



Inspection d'assemblages aéronautiques par vision 2D/3D en exploitant la maquette numérique et la pose estimée en temps réel

Hamdi Ben Abdallah

► To cite this version:

Hamdi Ben Abdallah. Inspection d'assemblages aéronautiques par vision 2D/3D en exploitant la maquette numérique et la pose estimée en temps réel. Matériaux. Ecole des Mines d'Albi-Carmaux, 2020. Français. NNT : 2020EMAC0001 . tel-02902981

HAL Id: tel-02902981

<https://theses.hal.science/tel-02902981>

Submitted on 20 Jul 2020

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.



THÈSE

en vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

délivré par

IMT - École Nationale Supérieure des Mines d'Albi-Carmaux

présentée et soutenue par

Hamdi BEN ABDALLAH

le 4 février 2020

**Inspection d'assemblages aéronautiques par vision
2D/3D, en exploitant la maquette numérique et la
pose estimée en temps réel**

École doctorale et discipline ou spécialité :

EDSYS : Informatique et Robotique

Unité de recherche :

Institut Clément Ader, UMR CNRS 5312, IMT Mines Albi

Directeurs de thèse :

Jean-José Orteu, Professeur, IMT Mines Albi (Directeur de Thèse)

Igor Jovancevic, Ingénieur de Recherche, DIOTA (Co-directeur)

Autres membres du jury :

Olivier Aubreton, Maître de Conférences HDR, Université de Bourgogne Franche-Comté
(*Rapporteur*)

Sylvie Treuillet, Maître de Conférences HDR, Université d'Orléans (*Rapporteur*)

Michel Devy, Directeur de Recherche Émérite, LAAS-CNRS (*Examineur*)

François Goulette, Professeur, Mines ParisTech (*Président*)

Ludovic Brèthes, Ingénieur de Recherche, DIOTA (*Invité*)

Je remercie la Région Occitanie, IMT Mines Albi et la société DIOTA pour avoir co-financé ma thèse.





Remerciements

CETTE thèse est le fruit d'une collaboration fructueuse entre l'Institut Clément Ader (ICA) UMR CNRS 5312 et la Sté DIOTA. Plus particulièrement entre l'équipe Métrologie, Identification Contrôle et Surveillance (MICS) de l'ICA et l'équipe R&D Contrôle automatique de DIOTA. Je souhaiterais remercier les directeurs de m'avoir accueilli au sein de leur entité et de m'avoir permis de participer à cette aventure.

Ces trois ans de travaux ont été une aventure particulièrement enrichissante sur le plan scientifique et humain. Il m'est très difficile de mentionner tout le monde car c'est grâce à l'aide de nombreuses personnes que j'ai pu mener cette thèse au bout.

Mes **plus sincères** remerciements vont à mon directeur de thèse Jean-José Orteu, professeur de l'Institut Mines-Télécom et chercheur à l'ICA, qui m'a été une source d'encouragements comme d'inspiration à travers ses qualités pédagogiques et scientifiques, son soutien, sa grande gentillesse et sa patience. Ce travail n'aurait pas pu être mené à bien sans sa disponibilité et ses nombreux conseils durant ces trois années de thèse.

Je voudrais également remercier mon co-directeur de thèse Igor Jovancevic pour le temps qu'il m'a consacré. Ses remarques pertinentes m'ont permis de mener à bien ces travaux de thèse.

L'ambiance très chaleureuse chez DIOTA a contribué à la réussite de ma thèse. Je tiens à remercier mes collègues : Igor, Ludovic, Yannick, Benoit, Nour, Pierre, Djamel et Xavier pour m'avoir soutenu et encouragé ainsi que pour tous les bons moments partagés à leur côté. Travailler avec eux a été un réel plaisir.

Merci à tous ceux qui m'ont aidé lors de la rédaction de ce manuscrit : Romain, Morgane, Maxime, Stella, William, Laurie, Emmanuel et Florian.

Je remercie sincèrement tous les collègues du laboratoire ICA pour leur soutien et leurs encouragements, notamment Maxime, Fatna, Benjamin, Mehdi, Camille, Anh Tuan, Clément (D).

Je tiens également à remercier tous les autres membres du laboratoire : enseignants-chercheurs, techniciens, doctorants, assistantes et stagiaires, avec lesquels j'ai eu beaucoup de plaisir à travailler au laboratoire et je voudrais remercier tous les collègues pour leur gentillesse, leur accueil et l'ambiance de travail.

Je souhaite remercier tous les professeurs de l'Université Paul Sabatier de Toulouse pour m'avoir accordé leur confiance et m'avoir permis d'encadrer des travaux pratiques durant mes années de thèse, notamment Lubomir Gavrilov, Francois Chaperon et Luca Amodei.

Je tiens à exprimer ma plus sincère gratitude envers toute ma famille et mes amis pour leur soutien sans faille durant ma thèse. Je tiens à remercier plus particulièrement ma mère de m'avoir soutenu, encouragé, porté, poussé depuis le début. Je n'oublie pas non plus de remercier tous mes amis qui m'ont apporté leur soutien moral pendant ces années d'études, je les en remercie sincèrement.

Je souhaiterais remercier Sanaa pour sa bonne humeur, son soutien, ses aventures et qui a su me redonner le sourire et la motivation pour terminer cette thèse.

Je ne peux malheureusement pas citer toutes les personnes que j'ai rencontrées durant mon parcours et qui ont contribué d'une façon ou d'une autre, de près ou de loin, à l'aboutissement de cette thèse. Je tiens à les remercier très sincèrement.

Last but not least, je souhaite vous remercier, vous, lecteur de ces quelques lignes, pour l'intérêt que vous portez à mon manuscrit.

— Hamdi, le 25 février 2020

Hamdi BEN ABDALLAH

*À ma mère Fatna et à mon père Mohamed,
à mes frères et soeurs,
à toute ma famille.*

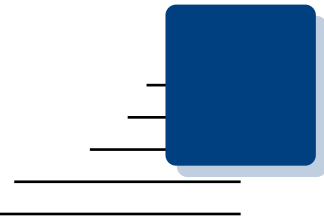


Table des matières

<i>Remerciements</i>	v
<i>Table des matières</i>	ix
<i>Table des figures</i>	xv
<i>Liste des tableaux</i>	xxv

Glossaire	1
1 Introduction générale	3
1.1 Contexte et problématique	4
1.2 Méthodes d'inspection automatique par vision 2D/3D	8
1.2.1 Inspection à partir d'images 2D	8
1.2.1.1 Inspection à partir d'images 2D sans exploiter un modèle CAO	9
1.2.1.2 Inspection à partir d'images 2D en exploitant un modèle CAO	10
1.2.2 Inspection à partir de nuages de points 3D	12
1.3 Méthodes de segmentation 3D de formes cylindriques dans un nuage de points	13
1.3.1 Segmentation de formes cylindriques rigides	14
1.3.2 Segmentation de formes cylindriques non-rigides	14
1.4 Conclusion sur les méthodes existantes	15
1.5 Solutions proposées	16
1.5.1 Inspection à partir d'images 2D	16
1.5.1.1 Inspection via une tablette portée par un opérateur	16
1.5.1.2 Inspection via un robot équipé de capteurs	18
1.5.2 Inspection à partir de nuages de points 3D	22
1.6 Localisation de la caméra	23

1.6.1	Les méthodes basées marqueurs	23
1.6.2	Les méthodes basées modèle reconstruit	25
1.6.3	Les méthodes basées modèle CAO	26
1.6.4	Synthèse des techniques permettant de localiser la caméra dans une scène complexe	27
1.6.5	La méthode disponible dans le cadre de cette thèse	27
1.7	Structuration du manuscrit	28
2	Sélection automatique des points de vue	29
2.1	Introduction et problématique	30
2.2	Méthodes de choix automatique de points de vue	31
2.3	Schéma général de la méthode proposée	32
2.4	Sphère de visibilité et points de vue	33
2.4.1	Centre de la sphère de visibilité	34
2.4.2	Choix du repère associé à la sphère de visibilité	35
2.5	Génération des points de vue candidats et des images synthétiques associées	36
2.5.1	Nombre de points de vue \mathcal{N}	37
2.5.2	Images synthétiques générées depuis la sphère de visibilité	38
2.6	Fonction score f_{score} pour sélectionner le meilleur point de vue	39
2.6.1	Critère de visibilité $f_{\text{visibilité}}$	39
2.6.2	Autres critères adaptés au type d'inspection	41
2.7	Exemples de choix de meilleur point de vue	42
2.8	Exploitation des points de vue dans le mode d'inspection interactive à l'aide de la réalité augmentée	44
2.9	Conclusion	45
3	Inspection à partir d'images 2D	47
3.1	Introduction	48
3.2	Schéma général de la méthode proposée	50
3.2.1	Inspection via une tablette portée par un opérateur	50
3.2.2	Inspection robotisée	51
3.3	Génération de contours 3D (edgelets)	52
3.4	Estimation de la pose de la caméra	53
3.5	Mise en correspondance 2D/3D	53
3.5.1	Projection 3D/2D	53
3.5.1.1	Projection des edgelets sur l'image 2D	53
3.5.1.2	Erreur de reprojection	54
3.5.2	Génération de lignes de recherche pour la mise en correspondance 2D/3D	55
3.6	Filtrage des edgelets indésirables	56
3.6.1	Gestion des contours parasites	57
3.6.1.1	Définitions	57
3.6.1.2	Filtrage des contours parasites	59
3.6.2	Gestion des occultations	61
3.6.3	Orientation des gradients	64

3.7	Prise de décision	65
3.7.1	Score de similarité des formes entre les edgelets filtrés et les points de contour 2D appariés	66
3.7.1.1	Contexte de forme	66
3.7.1.2	Mesure de similarité (calcul de distance basé sur le contexte de forme)	66
3.7.2	Taux d'appariement entre les edgelets filtrés et les points de contour 2D appariés	67
3.7.2.1	Pondération d'appariement	67
3.7.2.2	Taux d'appariement	68
3.7.3	Processus de décision	68
3.8	Choix de points de vue dans l'étape de préparation de l'inspection 2D	69
3.9	Résultats expérimentaux	70
3.9.1	Conditions expérimentales	70
3.9.2	Éclairage	71
3.9.3	Présentation des données	72
3.9.3.1	Données utilisées pour la solution robotisée	72
3.9.3.2	Données utilisées pour la solution exploitant une tablette	74
3.9.4	Détection de contours dans l'image	76
3.9.5	Détermination des paramètres optimaux	80
3.9.5.1	Détermination du seuil d'appariement	80
3.9.6	Métriques d'évaluation pour l'analyse de performances	82
3.9.6.1	Matrice de confusion	82
3.9.6.2	Indicateurs de succès	82
3.9.7	Résultats d'inspection	83
3.9.7.1	Inspection via une tablette portée par un opérateur	84
3.9.7.2	Inspection robotisée	85
3.9.7.3	Interprétation des résultats	85
3.10	Conclusion	86
3.10.1	Synthèse des travaux	86
3.10.2	Limitations de la méthode d'inspection proposée et améliorations futures	86
4	Inspection à partir de nuages de points 3D : segmentation 3D des câbles	89
4.1	Introduction	90
4.2	Schéma général de la méthode proposée	92
4.3	Pré-traitements	93
4.3.1	Filtrage du nuage de points 3D	93
4.3.2	Transformation d'un modèle CAO maillé en nuage de points 3D	95
4.3.3	Sous-échantillonnage uniforme du nuage de points	96
4.4	Initialisation de la segmentation 3D d'un câble	98
4.4.1	Alignement des deux nuages	99
4.4.2	Génération d'un point pour initialiser la segmentation	101
4.5	Segmentation 3D d'un câble	103
4.5.1	Recherche du premier sous-cylindre	105

4.5.1.1	Structure de recherche	105
4.5.1.2	Algorithme de recherche	107
4.5.2	Ajustement des paramètres du premier sous-cylindre	107
4.5.2.1	Ajustement de la longueur	108
4.5.2.2	Ajustement de l'axe principal	110
4.5.2.3	Ajustement du rayon	113
4.5.3	Propagation de la recherche/ajustement des sous-cylindres	115
4.6	Résultats expérimentaux	120
4.6.1	Conditions expérimentales	121
4.6.2	Présentation des données	121
4.6.2.1	Modèles CAO disponibles	121
4.6.2.2	Nuages de points synthétiques simulés	121
4.6.2.3	Nuages de points réels acquis par le scanner	123
4.6.3	Évaluation du processus d'ajustement des paramètres	123
4.6.3.1	Métriques d'évaluation	123
4.6.3.2	Comparaison avec les méthodes d'ajustement classiques les plus populaires	125
4.6.4	Évaluation de la méthode de modélisation-segmentation à partir de données synthétiques	129
4.6.5	Évaluation de la méthode de modélisation-segmentation avec des données réelles	132
4.7	Conclusion	136
4.7.1	Synthèse des travaux	136
4.7.2	Perspectives	137

5 Inspection à partir de nuages de points 3D : inspection

3D des câbles	141
5.1	Introduction et problématique	142
5.2	Schéma général de la méthode proposée	143
5.3	Mesure du rayon de courbure	144
5.3.1	Rayon de courbure minimal des câbles	144
5.3.2	Mesure du rayon de courbure à partir des résultats de modélisation	145
5.3.3	Prise de décision	148
5.4	Détection d'interférences	149
5.4.1	Les entrées	149
5.4.2	Concept d'interférence	150
5.4.3	Regroupement du nuage de points $\overline{\mathcal{P}}_s$	150
5.4.3.1	Méthode de regroupement basée sur la distance Euclidienne	151
5.4.3.2	Résultats	152
5.4.4	Détecteur d'interférences	153
5.5	Résultats expérimentaux	157
5.5.1	Conditions expérimentales	157
5.5.2	Présentation des données	157
5.5.3	Évaluation de la méthode de calcul du rayon de courbure des câbles	157
5.5.3.1	Métriques d'évaluation pour l'analyse de performances	159

5.5.3.2 Résultats de l'évaluation	159
5.5.4 Évaluation de la méthode de détection d'interférences	160
5.6 Conclusion	163
5.6.1 Synthèse des travaux	163
5.6.2 Améliorations futures	163
Conclusion générale	165
Travaux réalisés	165
Perspectives	168

A Extraction des éléments de la maquette numérique . .	171
B Détection automatique des connecteurs par Machine Learning	173
B.1 Choix de l'architecture de réseau	174
B.2 Augmentation de données d'apprentissage	174
B.3 Résultats de détection	174
B.4 Prise de décision	175
Bibliographie	177

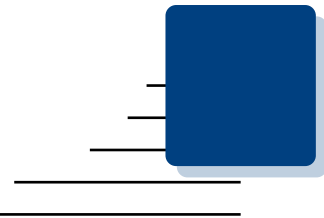


Table des figures

1.1	Un exemple d'assemblage mécanique aéronautiques complexe : (gauche) un moteur d'avion LEAP (Leading Edge Aviation Propulsion) de la famille CFM56, qui se compose de plusieurs dizaines de milliers d'éléments à contrôler, (droite) images zoomées	6
1.2	Inspection via une tablette, équipée d'une caméra, portée par un opérateur	17
1.3	Solution de contrôle via une tablette équipée d'une caméra	17
1.4	Inspection robotisée	18
1.5	Effecteur de contrôle autonome	19
1.6	Le capteur est équipé de 2 caméras	20
1.7	Système d'éclairage autour de la caméra d'inspection	21
1.8	Scanner 3D Ensenso N35	22
1.9	Localisation basée marqueurs : (a) exemple de trois marqueurs avec des motifs différents, et (b) localisation basée sur l'utilisation des marqueurs placés dans l'environnement	24
1.10	Exemple de résultat de détection de marqueurs sur une image en niveaux de gris	24
2.1	(a) contrôle semi-automatisé, (b) contrôle complètement automatisé .	30
2.2	Schéma général de la méthode proposée	33
2.3	Sphère de visibilité. Les points rouges représentent l'ensemble des points de vue candidats qui ont été générés	34

2.4	Centre de masse du modèle CAO : (a) maillage triangulaire d'un modèle CAO, les points violets et les lignes bleues claires représentent respectivement les sommets et les arêtes du maillage, (b) le point rouge représente le centre de masse de ce modèle CAO	35
2.5	Repère associé à la sphère de visibilité obtenu à partir du centre de masse de l'élément à inspecter et de l'ACP sur le modèle CAO de l'élément : (a) résultat de l'ACP obtenu sur le modèle CAO. Les 3 vecteurs (rouge, vert et bleu) représentent les 3 composantes principales estimées par l'ACP, et la sphère noire représente le centre de masse du modèle CAO. La composante qui contient la variance la plus grande est en rouge, (b) sphère de visibilité orientée résultante	36
2.6	Certaines sphères de visibilité générées avec différentes valeurs d'espacement angulaire (les points violets sont les points de vue candidats) .	37
2.7	Images synthétiques générées depuis la sphère de visibilité : (1 ^{ère} ligne) modèle CAO 3D, (2 ^{ème} ligne) sphères de visibilité. Les points rouges représentent la position de la caméra virtuelle, et (3 ^{ème} ligne) images synthétiques générées à partir du modèle CAO	38
2.8	Exemple de résultat de visibilité obtenu sur trois vues différentes : il s'agit d'inspecter l'élément d'indice i et de couleur violette avec le codage $(R, V, B) = (128, 17, 156)$	39
2.9	Distribution de la visibilité sur la sphère : (a) modèle CAO 3D et (b) scores de visibilité $f_{\text{visibilité}}$ normalisés (entre 0 et 1) (rouge : scores très faibles et bleu : scores maximaux). Les points de vue en rouge sont des points « inaccessibles » en raison d'une occultation liée à la nature de l'assemblage mécanique	40
2.10	Occultation partielle : une zone de l'élément à inspecter (rouge) est occultée ou masquée par un autre élément (vert semi-transparent) . .	41
2.11	Exemple de résultat obtenu pour l'inspection 3D du câble dans le rectangle rouge : (a) modèle CAO 3D, (b-i) images synthétiques générées depuis les différents points de vue de la sphère de visibilité .	42
2.12	Point de vue fronto-parallèle dans le cas de l'inspection d'une vis à tête plate : (a) le point de vue en vert représente le point de vue fronto-parallèle obtenu en fixant l'angle φ à zéro dans la sphère de visibilité. Le vecteur en noir représente la composante principale du modèle ACP, (b) image synthétique générée à partir du modèle CAO et à partir de ce point de vue, et (c) image réelle obtenue à partir de ce point de vue pour inspecter la vis à tête plate dans le cadre rouge .	43

2.13	Point de vue latéral dans le cas d'une mesure de distance minimale entre 1 support rigide et 1 câble : (a) les points de vue en bleu sont les vues latérales obtenues en fixant l'angle φ à 45° par rapport à la composante principale de l'élément à inspecter ; le point de vue vert est le meilleur point de vue selon la fonction score. Le vecteur en noir représente la composante principale du modèle ACP, (b) image synthétique obtenue à partir du modèle CAO et à partir de ce point de vue, et (c) image obtenue à partir de ce point de vue pour mesurer la distance entre le support dans le cadre rouge et le câble le plus proche	44
2.14	Guidage de l'utilisateur vers le bon angle de prise de vue à l'aide de réalité augmentée (voir les flèches bleues). L'élément à inspecter est dans le rectangle rouge	45
3.1	Exemple de problématique d'inspection : (a) panneau de test constitué d'éléments à contrôler (les éléments A et B ont été correctement assemblés, l'élément C est absent), (b) modèle CAO du panneau . .	49
3.2	Schéma général de la méthode proposée	50
3.3	Schéma général de la méthode proposée (inspection via une tablette, équipée d'une caméra, portée par un opérateur)	51
3.4	Schéma général de la méthode proposée (inspection robotisée)	52
3.5	Génération des edgelets : (a) exemple d'un modèle CAO, (b) edgelets générés à partir de ce modèle CAO	53
3.6	Projection des edgelets (points 3D) sur l'image d'inspection (image 2D) : (a) modèle CAO du panneau de test, (b) élément A présent sur le panneau (en rouge sur le modèle CAO), (c) élément C absent du panneau (en bleu sur le modèle CAO)	54
3.7	Erreur de reprojection : distance 2D entre la projection des edgelets prédits et des contours réels extraits dans l'image	55
3.8	Ligne de recherche	56
3.9	Génération de lignes de recherche (images zoomées) : (a) projection des edgelets sur l'image d'inspection, (b) un exemple de ligne de recherche avec le vecteur normal sortant dirigé vers l'extérieur de la surface (en bleu) et le vecteur normal rentrant dirigé vers l'intérieur de la surface (en rouge), (c) génération de lignes de recherche	56
3.10	Exemple de contours parasites. Les lignes de recherche correspondant aux appariements piégés par des contours parasites sont tracées en rouge dans deux cas : (a) cas où l'assemblage mécanique est composé d'un objet très proche de l'élément à inspecter, (b) cas où un changement de point de vue de la caméra provoque l'apparition de contours parasites.	58

3.11	Exemple d'image contexte : (a) image réelle, (b) modèle CAO, (c) image contexte de l'élément à inspecter (cas de l'élément bleu) : elle contient tous les éléments du modèle CAO sauf l'élément à inspecter .	59
3.12	Processus de filtrage des contours parasites	60
3.13	Filtrage de contours parasites : (a) image contexte de l'élément à inspecter, (b) gradient de l'image contexte avec les edgelets conservés (vert) et les edgelets rejetés (rouge)	61
3.14	Occultation partielle : une zone de l'élément à inspecter (rouge) est occultée ou masquée par un autre élément (vert semi-transparent) . .	61
3.15	Gestion des occultations : (a) image synthétique générée à partir du modèle CAO par une méthode de type CATIA [®] Composer, (b) image synthétique générée à partir du modèle CAO par la méthode d'indice de rendu de couleur, (c) masque généré par une méthode d'indice de rendu de couleur (la zone avec couleur violette représente la région à inspecter), (d) les edgelets projetés sur l'image d'inspection (e) les edgelets projetés sur la région d'intérêt spécifiée par le masque	63
3.16	Exemple de dégradation due à des aberrations géométriques (rayure sur la plaque). À gauche : projection des edgelets de l'élément à inspecter sur l'image 2D, à droite une zone de l'image agrandie 3200 fois	64
3.17	Critère de similarité d'orientation. (1 ^{ère} colonne) projection des edgelets de l'élément à inspecter sur l'image 2D, (2 ^{ème} colonne) une zone de l'image agrandie 3200 fois. (a) un exemple de point de contour c_i apparié (par erreur) avec un edgelet e_i ayant une orientation différente ($\theta_i = 120^\circ$), (b) un exemple de point de contour c_i apparié avec un edgelet e_i ayant une orientation similaire ($\theta_i = 0^\circ$)	65
3.18	Similarité des formes entre les edgelets filtrés et les contours détectés à partir de l'image : (a) exemple de deux formes jugées comme similaires et (b) exemple de deux formes jugées comme différentes	67
3.19	Gaussienne de pondération de la distance entre un edgelet e_i et deux exemples de contours 2D appariés (c_j, c_k). Typiquement, plus l'edgelet est proche du contour apparié et plus le facteur de pondération (poids w) est faible et inversement	68
3.20	Processus de décision	69
3.21	Système d'éclairage autour de la caméra d'inspection	71
3.22	Assemblages mécaniques utilisés dans le mode robotisé : (a) moteur LEAP de la famille CFM56, (b) et (c) deux panneaux de test différents constitués d'éléments à contrôler	73
3.23	Exemples de données acquises par le robot sur le moteur (figure 3.22a) utilisées pour évaluer notre approche dans le contexte robotisé	74

3.24	Exemples de données acquises par le robot sur les deux panneaux de test : Panneau 1 (figure 3.22b) et Panneau 2 (figure 3.22c), utilisées pour évaluer notre approche dans le contexte robotisé	74
3.25	Assemblages mécaniques utilisés dans le mode manuel : (a) un panneau test constitué d'éléments à contrôler, (b) une partie d'assemblage de l'avion Airbus A350 XWB et (c) une partie d'assemblage de l'avion 34 Rafale	75
3.26	Résultats de l'application du filtre de Canny avec différents seuils détaillés dans le tableau 3.1	76
3.27	Problèmes rencontrés avec le détecteur de contours de Canny. Le détecteur de contour de Canny échoue : (a) et (b) dans le cas de surfaces brillantes, de nombreuses détections manquées, et (c) et (d) dans le cas de surfaces texturées, beaucoup de fausses détections . . .	78
3.28	Comparaison qualitative de l'algorithme de Canny et de l'algorithme de détection de contours à base de réseaux de neurones. (1 ^{ère} colonne) image initiale, (2 ^{ème} colonne) carte de contours calculée par Canny, (3 ^{ème} colonne) carte de contours calculée par l'algorithme de détection de contours à base de réseaux de neurones. Nous constatons que l'algorithme de détection de contours à base de réseaux de neurones est capable de détecter beaucoup mieux les contours sur les surfaces brillantes et sur les surfaces texturées, avec une plus grande robustesse par rapport au bruit que l'algorithme de Canny	79
3.29	Détermination du seuil d'appariement (seuil _{appariement}) : (a) Panneau 1 constitué d'éléments à contrôler, (b) modèle CAO du Panneau 1, (c-e) pour les 3 éléments (l'élément A a été monté correctement, l'élément B a été monté à l'envers et l'élément C est absent) : la courbe représente le taux d'appariement calculé à partir de 100 images acquises avec des points de vue différents. La ligne bleu représente le taux moyen d'appariement. La ligne rouge représente le seuil d'appariement qui a été calculé pour cette expérience, en prenant le minimum ou le maximum de la courbe, et (f) résultats d'inspection (vert : élément monté correctement et rouge : éléments mal montés)	81
4.1	Exemple d'inspection : une scène complexe constituée des différents câbles à inspecter	90
4.2	Nuage de points acquis par le scanner (vu sous 2 angles différents) : la vue de droite montre que les données sont éparses et que l'on ne dispose que d'une faible proportion de la surface du câble cylindrique à segmenter (à peine 1/3 de la circonférence)	91
4.3	Schéma général de la méthode proposée	92
4.4	Schéma général de la phase de pré-traitements	93
4.5	Distance moyenne par rapport aux k -plus proches voisins (k NN) (nuage de points d'entrée en rouge, nuage de points filtré en vert)	94

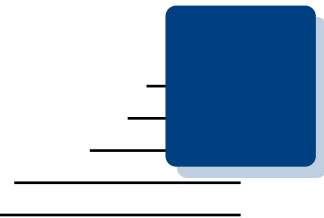
4.6	Interprétation des coordonnées barycentriques en termes de rapports d'aires	96
4.7	Comparaison quantitative entre la méthode proposée et les deux approches le plus fréquemment rencontrées dans la littérature pour l'échantillonnage du maillage : (a) maillage triangulaire d'un modèle CAO, les points verts et les lignes bleues représentent respectivement les sommets et les arêtes du maillage, (b) nuage de points généré avec la méthode d'échantillonnage aléatoire naïf, (c) nuage de points généré avec la technique de lancer de rayons, et (d) nuage de points généré avec notre méthode d'échantillonnage pondéré	96
4.8	Sous-échantillonnage du nuage de points : (a) une série de voxels entourant le nuage de points d'entrée; l' <i>octree</i> résultant est représenté en noir, (b-d) quelques nuages de points générés avec différentes valeurs de résolution détaillées dans le tableau 4.1	97
4.9	Les points de raccordement des câbles	98
4.10	Schéma général de la phase d'alignement proposée	99
4.11	Exemple de résultat obtenu : alignement du modèle CAO (vert) sur le nuage de points (rouge)	100
4.12	Projection des résultats d'écarts entre le modèle CAO du câble (après l'alignement en deux étapes : alignement global puis alignement local) et le nuage de points sur l'image 2D (bleu : différences faibles et rouge : différences importantes).	101
4.13	Sélection de points dans le nuage synthétique : (a) résultat d'alignement (en deux étapes), alignement global puis local, (b) valeurs des écarts entre le modèle CAO du câble et le nuage de points (bleu : différences faibles) , (c) un exemple des 10 meilleurs points trouvés (qui ont le plus grand nombre de voisins dans le nuage réel)	102
4.14	Processus d'ajustement d'un point d'initialisation au centre du câble. Étape 1 : sélection de tous les points qui sont dans une sphère centrée sur le point q_i ; étape n : fin du processus itératif de recherche du point q_c (point centré sur le câble)	103
4.15	Câble reconstruit à l'aide de cylindres élémentaires (sous-cylindres) .	104
4.16	Les étapes de segmentation d'un câble	105
4.17	(a) Exemple de deux points antipodaux A (en bleu) et B (en rouge) sur la surface de la sphère, (b) ensemble de points antipodaux répartis uniformément sur toute la surface de la sphère source selon la plage de colatitude θ (comprise entre 0 et π) et de longitude δ (comprise entre 0 et 2π)	105
4.18	Structure de recherche basée antipodes d'une sphère : (a) cylindre rotatif avec trois degrés de liberté (trois rotations R_u, R_v, R_w), (b) ensemble de points antipodaux répartis uniformément sur toute la surface de la sphère pour l'étape d'initialisation	106

4.19	Certaines sphères de recherche générées avec différentes valeurs d'espacement angulaire (les points bleus sont les antipodes des points rouges)	107
4.20	Ajustement des paramètres du sous-cylindre	108
4.21	Maillage triangulaire d'un modèle CAO du câble, les points rouges et les lignes bleues représentent respectivement les sommets et les arêtes du maillage. On voit que selon la courbure du câble, chaque sous-câble a une longueur différente (par exemple $l_i \neq l_j$)	109
4.22	Ajustement de la longueur : (a) cylindre ajusté sur le nuage de points correspondant au sous-câble, (b) projection du nuage de points situé à l'intérieur du cylindre de recherche sur l'axe principal, (c) graphique à barres indiquant le nombre des points par bin, et (d) sous-cylindre ajusté	109
4.23	Création d'un nuage de points virtuel à partir du modèle de sous-cylindre : (a) le modèle de sous-cylindre (en vert) et le nuage de points du sous-câble (en rouge), (b) le nuage de points virtuel (en vert) créé avec la même résolution que le nuage du sous-câble (en rouge)	110
4.24	Calcul de la distance d_i (ligne bleue) entre un point p_i du sous-câble (en rouge) et son plus proche voisin q_i de sous-cylindre (en vert) . . .	111
4.25	Évaluation de l'erreur quadratique moyenne de la distance MSE_d en fonction du nombre d'itérations entre le sous-cylindre (s-cy ₁ en vert) et le sous-câble (s-c ₁ en rouge) : (a) après 1 itération, (b) après 5 itérations, (c) après 10 itérations. La deuxième ligne fournit les histogrammes d'erreurs de distance : (d) après une 1 itération, (e) après 5 itérations, (f) après 10 itérations	111
4.26	Comparaison quantitative entre la méthode utilisée (ICP point à point avec 5 itérations) et la méthode de calcul de la moyenne des vecteurs normaux. (a) exemple des normales estimées (lignes noires) sur un sous-câble (nuage de points en rouge), (b) estimation par noyau de l'erreur de distance entre les deux méthodes	113
4.27	Ajustement du rayon	114
4.28	Une autre méthode pour estimer le rayon du cylindre. Si l'on projette les points rouges sur un plan π perpendiculaire à l'axe du cylindre, on obtient la section du cylindre, c'est-à-dire un cercle	114
4.29	Propagation du cycle k au cycle $k + 1$: (1 ^{ère} colonne) cylindre ajusté sur le nuage de points correspondant au sous-câble, (2 ^{ème} colonne) projection du nuage de points situé à l'intérieur du cylindre de recherche sur l'axe principal, (3 ^{ème} colonne) graphique à barres indiquant le nombre de points par bin, et (4 ^{ème} colonne) sous-cylindre ajusté . . .	115
4.30	Liaison rotule entre deux sous-cylindres consécutifs (s-cy _k et s-cy _{k+1})	116

4.31	(a) Structure de recherche utilisée pour construire le sous-cylindre initial s_{c_1} , (b) structure de recherche simplifiée après la première itération. Le vecteur directeur de la sphère de recherche de la deuxième itération est initialisé au vecteur directeur du cylindre de la première itération (u_0)	116
4.32	Exemple de résultat : (a) image 2D de la scène, (b) résultat de modélisation du câble, composé de 13 sous-cylindres (12 sous-cylindres en vert et un trou détecté dû à un chevauchement avec un autre câble occultant en bleu), (c) résultat de segmentation du câble (le nuage de points segmenté en vert)	118
4.33	Illustration des résultats de propagation obtenus dans l'exemple de la figure 4.32 par une succession (ou enchaînement) de 13 sous-cylindres consécutifs, étiquetés $s\text{-}cy_1$ à $s\text{-}cy_{13}$ (les différentes couleurs représentent les différents sous-cylindres)	119
4.34	Exemple d'un assemblage mécanique utilisé : (a) moteur LEAP de la famille CFM56, (c) une partie extraite du moteur LEAP contenant 20 câbles différents	122
4.35	Exemples de nuages de points synthétiques générés avec différentes résolutions	122
4.36	Exemple d'un nuage de points réel acquis par le scanner 3D : (a) image 2D de la scène, (b) nuage de points coloré (avec information de couleur), (c) nuage de points sans information de couleur	124
4.37	Mesure de l'angle entre l'axe principal v du cylindre connu (cylindre vert) et l'axe principal \tilde{v} d'un faux cylindre estimé (cylindre bleu). Le cylindre vert représente le cylindre connu (vérité terrain) qui correspond aux vrais points 3D (points rouges). Le cylindre bleu représente un faux cylindre qui a été estimé par erreur à partir de points 3D qui correspondent à des points de bruit (points bleus). . .	125
4.38	Exemples de jeux de données utilisés et les résultats obtenus : (1 ^{ère} ligne) cylindres synthétiques, (2 ^{ème} ligne) tuyau synthétique, (3 ^{ème} ligne) modèle d'arbre acquis par un système TLS. Les cylindres d'entrée sont en rouge et les cylindres ajustés par notre méthode sont en vert .	127
4.39	Exemple des modèles ACP obtenus sur : (a) un cylindre synthétique, (b) même cylindre avec bruit ajouté ($\sigma = 0.4$), (c) même cylindre avec données manquantes. Les 3 vecteurs (rouge, bleu et vert) représentent les 3 composantes principales estimées par l'ACP, et la sphère noire représente le centroïde du cylindre. On remarque que les données bruitées, manquantes ou incomplètes peuvent corrompre l'estimation de l'ACP	128
4.40	Exemple de données simulées utilisées : (a) modèle CAO, (b-f), nuages de points générés (simulés) à partir du modèle CAO avec différentes résolutions \mathcal{R}_P et différents niveaux du bruit σ	130

4.41	Modèle CAO d'un assemblage mécanique. Ce modèle contient 4 câbles différents étiquetés c_1 à c_4	131
4.42	Évolution de l'erreur quadratique moyenne de : (a) la distance MSE_{distance} et (b) du rayon MSE_{rayon} en fonction des niveaux de bruit (σ)	132
4.43	Un exemple de résultat de segmentation obtenu sur le câble c_1 (voir la figure 4.41) avec : (a) un nuage de points bruité (bruit Gaussien d'écart type $\sigma = 0.75$), (b) un nuage de point fortement bruité (bruit Gaussien d'écart type $\sigma = 1.8$)	132
4.44	Exemples de résultats de modélisation-segmentation obtenus sur les données réelles acquises par le scanner 3D : (a) un câble de rayon 3 mm, (b) un câble de rayon 6 mm, (c) un câble de rayon 7 mm, (d) un câble de rayon 10 mm	133
4.45	Résultats de modélisation en fonction du niveau de bruit (σ)	135
4.46	Évolution de l'erreur quadratique moyenne de : (a) la distance MSE_{distance} et (b) du rayon MSE_{rayon} en fonction du niveau de bruit (σ)	135
4.47	Résultat de modélisation sur un nuage de points fortement bruité ($\sigma = 1.4$). Les rectangles rouges représentent les zones où il y a des valeurs aberrantes (<i>outliers</i>) très proches du câble et, dans ce cas, le bruit a causé une connexion entre les autres câbles et le câble à segmenter	136
4.48	Exemples d'application dans le domaine : (a) industriel [Kum+17], (b) forestier [Hac+14], (c) médical [Ans+17]	138
5.1	Moteur d'avion	142
5.2	Schéma général de la méthode proposée	144
5.3	Rayon de courbure du câble	145
5.4	Calcul du rayon de courbure : (a) un exemple de résultat de modélisation dans lequel chaque sous-cylindre $s\text{-cy}_k$ (en vert) possède un point initial P_k^i et un point final P_k^f représentés en rose, (b) le rayon de courbure correspondant	146
5.5	Ajustement du plan 3D au sens des moindres carrés sur l'ensemble de données P^i (points initiaux) et P^f (points finaux) du modèle final du câble de la figure 5.4a	147
5.6	Les étapes de calcul du rayon de courbure : (a) un exemple de résultat de segmentation-modélisation dans lequel chaque sous-cylindre $s\text{-cy}_k$ (en vert) possède un point initial P_k^i et un point final P_k^f représentés en rose, (b) projection de l'ensemble de données P^i et P^f du modèle final du câble sur le plan 3D ajusté au sens des moindres carrés, et (c) ajustement d'un cercle 2D au sens des moindres carrés sur l'ensemble des points projetés	148

5.7	Les entrées : le nuage de points segmenté \mathcal{P}_s en vert et le nuage de points d'entrée sans le nuage de points segmenté $\overline{\mathcal{P}}_s$ en rouge	149
5.8	Illustration d'une extraction Euclidienne de <i>clusters</i> sur le nuage de points 3D $\overline{\mathcal{P}}_s$: (a) nuage de points d'entrée (le câble \mathcal{P}_s en vert et le nuage de points $\overline{\mathcal{P}}_s$ en rouge), et (b) 31 <i>clusters</i> différents, étiquetés \mathcal{C}_1 à \mathcal{C}_{31} . Chaque cluster est représenté par une couleur différente . . .	153
5.9	Résultats de la détection d'interférences pour une tolérance $d_{\mathcal{T}} = 12$ mm. Le câble segmenté (en vert) est en interférence avec deux <i>clusters</i> \mathcal{C}_5 ($d_5 = 9.6$ mm) et \mathcal{C}_{20} ($d_{20} = 10.2$ mm)	155
5.10	Résultat du processus de détection d'interférences : (a) nuage de points d'entrée, et (b-c) nuage de points segmenté (câble) en vert et les clusters qui causent des problèmes d'interférences (en rouge) pour deux valeurs différentes du seuil de tolérance	156
5.11	Exemple de données simulées utilisées : (a-c) trois câbles synthétiques différents avec trois rayons de courbure différents (r_c)	158
5.12	Modèle CAO d'une scène d'un assemblage mécanique. Ce modèle contient 4 câbles différents étiquetés de c_1 à c_4	160
5.13	Exemple de données simulées utilisées : (a) modèle CAO, (b) nuage de points synthétique généré à partir du modèle CAO, et (c) zoom de la zone encadrée dans le rectangle bleu	161
5.14	Un exemple de résultat, obtenu sur le câble c_1 : (a) résultat de modélisation, composé de 21 sous-cylindres, (b) résultat de segmentation. Le câble segmenté est en vert, et (c) résultat de la détection d'interférences pour une tolérance $d_{\mathcal{T}} = 42$ mm : le câble segmenté (vert) est en interférence avec le câble c_2 (bleu)	162
A.1	Extraction des éléments de la maquette numérique : (a) maquette globale, (b) maquette restreinte à la zone à inspecter, (c) un élément en particulier (un câble), et (d) l'élément avec son environnement proche	172
B.1	Un exemple de résultat obtenu avec le détecteur SSD	175
B.2	Exemple d'inspection d'un connecteur spécifié. (a) image synthétique et région d'intérêt générés à partir de la maquette numérique, (b) région d'intérêt pour la détection du connecteur par Deep Learning .	176
B.3	Quelques exemples d'images qui ont été analysées	176



Liste des tableaux

1.1	Caractéristiques techniques de la caméra d’inspection/localisation . .	17
1.2	Caractéristiques techniques de l’objectif de la caméra d’inspection/lo- calisation	18
1.3	Caractéristiques techniques de la caméra d’inspection	20
1.4	Caractéristiques techniques de l’objectif de la caméra d’inspection . .	20
1.5	Caractéristiques techniques de la caméra de localisation	21
1.6	Caractéristiques techniques de l’objectif de la caméra de localisation .	21
1.7	Caractéristiques techniques du scanner 3D	22
3.1	Les différents paramètres de filtre de Canny utilisés dans les résultats de la figure 3.26	76
3.2	Résultats quantitatifs du taux d’appariement de l’expérience 1 (figure 3.29)	80
3.3	Matrice de confusion	82
3.4	Résultats d’évaluation	84
3.5	Résultats des indicateurs de succès obtenus sur l’ensemble des données	84
3.6	Résultats d’évaluation	85
3.7	Indicateurs de succès obtenus sur l’ensemble des données	85
4.1	Nombre de voxels, nombre de points et résolution obtenus pour les différents nuages de points représentés sur la figure 4.8	97

4.2	Les détails des 13 sous-cylindres : leur rayon, leur longueur, l'angle entre deux axes principaux consécutifs et le sens de propagation, où « + » représente le premier sens et « - » représente le sens opposé. Il est à noter que s-cy ₈ est un trou détecté dû à un chevauchement avec un autre câble occultant (voir la figure 4.32) et son rayon est égal au rayon du sous-cylindre précédent s-cy ₇ .	120
4.3	Résultats de comparaison entre la méthode proposée et les méthodes d'ajustement classiques : RANSAC [Sch+07], MLESAC [Tor+00] et la méthode basée sur l'Analyse en Composantes Principales (ACP) [Lal+06]	126
4.4	Tableau récapitulatif des résultats de segmentation sur les données simulées. MSE_{distance} représente l'erreur quadratique moyenne de la distance entre le nuage de points synthétique et le nuage de points segmenté pour chaque câble présenté sur la figure 4.41	131
4.5	Tableau récapitulatif des résultats de segmentation sur les données réelles dans laquelle r représente le rayon de la vérité terrain et \tilde{r} le rayon moyen estimé	134
5.1	Les entrées	149
5.2	Les paramètres de la méthode Euclidienne d'extraction de clusters	151
5.3	Une liste de 31 <i>clusters</i> différents, étiquetés \mathcal{C}_1 à \mathcal{C}_{31} , et leur détail : libellé, taille et distance d_i par rapport au câble segmenté (\mathcal{P}_s)	154
5.4	Le rayon de courbure minimal $r_c(\text{min})$ autorisé pour chaque câble présenté sur la figure 5.11	158
5.5	Résultats des indicateurs de succès obtenus sur l'ensemble des données	160
5.6	Comparaison des résultats de mesure entre notre méthode et le logiciel CATIA [®] (distances en mm)	162
B.1	Matrice de confusion obtenue	176



Glossaire

- ACP** Analyse en Composantes Principales 14
- ACPR** Analyse en Composantes Principales Robuste 14
- CAO** Conception Assistée par Ordinateur 5
- CNN** Convolutional Neural Network 11
- FAA** *Federal Aviation Administration* 142
- GAN** Generative Adversarial Network 11
- ICA** Institut Clément Ader 5
- ICEA** Association des Ingénieurs de câbles Isolés 145
- ICP** *Iterative Closest Point* 12
- IHM** Interface Homme-Machine 14
- KDE** *Kernel Density Estimation* 113
- kNN** *k-plus proches voisins* xix, 94
- LEAP** *Leading Edge Aviation Propulsion* xv, 6
- LTS** *Least Trimmed Squares* 14
- MLESAC** *Maximum Likelihood Estimator SAmple Consensus* 120
- MSE** *Mean Square Error* 100
- NEC** Code National sur l'Electricité 145
- NURBS** *Non-Uniform Rational Basis Splines* 12
- OCR** *Optical Character Recognition* 66
- OpenCV** *Open Computer Vision* 70
- OpenGL** *Open Graphics Library* 70
- PC** *Personal Computer* 19

PCL *Point Cloud Library* [13](#)
PLM *Product Life-cycle Management* [5](#)
PTZ *Pan-Tilt-Zoom* [11](#)

QCAV *Quality Control by Artificial Vision* [8](#)

RAM *Random Access Memory* [70](#)
RANSAC *RANdom SAmple Consensus* [13](#)
R-FCN *Region-based Fully Convolutional Networks* [174](#)
RGB *Red, Green, Blue* [13](#)
RGB-D *Red, Green, Blue and Depth* [12](#)
RGC-sm *Right Generalized Cylinder State Model* [14](#)
ROS *Robot Operating System* [51](#)

SIFT *Scale-Invariant Feature Transform* [26](#)
SLAM *Simultaneous Localization And Mapping* [25](#)
SSD *Single Shot MultiBox Detector* [174](#)
SURF *Speeded Up Robust Feature* [26](#)

TLS *Terrestrial Laser Scanning* [xxii](#), [13](#), [126](#)

YOLO *You Only Look Once* [174](#)

Introduction générale

« Le danger qui menace les chercheurs aujourd'hui serait de conclure qu'il n'y a plus rien à découvrir. »

Pierre Joliot-Curie — Biologiste, Scientifique (1932)

1.1	Contexte et problématique	4
1.2	Méthodes d'inspection automatique par vision 2D/3D	8
1.2.1	Inspection à partir d'images 2D	8
1.2.1.1	Inspection à partir d'images 2D sans exploiter un modèle CAO	9
1.2.1.2	Inspection à partir d'images 2D en exploitant un modèle CAO	10
1.2.2	Inspection à partir de nuages de points 3D	12
1.3	Méthodes de segmentation 3D de formes cylindriques dans un nuage de points	13
1.3.1	Segmentation de formes cylindriques rigides	14
1.3.2	Segmentation de formes cylindriques non-rigides	14
1.4	Conclusion sur les méthodes existantes	15
1.5	Solutions proposées	16
1.5.1	Inspection à partir d'images 2D	16
1.5.1.1	Inspection via une tablette portée par un opérateur	16
1.5.1.2	Inspection via un robot équipé de capteurs	18
1.5.2	Inspection à partir de nuages de points 3D	22
1.6	Localisation de la caméra	23
1.6.1	Les méthodes basées marqueurs	23
1.6.2	Les méthodes basées modèle reconstruit	25
1.6.3	Les méthodes basées modèle CAO	26
1.6.4	Synthèse des techniques permettant de localiser la caméra dans une scène complexe	27

1.6.5	La méthode disponible dans le cadre de cette thèse	27
1.7	Structuration du manuscrit	28

Dans ce chapitre, nous allons introduire le contexte de la thèse ainsi que la problématique associée à l'inspection automatique d'assemblages aéronautiques complexes avec en arrière-plan la question du goulot d'étranglement induit par les méthodes d'inspection actuelles, qui limitent l'inspection rapide de grandes quantités de données. Ensuite nous présentons une étude bibliographique des domaines qui gravitent autour de notre sujet de thèse. Plus précisément, les domaines concernés sont l'inspection par analyse d'images 2D, l'inspection par analyse de nuages de points 3D, la segmentation 3D de formes cylindriques dans un nuage de points 3D et la localisation du capteur dans un environnement complexe. Nous présentons également les solutions proposées et les moyens mis en œuvre pour contribuer à l'automatisation des méthodes d'inspection.

1.1 Contexte et problématique

Cette thèse s'inscrit dans le cadre général de l'Industrie 4.0 et du développement d'outils numériques au service de ce qui est communément désigné par Usine du Futur.

Les industriels ont pris conscience de l'intérêt de déployer les outils numériques dans les usines pour trouver de nouveaux leviers de productivité afin de gagner en agilité et flexibilité. Cette transformation industrielle se traduit notamment par :

- une exploitation des données numériques sur le terrain ;
- un accompagnement des tâches humaines complexes sur le terrain, que ce soit dans la phase d'assemblage d'un produit, ou dans la phase d'inspection, avec un guidage des actions ;
- un déploiement de systèmes robotiques pour plus d'agilité dans les actions à réaliser et pour réduire les efforts physiques.

Si l'on prend l'exemple du secteur aéronautique (qui a principalement servi de support aux applications présentées dans cette thèse), on constate que les besoins en inspection et contrôle dans le domaine de la fabrication de pièces et structures aéronautiques sont sans cesse grandissants.

Il suffit de considérer les éléments suivants . . .

- le trafic aérien a été multiplié par 6 en 35 ans,
- plus de 4 milliards de personnes ont pris l'avion en 2018,
- la barre des 1000 avions commerciaux de type Airbus et Boeing produits par an avait été atteinte en 2011. La barre des 2000 avions produits par an devrait être franchie d'ici 2021. D'ici à 2035, les constructeurs devront livrer au total plus de 30000 nouveaux avions.

... pour comprendre qu'il va être nécessaire de fabriquer de plus en plus d'avions, de plus en plus vite.

Les cadences de fabrication qui augmentent induisent un besoin d'outils d'inspection et de contrôle automatique des pièces pour réduire le temps de contrôle tout en maîtrisant la qualité. De nombreuses phases d'assemblage sont réalisées manuellement, ce qui augmente les risques de défaillance par rapport à un assemblage automatisé. Il est donc important de mettre en place une vérification de la totalité des assemblages à chaque étape de fabrication (de manière à pouvoir détecter une anomalie le plus en amont possible). Pour limiter les risques de défauts, la plupart des industriels ont recours à des contrôleurs humains qui assurent la vérification de la conformité des assemblages. Ces efforts induisent des coûts importants pour assurer une qualité de production élevée, sans pour autant éliminer complètement les risques d'erreurs.

Dans ce contexte, le contrôle automatique par vision artificielle est un moyen de soulager les opérateurs de tâches de contrôle répétitives et d'assurer un contrôle plus rapide et de meilleure qualité, en fournissant également des outils de traçabilité des opérations de contrôle.

Fournir de tels outils de contrôle est un des objectifs de la Sté DIOTA, spécialisée dans le développement d'outils numériques pour l'Industrie 4.0, avec qui cette thèse a été menée en partenariat. L'objectif de DIOTA est de proposer des solutions innovantes qui exploitent les données du [PLM](#) (gestion du cycle de vie des produits), la technique de réalité augmentée et des techniques d'inspection sans contact pour répondre aux besoins de l'Industrie 4.0. DIOTA développe deux types d'outils numériques :

1. des outils de type « Digital-Assisted Operator », qui permettent d'assister les opérateurs dans leurs tâches d'assemblage ou de contrôle de pièces,
2. des outils de type « Digital-based Robotics », qui permettent de réaliser des opérations de contrôle robotisées.

Cette thèse a été menée dans le cadre du laboratoire de recherche commun « Inspection 4.0 » sur l'inspection et le contrôle de pièces et structures mécaniques par vision 2D/3D, mis en place depuis quelques années par l'Institut Clément Ader ([ICA](#)) UMR CNRS 5312 et la Sté DIOTA.

La problématique d'inspection et de contrôle de pièces industrielles est souvent découpée en deux catégories :

1. le contrôle de conformité (par exemple la détection de défauts d'assemblage), qui consiste à vérifier qu'une pièce ou un assemblage mécanique sont conformes à leur modèle [CAO](#).
2. le contrôle de dommages (ou de surfaces), qui consiste à détecter des dommages pouvant affecter la résistance ou l'aspect esthétique d'une pièce. Par exemple, un enfoncement, une fissure ou une rayure sur un fuselage d'avion [[Jov+17](#)].

Dans le cadre de cette thèse, nous nous sommes intéressés exclusivement à la problématique du contrôle de conformité.

L'objectif de la thèse est donc de développer des outils permettant l'inspection automatique d'assemblages mécaniques aéronautiques complexes tels que celui qui est visible¹ sur la figure 1.1. L'objectif de l'inspection est de vérifier que l'assemblage a été correctement assemblé, i.e. que tous les éléments constituant l'assemblage sont présents, dans la bonne position et à la bonne place.

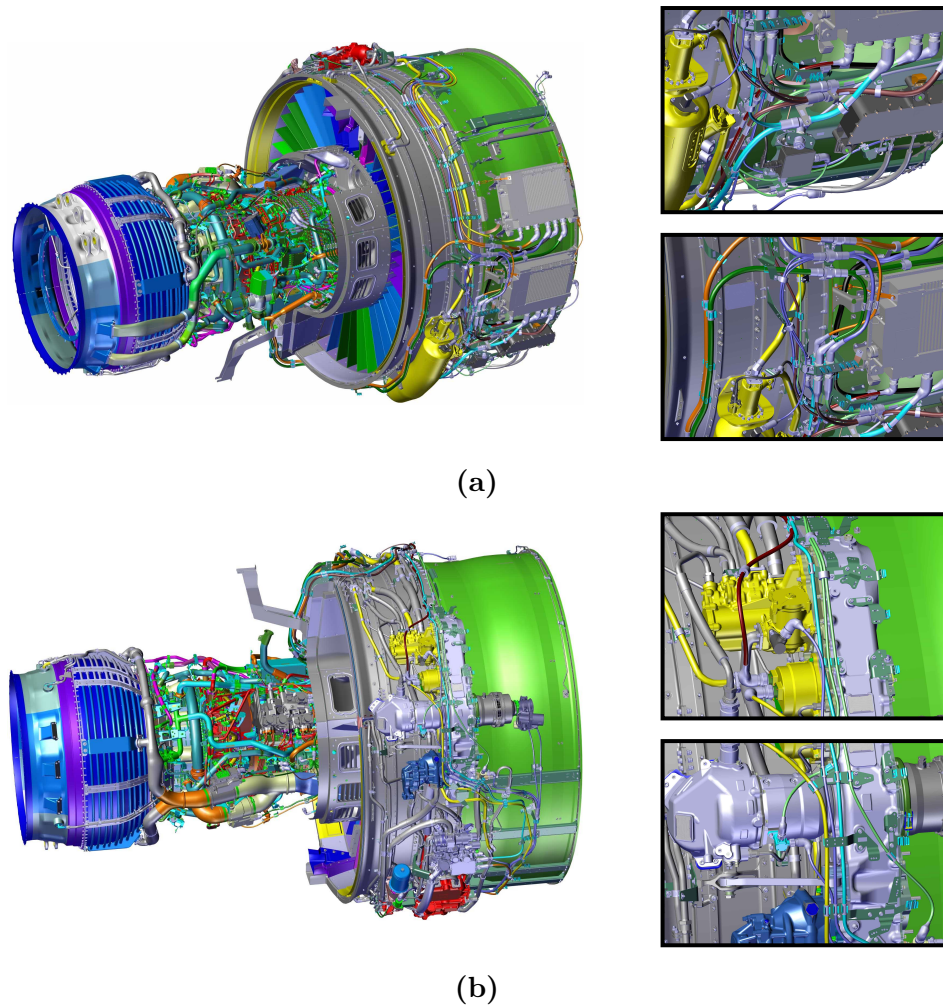


Figure 1.1 – Un exemple d'assemblage mécanique aéronautiques complexe : (gauche) un moteur d'avion **LEAP** (Leading Edge Aviation Propulsion) de la famille CFM56, qui se compose de plusieurs dizaines de milliers d'éléments à contrôler, (droite) images zoomées

Comme on peut le voir sur la figure 1.1, les pièces peuvent être complexes et constituées de nombreux éléments qui peuvent être agencés différemment au cours de l'assemblage, qui est souvent manuel. Ceci complexifie énormément la tâche d'inspection.

1. D'autres exemples d'assemblages seront fournis dans les chapitres 3 et 4.

Pour nous, inspecter une pièce consiste à vérifier que le résultat de la fabrication est conforme à la conception originale. Autrement dit, il s'agit de vérifier que le « tel que fabriqué » est conforme au « tel que conçu ». Dans cette thèse, nous considérerons que le modèle CAO de la pièce inspectée est disponible et nous utiliserons ce modèle CAO comme une référence de l'état dans lequel doit être la pièce inspectée. L'annexe [A](#) donne quelques détails sur la façon dont les informations nécessaires à l'inspection sont extraites de la maquette numérique.

À travers cette thèse, nous avons essayé de répondre au cahier des charges qui nous a été posé et aux contraintes qui nous ont été fixées :

- inspecter des pièces complexes dans un environnement industriel de type atelier d'assemblage
- proposer des méthodes rapides et robustes
- exploiter au mieux les moyens d'inspection développés par DIOTA : une tablette équipée d'une caméra et un robot équipé d'un capteur constitué de 2 caméras et d'un scanner 3D
- privilégier les méthodes basées sur l'analyse d'images 2D, pour des raisons de rapidité
- ne pas hésiter à exploiter le scanner 3D pour les types d'inspection qui ne peuvent pas être faits par analyse d'images 2D
- exploiter au mieux la maquette numérique (modèle CAO 3D) de la pièce inspectée
- exploiter au mieux le module de localisation temps-réel développé par DIOTA
- proposer des méthodes robustes même si les données sont parfois incomplètes ou bruitées (cas des nuages de points 3D incomplets en raison d'une acquisition de type one-shot)
- veiller à ce que les algorithmes d'inspection ne renvoient aucun faux négatif (il ne faut rater aucun défaut), tout en minimisant le taux de faux positifs (fausses alarmes)

Pour répondre à ce cahier des charges et à ces contraintes, deux scénarios d'inspection et plusieurs méthodes d'inspection ont été développées.

Les deux scénarios d'inspection exploitent les moyens d'inspection développés par DIOTA :

1. un scénario basé sur une tablette², équipé d'une caméra, portée par un opérateur pour un contrôle interactif,
2. un scénario basé sur un robot équipé de capteurs pour un contrôle totalement automatique.

2. ce que nous désignons par le terme *tablette* est souvent un ordinateur puissant, avec un écran de type tablette, équipé d'une caméra industrielle

Ces deux scénarios ont permis de contribuer au développement des deux types d'outils de contrôle proposés par DIOTA (« Digital-Assisted Operator » et « Digital-based Robotics »). À noter que nous n'avons pas participé au développement de ces moyens matériels. Nous avons développé des algorithmes d'inspection en traitant les données fournies par les capteurs portés par la tablette ou le robot.

Par ailleurs, deux types de méthodes d'inspection ont été développés :

1. une méthode basée sur l'analyse d'images 2D fournies par une caméra,
2. une méthode basée sur l'analyse de nuages de points 3D fournis par un scanner 3D.

Dans la section suivante, nous allons commencer par faire un bref état de l'art des méthodes existantes pour l'inspection automatique par vision 2D/3D.

1.2 Méthodes d'inspection automatique par vision 2D/3D

Les approches pour résoudre un problème d'inspection sont très différentes suivant le type et le nombre de capteurs utilisés (une caméra 2D, plusieurs caméras 2D, un scanner 3D, etc.) et le type d'objet à inspecter (forme simple ou forme complexe). Au niveau du type de caméra, certains travaux exploitent des images thermiques pour détecter des défauts [Bel+16 ; Her+19] mais nous avons décidé de restreindre notre état de l'art aux méthodes d'inspection exploitant des images dans le spectre visible.

Cette section a pour objectif de faire un tour d'horizon des méthodes existantes d'inspection automatique par vision 2D/3D. Nous évoquerons dans un premier temps les méthodes d'inspection automatique basées sur l'analyse d'images 2D. Ensuite, nous aborderons les méthodes d'inspection qui s'appuient sur l'analyse de nuages de points 3D.

Il est à noter que les *proceedings* de la conférence internationale « Quality Control by Artificial Vision (QCAV) », qui a lieu tous les deux ans depuis 1993, sont une source intéressante pour l'identification de travaux dans le domaine de l'inspection par vision 2D/3D.

1.2.1 Inspection à partir d'images 2D

L'inspection à partir de l'analyse d'images 2D a fait l'objet de nombreux travaux. Généralement l'utilisation des techniques d'analyse d'images 2D pour la détection de défauts est privilégiée pour diminuer le temps de calcul et accélérer l'inspection.

Dans l'état de l'art, on peut distinguer deux familles d'approches essayant de résoudre ce problème :

1. la première approche consiste à comparer les caractéristiques extraites de l'image 2D (qui décrivent l'objet réel) à une description ou à une liste de règles qui décrivent l'objet idéal, sans utiliser un modèle CAO.
2. la deuxième approche consiste à exploiter un modèle CAO qui décrit l'état dans lequel doit se trouver l'objet inspecté

1.2.1.1 Inspection à partir d'images 2D sans exploiter un modèle CAO

TIEN SY et al. [Tie+09] ont présenté un algorithme de détection des défauts sur la chaussée. Dans ce travail, les auteurs cherchent à détecter des fissures sur la chaussée par analyse d'images 2D. L'extraction et la classification de défauts dans l'image 2D sont réalisées par une transformation de Hough et par le calcul de l'orientation locale des pixels.

JOVANČEVIĆ et al. [Jov+15] ont développé des algorithmes d'analyses d'images 2D sans utilisation du modèle CAO pour établir un diagnostic sur l'état d'un avion (vérifier que le moteur n'est pas endommagé ou obstrué, vérifier qu'une trappe n'est pas restée ouverte, vérifier la qualité des pneus et vérifier que les cales ont été enlevées ou vérifier que les sondes ne sont pas endommagées). À noter néanmoins que le modèle CAO a été exploité pour guider le robot *Air-Cobot* vers la zone à inspecter.

VILLEMOT et al. [Lei+17; Vil+16] ont proposé une méthode de détection d'amers visuels sur un avion. Dans leur travail, ils ont utilisé le robot mobile autonome *Air-Cobot*. L'algorithme d'inspection présenté nécessite en entrée des sous-images contenant uniquement l'élément à inspecter et une image de la scène. Ensuite, pour déterminer l'état de l'élément à inspecter, ils ont utilisé les descripteurs SURF et l'algorithme d'appariement *template matching* pour faire l'appariement entre les sous-images de référence avec l'image de la scène.

Récemment, TOUT et al. [Tou+19] ont présenté un système automatisé pour détecter des défauts à la surface d'un vilebrequin. En raison de la surface très texturée des vilebrequins et de la variabilité des formes et des types de défauts, ils ont montré que l'utilisation des techniques d'apprentissage profond, plus précisément les réseaux neuronaux convolutifs (CNN), est une solution plus efficace que les méthodes traditionnelles.

MIRANDA et al. [Mir+19b] ont présenté un algorithme de détection des défauts ou anomalies sur le fuselage externe d'un avion (e.g vis, signalétique, fuite ou impact). L'acquisition des images est effectuée par un drone qui tourne autour de l'avion. Une fois le vol terminé, l'analyse des images 2D est effectuée sur une tablette portée par un opérateur. Ils ont utilisé un réseau de neurones entraîné de bout en bout (*end-to-end learning*) pour détecter et identifier les objets de la scène. L'opérateur en charge de l'inspection doit valider ou refuser manuellement chaque zone contenant des défauts détectés et il peut ajouter manuellement des défauts non détectés. Dans leur travail, ils ont abordé le problème du manque de données pour l'apprentissage en faisant un apprentissage sur des données synthétiques réalistes.

Nous nous intéresserons par la suite aux méthodes qui exploitent le modèle CAO. En effet, lorsque celui-ci est disponible, il y a tout intérêt à utiliser le modèle CAO car celui-ci fournit une description exacte et précise de la géométrie de l'objet inspecté. De plus, l'exploitation de ce modèle est le seul moyen efficace et précis pour une inspection automatisée [Pri+99]. Dans le cadre de cette thèse, nous nous placerons

donc dans le cadre où la maquette numérique de la pièce/structure à contrôler est disponible. Cette maquette servira de référence et décrira la pièce/structure ou l'assemblage tel qu'il doit être.

1.2.1.2 Inspection à partir d'images 2D en exploitant un modèle CAO

Nous présentons dans cette section quelques-uns des travaux sur l'inspection à partir d'images 2D et d'un modèle CAO de l'objet à inspecter. Ces méthodes consistent généralement à comparer les primitives 3D extraites du modèle CAO avec les mêmes primitives 2D extraites de l'image. Ces primitives peuvent être des points singuliers [Des+15], des contours 2D ou 3D [Loe+18], des squelettes [Mer04], etc.

MERAD [Mer04] a développé un système d'inspection automatique d'objets comportant des surfaces complexes à l'aide d'une caméra 2D fixe. Ce système d'inspection est dédié à une machine à usinage rapide. La méthode d'inspection proposée se base sur les informations géométriques et topologiques sous la forme d'un graphe de squelette. Elle consiste à effectuer une mise en correspondance entre le squelette 3D extrait hors-ligne du modèle CAO et le squelette 2D extrait en ligne de l'objet à inspecter. Ce processus de squelettisation permet de reconnaître l'objet et de le localiser. Une fois que l'objet est localisé, la projection du squelette 3D sur l'image 2D permet d'analyser localement l'objet à inspecter et donc de statuer sur sa conformité.

Cette méthode repose principalement sur l'exploitation de squelettes 2D/3D. La squelettisation est une méthode utilisée en analyse de formes. Elle permet de représenter un objet avec un minimum d'information. Ces informations représentent des courbes centrées sur la forme d'origine. Cependant, comme expliqué par LEBORGNE [Leb16], la squelettisation apporte parfois une simplification de l'information trop importante et la méthode s'avère très sensible au bruit sur les bords de la forme.

BERNDT [Ber13] a présenté une nouvelle technique de détection d'erreurs d'assemblage mécanique dans les chaînes de production de petites quantités. Cette technique a été développée par les chercheurs de l'institut Fraunhofer IFF. Ils ont développé un logiciel qui crée une image synthétique de l'objet avant assemblage, à partir de son modèle CAO et depuis une position particulière de la caméra. Une fois que le technicien a terminé d'assembler les éléments, une caméra fixée à une position connue par rapport à l'assemblage prend une image. Cette image possède exactement la même perspective que l'image synthétique de référence (créée à partir du modèle CAO). Enfin, une comparaison entre l'image synthétique de référence et l'image prise par la caméra permet de détecter des erreurs.

Cette méthode exige une connaissance a priori de la position de la caméra par rapport à l'élément à inspecter. Dans ce cas, l'intervention du technicien est souvent nécessaire pour manipuler la position de la caméra avec précision pour chaque élément à inspecter, ce qui n'est pas toujours réalisable.

KARABAGLI [Kar13] a présenté une technique de vérification automatique de montages d'usinage, basée sur la vision monoculaire (une caméra). La technique présentée permet de reconnaître automatiquement les éléments du montage (brut à

usiner, pions de positionnement, tiges de fixation, etc.) et d'assurer que les éléments de bridage constituant le montage d'usinage sont présents, au bon endroit et ont la bonne forme et dimension. Le système est basé sur l'exploitation d'une image de contours 2D. Cette image de contours permet ensuite d'extraire des primitives simples (e.g. centre des cercles et intersection des droites) permettant de faire une mise en correspondance avec une image théorique (créée à partir du modèle CAO). La décision sur la conformité du montage d'usinage est prise à partir d'un score d'appariement. Ce score est donné par le rapport entre la mesure de caractéristiques topologiques (squelette) et dimensionnelles (contour et surface) calculées sur l'objet réel et la mesure de son correspondant dans l'image binaire de référence.

VIANA DO ESPIRITO SANTO [Via16] a présenté un algorithme d'inspection automatique d'assemblages mécaniques aéronautiques basé sur l'utilisation d'une caméra PTZ (Pan-Tilt-Zoom) et un dispositif d'éclairage unidirectionnel qui se déplace d'une manière synchrone avec la caméra PTZ. La méthode présentée permet de vérifier la présence/absence d'un élément à inspecter dans un assemblage mécanique, la conformité de son positionnement et son exactitude géométrique par rapport à un élément idéal décrit dans un modèle CAO. Cette méthode consiste à comparer le contour 2D issue du modèle CAO avec le contour 2D de l'image réelle. Les contours sont appariés, en exploitant des concepts de la théorie de graphes, notamment l'utilisation d'un graphe biparti pour s'assurer du respect de la contrainte d'unicité dans le processus d'appariement. Dans sa thèse, JOVANČEVIĆ [Jov16] a également exploité le modèle CAO pour inspecter certains éléments d'un avion (par exemple les sondes Pitot), en comparant les contours extraits dans une image avec les contours extraits dans le modèle CAO projetés dans l'image.

Ces méthodes d'inspection ont l'avantage d'être à la fois peu coûteuses en terme de calculs et relativement simples à mettre en œuvre. Elles sont principalement basées sur l'exploitation des primitives simples (e.g. lignes, segments ou cercles). Cependant, ce type de primitive est peu discriminant et très spécifique. De plus, ces méthodes peuvent être sensibles aux changements d'illumination et elles manquent souvent de robustesse, surtout lorsqu'il s'agit de les mettre en œuvre dans un environnement industriel.

Récemment, MIRANDA et al. [Mir+19a] ont proposé une méthode pour détecter et inspecter les vis extérieures d'un avion à partir d'images acquises par un drone. Ils ont utilisé un réseau neuronal convolutionnel CNN (*Convolutional Neural Network* en anglais) pour détecter les zones d'intérêt et pour extraire les vis des images. Après avoir détecté les zones d'intérêt, ils ont utilisé une approche de type GAN (*Generative Adversarial Network*) proposée par MIRZA et al. [Mir+14] qui permet de générer des motifs de vis à partir du modèle 3D CAO et des images observées. L'appariement entre les vis du modèle attendues et les vis détectées se fait en exploitant un graphe biparti. Lorsque l'appariement échoue, ils concluent qu'une vis est manquante ou endommagée.

1.2.2 Inspection à partir de nuages de points 3D

Ces méthodes d'inspection se basent sur une comparaison directe entre le modèle CAO en 3D et les données 3D de l'objet réel acquises par un capteur 3D. Dans ce cas, on dispose de deux formes :

- le modèle CAO 3D qui sert de référence, que nous appellerons M . On dispose alors d'un modèle complet et exact décrit dans le système de coordonnées de modélisation,
- le nuage de points 3D fourni par le capteur 3D, que nous appellerons D . Ce nuage de points est décrit dans le système de coordonnées de mesure.

Ces deux formes étant décrites dans des systèmes de coordonnées différents, afin de comparer ces deux formes, il est nécessaire de les exprimer dans un système de coordonnées commun et de les recalculer. Cette phase de recalage entre ces deux formes consiste à trouver la transformation rigide (rotation et translation) à appliquer sur D afin de minimiser un critère de distance entre D et M .

Un article intéressant [Mor+14] fournit une comparaison très détaillée des différentes méthodes de l'état de l'art pour l'alignement de données acquises avec un capteur RGB-D.

Les deux approches le plus fréquemment rencontrées dans la littérature pour l'alignement du modèle CAO avec les nuages de points sont :

- recalage par l'algorithme du point le plus proche itéré (ICP - Iterative Closest Point) proposé par BESL et al. [Bes+92]. Cet algorithme consiste à calculer de façon itérative la transformation alignant le mieux deux ensembles de points 3D. Il s'agit de minimiser itérativement la distance entre la source (nuage de points d'entrée) et la cible (modèle CAO). Au bout de chaque itération, l'algorithme construit une liste de points appariés en établissant une correspondance point à point via le calcul d'une distance euclidienne. Puis, il estime la transformation de repère entre les points appariés qui minimise une mesure d'erreur quadratique moyenne définie entre eux. La convergence est atteinte lorsque l'erreur résiduelle de distance entre les points appariés est inférieure à un certain seuil ou en fixant un nombre d'itérations nécessaire,
- recalage par surfaces et primitives géométriques : ce recalage consiste à mettre en correspondance des primitives géométriques (cylindre, sphère, courbe, plan, etc.) extraites du modèle CAO et du nuage de points, puis à rechercher les paramètres de transformation permettant de minimiser l'erreur entre l'ensemble de ces formes géométriques [Ben15].

PRIETO et al. [Pri+02] ont présenté l'inspection automatisée d'objets manufacturés. Ils ont comparé le modèle de référence de la base de données CAO avec leur image 3D obtenue avec un capteur télémétrique laser. Le recalage du nuage de points 3D non ordonnés avec le modèle CAO au format NURBS a été obtenu par l'algorithme ICP. Ils ont présenté également une méthode de segmentation supervisée qui permet de décomposer le nuage de points en sous-ensembles. Enfin, la comparaison est effectuée

par un calcul de distance entre chaque point 3D de chaque sous-ensemble et son point le plus proche du modèle CAO.

HARVENT et al. [Har+08] ont présenté l'inspection de pièces aéronautiques à partir d'un système multi-caméras. Les nuages de points 3D ont été obtenus par corrélation multi-images. L'alignement du modèle CAO 3D avec les nuages de points, ainsi que la détection de défauts sur les nuages ont été obtenus via le logiciel GeoMagic qui permet très facilement d'aligner les deux modèles « nuage de points 3D/modèle CAO » et fournir les informations globales (écart-type, distance moyenne) de comparaison.

HONG-SEOK et al. [Hon+14] ont présenté l'inspection de pièces usinées à partir de nuages de points 3D fournis par un scanner laser. L'alignement du modèle CAO avec les nuages de points a été obtenu par un recalage de type ICP. La détection de défauts a été effectuée par comparaison entre les caractéristiques locales (e.g. normales à la surface, coins) extraites sur le modèle CAO et sur le nuage de points.

TAGUCHI et al. [Tag+18] ont présenté un système pour l'inspection des installations électriques par analyse de nuages de points 3D. Les nuages de points 3D sont acquis avec un capteur RGB-D. Ils ont utilisé des techniques d'apprentissage automatique pour localiser les fusibles dans l'installation électrique. Ensuite, ils ont utilisé l'informations de couleur (RGB) pour identifier et vérifier l'état du fusible dans chaque zone localisée.

NGUYEN et al. [Ngu+18] ont présenté un système pour l'inspection des lignes de tuyauterie dans un environnement industriel. Le nuage de points 3D acquis avec Scanner Laser (ou Lidar) à balayage 3D Terrestre (*TLS*, *Terrestrial Laser Scanning* en anglais) est comparé avec le modèle CAO 3D. L'alignement du modèle CAO avec les nuages de points a été obtenu par un recalage de type ICP. Ensuite, ils ont utilisé la méthode de segmentation par croissance de région qui se base principalement sur l'information de normale. Cette méthode est proposée par la bibliothèque *OpenSource PCL* (*Point Cloud Library*). Ensuite, ils ont utilisé l'algorithme *RANSAC* pour estimer les paramètres du cylindre (direction et rayon) dans chaque région segmentée. Enfin, pour statuer sur la conformité des composants de tuyauterie, ils ont combiné l'analyse d'écart entre le modèle CAO et le nuage de points avec une comparaison entre les paramètres estimés des cylindres et les paramètres définis par la CAO.

1.3 Méthodes de segmentation 3D de formes cylindriques dans un nuage de points

Nous verrons dans les chapitres 4 et 5 que nous avons dû traiter la problématique de la segmentation de câbles cylindriques dans un nuage de points 3D.

Les méthodes identifiées dans la littérature pour segmenter une forme cylindrique dans un nuage de points 3D peuvent être classées en deux catégories, selon le type d'application visé :

-
- Segmentation de formes cylindriques rigides (e.g. ligne de tuyauterie) (section 1.3.1).
 - Segmentation de formes cylindriques non-rigides ou flexibles (e.g. branches principales des arbres présents sur des données forestières) (section 1.3.2).

1.3.1 Segmentation de formes cylindriques rigides

Généralement, ces méthodes cherchent à ajuster un modèle mathématique de cylindre sur un nuage de points en utilisant la méthode de RANSAC [Jin+19; Sch+07; Su+19] ou des méthodes basées sur la transformation de Hough [Rab+05].

CHAPERON et al. [Cha+01; Cha02] ont proposé un algorithme pour la segmentation des lignes de tuyauterie de diamètre constant dans un environnement industriel complexe. Cette méthode repose sur une étude locale du nuage de points permettant la modélisation-segmentation de portions de cylindres, qui sont ensuite étendues par propagation dans le nuage de points, à partir d'un point de départ spécifié manuellement par l'utilisateur à travers une IHM.

LALONDE et al. [Lal+06] ont présenté un algorithme basé sur l'Analyse en Composantes Principales (ACP) pour extraire un cylindre dans un nuage de points et estimer son rayon. La méthode ACP permet d'estimer la position et la direction du cylindre. Ensuite, les points 3D du cylindre sont projetés sur le plan défini par les deux premières composantes du modèle ACP. Pour estimer le rayon du cylindre, un cercle est ajusté sur les points 2D projetés.

De la même manière, NURUNNABI et al. [Nur+17] combinent une méthode d'Analyse en Composantes Principales Robuste (ACPR), avec la méthode de régression LTS (Least Trimmed Squares) de ROUSSEEUW [Rou84]. La méthode ACPR permet d'estimer la direction du cylindre. La méthode de régression LTS permet d'ajuster un cercle 2D sur les points 3D du nuage projetés pour obtenir le rayon du cylindre.

1.3.2 Segmentation de formes cylindriques non-rigides

Les quelques méthodes que nous avons pu identifier pour segmenter des formes cylindriques non rigides sont généralement basées sur un processus itératif d'ajustement des primitives géométriques (e.g. cylindre, cercle) sur le nuage de points.

FLÓREZ VALENCIA [Fló06] a utilisé le modèle d'état de cylindre généralisé *RGC-sm* (*Right Generalized Cylinder State Model*) pour la segmentation semi-automatique d'images vasculaires tridimensionnelles. La segmentation a été utilisée pour quantifier le degré de sténose. Il a choisi une approche de segmentation par suivi de vaisseaux pour la construction d'une représentation cylindrique continue du vaisseau. Ce modèle de cylindre lui a permis une quantification précise du degré de sténose. Dans son travail, il a utilisé des données 3D denses et complètes acquises avec des techniques d'imagerie 3D, ce qui simplifie l'étape de segmentation.

HACKENBERG et al. [Hac+15; Rau+13] ont proposé une méthode (qu'ils ont appelé *SimpleTree*) pour segmenter des arbres (tronc et branches) dans un espace

forestiers à partir de l'ajustement de cylindres sur le nuage de points 3D acquis par Scanner Laser Terrestre. Le processus d'ajustement d'un premier cylindre démarre à partir de la base de l'arbre. Ce cylindre est ajusté sur le nuage de points par la méthode des moindres carrés. À partir de ce cylindre, une étape de propagation le long du tronc et le long des branches permet d'ajuster les cylindres successifs. Le modèle final de l'arbre est constitué par la collection de tous les cylindres ajustés.

Récemment, [Gél+17a ; Gél+17b] ont proposé deux méthodes pour segmenter la tige d'une plante de tournesol. La première méthode utilise l'algorithme de RANSAC. La deuxième méthode utilise un anneau, de rayon connu a priori, qui parcourt la tige depuis le pied jusqu'à son sommet. La concaténation des différents anneaux le long de la tige permet de segmenter la tige via un cylindre généralisé.

1.4 Conclusion sur les méthodes existantes

L'état de l'art que nous avons effectué ne nous a pas permis d'identifier de méthode permettant de répondre entièrement à notre problématique et au cahier des charges et contraintes qui nous ont été donnés. Néanmoins, nous avons identifié certains travaux intéressants dont nous nous sommes inspirés.

Pour l'inspection à partir d'images 2D en exploitant un modèle CAO, nous avons poursuivi les travaux initiés par DIOTA basé sur l'exploitation des edgelets extraits du modèles CAO et des contours extraits dans les images. Nous avons complété ces travaux pour rendre la méthode plus robuste dans un contexte industriel. Nos contributions seront exposées dans la suite de ce manuscrit et résumées dans le chapitre de conclusion.

Pour l'inspection à partir de nuages de points 3D, nous nous sommes inspirés des travaux de [Cha+01 ; Hac+15 ; Rau+13], pour proposer des méthodes efficaces et robustes capables de répondre à notre problématique, tout en veillant à garder un caractère générique pour pouvoir répondre à d'autres problématiques.

L'état de l'art a montré que certains travaux récents sur l'inspection ont été réalisés via des approches de type Machine Learning qui sont de plus en plus utilisées. La plupart de ces travaux concernent la problématique du contrôle de dommages (contrôle de surfaces) [Pyu+19 ; Tab+19] qui est différente de notre problématique. Pour ce qui concerne la problématique du contrôle de conformité, nous avons trouvé peu de travaux [Cas19].

Il est à noter que, dans le cadre de leur laboratoire commun de recherche, l'Institut Clément Ader et DIOTA ont commencé à explorer des méthodes d'inspection basées sur le Machine Learning, notamment à travers plusieurs stages de Master [Abu19 ; Ghi18 ; Mik19 ; Mok18]. Une nouvelle thèse intitulée « Inspection de pièces aéronautiques en exploitant la maquette numérique, des images 2D, des nuages de points 3D et des techniques de Machine Learning » a démarré en octobre 2019.

1.5 Solutions proposées

L'objectif de DIOTA est de proposer un ensemble de solutions pour répondre à une problématique générale de contrôle et de fournir une assistance aux contrôleurs à travers des moyens de contrôle portatifs et/ou automatisés. Dans cette thèse, nous avons exploité les moyens matériels de contrôle développés par DIOTA (tablette équipée d'une caméra et robot équipé de capteurs) mais nous n'avons pas participé au développement de ces moyens matériels. Nous avons développé des algorithmes d'inspection en traitant les données fournies par les capteurs portés par la tablette ou le robot.

Parmi les solutions d'inspection proposées dans cette thèse, certaines sont basées sur l'analyse des images 2D fournies par une ou plusieurs caméras, d'autres sont basées sur l'analyse des nuages de points 3D fournis par un scanner 3D.

1.5.1 Inspection à partir d'images 2D

L'utilisation des techniques d'analyse d'images 2D pour l'inspection est privilégiée pour diminuer le temps de calcul et accélérer l'inspection. Il s'agit de contrôler un maximum d'éléments par vision 2D (analyse d'images 2D) et de réserver l'utilisation du capteur 3D (analyse de nuages de points 3D) pour les éléments qui ne peuvent pas être contrôlés par vision 2D (voir chapitre 4).

Pour inspecter un élément dans un assemblage mécanique par analyse d'images 2D, nous proposons 2 scénarios : inspection via une tablette, équipée d'une caméra, portée par un opérateur, ou inspection via un robot mobile équipé de capteurs.

1.5.1.1 Inspection via une tablette portée par un opérateur

Dans un contexte « opérateur », l'utilisation d'une tablette permet un contrôle visuel interactif en proposant des capacités de visualisation et de manipulation des fiches d'instruction, en réalité augmentée, pour assister les opérations de contrôle (voir la figure 1.2).

Une tablette peut être aussi utilisée comme moyen de contrôle automatique, en développant des algorithmes d'analyse des images fournies par la caméra portée par la tablette. Cela fait l'objet de nos travaux.

Le dispositif de contrôle proposé par DIOTA (cf. figure 1.3) est équipé d'une caméra haute résolution qui permet non seulement d'assurer la localisation de l'élément à contrôler (fonction « localisation temps réel ») mais également de réaliser des observations de ce qui est présent dans le champ de la caméra pour contrôler la conformité de la pièce inspectée en temps réel (fonction « inspection »).



Figure 1.2 – Inspection via une tablette, équipée d’une caméra, portée par un opérateur



Figure 1.3 – Solution de contrôle via une tablette équipée d’une caméra

Les caractéristiques techniques de la caméra d’inspection/localisation utilisée dans ce mode d’inspection sont présentées dans le tableau 1.1 et le tableau 1.2.

Table 1.1 – Caractéristiques techniques de la caméra d’inspection/localisation

Nom	UI-3070CP-C-HQ
Fréquence d’acquisition	123 Fps
Résolution ($h \times v$)	2056×1542 pixels
Surface optique	$7.093 \text{ mm} \times 5.320 \text{ mm}$
Résolution	3.17 MPix
Taille de pixel	$3.45 \mu\text{m}$

Table 1.2 – Caractéristiques techniques de l’objectif de la caméra d’inspection/-localisation

Nom	KOWA LM5NCL
Distance focale	4.5 mm
Distance minimale de l’objet (M.O.D)	364 mm
Champ de vision angulaire ($h \times v$)	$79.0^\circ \times 59.4^\circ$

Afin de fournir à l’utilisateur une assistance pour le déplacement de la tablette, nous proposons dans le chapitre 2 une méthode automatique de choix de points de vue qui permet de guider en temps réel les opérateurs dans leur tâche de prise d’images au cours d’une phase d’inspection en réalité augmentée. Dans le chapitre 3, nous proposons un algorithme d’inspection qui permet d’analyser les images 2D en temps réel afin de fournir un diagnostic sur la conformité de la pièce inspectée.

1.5.1.2 Inspection via un robot équipé de capteurs

Dans un contexte robotisé où le moyen n’est plus porté par un opérateur mais déplacé de manière automatique et commandé, nous proposons une solution robotisée qui s’appuie sur un effecteur de contrôle monté sur une plateforme robotique (cf. figure 1.4).

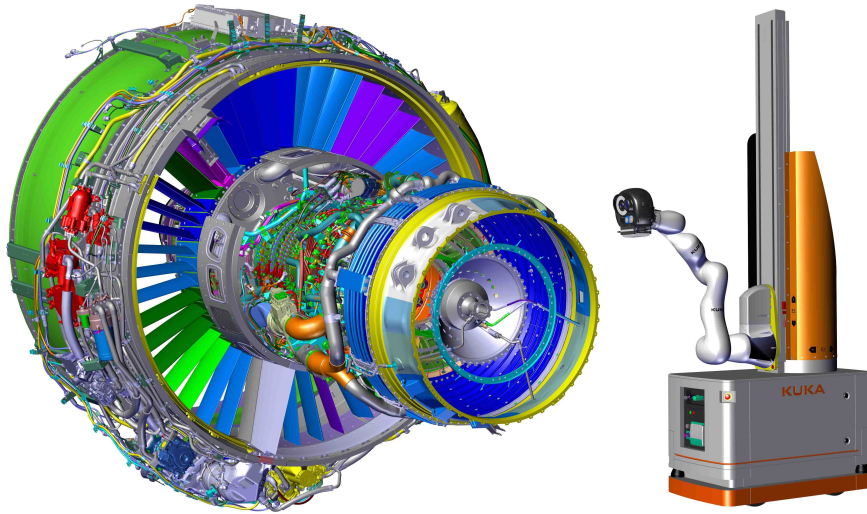


Figure 1.4 – Inspection robotisée

L’effecteur de contrôle présenté sur la figure 1.5 permet d’offrir une grande flexibilité et une large capacité de contrôle. Cet effecteur, qui est composé de plusieurs capteurs qui offrent plusieurs champs de perception, permet d’assurer plusieurs fonctions :

- Localisation : pour la localisation, une caméra à champ large permet d’assurer la localisation précise de l’effecteur par rapport à la pièce qu’il contrôle,

- Inspection : pour l'inspection, une caméra haute résolution avec un champ de vue réduit permet de capturer les détails et d'observer très finement les éléments à contrôler,
- Numérisation 3D : un capteur stéréo à lumière structurée vient compléter les capacités d'observation en permettant de numériser en 3D les zones observées. Ce capteur 3D est réservé pour les éléments que ne peuvent pas être contrôlés par vision 2D (voir la section 1.5.2),
- Pilotage : l'ensemble est piloté par un PC industriel qui assure la gestion des acquisitions de données et la localisation de l'effecteur directement en bout de bras.

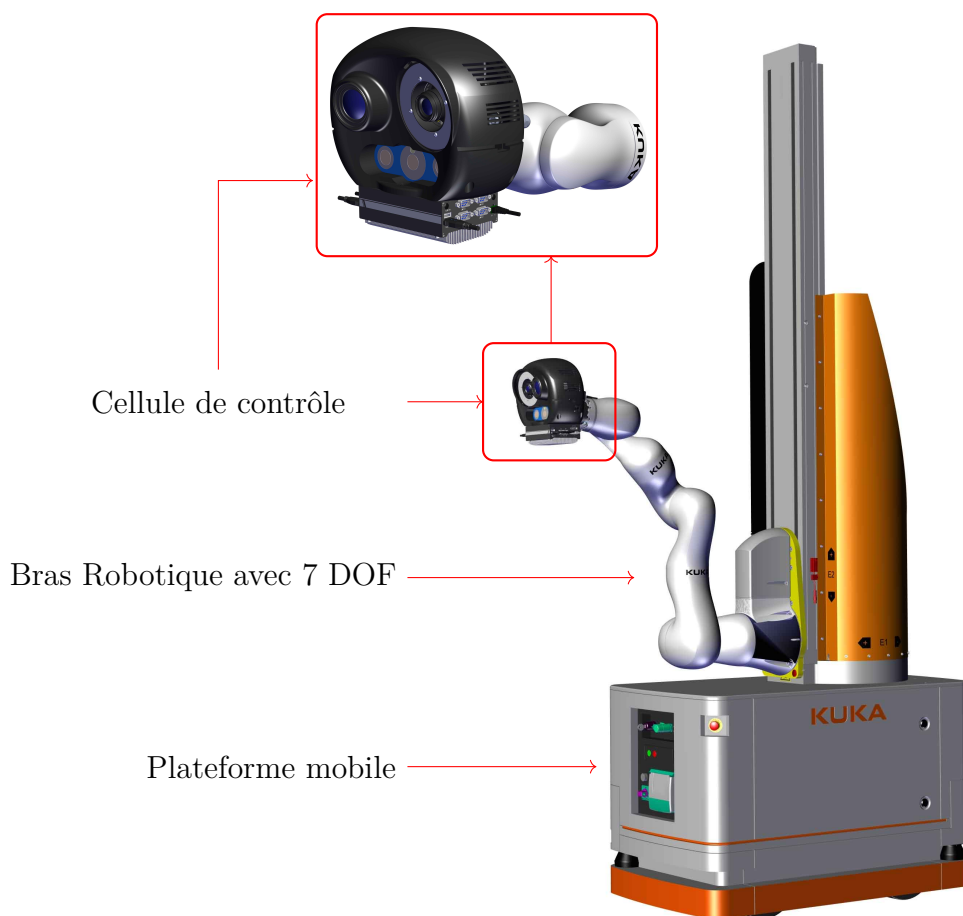


Figure 1.5 – Effecteur de contrôle autonome

Caméra d'inspection : cette caméra (cf. figure 1.6) dispose d'une résolution élevée (2592×2048 pixels) et se base sur un capteur de grande dimension ce qui permet d'obtenir une très bonne sensibilité et une résolution maximale sur la zone de travail. Le choix d'une focale 25 mm permet de couvrir une zone centrée sur les objets à contrôler et de conserver une bonne résolution pour le contrôle.

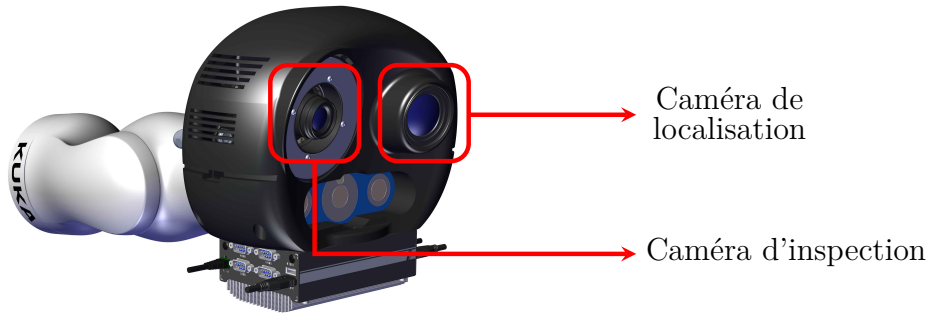


Figure 1.6 – Le capteur est équipé de 2 caméras

À la distance de travail de 600 mm, la caméra observe une zone de 311 mm \times 245 mm et chaque pixel couvre un carré de 0.12 mm \times 0.12 mm sur l'objet. La résolution spatiale que l'on obtient est donc d'environ un point tous les 0.12 mm sur les deux axes ce qui signifie dans un contexte de mesure qu'une erreur d'un pixel équivaut à une erreur d'environ 0.1 mm.

Les caractéristiques techniques de la caméra d'inspection et de son optique sont présentées dans le tableau 1.3 et le tableau 1.4.

Table 1.3 – Caractéristiques techniques de la caméra d'inspection

Nom	UI-3180 (PYTHON 5000)
Fréquence d'acquisition	73 Fps
Résolution ($h \times v$)	2592 \times 2048 pixels
Surface optique	12.441 mm \times 9.830 mm
Résolution	5.31 MPix
Taille de pixel	4.8 μm

Table 1.4 – Caractéristiques techniques de l'objectif de la caméra d'inspection

Nom	vs-2514h1
Distance focale	25 mm (25.42)
Distance minimale de l'objet (M.O.D)	300 mm
Champ de vision angulaire ($h \times v$)	21.5° \times 28.5°

Caméra de localisation : cette caméra à champ large (cf. figure 1.6) doit permettre au robot de localiser de manière précise la position réelle de l'effecteur par rapport à la pièce qu'il contrôle.

Les caractéristiques techniques de la caméra de localisation et son optique sont présentées dans le tableau 1.5 et le tableau 1.6.

Table 1.5 – Caractéristiques techniques de la caméra de localisation

Nom	IDS 3070
Fréquence d'acquisition	123 Fps
Résolution ($h \times v$)	2056×1542 pixels
Surface optique	$7.093 \text{ mm} \times 5.0320 \text{ mm}$
Résolution	3.17 MPix
Taille de pixel	$3.45 \mu\text{m}$

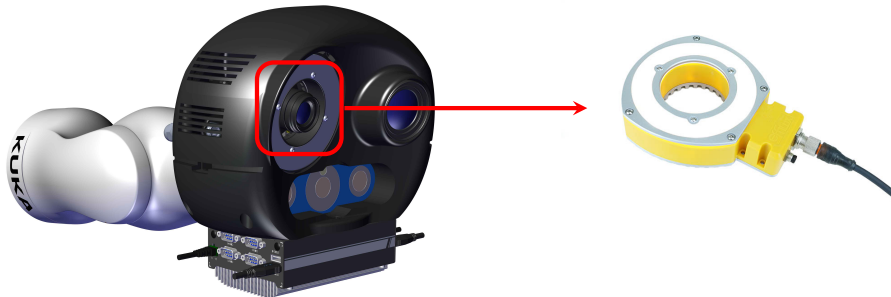
Table 1.6 – Caractéristiques techniques de l'objectif de la caméra de localisation

Nom	Kowa 8 mm
Distance focale	8 mm
Distance minimale de l'objet (M.O.D)	100 mm
Champ de vision angulaire ($h \times v$)	$79.7^\circ \times 63.0^\circ$

Eclairage : l'éclairage est un élément important de tout système d'inspection automatisé.

Lorsqu'on prend en compte les différentes fonctions du capteur (une caméra pour l'inspection et une caméra pour la localisation), il n'est pas nécessaire de positionner un éclairage global pour le capteur mais le plus important est de se concentrer sur la caméra de contrôle 2D qui a besoin dans certains contextes de pouvoir disposer de lumière additionnelle et de la contrôler. Par conséquent, nous avons ajouté une solution d'éclairage autour de la caméra d'inspection (voir figure 1.7). Cette source d'éclairage n'est activée qu'au moment de l'acquisition de l'image, lorsque le robot est en mode d'acquisition (le robot est en position fixe et ne bouge pas).

Pour la caméra de localisation, la vue de l'assemblage en champ large permet de capturer à la fois la pièce mais également son environnement. Il n'est pas nécessaire d'apporter de la lumière pour que les algorithmes fonctionnent et une lumière qui n'est pas statique dans la scène pourrait créer des erreurs et/ou perturber le tracking. Nous avons donc décidé de ne pas ajouter d'éclairage pour cette fonction mais, si nécessaire, de nous appuyer sur l'éclairage naturel ambiant pour apporter une lumière fixe constante.

**Figure 1.7** – Système d'éclairage autour de la caméra d'inspection

1.5.2 Inspection à partir de nuages de points 3D

L'effecteur de contrôle présenté précédemment est équipé d'un scanner 3D qui permet de capturer de l'information 3D. Cette perception 3D est nécessaire en cas de mesure nécessitant la prise en compte de l'aspect dimensionnel et notamment pour les relevés qui ont besoin d'interpréter la profondeur pour statuer sur la conformité. C'est le cas typiquement lors d'un relevé pour analyser les interférences entre un câble et les autres éléments de l'assemblage. Il s'agit de vérifier qu'un câble est à une distance minimale des éléments qui l'entourent. Dans ce cas, la mesure dans l'image ne suffit plus et la perception 3D vient lever des ambiguïtés ou identifier la position d'éléments 3D à partir d'une vue partielle.

Nous avons évalué plusieurs capteurs 3D et nous avons finalement sélectionné le capteur ENSENSO de la société IDS (voir figure 1.8).

Le scanner 3D ENSENSO (cf. figure 1.8) fonctionne selon le principe de stéréovision. Ce capteur dispose de deux caméras CMOS et d'un projecteur intégré, créant un motif aléatoire de points sur l'objet à saisir. Ce capteur contient un système actif avec encodage spatial permettant la capture « *one-shot* ».

Les caractéristiques techniques du scanner 3D sont présentées dans le tableau 1.7.

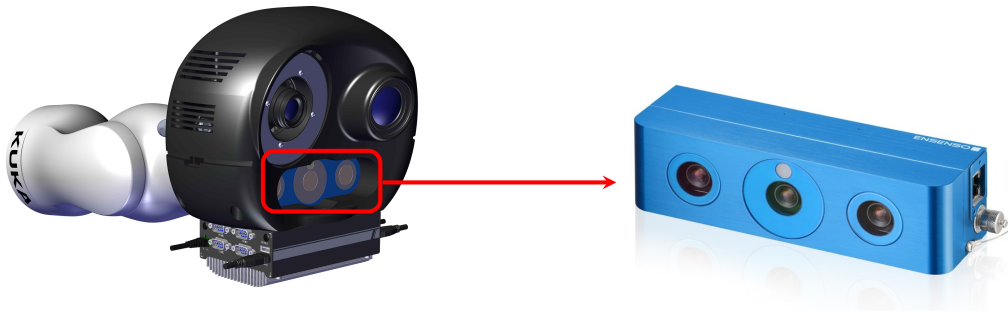


Figure 1.8 – Scanner 3D Ensenso N35

Table 1.7 – Caractéristiques techniques du scanner 3D

Nom	Scanner 3D Ensenso N35
Capteurs	2 capteurs CMOS (Global Shutter) + 1 projecteur
Résolution	1280 × 1024 (1.3 MPix)
Distance focale	6 – 16 mm
Distance de travail max	3000 mm
Fps max	10 (2× Binning : 30) et 64 disparités

Remarque : au niveau du scanner 3D, il n'est pas nécessaire d'ajouter un éclairage puisque c'est un système actif qui se base sur sa propre projection de motif lumineux pour réaliser ses traitements.

Dans les chapitres 4 et 5, nous nous intéressons à l'inspection d'éléments de type câbles électriques, qui a nécessité la mesure de distances 3D, et donc la mise en œuvre d'une méthode d'inspection basée sur l'analyse de nuages de points 3D fournis par un scanner.

1.6 Localisation de la caméra

La localisation de la caméra revient à déterminer la pose entre un objet connu et la caméra, modélisée par son modèle sténopé [Hor+95]. Cette pose correspond à la transformation rigide entre le repère de la caméra et celui de l'objet selon six degrés de liberté, trois translations et trois rotations. Dans ce cas, on suppose que la caméra est calibrée, c'est-à-dire que les paramètres internes (ou intrinsèques) à la caméra (focale, point principal, paramètres de distorsion) sont connus.

Il existe trois grandes classes de méthodes d'estimation de pose : les méthodes basées marqueurs, les méthodes basées modèle reconstruit et les méthodes basées modèle CAO. Chaque méthode possède des avantages et des inconvénients. Dans cette section, nous détaillons ces différentes méthodes et nous présentons in fine la méthode utilisée dans le cadre de nos travaux.

1.6.1 Les méthodes basées marqueurs

Les méthodes les plus anciennes de localisation se basent sur l'utilisation de marqueurs placés dans l'environnement et facilement détectables pour estimer le mouvement relatif (position et orientation dans l'espace) du capteur qui est déplacé autour la pièce à localiser. Cependant, ces méthodes présentent certaines limites qu'il est nécessaire de connaître et d'anticiper.

Ces méthodes ont été présentées la première fois en 1999 par KATO et al. [Kat+99], et ensuite par BILLINGHURST et al. [Bil+00]. Elles font aujourd'hui l'objet de plusieurs bibliothèques, les plus utilisés actuellement étant ARToolKit et Vuforia Studio. Le succès de ces bibliothèques est lié à la fiabilité et à la rapidité de détection des marqueurs dans l'image. La localisation basée marqueurs peut sembler aisée, mais elle nécessite de répondre à de nombreuses questions : la mise en œuvre [Fia05 ; Hav18 ; Kat+99], le choix de marqueurs appropriés [Dav+04 ; Zha+02] et leur détection et identification [Hav18 ; Kat+99].

Un marqueur est un objet de localisation artificiel qui peut être incorporé dans la scène. Ces marqueurs sont des panneaux plats, imprimés avec des dimensions normalisées connues dans la scène réelle. De plus, un marqueur (voir la figure 1.9a) est un élément graphique (i.e. une image) qui possède généralement les caractéristiques suivantes :

- image carrée en noir et blanc,
- sa dimension est connue dans la scène,
- il possède un motif unique.

Grâce à ces fortes contraintes, ces marqueurs peuvent être facilement repérés et ainsi être considérés comme des objets de localisation idéaux. Pour chaque image de la scène, le repérage de ces marqueurs permet de déduire facilement la pose de la caméra.

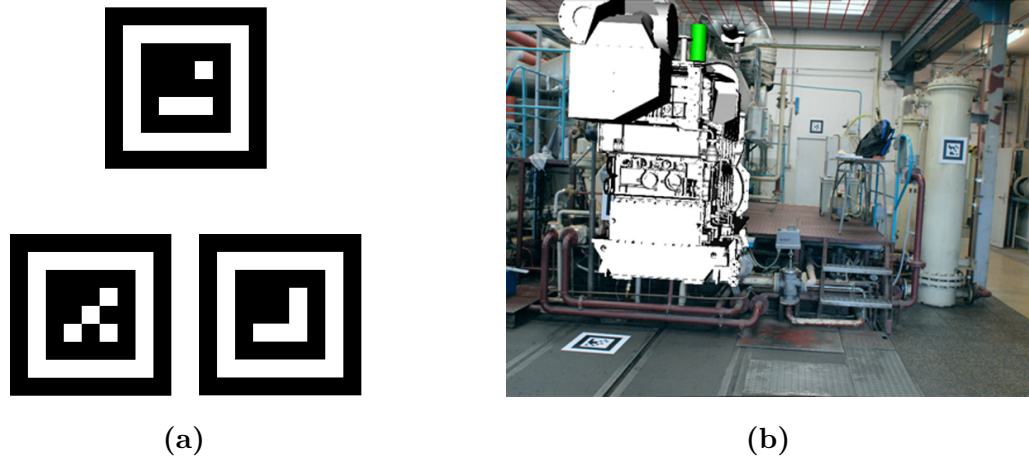


Figure 1.9 – Localisation basée marqueurs : (a) exemple de trois marqueurs avec des motifs différents, et (b) localisation basée sur l'utilisation des marqueurs placés dans l'environnement

La localisation basée marqueurs se déroule en plusieurs étapes [Hav18]. Tout d'abord, l'image d'entrée est binarisée : transformer l'image d'entrée (image en couleur ou en niveaux de gris) en une image dont les valeurs de pixels ne peuvent avoir que deux classes 1 ou 0. Ensuite, les contours du carré sont détectés grâce à une transformation de Hough [Hou+62]. Les coins du carré sont ensuite détectés à l'aide du détecteur de coin de Harris [Har+88] (voir la figure 1.10). Enfin, les positions de ces coins permettent de déterminer la pose de la caméra.

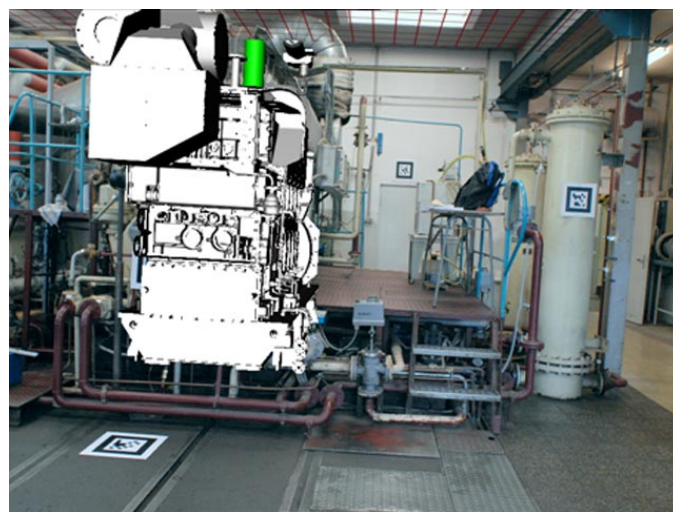


Figure 1.10 – Exemple de résultat de détection de marqueurs sur une image en niveaux de gris

Les avantages de cet algorithme sont :

- Calcul rapide. En effet, les opérations de traitement utilisées sur l'image d'entrée (binarisation, détection de contour et détection d'angle) ne nécessitent pas un temps de calcul important,
- Stabilité de la détection,
- La détection de multiples marqueurs est sans impact sur les résultats.

Cependant, cette approche nécessite de poser continuellement dans la scène réelle de marqueurs artificiels, ce qui n'est pas toujours réalisable. De plus, les marqueurs doivent rester visibles (i.e. dans le champ de la caméra) pendant la localisation. En effet, une occlusion partielle du marqueur empêche la détection [Hav18]. Ceci limite les mouvements possibles de la caméra.

Remarque : pour plus d'informations sur les techniques de localisation basée marqueurs, nous invitons le lecteur à se référer au manuscrit de thèse de HAVARD [Hav18].

1.6.2 Les méthodes basées modèle reconstruit

Les méthodes précédentes exigeaient une connaissance a priori de l'objet d'intérêt pour faire la localisation. Lorsque l'environnement est inconnu, de nouvelles techniques sans marqueurs ont été développées. Ces techniques de localisation appelées **SLAM** (*Simultaneous Localization And Mapping* en anglais) sont des méthodes qui permettent à un système de vision de se localiser tout en créant simultanément une cartographie de l'environnement qui est généralement inconnu. Généralement, ces méthodes utilisent la détection de points particuliers dans les images que l'on appelle points d'intérêts.

Les méthodes de SLAM ont été grandement améliorées depuis leur introduction par SMITH et al. [Smi+87] au début des années 1980.

MOURAGNON et al. [Mou+06] ont proposé un algorithme de SLAM monoculaire basé points d'intérêt afin d'assurer des performances temps réel de localisation. Cet algorithme a été récemment utilisé dans les travaux de thèse de LOESCH [Loe17]. Comme présenté par LOESCH [Loe17], le principe de fonctionnement de l'algorithme de SLAM est le suivant : à l'instant t la caméra observe en ligne les points d'intérêt présents dans l'environnement. Ensuite, à l'instant $t + 1$, ces points d'intérêt vont permettre de localiser la caméra par rapport aux points d'intérêt présents à l'instant t . La détection de points d'intérêt est généralement effectuée à l'aide de la méthode de HARRIS et al. [Har+88]. Ce principe de localisation a été utilisé par TREUILLET et al. [Tre+10] pour assister la navigation des personnes malvoyantes dans des environnements inconnus.

Récemment, ELLOUMI et al. [Ell+17] ont proposé une méthode de localisation pédestre, basée sur la vision monoculaire. Cette méthode se base principalement sur l'exploitation des points de fuite pour estimer l'orientation de la caméra dans les environnements inconnus. L'orientation de la caméra est estimée de façon rapide

par le suivi de 3 points de fuite orthogonaux dans une séquence vidéo. Dans leur travail, ils ont montré que l'utilisation des primitives de type lignes de fuite sont moins sensibles aux mouvements brusques et aux changements de scènes que les primitives de type points d'intérêt. Pour plus d'informations sur les techniques de localisation dans des environnements inconnus, nous invitons le lecteur à se référer au manuscrit de thèse de ELLOUMI [ELL12].

Les avantages des méthodes basées modèle reconstruit sont :

- qu'elles n'ont pas besoin d'une connaissance a priori de l'objet d'intérêt,
- elles sont robustes aux occultations partielles.

Le point faible de ces méthodes de localisation par reconstruction est qu'elles ont tendance à dériver et peuvent générer une instabilité qui rend le résultat de localisation peu robuste. Pour rendre ces méthodes plus performantes, il existe des techniques de fusion qui prennent en compte des informations complémentaires telles que des informations issues de capteurs inertiels ou des signaux GPS (*Global Positioning System* en anglais) pour renforcer la robustesse et augmenter la précision.

1.6.3 Les méthodes basées modèle CAO

Dans certains contextes où l'on dispose d'une connaissance a priori de la géométrie de la pièce visée ou de son environnement, on peut également utiliser une maquette numérique pour se localiser par rapport à la pièce que l'on souhaite inspecter. Dans ce cas, on parle de techniques de recalage basées modèle.

Le recalage basé modèle s'appuie sur un ensemble de primitives 3D provenant du modèle de l'objet à localiser. Ce recalage est effectué à partir d'un processus en deux étapes :

- Étape hors ligne : dans un premier temps, une extraction des primitives 3D du modèle 3D de l'objet d'intérêt ;
- Étape en ligne : dans un deuxième temps, les primitives 3D extraites dans l'étape hors ligne sont appariées avec leurs primitives 2D associées dans les images d'un flux vidéo. Enfin, la pose de la caméra est estimée en minimisant l'erreur de reprojection (voir la section 3.5.1.2) entre les primitives 3D extraites du modèle et leurs observations associées dans l'image 2D.

Le problème de localisation basée modèle peut sembler trivial mais il soulève cependant de nombreuses questions notamment sur le choix de primitives. Dans la littérature, des nombreuses approches de localisation se basent sur l'information de contour du modèle 3D [Loo17]. Ces méthodes s'appuient sur l'extraction de contours 2D dans l'image, suivi d'une étape de mise en correspondance entre ces contours et les contours 3D du modèle 3D [Kos+95 ; Ruf+97]. Cette approche a l'avantage d'être à la fois peu coûteuse en termes de calcul et peu sensibles aux changements d'illumination. Cependant, ces primitives étant peu discriminantes, elles sont parfois difficilement exploitables dans l'industrie.

D'autres travaux ont proposé d'utiliser l'association de points d'intérêt entre le modèle 3D et l'image 2D. Dans ces travaux, l'association de points d'intérêt se base sur différents descripteurs comme SURF [Bay+08] ou SIFT [Low04]. Bien que ces primitives soient plus discriminantes que la méthode des points de contour discutée précédemment, ces approches sont principalement basées sur l'apparence de l'objet à localiser et donc sensibles aux changements d'illumination et, par conséquent, difficilement exploitables dans l'industrie.

En conclusion, ces méthodes offrent une grande robustesse mais imposent de disposer d'une maquette précise de l'objet observé. Le principal inconvénient de cette approche est de devoir observer en permanence une partie importante de la maquette, ce qui ne donne aucune flexibilité sur le mouvement qui peut conduire à un « décrochage » dans certains cas. Typiquement, avec ces méthodes, il est difficile de s'approcher de la pièce sans risquer de la perdre lorsqu'une bonne partie est masquée.

1.6.4 Synthèse des techniques permettant de localiser la caméra dans une scène complexe

Pour aboutir à une localisation robuste et précise, la plupart des systèmes ont recouru à l'utilisation de marqueurs. Cette technique implique cependant de modifier l'environnement ce qui n'est pas toujours possible à réaliser. De plus, la présence obligatoire de marqueurs artificiels dans la scène nous semble être un frein majeur au déploiement de cette méthode en environnement industriel et opérationnel. Par conséquent, la suppression de la contrainte imposée par les marqueurs est un passage nécessaire pour une industrialisation effective de la localisation précise d'un élément dans une scène complexe.

Pour s'affranchir des contraintes liées aux marqueurs, de nouvelles techniques sans marqueurs ont été développées. Ces techniques ont démontré leur robustesse. Cependant, les contraintes associées à chacune de ces méthodes rendent leurs exploitations difficiles dans le cadre d'une application industrielle.

Remarque : pour les lecteurs souhaitant avoir plus de détails sur les algorithmes de localisation, une monographie de synthèse a été publiée par LEPETIT et al. [Lep+05]. MARCHAND et al. [Mar+09] ont aussi présenté une synthèse complète de ces différentes approches.

Nous avons présenté les avantages et les inconvénients des différentes méthodes de localisation existantes. Cependant, même si un nombre important d'algorithmes de localisation existent dans la littérature, ceux-ci ne sont pas toujours, pour des raisons diverses (e.g. temps de calcul, robustesse, précision), adaptés pour localiser un élément à inspecter dans une scène complexe.

1.6.5 La méthode disponible dans le cadre de cette thèse

Pour nos travaux, nous avons utilisé une méthode robuste de localisation développée par DIOTA qui exploite le modèle 3D de l'élément à localiser. Cette méthode

combine la localisation sans marqueurs avec un algorithme de SLAM contraint par le modèle. Avec cette méthode, nous obtenons une localisation qui profite de l'avantage des deux techniques à savoir la robustesse et la souplesse d'utilisation.

Grâce à cette technique de recalage basée SLAM contraint par le modèle, nous sommes en mesure de relier en temps réel la pièce physique avec sa représentation virtuelle (maquette numérique).

Le fait de pouvoir relier une pièce physique avec son modèle 3D va être exploité à des fins d'inspection en analysant les écarts entre le « tel que fabriqué » et le « tel que conçu ».

1.7 Structuration du manuscrit

Ce manuscrit décrit le développement d'outils numériques permettant l'inspection automatique d'assemblages mécaniques aéronautiques complexes.

En premier lieu, nous présentons dans le chapitre 2 une méthode de choix automatique des points de vue, réalisé hors ligne, pour une acquisition optimale des données qui sera réalisée en ligne dans l'étape ultérieure d'inspection.

Dans le chapitre 3, nous présentons une méthode automatique d'inspection basée uniquement sur l'analyse d'images 2D et l'exploitation du modèle CAO de l'élément à inspecter.

Dans le chapitre 4, nous présentons une méthode originale et automatique pour la segmentation 3D de câbles dans un environnement complexe, basée uniquement sur l'analyse des nuages de points 3D et l'exploitation du modèle CAO du câble.

Dans le chapitre 5, nous présentons une application de la méthode de segmentation 3D pour inspecter le système d'interconnexion de câblage électrique (*Electrical Wiring Interconnection System (EWIS)* en anglais), et, plus précisément, pour mesurer le rayon de courbure et aussi vérifier si les câbles sont à une distance minimale des éléments qui sont dans son environnement (problème d'interférences).

Enfin, nous terminons par une conclusion générale dans laquelle nous résumons nos travaux, nous listons nos principales contributions et nous présentons différentes perspectives envisageables pour la suite de ces travaux.

Sélection automatique des points de vue

« Sachant ce que nous savons à propos du monde, où devrions nous aller pour gagner le plus d'information possible ? »

Yamauchi (1997)

2.1	Introduction et problématique	30
2.2	Méthodes de choix automatique de points de vue	31
2.3	Schéma général de la méthode proposée	32
2.4	Sphère de visibilité et points de vue	33
2.4.1	Centre de la sphère de visibilité	34
2.4.2	Choix du repère associé à la sphère de visibilité	35
2.5	Génération des points de vue candidats et des images synthétiques associées	36
2.5.1	Nombre de points de vue \mathcal{N}	37
2.5.2	Images synthétiques générées depuis la sphère de visibilité	38
2.6	Fonction score f_{score} pour sélectionner le meilleur point de vue	39
2.6.1	Critère de visibilité $f_{\text{visibilité}}$	39
2.6.2	Autres critères adaptés au type d'inspection	41
2.7	Exemples de choix de meilleur point de vue	42
2.8	Exploitation des points de vue dans le mode d'inspection interactive à l'aide de la réalité augmentée	44
2.9	Conclusion	45

Le contenu de ce chapitre a été publié dans [Ben+19a].

2.1 Introduction et problématique

Un point de vue (ou point d’observation) peut être vu comme une fenêtre à travers laquelle le contenu de la scène est présenté à l’algorithme d’inspection. Le positionnement de la caméra d’inspection à un point de vue donné permet de fournir une vue (i.e. une image) montrant une partie de la scène. En fournissant cette vue, on peut aider l’algorithme d’inspection dans son analyse. Il est donc essentiel de bien choisir les points de vue que l’on va utiliser dans les algorithmes d’inspection. Pour une pièce à inspecter donnée, le choix de points de vue s’effectue hors ligne avant exécuter les algorithmes d’inspection (c’est une étape de préparation préalable à l’inspection proprement dite).

Un des principaux challenges du contrôle automatique de caméra est de proposer aux algorithmes d’inspection une vue sans occultation, et avec la meilleure observation possible de l’objet d’intérêt (élément à inspecter). Le choix des points de vue consiste donc en la recherche du meilleur point de vue par rapport à un ensemble de propriétés (e.g. visibilité de l’information, occultation, etc.).

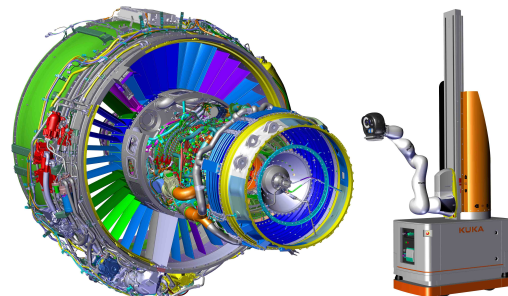
Le contrôle ou le pilotage du choix des points de vue peut varier selon les besoins, d’un contrôle semi-automatisé tel que dans les applications de contrôle interactives (e.g. tablette portée par un opérateur, voir la figure 2.1a), à un contrôle complètement automatisé (e.g. robot équipé de capteurs, voir la figure 2.1b). Il consiste à positionner la caméra d’inspection à la distance de travail de sorte que l’élément à contrôler soit dans le champ visuel de la caméra, afin de pouvoir en faire la meilleure observation possible. Par conséquent, dans le cas du contrôle robotisé, on peut le voir comme une sous-classe de problèmes de planification des trajectoires du robot d’inspection.

Généralement, la planification des trajectoires du robot d’inspection consiste en :

- ordonnancement et planification optimisés des trajectoires du robot en évitant les collisions ¹.
- choix automatique des points d’observation en fonction des éléments à contrôler.



(a) contrôle interactif



(b) contrôle robotisé

Figure 2.1 – (a) contrôle semi-automatisé, (b) contrôle complètement automatisé

1. Dans notre thèse, nous n'avons pas travaillé sur l'exécution de trajectoire sans collisions.

La planification des trajectoires du robot d'inspection est généralement utilisée pour rechercher un mouvement ou une trajectoire qui évite de rentrer en collision avec des éléments de l'environnement. Cependant, la planification des trajectoires du robot ne se résume pas simplement à trouver une trajectoire optimale pour éviter les collisions, cela nécessite en supplément que les trajectoires du robot respectent un certain nombre de contraintes (par exemple maintenir la visibilité de l'élément à contrôler).

Un certain nombre de difficultés sont sous-jacentes au choix de points de vue en environnement complexe :

- Comment déterminer de manière précise et efficace les meilleurs points d'observations dans un environnement complexe ?
- Quel point de vue choisir pour la caméra en fonction de la nature de l'assemblage mécanique et des risques potentiels d'occultation ?

Conscients de ces difficultés, nous proposons dans ce chapitre un algorithme de choix automatique des points de vue, qui va être utilisé dans deux scénarios de contrôle :

1. un scénario d'inspection complètement automatisé à l'aide d'un robot équipé de capteurs.
2. un scénario d'inspection interactive avec un guidage de l'opérateur vers le meilleur point de vue pour la prise d'images via la visualisation d'informations de guidage en réalité augmentée.

Nous présentons une nouvelle approche permettant de calculer la visibilité des objets à contrôler. Ce calcul de visibilité repose sur l'exploitation du modèle CAO pour générer des images synthétiques à partir d'un ensemble de points de vue générés sur une sphère de visibilité.

2.2 Méthodes de choix automatique de points de vue

Généralement, les méthodes de choix automatique de points de vue cherchent à optimiser une certaine métrique permettant de décider à quel endroit il faut placer le système d'acquisition.

LIU et al. [Liu+14] ont présenté une méthode de choix automatique de points de vue pour l'inspection des lignes à haute tension à l'aide d'un drone autonome. Dans leur travail, le choix de points de vue est effectué en deux étapes. La première étape consiste à générer un ensemble de points de vue candidats. Pour ce faire, ils ont utilisé un cylindre tessellé² par des triangles, centré sur l'élément à inspecter et chaque sommet des triangles sur la surface du cylindre est considéré comme un point de vue

2. La tessellation est une méthode qui permet d'assigner des triangles sur une surface donnée.

candidat. La deuxième étape consiste à évaluer chaque point de vue candidat. La pertinence de chaque point de vue est évaluée par une mesure d'entropie. Le point de vue le plus pertinent est celui qui a l'entropie la plus élevée. L'entropie utilisée pour mesurer la pertinence de chaque point de vue est l'entropie de SHANNON [Sha48].

L'évaluation des points de vue candidats par mesure d'entropie a été utilisée dans de nombreuses travaux [And+04 ; Luq+14 ; Sto+04 ; Vaz+01 ; Vaz03]. Comme expliqué dans la thèse de NORMAND [Nor10], l'entropie pour mesurer la pertinence des points de vue consiste à placer une sphère englobante autour du points de vue d'intérêt, puis à mesurer la taille des projections des faces visibles des objets de la scène sur cette sphère.

D'une manière similaire, VASQUEZ-GOMEZ et al. [Vas+09] ont proposé d'utiliser une sphère tessellée par 80 triangles, centrée sur l'élément à inspecter, dont chaque centroïde de sommet de triangles sur la surface de la sphère est considéré comme un point de vue candidat. L'évaluation de chaque point de vue candidat repose sur la méthode de lancer de rayons (*Ray tracing*) [Fol+90]. Cette méthode consiste à lancer des rayons depuis le point de vue candidat vers l'élément à inspecter. Ensuite, elle cherche les points d'impact entre les rayons lancés et l'objet. Le temps nécessaire pour obtenir les scores de l'ensemble de points de vue est très important. En effet, plusieurs de ces rayons lancés iront percuter des objets invisibles et d'autres rayons ne feront littéralement que traverser la scène.

WU et al. [Wu+15] ont proposé un réseau de neurones convolutif CNN (*Convolutional Neural Networks*) pour la prédiction de meilleur point de vue. Pour entraîner le réseau de neurones à trouver le meilleur point de vue, ils ont utilisé différentes catégories d'objets et des poses arbitraires générées à partir de données CAO brutes.

Contrairement aux approches qui exploitent l'espace objet et qui sont très coûteuses en temps de calcul, l'approche que nous avons proposée exploite l'espace image et elle est donc plus rapide. De plus, cette approche est plus efficace puisqu'elle considère la distance d'observation de la caméra en temps réel et elle exploite la totalité du modèle CAO.

2.3 Schéma général de la méthode proposée

La figure 2.2 présente le schéma général de la méthode de sélection du point de vue que nous proposons. Cette méthode exploite le modèle CAO de l'élément à inspecter. Nous générons dans un premier temps une sphère de visibilité centrée sur le centre de gravité de l'élément à inspecter. Elle fournit un ensemble de points de vue candidats. Ensuite, une caméra virtuelle est positionnée successivement à chaque point de vue candidat. Une image synthétique est générée à partir du modèle CAO depuis chacun de ces points de vue.

Afin d'évaluer les points de vue candidats, nous avons mis en place une fonction score f_{score} qui attribue un score (ou note) pour chaque point de vue candidat présent dans la sphère de visibilité. Elle permet donc de mesurer la pertinence de chaque point de vue candidat. Le point de vue ayant le score le plus élevé est sélectionné comme le « meilleur point de vue ».

Dans la suite de ce chapitre, nous détaillons chacune de ces étapes.

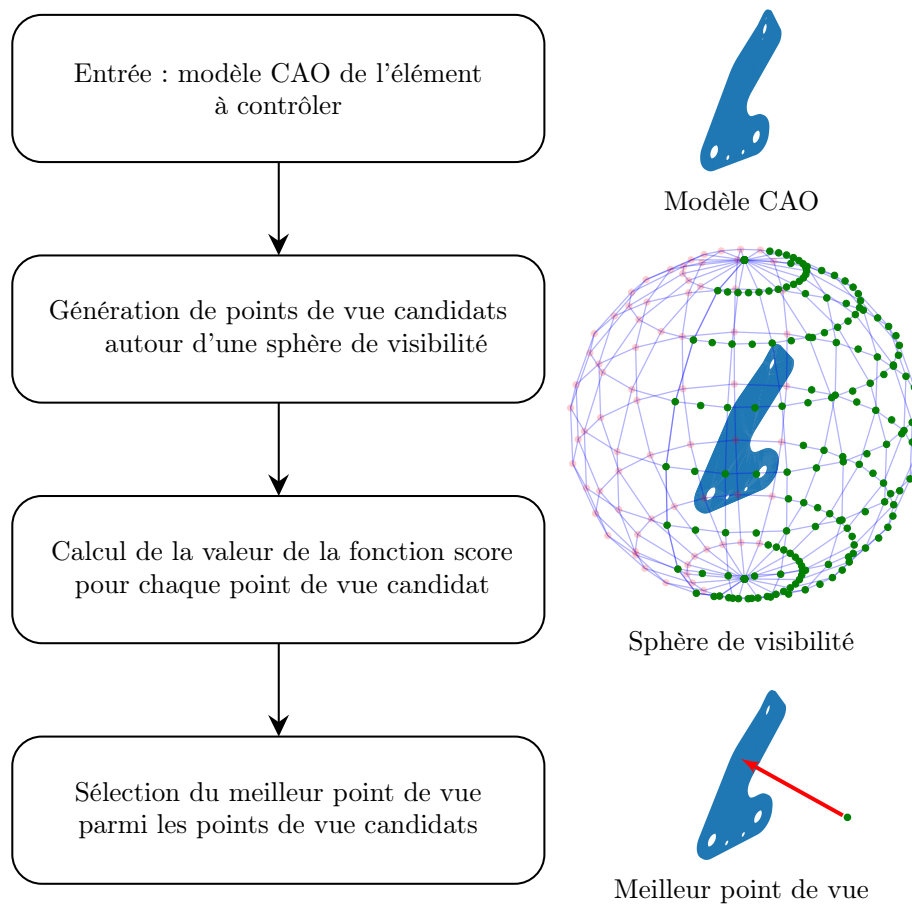


Figure 2.2 – Schéma général de la méthode proposée

2.4 Sphère de visibilité et points de vue

Dans cette section, nous définissons les éléments nécessaires à la compréhension de la méthode de choix de points de vue proposée : notion de point de vue et notion de sphère de visibilité utilisée pour générer les points de vue candidats.

La sphère de visibilité est définie comme un ensemble de points de vue répartis uniformément sur toute la surface d'une sphère 3D selon la plage de colatitude θ (comprise entre 0 et π) et de longitude φ (comprise entre 0 et 2π) (voir la figure 2.3). Cette sphère est centrée sur le centre de gravité de l'élément à inspecter. Le rayon de la sphère de visibilité ρ représente la distance entre la caméra et l'élément à inspecter.

Dans un repère cartésien orthonormé (O, x, y, z) , un point de vue $p_i^{\text{vue}}(x_i, y_i, z_i)$ sur la sphère de visibilité est défini par l'équation suivante :

$$\begin{cases} x_i = c_x + \rho \times \sin(\varphi_i) \times \cos(\theta_i) \\ y_i = c_y + \rho \times \sin(\varphi_i) \times \sin(\theta_i) \\ z_i = c_z + \rho \times \cos(\varphi_i) \end{cases} \begin{cases} \theta_{\min} \leq \theta_i \leq \theta_{\max} \\ \varphi_{\min} \leq \varphi_i \leq \varphi_{\max} \end{cases} \quad (2.1)$$

où la longitude θ_i et la colatitude φ_i désignent la position de la caméra dans l'espace 3D et $C(c_x, c_y, c_z)$ représente le centre de masse du modèle CAO de l'élément à inspecter, et ρ représente la distance de travail.

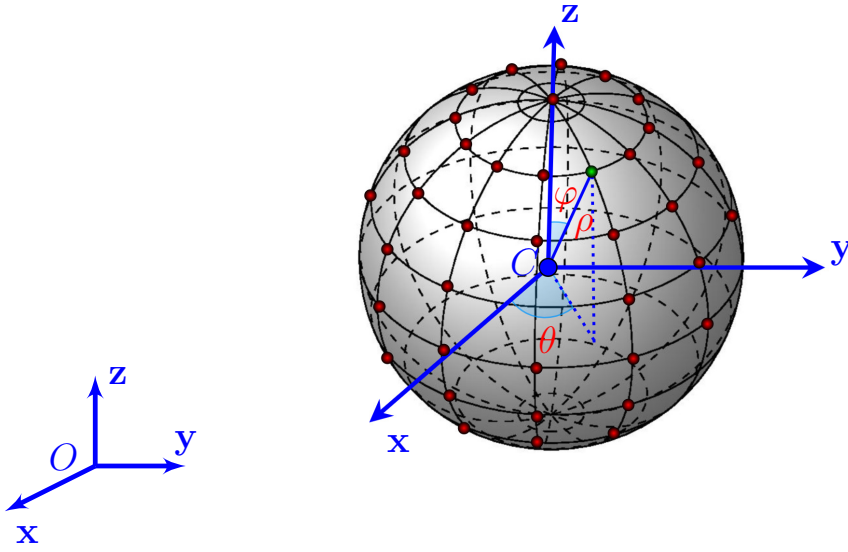


Figure 2.3 – Sphère de visibilité. Les points rouges représentent l'ensemble des points de vue candidats qui ont été générés

2.4.1 Centre de la sphère de visibilité

Le centre de la sphère de visibilité est égal au centre de masse (aussi nommé centre de gravité) du modèle CAO de l'élément à inspecter (voir la figure 2.4). Étant donné un maillage triangulaire 3D (modèle CAO), le centre de masse est calculé comme suit :

$$C = \frac{1}{n} \sum_{i=1}^N p_i \quad (2.2)$$

où les points $p_i = [x_i, y_i, z_i]^T$ pour $i = 1 \dots n$, représentent les sommets du maillage triangulaire.

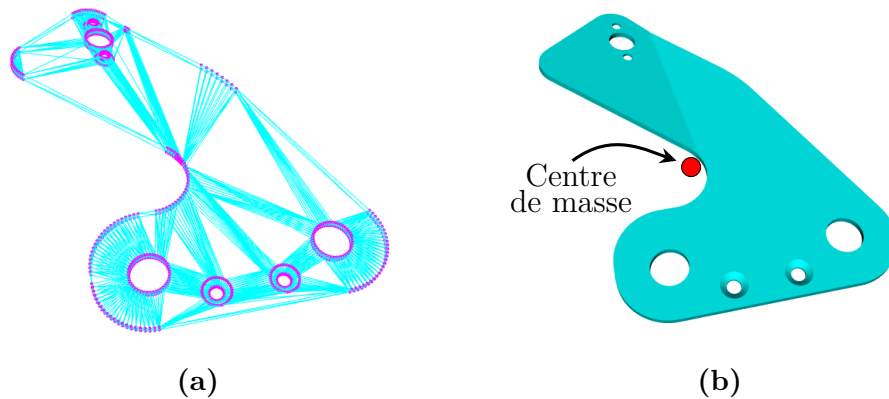


Figure 2.4 – Centre de masse du modèle CAO : (a) maillage triangulaire d'un modèle CAO, les points violets et les lignes bleues claires représentent respectivement les sommets et les arêtes du maillage, (b) le point rouge représente le centre de masse de ce modèle CAO

2.4.2 Choix du repère associé à la sphère de visibilité

Le repère x, y, z associé à la sphère de visibilité est défini comme suit : il est centré sur le centre de la sphère (qui coïncide avec le centre de gravité de l'élément à inspecter) et l'axe z de ce repère (voir figure 2.3) coïncide avec la normale à l'élément à inspecter.

La normale à l'élément à inspecter est déterminée par une méthode d'Analyse en Composantes Principales (ACP) appliquée au modèle CAO de l'élément à inspecter.

L'ACP est une méthode statistique consistant à transformer un ensemble de variables liées entre elles (dites « corrélées ») en un nouvel ensemble de variables orthogonales (dites « décorrélées »). Ces nouvelles variables sont appelées *composantes principales* (ou axes principaux).

Dans le modèle ACP (voir les 3 vecteurs rouge, vert et bleu de la figure 2.5), la première composante principale (vecteur rouge) est celle qui maximise la variance, c'est-à-dire que cette composante correspond à celle où la projection orthogonale des observations présente la variance la plus grande (vecteur rouge). Ensuite, la deuxième composante principale (vecteur vert) correspond à la droite affine maximisant la variance, sous la contrainte d'être orthogonale à la composante principale précédente. Enfin, la troisième composante principale (vecteur bleu) correspond à la droite affine maximisant la variance, sous la contrainte d'être orthogonale au plan engendré par les deux premières composantes principales.

D'une façon générale, le calcul des composantes principales se base principalement sur la diagonalisation de la matrice de covariance ou de la matrice de corrélation [Luu17].

Dans notre cas précis, la composante qui contient la contribution la plus importante, c'est-à-dire qui possède la variance la plus grande, est considérée comme l'axe principal (axe z) de la sphère de visibilité (voir la figure 2.5).

Remarques :

1. Généralement, les deux premières composantes principales du modèle ACP représentent la majeure partie de l'information totale.
2. Comme on peut le voir sur la figure 2.5, le vecteur vert correspond à la deuxième composante principale et le vecteur bleu correspond à la troisième composante principale. Ces deux vecteurs sont déduits du premier vecteur (vecteur rouge) grâce à la contrainte d'orthogonalité et au fait qu'il n'y a ici que trois variables.
3. Enfin, pour plus d'informations sur la méthode d'Analyse en Composantes Principale (ACP), nous invitons le lecteur à se référer au récent manuscrit de thèse de LUU [Luu17].

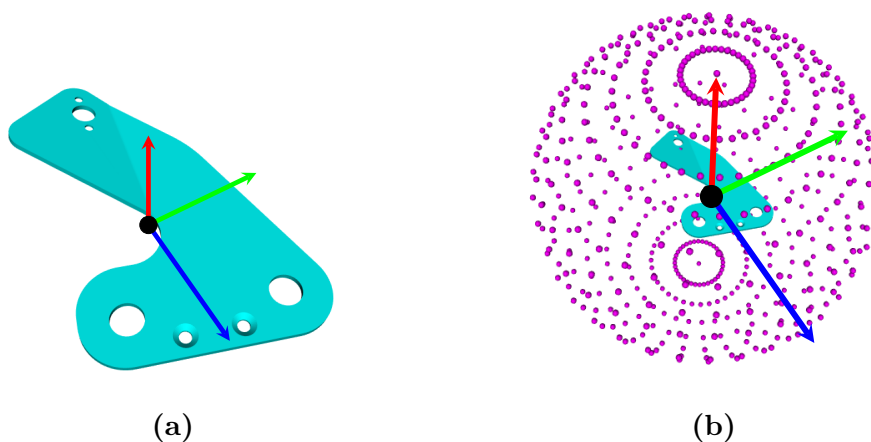


Figure 2.5 – Repère associé à la sphère de visibilité obtenu à partir du centre de masse de l'élément à inspecter et de l'ACP sur le modèle CAO de l'élément : (a) résultat de l'ACP obtenu sur le modèle CAO. Les 3 vecteurs (rouge, vert et bleu) représentent les 3 composantes principales estimées par l'ACP, et la sphère noire représente le centre de masse du modèle CAO. La composante qui contient la variance la plus grande est en rouge, (b) sphère de visibilité orientée résultante

Les raisons pour lesquelles nous avons utilisé l'ACP pour déterminer la direction normale à l'élément à inspecter sont multiples. Tout d'abord, l'ACP permet de retrouver les composantes principales de façon *non paramétrique*. De plus, avec la méthode ACP, on connaît précisément les positions de chaque point de vue par rapport à l'élément à inspecter. En exploitant la sphère de visibilité, nous pouvons choisir directement le point de vue optimal sans passer par une exploration systématique de l'ensemble des points de vue (cf. section 2.7).

2.5 Génération des points de vue candidats et des images synthétiques associées

Dans cette section, nous allons brièvement présenter le processus de génération des points de vue candidats. Tout d'abord, nous générons un ensemble de points de vue candidats autour de la sphère de visibilité présentée dans la section 2.4. Ensuite,

une caméra virtuelle est positionnée successivement à chaque point de vue candidat présent sur la sphère de visibilité. Enfin, une image synthétique est générée à partir du modèle CAO depuis chacun de ces points de vue.

2.5.1 Nombre de points de vue \mathcal{N}

Le nombre de points de vue \mathcal{N} sur la sphère de visibilité est défini à l'aide d'un espacement angulaire χ entre chaque point de vue (voir l'équation (2.3)).

$$\mathcal{N} = 2 \times \frac{(\theta_{max} - \theta_{min}) \times (\varphi_{max} - \varphi_{min})}{\chi^2} \quad (2.3)$$

où $[\theta_{min}, \theta_{max}]$ représente la plage angulaire en colatitute θ , $[\varphi_{min}, \varphi_{max}]$ représente la plage angulaire en longitude φ et χ représente l'espacement angulaire (pas angulaire) qui permet de contrôler la précision de déplacement de la caméra virtuelle.

Le nombre de points de vue sur la sphère de visibilité est défini à l'aide de l'espacement angulaire χ . Cet espacement angulaire permet de définir le degré de liberté de la caméra virtuelle. La figure 2.6 montre certaines sphères générées avec différents espacements angulaires. Conformément à l'équation (2.3), on voit que plus l'espacement angulaire χ est faible, plus le nombre de points de vue \mathcal{N} est élevé, et inversement.

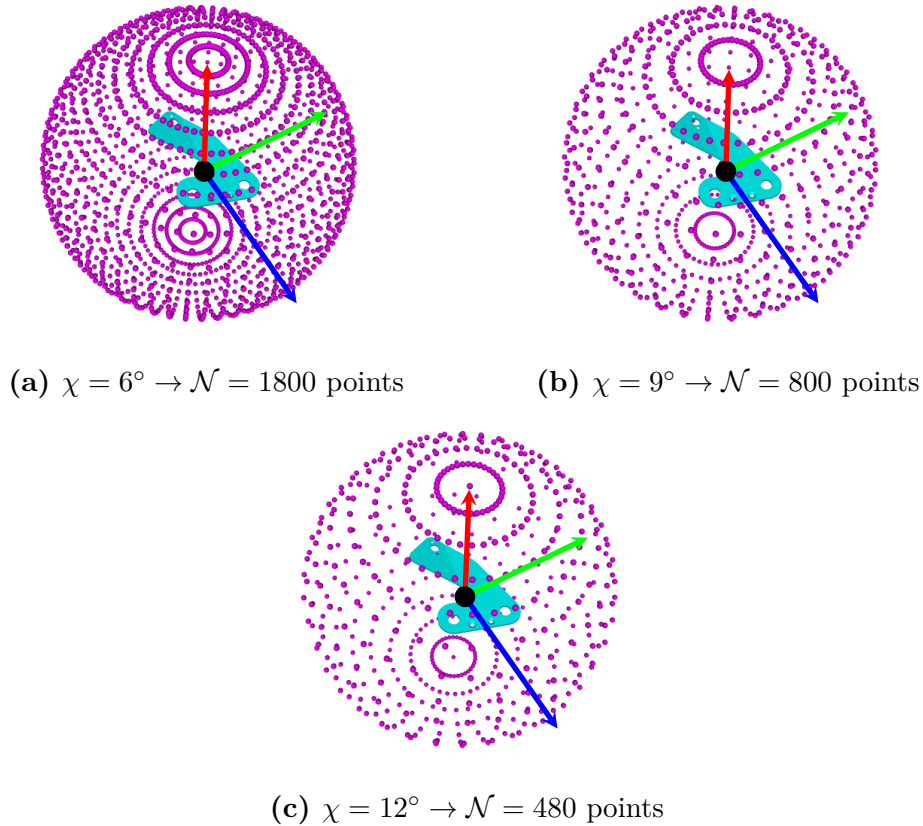


Figure 2.6 – Certaines sphères de visibilité générées avec différentes valeurs d'espacement angulaire (les points violets sont les points de vue candidats)

Pour que l'élément à contrôler soit dans le champ de vue de la caméra et afin de pouvoir en faire la meilleure observation possible, le choix d'un point de vue optimum est nécessaire. En effet, selon la nature de l'assemblage mécanique, les éléments à inspecter peuvent être partiellement cachés ou masqués par d'autres éléments (problème d'occultation). Ceci peut perturber la visibilité et donc l'inspection.

2.5.2 Images synthétiques générées depuis la sphère de visibilité

Une fois que l'ensemble des points de vue candidats est généré, une caméra virtuelle est positionnée successivement à chacun de ces points de vue. Ensuite, depuis chacune de ces positions nous effectuons un rendu colorimétrique de la scène observée par la caméra virtuelle. Ce rendu des couleurs consiste à attribuer aléatoirement, en l'absence de source lumineuse, un indice et une couleur unique pour chaque élément de la scène visible par la caméra.

Comme expliqué par Hertzmann [Her99], afin d'être indépendant des conditions d'illumination, ces images synthétiques générées à partir du modèle CAO doivent être basées sur des rendus non-réalistes.

Un exemple de résultat est présenté sur la figure 2.7.

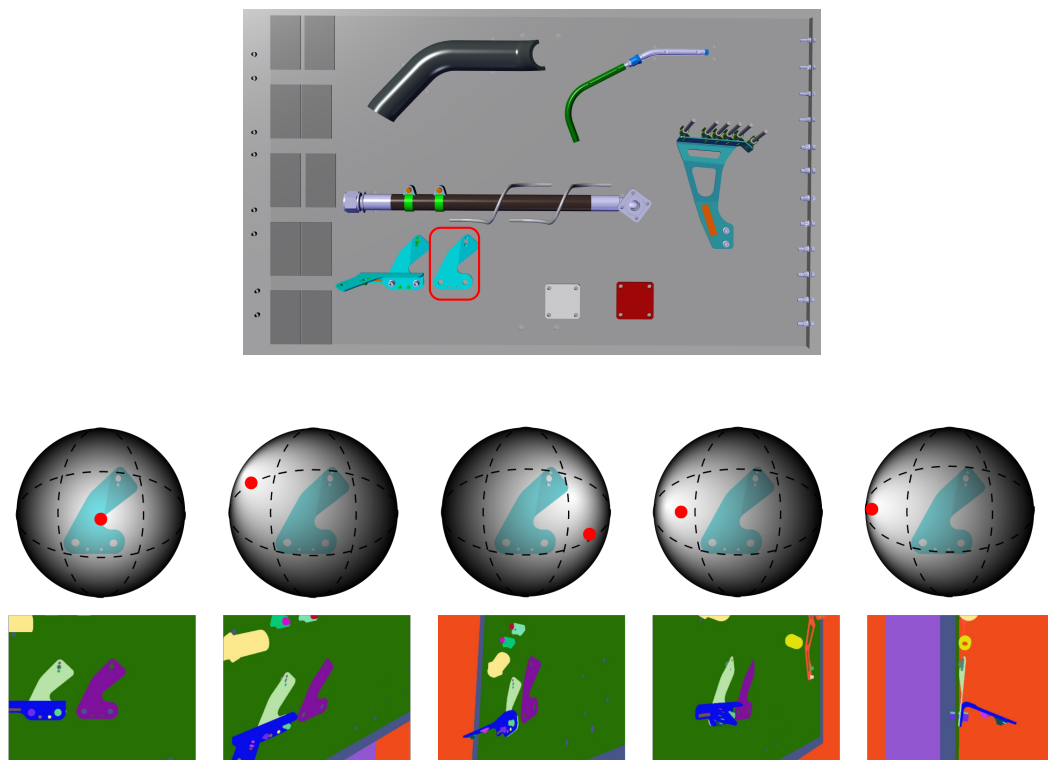


Figure 2.7 – Images synthétiques générées depuis la sphère de visibilité : (1^{ère} ligne) modèle CAO 3D, (2^{ème} ligne) sphères de visibilité. Les points rouges représentent la position de la caméra virtuelle, et (3^{ème} ligne) images synthétiques générées à partir du modèle CAO

2.6 Fonction score f_{score} pour sélectionner le meilleur point de vue

Afin d'évaluer les points de vue candidats, nous avons mis en place une fonction score f_{score} qui attribue un score (ou note) pour chaque point de vue candidat présent dans la sphère de visibilité. Cette fonction score permet donc de mesurer la pertinence de chaque point de vue candidat. Elle exploite un critère de visibilité défini dans la section suivante.

2.6.1 Critère de visibilité $f_{\text{visibilité}}$

Une fois que les images synthétiques sont générées pour chaque point de vue candidat, l'étape suivante consiste à évaluer la visibilité de l'élément à inspecter à partir de chaque image synthétique. Nous avons vu précédemment que le rendu des couleurs utilisé consiste à attribuer aléatoirement, en l'absence de source lumineuse, un indice et une couleur **unique** pour chaque élément visible par la caméra. Pour évaluer la visibilité de l'élément à inspecter, nous prenons l'indice de cet élément et nous calculons pour chaque image synthétique (c.-à-d. pour chaque point de vue candidat) le nombre de pixels visibles par la caméra qui possèdent la même couleur affectée à l'indice. Nous désignons ce score par $f_{\text{visibilité}}$.

La figure 2.8 montre un exemple de résultat obtenu sur trois vues différentes. Dans cet exemple, il s'agit d'inspecter l'élément violet d'indice i et de code (R, V, B) = (128, 17, 156).

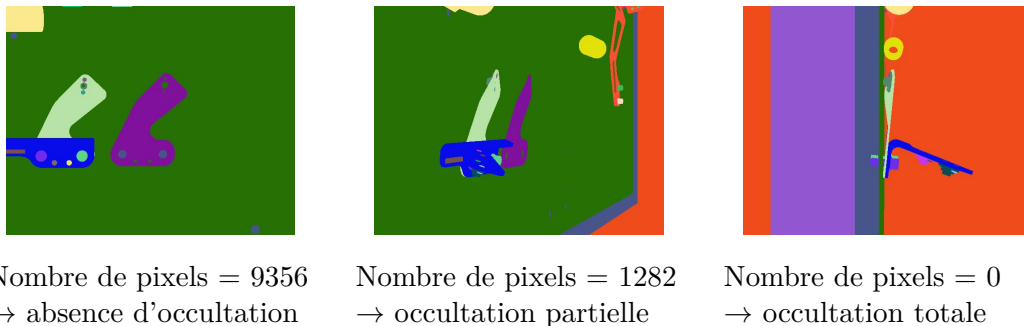
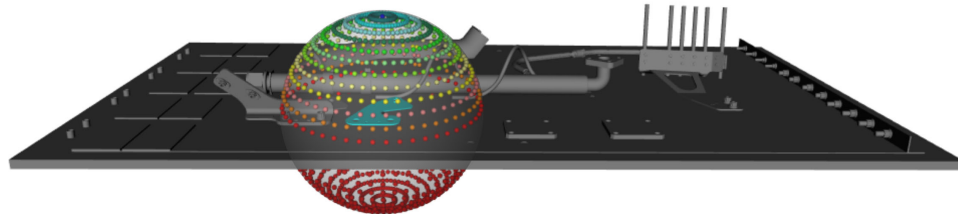


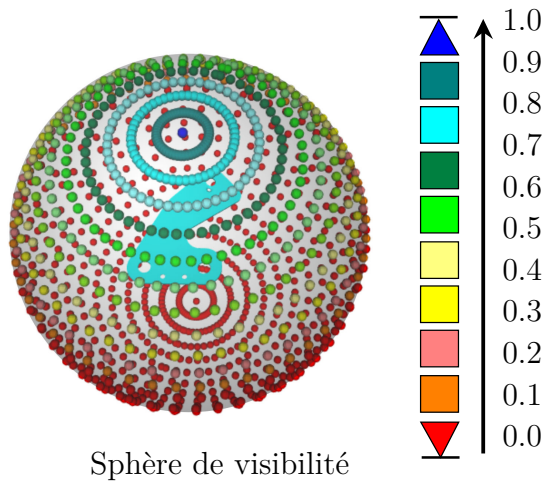
Figure 2.8 – Exemple de résultat de visibilité obtenu sur trois vues différentes : il s'agit d'inspecter l'élément d'indice i et de couleur violette avec le codage (R, V, B) = (128, 17, 156)

Pour un élément à inspecter donné d'indice i , le meilleur point de vue, selon le critère de visibilité, est le point de vue pour lequel l'image synthétique a le plus grand nombre de pixels de couleur identique à celle affectée à l'indice i . Le critère de visibilité consiste alors à privilégier les positions de caméra qui maximisent l'indice visuel de l'élément à inspecter dans l'image.

Selon le critère de visibilité, le meilleur point de vue (avec le score le plus élevé) de la caméra est probablement celui fronto-parallèle à l'objet d'intérêt et le pire point de vue est celui dans lequel l'objet d'intérêt est complètement masqué (voir la figure 2.9).



(a)



Sphère de visibilité

(b)

Figure 2.9 – Distribution de la visibilité sur la sphère : (a) modèle CAO 3D et (b) scores de visibilité $f_{\text{visibilité}}$ normalisés (entre 0 et 1) (rouge : scores très faibles et bleu : scores maximaux). Les points de vue en rouge sont des points « inaccessibles » en raison d'une occultation liée à la nature de l'assemblage mécanique

Remarque : parfois nous ne pouvons trouver que des configurations dans lesquelles l'élément à inspecter est occulté partiellement. Dans ce cas de figure, le choix de points de vue est fortement compromis à cause de la nature de l'assemblage. En effet, comme on le voit sur la figure 2.10, il n'y a aucun point de vue qui permette d'observer l'élément à inspecter dans son intégralité.

Afin de résoudre ce problème d'occultation partielle, nous avons fusionné l'algorithme d'inspection 2D avec un algorithme original de gestion des occultations partielles. Pour plus de détails, nous invitons le lecteur à se référer au chapitre 3.

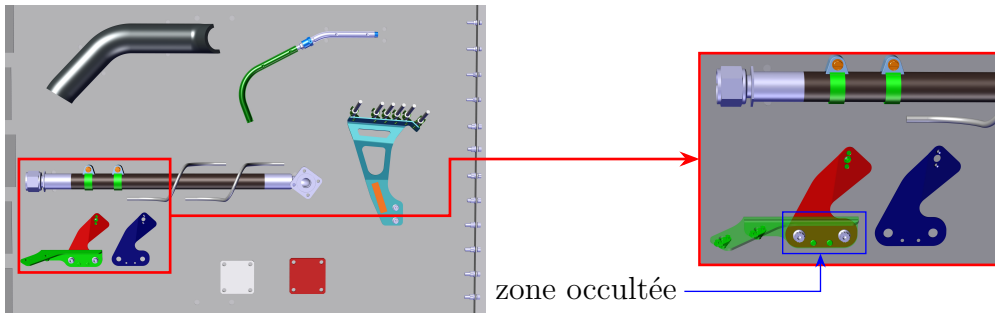


Figure 2.10 – Occultation partielle : une zone de l’élément à inspecter (rouge) est occultée ou masquée par un autre élément (vert semi-transparent)

2.6.2 Autres critères adaptés au type d’inspection

Nous verrons dans les chapitres 3, 4 et 5 que nous avons mis en œuvre 2 types d’inspection différents :

- inspection 2D basée sur l’analyse d’images 2D
- inspection 3D basée sur l’analyse de nuages de points 3D

Pour sélectionner les meilleurs points de vue, dans l’étape hors ligne de choix de point de vue, nous avons choisi une fonction score f_{score} adaptée à chacun des types d’inspection (2D ou 3D).

Inspection 3D : lorsqu’on cherche hors ligne le meilleur point de vue pour faire de l’inspection 3D, ce meilleur point de vue est déterminé en prenant une fonction score définie par :

$$f_{\text{score}} = f_{\text{visibilité}} \quad (2.4)$$

Inspection 2D : lorsqu’on cherche hors ligne le meilleur point de vue pour faire de l’inspection 2D, ce meilleur point de vue est déterminé en prenant une fonction score qui combine $f_{\text{visibilité}}$ et d’autres fonctions (f_{parasite} et $f_{\text{similarité}}$), qui seront définies dans la section 3.8, suivant l’équation suivante :

$$\begin{aligned} f_{\text{score}} &= w_v \times f_{\text{visibilité}} + w_p \times f_{\text{parasite}} + w_s \times f_{\text{similarité}} \\ w_v + w_p + w_s &= 1 \end{aligned} \quad (2.5)$$

Les w_i ($i = v, p, s$) représentent des facteurs de pondération qui attribuent l’importance des différents critères d’évaluation. Ils peuvent être ajustés par l’utilisateur en fonction de l’importance qu’il souhaite donner à chaque critère, pour une application spécifique d’inspection. Ils permettent aussi de normaliser le score final (f_{score}) entre 0 et 1.

Remarque : avant de pouvoir obtenir un score pour chaque point de vue, il est nécessaire de normaliser les fonctions $f_{\text{visibilité}}$, f_{parasite} et $f_{\text{similarité}}$. En effet, les scores provenant de ces différentes fonctions ne sont pas forcément dans le même intervalle. Une fonction peut être dans l'intervalle $[0, 1]$ et une autre dans $[0, k]$ avec $k > 1$. Il faut donc les normaliser afin que ces fonctions soient dans le même intervalle et que l'on puisse les combiner à l'aide des facteurs de pondération sans qu'une fonction ne prenne plus d'importance que les autres.

2.7 Exemples de choix de meilleur point de vue

Dans cette section nous donnons trois exemples qui illustrent comment la méthode proposée permet de choisir le meilleur point de vue en fonction de l'application visée :

Exemple 1 : la figure 2.11 montre un exemple de résultat obtenu pour l'inspection 3D d'un câble en exploitant le critère de visibilité.

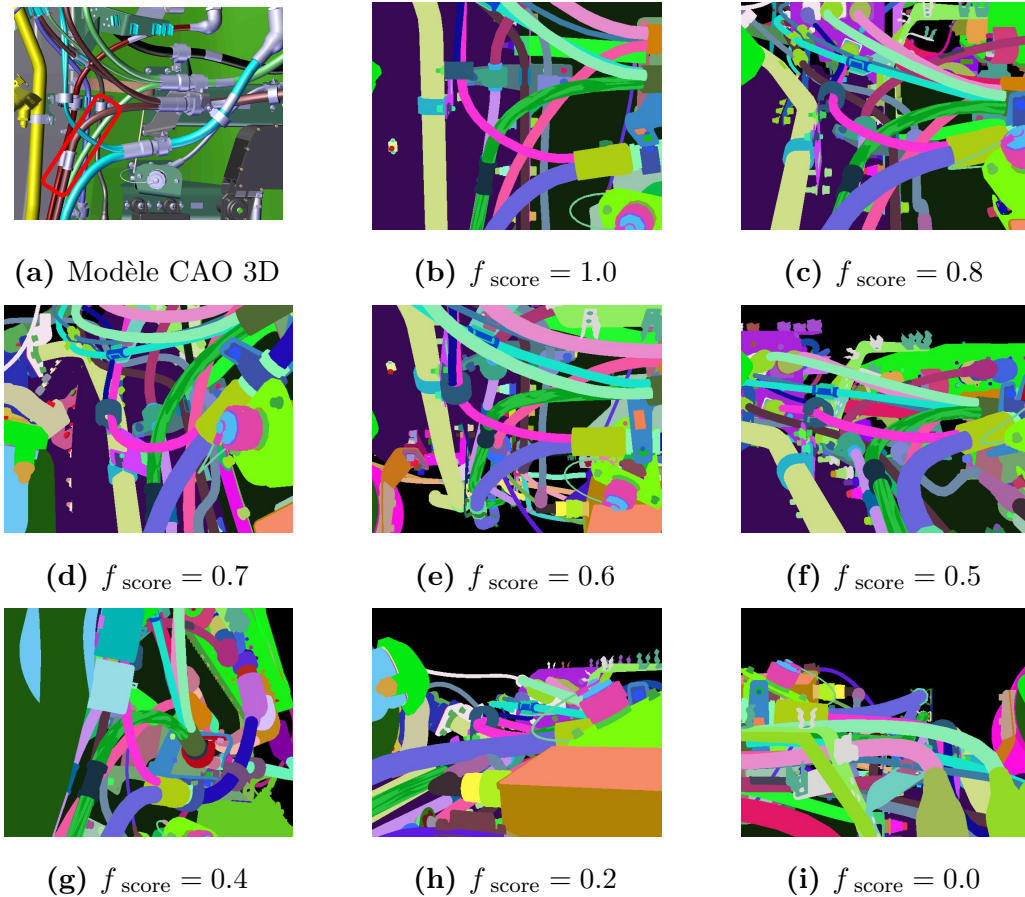


Figure 2.11 – Exemple de résultat obtenu pour l'inspection 3D du câble dans le rectangle rouge : (a) modèle CAO 3D, (b-i) images synthétiques générées depuis les différents points de vue de la sphère de visibilité

Exemple 2 : pour inspecter une vis à tête plate (vérification présence/absence), l'algorithme d'inspection nécessite un point de vue fronto-parallèle (en face de la vis). Le choix de ce point de vue adéquat se fait instantanément en fixant l'angle φ à zéro dans la sphère de visibilité (voir figure 2.12).

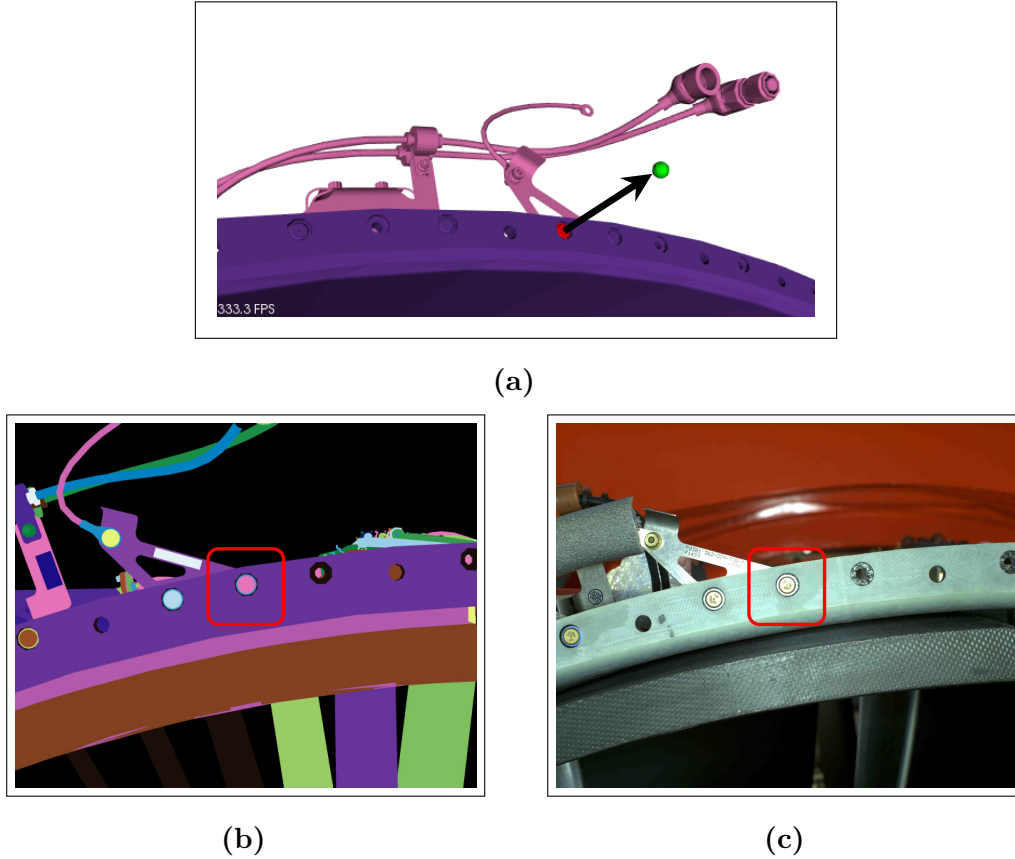
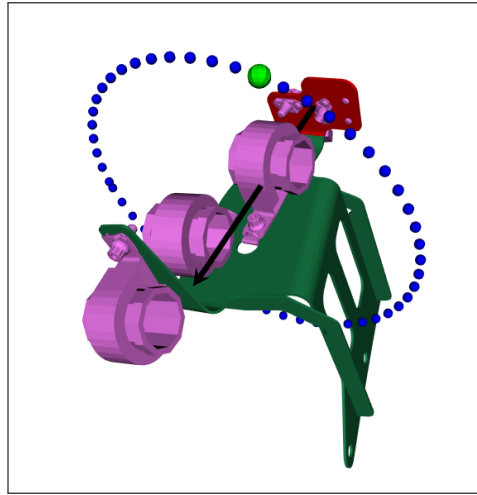
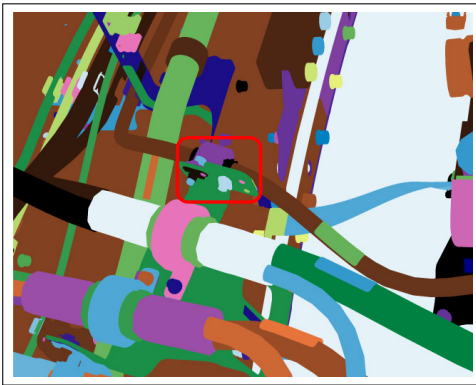


Figure 2.12 – Point de vue fronto-parallèle dans le cas de l'inspection d'une vis à tête plate : (a) le point de vue en vert représente le point de vue fronto-parallèle obtenu en fixant l'angle φ à zéro dans la sphère de visibilité. Le vecteur en noir représente la composante principale du modèle ACP, (b) image synthétique générée à partir du modèle CAO et à partir de ce point de vue, et (c) image réelle obtenue à partir de ce point de vue pour inspecter la vis à tête plate dans le cadre rouge

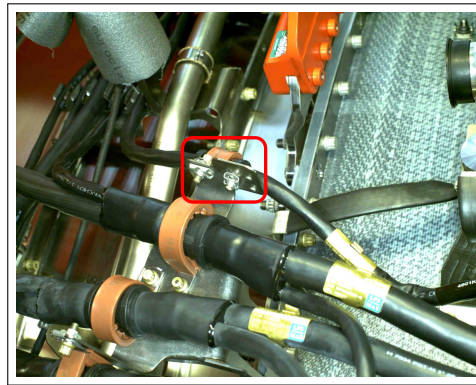
Exemple 3 : pour mesurer la distance minimale entre 1 support rigide et 1 câble, l'algorithme d'inspection nécessite un point de vue latéral (avec une inclinaison de 45° par rapport à la normale de l'élément). Le choix de ces points de vue appropriés se fait instantanément en fixant l'angle φ à 45° dans la sphère de visibilité (voir l'ensemble des points de vue en bleu sur la figure 2.13). Ensuite, pour évaluer ces points de vue latéraux, la fonction score f_{score} (voir la section 2.6) permet de mesurer la pertinence de chacun de ces points de vue simplifiés. Finalement, le point de vue avec le score le plus élevé sera sélectionné (voir le point de vue vert sur la figure 2.13).



(a)



(b)



(c)

Figure 2.13 – Point de vue latéral dans le cas d’une mesure de distance minimale entre 1 support rigide et 1 câble : (a) les points de vue en bleu sont les vues latérales obtenues en fixant l’angle φ à 45° par rapport à la composante principale de l’élément à inspecter ; le point de vue vert est le meilleur point de vue selon la fonction score. Le vecteur en noir représente la composante principale du modèle ACP, (b) image synthétique obtenue à partir du modèle CAO et à partir de ce point de vue, et (c) image obtenue à partir de ce point de vue pour mesurer la distance entre le support dans le cadre rouge et le câble le plus proche

2.8 Exploitation des points de vue dans le mode d’inspection interactive à l’aide de la réalité augmentée

Dans le mode d’inspection interactive où le moyen de guidage/contrôle nécessite de la mobilité, la méthode automatique de choix de points de vue précédemment présentée permet de guider en temps réel les opérateurs dans leur tâche de prises d’images au cours d’une phase d’inspection en réalité augmentée.

La réalité augmentée consiste à superposer en temps réel des éléments visuels 2D ou 3D à la réalité observée par la caméra. Dans notre cas précis, pour un élément à inspecter donné, nous utilisons la réalité augmentée pour superposer en temps réel les points de vue (de la sphère de visibilité) accessibles avec une bonne visibilité et l'image observée par la caméra.

La figure 2.14 montre un exemple d'inspection interactive guidée par la réalité augmentée. Les flèches bleues indiquent les points de vue de la sphère de visibilité accessibles avec une bonne visibilité pour inspecter l'élément dans le rectangle rouge. Dans cet exemple, la réalité augmentée demande à l'opérateur de se déplacer dans le bon angle de prise de vue (en face des flèches bleues).



Figure 2.14 – Guidage de l'utilisateur vers le bon angle de prise de vue à l'aide de réalité augmentée (voir les flèches bleues). L'élément à inspecter est dans le rectangle rouge

2.9 Conclusion

Dans ce chapitre, nous avons présenté une méthode de choix automatique de points de vue qui exploite uniquement la maquette numérique, et qui peut être mise en œuvre suivant 2 scénarios : via un robot équipé de capteurs (mode d'inspection complètement automatisée) ou via une tablette, équipée d'une caméra, portée par un opérateur (mode d'inspection semi-automatique avec l'option d'une assistance de l'opérateur vers le bon angle de vue en réalité augmentée).

En introduction, nous nous sommes demandés comment déterminer de manière précise et efficace les meilleurs points d'observations dans un environnement complexe.

La méthode proposée exploite le modèle CAO de l'élément à inspecter. Nous générons dans un premier temps une sphère de visibilité centrée sur le centre de gravité de l'élément à inspecter. Elle fournit un ensemble de points de vue candidats qui couvrent toutes les positions possibles de la caméra. Ensuite, une caméra virtuelle est positionnée successivement à chaque point de vue candidat. Une prise de vue virtuelle avec génération d'une image synthétique basse résolution est réalisée depuis chacun de ces points de vue. Enfin, nous avons mis en place une fonction score qui attribue un score pour chaque point de vue présent dans la sphère de visibilité et qui permet de mesurer la pertinence de chaque point de vue candidat.

Cette fonction score combine plusieurs critères dont certains sont utilisés dans l'étape même d'inspection (critère f_{parasite} et critère $f_{\text{similarité}}$ dans le cas de l'inspection 2D ; voir sections 3.6.1 et 3.7.1).

Contrairement aux approches existantes qui exploitent l'espace objet et qui sont très coûteuses en temps de calcul, l'approche que nous avons proposée exploite l'espace image et elle est donc plus rapide. L'originalité de la méthode proposée est le fait que le meilleur point de vue est déterminé en tenant compte des contours parasites dans le cas de l'inspection 2D. Les coefficients de pondération utilisés pour la fonction f_{score} permettent de choisir le meilleur point de vue adapté au type d'inspection qui sera réalisé ultérieurement.

Dans les chapitres suivants, nous nous intéressons à l'inspection automatique des éléments présents dans les assemblages mécaniques complexes à partir des meilleurs points d'observation déterminés par la méthode de choix de point de vue présentée dans ce chapitre.

Inspection à partir d'images 2D

« PICTURE, n. A representation in two dimensions of something wearisome in three. »

Ambrose Bierce (1842-1914)

3.1	Introduction	48
3.2	Schéma général de la méthode proposée	50
3.2.1	Inspection via une tablette portée par un opérateur	50
3.2.2	Inspection robotisée	51
3.3	Génération de contours 3D (edgelets)	52
3.4	Estimation de la pose de la caméra	53
3.5	Mise en correspondance 2D/3D	53
3.5.1	Projection 3D/2D	53
3.5.1.1	Projection des edgelets sur l'image 2D	53
3.5.1.2	Erreur de reprojexion	54
3.5.2	Génération de lignes de recherche pour la mise en correspondance 2D/3D	55
3.6	Filtrage des edgelets indésirables	56
3.6.1	Gestion des contours parasites	57
3.6.1.1	Définitions	57
3.6.1.2	Filtrage des contours parasites	59
3.6.2	Gestion des occultations	61
3.6.3	Orientation des gradients	64
3.7	Prise de décision	65
3.7.1	Score de similarité des formes entre les edgelets filtrés et les points de contour 2D appariés	66
3.7.1.1	Contexte de forme	66

3.7.1.2	Mesure de similarité (calcul de distance basé sur le contexte de forme)	66
3.7.2	Taux d'appariement entre les edgelets filtrés et les points de contour 2D appariés	67
3.7.2.1	Pondération d'appariement	67
3.7.2.2	Taux d'appariement	68
3.7.3	Processus de décision	68
3.8	Choix de points de vue dans l'étape de préparation de l'inspection 2D	69
3.9	Résultats expérimentaux	70
3.9.1	Conditions expérimentales	70
3.9.2	Éclairage	71
3.9.3	Présentation des données	72
3.9.3.1	Données utilisées pour la solution robotisée	72
3.9.3.2	Données utilisées pour la solution exploitant une tablette .	74
3.9.4	Détection de contours dans l'image	76
3.9.5	Détermination des paramètres optimaux	80
3.9.5.1	Détermination du seuil d'appariement	80
3.9.6	Métriques d'évaluation pour l'analyse de performances	82
3.9.6.1	Matrice de confusion	82
3.9.6.2	Indicateurs de succès	82
3.9.7	Résultats d'inspection	83
3.9.7.1	Inspection via une tablette portée par un opérateur	84
3.9.7.2	Inspection robotisée	85
3.9.7.3	Interprétation des résultats	85
3.10	Conclusion	86
3.10.1	Synthèse des travaux	86
3.10.2	Limitations de la méthode d'inspection proposée et améliorations futures	86

Le contenu de ce chapitre a été publié dans [Ben+19a].

3.1 Introduction

Dans ce chapitre, nous présentons les méthodes d'inspection basées uniquement sur l'analyse d'images 2D et l'exploitation du modèle CAO de l'élément à inspecter et de son environnement qui servent de référence.

Dans nos travaux, l'utilisation des techniques d'analyse d'images 2D pour l'inspection est privilégiée pour diminuer le temps de calcul et accélérer l'inspection. Il s'agit de contrôler un maximum d'éléments par vision 2D (analyse d'images 2D) et de réserver l'utilisation du capteur 3D (analyse de nuages de points 3D) pour les éléments qui ne peuvent pas être contrôlés par vision 2D. La méthode d'inspection 3D basée sur l'analyse de nuages de points 3D que nous avons développée sera présentée dans les chapitres 4 et 5.

En utilisant une caméra 2D (portée par une tablette, elle-même portée par un opérateur) ou deux caméras 2D (portées par un robot), il s'agit d'inspecter un

assemblage mécanique aéronautique pour vérifier qu'il a été correctement assemblé, i.e. que les éléments sont présents et dans la position souhaitée.

À titre d'exemple, la figure 3.1a présente un exemple d'assemblage mécanique (panneau de test) et la figure 3.1b présente le modèle CAO 3D correspondant. Sur ce panneau, les éléments A et B ont été correctement assemblés mais l'élément C est absent.

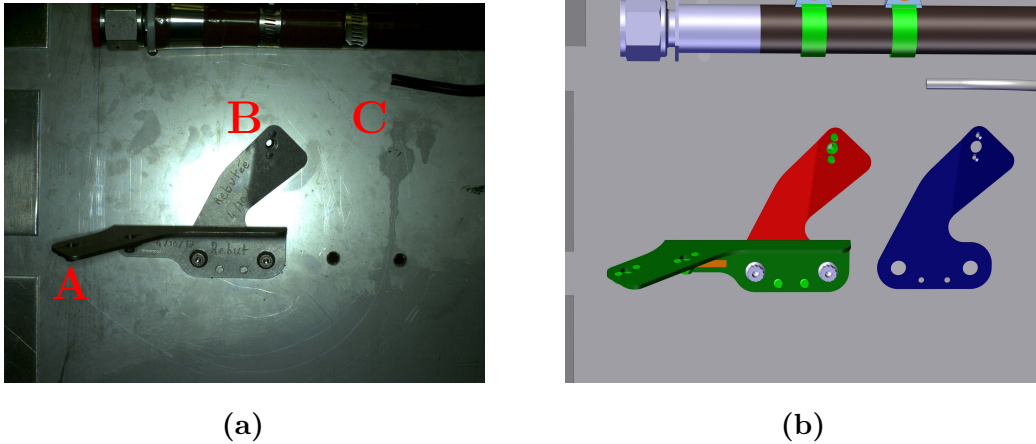


Figure 3.1 – Exemple de problématique d'inspection : (a) panneau de test constitué d'éléments à contrôler (les éléments A et B ont été correctement assemblés, l'élément C est absent), (b) modèle CAO du panneau

Notre méthode d'inspection s'appuie sur la mise en correspondances entre des primitives 2D extraites dans l'image et les primitives 3D issues du modèle CAO de l'objet à inspecter. La pose relative entre l'élément à inspecter et la caméra est fournie par un algorithme de SLAM contraint par le modèle (cf. section 3.4). Les edgelets 3D extraits du modèle CAO sont projetés dans l'image 2D en utilisant la projection perspective à l'aide des paramètres de calibration et la localisation de la caméra.

La mise en correspondance 2D/3D est réalisée en recherchant localement le long d'une ligne de recherche, dans la direction de la normale de l'edgelet projeté, le point de contour 2D le plus proche de l'edgelet (cf. section 3.5). Pour éviter les erreurs de mise en correspondance 2D/3D, nous avons développé une approche originale basée sur l'analyse d'une image synthétique générée à partir du modèle CAO (cf. section 3.6).

La mise en correspondance entre les edgelets projetés et les contours extraits dans l'image 2D permet de faire un diagnostic sur la conformité de l'assemblage (cf. section 3.7).

Dans la section 3.8, nous explicitons la fonction score f_{score} qui est utilisée pour le choix de points de vue en ligne en préparation de l'étape d'inspection 2D.

Enfin, le protocole expérimental et les résultats d'inspection 2D obtenus sont présentés et analysés (cf. section 3.9).

3.2 Schéma général de la méthode proposée

Notre algorithme d'inspection, illustré par la figure 3.2, s'appuie sur la mise en correspondance entre des primitives 2D extraites dans l'image et les primitives 3D issues du modèle CAO de l'objet à inspecter. La pose relative entre l'élément à inspecter et la caméra est fournie par un algorithme de SLAM contraint par le modèle. Les edgelets 3D extraits du modèle CAO sont projetés dans l'image 2D en utilisant la projection perspective à l'aide des paramètres de calibration et de localisation de la caméra. Enfin, la mise en correspondance entre les edgelets projetés et les contours extraits dans l'image 2D permet de faire un diagnostic sur la conformité de l'assemblage.

Les algorithmes de contrôle développés se basent sur l'analyse d'images/vidéos qui exploitent le modèle CAO de l'élément à inspecter qui sert de référence. Ces fonctionnalités de contrôle sont applicables dans un contexte tablette (cf. section 3.2.1) ou robotisé (cf. section 3.2.2) et quel que soit le porteur de la caméra (opérateur, robot, drone).

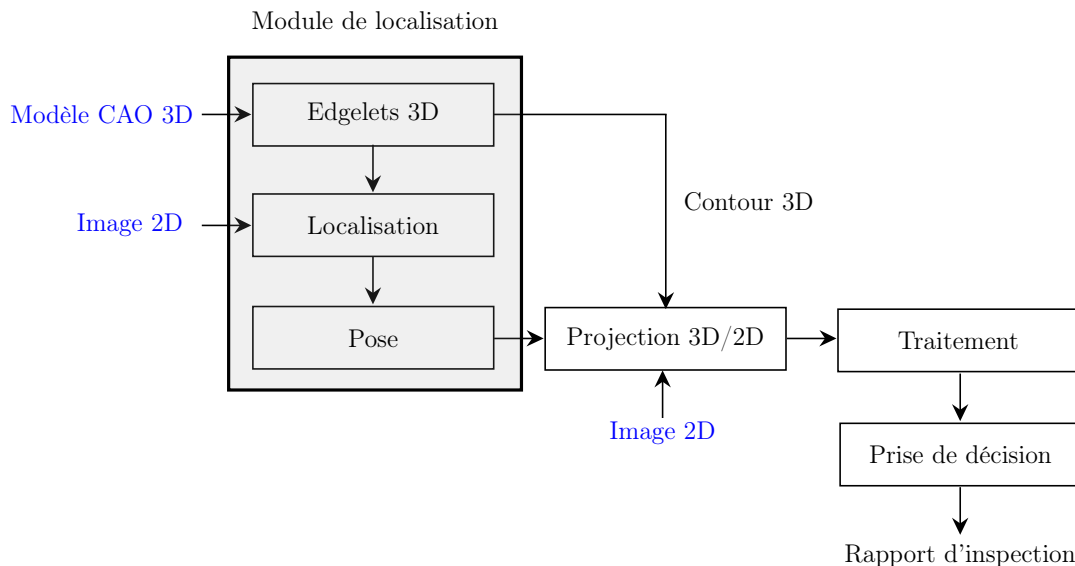


Figure 3.2 – Schéma général de la méthode proposée

3.2.1 Inspection via une tablette portée par un opérateur

Dans un contexte tablette où le moyen de guidage/contrôle nécessite de la mobilité, l'algorithme d'inspection développé se base sur l'analyse du flux vidéo, et la comparaison avec la maquette numérique recalée permet de faire un diagnostic sur la conformité de l'assemblage.

Le dispositif proposé (cf. figure 3.3) est équipé d'une caméra à haute résolution qui permet non seulement d'assurer la localisation de l'élément à contrôler mais

également de réaliser des observations de ce qui est présent dans le champ de la caméra et ainsi de contrôler la conformité des éléments en temps réel.

L'opérateur déplace la tablette devant l'élément à inspecter et des images sont prises en temps réel. L'opérateur peut décider (option de l'algorithme) si la décision sur l'état d'un élément doit être prise sur la base d'une seule image ou si plusieurs images successives doivent être analysées avant de prendre la décision finale (mode *mono-view* versus mode *multi-view*). Dans le cas où l'opérateur sélectionne le mode *multi-view*, les images sont acquises à partir des N meilleurs points de vue déterminés dans l'étape en ligne de choix de points de vue (N est un paramètre choisi par l'opérateur, typiquement entre 5 et 10).

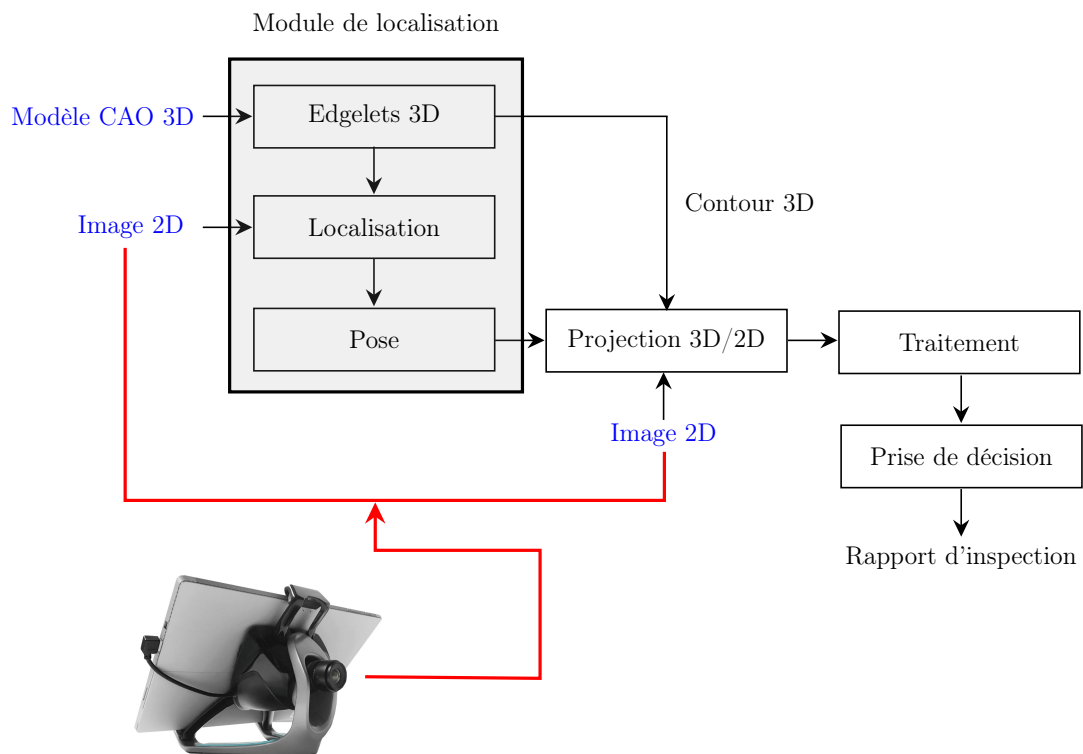


Figure 3.3 – Schéma général de la méthode proposée (inspection via une tablette, équipée d'une caméra, portée par un opérateur)

3.2.2 Inspection robotisée

Dans un contexte robotisé où le moyen de contrôle n'est plus porté par un opérateur mais déplacé de manière automatique et commandé, nous proposons une solution qui exploite un effecteur équipé de deux caméras montées sur la plateforme robotique (cf. figure 3.4). Une caméra de localisation à champ large permet d'assurer la localisation précise de l'effecteur par rapport à la pièce qu'il inspecte, et une caméra à haute résolution avec un champ de vue réduit permet de capturer les détails et d'observer très finement les éléments. Ces deux caméras sont pilotées par un PC industriel (superviseur) qui assure la gestion des caméras et la localisation de

l'effecteur directement en bout de bras. Afin d'analyser les observations, les images acquises sont envoyées vers un PC de traitement pour déterminer les écarts et en déduire la conformité des éléments. L'interaction entre l'effecteur, le superviseur et le PC de traitement est gérée par le middleware ROS.

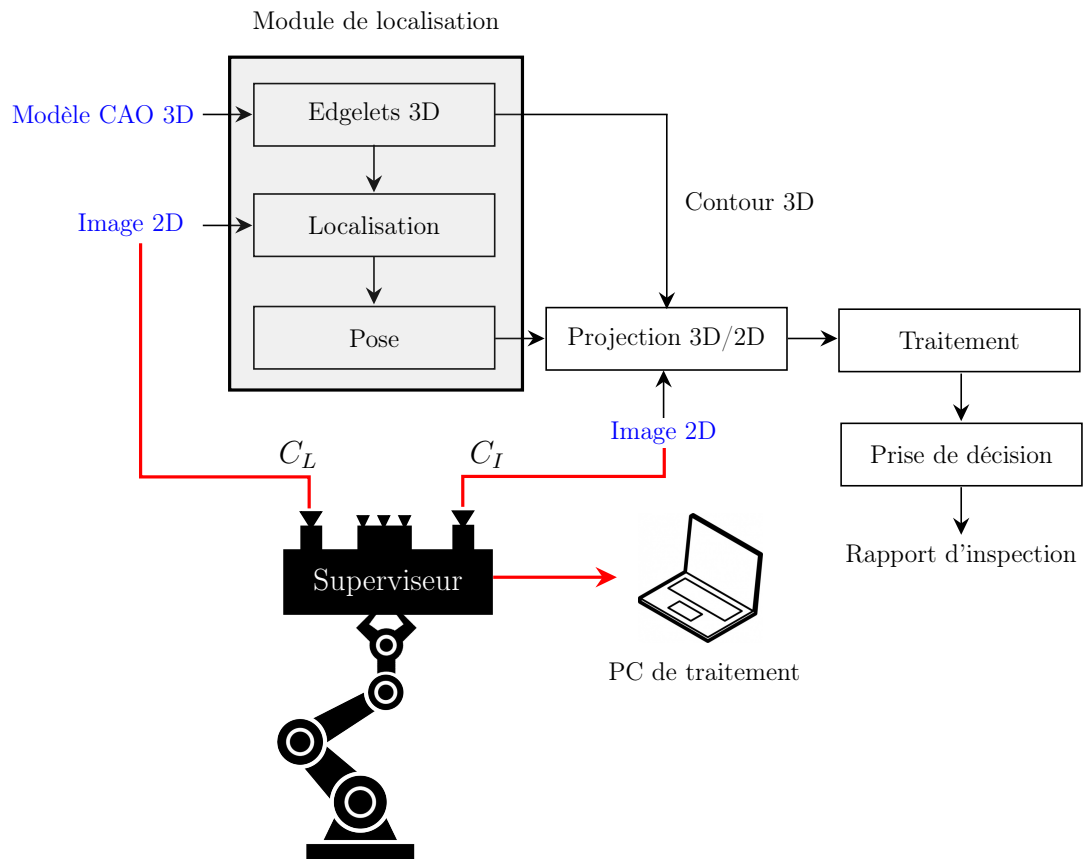


Figure 3.4 – Schéma général de la méthode proposée (inspection robotisée)

3.3 Génération de contours 3D (edgelets)

Un edgelet $E_i(p_i, \vec{t}_i)$ est défini par un point 3D p_i et un vecteur orientation \vec{t}_i , tangentiel à la surface au point p_i . L'ensemble des points p_i représentent les points de contour 3D du modèle CAO (cf. figure 3.5).

Ces edgelets (primitives géométriques 3D issues du modèle CAO) sont dits statiques du fait de leur génération en ligne et indépendamment du point de vue de la caméra. Ils sont extraits des arêtes franches du modèle 3D de l'élément à inspecter dans l'espace image par rendu de synthèse.

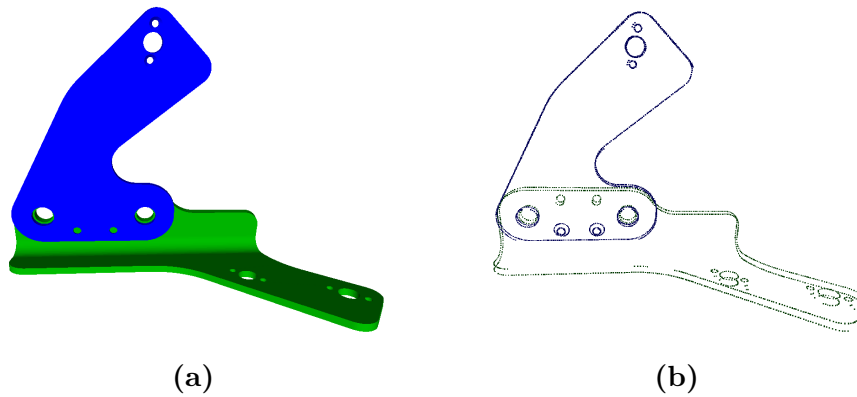


Figure 3.5 – Génération des edgelets : (a) exemple d'un modèle CAO, (b) edgelets générés à partir de ce modèle CAO

3.4 Estimation de la pose de la caméra

Il s'agit de localiser la caméra par rapport à un objet 3D complexe de la scène en nous appuyant sur la connaissance du modèle CAO de cet objet. La localisation d'objet consiste généralement à établir des correspondances entre des primitives 2D observées dans une image 2D issue de la caméra et les primitives 3D issues du modèle CAO de l'objet.

Afin de faciliter l'étape de mise en correspondance et d'améliorer l'estimation de la pose, le processus d'extraction d'edgelets (décrit dans la section 3.3) est combiné à un algorithme de SLAM contraint par le modèle développé par DIOTA.

La pose de la caméra est estimée en minimisant l'erreur de reprojection entre les edgelets 3D extraits du modèle et leurs observations associées dans l'image 2D. Pour plus d'informations nous invitons le lecteur à se référer au manuscrit de thèse de LOESCH [Loe17]. L'erreur de reprojection est définie dans la section suivante.

Grâce à cette technique de recalage basée modèle, nous sommes en mesure de relier en temps réel la pièce physique avec sa représentation virtuelle.

À noter que nous n'avons pas développé le module d'estimation de pose temps-réel. Nous avons utilisé le module développé par DIOTA (expertise DIOTA) et nous avons exploité dans nos travaux la pose fournie par ce module.

3.5 Mise en correspondance 2D/3D

3.5.1 Projection 3D/2D

3.5.1.1 Projection des edgelets sur l'image 2D

La pose relative entre l'élément à inspecter et la caméra étant connue (c'est une des forces des méthodes d'inspection proposées), les edgelets 3D extraits du modèle CAO sont projetés dans l'image 2D en utilisant la projection perspective à l'aide des

paramètres de calibration et la localisation de la caméra. La figure 3.6 montre deux exemples de résultats de projection des edgelets sur l'image 2D.

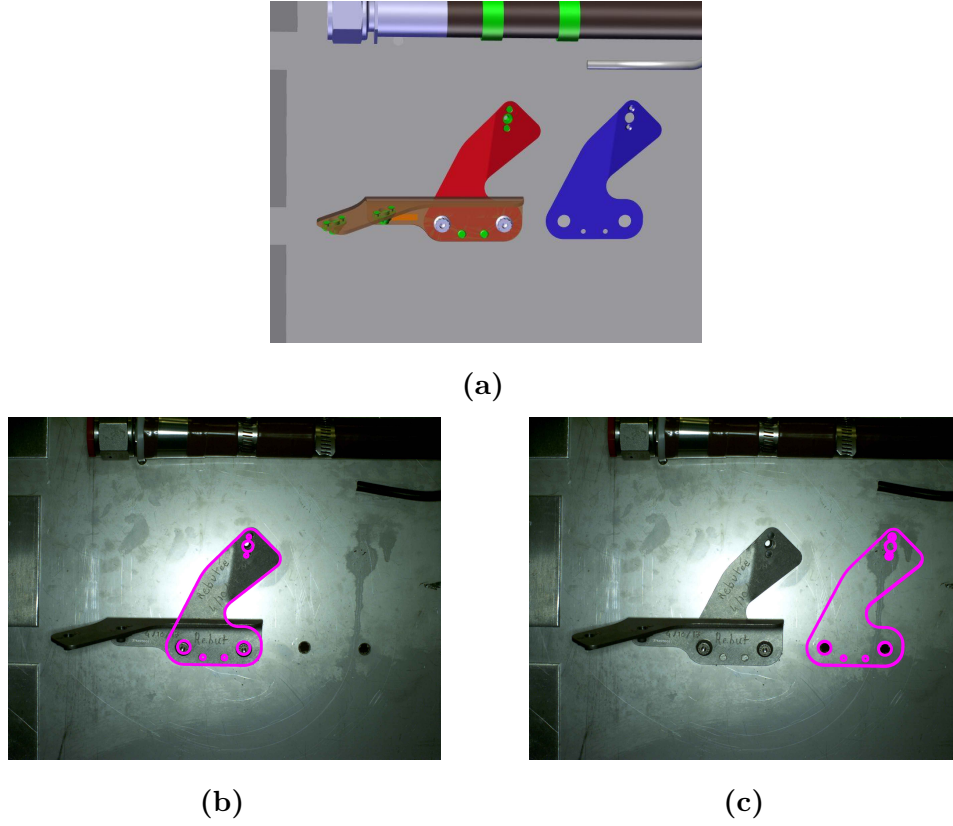


Figure 3.6 – Projection des edgelets (points 3D) sur l'image d'inspection (image 2D) : (a) modèle CAO du panneau de test, (b) élément A présent sur le panneau (en rouge sur le modèle CAO), (c) élément C absent du panneau (en bleu sur le modèle CAO)

3.5.1.2 Erreur de reprojection

Lorsque les edgelets sont projetés sur l'image 2D, il est nécessaire de définir une erreur permettant de mesurer la qualité de cette projection. Il s'agit de mesurer la distance 2D entre la projection des edgelets prédits et des contours réels extraits dans l'image (voir figure 3.7). Cette distance sera désignée sous le terme d'erreur de reprojection par la suite :

$$e_r = \|q - \mathcal{P}(P)\| = \|q - p\| \quad (3.1)$$

où \mathcal{P} correspond à la fonction de projection perspective qui permet de passer des coordonnées homogènes aux coordonnées euclidiennes [Hor+95], p représente la projection d'un edgelet P dans l'image 2D et q représente un point de contour 2D.

L'évaluation par DIOTA de son module de localisation a permis de déterminer une erreur de reprojection de l'ordre de 2 pixels.

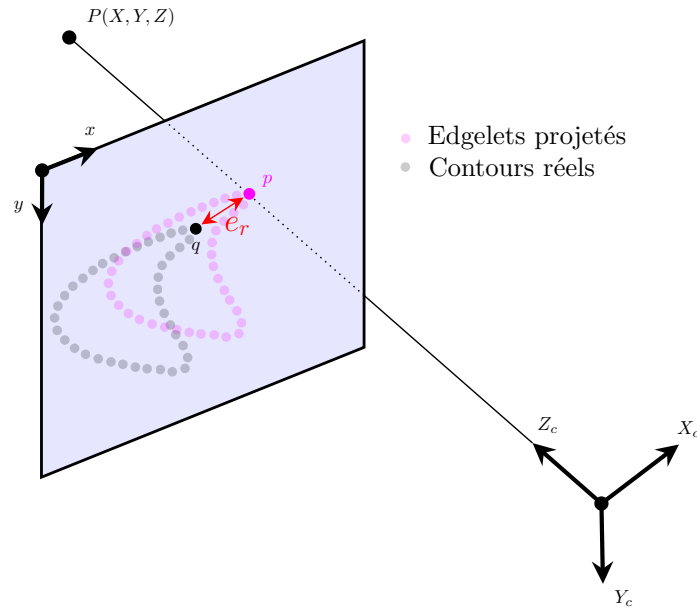


Figure 3.7 – Erreur de reprojection : distance 2D entre la projection des edgelets prédits et des contours réels extraits dans l'image

3.5.2 Génération de lignes de recherche pour la mise en correspondance 2D/3D

La mise en correspondance 2D/3D est réalisée en projetant dans l'image 2D les edgelets extraits du modèle CAO (voir figure 3.9a) et en recherchant localement le long d'une ligne de recherche l_i définie à l'aide de l'orientation 3D des edgelets (voir la figure 3.8) le point de contour 2D le plus proche de l'edgelet projeté.

Pour chaque edgelet $E_i(p_i, \vec{t}_i)$ nous générons un vecteur normal sortant de la surface \vec{n}_i^+ (dirigé de l'intérieur vers l'extérieur tel que $\angle(\vec{t}_i, \vec{n}_i^+) = +90^\circ$) et un vecteur normal rentrant de la surface \vec{n}_i^- (dirigé de l'extérieur vers l'intérieur tel que $\angle(\vec{t}_i, \vec{n}_i^-) = -90^\circ$) (voir figure 3.9b).

Une ligne de recherche l_i est définie par la concaténation de deux vecteurs \vec{n}_i^+ et \vec{n}_i^- . La longueur de la ligne de recherche $\|l_i\|$ est définie à l'aide de l'équation (3.2).

$$\|l_i\| = 4 \times e_r \quad (3.2)$$

où e_r correspond à l'erreur de reprojection définie par l'équation (3.1).

Compte tenu de l'erreur de reprojection moyenne estimée par DIOTA, nous avons dans notre cas : $\|l_i\| = 8$ pixels, $\|\vec{n}_i^+\| = 4$ pixels et $\|\vec{n}_i^-\| = 4$ pixels.

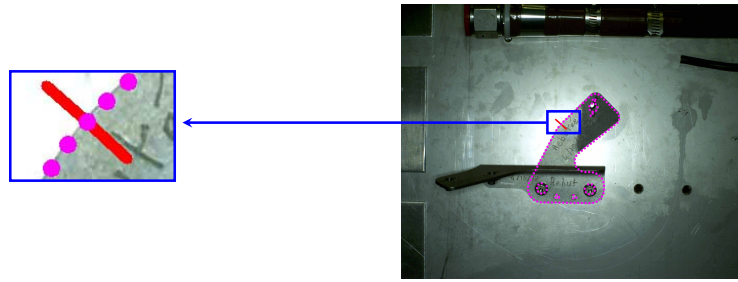
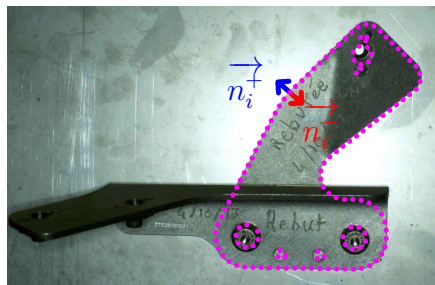


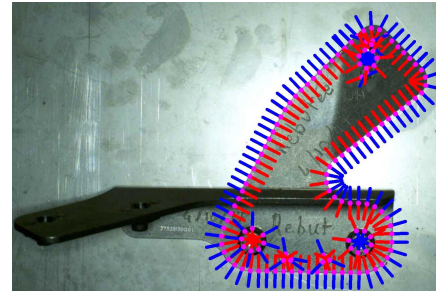
Figure 3.8 – Ligne de recherche



(a)



(b)



(c)

Figure 3.9 – Génération de lignes de recherche (images zoomées) : (a) projection des edgelets sur l'image d'inspection, (b) un exemple de ligne de recherche avec le vecteur normal sortant dirigé vers l'extérieur de la surface (en bleu) et le vecteur normal rentrant dirigé vers l'intérieur de la surface (en rouge), (c) génération de lignes de recherche

3.6 Filtrage des edgelets indésirables

La mise en correspondance 2D/3D est réalisée en projetant dans l'image 2D les edgelets extraits du modèle CAO de l'élément à inspecter et en recherchant localement, le long d'une ligne de recherche l_i dans la direction de la normale de l'edgelet projeté, le point de contour 2D le plus proche de l'edgelet. Cependant, au cours de la phase de mise en correspondance 2D/3D, le risque de mauvais appariement est possible lorsque plusieurs contours réels sont localement proches dans l'image 2D.

À noter que la méthode que nous avons utilisée pour extraire les contours 2D dans les images sera décrite et commentée dans la section 3.9.4.

Lorsque l'on cherche à mettre en correspondance un edgelet et un point de contour 2D, les problèmes les plus fréquemment rencontrés dans notre situation sont :

- Pour un edgelet donné, il est possible d'avoir un appariement avec un contour parasite (voir définition dans la section 3.6.1) d'un élément proche de l'élément à inspecter.
- Pour un edgelet donné, il est possible d'avoir un appariement avec un contour occultant d'un autre élément qui masque l'élément à inspecter.
- Pour un edgelet donné, il est possible d'avoir un appariement avec un contour ayant une orientation différente (par exemple appariement avec un bruit dans l'image contour).

Afin d'éliminer les appariements qui ne vérifient pas certaines propriétés, nous distinguons deux types de traitements qui seront détaillés par la suite :

- **Traitement préliminaire** : l'objectif est de limiter le risque d'effectuer un appariement erroné en éliminant les edgelets appariés avec des points de contour parasite (section 3.6.1) et/ou des points de contour occultant (section 3.6.2). Ce type de traitement permet de réduire la zone de recherche par l'élimination des edgelets indésirables.
- **Traitement a posteriori** : après avoir effectué la mise en correspondance, l'objectif est d'éliminer les edgelets appariés avec des points de contour 2D ayant une orientation différente de celle des edgelets projetés (section 3.6.3). Ce type de contour 2D indésirable est probablement dû au bruit dans l'image contour ou à des artefacts.

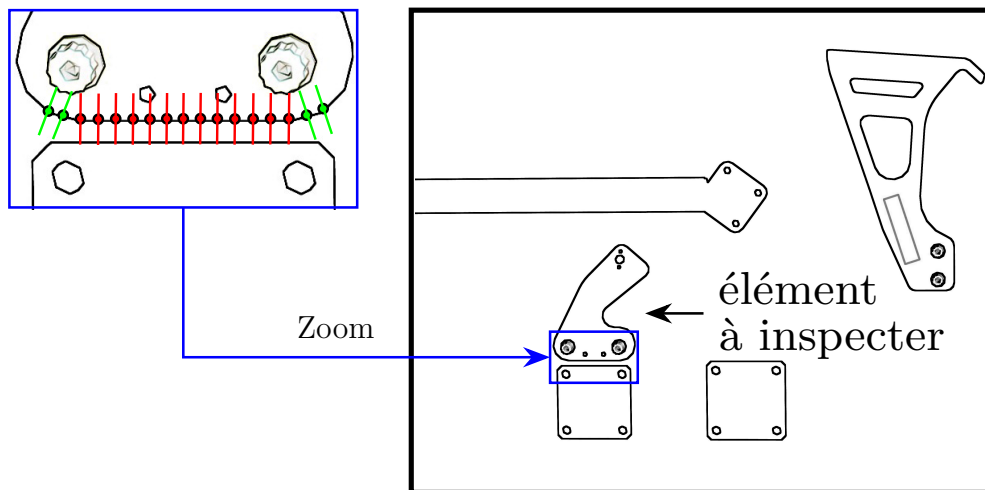
3.6.1 Gestion des contours parasites

3.6.1.1 Définitions

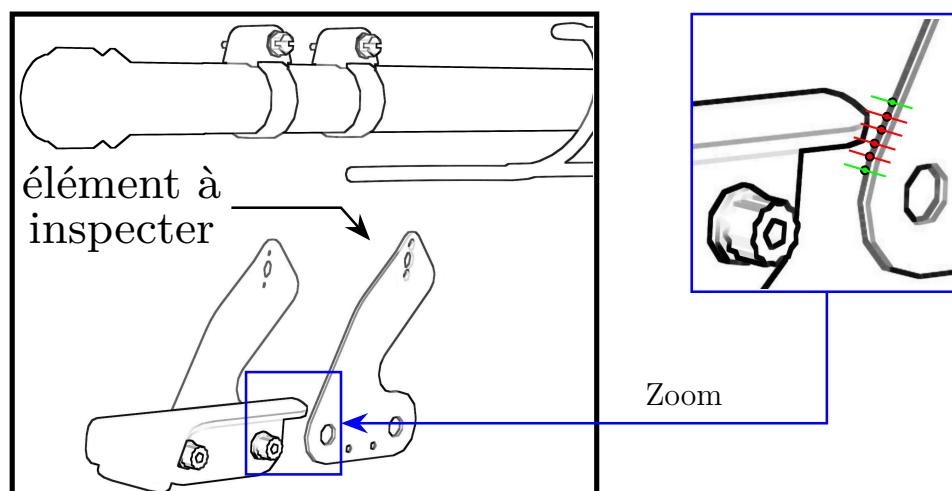
Définition 1 - Contour parasite : c'est un contour qui ne correspond pas aux contours que l'on cherche. Plus précisément, c'est un contour qui appartient à un autre élément très proche de l'élément à inspecter, comme illustré sur la figure 3.10. Nous avons décidé qu'un point 2D appartient à un contour parasite lorsque ce point n'appartient pas au contour de l'élément à inspecter et se situe à une distance d inférieure à la moitié de la longueur de la ligne de recherche l_i (présentée dans la section 3.5.2).

Les principales causes d'apparition des contours parasites sont :

- le cas où l'assemblage mécanique est composé d'objets qui se recouvrent partiellement ou très proches les uns des autres (voir figure 3.10a).
- le cas où le moyen de guidage/contrôle nécessite de la mobilité, le changement de point de vue de caméra peut provoquer l'apparition de contours parasites qui peuvent rendre l'appariement erroné (voir figure 3.10b).



(a)



(b)

Figure 3.10 – Exemple de contours parasites. Les lignes de recherche correspondant aux appariements piégés par des contours parasites sont tracées en rouge dans deux cas : (a) cas où l'assemblage mécanique est composé d'un objet très proche de l'élément à inspecter, (b) cas où un changement de point de vue de la caméra provoque l'apparition de contours parasites.

Définition 2 - Image contexte d'un élément à inspecter : l'image contexte d'un élément à inspecter est une image synthétique générée à partir du modèle CAO et d'une position particulière de la caméra *sans l'objet à inspecter* mais avec les objets faisant partie du contexte. Théoriquement (si la localisation est fiable), cette image possède exactement la même perspective que l'image réelle (voir figure 3.11).

Comme expliqué par Hertzmann [Her99], afin d'être indépendant des conditions d'illumination, ces images synthétiques générées à partir du modèle CAO doivent être basées sur des rendus non-réalistes. Travailler dans l'espace image (2D) plutôt que dans l'espace objet (3D) permet de réduire le temps de calcul.

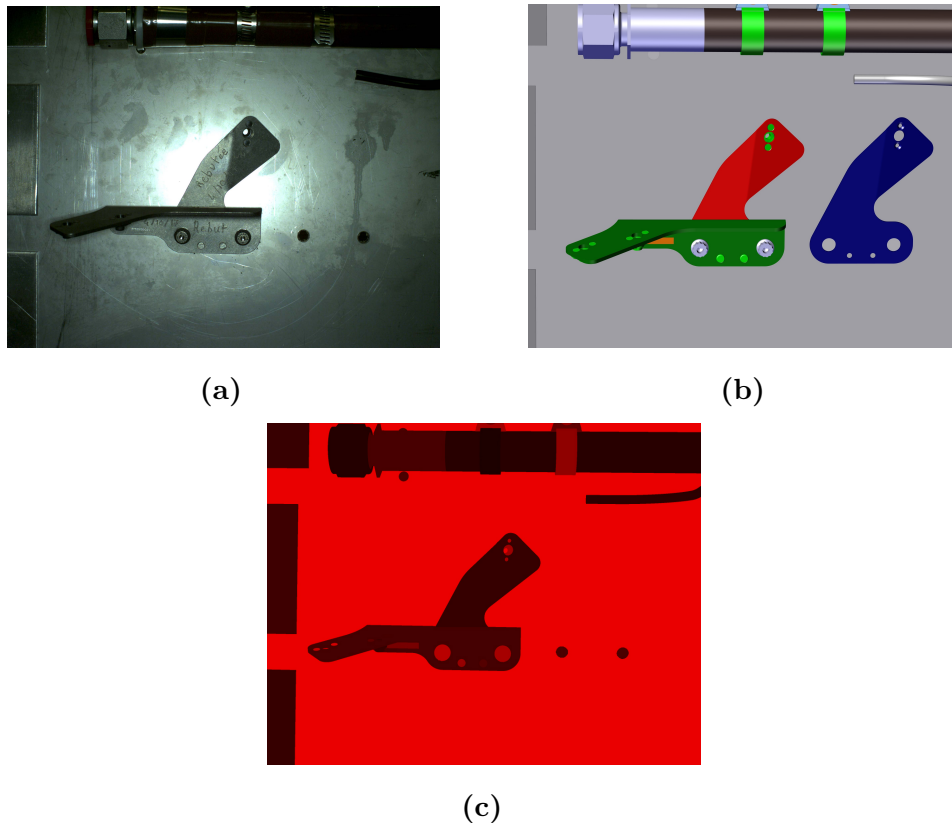


Figure 3.11 – Exemple d'image contexte : (a) image réelle, (b) modèle CAO, (c) image contexte de l'élément à inspecter (cas de l'élément bleu) : elle contient tous les éléments du modèle CAO sauf l'élément à inspecter

3.6.1.2 Filtrage des contours parasites

Les contours parasites peuvent conduire à un appariement erroné notamment dans le cas de l'absence de l'élément à inspecter. En effet, dans ce cas, il est possible d'avoir un appariement avec un contour parasite d'un élément proche de l'élément à inspecter. Par conséquent, il serait particulièrement intéressant de pouvoir détecter ces contours parasites et de limiter leur influence sur l'appariement.

Afin de pallier le problème d'appariement des edgelets avec des contours parasites, nous proposons une approche basée modèle qui exploite l'analyse d'une image synthétique générée à partir du modèle CAO (image contexte).

La figure 3.12 résume le processus de filtrage des contours parasites. Tout d'abord, nous générons l'image contexte (présentée précédemment) de l'élément à inspecter en générant une image synthétique (non-réaliste) à partir du modèle CAO. Ensuite, nous calculons l'image gradient de l'image contexte. Une fois l'image gradient calculée, nous projetons les edgelets sur cette image. Pour chaque edgelet projeté, nous cherchons localement le long de la ligne de recherche (définie dans la section 3.5.2), le point de contour 2D le plus proche de l'edgelet projeté. Enfin, les points de contours 2D trouvés sont nécessairement des contours parasites et les edgelets appariés avec ces contours sont rejetés (voir figure 3.13).

Par la suite, nous ne prendrons pas en compte les edgelets appariés avec des contours parasites, car, par définition, ils représentent de mauvais appariements avec des contours d'autres éléments proches de l'élément à inspecter.

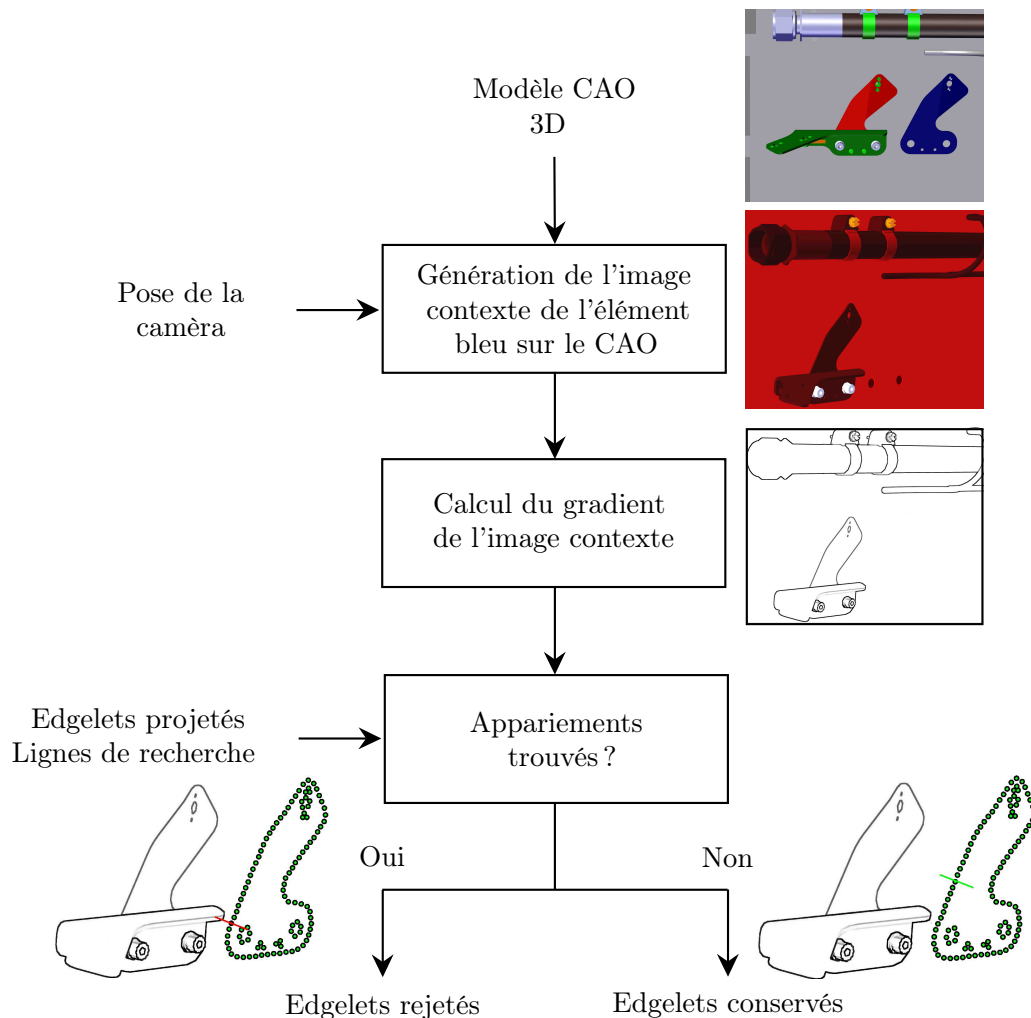


Figure 3.12 – Processus de filtrage des contours parasites

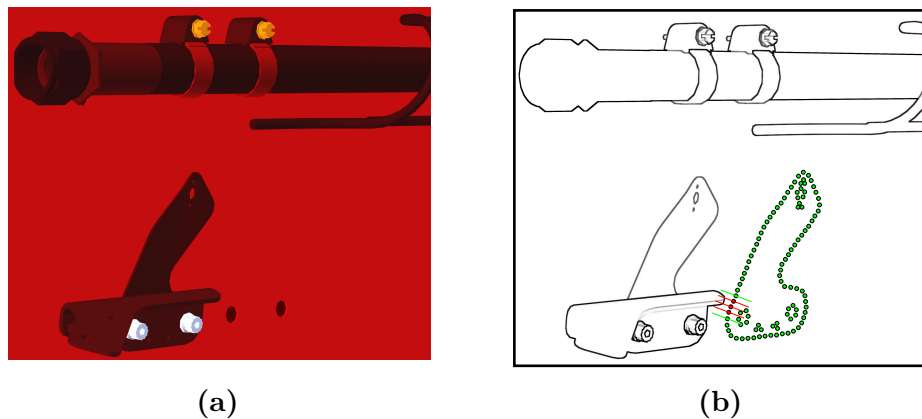


Figure 3.13 – Filtrage de contours parasites : (a) image contexte de l'élément à inspecter, (b) gradient de l'image contexte avec les edgelets conservés (vert) et les edgelets rejetés (rouge)

3.6.2 Gestion des occultations

La réalisation de tâches d'inspection suppose que les indices visuels nécessaires sont disponibles. Afin de résoudre le problème d'occultation totale des indices visuels et d'assurer la visibilité maximale de l'élément, un algorithme de sélection automatique des points de vue a été présenté dans le chapitre 2. Cependant, selon la nature de l'assemblage mécanique, les éléments à inspecter peuvent être partiellement cachés ou masqués par d'autres éléments, comme illustré sur la figure 3.14.

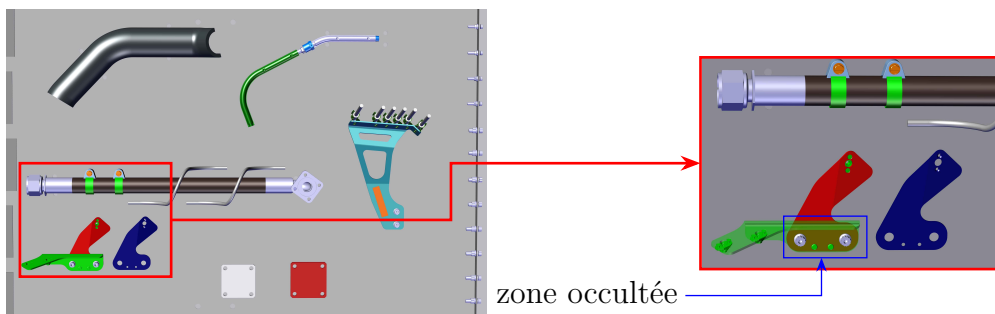


Figure 3.14 – Occultation partielle : une zone de l'élément à inspecter (rouge) est occultée ou masquée par un autre élément (vert semi-transparent)

Le problème d'occultation partielle est souvent considéré comme l'un des principaux verrous dans un processus d'inspection notamment dans le cas de l'absence de l'élément à inspecter. En effet, dans ce cas, il peut y avoir des mauvais appariements avec l'élément occulté et par conséquent le diagnostic¹ obtenu sur l'élément à inspecter est erroné.

1. Résultat d'inspection : élément monté correctement ou élément mal monté

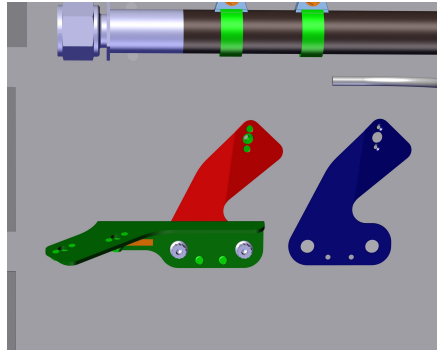
Le problème des occultations peut être géré de différentes façons :

- Dans l'espace objet : pour un élément de la scène donné, déterminer les parties de l'élément dont la vue n'est masquée par d'autres objets. Cette méthode est très coûteuse en temps de calcul. Elle utilise généralement l'algorithme du peintre [Ber93] (*Painter's algorithm* en anglais) pour résoudre le problème de visibilité d'un rendu en 3D.
- Dans l'espace image : pour chaque pixel de l'image, déterminer l'objet le plus proche de l'observateur qui est traversé par le projecteur passant par le pixel. C'est la méthode la plus populaire qui utilise soit la technique de tampon de profondeur (*z-buffer* en anglais) ou le principe de lancer de rayons (*Ray tracing* en anglais). Cependant, ces méthodes se basent sur des rendus réalistes et dépendent principalement des conditions d'illumination.

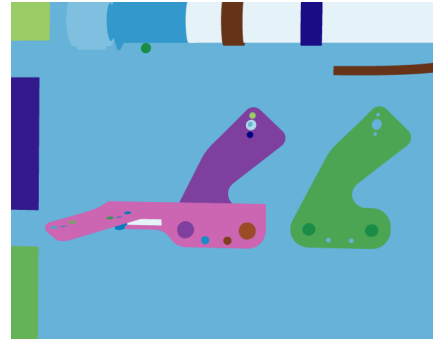
Afin de pallier le problème d'occultations partielles et d'être indépendant des conditions d'illumination, nous proposons une approche originale basée modèle qui exploite l'*indice de rendu de couleur* pour limiter ces mauvais appariements en filtrant les edgelets appariés dans la zone occultée.

Étant donné un ensemble d'éléments 3D et un point de vue fixé par la position de la caméra, pour un élément à inspecter donné, il s'agit de déterminer les parties (occultées ou masquées par d'autres éléments) susceptibles d'être éliminées. Pour ce faire, nous générons par rendu des couleurs (non-réaliste) une image de synthèse à partir du modèle CAO. Pour chaque élément du modèle CAO, nous affectons aléatoirement un indice et une couleur (voir figure 3.15b). La couleur permet de mesurer l'indice visuel de chaque élément indexé. Ensuite, nous prenons l'indice de l'élément à inspecter et nous générons un masque sur les pixels qui possèdent la même couleur affectée à l'indice. Enfin, ce masque sera utilisé comme une fenêtre de filtrage pour rejeter les edgelets appartenant à la zone occultée.

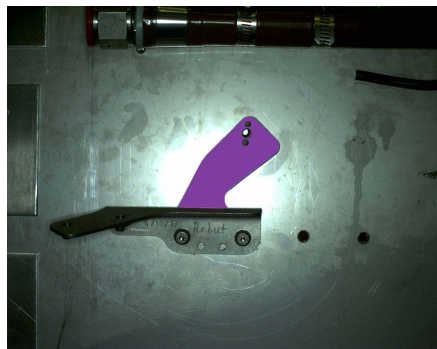
Par la suite, nous ne prendrons pas en compte les edgelets dans les zones occultées, car, par définition, ils représentent des mauvais appariements avec des contours d'autres éléments qui masquent l'élément à inspecter.



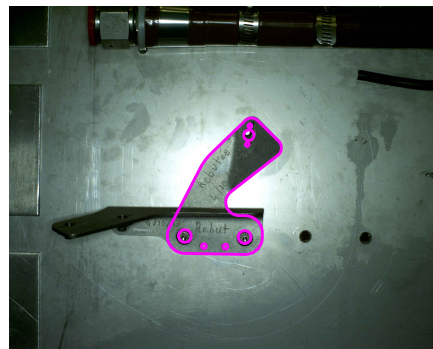
(a)



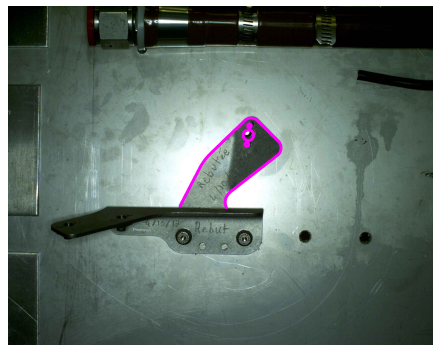
(b)



(c)



(d)



(e)

Figure 3.15 – Gestion des occultations : (a) image synthétique générée à partir du modèle CAO par une méthode de type CATIA® Composer, (b) image synthétique générée à partir du modèle CAO par la méthode d'indice de rendu de couleur, (c) masque généré par une méthode d'indice de rendu de couleur (la zone avec couleur violette représente la région à inspecter), (d) les edgelets projetés sur l'image d'inspection (e) les edgelets projetés sur la région d'intérêt spécifiée par le masque

3.6.3 Orientation des gradients

Les contours 2D correspondent à des discontinuités de la fonction d'intensité dans l'image. Généralement, les discontinuités de la fonction d'intensité correspondent aux discontinuités des propriétés géométriques (e.g. concavité, homogénéité) et photométriques (e.g. variations d'illumination, réflectance, variations dans les propriétés des matériaux des objets). Une image est souvent dégradée par des variations imprévisibles en intensité. Ces dégradations sont dues à des aberrations géométriques, photométriques, chromatiques et à des aberrations engendrées par le système d'acquisition. Un exemple de dégradation due à des aberrations géométriques (rayure sur la plaque) est illustré sur la figure 3.16.

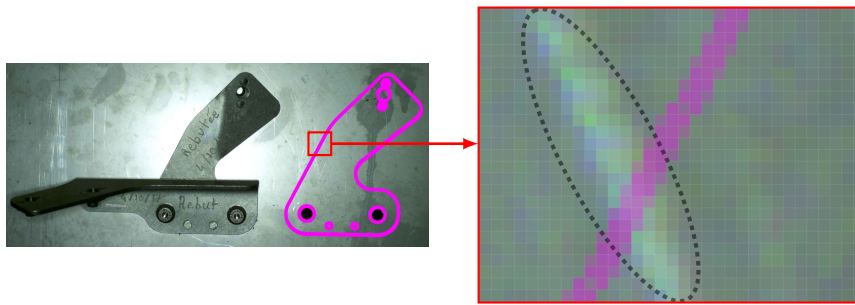


Figure 3.16 – Exemple de dégradation due à des aberrations géométriques (rayure sur la plaque). À gauche : projection des edgelets de l'élément à inspecter sur l'image 2D, à droite une zone de l'image agrandie 3200 fois

La réduction des variations imprévisibles en intensité dans l'image nécessite généralement une opération de lissage. Cependant, cette opération de lissage modifie, de façon indésirable, la structure de l'image par l'élimination et la modification de certains contours désirables.

Les variations imprévisibles en intensité dans l'image posent des problèmes au niveau de la mise en correspondance 2D/3D basée contours. En effet, dans l'étape de mise en correspondance 2D/3D entre un edgelet projeté et un point de contour 2D, il est possible d'avoir un appariement avec un point de contour 2D ayant une orientation différente. La robustesse du processus de mise en correspondance 2D/3D peut être donc améliorée en ne considérant que les edgelets projetés avec une orientation similaire à l'orientation des points de contour 2D appariés.

Pour un point de contour 2D donné, la similarité d'orientation est jugée selon le critère suivant :

- Un point de contour c_i est apparié avec un edgelet e_i ayant une orientation similaire lorsque $\theta_i < \theta_{\text{seuil}}$ (cf. figure 3.17b).
- Un point de contour c_i est apparié avec un edgelet e_i ayant une orientation différente lorsque $\theta_i > \theta_{\text{seuil}}$ (cf. figure 3.17a).

où θ_i correspond à l'angle entre l'orientation de contour c_i et celle de l'edgelet e_i défini par l'équation (3.3) et θ_{seuil} correspond au seuil de similarité d'orientation que nous avons fixé à 30° .

$$\theta_i = \arccos|\vec{n}_{e_i} \cdot \vec{d}_{c_i}| \quad (3.3)$$

où \vec{n}_{e_i} correspond à la normale à la projection dans l'image 2D de l'orientation \vec{t}_i de l'edgelet e_i et \vec{d}_{c_i} correspond à l'orientation du gradient du point de contour 2D c_i apparié avec l'edgelet e_i .

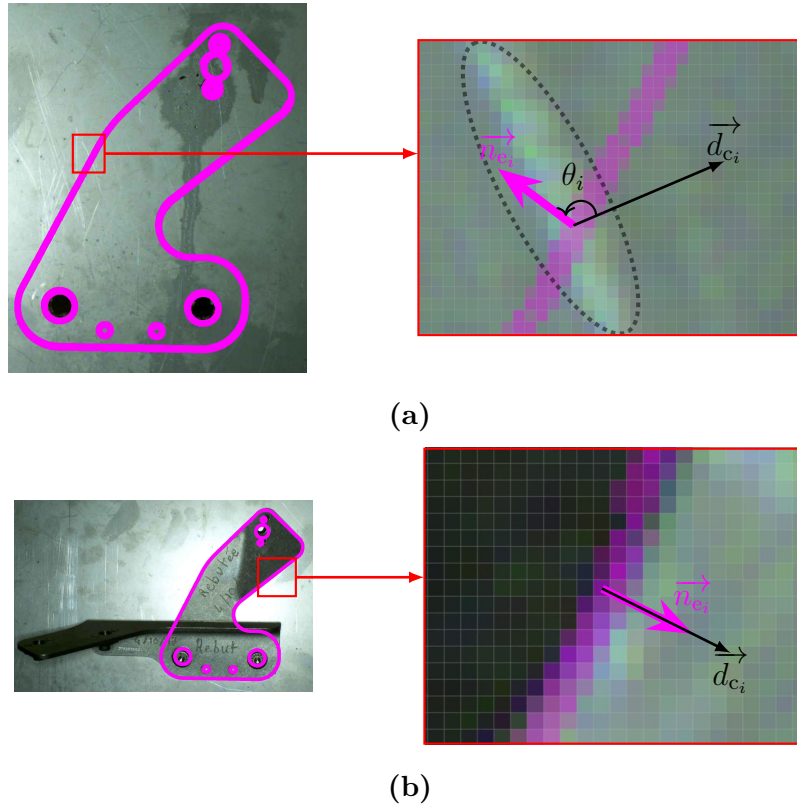


Figure 3.17 – Critère de similarité d'orientation. (1^{ère} colonne) projection des edgelets de l'élément à inspecter sur l'image 2D, (2^{ème} colonne) une zone de l'image agrandie 3200 fois. (a) un exemple de point de contour c_i apparié (par erreur) avec un edgelet e_i ayant une orientation différente ($\theta_i = 120^\circ$), (b) un exemple de point de contour c_i apparié avec un edgelet e_i ayant une orientation similaire ($\theta_i = 0^\circ$)

3.7 Prise de décision

À partir de l'analyse des résultats de mise en correspondance 2D/3D, une étape de prise de décision doit indiquer *in fine* si un défaut est présent ou non. La décision est prise à l'aide de deux indicateurs :

1. Similarité des formes entre les edgelets filtrés et les points de contour 2D appariés avec les edgelets filtrés (cf. section 3.7.1).

-
2. Taux d'appariement entre les edgelets filtrés et tous les edgelets appariés (cf. section 3.7.2).

3.7.1 Score de similarité des formes entre les edgelets filtrés et les points de contour 2D appariés

Une fois que la mise en correspondance entre les edgelets filtrés et les points de contour 2D est effectuée, l'étape suivante est la mesure de similarité qui donne le degré de ressemblance entre les edgelets filtrés et les points de contour 2D appariés.

3.7.1.1 Contexte de forme

Afin de mesurer le degré de ressemblance entre les edgelets filtrés et les points de contour 2D appariés, nous avons choisi d'exploiter un descripteur basé sur le contexte de forme (*shape context* en anglais) décrit par BELONGIE et al. [Bel+00]. Nous l'appliquons dans nos travaux à des données de contour. Ce descripteur a été utilisé sur des données non segmentées (totalité de l'image) pour des tâches de détection d'objet [Bel+02], ou encore sur les squelettes de mots pour la reconnaissance optique de caractères (*Optical Character Recognition (OCR)* en anglais) [Pil12].

Du fait de la polyvalence et de la richesse de ce descripteur, nous l'avons adopté pour mesurer la similarité entre les edgelets filtrés et les points de contour 2D appariés.

3.7.1.2 Mesure de similarité (calcul de distance basé sur le contexte de forme)

En général, la mesure de similarité est jugée par le calcul d'une distance (l'inverse de la similarité) entre les caractéristiques extraites de deux formes que ce soit sur la totalité des pixels de l'objet, ou seulement sur la frontière de l'objet (contour) [Mer17]. Dans notre cas, nous nous intéressons à la mesure de similarité définie à partir des informations extraites seulement de la frontière de l'objet (similarité basée contour).

Une mesure de similarité assigne une distance importante aux formes non similaires. Autrement dit, plus la mesure de similarité est faible plus les formes sont similaires. Par exemple, la figure 3.18 montre tous les edgelets projetés (noir), les edgelets filtrés (rouge) et les points de contour 2D appariés (vert).

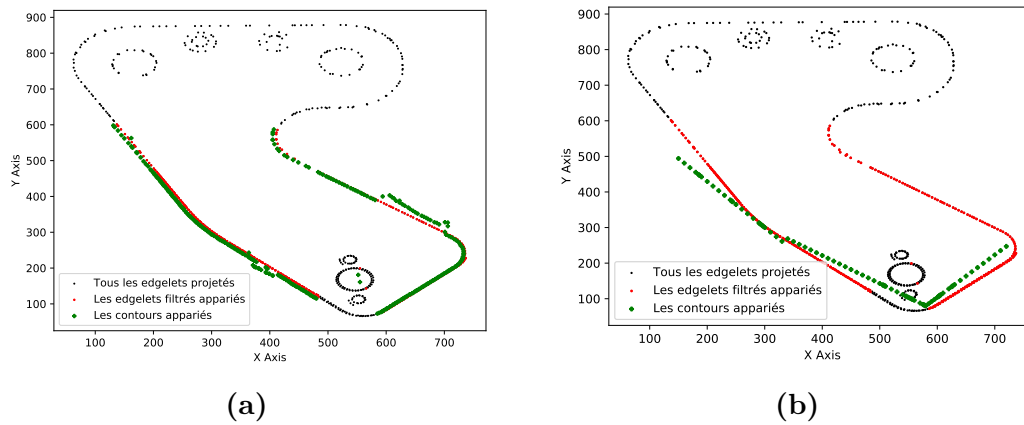


Figure 3.18 – Similarité des formes entre les edgelets filtrés et les contours détectés à partir de l'image : (a) exemple de deux formes jugées comme similaires et (b) exemple de deux formes jugées comme différentes

3.7.2 Taux d'appariement entre les edgelets filtrés et les points de contour 2D appariés

3.7.2.1 Pondération d'appariement

Lorsque les edgelets sont projetés sur l'image 2D, il peut résulter une erreur de reprojektion e_r . Cette erreur permet de mesurer la qualité de cette projection. Afin de prendre en compte ce type d'erreur, le processus de prise de décision requiert l'utilisation d'une pondération.

Cette pondération prend en compte le degré d'appariement des edgelets projetés avec les points de contour 2D et permet d'avoir une fonction de coût avec des mesures hétérogènes.

Chaque appariement (couple edgelet-contour) est pondéré en fonction de la distance entre l'edgelet et le point de contour 2D appariés (cf. figure 3.19). La pondération d'appariement est établie suivant une dispersion Gaussienne 2D définie par un paramètre : la distance entre edgelet et contour 2D appariés. La figure 3.19 montre l'allure de cette fonction de pondération. Typiquement, la longueur de la ligne de recherche l_i est la distance maximale d'appariement entre edgelet et point de contour 2D. Plus la distance entre edgelet et contour 2D appariés est faible (l'edgelet est proche du contour apparié) plus la qualité d'appariement est bonne et plus le facteur de pondération (poids w) est faible.

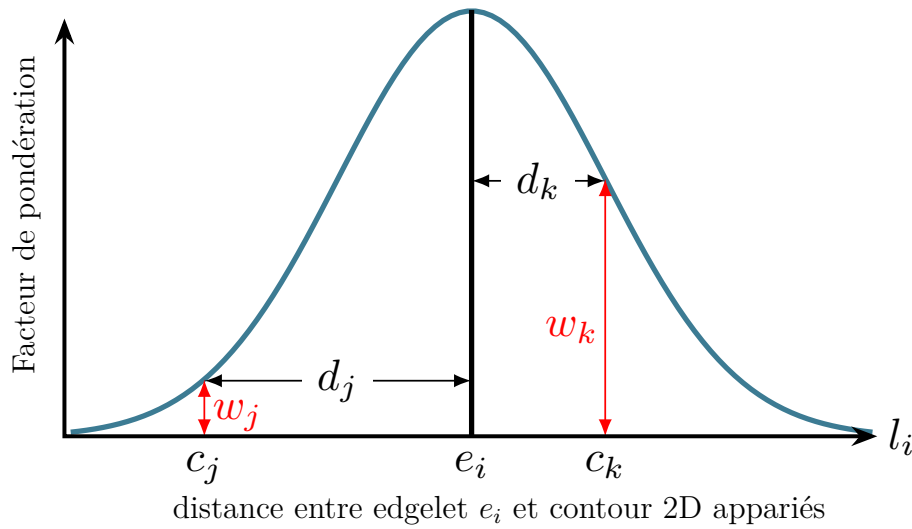


Figure 3.19 – Gaussienne de pondération de la distance entre un edgelet e_i et deux exemples de contours 2D appariés (c_j , c_k). Typiquement, plus l’edgelet est proche du contour apparié et plus le facteur de pondération (poids w) est faible et inversement

3.7.2.2 Taux d’appariement

Le taux d’appariement représente le taux d’edgelets mis en correspondance avec des contours dans l’image. Ce taux est présenté en pourcentage selon le nombre d’edgelets appariés correctement avec des contours 2D.

Nous avons trouvé expérimentalement (cf. section 3.9.5) que, dans le cas d’un élément présent, le taux moyen d’appariement est de l’ordre de 90% contre 6% dans le cas d’un élément absent et 10% dans le cas d’un élément monté à l’envers. Le taux moyen d’appariement est plus significatif lorsque l’objet est sur un fond uniforme. Un risque de mauvais appariement est cependant possible lorsque plusieurs contours sont localement proches dans l’image 2D.

3.7.3 Processus de décision

La décision finale est prise à l’aide de deux indicateurs : la similarité des formes entre les edgelets filtrés et les points de contour 2D appariés ($f_{\text{similarité}}$) et le taux d’appariement entre les edgelets filtrés et les points de contour 2D appariés ($\text{taux}_{\text{appariement}}$).

Les étapes du processus de décision sont illustrées sur la figure 3.20.

- Si $f_{\text{similarité}} < \text{seuil}_{\text{similarité}}$, l’élément est considéré comme non conforme (absent ou monté à l’envers).

Dans nos expériences, nous avons fixé le seuil de similarité ($\text{seuil}_{\text{similarité}}$) à 0.7, sur la base d’une comparaison approfondie entre différentes formes.

- Si $f_{\text{similarité}} > \text{seuil}_{\text{similarité}}$, nous calculons le rapport entre tous les edgelets appariés et les edgelets filtrés ($\text{taux}_{\text{appariement}}$).
- Si $\text{taux}_{\text{appariement}} > \text{seuil}_{\text{appariement}}$, l'élément est considéré comme conforme (présent). Sinon, l'élément est considéré comme non conforme.

Le seuil du taux d'appariement $\text{seuil}_{\text{appariement}}$ a été déterminé expérimentalement (voir section 3.9).

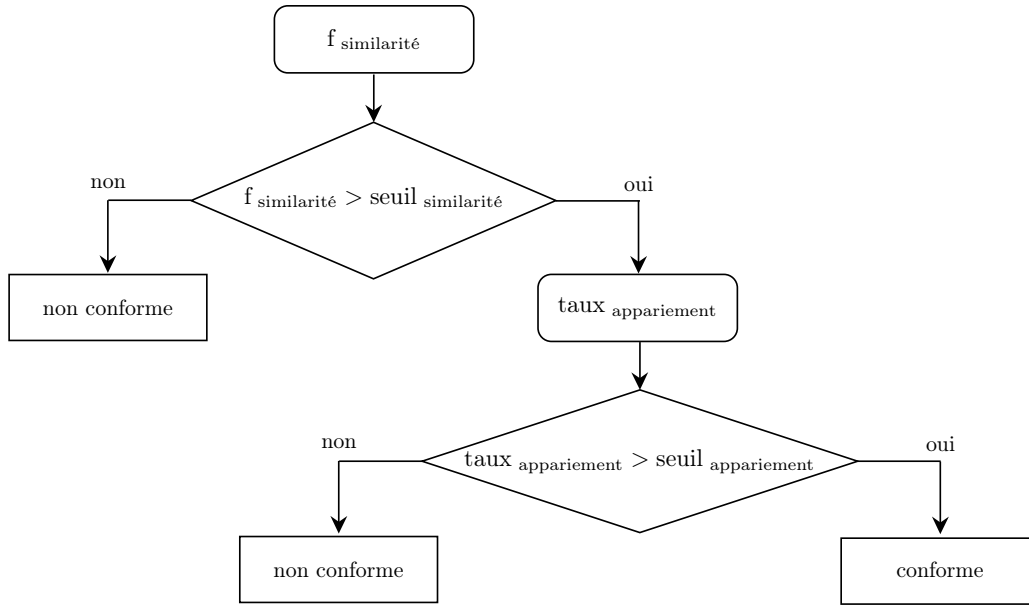


Figure 3.20 – Processus de décision

3.8 Choix de points de vue dans l'étape de préparation de l'inspection 2D

Comme nous l'avons vu dans le chapitre 2, le choix des meilleurs points de vue, déterminé à partir de la maquette numérique exploite une fonction score f_{score} .

Comme nous l'avons vu dans ce chapitre, nous avons mis en place un processus d'élimination des contours parasites dans l'étape d'inspection qui permet, à partir de la maquette numérique, de définir une fonction f_{parasite} définie par :

$$f_{\text{parasite}} = \frac{\text{Nombre d'edgelets non appariés avec des contours parasites}}{\text{Nombre total d'edgelets}} \quad (3.4)$$

Ce critère est maximal quand le nombre de contours parasites est très faible. Il a été utilisé dans l'étape en ligne d'inspection mais il peut aussi être utilisé dans l'étape hors ligne de choix de points de vue.

Pa ailleurs, nous avons également défini dans ce chapitre une méthode qui permet de maximiser les bons appariements en exploitant la similarité des formes entre les edgelets filtrés et les points de contour 2D appariés. Cette méthode s'appuie sur un critère noté $f_{\text{similarité}}$ qui est maximal lorsque les formes sont similaires. Dans l'étape en ligne d'inspection ce critère est calculé à partir des images 2D réelles mais ce critère peut aussi être calculé à partir d'images synthétiques générées à partir de la maquette numérique. Ce critère peut donc être utilisé dans l'étape hors ligne de choix de points de vue.

Par conséquent, pour sélectionner les meilleurs points de vue préalablement à la mise en œuvre de l'inspection 2D, nous avons choisi une fonction score f_{score} qui exploite les trois critères $f_{\text{visibilité}}$ (défini dans la section 2.6.1), f_{parasite} et $f_{\text{similarité}}$, suivant l'équation 2.5.

3.9 Résultats expérimentaux

Dans l'objectif d'évaluer les performances de l'approche présentée dans ce chapitre, nous avons procédé à des expérimentations et évaluations exhaustives sur des images issues de bases de données variées, constituées par DIOTA, provenant de différents assemblages mécaniques dans les deux modes : inspection robotisée et inspection via une tablette portée par un opérateur. Nous présentons d'abord les conditions expérimentales dans lesquelles les expérimentations ont été réalisées (cf. section 3.9.1 et section 3.9.2). Ensuite, nous présentons les entrées principales de notre approche. Nous introduisons dans un premier temps les données utilisées, notamment les modèles CAO 3D des assemblages mécaniques et les images testées dans les deux configurations, inspection via une tablette ou inspection robotisée (cf. section 3.9.3).

Nous présentons ensuite l'impact de la détection de contours sur l'étape de mise en correspondance 2D/3D à travers une comparaison qualitative entre les résultats de l'algorithme de détection de contours de Canny et les résultats de détection de contours à base de réseaux de neurones (cf. section 3.9.4). Puis, nous détaillons la détermination quantitative des paramètres optimaux de l'étape de prise de décision (cf. section 3.9.5). Enfin, nous présentons les résultats d'inspection obtenus sur plusieurs jeux de données dans les deux configurations, inspection via une tablette ou inspection robotisée (cf. section 3.9.6 et section 3.9.7).

3.9.1 Conditions expérimentales

L'implémentation de l'algorithme est effectuée en C++ et C# en utilisant la bibliothèque *Open Computer Vision (OpenCV)* et la bibliothèque *Open Graphics Library (OpenGL)*. La méthode d'inspection a été évaluée dans les deux modes : inspection robotisée et inspection via une tablette portée par un opérateur, en utilisant une machine ayant un processeur Quad-core i7-6700 cadencé à 2.6 Ghz et équipé de 16 Go de RAM pour la solution robotisée et une tablette ayant un processeur Intel Core mobiles Dual-core i7-7660 cadencé à 2.5 Ghz et équipé de 4 Go de RAM pour la solution utilisant une tablette.

Dans les expériences, les images ont été acquises dans les deux modes (robotisé et manuel) en utilisant une caméra UI-3180 (PYTHON 5000). Les images sont en RGB avec une résolution de 2592×2048 pixels avec une focale de 25 mm pour le mode robot et une résolution de 1920×1080 pixels avec une focale de 4.5 mm pour le mode tablette (pour plus de détails, voir la section 1.5).

Les images de la solution robotisée sont de meilleure résolution que les images de la solution exploitant une tablette car avec la solution tablette nous recherchons une inspection en temps réel.

3.9.2 Éclairage

L'éclairage est un élément important de tout système d'inspection automatisé. L'évaluation que nous proposons est réalisée en environnement industriel.

Lorsqu'on prend en compte les différentes fonctions du capteur (une caméra pour l'inspection et une caméra pour la localisation), il n'est pas nécessaire de positionner un éclairage global pour le capteur mais le plus important est de se concentrer sur la caméra d'inspection 2D qui a besoin dans certains contextes de pouvoir disposer de lumière additionnelle et de la contrôler. Par conséquent, nous avons ajouté une solution d'éclairage autour de la caméra d'inspection (voir figure 3.21). Cette source d'éclairage n'est activée qu'au moment de l'acquisition de l'image, lorsque le robot est en mode d'acquisition (le robot est en position fixe et ne bouge pas).

Pour la caméra de localisation, la vue de l'assemblage en champ large permet de capturer à la fois la pièce mais également son environnement. Il n'est pas nécessaire d'apporter de la lumière pour que les algorithmes fonctionnent et une lumière qui n'est pas statique dans la scène pourrait créer des erreurs et/ou perturber la localisation. Nous avons donc décidé de ne pas ajouter d'éclairage pour cette fonction mais, si nécessaire, de nous appuyer sur l'éclairage naturel ambiant pour apporter une lumière fixe constante.

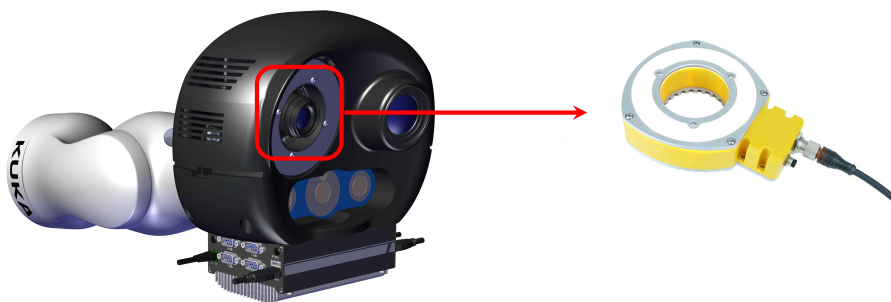


Figure 3.21 – Système d'éclairage autour de la caméra d'inspection

Pour la solution exploitant une tablette portée par un opérateur, nous utilisons une seule caméra qui permet à la fois la localisation et l'inspection de l'élément à contrôler. Dans ce cas, nous ne nous appuyons que sur la lumière ambiante provenant

du contexte de l'usine. En effet, nous travaillons toujours à l'intérieur de l'usine, ce qui nous permet d'avoir un environnement stable et des conditions d'éclairage stables (conditions normales de luminosité).

3.9.3 Présentation des données

Cette section présente les entrées principales de notre approche. Nous introduisons dans un premier temps les données utilisées au cours de nos travaux notamment les modèles CAO 3D des assemblages mécaniques et les images acquises dans les deux configurations, inspection via une tablette et inspection robotisée.

3.9.3.1 Données utilisées pour la solution robotisée

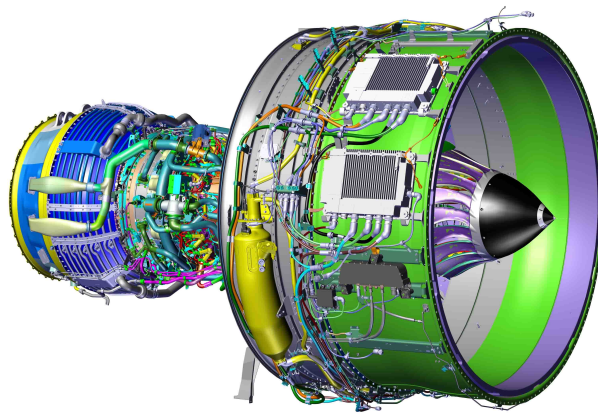
Assemblages mécaniques utilisés : le modèle CAO de l'assemblage mécanique, duquel on peut extraire le modèle CAO de l'élément à contrôler est disponible et il sert de référence. La figure 3.22 montre trois assemblages mécaniques différents utilisés dans les résultats expérimentaux :

- Moteur d'avion (figure 3.22a) : un moteur d'avion LEAP (Leading Edge Aviation Propulsion) de la famille CFM56, qui se compose de plusieurs dizaines de milliers d'éléments à contrôler.
- Panneau 1 (figure 3.22b) : utilisé pour tester nos algorithmes, qui contient 6 éléments à contrôler.
- Panneau 2 (figure 3.22c) : utilisé pour tester nos algorithmes, qui contient 4 éléments à contrôler.

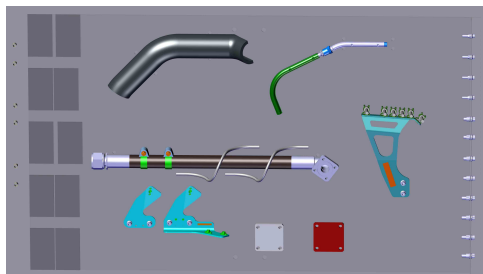
Remarque : l'intérêt d'utiliser les panneaux de test Panneau 1 et Panneau 2 est de pouvoir modifier l'agencement de leurs éléments ce qui permet ainsi de créer un grand nombre de configurations possibles.

Images acquises : la plateforme robotique porte deux caméras. Une caméra de localisation à champ large pour assurer la localisation précise de l'effecteur par rapport à la pièce à contrôler et une caméra haute résolution avec un champ de vue réduit pour capturer les détails et observer très finement les éléments à inspecter. Conformément aux normes de sécurité industrielle, pour assurer la sécurité de l'élément inspecté ainsi que celle du robot d'inspection, la distance entre la caméra et l'élément à contrôler est fixée à 600 mm avec une tolérance de ± 6 mm.

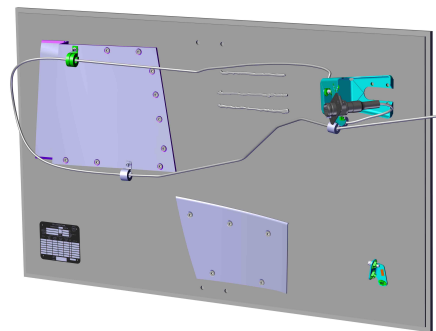
Dans le contexte robotisé, les images utilisées dans les résultats expérimentaux qui suivent sont acquises avec une résolution de 2592×2048 pixels. La figure 3.23 montre les données acquises par le robot sur le moteur LEAP (voir la figure 3.22a) et la figure 3.24 montre les données acquises par le robot sur le Panneau 1 (figure 3.22b) et le Panneau 2 (figure 3.22c).



(a)



(b)



(c)

Figure 3.22 – Assemblages mécaniques utilisés dans le mode robotisé : (a) moteur LEAP de la famille CFM56, (b) et (c) deux panneaux de test différents constitués d'éléments à contrôler

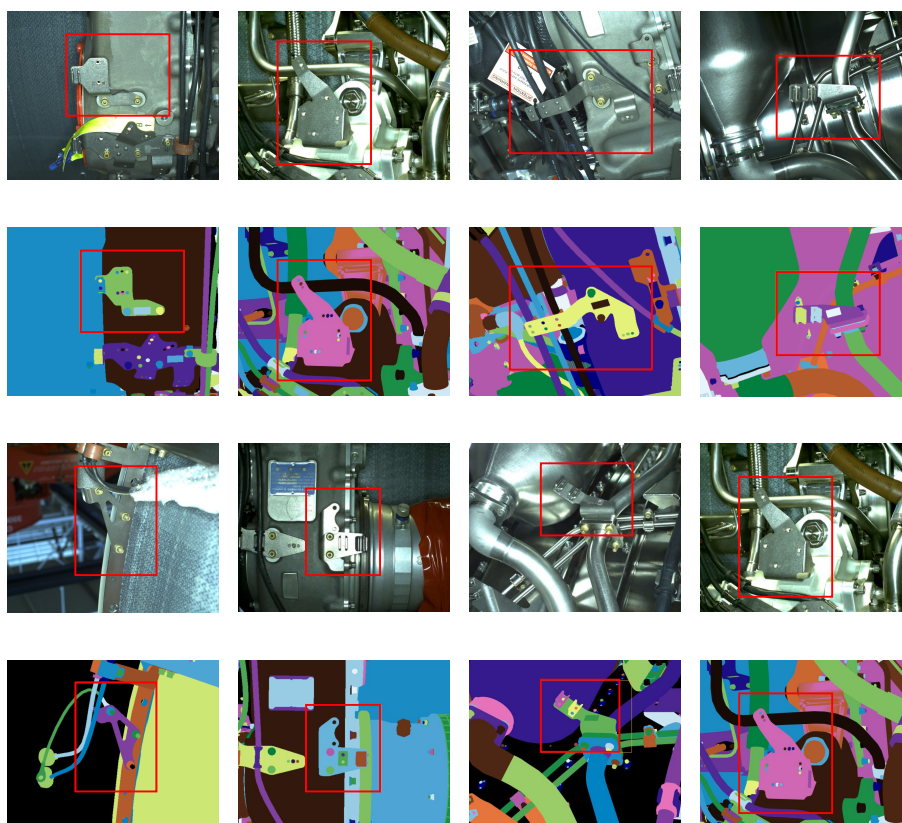


Figure 3.23 – Exemples de données acquises par le robot sur le moteur (figure 3.22a) utilisées pour évaluer notre approche dans le contexte robotisé

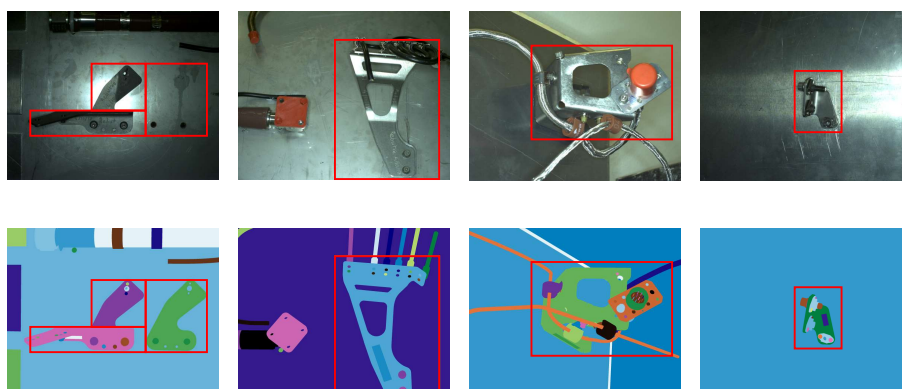


Figure 3.24 – Exemples de données acquises par le robot sur les deux panneaux de test : Panneau 1 (figure 3.22b) et Panneau 2 (figure 3.22c), utilisées pour évaluer notre approche dans le contexte robotisé

3.9.3.2 Données utilisées pour la solution exploitant une tablette

Assemblages mécaniques utilisés : l'utilisation d'une tablette est privilégiée dans l'inspection d'assemblages mécaniques comportant des éléments dispersés.

Inversement, le mode robotisé est favorisé dans l'inspection d'assemblages mécaniques composés d'un grand nombre d'éléments difficilement accessibles (e.g. moteur d'avion qui se compose de plusieurs dizaines de milliers d'éléments à contrôler).

La figure 3.25 montre trois assemblages mécaniques différents utilisés dans les résultats expérimentaux :

- Panneau 3 (figure 3.25a) : utilisé pour tester nos algorithmes, qui contient 9 éléments à contrôler.
- Moteur d'avion XWB (figure 3.25b) : une partie d'assemblage de l'avion Airbus A350 XWB, qui se compose de plusieurs dizaines d'éléments à contrôler.
- Moteur de l'avion de combat 34 Rafale (figure 3.25c) : une partie d'assemblage de l'avion 34 Rafale, qui se compose de plusieurs dizaines d'éléments à contrôler.

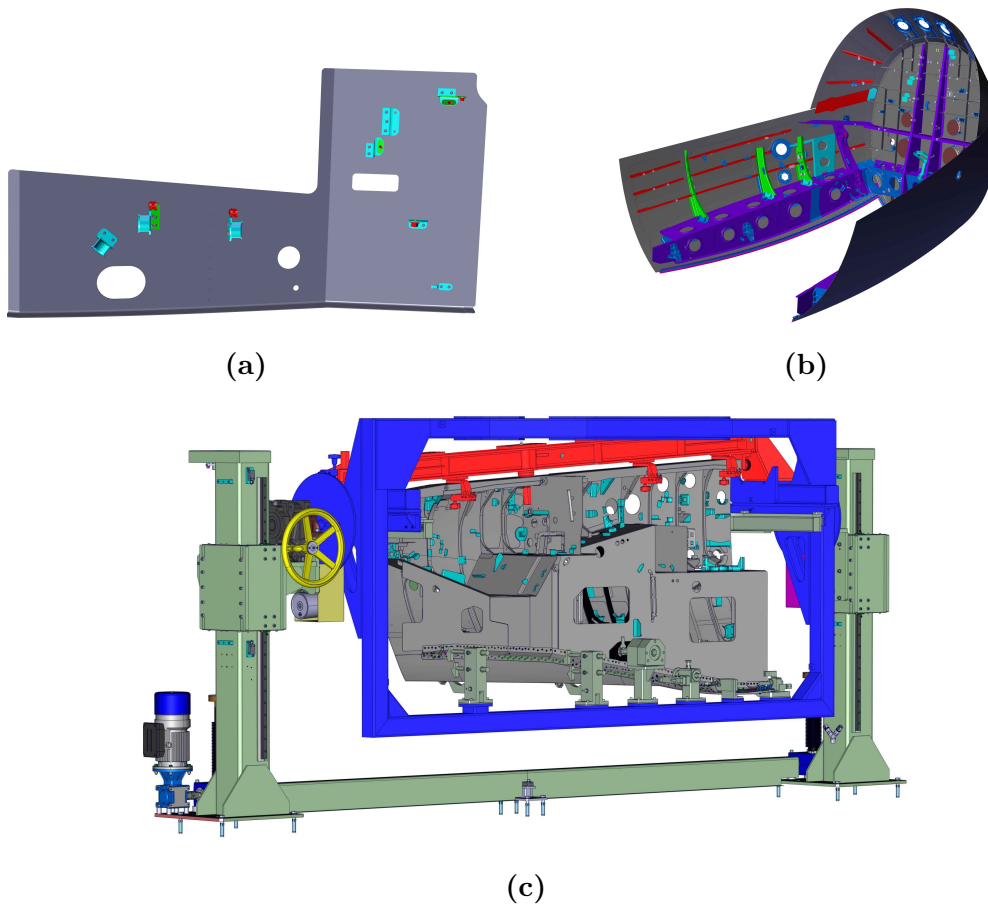


Figure 3.25 – Assemblages mécaniques utilisés dans le mode manuel : (a) un panneau test constitué d'éléments à contrôler, (b) une partie d'assemblage de l'avion Airbus A350 XWB et (c) une partie d'assemblage de l'avion 34 Rafale

Images acquises : Dans le contexte tablette, des séquences d'images (flux vidéo) sont acquises avec une résolution de 1920×1080 pixels en utilisant une seule caméra montée sur une tablette. Le flot des données (images + poses) est ensuite analysé en temps réel pour statuer sur l'état de conformité des éléments.

3.9.4 Détection de contours dans l'image

Dans un premier temps, pour détecter les contours dans l'image, nous avons utilisé le filtre de Canny [Can86]. Cette méthode reste à ce jour la méthode la plus connue pour la détection de contours dans une image. Le filtre de Canny se base sur l'orientation du gradient dans une image et il utilise le seuillage par hystérésis qui nécessite deux seuils, un seuil bas T_b et un seuil haut T_h . Le réglage de ces seuils peut être délicat et résulte toujours d'un compromis entre sous-détection et sur-détection.

- Seuil bas : contours appropriés détectés, mais beaucoup de bruit, même en présence d'un filtre de lissage (voir figure 3.26b). Autrement dit, un seuil très bas peut conduire à la détection de faux positifs.
- Seuil haut : pas de bruit, mais perte de certains contours (voir figure 3.26f). Autrement dit, un seuil très haut peut empêcher la détection de contours pertinents.

Un exemple de résultat de l'application du filtre de Canny avec différents seuils est visible sur la figure 3.26 et le tableau 3.1.

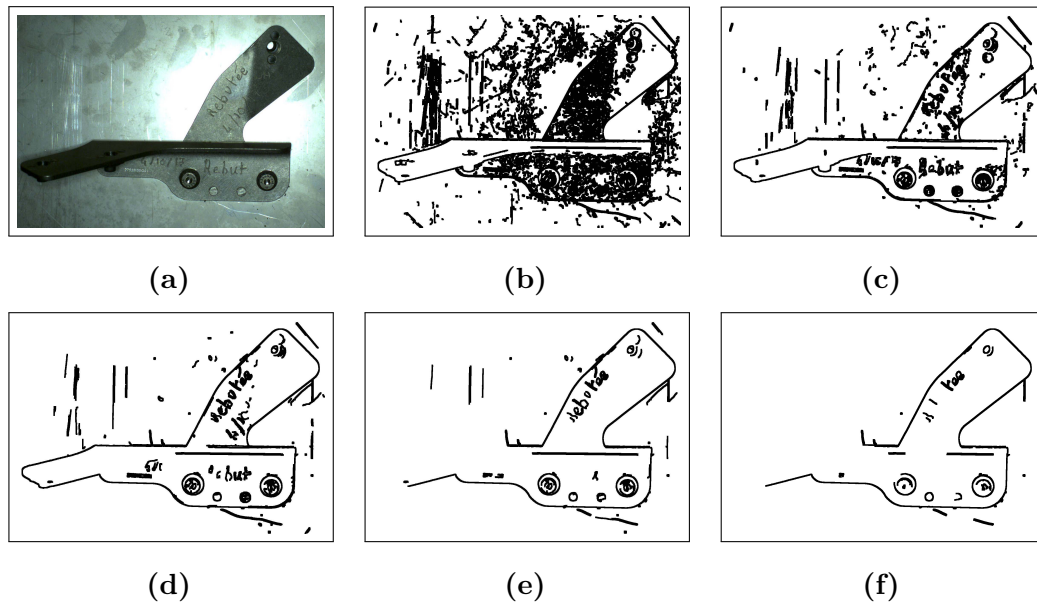


Figure 3.26 – Résultats de l'application du filtre de Canny avec différents seuils détaillés dans le tableau 3.1

Table 3.1 – Les différents paramètres de filtre de Canny utilisés dans les résultats de la figure 3.26

Figure	3.26b	3.26c	3.26d	3.26e	3.26f
(T_b, T_h)	(10, 30)	(20, 60)	(30, 90)	(40, 120)	(50, 150)

Remarques :

1. Selon les recommandations de CANNY [Can86] le seuil haut doit être égal à 3 fois le seuil bas ($T_h = 3 \times T_b$).
2. L'étape de mise en correspondance est réalisée en projetant dans l'image contour les edgelets extraits de modèle CAO et en recherchant localement le long de lignes de recherche le point de contour 2D le plus proche de l'edgelet projeté. Par conséquent, nous ne nous intéressons qu'aux contours 2D localement proches de l'élément à inspecter. La contrainte d'orientation du gradient présentée dans la section 3.6.3 permet de limiter l'effet des contours aberrants (bruit) sur l'étape de mise en correspondance 2D/3D.

En utilisant le filtre de Canny pour extraire les contours dans les images, les résultats préliminaires de l'évaluation de la méthode d'inspection sur le Panneau 1 (figure 3.22b) et le Panneau 2 (figure 3.22b) sont très satisfaisants. En revanche, sur le moteur (figure 3.22a) les résultats ne sont pas satisfaisants notamment sur les surfaces brillantes ou texturées. Une analyse approfondie de ce problème nous a amené à conclure que la source d'erreur principale vient soit de sous-détection ou de sur-détection de contours. En effet, il n'existe pas actuellement de méthode générique pour déterminer des seuils produisant des résultats satisfaisants sur tous les types d'images. Pour cette raison, les résultats sont très satisfaisants sur les panneaux de test Panneau 1 et Panneau 2 et non satisfaisants sur le moteur.

Le détecteur de contours de Canny ne se concentre que sur les changements locaux de propriétés de l'image. Pour cette raison, le détecteur de Canny n'a aucune compréhension sémantique², et sa précision est limitée (ce qui est très souvent le cas).

Sur la figure 3.27, on peut voir deux cas où le détecteur de contours de Canny échoue car il n'a aucune compréhension du contenu de l'image, notamment sur les surfaces brillantes (e.g. aluminium, acier, etc.) (voir figure 3.27a, de nombreuses détections manquées) et sur les surfaces texturées (voir figure 3.27c, beaucoup de fausses détections). De plus, dans le cas où l'image comporte une zone de luminance, les éléments sont peu contrastés par rapport au fond et la détection de contours par la méthode de Canny est compromise.

2. Compréhension du contenu de l'image

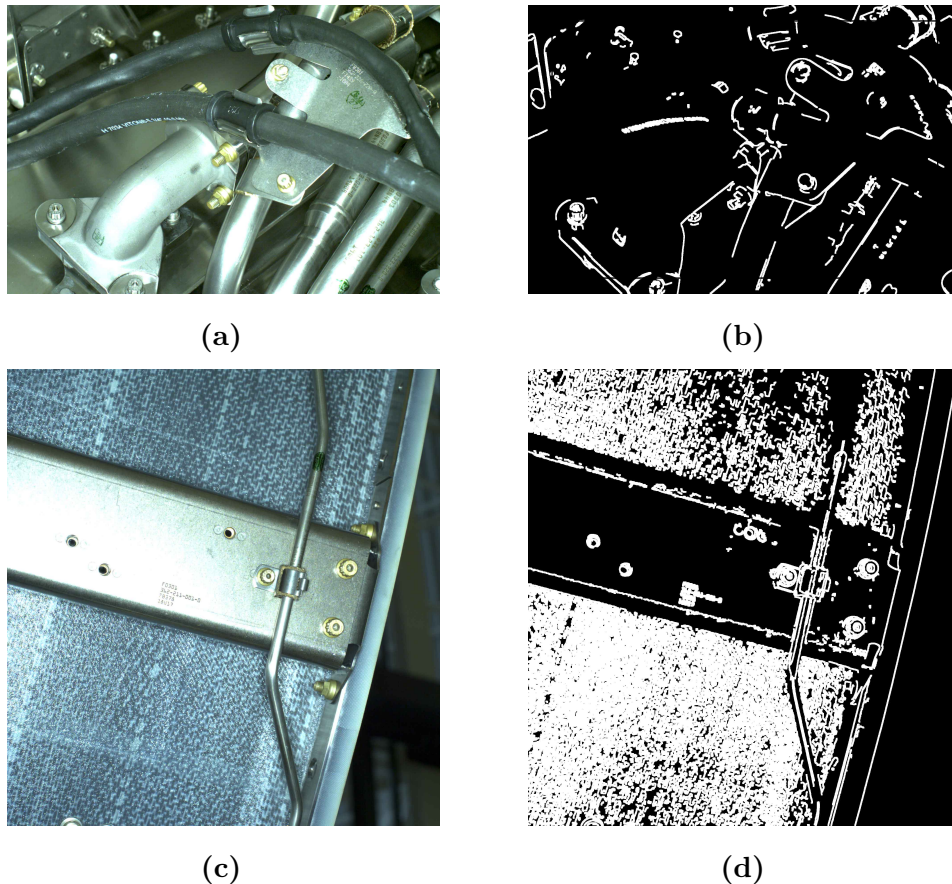


Figure 3.27 – Problèmes rencontrés avec le détecteur de contours de Canny. Le détecteur de contour de Canny échoue : (a) et (b) dans le cas de surfaces brillantes, de nombreuses détections manquées, et (c) et (d) dans le cas de surfaces texturées, beaucoup de fausses détections

Outre la méthode de Canny, certaines nouvelles méthodes ont été proposées pour la détection de contours à base d'apprentissage profond (Deep Learning). Très récemment, XIE et al. [Xie+17] et CHEN et al. [Che+16] ont proposé un réseau de neurones artificiels entièrement convolutionnel (en anglais *CNN* ou *Convolutional Neural Networks*) capable de détecter beaucoup mieux les contours, et ce, sans avoir à manuellement ajuster ses paramètres. Afin d'apprendre à classifier quels pixels de l'image correspondent à des contours, ces méthodes utilisent les forêts d'arbres décisionnels³ (*random forest classifier* en anglais) [Bre01].

La figure 3.28 illustre qualitativement les résultats de l'approche de détection de contours à base de réseaux de neurones. Dans cette partie expérimentale, nous avons utilisé ce détecteur pour sa résistance au bruit notamment sur les surfaces brillantes et sur les surfaces texturées. Il pourrait être intéressant d'investiguer d'autres algorithmes d'extraction de contours.

3. Elles font partie des technique d'apprentissage automatique qui permettent d'obtenir des modèles prédictifs pour la classification et la régression

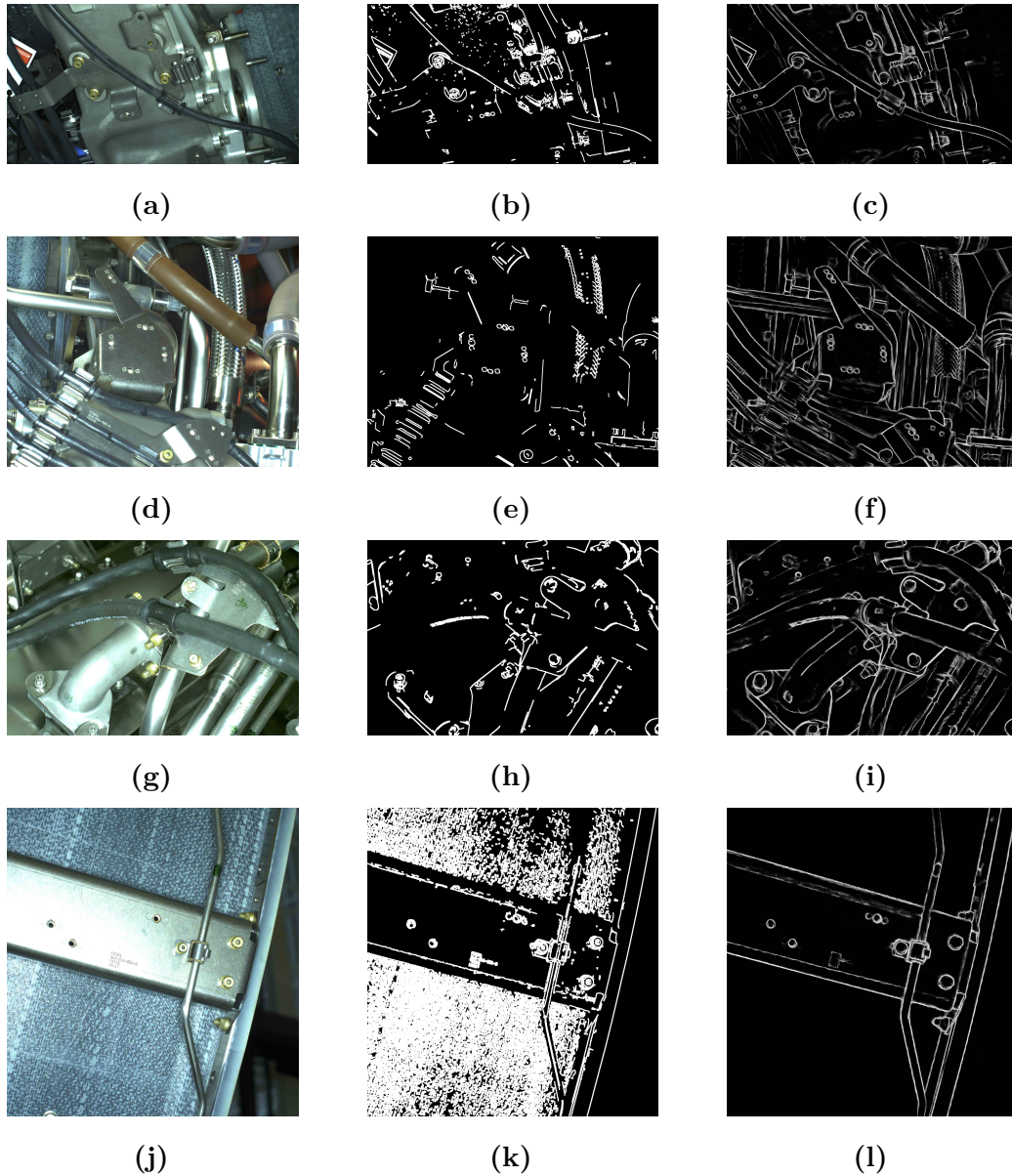


Figure 3.28 – Comparaison qualitative de l'algorithme de Canny et de l'algorithme de détection de contours à base de réseaux de neurones. (1^{ère} colonne) image initiale, (2^{ème} colonne) carte de contours calculée par Canny, (3^{ème} colonne) carte de contours calculée par l'algorithme de détection de contours à base de réseaux de neurones. Nous constatons que l'algorithme de détection de contours à base de réseaux de neurones est capable de détecter beaucoup mieux les contours sur les surfaces brillantes et sur les surfaces texturées, avec une plus grande robustesse par rapport au bruit que l'algorithme de Canny

Les détecteurs qui utilisent le Machine Learning ou le Deep Learning produisent de meilleurs résultats que les détecteurs de contours traditionnels dans notre cas. De plus, l'avantage majeur des méthodes par Machine Learning est qu'elles ne requièrent pas la calibration de paramètres.

Dans les résultats que nous présentons par la suite, l'extraction de contours a été réalisée avec le détecteur de Canny dans le scénario tablette et par la technique de Machine Learning dans le scénario robot.

3.9.5 Détermination des paramètres optimaux

3.9.5.1 Détermination du seuil d'appariement

Le seuil d'appariement ($\text{seuil}_{\text{appariement}}$), défini dans la section 3.7.3, joue un rôle important dans le processus de décision. Par conséquent, il nécessite une attention particulière.

Un exemple qui illustre le processus de détermination du seuil d'appariement sur le Panneau 1 est présenté en figure 3.29, et les résultats quantitatifs correspondants sont présentés dans le tableau 3.2. Ce type d'expérience nous a permis de déterminer la valeur des seuils d'appariement à fixer pour le cas conforme et pour le cas non conforme (voir plus loin).

Table 3.2 – Résultats quantitatifs du taux d'appariement de l'expérience 1 (figure 3.29)

Élément	Statut	Taux d'appariement [min, max]	Taux moyen
A	présent	[69 %, 100 %]	90.1 %
B	monté à l'envers	[0 %, 24 %]	9.02 %
C	absent	[0 %, 22 %]	5.62 %

Le type d'expérience décrit sur la figure 3.29 a été mené plusieurs fois sur une base de données contenant les 3 panneaux de test (avec des configurations différentes). Une analyse des résultats de ces expériences nous a permis de considérer qu'avec un seuil d'appariement fixe égal à 70 % pour le cas conforme et 25 % pour le cas non conforme, une décision correcte sur le statut des éléments est atteinte.

Nous avons donc décidé que :

- Si $\text{taux}_{\text{appariement}} \geq 70\%$, l'élément est considéré comme conforme (présent).
- Si $\text{taux}_{\text{appariement}} \leq 25\%$, l'élément est considéré comme non conforme (absent ou monté à l'envers).
- si $25\% < \text{taux}_{\text{appariement}} < 70\%$, l'élément est considéré comme *unknown*. Situation que nous n'avons jamais rencontrée dans nos expériences.

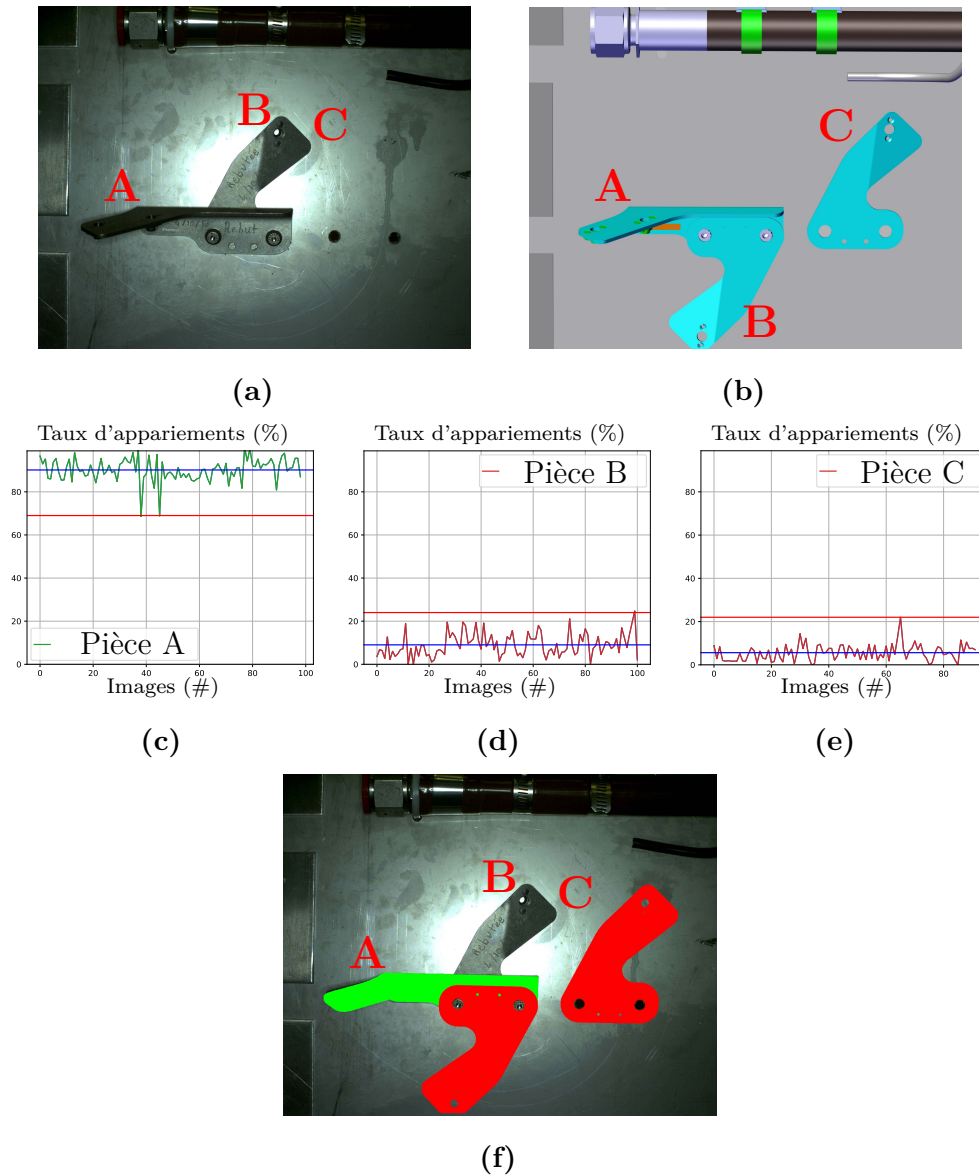


Figure 3.29 – Détermination du seuil d'appariement ($\text{seuil}_{\text{appariement}}$) : (a) Panneau 1 constitué d'éléments à contrôler, (b) modèle CAO du Panneau 1, (c-e) pour les 3 éléments (l'élément A a été monté correctement, l'élément B a été monté à l'envers et l'élément C est absent) : la courbe représente le taux d'appariement calculé à partir de 100 images acquises avec des points de vue différents. La ligne bleu représente le taux moyen d'appariement. La ligne rouge représente le seuil d'appariement qui a été calculé pour cette expérience, en prenant le minimum ou le maximum de la courbe, et (f) résultats d'inspection (vert : élément monté correctement et rouge : éléments mal montés)

3.9.6 Métriques d'évaluation pour l'analyse de performances

L'évaluation d'un algorithme d'inspection consiste à confronter les statuts retournés par l'algorithme et les statuts fournis par la vérité terrain. Pour ce faire, l'outil privilégié est la matrice de confusion. Plusieurs indicateurs de succès résumant les performances de l'algorithme en sont déduits.

3.9.6.1 Matrice de confusion

Lorsque l'algorithme retourne le statut de l'élément à inspecter, deux cas peuvent se présenter : l'élément est défectueux ou non défectueux **selon lui**. En face de ces deux choix de l'algorithme, nous avons les deux cas où l'élément est défectueux ou non défectueux **selon la vérité terrain**. Par conséquent, cela nous donne alors 4 cas possibles différents :

- Vrai Positif (VP) : l'élément est défectueux et l'algorithme trouve que l'élément est défectueux.
- Faux Positif (FP) : l'élément est non défectueux et l'algorithme trouve que l'élément est défectueux (fausse alarme).
- Vrai Négatif (VN) : l'élément est non défectueux et l'algorithme trouve que l'élément est non défectueux.
- Faux Négatif (FN) : l'élément est défectueux et l'algorithme trouve que l'élément est non défectueux (situation critique).

Ces notions sont résumées dans le tableau 3.3 de la matrice de confusion.

Table 3.3 – Matrice de confusion

Vérité terrain → Algorithme ↓	Non-défectueux	Défectueux
Non-défectueux	VN	FN
Défectueux	FP	VP

3.9.6.2 Indicateurs de succès

La mise en œuvre des indicateurs de succès a pour objectif d'évaluer la performance de notre système.

Ces indicateurs se basent sur les notions décrites dans la matrice de confusion (tableau 3.3). Les définitions et les formules correspondantes sont données ci-après :

- Précision (définie par l'équation (3.5)) : proportion de prédictions correctes parmi les prédictions positives (i.e. élément détecté comme défectueux)

$$\begin{aligned} \text{précision} &= \frac{\text{nbr d'éléments correctement attribués à la classe « défectueux »}}{\text{nbr d'éléments attribués à la classe « défectueux »}} \\ &= \frac{VP}{VP + FP} \in [0\%, 100\%] \end{aligned} \quad (3.5)$$

La précision mesure la capacité de l'algorithme à éviter les fausses alarmes.
Un système ne produisant aucun FP a une précision de 100%.

- Rappel (défini par l'équation (3.6)) : proportion de défauts correctement identifiés

$$\begin{aligned} \text{rappel} &= \frac{\text{nbr d'éléments correctement attribués à la classe « défectueux »}}{\text{nbr d'éléments appartenant à la classe « défectueux »}} \\ &= \frac{VP}{VP + FN} \in [0\%, 100\%] \end{aligned} \quad (3.6)$$

Le rappel mesure la capacité de l'algorithme à trouver tous les défauts.
Un système ne produisant aucun FN a un rappel de 100%.

- F_{mesure} (ou F_{β} , définie par l'équation (3.7)) : moyenne harmonique de la précision et du rappel. Elle est également connue sous le nom de mesure F_1 (pour $\beta = 1$). Cette fonction correspond à un compromis entre la précision et le rappel donnant la performance globale du système.

$$F_{\text{mesure}} = (1 + \beta^2) \frac{\text{précision} \cdot \text{rappel}}{\beta^2 \cdot \text{précision} + \text{rappel}} \in [0\%, 100\%] \quad (3.7)$$

où β correspond au coefficient de la moyenne harmonique pondérée.

Si $\beta < 1$, on privilégie la précision.

Si $\beta > 1$, on privilégie le rappel.

Pour sanctionner plus fortement les faux négatifs (situation critique) que les faux positifs (fausse alarme), il suffit de prendre $\beta > 1$.

Un « bon » algorithme de détection de défauts doit présenter d'une part un rappel élevé et, d'autre part, un taux de faux positifs faible. Autrement dit, le rappel et la précision doivent être proches de 100 %.

Ces indicateurs seront donc utilisés par la suite pour mesurer la performance de l'algorithme.

3.9.7 Résultats d'inspection

La performance de l'algorithme dans le deux modes (tablette et robotisé) est déterminée à l'aide des indicateurs de succès présentés à la section 3.9.6, que l'on

peut facilement déduire par la confrontation des valeurs observées (vérité terrain) et des valeurs prédites (résultats de l'algorithme) par l'intermédiaire de la matrice de confusion présentée dans le tableau 3.3.

Nous présentons dans cette section les résultats d'inspection obtenus dans les deux contextes, robotisé et manuel, ainsi que l'analyse de ces résultats.

3.9.7.1 Inspection via une tablette portée par un opérateur

Le tableau 3.4 (dérivé de la matrice de confusion présentée dans le tableau 3.3) donne les VP, VN, FP et FN obtenus avec notre méthode dans le contexte tablette pour chacun des assemblages utilisés dans cette étude. De plus, le tableau 3.5 donne la précision, le rappel et la fonction F_{mesure} , calculés pour l'ensemble des assemblages.

On peut constater que l'algorithme d'inspection se caractérise par un rappel très élevé (100 % c.-à-d. FN = 0), une très bonne précision (avoisinant 96 %) et une performance globale très élevée puisque $F_{1.5}$ avoisine 98 %.

L'analyse de ces résultats montre immédiatement que la performance de notre méthode est très élevée (précision et rappel proches de 100 %). Ces très bonnes performances sont facilitées par le fait que les données utilisées par le mode tablette sont acquises sur des assemblages mécaniques relativement simples, avec des éléments bien séparés.

Table 3.4 – Résultats d'évaluation

Assemblage	Nombre d'éléments	Nombre d'images	VP	VN	FP	FN
Panneau 1 (figure 3.22b)	6	250	114	136	0	0
Panneau 2 (figure 3.22c)	4	122	88	31	3	0
Panneau 3 (figure 3.25a)	9	300	158	142	0	0
Moteur XWB (figure 3.25b)	56	5135	1321	3686	128	0
Moteur 34 Rafale (figure 3.25c)	30	1200	832	364	4	0
Totale	105	7007	2513	4359	135	0

Table 3.5 – Résultats des indicateurs de succès obtenus sur l'ensemble des données

Précision	Rappel	F_{β}		
		$\beta = 0.5$	$\beta = 1.0$	$\beta = 1.5$
95.15 %	100 %	89.57 %	97.5 %	98.26 %

3.9.7.2 Inspection robotisée

Le tableau 3.6 (dérivé de la matrice de confusion présentée dans le tableau 3.3) donne les VP, VN, FP et FN obtenus avec notre méthode dans le contexte robotisé pour chacun des assemblages utilisés dans cette étude.

Une analyse du tableau 3.7 montre que notre système se caractérise par un rappel très élevé (100 %), quel que soit l'élément à inspecter. Autrement dit, la capacité de l'algorithme à détecter les défauts est très bonne ($FN = 0$) sur l'ensemble des assemblages inspectés.

En revanche, la précision n'est que de 84.52 %. Cela signifie que, parmi les alertes envoyées par l'algorithme d'inspection, une alerte sur dix correspond à une fausse alarme, c'est-à-dire que l'algorithme trouve à tort que l'élément est défectueux. Ceci s'explique par le fait que les données utilisées pour le mode robotisé, notamment sur le moteur LEAP (figure 3.22a), sont acquises en usine, sur un assemblage réel et très complexe. De plus, l'environnement autour de l'élément à inspecter est très encombré et la tâche d'appariement devient beaucoup plus difficile.

La performance globale du système d'inspection peut être évaluée par la fonction de mesure $F_{1.5}$ qui avoisine 94 %.

Table 3.6 – Résultats d'évaluation

Assemblage	Nombre d'éléments	Nombre d'images	VP	VN	FP	FN
Panneau 1 (figure 3.22b)	6	161	93	61	7	0
Panneau 2 (figure 3.22c)	4	154	87	58	9	0
Moteur LEAP (figure 3.22a)	33	328	164	92	72	0
Totale	43	643	344	236	63	0

Table 3.7 – Indicateurs de succès obtenus sur l'ensemble des données

Précision	Rappel	F_β		
		$\beta = 0.5$	$\beta = 1.0$	$\beta = 1.5$
84.52 %	100 %	87.22 %	91.6 %	94.01 %

3.9.7.3 Interprétation des résultats

Les résultats obtenus avec le mode tablette ($F_{1.5} = 98.3$ %) sont légèrement meilleurs que ceux obtenus avec la plateforme robotique ($F_{1.5} = 94$ %). Il y a deux raisons principales à cela :

1. les données utilisées pour le mode tablette ont été acquises sur des assemblages relativement simples (voir figure 3.25), alors que les données de la plateforme robotique ont été acquises en usine, sur un assemblage réel et très complexe, à la

toute fin de la chaîne de production (voir figure 3.23). De plus, l'environnement autour de l'élément à inspecter est très encombré et la tâche d'appariement devient beaucoup plus difficile.

2. la mobilité du système portatif permet une localisation beaucoup plus précise des éléments que celle obtenue avec le système robotisé. En effet, contrairement au mode tablette qui utilise une seule caméra pour la localisation et l'inspection, les erreurs de calibration de l'ensemble des caméras (inspection et localisation) dans le mode robotisé diminuent la précision de localisation.

3.10 Conclusion

Dans cette section, nous faisons une synthèse de nos travaux sur l'inspection 2D basée images et nous proposons des pistes pour la poursuite de ces travaux.

3.10.1 Synthèse des travaux

Dans ce chapitre, nous avons présenté une méthode pour l'inspection d'assemblages mécaniques aéronautiques par vision 2D (analyse d'images) qui exploite la maquette numérique et la localisation, et qui peut être mise en œuvre suivant 2 scénarios : via un robot, équipé de deux caméras, ou via une tablette, équipée d'une caméra, portée par un opérateur.

La méthode d'inspection proposée exploite le modèle CAO de la pièce inspectée, qui sert de référence. Dans cette méthode, nous avons utilisé la description des contours 3D sous forme d'edgelets pour analyser la correspondance dans l'image 2D et vérifier que les edgelets 3D projetés sont bien retrouvés dans l'image. La pose relative entre l'élément à inspecter et la caméra est fournie par un algorithme de SLAM contraint par le modèle. Les edgelets 3D extraits du modèle CAO sont projetés dans l'image 2D en utilisant la projection perspective à l'aide des paramètres de calibration et de localisation de la caméra. Ensuite, pour éviter les contours parasites et anticiper les occultations dans l'étape de mise en correspondance 2D/3D, nous avons développé une approche innovante basée sur l'analyse d'une image synthétique générée à partir du modèle CAO (image contexte). Enfin, l'analyse des mises en correspondance entre les edgelets projetés et les contours 2D extraits dans l'image permet de faire un diagnostic sur la conformité de l'assemblage.

3.10.2 Limitations de la méthode d'inspection proposée et améliorations futures

La méthode proposée est pertinente lorsque l'objet est sur un fond uniforme. Cette méthode est limitée à l'inspection d'éléments rigides qui ne changent pas de forme. Elle est plutôt adaptée à la vérification de présence/absence et du bon positionnement d'un objet dont la forme 3D est assez caractéristique et non ambiguë (il y a de nombreux éléments de ce type dans un assemblage mécanique aéronautique). Certains éléments de la maquette numérique peuvent être plus difficile à inspecter

avec cette méthode. Nous avons par exemple testé la méthode pour l'inspection (présence/absence) des connecteurs qui tiennent les câbles au sein de l'assemblage. Les résultats n'ont pas été très bons en raison d'un très grand nombre de contours parasites générés par le câble lui-même. Pour ce type particulier d'inspection, nous avons développé une méthode basée sur le Deep Learning qui est décrite dans l'annexe B. Il nous semble que cette méthode est difficilement généralisable pour l'inspection de n'importe quel élément de l'assemblage car ce dernier est généralement constitué d'éléments très variés (forme, taille, couleur, etc.). Il faudrait pour cela disposer d'une base d'apprentissage très grande représentative des éléments à inspecter. Une solution à explorer consiste à constituer une base d'apprentissage à partir de données synthétiques.

Bien que ce soit rare, il arrive parfois que la méthode d'inspection présentée dans ce chapitre échoue lorsque l'élément à inspecter possède la même couleur et luminance que le fond (par exemple éléments noirs sur fond noir). La cause est habituellement une mauvaise segmentation des contours (difficultés à discriminer les pixels entre eux et de séparer les objets du fond). Ce problème pourrait être pallié en complétant les informations 2D (image) par une information de profondeur 3D, à travers un couplage 2D/3D.

La gestion des contours parasites et des occultations est mise en œuvre via l'analyse d'une image synthétique générée à partir du modèle CAO. Un axe de recherche intéressant pour la poursuite de ce travail serait d'intégrer cette étape d'analyse dans l'étape de génération des contours 3D (edgelets). En effet, dans nos travaux, nous avons utilisé des edglets statiques (car générés en ligne et indépendamment du point de vue de la caméra). Nous pourrions générer des edgelets en ligne par rendu de synthèse sur la carte graphique à chaque changement de point de vue (edgelets dynamiques). Cela permettrait de rendre la localisation robuste aux occultations. En plus d'assurer la localisation d'objets partiellement occultés, cela permettrait aussi de maximiser les chances de mise en correspondance correcte avec les contours 2D dans les images.

En ce qui concerne la prise de décision sur l'état de l'élément, certaines autres possibilités pourraient être étudiées, par exemple, ajouter un réseau de neurones entièrement convolutionnel qui calculerait l'emplacement et l'orientation de l'élément en cas de mauvais assemblage. Cela viendrait en prolongement de l'une des pistes prometteuses que nous avons commencé à étudier en utilisant des réseaux de neurones entièrement convolutionnels et des rendus synthétiques plus réalistes pour la génération des primitives 3D à partir du modèle CAO et l'extraction des contours 2D dans l'image réelle.

À l'heure actuelle, l'inspection multi-vues n'est possible qu'avec une plate-forme portative. Cela signifie que nous prenons une décision finale sur le statut de l'élément en fusionnant les résultats obtenus à partir de vues multiples. En mode robotique, une seule acquisition d'image est effectuée (acquisition en mode one-shot). L'inspection multi-vues est une fonctionnalité à ajouter à notre plateforme robotique pour la suite de nos travaux.

Dans ce chapitre, nous avons présenté une méthode d'inspection par analyse d'images 2D qui permet de faire un diagnostic sur la présence ou l'absence d'éléments

rigides composant l'assemblage. Dans les chapitres suivants, nous nous intéressons à l'inspection d'éléments de type câbles électriques, qui a nécessité la mesure de distances 3D, et donc la mise en œuvre d'une méthode d'inspection basée sur l'analyse de nuages de points 3D fournis par un scanner.

Inspection à partir de nuages de points 3D : segmentation 3D des câbles

4.1	Introduction	90
4.2	Schéma général de la méthode proposée	92
4.3	Pré-traitements	93
4.3.1	Filtrage du nuage de points 3D	93
4.3.2	Transformation d'un modèle CAO maillé en nuage de points 3D	95
4.3.3	Sous-échantillonnage uniforme du nuage de points	96
4.4	Initialisation de la segmentation 3D d'un câble	98
4.4.1	Alignement des deux nuages	99
4.4.2	Génération d'un point pour initialiser la segmentation	101
4.5	Segmentation 3D d'un câble	103
4.5.1	Recherche du premier sous-cylindre	105
4.5.1.1	Structure de recherche	105
4.5.1.2	Algorithme de recherche	107
4.5.2	Ajustement des paramètres du premier sous-cylindre	107
4.5.2.1	Ajustement de la longueur	108
4.5.2.2	Ajustement de l'axe principal	110
4.5.2.3	Ajustement du rayon	113
4.5.3	Propagation de la recherche/ajustement des sous-cylindres	115
4.6	Résultats expérimentaux	120
4.6.1	Conditions expérimentales	121
4.6.2	Présentation des données	121
4.6.2.1	Modèles CAO disponibles	121
4.6.2.2	Nuages de points synthétiques simulés	121
4.6.2.3	Nuages de points réels acquis par le scanner	123

4.6.3	Évaluation du processus d'ajustement des paramètres	123
4.6.3.1	Métriques d'évaluation	123
4.6.3.2	Comparaison avec les méthodes d'ajustement classiques les plus populaires	125
4.6.4	Évaluation de la méthode de modélisation-segmentation à partir de données synthétiques	129
4.6.5	Évaluation de la méthode de modélisation-segmentation avec des données réelles	132
4.7	Conclusion	136
4.7.1	Synthèse des travaux	136
4.7.2	Perspectives	137

Le contenu de ce chapitre a été publié dans [Ben+20].

4.1 Introduction

Nous avons vu dans le chapitre précédent une méthode d'inspection basée uniquement sur l'analyse d'images 2D et l'exploitation du modèle CAO de l'élément à inspecter et de son environnement qui servent de référence. Cependant, certains types d'inspection ne peuvent pas être réalisés par « simple » analyse d'images 2D. Un exemple typique est l'inspection des câbles.

À titre d'exemple, on voit sur la figure 4.1 que, selon le point de vue, l'analyse d'une image 2D ne permet pas de déterminer la distance entre les câbles.

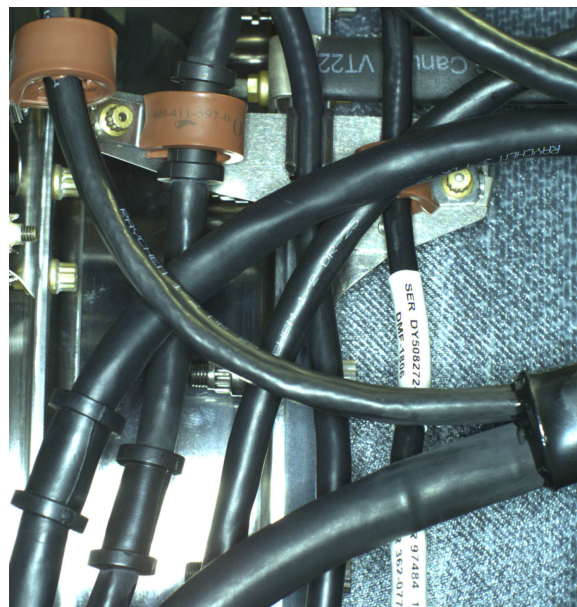


Figure 4.1 – Exemple d'inspection : une scène complexe constituée des différents câbles à inspecter

Pour ce type de diagnostic, nous utilisons un capteur 3D qui fournit des nuages de points 3D. Dans ce chapitre, nous nous intéressons à la segmentation 3D des

câbles. Les résultats de segmentation seront utilisés dans le chapitre 5 pour fournir un diagnostic sur les câbles. Plus précisément, pour mesurer le rayon de courbure de chaque câble et vérifier si les câbles sont à une distance minimale des autres éléments qui l’entourent (problème d’interférences).

Les câbles ou les harnais sont des éléments non-rigides (flexibles) qui peuvent changer de forme. Très peu de travaux de recherche existent sur la segmentation des formes cylindriques flexibles. Par ailleurs, à notre connaissance, il n’existe pas de logiciels qui permettent d’effectuer la segmentation des formes cylindriques flexibles (e.g. câbles).

Les câbles étant flexibles, leur forme après assemblage est souvent différente de la forme du câble dans le modèle CAO. Par conséquent, l’utilisation du modèle CAO pour l’inspection des câbles va être plus difficile. Nous avons donc développé une méthode de segmentation des câbles qui exploite au minimum le modèle CAO. Par ailleurs, une autre difficulté à surmonter sera liée au fait que le nuage de points 3D acquis par le scanner est bruité et contient des données éparses et parfois incomplètes du fait que nous avons utilisé une acquisition de type *one-shot* (un seul nuage acquis depuis un seul point de vue) (voir la figure 4.2).

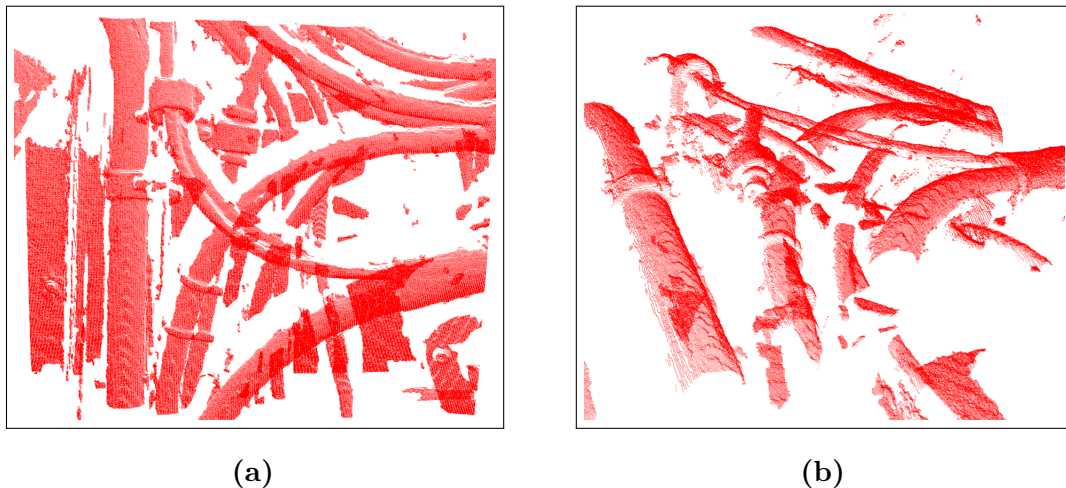


Figure 4.2 – Nuage de points acquis par le scanner (vu sous 2 angles différents) : la vue de droite montre que les données sont éparses et que l’on ne dispose que d’une faible proportion de la surface du câble cylindrique à segmenter (à peine 1/3 de la circonférence)

Dans ce chapitre, nous présentons une méthode originale de segmentation 3D de formes cylindriques flexibles dans un nuage de points 3D. La première étape consiste à effectuer certains pré-traitements sur les données disponibles (nuage de points 3D acquis par le scanner et modèle CAO 3D) : filtrage du nuage de points, afin de réduire le bruit et de ne conserver que les points d’intérêt, et échantillonnage pondéré du modèle CAO pour générer un nuage de points 3D synthétique à partir d’un maillage 3D. L’étape suivante consiste à segmenter le câble en plusieurs étapes : recherche d’un point 3D sur le câble pour initialiser la segmentation, et segmentation

proprement dite qui est basée sur une modélisation du câble par une succession de cylindres. Le protocole expérimental et les résultats obtenus, sur des données réelles et des données synthétiques, sont présentés et analysés dans la section 4.6.

4.2 Schéma général de la méthode proposée

La figure 4.3 présente le schéma général de la méthode de segmentation proposée. Tout d'abord, pour réaliser les acquisitions, nous utilisons un robot équipé de deux caméras 2D et d'un scanner 3D. Dans ce chapitre, nous utilisons le scanner 3D pour acquérir les nuages de points 3D, et une caméra 2D pour localiser le scanner 3D dans la scène. Afin d'analyser les observations, les nuages de points acquis sont envoyés vers un PC de traitement. L'interaction entre l'effecteur, le superviseur et ce PC est gérée par le middleware ROS. Afin de faciliter la segmentation et obtenir des données cohérentes, les nuages de points acquis par le scanner seront envoyés à un processus de pré-traitements. Les résultats de ce processus seront utilisés pour modéliser puis segmenter les câbles. La méthode de segmentation proposée s'appuie sur la détection et la modélisation de portions de cylindres, qui seront ensuite étendues par propagation sur le câble dans les deux sens à l'aide d'une structure de recherche basée antipodes d'une sphère.

Enfin, les résultats de segmentation obtenus seront utilisés dans le chapitre 5 pour fournir un diagnostic sur les câbles. Plus précisément, pour mesurer le rayon de courbure de chaque câble et vérifier si les câbles sont à une distance minimale des autres éléments de l'assemblage (problème d'interférence).

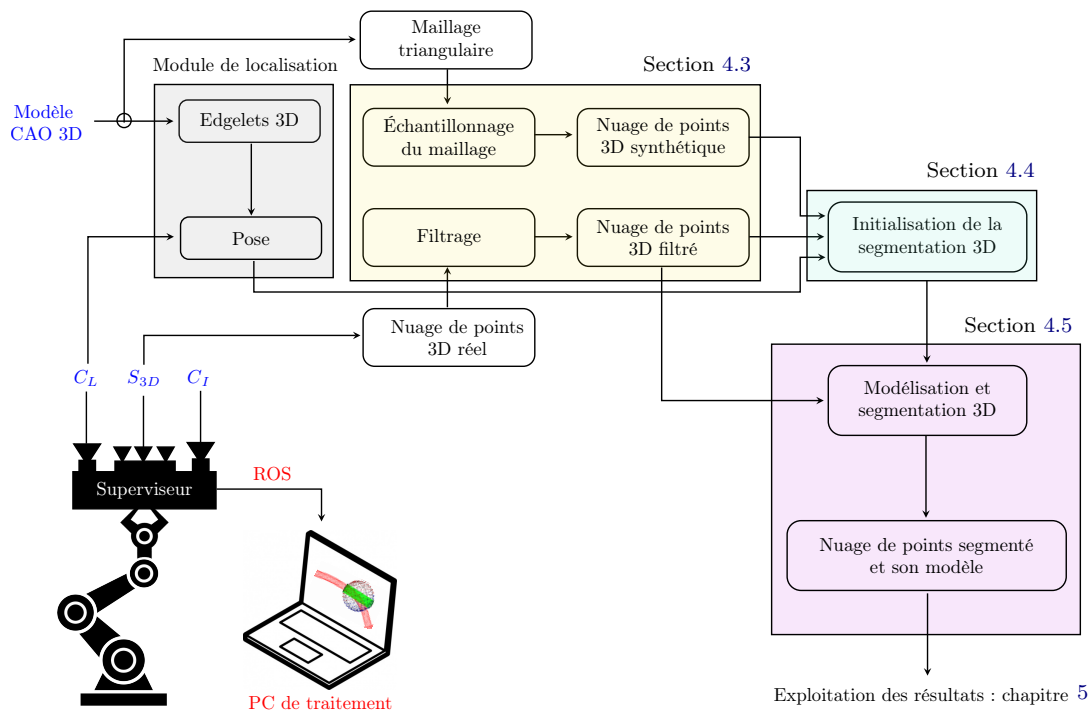


Figure 4.3 – Schéma général de la méthode proposée

4.3 Pré-traitements

La phase de pré-traitements est souvent une étape nécessaire à l'obtention de données cohérentes. Dans cette section, nous proposons une méthode de filtrage, afin de réduire le bruit et de ne conserver que les points d'intérêt dans le nuage de points. Ensuite, nous détaillons une méthode permettant de générer des nuages de points 3D à partir de maillages 3D afin de disposer d'une représentation du modèle CAO sous forme de nuage de points. Le schéma général de la phase de pré-traitements est illustré sur la figure 4.4.

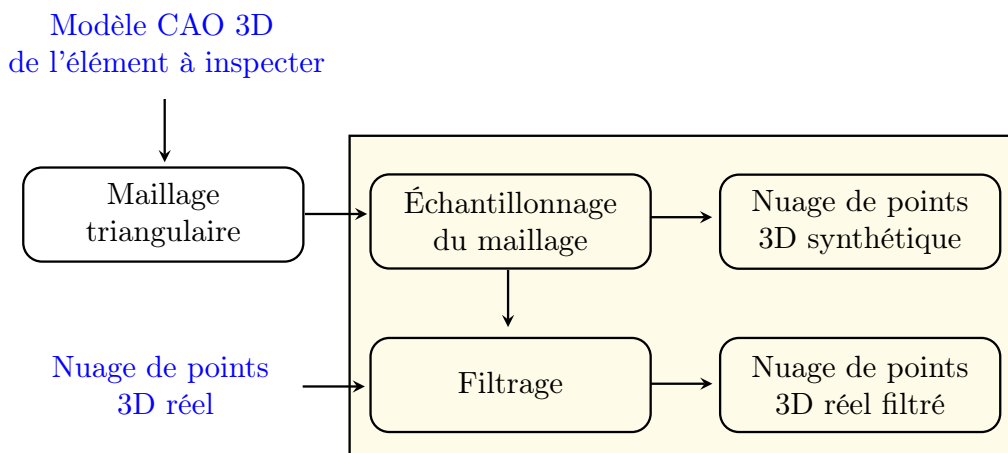


Figure 4.4 – Schéma général de la phase de pré-traitements

4.3.1 Filtrage du nuage de points 3D

Bien que la qualité des scanners ait été grandement améliorée ces dernières années, nous obtenons encore des erreurs de mesure inévitables et des valeurs aberrantes dans le nuage de points, causées par diverses raisons telles que :

- Les erreurs liées au bruit causé par les conditions environnementales (la lumière ambiante ou même la température).
- Les erreurs liées aux points 3D qui correspondent aux zones de discontinuité dans le nuage (effets de bord).
- Les erreurs liées à des problèmes de réflexion (dépend de la spécularité des surfaces et de l'orientation de la surface par rapport à la direction du faisceau lumineux).

Le principe du filtrage est de déterminer quels points constituent une donnée valide (points justes) et quels autres ne sont que des « faux points ».

Notre méthode de filtrage automatique consiste à supprimer les points isolés ou les groupes de points (clusters) trop éloignés de la surface théorique, par analyse sur les voisinages locaux. Il s'agit en quelque sorte de ne garder que les groupes de points dits « représentatifs ».

Afin d'éliminer les points isolés ou les groupes de points isolés (souvent des artefacts), nous considérons chaque point comme le centre d'une sphère 3D ayant un rayon r . Ensuite, nous déterminons le nombre de voisins à l'intérieur de cette sphère. Enfin, si le point a moins de k -voisins, il est retiré du nuage de points. Cette technique part du *postulat* que les données valides constituent des ensembles de données plus denses que les autres.

Les paramètres du filtre sont les suivants : rayon de la sphère (r) définissant le voisinage à étudier et le nombre entier positif (k) définissant le nombre minimum de points attendus à l'intérieur de la sphère. Leurs effets sur le nuage de points résultant dépendent du niveau de bruit et de sa distribution. Par conséquent, il n'existe pas de méthode générique pour déterminer ces paramètres. Cependant, nous avons constaté que, dans la pratique, l'expérience guide la détermination des paramètres optimaux nécessaires en fonction de la résolution du nuage de points \mathcal{R}_P , qui représente la moyenne de la distance entre chaque point et son plus proche voisin. De plus, les effets sont principalement locaux et généralement peu sensibles aux paramètres du filtre.

La figure 4.5 illustre un exemple de résultat de filtrage. Au départ le nuage de points comporte 131200 points pour une surface de 520 cm² (26 cm × 20 cm). À la fin du processus, le nuage de points comporte 95782 points pour 520 cm². On constate que 27 % des points sont considérés comme des point de bruit.

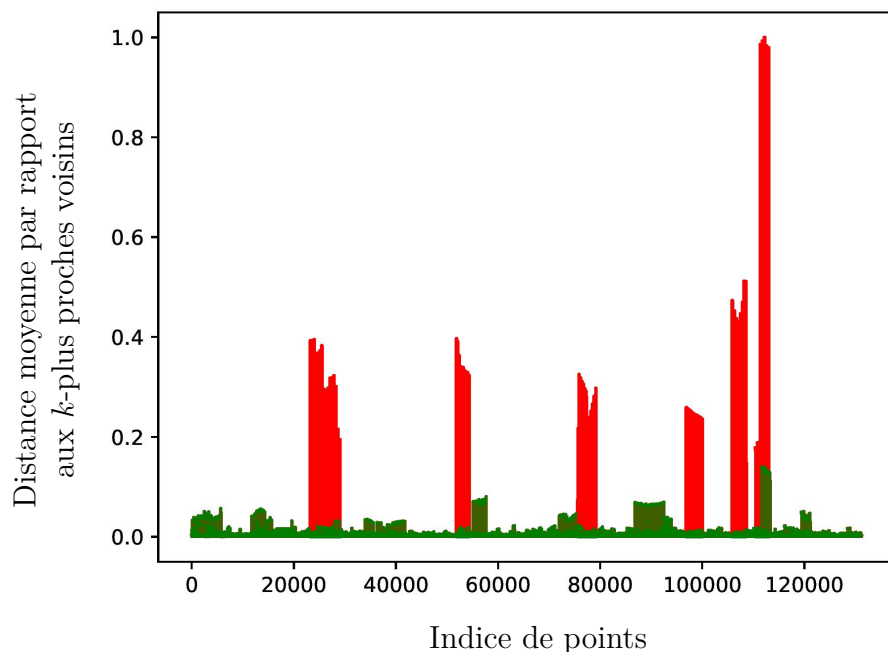


Figure 4.5 – Distance moyenne par rapport aux k -plus proches voisins (kNN) (nuage de points d'entrée en rouge, nuage de points filtré en vert)

Remarques :

1. Si le nuage de points d'entrée est composé de plusieurs groupes de points (clusters) suffisamment dissociés les uns des autres, il est possible de le décomposer

avec notre sphère de filtrage en sous-nuages compacts. Il s'agit en quelque sorte de faire une extraction des composantes connexes (*connected component* en anglais) dans le nuage de points.

2. Pour une version future de l'algorithme de filtrage, nous étudierons l'apprentissage automatique pour automatiser la détermination des paramètres du filtre et leur ajustement adaptatif.

À partir de maintenant, quand nous parlerons de nuage de points fourni par le scanner, nous parlerons du nuage de points filtré.

4.3.2 Transformation d'un modèle CAO maillé en nuage de points 3D

Il s'agit d'échantillonner de manière uniforme une surface décrite par un maillage triangulaire, tout en préservant les caractéristiques géométriques de la surface originale.

Les deux approches le plus fréquemment rencontrées dans la littérature pour l'échantillonnage du maillage (*mesh sampling* en anglais) sont :

- Technique de lancer de rayons. Cette méthode consiste à lancer un rayon depuis chaque pixel d'une caméra virtuelle et à stocker l'intersection de ce rayon avec le premier maillage 3D rencontré [Ber+16]. Les points d'observation de la caméra virtuelle sont définis par les sommets d'une sphère tessellée par des triangles (tessellation de type icosaèdre). Comme illustré sur la figure 4.7c, cette méthode engendre des erreurs d'échantillonnage.
- Échantillonnage aléatoire naïf. Cette méthode consiste à échantillonner de manière aléatoire les faces triangulaires du maillage. Comme illustré sur la figure 4.7b, cette méthode engendre un sur-échantillonnage des zones dont le maillage est le plus raffiné, et un sous-échantillonnage des zones dont le maillage est le moins raffiné.

Afin de résoudre ces problèmes, la méthode proposée consiste à pondérer pour chaque triangle l'échantillonnage en fonction de sa surface. Ensuite, pour empêcher les échantillons de dépasser les bords du triangle lors de leur génération, nous utilisons le concept de coordonnées barycentriques. En effet, pour un triangle $\triangle ABC$ non aplati (voir la figure 4.6), (dans ce cas (A, B, C) est un repère barycentrique), un point P quelconque du triangle a pour coordonnées :

$$u \times \overrightarrow{PA} + v \times \overrightarrow{PB} + w \times \overrightarrow{PC} = 0 \quad (4.1)$$

où u , v et w représentent les coordonnées normalisées vérifiant l'équation suivante :

$$u + v + w = 1 \quad (4.2)$$

Pour générer un point P à l'intérieur du triangle, nous générons aléatoirement les valeurs de u , v et w de telle sorte qu'elles satisfassent l'équation (4.2).

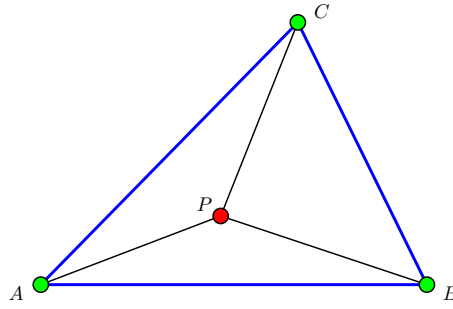


Figure 4.6 – Interprétation des coordonnées barycentriques en termes de rapports d’aires

Comme illustré sur la figure 4.7d, la méthode proposée produit une distribution uniforme des échantillons sur la surface décrite par le maillage du modèle CAO, tout en assurant une certaine régularité spatiale.

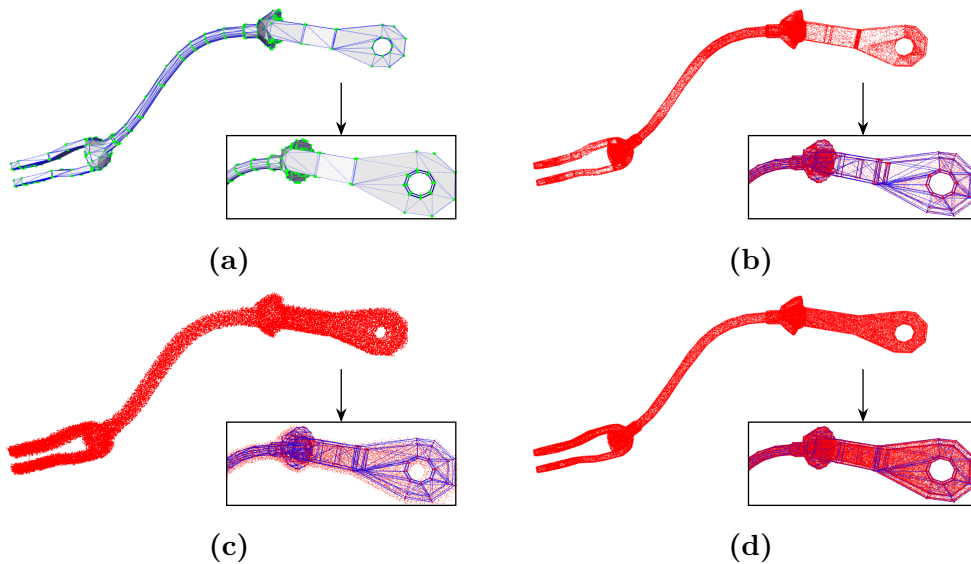


Figure 4.7 – Comparaison quantitative entre la méthode proposée et les deux approches le plus fréquemment rencontrées dans la littérature pour l’échantillonnage du maillage : (a) maillage triangulaire d’un modèle CAO, les points verts et les lignes bleues représentent respectivement les sommets et les arêtes du maillage, (b) nuage de points généré avec la méthode d’échantillonnage aléatoire naïf, (c) nuage de points généré avec la technique de lancer de rayons, et (d) nuage de points généré avec notre méthode d’échantillonnage pondéré

4.3.3 Sous-échantillonnage uniforme du nuage de points

Il peut être nécessaire d’aligner le nuage de points 3D avec le nuage de points généré à partir du modèle CAO (cf. section 4.4). Si les deux nuages ont des résolutions

différentes, il peut être difficile d'obtenir un alignement précis [Li+11]. Pour résoudre ce problème, le nuage de points du modèle (généralisé à partir du modèle CAO) doit être soumis à un processus de sous-échantillonnage. Pour cela, nous choisissons de partitionner le nuage de points d'entrée avec une structure hiérarchique et régulière de grille de voxels 3D (voir la figure 4.8a). La grille de voxels 3D peut être vue comme un ensemble de petites boîtes 3D dans l'espace structures sur une arbre de partitionnement 3D (ou *octree*). Ensuite, dans chaque voxel (c.-à-d. boîte 3D), tous les points présents seront sous-échantillonnés avec leur centroïde. Le sous-échantillonnage avec le centroïde représente plus précisément la surface sous-jacente, par rapport aux méthodes plus standards.

La figure 4.8 montre quelques nuages de points générés avec différentes valeurs de résolution détaillées dans le tableau 4.1.

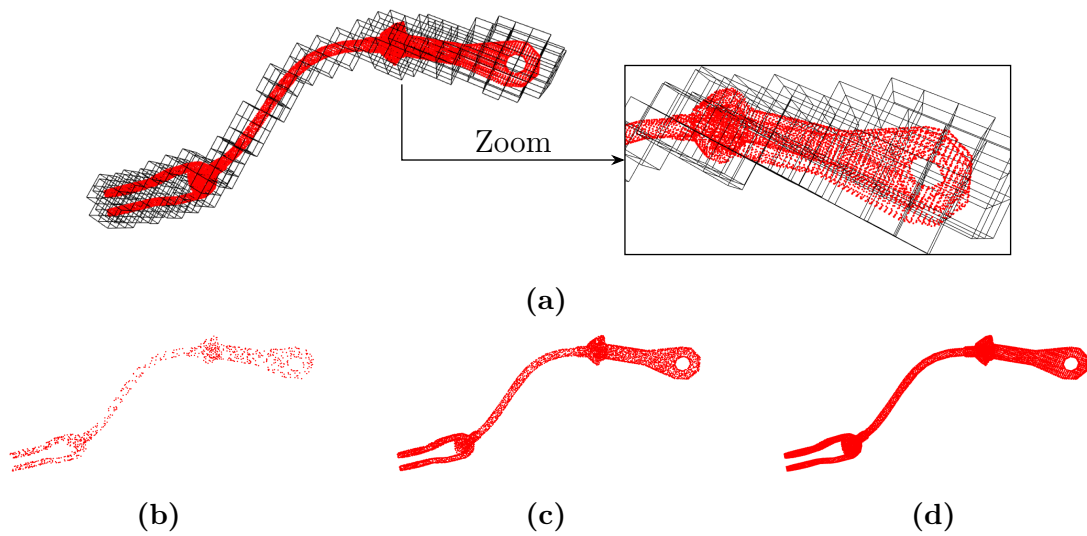


Figure 4.8 – Sous-échantillonnage du nuage de points : (a) une série de voxels entourant le nuage de points d'entrée ; l'*octree* résultant est représenté en noir, (b-d) quelques nuages de points générés avec différentes valeurs de résolution détaillées dans le tableau 4.1

Table 4.1 – Nombre de voxels, nombre de points et résolution obtenus pour les différents nuages de points représentés sur la figure 4.8

Figure	Nbr de voxels	Nbr de points	$\mathcal{R}_{\mathcal{P}}$ (mm)
4.8b	1000	947	1.06
4.8c	10000	6483	0.75
4.8d	100000	12433	0.66

Remarques :

1. Le **sous**-échantillonnage diffère du **ré**-échantillonnage dans le sens où il ne crée pas de nouveaux points mais se contente de sélectionner un sous-ensemble de points à partir du nuage d'entrée.

-
2. L'arbre de partitionnement 3D (ou *octree*) est une structure de données généralement utilisée pour accélérer une série d'opérations dans l'espace, comme la recherche des plus proches voisins d'un point, la méthode de lancer de rayons [Por+16] ou la détection de collisions [Jun+96].

4.4 Initialisation de la segmentation 3D d'un câble

L'initialisation de notre méthode de segmentation consiste à trouver un point 3D appartenant au câble que l'on souhaite inspecter.

Pour cela, nous avons développé une méthode qui exploite le modèle CAO via un recalage entre le nuage de points 3D généré à partir du modèle CAO et le nuage de points acquis par le scanner (filtré).

Étant donné que les câbles sont flexibles, leur position réelle dans le nuage n'est généralement pas la même que leur position théorique dans le modèle.

Néanmoins, les points de raccordement des câbles (connecteurs ou pinces) sont des éléments rigides. Par conséquent, ces points (clés) permettent de garantir une rigidité d'une partie du câble (voir la figure 4.9) et ils peuvent être utilisés pour initialiser la segmentation.

Cela signifie que la méthode que nous avons développée ne fonctionne à coup sûr que si nous avons dans le champ de vue au moins une portion rigide du câble (par exemple un connecteur), ce qui est généralement le cas.

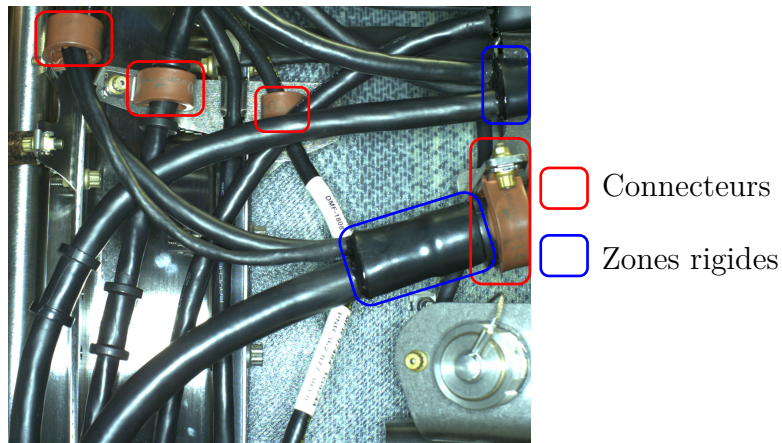


Figure 4.9 – Les points de raccordement des câbles

Pour exploiter le modèle CAO, nous commençons par aligner le modèle CAO du câble avec le nuage de points. Plus précisément, nous alignons le nuage de points synthétique généré à partir du modèle CAO et le nuage de points réel.

4.4.1 Alignement des deux nuages

Comme illustré sur la figure 4.10, cet alignement se déroule en deux étapes :

1. alignement global en exploitant la pose fournie par le module de localisation,
2. alignement local par une méthode de type ICP.

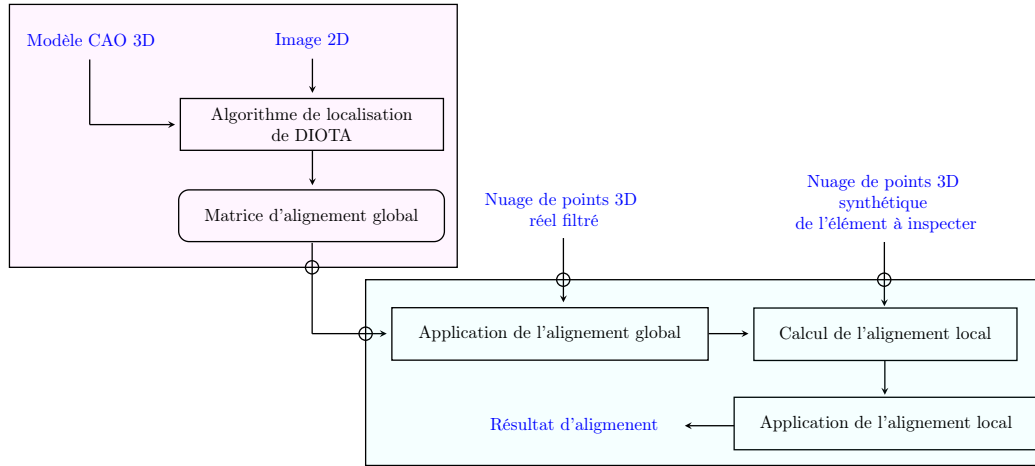


Figure 4.10 – Schéma général de la phase d'alignement proposée

Étape 1 (alignement global) : dans un premier temps, nous avons utilisé la caméra 2D de localisation et l'algorithme SLAM contraint par le modèle (décrit dans le chapitre 3) pour obtenir une estimation initiale de la pose caméra par rapport à la scène, en s'appuyant principalement sur la connaissance *a priori* du modèle CAO. Une fois que la pose de la caméra par rapport à la scène est estimée, nous pouvons facilement exprimer la pose du scanner 3D par rapport à la scène à l'aide de la matrice de passage entre la caméra de localisation et le scanner 3D obtenue par la calibration de l'ensemble des capteurs 2D-3D.

Grâce à cette technique de recalage basé modèle, nous sommes en mesure d'initialiser le recalage du nuage de points avec le modèle CAO. Cette initialisation se base principalement sur l'information globale de la scène. Cependant, dans notre cas, nous ne nous intéressons qu'aux éléments non-rigides (câbles) qui peuvent changer de forme. Par conséquent, l'alignement global n'est pas suffisant pour aligner avec précision le modèle CAO de l'élément à segmenter (câble) sur le nuage de points. Pour résoudre ce problème, nous avons dû ajouter une étape d'alignement local pour raffiner le modèle de câble sur le nuage de points.

Étape 2 (alignement local) : dans un deuxième temps, le résultat de l'alignement global est localement raffiné à l'aide de l'algorithme ICP proposé par BESL et al. [Bes+92] combiné avec l'algorithme de Levenberg-Marquardt [Cha+13 ; Fit02].

Les algorithmes ICP sont peu performants dès lors que les déplacements sont trop grands. Autrement dit, les algorithmes ICP ne peuvent converger¹ que lorsque les objets à recaler sont initialement très proches les uns des autres. Dans notre cas, l'alignement global permet de rapprocher le modèle CAO du nuage de points. Ceci permet d'assurer la convergence (rapide) de l'algorithme ICP.

Pour valider la méthode d'alignement en deux étapes, nous avons sélectionné un critère basé sur l'erreur *MSE* (*Mean Square Error* en anglais) entre le nuage de points généré à partir du modèle CAO et le nuage de points fourni par le scanner. On considère que l'alignement est réussi quand l'erreur *MSE* est inférieure à 1 mm.

La figure 4.11 présente un exemple de résultat obtenu avec notre méthode d'alignement (en deux étapes), alignement global puis local. Comme on le voit sur la figure 4.11a, l'alignement global permet de rapprocher le modèle CAO du nuage de points avec un *MSE* de 1.24 mm. Ensuite, l'alignement local (voir la figure 4.11b) permet de raffiner le résultat de l'alignement global avec un *MSE* très faible de 0.38 mm.

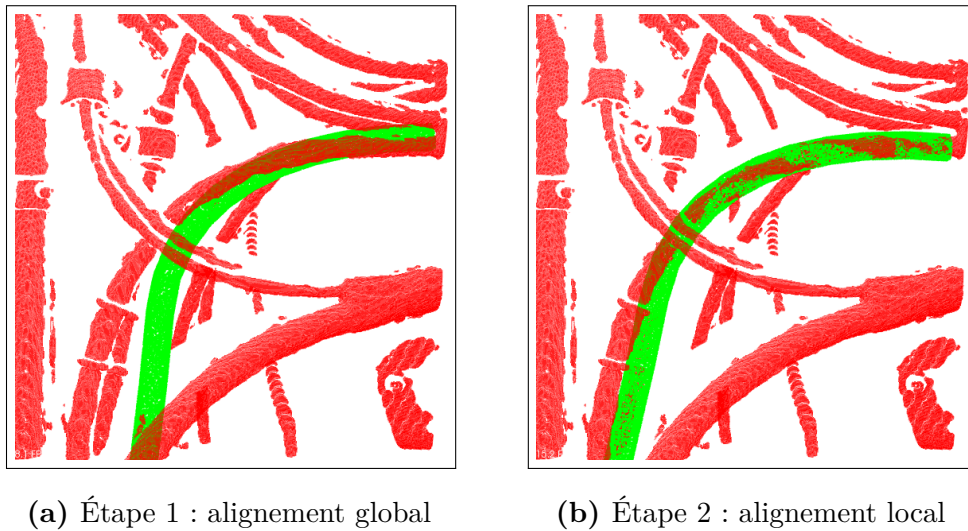


Figure 4.11 – Exemple de résultat obtenu : alignement du modèle CAO (vert) sur le nuage de points (rouge)

De la même manière, et sur la base d'une analyse approfondie effectuée sur une base de données contenant 50 nuages de points différents, nous avons trouvé une erreur *MSE* moyenne de 1.2 mm après l'alignement global et de 0.4 mm après l'alignement local.

Comme on le voit sur la figure 4.12, les différences d'écarts entre le modèle CAO du câble (après l'alignement en deux étapes : alignement global puis alignement local présenté précédemment) et le nuage de points sont très faibles dans les zones rigides (connecteurs).

1. On dit que l'algorithme ICP converge lorsque l'erreur résiduelle de distance entre le modèle CAO et le nuage de points est inférieure à un certain seuil ϵ

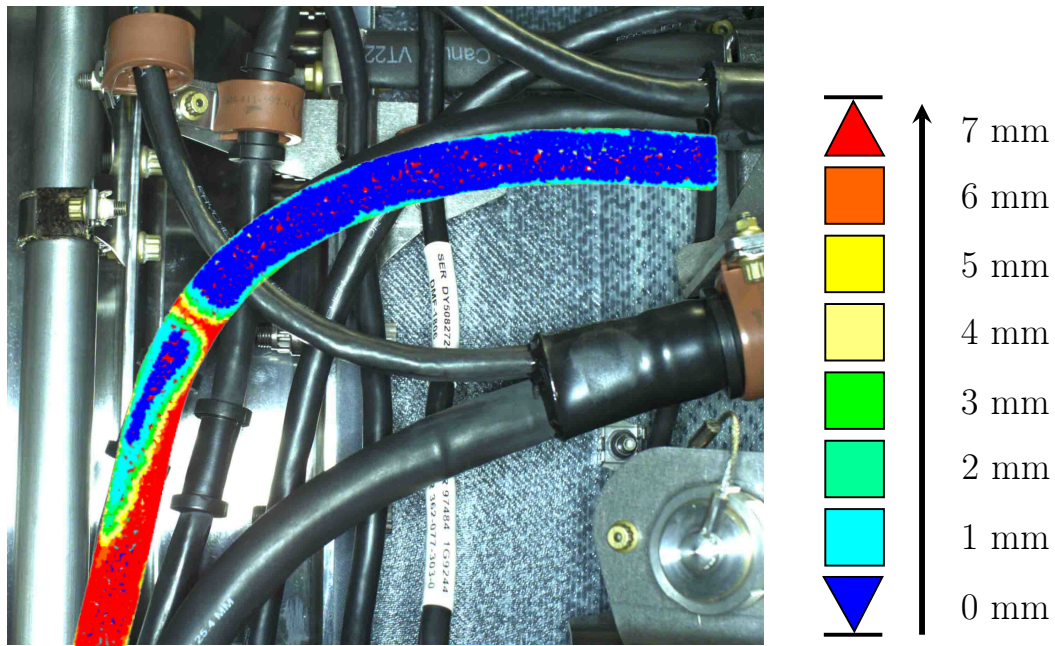


Figure 4.12 – Projection des résultats d'écarts entre le modèle CAO du câble (après l'alignement en deux étapes : alignement global puis alignement local) et le nuage de points sur l'image 2D (bleu : différences faibles et rouge : différences importantes).

La zone bleue de la figure 4.12 correspond à une zone dans laquelle on est certains de trouver un point appartenant au câble dans le nuage de points réel. C'est donc dans cette zone que nous allons générer des points d'initialisation candidats.

4.4.2 Génération d'un point pour initialiser la segmentation

Nous considérons chaque point p_i du nuage synthétique du câble comme le centre d'une sphère 3D ayant un rayon r égal au rayon du câble dans le modèle CAO (le modèle CAO nous fournit la valeur théorique du rayon du câble, mais ce rayon peut aussi être estimé à partir du maillage triangulaire). Ensuite, nous calculons le nombre de points 3D du nuage de points réel qui se trouvent dans cette sphère. Enfin, nous trions les points p_i par ordre décroissant en fonction du nombre de points voisins, à l'intérieur de chaque sphère (voir la figure 4.13).

Comme le montre la figure 4.13c, les 10 meilleurs points p_i trouvés sont principalement focalisés dans la zone rigide (zone bleue), où l'écart entre le modèle CAO du câble et le nuage de points est très faible.

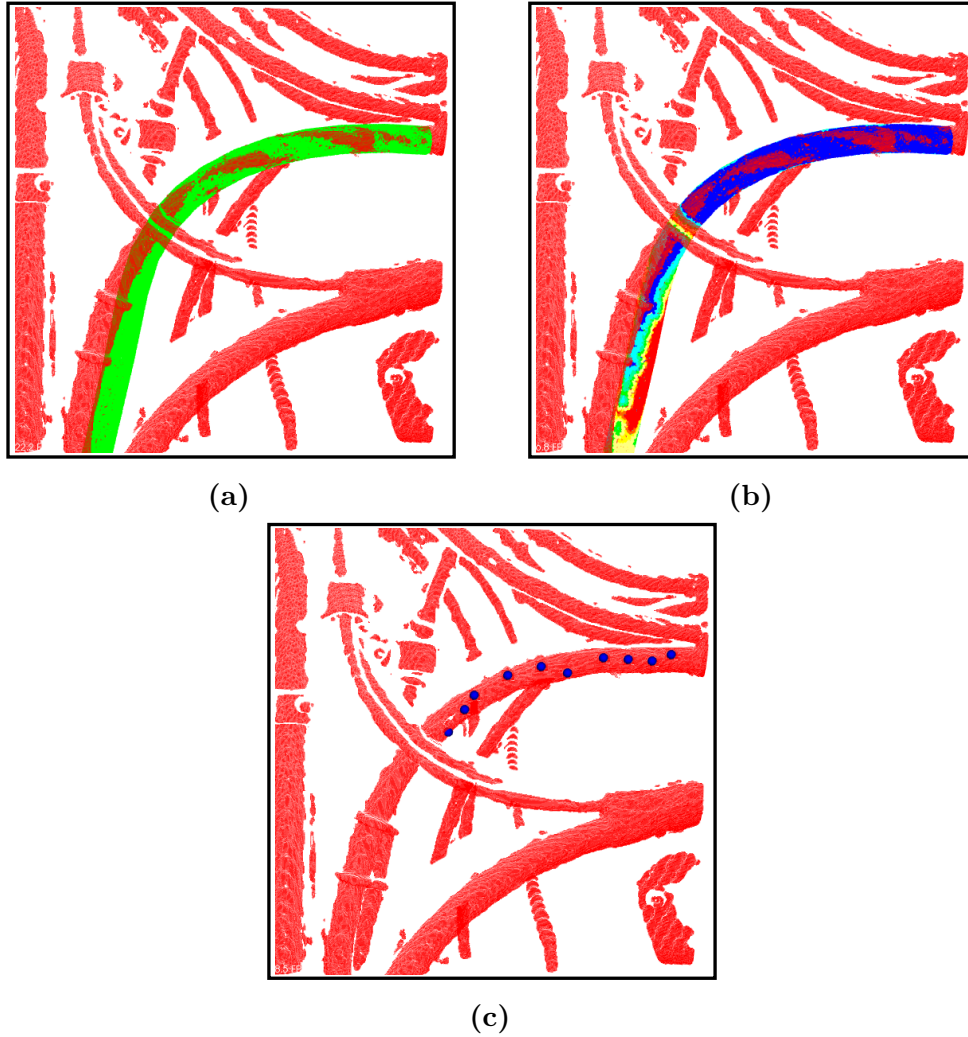


Figure 4.13 – Sélection de points dans le nuage synthétique : (a) résultat d’alignement (en deux étapes), alignement global puis local, (b) valeurs des écarts entre le modèle CAO du câble et le nuage de points (bleu : différences faibles) , (c) un exemple des 10 meilleurs points trouvés (qui ont le plus grand nombre de voisins dans le nuage réel)

Une fois que les points p_i du nuage synthétique sont triés par ordre décroissant, nous commençons par le premier point p_i qui a le plus grand nombre de voisins réels dans sa sphère. À ce point p_i du nuage synthétique, nous associons le point q_i le plus proche dans le nuage réel.

Nous recherchons tous les points du câble réel contenus dans la sphère centrée sur le point q_i (de même rayon que le modèle de câble) (voir les points verts dans l’étape 1 de la figure 4.14).

Parmi ces points, nous recherchons, par un processus itératif, le point qui possède le plus grand nombre de voisins à l’intérieur d’une sphère centrée sur ce point.

À la fin du processus itératif, nous trouvons un point q_c (voir étape n de la figure 4.14). Ce point q_c est choisi comme point d’initialisation de la segmentation.

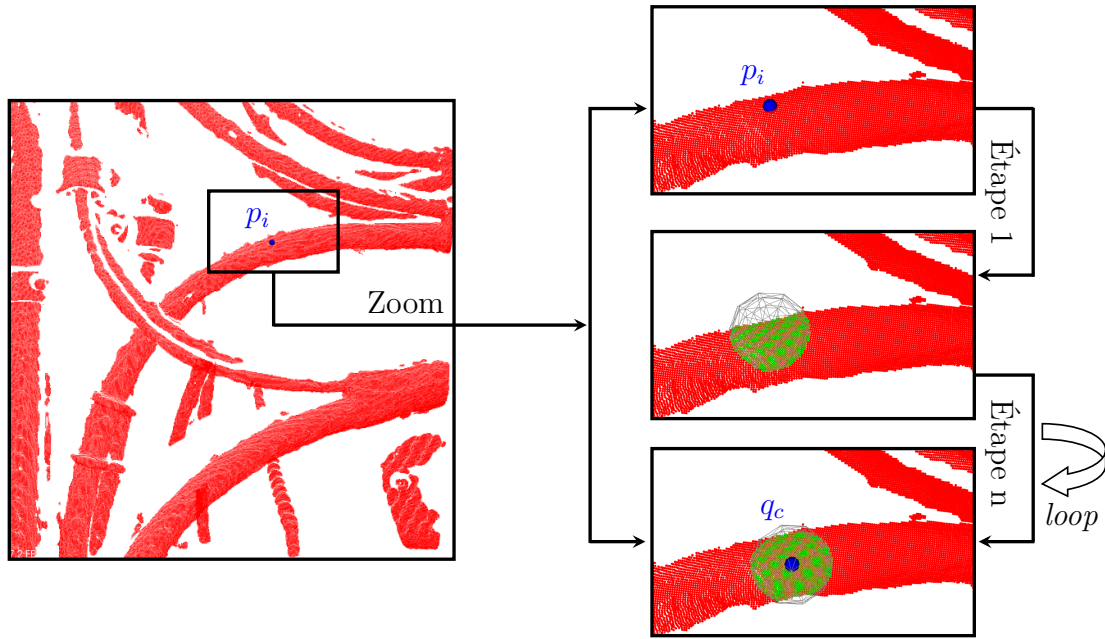


Figure 4.14 – Processus d’ajustement d’un point d’initialisation au centre du câble. Étape 1 : sélection de tous les points qui sont dans une sphère centrée sur le point q_i ; étape n : fin du processus itératif de recherche du point q_c (point centré sur le câble)

4.5 Segmentation 3D d’un câble

Le point 3D q_c trouvé précédemment va permettre d’initialiser la segmentation 3D du câble. Nous avons choisi de modéliser le câble par une succession de cylindres (appelés sous-cylindres par la suite).

Définition 1 - Sous-cylindre s-cy : chaque câble peut être vu comme une succession de cylindres élémentaires (sous-cylindres) dont les orientations dépendent de la courbure du câble (voir la figure 4.15).

Un sous-cylindre est un cylindre virtuel qui modélise une section du câble (appelé sous-câble). Nous considérons que chaque sous-cylindre $s\text{-cy}_k$ est un objet paramétré par sa position c_k , un point initial P_k^i , un point final P_k^f , la direction de son axe d_k (appelée axe principal du sous-cylindre) et son rayon r_k .

L’axe du sous-cylindre $s\text{-cy}_{k+1}$ est contraint à passer par l’extrémité du sous-cylindre $s\text{-cy}_k$. Autrement dit, le point initial P_{k+1}^i du cylindre $s\text{-cy}_{k+1}$ coïncide avec le point final P_k^f du sous-cylindre $s\text{-cy}_k$. Seule la direction de l’axe est autorisée à varier (principe de la liaison rotule).

Définition 2 - Sous-câble s-c : un sous-câble $s-c_k$ est une section de nuage de points du câble qui va être modélisé par un sous-cylindre $s-cy_k$ (voir la figure 4.15).

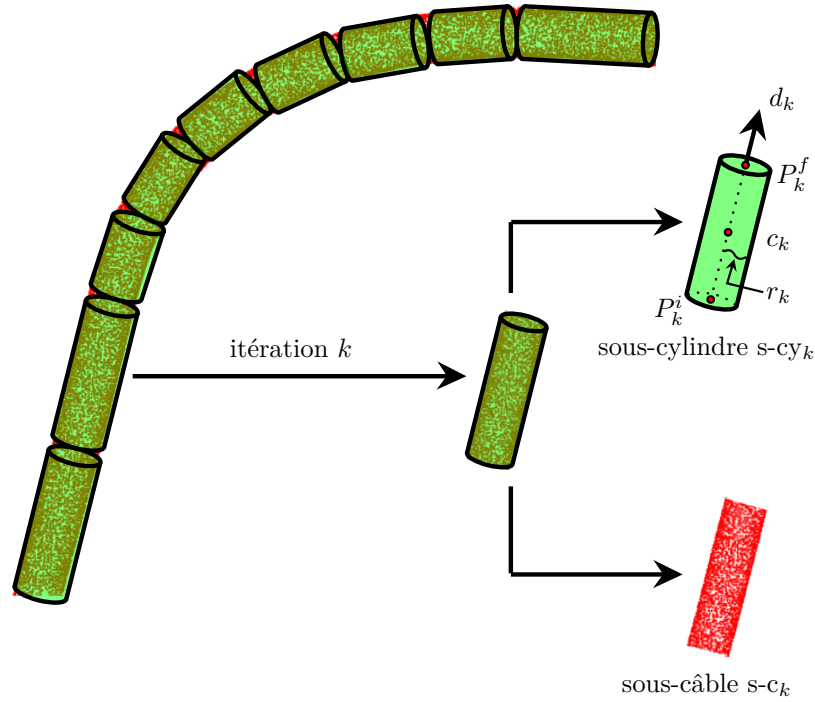


Figure 4.15 – Câble reconstruit à l'aide de cylindres élémentaires (sous-cylindres)

Algorithme global : la segmentation d'un câble est effectuée en plusieurs étapes :

1. à partir du point q_c trouvé lors de la procédure d'initialisation, nous cherchons le premier sous-cylindre (noté $s-cy_1$) grâce à une structure de recherche innovante que nous avons proposée et qui sera décrite dans la section 4.5.1.1,
2. une fois le sous-cylindre initial construit, nous ajustons les paramètres du sous-cylindre sur le nuage de points réel (voir la figure 4.16a),
3. pour rechercher le sous-cylindre suivant, l'algorithme de segmentation propage automatiquement la structure de recherche dans les deux sens à partir des deux extrémités du sous-cylindre initial (P_1^i, P_1^f) (voir la figure 4.16b),
4. le processus de recherche/ajustement est propagé jusqu'aux conditions d'arrêts (voir section 4.5.3).

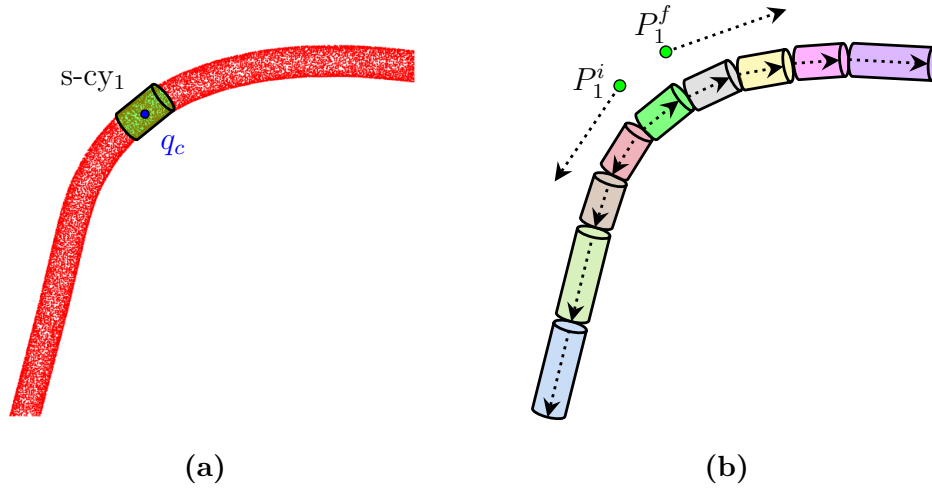


Figure 4.16 – Les étapes de segmentation d'un câble

4.5.1 Recherche du premier sous-cylindre

Pour rechercher le premier sous-cylindre (et les sous-cylindres successifs) nous avons défini une structure de recherche innovante qui va être décrite dans la section suivante.

4.5.1.1 Structure de recherche

Nous allons présenter dans cette section la structure de recherche basée antipodes d'une sphère utilisée pour modéliser/segmenter un câble dans un nuage de points.

Définition : deux points sont considérés comme antipodaux (i.e. chacun est l'antipode de l'autre) s'ils sont diamétralement opposés sur une sphère. De plus, un point antipodal est souvent appelé un antipode. Par exemple, sur la figure 4.17a, le point B est un antipode du point A . Ils sont donc situés de telle manière qu'une ligne verte tracée de l'un à l'autre passe par le centre de la sphère.

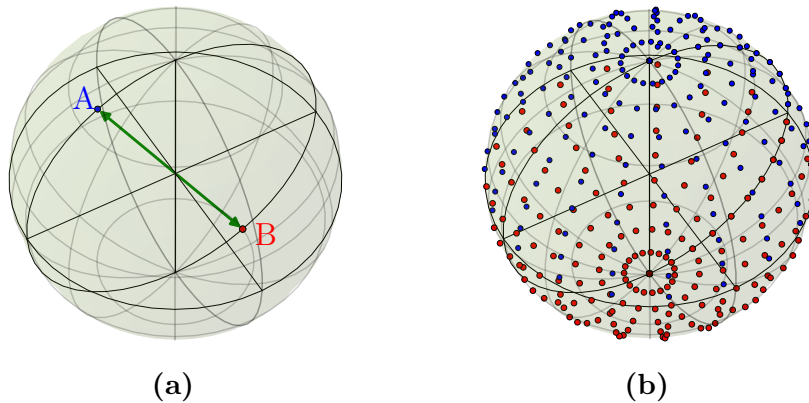


Figure 4.17 – (a) Exemple de deux points antipodaux A (en bleu) et B (en rouge) sur la surface de la sphère, (b) ensemble de points antipodaux répartis uniformément sur toute la surface de la sphère source selon la plage de colatitude θ (comprise entre 0 et π) et de longitude δ (comprise entre 0 et 2π)

La structure de recherche proposée se compose d'une sphère virtuelle et d'un cylindre virtuel rotatif centré sur la sphère virtuelle (voir la figure 4.18a). La sphère virtuelle est définie comme un ensemble de \mathcal{N} antipodes répartis uniformément sur la surface, selon la plage de colatitude θ (comprise entre θ_{min} et θ_{max}) et de longitude δ (comprise entre δ_{min} et δ_{max}) (voir la figure 4.18b). Le cylindre rotatif est défini par deux points antipodaux (antipodes) sur la sphère. Il tourne avec trois degrés de liberté (trois rotations R_x, R_y, R_z autour des axes x, y et z d'un repère aléatoire placé au centre de la sphère) à l'intérieur de la sphère virtuelle pour chaque antipode.

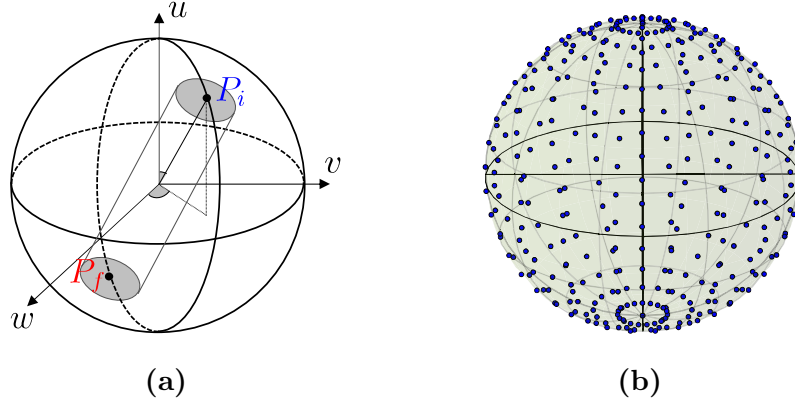


Figure 4.18 – Structure de recherche basée antipodes d'une sphère : (a) cylindre rotatif avec trois degrés de liberté (trois rotations R_u, R_v, R_w), (b) ensemble de points antipodaux répartis uniformément sur toute la surface de la sphère pour l'étape d'initialisation

Le pas de rotation du cylindre (ou le nombre d'antipodes \mathcal{N}) est défini à l'aide d'un espacement angulaire (ou pas angulaire) χ , qui définit la résolution de la structure de recherche.

Résolution de la structure de recherche : la sphère de recherche utilisée est définie comme un ensemble de \mathcal{N} antipodes répartis uniformément sur la surface, selon la plage de colatitude θ (comprise entre θ_{min} et θ_{max}) et de longitude δ (comprise entre δ_{min} et δ_{max}). Ces angles sont définis par rapport à une direction que nous appellerons par la suite *vecteur directeur de la sphère de recherche* (voir le vecteur bleu sur la figure 4.19).

Le nombre d'antipodes \mathcal{N} est défini à l'aide d'un espacement angulaire χ entre chaque antipode de la sphère de recherche (voir l'équation (4.3)).

$$\mathcal{N} = \frac{(\theta_{max} - \theta_{min}) \times (\delta_{max} - \delta_{min})}{\chi^2} \quad (4.3)$$

où $[\theta_{min}, \theta_{max}]$ représente la plage angulaire en colatitude θ , $[\delta_{min}, \delta_{max}]$ représente la plage angulaire en longitude δ et χ représente l'espacement angulaire (pas angulaire) qui permet de contrôler la résolution de la structure de recherche.

À partir de l'équation (4.3), on peut remarquer que l'espacement angulaire χ et le nombre d'antipodes \mathcal{N} sont inversement proportionnels au carré. Ainsi, lorsque

l'espacement angulaire χ est faible, le nombre d'antipodes \mathcal{N} est élevé, la résolution est plus grande et le temps d'exécution est plus élevé.

La résolution de notre structure de recherche est définie à l'aide de l'espacement angulaire χ . Cet espacement angulaire permet de définir le nombre de rotations du cylindre de recherche. La figure 4.19 montre certaines sphères de recherche générées avec différents degrés de résolution.

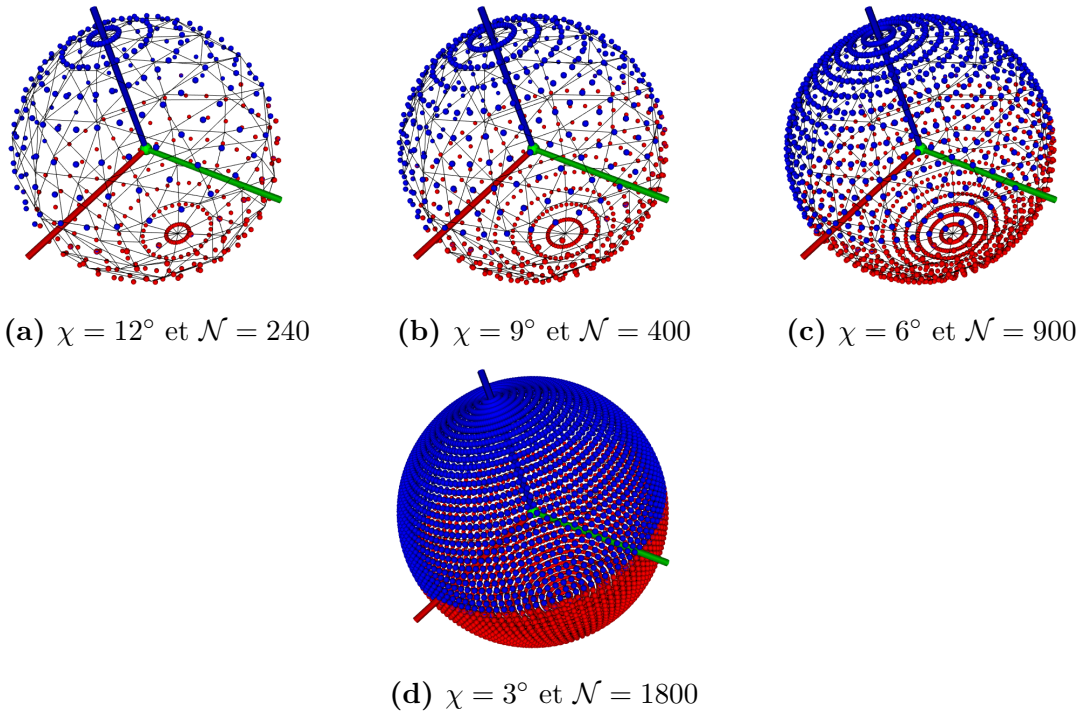


Figure 4.19 – Certaines sphères de recherche générées avec différentes valeurs d'espacement angulaire (les points bleus sont les antipodes des points rouges)

4.5.1.2 Algorithme de recherche

La structure de recherche est positionnée sur le point d'initialisation q_c décrit dans la section 4.4. Le cylindre virtuel rotatif de la structure de recherche tourne à l'intérieur de la sphère virtuelle avec trois degrés de liberté (trois rotations) pour chaque antipode. Pour chaque rotation, nous calculons le nombre de points à l'intérieur du cylindre virtuel rotatif. La position du cylindre ayant le plus grand nombre de points est utilisée pour définir le sous-cylindre initial s-cy₁.

Une fois le sous-cylindre initial s-cy₁ trouvé, nous allons ajuster ses paramètres pour qu'il s'ajuste au mieux sur le câble.

4.5.2 Ajustement des paramètres du premier sous-cylindre

Nous avons décidé de modéliser chaque sous-cylindre par trois paramètres : sa longueur, sa direction (son axe principal) et son rayon.

La figure 4.20 illustre le processus d'ajustement des paramètres du sous-cylindre.

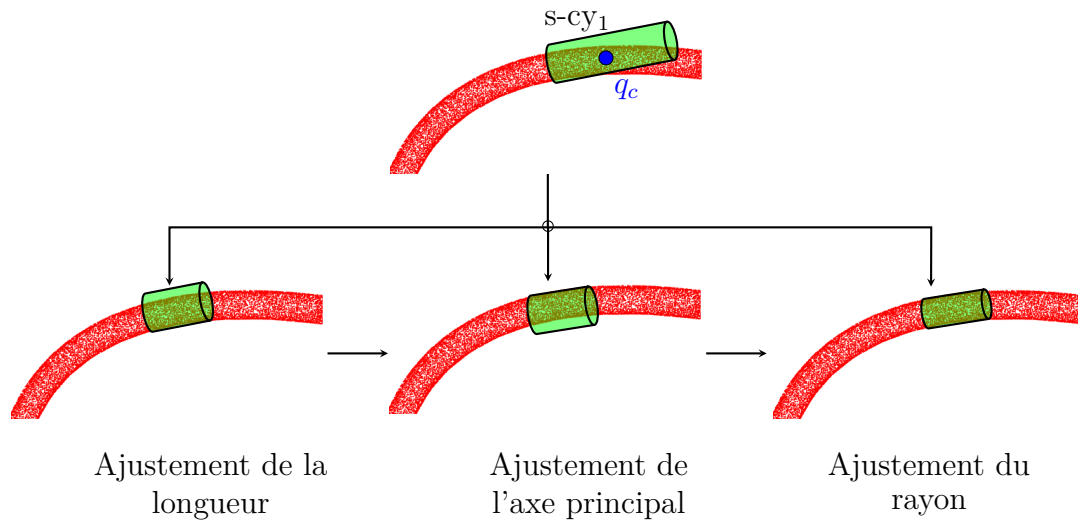


Figure 4.20 – Ajustement des paramètres du sous-cylindre

D'une façon générale, pour un ensemble de points, il s'agit de déterminer les paramètres du modèle, qui s'ajustent **au mieux** sur ces points. Dans notre cas précis, l'ajustement des paramètres du sous-cylindre se déroule en plusieurs étapes :

1. Ajustement de la longueur
2. Ajustement de l'axe principal
3. Ajustement du rayon

4.5.2.1 Ajustement de la longueur

En fonction de la courbure du câble, chaque sous-câble a une longueur différente (voir la figure 4.21).

Les longueurs des sous-cylindres (ou sous-câbles) sont calculées en projetant les points 3D situés à l'intérieur du cylindre de recherche sur son axe principal. Une fois que les points sont projetés, une pondération de ces points projetés permet d'estimer la longueur du sous-cylindre. Le processus d'ajustement de la longueur est illustré sur la figure 4.22.

On voit sur le graphique à barres (figure 4.22c) que le nombre des points par bin est très faible dans la zone courbée.

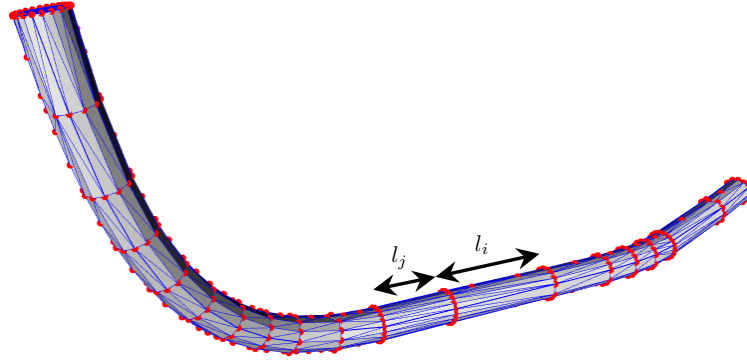


Figure 4.21 – Maillage triangulaire d'un modèle CAO du câble, les points rouges et les lignes bleues représentent respectivement les sommets et les arêtes du maillage. On voit que selon la courbure du câble, chaque sous-câble a une longueur différente (par exemple $l_i \neq l_j$)

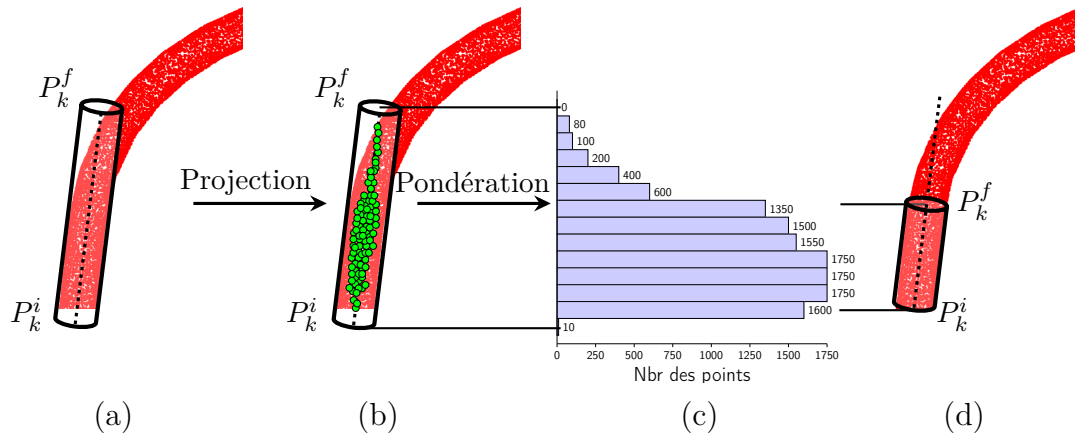


Figure 4.22 – Ajustement de la longueur : (a) cylindre ajusté sur le nuage de points correspondant au sous-câble, (b) projection du nuage de points situé à l'intérieur du cylindre de recherche sur l'axe principal, (c) graphique à barres indiquant le nombre des points par bin, et (d) sous-cylindre ajusté

Remarques :

1. Le centre du sous-cylindre final est égal au centre de masse du nuage de points situé à l'intérieur du cylindre de recherche.
2. Étant donné un nuage de points 3D, $p_i = [x_i, y_i, z_i]^T$ pour $i = 1 \dots n$, le centre de masse est calculé comme suit :

$$C = \frac{1}{n} \sum_{i=1}^N p_i \quad (4.4)$$

4.5.2.2 Ajustement de l'axe principal

Pour ajuster l'axe principal du modèle sous-cylindre, il suffit d'ajuster ce modèle sur le nuage de points du sous-câble. Pour ce faire, nous faisons appel à un algorithme d'ajustement de type ICP point à point [Bes+92].

Afin de pouvoir ajuster le modèle du sous-cylindre (cylindre vert sur la figure 4.23a) avec le nuage de points du sous-câble (les points rouges sur la figure 4.23a), nous créons un nuage de points virtuel du modèle de sous-cylindre. De plus, dans le but d'obtenir un ajustement précis, ce nuage de points virtuel est créé avec la même résolution que le nuage de points du sous-câble (voir la figure 4.23b).

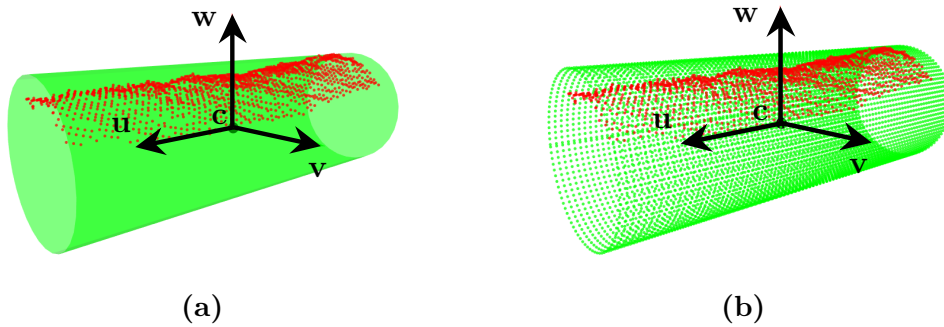


Figure 4.23 – Création d'un nuage de points virtuel à partir du modèle de sous-cylindre : (a) le modèle de sous-cylindre (en vert) et le nuage de points du sous-câble (en rouge), (b) le nuage de points virtuel (en vert) créé avec la même résolution que le nuage du sous-câble (en rouge)

L'algorithme ICP point à point est une méthode itérative d'alignement 3D. Cet algorithme consiste à calculer la transformation alignant au mieux deux ensembles de données 3D de façon itérative. En outre, l'objectif est de minimiser itérativement la distance entre les deux nuages de points considérés. Au bout de chaque itération, l'algorithme construit une liste de points appariés en établissant une correspondance point à point au moyen d'une distance euclidienne. Puis, il estime la transformation de repère entre les points appariés qui minimise une mesure d'erreur quadratique moyenne définie entre eux. La convergence est atteinte lorsque l'erreur résiduelle de distance entre les points appariés est inférieure à un certain seuil ou en fixant un nombre d'itérations nécessaire. Il est donc nécessaire de déterminer le nombre d'itérations de l'algorithme. Pour cela, nous allons utiliser l'erreur quadratique moyenne de la distance entre les points réel du sous-câble et les points virtuels du modèle de sous-cylindre. Cette erreur quadratique moyenne (*mean squared erreur* (MSE_d) en anglais) est définie par :

$$MSE_d = \frac{1}{n} \sum_{i=1}^n (d_i)^2 \quad (4.5)$$

où d_i représente la distance euclidienne entre un point p_i du nuage réel et son plus proche voisin q_i du nuage virtuel (voir la figure 4.24)

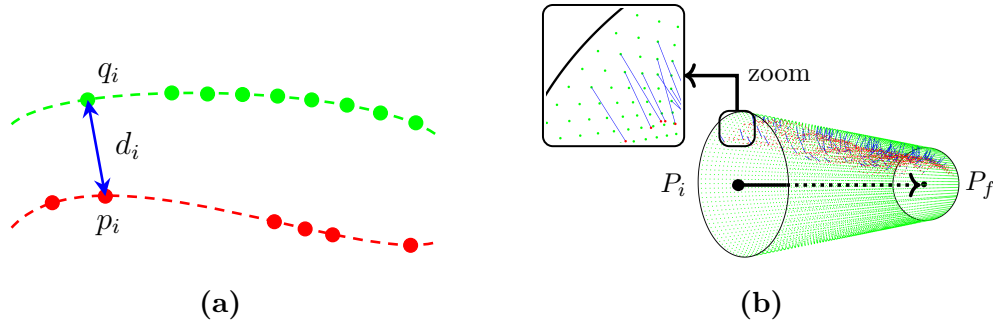


Figure 4.24 – Calcul de la distance d_i (ligne bleue) entre un point p_i du sous-câble (en rouge) et son plus proche voisin q_i de sous-cylindre (en vert)

Remarques :

1. L'algorithme itératif ICP converge toujours de façon monotone vers un minimum local par rapport à la fonction d'erreur quadratique moyenne.
2. Pour plus d'informations sur l'algorithme ICP et ses variantes, nous invitons le lecteur à se référer au mémoire de HADDEJI [Had13].

Afin de déterminer le nombre d'itérations nécessaire pour la convergence de l'algorithme ICP point à point, nous avons calculé et comparé pour chaque itération, les valeurs de l'erreur quadratique moyenne (MSE_d , défini à l'équation (4.5)).

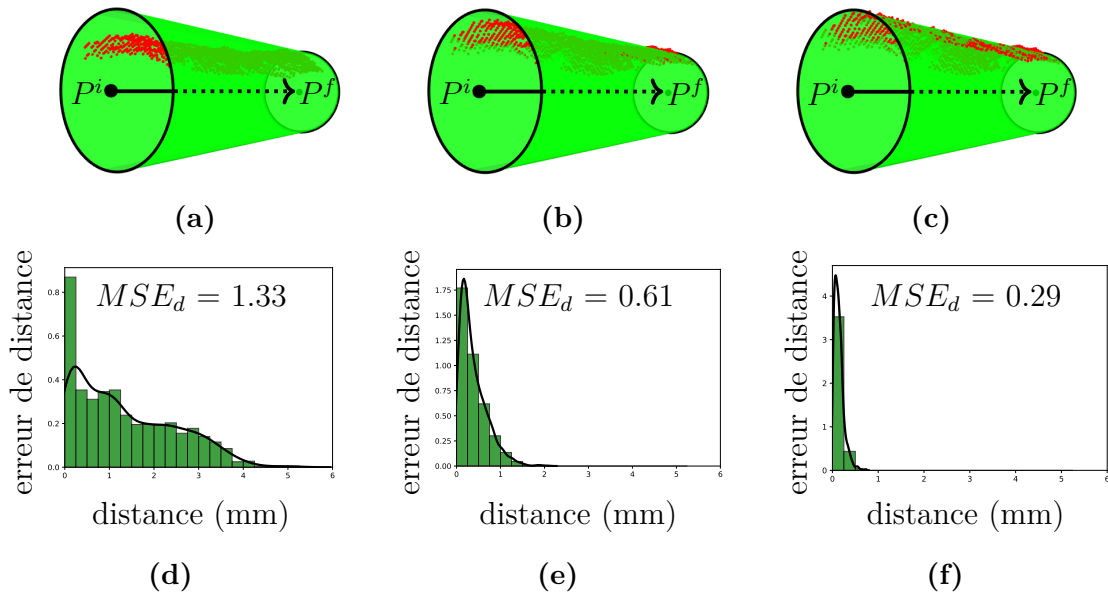


Figure 4.25 – Évaluation de l'erreur quadratique moyenne de la distance MSE_d en fonction du nombre d'itérations entre le sous-cylindre (s-cy₁ en vert) et le sous-câble (s-c₁ en rouge) : (a) après 1 itération, (b) après 5 itérations, (c) après 10 itérations. La deuxième ligne fournit les histogrammes d'erreurs de distance : (d) après une 1 itération, (e) après 5 itérations, (f) après 10 itérations

Comme attendu, les différentes erreurs mesurées dans cet exemple diminuent en fonction du nombre d'itérations. De plus, comme le montrent les histogrammes d'erreurs de distance (figures 4.25d, 4.25e et 4.25f), l'erreur est diminuée de moitié à chaque itération et, au bout de 10 itérations, l'erreur MSE_d converge vers une valeur proche de zéro (voir la figure 4.25f).

L'algorithme ICP point à point montre une convergence très rapide ($\simeq 6$ ms pour un nuage de 3500 points) durant les quelques premières itérations. Ceci s'explique par le fait que notre algorithme de recherche (présenté dans la section 4.5.1.1) nous donne un nuage de points segmenté (sous-câble) très proche de nuage de son modèle (sous-cylindre). Autrement dit, les points du sous-câble sont toujours à l'intérieur du sous-cylindre. Par conséquent, seuls les points pertinents (aussi appelées « inliers ») sont considérés comme des entrées pour l'algorithme d'ICP, ce qui donne un alignement précis avec une vitesse de convergence rapide, en un nombre minimum d'itérations.

Après la réalisation de plusieurs tests similaires à ce dernier sur d'autres nuages de points, nous avons remarqué que l'évolution de l'erreur devient insignifiante après un nombre d'itérations moyen égale à 10. Dans ce cas, le changement d'erreur devient de l'ordre de 10^{-6} et les résultats obtenus répondent à nos attentes. Par conséquent, le critère d'arrêt a été fixé par la combinaison de deux conditions :

- Un nombre d'itérations maximal égal à 10
- Une précision de l'erreur égale à 10^{-6}

Autres méthodes possibles pour ajuster l'axe principal du cylindre :

1. nous avons initialement développé une méthode qui consiste à utiliser l'estimation des normales à la surface en chaque point. En effet, si un point est situé sur un cylindre, sa normale est nécessairement perpendiculaire à l'axe principal du cylindre. Ainsi, pour trouver l'axe principal, il suffit de prendre l'axe perpendiculaire à la normale moyenne des vecteurs normaux.

Nous avons comparé cette méthode à la méthode ICP point à point (avec 5 itérations) afin de déterminer laquelle est la plus précise. La métrique de comparaison utilisée est l'erreur MSE_d (définie par l'équation (4.5)).

Sur la figure 4.26, nous fournissons une comparaison quantitative entre ces deux méthodes sur un exemple de données réelles.

Étant donné que l'estimation des normales est principalement basée sur l'utilisation des voisinages locaux, elle est donc sensible aux variations de densité de points, où certaines zones peuvent contenir un grand nombre de points, alors que d'autres très peu (comme on peut le voir sur la figure figure 4.26a).

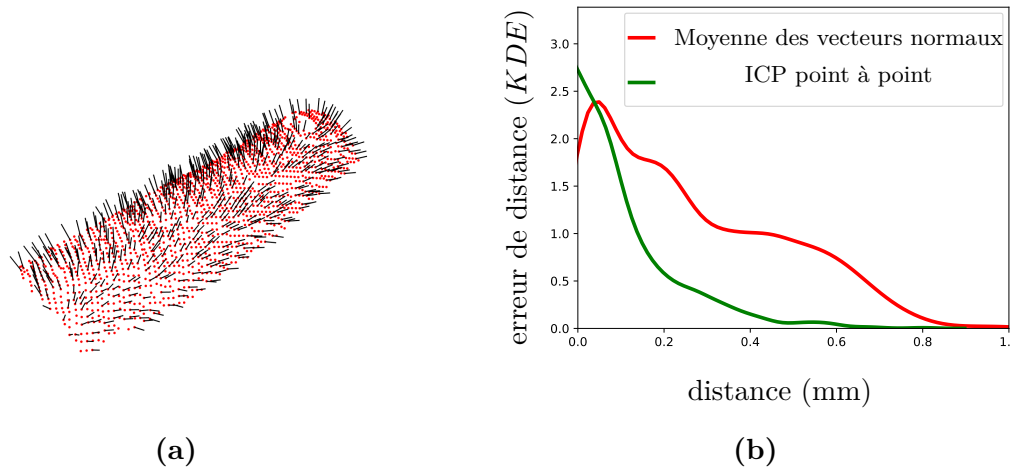


Figure 4.26 – Comparaison quantitative entre la méthode utilisée (ICP point à point avec 5 itérations) et la méthode de calcul de la moyenne des vecteurs normaux. (a) exemple des normales estimées (lignes noires) sur un sous-câble (nuage de points en rouge), (b) estimation par noyau de l’erreur de distance entre les deux méthodes

La figure 4.26b montre que la méthode ICP point à point est plus performante que la méthode basée sur la moyenne des vecteurs normaux. Sur cette figure, *KDE* signifie *Kernel Density Estimation*. C’est une méthode qui permet de créer une courbe continue à partir d’un histogramme de données.

2. une autre méthode pour ajuster l’axe principal du cylindre a été proposée dans la thèse de CHAPERON [Cha02], et récemment utilisée par KAWASHIMA et al. [Kaw+14] et par QIU et al. [Qiu+14]. Cette méthode est utilisée pour la segmentation de lignes de tuyauterie. Elle se base principalement sur l’utilisation des normales, de la sphère de Gauss (appelée *sphère unité*) et de l’image de Gauss (représentation des normales sur la sphère de Gauss [Car76 ; Cha02]). Dans ces travaux, les auteurs ont remarqué que l’image de Gauss d’un cylindre est un grand cercle sur la sphère de Gauss. De plus, ce cercle représente l’intersection entre la sphère de Gauss et un plan passant par son centre. Par conséquent, la normale de ce plan correspond exactement à l’axe principal du cylindre. Enfin, il suffit d’ajuster un plan sur les points de l’image de Gauss pour trouver l’axe principal du cylindre.

Nous n’avons pas testé cette méthode mais il est fort probable qu’elle devrait fournir de moins bons résultats sur nos données que la méthode ICP point à point, étant donné qu’elle exploite les normales à la surface.

4.5.2.3 Ajustement du rayon

Nous estimons le rayon du câble \tilde{r} en calculant la distance moyenne de tous les points 3D (à l’intérieur du cylindre) par rapport à l’axe principal, par la formule suivante :

$$\tilde{r} = \frac{1}{N} \sum_{i=1}^n d_i \quad (4.6)$$

où d_i représente la distance entre un point p_i et sa projection orthogonale p_i^* sur l'axe principal du cylindre (voir la figure 4.27).

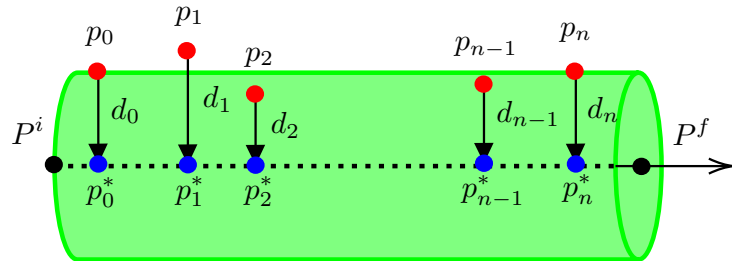


Figure 4.27 – Ajustement du rayon

Un autre moyen pour estimer le rayon du cylindre présenté dans les travaux [Cha02 ; Lal+06 ; Rau+13] consiste à projeter les points sur un plan perpendiculaire à l'axe du cylindre, puis à ajuster un cercle sur les points projetés au sens des moindres carrés ou par la méthode de Taubin [Tau91] (voir la figure 4.28). Nous avons testé cette méthode et elle s'est avérée moins robuste que celle que nous avons mise en place. En effet, dans notre cas, les nuages de points ont été acquis par une méthode de type *one-shot* (une seule acquisition depuis un seul point de vue). Par conséquent, les données acquises sont incomplètes et parfois localement manquantes, ce qui rend l'ajustement du cercle inefficace.

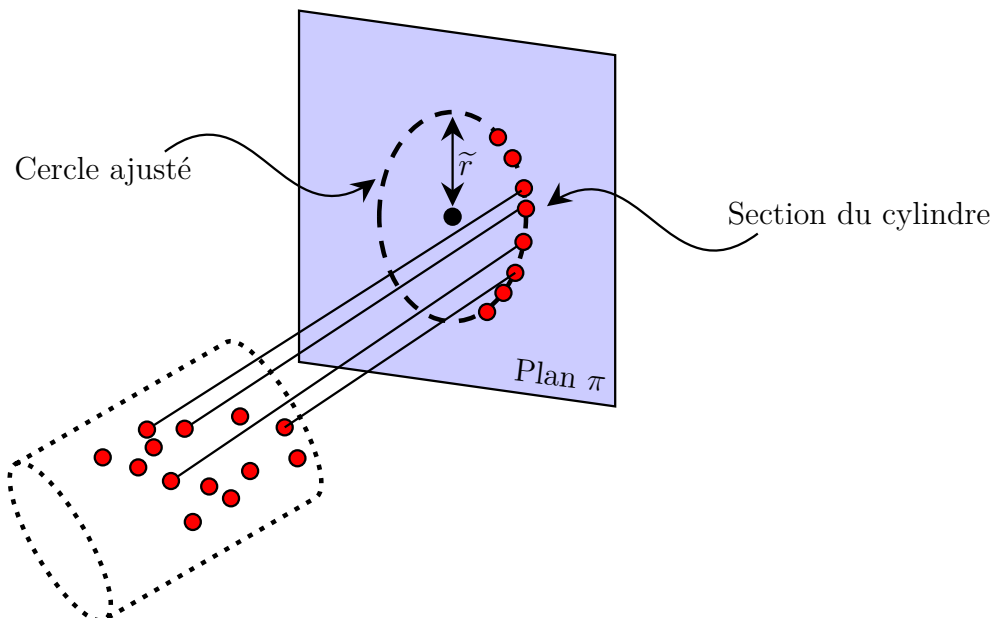


Figure 4.28 – Une autre méthode pour estimer le rayon du cylindre. Si l'on projette les points rouges sur un plan π perpendiculaire à l'axe du cylindre, on obtient la section du cylindre, c'est-à-dire un cercle

4.5.3 Propagation de la recherche/ajustement des sous-cylindres

Une fois le sous-cylindre $s\text{-}cy_1$ ajusté sur le nuage de points réel du sous-câble, le processus de recherche/ajustement des sous-cylindres est étendu de façon itérative par propagation sur le câble dans les deux sens (cf. figure 4.16 et figure 4.29).

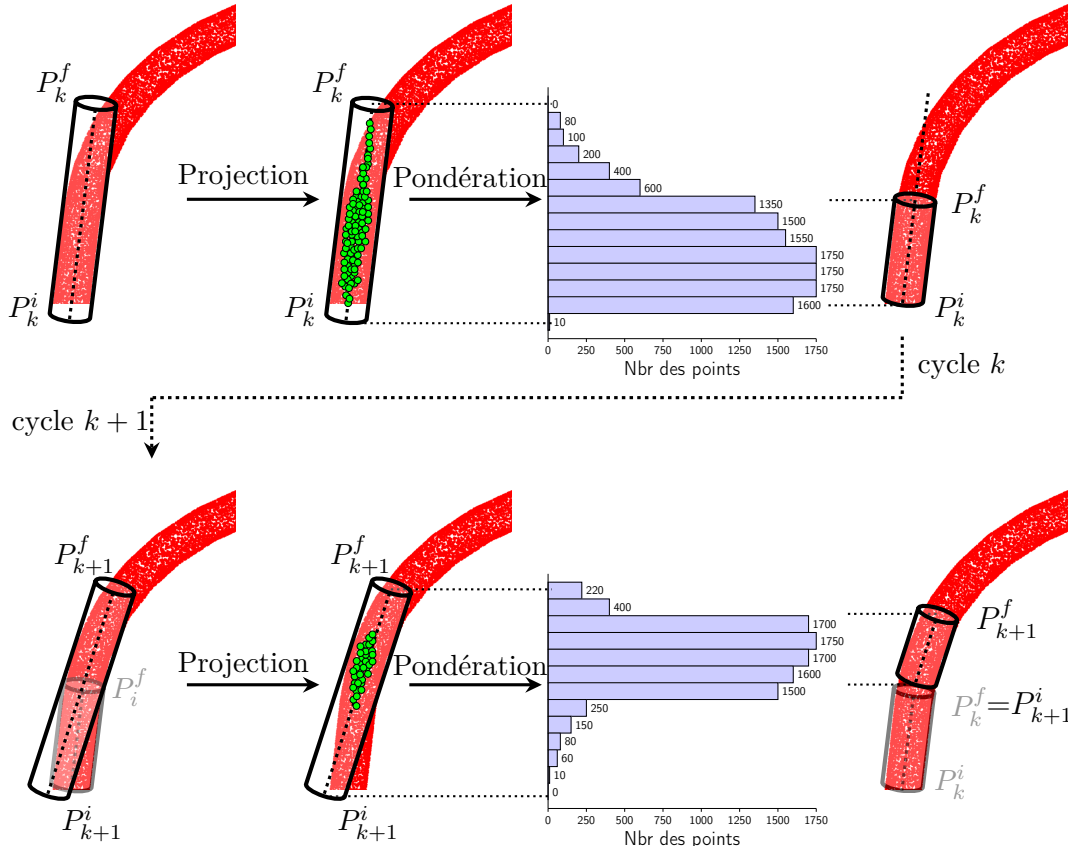


Figure 4.29 – Propagation du cycle k au cycle $k + 1$: (1^{ère} colonne) cylindre ajusté sur le nuage de points correspondant au sous-câble, (2^{ème} colonne) projection du nuage de points situé à l'intérieur du cylindre de recherche sur l'axe principal, (3^{ème} colonne) graphique à barres indiquant le nombre de points par bin, et (4^{ème} colonne) sous-cylindre ajusté

Remarques :

1. l'axe du sous-cylindre $s\text{-}cy_{k+1}$ est contraint à passer par l'extrémité du sous-cylindre $s\text{-}cy_k$. Autrement dit, le point initial P_{k+1}^i du cylindre $s\text{-}cy_{k+1}$ coïncide avec le point final P_k^f du sous-cylindre $s\text{-}cy_k$. Seule la direction de l'axe est autorisée à varier (principe de la liaison rotule, voir la figure 4.30).

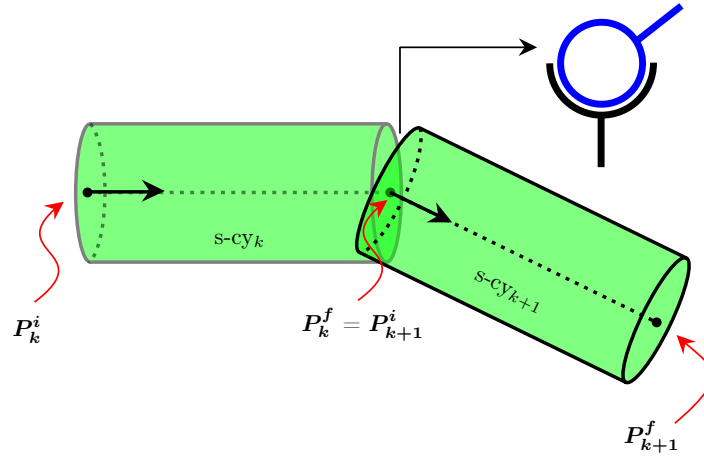


Figure 4.30 – Liaison rotule entre deux sous-cylindres consécutifs ($s\text{-cy}_k$ et $s\text{-cy}_{k+1}$)

2. Un câble ne peut pas se plier au-delà de 90° . Cette hypothèse permet de réduire le nombre d'antipodes \mathcal{N} sur la sphère et, par conséquent, le temps de calcul. Pour cela, une fois le sous-cylindre initial (noté $s\text{-cy}_1$) construit (après la première itération), le vecteur directeur de la sphère de recherche est initialisé au vecteur directeur du cylindre précédent u_0 (voir la figure 4.31). Ceci fournit une initialisation valable dans le cas où l'angle qui fait le sous-cylindre $s\text{-cy}_{k+1}$ avec le sous-cylindre précédent $s\text{-cy}_k$ n'est pas trop élevé.

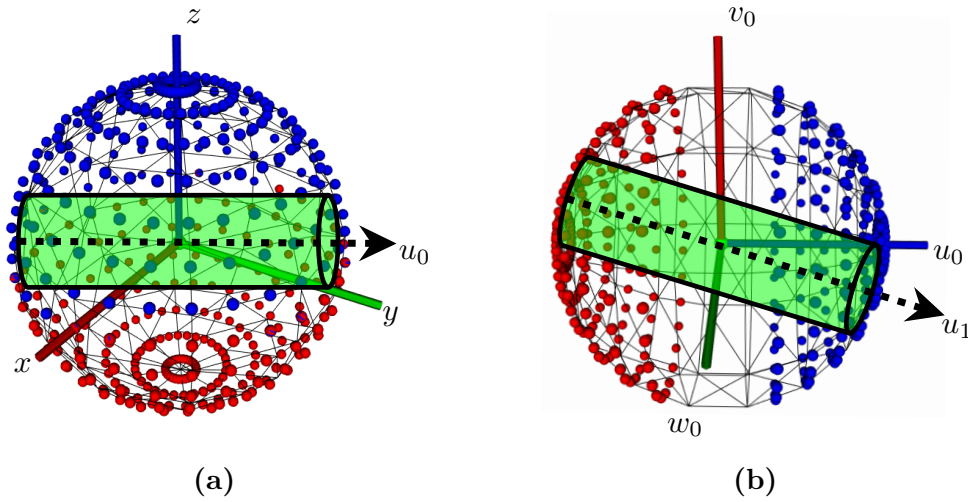


Figure 4.31 – (a) Structure de recherche utilisée pour construire le sous-cylindre initial s_{c1} , (b) structure de recherche simplifiée après la première itération. Le vecteur directeur de la sphère de recherche de la deuxième itération est initialisé au vecteur directeur du cylindre de la première itération (u_0)

3. l'hypothèse que le câble ne peut pas se plier au-delà de 90° permet de limiter la segmentation aux câbles dont le rayon de courbure maximal est inférieur

à 90°. Cependant, l'utilisateur a la possibilité de désactiver cette option pour segmenter les câbles dont la courbure dépasse 90°. Cette hypothèse a pour but de simplifier la structure de recherche, et donc de réduire le temps de calcul.

Critères d'arrêt de la propagation :

Les itérations s'arrêtent selon les trois critères d'arrêt suivants :

1. il n'y a pas assez de points dans le cylindre virtuel rotatif, ou/et
2. il y a un très grand nombre de points dans la sphère de recherche.

↪ Ces deux critères sont définis en fonction de la résolution du nuage de points (\mathcal{R}_p) et de la forme géométrique des caractéristiques géométriques du cylindre rotatif et de la sphère virtuelle.

3. l'erreur quadratique moyenne de la distance entre le sous-cylindre courant s-cy_k et le sous-câble s-c_k est supérieure à 3 mm.

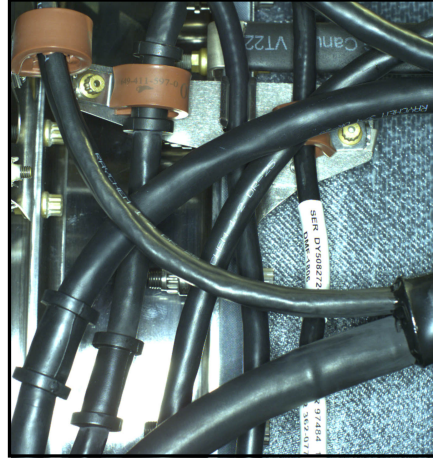
Un exemple de résultat :

La figure 4.32 montre un exemple de résultat de modélisation et de segmentation obtenu sur un câble courbé dans une scène complexe.

Comme le montre la figure 4.32b, un trou dû à un chevauchement avec un autre câble a été détecté. Pour cela, nous avons mis en œuvre le mécanisme suivant : lorsque les conditions d'arrêt de la propagation sont atteintes, nous essayons de poursuivre la recherche dans une direction calculée à partir de la direction précédente (liaison rotule) sur une distance égale au plus grand diamètre des câbles présents dans la maquette numérique. Si cette recherche est fructueuse, la propagation continue après le trou détecté. Si la recherche échoue, soit nous arrêtons définitivement la propagation, soit nous lançons une propagation dans le sens opposé si ce sens n'a pas déjà été testé.

À titre illustratif, la figure 4.33 illustre les résultats de propagation précédemment obtenus à partir des deux extrémités du sous-cylindre initial s-cy₁ sous forme d'une succession (ou enchaînement) de 13 sous-cylindres consécutifs, étiquetés s-cy₁ à s-cy₁₃. Le tableau 4.2 montre les détails de ces 13 sous-cylindres : leur rayon, leur longueur, l'angle θ_k entre deux axes principaux consécutifs et le sens de propagation, où « + » représente le premier sens et « - » représente le sens opposé. L'angle θ_k entre deux axes consécutifs (\vec{v}_k et \vec{v}_{k+1}) est calculé en considérant chaque axe principal comme un vecteur. Ensuite, cet angle s'obtient par le produit scalaire et la norme des vecteurs :

$$\cos(\theta_k) = \frac{\vec{v}_k \cdot \vec{v}_{k+1}}{\|\vec{v}_k\| \cdot \|\vec{v}_{k+1}\|} \quad (4.7)$$



(a)



(b)



(c)

Figure 4.32 – Exemple de résultat : (a) image 2D de la scène, (b) résultat de modélisation du câble, composé de 13 sous-cylindres (12 sous-cylindres en vert et un trou détecté dû à un chevauchement avec un autre câble occultant en bleu), (c) résultat de segmentation du câble (le nuage de points segmenté en vert)

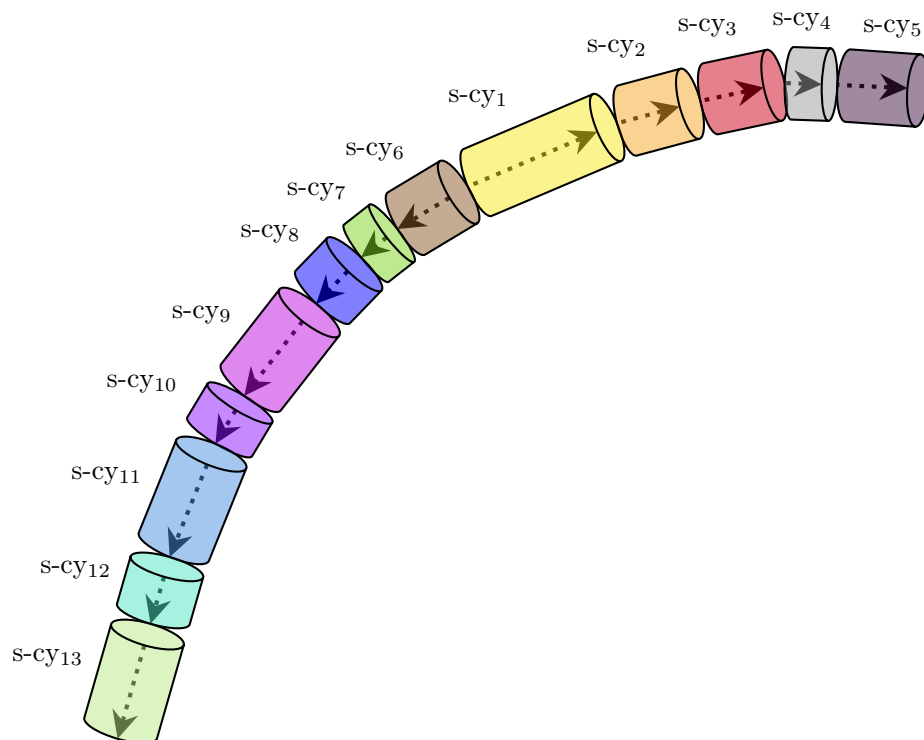


Figure 4.33 – Illustration des résultats de propagation obtenus dans l'exemple de la figure 4.32 par une succession (ou enchaînement) de 13 sous-cylindres consécutifs, étiquetés s-cy₁ à s-cy₁₃ (les différentes couleurs représentent les différents sous-cylindres)

Table 4.2 – Les détails des 13 sous-cylindres : leur rayon, leur longueur, l’angle entre deux axes principaux consécutifs et le sens de propagation, où « + » représente le premier sens et « - » représente le sens opposé. Il est à noter que s-cy₈ est un trou détecté dû à un chevauchement avec un autre câble occultant (voir la figure 4.32) et son rayon est égal au rayon du sous-cylindre précédent s-cy₇.

Sous-cylindre	Rayon	Longueur	Angle	Sens
s-cy ₁	5.81	33.55	–	+
s-cy ₂	5.93	22.09	8.51	+
s-cy ₃	5.89	17.83	2.12	+
s-cy ₄	5.97	17.97	11.4	+
s-cy ₅	5.91	19.26	3.7	+
s-cy ₆	5.90	16.43	4.12	-
s-cy ₇	5.91	10.01	2.02	-
s-cy ₈	–	13.20	3.22	-
s-cy ₉	5.93	22.01	1.7	-
s-cy ₁₀	5.98	10.66	1.1	-
s-cy ₁₁	5.92	19.44	4.3	-
s-cy ₁₂	5.79	11.81	12.3	-
s-cy ₁₃	5.80	18.02	11.2	-

4.6 Résultats expérimentaux

Afin de donner un large aperçu des résultats expérimentaux de notre méthode de segmentation, présentée dans ce chapitre, nous avons procédé à des expérimentations et évaluations exhaustives sur des données synthétiques et réelles issues de bases de données variées constituées par DIOTA.

Cette section sera consacrée à l’évaluation de l’approche de segmentation proposée. La première partie de l’évaluation s’intéressera aux résultats produits par la première partie de l’approche (ajustement des paramètres du sous-cylindre présenté dans la section 4.5.2) tandis que la seconde partie se penchera sur les résultats finaux de la modélisation-segmentation des câbles dans une scène complexe.

Nous présenterons d’abord les conditions expérimentales dans lesquelles les expérimentations ont été réalisées (cf. section 4.6.1). Ensuite, nous présenterons les données utilisées, notamment les modèles CAO 3D et les nuages de points synthétiques et réels testés (cf. section 4.6.2). Nous présenterons ensuite à la section 4.6.3 une évaluation de la performance du processus d’ajustement des paramètres à travers un comparaisn avec les trois méthodes d’ajustement les plus populaires : (i) RANSAC (*RANdom SAmple Consensus*) [Sch+07], (ii) MLESAC (*Maximum Likelihood Estimator SAmple Consensus*, une version améliorée de RANSAC) [Tor+00] et (iii) une méthode basée sur l’Analyse en Composantes Principales (ACP) [Lal+06]. Enfin, nous présenterons une évaluation de la performance de la méthode de segmentation avec des données synthétiques (cf. section 4.6.4) et réelles (cf. section 4.6.5).

4.6.1 Conditions expérimentales

L'implémentation de l'algorithme de segmentation présenté dans ce chapitre est effectuée en C++ et en Python en utilisant la bibliothèque *Point Cloud Library* (*PCL*). La méthode de segmentation proposée a été évaluée en utilisant une machine ayant un processeur Quad-core i7-6700 cadencé à 2.6 Ghz et équipé de 16 Go de RAM.

Dans les expériences, les nuages de points ont été acquis en utilisant le scanner 3D Ensenso N35 porté sur un robot mobile. Ce scanner fonctionne selon le principe de la stéréovision. Ce scanner est composé de deux caméras CMOS et d'un projecteur intégré, créant un motif aléatoire de points sur l'objet à saisir. En outre, cette technologie contient un système actif avec encodage spatial permettant la capture « *one-shot* ». Les caractéristiques techniques du scanner 3D sont présentées dans la section 1.5.2.

4.6.2 Présentation des données

Cette section présente les entrées principales de notre approche. Nous introduisons dans un premier temps les données utilisées au cours de nos travaux notamment les modèles CAO 3D des assemblages mécaniques et les nuages de points synthétiques et réels (acquis par le scanner).

4.6.2.1 Modèles CAO disponibles

Le modèle CAO de l'assemblage mécanique, duquel on peut extraire le modèle CAO des câbles, est disponible. La figure 4.34 montre un exemple d'assemblage mécanique utilisé dans les résultats expérimentaux :

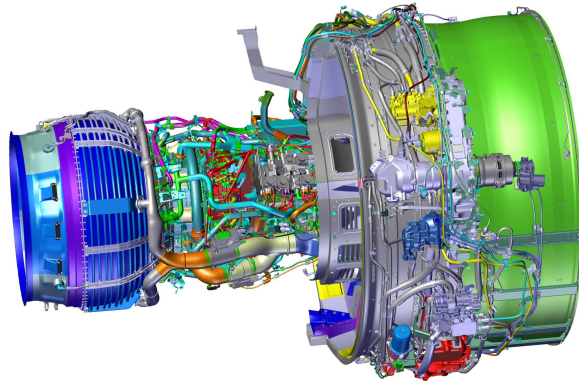
- Moteur d'avion (figure 4.34a) : un moteur d'avion LEAP (Leading Edge Aviation Propulsion) de la famille CFM56, qui se compose de plusieurs câbles à contrôler.
- Une partie extraite du moteur LEAP contenant 20 câbles à contrôler (figure 4.34b).

4.6.2.2 Nuages de points synthétiques simulés

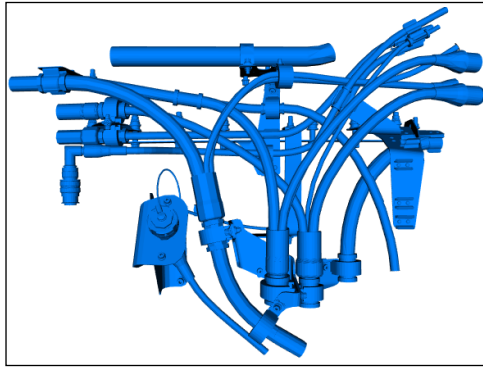
Afin d'enrichir la base de données, nous avons utilisé la méthode d'échantillonnage uniforme pondéré, présentée dans la section 4.3.2 pour générer différents nuages de points synthétiques. Pour examiner l'effet des variations de densité de points, nous avons généré ces nuages de points avec des résolutions différentes (voir la figure 4.35).

Avec notre méthode d'échantillonnage pondéré uniforme, nous pouvons représenter (simuler) n'importe quel modèle 3D CAO par des nuages de points avec des résolutions différentes.

D'autres données synthétiques seront présentées par la suite, dans la section 4.6.3 et la section 4.6.4.



(a)



(b)

Figure 4.34 – Exemple d'un assemblage mécanique utilisé : (a) moteur LEAP de la famille CFM56, (c) une partie extraite du moteur LEAP contenant 20 câbles différents

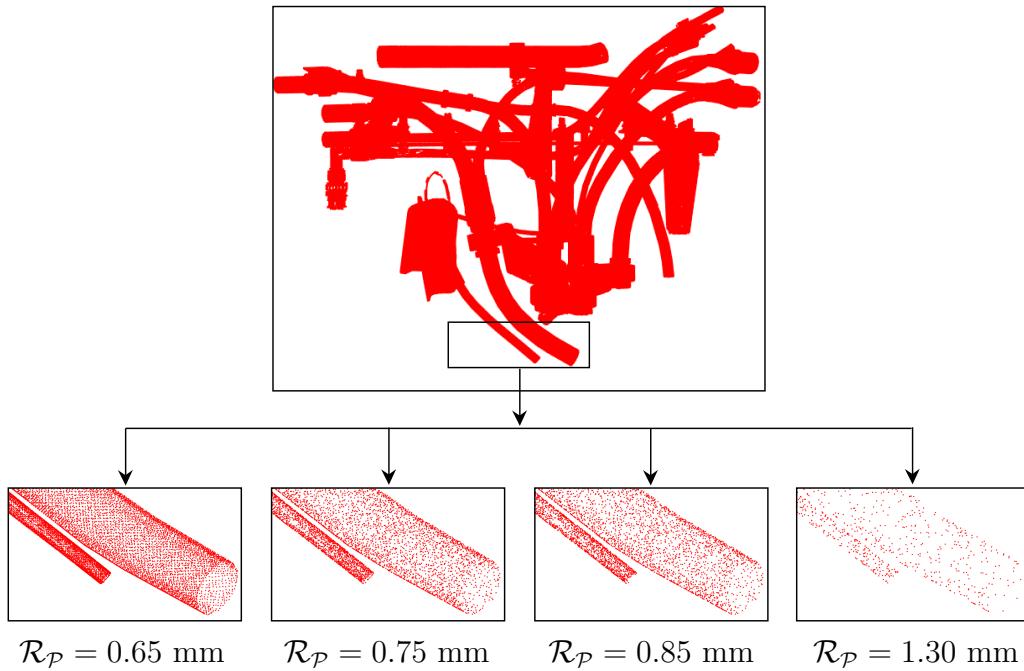


Figure 4.35 – Exemples de nuages de points synthétiques générés avec différentes résolutions

4.6.2.3 Nuages de points réels acquis par le scanner

Nous disposons des nuages de points réels du moteur LEAP (figures 4.34a et 4.34b), colorisés avec une information d'intensité. La méthode de segmentation proposée n'exige que l'information de profondeur. Par conséquent, l'information de couleur n'a pas été prise en compte.

Les nuages de points ont été acquis par une méthode de type *one-shot* (une seule acquisition depuis un seul point de vue). Par conséquent, les données acquises sont incomplètes et parfois localement manquantes.

Conformément aux normes de sécurité industrielle, pour assurer la sécurité de l'élément inspecté ainsi que celle du robot d'inspection, la distance entre le scanner 3D et l'élément à contrôler est fixé à 600 mm avec une tolérance de ± 6 mm.

Avant filtrage, ces nuages de points contiennent en moyenne 131200 points pour une surface numérisée de 520 cm² (26 cm \times 26 cm) et une résolution spatiale de 0.45 mm (voir un exemple sur la figure 4.36).

D'autres données réelles seront présentées par la suite, dans la section 4.6.5.

4.6.3 Évaluation du processus d'ajustement des paramètres

Afin d'évaluer la robustesse du processus d'ajustement des paramètres présenté dans la section 4.5.2, nous avons procédé à des expérimentations et évaluations exhaustives sur différents nuages de points synthétiques. En vue de valider le processus d'ajustement, nous avons dans un premier temps créé à partir du modèle CAO différents nuages de points représentant différents cylindres (différents rayons, longueurs, orientations et positions). Pour examiner l'effet des variations de densité de points, nous avons généré ces nuages de points avec des résolutions différentes à l'aide de la méthode d'échantillonnage uniforme pondéré, présentée dans la section 4.3.2. De plus, pour étudier la robustesse de la méthode d'ajustement proposée vis-à-vis du bruit et des outliers, les points du nuage ont été entachés d'une erreur Gaussienne de moyenne ($\mu = 0$) et de différentes valeurs d'écart-type (σ), qui définissent le niveau de bruit dans toutes les directions (x , y et z) du nuage de points.

Ensuite, notre méthode d'ajustement est comparée aux trois méthodes d'ajustement classiques les plus populaires : (i) RANSAC (*RANdom SAmple Consensus*) [Sch+07], (ii) MLESAC (*Maximum Likelihood Estimator SAmple Consensus*, une version améliorée de RANSAC) [Tor+00] et (iii) une méthode basée sur l'Analyse en Composantes Principales (ACP) [Lal+06].

4.6.3.1 Métriques d'évaluation

Pour évaluer notre algorithme de modélisation-segmentation, nous avons mis en place quatre mesures comparatives résumant les performances de l'algorithme :

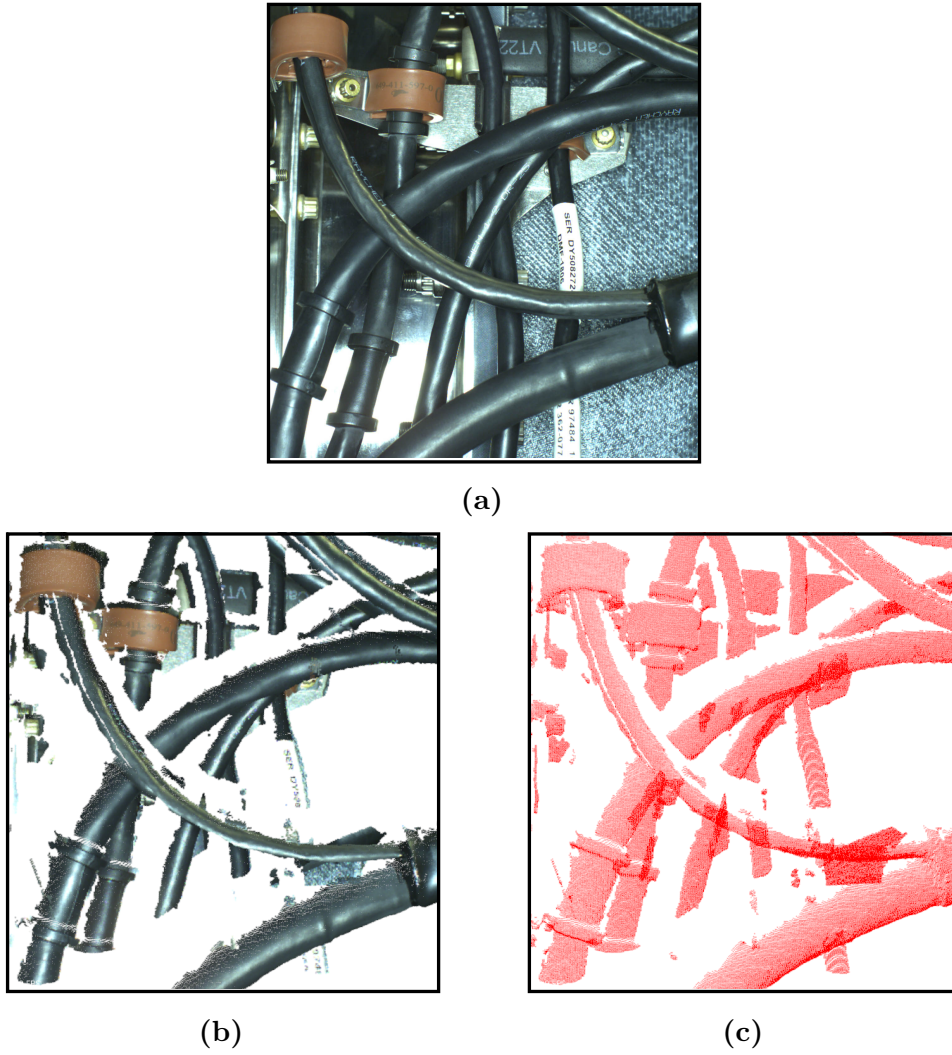


Figure 4.36 – Exemple d'un nuage de points réel acquis par le scanner 3D : (a) image 2D de la scène, (b) nuage de points coloré (avec information de couleur), (c) nuage de points sans information de couleur

1. La différence entre le rayon réel de la vérité terrain r et le rayon estimé \tilde{r} :

$$MSE_{\text{rayon}} = \frac{1}{n} \sum_{i=1}^n (r_i - \tilde{r}_i)^2 \quad (4.8)$$

2. La différence entre la longueur de la vérité terrain h et la longueur estimée \tilde{h} :

$$MSE_{\text{longueur}} = \frac{1}{n} \sum_{i=1}^n (h_i - \tilde{h}_i)^2 \quad (4.9)$$

3. L'angle entre l'axe principal de la vérité terrain et l'axe estimé :

$$MSE_{\text{orientation}} = \frac{1}{n} \sum_{i=1}^n (\theta_i)^2 \quad (4.10)$$

4. La distance euclidienne entre le centre de la vérité terrain c et le centre estimé \tilde{c} :

$$MSE_{\text{centre}} = \frac{1}{n} \sum_{i=1}^n \|c_i - \tilde{c}_i\|_2 \quad (4.11)$$

où n représente le nombre de nuages de points de l'ensemble des données.

Remarque : comme illustré sur la figure 4.37, l'angle entre l'axe principal de la vérité terrain v et l'axe estimé \tilde{v} est défini comme l'écart angulaire entre ces deux axes (voir équation (4.12)).

$$\theta = \arccos|v \cdot \tilde{v}| \quad (4.12)$$

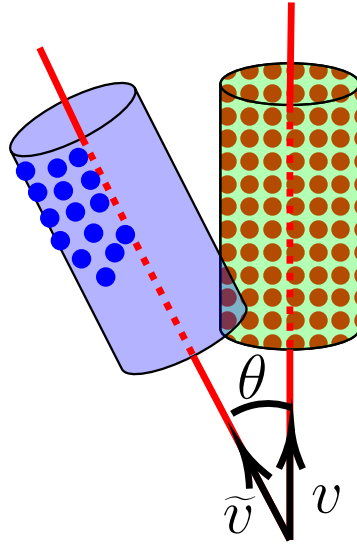


Figure 4.37 – Mesure de l'angle entre l'axe principal v du cylindre connu (cylindre vert) et l'axe principal \tilde{v} d'un faux cylindre estimé (cylindre bleu). Le cylindre vert représente le cylindre connu (vérité terrain) qui correspond aux vrais points 3D (points rouges). Le cylindre bleu représente un faux cylindre qui a été estimé par erreur à partir de points 3D qui correspondent à des points de bruit (points bleus).

4.6.3.2 Comparaison avec les méthodes d'ajustement classiques les plus populaires

Dans cette section, notre méthode d'ajustement est comparée aux trois méthodes d'ajustement classiques les plus populaires : (i) RANSAC (*RANdom SAmple Consensus*) [Sch+07], (ii) MLESAC (*Maximum Likelihood Estimator SAmple Consensus*,

une version améliorée de RANSAC) [Tor+00] et (iii) une méthode basée sur l'Analyse en Composantes Principales (ACP) [Lal+06].

La figure 4.38 montre quelques exemples de résultats d'ajustement obtenus avec notre méthode sur différents nuages de points avec des informations incomplètes, manquantes, bruitées et en présence de valeurs extrêmes.

Le tableau 4.3 montre les résultats de comparaison obtenus sur 20 nuages de points différents avec des résolutions différentes.

- Les points du nuage ont été entachés d'une erreur Gaussienne avec 5 valeurs différentes d'écart-type ($\sigma \in [0.4, 1.4]$).
- Étant donné que la génération du nuage de points bruité est aléatoire, les expériences ont été répétées 5 fois. Par conséquent, les valeurs des métriques présentées dans le tableau 4.3 sont obtenues en faisant la moyenne de ces valeurs.

Les résultats ont donc été obtenus à partir de 25 essais (5×5) pour chaque nuage de points.

Table 4.3 – Résultats de comparaison entre la méthode proposée et les méthodes d'ajustement classiques : RANSAC [Sch+07], MLESAC [Tor+00] et la méthode basée sur l'Analyse en Composantes Principales (ACP) [Lal+06]

Méthode	MSE_{rayon}	MSE_{longueur}	$MSE_{\text{orientation}}$	MSE_{centre}
RANSAC [Sch+07]	0.19	2.66	0.12	2.47
MLESAC [Tor+00]	0.22	1.16	0.28	1.1
ACP [Lal+06]	2.78	2.64	5.88	3.32
Notre méthode	0.54	0.07	0.019	0.06

Comme prévu, la méthode basée sur l'ACP [Lal+06] s'avère la moins robuste (voir les résultats de comparaison dans le tableau 4.3). Les résultats obtenus avec cette méthode nous permettent de discuter de l'influence de données incomplètes ou localement manquantes et du bruit sur les résultats obtenus par une méthode d'estimation par optimisation (comme la méthode des moindres carrés, l'analyse en composantes principales, etc.).

Tout d'abord, la méthode proposée par LALONDE et al. [Lal+06], consiste à utiliser le modèle ACP estimé à partir du nuage de points pour déterminer le centre et l'axe principal du cylindre. Ensuite, les points 3D du cylindre sont projetés sur un plan perpendiculaire à l'axe du cylindre déterminé précédemment. Enfin, pour déterminer le rayon du cylindre, un cercle est estimé aux points projetés au sens des moindres carrés.

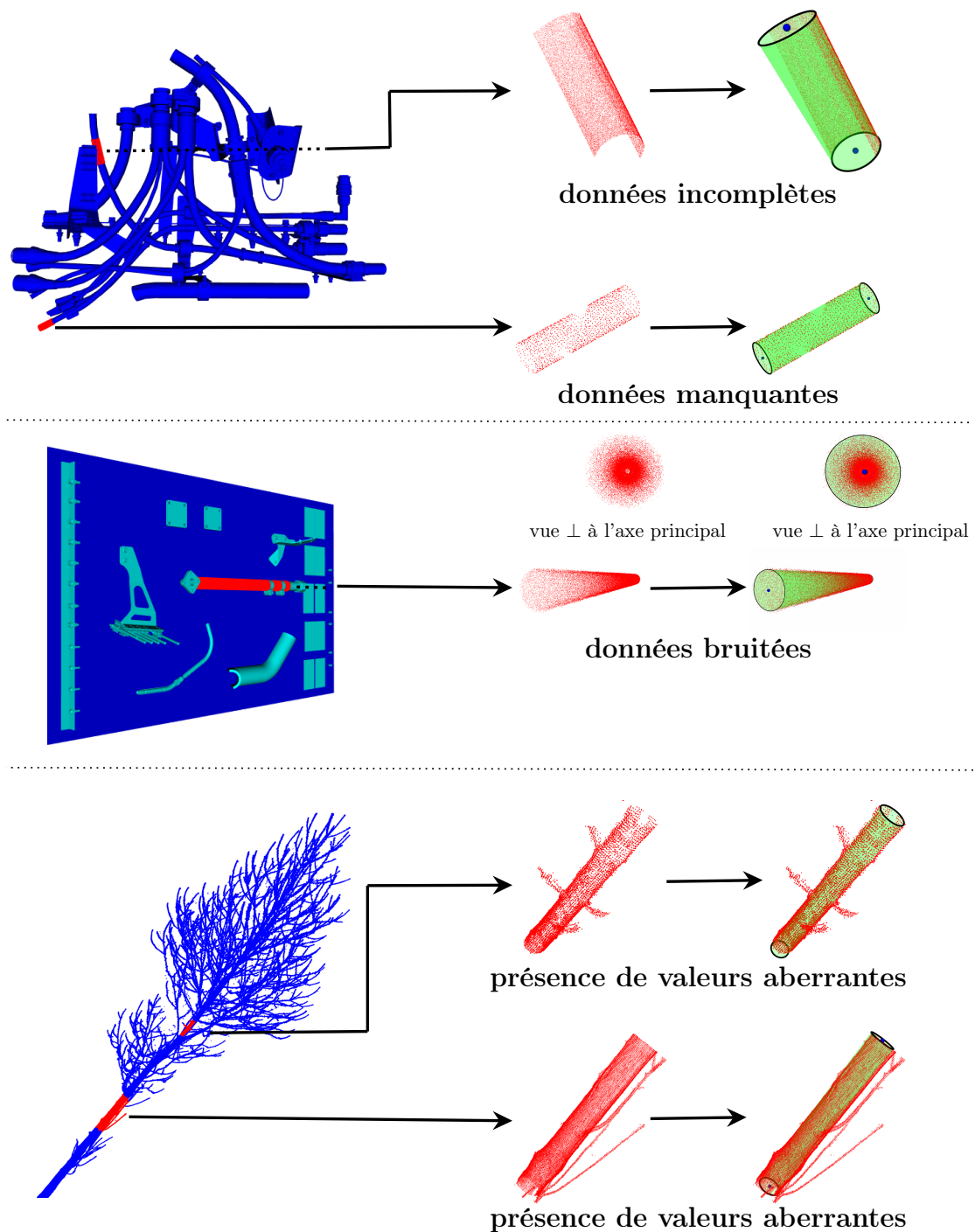


Figure 4.38 – Exemples de jeux de données utilisés et les résultats obtenus : (1^{ère} ligne) cylindres synthétiques, (2^{ème} ligne) tuyau synthétique, (3^{ème} ligne) modèle d'arbre acquis par un système TLS. Les cylindres d'entrée sont en rouge et les cylindres ajustés par notre méthode sont en vert

L'ACP utilise principalement un calcul préliminaire de la moyenne des données et de leur matrice de variance-covariance. Cependant, la moyenne et la variance sont sensibles à la présence du bruit ou de données incomplètes ou localement manquantes [Tha08].

Comme illustré sur la figure 4.39, le modèle ACP obtenu à partir de données bruitées et incomplètes ne représente pas fidèlement le système, et l'estimation de l'axe principal du cylindre est donc incorrecte. De plus, avec les données incomplètes ou localement manquantes, l'approche d'ajustement du cercle au sens des moindres carrés devient inefficace.

Enfin les résultats obtenus s'avèrent souvent inexploitable car trop biaisés par l'influence du bruit et de données incomplètes ou localement manquantes.

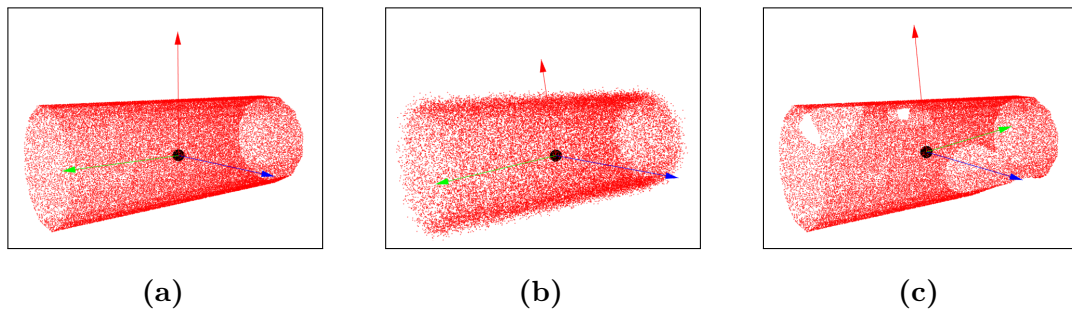


Figure 4.39 – Exemple des modèles ACP obtenus sur : (a) un cylindre synthétique, (b) même cylindre avec bruit ajouté ($\sigma = 0.4$), (c) même cylindre avec données manquantes. Les 3 vecteurs (rouge, bleu et vert) représentent les 3 composantes principales estimées par l'ACP, et la sphère noire représente le centroïde du cylindre. On remarque que les données bruitées, manquantes ou incomplètes peuvent corrompre l'estimation de l'ACP

La méthode basée sur l'algorithme RANSAC [Sch+07] a surpassé la méthode basée sur l'ACP, notamment sur les données incomplètes ou localement manquantes (voir les résultats de comparaison dans le tableau 4.3). En revanche, les résultats obtenus par l'algorithme RANSAC sur les données bruitées sont moins robustes que ceux obtenus avec les données incomplètes ou localement manquantes.

L'algorithme RANSAC est un algorithme stochastique permettant d'estimer les paramètres de certains modèles mathématiques. Cet algorithme est fait pour distinguer les données pertinentes qui suivent le modèle choisi (on les appelle *inliers*) et celles qui ne correspondent pas au modèle choisi (on les appelle *outliers*). Cependant, l'algorithme RANSAC est généralement influencé par l'effet de masque² [Nur+14] (*masking effect*). L'influence de l'effet de masque sur l'algorithme RANSAC a été prouvée par ROUSSEEUW et al. [Rou+11] et récemment confirmé par NURUNNABI et al. [Nur+17]. Dans ce cas l'algorithme RANSAC peut considérer les données

2. Selon ONDO et al. [Ond+01] : « l'effet de masque se produit lorsqu'un sous-ensemble d'observations singulières n'est pas détecté à cause de sa proximité avec une autre observation singulière »

outliers comme des données *inliers*. Pour cela, l'algorithme MLESAC (une version améliorée de RANSAC) a été proposé pour remédier à ce problème.

Conformément aux résultats du tableau 4.3, l'algorithme MLESAC est plus robuste face au bruit que l'algorithme RANSAC. Cet algorithme a été utilisé dans la thèse de DUJARDIN [Duj13] pour la segmentation des lignes de tuyauterie (représentées par des cylindres) dans un nuage de points. Pour plus d'informations sur l'estimateur robuste MLESAC, nous invitons le lecteur à se référer à ce manuscrit de thèse.

L'analyse des résultats du tableau 4.3 montre immédiatement que la performance de la méthode proposée a surpassé les méthodes précédentes, notamment pour l'estimation de la longueur ($MSE_{\text{longueur}} = 0.07$), de l'orientation ($MSE_{\text{orientation}} = 0.019$) et du centre ($MSE_{\text{centre}} = 0.06$).

Notre méthode est légèrement plus lente ($\simeq 190$ ms pour 16 000 points 3D) que les autres approches que nous avons testées ($\simeq 95$ ms pour 16 000 points 3D), mais elle nous a permis d'obtenir de très bons résultats pour l'ajustement de cylindres à partir de données 3D incomplètes et partiellement manquantes.

4.6.4 Évaluation de la méthode de modélisation-segmentation à partir de données synthétiques

Afin de valider notre méthode de segmentation de câble avec des données synthétiques, nous avons d'abord créé à partir des modèles CAO du moteur LEAP (voir la figure 4.34) différents nuages de points contenant différents câbles. Pour examiner l'effet des variations de densité de points, nous avons généré ces nuages de points avec des résolutions différentes à l'aide de la méthode d'échantillonnage uniforme pondéré présentée dans la section 4.3.2. De plus, pour étudier la résistance de la méthode proposée vis-à-vis du bruit et des outliers, les points du nuage sont entachés d'une erreur Gaussienne de moyenne ($\mu = 0$) et de différentes valeurs d'écart-type (σ) qui définissent le niveau de bruit dans toutes les directions (x , y et z) du nuage de points.

La figure 4.40 montre un exemple de données synthétiques utilisées dans nos simulations.

Dans le cas des données synthétiques, notre approche a été évaluée en comparant le câble segmenté avec le câble synthétique. Afin de valider notre méthode, nous avons sélectionné deux critères :

1. L'erreur quadratique moyenne de la distance MSE_{distance} entre le nuage de points de la source (modèle CAO) et le nuage de points segmenté.

$$MSE_{\text{distance}} = \frac{1}{n} \sum_{i=1}^n MSE_{d_i} \quad (4.13)$$

où MSE_{d_i} est défini par l'équation (4.5), et n représente le nombre de câbles testés.

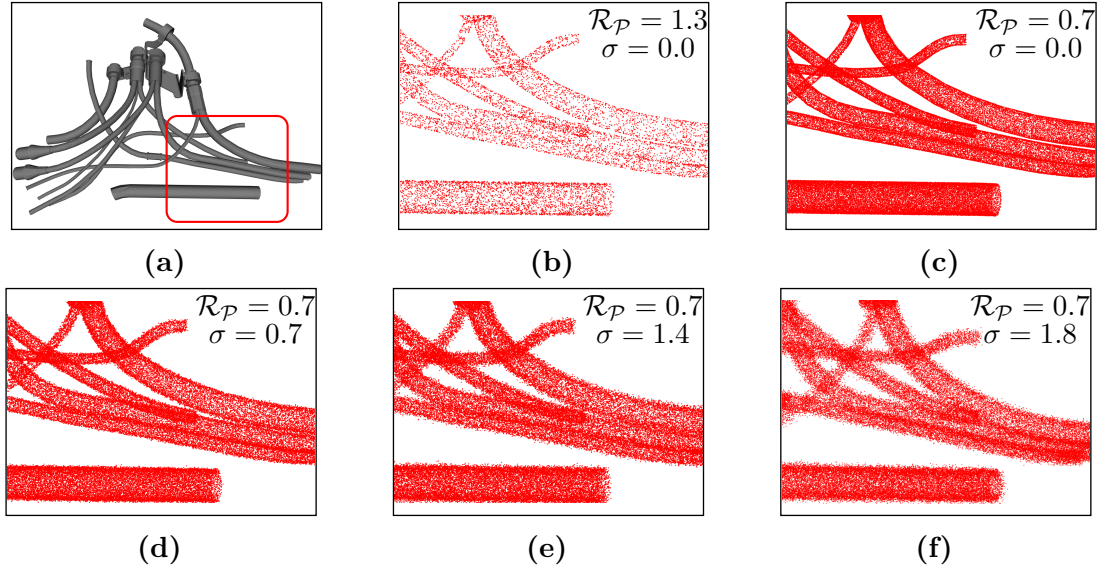


Figure 4.40 – Exemple de données simulées utilisées : (a) modèle CAO, (b-f), nuages de points générés (simulés) à partir du modèle CAO avec différentes résolutions $\mathcal{R}_{\mathcal{P}}$ et différents niveaux du bruit σ

2. L'erreur MSE_{rayon} (définie par l'équation (4.8)) qui représente l'erreur de différence entre le rayon de la vérité terrain (modèle CAO) r et le rayon estimé \tilde{r} .

Le tableau 4.4 présente les résultats expérimentaux de différents câbles (c_1 , c_2 , c_3 et c_4) présentés sur la figure 4.41. Ce tableau confirme que la méthode de segmentation proposée est fiable.

Il est à noter que ces résultats ont été obtenus à partir de 30 essais pour chaque câble :

- Les nuages de points ont été générés avec 3 résolutions différentes ($\mathcal{R}_{\mathcal{P}} = 0.7$, $\mathcal{R}_{\mathcal{P}} = 1.0$ et $\mathcal{R}_{\mathcal{P}} = 1.3$).
- Les points du nuage ont été entachés d'une erreur Gaussienne avec 10 valeurs différentes d'écart-type ($\sigma \in [0.0, 1.8]$).

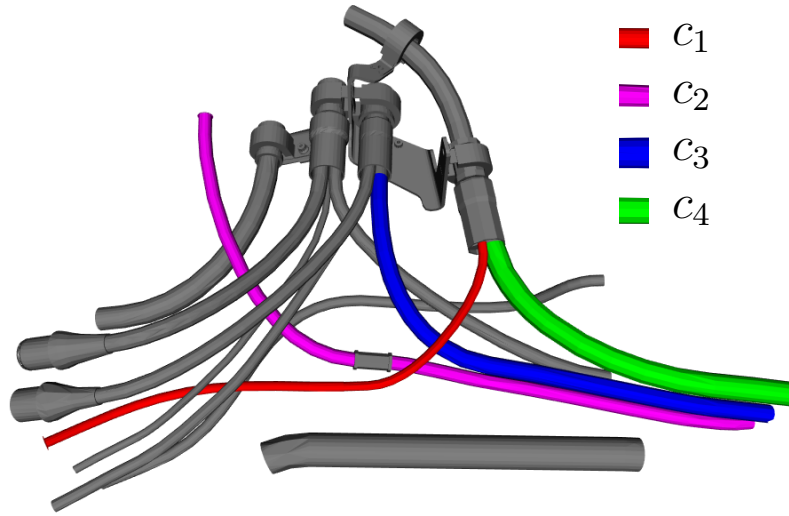


Figure 4.41 – Modèle CAO d’un assemblage mécanique. Ce modèle contient 4 câbles différents étiquetés c_1 à c_4

Table 4.4 – Tableau récapitulatif des résultats de segmentation sur les données simulées. MSE_{distance} représente l’erreur quadratique moyenne de la distance entre le nuage de points synthétique et le nuage de points segmenté pour chaque câble présenté sur la figure 4.41

Câble	MSE_{distance}
c_1	0.03
c_2	0.06
c_3	0.11
c_4	0.04
Moyenne	0.06

La robustesse de notre méthode face au bruit est démontrée par les courbes d’évolution de l’erreur quadratique moyenne de distance MSE_{distance} et du rayon MSE_{rayon} en fonction des niveaux de bruit (σ) (voir la figure 4.42). Pour un écart-type $\sigma < 1.3$, l’évolution de l’erreur est constante et très proche de « zéro » pour l’ MSE_{distance} et pour l’ MSE_{rayon} . Cependant, nous avons constaté qu’avec un écart-type $\sigma > 1.3$, il est difficile d’obtenir un résultat correct. Ceci est illustré par la figure 4.43.

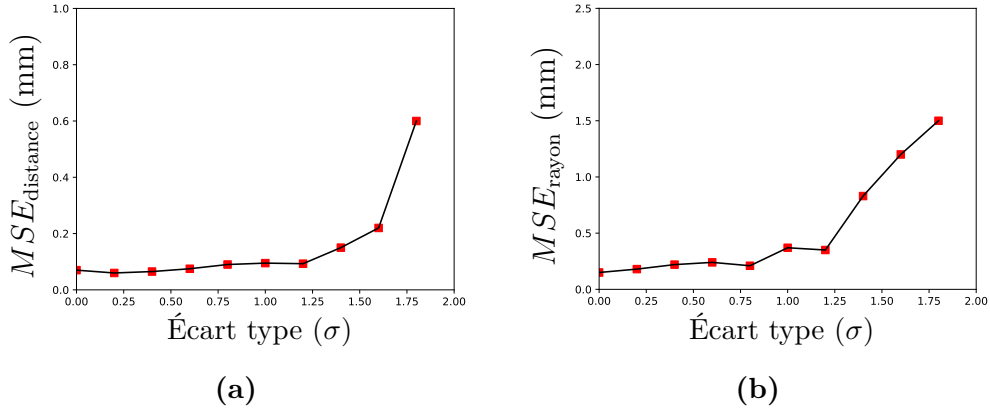


Figure 4.42 – Évolution de l’erreur quadratique moyenne de : (a) la distance MSE_{distance} et (b) du rayon MSE_{rayon} en fonction des niveaux de bruit (σ)

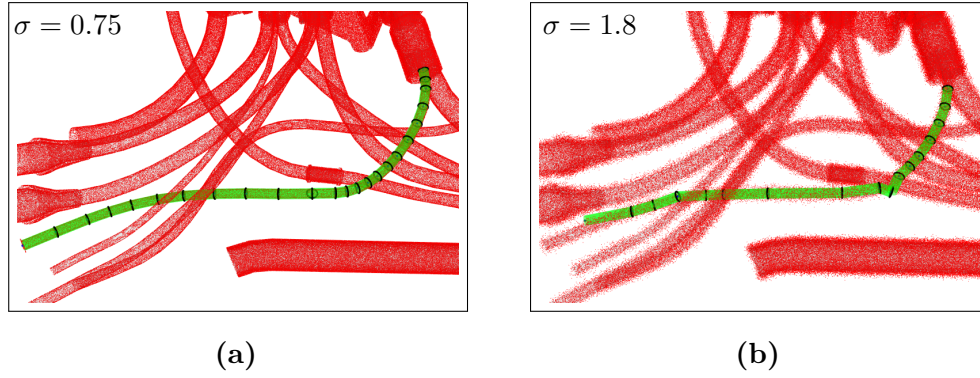


Figure 4.43 – Un exemple de résultat de segmentation obtenu sur le câble c_1 (voir la figure 4.41) avec : (a) un nuage de points bruité (bruit Gaussien d’écart type $\sigma = 0.75$), (b) un nuage de point fortement bruité (bruit Gaussien d’écart type $\sigma = 1.8$)

4.6.5 Évaluation de la méthode de modélisation-segmentation avec des données réelles

Dans cette section, notre algorithme est testé sur des données réelles capturées par le scanner 3D. Les nuages de points ont été acquis par une méthode de type *one-shot* (une seule acquisition depuis un seul point de vue). Par conséquent, les données acquises sont incomplètes et parfois localement manquantes.

Notre base de données réelles contient 74 nuages de points différents acquis pendant plusieurs mois dans des conditions environnementales variées. De plus, pour étudier la résistance de la méthode proposée vis-à-vis du bruit et des outliers, les points du nuage sont entachés d’une erreur Gaussienne de moyenne ($\mu = 0$) et de

différentes valeurs d'écart-type (σ) qui définissent le niveau de bruit dans toutes les directions (x , y et z) du nuage de points.

Dans un premier temps, nous avons utilisé une méthode de segmentation manuelle qui consiste à extraire manuellement les câbles dans un nuage de points. Pour ce faire, nous avons utilisé l'outil Meshlab[®] pour la segmentation/annotation manuelle des câbles dans le nuage de points. Ensuite, notre approche a été évaluée en comparant le câble segmenté manuellement et le câble segmenté avec notre algorithme. Dans ce cas, nous avons sélectionné deux critères :

- L'erreur quadratique moyenne de la distance MSE_{distance} (définie par l'équation (4.13)) entre le nuage de points segmenté manuellement et le nuage de points segmenté avec notre algorithme
- L'erreur MSE_{rayon} (définie par l'équation (4.8)) qui représente l'erreur de différence entre le rayon de la vérité terrain (modèle CAO) r et le rayon estimé \tilde{r} .

La figure 4.44 montre quelques exemples de données réelles utilisées avec les résultats de modélisation-segmentation obtenus sur différents câbles.

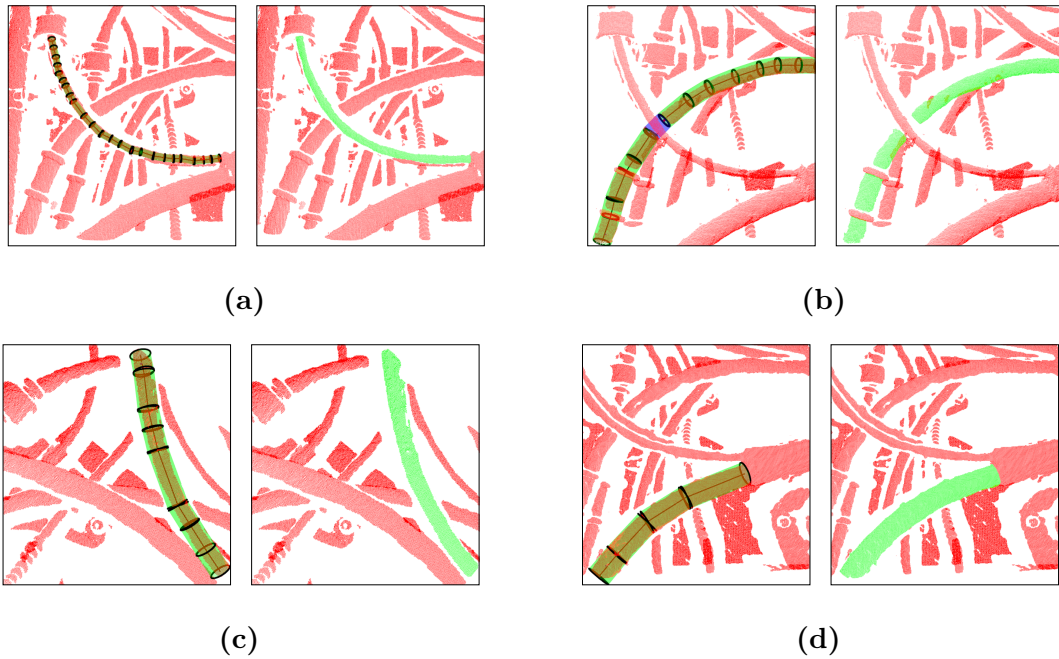


Figure 4.44 – Exemples de résultats de modélisation-segmentation obtenus sur les données réelles acquises par le scanner 3D : (a) un câble de rayon 3 mm, (b) un câble de rayon 6 mm, (c) un câble de rayon 7 mm, (d) un câble de rayon 10 mm

Le tableau 4.5 présente les résultats expérimentaux sur 74 nuages de points avec 6 câbles différents. Ce tableau confirme que la méthode de segmentation proposée est fiable. Les points du nuage ont été entachés d'une erreur Gaussienne avec 10 valeurs différentes d'écart-type ($\sigma \in [0.0, 1.8]$).

Remarque : Étant donné que la génération du nuage de points bruité est aléatoire, les expériences ont été répétées 5 fois. Par conséquent, les valeurs des métriques présentées dans le tableau 4.5 sont obtenues en faisant la moyenne de ces valeurs.

Les résultats ont donc été obtenus à partir de 50 essais (5×10) pour chaque câble.

Table 4.5 – Tableau récapitulatif des résultats de segmentation sur les données réelles dans laquelle r représente le rayon de la vérité terrain et \tilde{r} le rayon moyen estimé

Câble	Nombre de sous-câbles	r	\tilde{r}	MSE_{rayon}	MSE_{distance}
1	23	3.00	2.89	0.11	0.33
2	19	5.00	4.68	0.32	0.07
3	12	6.00	5.93	0.07	0.09
4	11	6.00	5.24	0.76	0.75
5	12 + 1 trou	7.00	6.44	0.56	0.12
6	3	10.00	8.98	1.02	0.08
Moyenne	–	–	–	0.47	0.12

La robustesse de notre méthode face au bruit est démontrée par les courbes d'évolution de l'erreur quadratique moyenne de distance MSE_{distance} et du rayon MSE_{rayon} en fonction des niveaux de bruit (σ) (voir la figure 4.46). Pour un écart-type $\sigma < 1.2$, l'évolution de l'erreur est constante et très proche de « zéro » pour l'erreur MSE_{distance} et pour l'erreur MSE_{rayon} . Cependant, nous avons constaté qu'avec un écart-type $\sigma > 1.2$, il est difficile d'obtenir un résultat correct. Ceci est illustré par la figure 4.45.

Contrairement à l'erreur MSE_{distance} , l'estimation du rayon est évaluée en utilisant le rayon théorique du modèle CAO du câble. Pour cela, l'erreur quadratique moyenne sur l'estimation du rayon MSE_{rayon} est deux fois plus élevée que l'erreur quadratique moyenne de la distance MSE_{distance} (voir le tableau 4.5). Ceci est expliqué par les erreurs provenant du système de mesure (scanner 3D).

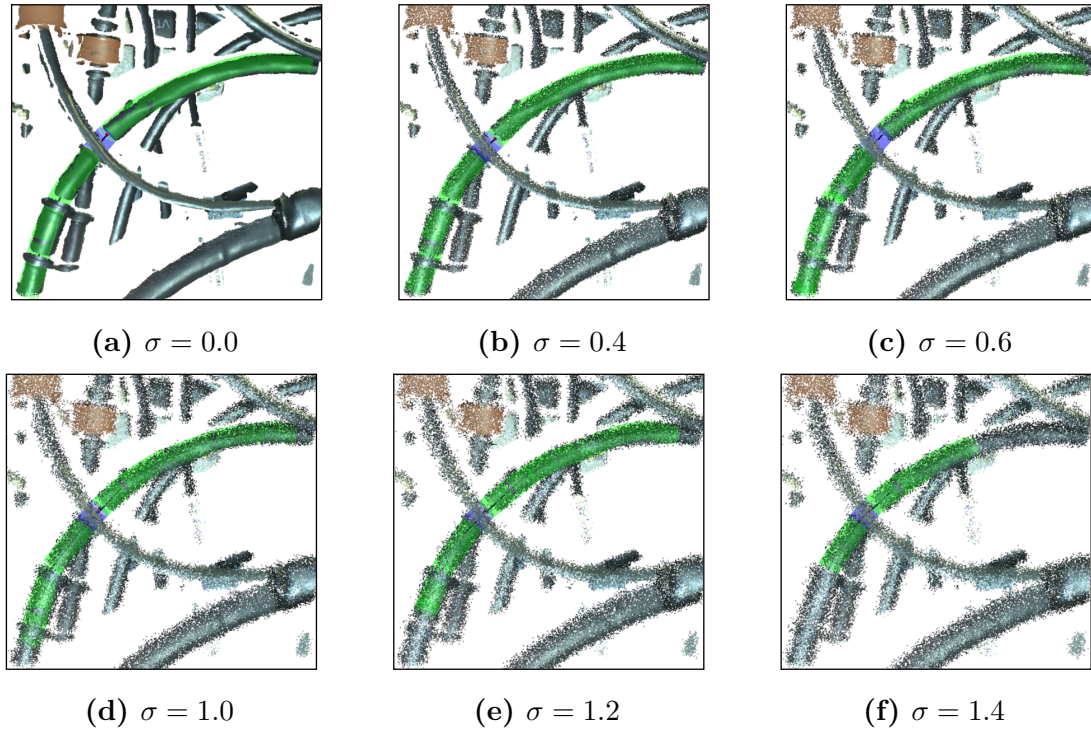


Figure 4.45 – Résultats de modélisation en fonction du niveau de bruit (σ)

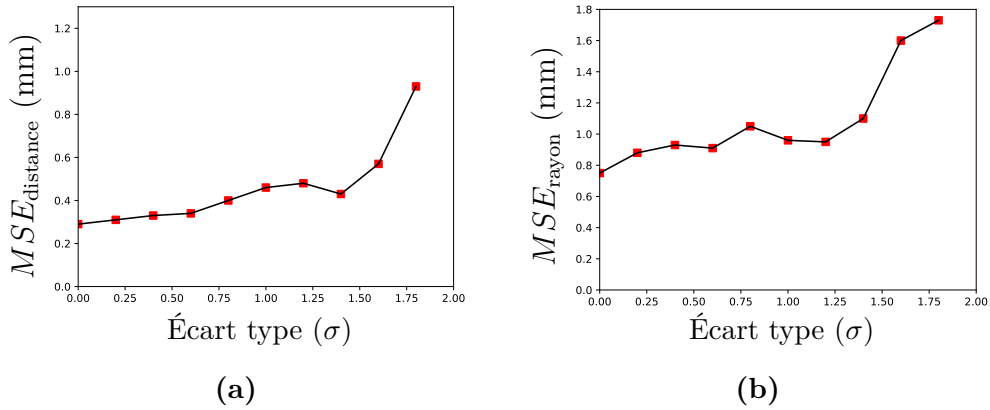


Figure 4.46 – Évolution de l'erreur quadratique moyenne de : (a) la distance $MSE_{distance}$ et (b) du rayon MSE_{rayon} en fonction du niveau de bruit (σ)

Comme illustré sur la figure 4.47, la modélisation et la segmentation ont échoué dans la zone où il y a des valeurs aberrantes (*outliers*) très proches du câble. En effet, l'ajout de niveau de bruit élevé au nuage de points entraîne également l'ajout de points entre les zones aberrantes et le câble à segmenter. Dans ce cas, le bruit provoque une **connexion** entre les autres câbles et le câble à segmenter. Par conséquent, nous perdons la forme cylindrique du câble.

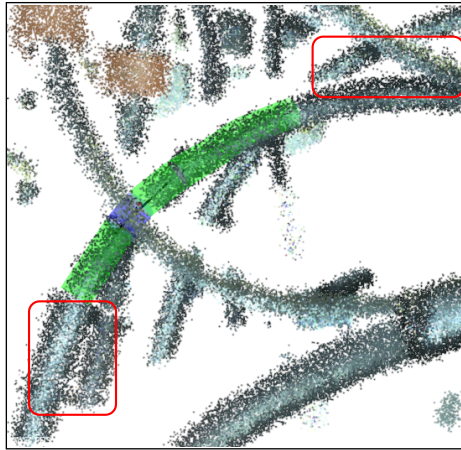


Figure 4.47 – Résultat de modélisation sur un nuage de points fortement bruité ($\sigma = 1.4$). Les rectangles rouges représentent les zones où il y a des valeurs aberrantes (*outliers*) très proches du câble et, dans ce cas, le bruit a causé une **connexion** entre les autres câbles et le câble à segmenter

Avec les moyens de calcul définis dans la section 4.6.1, le processus global de modélisation-segmentation d'un câble s'effectue environ en 7 s sur un nuage de points de l'ordre de 200000 points 3D. À noter que, pour l'instant, nous n'avons pas cherché à optimiser ce temps de calcul.

4.7 Conclusion

Dans cette section, nous faisons une synthèse de nos travaux sur la segmentation 3D des câbles et nous proposons des pistes pour la poursuite de ces travaux.

4.7.1 Synthèse des travaux

Dans ce chapitre, une méthode originale pour la modélisation et la segmentation automatique des câbles (formes non-rigides) dans une scène complexe a été présentée. Malgré la complexité des scènes traitées, nous avons réussi à obtenir des résultats prometteurs et la méthode a été validée dans des environnements industriels.

Nous avons tout d'abord mis en œuvre une étape de pré-traitement des données via une méthode de filtrage, afin de réduire le bruit et de ne conserver que le point d'intérêt dans le nuage de points. Ensuite, nous avons développé une méthode d'échantillonnage pondéré permettant de générer des nuages de points 3D à partir de maillages 3D afin de pouvoir exploiter le modèle CAO 3D sous forme de nuage de points 3D.

Nous avons ensuite proposé une méthode d'alignement (en deux étapes) : alignement global puis local. Dans l'étape d'alignement global nous rapprochons le modèle CAO du nuage de points à l'aide d'un algorithme de SLAM contraint par le modèle. Ensuite les résultats de l'alignement global sont localement raffinés à l'aide d'un algorithme ICP.

Ensuite, nous avons utilisé les résultats de l'alignement pour initialiser la segmentation 3D des câbles. La méthode de segmentation automatique présentée s'appuie sur la modélisation du câble par des portions de cylindres, qui sont ensuite étendues par propagation sur le câble dans les deux sens à l'aide d'un algorithme de recherche qui exploite une structure de recherche basée antipodes d'une sphère.

Enfin, la méthode a été évaluée sur des données synthétiques et réelles issues de bases de données variées constituées par DIOTA.

Dans la première partie de l'évaluation, nous avons montré que la performance de la méthode proposée pour l'ajustement des paramètres du sous-cylindre a surpassé les méthodes classiques très populaires : RANSAC, MLESAC et ACP. Dans la seconde partie de l'évaluation, nous avons montré la robustesse de notre méthode face aux bruits, aux données incomplètes, localement manquantes et aux variations de densité de points.

4.7.2 Perspectives

La solution que nous avons proposée a permis de modéliser et segmenter les câbles dans une scène complexe. Cette solution pourrait être mise en œuvre dans d'autres contextes applicatifs :

- modélisation et segmentation d'une ligne de tuyauterie de grandes structures industrielles (voir la figure 4.48a).
- modélisation et segmentation des arbres pour le domaine forestier (voir la figure 4.48b).
- modélisation et segmentation des artères en imagerie médicale 3D (voir la figure 4.48c).

Les travaux sur l'initialisation automatique basée modèle de la structure de recherche (présentés dans la section 4.4) devront être approfondis. En effet, les méthodologies présentées, qui exploitent la maquette numérique, donnent des résultats satisfaisants mais à condition de disposer du modèle CAO.

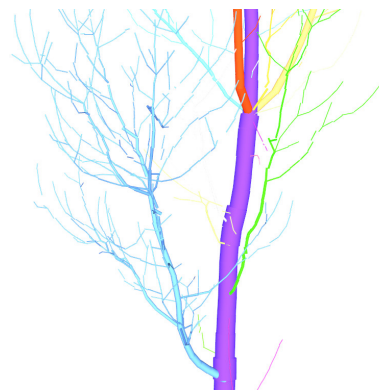
Une autre amélioration que nous avons envisagée, sans avoir pu l'étudier en détails, consisterait à exploiter l'apprentissage automatique pour automatiser la détermination des paramètres du filtre (proposé dans la section 4.3.1) et leur ajustement adaptatif.

Les temps de calcul n'ont pas été optimisés jusqu'ici. En particulier la propagation de la structure de recherche présentée dans la section 4.5.3. En vue d'améliorer les temps d'exécution, on peut envisager d'envoyer la structure de recherche dans les deux sens (« + » et « - ») à partir des deux extrémités du sous-cylindre initial en même temps à l'aide de la programmation parallèle³.

3. La parallélisme consiste à mettre en œuvre des programmes informatiques permettant de traiter des informations de manière simultanée.



(a) Domaine industriel



(b) Domaine forestier



(c) Domaine médical

Figure 4.48 – Exemples d’application dans le domaine : (a) industriel [Kum+17], (b) forestier [Hac+14], (c) médical [Ans+17]

Dans ce chapitre, nous avons présenté une méthode automatique de segmentation 3D des câbles dans une scène complexe. Dans le chapitre suivant, nous montrons l'exploitation des câbles segmentés pour mesurer le rayon de courbure de chaque câble et détecter des interférences éventuelles entre un câble et les éléments qui l'entourent.

Inspection à partir de nuages de points 3D : inspection 3D des câbles

5.1	Introduction et problématique	142
5.2	Schéma général de la méthode proposée	143
5.3	Mesure du rayon de courbure	144
5.3.1	Rayon de courbure minimal des câbles	144
5.3.2	Mesure du rayon de courbure à partir des résultats de modélisation	145
5.3.3	Prise de décision	148
5.4	Détection d'interférences	149
5.4.1	Les entrées	149
5.4.2	Concept d'interférence	150
5.4.3	Regroupement du nuage de points $\overline{\mathcal{P}}_s$	150
5.4.3.1	Méthode de regroupement basée sur la distance Euclidienne	151
5.4.3.2	Résultats	152
5.4.4	Détecteur d'interférences	153
5.5	Résultats expérimentaux	157
5.5.1	Conditions expérimentales	157
5.5.2	Présentation des données	157
5.5.3	Évaluation de la méthode de calcul du rayon de courbure des câbles	157
5.5.3.1	Métriques d'évaluation pour l'analyse de performances	159
5.5.3.2	Résultats de l'évaluation	159
5.5.4	Évaluation de la méthode de détection d'interférences	160
5.6	Conclusion	163
5.6.1	Synthèse des travaux	163
5.6.2	Améliorations futures	163

5.1 Introduction et problématique

Le moteur d'avion est équipé de différents harnais, canalisations et câbles, portés par des supports (voir figure 5.1). Le système d'interconnexion de câblage électrique est désigné par EWIS (*Electrical Wiring Interconnection System*). En fonction des routages qui sont faits, il peut arriver que les câbles soient trop proches les uns des autres ou qu'ils passent trop près de certains supports. Cette proximité engendre un risque, il est donc nécessaire de s'assurer que la distance de sécurité est respectée. En effet, les câbles sont des éléments flexibles qui peuvent changer de forme et d'emplacement. De plus, les différents éléments de l'assemblage mécanique auront des expositions diverses. Si l'environnement est trop chaud et que le câble est trop proche de ces éléments, il peut brûler et même faire fondre les câbles. Par conséquent, il faut identifier les zones de passage critiques et y relever une distance qui soit compatible avec les contraintes de sécurité. Une enquête effectuée par la [FAA](#) (*Federal Aviation Administration*) a montré que des défauts de câblage sur le système EWIS a été la cause de certains accidents d'avion [Sle+06].

Lors de l'installation, les câbles sont tordus ou fléchis dans diverses conditions environnementales. Tout rayon inférieur à un rayon de courbure minimal autorisé peut entraîner des affaiblissements de courbure et/ou des cassures du câble, pouvant alors causer des dommages aux blindages. La gaine extérieure pourrait alors craquer. Par conséquent, il est nécessaire de parvenir à mesurer ce rayon, afin de vérifier sa conformité par rapport aux exigences définies par les normes de sécurité.

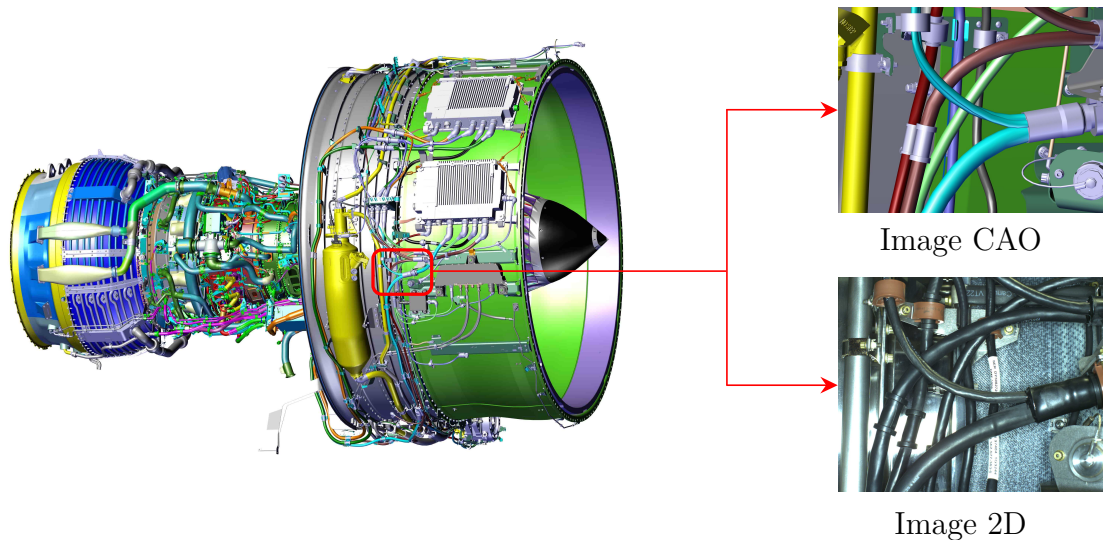


Figure 5.1 – Moteur d'avion

Nous n'avons identifié aucune publication sur ce sujet dans l'état de l'art. Par ailleurs, il existe un petit nombre de logiciels commerciaux qui permettent d'effectuer la mesure du rayon de courbure et la détection d'interférences entre les câbles et les autres éléments de l'assemblage mécanique, mais uniquement sur des modèles CAO (par exemple : SolidWorks®, CATIA®). Enfin, à notre connaissance, il n'existe pas de

logiciel ou d'algorithme, qui permette d'effectuer ces opérations sur des nuages de points 3D.

Nous avons vu dans le chapitre précédent une méthode originale et automatique pour la segmentation 3D de câbles, basée uniquement sur l'analyse des nuages de points 3D et l'exploitation du modèle CAO du câble. Dans ce chapitre, nous présentons une exploitation des câbles segmentés pour inspecter le système d'interconnexion de câblage électrique (*EWIS*), et, plus précisément, pour mesurer le rayon de courbure de chaque câble et aussi vérifier si les câbles sont à une distance minimale des autres éléments présents dans son environnement (problème d'interférences).

Tout d'abord, nous présenterons dans la section 5.3 une méthode pour mesurer le rayon de courbure des câbles. Cette méthode se déroule en trois étapes : (i) ajustement d'un plan 3D au sens des moindres carrés sur l'ensemble de données P^i (points initiaux) et P^f (points finaux) du modèle final du câble. (ii) projection de l'ensemble de données P^i (points initiaux) et P^f (points finaux) du modèle final du câble sur le plan déterminé dans l'étape précédente. (iii) ajustement d'un cercle 2D au sens des moindres carrés sur l'ensemble des points projetés.

Ensuite, nous présenterons dans la section 5.4 un algorithme pour la détection d'interférences entre les câbles et les autres éléments de l'assemblage dans une scène complexe où il peut s'avérer difficile de déterminer visuellement si les composants interfèrent entre eux.

Enfin, le protocole expérimental et les résultats obtenus seront présentés et analysés dans la section 5.5.

5.2 Schéma général de la méthode proposée

La figure 5.2 présente le schéma général de la méthode d'inspection proposée. Cette méthode repose principalement sur les résultats de la modélisation et de la segmentation du câble présentés dans le chapitre 4. Dans un premier temps, il s'agit de mesurer le rayon de courbure du câble segmenté, afin de vérifier que ce rayon mesuré est supérieur au rayon de courbure minimal autorisé $r_c(\min)$. Dans un second temps, il s'agit de détecter les interférences entre les câbles et les autres éléments. Autrement dit, il s'agit de vérifier que le câble segmenté et les éléments environnants présents dans l'ensemble mécanique sont à une distance minimale ($d_{\mathcal{T}}$) l'un de l'autre.

Dans la suite de ce chapitre, nous détaillons chacun de ces algorithmes d'inspection.

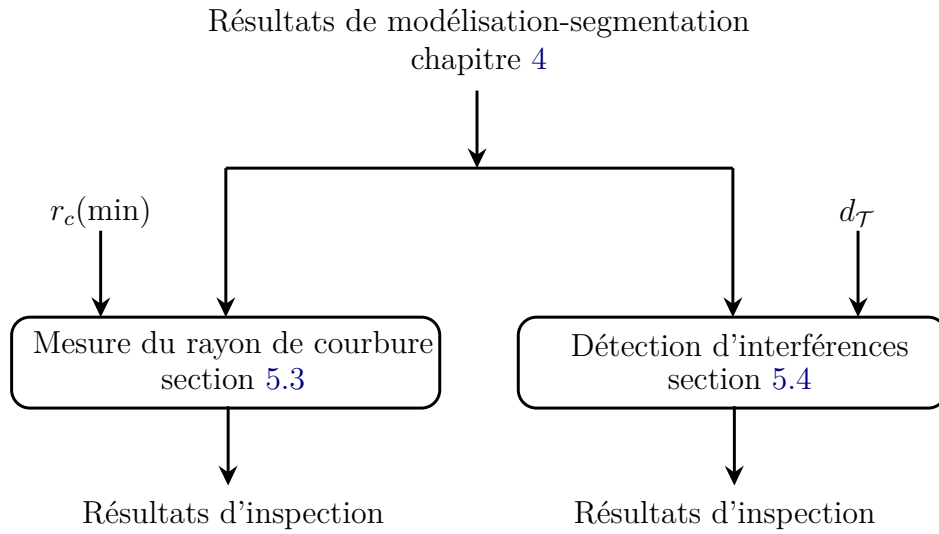


Figure 5.2 – Schéma général de la méthode proposée

5.3 Mesure du rayon de courbure

Nous allons présenter dans cette section l'algorithme de mesure du rayon de courbure des câbles.

5.3.1 Rayon de courbure minimal des câbles

Le rayon de courbure minimal $r_c(\min)$ est le plus petit rayon permis autour duquel le câble peut se plier. Il se base principalement sur le diamètre du câble et sur le type de câble :

$$r_c(\min) = k \times d \quad (5.1)$$

où d représente le diamètre extérieur du câble et k représente un multiplicateur qui définit le type de câble.

Les multiplicateurs du câble sont établis par les normes du secteur d'activité et varient en fonction du type de câble. Selon les normes de sécurité du câble, le rayon de courbure recommandé pour les harnais (faisceaux de câble) et les câbles dans les assemblages mécaniques aéronautiques ne devrait pas être inférieur à 10 fois le diamètre du plus gros câble. Dans ce cas :

$$r_c(\min) = 10 \times d \quad (5.2)$$

Par exemple, pour un câble de diamètre 14 mm, le rayon de courbure minimal $r_c(\min)$ ne devrait pas être inférieur à 140 mm.

Lors de l'installation, les câbles sont tordus ou fléchis dans diverses conditions environnementales. Cependant, tout rayon inférieur au rayon de courbure minimal autorisé peut entraîner :

- Des affaiblissements de courbure et/ou des cassures du câble.
- Un craquement de la gaine extérieure.
- Des dommages aux blindages.

Les exigences pour le rayon de courbure minimal sont définies par les normes :

- *NEC* (Code National sur l'Electricité) [NFP14],
- *ICEA* (Association des Ingénieurs de câbles Isolés) [ICE08].

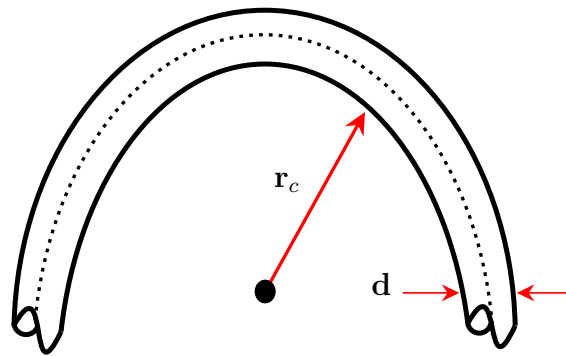


Figure 5.3 – Rayon de courbure du câble

5.3.2 Mesure du rayon de courbure à partir des résultats de modélisation

La méthode de modélisation présentée dans le chapitre 4 produit un modèle de câble à partir d'un nuage de points 3D. Ce modèle contient un ensemble de sous-cylindres $s\text{-}cy$ dans lequel chaque sous-cylindre $s\text{-}cy_k$ possède un point initial P_k^i et un point final P_k^f (voir la figure 5.4). À l'aide de ce modèle, nous pouvons effectuer une analyse quantitative du rayon de courbure du câble segmenté.

Dans cette sous-section, nous présenterons les étapes nécessaires pour mesurer le rayon de courbure à partir des résultats de modélisation.

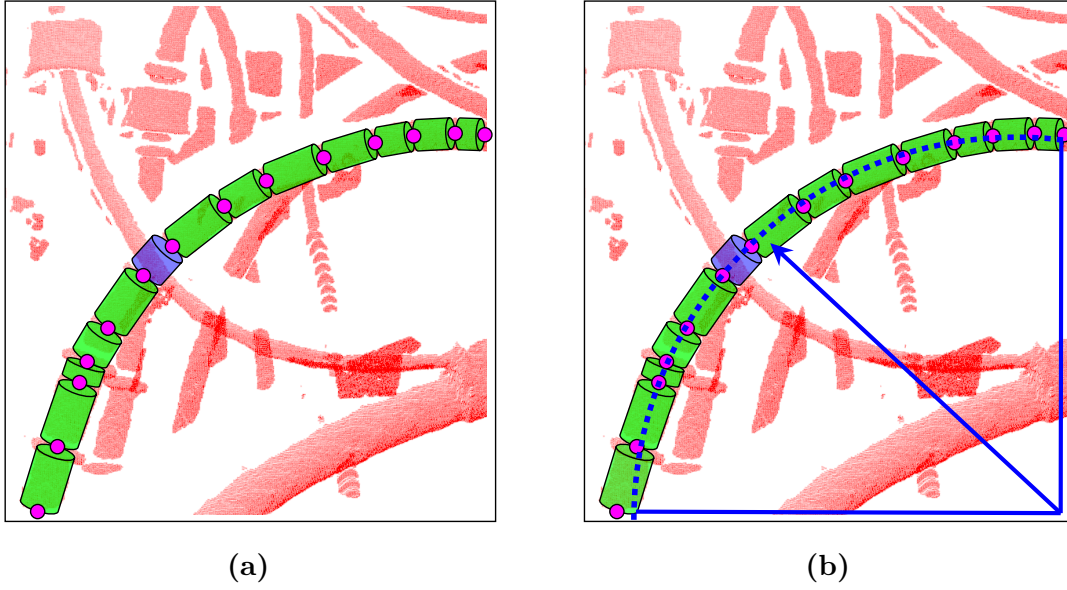


Figure 5.4 – Calcul du rayon de courbure : (a) un exemple de résultat de modélisation dans lequel chaque sous-cylindre $s\text{-cy}_k$ (en vert) possède un point initial P_k^i et un point final P_k^f représentés en rose, (b) le rayon de courbure correspondant

La mesure du rayon de courbure se déroule en trois étapes :

- **Étape 1** : ajustement d'un plan 3D au sens des moindres carrés sur l'ensemble des données P^i (points initiaux) et P^f (points finaux) du modèle final du câble. L'ensemble des points 3D $\{P_k^{i,f}(x_k, y_k, z_k), k = 1, \dots, n\}$ peut être approximé par un plan 3D. Rechercher une relation entre les variables x , y et z vérifiant l'équation du plan (équation (5.3)) revient à rechercher un plan qui s'ajuste le mieux possible à ces points 3D.

$$z = ax + by + c \quad (5.3)$$

Parmi tous les plans possibles, nous sélectionnons celui qui possède une propriété remarquable : c'est celui qui rend minimale la **somme des carrés des écarts** des valeurs observées z_k au plan $\hat{z}_k = ax_k + by_k + c$. Si ϵ_k représente cet écart, appelé aussi *résidu*, le principe des *moindres carrés* consiste à choisir les valeurs a , b et c qui minimisent :

$$E = \sum_{k=1}^n \epsilon_k^2 = \sum_{k=1}^n (z_k - (ax_k + by_k + c))^2 \quad (5.4)$$

Par exemple, la figure 5.5 montre le plan qui s'ajuste le mieux possible à l'ensemble de données P^i (points initiaux) et P^f (points finaux) du modèle final du câble de la figure 5.4a. Ce plan est déterminé en minimisant la somme des carrés des résidus E (voir équation (5.4)).

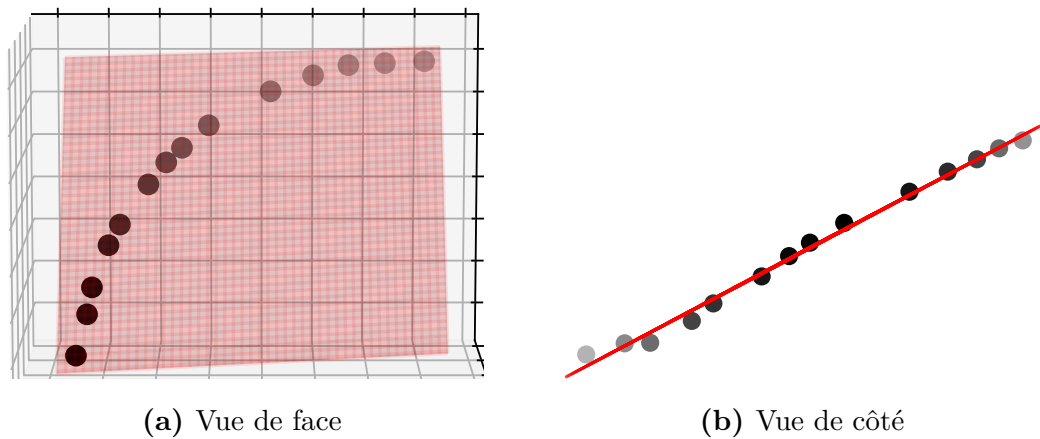


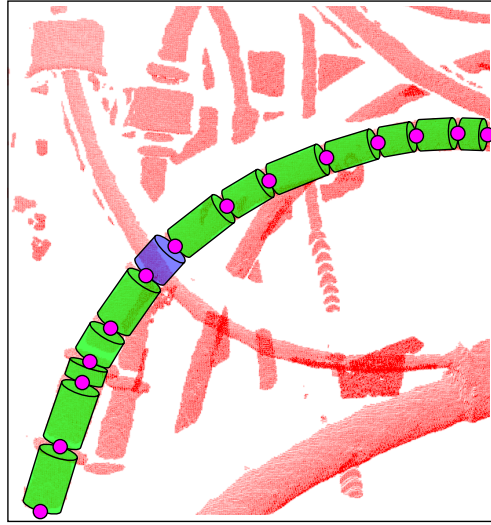
Figure 5.5 – Ajustement du plan 3D au sens des moindres carrés sur l'ensemble de données P^i (points initiaux) et P^f (points finaux) du modèle final du câble de la figure 5.4a

- **Étape 2** : projection de l'ensemble de données P^i (points initiaux) et P^f (points finaux) du modèle final du câble sur le plan déterminé dans l'étape précédente (voir la figure 5.6b).
- **Étape 3** : ajustement d'un cercle 2D au sens des moindres carrés sur l'ensemble des points projetés (voir la figure 5.6c).

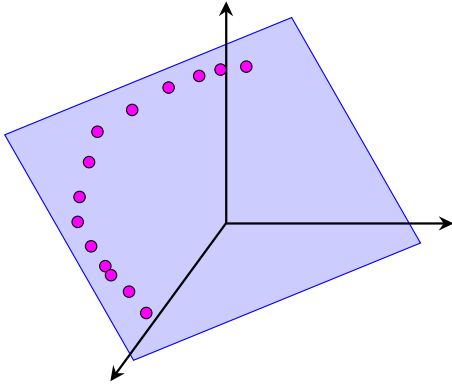
Enfin, le rayon de courbure du câble est égal au rayon du cercle 2D déterminé dans la dernière étape.

Remarque : plusieurs autres possibilités peuvent être utilisées pour mesurer le rayon de courbure d'un câble :

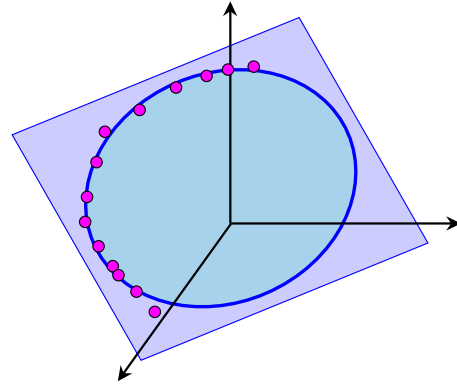
- Ajuster directement un cercle 3D sur l'ensemble de données P^i (points initiaux) et P^f (points finaux) du modèle final du câble. Cependant, nous avons testé cette méthode et elle s'est avérée moins robuste que celle que nous avons mise en place.
- Ajuster le plan 3D de l'étape 1 avec une analyse en composantes principales (ACP). Il s'agit de calculer le modèle ACP sur l'ensemble de données P^i et P^f du modèle final du câble. Ensuite, le plan 3D est déterminé à partir de deux premiers vecteurs principaux. Cependant, nous avons également testé cette méthode et elle s'est aussi avérée moins robuste que la méthode des moindres carrés.



(a)



(b)



(c)

Figure 5.6 – Les étapes de calcul du rayon de courbure : (a) un exemple de résultat de segmentation-modélisation dans lequel chaque sous-cylindre $s\text{-}cy_k$ (en vert) possède un point initial P_k^i et un point final P_k^f représentés en rose, (b) projection de l'ensemble de données P^i et P^f du modèle final du câble sur le plan 3D ajusté au sens des moindres carrés, et (c) ajustement d'un cercle 2D au sens des moindres carrés sur l'ensemble des points projetés

5.3.3 Prise de décision

À partir de la mesure du rayon de courbure, une étape de prise de décision doit indiquer *in fine* si un défaut est présent ou non. La décision est prise comme suit :

- le câble est considéré comme conforme si le rayon de courbure mesuré \tilde{r}_c est supérieur au rayon de courbure minimal $r_c(\min)$
- le câble est considéré comme non conforme si le rayon de courbure mesuré \tilde{r}_c est inférieur au rayon de courbure minimal $r_c(\min)$

Les résultats expérimentaux de la méthode présentée dans cette section seront présentés dans la section 5.5.

5.4 Détection d'interférences

Nous allons présenter dans cette section l'algorithme de détection d'interférences entre les composants d'un assemblage mécanique complexe.

5.4.1 Les entrées

Une fois la segmentation faite, nous avons deux nuages de points principaux : le nuage de points segmenté, que nous appellerons \mathcal{P}_s , et le nuage de points d'entrée sans le nuage de points segmenté, que nous appellerons $\overline{\mathcal{P}}_s$ (voir la figure 5.7).

Si nous appelons \mathcal{P} le nuage de points d'entrée fourni par le scanner 3D (puis filtré), alors nous obtenons l'équation suivante :

$$\mathcal{P} = \mathcal{P}_s + \overline{\mathcal{P}}_s \quad (5.5)$$

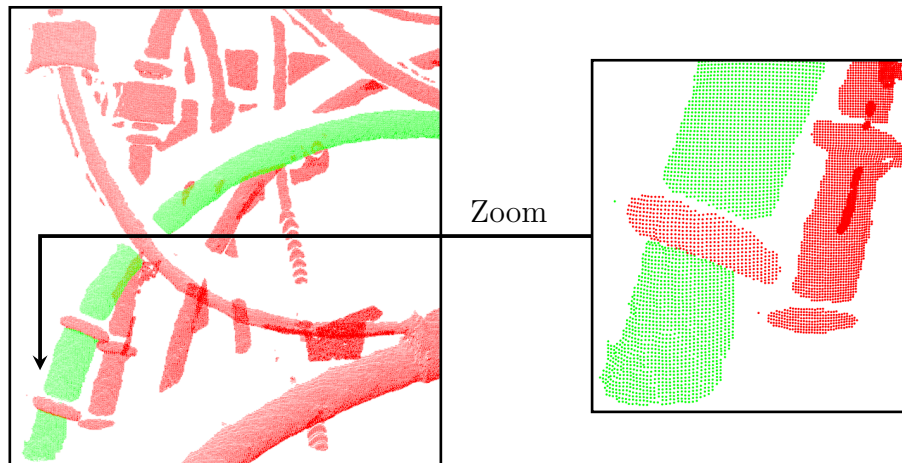


Figure 5.7 – Les entrées : le nuage de points segmenté \mathcal{P}_s en vert et le nuage de points d'entrée sans le nuage de points segmenté $\overline{\mathcal{P}}_s$ en rouge

Le tableau 5.1 résume les principales entrées de l'algorithme de détection d'interférences.

Table 5.1 – Les entrées

Nom	Rôle
\mathcal{P}	Nuage de points d'entrée filtré
\mathcal{P}_s	Nuage de points segmenté (câble)
$\overline{\mathcal{P}}_s$	Nuage de points d'entrée sans le nuage de points segmenté
d_{τ}	Tolérance (distance minimale autorisée entre les éléments d'assemblage)

5.4.2 Concept d'interférence

L'objectif est de vérifier que la distance minimale entre un câble et les autres composants d'assemblage mécanique respecte les contraintes de sécurité. En d'autres termes, il s'agit de vérifier si le câble à inspecter est situé à une distance de sécurité ($d_{\mathcal{T}}$) des autres éléments environnants présents dans l'ensemble mécanique.

Afin de mesurer la distance entre le nuage de points segmenté \mathcal{P}_s (qui représente le câble) et les éléments environnants appartenant à $\overline{\mathcal{P}}_s$, il est nécessaire dans un premier temps de diviser le nuage de points $\overline{\mathcal{P}}_s$ en groupes cohérents et significants. Ensuite, si la distance minimale entre le nuage de points segmenté (câble) et un cluster appartenant à $\overline{\mathcal{P}}_s$ est inférieure à la valeur de tolérance donnée $d_{\mathcal{T}}$, alors le câble est en interférence avec ce cluster.

Nous définissons une interférence de la manière suivante : un câble (ensemble de points $p_i \in \mathcal{P}_s$) est en interférence avec un *cluster* $E_j = \{p_j \in \overline{\mathcal{P}}_s\}$ si :

$$\min \|p_i - p_j\|_2 < d_{\mathcal{T}} \quad (5.6)$$

avec $d_{\mathcal{T}}$ représentant la distance minimale autorisée entre les éléments d'assemblage.

La section suivante fournit les détails nécessaires sur la façon de diviser un nuage de points 3D en groupes cohérents et significants.

5.4.3 Regroupement du nuage de points $\overline{\mathcal{P}}_s$

Le regroupement du nuage de points consiste à diviser le nuage de points $\overline{\mathcal{P}}_s$ en groupes cohérents et significants. Cette activité est le plus souvent désignée dans la communauté scientifique par le terme anglais *clustering*.

Le *clustering* est une méthode de classification non supervisée (pas de classes prédéfinies), qui cherche à décomposer le nuage de points 3D en plusieurs sous-ensembles les plus homogènes possibles.

Une bonne méthode de regroupement permet de garantir :

- Une grande similarité intra-groupe
- Une faible similarité inter-groupe

Les techniques de *clustering* font l'objet de très nombreux travaux.

Tout d'abord, nous trouvons l'algorithme de partitionnement de données *K-moyennes* (*K-means* en anglais) introduit par HARTIGAN et al. [Har+79] en 1979. Cet algorithme consiste à diviser le nuage de points 3D en K *clusters* de façon à minimiser la distance Euclidienne de chaque point 3D par rapport au centroïde de cluster. De nombreuses variantes se sont succédées depuis afin d'étendre ses capacités de séparation : *global-K-means* [Lik+02], *X-means* [Pel+02], *Gaussian-means* [Ham+03], *K-means++* [Art+07]. Dans la même optique, SANTOS et al. [San+14] ont montré qu'il est possible de diviser le nuage de points en groupes cohérents et significants en utilisant l'algorithme *Spectral Clustering*, proposé par NG et al. [Ng+01]. Cependant, ces deux méthodes ne sont pas aussi efficaces et ont

l'inconvénient de devoir connaître le nombre de clusters *a priori*, ce qui implique une interaction de la part d'un utilisateur. Elles ont néanmoins l'avantage d'être peu coûteuses en termes de mise en œuvre et de performance.

Dans notre cas, nous nous sommes orientés vers la solution de regroupement basée sur la distance Euclidienne proposée par RUSU [Rus10]. En effet, cette méthode donne de bons résultats sur nos données. De plus, elle est très rapide du fait qu'elle utilise la structure accélératrice *Kd-Tree* pour accélérer la recherche de plus proches voisins.

Remarques : pour plus d'informations sur les techniques de regroupement (*clustering*), nous invitons le lecteur à se référer aux manuscrits de thèse de RUSU [Rus09] et de GÉLARD [Gél18].

5.4.3.1 Méthode de regroupement basée sur la distance Euclidienne

La méthode de regroupement (ou *clustering*) que nous avons employée pour diviser le nuage de points en sous-ensembles (appelés *clusters*) cohérents et signifiants a été proposée par RUSU [Rus10]. Cette méthode permet de créer des *clusters* à partir d'un nuage de points 3D non organisé en se basant sur des contraintes géométriques.

Afin de décomposer le nuage de points 3D $\overline{\mathcal{P}_s}$ en plusieurs *clusters* les plus homogènes possible, l'algorithme doit être capable de comprendre ce qu'est un *cluster* et ce qui le différencie des autres. Un *cluster* $E_i = \{p_i \in \overline{\mathcal{P}_s}\}$ est un sous-ensemble distinct de $E_j = \{p_j \in \overline{\mathcal{P}_s}\}$ qui vérifie l'équation suivante :

$$\min \|p_i - p_j\|_2 < d_{th} \quad (5.7)$$

avec d_{th} représentant la distance de seuil maximale qui permet de faire la distinction entre deux *clusters*. Cette équation indique que si la distance minimale entre un ensemble de points $p_i \in \overline{\mathcal{P}_s}$ et un autre ensemble de points $p_j \in \overline{\mathcal{P}_s}$ est supérieure à la distance de seuil donnée, alors ces points sont dans deux *clusters* différents.

Les paramètres de cette méthode sont présentés dans le tableau 5.2.

Table 5.2 – Les paramètres de la méthode Euclidienne d'extraction de clusters

Paramètre	Rôle
d_{th}	Cette distance permet de faire la distinction entre deux <i>clusters</i>
$\min_{cluster}$	Nombre minimum de points qu'un <i>cluster</i> doit contenir pour être considéré comme valide
$\max_{cluster}$	Nombre maximum de points qu'un <i>cluster</i> doit contenir pour être considéré comme valide

Cette méthode Euclidienne d'extraction de *clusters* est résumée à travers le pseudo-code suivant :

-
1. Créer une structure *Kd-Tree* pour représenter le nuage de points d'entrée $\overline{\mathcal{P}_s}$
 2. Mettre en place une liste vide de *clusters* \mathcal{C} , et une file d'attente¹ des points \mathcal{Q}
 3. Pour chaque point $p_i \in \overline{\mathcal{P}_s}$, effectuer les étapes suivantes :
 - (a) ajouter p_i à la file d'attente courante \mathcal{Q}
 - (b) pour chaque point $p_i \in \mathcal{Q}$ faire :
 - i. rechercher l'ensemble E_i^k de points voisins de p_i dans une sphère de rayon $r < d_{th}$
 - ii. pour chaque voisin $p_i^k \in E_i^k$, vérifier si p_i^k a déjà été traité, et si ce n'est pas le cas, ajouter p_i^k dans la file d'attente \mathcal{Q}
 - (c) lorsque la liste de tous les points de \mathcal{Q} a été traitée, ajouter \mathcal{Q} à la liste des clusters \mathcal{C} , et réinitialiser \mathcal{Q} à une liste vide
 4. L'algorithme se termine lorsque tous les points $p_i \in \overline{\mathcal{P}_s}$ ont été traités et font maintenant partie de la liste de *clusters* \mathcal{C}

Comme nous avons vu précédemment, la distance d_{th} permet de faire la distinction entre deux *clusters*, de sorte que, si la distance minimale entre un ensemble de points $p_i \in \overline{\mathcal{P}_s}$ et un autre ensemble de points $p_j \in \overline{\mathcal{P}_s}$ est supérieure à cette distance, alors ces points sont dans deux *clusters* différents. Cependant, nous avons trouvé que, si cette distance est très grande, tous les points seront regroupés dans un seul et unique cluster, et si la distance est très petite, chaque point sera considéré comme un *cluster*. Afin de résoudre ce problème, nous avons paramétré cette distance d_{th} (trouvée de façon empirique) en fonction de la résolution spatiale du nuage de points \mathcal{R}_P .

Empiriquement, les paramètres présentés dans le tableau 5.2 ont été choisis comme suit : $d_{th} = 10$ mm, $\min_{cluster} = 100$ points et $\max_{cluster} = 100000$ points.

5.4.3.2 Résultats

Comme on le voit sur la figure 5.8, la méthode d'extraction de *clusters* présentée précédemment permet de diviser le nuage de points 3D en groupes (plusieurs parties) cohérents et significatifs.

Remarque : vu que nous avons fixé le nombre minimum de points qu'un cluster doit contenir pour être considéré comme valide ($\min_{cluster} = 100$), les plus petits clusters sont filtrés. C'est pourquoi ils n'apparaissent plus sur la figure 5.8b.

1. En informatique, une file d'attente (*queue* en anglais) est une structure de données (liste) basée sur le principe du premier entré, premier sorti. Autrement dit, les premiers éléments ajoutés à la file seront les premiers à en être retirés.

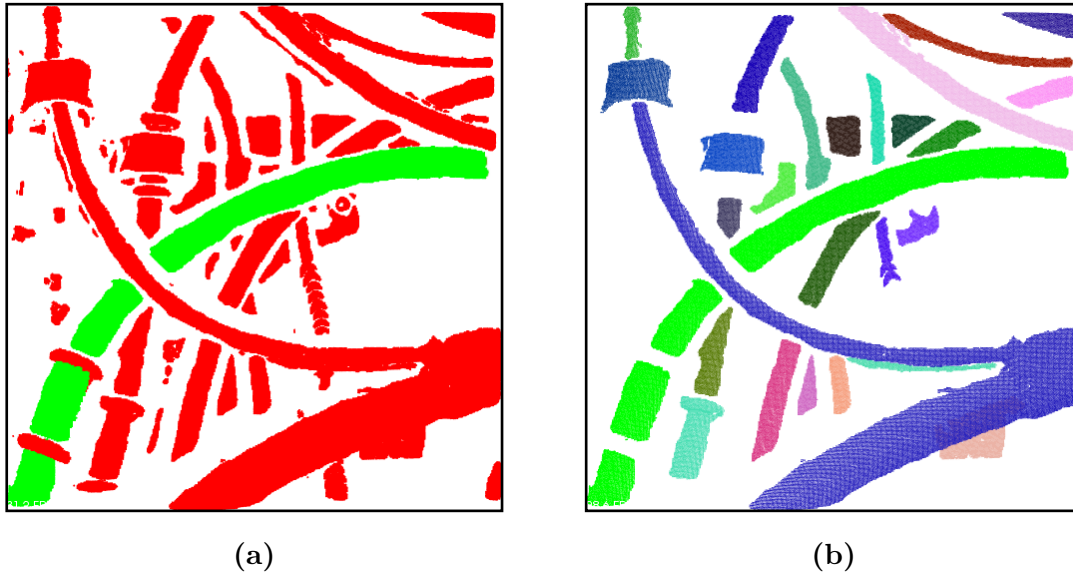


Figure 5.8 – Illustration d'une extraction Euclidienne de *clusters* sur le nuage de points 3D $\overline{\mathcal{P}}_s$: (a) nuage de points d'entrée (le câble \mathcal{P}_s en vert et le nuage de points $\overline{\mathcal{P}}_s$ en rouge), et (b) 31 *clusters* différents, étiquetés \mathcal{C}_1 à \mathcal{C}_{31} . Chaque cluster est représenté par une couleur différente

5.4.4 Détecteur d'interférences

Une fois le regroupement effectué, le nuage est divisé en sous-ensembles (*clusters*) cohérents et significants. À ce stade, nous avons le nuage de points segmenté \mathcal{P}_s (qui représente le câble) et une liste de n *clusters*, que nous appellerons \mathcal{C}_i avec $i = 1 \dots n$. Cette liste de n *clusters* vérifie l'équation suivante :

$$\overline{\mathcal{P}}_s = \sum_{i=1}^n \mathcal{C}_i \quad (5.8)$$

Afin de tester si un câble (ensemble de points 3D $p_i \in \mathcal{P}_s$) et les éléments environnants appartenant à $\overline{\mathcal{P}}_s$ (ensemble de n *clusters*) sont en interférence, nous calculons la distance d_i entre chaque point $p_i \in \mathcal{P}_s$ et son plus proche voisin \tilde{p}_i dans le *cluster* \mathcal{C}_i (voir l'équation (5.9)).

$$d_i = \|p_i - \tilde{p}_i\|_2 \quad (5.9)$$

Si la distance minimale d_i entre le câble et le *cluster* \mathcal{C}_i est inférieure ou égale à la tolérance $d_{\mathcal{T}}$ ($d_i \leq d_{\mathcal{T}}$), alors, le câble est considéré en interférence avec le *cluster* \mathcal{C}_i .

Par exemple, le tableau 5.3 montre les 31 *clusters* différents de la figure 5.8b, étiquetés \mathcal{C}_1 à \mathcal{C}_{31} , et leur détail : libellé, taille et distance d_i par rapport au câble segmenté \mathcal{P}_s .

Table 5.3 – Une liste de 31 *clusters* différents, étiquetés \mathcal{C}_1 à \mathcal{C}_{31} , et leur détail : libellé, taille et distance d_i par rapport au câble segmenté (\mathcal{P}_s)

<i>Cluster</i>	Taille (nombre de points)	Distance d_i
\mathcal{C}_1	17632	15.1
\mathcal{C}_2	8938	35.9
\mathcal{C}_3	6782	66.0
\mathcal{C}_4	6653	124.0
\mathcal{C}_5	4015	9.6
\mathcal{C}_6	2888	119.2
\mathcal{C}_7	1871	37.1
\mathcal{C}_8	1824	52.3
\mathcal{C}_9	1787	13.3
\mathcal{C}_{10}	1740	52.8
\mathcal{C}_{11}	1698	30.5
\mathcal{C}_{12}	1685	31.8
\mathcal{C}_{13}	1642	38.7
\mathcal{C}_{14}	1520	39.7
\mathcal{C}_{15}	1251	32.3
\mathcal{C}_{16}	1059	65.2
\mathcal{C}_{17}	960	51.2
\mathcal{C}_{18}	943	18.2
\mathcal{C}_{19}	932	121.3
\mathcal{C}_{20}	913	10.2
\mathcal{C}_{21}	889	36.1
\mathcal{C}_{22}	842	45.0
\mathcal{C}_{23}	786	52.3
\mathcal{C}_{24}	739	40.5
\mathcal{C}_{25}	719	58.0
\mathcal{C}_{26}	674	47.8
\mathcal{C}_{27}	627	47.5
\mathcal{C}_{28}	601	27.3
\mathcal{C}_{29}	600	32.5
\mathcal{C}_{30}	569	76.9
\mathcal{C}_{31}	511	75.3

La figure 5.9, montre les résultats de la détection d'interférences pour une tolérance $d_T = 12$ mm. Le câble segmenté (en vert) est en interférence avec deux *clusters* \mathcal{C}_5 ($d_5 = 9.6$ mm) et \mathcal{C}_{20} ($d_{20} = 10.2$ mm).

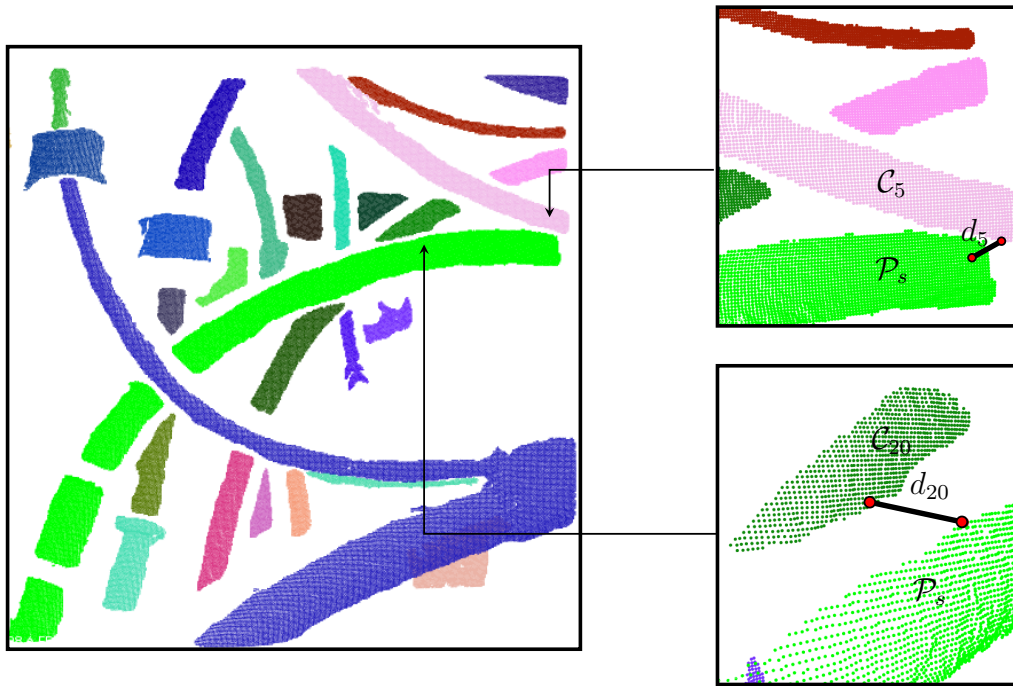
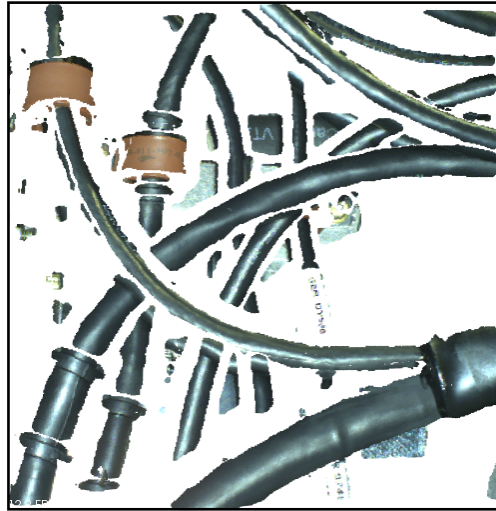
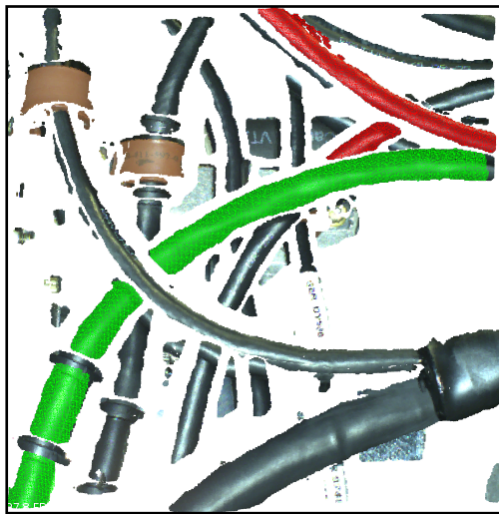


Figure 5.9 – Résultats de la détection d'interférences pour une tolérance $d_T = 12$ mm. Le câble segmenté (en vert) est en interférence avec deux *clusters* \mathcal{C}_5 ($d_5 = 9.6$ mm) et \mathcal{C}_{20} ($d_{20} = 10.2$ mm)

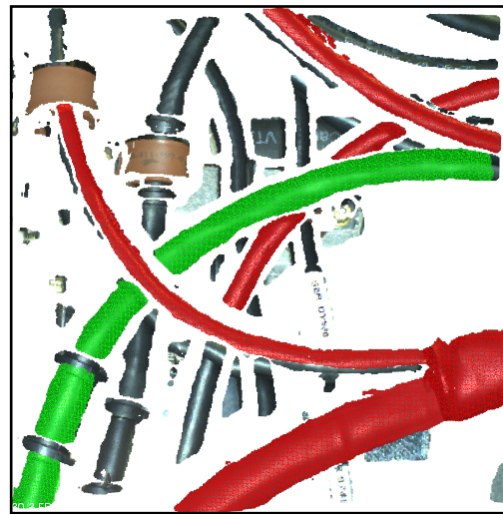
La figure 5.10 montre les résultats finaux du processus de détection d'interférences obtenus pour deux valeurs différentes du seuil de tolérance ($d_T = 12$ mm et $d_T = 20$ mm). Pour $d_T = 12$ mm deux éléments ont été détectés en interférence (C_5 et C_{20} dans le tableau 5.3) alors que pour $d_T = 20$ mm cinq éléments ont été détectés en interférence (C_1, C_5, C_9, C_{18} , et C_{20} dans le tableau 5.3).



(a)



(b) $d_T = 12$ mm



(c) $d_T = 20$ mm

Figure 5.10 – Résultat du processus de détection d'interférences : (a) nuage de points d'entrée, et (b-c) nuage de points segmenté (câble) en vert et les clusters qui causent des problèmes d'interférences (en rouge) pour deux valeurs différentes du seuil de tolérance

Les résultats expérimentaux de la méthode présentée dans cette section vont être présentés dans la section suivante.

5.5 Résultats expérimentaux

Dans l'objectif d'évaluer les performances de nos méthodes d'inspection, présentées dans ce chapitre, nous avons procédé à des expérimentations et évaluations exhaustives sur des données synthétiques que nous avons générées à partir des modèles CAO.

Cette section sera consacrée à l'évaluation des méthodes d'inspection proposées. La première partie de l'évaluation s'intéressera aux résultats produits par la méthode de calcul du rayon de courbure des câbles présentée dans la section 5.3, tandis que la seconde partie se penchera sur les résultats finaux de la méthode de détection d'interférences entre les composants d'un assemblage mécanique dans une scène complexe (présentée dans la section 5.4).

Nous présenterons d'abord les conditions expérimentales dans lesquelles les expérimentations ont été réalisées (section 5.5.1). Ensuite, nous rappellerons les données utilisées, notamment les modèles CAO 3D et les nuages de points synthétiques testés (section 5.5.2). Nous présenterons ensuite dans la section 5.5.3 une évaluation exhaustive de la performance de la méthode de calcul du rayon de courbure. Enfin, nous présenterons les résultats de la méthode de détection d'interférences entre les composants d'un assemblage mécanique complexe (section 5.5.4).

5.5.1 Conditions expérimentales

L'implémentation des algorithmes d'inspection présentés dans ce chapitre est effectuée en C++ et en Python en utilisant la bibliothèque *Point Cloud Library* (*PCL*). La méthode de segmentation proposée a été évaluée en utilisant une machine ayant un processeur Quad-core i7-6700 cadencé à 2.6 Ghz et équipé de 16 Go de RAM.

Dans les expériences, les nuages de points ont été acquis en utilisant le scanner 3D Ensenso N35 porté par un robot mobile. Ce scanner fonctionne selon le principe de la stéréovision. Il dispose de deux caméras CMOS et d'un projecteur intégré, créant un motif aléatoire de points sur l'objet à saisir. En outre, cette technologie contient un système actif avec encodage spatial permettant la capture « *one-shot* ». Les caractéristiques techniques du scanner 3D sont présentées dans la section 1.5.2.

5.5.2 Présentation des données

Ce chapitre présente une application de la méthode de segmentation exposée dans le chapitre 4. Par conséquent, les données utilisées dans les résultats expérimentaux qui vont suivre sont les mêmes que celles présentées dans la section 4.6.

5.5.3 Évaluation de la méthode de calcul du rayon de courbure des câbles

Comme il n'y a pas de système d'instrumentation pour mesurer avec précision le rayon de courbure du câble, nous avons décidé de tester notre algorithme sur des

données synthétiques simulées à partir du modèle CAO. Pour ce faire, nous avons utilisé la méthode d'échantillonnage uniforme pondéré présentée dans le chapitre 4 pour générer différents nuages de points synthétiques.

La figure 5.11 montre un exemple de données synthétiques utilisées dans nos simulations.

Selon les normes de sécurité du câble, le rayon de courbure recommandé pour les harnais (faisceaux de câble) et les câbles dans les assemblages mécaniques aéronautiques ne devrait pas être inférieur à 10 fois le diamètre du plus gros câble. Dans ce cas, $r_c(\min) = 10 \times d$.

Le tableau 5.4 indique le rayon de courbure minimal de chaque câble présenté sur la figure 5.11.

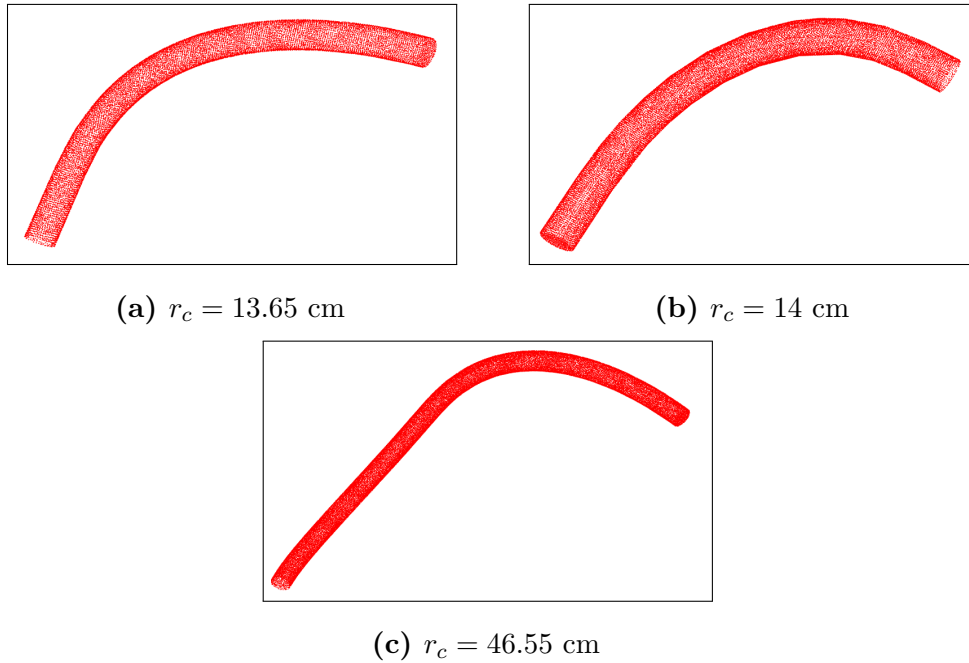


Figure 5.11 – Exemple de données simulées utilisées : (a-c) trois câbles synthétiques différents avec trois rayons de courbure différents (r_c)

Table 5.4 – Le rayon de courbure minimal $r_c(\min)$ autorisé pour chaque câble présenté sur la figure 5.11

câble	rayon du câble r	$r_c(\min)$
câble 1 (figure 5.11a)	3.00 mm	30 mm
câble 2 (figure 5.11b)	5.00 mm	50 mm
câble 3 (figure 5.11c)	7.00 mm	70 mm

5.5.3.1 Métriques d'évaluation pour l'analyse de performances

L'évaluation d'un algorithme d'inspection consiste à confronter les résultats retournés par l'algorithme et la vérité terrain. Pour ce faire, nous avons utilisé la métrique d'erreur quadratique moyenne (MSE , *Mean Square Error*) qui est un standard de l'industrie. Il s'agit d'une mesure de distance entre le rayon de courbure réel r_c (vérité du terrain) et le rayon de courbure prédit \tilde{r}_c (retourné par l'algorithme).

$$MSE_{r. \text{ courbure}} = \frac{1}{n} \sum_{i=1}^n (r_{c_i} - \tilde{r}_{c_i})^2 \quad (5.10)$$

Remarque : plus la valeur de la métrique MSE est petite, meilleure est la précision de l'algorithme. Un algorithme avec des prédictions parfaitement correctes aurait une métrique MSE proche de 0.

Afin d'évaluer la performance de notre algorithme, nous avons utilisé les indicateurs de succès présentés dans le chapitre 3 : la précision, le rappel et la F_{mesure} .

Remarque : un « bon » algorithme de détection de défauts doit présenter d'une part un rappel élevé et, d'autre part, un taux de faux positifs faible. Autrement dit, la précision et le rappel doivent être proches de 100 %.

5.5.3.2 Résultats de l'évaluation

Afin de calculer le rayon de courbure du câble, nous ajustons un plan 3D au sens des moindres carrés sur l'ensemble de données P^i (points initiaux) et P^f (points finaux) du modèle final du câble. Ensuite, nous projetons l'ensemble de données P^i (points initiaux) et P^f (points finaux) du modèle final du câble sur ce plan 3D. Enfin, nous ajustons un cercle 2D au sens des moindres carrés sur l'ensemble de points projetés. Par conséquent, la précision de la mesure est directement liée aux robustesses respectives de l'algorithme de modélisation, de l'algorithme d'ajustement du plan 3D et de l'algorithme d'ajustement du cercle.

Sur un ensemble de 25 nuages de points différents, nous avons obtenu une erreur $MSE_{b. \text{ radius}}$ voisine de 1.1 mm. Cette erreur de mesure est très faible par rapport à un rayon de courbure minimal moyen de 50 mm.

Le tableau 5.5 donne la précision, le rappel et la fonction F_{mesure} , calculés sur l'ensemble des données utilisées.

Une analyse du tableau 5.5 nous montre que notre système se caractérise par un rappel très élevé (100 %), quel que soit le câble à inspecter. Autrement dit, la capacité de l'algorithme à détecter les défauts est très bonne ($FN = 0$).

Table 5.5 – Résultats des indicateurs de succès obtenus sur l'ensemble des données

Précision	Rappel	F_β		
		$\beta = 0.5$	$\beta = 1.0$	$\beta = 1.5$
100 %	100 %	100%	100 %	100 %

Remarque : nous avons remarqué que les résultats de mesure sont meilleurs (mesure très proche de la vérité terrain) lorsque le nombre de sous-câbles trouvés est élevé (c.-à-d. l'ensemble de données P^i (points initiaux) et P^f (points finaux) contient plus de points). En effet, l'algorithme d'ajustement du plan 3D ou l'algorithme d'ajustement du cercle 2D est plus précis lorsque l'ensemble de données contient davantage de points.

5.5.4 Évaluation de la méthode de détection d'interférences

Comme nous ne disposons pas de l'instrumentation nécessaire pour mesurer avec une précision millimétrique la distance entre les éléments d'assemblage mécanique, nous avons décidé d'évaluer notre mesure sur des données synthétiques simulées à partir du modèle CAO. Pour ce faire, nous avons utilisé la méthode d'échantillonnage uniforme pondéré présentée dans le chapitre 4 pour générer différents nuages de points synthétiques.

Afin d'évaluer notre méthode, nous avons identifié 4 câbles différents (c_1 , c_2 , c_3 , et c_4) dans un assemblage mécanique complexe (voir la figure 5.12).

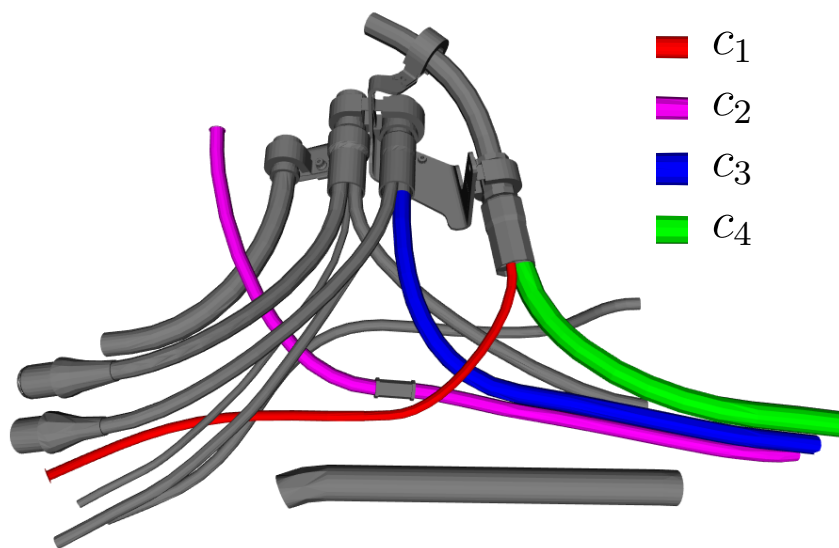


Figure 5.12 – Modèle CAO d'une scène d'un assemblage mécanique. Ce modèle contient 4 câbles différents étiquetés de c_1 à c_4

La figure 5.13 montre un exemple de données synthétiques utilisées dans nos simulations.

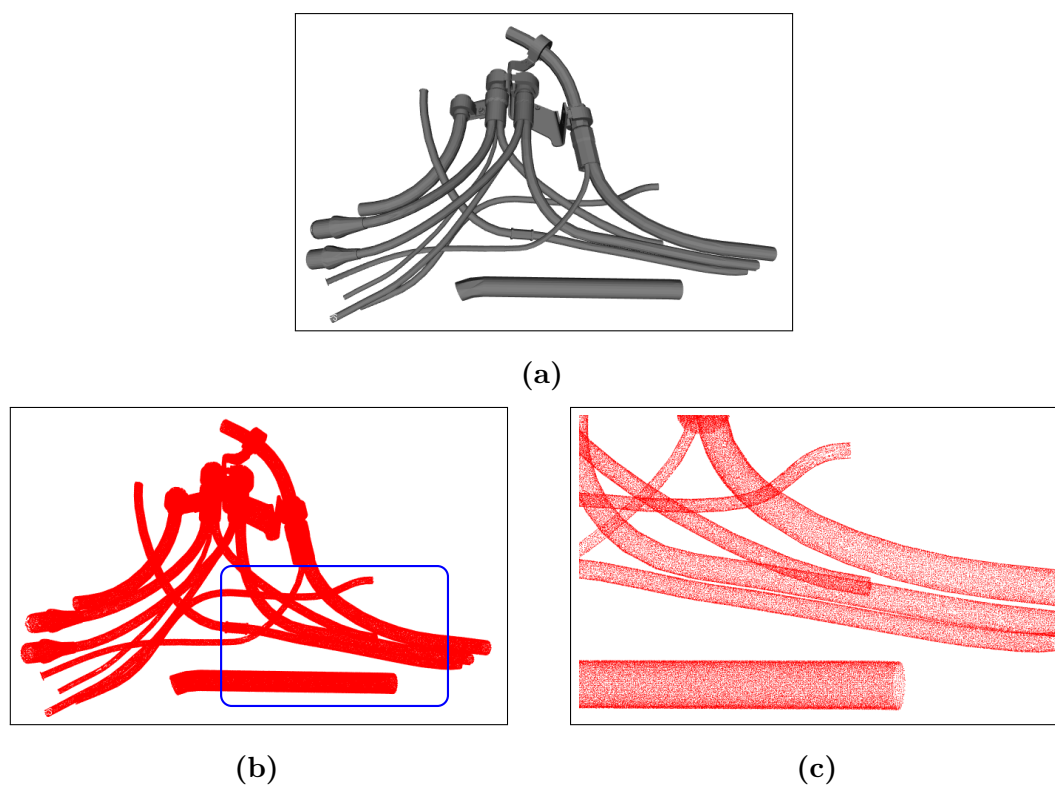


Figure 5.13 – Exemple de données simulées utilisées : (a) modèle CAO, (b) nuage de points synthétique généré à partir du modèle CAO, et (c) zoom de la zone encadrée dans le rectangle bleu

Dans un premier temps, nous avons utilisé le logiciel CATIA[®] pour mesurer les distances entre chaque câble et les éléments environnants dans l'assemblage. Ensuite, nous avons exécuté notre algorithme sur les différents nuages de points simulés. Enfin, nous avons comparé les résultats retournés par notre algorithme et la « vérité terrain » obtenue par le logiciel CATIA[®] (voir un exemple sur la figure 5.14).

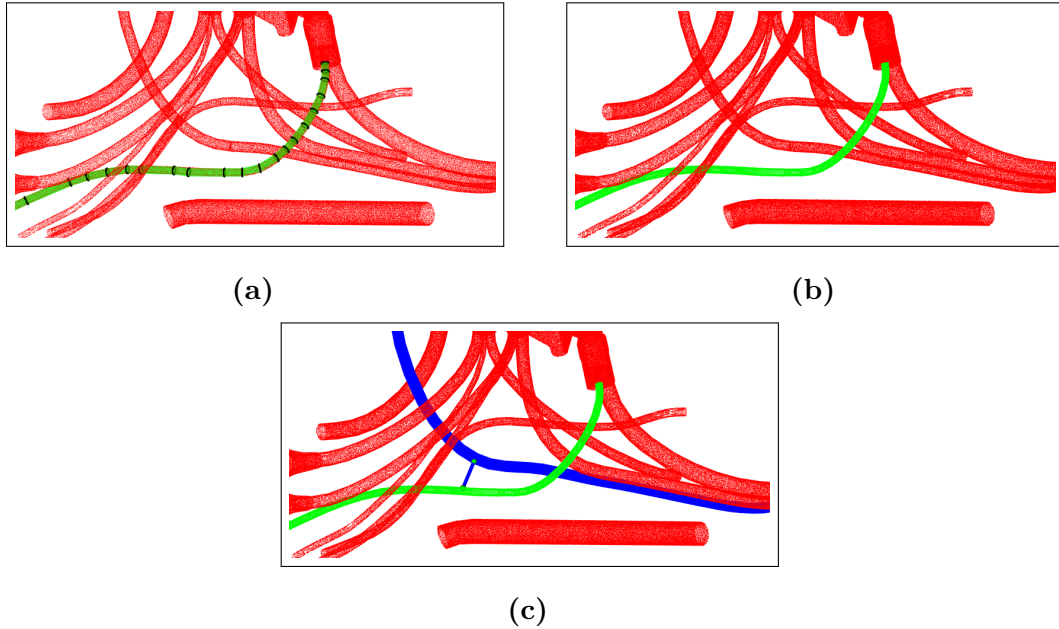


Figure 5.14 – Un exemple de résultat, obtenu sur le câble c_1 : (a) résultat de modélisation, composé de 21 sous-cylindres, (b) résultat de segmentation. Le câble segmenté est en vert, et (c) résultat de la détection d'interférences pour une tolérance $d_T = 42$ mm : le câble segmenté (vert) est en interférence avec le câble c_2 (bleu)

Le tableau 5.6 présente les résultats expérimentaux sur les 4 câbles (c_1 , c_2 , c_3 , et c_4). Ce tableau confirme que la méthode de détection d'interférence proposée est fiable.

Table 5.6 – Comparaison des résultats de mesure entre notre méthode et le logiciel CATIA[®] (distances en mm)

Mesure inter-câbles	Mesure obtenue avec CATIA [®]	Mesure obtenue avec notre algorithme
$c_1 - c_2$	40.7	41.2
$c_1 - c_3$	17.2	17.0
$c_1 - c_4$	0.1	0.2
$c_2 - c_3$	11.3	10.7
$c_2 - c_4$	37.0	37.2
$c_3 - c_4$	11.3	11.1

5.6 Conclusion

Dans cette section, nous présentons une synthèse de nos travaux sur l'inspection 3D des câbles et nous proposons des pistes pour la poursuite de ces travaux.

5.6.1 Synthèse des travaux

Dans ce chapitre, nous avons présenté une application de la méthode de segmentation présentée dans le chapitre 4.

Tout d'abord, nous avons présenté une méthode pour mesurer le rayon de courbure des câbles. Cette méthode se déroule en trois étapes : (i) ajustement d'un plan 3D au sens des moindres carrés sur l'ensemble de données P^i (points initiaux) et P^f (points finaux) du modèle final du câble. (ii) projection de l'ensemble de données P^i (points initiaux) et P^f (points finaux) du modèle final du câble sur le plan déterminé dans l'étape précédente. (iii) ajustement d'un cercle 2D au sens des moindres carrés sur l'ensemble des points projetés. Le rayon du cercle fournit le rayon de courbure du câble.

Ensuite, nous avons présenté un algorithme pour la détection d'interférences entre les câbles et les autres éléments d'assemblage dans une scène complexe. Cette détection est basée sur une mesure de distance 3D et une comparaison avec la valeur de tolérance spécifiée.

Enfin, les algorithmes proposés ont été évalués sur des données synthétiques générées à partir de plusieurs modèles CAO 3D.

Dans la première partie de l'évaluation, nous avons montré l'efficacité de la méthode proposée pour le calcul du rayon de courbure. Dans la seconde partie de l'évaluation, nous avons montré l'efficacité de la méthode de détection d'interférences entre les éléments d'un assemblage mécanique complexe.

5.6.2 Améliorations futures

La solution que nous avons proposée a permis d'inspecter avec efficacité les câbles dans une scène complexe. Nos travaux futurs pourraient porter principalement sur deux aspects :

La méthode mérite d'être évaluée davantage sur de nouveaux ensembles de données synthétiques et réelles pour assurer sa bonne adaptabilité aux différents contextes.

De plus, pour la détection d'interférences, la méthode de clustering basée sur la distance Euclidienne pourrait être améliorée en mettant en œuvre des méthodes d'apprentissage profond telles que les réseaux de neurones. L'idée est de parvenir à trouver automatiquement des paramètres optimaux permettant de diviser le nuage de points en groupes (sous-ensembles) cohérents et significatifs.



Conclusion générale

Travaux réalisés	165
Perspectives	168

Travaux réalisés

Dans cette thèse, nous nous sommes intéressés à l'inspection automatique de pièces aéronautiques complexes, dans un environnement industriel de type atelier d'assemblage, avec l'objectif de proposer des solutions qui répondent au mieux au cahier des charges qui nous avait été fixé :

- inspecter des pièces complexes dans un environnement industriel de type atelier d'assemblage,
- proposer des méthodes rapides et robustes,
- exploiter au mieux les moyens d'inspection développés par DIOTA : une tablette équipée d'une caméra et un robot équipé d'un capteur constitué de 2 caméras et d'un scanner 3D,
- privilégier les méthodes basées sur l'analyse d'images 2D, pour des raisons de rapidité,
- ne pas hésiter à exploiter le scanner 3D pour les types d'inspection qui ne peuvent pas être faits par analyse d'images 2D,
- exploiter au mieux la maquette numérique (modèle CAO 3D) de la pièce inspectée,
- exploiter au mieux le module de localisation temps-réel développé par DIOTA,
- proposer des méthodes robustes même si les données sont parfois incomplètes ou bruitées (cas des nuages de points 3D incomplets en raison d'une acquisition one-shot),

-
- veiller à ce que les algorithmes d’inspection ne renvoient aucun faux négatif (il ne faut rater aucun défaut), tout en minimisant le taux de faux positifs (fausses alarmes).

Nous avons traité différents types d’inspection, tels que la vérification de la présence/absence ou du bon positionnement des éléments dans l’assemblage mécanique, l’inspection des systèmes d’interconnexion de câbles électriques via la mesure du rayon de courbure des câbles, afin de vérifier sa conformité par rapport aux exigences définies par les normes de sécurité, et la vérification que les câbles sont à une distance minimale des autres éléments de l’assemblage (détection d’interférences).

Pour répondre à la problématique posée et au cahier des charges fixé, nous avons proposé plusieurs méthodes : une méthode de choix de point de vue automatique pour préparer l’inspection, des méthodes d’inspection 2D basées sur l’analyse d’images 2D, une méthode d’inspection 3D basée sur l’analyse de nuages de points 3D.

Dans le chapitre 1, nous avons présenté le contexte de la thèse et la problématique à résoudre. Ensuite, nous avons établi un bref état de l’art des méthodes d’inspection par vision 2D/3D. Cette étude bibliographique ne nous a pas permis d’identifier de méthode permettant de répondre entièrement à notre problématique mais nous a permis d’identifier des travaux intéressants dont nous nous sommes inspirés pour développer nos propres travaux.

Dans le chapitre 2, nous avons présenté une méthode de choix automatique de points de vue qui exploite la maquette numérique, et qui a été mise en œuvre suivant 2 scénarios : via un robot équipé de capteurs (mode complètement automatique) ou via une tablette, équipée d’une caméra, portée par un opérateur (mode semi-automatique avec l’option d’une assistance de l’opérateur vers le bon angle de vue en réalité augmentée). Nous générons dans un premier temps une sphère de visibilité centrée sur le centre de gravité du modèle CAO de l’élément à inspecter. Cette sphère de visibilité fournit un ensemble de points de vue candidats qui couvrent toutes les positions possibles de la caméra. Ensuite, une caméra virtuelle est positionnée successivement à chaque point de vue candidat et une prise de vue virtuelle, avec génération d’une image synthétique en basse résolution, est réalisée depuis chacun de ces points de vue. Enfin, nous avons défini une fonction qui attribue un score pour chaque point de vue candidat. Le point de vue qui remporte le meilleur score est choisi comme meilleur point de vue pour l’acquisition des données qui vont être utilisées pour la phase d’inspection ultérieure.

Dans le chapitre 3, nous avons présenté une méthode d’inspection 2D, basée sur l’analyse d’images 2D, pour l’inspection d’assemblages mécaniques complexes (typiquement un moteur d’avion). Cette méthode exploite la maquette numérique et la localisation temps réel de la maquette par rapport aux capteurs. La méthode a été mise en œuvre suivant 2 scénarios : via un robot, équipé de deux caméras, et via une tablette, équipée d’une caméra, portée par un opérateur. La méthode d’inspection est basée sur la mise en correspondance d’edgelets 3D extraits dans la maquette numérique et projetés dans l’image 2D (via la pose) avec des contours 2D extraits dans l’image. Pour s’affranchir des contours parasites et des occultations dans l’étape de mise en correspondance, nous avons développé une approche innovante basée sur

l'analyse d'une image synthétique générée à partir du modèle CAO (image contexte). Le pourcentage d'edgelets qui a pu être mis en correspondance permet de faire un diagnostic sur la conformité de l'assemblage inspecté. Les procédures d'évaluation de la méthode ont été présentées dans la section 3.9. Nous avons atteint un taux de réussite de 98.3% pour le mode d'inspection automatisé via un robot et 94% pour le mode d'inspection interactive via une tablette, tout en gardant les faux négatifs à un taux de 0% dans les deux scénarios. La plupart des algorithmes d'inspection 2D développés durant cette thèse ont été intégrés dans le logiciel DIOTA Inspect® et sont utilisés quotidiennement chez les clients de DIOTA pour réaliser des inspections sur site industriel.

Dans le chapitre 4, nous nous sommes intéressés à la segmentation 3D de formes cylindriques flexibles (des câbles ou des éléments de canalisation) à l'intérieur d'un nuage de points 3D. Nous avons dû faire face à des nuages de points 3D à la fois bruités et incomplets. En effet, dans notre cas, les nuages de points 3D ont été acquis par une méthode de type *one-shot* (une seule acquisition depuis un seul point de vue) qui conduit à des données incomplètes et parfois localement manquantes. Nous avons proposé une méthode originale pour la modélisation et la segmentation des câbles (formes non-rigides). Cette méthode se base sur la détection et la modélisation de portions de cylindres (appelées sous-cylindres), qui sont ensuite étendus par propagation sur le câble dans les deux sens à l'aide d'un algorithme de recherche qui exploite une structure de recherche innovante basée antipodes d'une sphère. La méthode que nous avons développée n'est pas spécifique à la segmentation de câbles et elle pourrait être employée, avec quelques adaptations, dans d'autres contextes applicatifs (e.g. modélisation et segmentation d'une ligne de tuyauterie de grandes structures industrielles, modélisation et segmentation des arbres dans le domaine forestier, ou modélisation et segmentation des artères en imagerie médicale 3D). Contrairement à la plupart des algorithmes de segmentation que nous avons identifiés, la méthode proposée est sans paramètres. Les performances de notre méthode ont été évaluées dans la section 4.6, sur des données réelles et synthétiques : robustesse de la méthode face aux bruits, aux données incomplètes, localement manquantes et aux variations de densité de points. La méthode a été validée dans des environnements industriels. Compte tenu de la complexité des scènes à traiter, nous avons réussi à obtenir des résultats très satisfaisants.

Dans le chapitre 5, nous avons présenté une application de notre méthode de segmentation 3D de câbles pour l'inspection 3D du système d'interconnexion de câblage électrique dans un assemblage aéronautique complexe (moteur d'avion). Nous avons proposé une méthode pour mesurer le rayon de courbure de chaque câble et aussi pour vérifier si les câbles sont à une distance minimale des autres éléments présents dans l'assemblage (détection d'interférence). Les tests ont montré que le taux de détection des défauts de courbure des câbles est de 100%. Pour la détection d'interférences, les tests ont montré un écart moyen de 0.5 mm par rapport aux mesures obtenues avec le logiciel CATIA®.

Nos principales contributions dans ces travaux ont été :

- pour la phase hors ligne de préparation de l'inspection, nous avons proposé un algorithme de choix de points de vue automatique qui exploite **uniquement**

la maquette numérique, et qui peut être mis en œuvre suivant 2 scénarios : via un robot équipé de capteurs (mode d’inspection complètement automatique) ou via une tablette, équipée d’une caméra, portée par un opérateur (mode d’inspection semi-automatique avec l’option d’une assistance de l’opérateur vers le bon angle de vue en réalité augmentée). Contrairement aux approches existantes qui exploitent l’espace objet et qui sont très coûteuses en temps de calcul, l’approche que nous avons proposée exploite l’espace image et elle est donc plus rapide. L’originalité de la méthode proposée est le fait que le meilleur point de vue est déterminé en tenant compte des contours parasites dans le cas de l’inspection 2D. De plus, la fonction score qui permet de déterminer le meilleur point de vue s’adapte au type d’inspection à réaliser.

- pour l’étape d’inspection 2D en ligne, nous avons développé une méthode d’inspection temps réel robuste qui exploite une ou plusieurs images 2D et la maquette numérique. Dans l’étape de mise en correspondance 2D/3D entre les primitives 2D extraites dans l’image et les primitives 3D issues du modèle CAO, nous avons proposé une méthode originale pour éviter les contours parasites et anticiper les occultations. Cette méthode exploite une approche innovante basée sur l’analyse d’une image synthétique générée à partir du modèle CAO, que nous avons désigné par *image contexte*. Elle conduit à un taux de bons appariements plus important.
- pour l’étape d’inspection 3D en ligne, nous avons développé une méthode entièrement automatique et robuste pour la modélisation-segmentation 3D de formes cylindriques flexibles à partir de nuages de points 3D bruités et incomplets. Pour identifier les points 3D d’un câble à l’intérieur d’un nuage de points, cette méthode exploite une structure de recherche innovante basée antipodes d’une sphère. Elle se compose d’une sphère virtuelle et d’un cylindre virtuel rotatif centré sur la sphère virtuelle. La méthode que nous avons développée n’est pas spécifique à la segmentation de câbles et elle pourrait être employée, avec quelques adaptations, dans d’autres contextes applicatifs.

Les travaux réalisés au cours de cette thèse ont donné lieu à deux communications internationales ([Ben+19b ; Ben+19c]), une communication nationale ([Ben+18]) et deux articles de revue ([Ben+19a ; Ben+20]).

Perspectives

Ces travaux de recherche et les différentes thématiques connexes abordées pendant cette thèse ouvrent différentes perspectives de recherche.

Pour l’inspection 2D, bien que ce soit rare, il arrive parfois que la méthode d’inspection échoue lorsque l’élément à inspecter possède la même couleur et luminance que le fond (par exemple éléments noirs sur fond noir). La cause est habituellement une mauvaise détection des contours (difficultés à discriminer les pixels entre eux et de séparer les objets du fond). Ce problème pourrait être pallié en complétant les

informations 2D (image) par une information de profondeur 3D (scanner), à travers un couplage 2D/3D (exploitation conjointe des images 2D et des nuages de points 3D).

Dans le scénario tablette, nous prenons une décision finale sur le statut de l'élément en fusionnant les résultats obtenus à partir de vues multiples. Cette inspection multi-vues n'est possible que dans le scénario tablette. En mode robot, une seule acquisition d'image est effectuée (acquisition en mode one-shot). L'inspection multi-vues est une fonctionnalité à ajouter à notre plateforme robotique pour la suite de nos travaux.

Dans le domaine 3D, nos travaux futurs pourraient porter principalement sur deux aspects :

1. segmentation 3D du câble : dans cette méthode, les temps de calcul n'ont pas été optimisés jusqu'ici, en particulier la propagation de la structure de recherche présentée dans la section 4.5.3. En vue d'améliorer le temps d'exécution, il est possible d'envisager de propager en même temps la structure de recherche virtuelle dans les deux sens à partir des deux extrémités du sous-cylindre initial, en faisant de la parallélisation.
2. mesure du rayon de courbure des câbles et détection d'interférences : la solution que nous avons proposée a permis d'inspecter les câbles dans une scène complexe et les tests que nous avons menés ont montré une bonne robustesse face au bruit et aux variations de densité de points. Cette méthode mérite d'être évaluée davantage sur de nouveaux ensembles de données synthétiques et réelles pour assurer sa bonne adaptabilité aux différents contextes. De plus, pour la détection d'interférences, la méthode de clustering basée sur la distance Euclidienne pourrait être améliorée en mettant en œuvre des méthodes d'apprentissage profond telles que les réseaux de neurones. L'idée est de parvenir à trouver automatiquement des paramètres optimaux permettant de diviser le nuage de points en groupes (sous-ensembles) cohérents et significatifs.

Ces dernières années, les techniques basées sur l'apprentissage automatique ont montré leur grand intérêt pour la résolution de nombreux problèmes dans le domaine de l'analyse d'images 2D ou de l'analyse de nuages de points 3D. Dans le cadre de leur laboratoire de recherche commun, l'Institut Clément Ader et DIOTA ont commencé à explorer ces techniques pour la résolution de problèmes d'inspection par vision 2D et/ou 3D. Pour poursuivre les travaux présentés dans cette thèse, une nouvelle thèse intitulée « Inspection de pièces aéronautiques en exploitant la maquette numérique, des images 2D, des nuages de points 3D et des techniques de Machine Learning » a démarré en octobre 2019.



Extraction des éléments de la maquette numérique

Dans cette thèse, nous considérons que la maquette numérique de l'assemblage mécanique à contrôler est disponible et sert de référence. Cette maquette décrit l'état dans lequel doivent se trouver les éléments à inspecter.

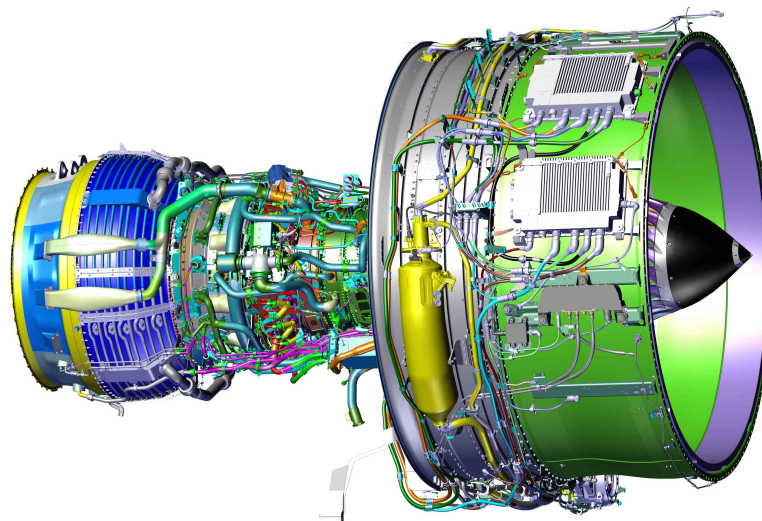
Pour préparer une opération d'inspection, DIOTA a développé un logiciel nommé Diota Connect[®], intégré dans CATIA Composer[®], qui permet de réaliser une opération appelée *préparation de gamme de contrôle*.

Le principe général de la préparation d'une gamme de contrôle est d'identifier dans la maquette numérique globale l'ensemble des éléments qui doivent être contrôlés et de leur affecter une classe de contrôle. Par exemple :

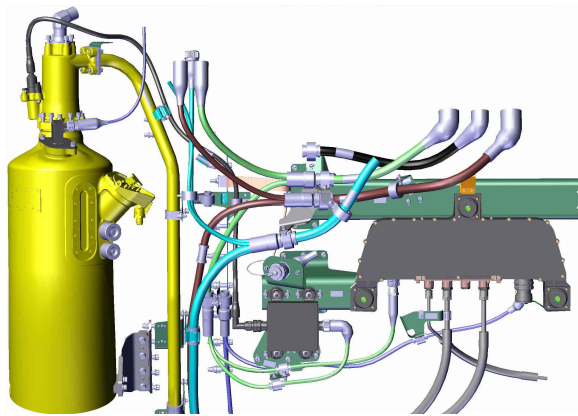
- pour vérifier la présence/absence et le bon positionnement d'un support, nous sélectionnons tous les supports présents dans la maquette numérique, puis nous affectons la classe *presence-absence*. Cette classe va permettre d'exécuter l'algorithme du chapitre 2,
- pour vérifier la conformité du rayon de courbure des câbles ou pour détecter les interférences entre un câble et son environnement, nous sélectionnons tous les câbles présents dans la maquette numérique, ensuite nous assignons la classe *cable*. Cette classe va permettre d'exécuter les algorithmes présentés dans les chapitres 4 et 5.

Cette étape de préparation hors ligne génère les données d'entrée nécessaires pour l'étape d'inspection en ligne qui suit.

A partir de ces données, il est possible d'extraire la maquette numérique restreinte à la zone à inspecter (cf. figure A.1b) ou un élément en particulier (cf. figure A.1c) ou l'élément avec son environnement proche (cf. figure A.1d).



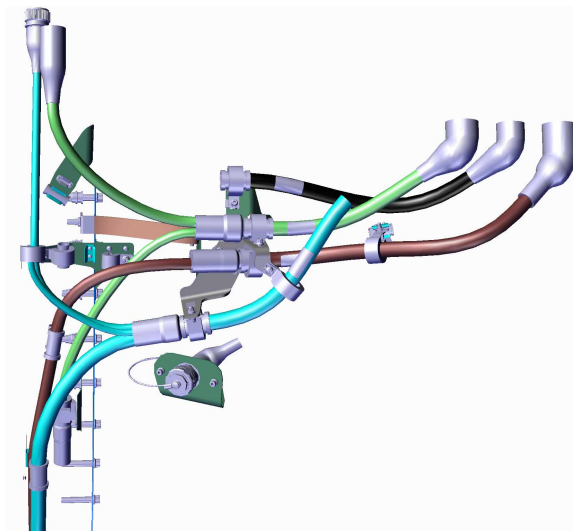
(a)



(b)



(c)



(d)

Figure A.1 – Extraction des éléments de la maquette numérique : (a) maquette globale, (b) maquette restreinte à la zone à inspecter, (c) un élément en particulier (un câble), et (d) l'élément avec son environnement proche

Détection automatique des connecteurs par Machine Learning

B.1	Choix de l'architecture de réseau	174
B.2	Augmentation de données d'apprentissage	174
B.3	Résultats de détection	174
B.4	Prise de décision	175

Dans cette annexe, nous décrivons l'algorithme basé Machine Learning que nous avons développé pour détecter automatiquement les connecteurs des câbles à partir d'images 2D d'un assemblage complexe.

L'objectif de la détection est de trouver dans l'image 2D des boîtes englobantes (régions) contenant l'objet d'intérêt.

D'une façon générale, les détecteurs d'objets sont divisés en deux catégories :

- les détecteur à une classe. Il s'agit de détecter un seul type d'objet,
- les détecteur multi-classes. Il s'agit de détecter plusieurs objets, tout en séparant les objets entre eux.

Dans notre cas précis, nous nous intéressons aux détecteurs d'objets à une classe afin de ne détecter que les connecteurs dans l'image.

Pour ce faire, nous avons décidé d'utiliser une méthode de détection d'objets basée Deep Learning avec un classifieur CNN. Cependant, les difficultés majeures concernant l'apprentissage sont :

- le choix de l'architecture de réseau,
- le manque de données d'apprentissage.

B.1 Choix de l'architecture de réseau

Un détecteur d'objets doit faire face à plusieurs difficultés comme : temps de calcul, robustesse de détection face aux occultations partielles de l'objet, variations de la luminosité, bruits. De plus, ce détecteur doit donner la position précise des objets.

Il existe de nombreux travaux basés sur les réseaux de neurones de convolution pour la détection d'objets [Dor+17]. Les détecteurs CNN les plus utilisés à ce jour sont : YOLO [Red+16], Faster R-CNN [Ren+17], SSD [Liu+16], R-FCN [Dai+16]. Récemment, HUANG et al. [Hua+17] ont proposé une analyse détaillée des différents détecteurs de l'état de l'art basées CNN (e.g. Faster R-CNN [Ren+17], SSD [Liu+16], R-FCN [Dai+16], etc.) en testant différentes architectures convolutives.

Afin de choisir l'approche la plus appropriée à nos données, nous avons testé trois détecteurs d'objets sur les données de DIOTA. Le premier détecteur testé pour la détection de connecteurs est l'algorithme Faster R-CNN proposé par REN et al. [Ren+17]. Le second détecteur testé est l'algorithme YOLO proposé par REDMON et al. [Red+16]. Nous avons testé aussi le détecteur SSD proposé par LIU et al. [Liu+16].

Finalement, nous avons choisi le détecteur SSD [Liu+16], car il permet d'obtenir les meilleurs résultats sur nos données. Les principales différences entre SSD et les autres détecteurs testés sont : le temps du calcul et la robustesse face aux occultations partielles.

B.2 Augmentation de données d'apprentissage

Le manque de données est un des problèmes fréquemment rencontrés dans l'apprentissage profond. Dans l'objectif d'améliorer la performance de détection, nous avons développé un outil d'augmentation de données d'entraînement (*Data Augmentation*). Pour ce faire, nous appliquons aux images d'entraînement des transformations affines (le flip horizontal et/ou vertical, la rotation). Nous avons ajouté également différents niveaux du bruit, contraste, luminosité à nos images d'apprentissage.

Une autre méthode a été proposée récemment par MIRANDA et al. [Mir+19b] pour l'augmentation de données d'apprentissage. Dans leur travail, ils ont abordé le problème du manque de données pour l'apprentissage en faisant un apprentissage sur des données synthétiques réalistes.

B.3 Résultats de détection

Nous avons utilisé 500 images réelles pour l'apprentissage. Ces images ont été augmentées à 5800 images. Nous avons utilisé 350 images de test. Comme on peut le voir sur la figure B.1 le score de confiance (probabilité que la région détectée

contienne ou pas le connecteur) atteint 100 % dans la plupart des tests réalisés. De plus, l'algorithme détecte avec succès les connecteurs partiellement occultés.

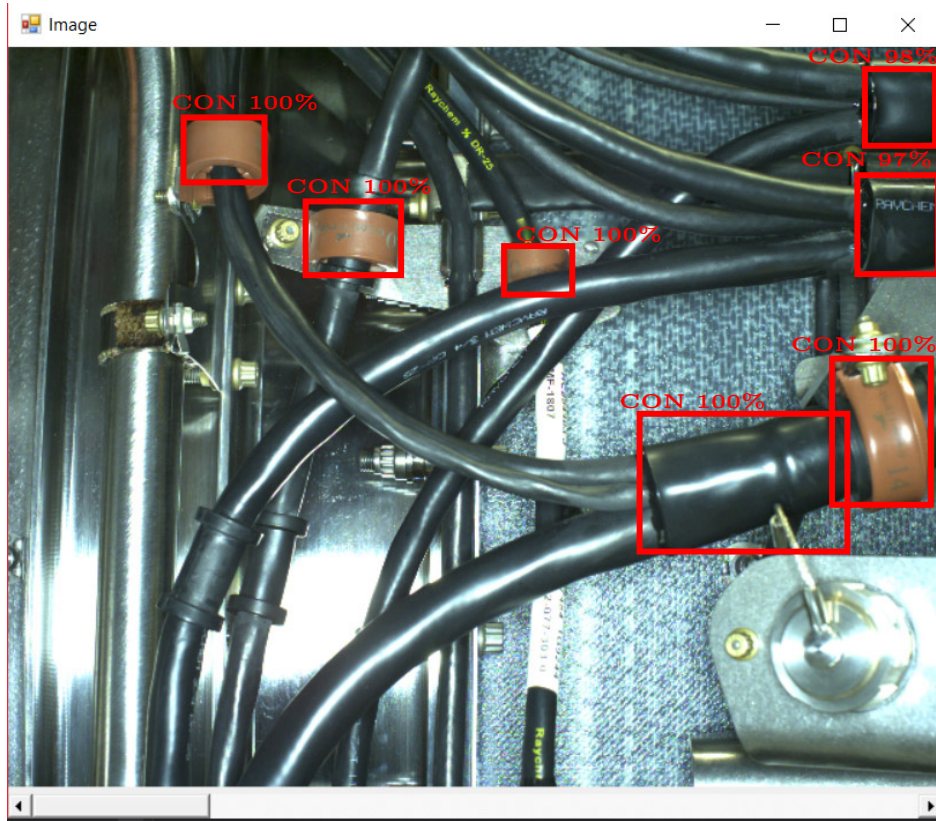
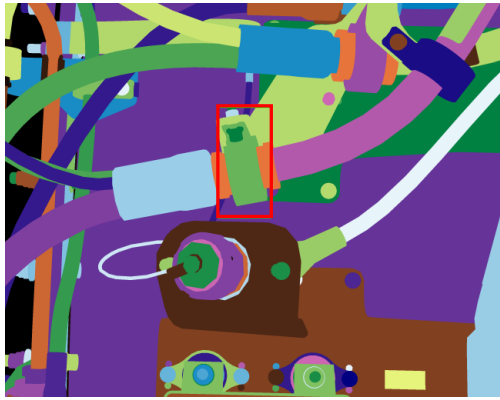


Figure B.1 – Un exemple de résultat obtenu avec le détecteur SSD

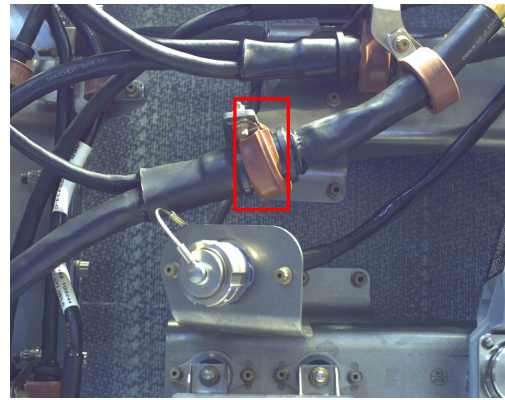
B.4 Prise de décision

L'étape de prise de décision en ligne doit indiquer *in fine* si le connecteur inspecté est présent ou non.

Pour ce faire, à partir du modèle CAO, nous générons une image synthétique correspondant au même point de vue que l'image analysée (cf. figure B.2a). À partir de cette image synthétique, nous générons un rectangle qui englobe le connecteur d'intérêt. Ce rectangle englobant est appliqué sur l'image réelle (cf. figure B.2b), ce qui permet de définir la zone de l'image qui est fournie à l'algorithme de détection par Deep Learning. Si l'algorithme détecte un connecteur, cela signifie qu'il est présent.



(a)



(b)

Figure B.2 – Exemple d’inspection d’un connecteur spécifié. (a) image synthétique et région d’intérêt générés à partir de la maquette numérique, (b) région d’intérêt pour la détection du connecteur par Deep Learning

Nous avons testé notre méthode sur 1284 images contenant 11 connecteurs différents (tous les connecteurs ne sont pas présents dans toutes les images). Le tableau B.1 donne la matrice de confusion obtenue (cf. tableau 3.3). On peut voir que les résultats sont très bons, avec un taux de faux négatifs égal à zéro, une précision de 95.3% et un rappel de 100%.

A titre d’exemple, la figure B.3 donne quelques exemples des images qui ont été analysées.

Table B.1 – Matrice de confusion obtenue

Nombre de connecteurs	Nombre d’images	VP	VN	FP	FN
11	1284	100%	94%	4.9%	0%

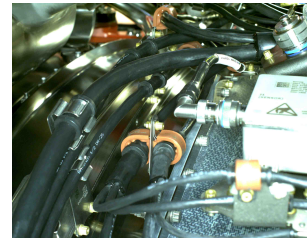
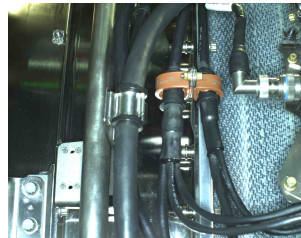
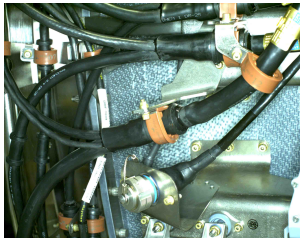


Figure B.3 – Quelques exemples d’images qui ont été analysées



Bibliographie

- [Abu19] A. Abubakr. “Industrial Inspection using Deep Learning and CAD model.” Master Thesis. MSc in Informatics at Grenoble (MoSIG - M2). Specialization: Graphics, Vision, and Robotics, 2019 (cit. on p. 15).
- [And+04] C. Andujar, P.-P. Vázquez Alcocer, and M. Fairen. “[Way-Finder: Guided Tours Through Complex Walkthrough Models](#).” In: *Computer Graphics Forum* 23 (Sept. 2004), pp. 499–508 (cit. on p. 32).
- [Ans+17] M. Ansari, S. Zai, and Y. Moon. “[Automatic segmentation of coronary arteries from computed tomography angiography data cloud using optimal thresholding](#).” In: *Optical Engineering* 56 (Jan. 2017), p. 013106 (cit. on p. 138).
- [Art+07] D. Arthur and S. Vassilvitskii. “K-Means++: The Advantages of Careful Seeding.” In: *Proceedings of the 18th annual ACM-SIAM symposium on Discrete algorithms*. Vol. 8. New Orleans, Louisiana, July 2007, pp. 1027–1035 (cit. on p. 150).
- [Bay+08] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. “[Speeded-up robust features \(SURF\)](#).” In: *Computer Vision and Image Understanding* 110 (June 2008), pp. 346–359 (cit. on p. 27).
- [Bel+00] S. Belongie and J. Malik. “Matching with shape contexts.” In: *Proceedings Workshop on Content-based Access of Image and Video Libraries*. Hilton Head Island, SC, USA, June 2000, pp. 20–26 (cit. on p. 66).
- [Bel+02] S. Belongie, J. Malik, and J. Puzicha. “[Shape matching and object recognition using shape contexts](#).” In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24.4 (Apr. 2002), pp. 509–522 (cit. on p. 66).
- [Bel+16] M. Belkacemi, C. Stolz, A. Mathieu, G. Lemaitre, and O. Aubreton. “A combined three-dimensional digitisation and subsurface defect detection data using active infrared thermography.” In: *13th Quantitative Infrared Thermography Conference (QIRT)*. Gdansk, Poland, July 2016 (cit. on p. 8).

-
- [Ben+18] H. Ben Abdallah, I. Jovancevic, J.-J. Orteu, L. Brèthes, and B. Dolives. “Inspection automatisée d’assemblages mécaniques aéronautiques par vision 2D/3D en exploitant la maquette numérique.” In: *Congrès Optique Toulouse 2018*. Oral presentation only. Toulouse, France, Mar. 2018 (cit. on p. 168).
- [Ben+19a] H. Ben Abdallah, I. Jovancevic, J.-J. Orteu, and L. Brèthes. “Automatic Inspection of Aeronautical Mechanical Assemblies by Matching the 3D CAD Model and Real 2D Images.” In: *Journal of Imaging* 5.10 (Oct. 2019), pp. 81–108 (cit. on pp. 29, 48, 168).
- [Ben+19b] H. Ben Abdallah, J.-J. Orteu, B. Dolives, and I. Jovancevic. “3D Point Cloud Analysis for Automatic Inspection of Aeronautical Mechanical Assemblies.” In: *Proceedings of 14th International Conference on Quality Control by Artificial Vision (QCAV)*. Mulhouse, France, 15-17 May 2019 (cit. on p. 168).
- [Ben+19c] H. Ben Abdallah, J.-J. Orteu, I. Jovancevic, B. Dolives, and L. Brèthes. “Automatic Inspection of Aeronautical Mechanical Assemblies using 2D and 3D computer vision.” In: *Proceedings of 11th International Symposium on NDT in Aerospace*. Paris-Saclay, France, 13-15 November 2019 (cit. on p. 168).
- [Ben+20] H. Ben Abdallah, J.-J. Orteu, I. Jovancevic, L. Brèthes, and B. Dolives. “3D Point Cloud Analysis for Automatic Inspection of Complex Aeronautical Mechanical Assemblies.” In: *Journal of Electronic Imaging* accepted (2020) (cit. on pp. 90, 168).
- [Ben15] A Bennis. “Contribution à la reconstruction 3D de bâtiments à partir de nuage de points de scanner laser terrestre.” Thèse. Université de Lorraine, Oct. 2015 (cit. on p. 12).
- [Ber+16] L.-P. Bergé, T. Duval, N. Aouf, and G. Coppin. “Visualisation de nuages de points 3D acquis par un laser : impact de la vision stéréoscopique et du suivi de la tête sur une recherche de cible.” In: *Actes de la 28ième conférence francophone sur l’Interaction Homme-Machine*. Fribourg, Switzerland, Oct. 2016, pp. 246–251 (cit. on p. 95).
- [Ber13] D. Berndt. *Digital Assembly Inspection: Automatic Quality Control Even for Small Quantities*. 2013. URL: <https://idw-online.de/en/news532655> (visited on 02/14/2018) (cit. on p. 10).
- [Ber93] M. Berg. *Ray Shooting, Depth Orders and Hidden Surface Removal*. Berlin, Heidelberg: Springer-Verlag, 1993 (cit. on p. 62).
- [Bes+92] P. J. Besl and N. D. McKay. “A method for registration of 3-D shapes.” In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14.2 (Feb. 1992), pp. 239–256 (cit. on pp. 12, 99, 110).
- [Bil+00] M. Billinghurst, I. Poupyrev, H. Kato, and R. May. “Mixing Realities in Shared Space: An Augmented Reality Interface for Collaborative Computing.” In: *Proceedings of IEEE International Conference on Multimedia and Expo. (ICME2000). Latest Advances in the Fast Changing World of Multimedia*. New York, NY, USA, 30 July - 2 August 2000, pp. 1641–1644 (cit. on p. 23).
- [Bre01] L. Breiman. “Random Forests.” In: *Machine Learning* 45.1 (Oct. 2001), pp. 5–32 (cit. on p. 78).

- [Can86] J. Canny. “A Computational Approach to Edge Detection.” In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-8.6 (Nov. 1986), pp. 679–698 (cit. on pp. 76, 77).
- [Car76] M. do Carmo. *Differential geometry of curves and surfaces*. Prentice Hall, 1976, pp. 113–191 (cit. on p. 113).
- [Cas19] N. Castet. “L’intelligence artificielle pour faciliter les activités d’inspection.” In: *Matlab Expo 2019*. Paris, France, 18 Juin 2019 (cit. on p. 15).
- [Cha+01] T. Chaperon and F. Goulette. “Extracting Cylinders in Full 3D Data Using a Random Sampling Method and the Gaussian Image.” In: *Proceedings of the Vision Modeling and Visualization Conference*. Stuttgart, Germany, 21-23 November 2001, pp. 35–42 (cit. on pp. 14, 15).
- [Cha+13] K. Changmin, S. Hyojoo, and K. Changwan. “Fully automated registration of 3D data to a 3D CAD model for project progress monitoring.” In: *Automation in Construction* 35 (Nov. 2013), pp. 587–594 (cit. on p. 99).
- [Cha02] T. Chaperon. “Segmentation de nuage de points 3D pour la modélisation automatique d’environnements industriels numérisés.” Thèse. École Nationale Supérieure des Mines de Paris, Sept. 2002 (cit. on pp. 14, 113, 114).
- [Che+16] L.-C. Chen, J. Barron, G. Papandreou, K. Murphy, and A. Yuille. “Semantic Image Segmentation with Task-Specific Edge Detection Using CNNs and a Discriminatively Trained Domain Transform.” In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2016 (cit. on p. 78).
- [Dai+16] J. Dai, Y. Li, K. He, and J. Sun. “R-FCN: Object Detection via Region-based Fully Convolutional Networks.” In: *30th Conference on Neural Information Processing Systems (NIPS 2016)*. Barcelona, Spain, 2016. arXiv: [1605.06409](https://arxiv.org/abs/1605.06409) (cit. on p. 174).
- [Dav+04] L. Davis, F. Hamza Lup, and J. Rolland. “A Method for Designing Marker-Based Tracking Probes.” In: *Proceedings of the Third IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR)*. Arlington, VA, USA, Nov. 2004, pp. 120–129 (cit. on p. 23).
- [Des+15] R. Destrez, S. Treuillet, and Y. Lucas. “Joint detection of anatomical points on surface meshes and color images for visual registration of 3D dental models.” In: *Proceedings SPIE 9534, 12th International Conference on Quality Control by Artificial Vision (QCAV)*. Le Creusot, France, Mar. 2015 (cit. on p. 10).
- [Dor+17] A. Dore, M. Devy, and A. Herbulot. “Détection d’objets en milieu naturel : application à l’arboriculture.” In: *16ième Journées Francophones des Jeunes Chercheurs en Vision par Ordinateur (ORASIS 2017)*. GREYC. Colleville-sur-Mer, France, June 2017 (cit. on p. 174).
- [Duj13] M. Dujardin. “Le scanner laser 3D : reconnaissance de formes et modélisation de déformations.” Mémoire CNAM. ESGT-CNAM - École supérieure des géomètres et topographes, Dec. 2013 (cit. on p. 129).

-
- [Ell+17] W. Elloumi, S. Treuillet, and R. Leconge. “[Real-time camera orientation estimation based on vanishing point tracking under Manhattan World assumption.](#)” In: *Journal of Real-Time Image Processing* 13.4 (Dec. 2017), pp. 669–684 (cit. on p. 25).
- [Ell12] W. Elloumi. “Contributions à la localisation de personnes par vision monoculaire embarquée.” Thèse. Université d’Orléans, Dec. 2012 (cit. on p. 26).
- [Fia05] M. Fiala. “ARTag, a fiducial marker system using digital techniques.” In: *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05)*. San Diego, CA, USA, 20–25 June 2005, pp. 590–596 (cit. on p. 23).
- [Fit02] A. Fitzgibbon. “[Robust Registration of 2D and 3D Point Sets.](#)” In: *Image and Vision Computing* 21 (Apr. 2002), pp. 1145–1153 (cit. on p. 99).
- [Fló06] L. Flórez Valencia. “Modèle d’état de cylindre généralisé et la quantification de sténoses artérielles en imagerie 3D.” Thèse. INSA Lyon, July 2006 (cit. on p. 14).
- [Fol+90] J. D. Foley, A. van Dam, S. K. Feiner, and J. F. Hughes. *Computer Graphics: Principles and Practice (2Nd Ed.)* Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1990 (cit. on p. 32).
- [Gél+17a] W. Gélard, M. Devy, A. Herbulot, and P. Burger. “Model-based Segmentation of 3D Point Clouds for Phenotyping Sunflower Plants.” In: *12th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISAPP 2017)*. Vol. 4. 2017, pp. 459–467 (cit. on p. 15).
- [Gél+17b] W. Gélard, A. Herbulot, M. Devy, P. Debaeke, R. McCormick, S. Truong, and J. Mullet. “Leaves Segmentation in 3D Point Cloud.” In: *International Conference on Advanced Concepts for Intelligent Vision Systems*. Antwerp, Belgium: Springer International Publishing, 18–21 September 2017, pp. 664–674 (cit. on p. 15).
- [Gél18] W. Gélard. “Modélisation 3D et suivi visuel pour caractériser le phénotype de variétés de tournesol.” Thèse. Université Paul Sabatier, Toulouse, Dec. 2018 (cit. on p. 151).
- [Ghi18] P. Ghimire. “Matching Renders with Real Images using Learned Features for Industrial Visual Inspection.” Master Thesis. Erasmus Mundus VIBOT Master, 2018 (cit. on p. 15).
- [Hac+14] J. Hackenberg, C. Morhart, J. Sheppard, H. Spiecker, and M. Disney. “[Highly Accurate Tree Models Derived from Terrestrial Laser Scan Data: A Method Description.](#)” In: *Forests* 5 (May 2014), pp. 1069–1105 (cit. on p. 138).
- [Hac+15] J. Hackenberg, H. Spiecker, K. Calders, M. Disney, and P. Raunonen. “[SimpleTree – An Efficient Open Source Tool to Build Tree Models from TLS Clouds.](#)” In: *Forests* 6 (Nov. 2015), pp. 4245–4294 (cit. on pp. 14, 15).
- [Had13] A. Haddeji. “Construction d’un Atlas 3D numérique de la cornée humaine par recalage d’images.” Master Thesis. Université de Montréal, 2013 (cit. on p. 111).

- [Ham+03] G. Hamerly and C. Elkan. “Learning the K in K-Means.” In: *Proceedings of the 16th International Conference on Neural Information Processing Systems (NIPS’2003)*. Whistler, British Columbia, Canada, Sept. 2003 (cit. on p. 150).
- [Har+08] J. Harvent, F. Bugarin, J.-J. Orteu, M. Devy, P. Barbeau, and G. Marin. “Inspection de pièces aéronautiques pour la détection de défauts de forme à partir d’un système multi-caméras.” In: *Journées COFREND : conférences et exposition sur les essais non destructifs*. Toulouse, France, May 2008 (cit. on p. 13).
- [Har+79] J. A. Hartigan and M. A. Wong. “[Algorithm AS 136: A K-Means Clustering Algorithm](#).” In: *Applied Statistics* 28.1 (1979), pp. 100–108 (cit. on p. 150).
- [Har+88] C. Harris and M. Stephens. “A Combined Corner and Edge Detector.” In: *Proceedings of the 4th Alvey Vision Conference*. Vol. 1988. Manchester, 31 August-2 September 1988, pp. 147–151 (cit. on pp. 24, 25).
- [Hav18] V. Havard. “Développement de méthodes et outils basés sur la réalité augmentée et virtuelle pour l’assistance ou l’apprentissage d’opérations dans un contexte industriel.” Thèse. Université de Normandie, Feb. 2018 (cit. on pp. 23–25).
- [Her+19] T. Herrmann, C. Migniot, and O. Aubreton. “Cracks detection on glass object based on active thermography approach.” In: *14th International Conference on Quality Control by Artificial Vision (QCAV)*. Vol. 11172. Mulhouse, France: SPIE, July 2019 (cit. on p. 8).
- [Her99] A. Hertzmann. “Introduction to 3D Non-Photorealistic Rendering: Silhouettes and Outlines.” In: *Non-Photorealistic Rendering, SIGGRAPH Course Notes*. 1999 (cit. on pp. 38, 59).
- [Hon+14] P. Hong-Seok and U. Tuladhar. “[Development of an Inspection System for Defect Detection in Pressed Parts Using Laser Scanned Data](#).” In: *Procedia Engineering* 69 (Dec. 2014) (cit. on p. 13).
- [Hor+95] R. Horaud and O. Monga. *Vision par ordinateur : outils fondamentaux*. Traité des nouvelles technologies. Hermès, 1995 (cit. on pp. 23, 54).
- [Hou+62] V. Hough and C. Paul. *Method And Means For Recognizing Complex Patterns*. US Patent US1771560A. Dec. 1962 (cit. on p. 24).
- [Hua+17] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, and K. Murphy. “Speed/accuracy trade-offs for modern convolutional object detectors.” In: *Proceedings of Computer Vision and Pattern Recognition (CVPR)*. 2017. arXiv: [1611.10012](#) (cit. on p. 174).
- [ICE08] ICEA. *Portable and Power Feeder Cables for Use in Mines and Similar Applications*. Tech. rep. ICEA S-75-381. Insulated Cable Engineers Association (ICEA), May 2008 (cit. on p. 145).
- [Jin+19] Y.-H. Jin and W.-H. Lee. “[Fast Cylinder Shape Matching Using Random Sample Consensus in Large Scale Point Cloud](#).” In: *Applied Sciences* 9.5 (Mar. 2019), p. 974 (cit. on p. 14).

-
- [Jov+15] I. Jovančević, J.-J. Orteu, T. Sentenac, and R. Gilblas. “Inspection d’un aéronef à partir d’un système multi-capteurs porté par un robot mobile.” In: *14ème Colloque Méthodes et Techniques Optiques pour l’Industrie*. Lannion, France, Nov. 2015 (cit. on p. 9).
- [Jov+17] I. Jovancevic, H.-H. Pham, J.-J. Orteu, R. Gilblas, J. Harvent, X. Maurice, and L. Brèthes. “3D point cloud analysis for detection and characterization of defects on airplane exterior surface.” In: *Journal of Non Destructive Evaluation* 36.4 (Dec. 2017), p. 74 (cit. on p. 5).
- [Jov16] I. Jovančević. “Exterior inspection of an aircraft using a Pan-Tilt-Zoom camera and a 3D scanner moved by a mobile robot: 2D image processing and 3D point cloud analysis.” Thèse. Ecole des Mines d’Albi-Carmaux, Nov. 2016 (cit. on p. 11).
- [Jun+96] D. Jung and K. Gupta. “Octree-based hierarchical distance maps for collision detection.” In: *Proceedings of IEEE International Conference on Robotics and Automation*. Minneapolis, MN, USA, Apr. 1996, pp. 454–459 (cit. on p. 98).
- [Kar13] B. Karabagli. “Vérification automatique des montages d’usinage par vision - Application à la sécurisation de l’usinage.” Thèse. Université de Toulouse le Mirail - Toulouse II, Nov. 2013 (cit. on p. 10).
- [Kat+99] H. Kato and M. Billinghurst. “Marker tracking and HMD calibration for a video-based augmented reality conferencing system.” In: *Proceedings 2nd IEEE and ACM International Workshop on Augmented Reality (IWAR’99)*. San Francisco, CA, USA, Feb. 1999, pp. 85–94 (cit. on p. 23).
- [Kaw+14] K. Kawashima, S. Kanai, and H. Date. “As-built modeling of piping system from terrestrial laser-scanned point clouds using normal-based region growing.” In: *Journal of Computational Design and Engineering* 1.1 (Jan. 2014), pp. 13–26 (cit. on p. 113).
- [Kos+95] A. Kosaka and G. Nakazawa. “Vision-Based Motion Tracking of Rigid Objects Using Prediction of Uncertainties.” In: *Proceedings of IEEE International Conference on Robotics and Automation*. Nagoya, Japan, 21-27 May 1995 (cit. on p. 26).
- [Kum+17] A. Kumar, P. Holi, S. Lee, and Y. Chai. “An adaptive approach for the reconstruction and modeling of as-built 3D pipelines from point clouds.” In: *Automation in Construction* 75 (Mar. 2017), pp. 65–78 (cit. on p. 138).
- [Lal+06] J.-F. Lalonde, N. Vandapel, and M. Hebert. *Automatic Three-Dimensional Point Cloud Processing for Forest Inventory*. Tech. rep. CMU-RI-TR-06-21. Pittsburgh, PA: Carnegie Mellon University, July 2006 (cit. on pp. 14, 114, 120, 123, 126).
- [Leb16] A. Leborgne. “Appariement de formes basé sur une squelettisation hiérarchique.” Thèse. Université de Lyon, July 2016 (cit. on p. 10).

- [Lei+17] J. Leiva, T. Villemot, G. Dangoumeau, M.-A. Bauda, and S. Larnier. “Automatic visual detection and verification of exterior aircraft elements.” In: *IEEE International Workshop of Electronics, Control, Measurement, Signals and their Application to Mechatronics (ECMSM)*. San Sebastian, Spain, 24-26 May 2017, pp. 1–5 (cit. on p. 9).
- [Lep+05] V. Lepetit and P. Fua. “Monocular Model-Based 3D Tracking of Rigid Objects: A Survey.” In: *Foundations and Trends in Computer Graphics and Vision* 1.1 (2005), pp. 1–89 (cit. on p. 27).
- [Li+11] W. Li, Z. Yin, Y. Huang, and Y. Xiong. “Automatic registration for 3D shapes using hybrid dimensionality-reduction shape descriptions.” In: *Pattern Recognition* 44 (Dec. 2011), pp. 2926–2943 (cit. on p. 97).
- [Lik+02] A. Likas, N. Vlassis, and J. Verbeek. “The Global K-Means Clustering Algorithm.” In: *Pattern Recognition* 36 (Aug. 2002), pp. 451–461 (cit. on p. 150).
- [Liu+14] C. Liu, R. Dong, and H. Wu. “Flying Robot Based Viewpoint Selection for the Electricity Transmission Equipment Inspection.” In: *Mathematical Problems in Engineering* 2014 (Dec. 2014) (cit. on p. 31).
- [Liu+16] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, Fu C.-Y., and A. Berg. “SSD: Single Shot MultiBox Detector.” In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2016. arXiv: [1512.02325](#) (cit. on p. 174).
- [Loe+18] A. Loesch, S. Bourgeois, V. Gay-Bellile, O. Gomez, and M. Dhome. “Localization of 3D objects using model-constrained SLAM.” In: *Machine Vision and Applications* 29.7 (Oct. 2018), pp. 1041–1068 (cit. on p. 10).
- [Loe17] A. Loesch. “Localisation d’objets 3D industriels à l’aide d’un algorithme de SLAM contraint au modèle.” Thèse. Université Clermont Auvergne, Dec. 2017 (cit. on pp. 25, 26, 53).
- [Low04] D. Lowe. “Distinctive Image Features from Scale-Invariant Keypoints.” In: *International Journal of Computer Vision* 60 (Nov. 2004), pp. 91–110 (cit. on p. 27).
- [Luq+14] L. Luque, B. Castillo-Toledo, A. Loukianov, and L. Gonzalez-Jimenez. “Power line inspection via an unmanned aerial system based on the quadrotor helicopter.” In: *Proceedings of the Mediterranean Electrotechnical Conference - MELECON*. Apr. 2014, pp. 393–397 (cit. on p. 32).
- [Luu17] K. Luu. “Application de l’Analyse en Composantes Principales pour étudier l’adaptation biologique en génomique des populations.” Thèse. Université Grenoble Alpes, Dec. 2017 (cit. on pp. 35, 36).
- [Mar+09] E. Marchand and M. Pressigout. “Un panorama des techniques de suivi visuel temps-réel pour la réalité augmentée.” In: *Technique et Science Informatiques* 28 (Sept. 2009), pp. 921–943 (cit. on p. 27).
- [Mer04] D. Merad. “Reconnaissance 2D/2D et 2D/3D d’objets à partir de leurs squelettes.” Thèse. Université d’Evry-Val d’Essonne, Dec. 2004 (cit. on p. 10).

-
- [Mer17] M. Merhy. “Reconnaissance de formes basée géodésiques et déformations locales de formes.” Thèse. Université de Bretagne occidentale - Brest, June 2017 (cit. on p. 66).
- [Mik19] I. Mikhailov. “Classification using a 3D sensor in a structured industrial environment.” Master Thesis. Erasmus Mundus VIBOT Master, 2019 (cit. on p. 15).
- [Mir+14] M. Mirza and S. Osindero. “Conditional Generative Adversarial Nets.” In: *Neural Information Processing Systems (NIPS)* (2014) (cit. on p. 11).
- [Mir+19a] J. Miranda, S. Larnier, A. Herbulot, and M. Devy. “UAV-based Inspection of Airplane Exterior Screws with Computer Vision.” In: *14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*. Prague, Czech Republic, Feb. 2019 (cit. on p. 11).
- [Mir+19b] J. Miranda, J. Veith, S. Larnier, A. Herbulot, and M. Devy. “Machine learning approaches for defect classification on aircraft fuselage images aquired by an UAV.” In: *14th International Conference on Quality Control by Artificial Vision (QCAV)*. Vol. 11172. Mulhouse, France, 2019, pp. 49–56 (cit. on pp. 9, 174).
- [Mok18] N. I. Mokhtari. “Application of state of the art Machine Learning to various Industrial Visual Inspection problems.” Master Thesis. Erasmus Mundus VIBOT Master, 2018 (cit. on p. 15).
- [Mor+14] V. Morell-Gimenez, M. Saval-Calvo, J. Azorin-Lopez, J. Garcia-Rodriguez, M. Cazorla, S. Orts-Escolano, and A. Fuster-Guillo. “[A Comparative Study of Registration Methods for RGB-D Video of Static Scenes](#).” In: *Sensors* 14.5 (May 2014), pp. 8547–8576 (cit. on p. 12).
- [Mou+06] E. Mouragnon, F. Dekeyser, P. Sayd, M. Lhuillier, and M. Dhome. “Real Time Localization and 3D Reconstruction.” In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. New York, NY, USA, 17-22 June 2006, pp. 363–370 (cit. on p. 25).
- [NFP14] NFPA. *NFPA 70, National Electrical Code, ARTICLE 300: General Requirements for Wiring Methods and Materials*. Tech. rep. National Fire Protection Association (NFPA), 2014 (cit. on p. 145).
- [Ng+01] A. Ng, M. Jordan, and Y. Weiss. “On Spectral Clustering: Analysis and an algorithm.” In: *Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic*. Vancouver, British Columbia, Canada, Mar. 2001, pp. 849–856 (cit. on p. 150).
- [Ngu+18] P. Nguyen and Y. Choi. “[Comparison of point cloud data and 3D CAD data for on-site dimensional inspection of industrial plant piping systems](#).” In: *Automation in Construction* 91 (July 2018), pp. 44–52 (cit. on p. 13).
- [Nor10] J.-M. Normand. “Placement de caméra en environnements virtuels.” Thèse. École Centrale de Nantes, 20 May 2010 (cit. on p. 32).

- [Nur+14] A. Nurunnabi, D. Belton, and W. Geoff. “[Robust Statistical Approaches for Local Planar Surface Fitting in 3D Laser Scanning Data.](#)” In: *ISPRS Journal of Photogrammetry and Remote Sensing* 96 (Oct. 2014), pp. 106–122 (cit. on p. 128).
- [Nur+17] A. Nurunnabi, Y. Sadahiro, and R. Lindenbergh. “Robust Cylinder Fitting In Three-Dimensional Point Cloud Data.” In: *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*. Vol. XLII-1/W1. May 2017, pp. 63–70 (cit. on pp. 14, 128).
- [Ond+01] J.-C. Ondo, T. Ouarda, V. Fortin, and B. Bobée. “[Procédures bayésiennes pour la détection d’observations singulières : synthèse bibliographique.](#)” In: *Journal de la société française de statistique* 142.2 (2001), pp. 41–64 (cit. on p. 128).
- [Pel+02] D. Pelleg and A. Moore. “X-means: Extending K-means with Efficient Estimation of the Number of Clusters.” In: *Proceedings of the 17th International Conference on Machine Learning*. 29 June - 02 July 2002 (cit. on p. 150).
- [Pil12] S. Pillai. *Shape Context Matching For Efficient OCR*. [click to get pdf](#). 2012. (Visited on 02/14/2018) (cit. on p. 66).
- [Por+16] R. Portalez and F. Duguet. *Ray tracing in an octree*. Research Report. Altimesh, Mar. 2016 (cit. on p. 98).
- [Pri+02] F. Prieto, T. Redarce, R. Lepage, and P. Boulanger. “[An Automated Inspection System.](#)” In: *The International Journal of Advanced Manufacturing Technology* 19.12 (June 2002), pp. 917–925 (cit. on p. 12).
- [Pri+99] F. Prieto, N. Grégoire, T. Redarce, R. Lepage, A. Jutard, and S. Doré. “Contrôle Automatique sans contact de pièces manufacturées.” In: *Troisième Congrès International de Génie Industriel*. Montréal, Canada, 1999, pp. 25–35 (cit. on p. 9).
- [Pyu+19] P. Pyun, A. Liu, C. Cheung, B. Mullins, and R. Sohigian. “Automatic Defect Inspection Using the NVIDIA End-to-End Deep Learning Platform.” In: *NVIDIA Developer Blog 2019*. 16 October 2019 (cit. on p. 15).
- [Qiu+14] R. Qiu, Q.-Y. Zhou, and Ulrich Neumann. “Pipe-Run Extraction and Reconstruction from Point Clouds.” In: *Computer Vision – ECCV 2014*. Zurich, Switzerland, June 2014, pp. 17–30 (cit. on p. 113).
- [Rab+05] T. Rabbani and F. Heuvel. “Efficient Hough transform for automatic detection of cylinders in point clouds.” In: *Proc ISPRS Workshop Laser Scanning 2005*. Vol. 36. Enschede, the Netherlands, Dec. 2005 (cit. on p. 14).
- [Rau+13] P. Raunonen, M. Kaasalainen, M. Akerblom, S. Kaasalainen, H. Kaartinen, M. Vastaranta, M. Holopainen, M. Disney, and P. Lewis. “[Fast Automatic Precision Tree Models from Terrestrial Laser Scanner Data.](#)” In: *Remote Sensing* 5 (Feb. 2013), pp. 491–520 (cit. on pp. 14, 15, 114).
- [Red+16] J. Redmon, S. Divvala, R. Girshick, and F. Farhadi. “You Only Look Once: Unified, Real-Time Object Detection.” In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 779–788 (cit. on p. 174).

-
- [Ren+17] S. Ren, K. He, R. Girshick, and J. Sun. “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks.” In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39.6 (June 2017), pp. 1137–1149 (cit. on p. 174).
- [Rou+11] P. Rousseeuw and M. Hubert. “Robust statistics for outlier detection.” In: *WIREs: Data Mining and Knowledge Discovery* 1.1 (Jan. 2011), pp. 73–79 (cit. on p. 128).
- [Rou84] P. Rousseeuw. “Least Median of Squares Regression.” In: *Journal of The American Statistical Association* 79.388 (Dec. 1984), pp. 871–880 (cit. on p. 14).
- [Ruf+97] A. Ruf, M. Tonko, R. Horaud, and H.-H. Nagel. “Visual Tracking of An End-Effector by Adaptive Kinematic Prediction.” In: *Proceedings of the 1997 IEEE/RSJ International Conference on Intelligent Robot and Systems (IROS). Innovative Robotics for Real-World Applications*. Grenoble, France, Sept. 1997, pp. 893–899 (cit. on p. 26).
- [Rus09] R. Rusu. “Semantic 3D Object Maps for Everyday Manipulation in Human Living Environments.” Thesis. Institut für Informatik der Technischen Universität München, 2009 (cit. on p. 151).
- [Rus10] R. Rusu. “Semantic 3D Object Maps for Everyday Manipulation in Human Living Environments.” In: *KI - Künstliche Intelligenz* 24.4 (Nov. 2010), pp. 345–348 (cit. on p. 151).
- [San+14] T. Santos, L. Koenigkan, J. Barbedo, and G. Rodrigues. “3D Plant Modeling: Localization, Mapping and Segmentation for Plant Phenotyping Using a Single Hand-held Camera.” In: *European Conference on Computer Vision (ECCV)*. Zurich, Switzerland, Sept. 2014, pp. 247–263 (cit. on p. 150).
- [Sch+07] R. Schnabel, R. Wahl, and R. Klein. “Efficient RANSAC for point-cloud shape detection.” In: *Comput. Graph. Forum* 26 (June 2007), pp. 214–226 (cit. on pp. 14, 120, 123, 125, 126, 128).
- [Sha48] C. E. Shannon. “A Mathematical Theory of Communication.” In: *Bell System Technical Journal* 27.3 (July 1948), pp. 379–423 (cit. on p. 32).
- [Sle+06] G. Slenski, P. McDowell, and M. Walz. “Novel Technologies For Improving Wire System Integrity.” In: *9th Joint FAA/DoD/NASA Aging Aircraft Conference*. June 2006, pp. 1–19 (cit. on p. 142).
- [Smi+87] R. Smith and P. Cheeseman. “On the Representation and Estimation of Spatial Uncertainty.” In: *The International Journal of Robotics Research* 5 (Feb. 1987) (cit. on p. 25).
- [Sto+04] D. Stobel, M. Hanheide, G. Sagerer, L. Kruger, and M. Ellenrieder. “Feature and Viewpoint Selection for Industrial Car Assembly.” In: *Lecture Notes in Computer Science*. Vol. 3175. Aug. 2004, pp. 528–535 (cit. on p. 32).
- [Su+19] Z. Su, S. Li, H. Liu, and Y. Liu. “Extracting Wood Point Cloud of Individual Trees Based on Geometric Features.” In: *IEEE Geoscience and Remote Sensing Letters* (Feb. 2019), pp. 1–5 (cit. on p. 14).

-
- [Tab+19] D. Tabernik, S. Šela, J. Skvarč, and D. Skočaj. “[Segmentation-based deep-learning approach for surface-defect detection](#).” In: *Journal of Intelligent Manufacturing* (May 2019) (cit. on p. 15).
- [Tag+18] K. Taguchi and M. Morimoto. “Appearance inspection of wire harness using RGB-D Sensor.” In: *Joint 7th International Conference on Informatics, Electronics and Vision (ICIEV) and 2nd International Conference on Imaging, Vision and Pattern Recognition (icIVPR)*. June 2018, pp. 416–419 (cit. on p. 13).
- [Tau91] G. Taubin. “[Estimation of planar curves, surfaces, and nonplanar space curves defined by implicit equations with applications to edge and range image segmentation](#).” In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 13.11 (Nov. 1991), pp. 1115–1138 (cit. on p. 114).
- [Tha08] Y. Tharrault. “Diagnostic de fonctionnement par analyse en composantes principales : application à une station de traitement des eaux usées.” Thèse. Université de Lorraine, Dec. 2008 (cit. on p. 128).
- [Tie+09] N. Tien Sy, M. Avila, S. Begot, F. Duculty, and J.-C. Bardet. “Étude d’un algorithme de détection de défauts sur des images de chaussées.” In: *colloque GRETSI*. Dijon, France, Sept. 2009 (cit. on p. 9).
- [Tor+00] P. Torr and A. Zisserman. “[MLESAC: A New Robust Estimator with Application to Estimating Image Geometry](#).” In: *Computer Vision and Image Understanding* 78 (Aug. 2000), pp. 138–156 (cit. on pp. 120, 123, 126).
- [Tou+19] K. Tout, M. Bouabdellah, C. Cudel, and J.-F. Urban. “Automated vision system for crankshaft inspection using deep learning approaches.” In: *14th International Conference on Quality Control by Artificial Vision (QCAV)*. Vol. 11172. 2019, pp. 162–167 (cit. on p. 9).
- [Tre+10] S. Treuillet and E. Royer. “[Outdoor/Indoor Vision Based Localization for Blind Pedestrian Navigation Assistance](#).” In: *Int. J. Image Graphics* 10 (Oct. 2010), pp. 481–496 (cit. on p. 25).
- [Vas+09] J. Vasquez-Gomez, E. Lopez-Damian, and L. Sucar. “View Planning for 3D Object Reconstruction.” In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Nov. 2009, pp. 4015–4020 (cit. on p. 32).
- [Vaz+01] P.-P. Vazquez Alcocer, M. Feixas, M. Sbert, and W. Heidrich. “Viewpoint Selection using Viewpoint Entropy.” In: *Proceedings of Vision, Modeling and Visualization Conference (VMV)*. Vol. 1010. Nov. 2001, pp. 273–280 (cit. on p. 32).
- [Vaz03] P.-P. Vazquez. “On the selection of good views and its application to computer graphics.” Ph.D. thesis. Departament de Llenguatges i Sistemes Informatics (LSI), Universitat Politècnica de Catalunya, Barcelona, Spain, 2003 (cit. on p. 32).
- [Via16] I. Viana do Espirito Santo. “Inspection automatisée d’assemblages mécaniques aéronautiques par vision artificielle - Une approche exploitant le modèle CAO.” Thèse. Ecole des Mines d’Albi-Carmaux, Dec. 2016 (cit. on p. 11).

-
- [Vil+16] T. Villemot, S. Larnier, and S. Vetault. “Détection d’amers visuels pour la navigation d’un robot autonome autour d’un avion et son inspection.” In: *Reconnaissance des Formes et Intelligence Artificielle (RFIA)*. Clermont-Ferrand, France, June 2016 (cit. on p. 9).
- [Wu+15] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao. “3D ShapeNets: A deep representation for volumetric shapes.” In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2015, pp. 1912–1920 (cit. on p. 32).
- [Xie+17] S. Xie and Z. Tu. “[Holistically-Nested Edge Detection](#).” In: *International Journal of Computer Vision* 125.1 (Dec. 2017), pp. 3–18 (cit. on p. 78).
- [Zha+02] X. Zhang, S. Fronz, and N. Navab. “Visual marker detection and decoding in AR systems: a comparative study.” In: *Proceedings of International Symposium on Mixed and Augmented Reality*. Darmstadt, Germany, Feb. 2002, pp. 97–106 (cit. on p. 23).

Author : Hamdi Ben Abdallah

Title : Inspection of aeronautical mechanical parts by 2D / 3D vision using the CAD model and the pose estimated in real time.

Abstract

This thesis makes part of a research aimed towards innovative digital tools for the service of what is commonly referred to as Factory of the Future. Our work was conducted in the scope of the joint research laboratory "Inspection 4.0" founded by IMT Mines Albi/ICA and the company DIOTA specialized in the development of numerical tools for Industry 4.0.

In the thesis, we were interested in the development of systems exploiting 2D images or (and) 3D point clouds for the automatic inspection of complex aeronautical mechanical assemblies (typically an aircraft engine). The CAD (Computer Aided Design) model of the assembly is at our disposal and our task is to verify that the assembly has been correctly assembled, i.e. that all the elements constituting the assembly are present in the right position and at the right place. The CAD model serves as a reference.

We have developed two inspection scenarios that exploit the inspection systems designed and implemented by DIOTA : (1) a scenario based on a tablet equipped with a camera, carried by a human operator for real-time interactive control, (2) a scenario based on a robot equipped with sensors (two cameras and a 3D scanner) for fully automatic control. In both scenarios, a so-called localisation camera provides in real-time the pose between the CAD model and the sensors (which allows to directly link the 3D digital model with the 2D images or the 3D point clouds analysed).

We first developed 2D inspection methods, based solely on the analysis of 2D images. Then, for certain types of inspection that could not be performed by using 2D images only (typically requiring the measurement of 3D distances), we developed 3D inspection methods based on the analysis of 3D point clouds. For the 3D inspection of electrical cables, we proposed an original method for segmenting a cable within a point cloud. We have also tackled the problem of automatic selection of best view point, which allows the inspection sensor to be placed in an optimal observation position.

The developed methods have been validated on many industrial cases. Some of the inspection algorithms developed during this thesis have been integrated into the DIOTA Inspect© software and are used daily by DIOTA's customers to perform inspections on industrial sites.

Keywords : Complex aeronautical assemblies, Robotic inspection system, CAD model, 2D Image analysis, 3D Point cloud analysis, 3D Segmentation, Automatic viewpoint selection.

Auteur : Hamdi Ben Abdallah

Titre : Inspection d'assemblages aéronautiques par vision 2D/3D, en exploitant la maquette numérique et la pose estimée en temps réel.

Résumé

Cette thèse s'inscrit dans le contexte du développement d'outils numériques innovants au service de ce qui est communément désigné par Usine du Futur. Nos travaux de recherche ont été menés dans le cadre du laboratoire de recherche commun "Inspection 4.0" entre IMT Mines Albi/ICA et la Sté DIOTA spécialisée dans le développement d'outils numériques pour l'Industrie 4.0.

Dans cette thèse, nous nous sommes intéressés au développement de systèmes exploitant des images 2D ou des nuages de points 3D pour l'inspection automatique d'assemblages mécaniques aéronautiques complexes (typiquement un moteur d'avion). Nous disposons du modèle CAO de l'assemblage (aussi désigné par maquette numérique) et il s'agit de vérifier que l'assemblage a été correctement assemblé, i.e que tous les éléments constituant l'assemblage sont présents, dans la bonne position et à la bonne place. La maquette numérique sert de référence.

Nous avons développé deux scénarios d'inspection qui exploitent les moyens d'inspection développés par DIOTA : (1) un scénario basé sur une tablette équipée d'une caméra, portée par un opérateur pour un contrôle interactif temps-réel, (2) un scénario basé sur un robot équipé de capteurs (deux caméras et un scanner 3D) pour un contrôle totalement automatique. Dans les deux scénarios, une caméra dite de localisation fournit en temps-réel la pose entre le modèle CAO et les capteurs mis en œuvre (ce qui permet de relier directement la maquette numérique 3D avec les images 2D ou les nuages de points 3D analysés).

Nous avons d'abord développé des méthodes d'inspection 2D, basées uniquement sur l'analyse d'images 2D puis, pour certains types d'inspection qui ne pouvaient pas être réalisés à partir d'images 2D (typiquement nécessitant la mesure de distances 3D), nous avons développé des méthodes d'inspection 3D basées sur l'analyse de nuages de points 3D. Pour l'inspection 3D de câbles électriques présents au sein de l'assemblage, nous avons proposé une méthode originale de segmentation 3D des câbles. Nous avons aussi traité la problématique de choix automatique de point de vue qui permet de positionner le capteur d'inspection dans une position d'observation optimale.

Les méthodes développées ont été validées sur de nombreux cas industriels. Certains des algorithmes d'inspection développés durant cette thèse ont été intégrés dans le logiciel DIOTA Inspect[®] et sont utilisés quotidiennement chez les clients de DIOTA pour réaliser des inspections sur site industriel.

Mots-clés : Assemblages aéronautiques complexes, Inspection robotisée, Maquette numérique, Analyse d'images 2D, Analyse de nuages de points 3D, Segmentation 3D, Sélection automatique de point de vue.