

# Développement et validation d'une méthode de comptage de lamelles d'une turbine à gaz

par

Imene BELKACEMI

MÉMOIRE PRÉSENTÉ À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE  
COMME EXIGENCE PARTIELLE À L'OBTENTION DE LA MAÎTRISE  
AVEC MÉMOIRE  
M. Sc. A.

MONTREAL, LE "22 SEPTEMBRE 2021"

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE  
UNIVERSITÉ DU QUÉBEC



Imene Belkacemi, 2021



Cette licence Creative Commons signifie qu'il est permis de diffuser, d'imprimer ou de sauvegarder sur un autre support une partie ou la totalité de cette oeuvre à condition de mentionner l'auteur, que ces utilisations soient faites à des fins non commerciales et que le contenu de l'oeuvre n'ait pas été modifié.

**PRÉSENTATION DU JURY**

CE MÉMOIRE A ÉTÉ ÉVALUÉ

PAR UN JURY COMPOSÉ DE:

M. Carlos Vázquez, directeur de mémoire

Département de génie logiciel et des technologies de l'information à l'École de technologie supérieure

M. David Labbé, co-directeur

Département de génie logiciel et des technologies de l'information à l'École de technologie supérieure

M. Souheil-Antoine Tahan, président du jury

Département de génie mécanique à l'École de technologie supérieure

M. Stéphane Coulombe, membre du jury

Département de génie logiciel et des technologies de l'information à l'École de technologie supérieure

IL A FAIT L'OBJET D'UNE SOUTENANCE DEVANT JURY ET PUBLIC

LE "10 SEPTEMBRE 2021"

À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE



## REMERCIEMENTS

Tout d'abord, je souhaite remercier mon directeur de mémoire Monsieur Carlos Vázquez ainsi que mon co-directeur Monsieur David Labbé qui m'ont permis de bénéficier de leur encadrement, se sont toujours montré à l'écoute et très disponible tout au long de la réalisation de ce projet, ainsi que pour l'inspiration, la patience, l'aide et le temps qu'ils ont bien voulu me consacrer.

Mes remerciements s'étendent à Monsieur Nicolas Hardy, Madame Budeanu Nausica et Monsieur Edgeley Bryn pour leur disponibilité et leur aide précieuse à Siemens.

Je tiens à remercier les membres du jury pour l'intérêt qu'ils ont apporté pour examiner et juger mon travail et leurs propositions qui viendront l'enrichir.

Mes vifs remerciements aux membres de ma famille, particulièrement mes parents, pour leurs sacrifices et pour le soutien moral qu'ils m'ont prodigué durant les années des études.

Je remercie très chaleureusement mon mari pour son écoute, sa présence, son soutien moral, ses encouragements et sa patience ainsi que la grande aide et le temps qu'il a bien voulu me consacrer tout au long de la réalisation de ce projet.

Enfin, je remercie tous ceux qui, d'une manière ou d'une autre, ont aidé à la réussite de ce travail et qui n'ont pas pu être cités ici.



# Développement et validation d'une méthode de comptage de lamelles d'une turbine à gaz

Imene BELKACEMI

## RÉSUMÉ

Les turbines à gaz sont des moteurs qui génèrent de l'énergie mécanique en compressant et brûlant de l'air ambiant. Cette énergie mécanique peut être utilisée dans plusieurs domaines comme la propulsion des avions. L'inspection des turbines à gaz est réalisée à Siemens Canada. Elle permet de vérifier la qualité des turbines à gaz nouvellement construites et de déterminer le niveau de maintenance requis pour les turbines à gaz usées. Avec le temps, celles-ci peuvent subir des dommages à l'intérieur pouvant nuire à leur bon fonctionnement. L'inspection est effectuée en utilisant un boroscope qui permet de vérifier l'état interne du moteur sans le démonter. Le boroscope est constitué d'une sonde munie d'une caméra, d'une source de lumière et d'un écran sur lequel les images acquises sont affichées. Pour accéder à la partie du moteur à inspecter, l'opérateur injecte la sonde dans le trou approprié présent sur la surface externe du moteur, manipule la sonde et oriente la caméra en se basant sur ses propres connaissances de la composition du moteur et en se référant aux images acquises dans le but de bien visualiser la zone cible. Les vidéos obtenues contiennent principalement des lamelles qui passent, l'une après l'autre, jusqu'à la fin de la partie du moteur sous inspection. L'inspection des turbines est composée de trois étapes effectuées par l'inspecteur en parallèle et en temps réel. Ce sont : (1) vérifier la qualité des images acquises, (2) compter les lamelles qui passent manuellement pour s'assurer de ne pas en manquer et de ne pas en vérifier plus qu'il faut, (3) détecter les défauts présents sur les composants internes du moteur. En fait, c'est difficile de savoir quand toutes les lamelles ont été inspectées et de ne pas en inspecter plus car la référence de la lamelle par laquelle l'inspection commence n'est pas absolue, les lamelles ne sont pas numérotées. Du coup, il faut compter manuellement jusqu'à la dernière lamelle de la partie du moteur sous inspection.

Le but de ce projet est de proposer une solution informatique qui libère l'inspecteur des deux premières tâches, à savoir : la qualification des images et le comptage des lamelles afin qu'il puisse se concentrer sur la détection des défauts. Cette solution, à travers une interface graphique, indiquera à l'utilisateur si les images capturées sont de bonne qualité à travers des métriques calculées. Aussi, elle contiendra un compteur de lamelles automatique, affichant le nombre de lamelles qui passent sur l'écran en temps réel. Pour cela, plusieurs traitements sur les images acquises doivent être effectués.

L'automatisation de la qualification des images et du comptage des lamelles libère l'inspecteur de ces deux tâches et le laisse se concentrer sur la partie la plus importante du point de vue cognitif qui est la détection des défauts. Ceci permettra un gain de productivité et une plus grande assurance de la qualité de l'inspection pour Siemens Canada.

**Mots-clés:** turbines à gaz, inspection par boroscope, qualification des images, comptage de lamelles



## **Development and validation of a gas turbine blade counting method**

Imene BELKACEMI

### **ABSTRACT**

Gas turbines are engines that generate mechanical energy by compressing and burning ambient air. This mechanical energy can be used in several fields such as the generation of electrical energy and the propulsion of airplanes. Inspection of gas turbines is performed at Siemens Canada. It verifies the compliance of newly built gas turbines with quality standards and determines the level of maintenance required for used gas turbines. Over time, these can suffer internal damage that could affect their proper functioning. The inspection is carried out using a borescope which allows the internal condition of the motor to be checked without dismantling it. The borescope consists of a probe fitted with a camera, a light source and a screen on which the acquired images are displayed. To access the part of the engine to be inspected, the operator injects the probe into the appropriate hole present on the external surface of the engine, manipulates the probe and orients the camera based on his own knowledge of the composition of the engine and based on his knowledge of the composition of the engine referring to the images acquired in order to visualize well at the target area. The resulting videos mainly contain blades that pass, one after the other, until the end of the part of the engine under inspection. Turbine inspection is made up of three stages carried out by the inspector in parallel and in real time. These are : (1) checking the quality of the images acquired, (2) counting the slides which pass manually to make sure not to run out and not to check more than necessary, (3) to detect the defects present on the internal engine components. In fact, it is difficult to know when all the blades have been inspected and not to inspect more because the reference of the blade that the inspection begins with is not absolute, the blades are not numbered. As a result, it is necessary to manually count until the last blade of the part of the engine under inspection.

The goal of this project is to provide an IT solution that frees the inspector from the first two tasks, namely : image qualification and blade counting so that he can concentrate on defect detection. This solution, through a user interface, will indicate to the user if the captured images are of good quality through calculated metrics. Also, it will contain an automatic blade counter, displaying the number of blades which pass on the screen in real time. To do this, several treatments on the acquired images will be carried out.

Automating image qualification and blade counting frees the inspector from these two tasks and lets them focus on the cognitively most important part of defect detection. This will lead to increased productivity and greater inspection quality assurance for Siemens Canada.

**Keywords:** gas turbine, borescope inspection, qualification of images, blade counting



## TABLE DES MATIÈRES

	Page
INTRODUCTION .....	1
CHAPITRE 1 CONTEXTE ET PROBLÉMATIQUE .....	5
1.1 Turbines à gaz .....	5
1.2 Inspection endoscopique .....	7
1.3 Déroulement de l'inspection .....	9
1.4 Défis reliés à l'inspection .....	12
1.5 Besoins techniques et fonctionnalités désirées .....	14
1.6 Requis essentiels et contraintes .....	16
1.7 Conclusion .....	16
CHAPITRE 2 REVUE DE LITTÉRATURE .....	19
2.1 Introduction .....	19
2.2 Elimination du bruit .....	19
2.2.1 Filtres dans le domaine spatial .....	20
2.2.1.1 Filtres locaux .....	20
2.2.1.2 Filtres non locaux .....	22
2.2.2 Filtres dans le domaine de transformation .....	23
2.2.3 Comparaison et discussion .....	25
2.3 Détection de contours .....	27
2.3.1 Opérateur Roberts .....	28
2.3.2 Opérateur Sobel .....	29
2.3.3 Opérateur Prewitt .....	29
2.3.4 Laplacien de Gaussien .....	29
2.3.5 Opérateur Canny .....	30
2.3.6 Comparaison et discussion .....	31
2.4 Détection du mouvement .....	33
2.4.1 Théorie du flux optique .....	33
2.4.2 Méthode Farneback .....	35
2.4.3 Méthode de Lucas-Kanade pyramidale .....	38
2.4.4 Comparaison et discussion .....	40
2.5 Conclusion .....	41
CHAPITRE 3 MÉTHODOLOGIE .....	43
3.1 Introduction .....	43
3.2 Situation de départ .....	43
3.2.1 Description globale de la solution existante .....	44
3.2.2 Fonctionnement de la solution existante .....	45
3.2.2.1 Qualification des images .....	47
3.2.2.2 Compteur de lamelles .....	50

3.2.3	Exécution et tests de la solution existante .....	53
3.2.3.1	Méthodes d'exécution de la solution .....	54
3.2.3.2	Tests et observations .....	55
3.3	Résolution des problèmes principaux avec la solution existante .....	57
3.3.1	L'exception " <i>Out Of Memory</i> " .....	58
3.3.2	Temps de calcul très long .....	58
3.3.3	Traitement de la même trame deux fois .....	59
3.3.4	Capture des trames en temps réel .....	60
3.4	Objectifs et matériel requis .....	60
3.5	Solution proposée .....	61
3.5.1	Qualification des images .....	62
3.5.1.1	Luminosité .....	62
3.5.1.2	Netteté .....	62
3.5.1.3	Contraste .....	63
3.5.1.4	Vitesse du mouvement .....	64
3.5.2	Compteur de lamelles .....	67
3.6	Présentation de l'application .....	74
3.7	Conclusion .....	76
CHAPITRE 4 TESTS, RÉSULTATS ET DISCUSSION .....		79
4.1	Introduction .....	79
4.2	Base de données des vidéos .....	79
4.3	Métriques de qualification des images .....	80
4.3.1	Tests unitaires .....	81
4.3.1.1	Netteté .....	81
4.3.1.2	Luminosité .....	82
4.3.1.3	Contraste .....	83
4.3.1.4	Vitesse .....	84
4.3.2	Tests avec les vidéos disponibles .....	84
4.4	Compteur de lamelles .....	85
4.5	Procédure de mise en place du carré .....	86
4.5.1	Tests et résultats obtenus .....	87
4.6	Test du logiciel à Siemens .....	92
4.7	Tableau récapitulatif et résultats du compteur .....	95
4.8	Analyse des résultats et discussion .....	101
4.8.1	Métriques de qualification des images .....	101
4.8.2	Compteur de lamelles .....	101
4.8.3	Procédure de sélection d'une bonne position .....	103
4.9	Conclusion .....	104
CONCLUSION ET PERSPECTIVES .....		105
ANNEXE I DIAGRAMME D'ACTIVITÉ DÉCRIVANT LA SOLUTION EXISTANTE .....		109

ANNEXE II LUMIÈRE FORTE DANS LA FIGURE ..... 111

ANNEXE III FORMULAIRE D'INFORMATION ET DE CONSENTEMENT ..... 113

BIBLIOGRAPHIE ..... 119



## LISTE DES TABLEAUX

	Page
Tableau 3.1	Comparaison du temps de calcul nécessaire pour le traitement de deux trames successives (CPU vs GPU) ..... 59
Tableau 4.1	Netteté obtenue pour les images modifiées ..... 81
Tableau 4.2	Luminosité obtenue pour les images modifiées ..... 83
Tableau 4.3	Contraste obtenu pour les images modifiées ..... 83
Tableau 4.4	Résultat de l'étude concernant la sélection d'une bonne position ..... 92
Tableau 4.6	Étages inspectés lors de la démonstration à Siemens ..... 94
Tableau 4.7	Tableau récapitulatif des tests des métriques de qualification des images ainsi que le compteur de lamelles ..... 96
Tableau 4.8	Résultats des tests effectués pour le compteur de lamelles ..... 102
Tableau 4.9	Ecart-type entre les positions choisies par les participants pour chaque vidéo ..... 104



## LISTE DES FIGURES

		Page
Figure 1.1	Turbine à gaz : SGT-A35 .....	5
Figure 1.2	Section d'un moteur à turbine à gaz avec les parties : compresseur (a), chambre de combustion (b) et turbine (c) Tirée de (Eliaz, Shemesh & Latanision, 2002) .....	6
Figure 1.3	Lamelles Tirée de (Chen, 2020) .....	6
Figure 1.4	Boroscope IPLEX GX/GT .....	8
Figure 1.5	Articulation de la sonde Tirée de (Olympus, 2021a) .....	9
Figure 1.6	Compteur manuel utilisé lors des inspections Tirée de (Nicolas, 2016) .....	11
Figure 2.1	Image Lena .....	26
Figure 2.2	Image Lena contenant différents types de bruit et filtrée en utilisant différents filtres spatiaux Tirée de (Singh & Neeru, 2014) .....	26
Figure 2.3	Les deux images originales utilisées Tirée de (Shah, Kedia, Raut, Ansari & Shroff, 2020) .....	32
Figure 2.4	Image (1) filtrée avec les filtres : Rebirths, Sobel, Prewitt, Laplacien et Canny dans l'ordre Tirée de (Shah <i>et al.</i> , 2020) .....	32
Figure 2.5	Polynômes de voisinage sur deux images consécutives, $d$ est le déplacement Tirée de (Husseini, 2017) .....	36
Figure 2.6	Pyramide d'images à quatre niveaux. À chaque niveau, l'image est réduite. Le calcul du flux optique commence au sommet de la pyramide (niveau 4) et se termine en bas (niveau 0) Tirée de (Husseini, 2017) .....	38
Figure 2.7	L'image de gauche est un exemple d'image de séquence extérieure et l'image de droite est une séquence d'image intérieure Tirée de (Husseini, 2017) .....	40
Figure 3.1	Matériel requis pour l'intégration du logiciel au processus d'inspection .....	44
Figure 3.2	Interface graphique de la solution existante .....	45

Figure 3.3	Diagramme de séquence global de la solution existante .....	47
Figure 3.4	Définir la zone du mouvement .....	51
Figure 3.5	Différence d'intensité due au passage d'une lamelle .....	53
Figure 3.6	Fonctionnement global de la solution existante .....	53
Figure 3.7	Simulation de fonctionnement en temps réel sans boroscope .....	54
Figure 3.8	Carré du calcul d'intensité différent entre deux exécutions (chaque ligne montre une exécution) .....	57
Figure 3.9	Fonctionnement global de la solution proposée .....	62
Figure 3.10	Points à suivre repérés dans la première trame .....	65
Figure 3.11	Bons et mauvais vecteurs .....	65
Figure 3.12	Images prises de différents moteurs, les lamelles principales sont encadrées en rouge .....	68
Figure 3.13	Une seule lamelle qui apparaît dans l'image .....	69
Figure 3.14	Une seule lamelle avec bordures supérieures, inférieures ou latérales (zones à éviter) .....	69
Figure 3.15	Plusieurs lamelles apparaissent, exemples de bonne zone .....	70
Figure 3.16	Trame originale (gauche) lissée avec le filtre bilatéral (droite) .....	72
Figure 3.17	Bordures extraites à partir de la trame lissée avec le filtre Canny .....	72
Figure 3.18	Piques d'intensité représentant le passage de six lamelles par une position fixée .....	73
Figure 3.19	Interface graphique améliorée .....	75
Figure 4.1	Différence entre la trame enregistrée avec la caméra (gauche) et la même trame ré-enregistrée après injection dans le module d'acquisition vidéo (droite) .....	80
Figure 4.2	Les trois images choisies pour les tests unitaires des métriques netteté, luminosité et contraste (dans l'ordre trame 1, trame 2 et trame 3) .....	81

Figure 4.3	Images floutée obtenues à partir des trame 1, 2 et 3. Dans chaque ligne, on trouve l'image originale, $\sigma = 0.5, 1, 2, 3, 4$ ..... 82	82
Figure 4.4	Images obtenues à partir de la trame 1. L'intensité rajoutée ou soustraite est mentionnée sous chaque image ..... 82	82
Figure 4.5	Modification du contraste de la trame 1. Les intervalles sont mentionnés sous chaque image ..... 84	84
Figure 4.6	Détection du passage d'une bordure par le carré fixé ..... 86	86
Figure 4.7	Positions choisies pour les deux premières vidéos ..... 88	88
Figure 4.8	Positions choisies pour les vidéos trois et quatre ..... 89	89
Figure 4.9	Positions choisies pour les vidéos cinq et six ..... 89	89
Figure 4.10	Positions choisies pour les vidéos sept et huit ..... 90	90
Figure 4.11	Positions choisies pour les vidéos neuf et dix ..... 91	91



## **LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES**

SGT	Siemens Gaz Turbine Aero derivative
FOD	Foreign Object Damages
DOD	Domestic Object Damage
fps	Frame Per Second
CPU	Central Processing Unit
GPU	Graphics Processing Unit



## INTRODUCTION

Siemens Canada fait l'assemblage, la révision et la réparation de plusieurs types de turbines à gaz. Ce sont des moteurs qui génèrent de l'énergie mécanique en compressant et brûlant de l'air ambiant. Cette énergie mécanique peut être utilisée dans plusieurs domaines entre autres la génération de l'énergie électrique, l'industrie du pétrole et du gaz ainsi que la propulsion des avions, des trains et des navires.

Les turbines usagées peuvent être abimées avec le temps. L'usure peut générer des dommages à l'intérieur des turbines nuisant au bon fonctionnement de celles-ci. Ces dommages sont généralement causés par les particules qui peuvent se présenter dans l'air admis dans le moteur (comme le sable), le sur-chauffage ou les réactions chimiques qui peuvent se produire à l'intérieur du moteur. D'où la nécessité d'effectuer des inspections d'évaluation de l'état interne des moteurs usagés périodiquement. C'est une étape critique pour déterminer le niveau de maintenance requis. Cette inspection d'évaluation est aussi importante pour s'assurer que les turbines à gaz nouvellement construites sont conformes aux normes de qualité.

Pour vérifier l'état d'une turbine à gaz, l'opérateur effectue une inspection endoscopique en utilisant un boroscope qui permet de voir l'intérieur de la turbine sans avoir à la démonter. En gros, il s'agit d'injecter une sonde munie d'une caméra et d'une source de lumière à travers des ports d'accès appropriés présents sur la surface externe du moteur pour accéder à l'étage voulu. Le moteur est composé de plusieurs étages, chacun est constitué de plusieurs lamelles (voir la figure 1.3). En tournant cet étage, les images capturées et affichées sur l'écran du boroscope montrent des lamelles qui passent, l'une après l'autre, jusqu'à la fin de l'étage. Cette fin correspond à l'atteinte du nombre total des lamelles qui composent l'étage en question. Pour le savoir, l'inspecteur compte les lamelles qui passent en utilisant un compteur manuel tenu par la main et se réfère à une feuille où les informations concernant l'étage sont inscrites. Les images acquises sont analysées, en temps réel, par l'inspecteur qui doit s'assurer que toutes les

composantes du moteur sont intactes. Le comptage manuel des lamelles peut distraire l'attention de l'inspecteur de la tâche la plus importante qui est la détection des défauts.

Cela nécessite une main-d'œuvre qualifiée difficile à former et à trouver. L'inspecteur doit avoir une bonne connaissance de la composition interne du moteur ainsi que des accessoires et outils qu'il doit manipuler pour acquérir de bonnes images qui lui permettront de juger l'état interne du moteur. En effet, les images acquises doivent être de bonne qualité pour garantir que les défauts y apparaissent clairement. Dans ce cas, l'inspection se déroule dans des conditions optimales. La qualité et la fiabilité des rapports finaux dépend de l'expertise des inspecteurs. Les inspecteurs juniors ont tendance à affronter beaucoup de difficultés qui mèneront au moins à une perte de temps, le pire est la non signalisation de dommages. Un mauvais diagnostic dans cette phase peut entraîner des coûts et des retards importants pour Siemens (Nicolas, 2016).

Une solution a été développée et fournie à Siemens dans le but de libérer l'inspecteur des tâches routinières et le laisser se concentrer sur la détection des défauts étant la tâche la plus stimulante et qui nécessite la concentration et la prise de décision humaine. Cette solution n'était pas utilisée car elle n'était pas fonctionnelle.

A travers ce projet, nous visons à assurer l'automatisation des tâches routinières de l'inspection par boroscope des turbines à gaz dont l'assemblage se fait à Siemens Canada. D'une part, il faut garantir que les inspections se déroulent dans des conditions optimales. Pour cela, il faut s'assurer que les images capturées par les inspecteurs sont de bonne qualité. En effet, plusieurs paramètres sont gérables par les inspecteurs lors de l'acquisition des vidéos, comme la luminosité projetée dans le moteur et la propreté des lentilles utilisées. Quatre métriques de qualification des images seront calculées tout au long de l'inspection et affichées à l'inspecteur avec un code couleur, en temps réel. Ces métriques reflètent la qualité des images capturées par les inspecteurs et les guident à acquérir des images de bonne qualité dans lesquelles les défauts apparaissent clairement.

D'autre part, il s'agit de libérer l'inspecteur du comptage manuel des lamelles dans le but d'éviter la sous et la sur inspection. Avant de commencer le comptage des lamelles, une position par laquelle uniquement la bordure principale de la lamelle passe doit être sélectionnée par l'inspecteur. Quand la lamelle passe par cette position, le compteur automatique est incrémenté et affiché en temps réel. A travers ces automatisations, les inspections seront guidées et le processus sera simplifié. Ceci pourra réduire le temps d'inspection et unifiera la qualité des vidéos acquises par les inspecteurs, même avec moins d'expertise.

Le premier chapitre de ce mémoire détaille le contexte et la problématique. Le second chapitre présente une revue de littérature des algorithmes et méthodes nécessaires pour la réalisation des deux parties de la solution. Le troisième chapitre détaille la méthodologie et la solution proposée. Les résultats obtenus ainsi qu'une discussion sont présentés dans le chapitre suivant. Les perspectives sont présentées à la conclusion générale de ce manuscrit.



# CHAPITRE 1

## CONTEXTE ET PROBLÉMATIQUE

### 1.1 Turbines à gaz

Une turbine à gaz est une « Machine rotative transformant l'énergie thermique en énergie mécanique, comprenant un ou plusieurs compresseur(s), un ou plusieurs dispositif(s) thermique(s) destiné(s) à réchauffer le fluide moteur, une ou plusieurs turbine(s), un système de régulation et un équipement auxiliaire » (ISO, 1996). Aussi appelée turbine à combustion, elle est capable de convertir du gaz naturel ou d'autres combustibles liquides en énergie mécanique (GEPower, 2021). La figure suivante montre la turbine à gaz SGT-A35 (SiemensEnergy, 2021).

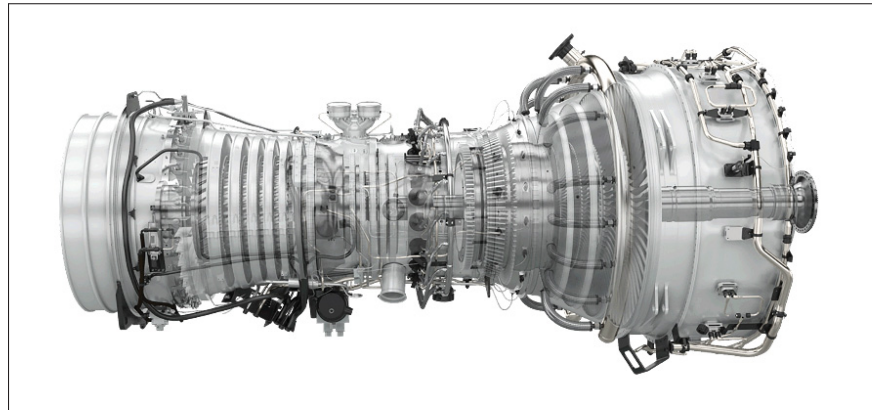


FIGURE 1.1 Turbine à gaz : SGT-A35

Dans son architecture la plus simple, une turbine à gaz est composée de trois éléments : un compresseur, une chambre à combustion et une turbine. La figure 1.2 montre une coupe d'une turbine à gaz. L'air admis de l'extérieur est dirigé vers le compresseur du moteur (a). Le rôle de ce dernier est de comprimer l'air avant d'ajouter du carburant. Dans un compresseur à écoulement axial, l'air s'écoule dans la direction de l'axe de l'arbre à travers des rangées alternées de lamelles fixes et rotatives appelés *stators* et *rotors* respectivement. Chaque ensemble de ces lamelles constitue un étage (voir la figure 1.3). Dans la chambre de combustion (b), un allumeur électrique est utilisé pour déclencher la combustion qui consiste à brûler l'air comprimé mélangé

avec du carburant. La chambre de combustion ajoute de l'énergie thermique au flux d'air et augmente sa température. En quittant la chambre de combustion, l'échappement chaud passe à travers la turbine (c), dans laquelle les gaz sont partiellement détendus à travers des rangées alternées de lamelles. Selon le type de moteur, la turbine peut être constituée d'un ou plusieurs étages. Les gaz, quittant la turbine à une pression intermédiaire, sont finalement accélérés par une buse pour atteindre la vitesse de sortie de jet élevée souhaitée (Eliaz *et al.*, 2002).

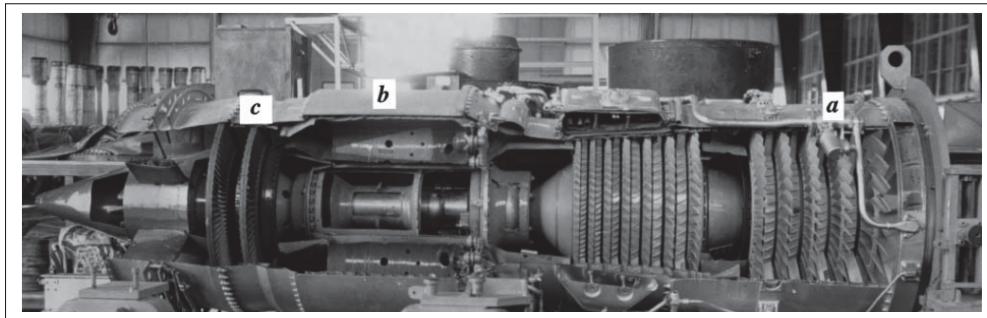


FIGURE 1.2 Section d'un moteur à turbine à gaz avec les parties : compresseur (a), chambre de combustion (b) et turbine (c)

Tirée de (Eliaz *et al.*, 2002)



FIGURE 1.3 Lamelles

Tirée de (Chen, 2020)

Dans le domaine de la propulsion aéronautique, la poussée produite (à cause de la vitesse

de sortie qui est supérieure à la vitesse du courant libre) est utilisée pour accélérer l'avion dans le sens contraire de la poussée (selon la troisième loi de Newton) (Eliaz *et al.*, 2002). L'énergie mécanique générée peut être utilisée pour générer de l'énergie électrique par le biais d'un générateur électrique (GEPower, 2021). Les turbines à gaz sont aussi utilisées dans la propulsion des trains et des navires ; dans l'industrie du pétrole et du gaz ainsi que dans la production d'énergie industrielle (ciment, aliments et boissons, bâtiments et infrastructures, mines et minéraux, produits chimiques et plastique).

L'assemblage, la réparation et la révision des turbines à gaz se fait à Siemens où trois types de turbines à gaz sont pris en charge : les turbines à gaz robustes (heavy-duty), industrielles et aérodérivé (Siemens, 2021, 2019).

L'inspection des turbines à gaz est nécessaire non seulement pour s'assurer que les moteurs construits sont conformes aux normes de qualité mais aussi pour classer les moteurs reçus dans l'atelier pour les affecter au champ de travail approprié (réparation ou révision). L'inspection consiste à vérifier l'état de tous les étages du compresseur ainsi que ceux de la turbine. Pour chaque étage, il faut vérifier l'état de toutes les lamelles qui le composent car des dommages peuvent s'y présenter comme des bosses, des rayures et des fissures. Généralement, ces dommages sont causés soit par des particules étrangères lors de l'admission de l'air dans le moteur (FOD) comme le sable ou bien par des matériaux internes (DOD) comme des particules traversant l'air secondaire provenant de débris laissés dans la tuyauterie d'air lors de la fabrication. D'autres dommages peuvent se présenter sur les lamelles à cause du sur-chauffage et des réactions chimiques comme la décoloration et la corrosion.

## **1.2 Inspection endoscopique**

L'inspection endoscopique est la méthode qui permet de vérifier l'état des turbines à gaz sans avoir à les démonter, ce qui facilite le processus d'inspection et permet de sauver du temps.

Pour appliquer cette méthode, un boroscope doit être utilisé. C'est un appareil qui permet de voir dans des endroits extrêmement petits et difficiles d'accès. Les boroscopes se retrouvent dans

de nombreuses applications médicales ou industrielles (LeParisien, 2016). A Siemens, deux modèles ont été utilisés : General Electrics XLG3 qui ne va plus être utilisé et Olympus iPlex RX/RT. Ce dernier est toujours utilisé en plus de Olympus iPlex GX/GT qui va être intégré dans le futur. La figure 1.4 montre le boroscope Olympus Iplex GX/GT, le modèle qui va être présenté ci-après (Chen, 2020).



FIGURE 1.4 Boroscope IPLEX GX/GT

Le boroscope Olympus Iplex GX/GT est constitué d'un écran LCD tactile de 8 pouces et d'une sonde se terminant par une source de lumière intense DEL et une caméra. Les sondes sont de différents diamètres (4 à 6 mm) et longueurs (2 à 10 m), ce qui permet d'adapter le matériel au besoin spécifique de l'inspection.

L'appareil contient un port HDMI pour la sortie vidéo et peut être alimenté (110 à 240 V) ou fonctionner avec une batterie assurant jusqu'à 150 minutes d'autonomie. Sur l'écran, les images obtenues, des boutons de commande, la date, l'heure, le titre, l'adaptateur d'embout optique et d'autres paramètres du système sont affichés.

Le long de la sonde est flexible et se termine par une articulation électronique pouvant être orientée avec précision dans les quatre sens (haut, bas, droite et gauche) via le menu tactile ou bien avec un bouton spécifique. Cette fonctionnalité permet de trouver la zone cible en changeant

le champ de vision de la caméra (voir la figure 1.5). De plus, on peut fixer cette position grâce à la fonction de verrouillage (Olympus, 2021b).



FIGURE 1.5 Articulation de la sonde  
Tirée de (Olympus, 2021a)

Lors de l'enregistrement des vidéos, des captures d'images peuvent être prises sans interrompre l'enregistrement en cours. La résolution des images enregistrées est de  $768 \times 576$  pixels, le format des captures enregistrées est Jpeg compressé tandis que le format des vidéos enregistrées est MPEG 4 AVC/H.264 avec un taux de trames de 60 ou 30 fps (selon le modèle du boroscope). Une carte de mémoire SDHC est fournie pour l'enregistrement des vidéos, avec la possibilité d'enregistrer les captures d'images fixes dans la mémoire interne (Olympus, 2021b).

Des fonctions logicielles sont fournies avec cet appareil. On peut effectuer un zoom numérique jusqu'à 5 fois, contrôler la luminosité projetée par la source de lumière comme on peut inverser les images affichées (droite, gauche, haut et bas) à  $180^\circ$  (Olympus, 2021b).

### 1.3 Déroulement de l'inspection

Avant de commencer, l'inspecteur identifie le port d'accès approprié qui mène à l'étage qu'il va examiner sur la surface externe du moteur. Pour cela, il se réfère à une documentation technique

disponible indiquant l'emplacement des ports sur la surface du moteur et le nombre de lamelles de chaque étage. Ensuite, il choisit les accessoires du boroscope à utiliser dans son inspection comme la lentille, la sonde et le type de lumière. Par la suite, il injecte la sonde du boroscope dans le port approprié et manipule l'articulation de la sonde dans le but de trouver la meilleure position de la caméra qui lui permet de voir les lamelles clairement. Une fois cette position trouvée, il verrouille l'articulation et fixe la sonde du boroscope de l'extérieur du moteur en serrant une vice.

La prochaine étape s'agit de s'assurer que la *qualité des images* obtenues à partir de la caméra est assez bonne pour effectuer une inspection dans des conditions optimales car elle affecte sensiblement les résultats de l'inspection. La qualité des images capturées peut être dégradée à cause de plusieurs facteurs, entre autres, la luminosité projetée par la lampe du boroscope, la saleté des lentilles utilisées, l'état interne du moteur et le champ de vision. Ce dernier est défini par l'adaptateur d'extrémité choisi. A cette phase, l'inspecteur effectue les modifications nécessaires comme l'ajustement de lumière ou le changement de l'adaptateur d'extrémité pour avoir la meilleure qualité des images possible.

A ce stade, la caméra est bien positionnée, les images capturées sont de bonne qualité, l'enregistrement de la vidéo est lancé. L'inspecteur commence à faire tourner le disque de lamelles de l'extérieur avec une main en utilisant un outil spécifique et commence son inspection qui consiste à exécuter les trois tâches principales suivantes, manuellement et en temps réel :

1. Vérifier, de façon continue, la *qualité des images* capturées et affichées sur l'écran du boroscope. La qualité des images peut se dégrader à cause de la vitesse avec laquelle l'inspecteur fait tourner l'étage. Une grande vitesse peut causer un flou du mouvement qui empêche d'effectuer une inspection satisfaisante ;
2. *Compter les lamelles* à partir de l'écran du boroscope. Pratiquement, on ne peut pas avoir un point de départ connu de l'extérieur du moteur car la référence est relative à la lamelle à partir de laquelle l'inspecteur commence son inspection. Donc, dans le but d'éviter d'examiner plus ou moins de lamelles qu'il faut (sur ou sous inspection), l'inspecteur doit compter les lamelles jusqu'à ce qu'il arrive au nombre de lamelles total de l'étage qui est

connu pour tous les étages de tous les moteurs. Pour réaliser cette tâche, l'inspecteur utilise un compteur manuel tenu avec l'autre main (voir la figure 1.6). Il appuie sur un bouton dès qu'une lamelle est passée et se réfère à une feuille où le nombre de lamelles de tous les étages du moteur en cours d'inspection est inscrit dans un tableau ;

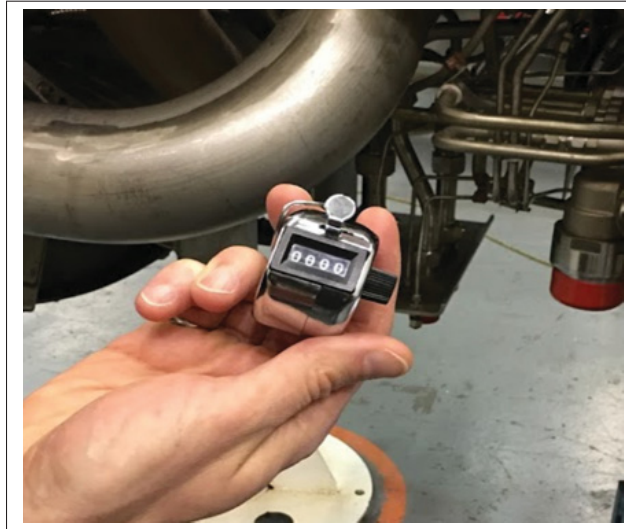


FIGURE 1.6 Compteur manuel utilisé lors des inspections  
Tirée de (Nicolas, 2016)

3. *Détecter les défauts* dans les lamelles qui passent par la caméra directement sur l'écran du boroscope. Il s'agit de vérifier l'état des bordures, la surface et le contour qui englobe l'étage. Si un défaut est détecté sur une lamelle, l'inspecteur prend une capture et continue son inspection si le défaut est minime, sinon il note le numéro de la lamelle X sur la feuille, termine le tour de l'étage puis revient à la fin pour prendre plusieurs captures du défaut détecté et le mesurer si nécessaire. Et c'est sur la même feuille qu'il écrit les résultats de son inspection à savoir : défauts détectés, leurs degrés et sur quelle lamelle pour pouvoir évaluer l'état global du moteur à la fin de l'inspection.

Pour enregistrer les vidéos et les captures lors de l'inspection, une clé de mémoire USB est branchée au boroscope avant de démarrer l'enregistrement. Les noms des fichiers et la description

doivent être inscrits manuellement à chaque fois pour refléter la zone d'inspection actuelle. La visualisation des captures enregistrées peut se faire sur l'écran du boroscope.

L'exécution de ces tâches en parallèle est primordiale pour l'inspection des turbines à gaz. En résumé, pour vérifier l'état des lamelles et détecter les défauts, il est important d'avoir une bonne qualité des images capturées, compter les lamelles pour garantir la vérification de toutes les lamelles de l'étage en plus de détecter les défauts.

Selon les exigences, une inspection complète d'un moteur peut prendre jusqu'à trois jours. A la fin de chaque inspection, un rapport est remis dans le but de montrer l'état de chaque zone du moteur inspectée par le boroscope. Ce rapport est évalué par un personnel spécialisé et discuté avec le client si des modifications sont recommandées ou requises. Les résultats sont ensuite inclus dans les rapports de réparation et de révision remis aux clients.

L'expertise de l'inspecteur et ses propres connaissances sont des facteurs clé déterminant le temps nécessaire pour une inspection de turbine et la fiabilité des rapports finaux. Les jeunes inspecteurs font face à plusieurs difficultés durant l'inspection. Par exemple, certains ont tendance à mal qualifier les images et à mal positionner la caméra ce qui peut les mener à effectuer des inspections dans des conditions sous optimales. D'autres trouvent de la difficulté durant l'injection de la sonde dans le moteur. Notons qu'un mauvais diagnostic dans cette phase peut entraîner des coûts et des retards importants pour Siemens. Dans la section suivante, les défis rencontrés par les inspecteurs sont détaillés.

#### **1.4 Défis reliés à l'inspection**

Avant d'injecter la sonde du boroscope dans le moteur, l'inspecteur doit sélectionner l'adaptateur d'extrémité optique. Ceci n'est pas documenté. Un inspecteur expérimenté utilise plusieurs adaptateurs en fonction de la section à inspecter. Par contre, un inspecteur inexpérimenté peut perdre beaucoup de temps à manœuvrer l'endoscope à l'intérieur et à l'extérieur pour changer cet accessoire (Nicolas, 2016).

Ensuite, le port menant à l'étage à inspecter doit être trouvé. Quelques étages se ressemblent et peuvent être facilement confondus par un inspecteur non expérimenté. Conduisant ainsi à une double inspection de certains étages et à en rater d'autres. Au moins, le temps d'inspection sera perdu en raison de cette confusion. Le pire est qu'un étage soit manquée par inadvertance et que les dommages passent inaperçus et ne soient pas signalés. Aussi, si le mauvais étage est inspecté, le nombre de lamelles peut être faux, ce qui peut induire à effectuer une inspection incomplète de l'étage (Nicolas, 2016).

Pendant la phase de positionnement de la caméra, l'inspecteur vise à atteindre la meilleure vue de la zone à inspecter. Pour cela, il se fie aux images obtenues de la caméra déjà injectée et à ses propres connaissances de la composition interne du moteur pour manœuvrer l'articulation du boroscope. Pratiquement, il n'y a aucun moyen de voir la position et l'orientation du bout de la sonde de l'extérieur. De plus, il n'y a pas de documentation facilement accessible pour guider un inspecteur inexpérimenté à manipuler l'articulation à l'intérieur du moteur pour atteindre la zone cible dans l'étage en question (Nicolas, 2016). D'autre part, certains étages sont larges et nécessitent plusieurs poses d'endoscope pour couvrir toute la longueur de la lamelle, ceci peut être causé par la position de la caméra (très proche des lamelles) ou du champ de vision de l'embout sélectionné (Nicolas, 2016). Dans ce cas, l'inspecteur doit déterminer si deux ou trois poses sont nécessaires. Dans ce cas, la caméra prendra plusieurs positions (à chercher avec l'articulation) et donc plusieurs vidéos vont être enregistrées pour le même étage.

En ce qui concerne la qualité des images, l'inspecteur doit garantir que l'image obtenue est la meilleur possible. Pour cela, il doit s'assurer de la propreté des lentilles utilisées et que la luminosité projetée est la meilleure. De plus, il doit faire attention à ralentir le mouvement des lamelles pour éviter l'effet du flou de mouvement. Rien n'indique si ces conditions d'acquisition d'image sont bonnes ni si la vitesse de mouvement est acceptable ou non à l'écran, ce qui revient au jugement de l'opérateur.

Lors de l'inspection, des vidéos et des captures d'images doivent être enregistrées pour une inspection approfondie ou une réévaluation ultérieure des résultats. Ces fichiers doivent refléter

avec précision la zone inspectée. Pour cela, plusieurs informations sont nécessaires, entre autres le nom du moteur et le nom de l'étage, la date et l'heure. Mais aucune dénomination standard n'est appliquée ou suggérée, c'est à la discrétion de l'inspecteur. De plus, l'affichage de l'heure et la date sur l'écran du boroscope peut être désactivée si l'opérateur juge que ça cache une partie de l'image. Ça peut augmenter le risque d'erreur pour l'identification de la zone inspectée et rend difficile la recherche dans les fichiers enregistrés dans le futur pour une post-inspection. Aussi, les données collectées ne sont pas centralisées et peuvent être assez volumineux (Go). La gestion du transfert vers le serveur local de Siemens et de l'archivage relève de la responsabilité de l'inspecteur.

### 1.5 Besoins techniques et fonctionnalités désirées

Grâce aux avancées technologiques en imagerie numérique, plusieurs tâches construisant la procédure d'inspection des turbines à gaz peuvent être automatisées. En plus de l'automatisation des tâches routinières (qualification des images et comptage de lamelles) qui réduit le multitâche, les informations requises pour identifier les parties des moteurs inspectées peuvent être unifiées. A travers ces améliorations, Siemens vise à standardiser et simplifier la procédure d'inspection, diminuer le temps nécessaire pour l'inspection et améliorer la qualité des résultats obtenus même avec une expertise limitée (Nicolas, 2016).

Pour atteindre ces objectifs, le processus d'inspection des turbines à gaz doit intégrer une solution logicielle. Cette dernière doit être en mesure de traiter les images acquises par le boroscope et de signaler visuellement à l'inspecteur si les conditions d'acquisition d'images sont bonnes tout au long de l'inspection et de prendre en charge le comptage de lamelles. Dans ce qui suit, nous détaillons les fonctionnalités que les inspecteurs désirent retrouver dans la solution logicielle.

- Champs à remplir obligatoirement avant de commencer l'inspection. Ce sont : le modèle du boroscope, l'adaptateur d'extrémité optique, le modèle de moteur ainsi que son numéro de série, le module, la section et le numéro de l'étage à inspecter. Aussi, selon la position de la caméra, l'opérateur peut préciser : le côté du bord qui peut être un *bord de fuite* ou un *bord d'attaque* et la pose du boroscope qui peut être *racine*, *centre* ou *extrémité*. Ceci

permettra d'associer les informations pertinentes à la vidéo et la capture correspondante systématiquement et ainsi structurer les noms des fichiers pour faciliter la recherche ultérieure ;

- Des indicateurs sur l'interface permettant d'avertir l'inspecteur visuellement si des modifications doivent être effectuées pour avoir une bonne qualité des images capturées. En termes de vitesse de rotation, luminosité, clarté et netteté des images capturées. Ça permet de détecter si la vitesse de rotation est assez lente pour avoir des images claires et nettes. Ça permet aussi de détecter quelques problèmes comme : lentilles sales, mal alignées ou mal sélectionnées. Cette fonctionnalité permet de réduire le risque de produire des vidéos inutiles pour la post-inspection et de réduire le risque d'effectuer une inspection dans des conditions sous optimales ;
- Des vidéos pertinentes et facilement accessibles, disponibles pour chaque étage de tous les moteurs. Ces vidéos montrent comment manipuler le boroscope pour arriver à la meilleure position de la caméra permettant de bien visualiser la zone cible. Ceci permet d'éviter le saut d'étage ou l'inspection d'un mauvais étage ;
- Compteur automatique en temps réel. Ceci permettra de s'en passer du compteur manuel utilisé et de libérer l'inspecteur de cette tâche. Il est aussi important d'indiquer visuellement à l'inspecteur que le tour complet est atteint. Pour cela, il faut détecter l'arrivée du compteur qui s'incrémente au cours de l'inspection au nombre de lamelles total de l'étage en cours d'inspection. Cette information est déduite à partir des informations entrées par l'utilisateur ;
- Une structure qui indique le nombre de lamelles à tour complet pour chaque étage. L'information est déduite en se basant sur les informations concernant le moteur et l'étage à inspecter entrées par l'inspecteur ;
- Sauvegarde automatique des fichiers sur le réseau au lieu de la clé de mémoire USB, directement avec les noms des fichiers construits à partir des informations entrées par l'utilisateur. Ce qui empêcherait la perte accidentelle de données ;
- Attirer l'attention de l'inspecteur aux trames qui contiennent une zone potentiellement défectueuse.

## 1.6 Requis essentiels et contraintes

Le but principal de ce projet de maîtrise est d'améliorer la qualité de l'inspection des turbines à gaz. Ceci en proposant une solution logicielle qui guide l'inspecteur pour acquérir des images de bonne qualité et le libère entièrement du comptage des lamelles pour le laisser se concentrer dans l'étape de détection des défauts.

Parmi les fonctionnalités désirées de la part des inspecteurs, les suivantes représentent l'objectif de ce projet :

1. Qualification des images acquises lors de l'inspection :

D'une part, il s'agit de valider les formules avec lesquelles les trois métriques de qualification des images sont implémentées. D'autre part, il faut valider les intervalles d'acceptance avec les inspecteurs ;

2. Compteur de lamelles :

La raison principale pour laquelle le compteur de lamelles de la solution existante ne fonctionne pas est la logique implémentée. Celle-ci devrait être remplacée pour avoir un compteur qui traite les images et affiche les résultats en temps réel ;

3. Mise à jour de la base de données des moteurs inspecté à Siemens.

## 1.7 Conclusion

Dans ce chapitre, nous avons détaillé le contexte et la problématique. Aussi, nous avons défini les objectifs de notre projet.

Une des métriques qui qualifient les images nécessite l'estimation du mouvement des lamelles tout au long de l'inspection. A partir de là, la vitesse des lamelles peut être calculée et utilisée pour indiquer à l'inspecteur si la vitesse de rotation qu'il applique sur l'étage est adéquate. Une autre métrique ainsi que le compteur des lamelles requièrent la détection des contours. Pour cela, nous avons besoin de traiter les images acquises en appliquant des filtres. Les contours détectés représenteront principalement les bordures des lamelles et seront utilisés dans le processus du comptage. Notons qu'avant la détection des contours, les images traitées doivent être dé-bruitées.

Dans le chapitre suivant, nous présentons une étude bibliographique des méthodes pouvant être utilisées pour atteindre nos objectifs. Cette étude entame les trois thématiques : élimination du bruit, détection des contours et estimation du mouvement.



## CHAPITRE 2

### REVUE DE LITTÉRATURE

#### 2.1 Introduction

La détection des contours est l'un des problèmes fondamentaux du traitement d'images qui a été largement étudié car il trouve son intérêt dans de nombreux processus comme la segmentation, la reconnaissance des formes, le suivi des objets, etc. En gros, il s'agit de détecter les variations locales, brusques et significatives de la fonction intensité de l'image. Le but est de réduire l'information dans l'image de façon significative afin de l'utiliser dans les étapes ultérieures (Rafael C. Gonzalez, 2018).

Du bruit peut se présenter sur les images lors de l'acquisition, la compression et la transmission. Ce bruit est aussi représenté par une variation d'intensité locale pouvant être détectée comme contour (Rafael C. Gonzalez, 2018). Pour cela, la suppression du bruit est une étape importante qui doit précéder la détection des contours.

Après avoir nettoyé les images du bruit et détecté les contours, les objets repérés peuvent être suivis. Ceci est intéressant dans plusieurs applications comme le suivi des personnes et la conduite autonome (Haering, Venetianer & Lipton, 2008), (Campbell, Egerstedt, How & Murray, 2010).

Ce chapitre comporte trois sections. Nous commençons par un aperçu des méthodes de suppression du bruit. Nous enchaînons avec les méthodes de détection de contour les plus référencées dans l'état de l'art. Ensuite, nous présentons les deux méthodes d'estimation de mouvement les plus utilisées. Une conclusion représente la dernière section du chapitre.

#### 2.2 Elimination du bruit

Une image est une matrice dont les éléments sont des valeurs d'intensité de pixel de niveau de gris ou de couleur. Elle peut être considérée comme un signal en deux dimensions, pour chaque pixel, une valeur d'intensité est associée, qu'elle soit unique (image en niveau de gris)

ou un triplet de valeurs, par exemple : rouge, vert et bleu (image en couleurs). L'image peut contenir des régions homogènes lisses, des textures et des contours résultants de la variation des valeurs des pixels. Les régions homogènes aident à déterminer la forme globale des objets dans une image, tandis que la variation rapide de l'intensité des pixels représente les bordures des objets. Ces informations sont importantes pour l'analyse, la compréhension et l'extraction d'informations des images (Szeliski, 2021).

Avec la présence du bruit, d'éventuelles tâches de traitement d'image ultérieures peuvent être affectées négativement. Par conséquent, la suppression du bruit joue un rôle important dans les systèmes de traitement d'image modernes. Il s'agit de supprimer le bruit sans perdre les caractéristiques importantes comme les bordures, les coins, les textures et les structures nettes (Shao, Yan, Li & Liu, 2013).

Plusieurs algorithmes ont été développés dans le but de supprimer le bruit tout en préservant autant de détails et d'informations que possible (Shao *et al.*, 2013; Buades, Coll & Morel, 2005).

Dans ce qui suit, nous exposons les méthodes du domaine spatial et les méthodes du domaine de transformation.

### **2.2.1 Filtres dans le domaine spatial**

Dans ce domaine, les opérations de suppression de bruit sont appliquées directement sur la matrice de l'image. Pour débruiter un patch, la corrélation entre les pixels candidats est exploitée. L'idée est que le bruit n'est pas corrélé entre les pixels et que les intensités de pixel de signal réel sont corrélées les unes aux autres. Selon le mode de sélection des pixels candidats, un filtre peut être catégorisé en tant que filtre local et filtre non local (Singh & Neeru, 2014).

#### **2.2.1.1 Filtres locaux**

Pour ce type de filtres, la sélection des pixels est basée sur la distance spatiale. Comme le filtre moyen, le filtre bilatéral, le filtre gaussien, le filtre SUSAN et le filtre Weiner.

Le filtre local le plus simple est le filtre moyen (Thanh, Engínođlu et al., 2019). Pour chaque pixel, la moyenne de la fenêtre sélectionnée est affectée au pixel central. Ce filtre génère souvent une quantité indésirable de lissage des bordures et une perte de détails (Thanh *et al.*, 2019). A leur tour, le filtre gaussien (Afshari, Gadsden & Habibi, 2017) et le filtre de Weiner (Benesty, Chen & Huang, 2010) sont des filtres linéaires qui n'emploient pas la moyenne des pixels voisins. Le filtre de Weiner est un filtre linéaire optimal qui implique une estimation linéaire d'une séquence de signal souhaitée à partir d'une autre séquence associée. Le filtre gaussien est un filtre linéaire qui entre dans la catégorie des filtres locaux, est de nature isotrope et est appliqué depuis longtemps dans la suppression de bruit. Il suppose que les pixels d'un voisinage sont homogènes et présentent des variations spatiales lisses. Cependant, un flou des bords apparaît lorsque la taille du noyau augmente (Afshari *et al.*, 2017).

Ces filtres constituent une solution simple et efficace mais ils ont tendance à supprimer les bordures qui sont des informations importantes pour l'analyse d'image (Fan, Zhang, Fan & Zhang, 2019). En particulier, dans notre projet où nous cherchons à isoler les bordures des lamelles pour les compter.

Pour répondre à ces limites, des filtres non linéaires ont été proposés. Entre autres, le filtre médian (Ng & Ma, 2006), le filtre de diffusion anisotrope (ADF) (Weickert, 1996), le filtre médian pondéré (Yang, Yin, Gabbouj, Astola & Neuvo, 1995) et le filtre Rank (Hardie & Barner, 1994).

Dans le filtrage médian (Ng & Ma, 2006), les pixels voisins sont classés en fonction de la luminosité et la valeur médiane devient la nouvelle valeur du pixel central. Le filtre de diffusion anisotrope s'agit d'un ensemble de techniques de filtrage non linéaire qui visent à réduire le bruit en préservant les bords (Weickert, 1996). Pour cela, il applique un processus inhomogène pour réduire la diffusivité aux emplacements des bords et effectue un processus de diffusion dans des régions homogènes ou avec de légères variations d'intensité.

La plupart des filtres de domaine spatial se basent sur la proximité spatiale, mais les filtres de voisinage prennent en compte les similitudes de niveaux de gris pour définir un patch de

voisinage. Citons le filtre SUSAN (Smith & Brady, 1997) qui est une approche de filtrage spatial basée sur le filtre de voisinage qui calcule la moyenne de tous les pixels équidistants du pixel central (Smith & Brady, 1997) et le filtre bilatéral qui considère à la fois la similitude des niveaux de gris et la proximité spatiale dans le processus de suppression du bruit (Tomasi & Manduchi, 1998). Ce dernier élimine le bruit dans les régions appartenant aux pixels de référence et préserve les bordures de l'image. Les pixels qui sont spatialement proches et qui ont des intensités de pixels similaires sont moyennés avec des poids décroissant avec la distance croissante par rapport au pixel de référence. Le processus de filtrage bilatéral est simple et non itératif, mais sa mise en œuvre directe est lente car sa complexité de calcul augmente de façon exponentielle avec l'augmentation du rayon de voisinage. Du coup, plusieurs versions sont apparues. Parmi elles : l'approche proposée par Chaudhary et al. qui emploie des noyaux de plage trigonométriques au lieu de considérer la distance spatiale euclidienne (Chaudhury, Sage & Unser, 2011); le filtre bilatéral accéléré qui a été donnée en 2013 (Chaudhury, 2012) et l'approche proposée par Chaudhary et coll. sûre et rapide en combinant le filtre bilatéral d'origine et sa modification de manière pondérée (Chaudhury & Rithwik, 2015).

Étant donné que l'idée de base pour la mise en œuvre des filtres locaux est l'exploitation de la corrélation entre les pixels, ils ont tendance à mal fonctionner à des niveaux de bruit élevés, car la corrélation entre les pixels est gravement altérée par des niveaux de bruit élevés.

### **2.2.1.2 Filtres non locaux**

Contrairement aux filtres locaux, les filtres non locaux exploitent la similitude entre les pixels de manière non locale, dans l'image entière. Dans (Buades *et al.*, 2005), Buades et al. ont introduit le filtre de moyenne non locale (NLM) qui exploite la présence de caractéristiques similaires dans l'image. C'est une approche typique de réduction du bruit, qui exploite la distance euclidienne entre deux patchs autour de la référence et le pixel sélectionné et génère un pixel sans bruit pour un seul point seulement.

Le filtrage morphologique (Teng, Zhang, Chen & Ti, 2015) est un autre concept mathématique utilisé dans la suppression du bruit. Il s'agit d'extraire des caractéristiques d'image utiles pour la description et la représentation d'une forme. Le filtrage morphologique comprend des opérations d'ouverture et de fermeture pour le lissage des images et l'élimination du bruit. L'opération d'ouverture lisse les contours d'un objet, rompt les creux étroits et supprime les pics lumineux, tandis que la fermeture fusionne généralement les creux étroits, élimine les petits trous et diminue les détails les plus sombres (Teng *et al.*, 2015).

L'objectif principal des algorithmes de suppression du bruit est de conserver autant de détails que possible. Malgré les niveaux de performance élevés atteints par les filtres du domaine spatial, ils s'efforcent à éliminer le bruit haute fréquence et non le signal haute fréquence qui est le problème le plus difficile du filtrage du domaine spatial. En effet, le bruit est aléatoire et a été présumé avoir une fréquence plus élevée, et le vrai signal a une fréquence plus basse, par conséquent, la fréquence plus élevée peut facilement être supprimée afin d'éliminer le bruit. Cependant, les bordures et les caractéristiques ne sont souvent pas répétitifs et ont une fréquence plus élevée et leur suppression entraîne une corruption des bords et un flou du signal. D'autre part, ces filtres sont lents à cause de la grande taille de la fenêtre des filtres (Goyal, Dogra, Agrawal, Sohi & Sharma, 2020).

### **2.2.2 Filtres dans le domaine de transformation**

Contrairement aux approches du domaine spatial, les méthodes du domaine de transformation transforment d'abord l'image bruitée dans un autre domaine, puis elles appliquent une procédure de suppression du bruit sur l'image transformée en fonction des différentes caractéristiques de l'image et de son bruit. Dans le domaine de fréquences, les caractéristiques des informations d'image (comme les bordures) sont différentes à celles du bruit. En fait, les coefficients plus petits dans la bande de haute fréquence peuvent être seuillés et les coefficients de grande amplitude dans la bande de fréquence inférieure sont préservés, lesquels sont moins susceptibles d'être du bruit. En seuillant les coefficients plus petits, le bruit peut être éliminé et finalement l'image est restaurée par transformation inverse des coefficients dans le domaine spatial (Goyal *et al.*, 2020).

Les méthodes de domaine de transformation ont été développées à partir de la transformée de Fourier (FT), mais depuis lors, une variété de méthodes a progressivement émergé, telles que la transformée de Fourier rapide (FFT), la transformée en cosinus discrète (DCT), les ondelettes (Hou, 2007; Zhang, Bao & Wu, 2005) et le filtrage 3D (BM3D) (Dabov, Foi, Katkovnik & Egiazarian, 2007).

La FT et la DCT ont été largement utilisés dans la suppression du bruit et la compression avec perte (Goyal *et al.*, 2020). En comparaison avec DCT, la transformée en ondelettes discrète (DWT) utilise un nombre plus élevé de jeux optimaux de fonctions pour représenter des arêtes vives que les cosinus. En plus de cela, dans le cas de sous-bandes passe-haut, la DCT est capable de fournir une résolution de fréquence plus élevée et une résolution spatiale plus faible. Une quantité considérable de littérature montre les performances supérieures de la transformée en ondelettes par rapport à FT et DCT, en termes de restauration d'image efficace et de compression avec perte (Xiong, Ramchandran, Orchard & Zhang, 1999).

DWT (Discrete Wavelet Transform) est un outil mathématique qui est obtenu par discrétisation de la transformée en ondelettes continue et constitue un facteur de translation et de dilatation englobant l'image entière à différentes échelles dans une fenêtre glissante (Nason & Silverman, 1994). DWT construit un niveau d'approximation des images à des niveaux plus grossiers en utilisant des bancs de filtres passe-bas et un niveau détaillé en utilisant des bancs de filtres passe-haut. Chaque niveau d'approximation est ensuite décomposé en représentation d'image à l'échelle suivante. Le DWT utilise un certain nombre d'ondelettes mères qui fournissent une fonction de base pour la décomposition du signal (Dogra, 2017). L'efficacité et l'aptitude des ondelettes mères sont régies par des facteurs tels que la longueur des filtres, l'orthogonalité, le support compact et la régularité. Daubechies est la fonction d'ondelette mère la plus couramment utilisée (Goyal *et al.*, 2020). Les méthodes de suppression de bruit basées sur l'ondelette transforment généralement le contenu de l'image en plusieurs sous-bandes à différentes résolutions et échelles.

En tant qu'extension efficace et puissante de l'approche NLM, BM3D (filtrage collaboratif 3-D par correspondance de blocs), qui a été proposée par Dabov et al. (Dabov *et al.*, 2007). BM3D est

une méthode de filtrage collaborative en deux étapes dans le domaine de transformation. Dans cette méthode, des patchs similaires sont empilés en groupes 3D par correspondance de blocs, et les groupes 3D sont transformés en domaine d'ondelettes. Ensuite, un seuillage dur ou un filtrage de Wiener avec des coefficients est utilisé dans le domaine des ondelettes. Enfin, après une transformation inverse des coefficients, tous les patchs estimés sont agrégés pour reconstruire l'image entière. Cependant, lorsque le bruit augmente progressivement, les performances de débruitage du BM3D diminuent considérablement et des artefacts sont introduits, en particulier dans les zones plates. Pour améliorer les performances de débruitage, de nombreuses versions améliorées de BM3D sont apparues, en l'occurrence (Dabov, Foi, Katkovnik & Egiazarian, 2009).

### 2.2.3 Comparaison et discussion

Pour comparer les filtres dans le domaine spatial, des expériences ont été réalisées avec des images sur lesquelles du bruit a été simulé. Ce bruit est par la suite supprimé en utilisant les algorithmes à évaluer et comparer.

Dans (Singh & Neeru, 2014) trois images ont été utilisées, parmi elles l'image Lena montrée dans la figure 2.1. Plusieurs modèles de bruit ont été simulé sur l'image originale, parmi eux : le bruit blanc gaussien et le bruit set et poivre. Pour le processus de suppression du bruit, six filtres spatiaux ont été appliqué sur les images bruitées, parmi eux : le filtre gaussien ( $3 \times 3$ ), le filtre moyen ( $3 \times 3$ ) et le filtre médian ( $3 \times 3$ ). La figure 2.2 montre l'image originale corrompue avec les différents types de bruit et chaque image bruitée filtrée à l'aide des filtres étudiés.

Les performances quantitatives des filtres ont été évaluées à l'aide de la métrique PSNR qui peut être définie avec l'équation suivante.

$$PSNR = 10 \log_{10} \frac{255^2}{MSE} \quad (2.1)$$

$$MSE = \frac{\sum_i \sum_j (r_{ij} - x_{ij})^2}{M \times N} \quad (2.2)$$



FIGURE 2.1 Image Lena

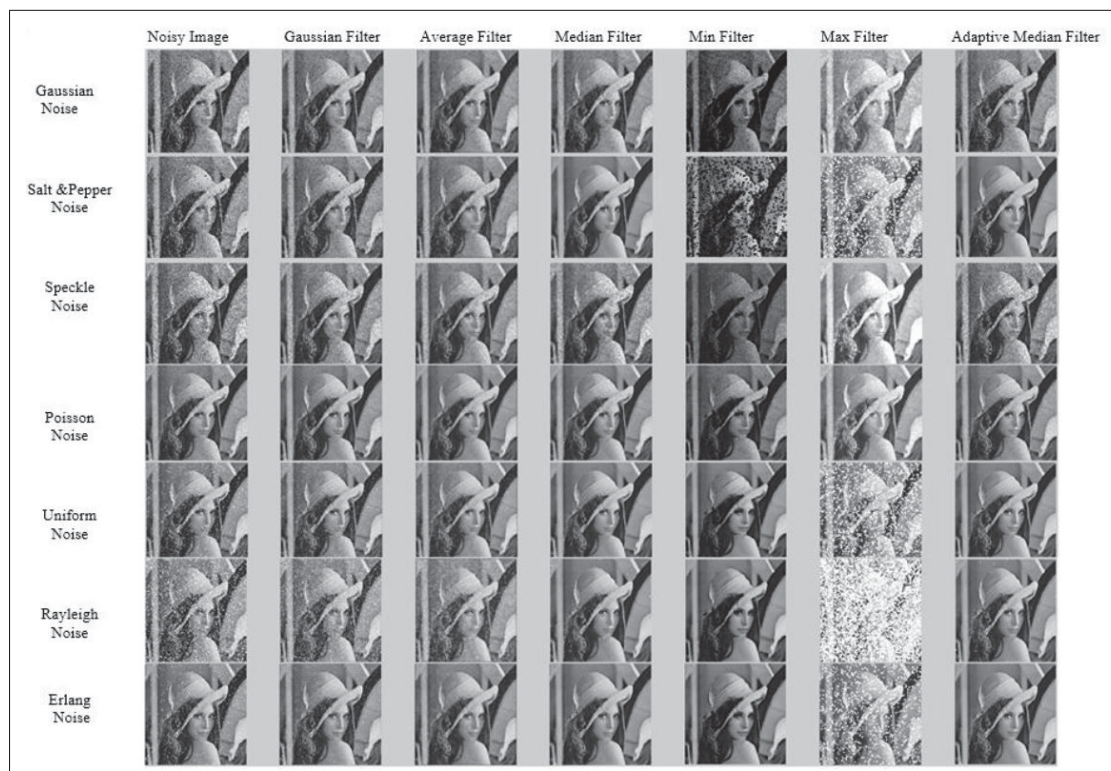


FIGURE 2.2 Image Lena contenant différents types de bruit et filtrée en utilisant différents filtres spatiaux  
Tirée de (Singh & Neeru, 2014)

où :  $r$  représente l'image originale,  $x$  représente l'image restaurée et  $M \times N$  est la taille de l'image traitée.

Les résultats de la comparaison montrent que le filtre moyen affiche de meilleures performances dans la suppression du bruit gaussien tandis que le filtre gaussien supprime efficacement le bruit de Poisson. Les filtres médians adaptatifs ont bien réussi à éliminer le bruits sel et poivre.

Concernant les filtres dans le domaine de transformation, la transformation en ondelettes est la mieux adaptée à la performance en raison de ses propriétés telles que la multi-résolution, la nature multi-échelle et la rareté. En plus des performances, les problèmes de complexité de calcul doivent également être pris en compte. Les techniques de seuillage utilisées avec la transformée en ondelettes discrète sont les plus simples à mettre en œuvre (Gupta et al., 2014).

### **2.3 Détection de contours**

La détection de contours est l'un des problèmes fondamentaux du traitement d'images et vision par ordinateur (Rafael C. Gonzalez, 2018). Une bordure est la limite entre un objet et l'arrière-plan ou un autre objet, elle indique les limites qui existent entre les régions dans une image. Généralement, les images sont converties en niveau de gris pour détecter plus facilement les contours. Dans ce cas, les valeurs d'intensité sont comprises entre 0 et 255. Dans les images en niveaux de gris, une bordure est caractérisée par une variation brusque d'intensité entre les pixels voisins. La détection de contours est donc la recherche des variations locales et significatives de la fonction intensité de l'image. Notons que le bruit doit être éliminé avant de détecter les bordures car il affecte le changement d'intensités dans l'image et réduit le taux de réussite des algorithmes de détection des contours (Rafael C. Gonzalez, 2018).

Les techniques de détection de contours sont divisées en deux catégories principales : les techniques basées sur le gradient et les techniques basées sur la laplacienne. La détection de contours basée sur le gradient essaie de faire des points maximum et minimum en prenant la première dérivée de l'image. Les pixels qui ont une valeur de gradient élevée sont considérés comme des régions du contour. En laplacienne, les opérateurs de détection des contours sont

pris en tant que deuxième dérivée de l'image et essaient de détecter les points de passage par zéro (Rafael C. Gonzalez, 2018).

Dans ce qui suit, nous présentons les opérateurs Robet, Sobel, Prewitt et Canny qui sont basés sur le gradient ainsi que l'opérateur laplacien de Gaussien qui est basé sur la laplacienne.

### 2.3.1 Opérateur Roberts

Cet opérateur (Roberts, 1963) effectue une mesure de gradient spatial 2D à calculer sur une image. Ainsi, il met en évidence des régions de haute fréquence spatiale qui correspondent aux contours. Les valeurs de pixels à chaque point de l'image en sortie représentent l'amplitude absolue estimée du gradient spatial de l'image d'entrée à ce point. L'opérateur se compose des deux noyaux  $2 \times 2$  suivants (Amer & Abushaala, 2015).

$$G_x = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, G_y = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \quad (2.3)$$

L'application de ces deux noyaux produit les composantes de gradient dans chaque orientation : horizontale et verticale. Celles-ci peuvent ensuite être combinées pour trouver l'amplitude absolue du gradient en chaque point et l'orientation de ce gradient. L'amplitude du gradient est donnée par (Amer & Abushaala, 2015) :

$$|G| = \sqrt{G_x^2 + G_y^2} \quad (2.4)$$

L'angle d'orientation du contour donnant lieu au gradient spatial (par rapport à l'orientation de la grille de pixels) est donné par (Amer & Abushaala, 2015) :

$$a = \arctan(G_y/G_x) - 3\pi/4 \quad (2.5)$$

### 2.3.2 Opérateur Sobel

Cet opérateur (Sobel, 1970) est connu comme un opérateur de différenciation discrète qui calcule l'approximation de gradient de la fonction d'intensité d'image pour la détection de bord d'image. Parmi les pixels de l'image, l'opérateur Sobel produit soit la normale à un vecteur, soit le vecteur gradient. Il applique, avec la convolution, les deux masques  $3 \times 3$  suivants pour calculer numériquement la première dérivée  $G_x$  et  $G_y$ .

$$G_x = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix}, G_y = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix} \quad (2.6)$$

### 2.3.3 Opérateur Prewitt

L'opérateur Prewitt (Prewitt, 1970) est très similaire à l'opérateur Sobel qui détecte également les bords verticaux et horizontaux d'une image. Il est basé sur le gradient et comporte les deux noyaux de taille  $3 \times 3$  suivants.

$$G_x = \begin{pmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}, G_y = \begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix} \quad (2.7)$$

### 2.3.4 Laplacien de Gaussien

Cet opérateur, également appelé LoG, a été proposé par (Marr & Hildreth, 1980). Il révèle que les points de densité évoluent rapidement dans l'image. Du coup, il peut être utilisé pour détecter les contours. Le filtre LoG prend une seconde dérivée dans l'image et essaie de trouver les points où la dérivée croise zéro, ce qui signifie que l'emplacement correspond à un niveau maximum (un summum) et est appelé emplacement périphérique (Rafael C. Gonzalez, 2018). Ce filtre est

très sensible au bruit à cause de l'utilisation de la seconde dérivée de l'image. C'est pourquoi un filtre réduisant le bruit doit être utilisé préalablement.

La valeur de densité de pixels laplacienne est calculée comme indiqué dans l'équation 2.8.

$$L(x, y) = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2} \quad (2.8)$$

Les filtres  $3 \times 3$  couramment utilisés pour le filtre LoG dans les applications de traitement d'image qui ont moins de bruit sont les suivants.

$$filtre1 = \begin{pmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{pmatrix}, \quad filtre2 = \begin{pmatrix} -1 & 2 & -1 \\ 2 & -4 & 2 \\ -1 & 2 & -1 \end{pmatrix} \quad (2.9)$$

### 2.3.5 Opérateur Canny

L'algorithme de détection de contours de Canny (Canny, 1986) est l'algorithme de détection de bord le plus couramment utilisé dans la pratique. Cet opérateur n'est pas sensible au bruit. Il fonctionne sur la base de trois critères : une bonne détection de contours (minimiser le taux d'erreur); une localisation des contours précise (afin de minimiser la distance entre les points détectés et le vrai contour); une faible multiplicité des maxima dû au bruit (afin de minimiser le nombre de réponses pour un seul contour). Chacun de ces critères est modélisé mathématiquement par Canny. Les principales étapes de l'algorithme de Canny sont (Rafael C. Gonzalez, 2018) :

1. Appliquer un filtre gaussien afin d'éliminer le bruit;
2. Calculer l'intensité du gradient en utilisant le filtre de Sobel;
3. Calculer les directions du gradient et les arrondir par multiples de 45 degrés dans les directions : verticale, horizontale et diagonales;

4. Supprimer les non-maxima dans le cas où la norme du gradient en un pixel est inférieure à la norme du gradient d'un de ses voisins le long de la direction du gradient. Cela signifie que les pixels non appartenant aux contours sont réduits au niveau 0 ;
5. Seuillage des contours. Pour chaque pixel :
  - Si la norme du gradient est inférieure à un seuil minimal, mettre le pixel à zéro
  - Si la norme du gradient est supérieure au seuil maximal marquer le pixel contour
  - Si la norme est supérieure au seuil minimal et inférieure au seuil maximal alors le pixel est contour s'il est connecté à autre pixel contour.

### **2.3.6 Comparaison et discussion**

Chacun des détecteurs de contours présentés dans cette section a ses propres avantages et inconvénients. Plusieurs travaux ont été menés pour comparer ces détecteurs, nous résumons les résultats obtenus dans ce qui suit.

Pour montrer les différences entre les différents opérateurs de détection de contours, les métriques MSE et PSNR sont utilisées. Dans (Singh & Singh, 2015), une comparaison entre toutes les techniques de détection de contours a été menée en utilisant l'image Lena (2.1). Tous les opérateurs ont été appliqués sur l'image et ont été comparés entre eux. Les résultats de comparaison ont été présentés dans des tableaux (Singh & Singh, 2015). Sur la base du MSE, l'opérateur Sobel montre le plus de similitude avec le Prewitt et montre le plus de dissemblance avec Canny. L'opérateur Canny montre le plus de dissemblance avec Roberts. L'opérateur Prewitt montre le plus de similitude avec le Sobel et montre le plus de dissimilarité avec Canny. L'opérateur LoG montre le plus de dissemblance avec Canny. L'opérateur Roberts montre le plus de similitude avec le Prewitt et montre le plus de dissemblance avec Canny. À partir des résultats obtenus, l'opérateur Sobel donne l'image de plus haute qualité et l'opérateur Roberts donne l'image de moindre qualité.

Dans (Shah *et al.*, 2020) les paramètres MSE et le PSNR ont été calculés pour les différentes techniques de détection de contour appliquées sur deux images différentes (figure 2.3). Les images obtenues après l'application des filtres sont présentées dans la figure 2.4.



FIGURE 2.3 Les deux images originales utilisées  
Tirée de (Shah *et al.*, 2020)

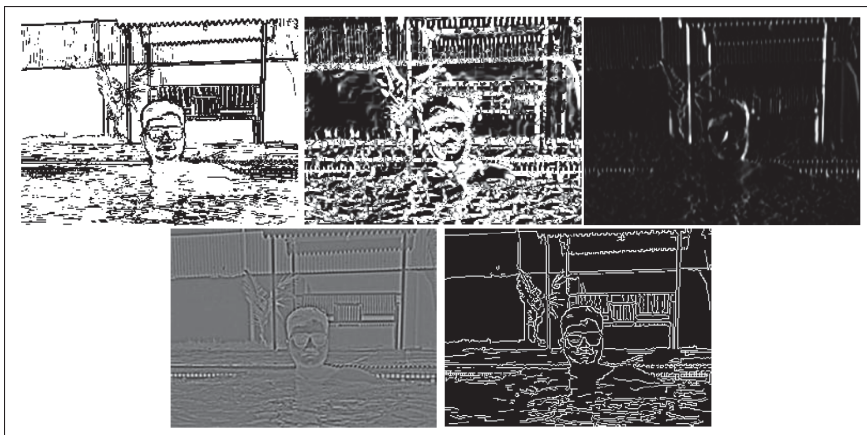


FIGURE 2.4 Image (1) filtrée avec les filtres : Reborts, Sobel, Prewitt, Laplacien et Canny dans l'ordre  
Tirée de (Shah *et al.*, 2020)

L'analyse a montré que l'opérateur Canny a été la méthode la plus efficace pour la détection des bords pour les deux images. L'efficacité de toutes les autres méthodes a été affectée par le bruit contrairement à l'opérateur Canny.

Les méthodes de détection des bords basées sur le gradient telles que Sobel, Prewitt et Roberts ne produisent pas de bons bords nets et leurs performances sont réduites pour les images bruitées.

Le Laplacien de Gaussien est efficace pour détecter l'emplacement correct des bords le long d'une grande zone de pixels, mais produit de faux bords en raison d'un changement de haute intensité aux coins et d'une image avec une mauvaise orientation des bords. Canny est une méthode optimale parmi toutes ces techniques. Il offre une bonne réponse sur l'emplacement des bords, en particulier lorsque les images contiennent du bruit. Il utilise deux seuils, l'un pour les arêtes fortes et l'autre pour les arêtes faibles.

## **2.4 Détection du mouvement**

Après avoir détecté les objets dans une séquence vidéo, différentes techniques peuvent être utilisées pour les suivre. Parmi elle, la différenciation temporelle, la soustraction de fond et la méthode de flux optique (Yilmaz, Javed & Shah, 2006). Dans cette section, nous présentons d'abord la théorie des flux optiques. Ensuite, les deux méthodes d'extraction de flux Farneback et Lucas-Kanade. Ces deux méthodes d'extraction de flux, parmi tant d'autres (Nagel, 1987), sont connues et largement utilisées dans différentes applications de vision par ordinateur.

### **2.4.1 Théorie du flux optique**

Le flux optique est la présentation de l'information de mouvement à partir d'une séquence d'images (Husseini, 2017). Ces informations sont extraites en comparant deux images successives capturées à deux moments différents. Généralement, le flux optique montre un vecteur de déplacement 2D pour chaque pixel montrant son mouvement de la première trame à la seconde dans les axes perpendiculaires de l'image. L'estimation du flux optique part du principe que les intensités des pixels ne changent pas avec le temps et que les pixels voisins ont un mouvement similaire (OpenCV, 2021a).

Les méthodes de flux optique sont classées en méthodes globales et locales (Husseini, 2017). Dans les méthodes locales, nous n'avons besoin de traiter que certains pixels de l'image entière, tandis que les méthodes globales (ou denses) traitent tous les pixels de l'image. Considérons un pixel  $I(x, y, t)$  dans la première trame, il passe à la trame suivante en un temps  $dt$ , le déplacement

des pixels de la première trame à la suivante  $(dx, dy)$ . Comme l'intensité des pixels ne change pas, nous pouvons dire que (OpenCV, 2021a) :

$$I(x, y, t) = I(x + \Delta x, y + \Delta y, t + \Delta t) \quad (2.10)$$

Pour trouver le déplacement des pixels, nous utilisons la méthode différentielle car elle utilise les dérivées partielles par rapport aux coordonnées spatiales et temporelles. En supposant que le déplacement est petit, nous prenons une approximation locale de la série de Taylor du signal de l'image et nous obtenons :

$$I(x + \Delta x, y + \Delta y, t + \Delta t) = I(x + y + t) + \frac{\partial I}{\partial x} \Delta x + \frac{\partial I}{\partial y} \Delta y + \frac{\partial I}{\partial t} \Delta t + c \quad (2.11)$$

Où  $I(x + y + t)$  est le pixel supposé à l'emplacement  $(x, y, t)$ ,  $\Delta x$ ,  $\Delta y$ ,  $\Delta t$  représentent le mouvement entre les deux trames et  $c$  est une constante de valeur réelle. À partir de ces équations, nous obtenons :

$$\frac{\partial I}{\partial x} \Delta x + \frac{\partial I}{\partial y} \Delta y + \frac{\partial I}{\partial t} \Delta t = 0 \quad (2.12)$$

Qui se traduit par :

$$\frac{\partial I}{\partial x} V_x + \frac{\partial I}{\partial y} V_y + \frac{\partial I}{\partial t} = 0 \quad (2.13)$$

Où  $V_x = \frac{dx}{dt}$  et  $V_y = \frac{dy}{dt}$  sont la vitesse (ou le flux optique) de  $I(x + y + t)$  et  $\frac{\partial I}{\partial x}$ ,  $\frac{\partial I}{\partial y}$ ,  $\frac{\partial I}{\partial t}$  sont les gradients de l'image en  $(x, y, t)$ . Les équations  $V_x$  et  $V_y$  sont inconnues et ne peuvent être résolues avec deux variables inconnues. Ainsi, plusieurs méthodes différentielles sont proposées pour résoudre ce problème (Nagel, 1987). Lukas-Kanade et Farneback sont deux d'entre eux, que nous pouvons classer respectivement en méthodes locales et globales (Bruhn, Weickert & Schnörr, 2005).

Dans les méthodes locales, juste quelques pixels de la trame entière sont traités. Ces pixels représentent des caractéristiques comme les coins, les arêtes et les tâches. Par contre, dans les méthodes globales, tous les pixels de la trame sont traités. Les méthodes locales sont plus efficaces pour calculer de manière robuste sous le bruit, mais les techniques denses sont plus

précises en termes de synchronisation. Les approches Farneback et Lucas-Kanade sont détaillées dans ce qui suit.

### 2.4.2 Méthode Farneback

La méthode Farneback est un algorithme d'estimation de mouvement entre deux images qui utilise l'expansion polynomiale, où un voisinage de chaque pixel d'image est approché par polynôme (Husseini, 2017).

Nous ne nous intéressons ici qu'aux polynômes quadratiques, qui donnent le modèle de signal local représenté dans un système de coordonnées local (Farneback, 2003).

$$f(x) \sim x^T A x + b^T x + c_1 \quad (2.14)$$

Où :

- $x$  est un vecteur  $1 \times 2$  contenant les variables courantes  $x$  et  $y$  ;
- $A$  est une matrice symétrique d'inconnues  $2 \times 2$ , qui capture des informations sur la partie paire du signal ;
- $b$  est la partie impaire du signal et un vecteur  $2 \times 1$  d'inconnues ;
- $c_1$  est un scalaire inconnu.

L'écrire en termes de tenseurs donne l'équation suivante, où les coefficients inconnus sont définis par le symbole  $r_i$ .

$$f(x) \sim \begin{bmatrix} x & y \end{bmatrix} \begin{bmatrix} r_4 & \frac{r_6}{2} \\ \frac{r_6}{2} & r_5 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} r_1 & r_3 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + r_1 \quad (2.15)$$

En calculant les polynômes de voisinage sur deux images successives, on peut obtenir directement le déplacement  $d$  en cas de translation idéale. Sur la figure 2.5, l'image  $f_1(x)$  est prise au temps  $t$  et  $f_2(x)$  au temps  $(t + dt)$ .

Par conséquent, si on peut mesurer  $f_1$  et  $f_2$ , on peut trouver  $d = [d1, d2]$ , qui est le déplacement 2D et donc le flux du pixel de l'image. Dans les équations 2.16, nous voyons comment les

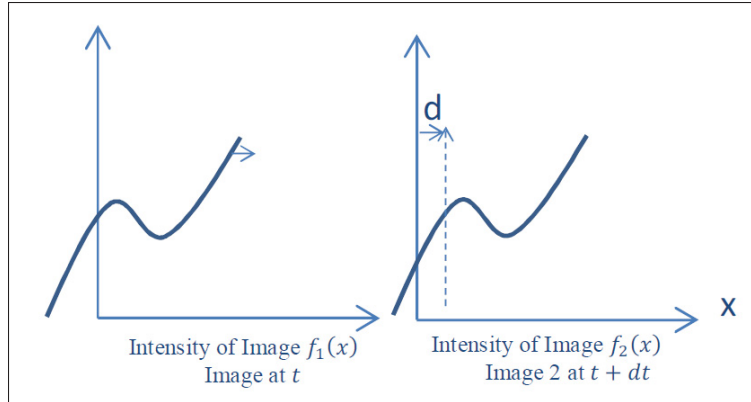


FIGURE 2.5 Polynômes de voisinage sur deux images consécutives,  $d$  est le déplacement  
Tirée de (Husseini, 2017)

coefficients polynomiaux de la deuxième trame sont reliés à ceux de la première trame et au déplacement  $d$ .

$$\begin{aligned}
 f_2(x) &= f_1(x - d) = (x - d)^T A_1 (x - d) + b_1^T (x - d) + c_1 & (2.16) \\
 &= x^T A_1 x + (b_1 - 2A_1 d)^T x + d^T A_1 d - b_1^T d + c_1 \\
 &= x^T A_2 x + b_2^T x + c_2
 \end{aligned}$$

En supposant que la luminosité est constante dans les deux images consécutives, nous pouvons égaliser les coefficients dans les deux polynômes.

$$A_2 = A_1 \quad (2.17)$$

$$b_2 = b_1 - 2A_1 d$$

$$c_2 = d^T A_1 d - b_1^T d + c_1$$

En tirant parti de l'équation pour  $b_2$  dans 2.17, on peut résoudre pour  $d$ , comme le montre l'équation 2.18.

$$d = -\frac{1}{2}A_1^{-1}(b_2 - b_1) \quad (2.18)$$

Il est également indiqué dans (Farneback, 2003) que l'équation 2.16 est pour le cas idéal et qu'elle pourrait être une approximation plus réaliste lorsque le terme  $-\frac{1}{2}(b_2 - b_1)$  de l'équation 2.18 est changé en  $\Delta b$ . Dans ce cas :

$$\Delta b = -\frac{1}{2}(b_2(x) - b_1(x)) \quad (2.19)$$

$$A(x) = \frac{A_1(x) + A_2(x)}{2} \quad (2.20)$$

Cela change l'équation 2.18 en l'équation 2.21

$$A(x)d(x) = \Delta b(x) \quad (2.21)$$

En principe, l'équation 2.21 peut être résolue ponctuellement, mais les résultats s'avèrent trop bruités. Nous supposons que tous les pixels d'une petite fenêtre se comportent de la même manière. Si nous pouvons mesurer  $A_1$ ,  $b_1$  et  $c_1$  à partir de  $f_1$  pour chaque pixel de  $I$  et  $A_2$ ,  $b_2$  et  $c_2$  à partir de  $f_2$ , nous pouvons calculer  $d$ .

Ainsi, nous essayons de trouver  $d(x)$  satisfaisant 2.21 aussi bien que possible sur un voisinage  $I$  de  $x$ , ou plus formellement minimisant

$$\sum_{\Delta x \in I} w(\Delta x) \| A(x + \Delta x)d(x) - \Delta b(x + \Delta x) \|^2 \quad (2.22)$$

Où nous laissons  $w$  une fonction de pondération des points du voisinage. Où la résolution de  $d(x; y)$  dans l'équation 2.22 donne l'équation 2.23

$$d(x) = \left( \sum w A^T A \right)^{-1} \sum w A^T \Delta b \quad (2.23)$$

### 2.4.3 Méthode de Lucas-Kanade pyramidale

Lucas-Kanade-Tomasi (KLT) est une estimation de mouvement à deux images qui utilise la minimisation pour résoudre l'équation de flux optique de base. L'algorithme est largement utilisé et développé par Bruce D. Lucas et Takeo Kanade (OpenCV, 2021a). La nature de la méthode Lucas-Kanade est de calculer le flux uniquement pour certains points et de suivre de très petits changements ou mouvements à travers les images. Cependant, quand il y a un grand mouvement ou une vitesse élevée qui doit être capturée à travers des images, nous devons utiliser une grande fenêtre et cela peut casser l'hypothèse de mouvement cohérent. Pour contourner ce problème, Lucas-Kanade a été utilisé dans une pyramide, montrée dans la figure 2.6. Cette figure montre une pyramide d'images avec quatre niveaux. L'estimation du mouvement commence au niveau supérieur de la pyramide et continue à descendre jusqu'au niveau inférieur. La taille de la fenêtre reste la même à tous les niveaux. Ainsi, il est possible de détecter de grands mouvements dans les niveaux plus grossiers de la pyramide.

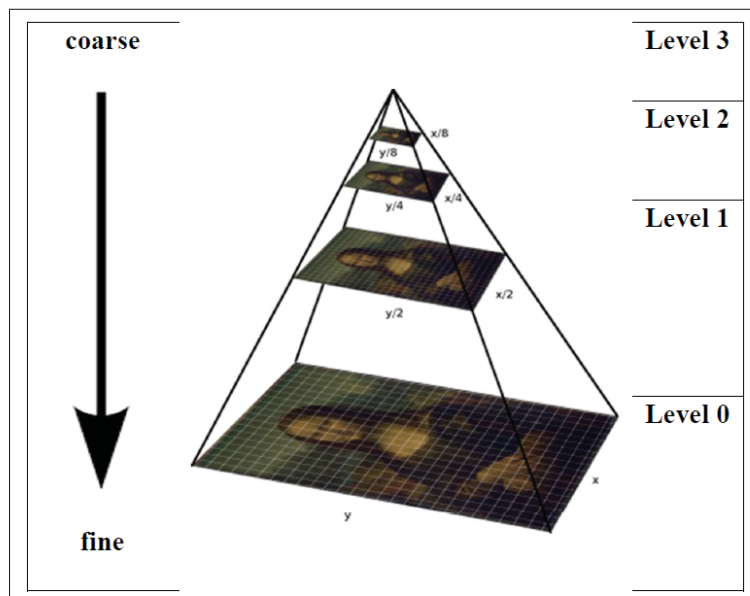


FIGURE 2.6 Pyramide d'images à quatre niveaux. À chaque niveau, l'image est réduite. Le calcul du flux optique commence au sommet de la pyramide (niveau 4) et se termine en bas (niveau 0)

Tirée de (Husseini, 2017)

À la fin de la section 2.4.1, nous avons abouti à une équation à deux variables inconnues ( $V_x, V_y$ ) et la résolution d'une équation à deux inconnues n'est pas possible, donc Lucas-Kanade utilise l'hypothèse de cohérence spatiale pour résoudre ce problème. Par conséquent, le procédé utilise des pixels voisins pour avoir plus d'équations, par exemple le voisinage  $5 \times 5$  autour du pixel courant nous donne 25 équations comme suit.

$$\underbrace{\begin{bmatrix} I_x(p_1) & I_y(p_1) \\ I_x(p_2) & I_y(p_2) \\ \vdots & \vdots \\ I_x(p_{25}) & I_y(p_{25}) \end{bmatrix}}_A \underbrace{\begin{bmatrix} V_x \\ V_y \end{bmatrix}}_v = - \underbrace{\begin{bmatrix} I_t(p_1) \\ I_t(p_2) \\ \vdots \\ I_t(p_{25}) \end{bmatrix}}_b \quad (2.24)$$

Où  $I_x(p_i)$ ,  $I_y(p_i)$ ,  $I_t(p_i)$  sont des dérivées spatiales de l'image  $I$  en position  $(x, y)$  et dans le temps  $t$ , et  $p_1, p_2, \dots, p_{25}$  sont les pixels à l'intérieur de la fenêtre. Dans ce cas,  $A$  est une matrice  $25 \times 2$ ,  $v$  est un vecteur de flux d'image local  $2 \times 1$  et  $b$  est un vecteur  $25 \times 1$ . Pour simplifier l'équation, nous l'écrivons sous cette forme  $Av = b$ . Ce système contient plus d'équations que d'inconnues donc c'est un système sur contraint. Pour résoudre ce système, la minimisation des moindres carrés est trouvée en multipliant l'équation par  $A^T$  et, elle résout le système  $2 \times 2$ .

$$\underbrace{(A^T A)}_{2 \times 2} \underbrace{v}_{2 \times 1} = \underbrace{A^T b}_{2 \times 2} \quad (2.25)$$

Où  $A^T$  est la matrice transposée de  $A$ . à partir de cette relation, nous obtenons l'équation 2.26, où la première matrice est une matrice inversible et les sommes sont de  $i = 1$  à  $n$ .

$$\begin{bmatrix} \sum_i I_x(p_i)^2 & \sum_i I_x(p_i)I_y(p_i) \\ \sum_i I_y(p_i)I_x(p_i) & \sum_i I_y(p_i)^2 \end{bmatrix} \begin{bmatrix} v_x \\ v_y \end{bmatrix} = \begin{bmatrix} -\sum_i I_x(p_i)I_t(p_i) \\ -\sum_i I_y(p_i)I_t(p_i) \end{bmatrix} \quad (2.26)$$

La solution de cette équation est alors :

$$(A^T A)^{-1} A^T b = \begin{bmatrix} v_x \\ v_y \end{bmatrix} \quad (2.27)$$

L'équation 2.27 peut être résolue lorsque  $A^T A$  est inversible ou lorsque les valeurs propres de  $A^T A$  ( $\lambda_1, \lambda_2$ ) sont suffisamment grandes et ont la même grandeur.

#### 2.4.4 Comparaison et discussion

Dans (Husseini, 2017), les deux méthodes d'estimation de flux optique présentées ci-dessous ont été évaluées et comparées. Pour mener l'étude, plusieurs clips vidéo comprenant des objets et scènes d'intérieur et d'extérieur ont été utilisés. Les deux ensembles de vidéos sont différents en termes de caractéristiques temporelles et spatiales (voir la figure 2.7). Le premier comprend des scènes extérieures, avec des mouvements d'objets à grande vitesse et la fréquence d'images est plus élevée. Par conséquent, certaines des images sont floues et empêchent le bon fonctionnement du système. Dans le deuxième ensemble de vidéos, les scènes sont dans un environnement intérieur, dans des conditions d'éclairage différentes avec des déplacements d'objets plus lents.



FIGURE 2.7 L'image de gauche est un exemple d'image de séquence extérieure et l'image de droite est une séquence d'image intérieure  
Tirée de (Husseini, 2017)

Selon les résultats obtenus, le modèle Farneback, évalué sur les deux ensembles de données, est le plus précis. Cependant, il souffre d'imprécision dans la scène extérieure.

Un des principaux défis rencontrés lors de l'étude de la méthode Lucas-Kanade était que dans certains cadres, la fonction n'était pas en mesure de trouver de bons points pour calculer les flux.

Un nouveau système de suivi d'objets pour les méthodes d'extraction de flux a été proposé en utilisant l'optimisation de flux inverse (Husseini, 2017). Avec le nouveau système, la précision du suivi a été augmentée d'environ 10% (Husseini, 2017).

## **2.5 Conclusion**

Plusieurs recherches ont été menées dans le domaine de la suppression du bruit, la détection des contours ainsi que l'estimation du mouvement. Selon le contexte et la nature de la problématique à résoudre, il est possible de sélectionner les méthodes et les adapter pour atteindre les meilleurs résultats.

Dans le chapitre suivant, nous présentons la méthodologie que nous proposons pour résoudre la problématique de qualification des images et comptage de lamelles.



## CHAPITRE 3

### MÉTHODOLOGIE

#### 3.1 Introduction

Avant d'implémenter la solution que nous proposons pour résoudre la problématique : qualifier les images capturées et compter les lamelles, nous allons d'abord analyser la solution existante et identifier pourquoi elle ne fonctionnait pas.

Dans ce chapitre, nous commençons par la description et l'analyse de la solution existante. Ensuite, nous présentons la solution que nous proposons pour atteindre les objectifs de ce projet.

#### 3.2 Situation de départ

Pour mettre en place les fonctionnalités et les inclure dans le processus d'inspection des turbines à gaz, un système informatique a été mis en place avec de nouvelles composantes matérielles :

- Un ordinateur pour exécuter le logiciel qui fournit l'interface usager et effectue l'analyse des images acquises par la caméra en temps réel. Pour minimiser les appareils à transporter autour de la turbine, un PC All-In-One a été utilisé ;
- Un module d'acquisition vidéo (Avio HD - Epiphan) qui relie le boroscope à l'ordinateur. Il relie la sortie VGA qui provient du boroscope à une entrée USB de l'ordinateur. Son rôle est de convertir le flux vidéo capturé par la caméra du boroscope en images pouvant être traitées sur un ordinateur (Epiphan, 2021).

La figure 3.1 montre la mise en place des composants ajoutés.

Un logiciel qui fournit une partie des fonctionnalités proposées a été développé et fourni à Siemens. Ce logiciel n'était pas utilisé car il est non-fonctionnel.

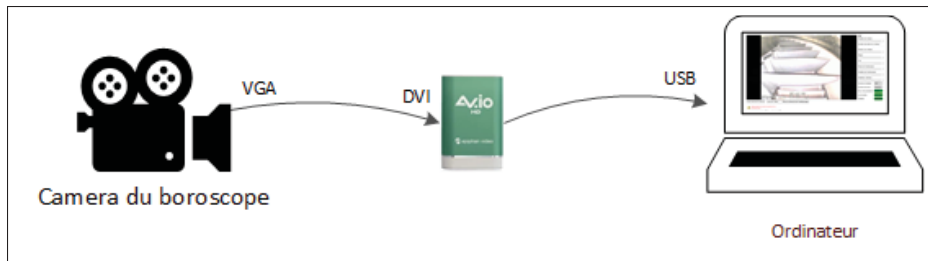


FIGURE 3.1 Matériel requis pour l'intégration du logiciel au processus d'inspection

### 3.2.1 Description globale de la solution existante

Parmi les fonctionnalités désirées présentées dans la section 1.5, quelques-unes ont été implémentées. Elles sont : la mise en place des champs obligatoires ; la base de données qui indique le nombre de lamelles de chaque étage pour trois moteurs ; trois indicateurs de qualité des images avec couleur, l'accès aux vidéos enregistrés ainsi que le compteur de lamelles. Pour cela, une interface graphique a été mise en place (voir la figure 3.2). Dans le côté gauche de l'interface, une zone est consacrée à l'affichage des images provenant de la caméra en temps réel. En bas de cette zone, l'utilisateur trouve les boutons qui permettent de démarrer et arrêter l'enregistrement des vidéos, la date et l'heure actuelles ainsi que les boutons qui permettent de lire des vidéos enregistrées. Dans le volet droit de l'interface se présentent les champs permettant l'identification de l'étage à inspecter. Ces informations sont mises dans des listes déroulantes dans le but de faciliter et unifier l'entrée de ces données. A partir de ces informations en entrées, le nombre de lamelles de l'étage en question à tour complet est déduit, aussi la vidéo montrant la façon avec laquelle l'endoscope peut être positionné dans l'étage en question peut être accédée via le bouton 'How To'. En bas de ces listes déroulantes, il y a des champs consacrés à l'affichage du compteur de lamelles ainsi que les métriques de qualité des images qui se mettent à jour en temps réel. Avec un code couleur, l'utilisateur peut savoir si les images acquises sont bonnes. Pour donner de la flexibilité au code source, le contenu des listes déroulantes a été mis dans des fichiers XML pouvant être facilement gérées par Siemens de manière autonome.

Notons que la langue peut être modifiée par le bouton ‘EN’ ou ‘FR’ qui se présente en haut du volet droit.



FIGURE 3.2 Interface graphique de la solution existante

### 3.2.2 Fonctionnement de la solution existante

La solution a été développée avec la technologie .NET, langage C#. Elle est constituée de plusieurs classes, chaque classe effectue une fonctionnalité précise. Le code source de l'interface graphique est implémenté dans un fichier nommé *MainWindow.xaml*. Les classes qui s'occupent de la qualification des images et du comptage des lamelles sont : *BladeCounter.cs*, *Blur.cs*, *Brightness.cs* et *Sharpness.cs*. les fonctions implémentées dans ces classes sont appelées à partir d'une autre classe nommée *MainWindowViewModel.cs*. La bibliothèque OpenCV est utilisée pour le traitement des images acquises.

Dans le fichier qui relie l'interface graphique et les modèles, il y a un minuteur qui déclenche la fonction *dispatcherTimer\_Tick* chaque 150 ms. Cette fonction met à jour l'image courante capturée du contenant qui affiche la vidéo en temps réel sur l'interface à la variable *latestGrab-*

*bedImage* de la classe *MainWindowViewModel.cs*. Suite à cette affectation, l'image précédente est remplacée par l'image courante qui est remplacée par la nouvelle image qui vient d'être capturée et la fonction *processNewImage* est appelée. Cette fonction prend en paramètre les deux images : précédente *Pim* et courante *Cim* qui vont être utilisées dans les deux parties de la solution, à savoir la qualification des images et le compteur de lamelles.

Il est important de noter qu'un temporisateur est mis en place pour bloquer les variables lors du traitement des trames capturées dans la fonction *processNewImage*. Donc, les images sont mises à jour chaque 150 ms si les traitements sont finis en moins de 150 ms. Dans le cas contraire, le prochain déclenchement aura lieu après 300 ms, 450 ms, etc. Une fois la nouvelle trame récupérée et affectée à la variable *latestGrabbedImage*, les deux parties du code : qualification des images et compteur de lamelles sont exécutées.

Notons que la qualification des images consiste à refléter la qualité des images acquises par la caméra par le biais de quelques métriques calculées. Ces métriques indiquent à l'inspecteur s'il doit améliorer les captures obtenues en effectuant quelques changements comme : modifier l'intensité projetée dans le moteur, changer le positionnement de la caméra, nettoyer les lentilles utilisées, diminuer la vitesse de rotation des lamelles.

Pour bien expliquer le fonctionnement du logiciel, un diagramme de séquences montrant le fonctionnement de la solution est présenté dans la figure 3.3 ainsi qu'un diagramme d'activités détaillé, présenté dans la figure I-1.

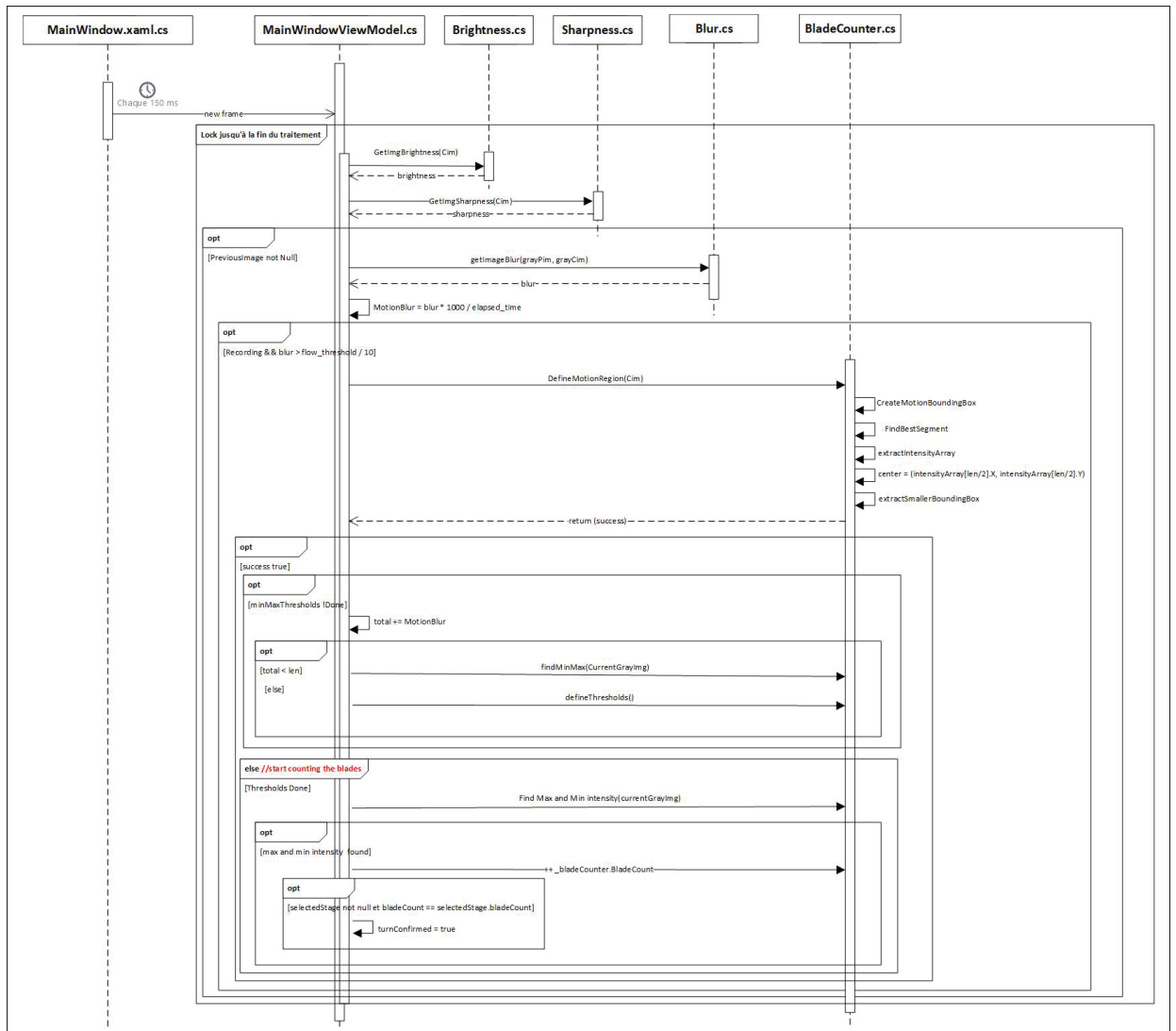


FIGURE 3.3 Diagramme de séquence global de la solution existante

### 3.2.2.1 Qualification des images

Le but de cette partie est de qualifier les images capturées par la caméra. Si elles ne sont pas bonnes, il faut le signaler à l'inspecteur pour qu'il effectue les modifications nécessaires. Pour cela, trois métriques sont calculées pour chaque image capturée et affectée à la variable *latestGrabbedImage*. A savoir : Brightness, Sharpness et Blur. Pour dire que les images obtenues sont bonnes, il faut que les valeurs trouvées appartiennent aux intervalles d'acceptance définis.

Dans ce cas, les champs réservés aux métriques prennent la couleur verte, sinon ça prend la couleur jaune ou rouge si la valeur est très petite ou très grande par rapport à la valeur acceptable.

- **Brightness** cette métrique est utilisée pour quantifier la luminosité globale de l'image afin d'assurer que la quantité de lumière projetée est bonne. Pour calculer la luminosité, les 3 composantes couleur R, G et B de chaque pixel sont récupérées. Ces composantes sont représentées par des valeurs entières entre 0 et 255 (8 bits) et sont utilisées pour calculer la luminosité du pixel en question en utilisant la formule ci-dessous (Al-Tairi, Rahmat, Sariipan & Sulaiman, 2014). A la fin, la moyenne des luminosités de tous les pixels est retournée et affichée à l'utilisateur.

$$luminosité = 0.2126 \times R + 0.7152 \times G + 0.0722 \times B \quad (3.1)$$

Intervalle d'acceptance et couleurs affichées :

- (0, 75) ou (230, 255) : image inacceptable (sombre ou très lumineuse). Couleur : rouge ;
  - (75, 100) ou (200, 230) : image un peu sombre ou un peu lumineuse. Couleur : jaune ;
  - (100, 200) : bonne luminosité. Couleur : vert.
- **Sharpness** cette métrique indique la netteté de l'image. Après avoir converti l'image courante en niveau de gris en utilisant l'équation 3.1, le filtre laplacien est appliqué pour détecter les contours. Ensuite, l'image est ajustée en soustrayant le min (qui est toujours négatif) de tous les pixels de l'image. Comme résultat, l'intervalle des valeurs passe de [min, max] à [0, max+|min|]. Par la suite, l'image est convertie dans l'intervalle [0, 255] avec la formule 3.2 appliquée à chaque pixel de l'image source :

$$image\_ajustée(I) = \left| image\_source(I) \times \frac{255}{max + |min|} \right| \quad (3.2)$$

Où I est l'intensité du pixel en cours.

La moyenne d'intensité de l'image ajustée est calculée. La netteté de l'image est calculée comme le montre la formule suivante.

$$sharpness = 255 - moyenne(image\_ajustée) \quad (3.3)$$

Intervalle d'acceptance :

- (0, 100) : image floue. Couleur : rouge ;
  - (100, 145) : image un peu floue. Couleur : jaune ;
  - (145 et plus) : image nette. Couleur : vert.
- **Blur et MotionBlur** cette métrique mesure le flou du mouvement causé par la rotation des lamelles. Ceci affecte considérablement la qualité des image aperçues, ce qui peut nuire à la détection des défauts. Pour calculer cette métrique, on a besoin de deux images successives : l'image courante et la précédente.

Pour commencer, les deux images sont converties en niveau de gris (équation 3.1) et leur taille est réduite par 4 (horizontalement et verticalement) dans le but de gagner du temps de calcul. Ensuite, l'algorithme *OptFlowDeepFlow* est utilisé pour calculer les vecteurs de mouvement pour chaque pixel entre les deux images en entrée. Comme résultat, on a une matrice de vecteurs de mouvement ayant la taille de l'image qui précise la taille du vecteur de mouvement vertical et horizontal pour chaque pixel. Après avoir divisé cette matrice en *flowr* et *flowc* (flux horizontal et vertical), on calcule la moyenne de la taille du vecteur de mouvement horizontal *mr* et vertical *mc* globales avec les pixels ayant une taille de vecteur de mouvement considérable ( $norm \geq 0.015$ ). La taille du vecteur de mouvement est calculée avec la formule suivante :

$$norm = \sqrt{flowr[r, c]^2 + flowc[r, c]^2} \quad (3.4)$$

Où : *r* et *c* sont, respectivement, les coordonnées horizontale et verticale du pixel en cours de traitement.

A la fin, *Blur* prend la valeur  $\sqrt{mr^2 + mc^2} \times 4$  pour revenir à l'échelle de l'image originale. Ceci représente la distance moyenne parcourue par la lamelle entre les deux trames considérées.

Ensuite, *MotionBlur* est calculé en divisant le blur par le temps écoulé entre l'acquisition des deux trames dans la variable *latestGrabbedImage*. Le résultat est une vitesse calculée comme ceci :

$$MotionBlur = Blur \times \frac{1000}{elapsed} (px/s) \quad (3.5)$$

Intervalle d'acceptance :

- (0, 12) : la vitesse est lente. Couleur : jaune ;
- (12, 17) : la vitesse est bonne. Couleur : vert ;
- (17 et plus) : la vitesse est grande. Couleur : rouge.

### 3.2.2.2 Compteur de lamelles

Avant de commencer le comptage, l'utilisateur doit remplir les champs obligatoires concernant le moteur à inspecter dans l'interface. A ce stade, le bouton qui permet de commencer l'enregistrement apparaît. Quand on appuie sur ce bouton, l'enregistrement de la vidéo commence ainsi que le traitement des images qui permet de compter les lamelles.

L'idée globale est de détecter la différence d'intensité importante qui représente le passage de la lamelle par une position fixée. La première étape est de définir la position du carré dans lequel la moyenne d'intensité sera calculée. Puis, des seuils d'intensité minimale et maximale seront définis pendant le passage de la première lamelle. Finalement, la moyenne d'intensité est calculée pour toutes les images acquises dans la même position. Dès qu'une intensité supérieure au seuil maximal est suivie par une intensité inférieure au seuil minimal, le compteur est incrémenté. Les étapes détaillées du fonctionnement sont présentées dans ce qui suit.

- **DefineMotionRegion**

Le but est de placer le carré dans lequel la moyenne d'intensité sera calculée. Ça se fait en plusieurs étapes :

- *CreateMotionBoundingBox* : trouver le rectangle qui englobe la zone de mouvement global dans l'image. Cette fonction prend en entrée les deux matrices *flowr* et *flowc* qui représentent les matrices du mouvement horizontal et vertical respectivement. A partir de

ces matrices, les vecteurs ayant une longueur de mouvement considérable (supérieure à une constante définie) sont repérés. En sortie, on a le point bas gauche et le point haut droit qui identifient le rectangle qui couvre les vecteurs repérés ;

- FindBestSegment : le but est de trouver, dans le rectangle de mouvement global, la zone qui englobe le mouvement représenté par l'angle du mouvement. Ce dernier est calculé à partir des mouvements moyens  $mr$  et  $mc$  calculées auparavant. Ensuite, on cherche la  $b$  qui donne la ligne de droite la plus proche du centre de l'image, avec la même direction de mouvement. Dans le schéma suivant (figure 3.4), la trame est représentée par le rectangle noir. Le rectangle qui englobe la zone de mouvement global est représenté par le rectangle bleu et la flèche noire représente la direction du mouvement global selon l'angle calculé ( $mr, mc$ ). Selon ce dernier, la zone rouge, qui représente une zone de mouvement plus précise, est déduite ;

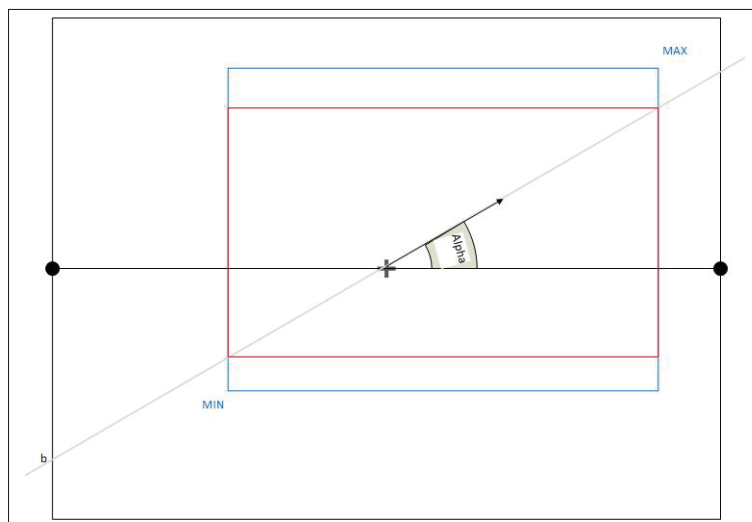


FIGURE 3.4 Définir la zone du mouvement

- ExtractIntensityArray : cette fonction s'occupe de la recherche des pixels les plus proches de la ligne  $a \times x + best\_b$  qui représente le mouvement. Après avoir calculé cette liste, on définit le point au milieu comme *center* autour duquel un carré de taille  $7 \times 7$  est créé, c'est dans ce carré que la moyenne d'intensité sera calculée.

- **Commencer le comptage**

L'idée est de définir les seuils d'intensité maximale et minimale pendant le passage de la première lamelle tout au long de l'écran. Pour cela, on cherche la moyenne d'intensité minimale et maximale dans le carré fixé pour les trames sélectionnées successivement jusqu'à ce que la lamelle traverse l'écran (voir la figure 3.5). Pour vérifier si la lamelle a traversé l'écran, on rajoute le motionBlur calculé à chaque fois à une variable qu'on compare avec la diagonale de l'image traitée. Une fois cette distance atteinte, on suppose que l'intensité minimale et maximale ont été trouvées.

En se basant sur ces intensités, on calcule les seuils auxquels les intensités seront comparées comme suit :

$$Seuil\_maximal = imax - (0.28 \times (imax - imin)) \quad (3.6)$$

$$Seuil\_minimal = imin + (0.30 \times (imax - imin)) \quad (3.7)$$

Où : *imax* et *imin* sont, respectivement, les intensités maximale et minimale trouvées.

Une fois ces seuils calculés, on calcule la moyenne d'intensité pour les trames suivantes jusqu'à l'obtention d'une intensité maximale (supérieure ou égale au seuil maximal). Ensuite, on cherche une intensité minimale (inférieure ou égale au seuil minimal). Dès qu'on trouve cette succession d'intensité maximale et minimale, le compteur est incrémenté. En réalité, cette succession d'intensité correspond au passage d'une lamelle qui se présente comme montré dans la figure 3.5 : le bord de la lamelle est très clair (voir la zone rouge) par rapport à l'ombre (voir la zone jaune) qui vient juste après.

À la fin, on vérifie si le nombre de lamelles trouvé a atteint le nombre de lamelle de l'étage inspecté pour notifier l'inspecteur qu'il a fini l'étage en cours.

Le schéma présenté dans la figure 3.6 montre le fonctionnement global de la solution existante.

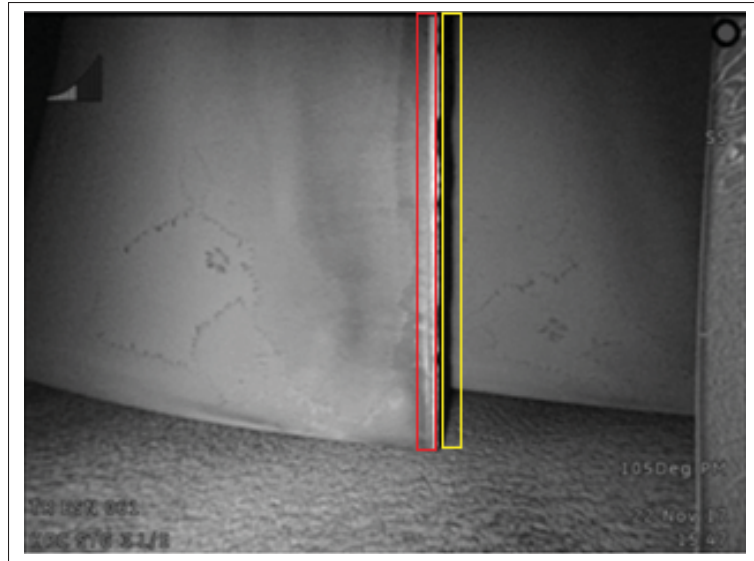


FIGURE 3.5 Différence d'intensité due au passage d'une lamelle

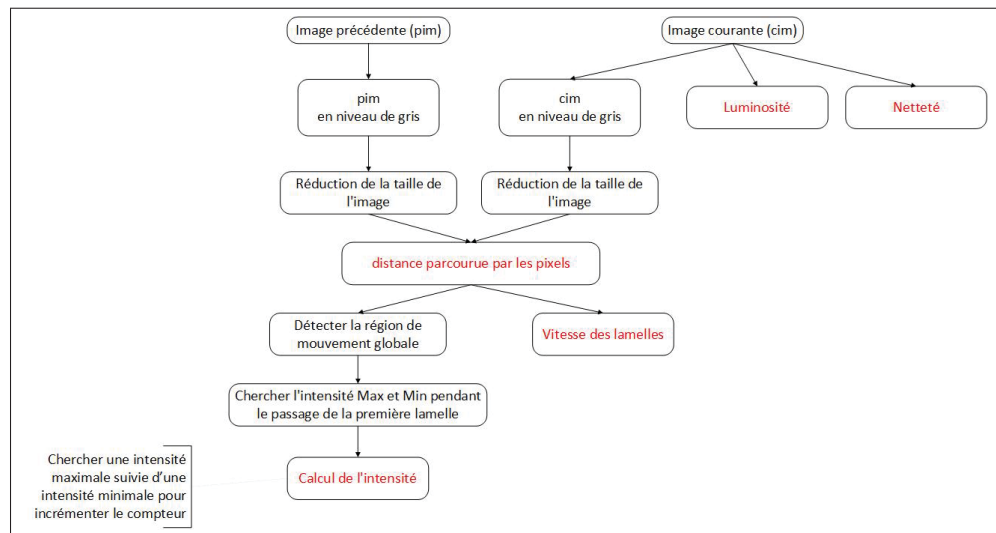


FIGURE 3.6 Fonctionnement global de la solution existante

### 3.2.3 Exécution et tests de la solution existante

Dans cette section, nous présentons comment la solution existante a été testée dans le but de l'analyser et trouver les raisons pour lesquelles elle ne fonctionne pas.

### 3.2.3.1 Méthodes d'exécution de la solution

Dans la solution actuelle, les trames sont capturées à partir du contenant qui les affiche sur l'interface chaque 150 ms. Pour exécuter le logiciel en dehors du plancher et sans avoir à utiliser le boroscope, il faut récupérer les trames à traiter autrement.

La première méthode consiste à remplacer les trames capturées de l'interface par des trames extraites des vidéos enregistrées auparavant, ces vidéos ont été fournies par Siemens. Pour cela, il faut extraire les trames à partir de la vidéo présente sur le disque et les traiter au lieu des trames qui proviennent du boroscope en temps réel.

La deuxième approche consiste à simuler l'exécution avec un boroscope en temps réel. Pour cela, nous avons d'abord projeté un deuxième écran à partir de la machine principale avec une sortie HDMI. C'est sur cet écran que l'interface de l'application sera affichée. Ensuite, nous projetons un troisième écran avec une sortie VGA, cette sortie représente l'entrée du video grabber dont la sortie USB est rebranchée à l'ordinateur. Comme le montre le schéma suivant, l'entrée USB provenant du video grabber (l'écran 3) est prise comme la source vidéo affichée dans le contenant spécifique à la source vidéo qui provient du boroscope. Sur l'écran, nous projetons un lecteur vidéo en plein écran qui va lire une vidéo enregistrée. C'est ce qui va simuler le flux provenant du boroscope en temps réel.

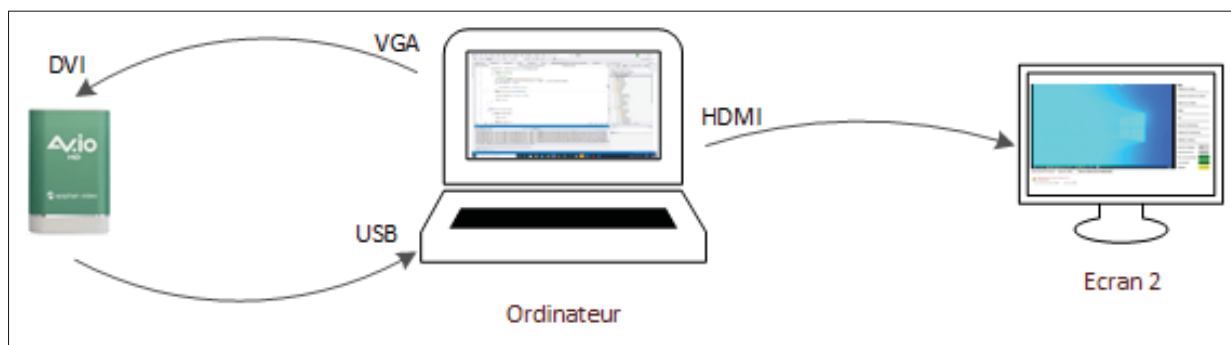


FIGURE 3.7 Simulation de fonctionnement en temps réel sans boroscope

### 3.2.3.2 Tests et observations

Après avoir testé la solution avec les deux approches décrites plus haut, nous avons remarqué que les champs consacrés aux valeurs des métriques de qualification des images étaient mis à jour régulièrement. Par contre, le compteur de lamelles ne marchait pas. Pour comprendre pourquoi, nous avons débogué le programme et conclu les points suivants :

- La sélection des trames à partir de la caméra en temps réel n'est pas unique si l'ordinateur sur lequel le logiciel s'exécute n'est pas le même. En effet, la capture de l'image à traiter dépend du temps de calcul qui varie selon les caractéristiques de la machine utilisée. La trame peut être capturée chaque 150, 300, 450 ms. Le logiciel a été testé sur trois machines avec des caractéristiques différentes et le temps qui sépare les captures varie selon la machine de 300 ms (1 trame sur 9) à 600 ms (1 trame sur 18) à 1050 (une trame sur 30) sachant que le taux d'acquisition est de 30 trames par seconde. Par conséquent, les trames sélectionnées diffèrent d'une machine à l'autre, et le résultat du compteur n'est pas le même pour la même vidéo.
- Même avec la même vidéo et la même machine, le résultat du compteur n'était pas le même entre plusieurs exécutions. Car les trames sélectionnées étaient différentes à chaque fois. Par conséquent, l'emplacement du carré et les seuils maximale et minimale étaient différents. La figure 3.8 montre deux exécutions pour une vidéo, chaque exécution est présentée dans une ligne. Chaque ligne montre la détection de l'intensité maximale (image de gauche) et l'intensité minimale (image de droite) au niveau de carré sélectionné pour l'exécution en cours. Entre les deux exécutions, l'emplacement du carré est différent.
- Au niveau de la lamelle, il y a une variation d'intensité dû à sa forme, à la luminosité projetée ainsi qu'à des tâches qui peuvent se présenter lors de l'inspection. Donc, l'intensité des pixels peut varier et passer d'une valeur minimale à une valeur maximale sans que la bordure de la lamelle passe par le carré. Dans la première ligne de la figure 3.8, l'intensité maximale correspond à la bordure de la lamelle mais la minimale ne correspond pas à l'ombre. Dans la deuxième ligne, ni la maximale ni la minimale correspond à ce qu'il faut. Ceci peut être dû à la capture des trames trop éloignées dans le temps.

- Il y a des vidéos qui peuvent contenir plusieurs lamelles qui apparaissent dans la même trame, ces lamelles sont très rapprochées. La détection des intensités minimale et maximale est difficile avec la sélection des trames éloignées.
- Avec cette méthode, on suppose que le sens de rotation des lamelles est toujours de droite à gauche. On cherche l'intensité maximale qui correspond à la bordure de la lamelle puis on cherche l'intensité minimale qui correspond à l'ombre. Si l'inspecteur fait tourner l'étage dans l'autre sens, on doit chercher les intensités inversement. Et ce n'est pas pris en considération.
- L'emplacement du carré dans lequel l'intensité est calculée n'est pas visible pour l'utilisateur
- La consommation de la mémoire RAM augmente durant l'exécution jusqu'au rejet de l'exception *Out Of Memory*. Ceci est dû à plusieurs raisons qui sont :
  - Traitement des deux images à chaque fois. L'image courante traitée dans une itération est retraitée en tant qu'image précédente dans la prochaine itération ;
  - Instanciation de l'algorithme qui calcule le mouvement dans la fonction qui est appelée à chaque fois qu'une trame est capturée ;
  - Des variables déclarées (images, matrices), utilisées mais l'espace mémoire alloué n'est pas libéré après l'utilisation ;
  - Des boucles inutiles consomment le temps d'exécution.
- Pour dire que la première lamelle a traversé tout l'écran, la distance parcourue doit être comparée à la distance totale qui est la largeur de l'écran, dans le cas d'un mouvement horizontal. Dans cette solution, on compare une vitesse à une distance totale, qui est la diagonale de la trame. Notons qu'il y a des cas où l'espace pouvant être parcouru par la lamelle est très petit par rapport à la largeur de la trame.

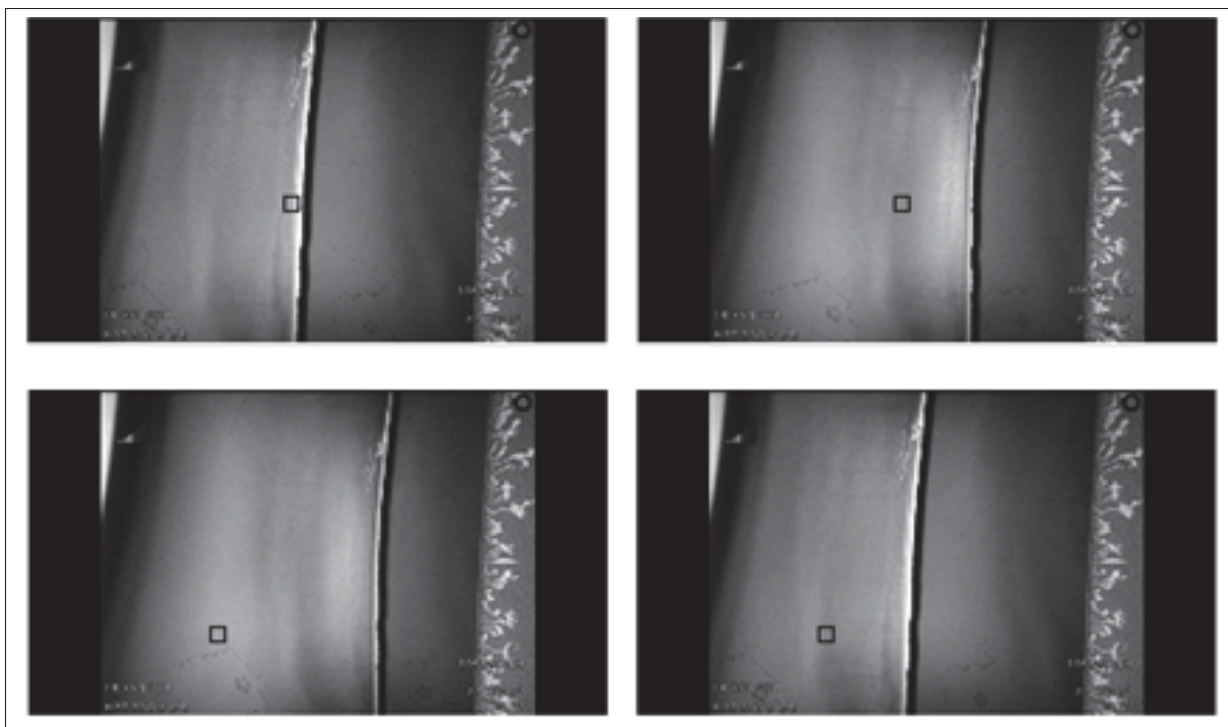


FIGURE 3.8 Carré du calcul d'intensité différent entre deux exécutions (chaque ligne montre une exécution)

### 3.3 Résolution des problèmes principaux avec la solution existante

L'idée globale de la solution existante est de calculer l'intensité dans un carré fixe pour des trames successives sélectionnées du flux provenant de la caméra en fonction du temps. Ce carré est fixé automatiquement selon le mouvement des lamelles détecté par le logiciel avant de commencer le comptage. Le passage d'une lamelle par le carré fixé est représenté par la détection d'une intensité maximale suivie d'une intensité minimale.

Quelques problèmes techniques empêchaient le bon fonctionnement du logiciel et donc la vérification de la méthodologie adoptée dans la solution existante pour résoudre la problématique. Les modifications présentées dans cette section ont été prioritairement effectuées afin de pouvoir analyser la solution existante et décider de son efficacité.

### 3.3.1 L'exception "*Out Of Memory*"

Pour ne pas rater la différence d'intensité importante qui traduit le passage de la lamelle par le carré fixe dans les trames, le logiciel doit calculer l'intensité dans cette position pour toutes les trames acquises par la caméra. Dans la solution existante, une trame est sélectionnée toutes les 150 ms. Et, le taux d'acquisition des trames est de 30 fps qui correspond à l'acquisition d'une seule trame chaque 33 ms. Sur une machine avec les caractéristiques suivantes : (Processeur : i5-7, RAM : 16 GB, Carte graphique : Nvidia GTX 1050 ti (4GB)), une trame sur 30 est sélectionnée pour le traitement. Dans le meilleur des cas, le temps nécessaire pour qu'une itération s'achève est inférieur à 33 ms, donc une trame sur cinq est sélectionnée pour le traitement ( $150/33 \approx 4.54$ ). Même dans ce cas, réellement impossible, il est très probable de rater la différence d'intensité importante qui correspond au passage de la lamelle par le carré fixe.

Pour garantir la détection du passage de la lamelle, nous avons décidé de traiter toutes les trames acquises par la caméra (ou encore, une trame sur deux). Mais, au bout de quelques minutes d'exécution, l'exception *Out Of Memory* a été jetée par OpenCv. La cause principale de cette exception est l'instanciation de l'algorithme qui calcule le flux du mouvement *IDenseOpticalFlow* dans une fonction qui est appelée autant de fois que de trames traitées. L'espace mémoire alloué à l'instance de l'algorithme n'est pas libérée à la fin de l'itération, et à la prochaine, l'algorithme est instancié une autre fois et ainsi de suite. Ce problème a été résolu en instanciant l'algorithme une seule fois. Cette unique instance de l'algorithme est utilisée pour estimer le mouvement dans toutes les itérations, tout au long de l'inspection.

Aussi, l'espace mémoire alloué aux variables comme les images et les matrices doit être libéré à la fin de chaque itération. Ceci est réalisé en utilisant des *using* et des *dispose* pour toutes les variables concernées.

### 3.3.2 Temps de calcul très long

Dans la solution existante, le logiciel devait capturer une trame toutes les 150 ms pour effectuer les traitements nécessaires, mais ceci n'était pas le cas. Sur la même machine, le temps de calcul

total écoulé entre l'acquisition d'une trame et la suivante était de 1043 ms dont 960 ms étaient réservées à la fonction qui exécute l'algorithme de calcul de mouvement entre deux trames successives. L'algorithme seul prend 855 ms, les 105 ms restantes sont prises par la boucle qui calcule la moyenne du flux du mouvement horizontal et vertical.

Pour pallier ce problème, deux modifications ont été apportées. Nous avons d'abord remplacé l'algorithme qui calcule le mouvement *IDensOpticalFlow* par son équivalent fourni par CUDA. Cette bibliothèque fournit trois instantiations possibles à l'algorithme *ICudaDensOpticalFlow* (OpenCV, 2021b). Parmi elles, *CudaSparsePyrLKOpticalFlow* a été utilisé. Ce dernier est basé sur la méthode de Lucas-Kanade pyramidale. Des changements doivent être apportés à la solution pour l'adapter à la nouvelle bibliothèque utilisée, entre autres la mise à jour de la version de EMGU.CV utilisée (passer de la version 3.2 à la version 4.9) et l'adaptation du code source en changeant les types des variables (par exemple Image passe à *CudaImage*). Cette modification a engendré la réduction du temps de traitement global d'une vidéo qui contient 1067 trames de 20 minutes à 1 minute 06 secondes. Le tableau 3.1 compare les temps d'exécution d'une seule itération avant et après les modifications apportées.

TABLEAU 3.1 Comparaison du temps de calcul nécessaire pour le traitement de deux trames successives (CPU vs GPU)

	Temps écoulé CPU (ms)	Temps écoulé GPU (ms)
Acquisitions de deux trames successives	1043	<b>80</b>
Estimation du mouvement	855	<b>4</b>
Boucle de la moyenne du mouvement	105	44

### 3.3.3 Traitement de la même trame deux fois

Dans la solution existante, chaque trame est traitée deux fois. L'image courante est traitée une fois en tant qu'image courante et une deuxième fois en tant qu'image précédente. Donc, les mêmes traitements sont appliqués à la même trame deux fois inutilement.

La nouvelle façon de faire est de conserver les images résultantes après le traitement de l'image courante et les utiliser dans la prochaine itération avec l'image nouvellement acquise.

### 3.3.4 Capture des trames en temps réel

Le flux vidéo capturé de la caméra injectée dans le moteur en temps réel était affiché dans l'interface via le composant *WebCameraControl*. Pour récupérer les trames à traiter, un évènement est déclenché chaque 150 ms. Suite à cet évènement, l'image est capturée à partir du composant et traitée en tant qu'image courante.

Nous avons remplacé ce composant par un *ImageViewer*. Ce dernier permet de déclencher un évènement à chaque fois qu'une trame est capturée de la source vidéo en temps réel. Suite à cet évènement, la trame récupérée directement de la caméra est affichée dans le composant approprié et est affectée à la variable correspondante pour déclencher le traitement. Cette méthode garantit l'acquisition de toutes les trames par le logiciel indépendamment du temps.

Les modifications effectuées dans la solution existante nous ont permis de l'exécuter en traitant toutes les trames acquises par la caméra. Ceci nous a laissé voir de près le fonctionnement de la solution et nous a permis de décider de son efficacité.

Les métriques de qualification des images sont calculées et affichées dans les champs appropriés en temps réel. Mais, la façon avec laquelle elles sont calculées est à améliorer.

Le compteur de lamelles ne fonctionne pas, même avec les modifications apportées à la solution existante. En effet, nous avons testé avec une vidéo qui contient 94 lamelles, 8 lamelles ont été comptées pour la première exécution et 4 ont été comptées pour la deuxième. La détection des lamelles n'est même pas garanti pour toutes les exécutions sur la même machine. Ceci est dû à la méthode adoptée pour le traitement des trames acquises.

## 3.4 Objectifs et matériel requis

L'objectif principal de ce projet est d'automatiser la qualification des images acquises et le comptage de lamelles. D'une part, le logiciel assurera une assistance lors de l'acquisition des trames en les qualifiant. Ceci assure l'acquisition de vidéos de bonne qualité (pour une post-inspection) ainsi que le déroulement de l'inspection dans des conditions optimales. D'autre

part, le logiciel notifiera l'inspecteur lorsque le tour complet est atteint ce qui le libère du comptage manuel pendant l'inspection. De cette manière, l'inspecteur se consacre à la détection des défauts qui est la tâche qui demande plus de concentration.

Pour pouvoir implémenter une solution qui traite les trames provenant de la caméra en temps réel, nous avons besoin d'un module d'acquisition vidéo qui permet de capturer le flux vidéo et le traiter sur l'ordinateur. Ce dernier doit avoir, au minimum, les caractéristiques suivantes :

- RAM : 16 G
- Processeur : i5, génération 7
- GPU : NVIDIA GTX 1050 ti (4 G)

En traitant les trames avec la carte graphique, la bibliothèque CUDA peut être utilisée. Elle fournit plusieurs algorithmes nécessaires pour résoudre la problématique et optimiser le temps d'exécution en prenant les calculs en charge au lieu du CPU.

### **3.5 Solution proposée**

Pour améliorer les métriques de qualification des images, nous changeons les algorithmes utilisés et optimisons le code source. Aussi, nous ajoutons une autre métrique de qualification des images qui est le contraste.

Concernant le compteur de lamelles, nous remplaçons toute la procédure adoptée dans la solution existante par d'autres traitements à effectuer pour pouvoir compter les lamelles qui passent et afficher le résultat en temps réel.

Notons que des bordures noires fixes sont toujours rajoutées aux trames capturées par la caméra avant d'être affichées dans l'interface. Avant de calculer les métriques de qualification des images, nous supprimons ces bordures car elles faussent les résultats obtenus. Par contre, nous traitons l'image avec bordures pour compter les lamelles car ça n'affecte pas le processus.

Le schéma présenté dans la figure 3.9 montre le fonctionnement global de la solution proposée. Dans ce qui suit, nous détaillons chacune de ces fonctions implémentées.

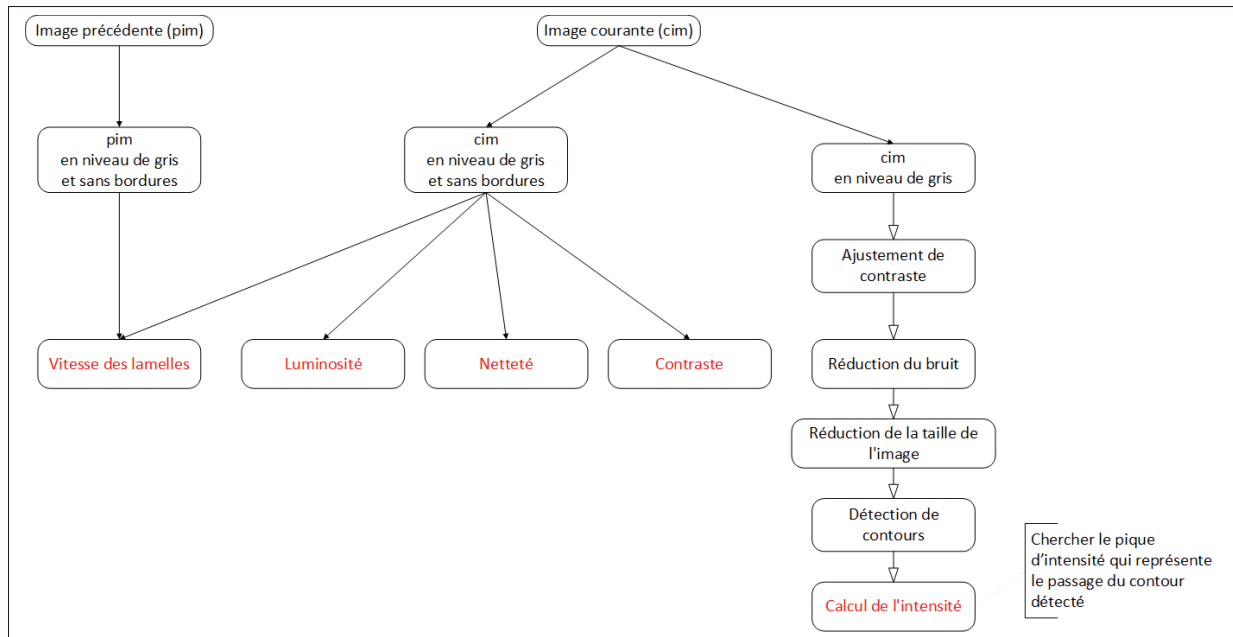


FIGURE 3.9 Fonctionnement global de la solution proposée

### 3.5.1 Qualification des images

#### 3.5.1.1 Luminosité

Cette métrique de qualification indique à l'utilisateur le niveau de lumière dans la trame acquise. L'inspecteur doit ajuster la quantité de lumière projetée dans le moteur si la lumière est très élevée ou pas suffisante pour effectuer une inspection dans des conditions optimales. Pour calculer cette métrique, la moyenne d'intensité de la trame traitée en niveau de gris est retournée.

#### 3.5.1.2 Netteté

Cette métrique indique la netteté des trames capturée. Le flou (opposé de la netteté) dans l'image dépend du taux d'acquisition des trames et peut être dû à plusieurs facteurs comme la défocalisation, secousses de la caméra et mouvement rapide des lamelles en cours d'inspection. Les bordures des lamelles sont généralement pointues et doivent être bien distinguées dans les images capturées. Ceci est important dans l'étape du comptage des lamelles vu que les trames

vont être filtrées pour pouvoir identifier les bordures des lamelles ainsi que dans l'étape de reconnaissance des défauts qui peuvent se présenter au niveau des bordures des lamelles.

Pour calculer cette métrique, nous avons choisi d'utiliser le filtre laplacien (Marr & Hildreth, 1980). Ce filtre nous permet de détecter les contours en calculant la deuxième dérivée de l'image en entrée. Si l'image est uniforme, ce filtre donnera zéro partout. Autour d'un contour, on aura : zéro loin du bord ; positif du côté le plus sombre du contour ; négatif du côté le plus clair et zéro à un certain point entre les deux valeurs (positive et négatives). De cette manière, nous sommes en mesure de déduire non seulement le nombre de contours présents dans l'image (croisements par zéro) mais aussi leur accentuation. Cette dernière représente l'amplitude des piques d'intensité (positive et négative) du contour détecté. Les piques d'intensité ne dépendent d'aucun seuil, plus le pique est grand, plus le contour est nette et vice versa. C'est pourquoi nous avons choisi ce filtre pour mesurer la netteté de l'image.

Après avoir appliqué le filtre laplacien sur l'image, nous calculons la valeur absolue du signal obtenu. Finalement, la moyenne des valeurs d'intensité de l'image traitée est retournée.

#### Algorithme 3.1 Calcul de la netteté de l'image

<p><b>Result :</b> Netteté</p> <ol style="list-style-type: none"> <li>1 Laplace = Appliquer le filtre laplacien sur l'image en entrée;</li> <li>2 AbsLaplace = Valeur absolue de Laplace;</li> <li>3 Netteté = Moyenne(absLaplace);</li> </ol>
--

### 3.5.1.3 Contraste

Le contraste indique si les intensités dans l'image sont bien distribuées. Pour le calculer, on doit d'abord obtenir l'histogramme de l'image. Ceci a été fait en utilisant la fonction *calcHist* fournie dans *EMGU.CV.CUDA*. Ensuite, nous calculons le pourcentage cumulatif des intensités dans l'image. Finalement, le contraste est calculé en effectuant la différence entre le dernier indice des 10% premières valeurs et le premier indice des 10% dernières valeurs (Matlab, 2021b).

### 3.5.1.4 Vitesse du mouvement

Pour décider si la vitesse avec laquelle l'inspecteur fait tourner l'étage est bonne, nous avons besoin d'estimer le mouvement entre les trames successives tout au long de l'inspection. Pour gagner du temps de calcul et pour que le déplacement de la lamelle soit considérable et peut être détecté et estimé clairement, nous avons décidé d'estimer le mouvement entre une trame et la troisième qui la succède.

L'algorithme *CudaSparsePyrLKOpticalFlow* est utilisé pour estimer le mouvement entre les trames sélectionnées. La première étape consiste à extraire les points à suivre à partir de la première trame par le détecteur *CudaGoodFeaturesToTrackDetector*. Ces points peuvent correspondre aux bordures des lamelles ou bien aux différences d'intensité présentes sur les lamelles comme le montre la figure 3.10. Ces points sont repérés par leurs coordonnées (x, y) et sauvegardés dans une liste. L'étape suivante consiste à trouver les points repérés dans la deuxième trame avec l'algorithme *CudaSparsePyrLKOpticalFlow* qui prend en paramètres les deux trames ainsi que la liste des points repérés. Pour chaque point, son correspondant est trouvé dans la deuxième image et sauvegardé dans une deuxième liste. En parcourant les deux listes de points, nous calculons la distance entre l'ancienne et la nouvelle position de chaque point avec la formule ci-après.

$$longueur = \sqrt{dy^2 + dx^2} \quad (3.8)$$

Si entre les deux trames, la lamelle principale s'est déplacée de quelque pixels dans un sens et elle reste toujours visible à l'écran, les vecteurs obtenues à partir des points repérés dans la première trame et trouvés dans la deuxième doivent être uniforme en termes de magnitude et sens comme le montre la figure 3.11a. Dans cette figure, les flèches (jaunes avec l'extrémité rouge) représentent le mouvement de la lamelle de gauche à droite entre les deux trames sélectionnées, dessinés sur la deuxième trame.

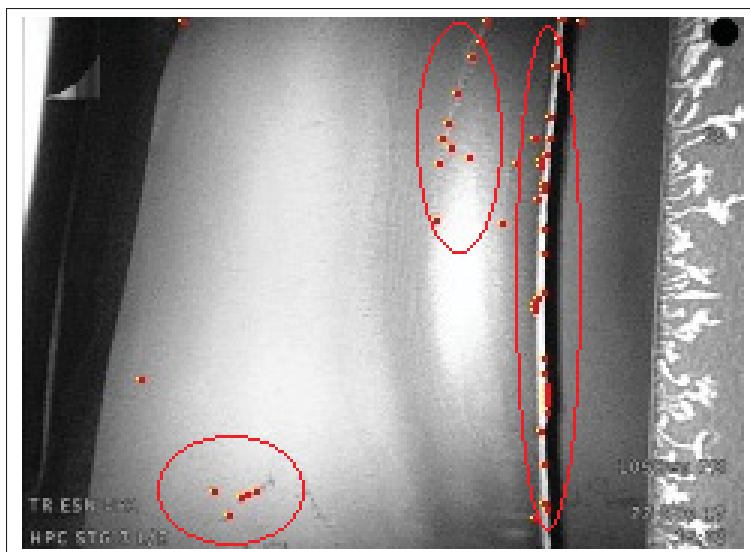


FIGURE 3.10 Points à suivre repérés dans la première trame

Par contre, la lamelle peut apparaître dans la première trame sélectionnée et disparaître dans la deuxième car elle est arrivée à l'extrémité, ou le contraire (apparaître juste dans la deuxième trame). Dans ce cas, les vecteurs obtenus ne représentent pas le mouvement car la lamelle est confondue avec l'extrémité de la trame. Dans ce cas, les vecteurs peuvent être très longs et vont dans des directions aléatoires. Ces trames ne peuvent être considérées pour le calcul de la vitesse de mouvement car elles ne le représentent pas (voir la figure 3.11b).

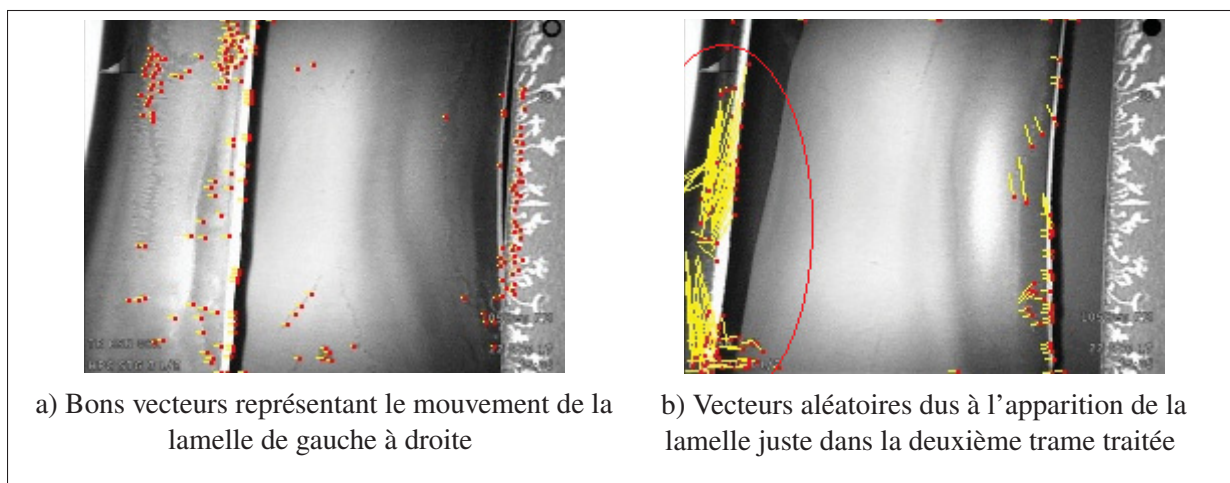


FIGURE 3.11 Bons et mauvais vecteurs

Pour pallier à ce problème, nous avons mis en place un filtre que nous appliquons sur l'ensemble de vecteurs obtenus après l'estimation du mouvement entre les deux trames traitées. D'abord, nous éliminons les vecteurs très longs, mal placés et mal orientés qui ont généralement une magnitude très longue par rapport à la majorité des vecteurs obtenus, qui représentent le mouvement.

Pour cela, nous éliminons les vecteurs aberrants en termes de magnitude en les comparant à un seuil maximal et minimal. Ces seuils sont obtenus selon l'ensemble de vecteurs actuel en calculant le premier, le troisième quartiles ( $Q_1$  et  $Q_3$ ) et l'inter-quartile ( $IQR$ ) de l'ensemble de magnitudes courant puis nous déduisons les seuils comme ceci (towards data science, 2020).

$$Seuil\_minimal = Q_1 - 1.5 \times IQR \quad (3.9)$$

$$Seuil\_maximal = Q_3 + 1.5 \times IQR \quad (3.10)$$

Les vecteurs ayant une magnitude supérieure au seuil maximal ou inférieure au seuil minimal sont éliminés.

Ensuite, nous éliminons les vecteurs aberrants en termes de coordonnées X et Y des vecteurs restants après la déduction des seuils minimal et maximal de la même manière.

Après ces deux étapes, si nous avons toujours des vecteurs ayant une magnitude plus grande que 45 pixels, les deux trames sélectionnées ne seront pas considérées pour le calcul de la vitesse car ça correspond à l'absence de la lamelle dans une des deux trames sélectionnées.

A la fin, la médiane des magnitudes des vecteurs restants est retournée. Cette valeur correspond à la distance (nombre de pixels) parcourue par la lamelle entre les deux trames traitées. Elle est multipliée par deux pour revenir aux dimensions originales des trames acquises.

La vitesse de la lamelle entre deux trames est obtenue en divisant la distance parcourue par le temps écoulé entre la capture des deux trames traitées (comme le montre l'équation suivante).

$$Vitesse = \frac{distance}{temps\_ecoulé} \quad (3.11)$$

Cette valeur représente la moyenne de la vitesse de la lamelle calculée chaque trois trames. Nous la divisons par trois pour obtenir la moyenne de la vitesse de la lamelle calculée pour toutes les trames acquises.

Finalement, la moyenne de cinq valeurs de vitesse successives est calculée et affichée dans le champs approprié dans l'interface. Ceci pour minimiser la variation rapide des valeurs affichées à l'écran.

### 3.5.2 Compteur de lamelles

Les moteurs inspectés à Siemens et les parties les composant sont différents, ce qui justifie la variété d'images qui peuvent être obtenue lors des inspections. Ci-dessous, la figure 3.12 montre l'intérieur de quelques moteurs. La bordure principale de la lamelle qui se déplace lors de l'inspection est encadrée (ou encerclée) en rouge. Les lamelles se déplacent tout au long de l'inspection dans un sens fixe (droit à gauche, bas en haut, ou le contraire). Elles se caractérisent par une bordure lumineuse suivie d'un ombre dû à la lumière projetée dans le moteur lors de l'inspection.

Pour compter les lamelles qui passent pendant l'inspection, le logiciel doit d'abord trouver la bordure principale de la lamelle qui se déplace et de trouver une position par laquelle seulement la bordure principale de la lamelle peut passer. Dans cette position fixe l'intensité est calculée pour toutes les trames acquises par la caméra. Le passage de la lamelle par cette position peut être détecté par un changement brusque d'intensité. Suite à ça, le compteur est incrémenté. Le nombre de ces piques étant le nombre de lamelles ayant passé par la position fixée.

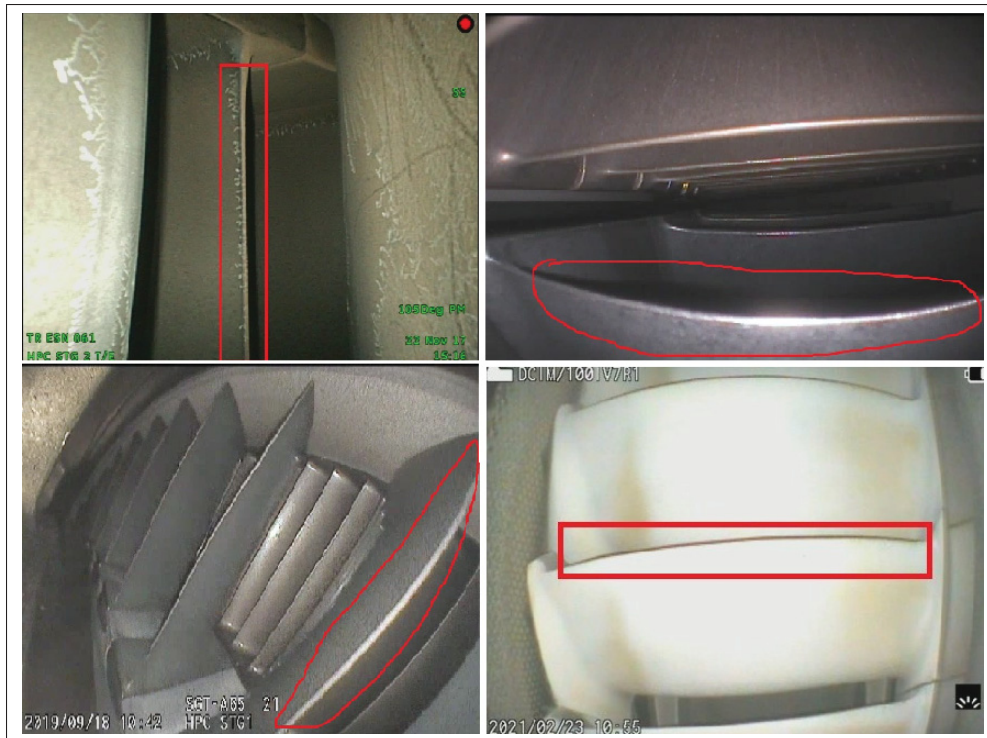


FIGURE 3.12 Images prises de différents moteurs, les lamelles principales sont encadrées en rouge

La première étape s'agit de choisir une bonne position par l'inspecteur dans l'interface. En règle générale, le carré doit être placé là où il y a beaucoup de contraste : la différence d'intensité entre la bordure de la lamelle et l'ombre qui se présente derrière. Trois cas peuvent se présenter :

1. Une seule lamelle principale apparaît dans la capture, comme montré dans la figure 3.13. Le carré peut être mis dans un endroit par lequel la lamelle passe. La région verte montre la zone possible ;
2. Une seule lamelle principale apparaît, avec la bordure supérieure ou inférieure. La deuxième bordure peut apparaître aussi (voir figure 3.14). Dans ce cas, il faut éviter de mettre le carré là où les bordures (autre que la principale) passent (zones rouges dans la figure suivante). Les zones vertes représentent de bonnes zones ;

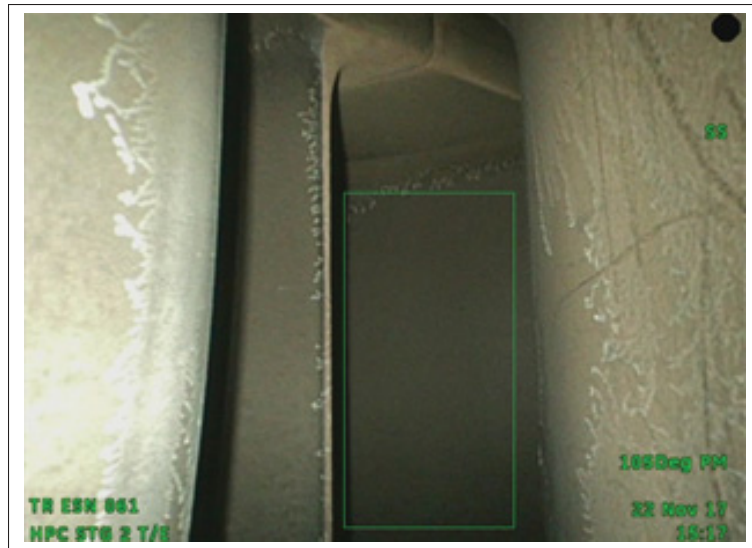


FIGURE 3.13 Une seule lamelle qui apparaît dans l'image

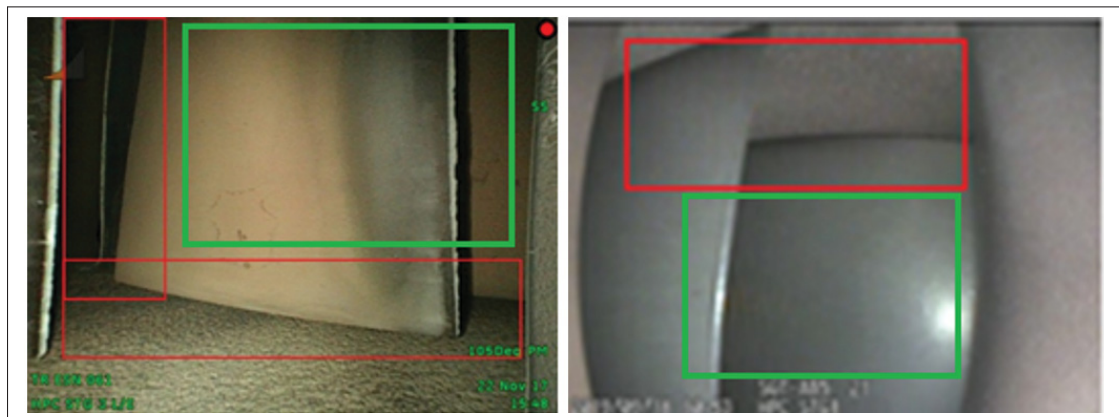


FIGURE 3.14 Une seule lamelle avec bordures supérieures, inférieures ou latérales (zones à éviter)

3. Plusieurs lamelles apparaissent, avec des lamelles qui ne bougent pas en arrière. Dans ce cas, il faut viser la lamelle la plus proche et éviter les positions par où les deux bordures de la lamelle principale peuvent passer. Des exemples de bonnes zones pour ce type de vidéos sont présentés dans les figures 3.15.

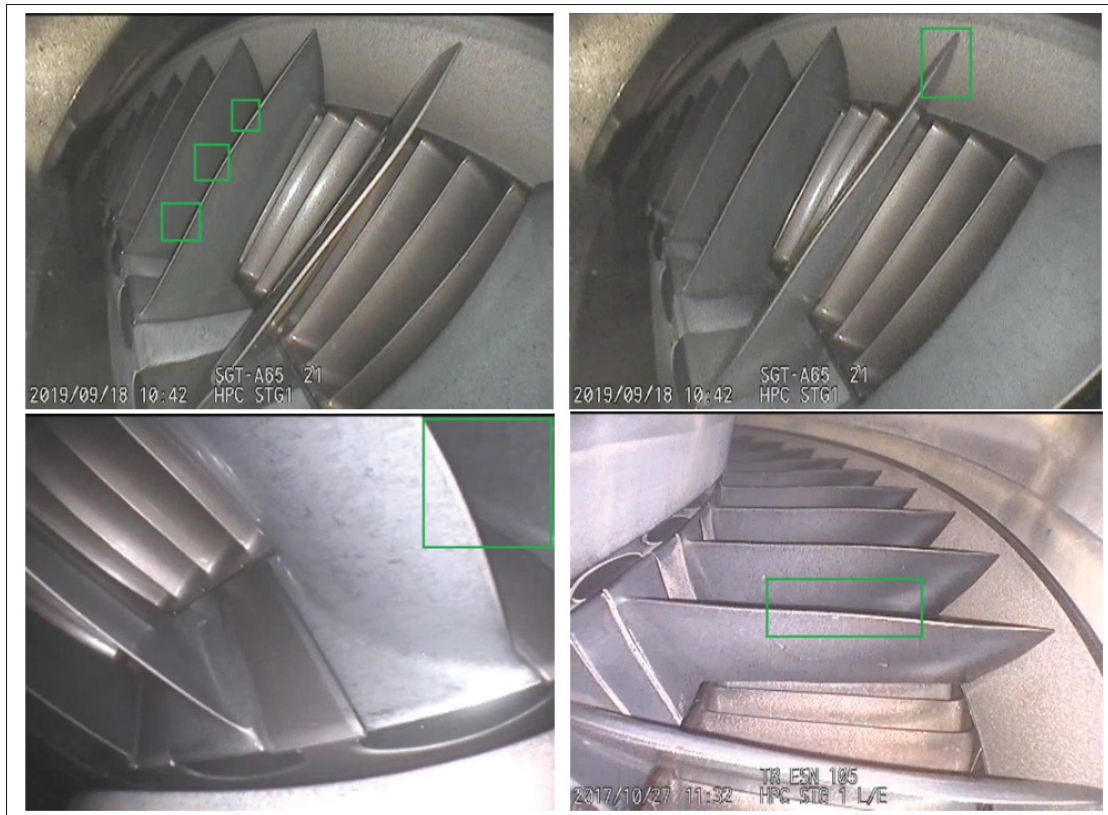


FIGURE 3.15 Plusieurs lamelles apparaissent, exemples de bonne zone

Sur la position choisie par l'utilisateur, un carré de  $10 \times 10$  pixels sera fixé pour calculer la moyenne d'intensité pour toutes les trames acquises par la caméra. Ce carré est dessiné sur l'interface pour permettre à l'inspecteur de voir par où la lamelle doit passer pour qu'elle soit comptabilisée.

Nous avons décidé de ne pas calculer l'intensité dans les trames converties en niveau de gris car l'intensité de la bordure principale de la lamelle ainsi que l'ombre peuvent changer d'intensité (donc la différence d'intensité à chercher) d'une vidéo à une autre, ça dépend de beaucoup de paramètres comme la luminosité projetée et la couleur des lamelles en cours d'inspection. Aussi, si nous calculons l'intensité dans des trames en niveau de gris, nous devons prendre le sens du mouvement en considération. En effet, dans la figure 3.13, si la lamelle se déplace de gauche à droite, le carré recevra l'intensité de l'ombre avant celle de la bordure de la lamelle, i.e. le

gradient entre les deux intensités sera positif. Dans le cas contraire, le gradient est négatif. Pour éviter ces problèmes, nous avons décidé de détecter les contours dans les trames et travailler avec des images noires et blancs où les pixels n'ont que deux valeurs possibles : 0 pour le noir qui représente les surfaces et 255 pour le blanc qui représente les bordures. Pour cela, l'algorithme Canny est utilisé (voir 2.3.5). Aussi, ce filtre permet de détecter le long des bordures même s'il n'y a pas une grande différence d'intensité entre la lamelle et l'arrière-plan et même si les bordures sont fines. Aussi, il y a un grand choix de seuils minimal et maximal pour extraire les bordures. Plusieurs tests ont été effectués sur toutes les vidéos disponibles.

Avant d'appliquer le filtre Canny, nous avons besoin de réduire le bruit présent dans les images. Classiquement, le filtre gaussien est utilisé. Dans notre cas, les bordures sont très importantes et beaucoup de bruit peut se présenter sur la surface des lamelles, nous avons choisi d'utiliser le filtre bilatéral car il garde les bordures en lissant toute la surface des lamelles (voir 2.2.1.1). Deux paramètres sont nécessaires pour le filtre bilatéral : *sigmaColor* et *sigmaSpatial*. *sigmaColor* filtre l'image dans l'espace colorimétrique. Une valeur plus élevée du paramètre signifie que des couleurs plus éloignées dans le voisinage des pixels (voir *sigmaSpace*) seront mélangées, résultant en de plus grandes zones de couleur semi-égale. *sigmaSpatial* filtre l'image dans l'espace de coordonnées. Une valeur plus élevée du paramètre signifie que les pixels plus éloignés s'influenceront mutuellement tant que leurs couleurs sont suffisamment proches (voir *sigmaColor*) (OpenCV, 2021c). Après plusieurs tests sur les trames tirées de l'ensemble des vidéos disponibles, nous avons fixé la valeur de *sigmaColor* à 200 et celle de *sigmaSpatial* à 5.

La figure 3.16 montre la trame originale (convertie en niveau de gris) lissées avec le filtre bilatéral. Les bruits présents sur la surface de la lamelle sont bien lissés tandis que la bordure de la lamelle est conservée. Les bordures de la trame lissée en utilisant le filtre Canny sont montrées dans la figure 3.17.

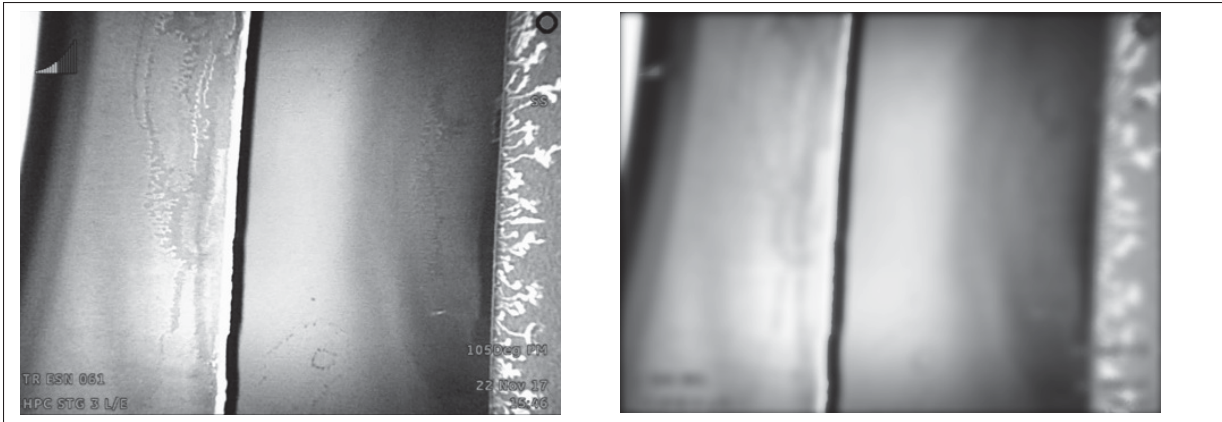


FIGURE 3.16 Trame originale (gauche) lissée avec le filtre bilatéral (droite)



FIGURE 3.17 Bordures extraites à partir de la trame lissée avec le filtre Canny

La fonction qui calcule la moyenne d'intensité dans le carré sélectionné est appelé après la sélection de celui-ci et prend en paramètre la trame courante pour laquelle les bordures sont détectées. La moyenne d'intensité est égale à zéro jusqu'à ce que la bordure de la lamelle passe par la position choisie. Le graphe présenté dans la figure 3.18 montre les moyennes intensités calculées pour une série de trames où six lamelles passent. Les six pics représentent le passage des lamelles.

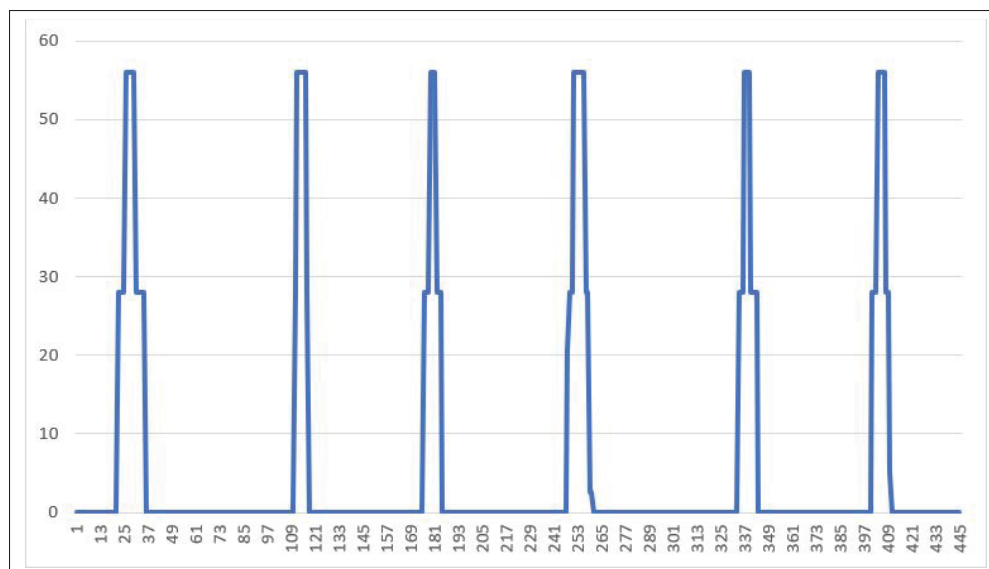


FIGURE 3.18 Piques d'intensité représentant le passage de six lamelles par une position fixée

Pour détecter les piques d'intensité représentant le passage des lamelles en temps réel, nous devons sauvegarder un certain nombre de valeurs successives pour pouvoir faire la différence. Pour cela, une liste de taille 10 est utilisée. Initialement, les valeurs d'intensité sont ajoutées dans cette liste au fur et à mesure jusqu'à son remplissage. Une fois remplie, nous calculons le gradient de chaque valeur de la première moitié de la liste avec les cinq qui la succèdent. Si cette valeur de gradient est supérieure à 13, nous considérons qu'une lamelle a été détectée dans cette succession de valeurs d'intensités. Dans ce cas, nous vidons la liste et la remplissons de nouveau pour chercher la prochaine lamelle. Au cas où aucune lamelle n'est détectée dans la suite de valeurs courante, nous supprimons le premier élément et ajoutons la moyenne d'intensité de la dernière trame acquise.

Un cas exceptionnel peut se produire si la largeur de l'ombre est supérieure à la taille du carré. Ce dernier peut se placer entre les deux bordures délimitant l'ombre. Dans ce cas, nous aurons un pique d'intensité représentant le premier contour suivi de plusieurs intensités nulles trouvées à l'intérieur de l'ombre suivies par un autre pique d'intensité correspondant au deuxième contour.

Par exemple, dans la liste **0 28 28 0 0 0 0 28 28 0 0 0**, les deux contours sont comptabilisés alors qu'elles ne représentent qu'une seule lamelle.

Aussi, si une pause survient lors de l'inspection, exactement au moment où le carré est sur la lamelle. Dans ce cas, les valeurs d'intensité trouvées sont toutes supérieures à zéro et une différence de plus de 13 peut se présenter dans cet ensemble de valeurs. Ce qui représente une détection d'une lamelle alors que ce n'est pas le cas.

Pour résoudre ces deux problèmes, un certain nombre de zéros successifs doit séparer les lamelles trouvées, ce qui représente la surface de la lamelle parcourue. Cette valeur est fixée à 9.

Notons que, rarement, les lamelles peuvent contenir des tâches qui peuvent être détectées comme bordure. Il y a une possibilité que ces fausses bordures soient comptabilisées si elles passent par le carré. D'autre part, il est courant que l'inspecteur fasse tourner l'étage dans le sens contraire pour vérifier l'état d'une lamelle déjà passée ou pour prendre des captures du défaut déjà rencontré. Dans ce cas, les lamelles vont être comptabilisées en double. Un bouton est disponible sur l'interface pour désactiver et réactiver le compteur de lamelles au besoin.

### **3.6 Présentation de l'application**

La figure suivante montre l'interface graphique après l'avoir amélioré. Dans le côté gauche de l'interface, les trames capturées par la caméra sont affichées en temps réel. A droite, dans les listes déroulantes disponibles, l'utilisateur peut choisir les informations concernant le moteur, le module et le stage à inspecter ainsi que les informations du boroscope utilisé. Ces listes déroulantes contiennent toutes les informations concernant tous les moteurs pouvant être inspecté à Siemens. Au-dessous de ces listes, le nombre de lamelles est affiché en temps réel. Suite à la sélection des informations du stage à inspecter, le nombre total des lamelles est déduit et le champ devient vert lorsque ce nombre est atteint. Juste au-dessous, les métriques qualifiant l'image capturée, avec un code couleur, sont affichées en temps réel. A savoir : le contraste, la netteté, la luminosité et la vitesse. Les taux d'acquisition et de traitement sont affichés à titre indicatif. Aussi, la date et l'heure actuelles sont affichées.

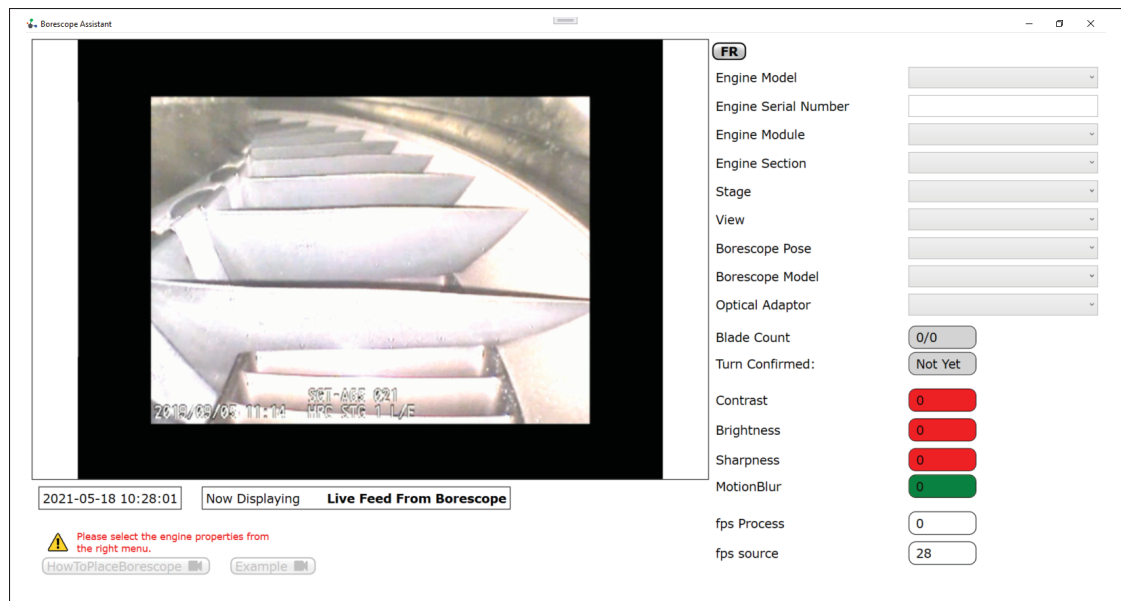


FIGURE 3.19 Interface graphique améliorée

Les instructions suivantes peuvent guider l'inspecteur à utiliser le logiciel optimalement.

1. Brancher le boroscope et le module d'acquisition vidéo ;
2. Brancher le module d'acquisition vidéo à l'ordinateur ;
3. Lancer l'application ;
4. Remplir les informations concernant le stage à inspecter ;
5. Vérifier les valeurs des trois métriques de qualification des images acquises : contraste (contrast), luminosité (brightness) et netteté (sharpness). Si les champs sont verts alors les images capturées sont bonnes ;
6. Cliquer sur l'espace réservé à l'affichage de la vidéo acquise pour choisir une bonne position ;
7. Remplir les informations concernant l'étage à inspecter. Suite à ça, le bouton *enregistrer* apparaît dans l'interface ;
8. Appuyer sur le bouton *enregistrer* pour commencer l'enregistrement de la vidéo. Dès que l'enregistrement est lancé, le bouton *arrêter l'enregistrement* est affiché dans l'interface. Aussi le bouton *pause* qui sert à mettre le compteur en pause ;
9. Si vous voulez réinitialiser le compteur, peser sur le bouton *arrêter l'enregistrement* ;

10. Si vous voulez mettre le compteur en pause pendant que vous faites tourner le stage, appuyer sur le bouton *stop* affiché à côté du compteur. Pour reprendre, peser sur le bouton *resume* ;
11. Faire tourner le stage avec une bonne vitesse. Pour vérifier si la vitesse est bonne tout au long de l'inspection, consulter la métrique *Speed*, le champ doit être vert ;
12. Le nombre de lamelles est incrémenté en temps réel dans le champ *blade count* quand la lamelle passe par la position choisie ;
13. Si la couleur de ce champ (*blade count*) devient verte, ça veut dire que le nombre total des lamelles du stage sous inspection est atteint ;
14. Appuyer sur le bouton *stop recording* à la fin de l'inspection.

### 3.7 Conclusion

Nous avons présenté, dans ce chapitre, la solution existante ainsi que la nouvelle méthodologie proposée pour résoudre la problématique. Aussi, nous avons présenté l'interface du logiciel et le manuel de son utilisation.

En comparant les deux schémas présentés dans les figures 3.6 et 3.9 (présentant le fonctionnement de la solution existante et celui de la nouvelle méthodologie proposée respectivement), nous pouvons voir clairement que nous avons changé toute la logique et les traitements appliqués sur les images acquises en implémentant une nouvelle solution qui nous permet d'avoir des résultats satisfaisants. Pour la qualification des images, nous avons optimisé le code source des métriques existantes et rajouté une nouvelle métrique qui est le contraste. Concernant le compteur de lamelles de la solution existante, la première étape consistait à trouver l'emplacement de calcul d'intensité automatiquement selon le mouvement des lamelles, ce qui a été remplacé par l'intervention humaine dans notre solution. Ce qui garantit la sélection du bon emplacement pour ce calcul étant très important pour le comptage de lamelles. D'autre part, la détection du changement d'intensité se faisait avec des images originales acquises mises en niveau de gris. Nous avons remplacé ces images par des images noires avec des contours blanches. Ce qui unifie la surface de la lamelle et met l'accent de la bordure de la lamelle. De cette manière, le passage

de la lamelle (blanche) est traduit par un changement d'intensité semblable pour tout les types des vidéos traités, ce qui n'a pas été pris en considération dans la solution existante.

Dans le chapitre suivant, nous présentons et discutons les tests effectués et les résultats obtenus.



## CHAPITRE 4

### TESTS, RÉSULTATS ET DISCUSSION

#### 4.1 Introduction

Trois aspects doivent être considérés pour tester le logiciel. D'une part, il faut tester et valider les intervalles des métriques de qualification des images. D'autre part, il faut tester le compteur de lamelles ainsi que la procédure de mise en place du carré (la position fixée) pour le calcul d'intensité.

Dans ce chapitre, nous présentons les tests effectués pour valider chacun de ces aspects et discutons les résultats obtenus. Nous commençons par la présentation de la base de données des vidéos fournies par Siemens pour effectuer les tests localement. Ensuite, nous présentons la méthodologie des tests ainsi que les résultats obtenus pour chacun des trois aspects. Nous récapitulons par la démonstration qui consiste à l'inspection d'un moteur en temps réel, à Siemens, en utilisant notre logiciel.

#### 4.2 Base de données des vidéos

28 vidéos différentes ont été fournies de la part de Siemens pour effectuer les tests du logiciel. Ces vidéos ne sont pas capturés avec le même boroscope et sont différentes en termes de résolution, illumination, types de lamelles, angles et position de capture.

Avant de passer aux tests, la base de données des vidéos a été préparée. Nous avons fractionné quelques vidéos car elles contiennent plusieurs vidéos différentes en termes d'orientation de la caméra, pour le même étage. En plus des vidéos fournies, d'autres ont été capturées lors d'une démonstration sur le plancher et ont été rajoutées à la base de données. Au final, 61 vidéos sont disponibles pour les tests.

Notons que quand les vidéos passent par le module d'acquisition vidéo, le contraste est amélioré (comme le montre la figure 4.1). Sachant que les vidéos fournies ont été enregistrées directement

par le boroscope dans la clé USB branchée, nous les avons injecté dans le module d'acquisition vidéo et les avons ré-enregistré. De cette manière, nous pouvons simuler l'exécution en temps réel localement et tester le logiciel avec toutes les vidéos disponibles.

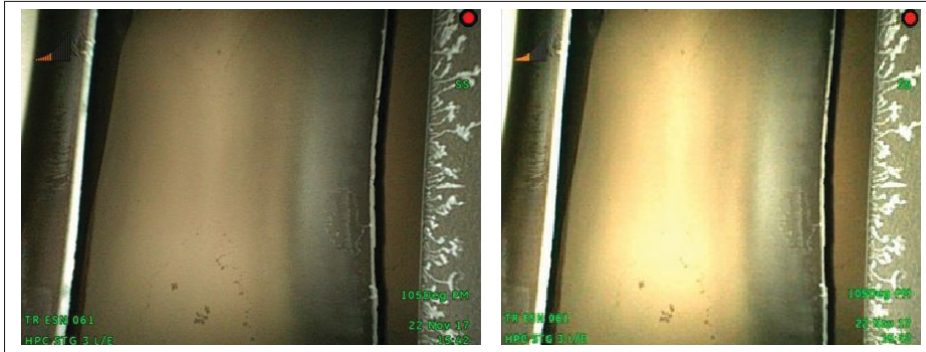


FIGURE 4.1 Différence entre la trame enregistrée avec la caméra (gauche) et la même trame ré-enregistrée après injection dans le module d'acquisition vidéo (droite)

### 4.3 Métriques de qualification des images

Pour les métriques de qualification des images, nous effectuons des tests unitaires avec des images générées localement à partir de trois trames tirées des vidéos disponibles (voir la figure 4.2). Nous avons appliqué des filtres sur ces images dans le but de modifier leur qualité en terme de netteté, luminosité et contraste. Nous calculons la métrique correspondante à chaque fois que nous modifions l'image. En ce qui concerne la vitesse des lamelles, nous l'avons testé en calculant la distance parcourue par la lamelle car nous ne pouvons pas fixer le temps écoulé entre l'acquisition des trames. D'autre part, nous calculons les quatre métriques pour chaque vidéo et prenons la moyenne des valeurs.

A partir de là, nous déduisons un intervalle d'acceptance pour chaque métrique dépendamment de la qualité des images obtenues. Ces intervalles sont proposés et validés avec l'inspecteur lors de la démonstration à Siemens. Les détails sont présentés dans ce qui suit.

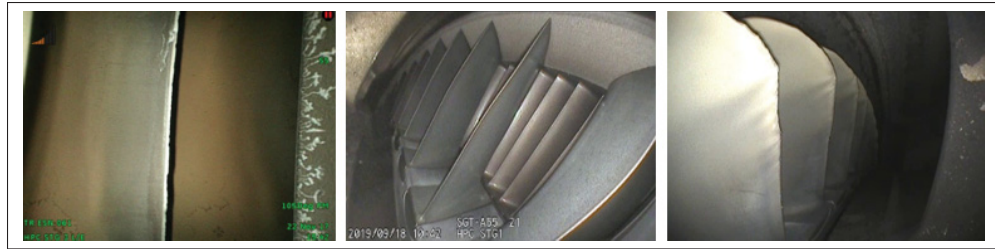


FIGURE 4.2 Les trois images choisies pour les tests unitaires des métriques netteté, luminosité et contraste (dans l'ordre trame 1, trame 2 et trame 3)

### 4.3.1 Tests unitaires

#### 4.3.1.1 Netteté

En termes de netteté, nous avons flouté les trois trames avec le filtre gaussien en utilisant la fonction *imgaussfilt(A,sigma)* disponible dans Matlab. Cette fonction filtre l'image *A* avec un noyau de lissage gaussien 2-D avec un écart type spécifié par *sigma*. Plus *sigma* est élevé, plus l'image est floue (Matlab, 2021c). Nous avons varié la valeur de *sigma* de 0.5 à 4. Les valeurs que nous avons obtenues pour les images modifiées sont présentées dans le tableau 4.1. Les images obtenues à partir des 3 trames sont présentées dans la figure 4.3.

TABLEAU 4.1 Netteté obtenue pour les images modifiées

Sigma	Trame 1	Trame 2	Trame 3
Image originale	7.52	17.01	5.49
0.5	5.449	10.029	3.18
1	3.119	4.75	2.01
2	1.84	2.19	1.38
3	1.48	1.63	1.2
4	1.27	1.44	1.12

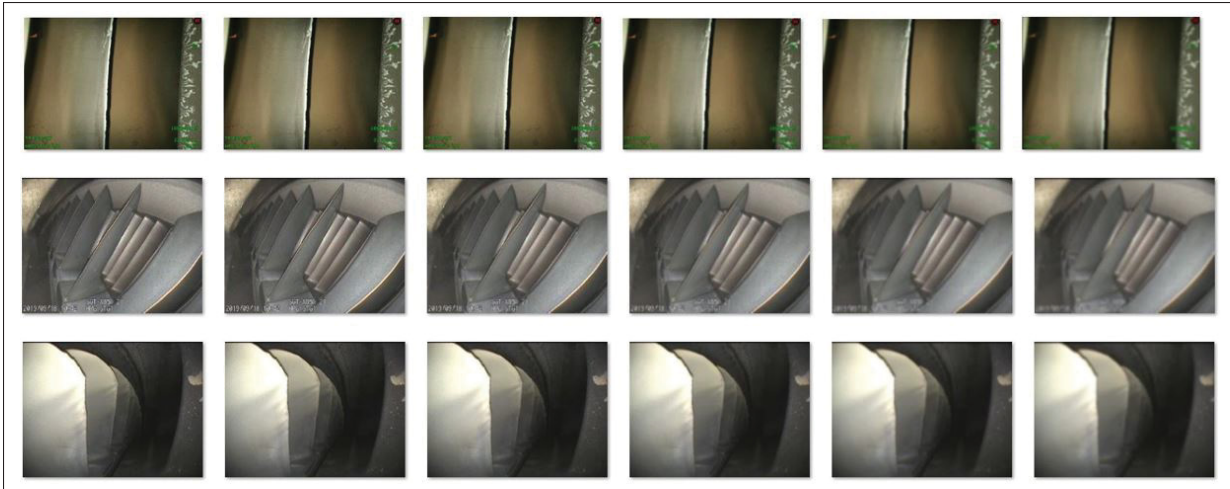


FIGURE 4.3 Images floutée obtenues à partir des trame 1, 2 et 3. Dans chaque ligne, on trouve l'image originale, sigma = 0.5, 1, 2, 3, 4

#### 4.3.1.2 Luminosité

En termes de luminosité, nous avons rajouté et soustrait de l'intensité aux pixels des trois images choisies. Les images obtenues à partir de la trame 1 sont présentées dans la figure 4.4. Les valeurs obtenues sont présentées dans le tableau 4.2.

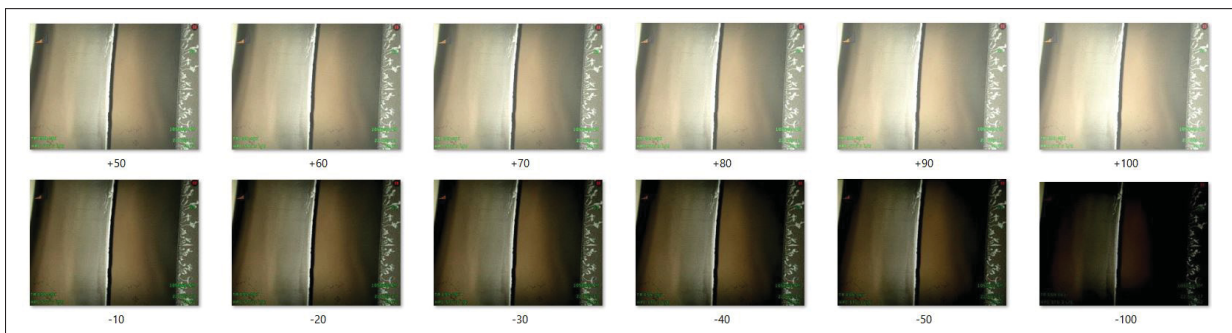


FIGURE 4.4 Images obtenues à partir de la trame 1. L'intensité rajoutée ou soustraite est mentionnée sous chaque image

TABLEAU 4.2 Luminosité obtenue pour les images modifiées

Image	Trame 1	Trame 2	Trame 3
Originale	90	107	99
-10	80	97	89
-20	71	87	79
-30	61	77	69
-40	52	67	61
-50	44	58	54
-100	13	20	29
+50	140	156	145
+60	150	166	155
+70	159	176	162
+80	169	186	170
+90	179	195	178
+100	189	204	186

#### 4.3.1.3 Contraste

Pour le contraste, nous l'avons modifié pour chaque image, après l'avoir converti en niveau de gris, en utilisant la fonction  $J = imadjust(I, [low\_in\ high\_in])$ . Cette fonction fait correspondre les valeurs d'intensité dans  $I$  aux nouvelles valeurs dans  $J$  de telle sorte que les valeurs entre  $low\_in$  et  $high\_in$  correspondent aux valeurs comprises entre 0 et 1 (Matlab, 2021a). Les intervalles que nous avons utilisé et les résultats obtenus sont montrés dans le tableau 4.3. Les images obtenues à partir de la trame 1 sont présentées dans la figure 4.5.

TABLEAU 4.3 Contraste obtenu pour les images modifiées

Image	Trame 1	Trame 2	Trame 3
Originale	101	98	181
[0, 1]	101	98	181
[0.2, 0.8]	113	159	71
[0.25, 0.75]	91	167	12
[0.3, 0.7]	60	136	1

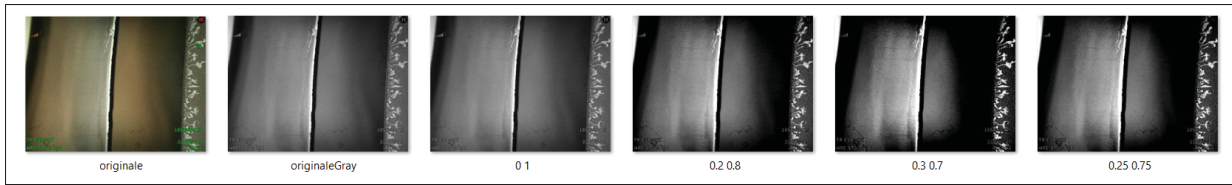


FIGURE 4.5 Modification du contraste de la trame 1. Les intervalles sont mentionnés sous chaque image

#### 4.3.1.4 Vitesse

Nous avons calculé la distance parcourue par les lamelles pour la vidéo à partir de laquelle la trame 1 a été sélectionnée pour les tests. D'une part, nous avons calculé la distance entre chaque deux trames acquises. Ensuite, Nous l'avons calculé entre chaque 4 trames acquises. Pour le premier cas, la moyenne de distance parcourue est de 7 pixels. Pour le deuxième cas, la moyenne de distance parcourue est de 13 pixels.

Si le temps écoulé pour l'acquisition des trames est le même dans les deux cas, la vitesse obtenue dans le deuxième cas est plus grande que la première, ce qui confirme le bon fonctionnement de la méthode de calcul.

#### 4.3.2 Tests avec les vidéos disponibles

Nous avons exécuté la solution en temps réel en simulant le fonctionnement sans boroscope comme présenté dans la figure 3.7. Ceci nous a permis d'observer la netteté, la luminosité, le contraste et la vitesse calculés tout au long de la vidéo. Nous avons noté la moyenne des valeurs obtenues pour chacune (sauf la vitesse) dans le tableau 4.7. Ce dernier contient les valeurs des métriques de qualification des images obtenues pour toutes les vidéos.

En se basant sur les valeurs présentées dans le tableau et aussi celles obtenues à partir des tests unitaires, nous proposons les intervalles d'acceptance suivants (avec : **acceptable** *moyennement acceptable* inacceptable).

- Netteté : [0, 5] [7 et plus]

- Luminosité : [0, 75] / 75, 100 / **[100, 210]** / 210, 230 / [230, 255]
- Contraste : [0, 80] / 80, 90 / **[90, 255]**
- Vitesse : **[0, 25]** / 25, 40 / [40 et plus] (px/s)

#### 4.4 Compteur de lamelles

Avant de passer aux tests du compteur de lamelles, nous devons d'abord nous assurer que le filtrage des images est optimal. Nous devons nous assurer que les valeurs des paramètres des filtres utilisés pour passer d'une trame brute acquise de la caméra à une image représentant les contours détectés sont bonnes pour toutes les vidéos de la base de données.

Dans notre processus, nous avons deux filtres pour lesquels les valeurs des paramètres doivent être fixées : le filtre bilatéral qui sert à éliminer le bruit tout en gardant les bordures des lamelles et le filtre Canny qui les extrait. Après plusieurs tests, nous avons retenu les valeurs suivantes :

- Filtre bilatéral(sigmaColor = 200, sigmaSpatial = 5);
- Filtre Canny(seuilMin = 70, seuilMax = 180).

Pour s'assurer du bon fonctionnement du compteur de lamelles, nous l'avons isolé dans une solution que nous exécutons localement avec toutes les vidéos disponibles dans la base de données présentée plus haut en utilisant un script batch. Ce dernier est un fichier texte contenant des lignes de commandes exécutées en séquence, l'une après l'autre. Chaque ligne de commande prend en paramètre le nom de la vidéo et la position du carré fixée pour le calcul de l'intensité. Pour chacune des vidéos, nous avons choisi deux bonnes positions différentes pour lesquelles nous comptons les lamelles dans la vidéos entière et comparons le résultat avec le nombre de lamelles réel compté manuellement. Nous avons aussi testé le compteur en temps réel en simulant le fonctionnement sans boroscope comme présenté dans la figure 3.7. Ceci nous permet de vérifier que l'affichage se fait correctement et en temps réel dans l'interface.

Pour vérifier le fonctionnement du compteur de près, nous enregistrons toutes les trames au moment de l'incrémement du compteur. Comme ça, nous pouvons vérifier précisément si le compteur a été incrémenté à cause du passage d'une bordure principale (cas correct) comme

montré dans la figure 4.6a ou bien une bordure latérale, un bruit ou une tâche qui n'a pas été filtrée (voir la figure 4.6b).

Les trames filtrées obtenues pour chaque vidéo et le nombre de lamelles pour les deux positions choisies pour chaque vidéo sont présentés dans la section 4.7.

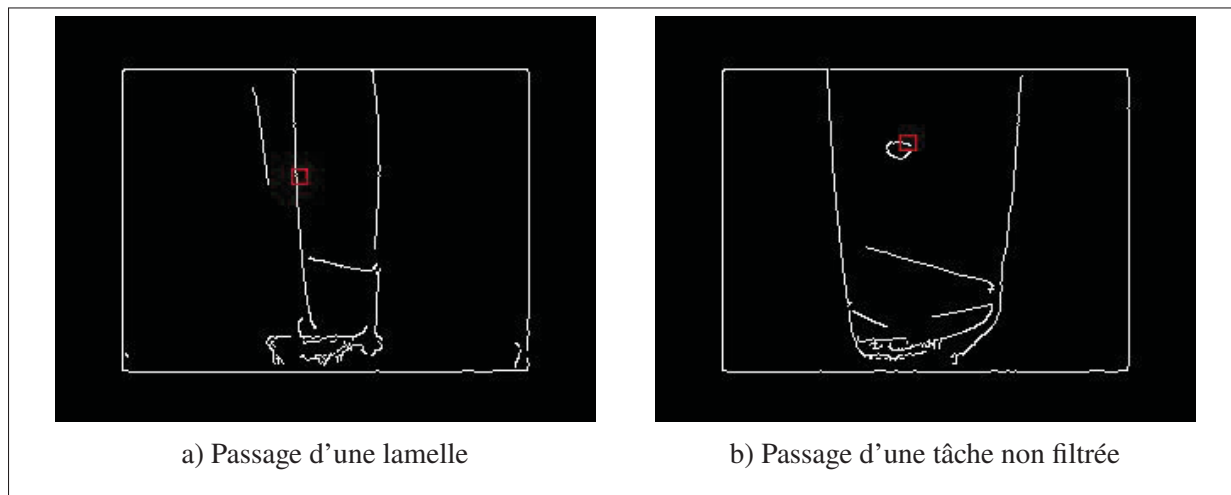


FIGURE 4.6 Détection du passage d'une bordure par le carré fixé

#### 4.5 Procédure de mise en place du carré

Comme l'intervention humaine est nécessaire pour choisir une bonne position qui sert à compter les lamelles, la procédure présentée dans la section 3.5.2 a été établie.

Il est important de s'assurer que l'exécution de cette procédure, par n'importe quelle personne, mène à des positions adéquates pour lesquelles le résultat du compteur est correct. En effet, le logiciel va être utilisé par de nombreux inspecteurs qui doivent confier le comptage de lamelles au logiciel pour se concentrer uniquement sur la détection des défauts. Pour cela, une étude a été menée sur 15 participants pour lesquels un énoncé présenté dans une page Web a été proposé. Cet énoncé comporte trois sections. Chacune contient 10 extraits de vidéo et 10 images correspondantes sur lesquelles ils doivent choisir une bonne position pour le comptage de

lamelles. Cette étude a été approuvée par le comité d'éthique de l'ETS. L'annexe III contient le formulaire d'information et de consentement présenté aux participants.

Après avoir lu et compris la procédure de mise en place du carré, le participant regarde la vidéo qui lui permet d'identifier la lamelle principale, le sens du mouvement, la bonne zone ainsi que les zones à éviter à cause des bordures secondaires (supérieures, inférieures, apparition de la deuxième bordure de la même lamelle). Ensuite, il clique sur l'image correspondante à la vidéo visualisée pour choisir la position qu'il juge bonne pour le comptage de lamelles. L'effet de cliquer sur l'image fera apparaître la prochaine image (partie de l'énoncé). Pour toutes les parties proposées, une bonne position doit être sélectionnée. A la fin, un email contenant toutes les coordonnées des positions choisies récupérées automatiquement ainsi que les adresses auxquelles ces coordonnées doivent être envoyées est rédigé, il suffit de l'envoyer via l'application d'envoi de courriel simplement en cliquant sur le bouton « Envoyer ». Les coordonnées récoltées de cette étude ont été utilisées pour compter les lamelles dans les vidéos entières correspondantes aux extraits présentés dans l'énoncé.

Nous avons choisi de recruter plusieurs participants pour juger de la répétabilité inter-participant. Celle-ci nous permet de vérifier si, pour les mêmes images, les participants tendent à choisir des positions appartenant à la bonne zone. Aussi, les trois parties de l'énoncé contiennent les mêmes 10 images. Dans ce cas, chaque image est présentée 3 fois au même participant. De cette manière, nous pouvons voir si le même participant choisit des positions semblables d'une fois à l'autre (répétabilité intra-participant).

#### **4.5.1 Tests et résultats obtenus**

Pour toutes les vidéos de l'énoncé, nous traçons les 45 positions (sélectionnées par tous les participants) sur les images fournies dans l'énoncé comme le montre les figures suivantes (4.7 à 4.11). Pour analyser les résultats selon les positions choisies, nous avons catégorisé les positions en :

- Mauvaises positions : encadrées avec un rectangle gris foncé ;

- Des positions pour lesquelles des lamelles ont été ratées : encadrées en orange ;
- Des positions pour lesquelles des lamelles de plus ont été comptabilisés : encadrées en rouge.

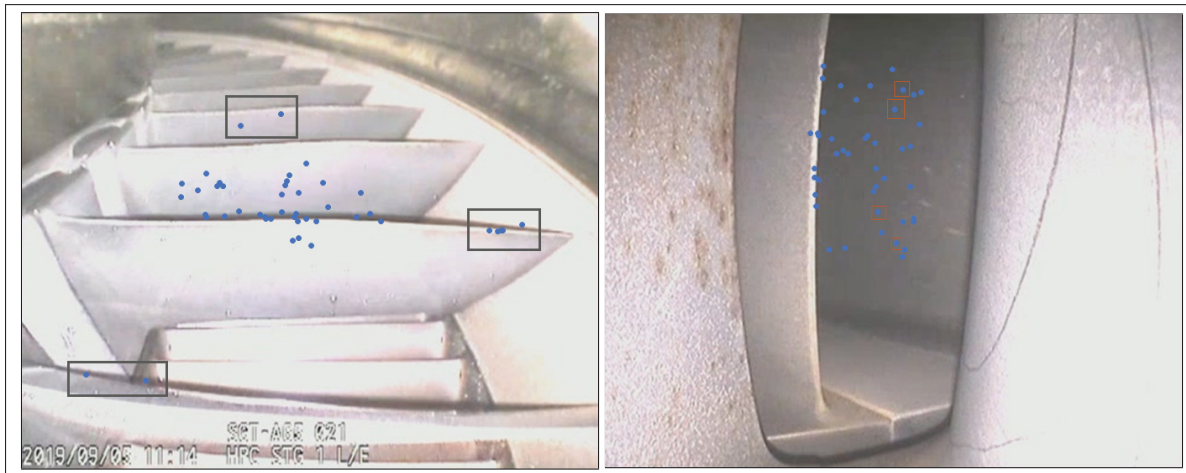


FIGURE 4.7 Positions choisies pour les deux premières vidéos

Pour la première vidéo représentée dans la figure 4.7, 36 positions parmi 45 sont correctes, ayant un nombre de lamelles correct. Les neuf autres positions ont été mal choisies, elles sont encadrées en gris foncé dans la figure correspondante. Les deux points en haut sont très éloignées de la lamelle principale, les lamelles y passent très rapidement ce qui rend difficile la détection de la différence d'intensité à cet endroit. Les deux bordures des lamelles passent par les sept autres positions, c'est pourquoi elles ne sont pas bonnes pour le comptage de lamelles. Pour la deuxième vidéo, toutes les positions choisies sont bonnes. Sauf que, une seule lamelle a été ratée pour chacune des quatre positions encadrées en orange.



FIGURE 4.8 Positions choisies pour les vidéos trois et quatre

Toutes les positions choisies pour la vidéo 3 sont bonnes sauf une seule (encadrée en gris foncé). Cette position est située dans une zone où beaucoup de lumière est reflétée, c'est pourquoi elle n'est pas bonne pour le comptage de lamelles (c'est le même cas que la figure II-1 présentée dans l'annexe II).

Pour la quatrième vidéo, une seule position est mauvaise, pour chacune des deux positions encadrées en orange, deux lamelles ont été ratées. Pour le reste des positions, le résultats du compteur est correct.



FIGURE 4.9 Positions choisies pour les vidéos cinq et six

Trente-six positions parmi quarante-cinq sélectionnées pour la vidéo 5 sont correctes, les autres sont encadrées en gris foncé. Lorsque la bordure de la lamelle (encerclée en rouge) passe par cette partie de la trame, le contraste est très faible et donc la lamelle ne peut être détectée (voir annexe II, figureII-1). Pour les trois positions encadrées en orange, quelques lamelles ont été ratées.

Pour la vidéo six, quelques lamelles ont été ratée pour la majorité des positions sélectionnées (voir les points qui ne sont pas encadrés). Les deux points encerclés en gris foncé ne sont pas bons car les deux bordures des lamelles y passent. Pour les points encadrés en vert, le nombre de lamelles trouvé est correct. Pour le point encadré en rouge, une seule deuxième bordure a été détectée comme lamelle en plus de toutes les autres lamelles principales qui devaient être trouvées.

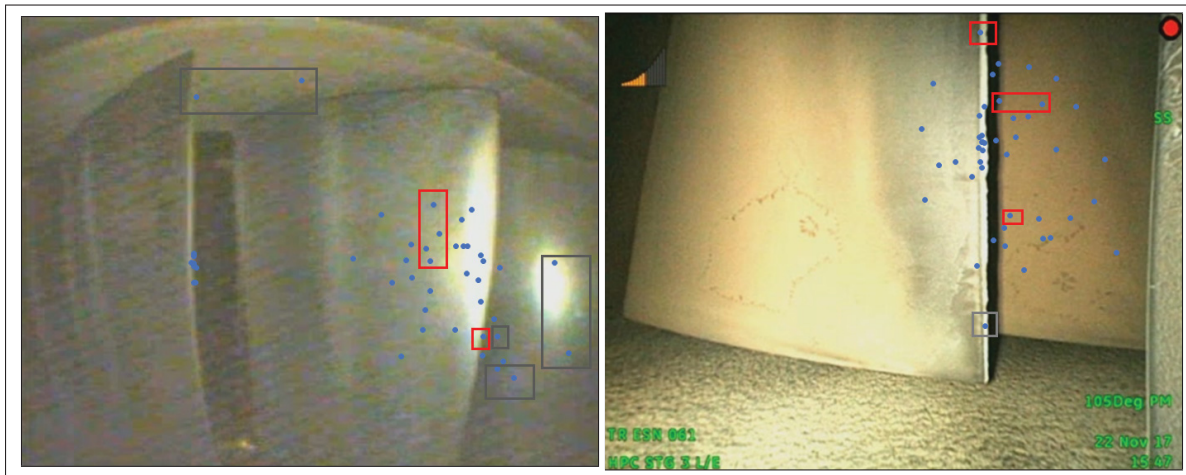


FIGURE 4.10 Positions choisies pour les vidéos sept et huit

Pour la vidéo 7, sept positions ne sont pas bonnes, elles sont mises dans des zones qui peuvent être très lumineuses lors de l'inspection ou bien dans une zone où les bordure latérales passent. Parmi les 38 positions restantes, cinq ont eu du reflet qui a été comptabilisé comme lamelle de plus. Le reste des positions a donné un résultat correct. Une seule position était mauvaise pour la vidéo huit car la bordure latérale y passe. Parmi les 44 positions restantes, 4 ont eu des bruits comptabilisés comme lamelle. Le reste des positions a donné un nombre de lamelles correct.

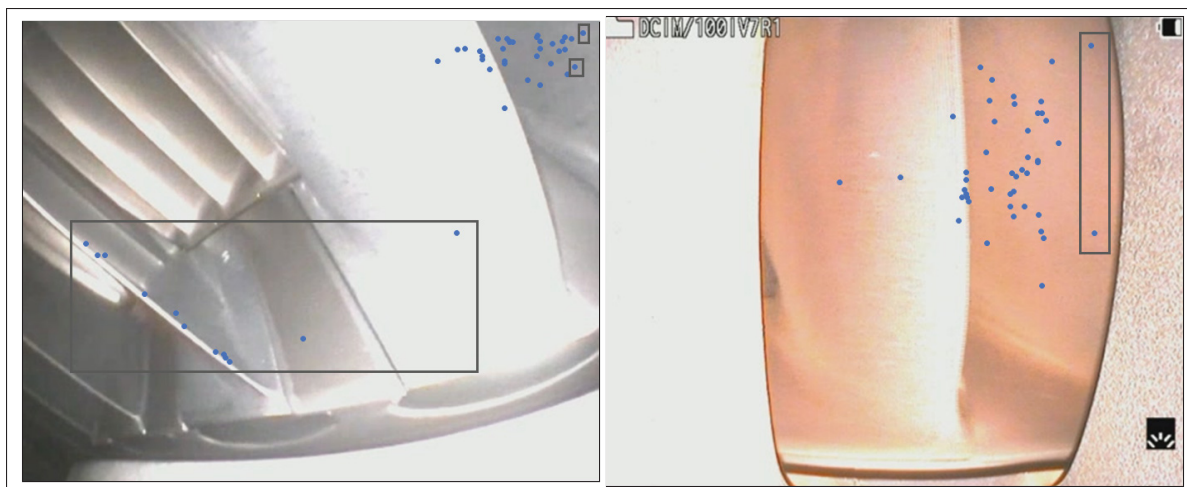


FIGURE 4.11 Positions choisies pour les vidéos neuf et dix

C'est un peu difficile de décider de la bonne zone dans la neuvième vidéo. Les positions ont été sélectionnées autour des deux lamelles qui paraissent principales mais une parmi elles (celle de gauche) n'est pas bonne pour la sélection des position car la deuxième bordure de la lamelle passe par les positions sélectionnées. Parmi les 45 positions, 14 étaient mauvaises. Parmi les 32 qui restent, 3 ont eu un bruit comptabilisé à cause d'une tâche qui passe dans la vidéo.

Pour la vidéo dix, toutes les positions sélectionnées ont donné un résultat correct sauf 2 qui étaient mauvaises.

Le tableau 4.4 résume les résultats obtenus pour toutes les vidéos. Nous y présentons le taux de positions bonnes et mauvaises sélectionnées par les participants. Parmi les bonnes positions, nous calculons le pourcentage de positions ayant donné un nombre de lamelles correct, raté quelques une ou ont eu des lamelles de plus.

TABLEAU 4.4 Résultat de l'étude concernant la sélection d'une bonne position

Vidéo	% bonnes positions	% mauvaises positions	% positions donnant un nombre de lamelles correct	% positions ayant des lamelles ratées	% positions ayant des lamelles de plus
Vidéo 1	80	20	100	0	0
Vidéo 2	100	0	91,1	8,9	0
Vidéo 3	97,8	2,2	100	0	0
Vidéo 4	97,8	2,2	95,5	4,5	0
Vidéo 5	80	20	91,7	8,3	0
Vidéo 6	95,6	4,4	27,9	69,8	2,3
Vidéo 7	84,4	15,6	86,8	0	13,2
Vidéo 8	97,8	2,2	90,9	0	9,1
Vidéo 9	68,9	31,1	90,3	0	9,7
Vidéo 10	95,6	4,4	100	0	0
Moyenne	89,8	10,2	87,4	9,2	3,4

#### 4.6 Test du logiciel à Siemens

Nous avons testé le logiciel avec le moteur ESN111 pour lequel nous avons inspecté 10 étages dont des captures sont montrées dans le tableau 4.6. Le but était de valider les intervalles des métriques de qualification des images et le résultat retourné par le compteur de lamelles.

Pour commencer, nous avons mis en place le matériel nécessaire : ordinateur, module d'acquisition vidéo et boroscope (comme le montre la figure 3.1). Nous avons lancé le logiciel et choisi l'étage à inspecter et rempli ses informations dans l'interface. Avant de passer au compteur, nous voulions valider les intervalles d'acceptance des quatre métriques de qualification des images proposés (déjà implémenté dans le logiciel). Nous les avons testé sur le premier étage (HPC\_stage1\_center) comme ceci :

1. Netteté : dans le boroscope, il y a un paramètre qui modifie la netteté des images avec quatre valeurs. Les valeurs de netteté obtenues pour chaque valeur sont : (0 : 17, 1 : 14, -1 : 13, 2 : 23);


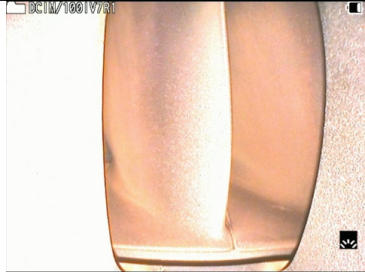






2. Luminosité : nous avons changé le niveau d'intensité de la lumière projetée dans le moteur par le boroscope. Ce dernier en a 6 niveaux. Les valeurs obtenues pour chacun sont : (Niveau 1 : 146, Niveau 2 : 169, Niveau 3 : 182, Niveau 4 : 202, Niveau 5 : 215, Niveau 6 : 223);
3. Contraste : pas de tests effectués ;
4. Vitesse : Nous avons vérifié que la pause donne la valeur zéro. Si l'inspecteur fait tourner l'étage rapidement, la valeur affichée augmente et vice-versa. Ceci dit que le calcul de la vitesse est correct.

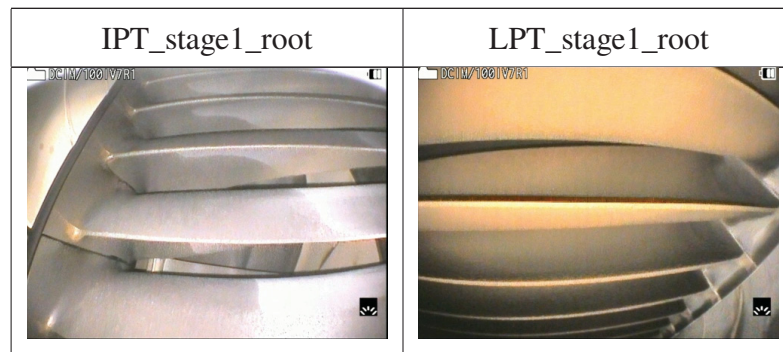
Ensuite, l'inspecteur choisit la position du carré en cliquant sur la vidéo affichée dans l'interface et commence l'enregistrement en cliquant sur le bouton *Enregistrer*. A ce stade, l'inspecteur fait tourner l'étage et compte les lamelles avec son compteur manuel. A la fin de l'étage, nous comparons le nombre de lamelles obtenu par le compteur manuel avec celui obtenu par le logiciel.

En comparant les résultats obtenus par le logiciel avec ceux obtenus par le compteur manuel, nous voyons que les résultats obtenus pour le compteur de lamelles étaient satisfaisants. Le nombre de lamelles obtenu est correct pour tous les stages inspectés. Aussi, le compteur s'affiche et s'incrémente en temps réel exactement au moment du passage de la lamelle. En effet, le taux de traitement des trames est égal au taux d'acquisition (autour de 30 trames par seconde). Lorsque le nombre total des lamelles de l'étage en inspection est atteint, le champ du compteur prend la couleur verte. Ce qui signifie que le tour complet est atteint. Nous avons aussi testé le compteur avec deux positions distinctes pour le stage (HPC\_stage3\_root). Les résultats étaient corrects, ce qui confirme que c'est indépendant de la position choisie (tant qu'elle est bonne).

Notons que toutes les positions choisies par l'inspecteur étaient bonnes. Par contre, parfois, la lamelle fait un allé retour sur le carré car l'inspecteur ne regardait pas l'interface du logiciel. Du coup, il se peut que des lamelles de plus soient comptabilisées à cause de ça, ce qui est correct.

TABLEAU 4.6 Étages inspectés lors de la démonstration à Siemens

<p>HPC_stage1_center</p> 	<p>HPC_stage3_root</p> 
<p>HPT_stage1_center</p> 	<p>IPC_stage1_root</p> 
<p>IPC_stage1_tip</p> 	<p>IPC_stage2_root</p> 
<p>LPT_stage1_tip</p> 	<p>LPT_stage4_tip</p> 



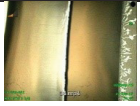
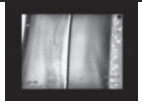

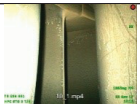

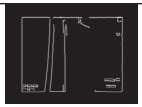

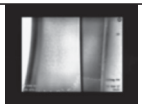
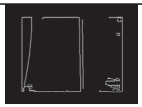
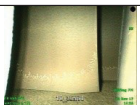
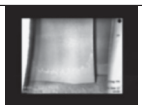
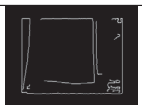

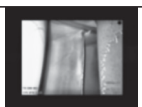

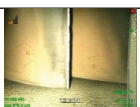








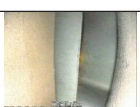


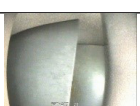

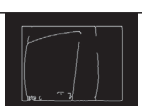






#### 4.7 Tableau récapitulatif et résultats du compteur

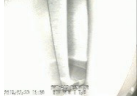



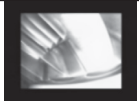

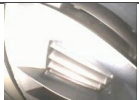



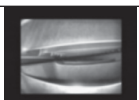
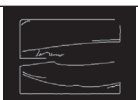






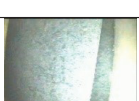


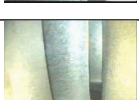

















Ce tableau suivant résume les résultats obtenus pour les métriques de qualification des images ainsi que le compteur. Pour chaque vidéo de la base de données, nous présentons une trame brute ainsi que la même trame filtrée par le filtre bilatéral et Canny. Aussi, la valeur majoritaire des trois métriques de qualification des images pour chaque vidéo. Les dernières colonnes contiennent le nombre de lamelles dans la vidéo et le nombre de lamelles trouvé pour les positions (p1 et p2) avec lesquelles les tests du compteur ont été effectués.


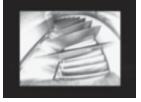



















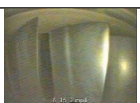








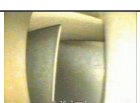





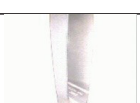


Pour que le résultat du compteur soit lisible, nous soulignons le nombre de lamelles si des lamelles sont ratées, et le mettons en **gras** si plus de lamelles ont été trouvé. Sinon, le nombre trouvé est correct.


























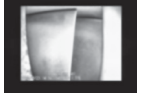













Notons que les vidéos 49, 50 et 57 n'ont pas été considérées pour le compteur de lamelles. La luminosité est très élevée dans les vidéos 49 et 50. De ce fait, les bordures sont confondues avec la surface de la lamelle et ne peuvent être détectées. Quant à la 57, les lamelles qui se déplacent se trouvent derrière celles qui apparaissent dans l'image et d'après l'inspecteur c'est difficile d'y accéder avec la sonde du boroscope.






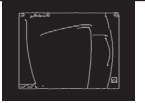
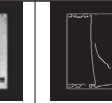

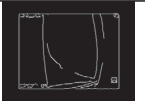



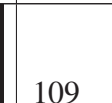



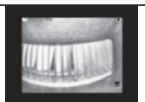

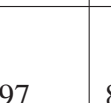


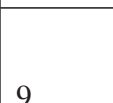

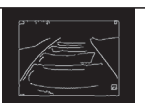
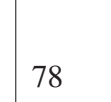





TABLEAU 4.7 Tableau récapitulatif des tests des métriques de qualification des images ainsi que le compteur de lamelles

N	Trame brute	Bilatéral	Canny	contra- ste	lumi- nosité	netteté	lame- lles	p 1	p 2
1				166	150	11	94	95	<b>107</b>
2				200	160	9	83	83	82
3				190	150	10	83	87	<u>82</u>
4				188	150	12	84	83	86
5				180	150	10	94	<b>107</b>	94
6				166	153	12	95	95	94
7				140	175	14	62	62	63
8				100	215	10	60	59	59
9				130	180	14	59	61	62
10				145	170	11	63	63	64
11				70	230	10	59	59	58
12				63	239	7	48	47	48

N	Trame brute	Bilatéral	Canny	contra- ste	lumi- nosité	netteté	lame- lles	p 1	p 2
13				66	240	7	58	58	58
14				140	190	6	59	58	58
15				150	185	5	61	62	61
16				114	115	6	70	70	71
17				125	130	7	87	86	86
18				130	195	8	61	63	64
19				140	190	6	63	<u>61</u>	<u>60</u>
20				120	190	7	57	57	57
21				150	170	9	55	56	<u>53</u>
22				150	170	9	5	5	<u>4</u>
23				140	180	11	55	55	55
24				145	180	12	54	56	55
25				140	180	10	57	60	56

N	Trame brute	Bilatéral	Canny	contra- ste	lumi- nosité	netteté	lame- lles	p 1	p 2
26				120	210	17	60	<u>48</u>	<u>25</u>
27				106	218	10	60	60	59
28				100	215	12	56	58	57
29				200	144	5	44	45	43
30				160	180	5	69	<u>68</u>	70
31				120	200	5	73	72	<u>72</u>
32				70	117	5	68	67	67
33				60	116	5	68	67	<u>58</u>
34				150	180	5	63	63	63
35				150	170	5	67	66	66
36				150	180	6	66	65	66
37				60	240	10	59	59	59
38				50	240	5	61	60	60

N	Trame brute	Bilatéral	Canny	contra- ste	lumi- nosité	netteté	lame- lles	p 1	p 2
39				49	240	5	60	58	<u>41</u>
40				154	170	11	65	66	66
41				111	205	11	64	65	64
42				150	175	10	65	65	64
43				155	170	10	63	69	64
44				150	150	7	54	53	55
45				100	218	8	64	<b>66</b>	65
46				120	200	8	65	66	65
47				126	190	8	68	71	68
48				68	236	8	60	60	61
49				30	240	6	60	/	/
50				60	230	7	58	/	/
51				105	205	13	14	17	13

N	Trame brute	Bilatéral	Canny	contra- ste	lumi- nosité	netteté	lame- lles	p 1	p 2
52				104	204	13	35	35	35
53				121	195	10	14	14	14
54				121	194	9	64	64	64
55				127	190	10	15	15	15
56				109	199	10	34	<u>31</u>	34
57				113	193	14	/	/	/
58				132	197	8	32	32	<u>30</u>
59				111	160	9	97	99	98
60				105	205	13	78	79	<b>87</b>
61				108	202	10	164	<u>155</u>	<u>160</u>

## 4.8 Analyse des résultats et discussion

### 4.8.1 Métriques de qualification des images

Lors de la démonstration du logiciel à Siemens, l'inspecteur a vérifié les intervalles proposées et a demandé de changer quelques-uns. Les nouveaux intervalles sont les suivants :

- Netteté : [0, 6] [**7 et plus**];
- Contraste : [0, 80] [80, 90] [**90, 255**]
- Luminosité : [0, 75][75, 100][**100, 180**]/[180, 210]/[210, 255]
- Vitesse : [**0, 30]** [30, 40] [40 et plus] (px/s)

### 4.8.2 Compteur de lamelles

Les six premières vidéos présentées dans le tableau 4.7 ont été enregistrées avec le boroscope GE (qui ne va plus être utilisé). Elles se caractérisent par des reflets accentués qui ne sont pas toujours bien filtrés par les paramètres fixés. Les vidéos 18, 19, 20 et 32 à 36 ont été enregistrées par le boroscope iPlex RX mais la résolution est de  $320 \times 240$  : la moitié de la résolution du reste des vidéos qui ont été également enregistrées par le même boroscope. Ceci peut être dû à une configuration effectuée par l'inspecteur lors de ces inspections. La qualité de ces vidéos est modérée par rapport à celles qui ont une résolution de  $640 \times 480$ .

Malgré ces différences entre les vidéos de la base de données, nous avons réussi à bien filtrer toutes les trames en ne gardant que les bordures des lamelles principales ce qui permet au logiciel de traiter des vidéos variantes en termes de qualité et résolution.

Pour les vidéos capturées par le boroscope GE, nous avons eu des bruits comptabilisés comme lamelle, ce qui est inévitable et imprévisible. Et ce pour les deux positions seulement parmi 12.

En ce qui concerne les vidéos prises par le boroscope Olympus iPlex RX et avec une mauvaise résolution, quelques lamelles ont été ratées dans 3 positions parmi 16.

Pour les vidéos capturées par le Olympus iPlex RX avec une bonne résolution, nous avons 88 positions au total. Parmi elles, 9 ont eu des lamelles ratées et 2 ont eu des lamelles détectées de plus à cause du bruit. Pour le reste, le nombre de lamelles trouvées est correct.

Notons que pour le nombre de lamelles contenues dans la vidéo a été compté à l'œil. Donc nous pouvons considérer un nombre retourné supérieur au nombre de lamelles de la vidéo correct si un aller-retour se produit exactement sur le carré car chaque passage par le carré compte. Aussi, nous pouvons avoir une lamelle qui n'arrive pas au carré (à la fin par exemple) et le nombre trouvé est moindre. Ceci est également considéré correct. La position par laquelle la lamelle passe pour qu'elle soit comptée est montrée dans l'interface, ce qui règle ce problème lors des prochaines inspections avec le logiciel.

Le tableau 4.8 résume les résultats des tests effectués pour le compteur de lamelles.

TABLEAU 4.8 Résultats des tests effectués pour le compteur de lamelles

	nb positions	nb lamelles correct	plus de lamelles	lamelles ratées
iPlex RX (320 × 240)	16	13	0	3
iPlex RX (640 × 480)	88	78	2	8
GE	12	9	2	1
Total	116	100	4	12
Total (%)		86,2	3,4	10,3

D'après ce tableau : 86,2% des positions ont donné un nombre de lamelles correct, 3,4% ont eu des bruits ou des tâches considérés comme lamelle et 10,3% ont eu des lamelles ratées.

Pour les positions ayant des lamelles de plus, les lamelles ont toutes été détectées en plus des bruits. Les lamelles sont surtout ratées dans les vidéos où l'inspecteur fait tourner les lamelles très rapidement.

Notons que nous avons sélectionné deux positions pour chaque vidéo pour prouver que le compteur ne dépend pas de la position choisie, tant qu'elle est bonne.

Pour les inspections à Siemens, ce taux de succès du logiciel est suffisant. D'une part, le GE ne sera plus utilisé, ce qui réduit le taux des bruits détectés. D'autre part, la cause principale des lamelles ratées est la position du carré loin de la lamelle principale surtout dans les vidéos qui ressemblent à la vidéo 1 montrée dans la figure 4.7. Si la position est bien choisie, le logiciel compte les lamelles qui y passent correctement.

### **4.8.3 Procédure de sélection d'une bonne position**

D'après le tableau 4.4, 89,8% des positions choisies étaient bonnes. Parmi elles, 87,4% ont donné un nombre de lamelles correct pour les vidéo correspondantes. Pour la vidéo 6, 69,8% des bonnes positions ont eu des lamelles ratées. Les positions appartenaient à une bonne zone mais l'inspecteur, dans cette vidéo, fait tourner les lamelles très rapidement et le sens est de gauche à droite. Ce qui fait que les lamelles qui passent par ces positions sont très rapprochées et très rapides, ce qui rend la détection de la différence d'intensité difficile. Si les lamelles se déplaçaient dans le sens contraire, ça aurait marché comme c'est le cas pour la vidéo 1. Les bruits et les tâches ne peuvent être prévisibles lors de la mise en place de la position par l'utilisateur. Mais généralement, ils sont filtrés et ne sont pas détectés comme lamelles.

D'après ces résultats, les participants tendent à choisir des positions qui appartiennent à une bonne zone. Et, la majorité des positions choisies retourne un nombre de lamelles correct pour les vidéos entières. Ce qui montre que la répétabilité inter-participant est vérifiée. En ce qui concerne la répétabilité intra-participant, nous l'avons évalué en calculant, pour chaque vidéo, l'écart type des trois positions choisies par chaque participant pour la coordonnée horizontale et verticale. A la fin, nous calculons la moyenne des 15 écart-types obtenus pour chaque vidéo (voir le tableau 4.9).

TABLEAU 4.9 Ecart-type entre les positions choisies par les participants pour chaque vidéo

Vidéo	Moyenne écart-type en x	Moyenne écart-type en y	remarque
Vidéo 1	34,026	8,04	lamelle horizontale
Vidéo 2	11,97	26,68	lamelle verticale
Vidéo 3	23,77	24,93	lamelle verticale
Vidéo 4	39,94	3,35	lamelle horizontale
Vidéo 5	26,47	26,18	lamelle verticale
Vidéo 6	36,92	23,66	lamelle inclinée
Vidéo 7	16,09	25,36	lamelle verticale
Vidéo 8	18,43	23,94	lamelle verticale
Vidéo 9	17,02	7,72	lamelle verticale
Vidéo 10	16,14	32,26	lamelle verticale

D'après le tableau, les écart-types trouvés entre les positions choisies par les participants sont relativement raisonnables. Notons que lorsque la lamelle est horizontale (resp. verticale), l'écart type entre les X (resp. Y) des positions choisies peut être élevé. Du coup, nous pouvons dire que le même participant tend à choisir les mêmes positions pour les mêmes images visualisées. Ce qui valide la répétabilité intra-participant.

Ceci confirme la fiabilité de la procédure proposée même si les participants ne sont pas des inspecteurs. En effet, nous croyons que si l'étude a été faite avec des inspecteurs, les résultats auraient été mieux car ils connaissent exactement de quoi il s'agit.

#### 4.9 Conclusion

Dans ce chapitre, nous avons présenté les tests effectués pour valider les méthodes de calcul des métriques de qualification des images, le compteur de lamelles et la procédure de sélection de la position utilisée pour le calcul d'intensité et la détection du passage des lamelles. Nous avons aussi analysé et discuté les résultats obtenus. Le chapitre suivant, présente les perspectives de ce projet et conclue ce manuscrit.

## CONCLUSION ET PERSPECTIVES

Dans ce mémoire, nous avons présenté une solution qui libère les inspecteurs de quelques tâches routinières. A Siemens, les inspecteurs de turbines à gaz doivent utiliser un boroscope pour vérifier l'état interne des moteurs. Après avoir injecté la sonde et la caméra pour visualiser la partie du moteur à inspecter, ils doivent effectuer plusieurs tâches simultanément : vérifier la qualité des images acquises, compter les lamelles en utilisant un compteur manuel et détecter les défauts tout au long de l'inspection. Les libérer de la qualification des images et du comptage des lamelles les laissera se concentrer sur la tâche qui demande plus de concentration et de prise de décision humaine : la détection des défauts.

Notre principal objectif consistait à proposer un logiciel qui prend en charge ces deux tâches tout au long de l'inspection et en temps réel. Un logiciel existait mais il était inopérant. Pour commencer, nous avons analysé ce logiciel et vu de près les méthodes de calcul proposées pour cerner le contexte et la problématique du projet. Ensuite, nous avons étudié l'état de l'art des différentes méthodes pouvant être utilisées pour résoudre la problématique : élimination du bruit, détection de contours et estimation du mouvement. En se basant sur cette étude, nous avons développé une nouvelle approche permettant de qualifier les images et de compter les lamelles en temps réel.

Beaucoup de traitements doivent être appliqués sur toutes les trames acquises pour atteindre les objectifs du projet. En effet, nous devons faire des calculs pour qualifier les images et aussi pour détecter les contours et compter les lamelles. Pour arriver à faire ça en temps réel, nous avons accéléré le processus de traitement en intégrant CUDA. Celle-ci permet d'utiliser un processeur graphique (GPU) pour exécuter les calculs au lieu du processeur central (CPU).

Dans un premier temps, nous avons rectifié les méthodes de calcul des trois métriques de qualification des images existantes : netteté, luminosité et vitesse. Aussi, nous avons rajouté une autre qui est le contraste. Dans un deuxième temps, nous avons proposé une nouvelle approche pour compter les lamelles. Pour compter les lamelles, il faut éliminer le bruit tout en gardant les bordures des lamelles bien claires pour les détecter dans l'étape suivante avec les filtres

appropriés. Avec les bordures détectées, nous comptons les lamelles qui passent par une position fixée en détectant la différence d'intensité importante qui représente le passage de la lamelle. Mais, nous avons besoin de l'intervention humaine pour choisir la position fixée. Pour cela, nous avons établi une procédure que nous proposons aux utilisateurs pour choisir une bonne position et avoir un résultat correct.

Dans le but de valider l'approche du comptage de lamelles et les méthodes de calcul des métriques de qualification des images et proposer un intervalle d'acceptance valide pour chacune, nous avons effectué les tests localement en utilisant un ensemble de vidéos fourni par Siemens. Aussi, nous avons testé le logiciel à Siemens directement en inspectant plusieurs étages d'un moteur qui était disponible. Aussi, nous avons validé la procédure de sélection de la position fixée pour le calcul d'intensité et le comptage de lamelles. Ceci a été réalisé en menant une étude sur un ensemble de participants.

### **Perspectives**

Les résultats auxquels nous sommes parvenus suscitent des réflexions et permettent d'envisager des poursuites de travaux de recherche. Concernant la qualification des images, les intervalles proposés sont fixés pour toutes les vidéos. Les moteurs sont différents et les conditions peuvent changer d'une partie du moteur à une autre. Par exemple, pour la vitesse, la lamelle peut être plus ou moins proche de la caméra, les intervalles ne doivent pas être identiques dans les deux cas. Les parties des moteurs peuvent être classifiées et pour chaque catégorie, des intervalles d'acceptance peuvent être proposés.

En ce qui concerne le compteur de lamelles, la méthode proposée est optimale car les valeurs des pixels des images sont 0 ou 255. Du coup, le passage de la lamelle est assurément détecté et ne peut être raté. L'amélioration que nous proposons concernant le compteur est de trouver le carré fixe automatiquement sans l'intervention humaine. En se basant sur le mouvement des lamelles. Aussi, le sens du mouvement peut être calculé dès les premières trames et déduit. Avec ça, nous pouvons ignorer les retours arrière qui se produisent souvent quand l'inspecteur veut

revoir une lamelles qui vient de passer. Dans ce cas, on peut ignorer les lamelles qui se déplacent dans le sens contraire et reprendre dès que le sens revient à la normale.

Une fonctionnalité pertinente qui peut rajouter plus de robustesse à l'algorithme est la détection automatique des défauts. En intégrant un module de reconnaissance de défauts qui montre la zone suspecte de contenir un défaut à l'utilisateur et apprend des décisions prises par l'inspecteur. Avec le temps, le logiciel sera en mesure de reconnaître les défauts et de les classifier avec précision.



# ANNEXE I

## DIAGRAMME D'ACTIVITÉ DÉCRIVANT LA SOLUTION EXISTANTE

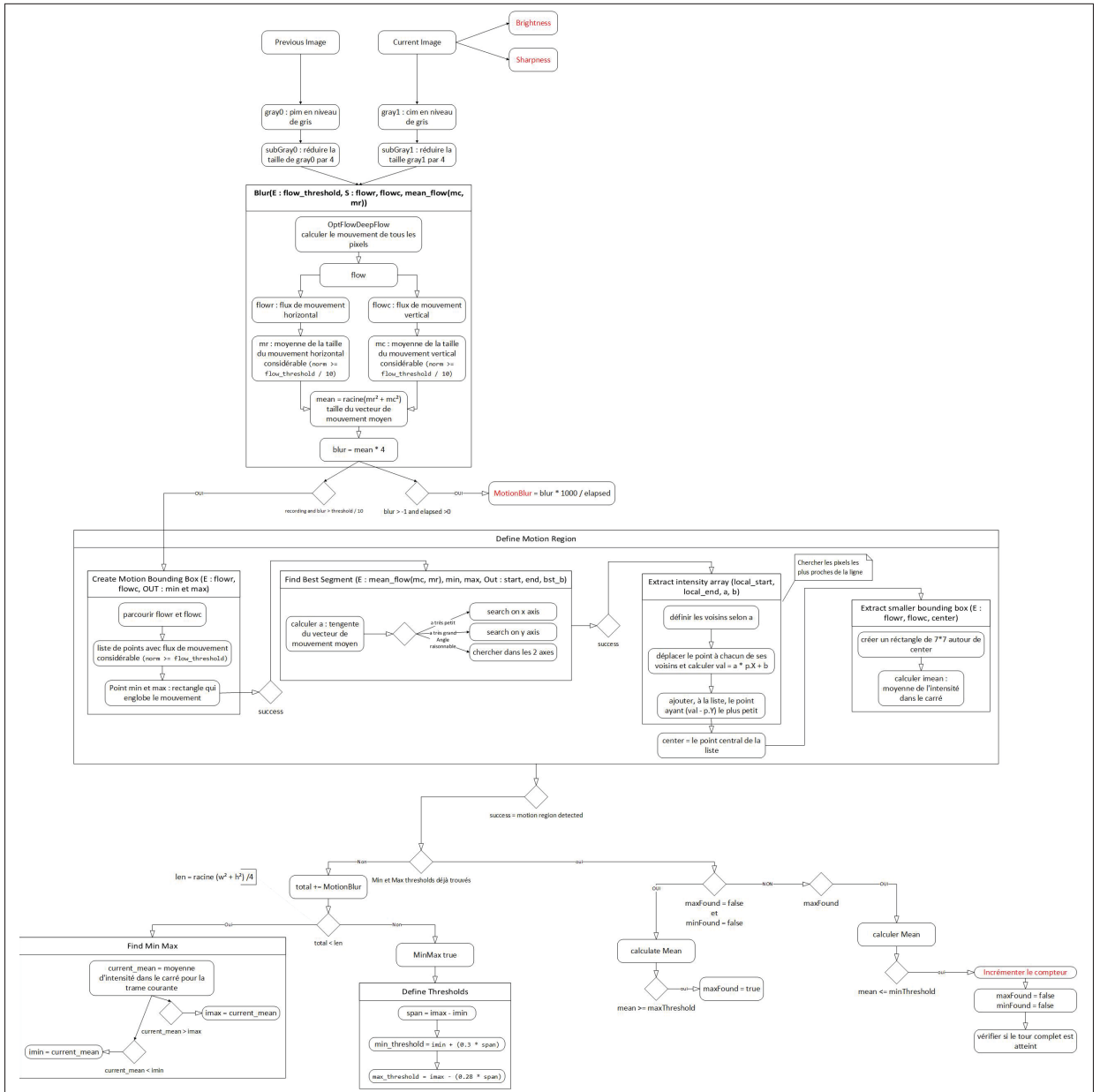


Figure-A I-1 Diagramme d'activité de la solution existante



## ANNEXE II

### LUMIÈRE FORTE DANS LA FIGURE

Cette figure représente l'impact de l'injection d'une lumière forte à l'intérieur du moteur. Quelques portions des bordures peuvent ne pas être détectées au niveau de certaines trames.

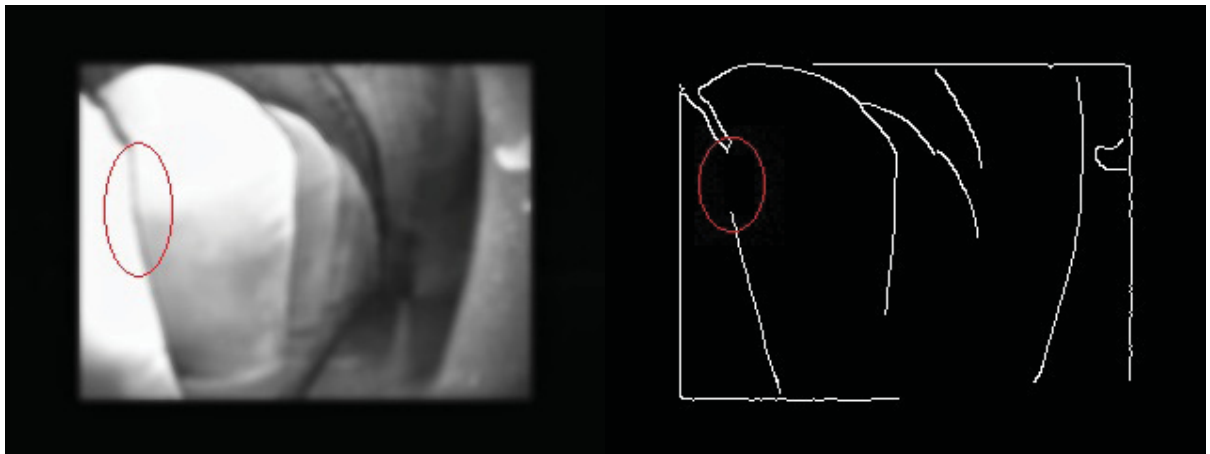


Figure-A II-1 Lumière forte qui engendre la non détection du contour



### **ANNEXE III**

#### **FORMULAIRE D'INFORMATION ET DE CONSENTEMENT**

Ceci représente le formulaire signé par les participants avant leur participation à notre étude.

## FORMULAIRE D'INFORMATION ET DE CONSENTEMENT

### TITRE DU PROJET DE RECHERCHE

---

Développement et validation d'une méthode de comptage de lamelles d'une turbine à gaz

### CHERCHEUR RESPONSABLE

---

Carlos Vazquez, professeur au département de LOG-TI – École de technologie supérieure (ÉTS)

### CO-CHERCHEUR

---

David Labbé, professeur au département de LOG-Ti, École de technologie supérieure (ÉTS)

### ÉTUDIANTE

---

Imene Belkacemi, Étudiante à la maîtrise au département de LOG-TI – ÉTS

### FINANCEMENT

---

Le chercheur responsable a reçu du financement de MITACS en partenariat avec Siemens pour mener à bien ce projet de recherche.

### INTRODUCTION

---

Nous vous invitons à participer à un projet de recherche. Cependant, avant d'accepter de participer à ce projet et de signer ce formulaire d'information et de consentement, veuillez prendre le temps de lire, de comprendre et de considérer attentivement les renseignements qui suivent.

Ce formulaire peut contenir des mots que vous ne comprenez pas. Nous vous invitons à poser toutes les questions que vous jugerez utiles au chercheur responsable de ce projet ou à un membre de l'équipe de recherche, et à leur demander de vous expliquer tout mot ou renseignement qui n'est pas clair.

### NATURE ET OBJECTIFS DU PROJET DE RECHERCHE

---

Les turbines à gaz contiennent des lamelles qui peuvent se détériorer avec le temps. Pour vérifier leur état et effectuer les réparations nécessaires, une inspection endoscopique doit avoir lieu. Cette dernière se fait en analysant les images obtenues par la caméra initialement injectée à l'intérieur du moteur par des inspecteurs experts. Chaque partie du moteur est composée d'un certain nombre de lamelles fixe et connu. Pour s'assurer que toutes les lamelles ont été vérifiées, l'inspecteur doit compter les lamelles jusqu'à ce qu'il arrive au nombre

total de la partie inspectée. Ceci est fait en utilisant un compteur manuel. Pour automatiser ce processus, une position adéquate doit être sélectionnée par l'utilisateur. Le passage de la lamelle par cette position est comptabilisé par le logiciel. Par contre, l'intervention humaine est nécessaire car beaucoup de conditions doivent être vérifiées afin de sélectionner la bonne position dépendamment de la partie du moteur inspectée. Cette sélection va être effectuée par plusieurs inspecteurs. Pour cela, une procédure qui comporte toutes les instructions à suivre pour arriver à la bonne position a été établie. Ainsi, ce projet de recherche vise à confirmer que l'application des instructions décrites mène toujours à une position qui donne un nombre correct de lamelles. Pour mener à bien ce projet, nous recruterons 15 participants, hommes et femmes, âgés de 18 ans et plus.

## DÉROULEMENT DU PROJET DE RECHERCHE

---

### 1. Lieu de réalisation du projet de recherche et durée de la participation

Ce projet de recherche se déroulera en ligne. Votre participation à ce projet durera 30 minutes et nécessitera une seule séance.

### 2. Nature de votre participation

#### 1. Phase de compréhension des instructions (15 minutes)

Nous vous demanderons dans un premier temps de lire et comprendre les instructions de sélection de la bonne position dans les images.

#### 2. Phase de sélection de la bonne position (15 minutes)

Dans cette étape, nous vous présenterons un énoncé de dix parties dans une page Web. Chaque partie contient un extrait de vidéo et trois images (voir la figure suivante) sur lesquelles nous vous demanderons de choisir une bonne position pour le comptage de lamelles. Après avoir regardé la vidéo, vous devrez identifier la lamelle principale, voir s'il y a des zones à éviter à cause des bordures secondaires (supérieures, inférieures, apparition de la deuxième bordure de la même lamelle). La sélection de la bonne position se fait en cliquant directement sur l'image.



Figure – exemples d'images capturées lors d'une inspection dans deux turbines à gaz différentes

## **AVANTAGES ET BÉNÉFICES ASSOCIÉS AU PROJET DE RECHERCHE**

---

Il se peut que vous retiriez un bénéfice personnel de votre participation à ce projet de recherche, mais nous ne pouvons vous l'assurer. Par ailleurs, les résultats obtenus contribueront à l'avancement des connaissances scientifiques dans ce domaine de recherche.

## **INCONVÉNIENTS ASSOCIÉS AU PROJET DE RECHERCHE**

---

Outre le temps consacré à la participation, votre participation à ce projet ne comporte aucun inconvénient.

## **PARTICIPATION VOLONTAIRE ET DROIT DE RETRAIT**

---

Votre participation à ce projet de recherche est volontaire. Vous êtes donc libre de refuser d'y participer. Vous pouvez également vous retirer de ce projet à n'importe quel moment, sans avoir à donner de raisons, en informant l'équipe de recherche.

Votre décision de ne pas participer à ce projet de recherche ou de vous en retirer n'aura aucune conséquence sur votre cheminement académique.

Le chercheur responsable de ce projet de recherche, le Comité d'éthique de la recherche de l'École de technologie supérieure ou l'organisme subventionnaire et son partenaire financier peuvent mettre fin à votre participation, sans votre consentement. Cela peut se produire si de nouvelles découvertes ou informations indiquent que votre participation au projet n'est plus dans votre intérêt, si vous ne respectez pas les consignes du projet de recherche ou encore s'il existe des raisons administratives d'abandonner le projet.

Si vous vous retirez du projet ou êtes retiré(e) du projet, l'information et le matériel déjà recueillis dans le cadre de ce projet seront néanmoins conservés, analysés ou utilisés pour assurer l'intégrité du projet.

Toute nouvelle connaissance acquise durant le déroulement du projet qui pourrait avoir un impact sur votre décision de continuer à participer à ce projet vous sera communiquée rapidement.

## **CONFIDENTIALITÉ**

---

Durant votre participation à ce projet de recherche, le chercheur responsable ainsi que les membres de l'équipe de recherche recueilleront, dans un dossier de recherche, les renseignements vous concernant et nécessaires pour répondre aux objectifs scientifiques de ce projet de recherche.

Ces renseignements peuvent comprendre votre nom, votre âge, votre adresse courriel ainsi que les résultats de tous les tests et procédures réalisés dans le cadre du projet.

Tous les renseignements recueillis demeureront confidentiels, dans les limites prévues par la loi. Afin de préserver votre identité et la confidentialité de vos renseignements, un numéro de code vous sera attribué. La clé du code reliant votre nom à votre dossier de recherche sera conservée par le chercheur responsable de ce projet de recherche.

Ces données de recherche seront conservées pendant au moins 10 ans par le chercheur responsable de ce projet de recherche.

Les données de recherche pourront être publiées ou faire l'objet de discussions scientifiques, mais il ne sera pas possible de vous identifier.

À des fins de surveillance, de contrôle, de protection, de sécurité, votre dossier de recherche pourra être consulté par une personne mandatée par des organismes réglementaires ainsi que par des représentants de l'organisme subventionnaire et de son partenaire financier, de l'École de technologie supérieure ou du Comité d'éthique de la recherche. Ces personnes et ces organismes adhèrent à une politique de confidentialité.

Vous avez le droit de consulter votre dossier de recherche pour vérifier les renseignements recueillis et les faire rectifier au besoin.

## **COMPENSATION**

---

Vous ne recevrez pas de compensation financière pour votre participation à ce projet de recherche.

## **EN CAS DE PRÉJUDICE**

---

En acceptant de participer à ce projet de recherche, vous ne renoncez à aucun de vos droits, ni ne libérez le chercheur responsable, l'École de technologie supérieure et l'organisme subventionnaire et son partenaire financier de leur responsabilité civile et professionnelle.

## **SUIVI ÉTHIQUE**

---

Le Comité d'éthique de la recherche de l'École de technologie supérieure a approuvé ce projet de recherche et en assure le suivi.

## **PERSONNES-RESSOURCES**

---

Pour toute question en lien avec le projet de recherche, vous pouvez contacter le chercheur responsable au [carlos.vazquez@etsmtl.ca](mailto:carlos.vazquez@etsmtl.ca). Vous pouvez également contacter [imene.belkacemi.1@ens.etsmtl.ca](mailto:imene.belkacemi.1@ens.etsmtl.ca).

Pour toute question en lien avec vos droits en tant que participant à la recherche, vous pouvez contacter le Comité d'éthique de la recherche de l'École de technologie supérieure par courriel à l'adresse [cer@etsmtl.ca](mailto:cer@etsmtl.ca) ou en téléphonant au (514) 396-8800 poste 7807.

## CONSENTEMENT

---

### **Engagement du chercheur responsable**

Je certifie qu'on a expliqué au(à la) participant(e) le présent formulaire d'information et de consentement et que l'on a répondu aux questions qu'il(elle) avait.

Je m'engage, avec l'équipe de recherche, à respecter ce qui a été convenu au formulaire d'information.

### **Participant(e)**

- J'ai lu le présent formulaire de consentement et j'ai disposé de suffisamment de renseignements et du temps nécessaire pour prendre ma décision.
- Après réflexion, je consens volontairement à participer à ce projet de recherche, aux conditions énoncées.
- Après réflexion, je refuse de participer au projet de recherche (*vous serez redirigé(e) vers la page de sortie du questionnaire*).

## BIBLIOGRAPHIE

- Afshari, H., Gadsden, S. & Habibi, S. (2017). Gaussian filters for parameter and state estimation : A general review of theory and recent trends. *Signal Processing*, 135, 218-238. doi : <https://doi.org/10.1016/j.sigpro.2017.01.001>.
- Al-Tairi, Z. H., Rahmat, R. W., Saripan, M. I. & Sulaiman, P. S. (2014). Skin segmentation using YUV and RGB color spaces. *Journal of information processing systems*, 10(2), 283–299.
- Amer, G. M. H. & Abushaala, A. M. (2015). Edge detection methods. *2015 2nd World Symposium on Web Applications and Networking (WSWAN)*, pp. 1–7.
- Benesty, J., Chen, J. & Huang, Y. (2010). Study of the widely linear Wiener filter for noise reduction. *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 205–208.
- Bruhn, A., Weickert, J. & Schnörr, C. (2005). Lucas/Kanade meets Horn/Schunck : Combining local and global optic flow methods. *International journal of computer vision*, 61(3), 211–231.
- Buades, A., Coll, B. & Morel, J.-M. (2005). A review of image denoising algorithms, with a new one. *Multiscale Modeling & Simulation*, 4(2), 490–530.
- Campbell, M., Egerstedt, M., How, J. P. & Murray, R. M. (2010). Autonomous driving in urban environments : approaches, lessons and challenges. *Philosophical Transactions of the Royal Society A : Mathematical, Physical and Engineering Sciences*, 368(1928), 4649–4672.
- Canny, J. (1986). A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, (6), 679–698.
- Chaudhury, K. N. (2012). Acceleration of the Shiftable  $O(1)$  Algorithm for Bilateral Filtering and Nonlocal Means. *IEEE Transactions on Image Processing*, 22(4), 1291–1300.
- Chaudhury, K. N. & Rithwik, K. (2015). Image denoising using optimally weighted bilateral filters : A sure and fast approach. *2015 IEEE International Conference on Image Processing (ICIP)*, pp. 108–112.
- Chaudhury, K. N., Sage, D. & Unser, M. (2011). Fast  $O(1)$  bilateral filtering using trigonometric range kernels. *IEEE transactions on image processing*, 20(12), 3376–3382.
- Chen, R. (2020, 09, 29). La prévoyance, un atout essentiel : les vidéoscopes contribuent à améliorer la qualité et l'efficacité de l'inspection des turbines à vapeur. Repéré à <https://urlz.fr/gjD7>.

- Dabov, K., Foi, A., Katkovnik, V. & Egiazarian, K. (2007). Image Denoising by Sparse 3-D Transform-Domain Collaborative Filtering. *IEEE Transactions on Image Processing*, 16(8), 2080-2095. doi : 10.1109/TIP.2007.901238.
- Dabov, K., Foi, A., Katkovnik, V. & Egiazarian, K. (2009). BM3D image denoising with shape-adaptive principal component analysis. *SPARS'09-Signal Processing with Adaptive Sparse Structured Representations*.
- Dogra, A. (2017). Performance comparison of different wavelet families based on bone vessel fusion. *Asian Journal of Pharmaceutics (AJP) : Free full text articles from Asian J Pharm*, 10(04).
- Donoho, D. L. (1995). De-noising by soft-thresholding. *IEEE transactions on information theory*, 41(3), 613–627.
- Donoho, D. L. & Johnstone, J. M. (1994). Ideal spatial adaptation by wavelet shrinkage. *biometrika*, 81(3), 425–455.
- Eliaz, N., Shemesh, G. & Latanision, R. (2002). Hot corrosion in gas turbine components. *Engineering Failure Analysis*, 9(1), 31-43. doi : [https://doi.org/10.1016/S1350-6307\(00\)00035-2](https://doi.org/10.1016/S1350-6307(00)00035-2).
- Epiphan. (2021). epiphan video. Repéré à <https://www.epiphan.com/products/avio-hd/>.
- Fan, L., Zhang, F., Fan, H. & Zhang, C. (2019). Brief review of image denoising techniques. *Visual Computing for Industry, Biomedicine, and Art*, 2(1), 1–12.
- Farneback, G. (2003). Two-frame motion estimation based on polynomial expansion. *Scandinavian conference on Image analysis*, pp. 363–370.
- GEPower. (2021). WHAT IS A GAS TURBINE? Repéré à <https://www.ge.com/power/resources/knowledge-base/what-is-a-gas-turbine>.
- Goyal, B., Dogra, A., Agrawal, S., Sohi, B. S. & Sharma, A. (2020). Image denoising review : From classical to state-of-the-art approaches. *Information fusion*, 55, 220–244.
- Gupta, S. et al. (2014). A review and comprehensive comparison of image denoising techniques. *2014 International Conference on Computing for Sustainable Global Development (INDIACom)*, pp. 972–976.
- Haering, N., Venetianer, P. L. & Lipton, A. (2008). The evolution of video surveillance : an overview. *Machine Vision and Applications*, 19(5), 279–290.

- Hardie, R. C. & Barner, K. E. (1994). Rank conditioned rank selection filters for signal restoration. *IEEE Transactions on Image Processing*, 3(2), 192–206.
- Hou, J. (2007). *Research on image denoising approach based on wavelet and its statistical characteristics*. (Thèse de doctorat, Dissertation, Huazhong University of Science and Technology).
- Husseini, S. (2017). *A survey of optical flow techniques for object tracking*. (B.S. thesis).
- ISO. (1996). Turbines à gaz — Vocabulaire. Repéré à <https://www.iso.org/obp/ui/#iso:std:iso:11086:ed-1:v1:fr:sec:A>.
- LeParisien. (2016). Boroscope. Repéré à <http://dictionnaire.sensagent.leparisien.fr/Boroscope/fr-fr/>.
- Marr, D. & Hildreth, E. (1980). Theory of edge detection. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, 207(1167), 187–217.
- Matlab. (2021a). imadjust, Adjust image intensity values or color map. Repéré à <https://www.mathworks.com/help/images/ref/imadjust.html>.
- Matlab. (2021b). imAdjust. Repéré à <https://www.mathworks.com/help/images/ref/imadjust.html>.
- Matlab. (2021c). imgaussfilt, 2-D Gaussian filtering of images. Repéré à <https://www.mathworks.com/help/images/ref/imgaussfilt.html>.
- Nagel, H.-H. (1987). On the estimation of optical flow : Relations between different approaches and some new results. *Artificial intelligence*, 33(3), 299–324.
- Nason, G. P. & Silverman, B. W. (1994). The discrete wavelet transform in S. *Journal of Computational and Graphical statistics*, 3(2), 163–191.
- Ng, P.-E. & Ma, K.-K. (2006). A switching median filter with boundary discriminative noise detection for extremely corrupted images. *IEEE Transactions on image processing*, 15(6), 1506–1516.
- Nicolas, H. (2016). *Boroscope inspection helper study* (Rapport n°REP-SIE001 - 2).
- Olympus. (2021a). Internal Inspection of Various Hoses of Automobiles Using Industrial Videoscopes. Repéré à <https://www.olympus-ims.com/fr/applications/rvi-automobile-hoses-inspection/>.

- Olympus. (2021b). IPLEX GX/GT Caractéristiques techniques. Repéré à [https://www.olympus-ims.com/fr/rvi-products/iplex-gx/#!cms\[focus\]=cmsContent14416](https://www.olympus-ims.com/fr/rvi-products/iplex-gx/#!cms[focus]=cmsContent14416).
- OpenCV. (2021a). Optical Flow. Repéré à [https://docs.opencv.org/master/d4/dee/tutorial\\_optical\\_flow.html](https://docs.opencv.org/master/d4/dee/tutorial_optical_flow.html).
- OpenCV. (2021b). Base interface for dense optical flow algorithms. Repéré à [https://docs.opencv.org/master/d6/d4a/classcv\\_1\\_1cuda\\_1\\_1DenseOpticalFlow.html](https://docs.opencv.org/master/d6/d4a/classcv_1_1cuda_1_1DenseOpticalFlow.html).
- OpenCV. (2021c). Functions and classes used to perform various linear or non-linear filtering operations on 2D images. Repéré à [https://docs.opencv.org/3.4/d4/d86/group\\_\\_imgproc\\_\\_filter.html](https://docs.opencv.org/3.4/d4/d86/group__imgproc__filter.html).
- Prewitt, J. M. (1970). Object enhancement and extraction. *Picture processing and Psychopictorics*, 10(1), 15–19.
- Rafael C. Gonzalez, R. E. W. (2018). *Digital image processing* (éd. 4). Pearson.
- Roberts, L. G. (1963). *Machine perception of three-dimensional solids*. (Thèse de doctorat, Massachusetts Institute of Technology).
- Shah, B. K., Kedia, V., Raut, R., Ansari, S. & Shroff, A. (2020). Evaluation and Comparative Study of Edge Detection Techniques. *IOSR Journal of Computer Engineering*, 22(5), 6–15.
- Shao, L., Yan, R., Li, X. & Liu, Y. (2013). From heuristic optimization to dictionary learning : A review and comprehensive comparison of image denoising algorithms. *IEEE transactions on cybernetics*, 44(7), 1001–1013.
- Siemens. (2019). Siemens gas turbine portfolio. Repéré à <https://assets.new.siemens.com/siemens/assets/api/uid:10f4860b140b2456f05d32629d8d758dc00bcc30/gas-turbines-siemens-interactive.pdf>.
- Siemens. (2021). Reliable gas turbines. Repéré à <https://www.siemens-energy.com/global/en/offerings/power-generation/gas-turbines.html>.
- SiemensEnergy. (2021). SGT - A35. Repéré à <https://www.siemens-energy.com/global/en/offerings/power-generation/gas-turbines/sgt-a30-a35-rb.html>.
- Singh, I. & Neeru, N. (2014). Performance comparison of various image denoising filters under spatial domain. *International Journal of Computer Applications*, 96(19), 21–30.
- Singh, S. & Singh, R. (2015). Comparison of various edge detection techniques. *2015 2nd International Conference on Computing for Sustainable Global Development (INDIACom)*,

- pp. 393–396.
- Smith, S. M. & Brady, J. M. (1997). SUSAN—a new approach to low level image processing. *International journal of computer vision*, 23(1), 45–78.
- Sobel, I. (1970). Camera models and perception. *Ph. D. Thesis*.
- Szeliski, R. (2021). *Computer Vision : Algorithms and Applications* (éd. 2). Springer.
- Teng, Y., Zhang, Y., Chen, Y. & Ti, C. (2015). Adaptive morphological filtering method for structural fusion restoration of hyperspectral images. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 9(2), 655–667.
- Thanh, D. N. H., Engínođlu, S. et al. (2019). An iterative mean filter for image denoising. *IEEE Access*, 7, 167847–167859.
- Tomasi, C. & Manduchi, R. (1998). Bilateral filtering for gray and color images. *Sixth international conference on computer vision (IEEE Cat. No. 98CH36271)*, pp. 839–846.
- towards data science. (2020). Practical implementation of outlier detection in python. Repéré à <https://towardsdatascience.com/practical-implementation-of-outlier-detection-in-python-90680453b3ce>.
- Treece, G. (2016). The bitonic filter : linear filtering in an edge-preserving morphological framework. *IEEE Transactions on Image Processing*, 25(11), 5199–5211.
- Weickert, J. (1996). Theoretical foundations of anisotropic diffusion in image processing. Dans *Theoretical foundations of computer vision* (pp. 221–236). Springer.
- Xiong, Z., Ramchandran, K., Orchard, M. T. & Zhang, Y.-Q. (1999). A comparative study of DCT-and wavelet-based image coding. *IEEE Transactions on circuits and systems for video technology*, 9(5), 692–695.
- Yang, R., Yin, L., Gabbouj, M., Astola, J. & Neuvo, Y. (1995). Optimal weighted median filtering under structural constraints. *IEEE transactions on signal processing*, 43(3), 591–604.
- Yilmaz, A., Javed, O. & Shah, M. (2006). Object tracking : A survey. *Acm computing surveys (CSUR)*, 38(4), 13–es.
- Zhang, L., Bao, P. & Wu, X. (2005). Multiscale LMMSE-based image denoising with optimal wavelet selection. *IEEE Transactions on circuits and systems for video technology*, 15(4), 469–481.