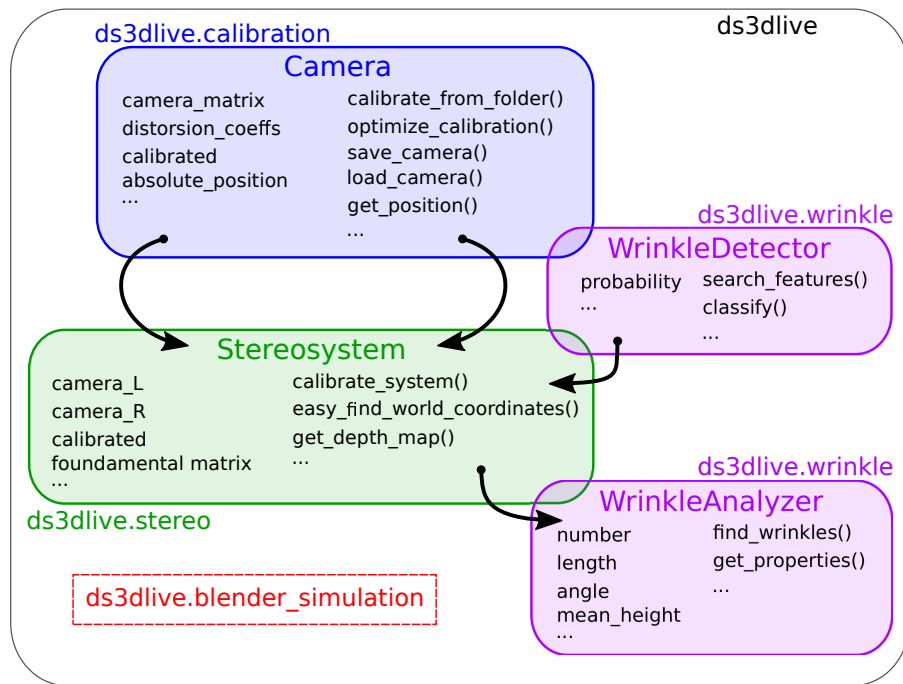


FIGURE 6 – Diagramme fonctionnel de la librairie ds3dlive. La librairie est constituée de plusieurs modules (calibration, stereo, ...) qui génère des objets de différentes classes (Camera, Stereosystem, ...) interagissant entre eux.

en cours de développement est en perpétuelle évolution. Lorsque l'occasion le permet, la version stable est mise à jour par la version en développement. Chaque ajout ou modification d'une fonctionnalité est traitée à part et est testée avant d'être envoyée sur la version de développement. A chaque fois qu'une mise à jour est faite, des tests unitaires sont réalisés afin de prévenir des problèmes pouvant intervenir suites aux modifications effectuées.

La librairie possède des modules dont certains sont déjà en cours de développement et bien avancés :

calibration : Il s'agit de la librairie qui gère les caméras / APN. Ce module permet de les calibrer, de sauvegarder leurs propriétés, de les charger mais aussi d'interagir directement avec les images fournies par le système (corriger la distorsion par exemple). Afin de permettre tout cela, le module génère des objets "Camera".

stereo : Ce module est basé sur l'objet "Stereosystem" qui est formé par deux objets "Camera". Le module se charge de définir les contraintes épipolaires associé au système de stéréovision et de connaître les positions exactes des caméras / APN afin d'être capable de situer des points dans l'espace (en cours de développement).

blender_simulation : Ce module n'est pas destiné à être utilisé en console ou avec une interface graphique. Il est destiné à créer un modèle 3D sur le logiciel Blender et s'ouvre donc depuis la console Python intégrée dans le logiciel.

wrinkle : Le module a pour objectif de détecter de manière automatique la présence de plis sur le papier (développement à faire) et de caractériser dimensionnellement les plis lorsqu'ils sont détectés et que l'information 3D est accessible (développement en cours).

La librairie est accessible sur le site internet GitHub sous la forme d'un projet privé détenu par le compte personnel "MaximeTeil". Il est possible de partager ce répertoire entre trois collaborateurs (deux autres comptes donc). A la demande de chacun, un accès peut être ouvert aux partenaires du projet. La limite du nombre de collaborateurs peut entraîner des accès temporaires et non permanents. Une discussion peut être ouverte concernant le passage du répertoire en mode "public". Ceci permettrait à tout le monde d'accéder au projet et de participer aux différentes tâches. En contrepartie, la confidentialité des travaux ne serait plus maintenue.

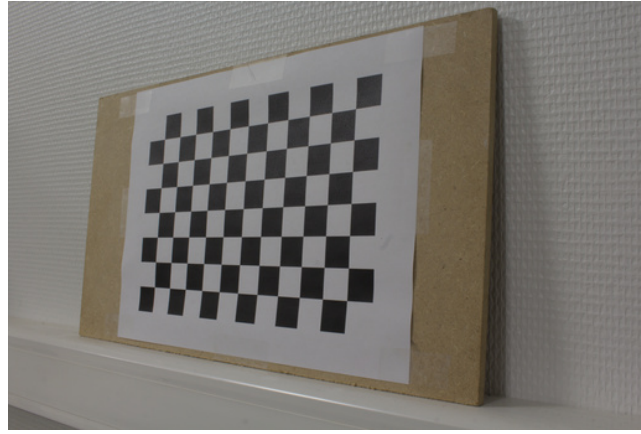


FIGURE 7 – Prise de vue permettant de calibrer une caméra avec une mire de type "damier" de 7×11 et des carrés de 20 mm.

3.2 Calibration de caméra / APN

Le module de calibration de caméra est adapté à la création d'objets de la classe "Camera". Ces objets ont pour objectif de stocker l'ensemble des propriétés intrinsèques à la caméra associée et de proposer un ensemble de méthodes pour traiter les données efficacement et faciliter les tâches de l'utilisateur. La création d'un objet de la classe Camera peut se faire sans aucune information. Cela aura pour conséquence de générer une caméra dont les propriétés sont toutes nulles par défaut. Une manière de mesurer et calculer les propriétés de la caméra est de la calibrer selon la théorie présentée dans l'annexe A.

La méthode de calibration consiste à prendre plusieurs clichés d'une mire bien connue, prise sous différents angles. Cette mire est constituée de zones à très fortes différences de contraste permettant ainsi de sélectionner de manière automatique et précise des points dont les positions sont connues. L'utilisation de cette mire permet de créer une table de correspondance entre la position des points réels dans l'espace et la position des points images associés dans les images obtenues. La mire utilisée actuellement dans le projet est de type "damier" : chaque contact entre les carrés noirs sont de points faciles à détecter automatiquement. La position des points est déterminée par la taille des carrés, le nombre de lignes et le nombre de colonnes du damier. Un script \LaTeX a été implémenté afin de générer automatiquement au damier au format PDF : il suffit de rentrer en variable le nombre de lignes, colonnes et la taille souhaitée des carrés. Les exemples donnés ici présentent une mire de 7 lignes, 11 colonnes et dont les carrés ont des côtés de 20 mm. Un cliché permettant de calibrer une caméra est montré sur la figure 7.

En fonction du nombre d'images acquises (n) et du nombre de points constituant la mire (p), un système d'équations linéaires en \bar{P} peut être obtenu :

$$\forall (n, p) \in (\mathbb{N}_+^*)^2, \quad \bar{q}_{n,p} \sim \bar{P}_n \cdot \bar{Q}_{n,p} \quad (1)$$

Puisque les images présentent obligatoirement du bruit et que le problème est surdimensionné, la résolution de ce système se fait au sens des moindres carrés. Une décomposition en valeurs singulières permet de trouver les éléments constituant les matrices \bar{P}_n qui minimisent l'erreur quadratique sur le système d'équations.

La connaissance des matrices de projection \bar{P}_n donne accès aux matrices de calibration \bar{K}_n (voir annexe A au besoin). La matrice de calibration de la caméra est déterminée comme étant la moyenne arithmétique des matrices de calibration pour chaque prise de vue :

$$\forall n \in \mathbb{N}_+^*, \quad \bar{K} = \frac{\sum_n \bar{K}_n}{n} \quad (2)$$

Le module permet également de déterminer la pose lorsque la caméra a été déplacée. En effet, ayant connaissance des paramètres intrinsèques à la caméra avec la matrice de calibration, il est possible

de déterminer les paramètres extrinsèques pour établir la matrice de projection. Une solution pour permettre cela est donc une nouvelle fois d'utiliser la mire pour être capable de déterminer la position et l'orientation de la caméra par rapport à cette mire. Il peut être envisagé de placer des autocollants sur le montage afin d'assurer une pose toujours relative au même repère. Dans ce cas, il faut s'assurer d'avoir au moins 3 points de mire (la méthode s'appelle d'ailleurs la méthode des 3 points).

Enfin, la connaissance des coefficients de distorsion lors de la phase de calibration permet au module d'utiliser les propriétés de la caméra pour déformer l'image acquise afin de corriger les déformations liées à la distorsion. Cela est très important pour des applications de métrologie.

Les algorithmes permettant la calibration des caméras sont basés sur la librairie Python d'OpenCV, qui possède un module de calibration 3D. Cette librairie est surtout utilisée pour détecter automatiquement et très précisément les points de mire et pour approximer efficacement les coefficients de distorsion. Dans la condition où ces deux outils pourraient être développés dans la librairie IPSDK, l'ensemble de ce module de calibration pourrait être implémenté avec IPSDK.

3.3 Système de stéréovision

Le module de stéréovision est constitué d'une classe "Stereosystem" qui est construit à partir de deux objets de la classe Camera. Les propriétés de chacune des caméras sont utilisées pour définir la contrainte épipolaire du système et être capable de situer un point dans l'espace à partir des points images associés sur chacun des capteurs. Ce module a finalement deux objectifs principaux :

- Être capable d'utiliser la contrainte épipolaire pour retrouver des points stéréocorrespondants ;
- À partir des points stéréocorrespondants, être capable de déterminer la position du point objet dans le repère monde (ou dans le repère caméra).

Géométrie épipolaire - La contrainte épipolaire est la contrainte qui permet de dire que si l'on s'intéresse à un point image sur une des deux images du système alors le point image associé dans l'autre image est placé sur une droite (dont on est capable de calculer l'équation). La position sur cette droite est déterminée par la profondeur du point objet par rapport au repère caméra : la position dépend donc de Z^C (voir annexe A). La contrainte épipolaire peut être définie par l'intermédiaire d'une matrice 3×3 $\overline{\overline{F}}$, appelée matrice fondamentale, grâce à la relation suivante :

$$\overline{q_2}^T \cdot \overline{\overline{F}} \cdot \overline{q_1} = 0 \quad (3)$$

où $\overline{q_1}$ et $\overline{q_2}$ sont les coordonnées dans le repère des pixels des points images sur chacune des images.

Géométrie épipolaire calibrée - Dans le cas où les matrices de calibration des caméras sont bien connues, il est possible de déterminer la contrainte épipolaire uniquement avec les paramètres extrinsèques des caméras (leurs positions et orientations) en travaillant dans les repères images et non plus dans les repères des pixels. La contrainte devient alors, en utilisant les termes de l'annexe A :

$$\overline{q_{i2}}^T \cdot \overline{\overline{E}} \cdot \overline{q_{i1}} = 0 \quad \text{avec} \quad \overline{\overline{E}} \sim \overline{\overline{K_2}}^T \cdot \overline{\overline{F}} \cdot \overline{\overline{K_1}} \sim \overline{\overline{R_2}} \cdot \overline{\overline{[t_1 - t_2]_\times}} \cdot \overline{\overline{R_1}}^T \quad (4)$$

Ici, la notation matricielle du produit vectoriel est utilisée :

$$\overline{\overline{[a]_\times}} = \begin{pmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{pmatrix} \quad (5)$$

Reconstruction 3D - À partir des paramètres intrinsèques et extrinsèques des deux caméras, il est possible de définir leur matrice de projection, respectivement $\overline{\overline{P_1}}$ et $\overline{\overline{P_2}}$. Lorsque deux points $\overline{q_1}$ et $\overline{q_2}$ sont stéréocorrespondants, il est possible de reconstruire le point objet \overline{Q} associé dans le repère monde par l'intermédiaire du système suivant, linéaire en les coefficients de \overline{Q} :

$$\begin{cases} \overline{\overline{P_1}} \cdot \overline{Q} \sim \overline{q_1} \\ \overline{\overline{P_2}} \cdot \overline{Q} \sim \overline{q_2} \end{cases} \quad (6)$$

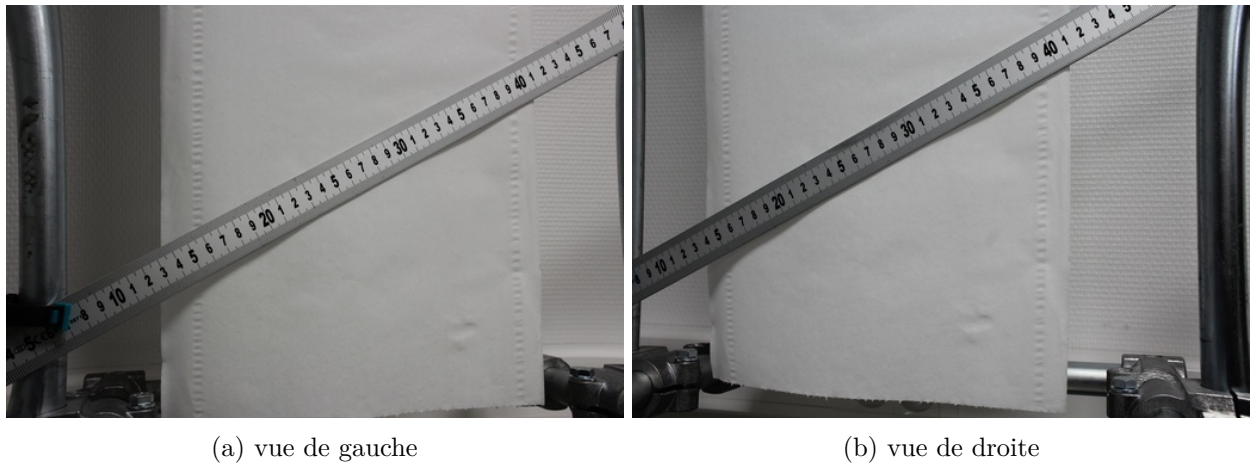


FIGURE 8 – Prise de vue stéréoscopique d'un réglet afin de vérifier la mesure 3D. Les images présentées sont corrigées de la distorsion.

Trait de réglet	$\overline{q_1}$	$\overline{q_2}$	\overline{Q}
297 mm	$\begin{pmatrix} 3227.14 \\ 1107.86 \end{pmatrix}$	$\begin{pmatrix} 2234.89 \\ 962.97 \end{pmatrix}$	$\begin{pmatrix} 98.90 \\ 15.46 \\ 10.09 \end{pmatrix}$
298 mm	$\begin{pmatrix} 3237.61 \\ 1101.94 \end{pmatrix}$	$\begin{pmatrix} 2245.79 \\ 956.81 \end{pmatrix}$	$\begin{pmatrix} 99.75 \\ 15.02 \\ 10.09 \end{pmatrix}$

TABLE 1 – Prise manuelle des coordonnées dans les repères des pixels pour chaque trait de réglet étudié.

Exemple de mesure 3D

Considérons la prise de vue stéréoscopique présentée sur la figure 8. Un réglet a été placé dans le champ de vision des deux caméras. Nous allons tenter de mesurer une longueur de 1 mm sur ce réglet. Pour cela, les coordonnées des extrémités de deux lignes adjacentes sur le réglet vont être prélevées manuellement : nous avons alors, pour chaque trait, $\overline{q_1}$ et $\overline{q_2}$. La calibration des caméras a été faite et le système stéréoscopique a également été calibré : nous avons $\overline{\overline{P_1}}$ et $\overline{\overline{P_2}}$. La table 1 indique les coordonnées retrouvées manuellement pour les traits de réglet de 297 mm et 298 mm ainsi que les coordonnées calculées dans le repère monde. La norme mesurée du vecteur reliant les deux traits de réglet dans le repère monde est de 0.959 mm pour une valeur théorique de 1 mm. Cela représente une erreur relative de 4.09 %.

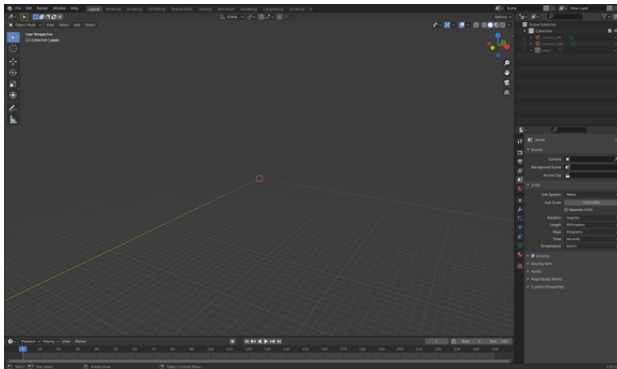
L'erreur relative semble rester aux alentours de cette valeur lorsque le même exercice est réalisé sur des longueurs plus grandes (*i.e.* de l'ordre de plusieurs centimètres). Cette erreur peut provenir :

- Du bruit présent dans les images ;
- De la précision de la calibration et de la correction de la distorsion ;
- D'une erreur de lecture humaine sur les points images ;
- De la précision de la mire.

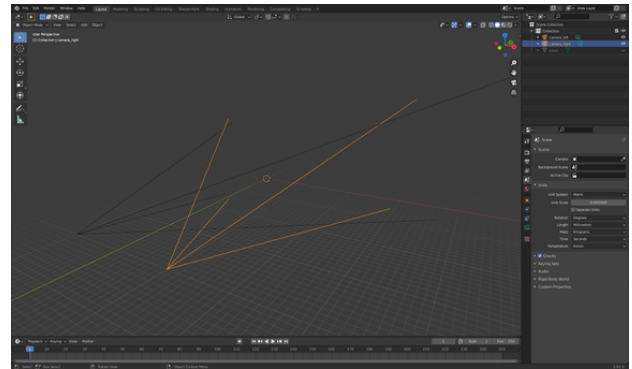
3.4 Simulation 3D

Afin de commencer les travaux numériques sur le traitements d'images et de permettre une mise en place facilitée du dispositif expérimental, des simulations peuvent être réalisées grâce au logiciel de modélisation 3D Blender.

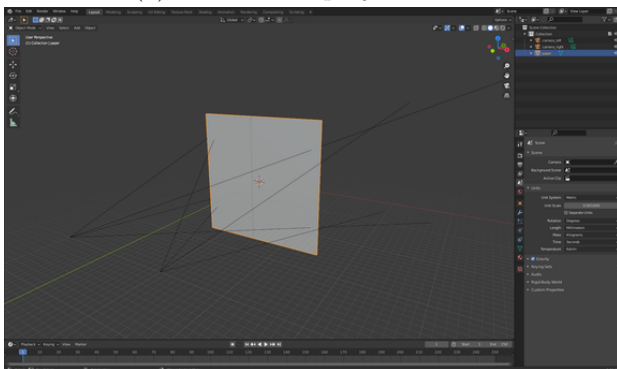
La figure 9 illustre les différentes étapes permettant de simuler le dispositif expérimental ainsi que le papier avec un pli vertical. Cela peut être réalisé par l'intermédiaire de l'interface graphique du logiciel mais il a été fait le choix de mener l'ensemble de cette procédure de manière automatique grâce à un



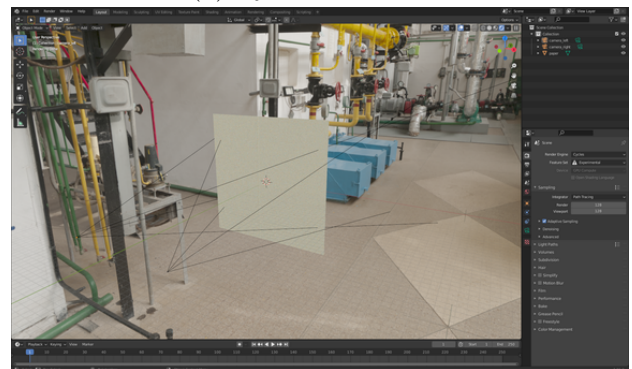
(a) Création du projet Blender



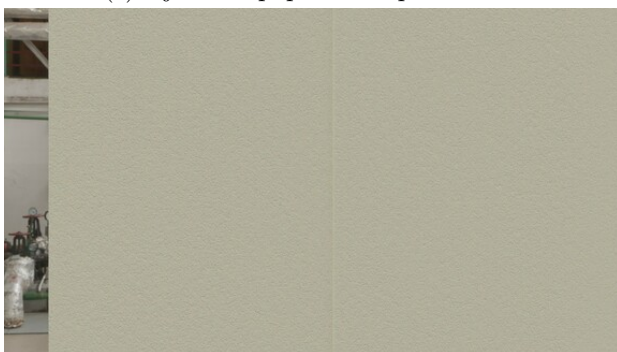
(b) Ajout des caméras



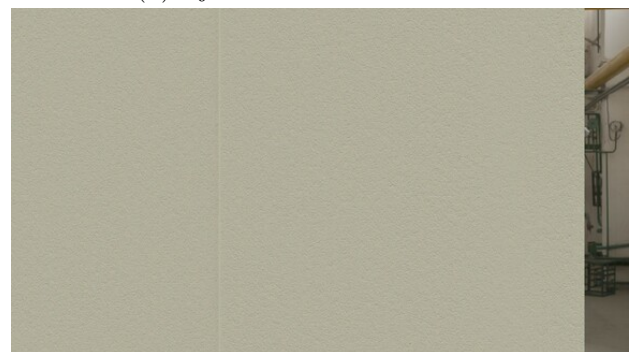
(c) Ajout du papier avec pli vertical



(d) Ajout d'un "monde" réaliste



(e) Acquisition de l'image de gauche



(f) Acquisition de l'image de droite

FIGURE 9 – Utilisation "graphique" de Blender dans ce projet

Paramètres propres au dispositif
→ Distance caméra 1 / caméra 2
→ Distance caméras / papier
→ Angles des caméras
→ Largeur du papier
→ Largeur, hauteur, position et résolution du pli
→ Longueurs focales des caméras
→ Éclairages
→ Autres à venir...
Paramètres physiques
→ Texture prédéfinie (pas obligatoire)
→ Rugosité
→ Effet "subsurface" (transmission partielle de l'éclairage sur l'épaisseur du matériau avec potentielle modification des couleurs)
→ Autres à venir...
Paramètres de rendu
→ Moteur de rendu
→ Insertion du monde dans la scène
→ Type et méthodes de calcul
→ Taille et format des images de sortie
→ Nombre d'images moyennées
→ Gestion des sauvegardes
→ Autres à venir...

TABLE 2 – Paramètres qui rentrent en considération dans la simulation Blender

script Python, qui peut être lu par le logiciel Blender. Un module Python a été créé pour générer des fonctions utiles à ce projet. Le principe consiste à générer un fichier de paramètres au format "tsv", par exemple dans un tableur numérique, puis de lire les paramètres de ce fichier dans les fonctions du module associé. L'annexe A présente le code interprété par le logiciel Blender. Les annexes B et C représentent quant à elles, respectivement, un exemple de fichier de paramètres au format tsv et le module qui regroupe l'ensemble des fonctions pouvant être utiles au projet.

À l'instant où ces lignes sont écrites, la simulation ne permet de générer que des plis dans la direction verticale. Les plis sont générés selon un profil gaussien décrit par l'équation (7). Ce profil est déterminé selon deux paramètres : la hauteur du pli et sa largeur à mi-hauteur. Un troisième paramètre permet de situer le pli sur le matériau papier (c'est-à-dire selon l'axe horizontal). Enfin, un quatrième paramètre définit la résolution du maillage pour être plus ou moins précis dans la forme donnée au profil gaussien. Les paramètres qui rentrent en considération sont présentés dans la table 2.

3.5 Caractérisation de plis par Alicona Infinite Focus

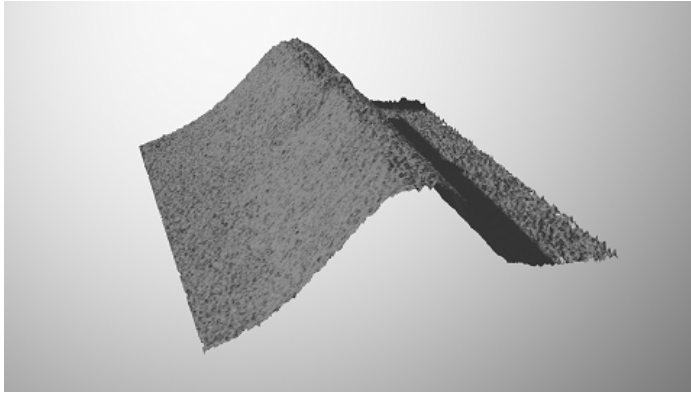
Dans le but de connaître les dimensions caractéristiques d'un pli, une analyse microscopique a été menée grâce à la machine Alicona Infinite Focus. Une carte topographique centrée sur une zone de dimensions $35\text{ mm} \times 2.5\text{ mm} \times 1\text{ mm}$ (dimensions selon, respectivement, l'axe, la largeur et la hauteur du pli) autour du pli a été calculée, ce qui permet de mailler la surface du papier et de déterminer certaines caractéristiques.

La figure 10 montre une représentation 3D de la topographie au niveau du pli ainsi que les courbes définissant 18 profils transversaux qui permettent de moyenner les caractéristiques géométriques sur la longueur de 35 mm. Deux caractéristiques ont été calculées :

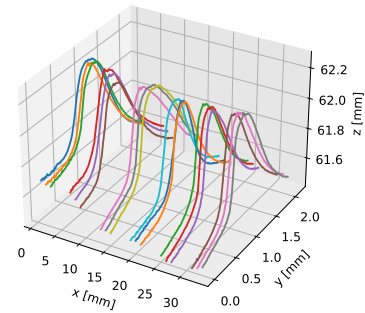
⇒ La hauteur du pli (selon la direction z) : $707 \pm 74\text{ }\mu\text{m}$;

⇒ La largeur du pli à mi hauteur (selon la direction y) : $680 \pm 167\text{ }\mu\text{m}$.

Le pli analysé a été généré lors de la formation sur la machine pilote et a été qualifié de "relativement grossier" dans le sens où il s'agit ici d'un pli dont la largeur semble relativement importante au regard



(a) Topographie 3D



(b) Analyse de profils transversaux

FIGURE 10 – Caractérisation géométrique d'un pli

de ceux qui ont été observés jusqu'à présent.

L'observation de la vue 3D et des profils transversaux nous entraîne à penser que le profil transversal d'un pli pourrait être décrit localement par une fonction gaussienne, pour laquelle il est facile de trouver l'équation (7) en fonction de sa position centrale X , de la largeur à mi-hauteur $l_{1/2}$ et de l'amplitude du pic A :

$$\begin{cases} \mathbb{R} \mapsto \mathbb{R} \\ x \rightarrow \frac{A}{\sigma\sqrt{2\pi}} e^{-\frac{(x-X)^2}{2\sigma^2}} \end{cases} \quad \text{avec} \quad \sigma = \frac{l_{1/2}}{2\sqrt{2\ln 2}} \approx 0.425 \cdot l_{1/2} \quad (7)$$

D'autres observations sur Alicona Infinite Focus sont prévues dans la suite pour caractériser des plis dont les dimensions sont plus caractéristiques de celles des plis qui se forment sur la machine pilote.

4 Gestion du projet

Perspectives

Certaines tâches n'ont pas encore été traitées et vont l'être dans les prochains mois :

- Traitement des images avec la librairie IP SDK.
- Développement d'un détecteur de plis : la méthode SIFT pourrait être utilisée (voir annexe B) avec la librairie IP SDK.
- Optimisation de la mesure 3D avec la triangulation [1].

Gestion

- Communication régulière sur les réseaux LinkedIn et ResearchGate à partir de maintenant.
- Dépenses réalisées en matériel : 1612.85 €.
- Collaborations sur le projet GitHub ? Privé / public ?

Références

- [1] R. I. Hartley and P. Sturm. Triangulation. *Computer Vision and Image Understanding*, 68(2) :146–157, November 1997.
- [2] David G Lowe. Object recognition from local scale-invariant features. In *Proceedings of the seventh IEEE international conference on computer vision*, volume 2, pages 1150–1157. Ieee, 1999.

- [3] J Liu, WP Fu, W Wang, and N Li. Image matching based on improved sift algorithm. *Chinese Journal of Scientific Instrument*, 34(5) :1107–1112, 2013.
- [4] Jian Wu, Zhiming Cui, Victor S Sheng, Pengpeng Zhao, Dongliang Su, and Shengrong Gong. A comparative study of sift and its variants. *Measurement science review*, 13(3) :122–131, 2013.
- [5] Xuelong Hu, Yingcheng Tang, and Zhenghua Zhang. Video object matching based on sift algorithm. In *2008 International Conference on Neural Networks and Signal Processing*, pages 412–415. IEEE, 2008.
- [6] Wikipédia. Scale-invariant feature transform — wikipédia, l’encyclopédie libre, 2020. [En ligne ; Page disponible le 19-décembre-2020].

Annexes

A Calibration des systèmes optiques - Théorie

Le modèle de sténopé est présenté ici. Nous allons voir comment calibrer une caméra en tenant compte de ce modèle.

Modèle de sténopé - Repères dans l'espace

Quatre repères sont nécessaires pour bien interpréter le modèle de sténopé et pouvoir situer un point dans l'espace. Les repères choisis sont les suivants :

- Repère monde : $Q(X, Y, Z)$. Repère fixe dans l'espace, qui est couramment utilisé en sciences physiques.
- Repère caméra : $Q^C(X^C, Y^C, Z^C)$. Repère qui a pour origine le centre de projection de la caméra et dont l'axe optique porte et oriente l'axe selon Z^C . X^C et Y^C sont orientés selon les axes des pixels sur le capteur de la caméra (respectivement, axe horizontal de gauche à droite et axe vertical de haut en bas).
- Repère image : $q_i(x, y)$. Repère à deux dimensions dont l'origine est le point principal (intersection entre l'axe optique et le plan image). Les axes déterminant x et y sont ceux définissant les axes X^C et Y^C , respectivement, dans le repère caméra.
- Repère des pixels : $q(u, v)$. Repère identique au repère image à la différence près que l'origine est le coin supérieur gauche de l'image et les dimensions sont exprimées en pixels.

La figure 11 illustre chacun de ces repères. Considérant un point dans l'espace à trois dimensions Q , il est possible de déterminer les coordonnées de sa projection q dans le plan image - dans un repère à deux dimensions donc. L'étude de la projection de points dans l'espace 3D dans un espace 2D nous amène à travailler avec les coordonnées homogènes. En présence de coordonnées homogènes, le symbole " \sim " correspond à une égalité vectorielle définie à un facteur près :

$$\exists s \in \mathbb{R}, \quad \bar{q} \sim \bar{Q} \Leftrightarrow s\bar{q} = \bar{Q} \quad (8)$$

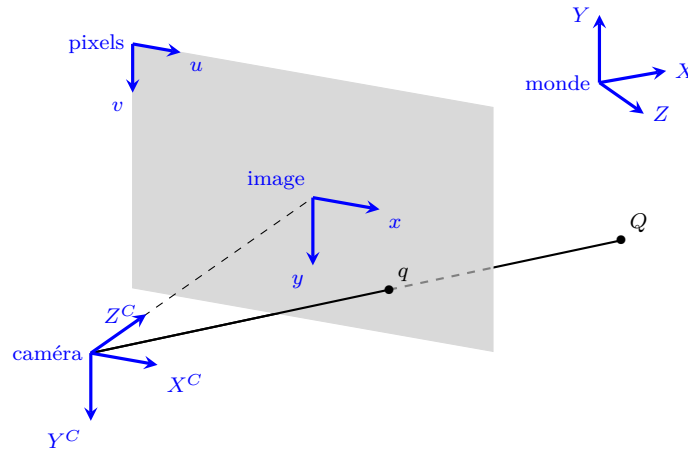


FIGURE 11 – Repère dans les différents espaces pour le modèle de sténopé.

L'objectif de la calibration est d'être capable de définir mathématiquement la relation qui relie la position d'un point dans l'espace (Q ici) à sa projection sur l'image (q ici). On cherche donc une matrice $\bar{\bar{P}}$ (dimensions 3 par 4), qui relie \bar{q} (dimension 3) à \bar{Q} (dimension 4) :

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \sim \bar{\bar{P}} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \Leftrightarrow \bar{q} \sim \bar{\bar{P}} \bar{Q} \quad (9)$$

La matrice P est appelée la matrice de projection de la caméra.

Passage du repère caméra au repère image

Par définition et notion d'optique géométrique, la distance focale f correspond à la distance entre le centre de projection du système optique et le point principal, qui est à la fois sur l'axe optique et sur le plan image. Cela veut dire que la distance focale correspond à la distance entre l'origine du repère caméra et l'origine du repère image.

On montre géométriquement (théorème de Thalès) :

$$\begin{cases} x = f \frac{X^C}{Z^C} \\ y = f \frac{Y^C}{Z^C} \end{cases} \Leftrightarrow \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \sim \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} X^C \\ Y^C \\ Z^C \\ 1 \end{pmatrix} \Leftrightarrow \bar{q}_i \sim \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \bar{Q}^C \quad (10)$$

Passage du repère image au repère des pixels

Le passage entre ces deux repères se fait par l'intermédiaire d'une translation (pour placer l'origine sur un coin de l'image) et d'un changement d'unité (pour passer d'une longueur métrique à une longueur en pixels).

Dans la suite, on appelle x_0 et y_0 les coordonnées, dans le repère image, de l'angle supérieur gauche de l'image. La translation suit donc le vecteur (x_0, y_0) .

On appelle également k_u et k_v les densités de pixels suivant, respectivement, les axes \bar{u} et \bar{v} . Ces grandeurs déterminent la quantité de pixels pour une longueur donnée. Dans le cas du capteur de l'APN Canon EOS 1200D, on peut calculer $k_u = k_v \approx 237$ px/mm. Multiplier une longueur métrique par la densité de pixel permet d'obtenir le nombre de pixel associé.

$$\bar{q} = \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} k_u & 0 & 0 \\ 0 & k_v & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & x_0 \\ 0 & 1 & y_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} k_u & 0 & k_u x_0 \\ 0 & k_v & k_v y_0 \\ 0 & 0 & 1 \end{pmatrix} \bar{q}_i \sim \begin{pmatrix} f k_u & 0 & k_u x_0 & 0 \\ 0 & f k_v & k_v y_0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \bar{Q}^C \quad (11)$$

Passage du repère monde au repère caméra

Le passage du repère monde vers le repère caméra est caractérisé par une translation de l'origine suivie d'une rotation.

On appelle \bar{t} le vecteur translation qui correspond finalement aux coordonnées de l'origine du repère monde dans le repère caméra. On appelle également \bar{R} la matrice de rotation définissant la rotation du repère monde dans le repère caméra.

$$\begin{pmatrix} X^C \\ Y^C \\ Z^C \end{pmatrix} = \bar{R} \left(\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} - \bar{t} \right) = \bar{R} \cdot \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} - \bar{R} \cdot \bar{t} \Leftrightarrow \bar{Q}^C = \begin{pmatrix} X^C \\ Y^C \\ Z^C \\ 1 \end{pmatrix} = \begin{pmatrix} \bar{R} & -\bar{t} \\ \bar{0}^T & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = \bar{Q} \quad (12)$$

Passage du repère monde au repère des pixels

Il est finalement possible de cumuler l'ensemble de changements de repère pour établir la matrice de projection qui relie directement les coordonnées de la projection dans l'image aux coordonnées réelles du point projeté.

$$\bar{q} \sim \begin{pmatrix} f k_u & 0 & k_u x_0 & 0 \\ 0 & f k_v & k_v y_0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} \bar{R} & -\bar{t} \\ \bar{0}^T & 1 \end{pmatrix} \bar{Q} \sim \bar{P} \cdot \bar{Q} \quad (13)$$

Propriétés intrinsèques et extrinsèques

L'équation (13) est intéressante dans sa première formulation puisqu'elle met en avant deux matrices bien distinctes. La première fait intervenir uniquement les propriétés intrinsèques au système optique (distance focale, densités de pixels et centre de l'image obtenue). La seconde fait intervenir des paramètres extrinsèques puisqu'elle dépend uniquement de l'emplacement et de l'orientation du

système optique lors d'une prise de vue. On voit ainsi que la matrice de projection $\overline{\overline{P}}$ peut varier entre deux prises de vue pour un même dispositif optique si ce dernier est déplacé et/ou tourné.

Par conséquent, c'est la première matrice qui va déterminer entièrement les propriétés de la caméra et c'est la raison pour laquelle la calibration d'un système optique a pour objectif de déterminer uniquement la matrice de calibration $\overline{\overline{K}}$. Cette matrice 3×3 est définie de la manière suivante :

$$\overline{\overline{K}} = \begin{pmatrix} fk_u & 0 & k_u x_o \\ 0 & fk_v & k_v y_o \\ 0 & 0 & 1 \end{pmatrix} \quad (14)$$

Distorsion des images

Une autre étape de la calibration porte sur le calcul des coefficients de distorsion. Ces coefficients permettent de corriger la déformation subie par l'image liée aux défauts de fabrication et de montage du dispositif optique. Il existe deux types de distorsion :

radiale : Liée essentiellement à des défauts de symétries dans les lentilles qui constituent l'objectif.

La distorsion radiale peut être décrite de la manière suivante, r étant un rayon depuis le centre de l'image :

$$\begin{cases} x \rightarrow x(k_1 r^2 + k_2 r^4 + k_3 r^6 + \dots) \\ y \rightarrow y(k_1 r^2 + k_2 r^4 + k_3 r^6 + \dots) \end{cases} \quad (15)$$

tangentielle : Liée essentiellement à des défauts d'alignement et de parallélisme des éléments du système optique. La distorsion tangentielle est décrite par les expressions suivantes :

$$\begin{cases} x \rightarrow x + (2p_1 xy + p_2(r^2 + 2x^2)) \\ y \rightarrow y + (p_1(r^2 + 2y^2) + 2p_2 xy) \end{cases} \quad (16)$$

Les coefficients k_1, k_2, k_3, p_1 et p_2 sont donc les coefficients recherchés. Une fonction inverse peut alors être calculée afin de permettre la correction des distorsions.

B Détection automatique de la présence de plis

L'objectif principal du projet est de détecter l'apparition de plis pendant la phase de couchage du papier. De plus, l'un des objectifs finaux du projet est d'être capable de caractériser de manière rapide les plis qui apparaissent. Afin de réduire la puissance de calcul nécessaire au traitement d'images et de réduire les délais de traitement, il est primordial, dans la mesure du possible, de détecter la présence d'un pli avant tout traitement sur l'image.

La description de l'objectif qui vient d'être mentionné nous emmène à traiter un problème de classification d'images. Pour être plus précis, il semblerait qu'une classification binaire soit suffisante : soit l'image présente un pli, soit elle n'en présente pas. Afin de le savoir, il faut être capable de le prédire grâce à un modèle pré-existant. Dans le cas où le modèle n'est pas simple à définir, il peut être avantageux d'utiliser un modèle pré-entraîné par une "intelligence artificielle" (machine learning). Deux cas se posent à nous : il est possible de décrire efficacement l'image pour permettre à un classifieur de déterminer la classe de l'image (classification par machine learning) ou alors on laisse l'algorithme trouver les éléments nécessaires à la classification par lui-même (deep learning).

Description et classification avec le Machine Learning

Dans le cas d'une classification intelligente, il est important de donner une description précise de l'image afin de donner suffisamment d'informations pour permettre au modèle de trouver une relation entre les entrées (la description de l'image) et la classe associée. Ces informations descriptives sont appelées généralement "features" dans la littérature. La phase de classification des images se fait selon trois phases successives qui sont décrites ci-dessous.

- Détection ou sélection des features : il s'agit généralement de la phase la plus importante car c'est en sélectionnant les features les plus adaptées que les descriptions seront les plus pertinentes.
- Description des features : il s'agit de décrire par un moyen mathématique la feature en question pour pouvoir y faire des comparaisons avec les features des images à classer.
- Classification : il s'agit de choisir la classe de l'image en fonction des features qui ont été trouvées ou sélectionnées.

Il existe plusieurs méthodes numériques pour chacune de ces phases. Il en est une qui est relativement populaire dans le domaine de la reconnaissance d'images : il s'agit de l'algorithme SIFT (Scale Invariant Feature Transform) [2]. Cet algorithme rend (semi-)automatique les tâches de détection et description des features et présente l'avantage de ne pas être dépendant des échelles et des rotations d'images pour les comparer entre elles. Plusieurs variantes de la méthode existent afin de l'optimiser et de la rendre plus rapide [3, 4], par exemple pour des applications en poursuite vidéo (video tracking) [5].

Algorithme SIFT : détection de features - L'ensemble des tâches de détection des features est réalisé dans l'espace des échelles : il s'agit de l'espace à 3 dimensions constitué des deux dimensions de l'image (x et y) et du facteur d'échelle σ (qui correspond à la convolution de l'image par un filtre gaussien de variance σ). Afin d'assurer une invariance du facteur d'échelle, il est également nécessaire de considérer un même espace des échelles pour une même image dont les dimensions varient. La figure 12 donne une illustration de cet espace de travail. Chacune des lignes correspond à une octave. La variation de taille de l'image et l'importance du flou gaussien déterminent la taille du feature en négligeant les détails qui sont de taille inférieure à σ_i .

Pour chaque octave, la différence de chaque images successives est calculée pour former les différences de gaussiennes (DoG : Differences of Gaussian). La DoG issue d'images ayant des flous gaussiens σ_i et σ_{i+1} contient alors uniquement les détails de l'image originale dont l'échelle varie entre σ_i et σ_{i+1} . La figure 13 donne une illustration des DoG obtenues à partir des images de la figure 12.

Enfin, la dernière étape consiste à trouver les points d'intérêt de l'image grâce aux DoG obtenues précédemment. Il s'agit en fait d'une recherche d'extrema locaux dans l'espace des échelles. Afin de permettre cela, les DoG obtenues successivement sont placées en couche et les pixels qui ont une intensité plus petite ou plus grande que ses 26 voisins sont considérés comme des points d'intérêt. La figure 14 illustre la recherche d'un point d'intérêt. A l'issue de cette recherche, de trop nombreux

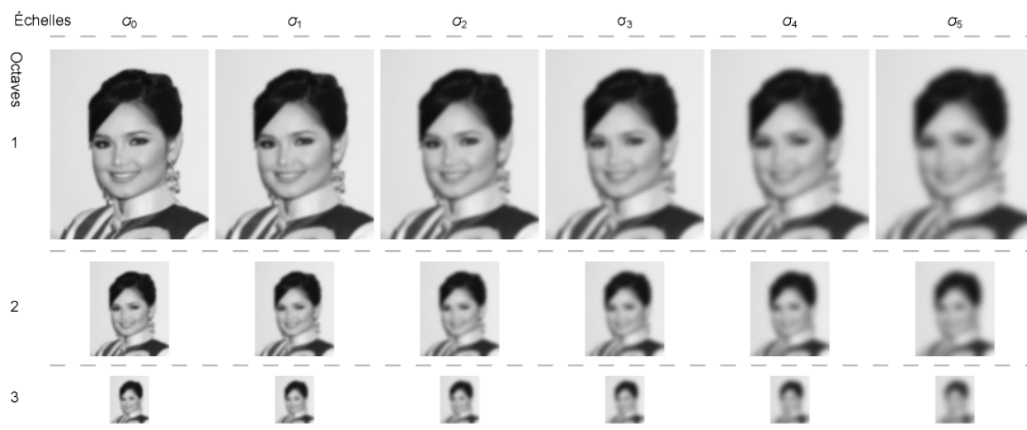


FIGURE 12 – Espace de travail pour la détection des features avec l’algorithme SIFT. Illustration issue de [6].

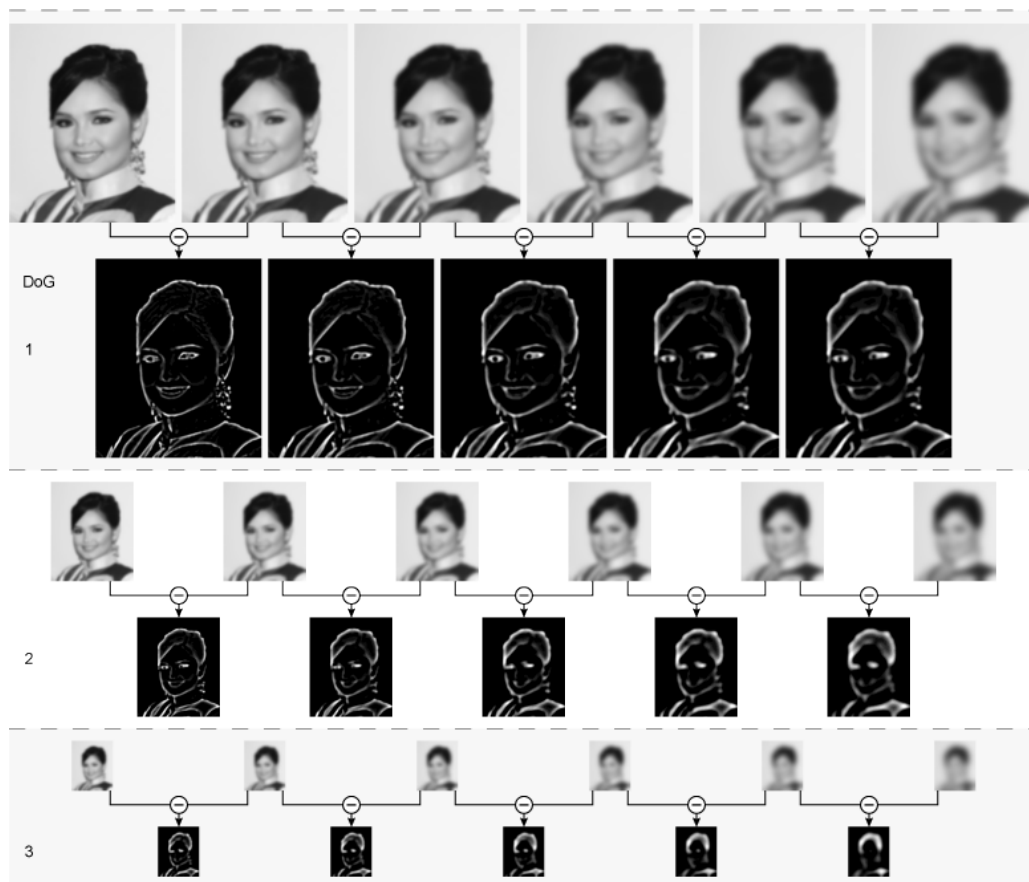


FIGURE 13 – Différences des gaussiennes. Illustration issue de [6].

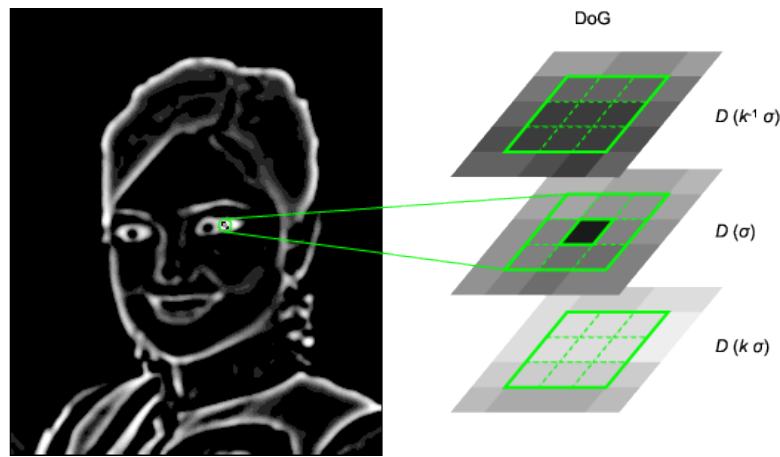


FIGURE 14 – Recherche des points d'intérêt. Illustration issue de [6].

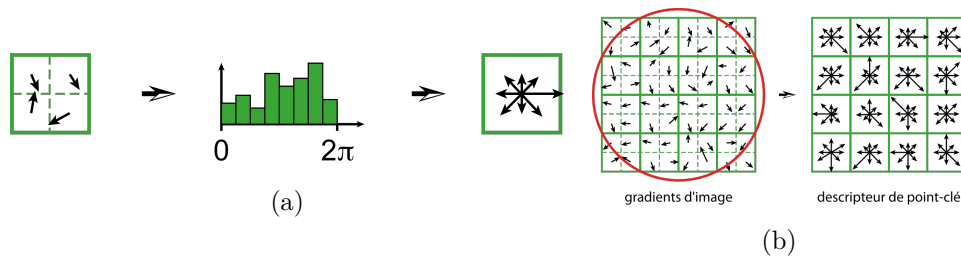


FIGURE 15 – Méthode de calcul des descripteurs avec la méthode SIFT. Illustration issue de [6].

points sont sélectionnés, une étape supplémentaire de tri est réalisé par l'intermédiaire de masques et seuillages par hystérésis.

Algorithme SIFT : description des features - Les features sont centrés sur les points d'intérêt et ont une taille caractéristique propre au flou gaussien en question. Une fois détectés, il est nécessaire de les décrire mathématiquement pour pouvoir les comparer. Pour accomplir cette tâche, une assignation de l'orientation et un descripteur sont calculés automatiquement de la manière présentée ci-dessous.

Le calcul de l'orientation se fait à partir du calcul du gradient de l'image dans le voisinage du centre d'intérêt (convolution de l'image avec un filtre de Sobel). L'ensemble des orientations obtenues est interposée sur un histogramme à 36 valeurs (pas d'orientation de 10°) et l'orientation choisie correspond au pic dans cet histogramme, lissé auparavant. Si l'histogramme présente plusieurs pics alors le descripteur est copié et plusieurs orientations sont données pour un même point d'intérêt. La contribution de chaque pixel est proportionnelle à l'amplitude de son gradient mais aussi à la proximité avec le point d'intérêt.

Vient désormais le calcul du descripteur qui est illustré sur la figure 15. On considère dans un premier temps un fenêtre carrée de taille $12\sigma \times 12\sigma$ pixels centrée autour du point d'intérêt et proportionnelle à son niveau de flou de détection σ . Cette fenêtre est découpée en 16 petites fenêtres de taille $3\sigma \times 3\sigma$ pixels. Pour chaque petite fenêtre, un histogramme de l'orientation du gradient est réalisé comme précédemment sur 8 valeurs, à la différence que maintenant l'orientation du point d'intérêt est soustrait. Pour un point d'intérêt on obtient donc 16 histogrammes de 8 valeurs. Ces histogrammes sont concaténés de manière à ne former qu'un seul vecteur de 128 valeurs. Afin de rendre le descripteur invariant au changement de contraste le vecteur est normalisé. Toutes les valeurs supérieures à 0.2 sont remplacées par 0.2 de manière à rendre le descripteur insensible aux changements de luminosité également. Enfin, une dernière normalisation vient générer le descripteur du point d'intérêt analysé.

Classification des images : machine learning - Après avoir déterminé une liste de descripteurs pour l'image, il faut maintenant savoir les utiliser et les analyser grâce à un classifieur entraîné sur des

images déjà classées manuellement. Il existe plusieurs moyens de classer les images, entre autres : régression logistique, SVM (Support Vector Machine), k-plus proches voisins, forêt d'arbres aléatoires ou encore réseau de neurones. Le principe reste relativement le même pour chacune de ces méthodes : des relations plus ou moins complexes sont pondérées dans la phase d'apprentissage. Lorsque l'on souhaite faire une prédiction, ces relations pondérées sont utilisées pour déterminer la meilleure probabilité du résultat.

Apprentissage automatique avec le Deep Learning

Les réseaux de neurones convolutifs sont très largement utilisés dans le domaine de la vision des machines. En effet, ces derniers permettent, lorsqu'ils présentent une solution de perceptrons multicouches, d'établir des relations très complexes mais surtout de retrouver automatiquement les features utiles à la classification lors de la phase d'apprentissage. Cette solution est cependant plus lourde à mettre en œuvre puisqu'elle nécessite généralement un très long entraînement avec de nombreuses images. On peut cependant envisager de générer une multitude d'images simulées avec le logiciel Blender afin d'entraîner le réseau de neurones.