



Segmentation et mesures géométriques : application aux objets tubulaires métalliques

Nicolas Aubry

► To cite this version:

Nicolas Aubry. Segmentation et mesures géométriques : application aux objets tubulaires métalliques. Synthèse d'image et réalité virtuelle [cs.GR]. Université de Lorraine, 2017. Français. NNT : 2017LORR0110 . tel-01649874

HAL Id: tel-01649874

<https://theses.hal.science/tel-01649874>

Submitted on 27 Nov 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



AVERTISSEMENT

Ce document est le fruit d'un long travail approuvé par le jury de soutenance et mis à disposition de l'ensemble de la communauté universitaire élargie.

Il est soumis à la propriété intellectuelle de l'auteur. Ceci implique une obligation de citation et de référencement lors de l'utilisation de ce document.

D'autre part, toute contrefaçon, plagiat, reproduction illicite encourt une poursuite pénale.

Contact : ddoc-theses-contact@univ-lorraine.fr

LIENS

Code de la Propriété Intellectuelle. articles L 122. 4

Code de la Propriété Intellectuelle. articles L 335.2- L 335.10

http://www.cfcopies.com/V2/leg/leg_droi.php

<http://www.culture.gouv.fr/culture/infos-pratiques/droits/protection.htm>

Segmentation et mesures géométriques : application aux objets tubulaires métalliques

THÈSE

présentée et soutenue publiquement le 12 Juillet 2017

pour l'obtention du

Doctorat de l'Université de Lorraine

(mention informatique)

par

Nicolas Aubry

Composition du jury

Président : Laurent WENDLING (Professeur, Université Paris Descartes)

Rapporteurs : Laure TOUGNE (Professeur, Université de Lyon 2)
Pascal DESBARATS (Professeur, Université de Bordeaux)

Examineurs : Sylvain CONTASSOT-VIVIER (Professeur, Université de Lorraine)
Olivier VALLET (Docteur, Entreprise Numalliance)
Laurent WENDLING (Professeur, Université Paris Descartes)

Directeurs : Isabelle DEBLED-RENNESSON (Professeur, Université de Lorraine)
Bertrand KERAUTRET (Maitre de Conférences, Université de Lorraine)

Mis en page avec la classe thesul.

*à mes parents, mon frère
et ma femme Carole
qui m'ont toujours soutenu.*

Résumé

La présence de spécularité sur un objet est un problème récurrent qui limite l'application de nombreuses méthodes de segmentation. En effet, les spécularités sont des zones ayant une intensité très élevée et perturbent énormément la détection dès lors que l'on utilise la notion de gradient de l'image. Les travaux menés dans cette thèse permettent de proposer une nouvelle méthode de détection d'un objet tubulaire métallique dans une image. La méthode s'affranchit de la notion de gradient en utilisant la notion de profil d'intensité. Nous proposons dans ce manuscrit, un processus qui parcourt des zones rectangulaires prédéfinies de l'image, par balayage d'un segment discret à la recherche d'un profil d'intensité référence.

Ces travaux s'inscrivent dans une collaboration avec Numalliance, une entreprise qui fabrique des machines-outils. Cette collaboration permet de mettre en pratique cette méthode dans le cadre d'un système de contrôle qualité automatique et temps-réel des pièces manufacturées par les machines-outils. Pour cela, la méthode présentée doit être rapide, robuste aux spécularités et à l'environnement industriel tout en étant suffisamment précise pour permettre de conclure sur la conformité ou non de la pièce.

Mots-clés: segmentation, tubulaire, métallique, spécularité, géométrie discrète, contrôle qualité, environnement industriel.

Abstract

The presence of specularities on an object is a recurring problem that limits the application of many segmentation methods. Indeed, specularities are areas with a very high intensity and greatly disturb the detection when the notion of gradient of the image is used. The work carried out in this thesis makes it possible to propose a new detection method for a metallic tubular object in an image. The method avoids the notion of gradient by using the notion of intensity profile. We propose in this manuscript a process which traverses predefined rectangular areas of the image by scanning a discrete segment in search of a reference intensity profile.

This work is part of a collaboration with Numalliance, a company that manufactures machine tools. This collaboration enables this method to be put into a real industrial application as part of an automatic and real-time quality control system for parts manufactured by machine tools. To this end, the method presented must be fast, robust to the specularities and to the industrial environment while being sufficiently precise to make it possible to conclude on the conformity or not of the part.

Keywords: segmentation, tubular, metallic, specularity, discrete geometry, quality control, industrial environment.

Table des matières

Table des figures	vii
Chapitre 1 Introduction	1
Partie I État de l’art	9
Chapitre 2 Segmentation & détection de segments/droites	11
2.1 Segmentation d’images	12
2.1.1 Approches « contours »	14
2.1.2 Approches « régions »	19
2.1.3 Expérimentations en milieu industriel	24
2.2 Détection de segments ou droites	29
2.2.1 Méthodes choisies	29
2.2.2 Expérimentations en milieu industriel	42
Chapitre 3 Géométrie discrète	49
3.1 Droites discrètes 2D	50
3.2 Segments flous	51
3.3 Détection de segments dans une image en niveaux de gris	57
Partie II Contributions	63
Chapitre 4 Méthode des profils d’intensité	65

4.1	Motivation	66
4.2	Notre approche	71
4.2.1	Notions importantes	72
4.2.2	Exploration d'une zone de recherche	74
4.3	Extension de la méthode : interpolation	82
4.4	Analyse des deux algorithmes	86
4.5	Extension aux images non industrielles et limites	94
Chapitre 5 Validation & intégration industrielle		97
5.1	Validation	98
5.1.1	Résultats obtenus avec les critères respectés	98
5.1.2	Résultats obtenus avec les critères non respectés	100
5.1.3	Résultats avec translation des zones de recherche	105
5.2	Intégration industrielle	113
5.3	Description d'un premier prototype	117
Chapitre 6 Conclusion & perspectives		123
Bibliographie		127

Table des figures

1.1	Illustration des spécularités sur une pièce métallique.	2
1.2	Illustration d'une machine à commandes numériques Numalliance (a) et de quelques exemples de pièces manufacturées avec celle-ci (b)	3
1.3	Gabarit avec une pièce que l'on veut contrôler (a) et résultat du contrôle (b)	4
1.4	Exemple de scanner 3D proposé aux industriels.	4
1.5	Définition des segments droits en rouge et des segments de tores en vert de la pièce.	6
1.6	Gabarit 2D virtuel en rouge avec la détection des segments droits de la pièce en vert. Si toutes les parties sont comprises dans le gabarit alors la pièce est conforme.	6
2.1	Les quatre modèles utilisés pour les différentes expérimentations.	13
2.2	Modèle avec spécularités sur fond uni.	13
2.3	Segmentation des cheveux de Lena (b). Résultat idéal de la segmentation d'un objet tubulaire métallique (d).	15
2.4	Illustration des superpixels avec différents paramètres [ASS ⁺ 12].	22
2.5	Une image et son arbre de composantes connexes [PNR ⁺ 11].	23
2.6	Résultats obtenus avec le détecteur de contours Canny avec une taille de noyau de 3, un seuil bas de 20 et haut de 60.	25
2.7	Résultats obtenus avec l'algorithme de détection de contours Marr-Hildreth [IPO16] avec une déviation standard $\sigma = 3$, une taille de noyau de 2 et un seuil de passage par 0 de $tcz = 0,07$	26
2.8	Résultats obtenus avec l'algorithme des snakes avec un paramètre de seuil de gradient de 10.	27

2.9	Résultats obtenus avec la segmentation de Chan-Vese [IPO16] avec le paramètre $\mu = 0.2$.	28
2.10	Résultats obtenus avec l'algorithme de Watershed (Partie 1).	30
2.11	Résultats obtenus avec l'algorithme de Watershed (Partie 2).	31
2.12	Résultats obtenus avec l'algorithme des Superpixels SLIC.	32
2.13	Résultats obtenus à l'aide des arbres de composantes connexes avec un paramètre $\alpha = 0.1$ (Partie 1).	33
2.14	Résultats obtenus à l'aide des arbres de composantes connexes (Partie 2).	34
2.15	Résultat idéal de la détection des couples de segments parallèles représentant les zones droites de l'objet tubulaire métallique.	35
2.16	Définition d'une droite en coordonnées polaires pour la transformée de Hough.	36
2.17	(à gauche) Trois segments avec (à droite) leur correspondance dans l'espace de Hough.	36
2.18	Résultats obtenus avec la transformée de Hough probabiliste pour la détection de droites.	43
2.19	Résultats obtenus avec une méthode interactive utilisant la transformée de Hough [EM00].	44
2.20	Résultats obtenus avec l'algorithme de détection de droites : LSD (Line Segment Detector).	45
2.21	Résultats obtenus avec l'algorithme de détection de droites : EDLines.	46
2.22	Résultats obtenus avec l'algorithme de détection de droites : CannyLines.	47
3.1	Exemple de segment discrète arithmétique.	51
3.2	Illustration des différentes connexités des droites discrètes par rapport à l'épaisseur ω .	52
3.3	Image extraite du manuscrit [DR07] pour l'obtention d'une Habilitation à diriger des recherches : (a) Un exemple de courbe discrète obtenue dans un contour de l'image d'une feuille. (b) Un agrandissement sur une partie du contour et segmentation obtenue avec un algorithme de reconnaissance de segments de droites naïves. (c) La segmentation souhaitée qui présente mieux la forme géométrique sur la même partie de la courbe.	53

3.4	Segment flou d'épaisseur 1 encadré par droite discrète $D(5, 8, -8, 11)$ - Les points considérés sont les points gris et les points bleus sont optimaux.	54
3.5	Parcours du segment S_0 (en noir) dans la direction orthogonale à droite et à gauche.	58
3.6	Illustration de la sélection d'un pixel sur un segment de droite ayant le gradient le plus fort et dont l'angle entre la direction de parcours et la direction du gradient est inférieur au seuil $\theta_{max} = \{\frac{\pi}{2}, \frac{\pi}{8}, \frac{\pi}{10}\}$ (Les auteurs précisent que se sont des valeurs empiriques validées sur des images réelles).	59
3.7	Illustration de segments flous obtenus (en bleu) sur une image réelle à partir de sélections représentées par les segments rouges sur l'image.	60
4.1	Illustration des trois composantes du modèle de réflectance de Nayar [NIK91] obtenues à l'aide d'une source de lumière fixe (a). L'image (b) illustre la réflexion de la lumière sur une surface d'un objet cylindrique.	67
4.2	Illustration des structures linéaires que nous retrouvons sur un objet tubulaire.	68
4.3	Expérimentations autour de l'étude des variations de profil d'intensité sur une forme tubulaire réaliste lorsque la distance de la source de la lumière est modifiée ou encore les paramètres de rendu. La première et deuxième lignes montrent deux exemples où nous avons utilisé deux positions de sources lumineuses différentes ($L_d = 100$ and $L_d = 500$). La troisième et dernière ligne montrent les variations de profil d'intensité définies à partir de l'erreur quadratique moyenne à partir d'un profil de référence.	69
4.4	Comparaison entre le profil d'intensité moyen d'un segment droit du tube et trois profils d'intensité pris au début, milieu et fin du segment droit du tube.	70
4.5	Visualisation 3D des intensités de la zone 3 - source A (a) et de la zone 3 - source B (b).	71
4.6	Illustration montrant les différentes structures utilisées par la méthode.	72

4.7	Illustration d'une droite discrète naïve $D(2, 5, 0, 5)$ en rouge avec ses paramètres a et b . La valeur t_r permettant d'obtenir les segments $D(2, 5, 0, 5)$ (comme par exemple le segment discret en bleu) « parallèles » de celui-ci est aussi illustrée.	74
4.8	Définition manuelle des différentes zones de recherche $\mathcal{A}_k$ en rouge et profils d'intensité $\mathcal{P}_{ref}^k$ associés en bleu.	75
4.9	Parcours de la zone de recherche $\mathcal{A}_0$ en fuchsia à l'aide d'une génération de segments de recherche $\mathcal{R}_0^i$ en rouge dans la direction du vecteur en noir. . .	76
4.10	Parcours du segment de recherche $\mathcal{R}_0^{50}$ par translation unitaire d'une zone d'intérêt $\mathcal{I}_{50}^{l,m}$ représenté en vert.	77
4.11	Visualisation du résultat après choix des zones d'intérêt par comparaison de profils d'intensité (a) — Illustration du problème de reconnaissance des segments flous si les premières zones d'intérêt sont fausses (b).	81
4.12	Visualisation des deux segments flous résultant (a) — Visualisation des deux segments flous résultats lors d'une mauvaise détection dès le départ (b) — en rouge les points refusés par l'algorithme de construction des segments flous.	82
4.13	Effet de la discrétisation de l'image	83
4.14	Illustration de la méthode de projection des profils références.	84
4.15	Illustration de la méthode de projection des zones d'intérêt.	85
4.16	Illustration utilisée pour l'étude de l'algorithme sans interpolation.	86
4.17	Illustration utilisée pour l'étude de l'algorithme avec interpolation.	87
4.18	Graphique du profil d'intensité référence $\mathcal{P}_{ref}$ pour l'illustration de la méthode sans interpolation (a - b) et avec interpolation (c - d).	88
4.19	Résultats obtenus après parcours de la zone de recherche $\mathcal{I}_i^{l,m}$ sur le segment de recherche $\mathcal{R}_k^i$ avec la méthode avec interpolation — Les MSE en vert correspondent aux nouveaux résultats obtenus avec la méthode avec interpolation.	90
4.20	Résultats obtenus après parcours de la zone de recherche $\mathcal{I}_j^{l,m}$ sur le segment de recherche $\mathcal{R}_k^j$ avec la méthode avec interpolation — Les MSE en vert correspondent aux nouveaux résultats obtenus avec la méthode avec interpolation.	91

4.21	Résultats obtenus après parcours de la zone de recherche $\mathcal{I}_q^{l,m}$ sur le segment de recherche $\mathcal{R}_k^q$ avec la méthode avec interpolation — Les MSE en vert correspondent aux nouveaux résultats obtenus avec la méthode avec interpolation.	92
4.22	Graphiques représentant le profil d'intensité référence et les profils d'intensité représentant les zones d'intérêt sélectionnées $\{\mathcal{I}_i^{l,m}, \mathcal{I}_j^{l,m}, \mathcal{I}_q^{l,m}\}$ sur les segments de recherche $\{\mathcal{R}_k^i, \mathcal{R}_k^j, \mathcal{R}_k^q\}$ avec la méthode sans (a) et avec (b) interpolation.	93
4.23	Résultats obtenus sur une image urbaine — en vert le profil d'intensité référence, en violet la zone de recherche et en bleu le résultat de la détection.	95
4.24	Résultats obtenus sur un bâtiment — en vert le profil d'intensité référence, en violet la zone de recherche et en bleu le résultat de la détection.	96
5.1	Résultats de l'algorithme sur le modèle 1 — (a) Initialisation de l'algorithme — (b) version sans interpolation — (c) version avec interpolation.	99
5.2	Résultats de l'algorithme sur le modèle 3 — (a) Initialisation de l'algorithme — (b) version sans interpolation — (c) version avec interpolation.	101
5.3	Résultats de l'algorithme sur le modèle 4 — (a) Initialisation de l'algorithme — (b) version sans interpolation — (c) version avec interpolation.	102
5.4	Résultats de l'algorithme sur le modèle « fond vert » — (a) Initialisation de l'algorithme — (b) version sans interpolation — (c) version avec interpolation.	103
5.5	Résultats de l'algorithme sur le modèle 2 — (a) Initialisation de l'algorithme — (b) version sans interpolation — (c) version avec interpolation.	104
5.6	Mesures de temps de l'exécution de l'algorithme sur toutes les zones de recherche d'une image — (a) sans interpolation — (b) avec interpolation.	105
5.7	Résultats de l'algorithme sur les modèles 1 et 3 après translation des zones de recherche.	106
5.8	Histogrammes représentant la répartition des scores MSE des zones d'intérêt qui ont été acceptées avec la méthode <i>sans interpolation</i> sur le modèle 1 après translation des zones de recherche — (a) Zone de recherche 1 — (b) Zone de recherche 2 — (c) Zone de recherche 3 — (d) Zone de recherche 4.	107

5.9	Histogrammes représentant la répartition des scores MSE des zones d'intérêt qui ont été acceptées avec la méthode <i>sans interpolation</i> sur le modèle 3 après translation des zones de recherche — (a) Zone de recherche 1 — (b) Zone de recherche 2 — (c) Zone de recherche 3 — (d) Zone de recherche 4.	108
5.10	Histogrammes représentant la répartition des scores MSE des zones d'intérêt qui ont été acceptées avec la méthode <i>sans interpolation</i> sur le modèle 1 — (a) Zone de recherche 1 — (b) Zone de recherche 2 — (c) Zone de recherche 3 — (d) Zone de recherche 4.	109
5.11	Histogrammes représentant la répartition des scores MSE des zones d'intérêt qui ont été acceptées avec la méthode <i>sans interpolation</i> sur le modèle 3 — (a) Zone de recherche 1 — (b) Zone de recherche 2 — (c) Zone de recherche 3 — (d) Zone de recherche 4.	110
5.12	Histogrammes représentant la répartition des scores MSE des zones d'intérêt qui ont été acceptées avec la méthode <i>avec interpolation</i> sur le modèle 1 — (a) Zone de recherche 1 — (b) Zone de recherche 2 — (c) Zone de recherche 3 — (d) Zone de recherche 4.	111
5.13	Histogrammes représentant la répartition des scores MSE des zones d'intérêt qui ont été acceptées avec la méthode <i>avec interpolation</i> sur le modèle 3 — (a) Zone de recherche 1 — (b) Zone de recherche 2 — (c) Zone de recherche 3 — (d) Zone de recherche 4.	112
5.14	Mesures d'angles entre la direction de la détection et la direction théorique du tube pour chaque zone de recherche sur les modèles 1 et 3 — en rouge les grandes déviations par rapport à la direction théorique du tube.	113
5.15	Schéma global du processus du système de contrôle qualité.	114
5.16	Définition d'une pièce dans son modèle $LRAr$. On retrouve cette écran de spécification de pièce directement dans l'interface de la machine-outil.	115
5.17	Illustration de la partie simulation du logiciel produit par Numalliance.	115
5.18	Visualisation d'un modèle 3D d'une pièce (a) — Illustration d'un recalage d'une pièce 3D virtuelle sur la pièce 2D dans l'image (b).	118
5.19	Illustration d'une pièce 3D après recalage (a) — Illustration de la génération des zones de recherche, tolérance et profils d'intensité références sur le modèle 3D recalé (b).	118

5.20	Illustration de la visualisation de la projection des profils d'intensité références dans l'espace image.	119
5.21	Illustration de la visualisation des résultats, obtenus à l'aide des informations extraites du processus — en rouge la dernière détection obtenue. Il est possible de naviguer entre les zones de recherche pour visualiser la détection obtenue.	121
6.1	Zones droites (en rouge) et courbes (en vert) d'un objet tubulaire.	126

Chapitre 1

Introduction

DANS le domaine de l'image numérique, la recherche ou l'identification d'un ou plusieurs objets précis dans une image est un problème classique que l'on retrouve dans de nombreuses applications. Nous l'effectuons même tous les jours, naturellement, à l'aide de notre propre système visuel. Pour accomplir cette tâche, nous effectuons ce que l'on appelle une segmentation de l'image. L'objectif de la segmentation est de partitionner une image en différentes régions cohérentes selon des critères choisis en amont. Ces différentes régions représentent des objets dans l'image que l'on va ainsi pouvoir manipuler ou encore analyser.

Il existe pour cela une grande quantité de méthodes (voir [Chapitre 2](#)) et ce domaine reste un champ de recherche très actif. Pour se convaincre, il suffit de se rendre sur le site du journal en ligne IPOL [[IPO16](#)] et de voir les centaines, voire milliers d'essais qui sont réalisés sur différents algorithmes de la section *Segmentation and Edges*. Ces méthodes, selon la complexité de l'image (i.e. type(s) d'objet(s), environnement de(s) l'objet(s), etc.), vont se montrer plus ou moins efficaces. Il existe deux caractéristiques qui posent la majeure partie des problèmes aux différentes méthodes de segmentation :

- Les objets comportant des spécularités comme par exemple les objets métalliques (figure [1.1](#)).
- Environnement non contrôlé (e.g. éclairage, fond, etc.) comme un environnement industriel ou naturel.
- Les objets qui sont fondus dans l'environnement, qui ne présentent donc pas ou peu de différences d'intensités entre eux et le fond (i.e. absence de contours).

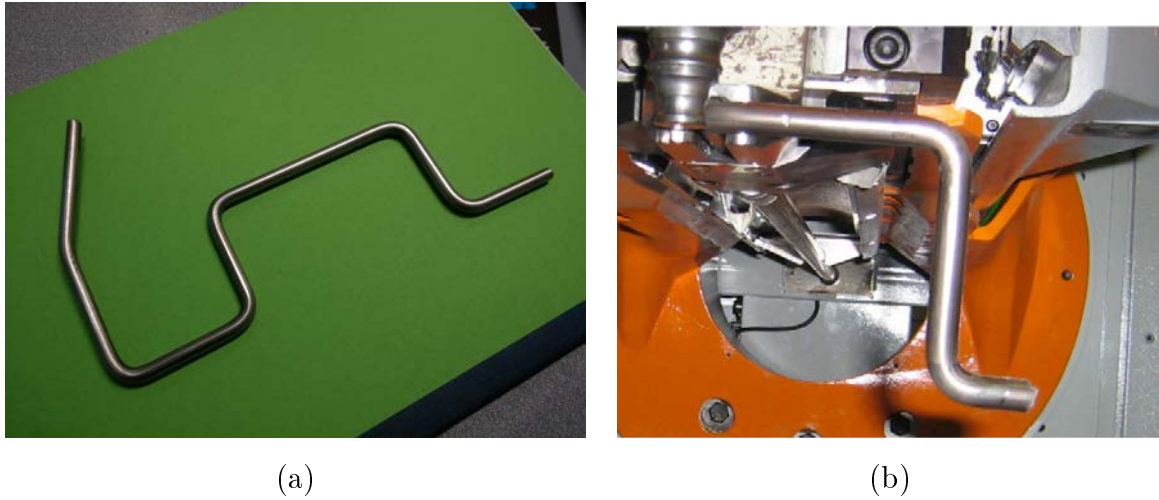


FIGURE 1.1 – Illustration des spécularités sur une pièce métallique.

Ce manuscrit de thèse s’inscrit précisément dans cette problématique. Il est le fruit d’une collaboration entre l’équipe Algorithmique Discrète et ses Applications à la Géométrie et à l’Imagerie (ADAGIo) du Laboratoire Lorrain de Recherche en Informatique et ses Applications (LORIA) et la société Numalliance, concepteur et fabricant de machines à commandes numériques (figure 1.2 (a)) pour plier le fil, le tube et le méplat métallique (figure 1.2 (b)). Cette collaboration a pour but de proposer une méthode alternative aux contrôles qualité [ISO15] qui sont effectués jusqu’à présent par les différents clients qui fabriquent des pièces à l’aide des machines de Numalliance. Cette thèse se place donc à la croisée de plusieurs domaines tels que la vision par ordinateur, le traitement d’images ainsi que la géométrie discrète. Ce manuscrit n’abordera cependant en détails que les deux derniers domaines et traitera en substance de la partie vision par ordinateur.

Le contrôle qualité est une facette de la gestion de la qualité d’un produit qui peut être un objet manufacturé, un service, un document ou encore un code source. Ce contrôle est une opération qui permet de déterminer à l’aide de différentes méthodes (contrôles par un humain ou automatique) si l’entité en contrôle est conforme ou non à des spécifications ou des exigences définies en amont. La conclusion du contrôle qualité est une décision qui est le plus souvent binaire : acceptation ou rejet mais il est possible d’ajouter de la nuance au système (e.g. ajouter une incertitude au contrôle pour faire une demande d’un second contrôle ou encore ajouter une notion intermédiaire de correction de l’entité).

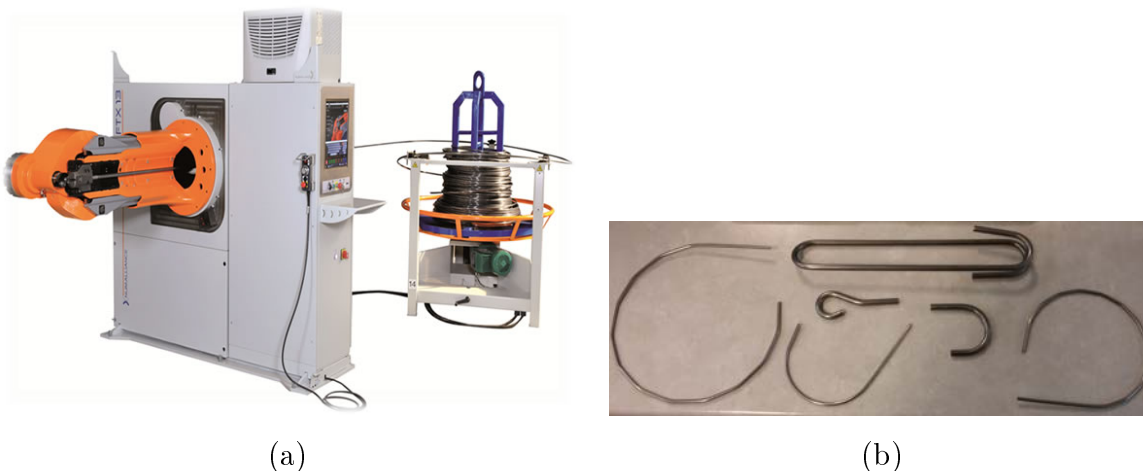


FIGURE 1.2 – Illustration d’une machine à commandes numériques Numalliance (a) et de quelques exemples de pièces manufacturées avec celle-ci (b)

Pour bien comprendre le processus dans ce domaine industriel voici les différentes caractéristiques des contrôles qualité effectués par les clients de Numalliance :

- Une fréquence de contrôle par prélèvement, c’est-à-dire que pour toutes les x pièces fabriquées, la dernière pièce produite est soumise au contrôle pour vérifier que la production n’a pas subi une déviation de qualité.
- Une méthode de contrôle par comparaison en utilisant un moule solide de la pièce que l’on veut vérifier et que l’on appelle communément un gabarit (figure 1.3 (a)). Ce gabarit reflète les tolérances que l’on a définies pour la pièce en fabrication. L’opérateur de contrôle qualité insère donc la pièce qui a été choisie pour vérification dans le gabarit. Si celle-ci entre entièrement et sans résistance alors elle est considérée comme conforme aux spécifications ainsi que toutes les pièces fabriquées précédemment (figure 1.3 (b)).
- Une méthode de contrôle par comparaison à l’aide d’un scanner 3D (figure 1.4) peut être aussi envisagée. Cela consiste à scanner une pièce pour effectuer ensuite une comparaison avec un modèle 3D de la pièce théorique.

Il est important de noter que cette étape de contrôle n’est pas obligatoire mais lorsque celle-ci est introduite dans le processus de fabrication, elle est critique sur plusieurs aspects. En effet, il est indispensable que cet ajout impacte le moins possible le temps de production soit parce que la méthode n’est pas suffisamment rapide, soit parce qu’elle



FIGURE 1.3 – Gabarit avec une pièce que l'on veut contrôler (a) et résultat du contrôle (b)

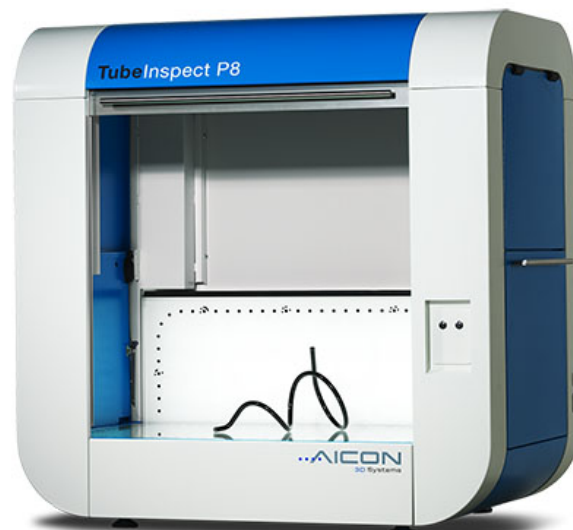


FIGURE 1.4 – Exemple de scanner 3D proposé aux industriels.

n'est pas assez robuste et insère donc des faux positifs (resp. faux négatifs) aux résultats. De plus, suivant la fréquence des vérifications, il existe un fort impact économique. En effet, pour ne pas trop interrompre la production, il est judicieux de choisir une faible fréquence. Mais lorsque l'on détecte une anomalie, nous devons soit retirer toutes les pièces fabriquées entre les deux pièces contrôlées (i.e. perte économique) ou tester toutes les pièces pour éliminer uniquement les mauvaises pièces (i.e. perte de temps et économique).

Les exigences de Numalliance sont de proposer une méthode de contrôle qualité simple à mettre en œuvre pour le client sans modifications excessives sur les machines déjà existantes tout en utilisant les caractéristiques suivantes :

- Une fréquence de contrôle systématique de la production.
- Une méthode de comparaison automatique à l'aide d'un gabarit virtuel pour supprimer la fabrication longue et coûteuse d'un modèle physique de gabarit.
- Une méthode rapide pour impacter le moins possible le temps de production.
- Une méthode qui permet de fonctionner dans de nombreuses conditions, c'est-à-dire qu'elle doit fonctionner dans un environnement non contrôlé. Beaucoup de méthodes de contrôle proposent un environnement contrôlé comme par exemple une boîte noire où l'on peut facilement contrôler l'éclairage de la scène et donc s'affranchir des problèmes liés au fond.

Pour satisfaire ces différentes conditions, nous nous sommes dirigés vers un contrôle à partir d'une unique caméra qui va effectuer un cliché de la pièce après ou pendant production et la méthode devra, à partir de cette photo, déterminer si cette pièce est conforme ou non aux spécifications. Ce procédé implique que nous devons être capables d'identifier dans l'image, et le plus précisément possible, la pièce pour pouvoir effectuer des mesures. Dans ce manuscrit, nous allons nous intéresser à la détection des segments droits de la pièce sans se préoccuper des segments de tores (figure 1.5). En effet, comme nous pouvons le voir dans l'illustration (figure 1.6), cette information est tout à fait suffisante pour effectuer une vérification du type « comparaison » avec un gabarit virtuel.

Pour identifier et analyser la pièce, nous allons nous intéresser, en plus de la segmentation d'image, à la reconnaissance de segments discrets. Ce type d'algorithme met en exergue, contrairement aux méthodes de segmentation, uniquement les ensembles de pixels de l'image qui forment des segments discrets selon certains critères mathématiques.

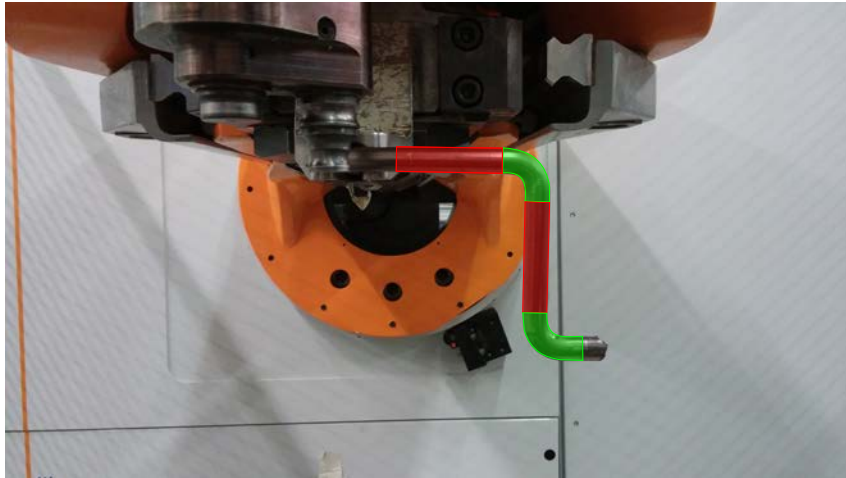


FIGURE 1.5 – Définition des segments droits en rouge et des segments de tores en vert de la pièce.

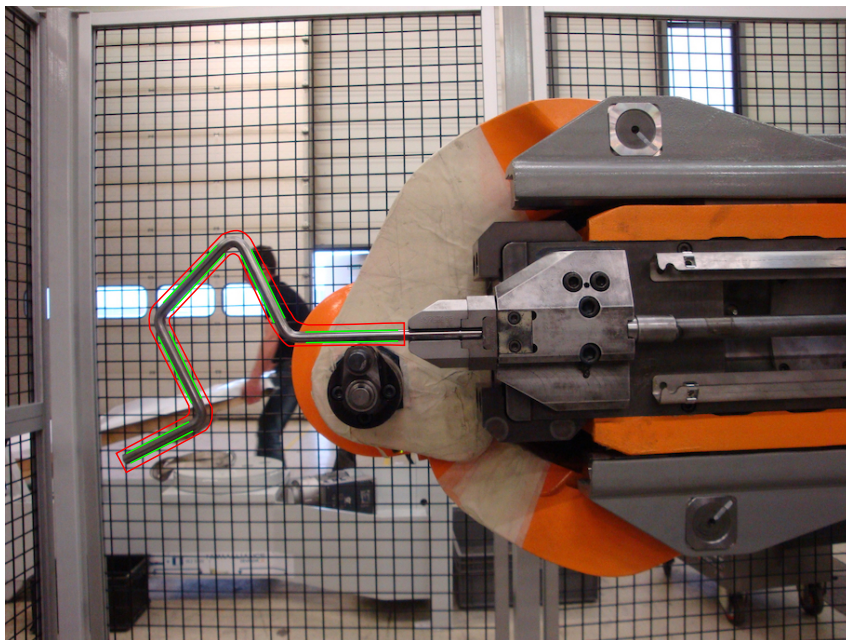


FIGURE 1.6 – Gabarit 2D virtuel en rouge avec la détection des segments droits de la pièce en vert. Si toutes les parties sont comprises dans le gabarit alors la pièce est conforme.

Ces deux types de méthodes paraissent les plus adaptés pour la détection des zones droites de la pièce (i.e. nous pouvons voir une zone droite de la pièce comme deux segments de droites parallèles).

Pour développer plus en détail la solution que nous proposons, la suite du manuscrit sera découpée en cinq chapitres. Nous présenterons dans le [Chapitre 2](#), différentes méthodes dites « classiques » ou plus récentes de segmentation et de détection de droites. Nous verrons en détails leurs comportements sur des images de pièces en production dans un environnement industriel contrôlé ou non. Nous pourrions mettre en avant les problèmes récurrents que l'on obtient sur des images possédant des spécularités. Ces spécularités sont une des caractéristiques intrinsèques des objets fabriqués par les machines de Numalliance. En effet, la majeure partie des pièces que les machines manufacturent sont des fils (ou tubes) métalliques donc à fort potentiel spéculaire. Les spécularités sont représentées sur une image par une ou plusieurs augmentation(s) brusque(s) de l'intensité, ce qui a pour effet de perturber la majeure partie des algorithmes qui utilisent justement les changements d'intensité pour identifier les bords d'un objet (i.e. notion de gradient - dans la figure [2.2](#) les spécularités au milieu de la pièce provoquent des erreurs de détection).

Dans le [Chapitre 3](#), nous verrons certains concepts et outils utilisés dans la méthode et qui font partie du domaine de la géométrie discrète comme par exemple la notion de segment flou. Nous verrons une illustration de l'utilisation des segments flous, dans une image en niveaux de gris, à l'aide d'une méthode proposée par Kerautret et Even [[KE09](#)]. Cette méthode propose un parcours incrémental de l'image qui sera le point de départ de notre méthode.

Le [Chapitre 4](#) présente notre proposition pour identifier les différents segments droits d'une pièce en s'affranchissant le plus possible des problèmes récurrents dus aux spécularités. Cette proposition introduit notamment la notion de profil d'intensité et exploite un certain nombre de données générées depuis l'étape de vision par ordinateur. Cette étape sera uniquement survolée car cela ne fait pas partie de la problématique étudiée dans ce manuscrit.

Dans le [Chapitre 5](#), nous montrerons comment la méthode s'intègre globalement dans le processus de contrôle qualité. Nous caractériserons pour cela les différents éléments qui le compose. Nous explorerons ensuite les résultats obtenus à l'aide de cet algorithme en

l'exécutant sur des séries de photos représentant la production de plusieurs modèles de pièces dont on aimerait connaître la justesse par rapport à leurs modèles théoriques.

Nous conclurons ce manuscrit dans le [Chapitre 6](#) en résumant les contributions apportées par cette méthode, ainsi que les limites abordées précédemment. Nous terminerons ainsi par une discussion sur les perspectives envisageables pour résoudre ou atténuer ces différentes limitations. Nous verrons aussi des perspectives possibles pour étendre la méthode à son utilisation dans des cadres différents, plus génériques que celui de l'industrie.

Première partie

État de l'art

Chapitre 2

Segmentation & détection de segments/droites

Sommaire

2.1	Segmentation d'images	12
2.1.1	Approches « contours »	14
2.1.2	Approches « régions »	19
2.1.3	Expérimentations en milieu industriel	24
2.2	Détection de segments ou droites	29
2.2.1	Méthodes choisies	29
2.2.2	Expérimentations en milieu industriel	42

COMME nous l'avons évoqué dans le chapitre précédent, l'objectif de cette partie est d'explorer les différentes méthodes capables de retrouver dans une image un objet tubulaire métallique. Plus précisément, nous voulons extraire les zones droites de l'objet (i.e. les zones courbes ne sont pas la priorité du sujet et feront l'objet d'une discussion dans les perspectives). Pour mener à bien cet objectif, les candidats les plus adaptés parmi les méthodes du domaine du traitement d'images apparaissent être les méthodes de segmentation et les détections de segments/droites.

Mais avant d'aller plus dans les détails, nous devons définir plus formellement les éléments que nous allons utiliser. En particulier, l'objet que nous allons manipuler tout au long de ce manuscrit est une image en deux dimensions que l'on peut définir comme

ceci :

Definition 1. Une *image* I en deux dimensions est une fonction du type :

- $\mathbb{N}^2 \rightarrow \mathbb{Z}$
- $(x, y) \mapsto i$

où le couple (x, y) correspond aux coordonnées d'un point dans l'image que l'on nomme *pixel* et i la valeur de ce pixel.

Dans ce chapitre, nous allons présenter un certain nombre de méthodes de segmentation qui vont permettre d'extraire de l'image I soit directement les contours de l'objet par une approche « contours » ou alors l'objet lui-même par une approche « régions ». Puis nous allons étudier des méthodes qui permettent d'obtenir les caractéristiques du contour de notre objet à l'aide d'algorithmes de détection de droites ou segments de droites. Pour cela, le chapitre sera découpé en deux sections correspondant à différentes familles d'algorithmes :

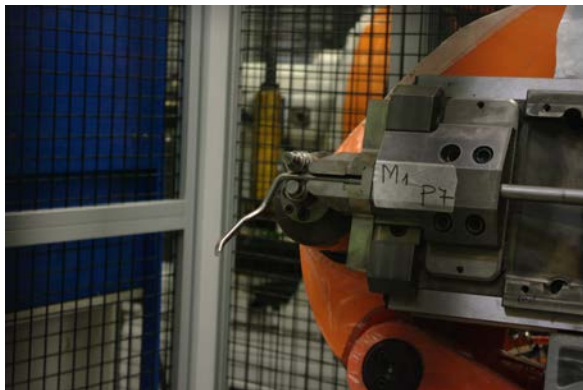
- [Segmentation](#)
- [Détection de segments ou droites](#)

Pour chaque algorithme, nous analyserons leur comportement lorsqu'on les applique sur quatre images de référence (figure [2.1](#)). Ces quatre images représentent des cas de prise de vue de pièces métalliques en fabrication dans leur environnement industriel. Pour chacun des algorithmes, nous conclurons sur les résultats obtenus, par rapport à l'aspect détection d'objet tubulaire métallique ainsi que pour l'aspect applicatif (i.e. le contrôle qualité).

Nous avons aussi ajouté une cinquième image [2.2](#) dans le but d'étudier le comportement des différents algorithmes uniquement sur les reflets spéculaires de la pièce grâce à l'utilisation d'un fond uni derrière elle.

2.1 Segmentation d'images

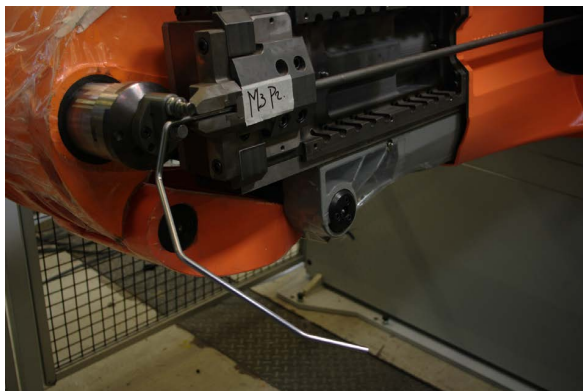
Dans la littérature parlant de la segmentation, nous pouvons retrouver communément, deux grandes catégories qui sont les approches « contours » et les approches « régions ». Ce sont des catégories que l'on peut qualifier de duales et nous les définissons ainsi :



(a)



(b)



(c)



(d)

FIGURE 2.1 – Les quatre modèles utilisés pour les différentes expérimentations.



FIGURE 2.2 – Modèle avec spécularités sur fond uni.

Definition 2. Une « région » est un sous-ensemble de pixels d'une image I . Ce sous-ensemble est « homogène » selon certains attributs comme par exemple le niveau de gris ou la texture des pixels.

Definition 3. Un « contour » est un sous-ensemble de pixels d'une image I qui marque une séparation entre une région et d'autres régions de cette même image.

Comme nous l'avons précisé dans l'introduction de ce manuscrit, la segmentation est une opération naturelle pour notre système visuel. En effet, nous identifions rapidement l'homogénéité d'une zone (i.e. régions) en terme de texture, intensité, couleurs etc. ou bien les contours des objets en considérant le changement plus ou moins brutal de l'homogénéité d'une zone (i.e. contours). A cette segmentation naturelle, nous y ajoutons une notion de sémantique, c'est-à-dire qu'une fois l'objet segmenté, nous l'identifions clairement. Par exemple, notre système visuel est capable d'identifier clairement les zones de l'image associées aux cheveux comme illustré sur la figure 2.3 (b).

Dans notre cas d'étude, nous voulons effectuer une segmentation avec un sens précis. Le but est d'obtenir une ou plusieurs régions qui représentent un objet tubulaire métallique dans une image (figure 2.3 (d)).

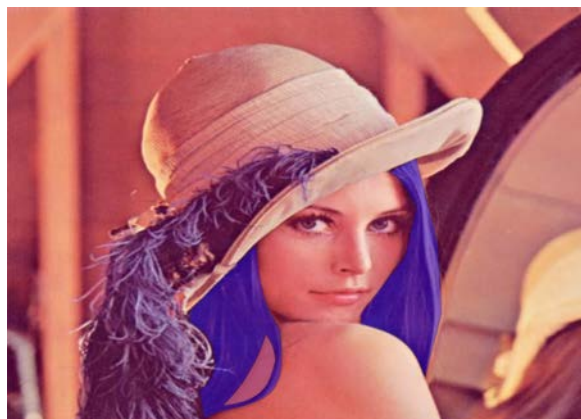
Il existe de nombreux algorithmes de segmentation, nous allons donc étudier un nombre non exhaustif de ceux-ci en commençant par les méthodes qui utilisent une approche « contours », puis une approche « régions ». Comme nous l'avons évoqué précédemment, nous les appliquerons sur les images références (figures 2.1 et 2.2) pour étudier leur comportement dans le cadre que nous nous sommes fixé tout au long de ce manuscrit. La figure 2.3 (d) représente un résultat idéal de ce que nous aimerions obtenir après une segmentation par approche « contours » (en rouge sur l'image) ou par approche « régions » (en bleu sur l'image).

2.1.1 Approches « contours »

Dans cette partie, nous étudierons dans un premier temps les algorithmes considérés comme classiques par leurs nombreuses utilisations ou citations dans d'autres productions scientifiques, puis nous verrons les méthodes utilisant le modèle de contour actif.



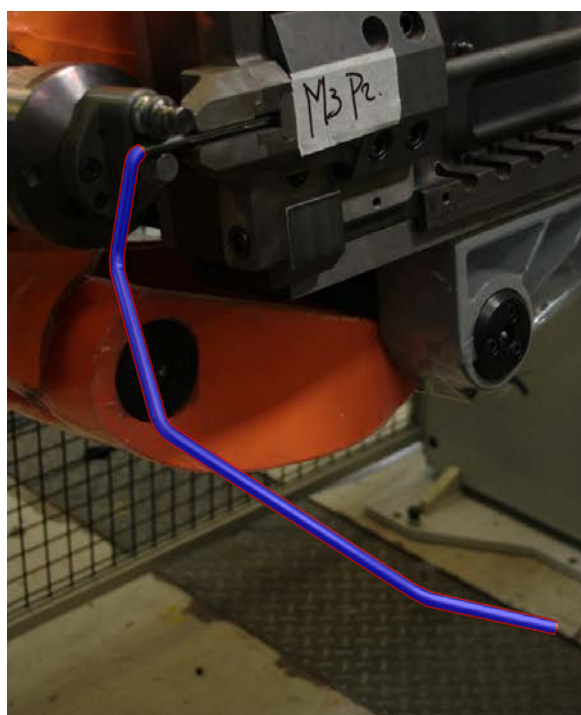
(a)



(b)



(c)



(d)

FIGURE 2.3 – Segmentation des cheveux de Lena (b). Résultat idéal de la segmentation d'un objet tubulaire métallique (d).

Algorithmes « classiques »

Dans la catégorie de la segmentation par approche « contours », la majeure partie des méthodes utilise la notion de dérivée locale du premier et second degré. Cette notion que l'on nomme « gradient » se prête effectivement très bien à la tâche de détection de contours car le changement plus ou moins brutal de l'information, par exemple de niveau de gris¹ est représenté par un maximum local dans le cas de la dérivée du premier degré et par un passage par zéro dans celui de la dérivée seconde.

En 1986, J. Canny [Can86] propose un processus en plusieurs étapes pour effectuer une détection de contours à l'aide du gradient calculé à partir de la dérivée du premier degré. Ce processus est utilisé dans de très nombreux algorithmes et se découpe ainsi :

1. On effectue un pré-traitement pour réduire le bruit présent dans l'image.
2. On calcule en chaque pixel de l'image I une approximation du gradient :

$$\nabla I = (g_x, g_y) = \left(\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y} \right) \quad (2.1)$$

3. On calcule pour chacun des gradients sa norme ainsi que son orientation :

$$Norme(x, y) = \sqrt{g_x^2 + g_y^2} \quad Direction(x, y) = \arctan \frac{g_x}{g_y} \quad (2.2)$$

4. On supprime les différents pixels représentés par un non-maximum local dans la direction du gradient.
5. On effectue un seuillage des maximums locaux sur les pixels restants de l'étape précédente.

Pour la première partie de pré-traitement, Canny propose d'utiliser un filtrage gaussien. Pour l'approximation du calcul du gradient, Canny adopte le filtre de Sobel [SF68] de la forme :

$$g_x = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix} \quad g_y = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix} \quad (2.3)$$

A l'aide de ce filtre, il procède par une première étape de suppression de pixels. Pour cela, il considère que chaque pixel, dont la norme $Norme(x, y)$ du gradient est inférieure

1. d'autres types d'informations peuvent être considérées comme par exemple la couleur, la texture, etc.

à celle de ces deux voisins dans la direction $Direction(x, y)$, n'est pas un bon candidat pour faire partie d'un contour. Puis, il effectue un seuillage par hystérésis en définissant un seuil bas et haut. Ces deux seuils permettent d'effectuer un second écrémage des pixels en suivant ces trois règles :

1. Si la norme $Norme(x, y)$ est inférieure au seuil bas, le pixel est rejeté.
2. Si elle est comprise entre le seuil bas et haut, le pixel est gardé uniquement si un de ses voisins a été précédemment accepté.
3. Si elle est supérieure au seuil haut, le pixel est considéré comme faisant partie d'un contour.

Pour ce qui est des méthodes utilisant plutôt la dérivée du second ordre, le processus introduit par Canny reste sensiblement le même à la différence que l'on utilise ici une approximation du laplacien :

$$\Delta I = (g_x, g_y) = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2} \quad (2.4)$$

Étant donné l'utilisation de la dérivée seconde, nous ne recherchons plus les maximums locaux, mais les passages par zéro du Laplacien.

L'utilisation de cet opérateur, introduite en 1980 par Marr et Hildreth [MH80], est historiquement intéressante car cela produit théoriquement des contours fermés contrairement aux méthodes utilisant la dérivée première. Au final, cela permet d'éviter une étape supplémentaire de poursuite et fermeture des contours mais l'approximation de la dérivée du second ordre est très sensible aux bruits, il faut donc fortement filtrer avant de la calculer.

Contours Actifs

Le modèle de contour actif est une technique itérative dont le principe est de placer une courbe (fermée ou non), puis de la faire évoluer selon certaines contraintes vers les bords de l'objet que l'on aimerait segmenter. L'évolution de la courbe est liée aux contraintes géométriques de l'objet à segmenter et de certaines données internes de l'image.

Le modèle classique des contours actifs ou « snakes » a été introduit en 1988 par Kass, Witkins et Terzopoulos [KWT88]. Il s'agit d'une méthode semi-automatique où nous

plaçons manuellement une courbe v dans l'image au voisinage de l'entité à segmenter. Cette courbe v est paramétrée par une variable s et s'écrit sous la forme :

$$\begin{cases} [0, 1] & \rightarrow \mathbb{R}^2 \\ s & \mapsto v(s) = (x(s), y(s)) \end{cases} \quad (2.5)$$

L'étape suivante de la méthode est de faire évoluer la courbe v sous la contrainte de plusieurs énergies. Le but final étant de minimiser cette énergie en gardant certaines contraintes géométriques correctes. La fonction de calcul de l'énergie est de la forme :

$$E_{totale}(v) = \int_0^1 E_{int}(v(s)) + \int_0^1 E_{img}(v(s)) + \int_0^1 E_{ext}(v(s)) ds \quad (2.6)$$

La première énergie importante E_{img} est calculée à partir d'informations de l'image comme par exemple l'intensité. Dans la majeure partie des applications, l'information de gradient est utilisée dans le but d'attirer la courbe vers les zones où celui-ci est fort (i.e. contours d'objets). La seconde énergie E_{ext} représente une énergie définie par l'utilisateur de l'algorithme et peut être différente selon les cas d'utilisation. Nous pouvons par exemple pré-définir une forme que nous voulons que la courbe atteigne si nous avons cette information au préalable. La dernière énergie E_{int} permet de régler la rigidité de la courbe, c'est à dire la possibilité de la courbe de se déformer pour épouser un contour. Techniquement cela permet d'éviter de rester bloqué dans un minimum local, ce qui est récurrent dans les problèmes de minimisation d'une fonction.

Nous allons utiliser ici deux méthodes différentes de la famille des contours actifs. La première est proposée par Luis Alvarez et al. [[ÁBHMN10](#)] [[ABMNH12](#)]. Cette méthode propose d'utiliser des opérateurs morphologiques en lieu et place des opérateurs différentiels utilisés dans le modèle standard des « snakes » utilisant la notion d'Équation aux Dérivées Partielles (EDP) pour l'évolution de la courbe. Ces méthodes utilisent la notion d'ensemble de niveaux (Level-Sets), introduite par Sher et Sethian [[SS88](#)], qui permet de représenter la courbe de manière implicite contrairement au modèle classique qui utilise une définition paramétrique (i.e. explicite) de la courbe. Cela permet par exemple de gérer les changements topologiques de la courbe, c'est-à-dire qu'il est possible de séparer ou fusionner une courbe. La seconde méthode est une proposition de Pascal Getreuer [[Get12](#)] sur une implémentation du modèle de Chan-Vese [[CV01](#)] à l'aide de la notion d'ensemble

de niveaux (Level-Sets) ainsi que sa résolution en utilisant une descente de gradient semi-implicite. Le modèle de Chan-Vese part des travaux initiés par Mumford et Shah [MS89] qui proposent d'approximer une image par une fonction lisse par morceaux comme solution du problème de minimisation. Le modèle de Chan-Vese est un cas particulier qui n'utilise pas l'information gradient.

2.1.2 Approches « régions »

Dans cette partie, nous allons maintenant nous intéresser à la seconde grande catégorie de segmentation, c'est-à-dire les méthodes utilisant une approche basée « régions ». Pour cela, nous avons considéré trois méthodes différentes qui sont la ligne de partage des eaux (watershed), les algorithmes « superpixel » et l'arbre des composantes connexes de l'image (Component-Tree).

Ligne de partage des eaux (LPE)

Cette méthode a été introduite par Beucher et Lantuejoul en 1979 [BL79]. Elle est issue de la morphologie mathématique et le principe est de considérer une image en niveaux de gris comme une carte topographique où l'intensité serait, par analogie, vue comme l'altitude des reliefs. Il faut s'imaginer que nous plongeons doucement et progressivement notre image en « relief » dans un bassin d'eau. Cela implique que les bassins de faible intensité seront remplis en premier (minimums locaux de l'image) et qu'au fur et à mesure de la plongée certains bassins vont se rejoindre. C'est justement au moment où deux bassins vont fusionner que nous obtenons une ligne de partage des eaux.

Malgré son approche simple et intuitive, la méthode laisse beaucoup de liberté sur sa formulation. Il existe donc plusieurs approches pour « simuler » la plongée de l'image dans l'eau (ou la montée des eaux dans l'image). Une première méthode consiste à faire monter l'eau de manière homogène sur toute l'image à l'aide d'un algorithme récursif d'immersion par exemple [VS91]. Une seconde approche consiste à utiliser la notion de distance géodésique [Mey88], c'est-à-dire que l'on considère la plus courte distance que l'eau emprunterait si elle chutait d'une pente (i.e. la zone où la pente est la plus forte). Pour cela, nous utilisons plutôt une image des normes du gradient au lieu de l'image en niveaux de gris. L'utilisation de la norme des gradients implique un grand nombre

de minima, ce qui génère pour chacun d’eux un bassin, produisant donc au final une sur-segmentation. Pour pallier à la sur-segmentation, il existe plusieurs méthodes :

- Effectuer un premier filtre moyenneur pour supprimer les différents minima non-significatifs.
- Utilisation de marqueurs sur l’image (swamping) pour sélectionner les zones que l’on aimerait mettre en évidence. Nous contrôlons ainsi le nombre de minima locaux soit le nombre de régions finales.
- Utilisation d’une première LPE comme marqueur pour ensuite en effectuer une seconde. Nous obtenons un graphe planaire sur lequel il est possible d’effectuer à nouveau une LPE ou toute autre opération (filtrage, swamping, etc.). Ceci donne lieu à un processus itératif dit « segmentation hiérarchique ».

Nous utiliserons pour nos expériences, une version décrite dans [Mey92] et qui est implémenté dans la librairie OpenCV [ope16].

Superpixels

Les algorithmes de « superpixels » génèrent des régions de pixels possédant une signification visuelle commune. Ce nouveau découpage de l’image remplace la structure de pixels simples de l’image, plus précisément les régions deviennent la nouvelle primitive de l’image. Cette méthode peut être une étape préliminaire efficace pour l’application d’autres méthodes comme la segmentation, la localisation d’objet ou encore l’estimation de profondeur [BDTL15]. Il existe différentes approches pour générer les superpixels avec leurs avantages et leurs défauts. Dans [ASS⁺12], les auteurs mettent en avant les propriétés qui sont généralement attendues, quelque soit l’approche choisie :

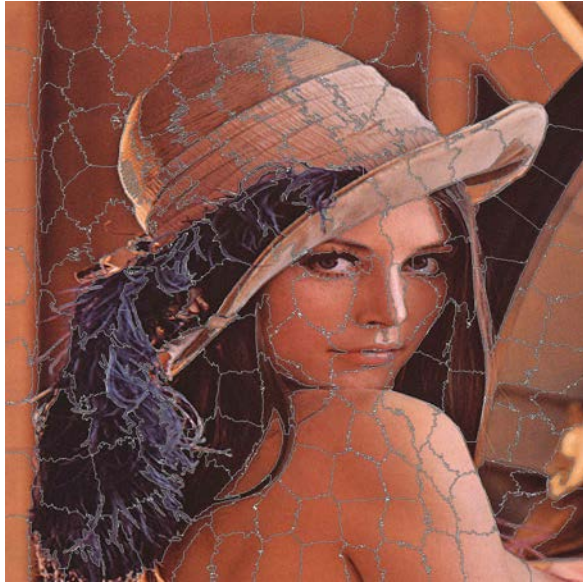
- Les superpixels doivent suivre les différents contours dans l’image.
- Si les superpixels sont une étape de pré-traitement dans le but de réduire la complexité d’un algorithme, alors les superpixels doivent être rapides à générer, avoir une empreinte mémoire faible et être simples d’utilisation.
- Si les superpixels sont utilisés dans une optique de segmentation de l’image, alors les superpixels doivent améliorer la rapidité d’exécution ainsi que les résultats de l’algorithme de segmentation.

Les deux principales familles de méthodes pour générer des superpixels sont les ap-

proches à base de graphe comme dans [SM00] et celles par descente de gradient comme dans [CM02]. La première consiste à partitionner un graphe de tous les pixels de l'image en utilisant la minimisation d'une fonction de coût définie sur les contours au niveau des limites des partitions. La seconde localise des maxima locaux d'une fonction de densité pour déterminer des caractéristiques d'intensité. Les pixels qui convergent vers les mêmes caractéristiques définissent les superpixels. Pour notre étude, nous utiliserons l'algorithme de superpixels proposé par Achanta et al. [ASS⁺12]. Ils proposent dans ce papier une méthode différente pour générer les superpixels, plus rapide, avec une empreinte mémoire plus faible et améliorant la performance des algorithmes de segmentation. La méthode Simple Linear Iterative Clustering (SLIC) est une adaptation du partitionnement en k-moyennes (k-means) pour la génération des superpixels avec deux importantes notions. La k-moyenne standard calcule la distance moyenne entre un pixel et tous les pixels de l'image. Ici, l'espace de recherche autour d'un pixel est limité à une région proportionnelle à la taille d'un superpixel. La seconde notion est la mesure de distance qui combine à la fois une proximité spatiale mais aussi colorimétrique. Cela permet d'obtenir un contrôle simultané de la taille et de la compacité des superpixels. Nous pouvons visualiser sur la figure 2.4 l'influence des paramètres de taille (i.e. ici le nombre total de superpixel que l'on désire dans l'image) et de compacité (i.e. permet de régler plus ou moins l'homogénéité des superpixels d'un point de vue colorimétrie).

Arbre de composantes connexes (Component-Trees)

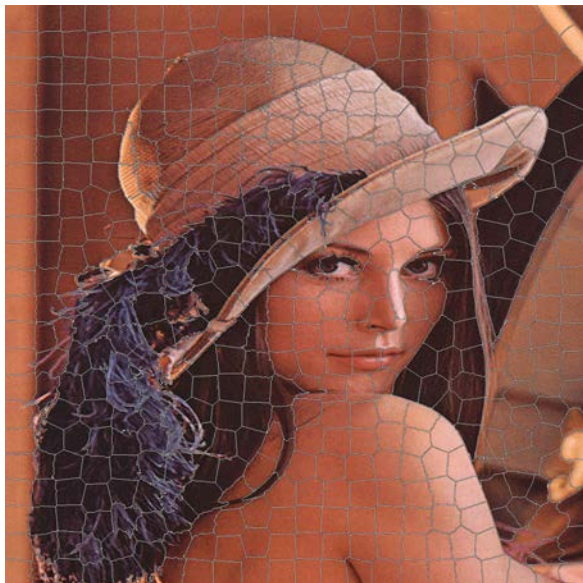
Tout comme les Superpixels, nous pouvons définir les arbres de composantes connexes (Component-Tree) comme une structure facilitant ou accélérant certaines méthodes de traitement d'image. Dans cette représentation de l'image (voir figure 2.5), un nœud de l'arbre matérialise une composante connexe à qui l'on peut associer un ou plusieurs attributs comme la surface de la composante ou encore un descripteur de forme associé à la composante. Les relations d'héritage entre les nœuds reflètent une relation d'inclusion spatiale entre les différentes composantes. Cette structure de données est utilisée dans de nombreuses applications comme la classification, le filtrage d'images, la compression ou encore la segmentation. C'est cette dernière application que nous allons étudier au travers d'une proposition faite par Benoît Naegel et Nicolas Passat dans [PNR⁺11].



(a) Taille : 200 ; Compacité : 10



(b) Taille : 200 ; Compacité : 40

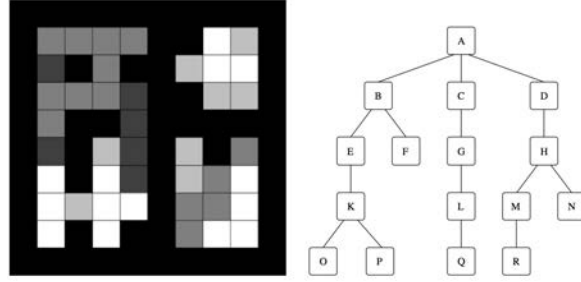


(c) Taille : 500 ; Compacité : 30



(d) Taille : 50 ; Compacité : 30

FIGURE 2.4 – Illustration des superpixels avec différents paramètres [ASS+12].

FIGURE 2.5 – Une image et son arbre de composantes connexes [PNR⁺11].

Dans cet article, les auteurs proposent un algorithme interactif de segmentation basé sur un arbre de composantes connexes. La création de l'arbre est effectuée par l'algorithme de Salembier [SOG98] qui est basé sur un remplissage par diffusion dans l'image ou l'algorithme de Najman [NC06] qui utilise le paradigme « Union-Find ». L'algorithme consiste à extraire un sous-ensemble de l'arbre de composantes connexes qui correspond à un objet binaire qui se « rapproche le plus » d'un autre objet binaire défini manuellement sur l'image (marqueur). En considérant une pseudo-distance d^α , le problème est ramené à un problème de minimisation, défini par :

$$\hat{\mathcal{K}} = \arg \min_{\mathcal{K}' \in \mathcal{P}(\mathcal{K})} \{d^\alpha(\bigcup_{N \in \mathcal{K}'} N, G)\} \quad (2.7)$$

où G correspond au marqueur (une image binaire), N un nœud appartenant à $\mathcal{K}'$ qui est une partie de l'arbre K représentant toutes les composantes connexes de l'image. Le problème est résolu à l'aide d'une méthode de programmation dynamique. La pseudo-distance utilisée lors de ce processus est définie ainsi :

$$d^\alpha(X, Y) = \alpha \cdot |X \setminus Y| + (1 - \alpha) \cdot |Y \setminus X| \quad (2.8)$$

où X et Y sont respectivement deux ensembles de pixels qui représentent un sous-ensemble de l'arbre des composantes connexes $\bigcup_{N \in \mathcal{K}'} N$ et le marqueur G . La valeur $\alpha \in [0, 1]$ est un paramètre correspondant à la distance que nous souhaitons entre les deux objets binaires et qui est définie manuellement. Par exemple, la distance d^0 permet de s'éloigner du marqueur alors que la distance d^1 va tendre vers le marqueur et ainsi éviter les sur-segmentation. La suppression de ce paramètre a été proposée par Kowalczyk et al. [KKNW12] à l'aide de la notion d'échelle significative [KL12] [KL14] qui permet de détecter la quantité de bruit présent le long d'un contour discret.

2.1.3 Expérimentations en milieu industriel

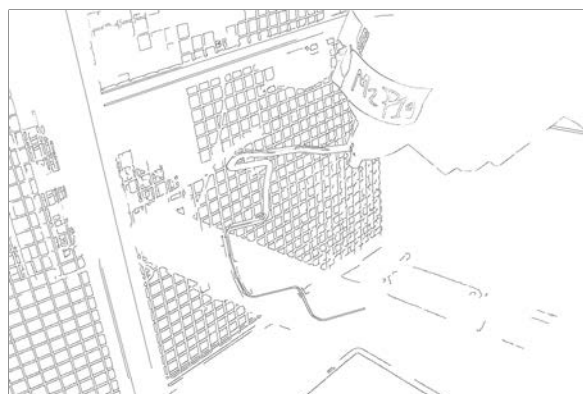
A l'aide des images de référence (figures 2.1 et 2.2), nous allons étudier ici le comportement des différentes méthodes évoquées précédemment. Cette étude va nous permettre d'observer si ces méthodes sont capables d'effectuer une segmentation suffisante d'un objet tubulaire métallique dans un environnement industriel non contrôlé. Une segmentation suffisante se définit par l'obtention des contours de l'objet et uniquement ceux-ci, ou par le fait que ces contours soient facilement distinguables du « bruit » engendré par la segmentation. Pour chacune des méthodes, nous garderons uniquement les résultats (avec leurs paramètres si il y en a) qui visuellement seraient les plus intéressants pour notre cas de figure. Nous considérons uniquement l'aspect visuel car les différentes mesures classiques que l'on pourrait effectuer, ne reflètent pas la qualité géométriques de l'objet (i.e. mesure de dissimilarité, mesure par rapport à un modèle segmenté, etc.). De plus, l'objectif ici n'est pas de proposer une nouvelle méthode de segmentation. Les différentes expériences ont été réalisées pour la plupart à l'aide des démos que les auteurs proposent, soit directement sur leur site personnel ou sur la plate-forme *IPO*L [IPO16]. Pour les autres cas, c'est la librairie C++ *OpenCV* [ope16] qui a été utilisée.

Dans un premier temps, si nous regardons les résultats obtenus avec Canny (figure 2.6) et Marr-Hildreth (figure 2.7) [MH80] [SC15], nous pouvons constater que la détection sur un fond vert uni (figure 2.6 (e) et 2.7 (e)) donne de relativement bons résultats sur la détection des bords du tube, malgré quelques discontinuités et détections internes dues aux spécularités. La conclusion est par contre plus mitigée lorsque la détection s'effectue dans un milieu industriel non contrôlé. En effet, nous pouvons observer, par exemple sur les images (a - d) des figures 2.6 et 2.7 que les discontinuités sont bien plus présentes, ainsi que la détection des spécularités du tube, au détriment des « réels » bords du tube. Cela s'explique par le fait que parfois le tube peut avoir une intensité proche du fond, ce qui ne produit pas de délimitation franche entre le tube et le fond.

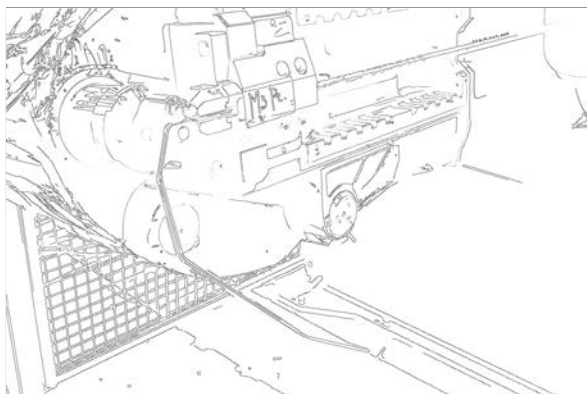
En ce qui concerne les méthodes dites « à contours actifs », nous obtenons pour l'algorithme « Morphological Snakes [ABMNH12] » avec un paramètre de seuil de gradient de 10 et 50 itérations, des résultats proches de nos attentes sur les figures 2.8 (d) et (e). Cela n'est plus le cas pour les autres figures bien que l'initialisation (en bleu sur l'image) soit relativement proche du tube. Nous constatons, en effet, que l'attraction dans la direction



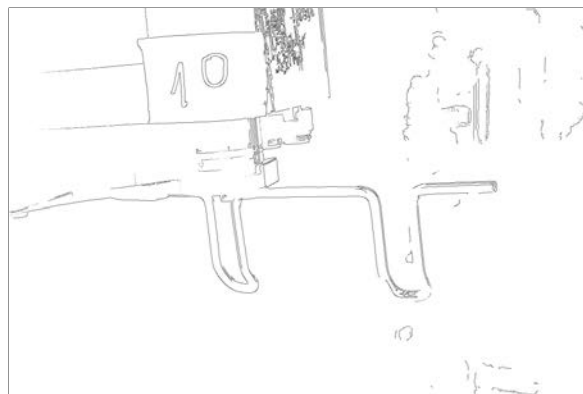
(a)



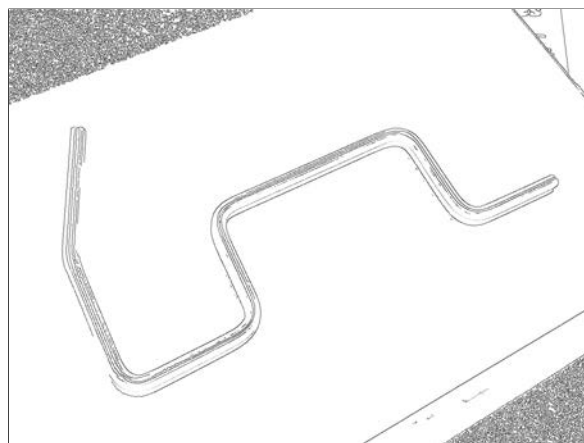
(b)



(c)



(d)



(e)

FIGURE 2.6 – Résultats obtenus avec le détecteur de contours Canny avec une taille de noyau de 3, un seuil bas de 20 et haut de 60.

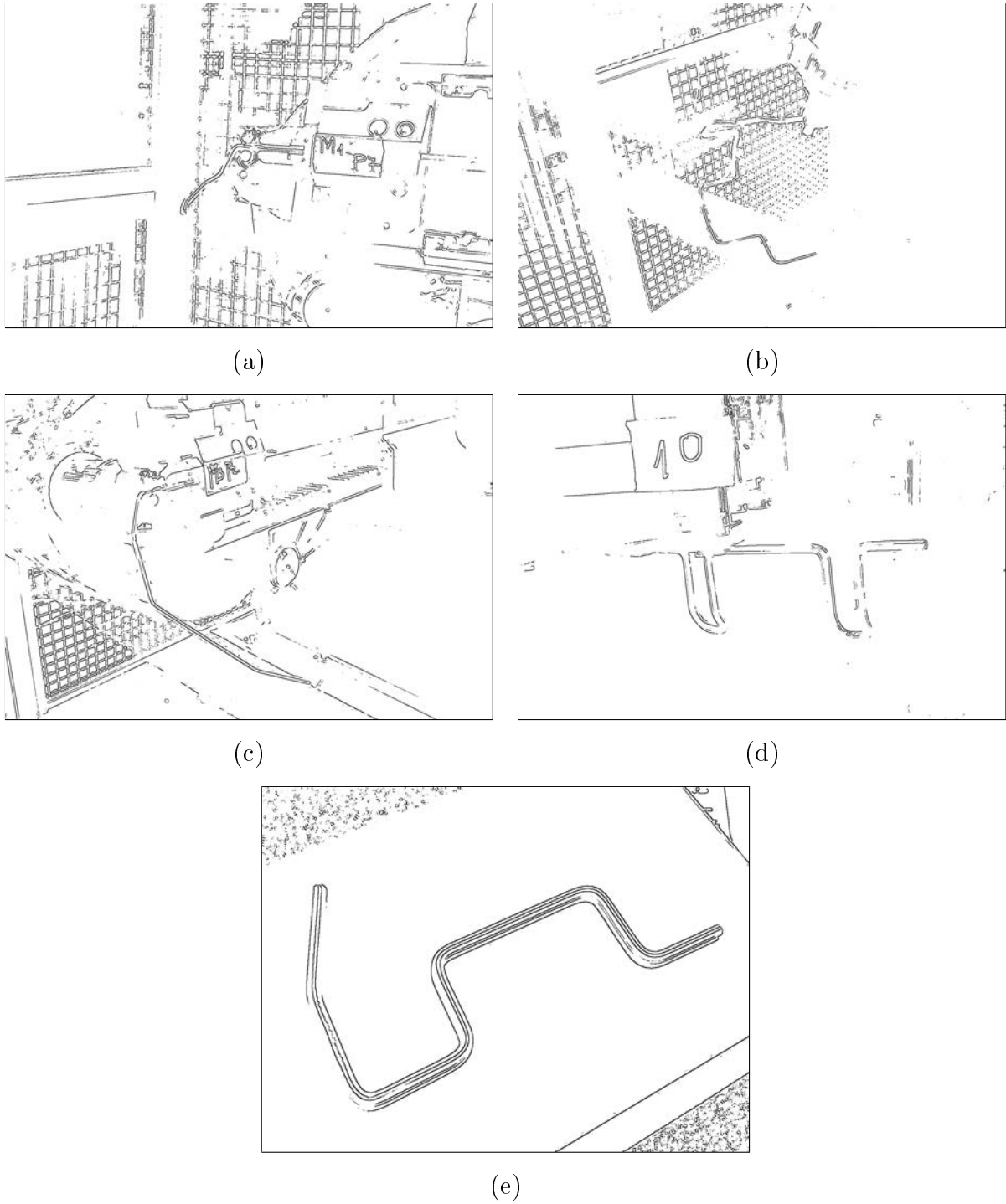
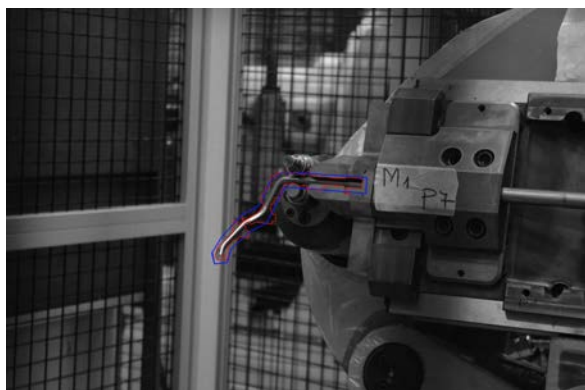
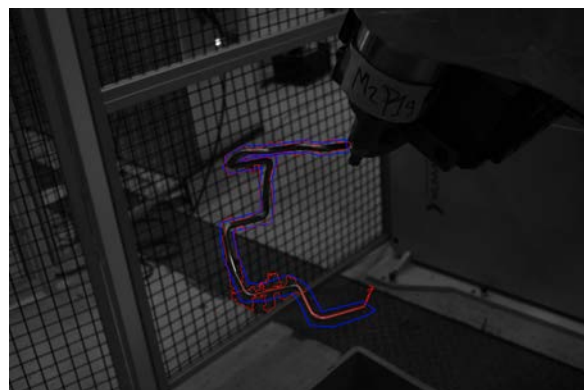


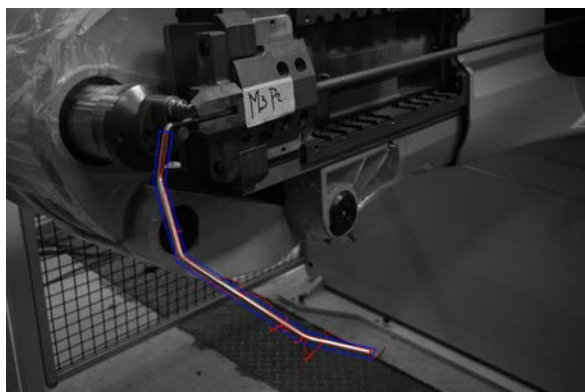
FIGURE 2.7 – Résultats obtenus avec l'algorithme de détection de contours Marr-Hildreth [IPO16] avec une déviation standard $\sigma = 3$, une taille de noyau de 2 et un seuil de passage par 0 de $tcz = 0,07$.



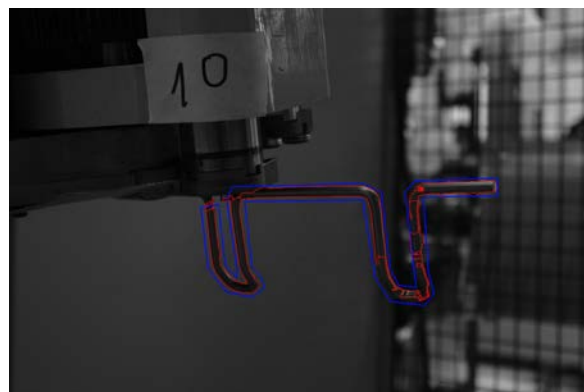
(a)



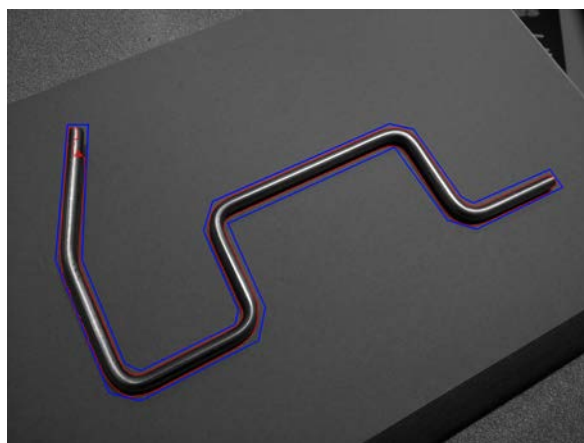
(b)



(c)



(d)



(e)

FIGURE 2.8 – Résultats obtenus avec l'algorithme des snakes avec un paramètre de seuil de gradient de 10.

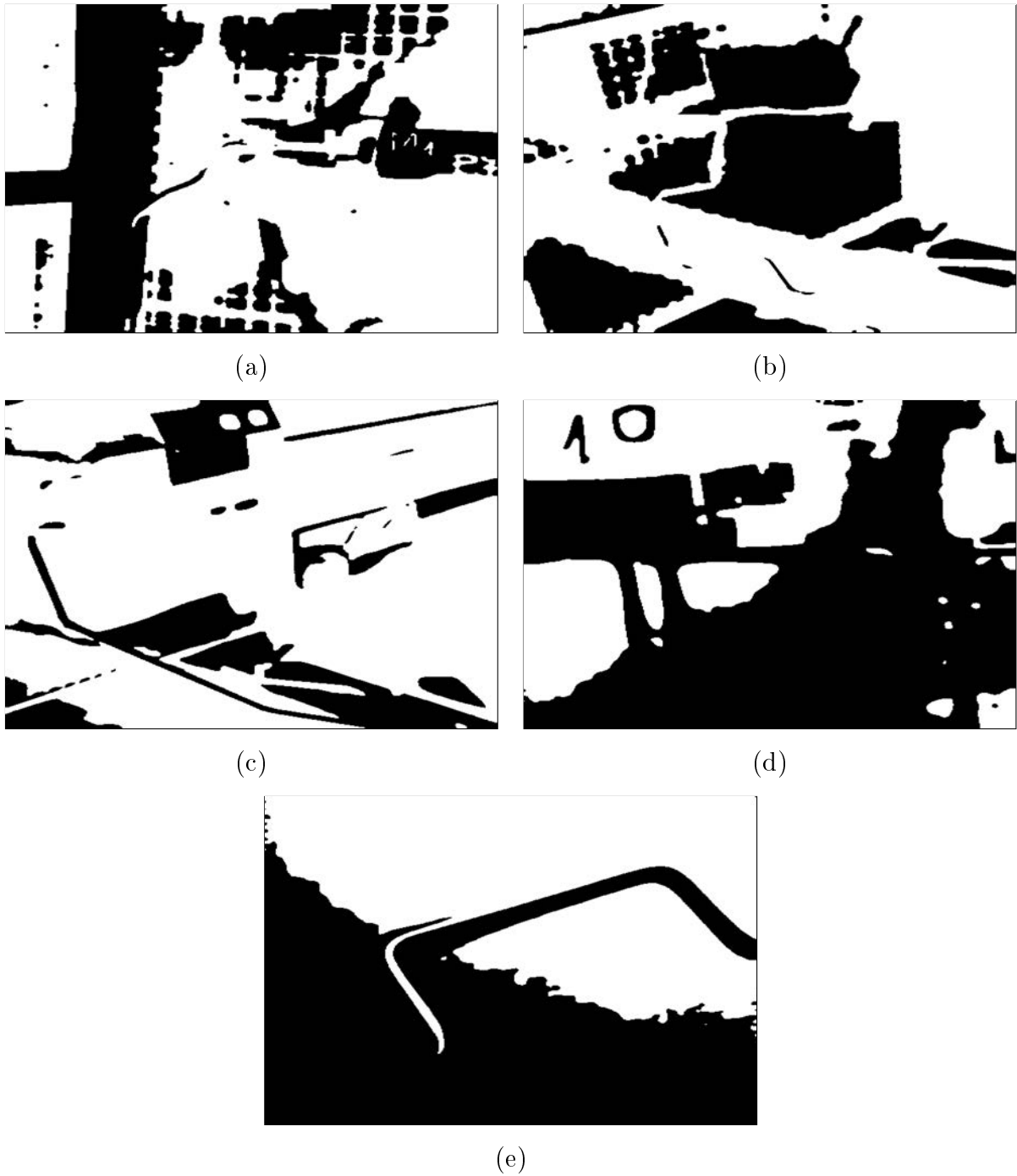


FIGURE 2.9 – Résultats obtenus avec la segmentation de Chan-Vese [IPO16] avec le paramètre $\mu = 0.2$.

du tube ne s'effectue pas suffisant pour coller aux contours de celui-ci. Pour l'algorithme de Chan-Vese [Get12], les images binaires (figure 2.9) qu'il produit ne permettent pas, même visuellement, d'extraire le tube ou alors que très partiellement.

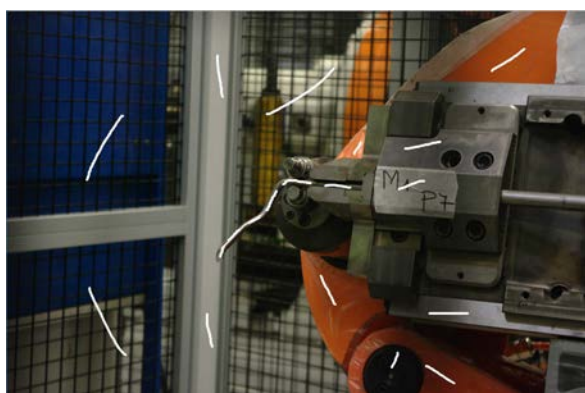
Les résultats obtenus par des algorithmes qui permettent d'effectuer un partitionnement différent de l'image comme le « Watershed » (figure 2.10 et 2.11) ou les « Superpixels — SLIC » (figure 2.12) ne sont pas assez précis car il y a des partitions du fond qui possèdent parfois de grandes parties du tube. Cela empêche d'effectuer une sélection fine des partitions pour obtenir une segmentation unique du tube. Nous obtenons ce type de résultat alors que nous essayons, par exemple pour le Watershed, d'identifier le plus possible d'éléments grâce à des marqueurs (voir images (a-c-e-g) sur figure 2.10 et 2.11) dont plusieurs sur le tube lui-même. Mais cela ne permet pas malgré tout d'obtenir une segmentation du tube exploitable.

Pour terminer, les résultats (figure 2.13 et 2.14) obtenus à l'aide des arbres de composantes connexes permettent, à l'aide de marqueurs sur le tube, d'éliminer vraiment efficacement la totalité de l'environnement autour de la pièce mais le résultat final n'est pas complet. En effet, comme nous pouvons le voir sur la figure 2.13 (d), il y a beaucoup de « trous » dans la segmentation du tube et la détection se focalise très fortement sur les spécularités du tube plutôt que sur ses bords. A noter que le choix du paramètre α s'est fait sur un critère uniquement visuel (i.e. celui qui permet d'obtenir visuellement l'objet et uniquement celui-ci).

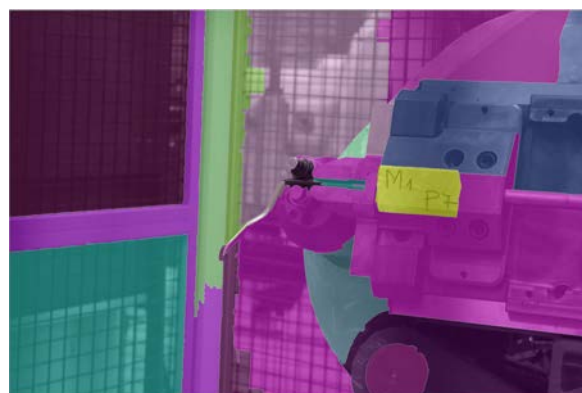
2.2 Détection de segments ou droites

2.2.1 Méthodes choisies

Alors que la segmentation permettait, selon l'approche choisie, de mettre en exergue les pixels représentant les contours d'un objet ou l'objet lui-même, nous allons porter notre intérêt sur des méthodes qui s'intéressent plutôt à la détection de primitives géométriques. Ce type de méthode est très souvent une étape préliminaire pour effectuer de la reconnaissance de formes. Dans notre étude d'algorithmes permettant d'extraire un objet tubulaire métallique de son environnement industriel, nous pouvons réduire cela à la détection de segments, plus précisément à la détection de couple de segments paral-



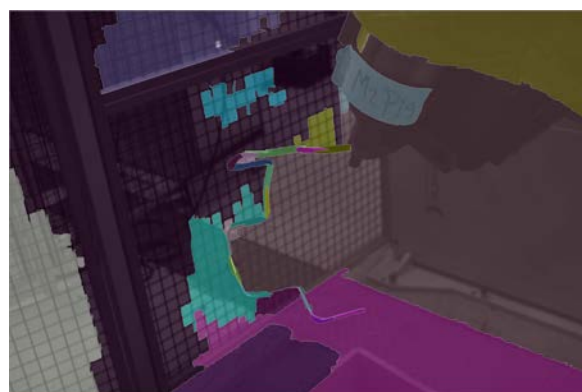
(a)



(b)



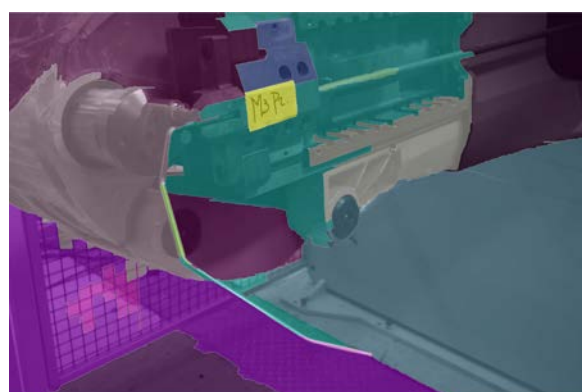
(c)



(d)

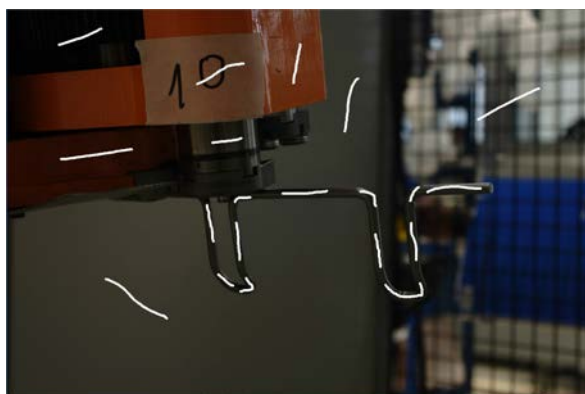


(e)



(f)

FIGURE 2.10 – Résultats obtenus avec l'algorithme de Watershed (Partie 1).



(g)



(h)

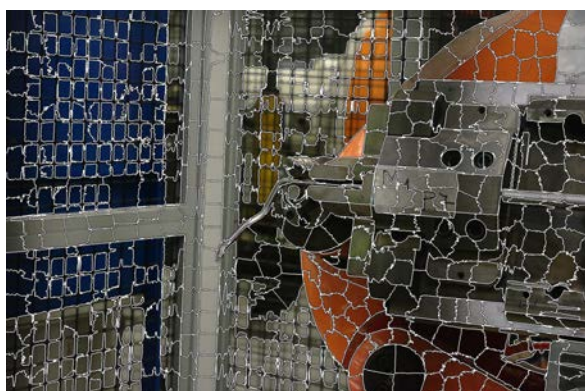


(i)

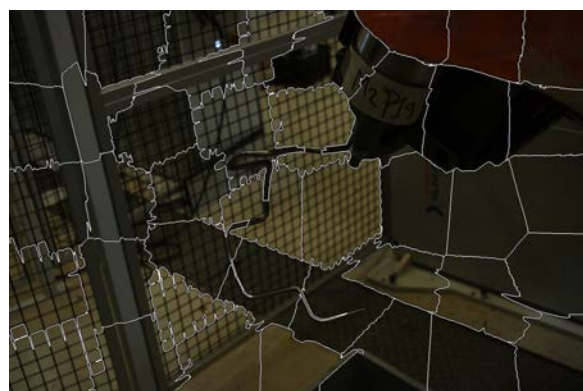


(j)

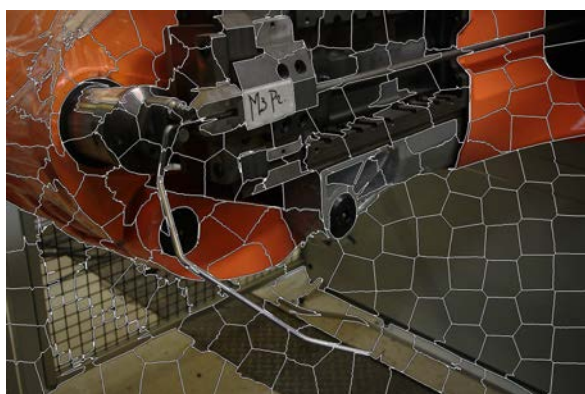
FIGURE 2.11 – Résultats obtenus avec l'algorithme de Watershed (Partie 2).



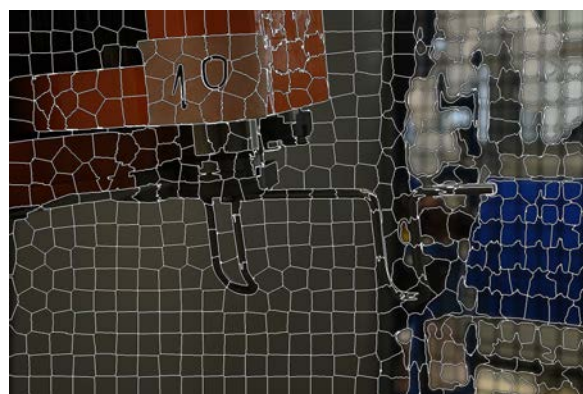
(a) Taille : 500 ; Compacité : 10



(b) Taille : 50 ; Compacité : 20



(c) Taille : 200 ; Compacité : 20



(d) Taille : 500 ; Compacité : 10



(e) Taille : 50 ; Compacité : 10

FIGURE 2.12 – Résultats obtenus avec l'algorithme des Superpixels SLIC.

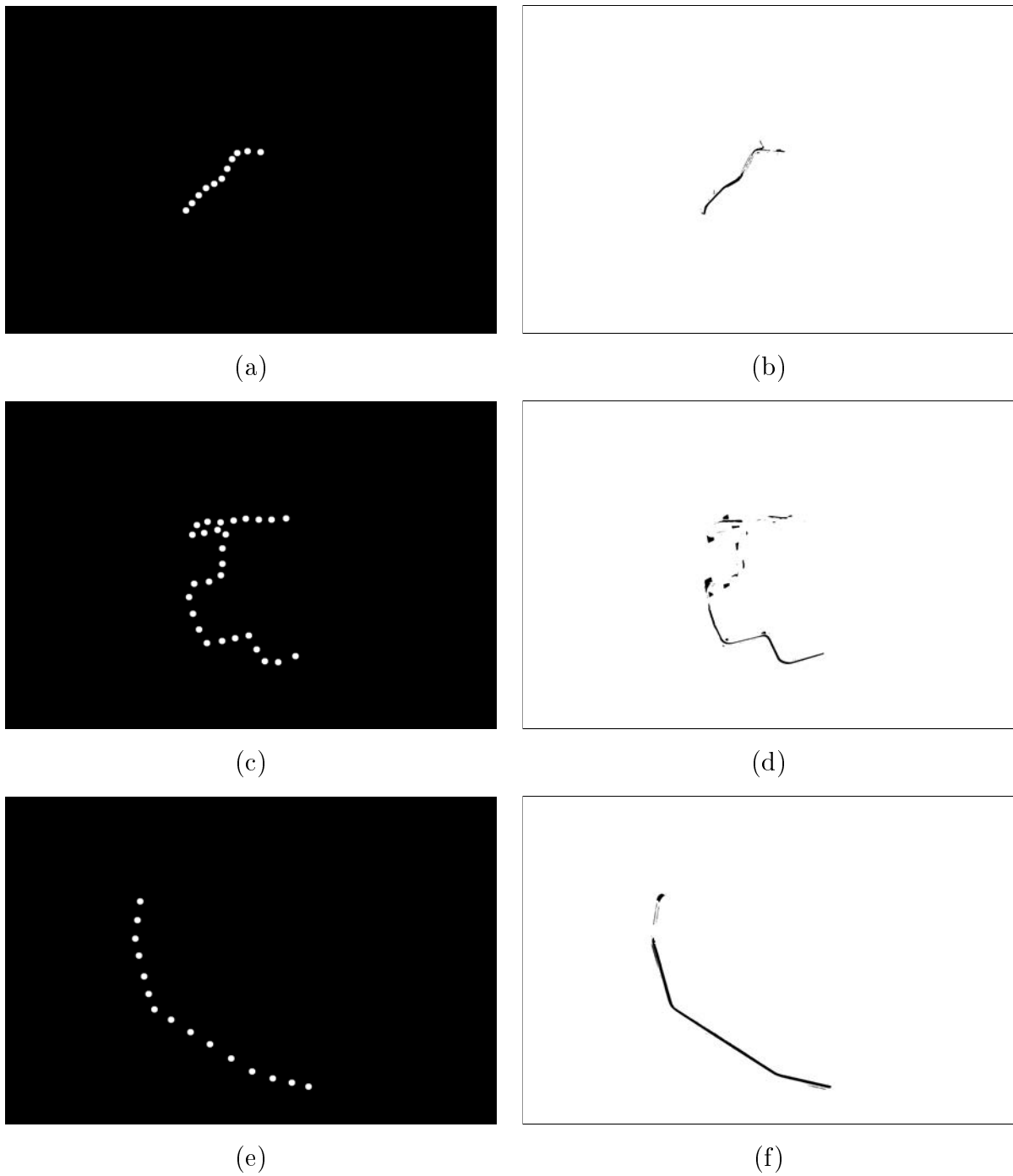
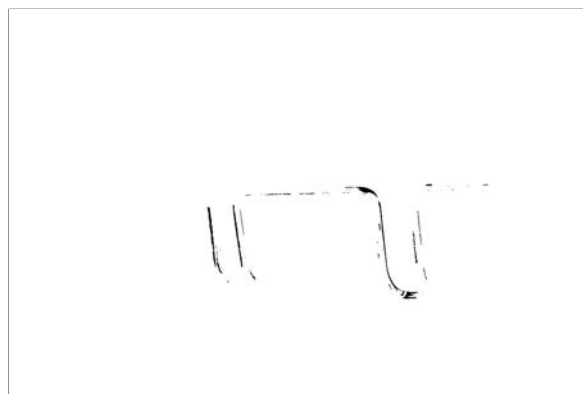


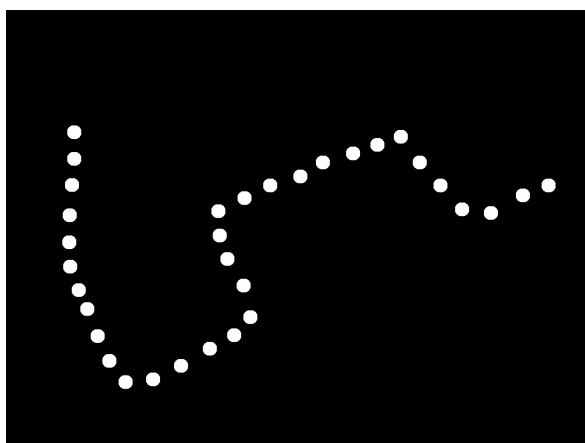
FIGURE 2.13 – Résultats obtenus à l'aide des arbres de composantes connexes avec un paramètre $\alpha = 0.1$ (Partie 1).



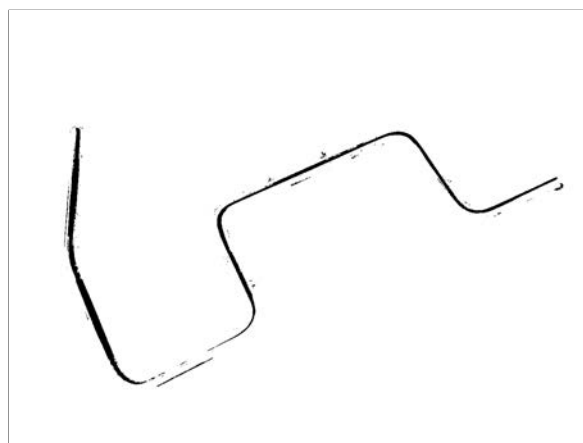
(g)



(h)



(i)



(j)

FIGURE 2.14 – Résultats obtenus à l'aide des arbres de composantes connexes (Partie 2).

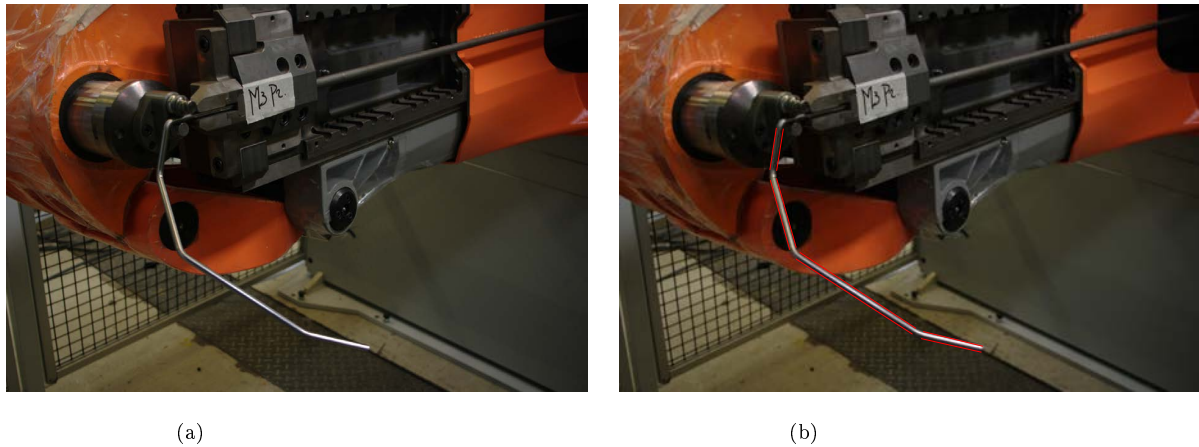


FIGURE 2.15 – Résultat idéal de la détection des couples de segments parallèles représentant les zones droites de l’objet tubulaire métallique.

lèles (figure 2.15). Pour cela, nous allons passer en revue de manière non exhaustive des algorithmes de détection de droites comme la transformée de Hough ou encore LSD (Line Segment Detector).

Historiquement, la première méthode bien connue permettant de mettre en évidence dans une image la structure de droites (ou segments) est la transformée de Hough. Cette méthode a été publiée par Paul Hough en 1959 [Hou59] et fait l’objet d’un brevet (IBM) en 1962 [Hou62]. Dans ce papier, Hough propose une méthode permettant de détecter des alignements au sein d’un ensemble de points dans une image binaire. La transformée de Hough (TH) a été généralisée par Duda et al. [DH72] pour permettre de détecter des primitives comme les droites, les cercles ou encore les ellipses selon la paramétrisation du problème (cf. Analyse pilotée par modèle). La primitive que nous voulons détecter est caractérisée par m paramètres et une instance peut s’exprimer par m éléments. Par exemple, une droite dépend de deux paramètres et passe par deux points ou encore un cercle dépend de trois paramètres et passe par trois points. La philosophie de la méthode est de transformer un problème de reconnaissance « paramétré » en une recherche d’extrema dans l’espace des paramètres.

Soit une droite (figure 2.16) définie en coordonnées polaires par cette équation :

$$\rho = x \cos \theta + y \sin \theta \quad (2.9)$$

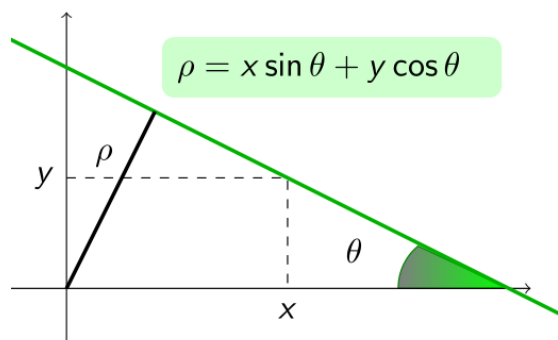


FIGURE 2.16 – Définition d’une droite en coordonnées polaires pour la transformée de Hough.

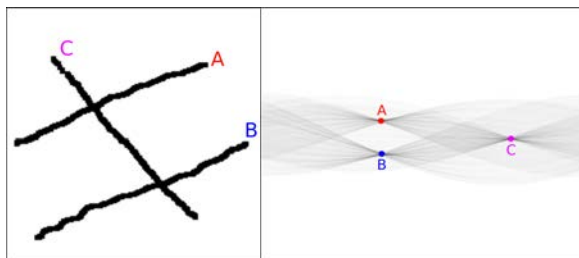


FIGURE 2.17 – (à gauche) Trois segments avec (à droite) leur correspondance dans l’espace de Hough.

La transformée de Hough propose de remplir une grille d’accumulation dont les axes sont les paramètres de la droite (ρ, θ) . Au final, cette grille possède pour une droite (ρ_n, θ_n) le nombre de pixels appartenant à l’objet² faisant partie de cette droite. Une case (ρ_n, θ_n) de la grille d’accumulation qui possède un gros score correspond, dans l’espace de l’image, à une droite de paramètres (ρ_n, θ_n) (figure 2.17).

Il existe de nombreuses implémentations de la transformée de Hough. Pour notre étude, nous nous sommes tournés vers la librairie C++ OpenCV [[ope16](#)] qui propose une implémentation de la transformée de Hough Standard ainsi qu’une version nommée « progressive probabilistic Hough transform » proposée par Matas et al. [[MGK00](#)]. Nous testerons également une méthode semi-automatique, utilisant la transformée de Hough, proposée par Even et al. [[EM00](#)]. Cette méthode consiste à définir manuellement un seg-

². rappel : nous travaillons sur une image binaire.

ment qui, à partir de deux points et d'une transformée de Hough, va être attirée vers un contour.

Bien que la transformée de Hough soit une méthode relativement ancienne, elle est encore très utilisée comme base pour de nouvelles méthodes ou à travers ses variantes car ses résultats applicatifs restent très intéressants. Malgré cela, la recherche dans le domaine de la détection de droites reste très active et de nouvelles propositions sans lien avec la transformée de Hough sont faites. De récents travaux se sont intéressés à l'un des problèmes majeurs de la plupart des algorithmes de détection de droites. Ce problème est le contrôle du nombre de fausses détections et plus particulièrement comment établir des seuils permettant d'inférer sur le statut d'une détection : bonne ou mauvaise détection ?

Désolneux et al. proposent une formalisation des concepts de la psychologie des formes et plus particulièrement des lois de Gestalt qui sont les suivantes [DMM07] :

- La loi de la symétrie : un ensemble de points possédant une symétrie quelconque sera plus facilement perçu comme une forme globale.
- La loi de clôture : lorsqu'une image est incomplète, notre cerveau a tendance à combler les trous pour identifier des formes.
- La loi de la proximité : nous regroupons en priorité les points proches les uns des autres.
- La loi de similitude : nous regroupons les points les plus similaires entre eux pour percevoir une forme dans le cas où la distance entre les points ne le permette pas.
- La loi de destin commun : des objets en mouvement ayant la même trajectoire seront considérés comme faisant partie de la même entité.
- La loi de familiarité : plus une entité est familière plus elle est significative.

Ces différentes lois permettent de décrire ce qui est observable dans une image. Elles sont couplées au principe de Helmholtz qui stipule qu'une structure perceptuelle est d'autant plus visible qu'elle est non-générique. Ces notions forment ce que l'on appelle l'approche *a contrario* et c'est donc à partir de cette approche que Désolneux et al. [DMM07] propose une méthode efficace de validation des candidats sélectionnés³.

Une méthode de validation permet de dire si un alignement dans une image est valide ou pas. L'approche *a contrario* va permettre cela en formalisant mathématiquement le

3. La plupart des algorithmes de détection comportent deux phases : sélection des candidats et validation de ceux-ci

principe de Helmholtz qui dit, si l'on se place dans le domaine de la détection de droites, que tout alignement existant dans une image de bruit blanc n'est pas perceptible par notre système visuel. Par conséquent, si l'on détecte dans notre image un alignement qui *pourrait* être observé dans une image de bruit blanc, celui-ci ne devrait en aucun cas être évalué comme un alignement valide car il est dû au hasard. On désigne une image de bruit blanc comme une image dont les pixels ont une valeur de niveaux de gris qui suivent une loi normale et que l'on nomme *hypothèse a contrario* H_0 .

En considérant la direction du gradient pour chaque pixel de l'image que l'on analyse, nous considérons que deux pixels p_1 et p_2 d'un même segment discret sont alignés si la direction de leurs gradients, respectivement g_1 et g_2 , concorde avec une certaine précision $p \in [0, 1]$. On nomme un alignement $k(l, n, p)$ tout segment k de longueur l en pixels contenant un nombre de pixels n alignés avec une précision p . En considérant la direction en chaque pixel, à l'aide de l'hypothèse H_0 comme une variable aléatoire indépendante et uniformément distribuée, on définit la probabilité qu'un alignement $k(l, n, p)$ soit dû au hasard par une loi binomiale :

$$B(l, n, p) = \sum_{i=n}^l \binom{l}{i} p^i (1-p)^{l-i} \quad (2.10)$$

On désigne un alignement $k(l, n, p)$ comme ϵ -significatif lorsque sa probabilité d'être dû au hasard vérifie :

$$B(l, n, p) \leq \frac{\epsilon}{\mathcal{N}_t} \quad (2.11)$$

avec $\mathcal{N}_t$ qui représente le nombre total de tests⁴ que l'on peut effectuer dans l'image.

À partir de l'équation 2.10, nous pouvons définir ce l'on nomme dans la littérature, le nombre de fausses alarmes d'un alignement k :

$$NFA(k) = \mathcal{N}_t \cdot B(l, n, p) \quad (2.12)$$

Cette définition implique qu'une séquence est ϵ -significative si $NFA(k) \leq \epsilon$ et nous pouvons donc déterminer si un alignement est valide ou non⁵.

4. alignement dû au hasard

5. dû au hasard

Dans [vGJMR12], Grompone von Gioi et al. proposent un algorithme dénommé *Line Segment Detector (LSD)* utilisant une validation « a contrario » comme décrite précédemment et un système de sélection des candidats énoncé par Burns et al. [BHR86]. L'idée pour la sélection est dans un premier d'effectuer un regroupement de pixels 8-connexe possédant la même orientation à un angle de tolérance τ près. L'orientation de chaque pixel correspond à la direction orthogonale au gradient (Level-Line). Pour limiter le nombre de candidats, le regroupement commence par les pixels qui possèdent un gradient élevé car ils sont susceptibles de faire partie d'un contour, donc d'un segment. Dans un second temps, une approximation de ces groupes est effectuée. Chaque regroupement est contenu dans un rectangle dont le centre est le barycentre de tous les pixels du groupe calculé par pondération de la norme du gradient de ceux-ci.

La validation d'un rectangle s'effectue à partir de la formulation suivante :

$$NFA(k) = \mathcal{N}^5 \cdot B(l, n, p) \leq 1 \quad (2.13)$$

avec l représentant le nombre de pixels total dans le rectangle et n le nombre de pixels alignés dans la direction du rectangle avec une précision p . Le nombre de test⁶ a été choisi par rapport au nombre de degrés de liberté qu'un rectangle possède.

Dans [AT11], Akinlar et Topal mettent en avant les bons résultats obtenus par l'algorithme LSD mais pointent le problème de temps d'exécution. Ils proposent, pour rendre la détection « temps-réel », une méthode nommée « EDlines » utilisant aussi une étape de validation à l'aide du principe de Helmholtz pour éliminer les détections de segments du type faux positifs. Leur contribution réside dans l'utilisation de leur précédente contribution [TAG10] en lieu et place de la création d'une carte de regroupement de pixels ayant la même orientation. Cette contribution est un nouvel algorithme de détection de contours nommé « Edge Drawing (ED) ».

La méthode « Edge Drawing (ED) » se découpe en plusieurs étapes :

1. Utilisation d'un filtre de Gauss de taille 5 avec un $\sigma = 1$ pour supprimer le bruit et lisser l'image.
2. Calcul des gradients (norme et direction) à l'aide d'un opérateur quelconque tel que Prewitt, Sobel, Scharr, etc.

6. $\mathcal{N}_t \sim \mathcal{N}^5$

3. Calcul d'un ensemble de pixels appelés « ancrs (anchors) » qui correspondent aux pixels ayant les valeurs maximales de gradient (i.e. les valeurs pics de la carte des gradients). Ils sont donc des candidats, avec une forte probabilité, pour faire partie d'une droite.
4. Connexion des différentes « ancrs » en dessinant un segment entre chacun. Nous relierons uniquement les « ancrs » ayant un gradient pointant dans la même direction.

Cette méthode suit le processus énoncé par Canny mais diffère au niveau des deux derniers points. Ces différentes étapes vont produire un ensemble de segments sans bruit et connectés (i.e. 8-connexes). L'étape suivante de l'algorithme «EDlines» consiste à parcourir chacun des segments précédemment générés pixel par pixel en essayant de les adapter à une droite à l'aide d'une méthode des moindres carrés et cela tant que l'erreur ne dépasse pas un certain seuil (e.g. un pixel d'erreur). Lorsque le seuil est dépassé, nous décidons de stopper le segment en cours et en commençons un nouveau avec le pixel courant (i.e. celui qui a provoqué le dépassement de seuil). Pour terminer, l'ensemble des segments est soumis à l'étape de validation identique à l'algorithme LSD.

Plus récemment, Lu et al. ont proposé une méthode [LYLL15] nommée « CannyLines » utilisant la même philosophie autour du principe de Helmholtz, exploité par les méthodes précédemment évoquées. Cette méthode contient aussi les mêmes étapes qui sont la création d'une carte des contours, la connexion/déconnexion des différents segments puis la validation. Mais les auteurs proposent ici plusieurs améliorations comme :

- l'utilisation d'un nouvel opérateur de détection de contours basé sur Canny nommé *CannyPF*.
- Une méthode de « connexion - déconnexion » des segments inspirée de [FO08].
- Une validation utilisant en plus de l'information de direction, la norme des gradients.

Le nouvel opérateur *CannyPF* utilise le principe de Helmholtz pour calculer ce que les auteurs appellent « la norme minimale du gradient significatif » et « la norme maximale du gradient significatif » qui vont être utilisées comme paramètres de seuil bas et haut dans le filtre de Canny. Ce procédé permet de rendre le filtre de Canny automatique sans ajustement des paramètres selon les images que nous avons à traiter. L'étape suivante part

du principe connexion/déconnexion utilisé dans « kernel-based Hough Transform (KHF) voting scheme » [FO08] et se découpe en plusieurs étapes :

1. Tri : tous les pixels obtenus à partir de l'étape *CannyPF* sont triés par rapport à la norme de leur gradient.
2. Connexion : en partant du pixel ayant la plus grande norme, nous cherchons le pixel voisin⁷ ayant une orientation équivalente à un angle τ près. Lorsqu'il n'y a plus de candidat, nous recommençons l'opération à partir d'un des pixels restant.
3. Déconnexion : tous les segments résultant de l'étape « connexion » ayant une longueur supérieure à seuil θ_s sont découpés en deux au point possédant une déviation maximale d'un pixel de largeur par rapport à la droite formée par les deux points extrémités du segment.
4. Extension : pour chacun des segments construits avec les étapes « connexion - déconnexion », nous tentons d'étendre les segments à partir de chaque extrémité en cherchant si il existe un point de contour relativement proche pour être ajouté au segment et ayant une direction de gradient identique⁸. Si on découvre un point de contour faisant déjà partie d'un autre segment, nous effectuons une fusion si cette fusion a une erreur inférieure à un pixel entre la droite obtenue par la méthode des moindres carrés du segment en cours et la droite du segment étendu.

Pour terminer, cette méthode exécute une étape de validation sur les différents segments construits. Les auteurs proposent une version étendue de la validation que nous retrouvons dans *LSD* ou encore *EDLines*. La fonction de validation est définie par :

$$NFA(\mathcal{L}) = NFA(n, k) \times NFA(S) = N^4 \cdot \sum_{i=k}^n \binom{n}{i} p^i (1-p)^{n-i} \times N_p \times H(u)^n \quad (2.14)$$

avec u la norme minimale du gradient du segment S , H la probabilité pour un point d'avoir une valeur d'intensité supérieure à u et N_p définie par :

$$N_p = \sum_k l_k \times (l_k - 1) / 2 \quad (2.15)$$

représentant le nombre total de parties connectées de tous les segments S_k de longueur l_k .

7. Recherche sur un voisinage 8-connexe.

8. Toujours à un angle τ près.

2.2.2 Expérimentations en milieu industriel

Comme pour la section 2.1.3, avec les mêmes remarques et conditions, nous allons maintenant mettre à l'épreuve les algorithmes de détection de droites sur nos images références.

La première expérience s'effectue à l'aide de méthodes utilisant la transformée de Hough dans un contexte de détection de droites. La première méthode est une transformée de Hough probabiliste. Pour cela, nous utilisons, comme nous l'avons déjà précisé, une implémentation proposée dans la librairie OpenCV [ope16]. L'exemple d'OpenCV propose d'effectuer dans un premier temps un filtrage avec un filtre de Canny pour garder au maximum les pixels susceptibles de faire partie d'une droite, puis d'effectuer la détection des droites avec la transformée de Hough probabiliste. Nous pouvons observer (figure 2.18) qu'en règle générale, en négligeant les détections de droites (les pixels bleus sur la figure 2.18) faisant partie de l'environnement, la détection se fait très partiellement. Nous obtenons soit des droites appartenant effectivement au tube mais incomplètes, soit des droites appartenant à la spécularité du tube. Cela peut s'expliquer par le fait qu'une première étape de filtrage s'effectue à l'aide d'un filtre de Canny (les pixels blancs sur la figure 2.18), précédemment abordé, qui élimine parfois les bords du tube, qui ne seront donc jamais détectés comme étant des droites par la transformée.

Malgré le côté interactif de la seconde méthode [EM00], les spécularités ou encore l'environnement industriel constituent toujours un problème. En effet, malgré une initialisation assez proche du tube, nous pouvons constater sur la figure 2.19 que nous obtenons parfois de très bons résultats mais aussi des plus mitigés où le segment jaune d'initialisation va soit ne pas être « attiré » en restant plus ou moins à sa position initiale ou bien être « attiré » par les spécularités du tube. De plus, ces résultats dépendent très fortement de l'initialisation, c'est à dire, la position des segments en jaunes sur la figure 2.19 qui sont définis manuellement. Cela implique un travail supplémentaire de « bonne » initialisation et d'ajustement de celle-ci pour chaque image que l'on devra étudier.

Pour les trois dernières méthodes « LSD », « EDLines » et « CannyLines » qui possèdent la même philosophie, nous pouvons observer que nous obtenons, encore une fois, une détection de droites possédant les mêmes caractéristiques. Nous avons, sur les figures 2.20, 2.21 et 2.22, une sur-détection au niveau du tube avec les spécularités et une

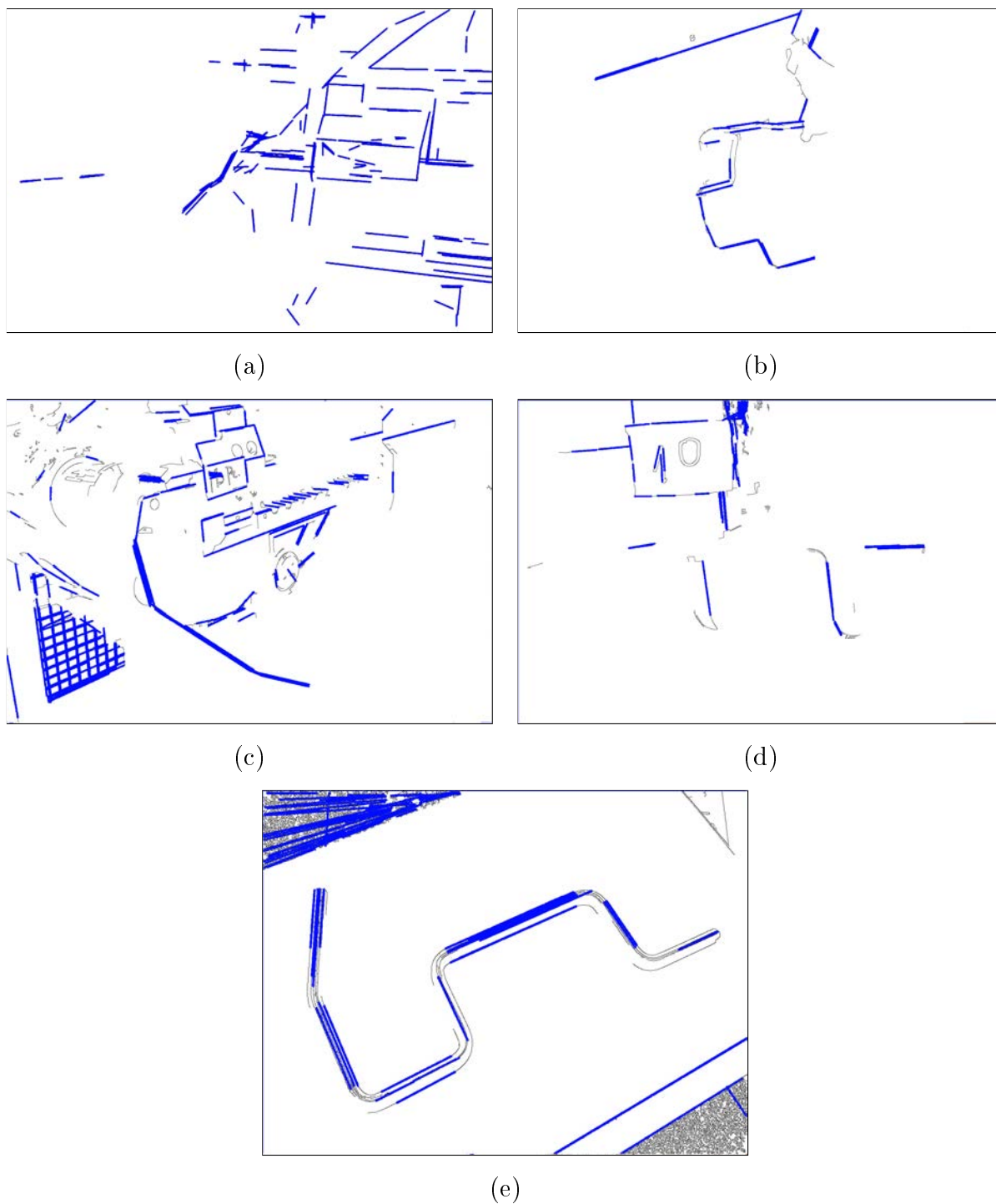
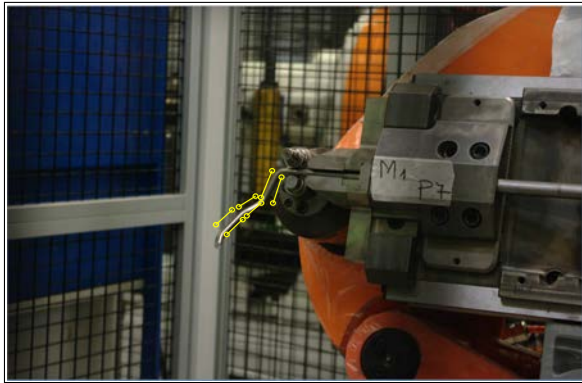
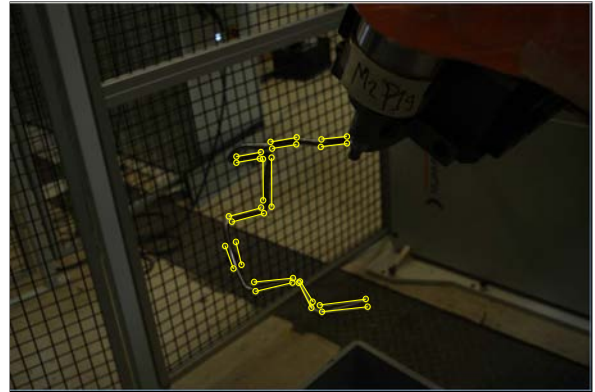


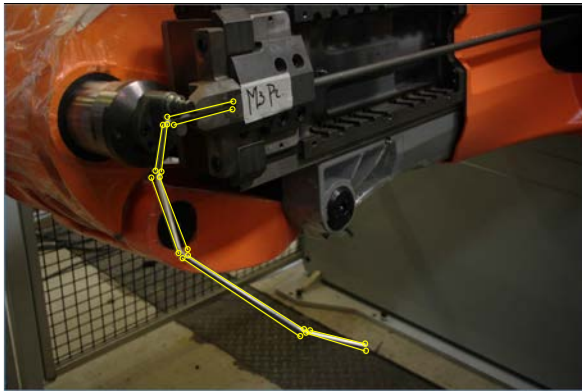
FIGURE 2.18 – Résultats obtenus avec la transformée de Hough probabiliste pour la détection de droites.



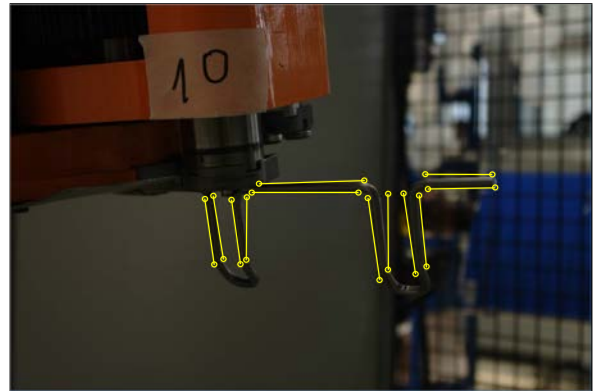
(a)



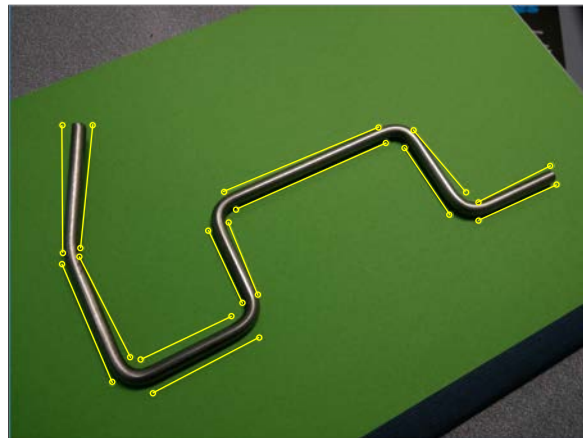
(b)



(c)



(d)



(e)

FIGURE 2.19 – Résultats obtenus avec une méthode interactive utilisant la transformée de Hough [EM00].

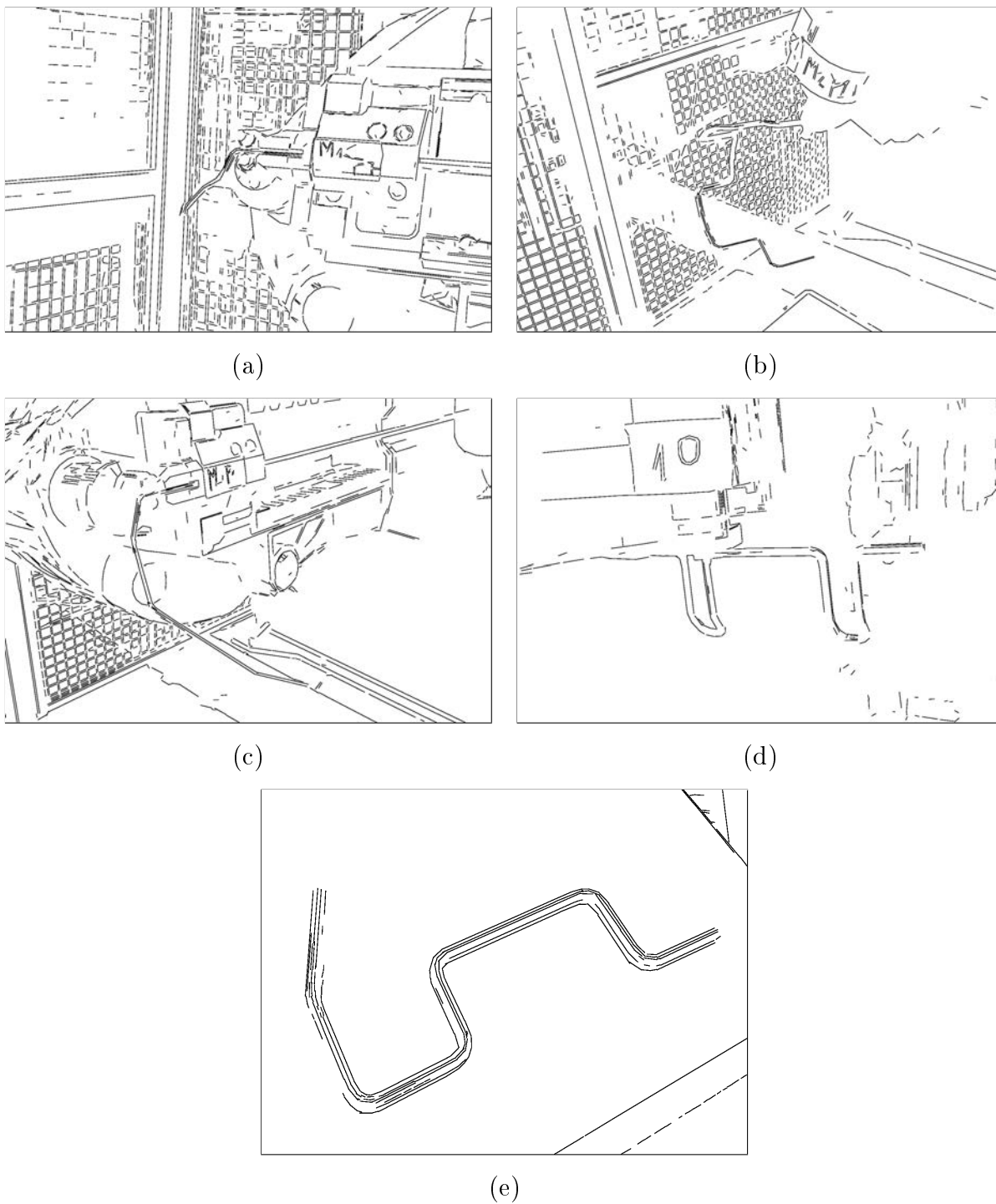
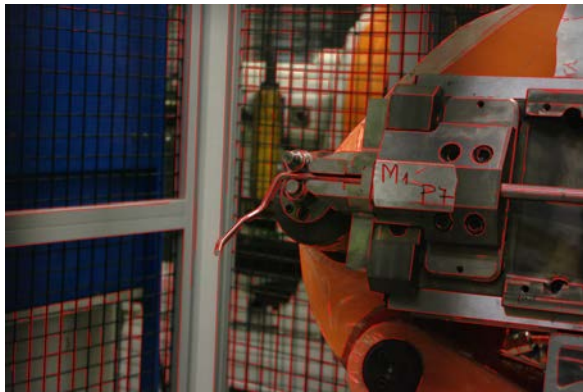
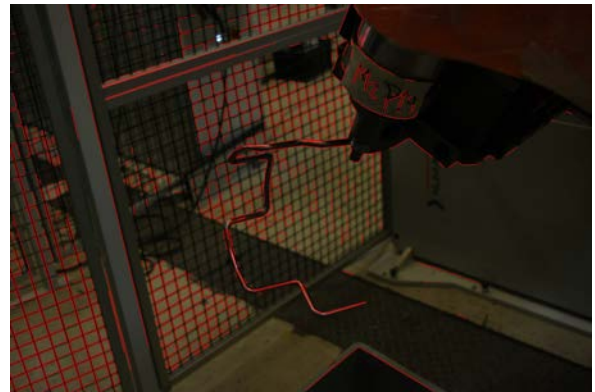


FIGURE 2.20 – Résultats obtenus avec l'algorithme de détection de droites : LSD (Line Segment Detector).



(a)



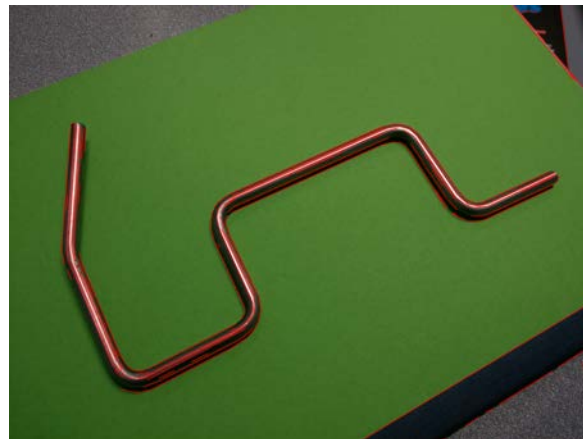
(b)



(c)



(d)



(e)

FIGURE 2.21 – Résultats obtenus avec l'algorithme de détection de droites : EDLines.

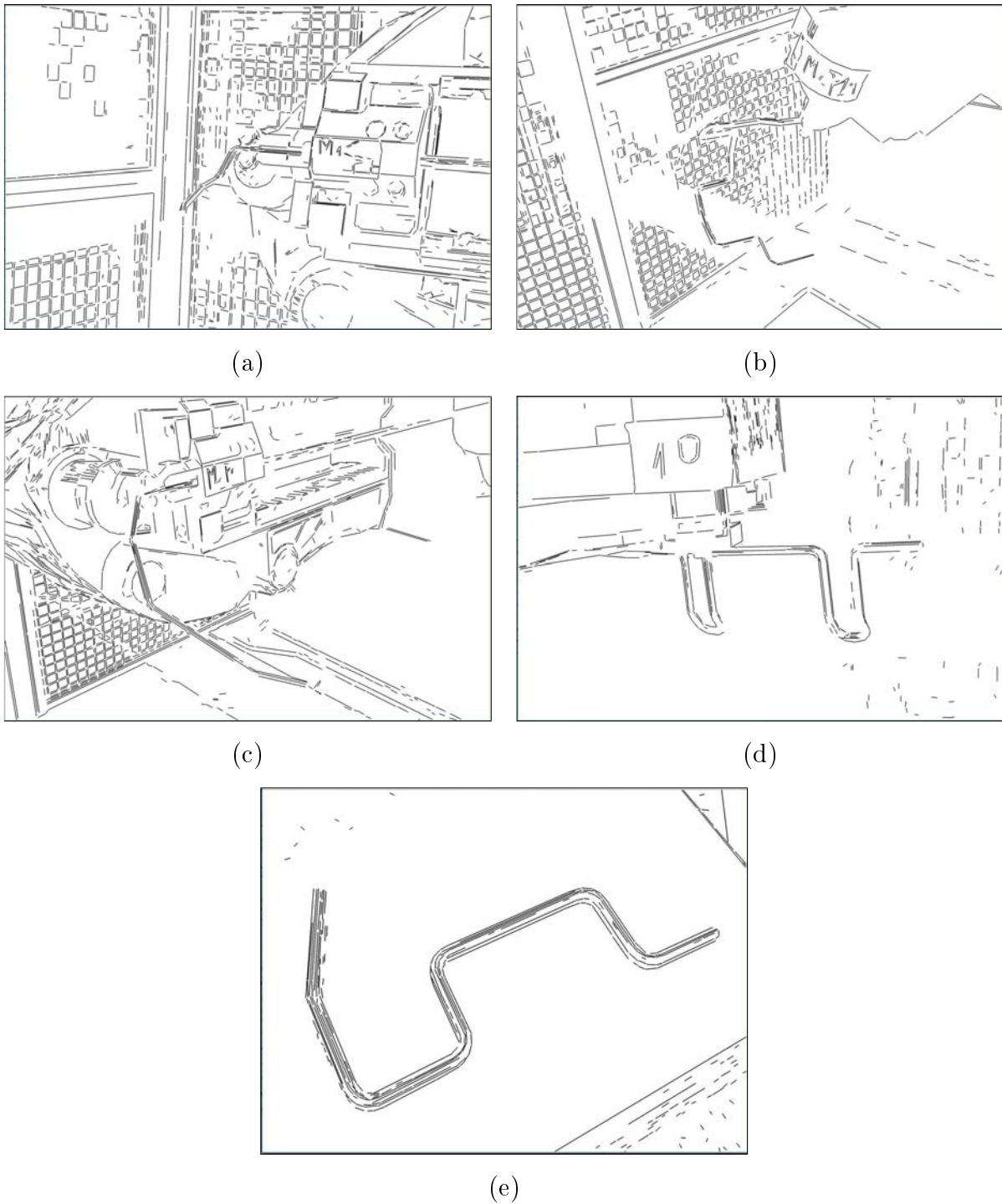


FIGURE 2.22 – Résultats obtenus avec l'algorithme de détection de droites : CannyLines.

détection parfois incomplète lorsque la droite détectée fait bien partie d'un bord du tube.

Pour conclure, nous avons expérimenté des méthodes avec des approches différentes qui pourraient, en théorie, nous permettre d'obtenir in fine les bords d'un tube métallique et si possible uniquement les bords. Nous avons pu constater que malgré des résultats parfois intéressants, aucune de ces méthodes ne nous a réellement permis d'obtenir les bords du tube. À partir de ces remarques et expérimentations, nous allons proposer une approche différente dans le [Chapitre 4](#) pour répondre au mieux à la segmentation d'un tube métallique dans son environnement industriel non contrôlé.

Chapitre 3

Géométrie discrète

Sommaire

3.1 Droites discrètes 2D	50
3.2 Segments flous	51
3.3 Détection de segments dans une image en niveaux de gris	57

DANS ce manuscrit, nous manipulons tout au long du processus une image numérique. Obtenu à travers un appareil d'acquisition tel qu'une caméra, une image numérique est par nature une entité « discrète ». Ce terme provient des mathématiques discrètes qui est un domaine où l'on traite des objets qui sont des ensembles dénombrables comme l'est une image numérique. En effet, nous pouvons définir une image en deux dimensions comme un ensemble de points (pixels) sur un plan à coordonnées entières. Comme pour le cas « continu », il a été nécessaire de définir une géométrie pour caractériser des objets mathématiques tels que les droites, les cercles, etc. dans un univers discret. Dans ce chapitre, nous allons présenter un objet de base de la géométrie discrète : les droites discrètes 2D. Il en existe plusieurs définitions mais nous nous intéressons en particulier à la définition faite par Reveillès [Rev91] basée sur une approche arithmétique qui permet d'avoir des propriétés intéressantes au niveau algorithmique. Cette définition est un point de départ pour le concept de « segment flou » qui peut être défini simplement, dans un premier temps, comme une formalisation de la notion de segment discret pour les données bruitées. Nous terminerons par l'étude d'une proposition d'extension des segments flous aux images en niveaux de gris qui sera le point de départ de la contribution de ce

manuscrit.

3.1 Droites discrètes 2D

La droite discrète est un élément que nous retrouvons dans de nombreux autres éléments discrets, ce qui en fait un objet de base essentiel de la géométrie discrète. En effet, les différentes définitions des droites discrètes ont permis d'ouvrir un large champs de recherche dans le domaine de la géométrie discrète et encore plus particulièrement dans l'étude des droites discrètes. De nombreux travaux ont été effectués pour fournir des estimateurs géométriques tels que des estimateurs des normales à une droite discrète, des tangentes [FT99, LVdV07] ou encore la courbure [FT99]. Il existe donc différentes définitions, mais toutes équivalentes, pour caractériser la droite discrète comme *le codage de Freeman* proposé par H. Freeman [Fre61] pour coder des courbes 8-connexe dans $\mathbf{Z}^2$ ou encore *la propriété de la corde* définie par Rosenfeld en 1974 dans [Ros74]. Nous ne rentrerons pas dans les détails de ces caractérisations pour nous attarder sur celle proposée par Reveillès en 1989 dans [Rev91].

Reveillès propose donc en 1989 une nouvelle approche pour caractériser un segment de droite discrète en utilisant une définition arithmétique.

Definition 4. Une droite discrète de pente $\frac{a}{b}$, de borne inférieure μ et d'épaisseur ω est définie par $D(a, b, \mu, \omega)$ avec $a, b, \mu, \omega \in \mathbf{Z}$. Les points $(x, y) \in \mathbf{Z}^2$ qui la composent vérifient la double inéquation suivante :

$$\mu \leq ax - by < \mu + \omega \quad (3.1)$$

Comme nous pouvons le voir sur la figure 3.1, nous considérons géométriquement une droite discrète comme un ensemble de points qui sont encadrés par deux droites réelles.

$$\begin{cases} ax - by = \mu \\ ax - by = \mu + \omega - 1 \end{cases} \quad (3.2)$$

Les droites discrètes peuvent avoir des connexités différentes (voir figure 3.2). Dans [Rev91], Reveillès introduit pour cela un théorème permettant, à partir des paramètres a, b et ω , de connaître la connexité de la droite discrète résultante. Le théorème est le suivant :

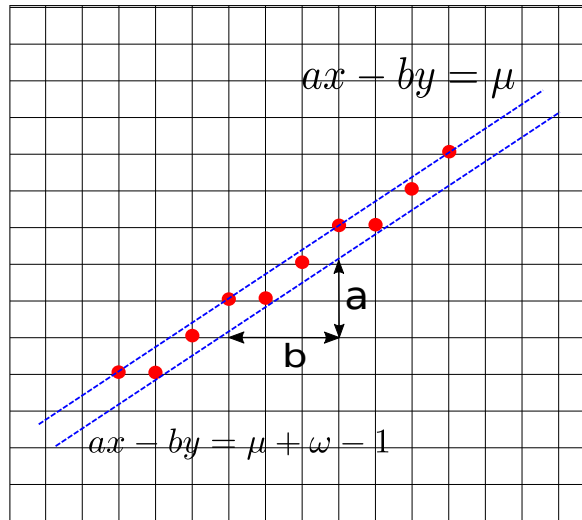


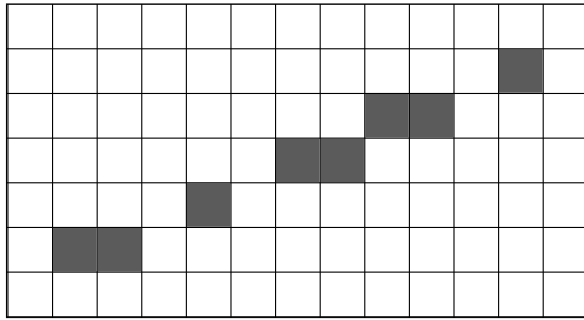
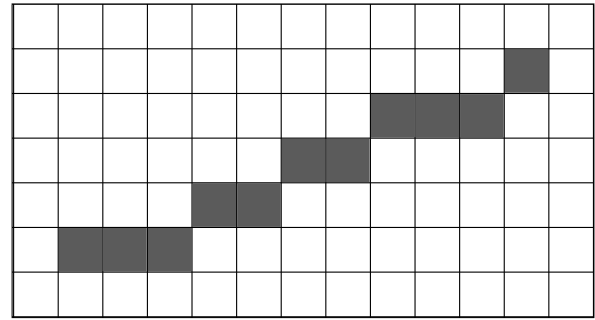
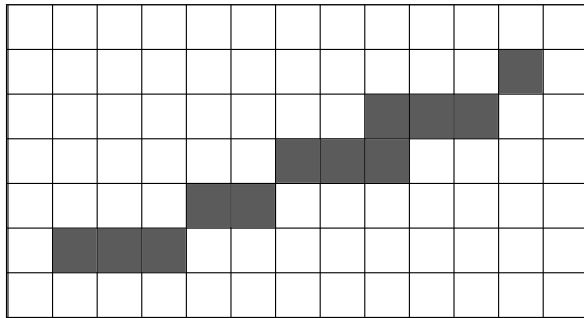
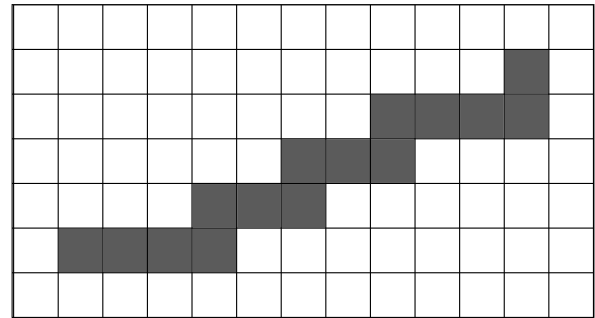
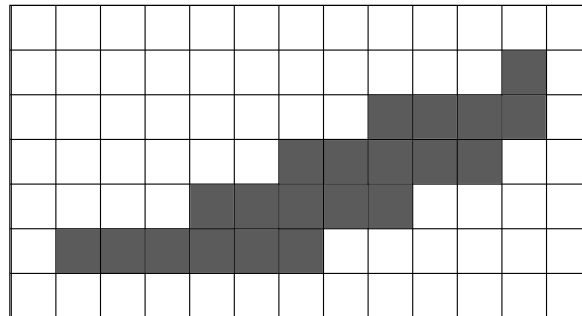
FIGURE 3.1 – Exemple de segment discrète arithmétique.

Théorème 1. *Étant donnée une droite discrète $D(a, b, \mu, \omega)$:*

- *elle est déconnectée, si $\omega < \max(|a|, |b|)$. (figure 3.2 (a))*
- *elle est 8-connexe, si $\omega = \max(|a|, |b|)$. Nous parlons aussi dans ce cas de droite naïve. (figure 3.2 (b))*
- *elle est *-connexe, c'est-à-dire qu'elle possède des 4-voisinages et 8-voisinages, si $\max(|a|, |b|) < \omega < |a| + |b|$. (figure 3.2 (c))*
- *elle est uniquement 4-connexe, si $\omega = |a| + |b|$. Nous parlons aussi dans ce cas de droite standard. (figure 3.2 (d))*
- *elle est épaisse, si $|a| + |b| < \omega$. (figure 3.2 (e))*

3.2 Segments flous

A partir de la définition des droites discrètes énoncée par Reveillès, plusieurs outils d'analyse ou de reconnaissance ont été proposés [FT99, DR95]. Ces outils sont malheureusement peu adaptés lorsque les droites sont bruitées. En effet, comme nous pouvons le constater sur la figure 3.3, un algorithme de reconnaissance de droites discrètes va produire un certain nombre de petits segments alors que nous aimerions obtenir un résultat beaucoup moins segmenté.

(a) segment déconnecté $D(3, 7, 0, 5)$ (b) segment naïf $D(3, 7, 0, 7)$ (c) segment *-connexe $D(3, 7, 0, 8)$ (d) segment standard $D(3, 7, 0, 10)$ (e) segment épais $D(3, 7, 0, 16)$ FIGURE 3.2 – Illustration des différentes connexités des droites discrètes par rapport à l'épaisseur ω .

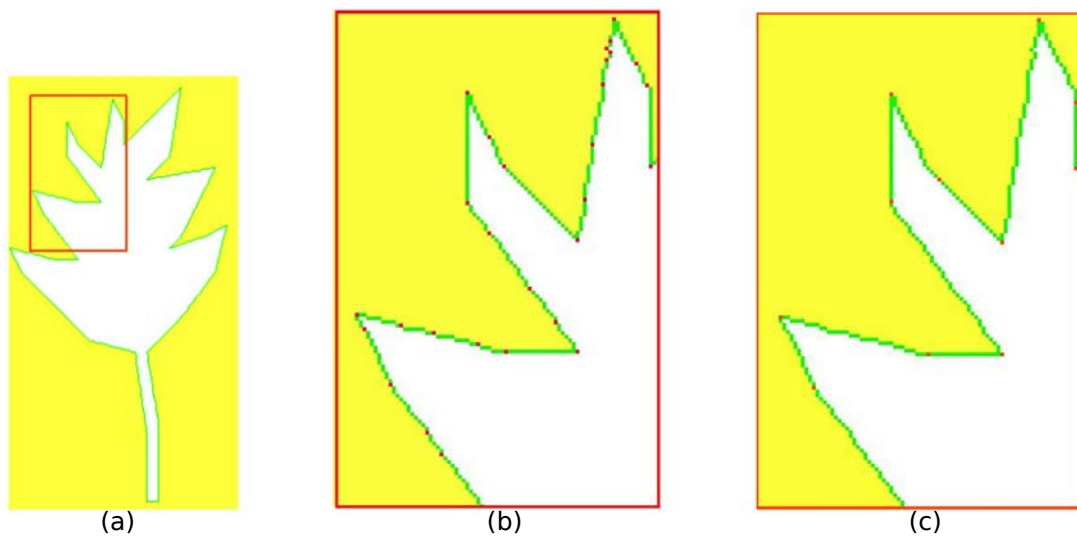


FIGURE 3.3 – Image extraite du manuscrit [DR07] pour l’obtention d’une Habilitation à diriger des recherches : (a) Un exemple de courbe discrète obtenue dans un contour de l’image d’une feuille. (b) Un agrandissement sur une partie du contour et segmentation obtenue avec un algorithme de reconnaissance de segments de droites naïves. (c) La segmentation souhaitée qui présente mieux la forme géométrique sur la même partie de la courbe.

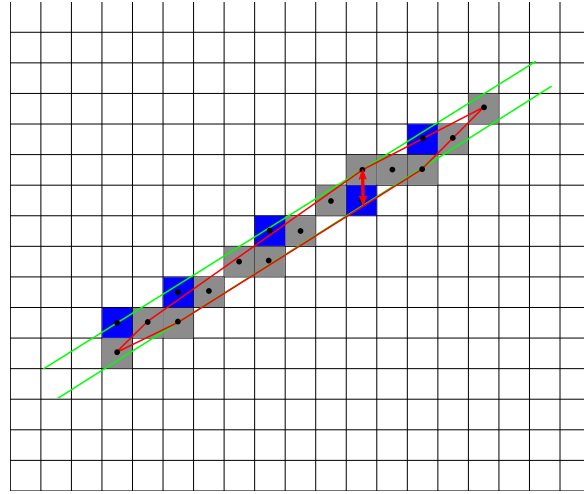


FIGURE 3.4 – Segment flou d'épaisseur 1 encadré par droite discrète $D(5, 8, -8, 11)$ - Les points considérés sont les points gris et les points bleus sont optimaux.

C'est dans cette optique que Debled-Rennesson a proposé une nouvelle notion : le segment flou [DRJLRD03, DRFRD06]. Le segment flou est défini comme un ensemble de points (sans notion de connexité particulière) qui appartient à une droite discrète dont on aurait fixé l'épaisseur. Debled-Rennesson propose deux approches pour la reconnaissance d'un segment flou : une approche arithmétique [DRJLRD03] et une approche géométrique [DRFRD06]. C'est sur cette seconde notion que nous allons nous attarder car c'est celle que nous utiliserons dans notre proposition.

L'approche géométrique des segments flous repose sur la notion d'enveloppe convexe de l'ensemble des points que nous étudions. Debled-Rennesson et al. proposent plusieurs définitions fondamentales qui vont conduire à la définition de cette nouvelle notion qu'est le segment flou [DRFRD06].

Définition 5. *L'épaisseur verticale d'une droite discrète $D(a, b, \mu, \omega)$ est la distance verticale (différence des ordonnées) entre les droites d'appui supérieures et inférieures de D .*

Définition 6. *La distance verticale d'un ensemble convexe C est la longueur maximale de l'intersection de l'enveloppe convexe de C avec une droite verticale.*

L'épaisseur verticale (figure 3.4 - double flèche verticale rouge) de $D(a, b, \mu, \omega)$ peut

être obtenue à l'aide de la formule $\frac{\omega-1}{\max(|a|,|b|)}$.

Definition 7. Une droite englobante $D(a, b, \mu, \omega)$ d'une suite de points entiers S_f est dite **optimale** si son épaisseur est égale à l'épaisseur verticale convexe de S_f .

A partir des définitions précédentes, nous pouvons spécifier un segment flou ainsi :

Definition 8. Soit S_f une suite de points entiers, S_f est un segment flou d'épaisseur v si et seulement si sa droite optimale englobante a une épaisseur verticale $(\frac{\omega-1}{\max(|a|,|b|)})$ inférieure ou égale à v .

À partir de cette définition, il a été proposé par Debled-Rennesson et al. dans [DRFRD06] un algorithme pour définir si un ensemble de points est un segment flou d'une certaine épaisseur v fixée ou non. Cet algorithme (Alg. 5) est linéaire et incrémental et fournit en plus des caractéristiques de la droite englobante optimale.

L'algorithme de reconnaissance d'un segment flou S_f revient à calculer la distance verticale de l'enveloppe convexe de S_f notée $\text{conv}(S_f)$. Dans l'algorithme présenté, nous considérons que la suite de points S se trouve dans le premier octant et les points sont ordonnés par ordre croissant des abscisses. Étant donné un ensemble de points S , à chaque itération nous essayons d'ajouter au segment flou S_f un nouveau point M de l'ensemble S . Nous avons trois possibilités : soit ce nouveau point se situe au-dessus, en-dessous ou appartient à la droite discrète englobante D . Dans les trois cas, nous recalculons l'enveloppe convexe de $S'f = S_f \cup \{M\}$ à l'aide de l'algorithme de Melkman [Mel87]. Dans le cas où le point M appartient à la droite D , alors la distance verticale n'est pas recalculée et S'_f est un segment flou de même épaisseur et avec la même droite englobante optimale D . Dans les deux autres cas, nous devons redéfinir la droite englobante D pour ce nouvel ensemble de points S'_f . Nous arrêtons notre parcours lorsque l'épaisseur verticale devient supérieure à l'épaisseur v . L'ensemble S est considéré au final comme un segment flou si, à la fin du parcours de celui-ci, nous avons une épaisseur verticale qui est inférieure à l'épaisseur v .

Dans la suite du manuscrit, nous utiliserons une implémentation des segments flous proposée par la librairie DGTal [dgt16]¹. Cette implémentation est très intéressante car

1. <http://dgtal.org/doc/nightly/moduleAlphaThickSegmentReco.html>

Algorithme 1 : Reconnaissance incrémentale d'un segment flou à l'épaisseur v dans le premier octant.

Données : S une suite de points entiers, v un réel

Résultat : $isSegment$ un booléen de valeur vraie si S est un segment flou d'épaisseur v et faux sinon ; $a_f, b_f, \mu_f, \omega_f$ entiers, les caractéristiques de la droite englobante optimale de S si elle existe pour l'épaisseur v

```

1 Initialisation :  $isSegment = true$ ;  $a = 0$ ;  $b = 1$ ;  $\omega = b$ ;  $\mu = 0$ ;  $M = (x_0, y_0)$ ;
2 tant que  $S$  n'est pas entièrement parcourue et  $isSegment$  faire
3    $M =$  point suivant de  $S$ ;
4   Ajouter  $M$  aux enveloppes convexes inférieures et supérieures de la suite de
     points de  $S$  parcourue;
5    $r = ax_M - by_M$ ;
6   si  $r = \mu$  alors
7      $U_L = M$ ;
8   si  $r = \mu + \omega - 1$  alors
9      $L_L = M$ ;
10  si  $r \leq \mu - 1$  alors
11     $U_L = M$ ;
12     $N =$  le point situé avant  $M$  dans l'enveloppe convexe supérieure;
13     $a = y_M - y_N$ ;  $b = x_M - x_N$ ;  $\mu = ax_M - by_M$ ;
14    Trouver le premier point  $C$  dans la partie inférieure de l'enveloppe convexe
     en commençant par le point  $L_L$ , tel que la pente de  $[C, C_{next}] > \frac{a}{b}$ ;
15     $L_L = C$ ;
16     $\omega = ax_{L_L} - by_{L_L} - \mu + 1$ ;
17  sinon
18    si  $r \geq \mu + \omega - 1$  alors
19      cas symétrique;
20   $isSegment = \frac{\omega-1}{b} \leq v$ ;
21 si  $isSegment$  alors
22    $a_f = \frac{a}{\gcd(a,b)}$ ;  $b_f = \frac{b}{\gcd(a,b)}$ ;  $\mu_f = \frac{\mu}{\gcd(a,b)}$ ;  $\omega_f = \frac{\omega-1}{\gcd(a,b)} + 1$ ;

```

elle supprime, entre autre, la contrainte sur l'ordre croissant des abscisses des points. Cela permettra de considérer n'importe quel ensemble de points dans notre algorithme.

Pour finir, nous allons introduire deux notions qui pourront être utiles en perspective car elles sont centrales dans l'estimation des paramètres géométriques. La première notion est ce que l'on nomme un segment flou maximal. La définition d'un segment flou maximal est la suivante² :

Definition 9. Soit $\mathcal{C}$ une courbe discrète et $\mathcal{C}_{i,j}$ une séquence de points de $\mathcal{C}$ indexés de i à j . Soit $SF(i, j, v)$ une notation pour définir un segment flou $\mathcal{C}_{i,j}$ d'épaisseur v . Alors $\mathcal{C}_{i,j}$ est un **segment flou maximal d'épaisseur v** , que l'on note $SFM(i, j, v)$ si il vérifie :

$$SFM(i, j, v) \Leftrightarrow SF(i, j, v) \wedge \neg SF(i-1, j, v) \wedge \neg SF(i, j+1, v) \quad (3.3)$$

La seconde notion est liée à la première et se nomme la couverture tangentielle. Cette notion est définie par un contour discret $\mathcal{C}$, constitué d'un ensemble de points $\mathcal{C}_i$. À partir du point initial $\mathcal{C}_0$ du contour, nous détectons l'ensemble des segments flous maximaux. Cet ensemble représente la couverture tangentielle et permet d'estimer la direction de la tangente de la courbe continue que la courbe discrète $\mathcal{C}$ représente. Cette notion permet, par exemple, comme le propose Kerautret et Lachaud [KL09] d'obtenir un estimateur de courbure dont l'objectif est de déterminer la forme continue qui se rapproche le plus du contour discret $\mathcal{C}$ car un contour discret peut être le résultat de la discrétisation d'une infinité d'objets continus.

3.3 Détection de segments dans une image en niveaux de gris

Maintenant que les segments flous ainsi que l'algorithme de reconnaissance de ceux-ci ont été définis, nous allons nous intéresser à une proposition d'extension pour les images en niveaux de gris. Pour cela, Kerautret et Even énoncent un parcours incrémental de l'image en niveaux de gris pour effectuer une extraction de segments de droites [KE09].

La méthode pour extraire un segment de droite de l'image se déroule de la façon suivante :

2. Définition obtenue dans [KL09]

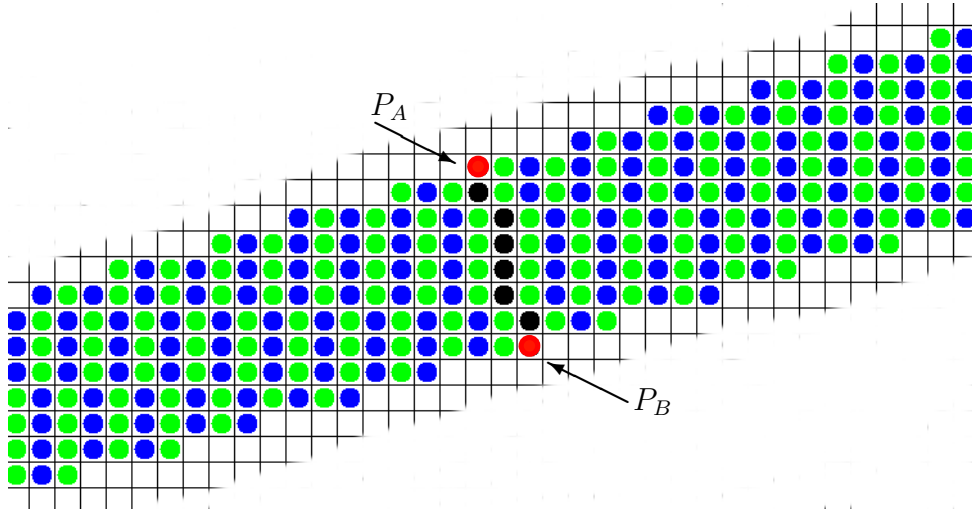


FIGURE 3.5 – Parcours du segment S_0 (en noir) dans la direction orthogonale à droite et à gauche.

1. un segment de droite discrète naïve (8-connex) S_0 qui coupe le segment de droite que nous aimerions extraire de l'image est défini. Cette définition se fait à partir de deux points P_A et P_B (figure. 3.5).
2. à partir de S_0 , nous parcourons un sous-ensemble de l'image en définissant d'autres segments de droite S_n correspondant à une translation de S_0 dans les deux directions orthogonales à celui-ci.
3. pour chacun des segments S_n , nous sélectionnons le pixel qui possède la norme du gradient la plus forte et dont l'angle entre la direction de parcours et la direction du gradient soit inférieur à un seuil $\theta_{max} = \frac{\pi}{2}$ (figure. 3.6). Nous obtenons à l'issue de cela, une liste de pixels candidats.
4. une reconnaissance de segment flou S_f à partir de la liste des pixels candidats est effectuée.
5. deux nouveaux points P_A et P_B sont définis de telle sorte que le segment S_0 soit maintenant orthogonal à la direction du segment flou S_f .
6. les étapes 2, 3 et 4 sont exécutées à nouveau avec le paramètre $\theta_{max} = \frac{\pi}{8}$.
7. de même avec le paramètre $\theta_{max} = \frac{\pi}{10}$.

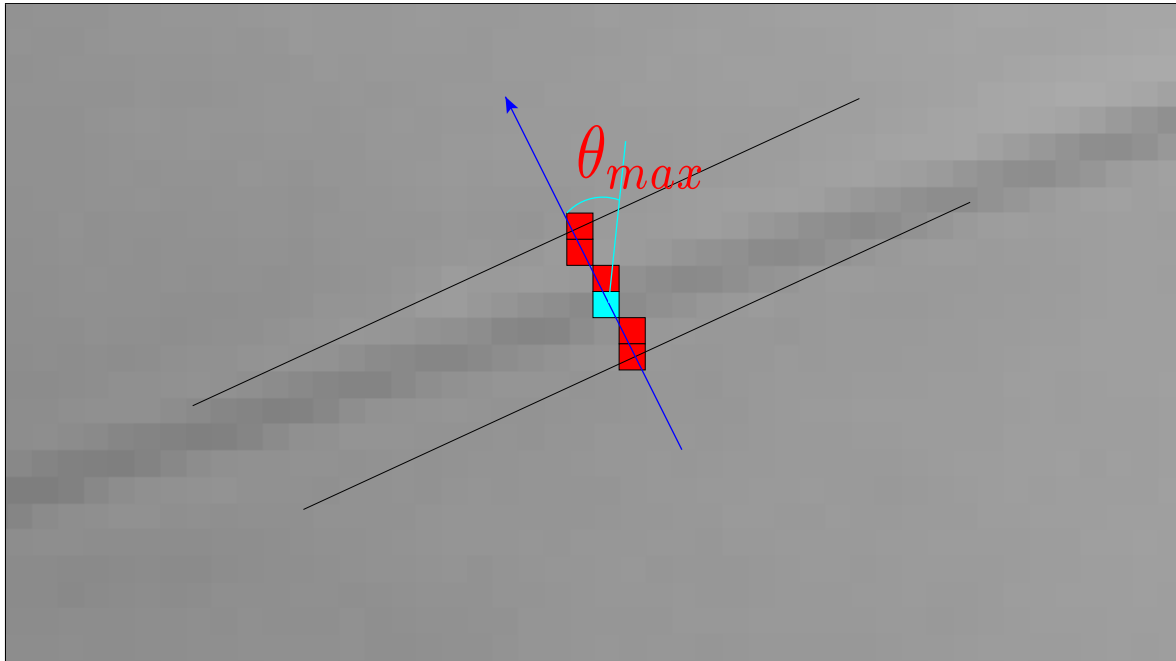


FIGURE 3.6 – Illustration de la sélection d'un pixel sur un segment de droite ayant le gradient le plus fort et dont l'angle entre la direction de parcours et la direction du gradient est inférieur au seuil $\theta_{max} = \{\frac{\pi}{2}, \frac{\pi}{8}, \frac{\pi}{10}\}$ (Les auteurs précisent que se sont des valeurs empiriques validées sur des images réelles).

Le but recherché en exécutant trois fois la procédure est d'essayer d'étendre le plus possible le segment flou pour extraire le segment de droite le plus long possible.

Nous voyons sur la figure 3.7 ce qu'il est possible d'obtenir comme reconnaissance à partir des différentes initialisations matérialisées par les segments rouges sur l'image. Les différents segments bleus illustrent les droites d'appui supérieures et inférieures du segment flou reconnu par l'algorithme.

Dans le Chapitre 2, nous avons expérimenté diverses méthodes de segmentation appliquées sur des images dans un environnement industriel afin d'en extraire un objet tubulaire comportant des spécularités. Nous avons pu constater que ces différents algorithmes, qui sont pour la plupart globaux et qui utilisaient l'information du gradient de l'image, ne produisaient pas une segmentation suffisante pour obtenir uniquement les contours de notre objet d'étude. Étant donné que nous possédons la possibilité d'identifier une zone dans l'image où le tube devrait être présent, nous allons dans le chapitre suivant proposer

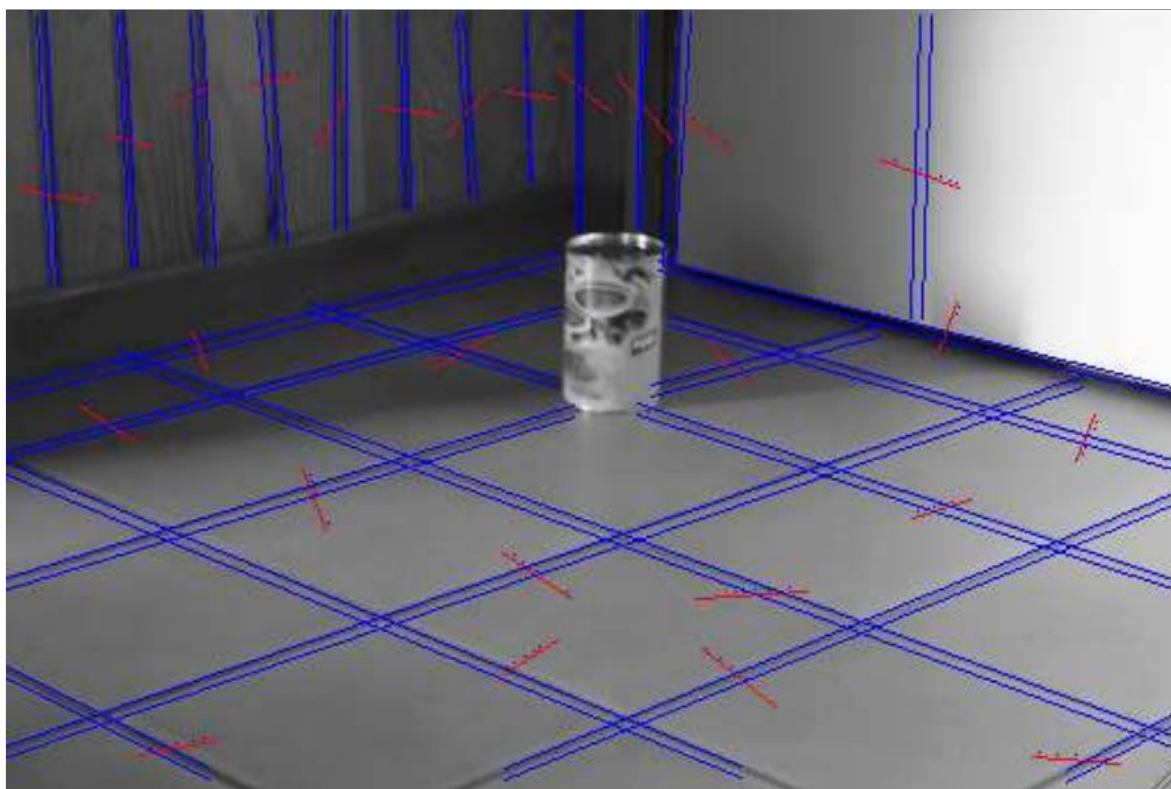


FIGURE 3.7 – Illustration de segments flous obtenus (en bleu) sur une image réelle à partir de sélections représentées par les segments rouges sur l'image.

une méthode se basant sur le parcours incrémental de l'image que nous venons de décrire dans cette section. Nous présenterons de plus une alternative à l'utilisation de l'information des gradients de l'image pour la sélection des pixels appartenant aux contours de l'objet.

Deuxième partie

Contributions

Chapitre 4

Méthode des profils d'intensité

Sommaire

4.1	Motivation	66
4.2	Notre approche	71
4.2.1	Notions importantes	72
4.2.2	Exploration d'une zone de recherche	74
4.3	Extension de la méthode : interpolation	82
4.4	Analyse des deux algorithmes	86
4.5	Extension aux images non industrielles et limites	94

COMME nous avons pu le constater dans le [Chapitre 2](#), les méthodes étudiées rencontrent différents problèmes avec les images que nous proposons d'analyser. Nous pouvons identifier deux situations qui, en général, génèrent des solutions non satisfaisantes. La première situation, le cœur de notre étude, est l'aspect spéculaire des objets que l'on souhaite reconnaître. La seconde est liée à l'environnement non contrôlé dans lequel l'image a été acquise. En effet, il peut arriver que le fond se confonde avec la pièce, ce qui produit des images où certains bords de l'objet ne se dégagent pas clairement et compliquent énormément sa segmentation.

Dans ce chapitre, nous allons présenter une approche différente n'utilisant aucune information de gradient mais exploitant la notion de profil d'intensité. Pour cela, nous commencerons par motiver notre choix de l'utilisation de cette notion qui sera détaillée par la suite. Nous terminerons ce chapitre par la présentation de deux algorithmes utili-

sant la notion de profil d'intensité qui permettront de contourner les problèmes évoqués précédemment.

4.1 Motivation

La stratégie que nous proposons consiste à exploiter les caractéristiques géométriques des objets tubulaires en y associant leurs propriétés photométriques. L'idée principale est plus précisément d'utiliser la cohérence de l'intensité de l'image que l'on peut mettre en évidence en considérant un modèle de réflectance comme par exemple le modèle proposé par Nayar [NIK91]. Un modèle comme celui-ci est capable de modéliser les parties diffuses et spéculaires de la lumière réfléchie. La figure 4.1 (a) illustre les valeurs d'intensité I réfléchies sur la surface de l'objet en fonction de la position de l'observateur avec une source de lumière fixe. Nous considérons la source de lumière comme située à l'infinie, c'est à dire que les rayons de lumière sont parallèles. Le lobe de réflexion diffus représente la partie de la lumière qui est réfléchie de manière équivalente dans toutes les directions (e.g. une surface mate comme le papier) alors que le lobe de réflexion spéculaire ainsi que le pic spéculaire dépendent de la position de l'observateur (e.g. une surface comme de l'eau sans mouvement). Formellement, pour une position fixe de lumière, la quantité de lumière réfléchie peut être définie comme une composition de ces trois composantes :

$$I = K_{dl} \cos(\theta_i) + K_{sl} \frac{1}{\cos(\theta_r)} e^{\frac{-\alpha^2}{2\sigma^2}} + K_{ss} \delta(\theta_i - \theta_r) \delta(\phi_r) \quad (4.1)$$

Les angles θ_i et (θ_r, ϕ_r) sont respectivement associés au rayon lumineux incident et réfléchi et à leur vecteur bissecteur par rapport à la normale à la surface notée $\vec{n}$ (voir Figure. 4.1 (b)). Pour terminer, les paramètres K_{dl} (diffus), K_{sl} (lobe spéculaire), K_{ss} (pic spéculaire) sont liés aux propriétés physiques de l'objet.

À partir de ce modèle, considérons l'ensemble des points appartenant à une même droite d , parallèle à l'axe principal du cylindre (droite rouge sur la Fig. 4.1 (b)). Par définition, nous avons l'angle entre la direction de la source de lumière et la normale (θ_i) qui reste constant. En conséquence, par rapport à l'équation 4.1, la valeur d'intensité de tous les points de la droite d est constante dans le cas où la surface est Lambertienne (i.e. $K_{sl}, K_{ss} = 0$). Dans le cas de matériaux spéculaires, comme les objets métalliques,

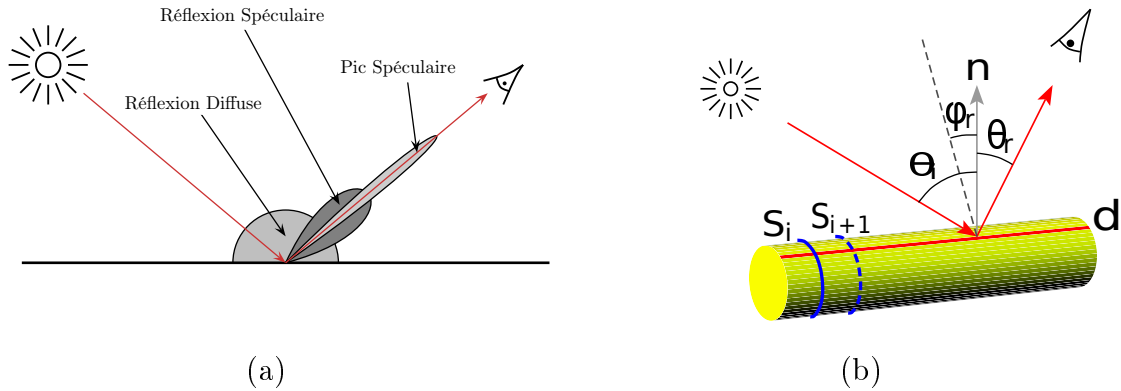


FIGURE 4.1 – Illustration des trois composantes du modèle de réflectance de Nayar [NIK91] obtenues à l’aide d’une source de lumière fixe (a). L’image (b) illustre la réflexion de la lumière sur une surface d’un objet cylindrique.

la valeur d’intensité I de la lumière réfléchie dépendra de la position de l’observateur (θ_r et ϕ_r). Si l’observateur est suffisamment loin de l’objet tubulaire qu’il observe, alors les variations de l’angle θ_r , provoquées par la translation le long de la droite d , peuvent être considérées comme limitées. Alors, le lobe spéculaire de la lumière réfléchie ne présente pas une grande variation autour de la direction du pic spéculaire (voir la zone sombre dans la Fig. 4.1 (a)). Pour terminer, le pic spéculaire ($K_{ss} \neq 0$) est relativement sensible à la position de l’observateur, cependant ce type de particularités est plutôt spécifique aux objets brillants comme les objets en plastique, réduit à un secteur fin du champ d’observation et n’est pas forcément pertinent pour les objets métalliques que nous allons rencontrer au cours de notre étude.

En dehors des considérations mathématiques, nous pouvons constater visuellement que dans de nombreux cas de figures, un objet tubulaire métallique joue un rôle plus ou moins de miroir. Nous pouvons voir, par exemple, sur la figure 4.2 (a) que le crapaud se projette sur le tube en arrière plan en subissant un étirement dans la direction du tube. De plus, ces structures linéaires obtenues dans la direction du tube peuvent être plus ou moins fines selon le diamètre du tube (i.e. plus le rayon est petit, plus la structure sera compressée) comme nous pouvons le voir sur un exemple plus commun sur la figure 4.2 (b).

Fort de ces observations photométriques, l’idée principale est de partir du postulat



FIGURE 4.2 – Illustration des structures linéaires que nous retrouvons sur un objet tubulaire.

suivant : si l'on prend une coupe transversale S_i , comme nous pouvons le voir en bleu sur la figure 4.1 (b), d'un objet tubulaire représentant un profil d'intensité, alors en *moyenne* tous les $S_i + n$ qui représentent aussi un profil d'intensité seront relativement proches en terme de variations.

Étant donné que dans un environnement réel, l'hypothèse d'une source lumineuse avec une direction constante n'est pas parfaitement garantie, nous avons effectué des expériences sur des données synthétiques en utilisant une source de lumière ponctuelle et en changeant sa position par rapport au centre de la scène (figure 4.3) puis sur des images de cas industriel (figure 4.4).

Dans de tels cas, la source lumineuse émet dans toutes les directions et les variations du profil d'intensité peuvent changer sur les parties linéaires de l'objet tubulaire. Afin de mesurer cette influence pour différentes distances de source de lumière dL , les mesures d'erreur quadratique moyenne ont été calculées entre un profil de référence et une série de profils pris le long de la boîte violette (voir première et deuxième lignes de la figure 4.3). Pour tous les différents paramètres de rendu, la variation d'erreur diminue rapidement

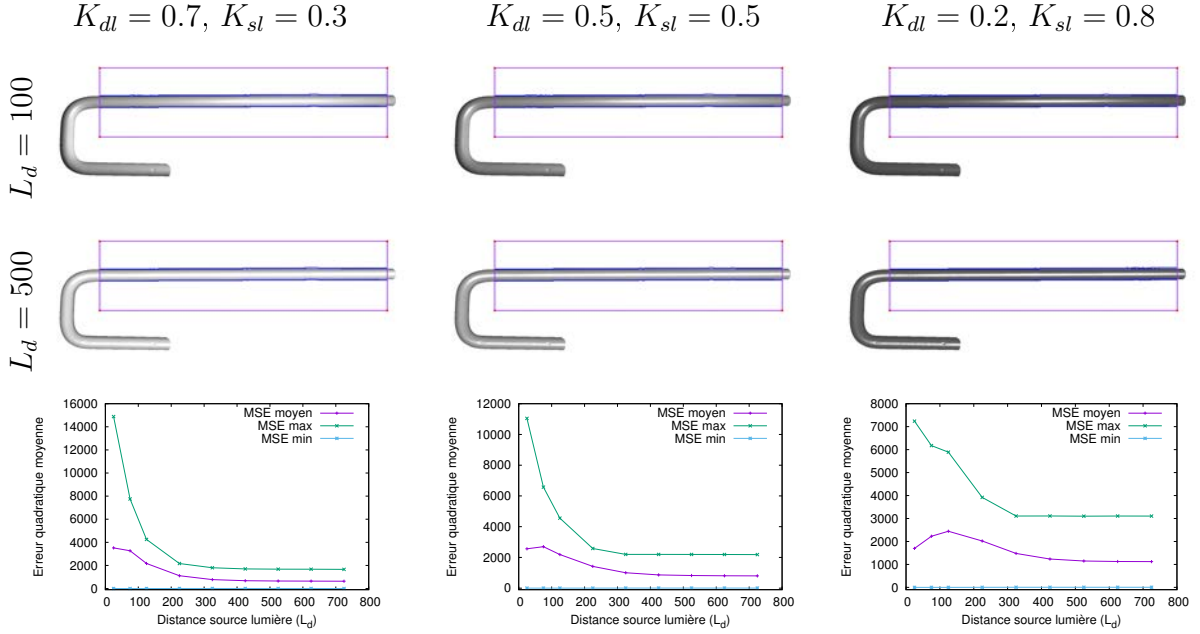
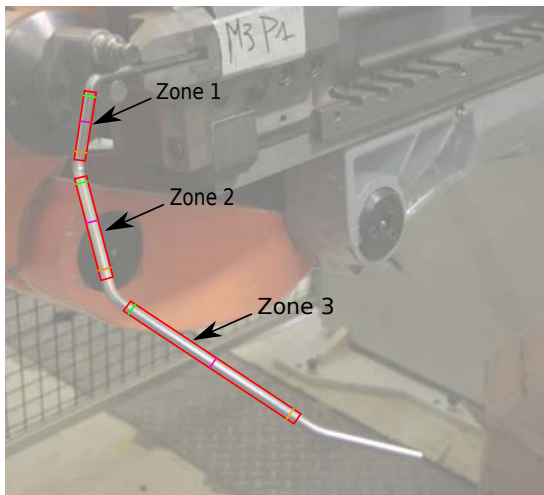


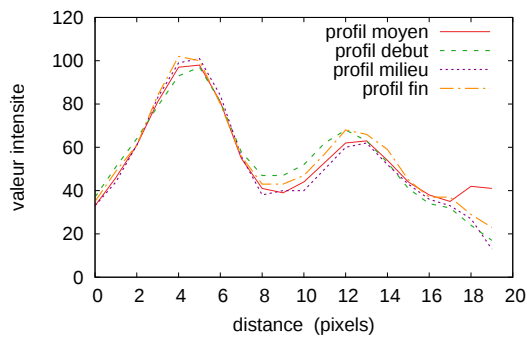
FIGURE 4.3 – Expérimentations autour de l’étude des variations de profil d’intensité sur une forme tubulaire réaliste lorsque la distance de la source de la lumière est modifiée ou encore les paramètres de rendu. La première et deuxième lignes montrent deux exemples où nous avons utilisé deux positions de sources lumineuses différentes ($L_d = 100$ and $L_d = 500$). La troisième et dernière ligne montrent les variations de profil d’intensité définies à partir de l’erreur quadratique moyenne à partir d’un profil de référence.

et devient alors stable (près de la distance 300, pour comparaisons, la partie linéaire du tube est de 255). Ces expériences confirment bien que l’hypothèse d’un profil constant est robuste même si une source de lumière uniforme non parfaite est présente dans la scène.

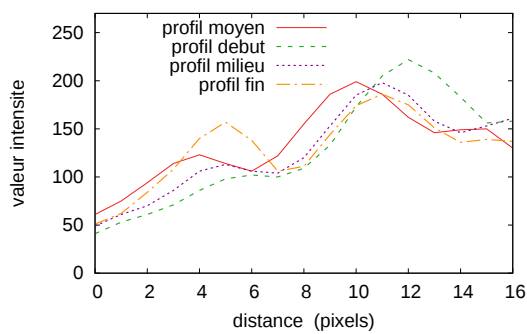
Pour les cas industriels (figure 4.4), nous avons calculé le profil d’intensité moyen de la zone (i.e. la moyenne de tous les profils d’intensité contenus dans la zone rectangulaire) complète représentée par un rectangle rouge (cela représente un segment droit de tube) puis nous l’avons comparé à trois profils d’intensité pris au début, au milieu et à la fin de cette zone, représentés respectivement par un segment vert, fuchsia et orange. Nous pouvons voir sur les différents graphiques qu’en effet les différentes courbes sont relativement proches, ce qui veut dire que le profil d’intensité le long d’un segment droit



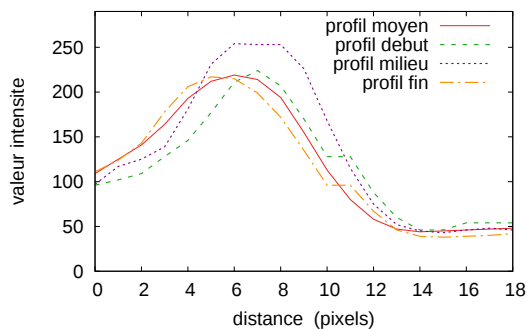
(a) source A



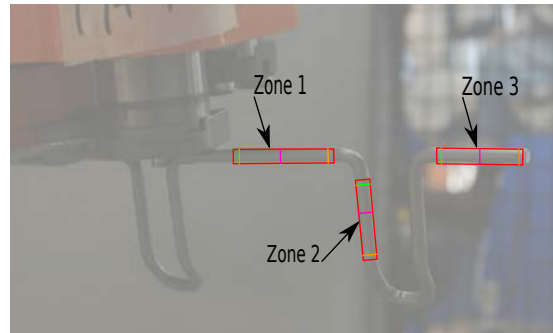
(b) zone 1 source A



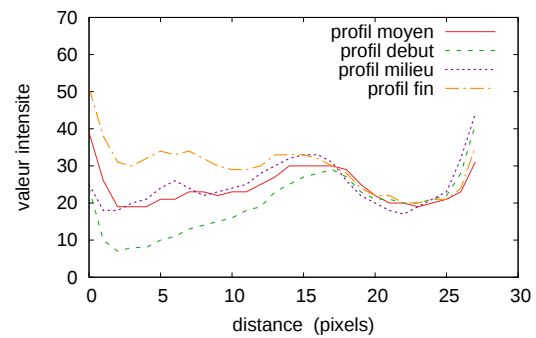
(c) zone 2 source A



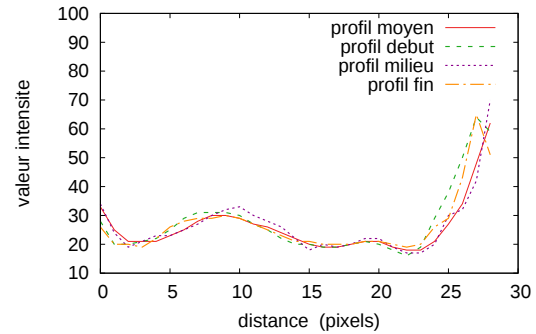
(g) zone 3 source A



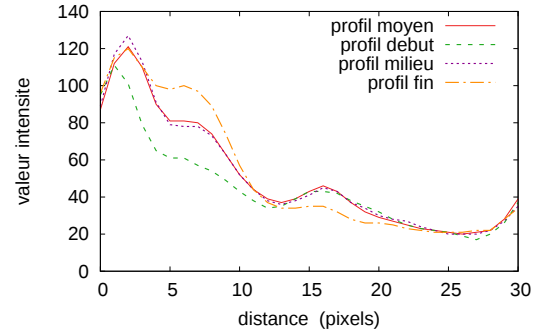
(e) source B



(f) zone 1 source B



(d) zone 2 source B



(h) zone 3 source B

FIGURE 4.4 – Comparaison entre le profil d'intensité moyen d'un segment droit du tube et trois profils d'intensité pris au début, milieu et fin du segment droit du tube.

de tube n'évolue pas de manière radicalement différente. Une visualisation supplémentaire, cette fois en 3D, montre sous forme de champs de hauteur, les niveaux de gris de la zone 3 de la source A et de la zone 3 de la source B (voir figure 4.5). Nous pouvons, là aussi, constater une certaine régularité au niveau des intensités quand on se déplace sur la même ligne le long de la surface.

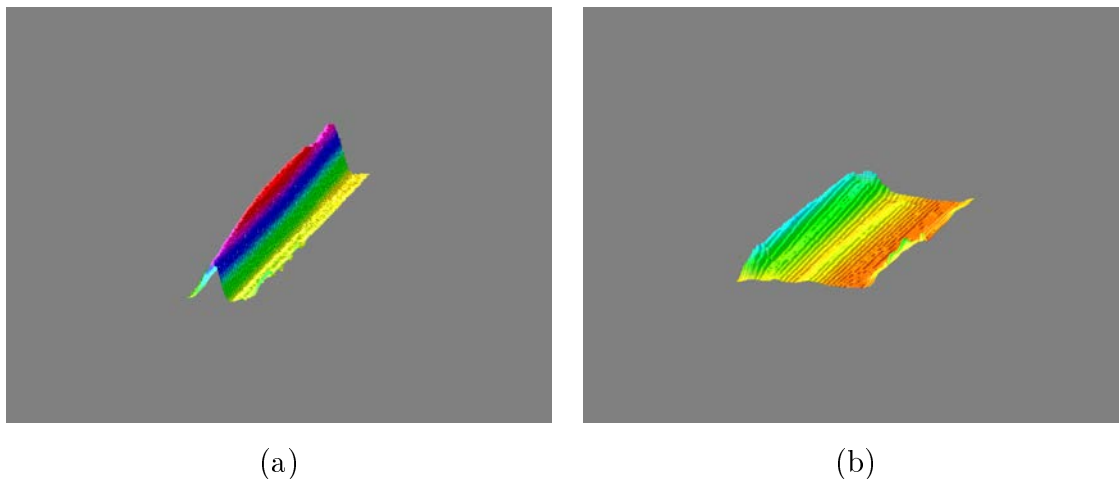


FIGURE 4.5 – Visualisation 3D des intensités de la zone 3 - source A (a) et de la zone 3 - source B (b).

4.2 Notre approche

Nous allons, dans cette section, présenter en détails cette nouvelle méthode qui s'appuie sur les différentes observations que nous avons faites dans la section précédente.

De manière succincte, notre méthode consiste, dans une ou plusieurs zone(s) rectangulaire(s) de l'image définies manuellement par l'utilisateur, à la(les) parcourir incrémentalement à l'aide de segments discrets comme cela a été proposé par Kerautret et Even [KE09]. À partir d'un profil d'intensité référence qui représente l'empreinte du tube dans une zone donnée, l'objectif est de retrouver, sur chacun des segments discrets, le sous-segment discret qui représente un profil d'intensité le plus proche possible de celui représentant l'empreinte du tube. Avant de décrire plus en détails l'algorithme, nous allons introduire les notions et définitions dont nous aurons besoin tout au long de cette section.

4.2.1 Notions importantes

Pour notre méthode, nous avons besoin de définir plusieurs structures visibles sur la figure 4.6 et dont les définitions sont les suivantes :

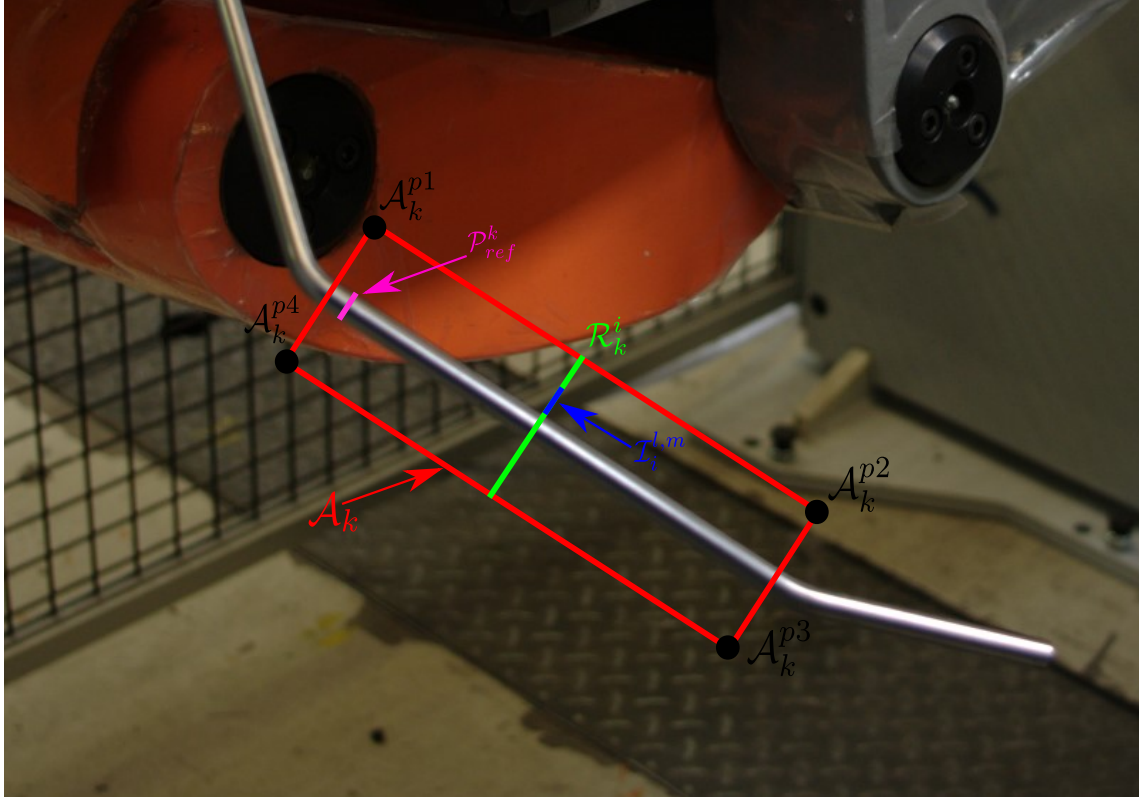


FIGURE 4.6 – Illustration montrant les différentes structures utilisées par la méthode.

- **Image référence** : l'image référence est l'image dans laquelle l'utilisateur va définir les différents éléments utiles pour l'algorithme.
- **Zone de recherche $\mathcal{A}_k$** : est une zone rectangulaire définie par l'utilisateur dans l'image référence. Cette zone est définie par quatre points $(\mathcal{A}_k^{p1}, \mathcal{A}_k^{p2}, \mathcal{A}_k^{p3}, \mathcal{A}_k^{p4})$ en coordonnées entières. k représente l'indice pour désigner une zone de recherche particulière dont les bornes sont $0 \leq k < n_a - 1$ et où n_a représente le nombre de zones de recherche définies dans l'image.
- **Profil d'intensité $\mathcal{P}$** : est une suite de valeurs associées à une séquence de pixels inscrite dans un segment discret naïf (8-connexe) où la valeur des pixels est com-

prise entre 0 et 255 (i.e. valeur des niveaux de gris).

- **Profil d'intensité référence** $\mathcal{P}_{ref}^k$: est un profil d'intensité dont les valeurs sont déduites d'un positionnement manuel d'un segment sur l'image référence. Il est utilisé comme initialisation de l'algorithme et chaque zone de recherche $\mathcal{A}_k$ possède un unique profil d'intensité référence $\mathcal{P}_{ref}^k$.
- **Segment de recherche** $\mathcal{R}_k^i$: pour parcourir la totalité d'une zone de recherche $\mathcal{A}_k$ de longueur d_r , nous déplaçons un segment discret que l'on nomme segment de recherche $\mathcal{R}_k^i$ par translation successive dans la direction donnée par le vecteur $(\mathcal{A}_k^{p2} - \mathcal{A}_k^{p1})$. L'indice i représente le segment de recherche en cours dont les bornes sont $0 \leq i < n_r - 1$ où n_r représente le nombre de segments de recherche dont nous avons besoin pour parcourir la distance d_r . Nous calculons n_r de la façon suivante : étant donnée l'épaisseur d'un segment de recherche $t_r = \frac{\max(a,b)}{\sqrt{a^2+b^2}}$ où $\frac{a}{b}$ la pente de la droite, alors $n_r = \frac{d_r}{t_r}$ (voir figure 4.7). L'initialisation se fait en définissant le premier segment de recherche $\mathcal{R}_k^0$ à partir des points $\mathcal{A}_k^{p1}$ et $\mathcal{A}_k^{p4}$.
Nous utilisons la définition de recherche directionnelle énoncée par Kerautret et Even. [KE09] qui garantit un parcours où tous les pixels de la zone sont parcourus une et une seule fois. Chaque segment de recherche est inscrit dans une droite discrète naïve (8-connexe) et obtenu par décalage horizontal ou vertical d'un pixel à partir de la droite précédente.
Étant donnée la nature d'un segment de recherche, c'est-à-dire un segment discret naïf (8-connexe), nous pouvons par extension considérer qu'il représente un profil d'intensité.
- **Zone d'intérêt** : une zone d'intérêt $\mathcal{I}_i^{l,m}$ représente un sous-segment d'un segment de recherche $\mathcal{R}_k^i$. La zone d'intérêt $\mathcal{I}_i^{l,m}$ commence au pixel d'indice l du segment de recherche $\mathcal{R}_k^i$ et se termine au pixel d'indice m .
De plus, nous pouvons, par définition, considérer une zone d'intérêt comme un profil d'intensité, tout comme nous l'avons fait pour un segment de recherche.

Nous avons maintenant à disposition plusieurs structures, nous pouvons donc nous intéresser plus en détail à notre proposition. Pour cela, nous allons décrire les différentes étapes de celle-ci à l'aide d'illustrations concrètes. Mais dans un premier temps, nous devons définir manuellement sur l'image référence les zones de recherche $\mathcal{A}_k$ (illustrées

en rouge sur la figure 4.8) ainsi que, pour chacune d'elles, la zone du tube qui fera office d'empreinte, que l'on nomme profil d'intensité référence $\mathcal{P}_{ref}^k$ (illustrée par les segments en bleus sur la figure 4.8).

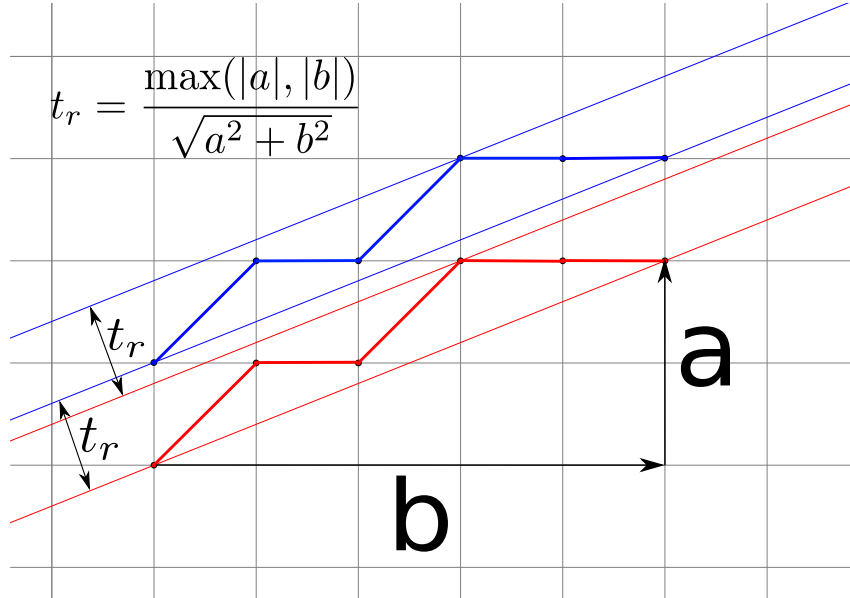


FIGURE 4.7 – Illustration d'une droite discrète naïve $D(2, 5, 0, 5)$ en rouge avec ses paramètres a et b . La valeur t_r permettant d'obtenir les segments $D(2, 5, 0, 5)$ (comme par exemple le segment discret en bleu) « parallèles » de celui-ci est aussi illustrée.

Étant donné que l'algorithme va s'exécuter de la même façon pour chacune des zones de recherche définies dans l'image référence, nous allons dans la suite considérer que nous avons défini une seule et unique zone de recherche $\mathcal{A}_0$.

4.2.2 Exploration d'une zone de recherche

À partir de la zone de recherche $\mathcal{A}_0$ et de son profil d'intensité référence $\mathcal{P}_{ref}^0$ associé, nous pouvons décrire l'algorithme général.

Parcours de la zone de recherche

La première étape (voir Algorithme 2) consiste à parcourir entièrement la zone de recherche $\mathcal{A}_0$ comme nous pouvons le voir sur la figure 4.9. Ce parcours consiste à partir

d'un segment de recherche $\mathcal{R}_0^0$ défini par les points $\mathcal{A}_k^{p1}$ et $\mathcal{A}_k^{p4}$ à scanner complètement la zone de recherche par translation successive dans la direction donnée par le vecteur $(\mathcal{A}_k^{p2} - \mathcal{A}_k^{p1})$ (en noir sur la figure 4.9).

Parcours des segments de recherche

La seconde étape (voir Algorithme 3) consiste, pour chacun des segments de recherche $\mathcal{R}_0^i$, à effectuer un parcours incrémental à l'aide d'une zone d'intérêt $\mathcal{I}_i^{l,m}$. Pour mieux visualiser ce parcours, mettons nous dans la situation où nous sommes au cinquantième segment de recherche ($\mathcal{R}_0^{50}$) représenté en bleu sur la figure 4.10 (a) dans la zone de recherche $\mathcal{A}_0$. Ce segment de recherche est, pour rappel, composé d'un nombre x de pixels qui forment une suite connexe. Nous pouvons donc indexer chacun de ces pixels de 0 à $x - 1$. De plus, nous avons également à notre disposition un profil d'intensité référence $\mathcal{P}_{ref}^0$ qui lui aussi est composé d'un nombre y de pixels.

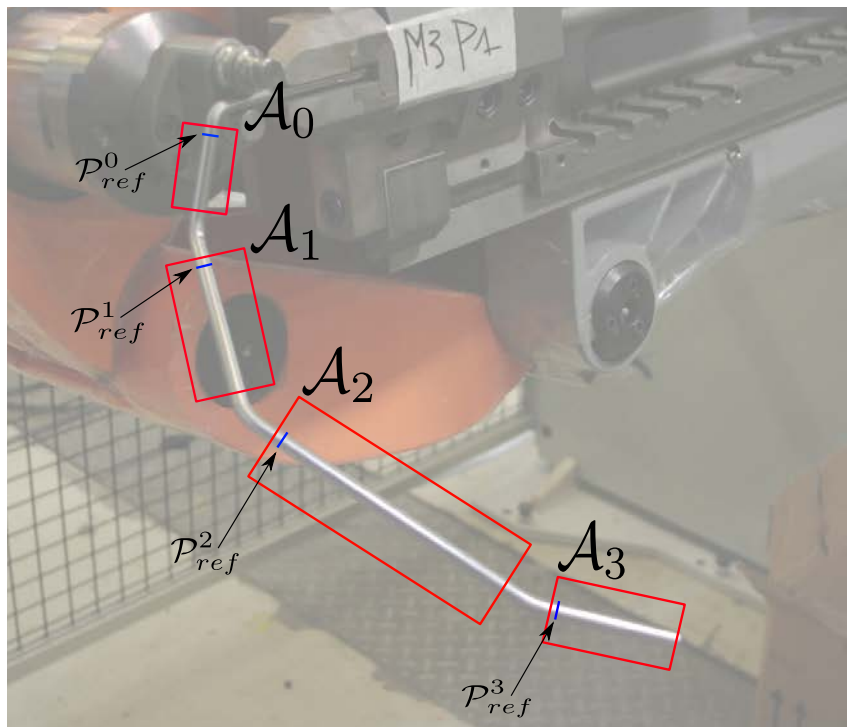


FIGURE 4.8 – Définition manuelle des différentes zones de recherche $\mathcal{A}_k$ en rouge et profils d'intensité $\mathcal{P}_{ref}^k$ associés en bleu.

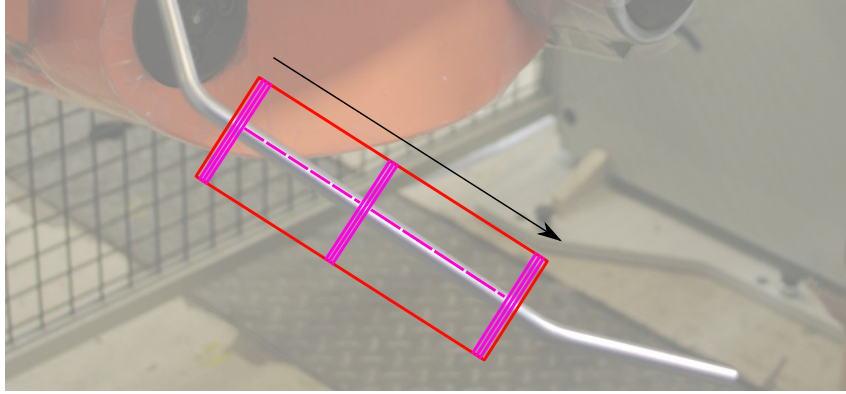


FIGURE 4.9 – Parcours de la zone de recherche $\mathcal{A}_0$ en fuchsia à l'aide d'une génération de segments de recherche $\mathcal{R}_0^i$ en rouge dans la direction du vecteur en noir.

Algorithme 2 : Algorithme principal de parcours de la zone de recherche $\mathcal{A}_0$ pour chercher le tube à partir du profil d'intensité référence $\mathcal{P}_{ref}$.

Données : $\mathcal{A}_0$ une zone de recherche définie par quatre points $(\mathcal{A}_0^{p1}, \mathcal{A}_0^{p2}, \mathcal{A}_0^{p3}, \mathcal{A}_0^{p4})$,
 $\mathcal{P}_{ref}$ un profil d'intensité référence

Résultat : $\mathcal{S}f_l$ et $\mathcal{S}f_m$ deux segments flous représentant les bords du tube dans la zone de recherche $\mathcal{A}_0$

```

1   $\mathcal{R}_0^0 \leftarrow \text{creationSegmentRecherche}(\mathcal{A}_0^{p1}, \mathcal{A}_0^{p4});$ 
2   $\text{direction} \leftarrow (\mathcal{A}_0^{p2} - \mathcal{A}_0^{p1});$ 
3   $\text{distance} \leftarrow \text{direction.norm}();$ 
4  /*  $\frac{a}{b}$  la pente du segment de recherche  $\mathcal{R}_0^0$  */
5   $t_r \leftarrow \frac{\max(a,b)}{\sqrt{a^2+b^2}};$ 
6   $\text{pas} \leftarrow \frac{\text{distance}}{t_r};$ 
7  /* rechercheTube  $\Rightarrow$  voir algorithme 3 */
8   $\text{result}[0] \leftarrow \text{rechercheTube}(\mathcal{R}_0^0, \mathcal{P}_{ref});$ 
9  pour  $i \leftarrow 1$  à pas faire
10  |    $\mathcal{R}_0^i \leftarrow \text{segmentRechercheSuivant}(\mathcal{R}_0^{i-1}, \text{direction.normalize}());$ 
11  |    $\text{result}[i] \leftarrow \text{rechercheTube}(\mathcal{R}_0^i, \mathcal{P}_{ref});$ 
12  $\mathcal{S}f_l, \mathcal{S}f_m \leftarrow \text{creationSegmentsFlous}(\text{result});$ 
```

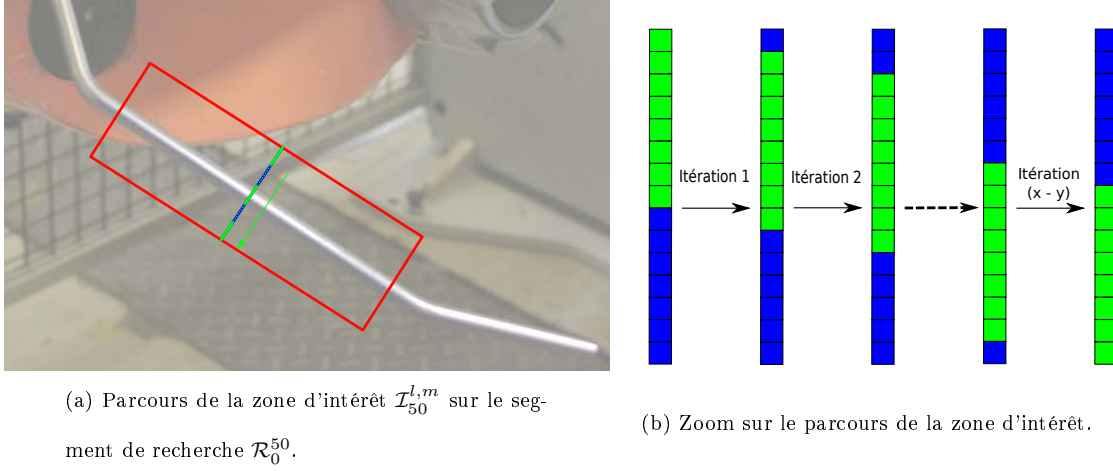


FIGURE 4.10 – Parcours du segment de recherche $\mathcal{R}_0^{50}$ par translation unitaire d’une zone d’intérêt $\mathcal{I}_{50}^{l,m}$ représenté en vert.

À partir de ces quelques rappels, nous définissons une zone d’intérêt $\mathcal{I}_{50}^{0,(y-1)}$. Cette notation indique que nous déclarons une zone sur le segment de recherche commençant à l’indice 0 et terminant à l’indice $(y - 1)$ avec y le nombre de pixels du profil d’intensité référence $\mathcal{P}_{ref}^0$. Cette zone est donc de taille équivalente, en terme de nombre de pixels, au profil d’intensité référence $\mathcal{P}_{ref}^0$. Comme nous pouvons le voir sur la figure 4.10 (a), le parcours consiste à déplacer notre zone d’intérêt en bleu incrémentalement sur le segment de recherche en vert en incrémentant d’une valeur de 1 les deux bornes (l, m) de la zone d’intérêt. Si l’on regarde la figure 4.10 (b), nous avons la zone d’intérêt en orange noté $\mathcal{I}_i^{0,7}$. Après une itération, la zone d’intérêt suivante devient $\mathcal{I}_i^{1,8}$, puis $\mathcal{I}_i^{2,9}$, etc. Après $(x - y) - 1$ itérations dans le cas de notre exemple où $x = 15$, nous avons $\mathcal{I}_i^{6,13}$ et le parcours s’arrête lorsque l’indice m de la zone d’intérêt est égal à $(x - y)$ soit dans notre exemple $\mathcal{I}_i^{7,14}$.

Sélection des zones d’intérêt

La troisième étape (voir Algorithme 4) représente la sélection de la « bonne » zone d’intérêt, représentant donc l’objet que l’on recherche, située sur chaque segment de recherche. C’est-à-dire que lorsque l’on parcourt un segment de recherche $\mathcal{R}_k^i$ à l’aide d’une zone d’intérêt $\mathcal{I}_i^{l,m}$, nous voulons connaître la position (l, m) sur le segment de recherche qui correspond le plus au profil d’intensité référence $\mathcal{P}_{ref}^k$. Pour cela, il nous faut un outil nous permettant, à partir d’un profil d’intensité référence et une zone d’intérêt, d’obtenir

Algorithme 3 : Algorithme qui permet de chercher la zone d'intérêt $\mathcal{I}^{l,m}$ sur le segment de recherche $\mathcal{R}$ qui obtient le plus petit score calculé par un MSE entre le profil d'intensité référence $\mathcal{P}_{ref}$ et les différentes zones d'intérêt.

Données : $\mathcal{R}$ un segment de recherche, $\mathcal{P}_{ref}$ un profil d'intensité référence

Résultat : Couple (l, m) qui représente les deux points extrémités de la zone d'intérêt, et qui a obtenue le plus petit score sur le segment de recherche $\mathcal{R}$

```

1   $(l, m) \leftarrow (0, \mathcal{P}_{ref}.size() - 1);$ 
2   $minMSE \leftarrow$  la valeur max représentable par un float/double;
3   $result \leftarrow (0, 0);$ 
4  répéter
5      /* On considère profil comme un tableau de valeurs d'intensité
        comme pour  $\mathcal{R}$ . */
6       $profil \leftarrow$  extractionProfil( $\mathcal{R}, l, m$ );
7       $mse =$  calculMSE( $\mathcal{P}_{ref}, profil$ );
8      si  $mse < minMSE$  alors
9           $minMSE \leftarrow mse;$ 
10          $result \leftarrow (l, m);$ 
11          $(l, m) \leftarrow (l + 1, m + 1);$ 
12 jusqu'à  $m! = \mathcal{R}.size();$ 
13 /* On considère que le couple result représente deux points,
    c'est-à-dire position dans l'image */
```

un score reflétant la distance entre ces deux entités.

Si l'on considère un profil d'intensité référence et une zone d'intérêt comme des vecteurs de taille n , nous pouvons évaluer leur distance à l'aide de la notion de *norme*. Il existe plusieurs types de normes mais nous nous intéresserons ici uniquement à la norme où $p = 1$ et $p = 2$ dans :

$$\|\vec{x}\|_p = (|x_1|^p + \dots + |x_n|^p)^{\frac{1}{p}} \quad (4.2)$$

Lorsque la norme est considérée comme une différence entre deux vecteurs (ou matrices),

Algorithme 4 : calculMSE— Algorithme qui calcule une *erreur quadratique moyenne (MSE)* entre deux profils d'intensité avec interpolation des valeurs

Données : $\mathcal{P}_0$ et $\mathcal{P}_1$ deux profils d'intensité sous forme de tableau de couples (x, y) , *interpolation* un booléen pour choisir si nous effectuons un MSE avec ou sans interpolation

Résultat : Distance entre $\mathcal{P}_0$ et $\mathcal{P}_1$

```

1 result ← 0.0;
2 si interpolation alors
3   pour i ← 0 à  $\mathcal{P}_0.size() - 1$  faire
4     /* IntensiteInterpolee est une fonction d'interpolation
       linéaire */
5     intensiteRef ← intensiteInterpolee( $\mathcal{P}_0$ ,  $\mathcal{P}_1[i].x$ );
6     result ← result + (intensiteRef -  $\mathcal{P}_1[i].y$ )2;
7 sinon
8   pour i ← 0 à  $\mathcal{P}_0.size() - 1$  faire
9     result ← result + ( $\mathcal{P}_0[i] - \mathcal{P}_1[i]$ )2;
10 result ←  $\frac{result}{\mathcal{P}_0.size()};$ 

```

nous pouvons définir cette distance de différentes façons. Pour la norme où $p = 1$, nous avons à disposition, par exemple, ces deux définitions :

$$SAD(\vec{x}, \vec{y}) = \|\vec{x} - \vec{y}\|_1 = \sum_{i=0}^{n-1} |x_i - y_i| \quad (4.3)$$

$$MAE(\vec{x}, \vec{y}) = \frac{1}{n} \|\vec{x} - \vec{y}\|_1 = \frac{1}{n} \sum_{i=0}^{n-1} |x_i - y_i| \quad (4.4)$$

où *SAD* correspond à la somme des différences absolues et *MAE* à l'erreur moyenne absolue. Pour la norme où $p = 2$:

$$SSD(\vec{x}, \vec{y}) = \|\vec{x} - \vec{y}\|_2^2 = \sum_{i=0}^{n-1} |x_i - y_i|^2 \quad (4.5)$$

$$MSE(\vec{x}, \vec{y}) = \frac{1}{n} \|\vec{x} - \vec{y}\|_2^2 = \frac{1}{n} \sum_{i=0}^{n-1} |x_i - y_i|^2 \quad (4.6)$$

où SSD correspond à la somme des différences au carré et MSE la moyenne arithmétique des carrés des différences. Pour notre cas, nous allons choisir la méthode du MSE qui va permettre d'obtenir un score moyen représentant la distance entre deux profils d'intensité pertinents. Étant donné que le MSE est une somme de différences au carré, cela permet de mettre en lumière les extrêmes contrairement à la méthode MAE qui assigne un poids équivalent à toutes les données. C'est-à-dire que lorsque nous avons une grande différence, nous allons accentuer cette différence mais, dans le cas contraire, la différence restera petite. Et c'est exactement ce que l'on recherche, puisque nous voulons obtenir au final la zone d'intérêt sur un segment de recherche qui obtient le score (ou la distance) le plus faible par rapport à un profil d'intensité référence.

Une fois que le parcours de tous les segments de recherche a été effectué et que sur chacun d'eux, une zone d'intérêt a été sélectionné, nous obtenons donc un ensemble de couple $(l, m)_i$ qui représente un intervalle sur un segment de recherche $\mathcal{R}_k^i$ (voir figure 4.11 (a) le couple de points en noir).

Construction des segments flous

À partir de cet ensemble de couples $(l, m)_i$, nous allons effectuer une reconnaissance de segment flou (voir Algorithme 5) afin d'estimer au mieux la direction du tube en considérant uniquement les couples proches de la zone du tube. Par exemple, sur le cas de la figure 4.11 (a) nous désirerions garder uniquement les couples en bleu et supprimer ceux en rouge. Pour cela, nous allons incrémentalement construire deux segments flous $\mathcal{S}f_l$ et $\mathcal{S}f_m$ en ajoutant au fur et à mesure les points l_i au segment flou $\mathcal{S}f_l$ et les points m_i à $\mathcal{S}f_m$. La construction des deux segments flous peut se faire en parallèle ou l'une après l'autre sans aucun changement au niveau du résultat obtenu.

Nous obtenons, après cette étape, deux segments flous qui doivent représenter, au plus près possible, les deux bords du segment droit du tube que nous analysons (figure 4.12 (a)). Mais si l'on regarde la figure 4.11 (b) où les premières zones d'intérêt ne sont pas correctes, nous obtiendrons au final deux segments flous très courts et dans la mauvaise direction (figure 4.12 (b)). Pour limiter ce type de problème, nous pourrions compléter

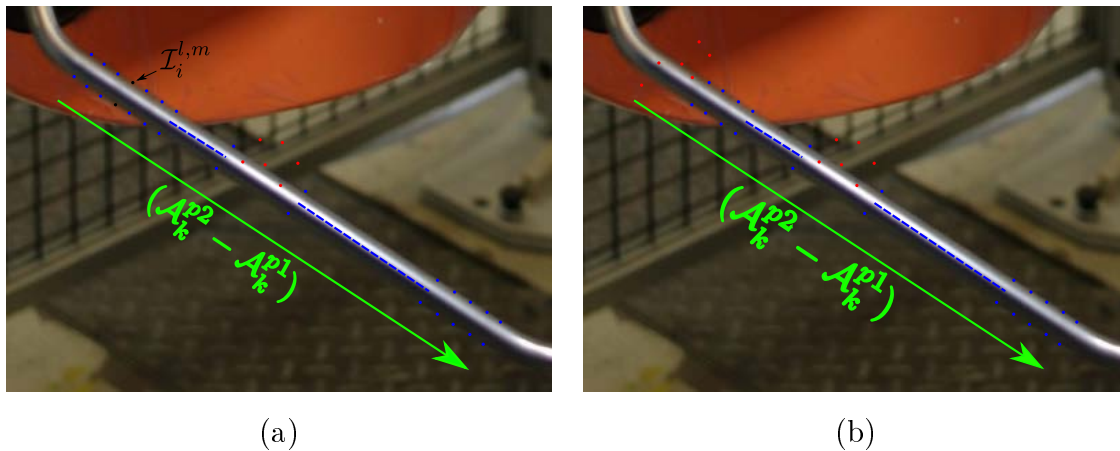


FIGURE 4.11 – Visualisation du résultat après choix des zones d'intérêt par comparaison de profils d'intensité (a) — Illustration du problème de reconnaissance des segments flous si les premières zones d'intérêt sont fausses (b).

l'algorithme de base en ajoutant un test sur le nombre de pixels composant les segments flous reconnus. En effet, si le nombre de pixels est inférieur à un certain seuil, nous pouvons effectuer une nouvelle reconnaissance en supprimant le premier couple $(l, m)_0$ pour étendre possiblement la reconnaissance. Si besoin, nous pouvons réitérer plusieurs fois ce processus en supprimant au fur et à mesure les premiers couples $(l, m)_i$. Le but final de cette opération est d'étendre la reconnaissance en supprimant les premiers couples $(l, m)_i$

Algorithme 5 : Algorithme de construction d'une paire de segments flous $(\mathcal{S}f_l, \mathcal{S}f_m)$ à partir d'une liste de couples de points

Données : $\mathcal{L}$ une liste de couples de points (l, m)

Résultat : $\mathcal{S}f_l$ et $\mathcal{S}f_m$ les deux segments flous construits à partir d'une liste de couples de points

```

1  $\mathcal{S}f_l \leftarrow \text{segmentFlou}();$ 
2  $\mathcal{S}f_m \leftarrow \text{segmentFlou}();$ 
3 pour  $i \leftarrow 0$  à  $\mathcal{L}.size()$  faire
4    $(l, m) \leftarrow \mathcal{L}[i];$ 
5    $\mathcal{S}f_l.\text{ajoutPoint}(l);$ 
6    $\mathcal{S}f_m.\text{ajoutPoint}(m);$ 
```

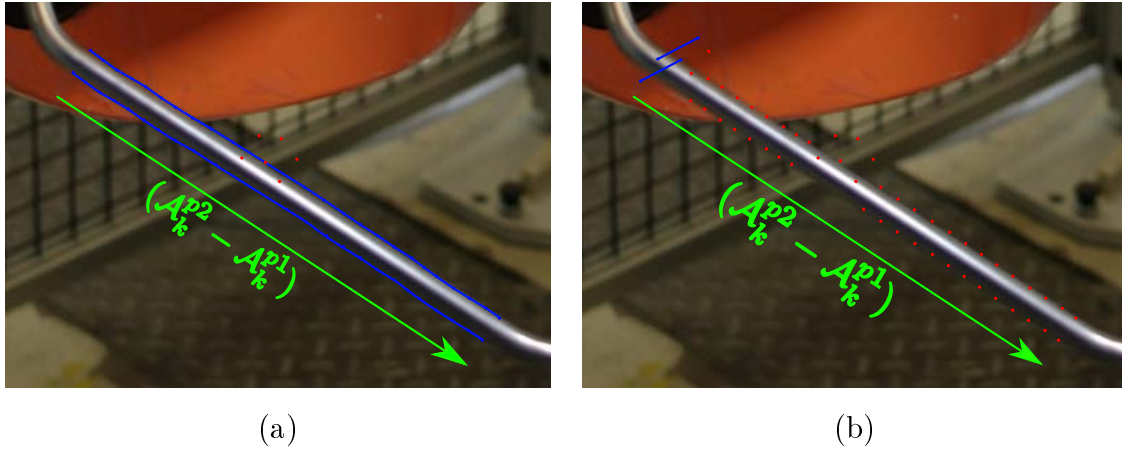


FIGURE 4.12 – Visualisation des deux segments flous résultant (a) — Visualisation des deux segments flous résultats lors d'une mauvaise détection dès le départ (b) — en rouge les points refusés par l'algorithme de construction des segments flous.

qui sont des mauvaises sélections. Pour ne pas alourdir l'algorithme 5, cette opération n'est pas stipulée.

Complexité

Comme nous avons pu le constater, l'algorithme dispose de plusieurs structures et la complexité dépend fortement de la taille de ces structures. Nous avons tout d'abord les zones de recherche qui sont composées de $L * M$ pixels qui sont tous consultés une unique fois lors du parcours incrémental à l'aide des segments de recherche. De plus, nous avons, pour chaque segment de recherche de M pixels, un parcours où l'on va consulter $(M - I_s + 1) * I_s$ pixels avec I_s le nombre de pixels de la zone d'intérêt.

4.3 Extension de la méthode : interpolation

Nous avons désormais à disposition un algorithme qui nous permet de scanner une zone de l'image et d'y repérer à l'intérieur un segment droit d'objet tubulaire à l'aide d'un profil d'intensité référence. Si l'on se réfère au postulat que nous avons énoncé au début de ce chapitre, l'intensité devrait être en moyenne la même tout le long de la droite orange sur la figure 4.13 allant dans la direction du tube représentée par deux droites

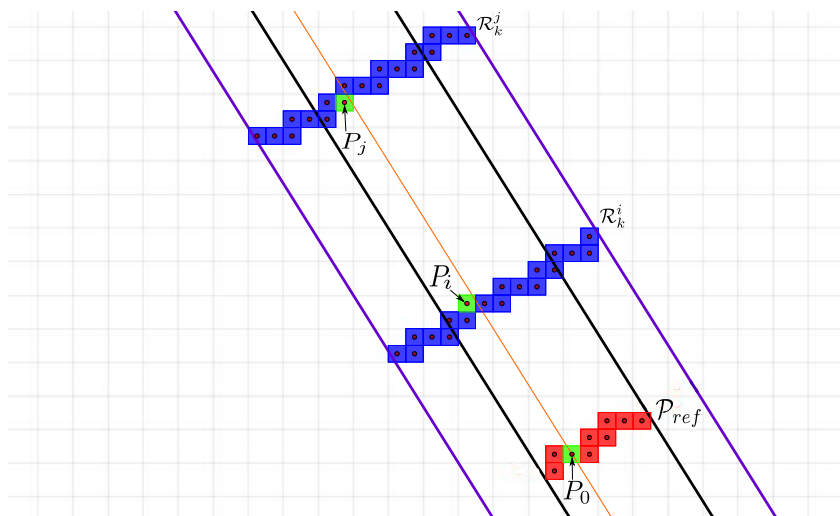


FIGURE 4.13 – Effet de la discrétisation de l'image

parallèles noires. Cela est vrai si l'on considère nos droites dans l'ensemble $\mathbb{R}$ mais lorsque l'on prend une photo de l'objet, la discrétisation du signal provoque une approximation de la scène par le processus d'échantillonnage et de quantification. Ce phénomène d'aliasing peut amener certaines questions sur l'exactitude du postulat de départ.

La figure 4.13 permet de mieux visualiser ce phénomène. Dans l'algorithme que nous venons d'exposer, nous pouvons être confronté à la situation suivante : lors de l'étape de comparaison, nous allons comparer le pixel P_0 du profil d'intensité référence P_{ref} avec les pixels P_i et P_j des segments de recherche R_k^i et R_k^j . Mais si l'on trace une droite continue passant par le centre du pixel P_0 allant dans la direction du tube, nous constatons que cette droite ne coupe pas les pixels P_i et P_j en leur centre. Cela signifie que l'on ne compare pas exactement des données équivalentes, ce qui pourrait casser notre postulat de départ. Nous allons pour cela proposer une modification de notre algorithme pour tenir compte de cet effet indésirable et nous verrons dans le chapitre suivant si cette modification théorique est réellement significative en pratique.

Modifications de l'algorithme sans interpolation

La première modification apportée à l'algorithme est d'obtenir un profil d'intensité référence en projetant chacun des pixels sur une droite perpendiculaire à la direction du

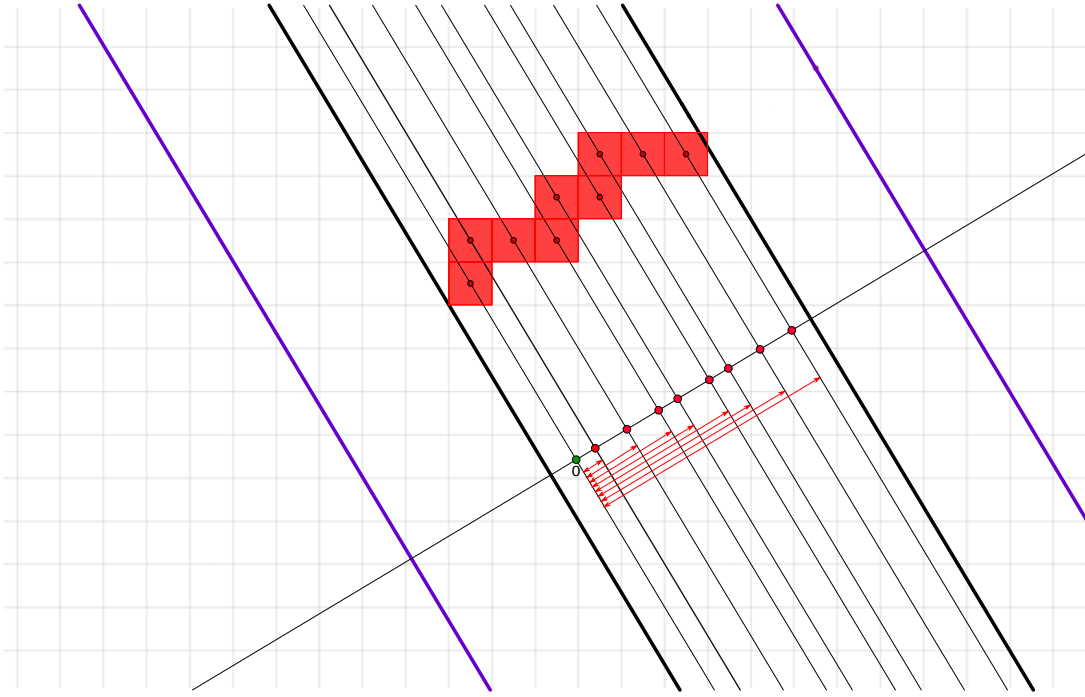


FIGURE 4.14 – Illustration de la méthode de projection des profils références.

tube. Nous nous retrouvons avec un profil d'intensité référence ayant les abscisses dans $\mathbb{R}$ en considérant le premier pixel comme l'origine (point vert sur la figure 4.14) et les autres comme leur distance euclidienne par rapport à celui-ci (flèches rouges sur la figure 4.14). Cela permet d'obtenir un profil d'intensité référence plus lisse de par son aspect continue plutôt que discrète.

L'algorithme modifié est relativement identique à celui présenté précédemment à l'exception de la génération du profil d'intensité référence que l'on vient d'explicitier et de la comparaison entre chaque zone d'intérêt et le profil d'intensité référence. Dans un premier temps, nous effectuons pour chacune des zones d'intérêt le même processus de projection (voir figure 4.15). Le calcul de distance à l'aide du *MSE* s'effectue maintenant de la façon suivante : comme nous pouvons le voir dans l'algorithme 4, nous avons en entrée deux profils d'intensité. Le calcul général du *MSE* reste équivalent à celui présenté précédemment excepté que pour chaque abscisse x (obtenue par projection) du profil d'intensité $\mathcal{P}_1$, nous récupérons à l'aide de la fonction *intensiteInterpoele* l'intensité du profil d'intensité $\mathcal{P}_0$ correspondant à cette même abscisse x . La fonction retourne directement l'intensité y

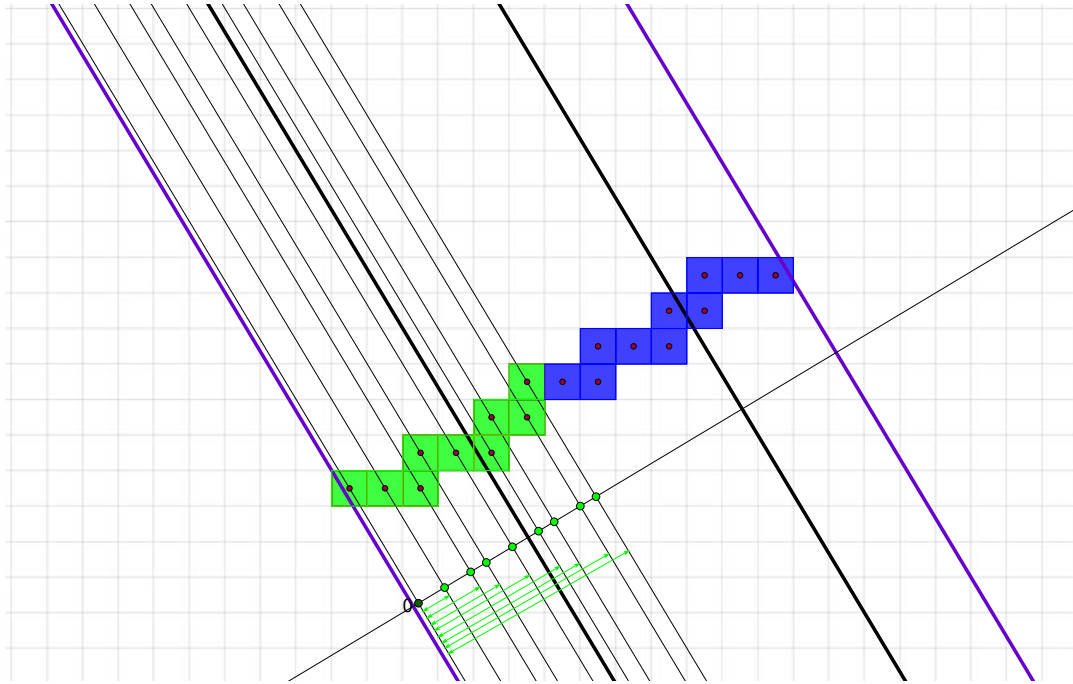


FIGURE 4.15 – Illustration de la méthode de projection des zones d'intérêt.

si le couple (x, y) existe sinon elle calcule par interpolation linéaire l'intensité y à l'aide des deux valeurs d'intensité correspondant aux abscisses qui encadrent x . Dans notre cas de figure, rappelons que $\mathcal{P}_0$ et $\mathcal{P}_1$ correspondent toujours respectivement au profil d'intensité référence et au profil d'intensité que l'on veut comparer. Cela veut dire que nous effectuons constamment l'interpolation sur le profil d'intensité référence (i.e. on utilise la fonction *intensiteInterpolee* uniquement avec le profil d'intensité référence).

Avec cette nouvelle stratégie, nous espérons obtenir de meilleures détections en améliorant la pertinence du MSE. C'est à dire obtenir des scores plus faibles lorsque la zone d'intérêt correspond bien à l'objet que l'on recherche et des scores plus élevés dans le cas contraire. Cela pourra permettre au final d'effectuer plus facilement une discrimination des différents scores obtenus pour un segment de recherche.

4.4 Analyse des deux algorithmes

Pour illustrer ces deux algorithmes, nous proposons d'effectuer une exécution de ceux-ci sur des données synthétiques. Cette étude permettra de voir comment chacun des algorithmes se comporte et ainsi pointer les différences de résultats. Pour cela, nous avons construit deux scénarios autour d'une même situation illustrée sur la figure 4.16 pour la méthode sans interpolation et la figure 4.17 pour la méthode avec interpolation.

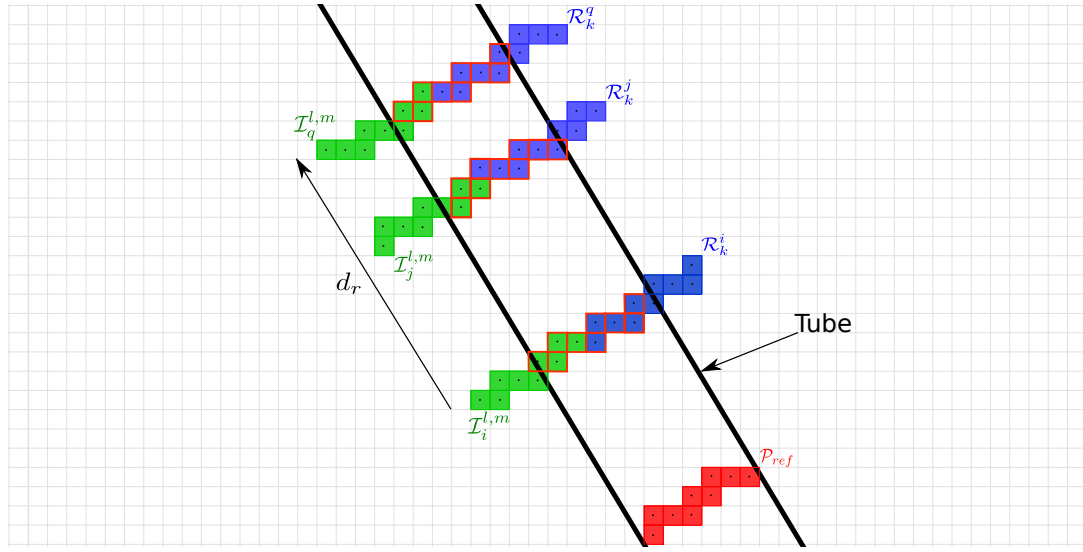


FIGURE 4.16 – Illustration utilisée pour l'étude de l'algorithme sans interpolation.

Chacun des scénarios se compose d'une grille de pixels où l'on dispose le tube que l'on veut détecter (deux droites parallèles continues en noir), une droite discrète représentant le profil d'intensité référence $\mathcal{P}_{ref}$ en rouge, composée de neuf pixels¹, ainsi que trois autres droites discrètes représentant trois segments de recherche $\mathcal{R}_k^i$ en bleu, se trouvant sur le parcours global de recherche dans la direction d_r . Sur chacun des segments de recherche, nous avons placé une zone d'intérêt $\mathcal{I}_i^{l,m}$ en vert. La différence entre la méthode sans et avec interpolation se situe au niveau de la reconstruction du profil d'intensité référence ainsi que des différentes zones d'intérêt. En effet, dans la version sans interpolation, nous considérons ces éléments comme des suites indicées à l'instar d'un tableau. Alors que pour la version avec interpolation, nous effectuons une projection du profil d'intensité référence

1. donc de neuf valeurs d'intensité

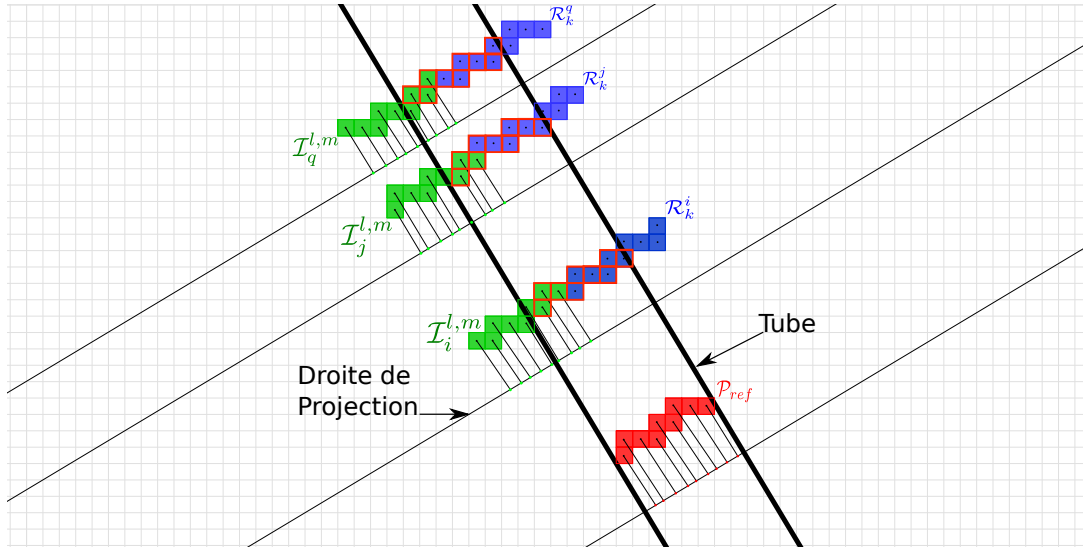


FIGURE 4.17 – Illustration utilisée pour l'étude de l'algorithme avec interpolation.

et des différentes zones d'intérêt sur une droite perpendiculaire au tube, comme nous pouvons le voir sur la figure 4.17. Pour finir, nous définissons l'intensité de chaque pixel de la grille de la façon suivante :

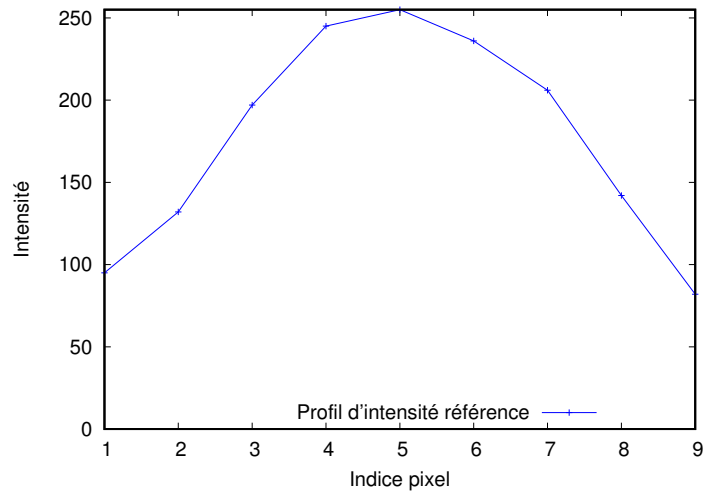
- Si le pixel n'est pas compris dans le tube, alors sa valeur est de 128.
- Si le pixel est dans le tube alors sa valeur est calculée à l'aide d'une gaussienne centrée sur le centre du tube (i.e. le centre du tube vaut donc 255). Sa valeur est déterminée en utilisant sa distance euclidienne entre lui-même et le bord gauche du tube. Ce choix d'utiliser une gaussienne comme profil d'intensité référence est justifié par le fait que cela nous génère un pic spéculaire entouré de différents niveaux de gris. Ce qui donne un cas d'étude intéressant et réaliste.

Nous pouvons voir ce que cela représente en regardant par exemple les deux profils d'intensité référence sur la figure 4.18.

Le principe général est d'effectuer un calcul de MSE entre le profil d'intensité référence et chacune des zones d'intérêt que l'on obtient par translation sur chacun des segments de recherche. Pour cela, nous créons un tableau de valeurs comme sur le tableau 4.1 où l'on enregistre les différentes valeurs d'intensité de chacun des pixels de chaque zone d'intérêt, ainsi que le score de MSE obtenu. Pour chacun des tableaux (4.1, 4.2, 4.3, 4.19, 4.20 et 4.21), nous sélectionnons l'itération qui a obtenue le score de MSE le plus faible. Nous

Indice	Intensité
1	95
2	132
3	197
4	245
5	255
6	236
7	206
8	142
9	82

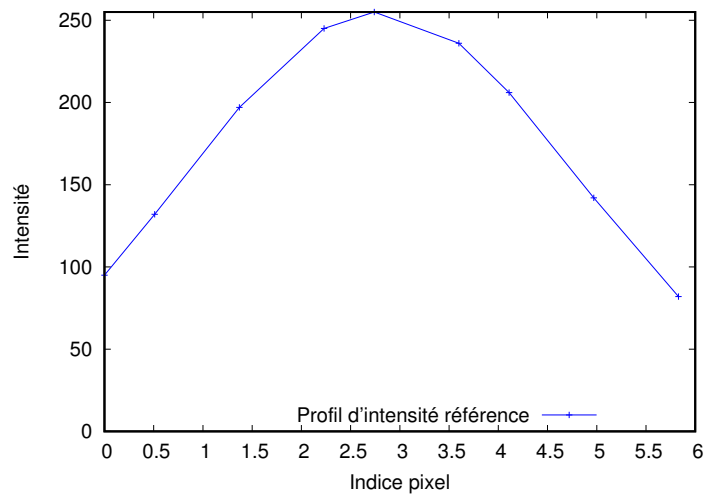
(a)



(b)

Distance	Intensité
0	95
0,51	132
1,37	197
2,23	245
2,74	255
3,6	236
4,11	206
4,97	142
5,83	82

(c)



(d)

FIGURE 4.18 – Graphique du profil d'intensité référence $\mathcal{P}_{ref}$ pour l'illustration de la méthode sans interpolation (a - b) et avec interpolation (c - d).

	Itération 1	Itération 2	Itération 3	Itération 4	Itération 5	Itération 6	Itération 7	Itération 8	Itération 9	Itération 10	Itération 11
1	128	128	128	128	128	64	119	158	219	253	253
2	128	128	128	128	64	119	158	219	253	253	217
3	128	128	128	64	119	158	219	253	253	217	155
4	128	128	64	119	158	219	253	253	217	155	117
5	128	64	119	158	219	253	253	217	155	117	62
6	64	119	158	219	253	253	217	155	117	62	128
7	119	158	219	253	253	217	155	117	62	128	128
8	158	219	253	253	217	155	117	62	128	128	128
9	219	253	253	217	155	117	62	128	128	128	128
MSE	10206,89	11911	11659,78	8569,22	3790,44	570,56	643,44	4353,33	9016,22	11757,89	11960,67

TABLE 4.1 – Résultats obtenus après parcours de la zone de recherche $\mathcal{I}_i^{l,m}$ sur le segment de recherche $\mathcal{R}_k^i$ avec la méthode sans interpolation.

pouvons visualiser ce choix par un encadré rouge de la colonne sur les différents tableaux et leur correspondance sur les figures 4.16 et 4.17 à travers un encadrement rouge des pixels appartenant à la zone d'intérêt sélectionnée.

	Itération 1	Itération 2	Itération 3	Itération 4	Itération 5	Itération 6	Itération 7	Itération 8	Itération 9	Itération 10	Itération 11
1	128	128	128	128	128	55	107	145	208	237	255
2	128	128	128	128	55	107	145	208	237	255	227
3	128	128	128	55	107	145	208	237	255	227	195
4	128	128	55	107	145	208	237	255	227	195	129
5	128	55	107	145	208	237	255	227	195	129	71
6	55	107	145	208	237	255	227	195	129	71	128
7	107	145	208	237	255	227	195	129	71	128	128
8	145	208	237	255	227	195	129	71	128	128	128
9	208	237	255	227	195	129	71	128	128	128	128
MSE	10459	12033,11	12345,44	9772,44	5524,78	1382,44	110	2836,33	7007,56	10021,11	11333,44

TABLE 4.2 – Résultats obtenus après parcours de la zone de recherche $\mathcal{I}_j^{l,m}$ sur le segment de recherche $\mathcal{R}_k^j$ avec la méthode sans interpolation.

	Itération 1	Itération 2	Itération 3	Itération 4	Itération 5	Itération 6	Itération 7	Itération 8	Itération 9	Itération 10	Itération 11
1	128	128	128	128	128	73	107	171	208	250	250
2	128	128	128	128	73	107	171	208	250	250	227
3	128	128	128	73	107	171	208	250	250	227	169
4	128	128	73	107	171	208	250	250	227	169	105
5	128	73	107	171	208	250	250	227	169	105	71
6	73	107	171	208	250	250	227	169	105	71	128
7	107	171	208	250	250	227	169	105	71	128	128
8	171	208	250	250	227	169	105	71	128	128	128
9	208	250	250	227	169	105	71	128	128	128	128
MSE	9863,44	11458,33	11274,56	8665,56	4142,33	563,78	530,67	4113	8324,44	11416,22	11927,78

TABLE 4.3 – Résultats obtenus après parcours de la zone de recherche $\mathcal{I}_q^{l,m}$ sur le segment de recherche $\mathcal{R}_k^q$ avec la méthode sans interpolation.

Itération 4				Itération 5			
Distances	Intensités	Interpolation	Nouvelle Référence	Distances	Intensités	Interpolation	Nouvelle Référence
0	128	OK	95	0	128	OK	95
0,86	128	Interpolation	158	0,51	64	OK	132
1,37	64	OK	197	1,37	119	OK	197
2,23	119	OK	245	1,88	158	Interpolation	225
2,74	158	OK	255	2,74	219	OK	255
3,6	219	OK	236	3,6	253	OK	236
4,46	253	Interpolation	180	4,11	253	OK	206
4,97	253	OK	142	4,97	217	OK	142
5,83	217	OK	82	5,83	155	OK	82
9014,11				3448,22			
8569,22				3790,44			
Itération 6				Itération 7			
Distances	Intensités	Interpolation	Nouvelle Référence	Distances	Intensités	Interpolation	Nouvelle Référence
0	64	OK	95	0	119	OK	95
0,86	119	Interpolation	158	0,51	158	OK	132
1,37	158	OK	197	1,37	219	OK	197
2,23	219	OK	245	2,23	253	OK	245
3,09	253	Interpolation	247	2,74	253	OK	255
3,6	253	OK	236	3,6	217	OK	236
4,46	217	Interpolation	180	4,46	155	Interpolation	180
5,32	155	Interpolation	118	4,97	117	OK	142
5,83	117	OK	82	5,83	62	OK	82
996,33				423,89			
570,56				643,44			
Itération 8				Itération 9			
Distances	Intensités	Interpolation	Nouvelle Référence	Distances	Intensités	Interpolation	Nouvelle Référence
0	158	OK	95	0	219	OK	95
0,86	219	Interpolation	158	0,86	253	Interpolation	158
1,72	253	Interpolation	218	1,37	253	OK	197
2,23	253	OK	245	2,23	217	OK	245
3,09	217	Interpolation	247	3,09	155	Interpolation	247
3,95	155	Interpolation	215	3,6	117	OK	236
4,46	117	Interpolation	180	4,46	62	Interpolation	180
5,32	62	Interpolation	118	4,97	128	OK	142
5,83	128	OK	82	5,83	128	OK	82
2522,22				7464,67			
4353,33				9016,22			
Itération 10				Itération 11			
Distances	Intensités	Interpolation	Nouvelle Référence	Distances	Intensités	Interpolation	Nouvelle Référence
0	253	OK	95	0	253	OK	95
0,51	253	OK	132	0,86	217	Interpolation	158
1,37	217	OK	197	1,72	155	Interpolation	218
2,23	155	OK	245	2,23	117	OK	245
2,74	117	OK	255	3,09	62	Interpolation	247
3,6	62	OK	236	3,6	128	OK	236
4,11	128	OK	206	4,46	128	Interpolation	180
4,97	128	OK	142	5,32	128	Interpolation	118
5,83	128	OK	82	5,83	128	OK	82
11757,9				11067,4			
11757,89				11960,67			

FIGURE 4.19 – Résultats obtenus après parcours de la zone de recherche $\mathcal{I}_i^{l,m}$ sur le segment de recherche $\mathcal{R}_k^i$ avec la méthode avec interpolation — Les MSE en vert correspondent aux nouveaux résultats obtenus avec la méthode avec interpolation.

Itération 4				Itération 5			
Distances	Intensités	Interpolation	Nouvelle Référence	Distances	Intensités	Interpolation	Nouvelle Référence
0	128	OK	95	0	128	OK	95
0,51	128	OK	132	0,86	55	Interpolation	158
1,37	55	OK	197	1,72	107	Interpolation	218
2,23	107	OK	245	2,23	145	OK	245
2,74	145	OK	255	3,09	208	Interpolation	247
3,6	208	OK	236	3,6	237	OK	236
4,11	237	OK	206	4,46	255	Interpolation	180
4,97	255	OK	142	5,32	227	Interpolation	118
5,83	227	OK	82	5,83	195	OK	82
9772,44				7312,89			
9772,44				5524,78			
Itération 6				Itération 7			
Distances	Intensités	Interpolation	Nouvelle Référence	Distances	Intensités	Interpolation	Nouvelle Référence
0	55	OK	95	0	107	OK	95
0,86	107	Interpolation	158	0,51	145	OK	132
1,37	145	OK	197	1,37	208	OK	197
2,23	208	OK	245	1,88	237	Interpolation	225
2,74	237	OK	255	2,74	255	OK	255
3,6	255	OK	236	3,6	227	OK	236
4,46	227	Interpolation	180	4,11	195	OK	206
4,97	195	OK	142	4,97	129	OK	142
5,83	129	OK	82	5,83	71	OK	82
1798,44				118,89			
1382,44				110			
Itération 8				Itération 9			
Distances	Intensités	Interpolation	Nouvelle Référence	Distances	Intensités	Interpolation	Nouvelle Référence
0	145	OK	95	0	208	OK	95
0,86	208	Interpolation	158	0,51	237	OK	132
1,37	237	OK	197	1,37	255	OK	197
2,23	255	OK	245	2,23	227	OK	245
3,09	227	Interpolation	247	2,74	195	OK	255
3,6	195	OK	236	3,6	129	OK	236
4,46	129	Interpolation	180	4,46	71	Interpolation	180
5,32	71	Interpolation	118	4,97	128	OK	142
5,83	128	OK	82	5,83	128	OK	82
1745,22				6302,67			
2836,33				7007,56			
Itération 10				Itération 11			
Distances	Intensités	Interpolation	Nouvelle Référence	Distances	Intensités	Interpolation	Nouvelle Référence
0	237	OK	95	0	255	OK	95
0,86	255	Interpolation	158	0,86	227	Interpolation	158
1,72	227	Interpolation	218	1,37	195	OK	197
2,23	195	OK	245	2,23	129	OK	245
3,09	129	Interpolation	247	3,09	71	Interpolation	247
3,95	71	Interpolation	215	3,6	128	OK	236
4,46	128	Interpolation	180	4,46	128	Interpolation	180
5,32	128	Interpolation	118	4,97	128	OK	142
5,83	128	OK	82	5,83	128	OK	82
7970,44				10164,1			
10021,11				11333,44			

FIGURE 4.20 – Résultats obtenus après parcours de la zone de recherche $\mathcal{I}_j^{l,m}$ sur le segment de recherche $\mathcal{R}_k^j$ avec la méthode avec interpolation — Les MSE en vert correspondent aux nouveaux résultats obtenus avec la méthode avec interpolation.

Itération 4				Itération 5			
Distances	Intensités	Interpolation	Nouvelle Référence	Distances	Intensités	Interpolation	Nouvelle Référence
0	128	OK	95	0	128	OK	95
0,86	128	Interpolation	158	0,86	73	Interpolation	158
1,72	73	Interpolation	218	1,37	107	OK	197
2,23	107	OK	245	2,23	171	OK	245
3,09	171	Interpolation	247	2,74	208	OK	255
3,6	208	OK	236	3,6	250	OK	236
4,46	250	Interpolation	180	4,46	250	Interpolation	180
5,32	250	Interpolation	118	4,97	227	OK	142
5,83	227	OK	82	5,83	169	OK	82
10218,6				4887,67			
8665,56				4142,33			
Itération 6				Itération 7			
Distances	Intensités	Interpolation	Nouvelle Référence	Distances	Intensités	Interpolation	Nouvelle Référence
0	73	OK	95	0	107	OK	95
0,51	107	OK	132	0,86	171	Interpolation	158
1,37	171	OK	197	1,37	208	OK	197
1,88	208	Interpolation	225	2,23	250	OK	245
2,74	250	OK	255	3,09	250	Interpolation	247
3,6	250	OK	236	3,6	227	OK	236
4,11	227	OK	206	4,46	169	Interpolation	180
4,97	169	OK	142	5,32	105	Interpolation	118
5,83	105	OK	82	5,83	71	OK	82
443,778				106,67			
563,78				530,67			
Itération 8				Itération 9			
Distances	Intensités	Interpolation	Nouvelle Référence	Distances	Intensités	Interpolation	Nouvelle Référence
0	171	OK	95	0	208	OK	95
0,51	208	OK	132	0,86	250	Interpolation	158
1,37	250	OK	197	1,72	250	Interpolation	218
2,23	250	OK	245	2,23	227	OK	245
2,74	227	OK	255	3,09	169	Interpolation	247
3,6	169	OK	236	3,95	105	Interpolation	215
4,46	105	Interpolation	180	4,46	71	Interpolation	180
4,97	71	OK	142	5,32	128	Interpolation	118
5,83	128	OK	82	5,83	128	OK	82
3604,56				6095,78			
4113				8324,44			
Itération 10				Itération 11			
Distances	Intensités	Interpolation	Nouvelle Référence	Distances	Intensités	Interpolation	Nouvelle Référence
0	250	OK	95	0	250	OK	95
0,86	250	Interpolation	158	0,51	227	OK	132
1,37	227	OK	197	1,37	169	OK	197
2,23	169	OK	245	2,23	105	OK	245
3,09	105	Interpolation	247	2,74	71	OK	255
3,6	71	OK	236	3,6	128	OK	236
4,46	128	Interpolation	180	4,11	128	OK	206
4,97	128	OK	142	4,97	128	OK	142
5,83	128	OK	82	5,83	128	OK	82
10174,4				11927,8			
11416,22				11927,78			

FIGURE 4.21 – Résultats obtenus après parcours de la zone de recherche $\mathcal{I}_q^{l,m}$ sur le segment de recherche $\mathcal{R}_k^q$ avec la méthode avec interpolation — Les MSE en vert correspondent aux nouveaux résultats obtenus avec la méthode avec interpolation.

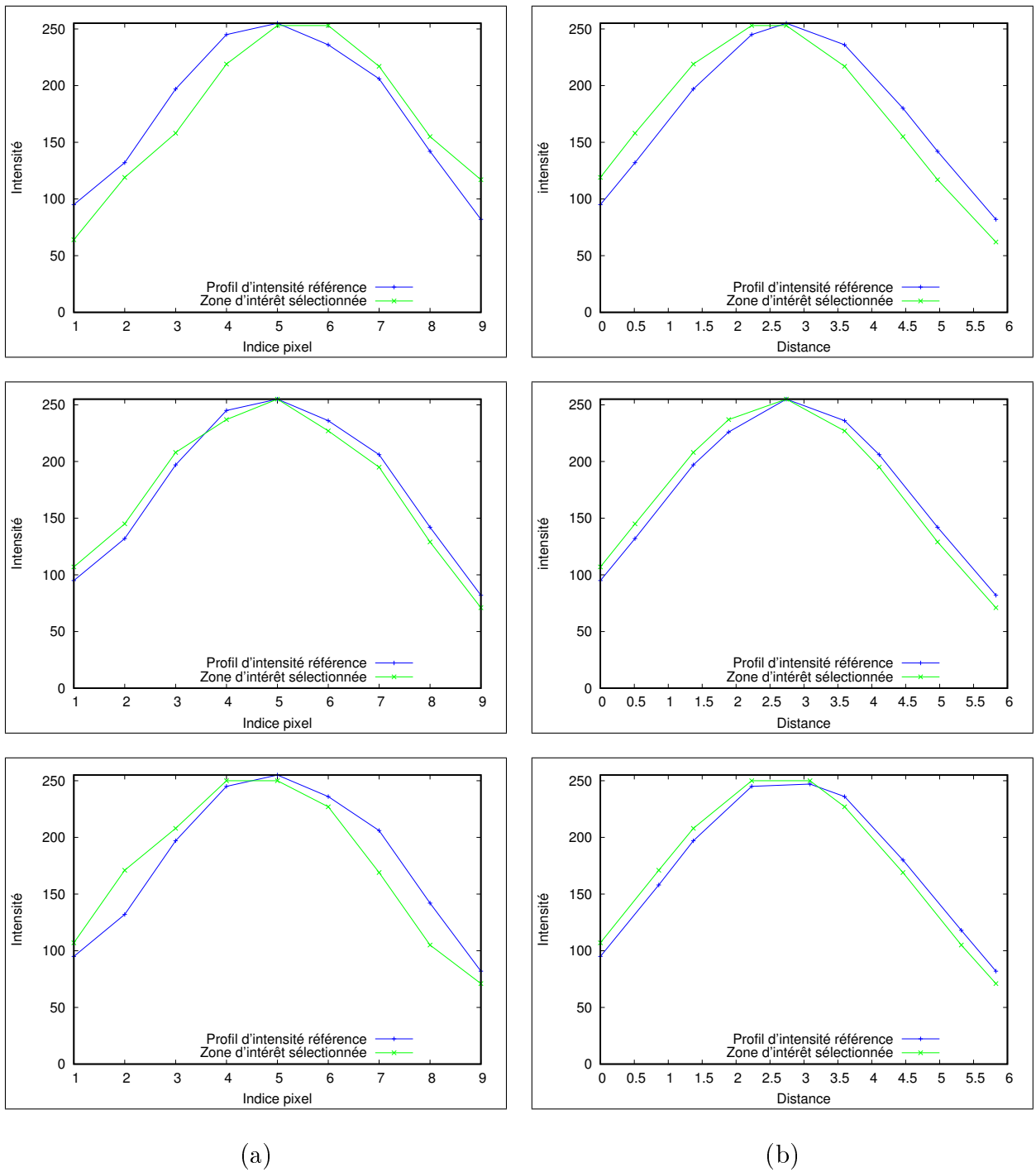


FIGURE 4.22 – Graphiques représentant le profil d'intensité référence et les profils d'intensité représentant les zones d'intérêt sélectionnées $\{\mathcal{I}_i^{l,m}, \mathcal{I}_j^{l,m}, \mathcal{I}_q^{l,m}\}$ sur les segments de recherche $\{\mathcal{R}_k^i, \mathcal{R}_k^j, \mathcal{R}_k^q\}$ avec la méthode sans (a) et avec (b) interpolation.

Si l'on observe les différents graphiques (voir figure 4.22 (a) et 4.22 (b)) comparant les zones d'intérêt sélectionnées et le profil d'intensité référence, nous obtenons bien une concordance relativement proche et cela se matérialise par une sélection correcte des pixels appartenant au tube (voir figure 4.16 et 4.17). Nous pouvons aussi constater que la version avec interpolation peut produire des concordances plus prononcées sans pour autant vraiment influencer le résultat qui est déjà exact avec la méthode sans interpolation sauf pour le segment $\mathcal{R}_k^i$ (voir figure 4.19) où la méthode avec interpolation ne sélectionne pas la même zone d'intérêt que la méthode sans et paraît être visuellement un meilleur choix. De plus, si l'on regarde de plus près les résultats de MSE obtenus par interpolation (en vert sur les figures 4.19, 4.20 et 4.21) et ceux sans interpolation (en noir sur les mêmes figures), nous pouvons constater que le MSE est plus discriminant pour la version avec interpolation que sans. Cela pourrait, en pratique, être un avantage pour la méthode avec interpolation lors de l'analyse de certaines images où par exemple le fond n'est pas très éloigné de l'intensité du tube que l'on recherche. Nous verrons dans le chapitre suivante, si cette avantage théorique se concrétise sur notre ensemble d'images en milieu industriel et à quel prix en terme de complexité ou temps de calcul.

4.5 Extension aux images non industrielles et limites

Avant de passer au chapitre suivant pour étudier notre méthode sur des images industrielles, nous allons montrer quelques résultats sur des images quelconques prises dans un contexte de détection de structures linéaires épaisses.

Pour cela, nous avons exécuté notre algorithme sur deux images dans un milieu urbain (figures 4.23 et 4.24). Sur la figure 4.23, nous pouvons voir que la détection s'effectue très bien dans la « zone 1 » alors que dans la « zone 2 » le résultat reste pertinent au départ de la détection, puis cela se dégrade. Ce problème de détection peut s'expliquer dans un premier temps par un changement de l'épaisseur de la zone que l'on doit détecter. Si l'épaisseur au cours de la détection change, notre profil d'intensité référence n'est plus valide. Dans un second temps, nous pouvons aussi constater qu'il y a des changements d'intensité le long de la « zone 2 » car nous sommes sur une reconnaissance d'une structure texturée contrairement aux objets tubulaires métalliques. Ce changement d'intensité implique, comme dans l'observation précédente, que le profil d'intensité référence n'est

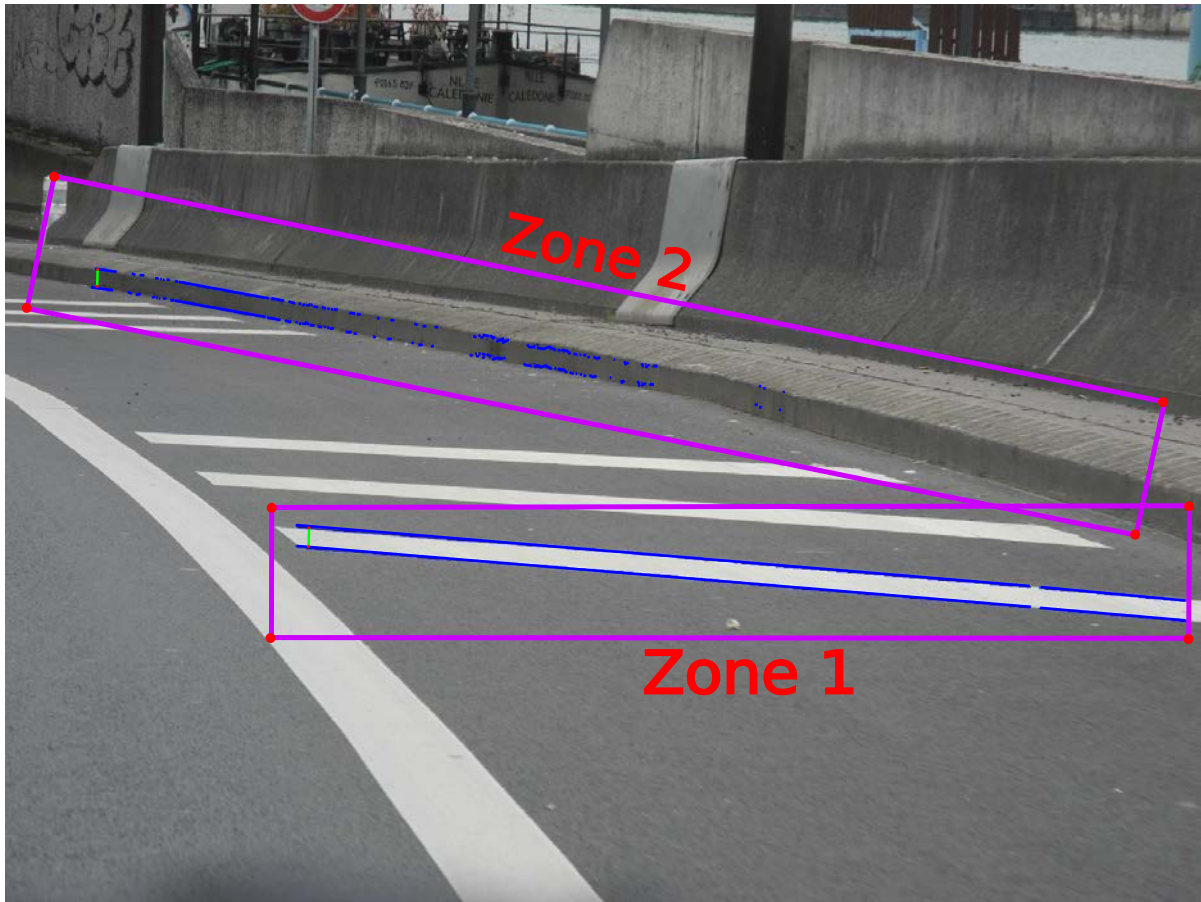


FIGURE 4.23 – Résultats obtenus sur une image urbaine — en vert le profil d'intensité référence, en violet la zone de recherche et en bleu le résultat de la détection.

plus valide. Ces différentes remarques s'appliquent aussi sur les résultats obtenus sur la figure 4.24. Nous remarquons très bien que dans la « zone 4 », par exemple, la détection s'effectue difficilement à cause d'un changement régulier de l'intensité le long de la zone de recherche. Il est intéressant de remarquer que dans la « zone 3 » la détection reste relativement bonne sur la totalité de la zone malgré une épaisseur de la structure à détecter très petite (i.e. un profil d'intensité référence avec seulement quatre valeurs).

Pour palier à ces différents problèmes, nous pouvons mettre en avant certaines perspectives qui ne seront bien entendu pas utilisées dans le chapitre suivant. Pour le problème de changement d'intensité, nous pouvons imaginer la possibilité d'utiliser plusieurs profils d'intensité références pour une même zone de recherche et effectuer un changement au

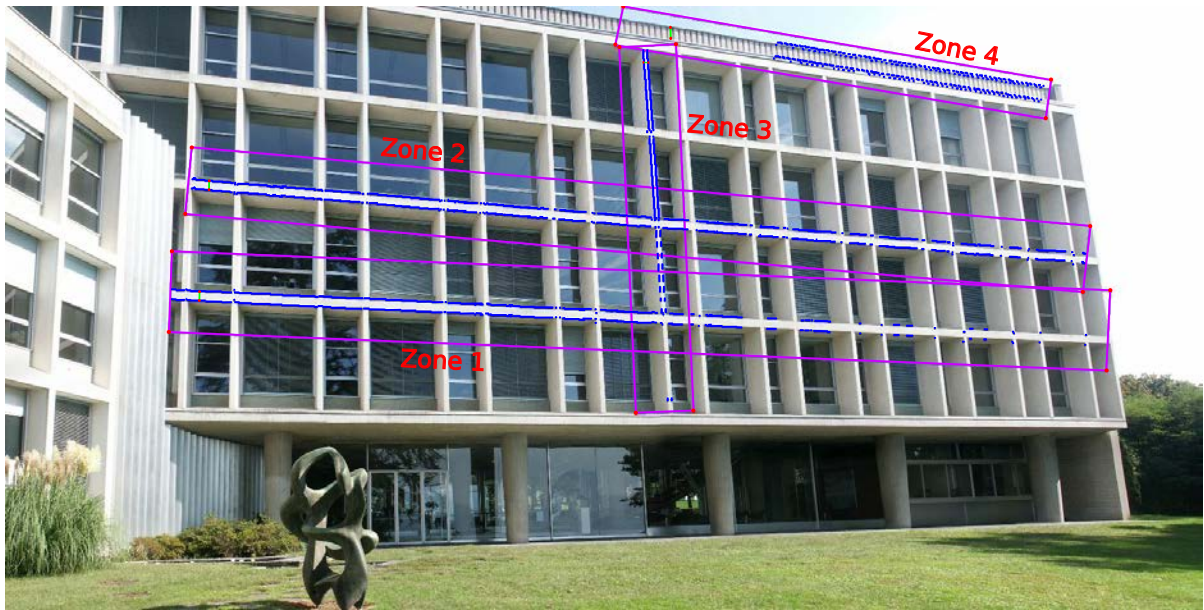


FIGURE 4.24 – Résultats obtenus sur un bâtiment — en vert le profil d'intensité référence, en violet la zone de recherche et en bleu le résultat de la détection.

cours du parcours de celle-ci. En ce qui concerne le problème sur le changement d'épaisseur, cela reste plus délicat car nous devons avoir des informations sur les transformations géométriques qui ont été effectuées lors de l'acquisition de l'image (e.g. projection perspective). Ces informations pourraient, par exemple, permettre d'effectuer un ajustement de la taille du profil d'intensité référence par rapport à l'endroit où l'on se situe dans la zone de recherche.

Nous avons maintenant à disposition deux versions de notre algorithme et nous allons confronter ces deux versions à une suite de photos contenant une pièce tubulaire à extraire de son milieu industriel. Nous pourrions ainsi étudier leur comportement et conclure sur l'intérêt de la version par interpolation sachant que cette dernière version sera par nature plus coûteuse en temps de calcul du fait de l'utilisation d'une interpolation linéaire à chaque utilisation du MSE ainsi que le besoin d'effectuer des projections des pixels sur une droite. Nous pourrions aussi conclure sur la pertinence des résultats obtenus avec notre méthode par rapport aux différentes méthodes étudiées dans le [Chapitre 2](#).

Chapitre 5

Validation & intégration industrielle

Sommaire

5.1	Validation	98
5.1.1	Résultats obtenus avec les critères respectés	98
5.1.2	Résultats obtenus avec les critères non respectés	100
5.1.3	Résultats avec translation des zones de recherche	105
5.2	Intégration industrielle	113
5.3	Description d'un premier prototype	117

DANS ce chapitre, nous allons dans un premier temps tester notre méthode sur un ensemble d'images reproduisant typiquement les images que l'on pourra rencontrer lors d'une utilisation industrielle. Pour cela, nous allons exécuter notre méthode sur les images « test » utilisées lors du [Chapitre 2](#) (Figures [2.1](#) et [2.2](#)). Nous effectuerons aussi des mesures de temps d'exécution, des visualisations de l'évolution du MSE au cours de l'algorithme, ainsi que des mesures d'angles entre la direction détectée et la direction théorique du tube.

Pour cette validation, nous nous plaçons en premier lieu dans une situation comportant ces différents critères :

- Chaque zone de recherche ne contient qu'une zone droite du tube pour respecter la garantie qu'en moyenne, l'intensité est constante le long d'un tube droit pour un point de vue fixe.
- Un seul profil d'intensité référence par zone de recherche.

- Pas de modification de l'éclairage entre l'image d'extraction des profils d'intensité références et sur laquelle l'algorithme est exécuté.

Ces critères permettront de mettre en avant le comportement de l'algorithme lorsque les conditions sont dites « idéales ». Après avoir étudié les résultats obtenus à l'aide de notre algorithme dans les conditions énoncées précédemment, nous continuerons notre analyse en effectuant des tests en dehors de ces conditions. Nous effectuerons des essais, par exemple, sur les images lorsque les zones de recherche ne contiennent que partiellement ou pas du tout le tube.

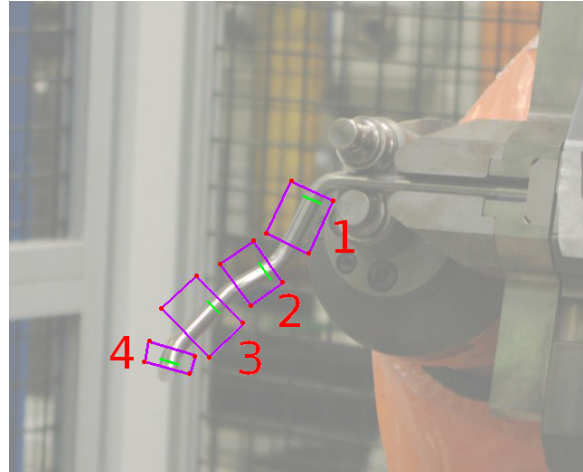
Nous terminerons ce chapitre par une présentation de l'intégration de la méthode au sein d'un processus de contrôle qualité de pièces manufacturées par des machines-outils conçues par l'entreprise Numalliance dans laquelle le travail de cette thèse a été effectué. Le travail dans ce chapitre a été publié [AKDRE15] et présenté à la conférence IWCIA 2015 et a donné lieu à une version journal en cours de validation.

5.1 Validation

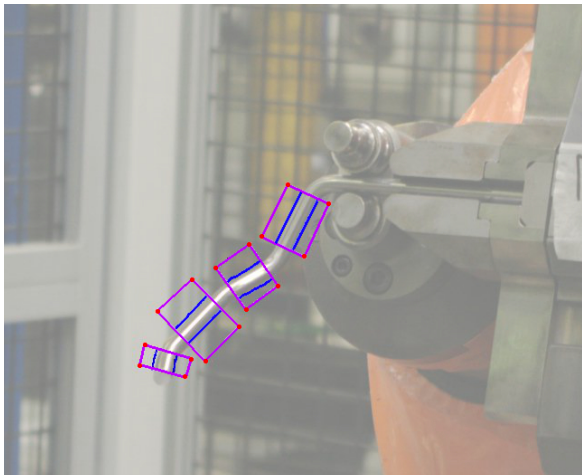
Pour commencer notre validation, nous allons définir sur chacune des images « test » les zones de recherche ainsi que les profils d'intensité références. À partir de ces données définies pour chacune des images (voir figures 5.1 (a), 5.5 (a), 5.2 (a), 5.3 (a) et 5.4 (a)), nous exécutons notre algorithme dans sa version sans et avec interpolation. Nous commenterons tout d'abord les résultats des expériences respectant nos critères, puis ceux obtenus lorsque nous nous en éloignons.

5.1.1 Résultats obtenus avec les critères respectés

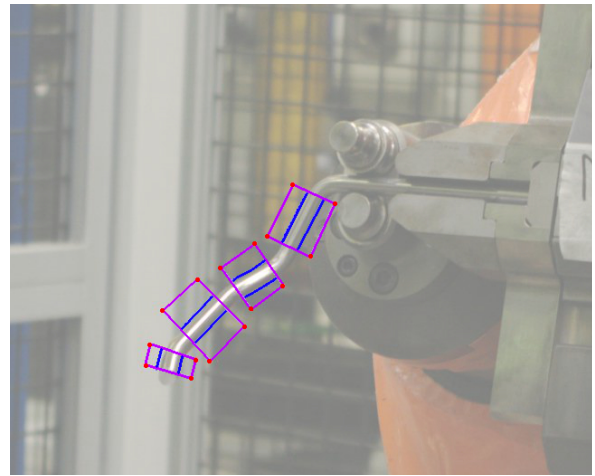
Si nous regardons les résultats obtenus sur chacun des modèles et plus particulièrement les zones de recherche respectant les critères, comme par exemple la totalité de la figure 5.1, nous obtenons pour chacune des zones un résultat très proche des contours réels du tube. Cette observation peut être aussi faite sur les figures 5.2 et 5.3. Nous pouvons aussi constater que nous obtenons les résultats attendus lorsque nous effectuons une reconnaissance d'un tube se trouvant dans un environnement contrôlé en le plaçant sur un « fond vert » (figure 5.4).



(a)



(b)



(c)

FIGURE 5.1 – Résultats de l'algorithme sur le modèle 1 — (a) Initialisation de l'algorithme — (b) version sans interpolation — (c) version avec interpolation.

Nous pouvons aussi noter que parfois, comme nous pouvons le voir sur la figure 5.3 (b) dans la zone de recherche n°3, certaines oscillations d'un pixel apparaissent. Ce phénomène de « décalage » peut se justifier par la remarque que nous avons formulée lors du Chapitre 4. Étant donné le caractère discret du parcours, nous pouvons avoir le profil d'intensité référence qui n'est pas exactement sur le même « pattern » de pixels 8-connexe que la zone d'intérêt qui minimise pourtant le score MSE .

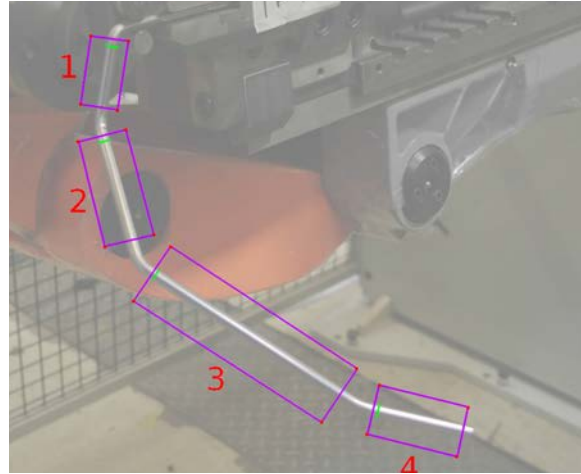
Ces résultats, obtenus à partir de critères reflétant les différents points théoriques énoncées dans le chapitre précédent, démontrent la pertinence de l'utilisation de l'information du profil d'intensité pour la reconnaissance d'objet tubulaire métallique dans un environnement industriel.

5.1.2 Résultats obtenus avec les critères non respectés

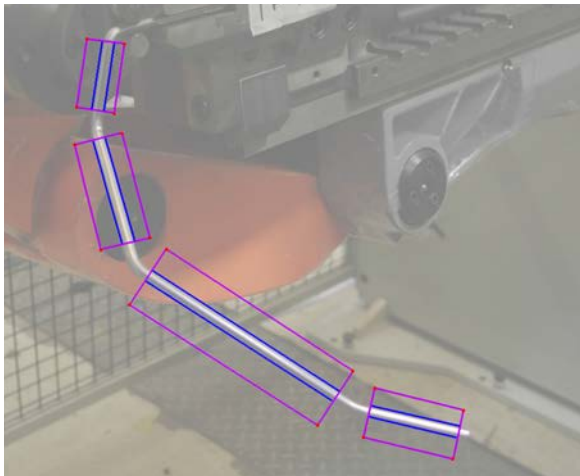
Nous venons de voir le comportement de la méthode lorsque nous nous fixons un cadre pratique reflétant au mieux les différents critères théoriques énoncés dans le Chapitre 4. Nous allons maintenant nous intéresser aux résultats que nous pouvons obtenir lorsque nous relâchons ces critères.

Sur la figure 5.5 dans la zone de recherche n°1 et n°6, nous visualisons une détection assez courte du tube, puis nous obtenons soit une fausse détection, soit aucun résultat. Nous pouvons expliquer ce phénomène par le fait que ces deux zones de recherche ne sont pas définies comme elles devraient l'être. En effet, si nous regardons de plus près, ces deux zones comportent plusieurs zones droites du tube au lieu d'une seule. Nous avons donc des zones droites différentes qui ne sont pas forcément orientées de la même façon, ce qui entraîne qu'il n'y a plus la garantie d'avoir, en moyenne, tout le long du tube la même intensité. Il est même possible d'avoir de très grosses fractures dans l'intensité comme dans la zone de recherche n°6 où nous observons une forte spécularité qui apparaît sur le tube au centre de la zone de recherche. Sachant que nous avons un unique profil d'intensité référence, il est donc impossible dans ces cas de figure de garantir une détection valide.

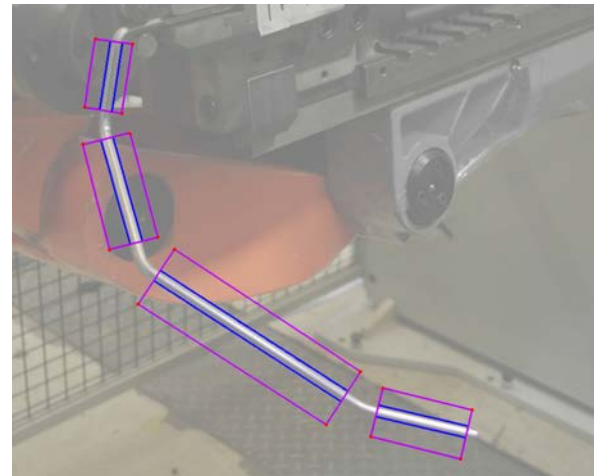
Pour terminer sur ces observations visuelles des résultats, nous pouvons aussi dire que nous ne constatons pas de réelle différence entre la méthode utilisant la version sans interpolation et avec interpolation. Nous avons effectué des mesures de temps d'exécution pour chacun des modèles avec la méthode avec et sans interpolation. Les mesures peuvent



(a)



(b)



(c)

FIGURE 5.2 – Résultats de l'algorithme sur le modèle 3 — (a) Initialisation de l'algorithme — (b) version sans interpolation — (c) version avec interpolation.

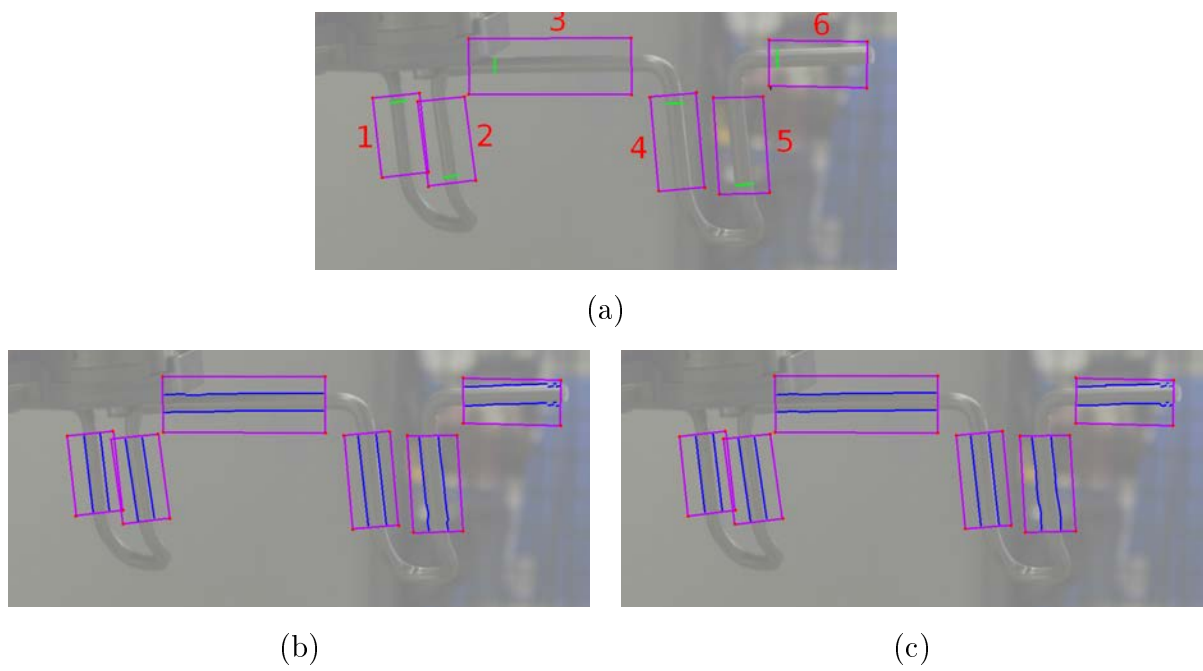


FIGURE 5.3 – Résultats de l'algorithme sur le modèle 4 — (a) Initialisation de l'algorithme — (b) version sans interpolation — (c) version avec interpolation.

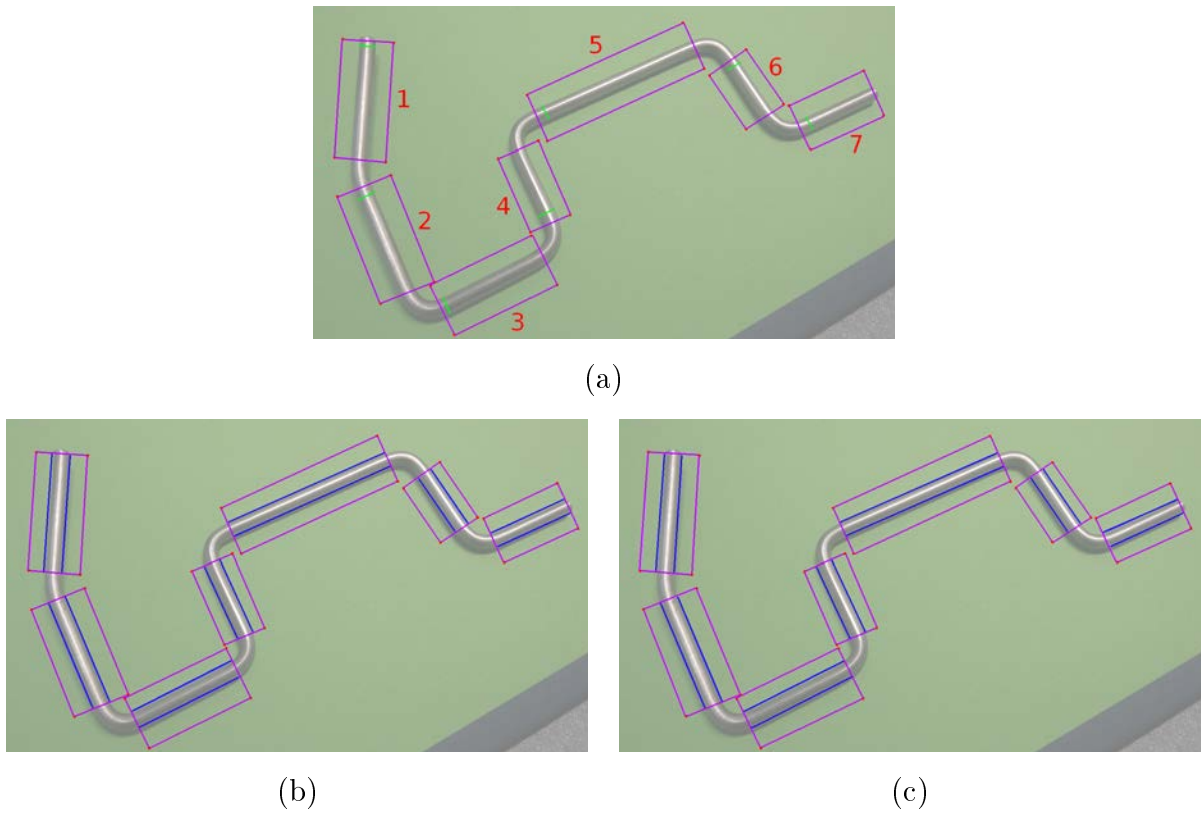
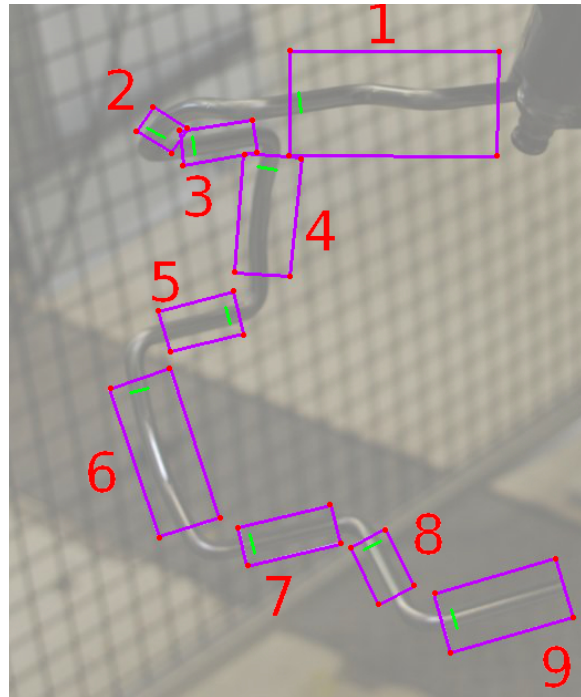
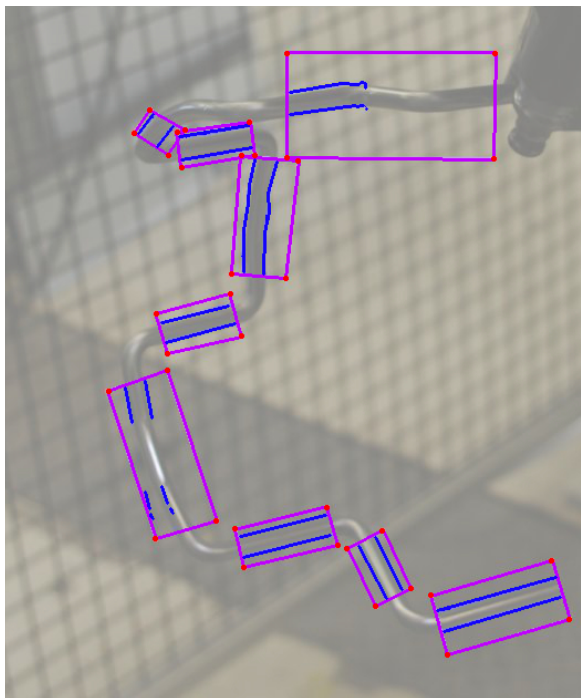


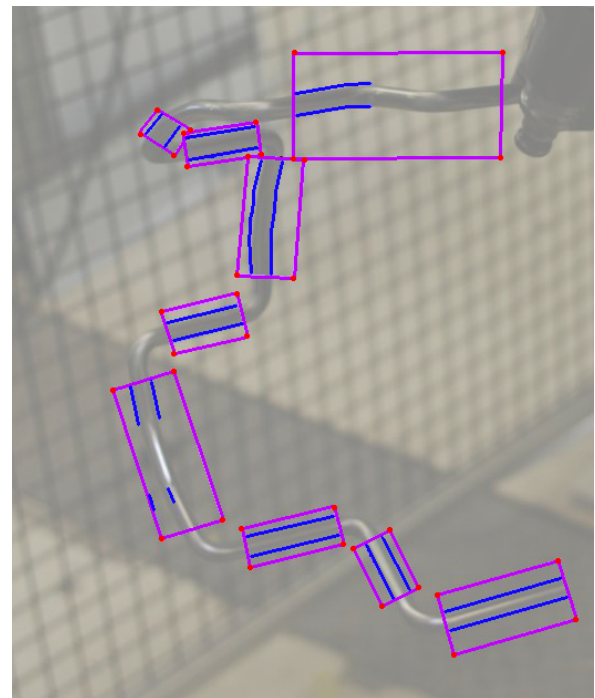
FIGURE 5.4 – Résultats de l’algorithme sur le modèle « fond vert » — (a) Initialisation de l’algorithme — (b) version sans interpolation — (c) version avec interpolation.



(a)



(b)



(c)

FIGURE 5.5 – Résultats de l'algorithme sur le modèle 2 — (a) Initialisation de l'algorithme — (b) version sans interpolation — (c) version avec interpolation.

Benchmark sans interpolation		Benchmark avec interpolation	
Modèle	Temps (ms)	Modèle	Temps (ms)
1	40	1	140
2	141	2	662
3	232	3	952
4	306	4	1804
Fond vert	701	Fond vert	4108

(a) (b)

FIGURE 5.6 – Mesures de temps de l'exécution de l'algorithme sur toutes les zones de recherche d'une image — (a) sans interpolation — (b) avec interpolation.

être visualisées sur la figure 5.6. Le protocole de mesure est le suivant : pour chacune des images, nous avons exécuté l'algorithme dix fois (Nous effectuons cela pour lisser le temps de l'exécution de l'algorithme qui s'exécute sur un ordinateur pas uniquement dédié à celui-ci) et nous avons gardé la moyenne des temps mesurés. Étant donnée la nature de l'algorithme, il est relativement difficile de conclure de manière générale. En effet, le temps d'exécution va dépendre fortement du nombre de zones de recherche que l'on doit inspecter, donc par extension au nombre de zones droites du tube que l'on doit détecter. La taille de chacune des zones de recherche impacte aussi le temps d'exécution final. En revanche, nous pouvons clairement identifier que la version avec interpolation reste plus gourmande que l'autre version. Comme nous n'avons pour l'instant pas identifié de réel avantage d'utiliser la version avec interpolation, le choix de la version sans interpolation devient naturel.

5.1.3 Résultats avec translation des zones de recherche

Nous venons de voir des cas où chacune des zones de recherche possède bien le tube que l'on recherche à l'intérieur. Intéressons nous maintenant aux cas de figure où nous n'avons pas ou partiellement le tube dans les zones de recherche. Pour cela, nous effectuons sur

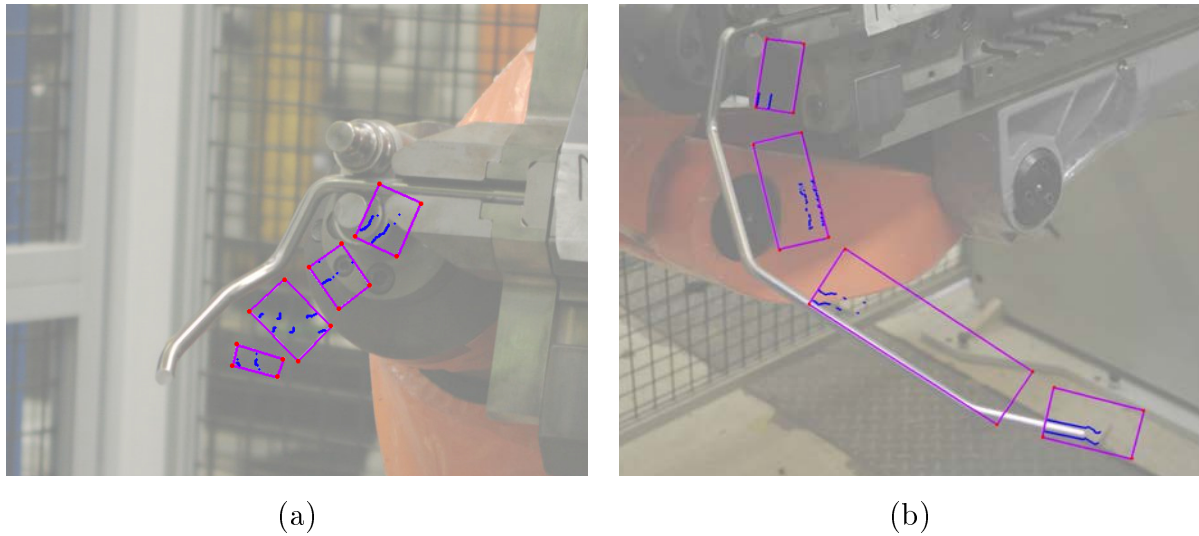


FIGURE 5.7 – Résultats de l’algorithme sur les modèles 1 et 3 après translation des zones de recherche.

les modèles n° 1 et n° 3 une translation de ces zones. Les résultats obtenus sont visibles sur la figure 5.7. Dans un premier temps, nous notons que dans l’ensemble, les résultats sont sous la forme d’un ensemble de points plus ou moins alignés, très peu connectés et surtout très peu nombreux par rapport au nombre de points que les bords du tube peuvent posséder.

De plus, si nous observons la distribution des scores MSE obtenus par les zones d’intérêt qui ont été retenues lors de l’exécution de l’algorithme sur les images après translation des zones de recherche (voir figures 5.8 et 5.9) et sur les images où nous avons obtenu de bonnes détections (voir figures 5.10 et 5.11), nous pouvons constater deux choses.

La première est que les scores de MSE sur les images des modèles n° 1 et n° 3 après translation des zones de recherche sont relativement concentrées autour de gros scores de MSE . La seconde est, qu’à contrario, sur les images où nous avons une bonne détection, les scores de MSE se retrouvent autour de valeurs bien plus faibles. Cette observation peut nous laisser penser que la distribution des MSE est un bon indicateur pour obtenir un indice de confiance sur ce qui a été détecté. Mais cela mérite pour l’affirmer de plus nombreuses expérimentations dans des conditions très différentes. Cela fera partie des

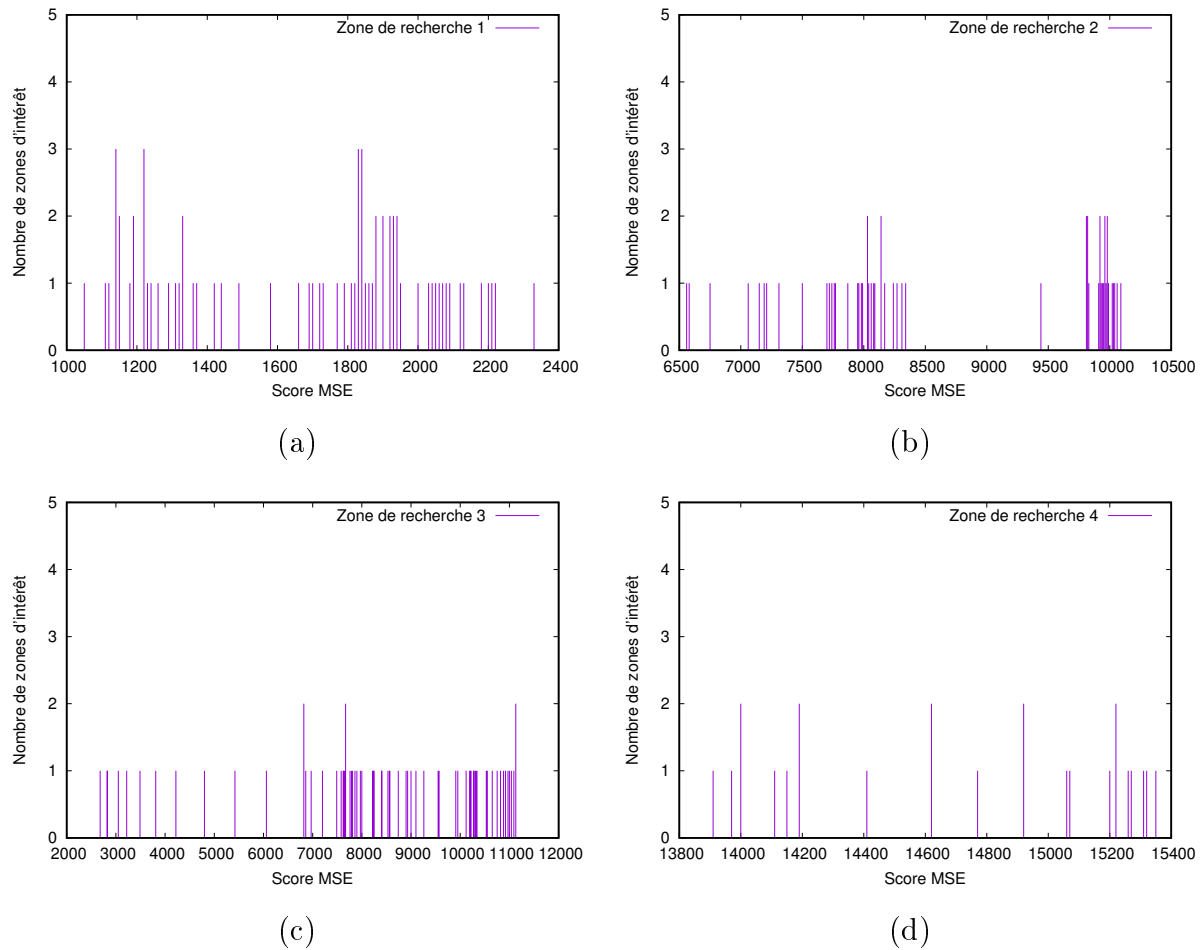


FIGURE 5.8 – Histogrammes représentant la répartition des scores MSE des zones d'intérêt qui ont été acceptées avec la méthode *sans interpolation* sur le modèle 1 après translation des zones de recherche — (a) Zone de recherche 1 — (b) Zone de recherche 2 — (c) Zone de recherche 3 — (d) Zone de recherche 4.

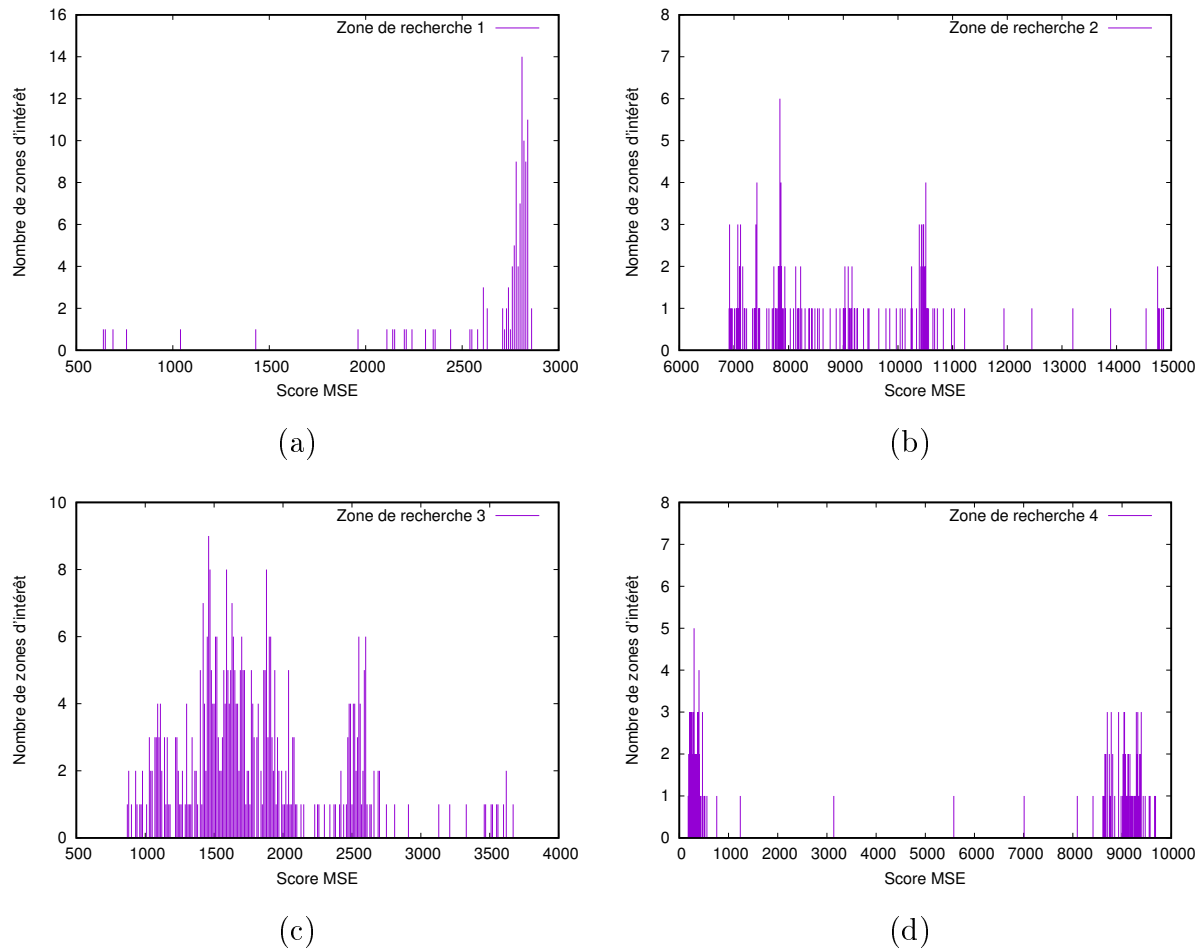


FIGURE 5.9 – Histogrammes représentant la répartition des scores MSE des zones d'intérêt qui ont été acceptées avec la méthode *sans interpolation* sur le modèle 3 après translation des zones de recherche — (a) Zone de recherche 1 — (b) Zone de recherche 2 — (c) Zone de recherche 3 — (d) Zone de recherche 4.

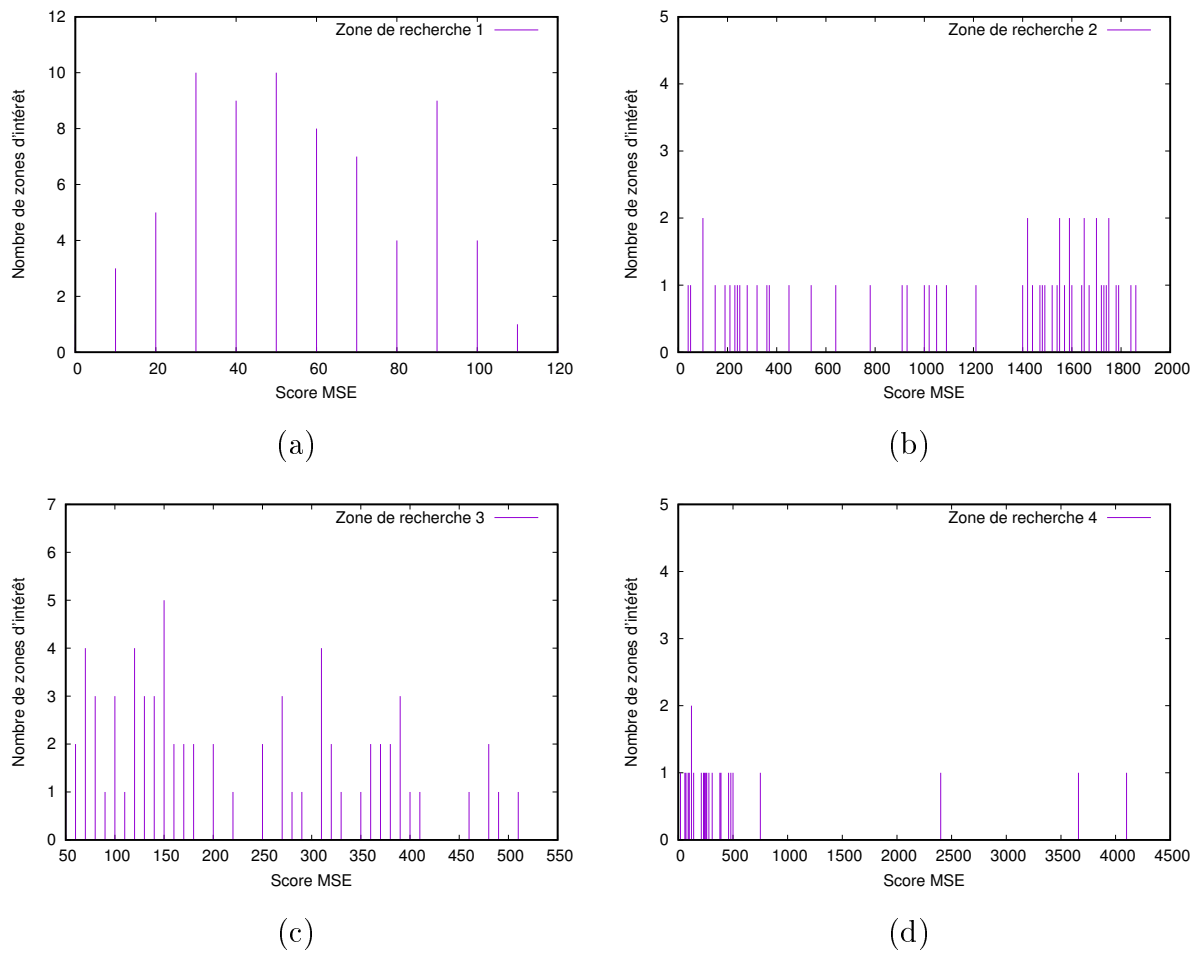


FIGURE 5.10 – Histogrammes représentant la répartition des scores MSE des zones d'intérêt qui ont été acceptées avec la méthode *sans interpolation* sur le modèle 1 — (a) Zone de recherche 1 — (b) Zone de recherche 2 — (c) Zone de recherche 3 — (d) Zone de recherche 4.

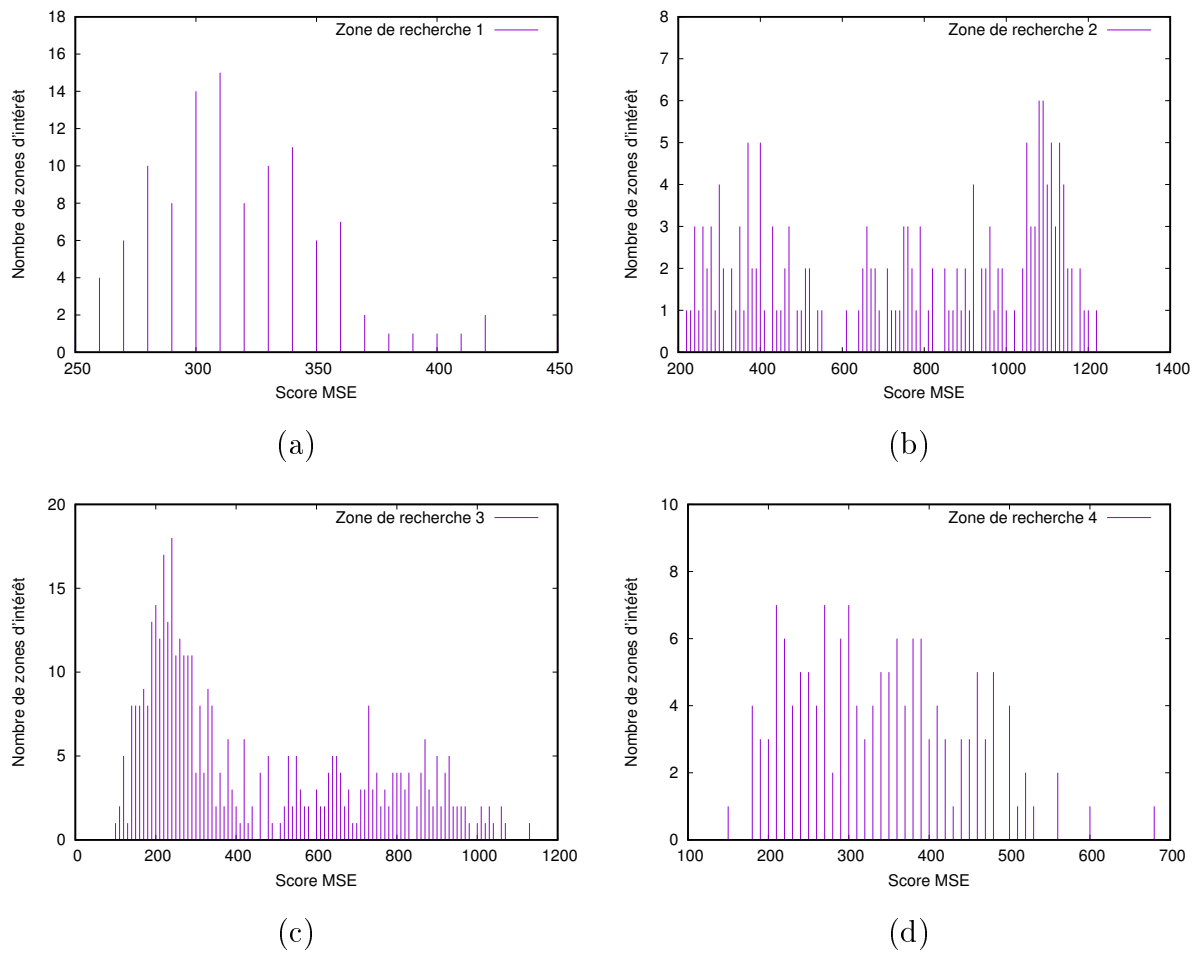


FIGURE 5.11 – Histogrammes représentant la répartition des scores MSE des zones d'intérêt qui ont été acceptées avec la méthode *sans interpolation* sur le modèle 3 — (a) Zone de recherche 1 — (b) Zone de recherche 2 — (c) Zone de recherche 3 — (d) Zone de recherche 4.

perspectives que nous aborderons dans le [Chapitre 6](#) de conclusion.

Si nous regardons la répartition des scores des MSE avec la méthode utilisant l'interpolation que nous retrouvons sur les graphiques de la figure 5.12 et 5.13, une nouvelle fois nous n'observons pas une réelle différence qui pourrait nous faire conclure que la version avec interpolation soit plus pertinente que l'autre.

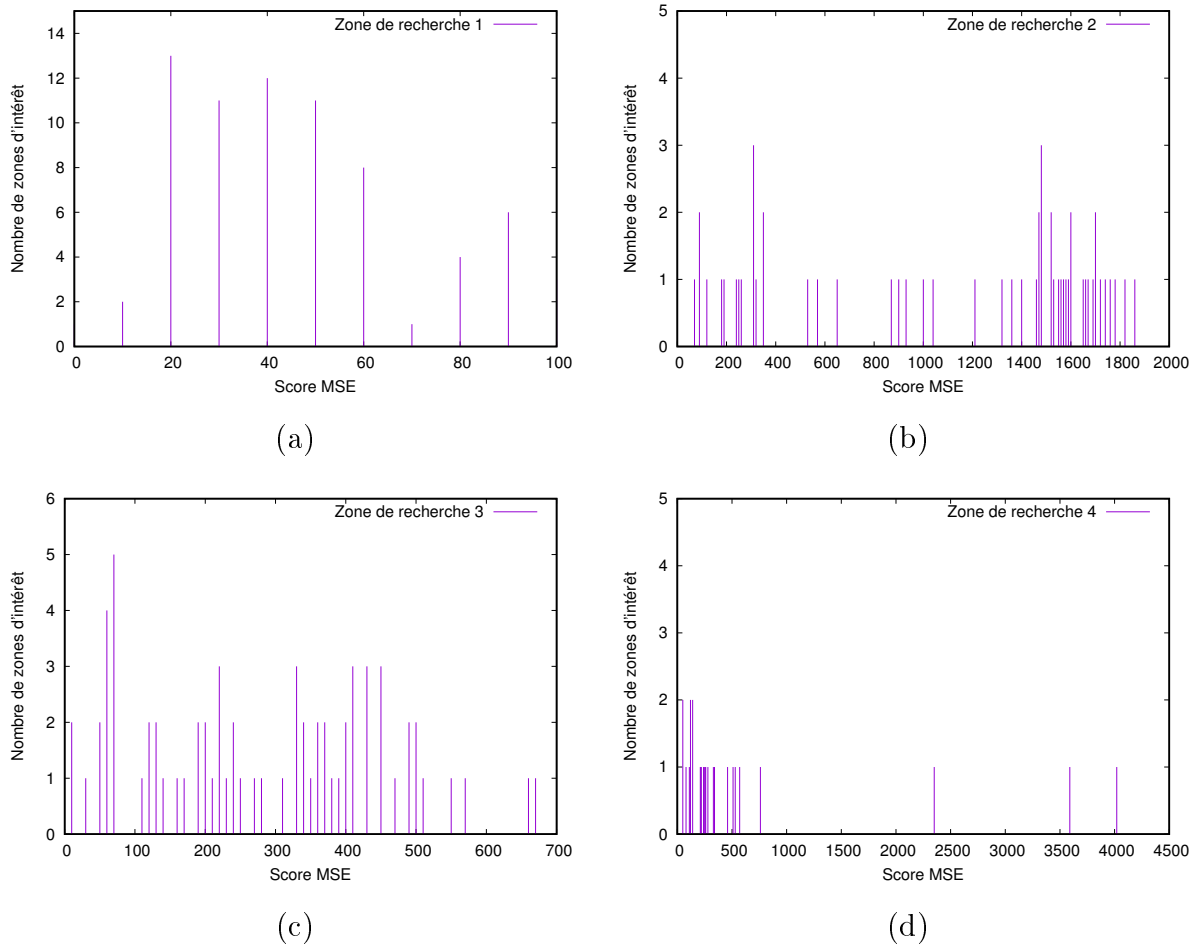


FIGURE 5.12 – Histogrammes représentant la répartition des scores MSE des zones d'intérêt qui ont été acceptées avec la méthode *avec interpolation* sur le modèle 1 — (a) Zone de recherche 1 — (b) Zone de recherche 2 — (c) Zone de recherche 3 — (d) Zone de recherche 4.

Sachant que nous avons à notre disposition la direction théorique du tube pour chaque zone de recherche, nous pouvons effectuer une mesure de déviation en mesurant l'angle

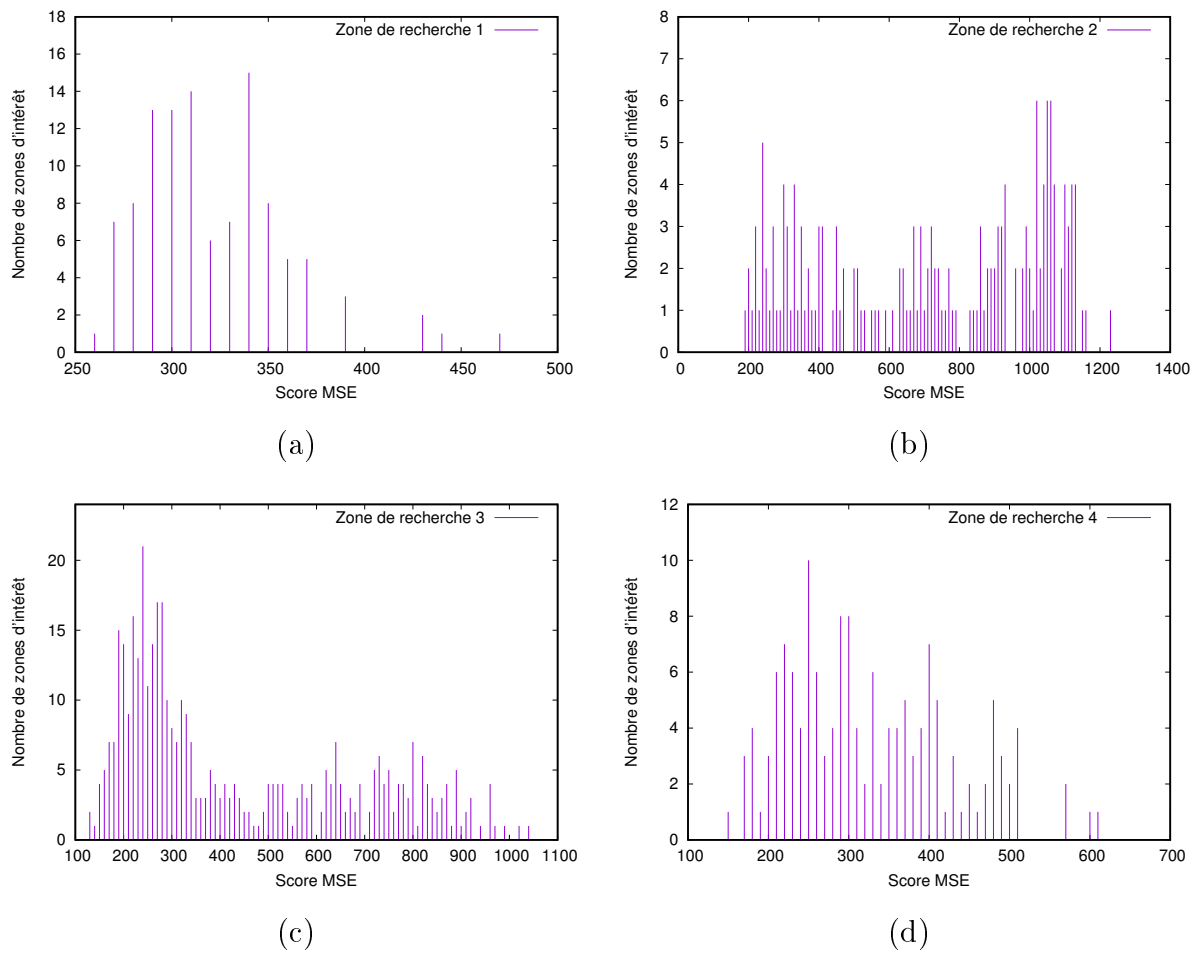


FIGURE 5.13 – Histogrammes représentant la répartition des scores MSE des zones d'intérêt qui ont été acceptées avec la méthode *avec interpolation* sur le modèle 3 — (a) Zone de recherche 1 — (b) Zone de recherche 2 — (c) Zone de recherche 3 — (d) Zone de recherche 4.

entre les segments flous obtenus et la direction théorique du tube. Des mesures ont été effectuées sur les modèles n° 1 et n° 3 dans le cas où nous avons obtenu de bonnes détections et dans le cas où nous effectuons une translation des zones de recherche. Les mesures qui ont été réalisées sont visibles dans les tableaux de la figure 5.14. Dans le cas des images où l'on avait de bonnes détections, nous obtenons des déviations avec la direction théorique relativement faible. Pour ce qui est des images avec translation des zones de recherche, nous obtenons des angles plus élevés, et c'est ce que nous recherchons, mais nous pouvons constater que ce n'est pas toujours le cas. En effet, nous pouvons obtenir malgré une mauvaise détection, des angles relativement petits. Cela s'explique par le fait que le peu de pixels qui sont alignés peuvent parfois se retrouver dans la bonne direction. Cela montre que la mesure d'angle n'est pas un critère suffisant pour noter la qualité d'une détection, tout du moins sans d'autres informations complémentaires. Mais si nous regardons à nouveau la figure 5.7, étant donnée la pauvreté du résultat obtenu par rapport aux nombres de points que les différents segments flous possèdent, il n'est clairement pas pertinent d'effectuer une mesure de déviation.

Modèle 1			Modèle 3		
Zone de recherche	Angle sans interpolation (°)	Angle Translation (°)	Zone de recherche	Angle sans interpolation (°)	Angle Translation (°)
1	0.0970292	22.3528	1	0.468047	0.374477
2	4.63027	0.103797	2	0.784153	0.982117
3	4.17498	41.9059	3	0.436897	20.8973
4	0.264034	18.2738	4	1.04163	0.345151

(a)
(b)

FIGURE 5.14 – Mesures d'angles entre la direction de la détection et la direction théorique du tube pour chaque zone de recherche sur les modèles 1 et 3 — en rouge les grandes déviations par rapport à la direction théorique du tube.

5.2 Intégration industrielle

Comme nous l'avons précisé auparavant dans ce manuscrit, cette thèse a été effectuée au sein d'une entreprise dont l'objectif principal est de proposer un système de contrôle qualité systématique des pièces produites. Dans un premier temps, le contrôle qui doit être

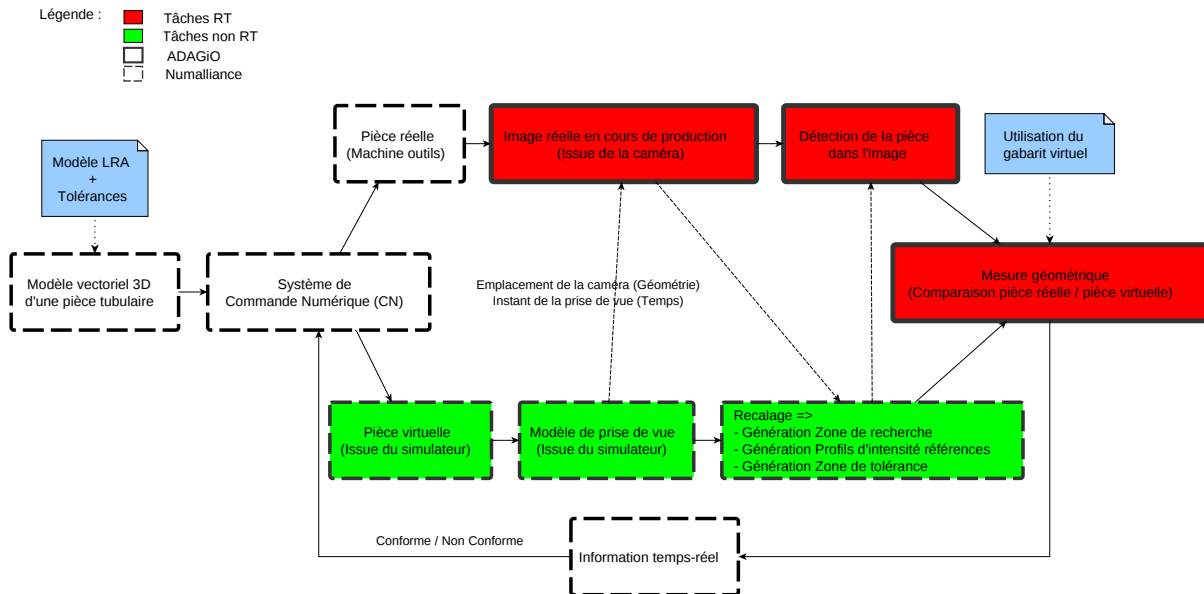
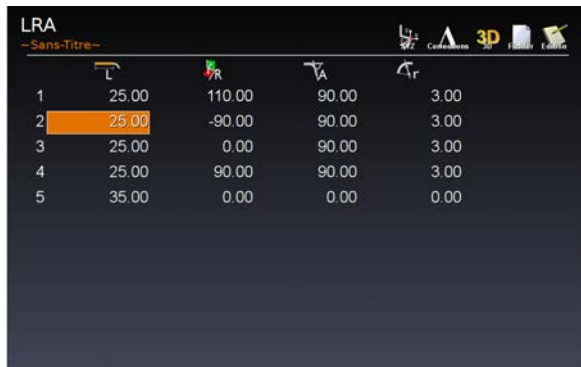


FIGURE 5.15 – Schéma global du processus du système de contrôle qualité.

mis en place est un contrôle binaire qui doit uniquement dire si une pièce est valide ou pas par rapport à une tolérance pré-définie. De plus, ce processus devra être automatique avec un paramétrage simple et rapide. Pour cela, la méthode que nous avons présentée s'inscrit dans un schéma global représenté par la figure 5.15 où nous pouvons voir des modules que nous allons expliciter ici, de manière non exhaustive et informelle, pour introduire des notions qui vont permettre de rendre notre méthode automatique.

La première donnée importante que fournit le processus en place dans les machines de Numalliance est la pièce virtuelle ainsi que la simulation. En effet, le logiciel en place sur les machines propose de créer facilement une pièce à l'aide d'une représentation *LRAr*. Cette représentation permet de spécifier une pièce à l'aide d'une seule et unique « primitive ». La « primitive » correspond à une longueur droite L , une orientation pour le pli R , un angle A et un rayon r . Au final, une pièce est un assemblage de « primitives » que nous pouvons visualiser sur la figure 5.16 où chacune des lignes correspond à une primitive *LRAr*. Après avoir spécifié sa pièce, l'opérateur peut dérouler la fabrication de celle-ci par la machine dans une simulation en 3-dimensions. Nous avons donc à notre disposition la pièce en fabrication dans son environnement machine virtuel (figure 5.17).

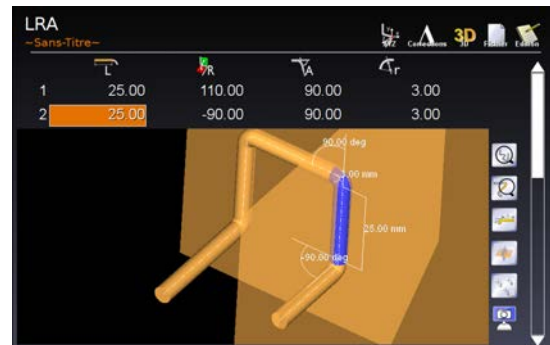
La seconde donnée est la prise de photo de la pièce réelle par un appareil d'acquisition



LRA
~Sans-Titre~

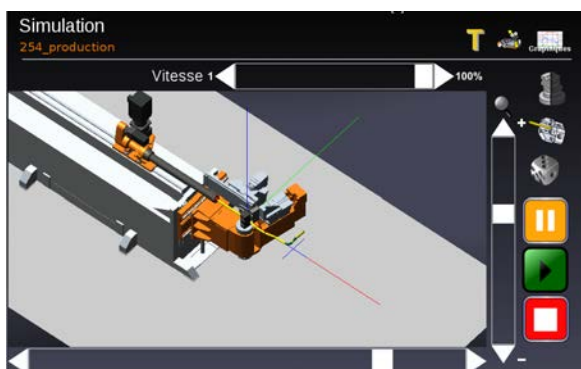
	L	R	A	r
1	25.00	110.00	90.00	3.00
2	25.00	-90.00	90.00	3.00
3	25.00	0.00	90.00	3.00
4	25.00	90.00	90.00	3.00
5	35.00	0.00	0.00	0.00

(a)

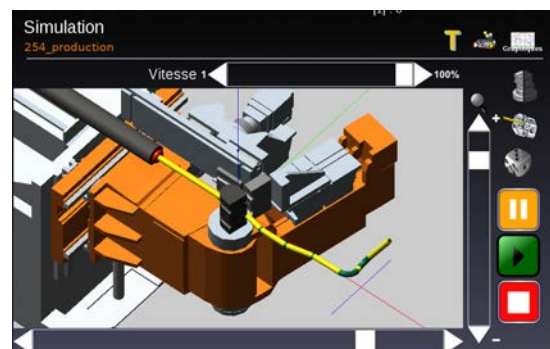


(b)

FIGURE 5.16 – Définition d’une pièce dans son modèle *LRAr*. On retrouve cette écran de spécification de pièce directement dans l’interface de la machine-outil.



(a)



(b)

FIGURE 5.17 – Illustration de la partie simulation du logiciel produit par Numalliance.

type caméra industrielle. Cette caméra sera fixe pour toute la durée de la production, donc le point de vue par rapport à la machine ne changera pas une fois sélectionné.

Le but ici est donc d'automatiser la méthode à l'aide de ces différentes données. Nous devons donc trouver un moyen de rendre la génération automatique des zones de recherche, des profils d'intensité références pour chacune d'elles, ainsi qu'une zone de tolérance. La zone de tolérance est de la même forme qu'une zone de recherche et incluse dedans. Cette zone permet de simuler un gabarit physique de telle sorte que nous sommes capables, après reconnaissance du tube, de dire si il est conforme ou non à la spécification par rapport à des tolérances définies.

L'idée générale est que nous soyons capables, à partir d'une méthode de recalage d'un objet virtuel avec un objet réel comme nous pouvons le trouver dans le domaine de la réalité augmentée [RU13], de générer ces différentes données importantes pour l'initialisation de l'algorithme. Aucune expérimentation n'a été pour l'instant effectué mais à partir de la classification des différentes méthodes énoncée dans [RU13], nous pouvons effectuer un premier choix reflétant les contraintes que nous avons. Nous nous intéresserons, dans un premier temps, aux méthodes utilisant un capteur optique « sensor-based tracking » et plus particulièrement celles utilisant la notion de marqueurs visuels « marker-based tracking ». Ces marqueurs visuels « fuducials » comportent des propriétés spécifiques qui permettent d'identifier facilement leur position dans le monde « réel » et par conséquent la position du capteur optique.

Pour commencer, nous devons calibrer notre dispositif de visualisation pour connaître ses paramètres intrinsèques, c'est-à-dire connaître les déformations qui se produisent lors de l'acquisition d'une image, induites par les différents éléments du dispositif (i.e. capteur, lentille, etc.) ainsi que la distance focale. De plus, ces paramètres sous forme d'une matrice, nous permettent d'effectuer la projection d'un point 3D dans l'espace du dispositif sur le plan image (2D). Cela peut être effectué facilement à l'aide d'une librairie telle que OpenCV [ope16] avec l'utilisation d'une image connue en forme de damier.

Ensuite, nous avons notre dispositif de visualisation qui est placé dans l'environnement de la machine-outil et qui ne bougera pas pendant toute la production. Notre objectif est de positionner virtuellement ce dispositif pour obtenir une correspondance entre la machine-outil virtuelle et la machine-outil réelle. Pour cela, nous devons calculer les paramètres extrinsèques de la caméra, c'est-à-dire la matrice de rotation et le vecteur de

translation qui nous permettent d'obtenir la transformation d'un point 3D de la scène observée dans l'espace du dispositif.

Une fois les informations intrinsèques et extrinsèques du dispositif obtenues, nous devons mettre en correspondance les points d'intérêt qui sont sur la machine-outil réelle et virtuelle de telle sorte que la scène virtuelle 3D représente le plus précisément possible ce que le dispositif de visualisation observe. À partir de là, nous obtenons donc un modèle 3D de la pièce sortant de la machine-outil avec la même vue que le dispositif de visualisation. Nous pouvons donc générer virtuellement les différentes zones (i.e. recherche, tolérance, etc.) puis les projeter dans l'espace 2D des images que le dispositif d'acquisition va prendre.

5.3 Description d'un premier prototype

La partie vision par ordinateur a été réalisée en début de thèse. N'étant pas le cœur du sujet de thèse mais malgré tout une partie importante du processus, nous avons mis en place un logiciel permettant de simuler cette étape. À l'aide d'un appareil photo calibré (i.e. nous avons les paramètres intrinsèques), nous prenons un cliché d'une pièce manufacturée. Puis nous effectuons un recalage avec le modèle 3D (figure 5.18 (a)) manuellement par transparence (figure 5.18 (b)). Nous avons maintenant à disposition le modèle 3D de la pièce recalée sur la pièce réelle (figure 5.19 (a)), nous pouvons virtuellement générer (figure 5.19 (b)) les zones de recherche (en bleu), les zones de tolérance (en vert), ainsi que les zones où nous allons extraire les profils d'intensité références (en rose).

Nous effectuons ensuite la projection dans l'image de ces différentes zones, grâce aux paramètres intrinsèques du dispositif. Nous visualisons la projection des profils d'intensité références sur une visualisation particulière (voir figure 5.20) où nous pouvons éventuellement corriger le placement de ceux-ci lors d'une erreur de projection. Lorsque nous validons nos profils d'intensité référence, l'extraction s'opère et nous avons toutes les informations utiles pour exécuter notre méthode de reconnaissance de tube.

Nous pouvons voir le résultat sur la figure 5.21 où nous observons les différentes zones de recherche en bleu, zones de tolérance en vert, ainsi que le résultat obtenu en rouge dans la première zone de recherche. Cette interface permet de se déplacer de zones de recherche en zones de recherche pour visualiser le résultat de la détection. Nous pouvons aussi voir qu'une indication est disponible pour nous prévenir si la détection effectuée se

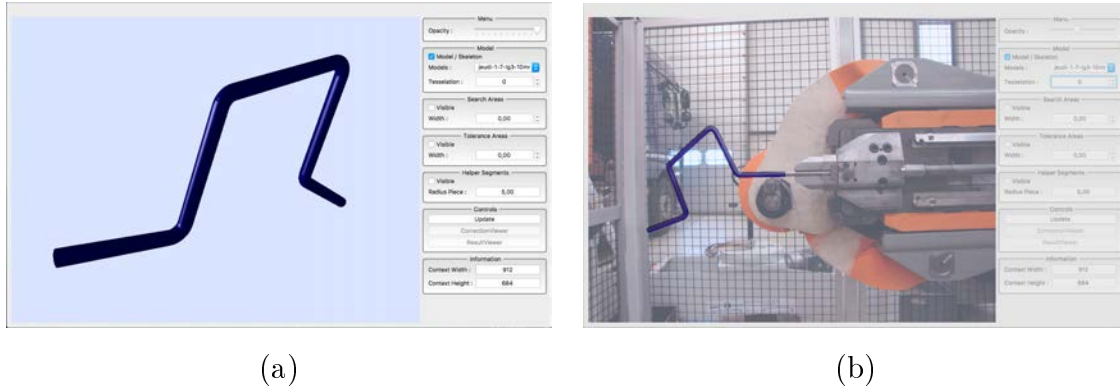


FIGURE 5.18 – Visualisation d'un modèle 3D d'une pièce (a) — Illustration d'un recalage d'une pièce 3D virtuelle sur la pièce 2D dans l'image (b).

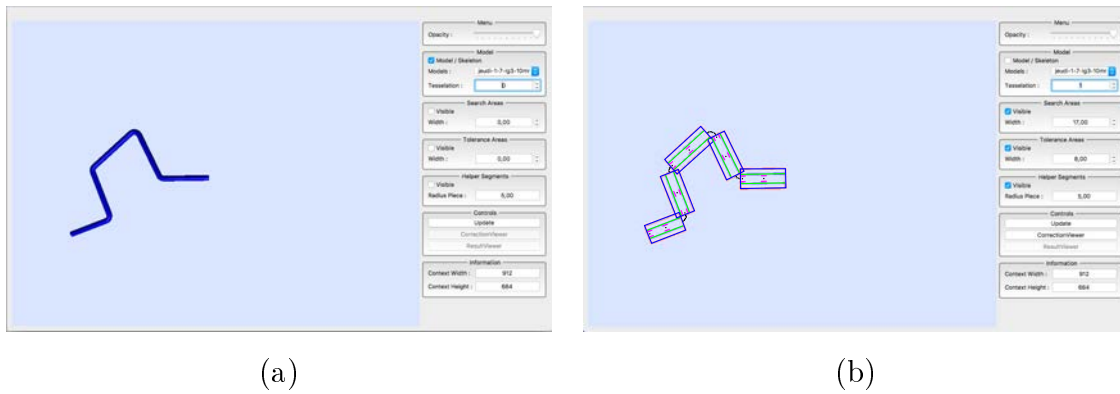


FIGURE 5.19 – Illustration d'une pièce 3D après recalage (a) — Illustration de la génération des zones de recherche, tolérance et profils d'intensité références sur le modèle 3D recalé (b).

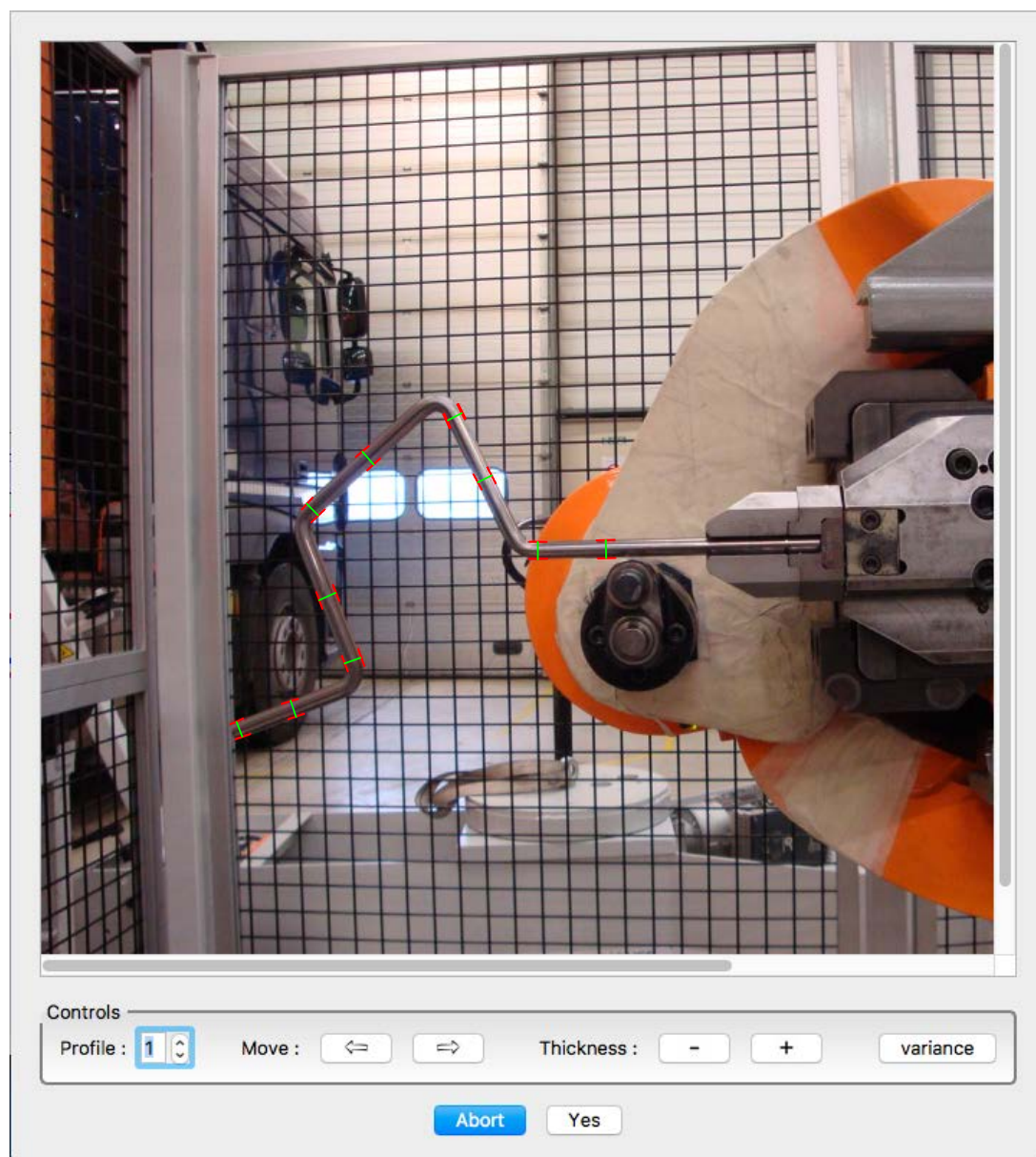


FIGURE 5.20 – Illustration de la visualisation de la projection des profils d'intensité références dans l'espace image.

trouve bien dans la zone de tolérance ou non. Ce qui nous permet d’avoir une information de conformité par exemple de ce segment droit du tube. Nous avons aussi pour terminer, le temps d’exécution de la détection pour l’ensemble des zones de recherche définis dans l’image.

Pour conclure, nous pouvons prendre la décision qu’une pièce est conforme aux spécifications lorsque ces différentes conditions sont remplies :

- Toutes les zones de recherche doivent être valides.
- Une zone de recherche est valide si :
 - les deux segments flous représentant les bords du tube sont bien compris dans la zone de tolérance.
 - l’angle entre les deux segments flous et la direction théorique du tube est inférieur à une tolérance fixée.

Dans ce chapitre, nous avons effectué différentes expériences pour valider le comportement de l’algorithme. Dans en premier temps nous l’avons exécuté dans des conditions qui respectaient les différents critères énoncés en introduction. Dans un second temps, nous avons levé certains de ces critères. Nous avons obtenu des résultats prometteurs autant dans les cas respectant les critères que ceux ne les respectant pas. En effet, nous avons pu constater que l’algorithme produisait les résultats attendus dans les cas respectant les critères. Dans les cas ne respectant pas les critères, nous obtenons des résultats moins précis mais dont nous pouvons en extraire des informations intéressantes comme une direction approximative du tube. Nous avons aussi pu identifier que l’algorithme, dans la majorité des cas, produisait une détection disparate dans les conditions où nous n’avions pas de tube dans la zone de recherche ou un profil d’intensité référence faux. Ce comportement est tout à fait normal et même souhaité car cela permet de discriminer facilement les cas où la pièce n’est pas conforme à la spécification. Pour terminer, nous avons ensuite présenté un premier prototype d’un logiciel permettant de « simuler » la totalité du processus qui sera amené à être implanter dans des machines-outils de production.

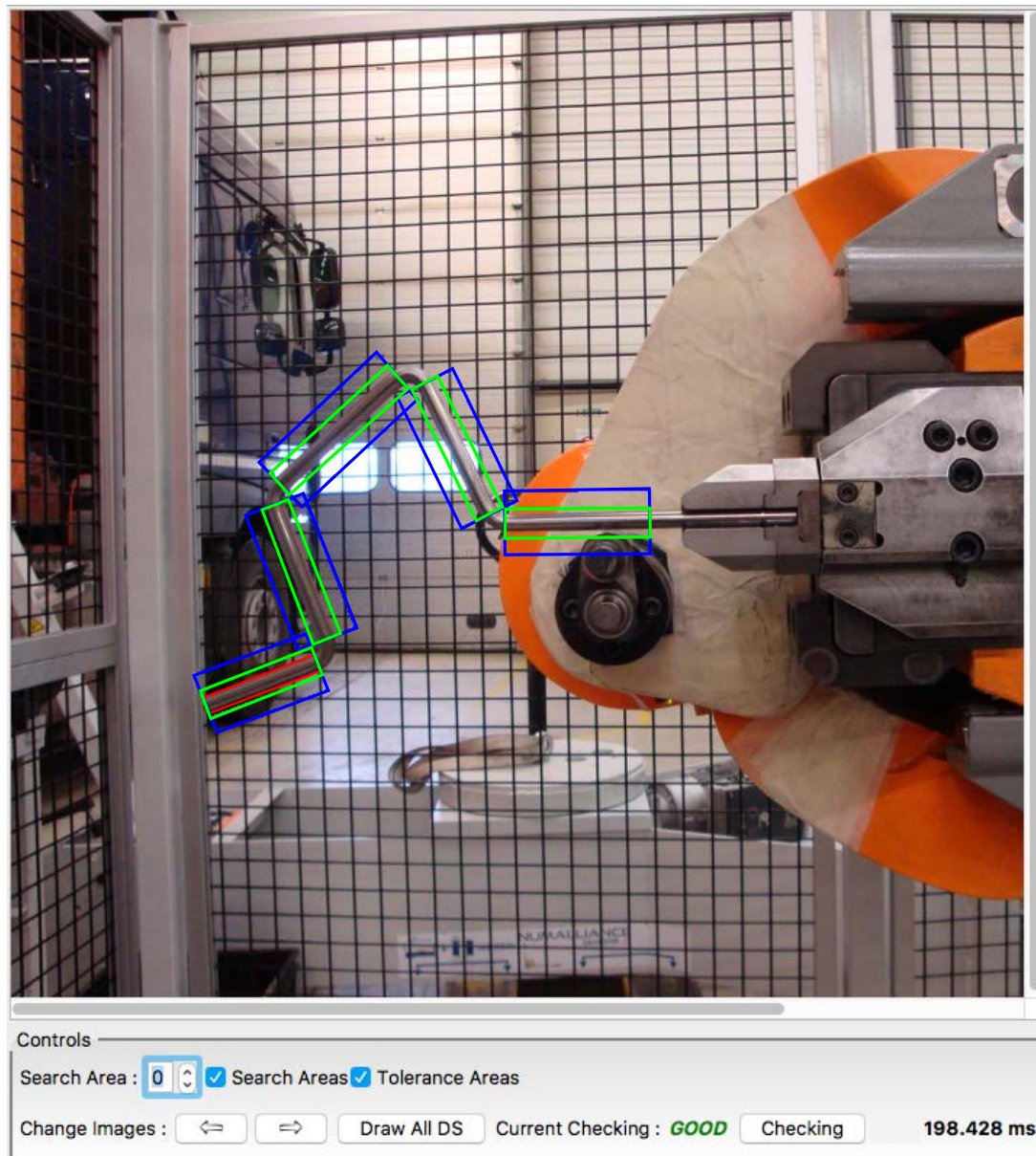


FIGURE 5.21 – Illustration de la visualisation des résultats, obtenus à l'aide des informations extraites du processus — en rouge la dernière détection obtenue. Il est possible de naviguer entre les zones de recherche pour visualiser la détection obtenue.

Chapitre 6

Conclusion & perspectives

LES travaux qui sont présentés dans ce manuscrit ont permis d'effectuer dans un premier temps un état de l'art des méthodes de segmentation ainsi que des méthodes de détection de droites. Cet état de l'art a été plus précisément réalisé dans un cadre de détection d'objet tubulaire comportant des reflets spéculaires dans un environnement industriel non contrôlé. Dans ce contexte, les difficultés constatées des différentes méthodes, nous ont amené à proposer une solution utilisant un outil particulier : le profil d'intensité.

La méthode proposée, en plus du profil d'intensité, utilise différents outils issus du domaine de la géométrie discrète. En effet, un parcours incrémental [KE09] de l'image à l'aide de segments discrets naïfs (8-connexe) a été mis en avant avec une utilisation conjointe de la notion de segment flou [DRFRD06]. L'ensemble de ces outils a permis de proposer un processus spécifique pour la détection de contours d'objets tubulaires spéculaires n'apparaissant pas dans la littérature sur le sujet.

Dans un premier temps, le [Chapitre 4](#) a introduit une méthode semi-automatique où les notions de zone de recherche, segment de recherche et zone d'intérêt ont été définies. Ces différentes notions permettent de parcourir des sous-ensembles d'une image (i.e. zones de recherche) à l'aide de segments discrets naïfs que nous translatons (i.e. segment de recherche) et de détecter l'objet tubulaire par une méthode de comparaison de profils d'intensité (i.e. comparaison de deux profils d'intensité à l'aide d'une mesure d'erreur quadratique moyenne (MSE)). Cette méthode est semi-automatique car nous devons, pour initialiser l'algorithme, définir à la main les zones de recherche ainsi que les profils d'intensité références qui sont utilisés lors de la comparaison à l'aide du MSE .

Dans un second temps, nous avons au [Chapitre 5](#) intégré la méthode dans un contexte industriel pour une tâche de contrôle qualité d'objet tubulaire métallique manufacturé par des machines-outils. Ce contexte a permis de mettre la méthode à l'épreuve dans un environnement industriel non contrôlé (i.e. conditions lumineuses et fonds différents selon les cas). Le but de cette intégration était de proposer un processus complet permettant, à partir d'une image, de dire si une pièce manufacturée est conforme à sa spécification et cela de manière automatique et systématique (i.e. pour chaque pièce fabriquée). Pour cela, nous avons proposé d'étendre la méthode avec des informations supplémentaires qui sont déjà disponibles dans le système des machines-outils. Ces informations alimentent un processus de recalage d'images issu du domaine de la vision par ordinateur qui permet de générer automatiquement les différents objets dont l'algorithme a besoin (i.e. zones de recherche, profils d'intensité références, zones de tolérance). Nous avons donc, grâce à l'ajout de cette étape, une méthode automatique.

Un logiciel divisé en plusieurs parties a été construit et présenté dans la section [Intégration Industrielle](#) du [Chapitre 5](#) qui nous a permis de simuler ce processus et effectuer des premiers tests de validation. Nous avons vu dans la section [Validation](#) que, visuellement et à l'aide de mesures de déviation, les détections étaient relativement très proches des contours des zones droites des objets tubulaires analysés. Mais nous avons pu voir aussi que dans certains cas, qui ne sont pas forcément des cas rares, nous obtenons soit aucune détection soit des détections que nous pouvons nommer « faux-positifs ». Ces mauvaises détections sont dues principalement au fait que ce que nous recherchons n'est pas ou partiellement présent dans les zones de recherche (i.e. cas des pièces fausses par exemple) ou encore si l'environnement lumineux a changé (i.e. les profils d'intensité références ne sont plus valables).

Nous avons exposé aussi quelques mesures possibles (e.g. Longueur des segments flous, angle entre les segments flous et la direction théorique du tube, etc.) permettant certainement, en les combinant, d'obtenir un indice de confiance sur la détection qui a été effectuée. Pour cela, nous devons effectuer de nombreux tests pour juger de la pertinence de ces indicateurs. De plus, cela permettra d'obtenir *in fine* un processus entièrement opérationnel pour la mise en production.

Nous pourrions aussi tester certains indicateurs qui n'ont pas encore été implémentés ou bien définis comme par exemple la création d'un score d'intensité référence pour chaque

zone de recherche (e.g. l'intensité moyenne de la zone de recherche) qui pourrait être comparé au score d'intensité courant lorsqu'une détection semble mauvaise dans l'optique de repérer qu'il y a eu un changement lumineux de la scène et qu'il faut par conséquent effectuer une nouvelle extraction des profils d'intensité références.

Le dernier point intéressant que nous devrons aborder dans la suite de ces travaux est le changement de diamètre du tube dans l'image. En effet, du fait de l'angle de prise de l'image et de la méthode de projection « perspective », nous pouvons très bien obtenir un segment droit du tube avec un diamètre différent au début et à la fin de celui-ci. Étant donné que nous prenons un profil d'intensité référence à un unique endroit du segment droit du tube, si il y a un changement de diamètre alors la détection pourra être perturbée (i.e. détection trop grand ou trop petite). Nous pourrons certainement détecter ce type de changement à partir du processus de vision par ordinateur à l'aide du modèle 3D. Effectivement, si le segment droit du tube du modèle 3D n'est pas perpendiculaire au plan de projection, nous savons qu'il y aura des changements de diamètre plus ou moins important mais il pourra être calculé à l'aide du modèle 3D et des informations de projection.

Dans les perspectives du projet industriel, nous pouvons imaginer, par exemple, modifier le processus pour utiliser plusieurs caméras et ainsi effectuer une reconstruction 3D. Si cette reconstruction est suffisante, nous pourrons exécuter une étape de mesures géométriques et proposer une correction automatique à la machine-outil. Cela permettra à terme de ne plus ou très peu arrêter la production d'une pièce. De plus, en l'état, notre méthode s'intéresse uniquement aux parties droites de l'objet tubulaire (en rouge sur la figure 6.1) mais pour effectuer une reconstruction 3D entière d'une pièce, il sera important de commencer à proposer en 2D la détection de l'objet entier et pour cela s'intéresser à la détection des arcs de cercle (i.e. les parties courbes de l'objet tubulaire — en vert sur la figure 6.1).

Nous pouvons aussi mettre en avant des perspectives plus générales. Comme il a été montré dans la section « [Extension aux images non industrielles & Limites](#) » du [Chapitre 4](#), il est tout fait envisageable d'adapter notre méthode pour la détection de segments épais dans une image en niveau de gris ou plus généralement des structures épaisses. Cette détection pourra être plus robuste avec l'ajout d'extensions comme le changement à la volée des profils d'intensité références ou encore le changement de la taille de ceux-ci

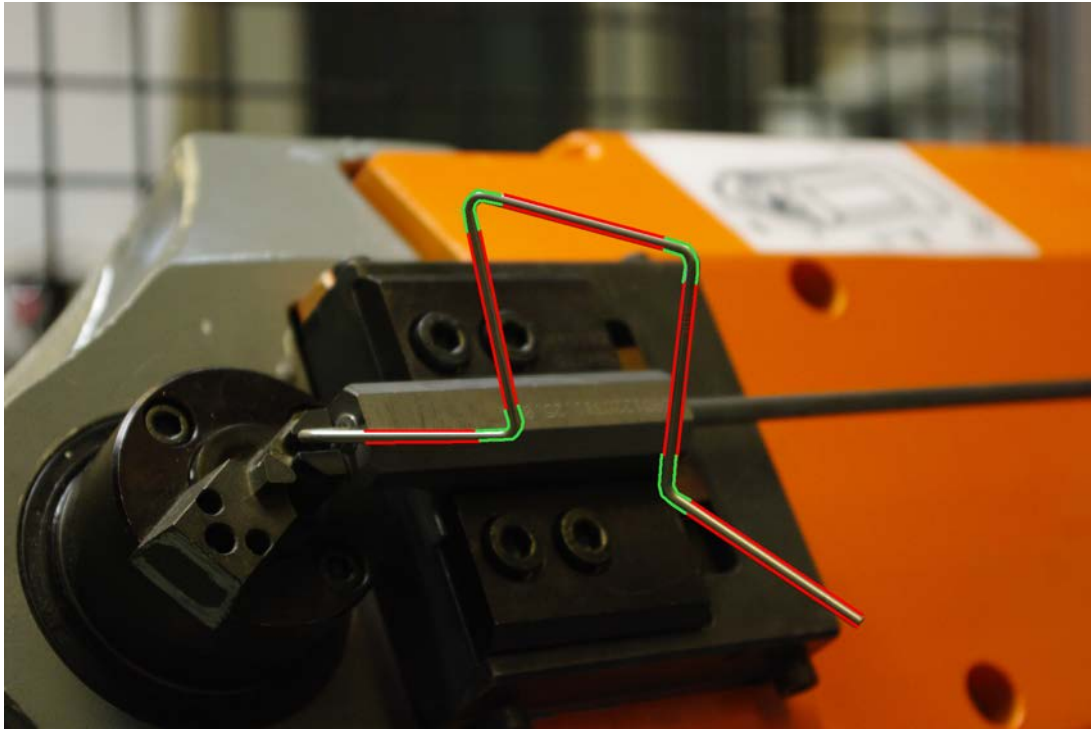


FIGURE 6.1 – Zones droites (en rouge) et courbes (en vert) d'un objet tubulaire.

pour épouser le changement d'épaisseur de la structure dû, par exemple, à la perspective dans l'image. Ces différentes perspectives seront aussi utiles pour le cas industriel qui rencontre les mêmes types de problèmes. Mais dans ce cadre, nous pourrions proposer une solution plus facilement que sur une image quelconque puisque nous avons les informations des transformations géométriques ainsi que le modèle 3D à notre disposition.

Pour terminer, d'un point de vue perspective industrielle, ces travaux vont être implémentés directement sur une machine-outil Numalliance pour, dans un premier temps, valider la méthode de reconnaissance du tube et tester sa robustesse en effectuant des tests dans des conditions différentes d'éclairages et avec différents types de pièces. Si ces différentes expériences sont concluantes, nous pourrions effectuer des tests en production (i.e. sur de longue période) pour vérifier que nous obtenons un bon processus de contrôle qualité et ainsi envisager une implémentation durable de la méthode pour des machines commercialisées.

Bibliographie

- [ÁBHMN10] Luis Álvarez, Luis Baumela, Pedro Henríquez, and Pablo Márquez-Neila. Morphological snakes. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2197–2202. IEEE, 2010.
- [ABMNH12] Luis Alvarez, Luis Baumela, Pablo Márquez-Neila, and Pedro Henríquez. A real time morphological snakes algorithm. *Image Processing On Line*, 2 :1–7, 2012.
- [AKDRE15] Nicolas Aubry, Bertrand Kerautret, Isabelle Debled-Rennesson, and Philippe Even. Parallel strip segment recognition and application to metallic tubular object measure. In *International Workshop on Combinatorial Image Analysis*, pages 311–322. Springer, 2015.
- [ASS⁺12] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE transactions on pattern analysis and machine intelligence*, 34(11) :2274–2282, 2012.
- [AT11] Cuneyt Akinlar and Cihan Topal. Edlines : A real-time line segment detector with a false detection control. *Pattern Recognition Letters*, 32(13) :1633–1642, 2011.
- [BDTL15] Pierre Buysens, Maxime Daisy, David Tschumperlé, and Olivier Lézoray. Restauration de cartes de profondeur basée sur les superpixels pour la synthèse de vue rgb-d. In *gretsi*, 2015.
- [BHR86] J Brian Burns, Allen R Hanson, and Edward M Riseman. Extracting straight lines. *IEEE transactions on pattern analysis and machine intelligence*, (4) :425–455, 1986.

- [BL79] Serge Beucher and Christian Lantuéjoul. Use of watersheds in contour detection. <http://cmm.ensmp.fr/~beucher/publi/watershed.pdf>, September 1979.
- [Can86] J. Canny. A Computational Approach to Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6) :679–698, November 1986.
- [CM02] Dorin Comaniciu and Peter Meer. Mean shift : A robust approach toward feature space analysis. *IEEE Transactions on pattern analysis and machine intelligence*, 24(5) :603–619, 2002.
- [CV01] Tony F Chan and Luminita A Vese. Active contours without edges. *IEEE Transactions on image processing*, 10(2) :266–277, 2001.
- [dgt16] DGtal : Digital geometry tools and algorithms library. <http://dgtal.org>, 2016.
- [DH72] Richard O Duda and Peter E Hart. Use of the hough transformation to detect lines and curves in pictures. *Communications of the ACM*, 15(1) :11–15, 1972.
- [DMM07] Agnes Desolneux, Lionel Moisan, and Jean-Michel Morel. *From gestalt theory to image analysis : a probabilistic approach*, volume 34. Springer Science & Business Media, 2007.
- [DR95] Isabelle Debled-Rennesson. *Etude et reconnaissance des droites et plans discrets*. PhD thesis, 1995.
- [DR07] Isabelle Debled-Rennesson. Éléments de géométrie discrète : Vers une étude des structures discrètes bruitées. *Habilitation à diriger des recherches, Université Henri Poincaré, Nancy*, 2007.
- [DRFRD06] Isabelle Debled-Rennesson, Fabien Feschet, and Jocelyne Rouyer-Degli. Optimal blurred segments decomposition of noisy shapes in linear time. *Computers & Graphics*, 30(1) :30–36, 2006.
- [DRJLRD03] Isabelle Debled-Rennesson, Remy Jean-Luc, and Jocelyne Rouyer-Degli. Segmentation of discrete curves into fuzzy segments. *Electronic Notes in Discrete Mathematics*, 12 :372–383, 2003.

- [EM00] Philippe Even and Anne Malavaud. Semi-automated edge segment specification for an interactive modelling system of robot environments. *International Archives of Photogrammetry and Remote Sensing*, 33(B5/1 ; PART 5) :222–229, 2000.
- [FO08] Leandro AF Fernandes and Manuel M Oliveira. Real-time line detection through an improved hough transform voting scheme. *Pattern Recognition*, 41(1) :299–314, 2008.
- [Fre61] Herbert Freeman. On the encoding of arbitrary geometric configurations. *IRE Transactions on Electronic Computers*, (2) :260–268, 1961.
- [FT99] Fabien Feschet and Laure Tougne. Optimal time computation of the tangent of a discrete curve : Application to the curvature. In *International Conference on Discrete Geometry for Computer Imagery*, pages 31–40. Springer, 1999.
- [Get12] Pascal Getreuer. Chan-Vese Segmentation. *Image Processing On Line*, 2 :214–224, 2012.
- [Hou59] Paul VC Hough. Machine analysis of bubble chamber pictures. In *International conference on high energy accelerators and instrumentation*, volume 73, 1959.
- [Hou62] Paul VC Hough. Method and means for recognizing complex patterns. Technical report, 1962.
- [IPO16] IPOL : Image processing on line. <http://www.ipol.im>, 2016.
- [ISO15] ISO 9000 :2015 - Systèmes de management de la qualité – Principes essentiels et vocabulaire. http://www.iso.org/iso/fr/catalogue_detail?csnumber=45481, 2015.
- [KE09] Bertrand Kerautret and Philippe Even. Blurred Segments in Gray Level Images for Interactive Line Extraction. In P. Wiederhold and R. P. Barneva, editors, *Proceedings of the 13th IWCI*, volume 5852 of *LNCS*, pages 176–186. Springer, November 2009.
- [KKNW12] Michal Kazimierz Kowalczyk, Bertrand Kerautret, Benoît Naegel, and Jonathan Weber. Revisiting Component Tree Based Segmentation Using Mea-

- ningful Photometric Informations. In Leonard Bolc, Ryszard Tadeusiewicz, Leszek J. Chmielewski, and Konrad W. Wojciechowski, editors, *Computer Vision and Graphics - International Conference, ICCVG 2012, Warsaw, Poland, September 24-26, 2012. Proceedings*, volume 7594 of *Lecture Notes in Computer Science*, pages 475–482. Springer, 2012.
- [KL09] Bertrand Kerautret and J-O Lachaud. Curvature estimation along noisy digital contours by approximate global optimization. *Pattern Recognition*, 42(10) :2265–2278, 2009.
- [KL12] Bertrand Kerautret and Jacques-Olivier Lachaud. Meaningful scales detection along digital contours for unsupervised local noise estimation. *IEEE transactions on pattern analysis and machine intelligence*, 34(12) :2379–2392, 2012.
- [KL14] Bertrand Kerautret and Jacques-Olivier Lachaud. Meaningful Scales Detection : an Unsupervised Noise Detection Algorithm for Digital Contours. *Image Processing On Line*, 4 :98–115, 2014.
- [KWT88] Michael Kass, Andrew Witkin, and Demetri Terzopoulos. Snakes : Active contour models. *International journal of computer vision*, 1(4) :321–331, 1988.
- [LVdV07] Jacques-Olivier Lachaud, Anne Vialard, and François de Vieilleville. Fast, accurate and convergent tangent estimation on digital contours. *Image and Vision Computing*, 25(10) :1572–1587, 2007.
- [LYLL15] Xiaohu Lu, Jian Yao, Kai Li, and Li Li. Cannylines : A parameter-free line segment detector. In *Image Processing (ICIP), 2015 IEEE International Conference on*, pages 507–511. IEEE, 2015.
- [Mel87] Avraham A Melkman. On-line construction of the convex hull of a simple polyline. *Information Processing Letters*, 25(1) :11–12, 1987.
- [Mey88] F Meyer. Skeletons in digital spaces. *Image analysis and mathematical morphology*, 2 :257–296, 1988.
- [Mey92] Fernand Meyer. Color image segmentation. In *Image Processing and its Applications, 1992., International Conference on*, pages 303–306. IET, 1992.

- [MGK00] Jiri Matas, Charles Galambos, and Josef Kittler. Robust detection of lines using the progressive probabilistic hough transform. *Computer Vision and Image Understanding*, 78(1) :119–137, 2000.
- [MH80] David Marr and Ellen Hildreth. Theory of edge detection. *Proceedings of the Royal Society of London B : Biological Sciences*, 207(1167) :187–217, 1980.
- [MS89] David Mumford and Jayant Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *Communications on pure and applied mathematics*, 42(5) :577–685, 1989.
- [NC06] Laurent Najman and Michel Couprie. Building the component tree in quasi-linear time. *IEEE Transactions on Image Processing*, 15(11) :3531–3539, 2006.
- [NIK91] S. K. Nayar, K. Ikeuchi, and T. Kanade. Surface reflection : physical and geometrical perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(7) :611–634, July 1991.
- [ope16] OpenCV : Open Source Computer Vision Library . <http://opencv.org>, 2016.
- [PNR⁺11] Nicolas Passat, Benoît Naegel, François Rousseau, Mériam Koob, and Jean-Louis Dietemann. Interactive segmentation based on component-trees. *Pattern Recognition*, 44(10) :2539–2554, 2011.
- [Rev91] Jean-Pierre Reveillès. *Géométrie discrete, calcul en nombres entiers et algorithmique*. PhD thesis, 1991.
- [Ros74] Azriel Rosenfeld. Digital straight line segments. *IEEE Trans. Comput*, 23(12) :1264–1269, 1974.
- [RU13] Ihsan Rabbi and Sehat Ullah. A survey on augmented reality challenges and tracking. *Acta Graphica znanstveni časopis za tiskarstvo i grafičke komunikacije*, 24(1-2) :29–46, 2013.
- [SC15] Haldo Spontón and Juan Cardelino. A Review of Classic Edge Detectors. *Image Processing On Line*, 5 :90–123, 2015.

- [SF68] Irwin Sobel and Gary Feldman. A 3x3 isotropic gradient operator for image processing. *a talk at the Stanford Artificial Project in*, pages 271–272, 1968.
- [SM00] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8) :888–905, 2000.
- [SOG98] Philippe Salembier, Albert Oliveras, and Luis Garrido. Antiextensive connected operators for image and sequence processing. *IEEE Transactions on Image Processing*, 7(4) :555–570, 1998.
- [SS88] S Sher and JA Sethian. Fronts propagating with curvature dependent speed : algorithms based on hamilton-jacobi formulation. *Journal of Computational Physics*, 79(1) :12–49, 1988.
- [TAG10] Cihan Topal, Cuneyt Akinlar, and Yakup Genc. Edge drawing : a heuristic approach to robust real-time edge detection. In *Pattern Recognition (ICPR), 2010 20th International Conference on*, pages 2424–2427. IEEE, 2010.
- [vGJMR12] Rafael Grompone von Gioi, Jérémie Jakubowicz, Jean-Michel Morel, and Gregory Randall. Lsd : a line segment detector. *Image Processing On Line*, 2 :35–55, 2012.
- [VS91] Luc Vincent and Pierre Soille. Watersheds in digital spaces : an efficient algorithm based on immersion simulations. *IEEE transactions on pattern analysis and machine intelligence*, 13(6) :583–598, 1991.