



HAL
open science

Inspection automatisée d'assemblages mécaniques aéronautiques par vision artificielle : une approche exploitant le modèle CAO

Ilísio Viana Do Espírigo Santo

► **To cite this version:**

Ilísio Viana Do Espírigo Santo. Inspection automatisée d'assemblages mécaniques aéronautiques par vision artificielle : une approche exploitant le modèle CAO. Optique [physics.optics]. Ecole des Mines d'Albi-Carmaux, 2016. Français. NNT : 2016EMAC0022 . tel-01590619

HAL Id: tel-01590619

<https://theses.hal.science/tel-01590619>

Submitted on 19 Sep 2017

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

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



THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par : *École Nationale Supérieure des Mines d'Albi-Carmaux*

Présentée et soutenue le *12 décembre 2016* par :
ILÍRIO VIANA DO ESPIRITO SANTO

**Inspection automatisée d'assemblages mécaniques aéronautiques par
vision artificielle : une approche exploitant le modèle CAO**

JURY

FAKHR-EDDINE ABABSA	Maître de Conférences-HDR, Université d'Evry	Rapporteur
YVES LUCAS	Maître de Conférences-HDR, Université d'Orléans	Rapporteur
FLORIAN BUGARIN	Maître de Conférences, Université de Toulouse III	Co-Directeur de Thèse
MICHEL DEVY	Directeur de Recherche CNRS, LAAS-CNRS	Président du Jury
ALBERTO IZAGUIRRE	Professeur, Université de Mondragon (Espagne)	Examineur
JEAN-JOSÉ ORTEU	Professeur de l'Institut Mines-Télécom, Mines-Albi	Directeur de Thèse

École doctorale et spécialité :

*École Doctorale Systèmes (EDSYS), dans la spécialité :
Informatique et Robotique*

Thèse réalisée à :

Institut Clément Ader, UMR CNRS 5312, Université de Toulouse - Mines Albi

Table des matières

1	Introduction générale	6
1.1	Structuration du manuscrit	7
1.2	Problématique : Comment exploiter la CAO	8
1.2.1	Le projet CAAMVis	8
1.2.1.1	Motivation et objectifs	8
1.2.1.2	Moyens matériels	9
1.2.1.3	Comment le modèle CAO est-il exploité dans ce projet	10
1.2.2	Le projet Lynx©	10
1.2.2.1	Motivation et objectifs	10
1.2.2.2	Moyens matériels	11
1.2.2.3	Comment le modèle CAO est-il exploité dans ce projet	13
1.3	Contributions	13
1.4	Conclusion	14
2	Inspection visuelle automatisée et reconnaissance d’objets	16
2.1	Considérations générales	17
2.1.1	Définitions	17
2.1.1.1	Reconnaissance d’objets	17
2.1.1.2	Inspection visuelle automatisée	17
2.1.2	Quelques systèmes d’acquisition utilisés en inspection visuelle	17
2.1.2.1	Caméra matricielle	18
2.1.2.2	Systèmes à base de caméra linéaire	18
2.1.2.3	La flexibilité du système Pan-Tilt-Zoom	19
2.1.2.4	Au-delà de l’œil humain	19
2.1.2.5	Nos systèmes d’acquisition	20
2.2	Méthodes de reconnaissance d’objets	20
2.2.1	La photométrie	20
2.2.1.1	Reconnaissance d’objets par Template Matching	20
2.2.1.2	La comparaison basée couleur	23
2.2.2	Les descripteurs locaux et méthodes à base d’histogrammes	23
2.2.2.1	SIFT	24
2.2.2.2	SURF	24
2.2.2.3	KAZE et AKAZE	24
2.2.2.4	BRISK	25

2.2.2.5	ORB	25
2.2.2.6	FREAK	26
2.2.2.7	Résumé sur les méthodes de détection et description de points d'intérêt	26
2.2.3	Reconnaissance par apprentissage	27
2.2.3.1	Apprentissage supervisé	27
2.2.3.1.1	K-NN	28
2.2.3.1.2	Classifieur bayésien	28
2.2.3.1.3	Arbres de décision	29
2.2.3.1.4	L'algorithme du Perceptron	29
2.2.3.1.5	Perceptron multicouche ou réseau de neurones	30
2.2.3.2	Apprentissage non-supervisé	32
2.2.3.2.1	L'algorithme du K-means	32
2.2.3.3	Apprentissage profond	32
2.2.3.4	Réduction de dimension en apprentissage	32
2.2.3.5	Lien entre l'apprentissage et l'inspection	33
2.2.4	La géométrie 2D	34
2.2.4.1	Primitives et attributs géométriques	34
2.2.4.2	Contour et squelette	34
2.3	Inspection sans modèle CAO	35
2.3.1	Inspection dans l'industrie agroalimentaire	36
2.3.2	Inspection dans le secteur de la banque et de la finance	36
2.3.3	Inspection dans l'industrie pharmaceutique	37
2.3.4	Inspection dans l'industrie électronique	37
2.3.5	Inspection dans l'industrie automobile	38
2.3.6	Inspection dans l'industrie aéronautique	39
2.3.7	Autres	40
2.4	Inspection de pièces mécaniques à l'aide d'un modèle CAO	40
2.4.1	Méthodes basées sur une comparaison 3D/3D	41
2.4.2	Méthodes basées sur une comparaison 2D/2D issue du modèle 3D	42
2.5	Conclusion	43
3	Application ayant servi de support à nos travaux	44
3.1	Projet CAAMVis : inspection à partir d'un capteur stéréo porté par un robot	44
3.1.1	Rappel des objectifs et stratégie d'inspection	44
3.1.1.1	Rappel des objectifs	44
3.1.1.2	Stratégie d'inspection	45

3.1.2	Rappel de la méthode développée par la société Noomeo, comparaison 3D /3D	45
3.1.2.1	Constat : Le problème du temps de calcul	46
3.1.3	Inspection basée sur la comparaison 2D/2D via la transformée en distance	46
3.1.3.1	Obtenir des contours 2D à partir d'un modèle 3D	48
3.1.3.2	Construction de la fonction de mesure basée sur l'utilisation d'une carte de distances	49
3.1.3.3	Algorithme d'optimisation utilisé	50
3.1.3.4	Description de la fonction de décision	52
3.1.4	Résultats préliminaires et discussions	52
3.1.5	Bilan du projet CAAMVis : améliorations envisagées	54
3.2	Introduction au projet Lynx et à l'inspection à partir d'une caméra PTZ	56
3.2.1	Rappel des objectifs	56
3.2.2	Les deux stratégies d'utilisation du modèle CAO dans le projet Lynx©	57

4 Appariement de primitives géométriques par la théorie des graphes et mesure de ressemblance 58

4.1	Concepts de base en théorie des graphes	59
4.1.1	Graphe et notion de degré	59
4.1.1.1	Exemple	59
4.1.2	Incidence et adjacence, prédécesseurs et successeurs	60
4.1.2.1	Exemple	60
4.1.3	Isomorphisme de graphe	62
4.1.3.1	Exemple	62
4.1.4	Graphes et sous-graphes	63
4.1.4.1	Exemple de graphe et sous-graphes	63
4.1.5	Bipartisme de graphe	64
4.1.5.1	Exemple	64
4.1.6	Graphe attribué relationnel (ARG en anglais pour Attributed Relational Graph)	65
4.1.7	Graphes et le problème de voisinage	65
4.1.8	Graphe de Voisinage Relatif RNG en anglais pour Relative Neighbourhood Graph	65
4.1.9	Le graphe de Gabriel	66
4.1.10	Autres graphes de voisinage	66
4.1.11	Mise en correspondance de deux graphes	66
4.2	Schéma fonctionnel de la méthode	67

4.3	Développement de la méthode d'appariement de primitives en exploitant des images de synthèse	69
4.3.1	Segmentation des sept images de la figure 4.7	70
4.3.2	Recalage d'images	73
4.3.2.1	Notre choix	74
4.3.3	Extraction de primitives géométriques	75
4.3.3.1	Méthodes à base de système de votes et d'ajustement de courbes aux données	75
4.3.3.1.1	Méthodes de type Transformée de Hough	75
4.3.3.1.2	Méthodes d'ajustement de type Prasad	75
4.3.3.2	Méthodes basées sur le regroupement de contours	75
4.3.3.3	Autres méthodes	76
4.3.3.4	Notre choix	76
4.3.3.4.1	Extraction d'ellipses par la méthode de Prasad et leur caractérisation	77
4.3.3.4.2	Extraction de segments de droite par la transformée de Hough	78
4.3.3.4.3	Résultat de l'extraction des primitives	79
4.3.4	Appariement des primitives par combinaison de graphes - fonction de ressemblance	81
4.3.4.1	Construction des deux graphes en vue de l'appariement	81
4.3.4.2	Graphes attribués et mesure de similarité versus graphe topologique	84
4.3.4.3	Fonction de ressemblance	84
4.3.5	Recherche du meilleur couple d'appariement à l'aide d'un graphe biparti	87
4.4	Prise de décision	90
4.5	Conclusion	91
5	Résultats et discussion	92
5.1	Résultat d'appariement entre deux images synthétiques	92
5.1.1	Cas où l'image test a moins d'éléments que l'image de référence	93
5.1.1.1	Cas n ° 1	93
5.1.1.2	Cas n ° 2	94
5.1.1.3	Cas n ° 3	97
5.1.2	Cas où l'image test a plus d'éléments que l'image de référence	100
5.1.2.1	Cas n ° 1	100
5.1.3	Combinaison de translation et éléments manquants dans l'image test	103

5.1.3.1	Cas n ° 1	103
5.1.4	Combinaison de zoom + nombre d'éléments différents	106
5.1.4.1	Cas n ° 1	106
5.1.4.2	Cas n ° 2	109
5.1.4.3	Cas n ° 3	112
5.1.5	Remarque concernant les objets tests ayant subi une rotation	115
5.2	Résultats d'appariement entre une image synthétique et une image réelle	115
5.2.1	Tests réalisés sur les données du projet AirCOBOT	115
5.2.1.1	Contexte	116
5.2.1.2	Cas n ° 1	118
5.2.1.3	Cas n ° 2	119
5.2.1.4	Cas n ° 3	120
5.2.1.5	Cas n ° 4	121
5.3	Conclusion	122
6	Conclusion générale	123
A	Assemblages mécaniques utilisés pour nos tests	126
A.1	Assemblage et modèle CAO utilisés dans le projet CAAMVis	126
A.1.1	Zoom sur quelques éléments du modèle CAO et de l'assemblage réel à contrôler	127
A.1.2	Exemple d'élément réel à contrôler reconstruit en 3D	127
A.2	Assemblage et modèle CAO utilisés dans le projet Lynx©	129
A.2.1	Assemblage et modèle CAO fournis par G2Metric	129
A.2.2	Zoom sur quelques éléments du modèle CAO, utilisés lors de tests avec données de synthèse	130
A.2.3	Assemblage et modèle CAO fournis par AéroliA	130
A.2.3.1	Éléments de l'assemblage AéroliA	132
A.3	Données utilisées pour tester notre méthode lors de la collaboration avec l'équipe projet AirCOBOT	132
B	Résultats de la segmentation des images de la figure 4.7	133
B.1	Résultat de segmentation de l'objet 4.7a	133
B.2	Résultat de segmentation de l'objet 4.7b	134
B.3	Résultat de segmentation de l'objet 4.7c	135
B.4	Résultat de segmentation de l'objet 4.7d	136
B.5	Résultat de segmentation de l'objet 4.7e	137
B.6	Résultat de segmentation de l'objet 4.7f	138

B.7	Résultat de segmentation de l'objet 4.7g	139
C	Résultats d'extraction de primitives sur les images segmentées de l'annexe B	140
C.1	Extraction de primitives de l'image C.1a	141
C.2	Extraction de primitives de l'image C.1b	141
C.3	Extraction de primitives de l'image C.1c	142
C.4	Extraction de primitives de l'image C.1d	142
C.5	Extraction de primitives de l'image C.1e	142
C.6	Extraction de primitives de l'image C.1f	143
C.7	Extraction de primitives de l'image C.1g	143
D	Expérimentations réalisées pour obtenir les données réelles	144
D.1	Positionner d'abord la caméra virtuelle dans la maquette numérique puis positionner la caméra réelle dans l'environnement réel.	144
D.2	Positionner d'abord la caméra réelle calibrée dans l'environnement réel et estimer sa pose	145
D.3	Expérimentation avec la caméra PTZ et assemblage mis à disposition par G2Metric	148
D.4	Expérimentation avec une caméra CCD standard et l'assemblage mis à disposition par AéroliA	149

Table des figures

1.1	CAAMVis : modèle CAO et assemblage réel.	9
1.2	Robot et capteur utilisés dans le projet CAAMVis	10
1.3	Lynx© : schéma de principe	11
1.4	Exemple de contrôle réalisé avec Lynx	12
1.5	Assemblage utilisé dans le projet Lynx : copie d'écran du modèle CAO dans la maquette Catia V5 et image de l'assemblage réel correspondant	12
1.6	Système d'acquisition utilisé dans Lynx©	13
2.1	Exemple d'un neurone biologique à gauche (extrait de [Hackeling, 2014]), neurone artificiel à droite (extrait de [Murty and Devi, 2011])	30
2.2	Exemple d'un réseau de neurones (extrait de [Murty and Devi, 2011])	31
2.3	Système ESOBASIS d'Enermeter, partie matérielle pendant le fonctionne- ment (gauche) et à l'arrêt (droite)	38
2.4	Système ESOTIREINSP d'Enermeter, partie matérielle	39
3.1	A gauche : modèle CAO d'un élément à contrôler, à droite : image de luminance de l'élément réel	46
3.2	Schéma de la méthode proposée. a) modèle CAO de l'élément, b) image de profondeur issue du modèle CAO, c) contour 2D issu de l'image de profondeur, d) image en niveau de gris de l'élément réel, e) contour extrait de l'image réelle.	47
3.3	Image de profondeur obtenue par rasterisation de la CAO.	48
3.4	Contours extraits de l'image synthétique et de l'image réelle	49
3.5	Image de transformée en distance des contours d'un élément réel.	49
3.6	Exemple de résultat obtenu par notre méthode d'appariement 2D/3D basée sur recalage de contour grâce à la transformée en distance	51
3.7	Test de la méthode sur l'objet réel 3.7b	53
3.8	Test de la méthode sur l'objet réel 3.8b	53
3.9	Test de la méthode sur l'objet réel 3.9b	53
3.10	Test de la méthode sur l'objet réel 3.10b	53
3.11	Test de la méthode sur l'objet réel 3.11b	54
3.12	Exemple de projection d'un support contenant trois classes d'objets. L'objet à contrôler en bleu, le support en rouge, un objet de même type que l'objet à contrôler en vert.	55

4.1	Exemple d'un graphe orienté	59
4.2	Exemple d'un graphe non orienté avec les arêtes numérotées pour illustrer le concept d'incidence	60
4.3	Illustration d'isomorphisme	62
4.4	Illustration de graphes et sous-graphes	64
4.5	Exemple d'un graphe biparti, formé de l'ensemble de nombre impairs et pairs de 1 à 6	64
4.6	Schéma fonctionnel de notre méthode d'appariement de primitives	67
4.7	Images générées à partir du modèle CAO de l'assemblage, en format ".bmp"	69
4.8	Première ligne : copie d'écran du modèle CAO. Deuxième ligne, à gauche : image générée à partir du modèle CAO, au format vectoriel et son contour, extrait après le passage du format vectoriel au format matriciel (".tiff"); à droite : images du modèle CAO modifiées, au format vectoriel (SVG) et leur contours respectifs après la conversion au format matriciel (".tiff").	72
4.9	Deux contours avant recalage : contour de l'image de référence en vert et contour de l'image test en magenta	73
4.10	Deux contours recalés : contour de l'image de référence en vert et contour de l'image test en magenta. Les pixels noirs concernent les endroits où les pixels verts et magenta se superposent.	74
4.11	Extraction d'ellipses par la méthode de Prasad : les ellipses considérées correctes sont dessinées en vert et les candidates non retenues sont dessinées en bleu. Le contour est affiché en noir sur fond blanc pour plus de clarté	77
4.12	Extraction de segments de droite par la transformée de Hough standard. Les quatre segments de droite correspondant aux quatre pics les plus importants de l'accumulateur de Hough sont dessinés en vert. Les débuts des segments de droite sont dessinés en jaune et leur fin en rouge.	78
4.13	Extraction d'ellipses et de segments de droite : affichage de l'ensemble des primitives sélectionnées (en vert)	80
4.14	Ensemble de primitives (segments de droite et ellipses) extraits de l'image de référence (gauche) et de l'image test (droite), après recalage de contour.	80
4.15	Passage de primitives (ellipses) au graphe relationnel.	82
4.16	Graphe relationnel, au sens 4.1.8, représentant les ellipses de la figure 4.7e.	83

4.17	Graphes attribués pour les deux ensembles de primitives de la figure 4.14. Première ligne : graphe correspondant à l'ensemble d'ellipses extraites de l'image de référence (gauche); graphe correspondant à l'ensemble d'ellipses extraites de l'image de l'objet à évaluer (droite). Deuxième ligne : graphe correspondant à l'ensemble des segments de droite extraits de l'image de référence (gauche); graphe correspondant à l'ensemble des segments de droite de l'image test (droite).	83
4.18	Graphe biparti montrant les candidats à l'appariement final pour le cas présenté dans la figure 4.14	87
4.19	Graphe biparti après optimisation des graphes de la figure 4.18 pour respecter la contrainte d'unicité	89
5.1	Ensemble des primitives à apparier pour le cas de l'objet 4.7c versus objet 4.7c modifié	93
5.2	Graphe biparti correspondant à l'appariement des primitives du cas test 4.7c versus test 4.7c modifié	94
5.3	Ensemble des primitives à apparier pour le cas de l'objet 4.7d versus objet 4.7d modifié	94
5.4	Graphe biparti correspondant à l'appariement des primitives du cas test 4.7d versus 4.7d modifié	95
5.5	Graphe biparti après optimisation des graphes de la figure 5.4 pour respecter la contrainte d'unicité	96
5.6	Ensemble des primitives à apparier pour le cas de l'objet 4.7a versus objet 4.7a modifié	97
5.7	Graphe biparti correspondant à l'appariement des primitives du cas au cas test 4.7a versus 4.7a modifié	98
5.8	Graphe biparti après optimisation des graphes de la figure 5.7 pour respecter la contrainte d'unicité	99
5.9	Première ligne : contours des images de référence (gauche) et test(droite) avant recalage. Deuxième ligne : Les deux contours recalés. Troisième ligne : Primitives extraites des contours	100
5.10	Graphe biparti correspondant à l'appariement des primitives du cas au cas test 4.7b versus 4.7b modifié	101
5.11	Graphe biparti après optimisation des graphes de la figure 5.10 pour respecter la contrainte d'unicité	102

5.12	Illustration d'un cas test contenant des éléments manquants. Première colonne : Avant le recalage, référence (vert) et test (magenta). Deuxième colonne : après le recalage. Troisième colonne : primitives extraites des deux contours après le recalage.	103
5.13	Graphe biparti correspondant à l'appariement des primitives du cas test 4.7b versus 4.7b modifié	104
5.14	Graphe biparti après optimisation des graphes de la figure 5.13 pour respecter la contrainte d'unicité	105
5.15	Appariement de primitives en prenant comme référence 4.7e versus 4.7f	106
5.16	Graphe biparti correspondant à l'appariement des primitives du cas test 4.7e versus 4.7f	107
5.17	Graphe biparti après optimisation des graphes de la figure 5.16 pour respecter la contrainte d'unicité	108
5.18	Appariement de primitives en prenant comme référence 4.7f versus 4.7g	109
5.19	Graphe biparti correspondant à l'appariement des primitives du cas 4.7e versus 4.7f	110
5.20	Graphe biparti après optimisation des graphes de la figure 5.19 pour respecter la contrainte d'unicité	111
5.21	Appariement de primitives en prenant comme référence 4.7e versus 4.7g	112
5.22	Graphe biparti correspondant à l'appariement des primitives du cas 4.7e versus 4.7g	113
5.23	Graphe biparti après optimisation des graphes de la figure 5.22 pour respecter la contrainte d'unicité	114
5.24	Exemple de rotation (image de référence et image de l'objet à évaluer) avant et après recalage. À gauche : image de référence (green), image test (magenta), avant le recalage. À droite image de référence et image test après le recalage.	115
5.25	CAO d'un avion à inspecter et son correspondant acquis avec la caméra PTZ, projet AirCOBOT	117
5.26	Navigation : robot loin de l'avion.	118
5.27	Graphe biparti illustrant des cas de conflits (non respect de la contrainte d'unicité de l'appariement) lors de la mise en correspondance des éléments de la figure 5.26	118
5.28	Graphe biparti de la figure 5.27 élagué pour respecter l'unicité de l'appariement	119
5.29	Navigation, robot proche de l'avion.	119

5.30	Graphe biparti correspondant à la matrice de scores 5.16, avec les meilleurs appariements mis en évidence par la couleur bleue, dans la figure 5.16.	120
5.31	Robot à un point de contrôle pour inspecter.	120
5.32	Graphe biparti correspondant au tableau des scores 5.17, avec les meilleurs appariements mis en évidence par la couleur bleue.	121
5.33	Robot positionné à un point de contrôle, pour réaliser l'inspection	121
5.34	Graphe biparti correspondant à la mise en correspondance des primitives de la figure 5.33 meilleurs appariements mis en évidence par la couleur bleue.121	
A.1	CAAMVis : zoom sur modèle CAO et assemblage réel	126
A.2	CAAMVis : zoom sur quelques éléments du modèle CAO, de l'assemblage réel	127
A.3	CAAMVis : zoom sur un élément CAO reconstruit en 3D	127
A.4	Modèle CAO et assemblage réel mis à disposition par G2Metric dans le cadre du projet Lynx©	129
A.5	Images générées à partir du modèle CAO et utilisés lors de tests synthétiques	130
A.6	CAAMVis : Zoom sur modèle CAO et assemblage réel	131
A.7	CAAMVis : zoom sur quelques éléments de l'assemblage	132
A.8	Données du projet AirCOBOT utilisées pour tester notre méthode développée dans le cadre du projet Lynx©	132
B.1	Image synthétique segmentée 4.7a par application de plusieurs pré-traitements successifs	133
B.2	Image synthétique 4.7b segmentée par application de plusieurs pré-traitements successifs	134
B.3	Image synthétique 4.7c segmentée par application de plusieurs pré-traitements successifs	135
B.4	Image synthétiques segmentée 4.7d par application de plusieurs pré-traitements successifs	136
B.5	Image synthétique segmentée 4.7e par application de plusieurs pré-traitements successifs	137
B.6	Image synthétique segmentée par application de plusieurs pré-traitements successifs	138
B.7	Image synthétique 4.7g segmentée par application de plusieurs pré-traitements successifs	139
C.1	Images générées à partir du modèle CAO de l'assemblage, en format ".bmp"	140

C.2	Extraction d'ellipses et de segments de droite (en vert) de l'image segmentée	
C.2a	141
C.3	Extraction d'ellipses et de segments de droite (en vert) de l'image segmentée	
C.3a	141
C.4	Extraction d'ellipses et de segments de droite (en vert) de l'image segmentée	
C.4a	142
C.5	Extraction d'ellipses et de segments de droite (en vert) de l'image segmentée	
C.5a	142
C.6	Extraction d'ellipses et de segments de droite (en vert) de l'image segmentée	
C.6a	142
C.7	Extraction d'ellipses et de segments de droite (en vert) de l'image segmentée	
C.7a	143
C.8	Extraction d'ellipses et de segments de droite (en vert) de l'image segmentée	
C.8a	143
D.1	Projection d'un élément CAO sur l'image réelle. Illustration du problème de point de vue. Les détails (parties saillantes) de l'objet qui se trouvent dans l'orientation de la caméra sont écrasés pendant la projection	148
D.2	CAO de l'assemblage et assemblage réel	149
D.3	Conventions de repères	150
D.4	Projections d'éléments de la CAO sur l'assemblage réel	151
D.5	Projection du modèle CAO correspondant à l'objet réel "C"	152

Liste des tableaux

3.1	Vocabulaire utilisé en test	53
3.2	Résultats obtenus avec notre la méthode d'inspection décrite dans ce chapitre	54
4.1	Tableau montrant les degrés entrant et sortant du graphe orienté de la figure	
4.1	60
4.2	Tableau montrant la liste d'incidence du graphe de la figure 4.2	61
4.3	Tableau montrant la liste d'incidence du graphe de la figure 4.2	61
4.4	Tableau montrant la liste des prédécesseurs et des successeurs de du graphe de la figure 4.2	61
4.5	Matrice d'adjacence du graphe de la figure 4.2	61
4.6	Correspondance entre les graphes G et H	63
4.7	Tableau décrivant les primitives de type ellipse, cf. figure 4.11	78
4.8	Tableau décrivant les primitives de type segments de droite cf. figure 4.12 .	79
4.9	Tableau d'appariement entre les ellipses de l'image test et celles de l'image de référence, montrées dans la figure 4.14.	86
4.10	Tableau d'appariement entre les segments de droite de l'image test et ceux de l'image de référence, montrées dans la figure 4.14.	86
5.1	Tableau d'appariement de segments de droite pour le couple référence-test de la figure 5.1	93
5.2	Tableau d'appariement de segments de droite pour le couple référence-test de la figure 5.3	95
5.3	Tableau d'appariement d'ellipses pour le couple référence-test de la figure 5.3	95
5.4	Tableau d'appariement de segments de droite pour le couple référence-test de la figure 5.6	97
5.5	Tableau d'appariement d'ellipses pour le couple référence-test de la figure 5.6	97
5.6	Tableau d'appariement de segments de droite pour le couple référence-test de la figure 5.9	100
5.7	Tableau d'appariement d'ellipses pour le couple référence-test de la figure 5.9	101
5.8	Matrice d'appariement entre les segments de droite de l'image de référence et celles de l'image test (cf. figure 5.12).	103
5.9	Matrice d'appariement entre les ellipses de l'image de référence et leurs correspondants de l'image de l'objet à inspecter, cf. Fig. 5.12.	104
5.10	Tableau d'appariement pour le couple référence-test de la figure 5.15	106
5.11	Tableau d'appariement pour le couple référence-test de la figure 5.15	107

5.12	Tableau d'appariement de segments de droite pour le couple référence-test de la figure 5.18	109
5.13	Tableau d'appariement d'ellipses pour le couple référence-test de la figure 5.18	110
5.14	Tableau d'appariement de segments de droite pour le couple référence-test de la figure 5.21	112
5.15	Tableau d'appariement d'ellipses pour le couple référence-test de la figure 5.21	112
5.16	Matrice de scores pour le jeu de primitives de la figure 5.29	119
5.17	Matrice de scores pour le jeu de primitives de la figure 5.31	120

Remerciements

Nous y voilà! Parti d'ô combien loin village de Messias Alves à Santana (São Tomé et Príncipe) et être en train d'écrire les dernières lignes de mon manuscrit de thèse est une fierté, mais surtout la fin d'un parcours jalonné de nombreuses rencontres, d'obstacles et de dur labeur. Loin est le temps où enfant je me laissais impressionner par le mystère des ondes hertziennes qui sans cesse me laissaient ébahi au travers des postes radios à pille. Et que je m'étais promis de comprendre. Bien après les études en Génie Électrique et Informatique Industrielle et d'Électronique, Électrotechnique, Automatique à Toulouse (ville que j'affectionnais avant même de connaître) m'ont fait découvrir le traitement du signal puis le traitement d'images.

Le traitement d'images!!! ça me rappelle mon appareil photo argentique, enfin celui de mon frère que j'ai pris lorsqu'il était parti faire son service militaire et qui m'a permis découvrir la joie de la photographie et la richesse des images. Mais imaginer que ces mêmes images pouvaient servir à des choses plus sérieuses et intéressantes, je dois remercier Professeur Frédéric LERASLE pour son cours d'introduction à l'imagerie à l'Université Paul Sabatier lors de mon Master ISII et Dr. Ariane HERBULOT qui par ses différents cours n'a fait qu'accroître mon intérêt pour l'analyse d'images.

Le rêve n'est plus de comprendre les ondes hertziennes! Je songe maintenant à faire de la vision dans un contexte lié à la robotique. C'est donc le moment de remercier mon directeur de thèse, Professeur Jean-José ORTEU, qui m'a pris en stage lors de mon Master en Intelligence, Artificielle, Reconnaissance des Formes et Robotique. Il m'a donné l'opportunité de faire ma vraie première application de vision dans un contexte lié à la robotique, au sein de Noomeo et m'a ensuite donné la possibilité de faire cette thèse. Je dois remercier une deuxième fois mon directeur de thèse pour son soutien inconditionnel dans un contexte difficile, où notamment le premier partenaire industriel de ma thèse (celui qui me fournissait les données) a cessé son activité. Grâce à lui, nous avons pu compter avec la collaboration de G2Metric qui a permis de financer une grande partie de cette thèse. Je remercie donc Dr. Nicolas Cornille de la société G2Metric et Mr. François Gastou PDG de cette société, pour avoir accepté de collaborer avec nous et de financer la thèse. J'ai vraiment apprécié sa disponibilité et son soutien pendant la thèse et en particulier pendant les moments difficiles et notamment lors de la rédaction du manuscrit. Je remercie aussi mon encadrant, Dr. Florian Bugarin, pour son soutien, les rendez-vous chez G2Metric et la relecture du manuscrit.

J'en profite pour remercier Professeur Phillippe OLIVIER et Professeur Thierry CUTARD directeurs respectifs des laboratoires Institut Clément Ader et Institut Clément Ader, site d'Albi, pour m'avoir accueilli au sein de leurs équipes.

Je voudrais remercier sincèrement les membres du jury de cette thèse. D'abord M. ABABSA et M. LUCAS pour m'avoir fait l'honneur de rapporter sur mes travaux, et pour leurs rapports qui ont contribué à rendre ce manuscrit plus clair.

Je remercie M. DEVY pour avoir présidé le jury de ma thèse. Et je remercie M. IZAGUIRE pour avoir participé à ce jury. Ces remerciements sont extensibles à mes directeur et co-directeur de thèse, M. ORTEU et M. BUGARIN.

Je voudrais remercier Manuel João Ferreira de la société Enermeter (pour les illustrations 2.3 et 2.4) ainsi que Packt Publishing (pour l'illustration 2.2.3.1.4) et Springer / Universities Press (pour les illustrations 2.2.3.1.4 et 2.2.3.1.5) de m'avoir accordé la permission de reproduction de quelques images utilisées à des fins d'illustration dans mon chapitre 2.

Une thèse est un marathon, pendant ces années de thèse j'ai pu compter sur le soutien de nombreuses personnes, parmi lesquelles Dr. Ludovic BRETHERS, Dr. Bilal KARABAGLI, Dr. Xavier Maurice, Dr. Nicollas CORNILLE, Renaud JULIEN, Eric Briche, Didier ADE, qu'ils trouvent ici l'expression de ma gratitude.

Je voudrais aussi remercier les Enseignants-chercheurs de Mines-Albi qui m'ont ouvert les portes de l'enseignement supérieur en me donnant l'opportunité d'enseigner à leurs côtés. Merci donc à Prof. Jean-José ORTEU, Dr. Jean-Louis DIRION, Prof. Yannick LEMAULT, Dr. Vincent VELAY, Dr. Fabien BAILLON, Dr. Martial SAUCEAUX.

Je saisis ce moment pour remercier, les différents collègues de bureau :

Marie POITRIMOLT, Rafael d'ELIA, Aurélien MONDIERE, Charline LE NUÉ, Damien MAUDUIT, Anaïs FARRUGIA, Pauline LEPESANT, Samuel PETROCELLI, Igor JOVANCEVIC, Lilia ZEMINI.

Que dire du soutien des amis? Augustina EPHRAIM (Tina), Jonathan BOUILLONEC (Dr. JONJON), Karin FARIA PINHO, Jean-Pierre HOUEDJISSIN (JP), Sylvie MAGNANON et Luc GUIHARD, Florence MAGNANON et Pierre Menz-Pégail, Hélène MAGNANON, Arlette MAGNANON, Adnette LOKO, Renliw FLEURMOND, Guy-Aurel LOKO, Jasper LEAU-CHOY, Denyse AKAYEZU, Marcos PIRES DOS SANTOS, António LIMA BONFIM, Helmer-Rename DOS SANTOS, Miller FERNANDES, Maldinne TRINDADE

(Wiwi), Leeley DAIO, Alexander PINHEIRO, Geoffroy NGAÏKOMOS, merci pour votre amitié et votre soutien.

Et la dream team ? Sam, thank you for sharpenning my interest on visual tracking during these memorable conversations while Igor was deepy lost in his shape matching applications. So, thank you Igor for such moments. I am proud of you guys ; we converted Jonjon to the team ! Also, I would like to show my gratitude to Samuel for outstanding geographic lessons.

Remerciements en portugais/Agradecimentos em português :

Antes de terminar esta sessão de agradecimentos, queria agradecer aos meus pais Manuel João e Gelardina, e aos meus irmãos, Luísa, Clementino, Esperança, Jorge, Edgar, Gualter e Edmeia.

Queria também agradecer os meus amigos de infância que sempre deram uma força. Obrigado à : *Lúcio Serôdio, Josias Mayker, Jurácio Diogo, Ludmilo Aguiar, Alex Faleiro, Amilcar da Costa, Oldnyn dos Reis d'Alva !*

Estes agradecimentos são extensivos à Maísa Pina, Suzete Brandão, Euridse Viegas, Maria Gonçalves e todos aqueles que contribuíram de uma forma ou de outra nessa etapa.

Je voudrais témoigner de toute ma gratitude à celle qui a vécu de près ces années de travail et qui a supporté mes sautes d'humeur et mon manque de temps pendant ces années. Merci à toi chère Znayda !!! Et merci à notre fille Lisenay qui a égayé la fin de cette thèse.

Pour finir je voudrais dédier ce mémoire à ma regrettée grand-mère "avó Yana", à António Almeida (Toninho) le grand frère que j'aurais aimé avoir et qui est parti beaucoup trop tôt et à Marc Vellut.

Liste d'acronymes

AirCOBOT : Aircraft enhanced Inspection by smaRt & Collaborative rOBOT	IUTAM : International Union of Theoretical and Applied Mechanics
AGAST : Adaptive and Generic Accelerated Segment Test	KDR : Knots Depth Ratio
AKAZE : Accelerated-KAZE Features	LAAS : Laboratoire d'Analyse et d'Architecture des Systèmes
BPC : Bounded Partial Correlation	LDB : Local Difference Binary
BRIEF : Binary Robust Independent Elementary Features	LED : Light-Emitting Diode
BRISK : Binary Robust Invariant Scalable Keypoints	Matlab : Matrix Laboratory
CAD : Computer-Aided Design	MSER : Maximally Stable Extremal Regions
CAAMVis : Contrôle Automatisé d'Assemblages Mécaniques par Vision Artificielle	MOE : Modulus of elasticity
CCD : Charge-coupled device	MOR : Modulus of rupture
CAO : Conception Assistée par Ordinateur	MST : Mminimum Spanning Tree
CMOS : Complementary Metal-Oxide Semiconductor	NCC : Normalized Cross Correlation
CMY : Cyan, Magenta, Yellow	OpenCV : Open source Computer Vision
DoG : Difference of Gaussians	ORB : Oriented FAST and Rotated BRIEF
DCE : Discrete Curve Evolution	PCA : Principal Component Analysis
FAST : Features from Accelerated Segment Test	PDE : Partial Distorsion Elimination
FREAK : Fast Retina Keypoint	PTZ : Pan-Tilt-Zoom
HoG : Histogram of Oriented Gradient	QCAV : International Conference on Quality Control by Artificial Vision
HSV : Hue, Saturation, Value	RAM : Random Access Memory
GLOH : Gradient Location and Orientation Histogram	RNG : Relative Neighbourhood Graph
ICA : Institut Clément Ader	RGB : Red, Green, Blue
ICP : Iterative Closest Point	SAD : Sum of Absolute Differences
ICPRAM : International Conference on Pattern Recognition Applications and Methods	SEA : Successive Elimination Algorithm
	SSD : Sum of Squared Differences
	SIFT : Scale-invariant feature transform
	SURF : Speeded Up Robust Features
	STL : Standard Template Library
	SVG : Scalable Vector Graphics

VISUM : VISion Understanding and Machine intelligence

YCrB : espace colorimétrique représenté par luminance (Y) et différence chroma-

tique rouge et bleu

ZNCC : Zero Mean Normalized Cross Correlation

"The idea is to try to give all the information to help others to judge the value of your contribution, not just the information that leads to judgment in one particular direction or another"

[R.P. Feynman, Physicien]

1

Introduction générale

La vérification de la conformité de produits est pour une entreprise une garantie d'assurer la livraison de produits de bonne qualité à ses clients. Cela constitue également un moyen de réduire les coûts en évitant le retour de produits non conformes et de prévenir un accident en cas applications cruciales telles qu'en médecine ou dans certaines applications aéronautiques ou encore dans le domaine alimentaire. En ce sens, des efforts continus sont mis en œuvre par les organisations de qualité et de sécurité, par les industriels et les institutions de recherche avec l'objectif d'améliorer les normes et les méthodologies de contrôle-qualité. Le contrôle qualité par vision artificielle est un sujet largement étudié dans lequel il existe une vaste littérature [Modayur et al., 1992, Newman and Jain, 1995, Malamas et al., 2003, Golnabi and Asadpour, 2007].

Ce type d'inspection est adapté lorsque l'on souhaite effectuer le contrôle de façon non destructive. De plus, l'inspection visuelle automatisée permet d'assurer l'uniformité du contrôle et de maintenir une cadence soutenue de contrôle. Au début de l'inspection visuelle par les caméras, l'objet de référence était une image en noir et blanc qui servait de référence lors de la comparaison avec l'objet à contrôler. Progressivement, différentes approches ont émergé telles que la corrélation d'un objet non défectueux avec l'image test, la comparaison de contours ou des approches plus sophistiquées, notamment la comparaison d'un modèle représentant l'objet non défectueux avec l'objet test pouvant être défectueux. Ce dernier type d'approche est développé notamment dans les articles de Bukovec et al. [Bukovec et al., 2007] et de Mozina et al. [Mozina et al., 2013].

Ce manuscrit présente les travaux de recherche intitulés "Inspection d'assemblages mécaniques aéronautiques par vision artificielle", menés dans le cadre d'une thèse de doctorat réalisée au sein du laboratoire Institut Clément Ader (ICA) à l'Ecole des Mines d'Albi. Ces travaux résultent de deux collaborations successives entre *l'Institut Clément Ader* et deux partenaires industriels. D'abord, une première collaboration avec la société Noomeo dans le cadre du projet CAAMVis financé par la région Midi-Pyrénées, qui a permis de travailler sur le contrôle automatisé d'assemblages mécaniques par vision artificielle lors d'un stage de fin d'études de master entre mars et septembre 2012 et qui s'est poursuivi par cette thèse de doctorat qui a débuté en septembre 2012. La collaboration avec la société Noomeo a duré jusqu'en décembre 2013. Elle était sensée durer jusqu'en septembre 2015 mais la cessation d'activité de la société Noomeo a écourté sa durée. Nous avons par la suite entamé une nouvelle collaboration, avec la société G2Metric, dans le cadre du projet

Lynx©. Cette collaboration s’est étendue sur la période de janvier 2014 à juin 2015.

Sommaire

1.1	Structuration du manuscrit	7
1.2	Problématique : Comment exploiter la CAO	8
1.2.1	Le projet CAAMVis	8
1.2.1.1	Motivation et objectifs	8
1.2.1.2	Moyens matériels	9
1.2.1.3	Comment le modèle CAO est-il exploité dans ce projet	10
1.2.2	Le projet Lynx©	10
1.2.2.1	Motivation et objectifs	10
1.2.2.2	Moyens matériels	11
1.2.2.3	Comment le modèle CAO est-il exploité dans ce projet	13
1.3	Contributions	13
1.4	Conclusion	14

1.1 Structuration du manuscrit

Ce document se décompose en six chapitres, commençant par ce **chapitre 1** qui nous sert à introduire nos travaux, en expliquant notamment la problématique, le contexte des travaux menés dans le cadre de cette thèse ainsi que les moyens mis en œuvre pour pouvoir atteindre les objectifs fixés et les contributions qui ont pu être apportées. Le **chapitre 2** se focalise sur l’état de l’art en inspection par vision artificielle. Nous y abordons plusieurs aspects de l’inspection par vision, notamment la reconnaissance d’objets, qui est souvent une étape préalable à l’inspection, les systèmes d’acquisition rencontrés en inspection par vision artificielle ainsi que les approches d’inspection 2D ou 3D avec ou sans utilisation d’un modèle de Conception Assistée par Ordinateur, dit CAO. La première méthode d’inspection développée dans le cadre du projet CAAMVis est présentée dans le **chapitre 3**. Les travaux plus conséquents ont été réalisés dans le cadre du deuxième projet (Lynx©), avec la société G2Metric. Ces travaux sont décrits dans le **chapitre 4**. Le **chapitre 5** présente et discute des résultats obtenus, en décrivant notamment les tests réalisés avec des données synthétiques puis des tests réalisés avec des données réelles. Le **chapitre 6** conclut le manuscrit tout en donnant quelques pistes pouvant être exploitées dans des travaux futurs.

1.2 Problématique : Comment exploiter la CAO

Comment exploiter le modèle CAO et des techniques d’analyse d’images pour réaliser l’inspection automatisée ?. La problématique de cette thèse consiste à exploiter au mieux le modèle CAO afin de réaliser l’inspection d’assemblages mécaniques en combinant l’exploitation du modèle CAO et des techniques d’analyse d’images. Nous allons à travers deux projets différents expliquer comment atteindre cet objectif. Les annexes A et B montrent quelques exemples d’éléments présents sur l’assemblage à contrôler.

D’abord nous nous concentrerons sur les objectifs et les moyens disponibles pour la réalisation de chaque projet, ensuite nous évoquerons les deux façons selon lesquelles nous envisageons d’utiliser le modèle CAO, afin de pouvoir combiner son utilisation avec les outils d’analyse d’images pour réaliser l’inspection d’assemblages mécaniques. Dans ce chapitre introductif, nous nous contenterons d’évoquer comment le modèle CAO peut être exploité pour atteindre l’objectif d’inspecter un assemblage mécanique constitué notamment de rivets et des supports de fixation. Nous détaillerons ces deux stratégies d’utilisation du modèle CAO dans le **chapitre 3**.

1.2.1 Le projet CAAMVis

Dans le projet CAAMVis le modèle CAO (cf. figure 1.1) est combiné avec un capteur de type stéréovision (cf. figure 1.2) pour faire l’inspection d’un assemblage réel. Nous reviendrons sur ce point dans le **chapitre 3** qui détaille la méthode d’inspection développée pendant ce projet. Ici, nous allons commencer par expliquer le contexte du projet CAAMVis, les objectifs poursuivis et les moyens disponibles pour la réalisation de ce projet.

1.2.1.1 Motivation et objectifs

CAAMVis est un projet collaboratif qui a réuni deux entreprises (Noomeo et Kinéocam) et deux laboratoires de recherche (LAAS-CNRS et Institut Clément Ader) de la région Midi-Pyrénées autour d’un objectif commun, à savoir le **C**ontrôle **A**utomatisée d’**A**ssemblages **M**écaniques par **V**ision artificielle. Ce projet a duré de janvier 2011 à décembre 2013 et a été financé par la région Midi-Pyrénées. Le projet a été porté par la société Noomeo, avec qui nous avons travaillé plus étroitement. En tant qu’utilisateur final la société Aéroliia a mis à notre disposition un assemblage mécanique ainsi que le modèle CAO de cet assemblage pour que nous puissions réaliser nos tests. En amont de ce projet, la société Noomeo avait développé une application commerciale qui consistait en l’inspection d’assemblages mécaniques par vision en utilisant un capteur de stéréovision embarqué sur un bras robotisé, cf. figure 1.2a. Le but de ces travaux consistait à inspecter un assemblage mécanique constitué d’une quantité importante d’éléments, cf. figure 1.1a. Les points considérés dans

cette application, concernaient notamment la présence et l'absence d'un élément à contrôler, son placement et la vérification que l'élément présent était celui attendu. Lors de ses développements initiaux la société Noomeo s'était focalisée sur une méthode d'inspection basée sur une comparaison 3D/3D, en comparant le modèle 3D des éléments à contrôler avec son correspondant réel après une phase de reconstruction 3D. Dans le cadre du projet CAAMVis, nous nous sommes appuyés sur le matériel mis à disposition par la société Noomeo afin de prouver la faisabilité de la même application en combinant cette fois l'utilisation du modèle CAO à des techniques d'analyse d'images, sans passer par la phase de reconstruction 3D des objets réels. L'objectif était de développer une méthode d'inspection basée sur une comparaison 2D/2D, en comparant une image virtuelle issue du modèle CAO avec une image acquise avec le capteur physique. Afin de mieux comprendre notre contribution à ce projet, nous allons décrire les moyens mis à notre disposition dans le cadre du projet et ensuite nous expliquerons comment nous avons exploité le modèle CAO pour y extraire des informations pertinentes comparables aux informations extraites d'une image réelle. La méthode permettant d'effectuer la comparaison est décrite dans le **chapitre 3**.

1.2.1.2 Moyens matériels

En entrée du système d'inspection il y a le modèle CAO de l'assemblage à contrôler (cf. figure 1.1a), duquel on peut déduire le modèle de l'élément à contrôler. Le système comprend également un capteur de vision à double tête stéréo (développé par la société Noomeo) qui peut fournir le nuage des points 3D de l'élément à contrôler (cf. figure 1.2a), ainsi que deux images 2D (gauche et droite) de ce même élément à inspecter.

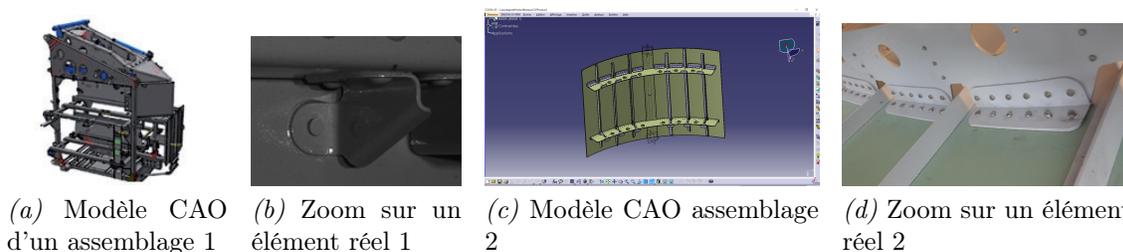


Figure 1.1 – CAAMVis : modèle CAO et assemblage réel. ⁽¹⁾

Le capteur constitué de deux têtes stéréo, comprend deux caméras grand champ et deux caméras petit champ. Deux projecteurs de mouchetis complètent le capteur, afin de favoriser la reconstruction 3D.

1. Des images de plus grande taille sont disponibles dans l'annexe D

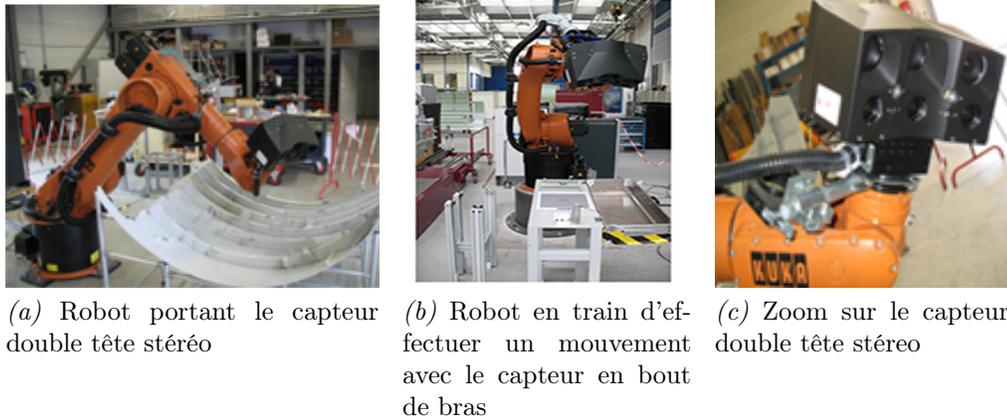


Figure 1.2 – Robot et capteur utilisés dans le projet CAAMVis

Les deux caméras grand champ permettent le recalage global du support d'assemblage physique avec la structure de l'assemblage du modèle CAO. Les deux caméras petit champ permettent de se focaliser sur l'objet d'intérêt.

Le capteur à double tête stéréo est porté par un robot manipulateur possédant six degrés de liberté. Les deux têtes stéréos sont calibrées et la transformation entre le robot et le capteur est aussi connue. En prenant l'une des deux caméras de la paire stéréo comme référence, la transformation entre le repère du monde et le repère de la caméra de référence est connue.

1.2.1.3 Comment le modèle CAO est-il exploité dans ce projet

Dans le cadre de ce projet, nous avons développé une méthode d'inspection d'assemblages mécaniques, basée sur la comparaison d'une image réelle provenant du capteur à double tête stéréo (cf. figure 1.2c) et d'une image synthétique issue du modèle CAO (cf. figure 1.1a). La stratégie de génération des données à partir du modèle CAO dans ce projet est décrite dans le **chapitre 3**, où nous présentons notre méthode d'inspection, qui cherche à optimiser la position et l'orientation d'un contour théorique au sein d'une carte de distance de l'élément réel à évaluer, après une étape de rasterisation qui nous permet d'obtenir une image de profondeur de l'élément théorique de référence dérivé de la CAO.

1.2.2 Le projet Lynx©

1.2.2.1 Motivation et objectifs

Suite à la cessation d'activité de la société Noomeo avec qui nous collaborions dans le cadre du projet CAAMVis, nous avons commencé une nouvelle collaboration avec la société G2Metric, dans le cadre du développement de leur projet Lynx©. Lynx© est un système de contrôle automatisé mis au point par la société G2Metric et qui se base sur l'analyse d'images fournies par une caméra PTZ (vision monoculaire) afin de vérifier la

conformité d'assemblages composés de rivets et de supports de fixation. Le contrôle dans Lynx© se base sur la comparaison (des régions d'intérêt) d'une image de référence d'un élément non défectueux et d'une image d'un objet potentiellement défectueux. De ce fait, le fonctionnement de Lynx© requiert la construction d'une base des données d'éléments de référence pour tout assemblage (cf. figure 1.3). La construction de la base des données est une étape chronophage et fastidieuse.

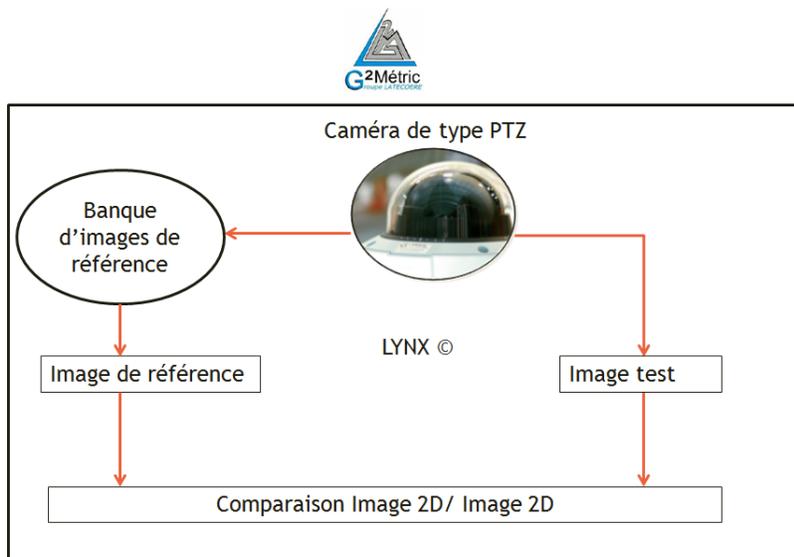


Figure 1.3 – Lynx© : schéma de principe

Dans le cadre de notre collaboration avec la société G2Metric, l'enjeu consiste à remplacer la base des données d'images de référence par le modèle CAO pour avoir plus de souplesse et de fiabilité au niveau du contrôle. En effet, lors de la construction d'une banque d'images de référence il se pose aussi la question de savoir qu'est-ce qu'une bonne image de référence, car l'image de référence peut elle-même changer selon les conditions d'acquisition. En revanche, en utilisant le modèle CAO, on s'affranchit de ce problème. Il faut tout de même garder à l'esprit que utilisation du modèle CAO pose d'autres questions telles que le choix du point de vue et la localisation (nous entendons par localisation le fait d'avoir l'élément issu de la CAO dans le même repère que l'objet réel à évaluer.) Outre ces problèmes, parfois, il se pose également le problème d'éclairage de la scène. Néanmoins, de manière générale on gagne en répétabilité en utilisant le modèle CAO. La figure 1.4 illustre à la fois la comparaison des régions d'intérêt dans Lynx© et le problème d'images de référence que nous évoquions précédemment.

1.2.2.2 Moyens matériels

Au début du projet Lynx© nous ne disposions pas encore d'un couple "modèle CAO + assemblage réel". La société G2Metric a fabriqué un assemblage réel à partir duquel le mo-

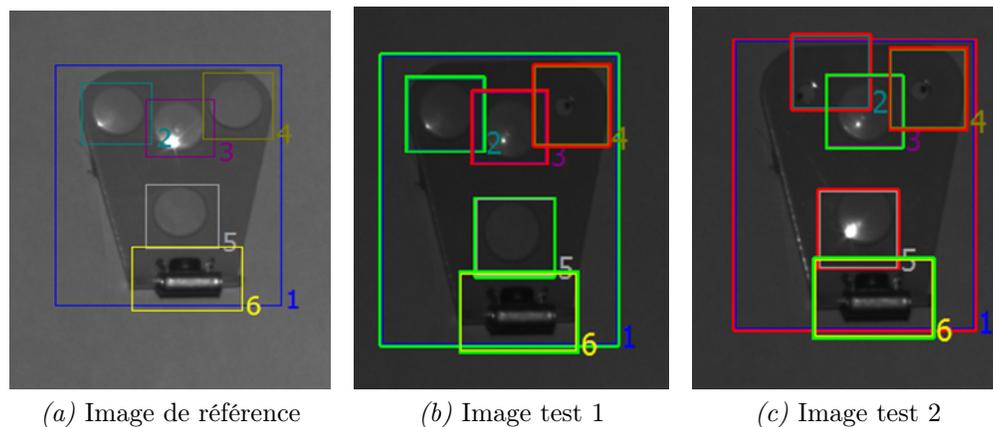


Figure 1.4 – Exemple de contrôle réalisé avec Lynx

dèle CAO a été créé en rétro-conception. Sur la figure 1.5, nous pouvons observer une copie d'écran de ce modèle CAO créé en rétro-conception ainsi qu'une image de l'assemblage réel.

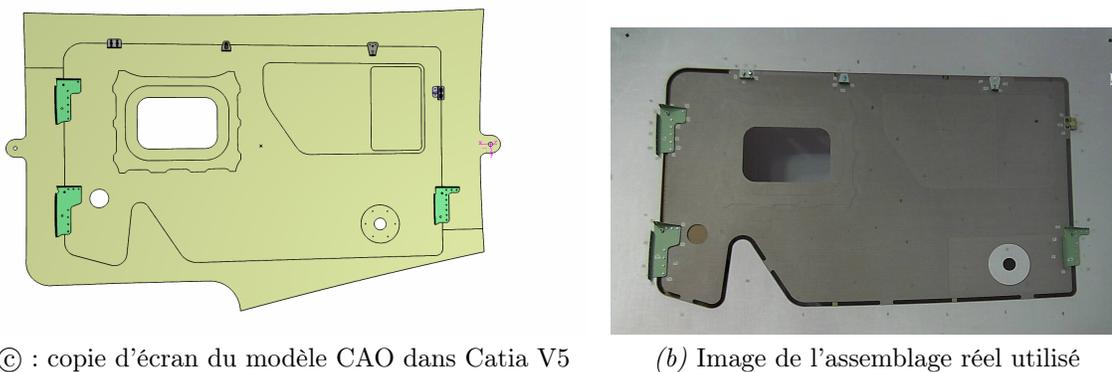


Figure 1.5 – Assemblage utilisé dans le projet Lynx : copie d'écran du modèle CAO dans la maquette Catia V5 et image de l'assemblage réel correspondant

Le système d'acquisition utilisé pendant ce projet consiste en le couplage d'une caméra PTZ avec un dispositif d'éclairage (cf. figure 1.6). À des fins de tests, nous avons aussi travaillé avec une caméra CCD classique.



(a) Système d'acquisition avant packaging



(b) Système d'acquisition après packaging

Figure 1.6 – Système d'acquisition utilisé dans Lynx©

Nous avons également utilisé l'assemblage mécanique prévu pour le projet CAAMVis pour nos tests, notamment lors de la phase d'estimation de pose préalable à la projection du modèle CAO sur le plan image (cf. annexe D).

1.2.2.3 Comment le modèle CAO est-il exploité dans ce projet

Comme nous l'avons vu, l'objectif du projet Lynx© consiste à introduire le modèle CAO dans l'inspection automatisée, afin de rendre l'inspection plus flexible et de renforcer sa fiabilité par l'utilisation d'un modèle de référence plus pertinent qui est le modèle CAO. L'une des premières difficultés rencontrées lorsqu'un modèle CAO est utilisé pour l'inspection réside dans la localisation de celui-ci par rapport à la caméra réelle. Dans ce projet, le modèle CAO utilisé est produit avec le progiciel CATIA V5 © de Dassault Systèmes et nous disposons des fichiers au format CATPRODUCT et CATPART. Nous avons envisagé deux façons distinctes d'utiliser le modèle CAO dans une tâche d'inspection :

- ◊ Premièrement, positionner d'abord la caméra virtuelle dans la maquette numérique puis positionner la caméra réelle dans l'environnement réel.
- ◊ Deuxièmement, positionner d'abord la caméra réelle dans l'environnement réel, la calibrer et estimer sa pose et réutiliser ces éléments pour projeter le modèle CAO sur le plan image.

Nous reviendrons sur ces deux stratégies dans le **chapitre 3** et donnerons des détails concernant les expérimentations réalisées en suivant ces deux stratégies dans l'annexe D.

1.3 Contributions

La première méthode que nous présenterons dans le **chapitre 3**, développée dans le cadre du projet CAAMVis, a été présentée lors du congrès international IUTAM 2012, qui s'est déroulé entre le 3 et le 6 novembre à Taiwan [Viana et al., 2012]. La version française de cet

article est disponible dans les archives ouvertes [Viana et al., 2013]. Nous avons communiqué sur cette thèse dans le cadre de la conférence annuelle réalisée par l'Ecole Doctorale Systèmes (EDSYS), en avril 2014 [Viana et al., 2014]. Dans le cadre du projet Lynx©, nous avons développé une méthode d'inspection d'assemblages mécaniques en exploitant conjointement le modèle CAO et une image réelle de l'objet à évaluer. Cette méthode est basée sur la mise en correspondance de primitives issues du modèle CAO servant de référence et des primitives issues de l'objet à inspecter. La mise en correspondance, dit appariement, est effectuée à l'aide de deux graphes attribués dont les attributs des nœuds servent comme points d'entrée d'une fonction de similarité qui calcule la ressemblance entre deux graphes. Pour garantir l'unicité, les scores d'appariements sont injectés dans un graphe biparti qui est optimisé pour garder la meilleure paire formée entre un nœud de référence et un nœud représentant l'élément à évaluer. La méthode a été testée avec les données de synthèse et avec des données réelles ce qui confirme son efficacité pour la tâche à accomplir. Une première version de cette méthode a fait l'objet d'une communication orale [Viana et al., 2015b] au congrès scientifique international sur Contrôle et Qualité par Vision Artificielle (QCAV en anglais), qui a eu lieu du 3 au 5 juin 2015 au Creusot, France. Nous avons communiqué sur ces travaux par poster lors de l'école d'été VISUM 2015 à Porto, Portugal [Viana et al., 2015a]. Une version étendue de cette méthode a été publiée dans *Journal of Electronic Imaging*, en ligne depuis décembre 2015 [Viana et al., 2015c]. Par ailleurs, dans le cadre d'un travail collaboratif entre l'ICA et le LAAS-CNRS (travail dans le cadre du projet AirCOBOT) nous avons présenté nos travaux lors de la conférence internationale ICPRAM 2016. Dans ces travaux la méthode est utilisée pour la mise en correspondance de primitives du modèle d'un avion avec des primitives extraites de l'image réelle, afin d'aider à la tâche de navigation d'un robot et d'inspection d'un aéronef, après la localisation du robot [Jovancevic et al., 2016].

1.4 Conclusion

Dans ce chapitre d'introduction, après avoir décrit la façon dont le manuscrit est structuré, nous avons exposé la thématique de la thèse présentée dans ce manuscrit et qui consiste en l'exploitation du modèle CAO, couplé à des techniques d'analyse d'images afin de réaliser l'inspection d'assemblages mécaniques, dans un contexte aéronautique. S'il n'y a pas de doute quant à l'utilité du modèle CAO en inspection, son utilisation suscite incontestablement la question de la localisation du modèle CAO par rapport à l'objet acquis avec un capteur physique. Nous avons abordé, à travers le projet CAAMVis et à travers le projet Lynx© deux façons différentes dont le modèle CAO peut être exploité afin d'obtenir des données synthétiques pouvant être comparées aux données réelles. Ces données sont ensuite utilisées dans le développement de deux méthodes de comparaison entre objets

théoriques du modèle et objets réels. Ces deux méthodes sont présentées dans les **chapitres 3 et 4**.

Nous avons également détaillé nos contributions en termes de publications et communication réalisées pendant la thèse. Avant d'entamer la discussion sur nos méthodes d'inspection (cf. **chapitres 3 et 4**), dans le **chapitre 2** suivant nous dressons un état de l'art sur l'inspection visuelle automatisée. Cet état de l'art abordera l'inspection visuelle automatisée dans sa généralité et l'inspection avec le modèle CAO en particulier.

2

Inspection visuelle automatisée et reconnaissance d'objets

L'objectif de ce chapitre est d'aborder la problématique de l'inspection visuelle automatisée dans sa généralité. En effet les techniques d'inspection sont nombreuses et elles se basent sur différents types d'information (photométriques, géométriques ou hybrides), selon la nature des éléments à inspecter. Parce que l'inspection visuelle et la reconnaissance d'objets sont deux sujets très proches, dans ce chapitre nous verrons la différence entre ces deux sujets et nous expliquerons comment les méthodes de reconnaissances d'objets (locales, globales) sont utilisées dans l'inspection. Ensuite le problème d'inspection de pièces mécaniques, avec ou sans l'utilisation du modèle de CAO sera abordé.

Sommaire

2.1	Considérations générales	17
2.1.1	Définitions	17
2.1.2	Quelques systèmes d'acquisition utilisés en inspection visuelle	17
2.2	Méthodes de reconnaissance d'objets	20
2.2.1	La photométrie	20
2.2.2	Les descripteurs locaux et méthodes à base d'histogrammes	23
2.2.3	Reconnaissance par apprentissage	27
2.2.4	La géométrie 2D	34
2.3	Inspection sans modèle CAO	35
2.3.1	Inspection dans l'industrie agroalimentaire	36
2.3.2	Inspection dans le secteur de la banque et de la finance	36
2.3.3	Inspection dans l'industrie pharmaceutique	37
2.3.4	Inspection dans l'industrie électronique	37
2.3.5	Inspection dans l'industrie automobile	38
2.3.6	Inspection dans l'industrie aéronautique	39
2.3.7	Autres	40
2.4	Inspection de pièces mécaniques à l'aide d'un modèle CAO	40
2.4.1	Méthodes basées sur une comparaison 3D/3D	41
2.4.2	Méthodes basées sur une comparaison 2D/2D issue du modèle 3D	42

2.5 Conclusion	43
--------------------------	----

2.1 Considérations générales

2.1.1 Définitions

2.1.1.1 Reconnaissance d'objets

En vision par ordinateur, la reconnaissance d'objets est l'action de retrouver un objet dans une image, séquence vidéo ou nuage de points 3D. L'invariance par rapport aux transformations géométriques est l'un des éléments essentiels en reconnaissance d'objets. Ainsi, la recherche d'un objet d'intérêt dans une image, par rapport à une ou N images contenues dans une base de données est considérée satisfaisante si le même objet est contenu dans l'image test et dans l'image de référence. Les méthodes de reconnaissance doivent permettre de retrouver l'objet cible même en cas de rotation, translation, ou changement d'échelle. Mais leur but ne consiste pas à aller mesurer ces transformations géométriques afin d'établir la conformité de l'objet tel qu'on le fait en inspection visuelle automatisée.

2.1.1.2 Inspection visuelle automatisée

L'inspection est le processus selon lequel on détermine si un produit dévie ou non d'un ensemble de spécifications [Newman and Jain, 1995, Moron, 1996, Prieto et al., 2002]. L'inspection visuelle automatisée consiste en la vérification automatique de la qualité d'un objet par rapport à l'ensemble des spécifications prédéfinies, en utilisant des données obtenues grâce à un capteur de vision artificielle (capteur de vision monoculaire, capteur de type stéréovision, rayon X, caméra hyperspectrale, caméra infrarouge, etc).

2.1.2 Quelques systèmes d'acquisition utilisés en inspection visuelle

En inspection visuelle automatisée et de manière générale en vision par ordinateur le succès de l'algorithme utilisé pour une tâche donnée est étroitement lié à la qualité des images d'entrée. Ces images proviennent de différentes sources d'acquisition, et bien souvent elles sont associées à un système d'éclairage lorsqu'il est possible de contrôler l'environnement de travail. Un système d'acquisition bien adapté permet d'obtenir des images de qualité c'est-à-dire avec une bonne résolution, un faible taux de bruit, de sorte à mettre en relief la zone d'intérêt, afin de permettre de mieux voir ce dont il est question. Le choix de systèmes imageurs est très vaste et dans cette section nous nous attacherons à présenter quelques systèmes d'acquisition et à expliquer les raisons qui justifient un choix au détriment d'un autre pour une tâche en particulier car les facteurs comme le coût du système et la place

qu'il peut occuper sont autant d'éléments importants. Cette description des systèmes d'acquisition utilisés en inspection commence par une brève présentation de la caméra matricielle standard CCD, suivi de la caméra linéaire, fonctionnant comme un scanner et adaptée pour les lignes de production (tapis roulant). Ensuite les systèmes d'acquisition d'images tels que la caméra Pan-Tilt-Zoom (PTZ), une caméra qui a la possibilité de pivoter horizontalement, verticalement en plus de pouvoir zoomer sera décrite. Puisqu'il y a des choses qui se passent et que l'être humain n'est pas capable d'observer à l'œil nu, les systèmes imageurs tels que les rayons-X et des caméras fonctionnant sur des plages spectrales au-delà du visible ou avec plus de résolution spectrale (trois à quelques centaines de canaux) seront aussi présentés dans cette section, qui finira par une description des systèmes d'acquisition utilisés dans nos travaux et auxquels nous avons déjà fait référence dans le chapitre précédent.

2.1.2.1 Caméra matricielle

Le terme caméra matricielle vient du fait que les photo-détecteurs (surface sensible du capteur) à l'intérieur de la caméra sont rangés en une grille contrairement à la caméra linéaire (cf. section 2.1.2.2) où ils sont rangés en une simple ligne.

Les caméras matricielles sont les plus répandues. En 2013, Stemmer Imaging [Stemmer, 2013] rapporte que dans les application de vision, la vente des caméras matricielles dépasse celles des caméras linéaires d'un facteur 10. En vision, on rencontre couramment les caméras matricielles (cf. figure 2.3), à technologie CCD [Perng et al., 2011] ou CMOS. Les deux technologies existent depuis des nombreuses années mais la technologie CCD est plus communément utilisée. Néanmoins, la technologie CMOS semble rattraper son retard ces dernières années. Ces technologies standards sont couramment employées notamment en raison de leur faible coût. En général, ce faible coût a comme contrepartie des problèmes liés à l'objectif. En pratique ce problème peut être résolu grâce à une étape de calibrage de la caméra. Si les caméras matricielles sont de manière générale massivement utilisées en vision, elles le sont particulièrement en inspection. Par exemple, dans le tableau de Cubero et al. [Cubero et al., 2011], répertoriant les systèmes d'acquisition utilisés en inspection visuelle, dans le domaine agroalimentaire, on constate une présence significative des caméras matricielles CCD notamment dans l'inspection de la partie extérieure des fruits.

2.1.2.2 Systèmes à base de caméra linéaire

Contrairement à la caméra matricielle, la caméra linéaire utilise une seule ligne de photo-détecteurs pour capturer un flot d'images d'un objet en déplacement. Ce flot d'images est traité par un système de vision intégré à la caméra afin de créer des images 2D qui sont généralement utilisées à des fins de contrôle-qualité industriel. Les caméras linéaires sont très utilisées dans les lignes de production où l'on cherche à inspecter les articles à un

rythme soutenu. De plus, elles offrent souvent une haute résolution et elles permettent de travailler à des distances de travail réduite pouvant même atteindre 10 *mm*.

Elles existent en plusieurs formats et utilisent aussi bien la technologie CCD [Excelitas, 2015] que la technologie CMOS [Teledyne, 2015].

Ce type de système d'acquisition a été utilisé notamment par la société portugaise Enermeter dans une application d'inspection de marquage pneumatique, en association avec d'autres types de capteurs, cf. figure 2.4.

2.1.2.3 La flexibilité du système Pan-Tilt-Zoom

Le système PTZ est intéressant en ce sens qu'il permet à la fois de viser une cible et de la suivre grâce à sa capacité de zoomer et de pivoter selon les deux axes (X et Y). Cette flexibilité lui vaut d'être utilisé dans différents types d'applications notamment en télé-surveillance où il est omniprésent mais son utilisation n'est pas limitée à ce seul domaine. Ce système PTZ est aussi utilisé dans d'autres domaines, notamment dans la navigation robotique terrestre.

Au même titre que les caméras PTZ on pourrait citer les caméras omnidirectionnelles de type fishe-eye grand angle, qui ont un angle de vue important et couvrent un champ de vision pouvant atteindre 360°. Elles sont utilisées dans des applications d'inspection, de surveillance, d'observation du ciel (suivi de nuages), où encore dans des applications de navigations terrestre (Google Street View, Globbespotter et Cyclorama de Cyclomedia). En comparaison avec les caméras grand angle de type fishe-eye, le système PTZ a l'avantage de permettre de zoomer sur l'objet d'intérêt, sans que l'on ait à déplacer la caméra.

2.1.2.4 Au-delà de l'œil humain

En plus des systèmes conventionnels déjà décrits dans cette section, les systèmes fonctionnant sur des plages fréquentielles non visibles à l'échelle de l'œil humain tels que les caméras infrarouges, ultra-violettes, hyperspectrales, ou encore les capteurs à rayon-X [Chuang et al., 2010] existent et sont utilisés dans des applications d'inspection visuelle automatisées. L'intérêt de ces systèmes est évident car ils permettent de percevoir ce que l'œil humain n'est pas capable d'observer. La caméra hyperspectrale est par exemple très utilisée dans l'industrie agroalimentaire, elle permet d'observer jusqu'à l'intérieur des fruits [Cubero et al., 2011]. Les caméras infrarouges sont quant à elles très utiles pour leur capacité à détecter des objets dans des conditions où il y a peu ou pas d'éclairage. La contrepartie de ce type de système d'acquisition est en revanche leur coût élevé qui peut atteindre plusieurs dizaines de milliers d'euros. Dans la section 2.3 ci-dessous nous décrivons quelques applications utilisant ce type de système d'acquisition permettant de voir dans le domaine du "non visible" à des fins d'inspection automatisée. Pour une lecture plus complète sur les systèmes d'acquisition d'images, le lecteur est invité à lire l'ouvrage

The Imaging and Vision Handobook [Stemmer, 2013].

2.1.2.5 Nos systèmes d'acquisition

Comme nous l'avons expliqué dans le **chapitre 1**, pour effectuer les travaux présentés dans ce manuscrit nous avons eu recours à deux systèmes d'acquisition différents. Pour le projet CAAMVis, présenté dans le **chapitre 3**, le système d'acquisition utilisé consiste en l'utilisation d'un robot portant un capteur de type stéréo-vision à double tête. Pour le développement de la méthode d'inspection présentée dans le **chapitre 4** le système d'acquisition est constitué d'une caméra PTZ, associée à un système d'éclairage directionnel qui se déplace de façon synchronisée avec la caméra PTZ lorsque cette dernière est dirigée vers un objet à évaluer. La solution consistant à utiliser un robot portant un capteur est économiquement plus couteuse.

2.2 Méthodes de reconnaissance d'objets

En inspection, reconnaître un objet est une condition nécessaire mais elle n'est pas suffisante. La reconnaissance d'objet est un pré-requis à l'inspection. Les méthodes de reconnaissance d'objet, lorsqu'elles sont adaptées au type de données à traiter, peuvent être utilisées en inspection visuelle automatisée, en y intégrant la notion de respect de spécifications (cf. définition de l'inspection dans la section 2.1.1.2). Cette partie de la thèse s'attache à décrire, de manière non exhaustive, quelques méthodes de reconnaissance d'objets répertoriées dans la littérature et aussi le rôle qu'elles peuvent avoir dans l'inspection visuelle.

2.2.1 La photométrie

En reconnaissance d'objets ou même en inspection, la couleur ou la luminance peut jouer un rôle clé lorsque les objets à reconnaître peuvent être ainsi différenciés. Parmi les techniques de reconnaissance exploitant l'information de luminance directement, on peut citer :

- ◇ Template matching : mise en correspondance de deux motifs. Le "template matching", comparaison d'un motif ou modèle de référence à l'objet test que l'on cherche à reconnaître, et ses différentes variantes.
- ◇ Comparaison colorimétrique en utilisant l'espace colorimétrique adapté ainsi que le système d'acquisition adéquat.

2.2.1.1 Reconnaissance d'objets par Template Matching

Le template matching est une technique d'analyse d'image, conceptuellement simple qui consiste à trouver un "template" T (modèle, motif ou imagerie contenant la forme à trouver) dans une image I [Nixon and Aguado, 2008, Kohandani et al., 2006]. En analyse

d'images, le "template matching" est très utilisé pour la reconnaissance d'objets. La corrélation [Kohandani et al., 2006, Briechele and Hanebeck, 2001, Malinsky et al., 2012, Gangodkar et al., 2012, Konishi et al., 2012] est souvent employée à cette fin. Dans ce cas, le template est normalement défini comme une imagerie en niveau de gris, à comparer à l'image test. C'est un produit scalaire des niveaux de gris de l'image servant de référence avec l'image cible, dans une fenêtre de taille de l'imagerie de référence. Minimiser le critère SSD eq.(2.9) revient à maximiser la corrélation eq. (2.1) [Martin and Crowley, 1995].

$$C(x, y) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} T_{m,n}(x, y) I(x + m, y + n) \quad (2.1)$$

où $T_{m,n}$ représente l'imagerie de référence T , de dimension (M, N) .

$I(x + m, y + n)$ est l'image test à la position (m, n) , décalée de x et de y dans un plan.

$C(x, y)$ représente la valeur de la fonction de corrélation entre I et T pour chaque position du template T dans l'image test I .

Pour des raisons de robustesse aux changements d'illumination de la scène et pour la non-sensibilité aux bruits, la corrélation croisée (nous utiliserons le terme corrélation par souci de concision) a été étendue, donnant lieu à plusieurs variantes dont certaines décrites ci-dessous.

Afin de mieux appréhender les différentes versions de la corrélation, nous allons introduire quelques notions de base.

Soit :

$$E_I(x, y) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} I(x + m, y + n)^2 \quad (2.2)$$

Énergie de l'image I pour une fenêtre de la taille du template T

$$E_T = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} T_{m,n}(x, y)^2 \quad (2.3)$$

Énergie du template T

$$\mu_I = \frac{1}{M \cdot N} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} I(x + m, y + n) \quad (2.4)$$

moyenne de l'image I pour une fenêtre de la taille du template T

$$\mu_T = \frac{1}{m.n} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} T_{m,n}(x, y) : \quad (2.5)$$

moyenne du template

Parmi les différents scores de corrélation on peut citer :

- ◇ Normalized Cross Correlation (NCC)

$$NCC(x, y) = \frac{T_{m,n}(x, y)I(x + m, y + n)}{\sqrt{E_I \cdot E_T}} \quad (2.6)$$

- ◇ Zero mean Normalized Cross-Correlation (ZNCC)

$$ZNCC(x, y) = \frac{\sum_{m=0}^{M-1} \sum_{n=0}^{N-1} (T(x, y) - \mu_T)(I(m + x, n + y) - \mu_I)}{\sqrt{\sum_{m=0}^{M-1} \sum_{n=0}^{N-1} (I(m + x, n + y) - \mu_I)^2 \cdot \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} (T_{m,n}(x, y) - \mu_T)^2}} \quad (2.7)$$

Par ailleurs, les critères SAD eq. (2.8) et SSD eq. (2.9) (proche de la corrélation) sont des mesures de similarité ou de distorsion [Kohandani et al., 2006, Gangodkar et al., 2012] régulièrement retenues pour le matching, en raison de leur faible coût de calcul.

- ◇ Somme des différences absolues

$$SAD(x, y) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} |I(x + m, y + n) - T_{m,n}(x, y)| \quad (2.8)$$

- ◇ Somme des différences carrées

$$SSD(x, y) = \sum_{m=0}^M \sum_{n=0}^N (I(m + x, n + y) - T_{m,n}(x, y))^2 \quad (2.9)$$

Parmi toutes ces techniques de mises en correspondance entre deux images, celles basées sur la NCC sont généralement les moins sensibles aux variations d'illumination de la scène et aussi plus robustes aux bruits si elles sont comparées à des techniques de mesure de distorsions telles que MAD, SAD, SSD [Gangodkar et al., 2012]. Ces techniques trouvent plusieurs applications dans la vision artificielle, notamment : le recalage d'image, l'estimation de mouvement, l'estimation de carte de disparité etc, comme rapporté par Gangodkar dans [Gangodkar et al., 2012]. Néanmoins ces techniques souffrent d'un handicap majeur qui est le temps de calcul, en particulier celles basées sur la corrélation et ses variantes.

Dans le but de minimiser ce problème, plusieurs méthodes ont été proposées dans la littérature. Par exemple, Luigi Di Stefano et Stefano Mattoccia ont présenté deux travaux très intéressants sur des variantes de la NCC et de ZNCC.

L'intérêt de leur méthode *Bounded Partial Correlation (BPC)* présentée dans l'article [Stefano and Mattoccia, 2003] réside dans le fait que les pixels ne pouvant pas permettre un score de corrélation meilleur que le pixel courant sont éliminés. Les auteurs ont utilisé les algorithmes *Successive Elimination Algorithm –SEA–* et *Partial Distorsion Elimination –PDE–* [Jung et al., 2000] ce qui, d'après eux, permet un gain considérable de temps de calcul. Dans un autre travail, [Stefano et al., 2005] une amélioration de la BPC est proposée, celle-ci utilise ZNCC (plus robuste que NCC) comme fonction de mesure.

2.2.1.2 La comparaison basée couleur

La couleur est une perception créée en réponse à l'excitation de notre système de vision dû à la lumière incidente sur la rétine dans la plage spectrale visible. La couleur d'une image peut contenir un nombre important d'informations qui peuvent être utilisées pour simplifier l'analyse d'images, l'identification d'objets, et l'extraction des caractéristiques distinctives grâce à la couleur [García et al., 2015]. Dans cette section nous abordons la comparaison basée couleur dans le but de faire de la reconnaissance. L'extraction des indices colorimétriques doit être effectuée dans un espace colorimétrique adapté. Si nous prenons comme exemple, l'espace colorimétrique RGB, composé des trois couleurs primaires et qui est très couramment utilisé en vision, cet espace colorimétrique est affecté par la brillance, c'est-à-dire qu'il n'est pas invariant à la différence de luminance, donc il n'est pas forcément le mieux adapté pour des opérations sur des images où l'on sait qu'il peut y avoir une variation de brillance [García et al., 2015]. Parmi les différents espaces colorimétriques, qui peuvent être utilisés afin de différencier des objets selon leur couleur, on peut citer HSV, CMY, YCrB, etc.

2.2.2 Les descripteurs locaux et méthodes à base d'histogrammes

Les descripteurs locaux font partie des méthodes utilisées en première ligne pour reconnaître les objets, que ce soit une reconnaissance 2D ou même 3D, où ils interviennent notamment dans le processus de mise en correspondance entre deux images dans le but d'effectuer la reconstruction tridimensionnelle de l'objet. Fondamentalement, on peut dire que la reconnaissance par les méthodes à base de descripteurs locaux repose sur trois points, consistant en la détection de points d'intérêt suivie d'une étape de description/caractérisation des points les plus robustes avant de finir par une la mise en correspondance entre les points de deux images.

L'objectif premier de ces méthodes est d'assurer un taux élevé de répétabilité tout en

permettant aux éléments d'intérêt d'être les plus discriminants possibles et ceci malgré le fait que l'image soit susceptible de subir des transformations géométriques (changement d'échelle, rotation, par exemple) ou contenir du bruit. Parmi ces méthodes, on distingue :

2.2.2.1 SIFT

SIFT, Scale Invariant Features Transform a été présentée par David Lowe en 2004 [Lowe, 2004]. Cette méthode de détection et caractérisation de points d'intérêt dans l'image est l'une des méthodes fondatrices de ce type d'approche où la reconnaissance est basée sur les descripteurs locaux. Comme l'indique son nom, SIFT est invariante au changement d'échelle. SIFT présente également l'invariance par rapport à la rotation. La détection des points d'intérêt dans SIFT, est réalisée dans un espace linéaire gaussien, suite à la convolution d'un noyau gaussien avec des pyramides d'images originales. Les points d'intérêt correspondent en effet à des extremas dans l'image résultante de la convolution des différences de gaussiennes (DoG) avec différentes résolutions de l'image originale. La description intègre l'orientation des gradients dans une fenêtre autour du point d'intérêt et permet d'assurer l'invariabilité du descripteur par rapport à la rotation. Le résultat de la description de SIFT est un vecteur de dimension 128.

2.2.2.2 SURF

SURF est l'acronyme de Speeded-Up Robust Features [Bay et al., 2006]. Il s'agit d'une méthode de descripteurs locaux qui s'inspire de SIFT, avec deux objectifs majeurs, la réduction du temps de calcul et plus de liberté dans l'utilisation comparée à SIFT qui est propriétaire. Dans la littérature, SURF est reconnu être plus rapide que SIFT, tout en permettant une caractérisation des points d'intérêt garantissant un niveau équivalent de répétabilité et de robustesse. La taille du vecteur résultant de la description par SURF est un vecteur de dimension 64 mais certaines versions étendues existent avec une dimension supérieure (128 par exemple). La rapidité de SURF comparée à SIFT réside notamment dans l'utilisation d'images intégrales lors de la description des points d'intérêt, c'est-à-dire des blocs rectangulaires dans lesquels est réalisée la description.

2.2.2.3 KAZE et AKAZE

KAZE est une technique récente de détection et caractérisation de points d'intérêt dans l'image. Cette méthode constitue une alternative sérieuse à des méthodes telles que SIFT et SURF qui ne sont pas tout à fait libres d'utilisation (c'est d'ailleurs pour cette raison qu'elle ne sont pas dans la partie standard de la très célèbre librairie de vision OpenCV). La nouveauté dans KAZE lorsque comparée avec les autres méthodes vient du fait que la détection et la description des points d'intérêt sont réalisées dans un espace non linéaire

permettant de préserver notamment les contrastes locaux, ce qui améliore la détection, contrairement à des méthodes telles que SIFT où la recherche de l'invariance à l'échelle est réalisée dans un espace linéaire suite à la convolution d'images pyramidales avec un noyau gaussien. Selon les auteurs de KAZE, bien que cette étape de détection soit plus lente que SURF par exemple, la méthode permet de dépasser SIFT et SURF et en terme de performance machine et en terme de résultats. À noter que dans la méthode KAZE, la description est basée sur une modification de la méthode SURF, nommée M-SURF (pour SURF modifié). AKAZE est la version accélérée de KAZE, développée par les mêmes auteurs de KAZE. Ils ont proposé une amélioration de leur propre méthode en introduisant le concept de FAST Explicit Diffusion qui permet de mieux gérer la non linéarité de leur étape de détection. Et dans la description, les auteurs utilisent une modification de LDB (Local Difference Binary [Yang and Cheng, 2014]) afin d'accélérer les calculs. Bien que très rapide, la rapidité d'AKAZE n'entraîne pas de baisse dans sa qualité. En effet, AKAZE permet d'obtenir des résultats comparables aux méthodes de description plus réputées et permet d'après ses auteurs de les dépasser parfois. AKAZE est une méthode nouvelle qui n'a pas tardé à séduire la communauté de vision artificielle, elle est déjà implémentée dans la populaire librairie de vision OpenCV.

2.2.2.4 BRISK

BRISK signifie Binary Robust Invariant Scalable Keypoint [Leutenegger et al., 2011] Pour sa partie détection, BRISK est une combinaison de FAST améliorée, en rendant ce dernier invariant à l'échelle. Sa partie description se base sur le descripteur BRIEF [Calonder et al.,], le permettant d'atteindre des performances très intéressantes au niveau du temps de calcul. Dans la littérature, BRISK a la réputation d'être plus rapide que SIFT et SURF [Haller, 2015] tout en permettant d'obtenir des performances comparables à ces deux descripteurs bien établis depuis plus d'une décennie.

2.2.2.5 ORB

Oriented Fast and Rotated BRIEF, dit ORB [Rublee et al., 2011] est une méthode qui combine les forces de FAST et de BRIEF (détecteur et descripteur bien établis) pour renforcer leurs points forts tout en compensant leurs points faibles. FAST par exemple est amélioré en termes de qualité de détection. Plutôt que d'utiliser un détecteur de coins simplifié, la métrique utilisée est celle de détecteurs de coins de Harris. De même, FAST est rendu invariant à l'échelle par utilisation de pyramides d'images à différentes échelles tout comme dans SIFT. L'invariance à la rotation est obtenue en exploitant la théorie des moments. La description, à l'instar de ce qui est fait dans BRISK, est réalisée en employant BRIEF, ce qui encore une fois permet d'atteindre des performances élevées en temps de calcul puisque BRIEF se base sur des opérations binaires très simples.

2.2.2.6 FREAK

Fast Retinal Keypoints [Alahi et al., 2012] est un descripteur local bioinspiré et récent (2012) qui essaye d'émuler le comportement de l'oeil humain afin d'arriver à une performance élevée en termes de qualité de description des points d'intérêt. Pour utiliser FREAK il faut au préalable faire une détection des points d'intérêt. Son intérêt réside dans la caractérisation à la fois rapide et efficace des points, ce qui permet de garantir la robustesse (recherche d'invariants) et la discrimination recherchée lorsqu'on utilise un descripteur local de points d'intérêt. Il s'inspire de BRISK pour la partie orientation et en partie d'ORB pour la partie matching.

2.2.2.7 Résumé sur les méthodes de détection et description de points d'intérêt

Dans ces travaux, le but n'est pas d'établir un outil de comparaison pour ces méthodes de détection et/ou description des points d'intérêt dans l'image. L'évaluation de ces méthodes est un sujet aussi ancien que les méthodes elles-mêmes. Néanmoins, ce sujet reste ouvert puisque des nouvelles apparaissent régulièrement. Le lecteur intéressé par l'évaluation de ces types de méthodes est invité à consulter les travaux suivants : [Haller, 2015, Canclini et al., 2013, Heinly et al., 2012, Ziegler et al., 2012, Mikolajczyk and Schmid, 2005, Miksik and Mikolajczyk, 2012].

Nombreuses sont les méthodes à base de descripteurs locaux ou des méthodes utilisant l'information d'histogramme de l'image pour faire la reconnaissance. Le sujet principal de ce manuscrit n'étant pas la reconnaissance d'objets, nous invitons le lecteur intéressé à développer cette thématique de reconnaissance d'objets à lire les articles sur DAISY [Tola et al., 2010], MESR [Matas et al., 2004], FAST, [Rosten and Drummond, 2006], AGAST [Weiss et al., 2008], GLOH, [Mikolajczyk and Schmid, 2005], HoG [Dalal and Triggs, 2005], ou encore les contributions plus récentes de David Lowe (créateur de SIFT) sur ce sujet [Muja and Lowe, 2009, Muja and Lowe, 2012].

Cette section du manuscrit a pour seul but de mentionner ces méthodes en ce sens qu'elles permettent de faire la reconnaissance qui est souvent une tâche préalable à d'autres tâches en vision artificielle y compris l'inspection visuelle automatisée. Néanmoins, compte tenu de la spécificité de notre tâche d'inspection (exploitant le modèle CAO pour comparaison d'une image théorique à une image réelle), ces types de méthodes ne sont pas employables car celles-ci se basent toutes sur l'intensité de l'image. Or, les images que nous utilisons dans nos méthodes d'inspection ne sont pas comparables au niveau de leur luminance. En effet nous utilisons une image de référence issue du modèle CAO, où seule la géométrie est comparable à l'image réelle acquise par le capteur PTZ.

2.2.3 Reconnaissance par apprentissage

Pourquoi est-il intéressant d'effectuer l'apprentissage automatique? À l'instar de l'inspection automatique, l'apprentissage automatique a un double but. Selon [Webb, 2002], en applications pratiques on aimerait épargner à l'opérateur humain des tâches difficiles (classification) et si possible rendre cette tâche plus fiable. L'auteur explique qu'il y a des environnements qui sont difficiles et d'autres inaccessibles pour l'homme et dans lesquels la tâche de classification n'est pas fiable si réalisée par l'humain. C'est dans ce contexte que l'on utilise des méthodes d'apprentissage automatique. En apprentissage artificiel on assigne des étiquettes aux données.

Selon [Murty and Devi, 2011] il existe essentiellement deux grands paradigmes utilisés en reconnaissance des formes, à savoir, les approches basées sur l'analyse statistique et celles basées sur l'analyse syntaxique.

La première est utilisée largement parce que les données contiennent des imperfections (bruit) et une modélisation statistique ou probabiliste permet de traiter ces données.

L'approche syntaxique est plus utilisée en théorie des langages. Dans cette section nous abordons la reconnaissance au sens du premier paradigme. L'apprentissage artificiel (au sens statistique) se décompose en deux grandes familles :

- ◇ Apprentissage supervisé
- ◇ Apprentissage non-supervisé

Nous revenons sur ces deux méthodes d'apprentissage dans les sections suivantes. Il est important de noter qu'on peut combiner ces approches entre elles pour obtenir une approche de reconnaissance hybride, où l'apprentissage supervisé peut servir à initialiser le système et l'apprentissage non-supervisé le complète en rajoutant de l'abstraction au moment où il n'a plus de connaissance *a priori* à injecter dans le système.

2.2.3.1 Apprentissage supervisé

L'apprentissage supervisé est une famille de méthodes de "machine-learning" ou apprentissage artificiel qui apprend à travers l'usage des données connues et étiquetées. En apprentissage artificiel supervisé, on entraîne la méthode grâce à des données labellisées et appartenant à une classe connue [Baggio et al., 2012], le but étant d'associer toute nouvelle donnée à une nouvelle classe [Murty and Devi, 2011]. Reinhard Klette [Klette, 2014] rajoute que cela est fait manuellement, à base d'expertise. Cette étape où l'on assigne des étiquettes connues à des données est dite phase d'apprentissage. L'autre étape de la méthode est la phase test. Celle-ci consiste à évaluer la méthode afin d'observer son

comportement face à une nouvelle donnée. Est-ce qu'elle le classe correctement ? Les méthodes d'apprentissage statistiques sont nombreuses. Nous allons décrire celles qui sont le plus couramment rencontrées.

2.2.3.1.1 K-NN

L'algorithme de K-NN (K-Nearest Neighbours) est un algorithme intuitif qui classe toute nouvelle donnée en fonction de la classe d'appartenance de ses K plus proches voisins qui ont été classés au préalable. Cet algorithme est intuitif mais n'est pas robuste aux *outliers*.

2.2.3.1.2 Classifieur bayésien

Le classifieur bayésien est populaire en reconnaissance des formes parce qu'il est un classifieur optimal [Murty and Devi, 2011]. En d'autres mots, il est possible de démontrer que la classification obtenue avec ce classifieur minimise la probabilité moyenne de l'erreur. Andrew Webb [Webb, 2002] démontre cela (cf. page 7, section intitulée "*Bayes decision rule for minimum error*"). Le classifieur bayésien utilise la règle de probabilité conditionnelle de Bayes afin de déduire l'information *a posteriori* de l'information *a priori*. Considérons C classes, w_1, \dots, w_c avec $p(w_1), \dots, p(w_c)$ représentant les probabilités *a priori* de chacune des classes (supposées connues). Minimiser la probabilité de commettre une erreur en regardant uniquement la donnée à classer sans tenir compte de la distribution de la probabilité de la classe impliquerait de classer un élément dans une classe w_j à condition que :

$$p(w_j) > p(w_k) \quad k = 1, \dots, C; k \neq j \quad (2.10)$$

En connaissance d'un vecteur d'observation x , si on veut associer les éléments de x à une classe. Il faut construire une règle de probabilité basée sur $x, p(w_j|x)$. À ce moment, on associe x à w_j si :

$$p(w_j|x) > p(w_k|x) \quad k = 1, \dots, C; k \neq j \quad (2.11)$$

Sauf que la probabilité $p(w_j|x)$ n'est pas connue à ce stade. On connaît $p(x)$ la probabilité d'avoir x , la probabilité d'avoir $p(w_j)$ et $p(x|w_j)$. Grâce au théorème de probabilité conditionnelle de Bayes on déduit la probabilité *a posteriori* suivante :

$$p(w_j|x) = \frac{p(x|w_j)p(w_j)}{p(x)} \quad (2.12)$$

Ainsi, la règle de probabilité d'erreur minimale consiste à associer à x à w_j si :

$$p(w_j|x)p(w_j) > p(w_k|x)p(w_k) \quad k = 1, \dots, C; k \neq j \quad (2.13)$$

2.2.3.1.3 Arbres de décision

Les arbres de décisions sont des graphes en forme d'arbres [Hackeling, 2014]. On peut les voir comme un jeu de questions sur des attributs d'un objet permettant de découvrir celui-ci. Le but étant de découvrir l'objet le plus tôt possible. Dans un arbre de décision on décompose l'ensemble de variables d'apprentissage en sous-ensembles jusqu'à ce qu'il ne soit plus possible de décomposer. Entre un ensemble et ses sous-ensembles il y a une flèche qui pointe vers le sous-ensemble. Souvent le sous-ensemble est représenté par un rectangle à l'intérieur duquel on note ces caractéristiques [Hackeling, 2014]. D'autres représentations consistent à utiliser des ronds à tous les niveaux, exception faite à un sous-ensemble constituant une feuille (conduisant à une classe), [Webb, 2002] La décomposition en sous-ensemble doit être effectuée de façon intelligente, c'est-à-dire, permettant d'obtenir une décision (correcte) le plus tôt possible. En ce sens, une de métriques couramment utilisées est l'entropie qui représente la quantité d'information ou l'incertitude sur les données. Dans la décomposition en sous-ensemble on cherche à minimiser l'incertitude sur les variables utilisées. On minimise l'entropie :

$$H(x) = - \sum_{i=1}^n p(x_i) \log_b p(x_i) \quad \text{avec } b = \text{base logarithmique au choix} \quad (2.14)$$

D'autres métriques sont utilisées, notamment la métrique dite de Gini impurity qui mesure la proportion de classes dans un ensemble. Elle est définie par :

$$Gini(x) = 1 - \sum_{i=1}^n p(i|x)^2 \quad (2.15)$$

ou encore la métrique permettant de maximiser le gain. Cette métrique est basée sur l'entropie. Le livre [Murty and Devi, 2011] présente plusieurs exemples employant cette méthode d'apprentissage afin de réaliser la classification automatique.

2.2.3.1.4 L'algorithme du Perceptron

Le Perceptron est un algorithme de classification binaire, qui tente de simuler le cerveau humain. Le cerveau est composé de cellules appelées neurones qui traitent l'information. Les neurones se connectent entre eux et échangent l'information via les synapses. Un neurone est composé de **dendrites** qui reçoivent les signaux électriques des autres neurones. Les signaux sont traités au niveau du **corps** du neurone qui envoie un signal à un autre

neurone via l'**axone**. Un perceptron fonctionne de la même façon. Il reçoit une ou plusieurs entrées, les traite et renvoie une sortie. [Hackeling, 2014]

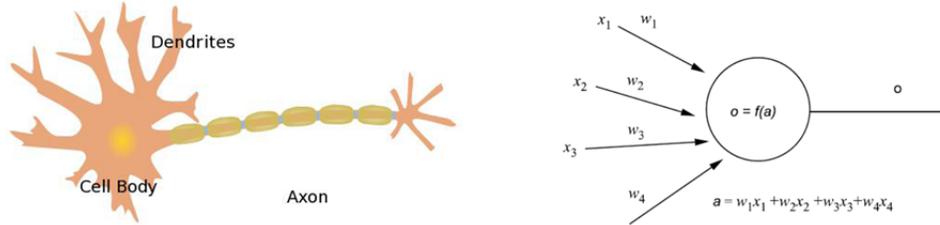


Figure 2.1 – Exemple d'un neurone biologique à gauche (extrait de [Hackeling, 2014]), neurone artificiel à droite (extrait de [Murty and Devi, 2011])

Un neurone artificiel est donc une combinaison linéaire d'entrées pondérées, mises ensemble pour produire une décision binaire. Souvent, il y a une entrée supplémentaire qui représente le biais. La sortie s'écrit alors :

$$y = \Phi\left(\sum_{i=1}^n w_i x_i + b\right) \quad (2.16)$$

La décision binaire en sortie dépend de la fonction d'activation. Souvent sont utilisées les fonctions d'Heaviside définie par :

$$g(x) = \begin{cases} 1 & \text{si } x > 0 \\ 0 & \text{sinon} \end{cases} \quad (2.17)$$

$$(2.18)$$

2.2.3.1.5 Perceptron multicouche ou réseau de neurones

Le Perceptron est capable de séparer les classes de manière linéaire et incapable de séparer des données non linéairement séparables. Pour vaincre cette limitation l'algorithme du Perceptron multicouche utilise des cascades de perceptrons. On parle alors de réseaux de neurones. Le principe est le même que celui du Perceptron mais on n'a plus une seule couche mais plusieurs couches, dont les couches cachées permettant de gérer la non linéarité. L'algorithme du perceptron multicouches procède en deux temps. Dans un premier temps (*forward*), il calcule les sorties en fonction des entrées et des couches cachées. L'activation des couches cachées en fonction des entrées x_1, x_2, \dots, x_d et des poids $(w_{11}, w_{12}, \dots, w_{21}, w_{22}, \dots, w_{dk})$ est donnée par l'équation suivante :

$$ah_j = x_1 \times w_{1j} + x_2 \times w_{2j} + \dots + x_m \times w_{mj} \quad \text{pour } 1 < j < k, \text{ avec } k \neq j \quad (2.19)$$

Et la $j^{\text{ième}}$ sortie d'une couche cachée en fonction de la couche cachée précédente et des différents poids des couches cachés (h_{11}, h_{kn}) est activée par :

$$a_1 = oh_1 \times h_{11} + oh_2 \times h_{21} + \dots + oh_k \times h_{k1} \quad (2.20)$$

Si on considère les sorties $o_i = f(a_i), \dots, o_n = f(a_n)$ et si on dispose des mesures t_i, \dots, t_n , alors on peut calculer l'erreur entre la mesure et la sortie de l'algorithme du réseau des neurones en écrivant :

$$e_i = t_i - o_i \quad (2.21)$$

Dans un second temps (*backtrack*) on ré-injecte les informations de mesures permettant de minimiser l'erreur entre les différentes sorties et les mesures et réajuste les différents poids à partir des sorties vers les entrées, par l'équation suivante (algorithme du gradient) :

$$h_{ji} = h_{ji} + \alpha \times o_j \times e_i \times g'(a_i) \quad (2.22)$$

Une fonction d'activation très utilisée dans cet algorithme est la fonction sigmoïde :

$$f(x) = \frac{1}{1 + e^{-x}} \quad (2.23)$$

qui a des propriétés intéressantes en termes de dérivées et qui sont exploitées notamment dans la phase du backtrack utilisant souvent la descente de gradient et donc les dérivées.

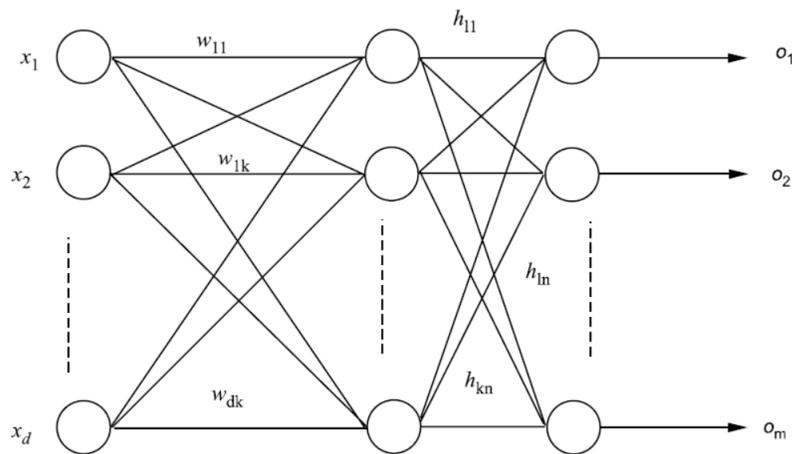


Figure 2.2 – Exemple d'un réseau de neurones (extrait de [Murty and Devi, 2011])

L'apprentissage automatique supervisé ne se limite pas aux méthodes que nous venons de

présenter. Il existe d'autres méthodes tels que les Chaines de Markov Cachées, Séparateur à vastes marges dit aussi machine (SVM) à support de vecteurs ou des méthodes de régression linéaires et non linéaires que nous n'avons pas abordées. La littérature sur ce sujet est vaste, et nous recommandons la consultation des ouvrages [Murty and Devi, 2011, Hackeling, 2014, Webb, 2002].

2.2.3.2 Apprentissage non-supervisé

En apprentissage non-supervisé, on n'a pas la connaissance *a priori* sur l'appartenance des données à une classe [Klette, 2014].

2.2.3.2.1 L'algorithme du K-means

K-means est une méthode dite de clustering (partitionnement de données) en apprentissage non-supervisé, très populaire grâce à sa rapidité et à sa capacité à s'adapter au changement d'échelle au niveau de la quantité des données [Hackeling, 2014]. *A priori*, on ne connaît pas le nombre de classes que l'on cherche à avoir pour le problème considéré. K est un hyper-paramètre qui permet alors de fixer le nombre de classes de l'algorithme qui déplace le centre des classe de manière itérative vers le point moyen des instances qui la composent et recommence le processus d'affectation d'une instance à une classe jusqu'à convergence. La fonction objectif à minimiser dans un tel processus est donnée par :

$$J = \sum_{k=1}^K \sum_i^{C_k} \|x_i - \mu_k\|^2 \quad (2.24)$$

2.2.3.3 Apprentissage profond

Cette famille de méthodes en expansion actuellement vise à remplacer l'apprentissage supervisé où on est obligé d'étiqueter les objets avec leur classe d'appartenance pendant la phase d'entraînement de la méthode, en rajoutant de l'abstraction grâce notamment à l'introduction d'un nombre élevé de couches. Il existe plusieurs architectures de ce type de méthodes, à savoir, réseau de neurones profond, réseau de neurones convolutifs ou encore réseau de neurones récurrent. Dans ce manuscrit nous n'avons pas la prétention de décrire ces méthodes. Le lecteur souhaitant apprendre plus sur ces méthodes est invité à consulter la page de référence <http://www.deeplearningbook.org/> du livre [Goodfellow et al., 2016] sur l'apprentissage profond.

2.2.3.4 Réduction de dimension en apprentissage

Lorsque l'on manipule une grande quantité de données ou que l'on cherche à avoir un algorithme d'apprentissage performant et rapide, incontestablement on se pose cette

question de réduction de dimensions des données. La réduction des données implique supprimer du vecteur de description des données toutes celles qui sont redondantes ou qui ont peu d'importance. Willi Richert et Pedro Luis Coelho [Richert and Coelho, 2013] évoquent cinq bonnes raisons pour pratiquer la réduction de dimensions :

- ◇ En général, les données superflues peuvent tromper le classifieur
- ◇ Plus la dimension des données est importante, plus il y a de paramètres à régler et plus il y a un risque de surapprentissage (*overfitting*)
- ◇ Les données utilisés pour résoudre le problème sont souvent de dimension élevée tandis que le problème lui même peut se résoudre avec des données d'une dimension moins importante.
- ◇ Plus la dimension est faible, plus l'apprentissage est rapide et plus on a la possibilité d'essayer des combinaisons différentes.
- ◇ Si on veut visualiser les données, on est limité en nombre de dimensions.

Il existe plusieurs types d'approches permettant de réduire la dimension des données. L'analyse en composante principale (PCA) est l'un des plus simples et le plus répandu des algorithmes de réduction de dimension.

Une autre méthode de réduction des dimensions couramment employée est l'analyse linéaire discriminante (LDA). Cette méthode maximise la distance entre points appartenant à des classes différentes et minimise celle entre les points au sein d'une même classe. On dit qu'elle maximise la variance inter classe et minimise la variance intra-classe. Un autre type d'approche de réduction des dimensions est l'utilisation des méthodes de positionnement multidimensionnel (Multidimensional Scaling, MDS) [Webb, 2002].

2.2.3.5 Lien entre l'apprentissage et l'inspection

L'apprentissage automatique permet de classer les objets et le lien qu'il peut y avoir entre ces méthodes de reconnaissance de formes et l'inspection est évident. On a dit en début de chapitre que la reconnaissance est un pré-requis en inspection. Parfois ces méthodes de reconnaissance suffisent à elles seules pour effectuer l'inspection. Typiquement en inspection on cherche à classer les objets en deux catégories : défectueux ou non défectueux. À titre d'exemple, l'inspection consistant à vérifier si un billet de banque est un faux ou un vrai, se limite au classement binaire. Parfois, en inspection on cherche dans un premier temps à regrouper des objets en différentes familles (ex. vis, supports, rivets, etc, en inspection de pièces métallique) avant la classification finale en défectueux ou non défectueux. Dans un cas comme dans l'autre on peut aisément faire appel à ces méthodes de reconnaissance des formes. Dans les sections 2.3 on verra des exemples d'applications d'inspection qui

utilisent une méthode d'apprentissage automatique. Même si ces méthodes sont très populaires, la nécessité d'avoir recours à une quantité importante de données pour enrichir la base d'apprentissage conditionne parfois leur utilisation.

2.2.4 La géométrie 2D

La géométrie est intrinsèque à l'objet, que l'on utilise une caméra en niveaux de gris, en couleur, ou même lorsque l'on crée une image virtuelle d'un objet, sa géométrie reste invariante. De ce fait, il existe énormément de méthodes de reconnaissance d'objets basées sur l'exploitation d'attributs géométriques. Dans [Musé, 2004] Pablo Musé présente les trois étapes importantes de la reconnaissance basée sur l'exploitation de la géométrie, concrètement sur l'aspect forme de l'objet. Ces étapes sont l'extraction/description des attributs leur appariement et la décision après l'appariement. Ici, nous allons mettre l'accent sur l'extraction et description car c'est l'étape qui conditionne tout le processus. D'ailleurs, selon [Musé, 2004] "Ce qui permet la reconnaissance des formes est le fait qu'elles aient des caractéristiques spécifiques."

"Le problème d'extraction de primitives consiste à définir un ensemble (le plus petit possible) de caractéristiques avec un grand pouvoir de discrimination", [Musé, 2004]. Ces caractéristiques spécifiques peuvent être globales (tout l'objet), locales ou semi-locales. Nous allons aborder quelques-unes de ces caractéristiques dans les sections suivantes.

2.2.4.1 Primitives et attributs géométriques

En observant notre environnement, on constate que les objets autour de nous, aussi bien les objets naturels que ceux fabriqués par l'Homme peuvent souvent être approximés par un ensemble de formes géométriques de base. Ils sont souvent constitués de segments de droite, de cercles, d'ellipses ou arcs d'ellipses, présentent plus au moins une courbure, etc. Aussi, en vision artificielle, lorsque cela est possible, ces types de primitives géométriques sont utilisés pour reconnaître les objets. Ce sont des caractéristiques plutôt locales, mais selon l'objet considéré, ces caractéristiques peuvent être globales.

2.2.4.2 Contour et squelette

Détecter le contour d'un objet dans une image permet d'alléger la quantité de données à manipuler, sans pour autant perdre l'essentiel de l'information géométrique nécessaire à l'analyse de l'objet. De plus, le contour de l'objet permet un passage aisé vers d'autres types d'informations. On peut par exemple extraire la topologie de l'objet à partir du contour en effectuant une transformation de l'axe médian, dit squelette. Ces caractéristiques sont des caractéristiques globales de l'objet. Le squelette peut aussi être obtenu à

partir de la transformée en distance, du diagramme de Voronoi, d'algorithmes d'amincissement ou encore en exploitant la morphologie mathématique. Le squelette est largement utilisé pour reconnaître des objets en effectuant l'appariement topologique de l'objet test avec l'objet de référence. Un des problèmes auquel font face les approches basées sur le squelette est la sensibilité de celui-ci aux déformations du contour de l'objet. En effet lorsque le contour de l'objet est un peu bruité, cela peut engendrer un changement topologique de sa forme. En conséquence, les méthodes basées sur le squelette nécessitent souvent d'être régularisées. Bai et Latecki [Bai and Latecki, 2008] ont pu reconnaître des objets en effectuant une recherche de similarité de squelette après son élagage en utilisant l'algorithme "Discrete Curve Evolution" (DCE). Un autre élément important dont il faut tenir compte est le fait que deux objets très semblables peuvent avoir deux squelettes très différents. Et réciproquement, deux objets différents peuvent avoir deux squelettes très similaires. Ceci est illustré dans l'article de Bai et Latecki. Pour remédier à ce problème, Karabagli [Karabagli, 2013] a par exemple couplé le résultat de l'appariement topologique du squelette avec d'autres mesures de similarité : la forme et le recouvrement de points de contours. De manière générale, l'utilisation de la topologie est efficace pour reconnaître les objets, mais dans une application d'inspection dimensionnelle où la reconnaissance seule n'est pas suffisante il est nécessaire de coupler l'information topologique avec un autre type de mesure permettant d'obtenir l'information dimensionnelle afin de savoir si les spécifications et tolérances dimensionnelles sont respectées.

2.3 Inspection sans modèle CAO

Dans cette section, nous verrons différentes méthodologies d'inspection visuelle automatisée et pour souligner l'omniprésence de cette étape très importante dans le cycle de fabrication et même de vie d'un produit, nous allons marquer cette transversalité par une description touchant à des domaines très variés tels que l'agroalimentaire et l'aéronautique, en passant par la banque, l'industrie pharmaceutique, etc. Traditionnellement, l'inspection visuelle automatisée est réalisée sans le modèle CAO, car celui-ci n'est pas toujours disponible ou n'est pas toujours approprié. C'est le notamment le cas dans le domaine agroalimentaire. Aussi, notre description concernera dans un premier temps les applications d'inspection plus traditionnelles i.e qui n'utilisent pas de modèle CAO comme élément de référence. Les applications utilisant le modèle CAO seront présentées dans la section 2.4. Dans un premier cas nous nous concentrerons sur les applications 2D et dans un deuxième cas, nous décrirons également quelques applications touchant à la 3D.

2.3.1 Inspection dans l'industrie agroalimentaire

L'industrie agroalimentaire est l'une de celles qui bénéficient depuis toujours de la fiabilité et de l'objectivité de la vision artificielle pour accomplir la tâche d'inspection. Elle est riche en littérature et cette richesse peut être exploitée dans d'autres domaines car même si les systèmes de vision sont souvent dédiés pour une tâche particulière [Newman and Jain, 1995], il y a souvent des points communs entre les tâches ou des sous-tâches de vision, soit dans la méthodologie, soit dans le type de capteur à utiliser. L'inspection automatisée dans l'industrie agroalimentaire est très diversifiée et concerne la vérification de la maturité de fruits, la couleur, l'aspect, la forme et les dimensions des fruits et légumes. Elle concerne aussi l'évaluation d'autres caractéristiques notamment le classement des graines, la fraîcheur des poissons, le séchage de cacao, etc. Pour pouvoir rendre compte d'une recherche de qualité dans des produits aussi variés, différentes modalités d'acquisition d'images et méthodes d'inspection sont employées. Dans [Soto et al., 2015], Juan Soto et al. combinent la richesse de l'imagerie hyperspectrale avec l'analyse en composante principale (PCA) et la logique floue afin de vérifier la qualité des graines de cacao (donnée par le degré de fermentation) pendant la phase de séchage. En effet, l'imagerie hyperspectrale (caméra Resonon, 200x640 pixels par image, composée de 240 bandes spectrales sur une plage spectrale allant de 400 nm à 900 nm) permet de capturer suffisamment de détails qui permettent de distinguer clairement non seulement les couleurs mais aussi les fissures à l'intérieur de graines de cacao. L'analyse en composante principale est utilisée afin de sélectionner les informations pertinentes adaptées à l'étude de séchage de graines. Dans cette application deux attributs permettent de caractériser la graine : l'indicateur de fissure (zone fissurée par rapport à la surface de la graine) et l'indicateur de couleur interne. Les deux attributs sont normalisés entre $[0, 1]$ et décomposés en 3 et 4 familles respectivement. Les auteurs utilisent ensuite la logique floue pour classifier les graines selon un ensemble de 12 règles pré-établies. Dans le livre [Davies, 2012] au chapitre 21, l'auteur décrit une application d'inspection automatisée de graines de céréales.

2.3.2 Inspection dans le secteur de la banque et de la finance

De nos jours, la vision est partout. Dans le secteur bancaire elle est utilisée pour la surveillance des lieux, mais elle a aussi un autre rôle très important qui consiste en la prévention de la contrefaçon. La tâche d'inspection visuelle automatisée dans ce domaine concerne la vérification des billets de banque. Les systèmes déployés à cette fin utilisent des systèmes imageurs qui permettent à la fois la recherche de signatures colorimétriques dans le domaine du visible et aussi des signatures invisibles à l'œil nu. Dans [Bruna et al., 2013] les auteurs décrivent une méthode de contrôle automatique de billets de banque

permettant à la fois d'identifier la valeur des billets d'euro et d'identifier les faux billets. La méthode est basée sur l'imagerie infra-rouge et utilise un système d'apprentissage pour identifier les signatures fiables dans des vrais et des faux billets de banque lors de la phase dite d'apprentissage. Pendant cette phase le système apprend également les caractéristiques des signatures permettant de reconnaître la valeur du billet contrôlé. Lors de la phase de test, les caractéristiques apprises sont comparées aux caractéristiques des billets à contrôler. La décision quant à la validité ou non du billet est prise en comparant différents attributs statistiques des signatures du billet à valider avec les statistiques disponibles dans la base d'apprentissage.

2.3.3 Inspection dans l'industrie pharmaceutique

L'industrie pharmaceutique utilise les systèmes d'inspection basés sur la vision afin de vérifier notamment l'étiquetage des flacons et le marquage sur les tablettes de médicaments. En effet, pour la protection des consommateurs, la réglementation impose aux agents pharmaceutiques de produire des médicaments qui puissent être identifiés sans ambiguïté. Ces réglementations concernent l'identification du principe actif contenu dans le médicament, le dosage, les dimensions des tablettes, le marquage, etc, [Mozina et al., 2013]. Bukovec et al. proposent dans [Bukovec et al., 2007] deux approches d'inspection de tablettes : une approche basée sur les *a priori* géométriques et une approche basée sur l'exploitation d'un modèle statistique d'apparence de la tablette, construit à l'aide de l'analyse en composante principale (PCA). Dans cette approche, les auteurs comparent la tablette à inspecter à un modèle d'apparence d'une tablette non défectueuse. Le modèle d'apparence est construit dans la phase dite d'apprentissage (hors ligne) et l'inspection est réalisée pendant la phase dite de test (en ligne). Mozina et al. rapportent dans [Mozina et al., 2013] que cette approche basée sur le modèle statistique d'apparence fonctionne mais qu'elle n'est pas robuste au problème de rotation qui peut apparaître pendant la production des comprimés. Ils proposent donc de coupler le modèle statistique d'apparence à un modèle de rotation et obtiennent ainsi un meilleur résultat.

2.3.4 Inspection dans l'industrie électronique

L'une des plus importantes applications de la vision artificielle dans l'industrie électronique concerne la vérification de circuits électroniques [Chuang et al., 2010] intégrés et la production de composants électroniques de manière plus générale. Perng et al. dans [Perng et al., 2011] ont proposé un système automatique pour vérifier la conformité des LED (*Light Electroluminescent Diodes*) montées en surface d'un circuit intégré, aussi bien pendant la phase de fabrication que dans la phase d'emballage des LED. Dans cette application le but était de vérifier plusieurs éléments de conformité parmi lesquels le bon placement de la LED (polarité et orientation), l'état de surface, la présence/absence d'éléments, etc.

L'algorithme d'inspection utilisé dans cette application se décompose en deux parties. La première partie concerne la phase d'apprentissage où les auteurs cherchent à identifier la zone à inspecter (précisée par un expert de contrôle qualité) dans l'image de l'élément de référence, en appliquant séquentiellement les opérations de binarisation, fermeture morphologique, et une transformation affine, afin d'extraire les caractéristiques discriminantes de l'objet de référence. Ensuite, pendant la phase de test, une comparaison (seuillage) de paramètres de l'objet à évaluer est faite avec les valeurs obtenues pendant la phase d'apprentissage afin de classer l'objet à inspecter en "défectueux" ou "non défectueux".

2.3.5 Inspection dans l'industrie automobile

La tâche d'inspection automatisée est très présente dans l'industrie automobile et concerne entre autres, la vérification du bon état de pneus, pendant la phase de production. Elle concerne également, la vérification d'autres équipements comme le siège de voiture ou l'uniformité de l'état de surface des objets, etc. Dans [Ferreira, 2015] Manuel João Ferreira a présenté deux applications d'inspection automatique par la vision artificielle. La première application, composée d'une partie matérielle (cf. 2.3) et d'une partie logicielle, concerne l'inspection du dossier de siège de voiture. Dans cette application le but est de vérifier l'uniformité de la couleur. Aussi les auteurs cherchent à s'assurer que la couleur utilisée est la bonne couleur, qu'il n'y a pas des petites différences de nuances. La partie matérielle est composée de deux caméras haute résolution et de deux diffuseurs de lumière, un de chaque côté du dossier de siège à inspecter. L'approche algorithmique dans cette application consiste en une trame séquentielle où les opérations suivantes sont réalisées. D'abord l'acquisition de dix images avec différentes ouvertures de diaphragme, suivie de l'identification de la région d'intérêt dans l'image ayant la luminance moyenne sur les 10 images prises. Ensuite, la texture est supprimée grâce à la morphologie mathématique. Les signatures de couleurs sont récupérées et classées par une approche bayésienne.

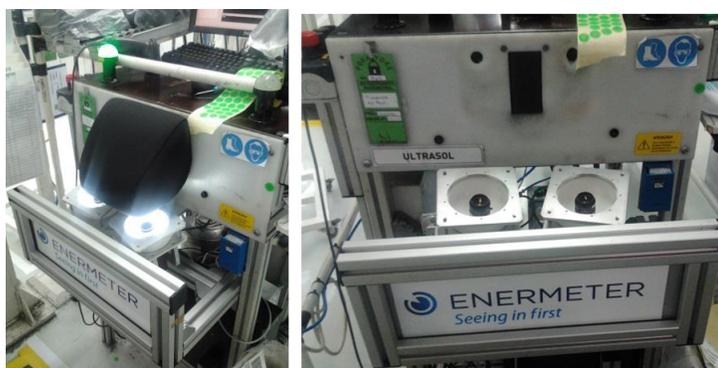


Figure 2.3 – Système ESOBASIS d’Enermeter, partie matérielle pendant le fonctionnement (gauche) et à l’arrêt (droite)

La deuxième application d'inspection présentée par Manuel João Ferreira, ESOTIREINSP,

concerne l'inspection de pneus pendant la phase de production afin de vérifier les marquages sur les flans ainsi que la conformité de la bande de roulement. Le système d'acquisition (cf. figure 2.4) de cette application d'inspection est composé d'une caméra linéaire (pour la bande de roulement des pneus), de deux caméras CCD couleur (pour les flans des pneus) et de deux LEDS, le tout associé à un éclairage par l'arrière. Un axe vertical, permet de bien placer le pneu à inspecter par rapport à la caméra.

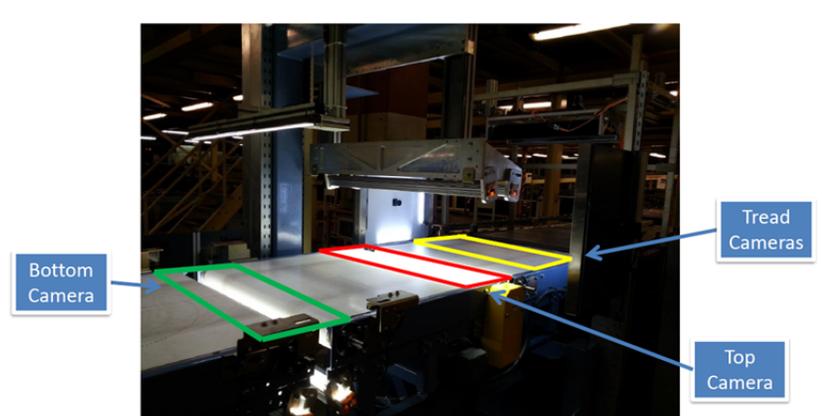


Figure 2.4 – Système ESOTIREINSP d'Enermeter, partie matérielle

Au niveau algorithmique, ce système se divise en deux parties, une pour l'inspection de la bande de roulement et l'autre pour les flans. La partie concernant l'inspection de la bande de roulement est essentiellement axée sur la détection, le positionnement et les mesures de dimensions des rainures des pneus. Et celle concernant l'inspection des flans cherche à identifier les lettres, les points et le code barre sur le pneu.

2.3.6 Inspection dans l'industrie aéronautique

En aéronautique l'inspection joue un rôle fondamental. Elle est employée afin de garantir aussi bien le contrôle et la qualité des équipements lors de la phase de fabrication comme l'inspection de l'état des appareils entre deux vols successifs [Harvent, 2010, Jovancevic et al., 2015]. On comprend de ce fait l'impact qu'elle peut avoir dans ce secteur. Pour mieux souligner cette importance, nous pouvons citer le projet AirCOBOT, un projet impliquant plusieurs acteurs socio-économiques et académiques de la région Midi-Pyrénées, autour d'un robot collaboratif destiné à effectuer des tâches d'inspection d'aéronefs au sol, entre deux vols.

Dans le cadre de ce projet, on peut citer les travaux relatifs à l'inspection de la partie extérieure d'un avion présenté par Jovancevic et al. dans [Jovancevic et al., 2015]. Plusieurs techniques différentes d'analyse d'images sont appliquées afin de traiter différents types de défauts. Cette question d'inspection visuelle automatisée est très importante; toute l'activité professionnelle de la société Doneclé par exemple, repose sur l'inspection visuelle

automatisée d'avions entre deux vols, à l'aide de drones. Dans ce domaine, on constate à la fois des approches classiques, c'est-à-dire, n'utilisant pas de modèle CAO et de plus en plus de volonté d'utiliser le modèle CAO. Dans la section 2.4 nous dresserons une liste (non exhaustive) des méthodes d'inspection utilisant le modèle CAO.

2.3.7 Autres

L'inspection visuelle automatisée ne se limite pas aux seuls domaines que nous venons de décrire. Elle est présente dans d'autres domaines. Hittawe et al. dans [Hittawe et al., 2015] décrivent une méthode de vérification de la qualité des planches de bois basée sur l'exploitation d'informations obtenues grâce à l'imagerie par rayons-X. Grâce à l'analyse des images obtenues par rayons-X, les auteurs calculent des paramètres physico-mécaniques (KDR, MOE and MOR) du bois et déterminent ainsi sa qualité. Dans le chapitre 10 du livre [Dawson-Howe, 2014] intitulé "Vision Problems", l'auteur évoque une série d'applications possibles d'inspection. Celles-ci touchent plusieurs domaines et incluent notamment la vérification de la nourriture de bébé pendant le processus industriel, vérification d'étiquettes sur flacons de colle, vérification d'état de gomme, etc. Cependant ces applications ne sont pas décrites par l'auteur. Dans [Davies, 2012] l'auteur décrit plusieurs applications d'inspection dans divers domaines, notamment l'inspection de circuit électronique intégré, inspection de bois, légumes, aciers, etc.

2.4 Inspection de pièces mécaniques à l'aide d'un modèle CAO

De nos jours, le modèle CAO est de plus en plus exploité dans le cadre d'applications d'inspection visuelle [Newman and Jain, 1995, Bourgeois, 2006, Boutarfa et al., 2006, Far, 2005, Fishkel et al., 2006, Karabagli, 2013].

[Chawla and Deb, 2012] présentent une méthode d'inspection de pièces mécaniques, où ils exploitent trois vues d'un même objet, pour y extraire des attributs géométriques qu'ils mettent en relation afin d'obtenir des primitives qu'ils appartiennent par la suite. Les primitives appariées sont décrites par une matrice qui permet de classer les pièces en différentes familles. Mais l'appariement est réalisé au sein des différentes vues utilisées. Bien que dans son titre il soit écrit "A Computer-Aided Inspection Methodology for Mechanical Parts based on Machine Vision", l'approche n'utilise pas un modèle CAO tel que nous l'entendons, c'est-à-dire conçu avec un logiciel de modélisation numérique en 3D. Ici, ce qu'on pourrait qualifier de modèle est le fait d'utiliser plusieurs vues. Les auteurs parlent d'ailleurs d'information 2.5 D.

Dans [Lucas, 2010] Yves Lucas décrit des applications de manipulation et inspection d'objet grâce à l'exploitation du modèle CAO. La première application concerne la dépalettisation robotisée d'objets. Elle est guidée par le modèle CAO, développée par la société danoise Scape Technologies. Dans cette application, le modèle CAO permet la génération d'images virtuelles et de gérer l'ordre de saisie des objets par le robot, dans une scène complexe. Une autre application, celle de la société Visio Nerf utilise le modèle CAO qui compare le modèle CAO de la pièce à un nuage 3D obtenu par vision active, à l'instar de ce qui est fait dans l'application de Noomeo (à l'origine de nos travaux décrits dans le **chapitre 3**) apportant une aide à l'ordonnancement de la tâche de manipulation d'objets.

La dernière de ces trois applications est celle de la société Sensor Desk qui utilise la CAO dans une tâche d'inspection. Le modèle CAO permet dans cet exemple de simuler la scène virtuelle (caméra, lumière, choix de point de vue) et de réaliser des traitements dans la scène virtuelle pour se rendre compte de leur efficacité. En résumé, on simule l'inspection dans un monde virtuel avant de la réaliser en pratique.

Au début de cette section, nous évoquons le nombre croissant d'approches d'inspection automatisée utilisant un modèle CAO comme référence. Ci-dessous nous dressons une liste non exhaustive de ces méthodes. Les méthodes d'inspection automatisée par vision artificielle utilisant un modèle CAO se décomposent en deux grandes familles :

- ◇ Méthodes basées sur une comparaison 3D/3D. Nous les appellerons méthodes "3D pure". Ces méthodes comparent un objet 3D réel à son modèle CAO 3D.
- ◇ Les méthodes basées sur une comparaison 2D/2D issue du modèle 3D. Ces méthodes reposent sur la comparaison d'une image réelle à une image 2D issue du modèle CAO. En ce sens, elles exploitent les *a priori* 3D du modèle, mais effectuent les calculs en deux dimensions.

2.4.1 Méthodes basées sur une comparaison 3D/3D

En comparaison avec les méthodes 2D, on peut dire que les méthodes purement 3D se sont beaucoup développées et cela de façon naturelle, car elles permettent une comparaison directe du modèle CAO en 3D avec la forme 3D de l'objet réel à évaluer. Beaucoup de travaux exploitent notamment les résultats des travaux de Besl et McKay [Besl and McKay, 1992] sur le recalage 3D. Malgré les années, cette approche continue d'être utilisée. [Berndt, 2011] propose une approche de comparaison 3D/3D exploitant le recalage global. Dans [Hong-Seok and Mani, 2014] une approche d'inspection de pièces usinées est présentée. Elle se base sur un recalage de type ICP (modifié) suivi de comparaison de caractéristiques locales (normales à la surface, coins).

D'autres méthodes exploitent plutôt des caractéristiques plus locales ou exploitent conjointement les caractéristiques globales et locales. En ce sens, on peut citer les travaux de Prieto et al., [Prieto et al., 2002], qui effectuent d'abord un recalage global puis s'intéressent à des éléments de surface plus locaux afin de déterminer si l'élément inspecté respecte les tolérances indiquées dans le modèle CAO. Boutarfa et al., dans [Boutarfa et al., 2006], ont utilisé à la fois le recalage global et la comparaison d'attributs géométriques tels que la normale à la surface, les courbures gaussiennes et moyennes afin de comparer l'objet réel avec son modèle CAO de référence.

Dans [Fishkel et al., 2006] les auteurs présentent une approche élégante de contrôle automatisé en comparant les caractéristiques locales de la pièce à évaluer avec celles du modèle CAO de référence. Cette approche est élégante en ce sens qu'après la segmentation des formes 3D en plusieurs régions, les caractéristiques locales de chaque région sont extraites (courbure, taille, etc). Ensuite les auteurs transforment le problème d'inspection en un problème de recherche opérationnelle en construisant deux graphes attribués qui sont appariés en utilisant un graphe biparti et à la fin du processus, un ensemble des règles injectées dans un arbre de décision permet de prendre la décision sur la qualité de l'objet à évaluer.

2.4.2 Méthodes basées sur une comparaison 2D/2D issue du modèle 3D

Parmi ces méthodes, on peut citer les travaux de Newman et Jain au milieu des années 90 [Newman and Jain, 1994]. Dans ces travaux, les auteurs ont proposé un système de contrôle basé sur la CAO, en utilisant les images de profondeur. L'apport de l'utilisation des images de profondeur et l'utilité du contrôle automatisé y sont clairement décrits. L'objectif de leur système consistait en l'inspection de défauts dans les surfaces de moulage ainsi qu'en l'inspection des tolérances dimensionnelles. Tout d'abord, ils ont segmenté la surface de l'élément à contrôler, en faisant une classification des surfaces et de certaines caractéristiques. Ils ont ensuite fait une mise en correspondance des résultats de cette classification avec des modèles présents dans une base des données, pour détecter la configuration de l'élément, la présence d'un artefact ou des parties manquantes. Les travaux de Newman et Jain font partie des travaux pionniers de contrôle automatisé avec le modèle CAO, bénéficiant de l'augmentation croissante de la puissance des ordinateurs. Des travaux plus récents ont aussi employé des stratégies similaires.

Far, dans [Far, 2005], exploite les *a priori* 3D au niveau des images 2D dans une application d'alignement d'objets. Dans [Viana et al., 2012], nous générons des images 2D à partir du modèle CAO pour établir une vérification de contrôle d'assemblages mécaniques au niveau 2D, en comparant les contours d'un objet extrait d'une image issue du modèle CAO au contour d'un objet extrait d'une image réelle. L'approche d'inspection de [Berndt,

2011], exploitant la de réalité augmentée [Zendjebil et al., 2008] suit la même stratégie. De même, Karabagli [Bilal et al., 2015], dans une application d’usinage à grande vitesse, emploie ce type de stratégie consistant à comparer des éléments extraits d’une image réelle à des éléments extraits d’une image synthétique générée à partir du modèle CAO. Il s’assure qu’il n’a pas de collision entre les éléments du montage (brut à usiner, tiges de fixation, pions de positionnement) et l’outil d’usinage. Les attributs utilisés dans sa comparaison sont d’ordre topologique (squelette) et dimensionnel (contour et surface)

2.5 Conclusion

Dans ce chapitre, nous avons défini l’inspection automatisée, tout en faisant le parallèle entre l’inspection et la reconnaissance d’objets, deux sujets parfois semblables mais différents. En effet, la reconnaissance est souvent en amont de l’inspection. De ce fait, les méthodes d’inspection peuvent s’appuyer sur les méthodes de reconnaissance d’objet, tout en les complétant afin d’y apporter la notion du respect de spécifications ou de tolérances qui caractérisent l’inspection. Dans ce chapitre nous avons aussi montré l’importance de l’inspection, présente dans des secteurs d’activité très variés, ce qui fait la diversité des méthodes d’inspection et accentue la nécessité de s’adapter, notamment au niveau des systèmes d’acquisition choisis pour la tâche d’inspection visuelle automatisée. Nous avons proposé une classification des méthodes d’inspection en méthodes traditionnelles, i.e sans utilisation d’un modèle CAO, et avec utilisation du modèle CAO. Cette classification permet clairement de mettre en valeur l’importance croissante de l’inspection ainsi que l’utilité du modèle CAO, notamment dans l’industrie aéronautique.

3

Application ayant servi de support à nos travaux

Ce chapitre décrit la méthode d'inspection réalisée dans le cadre du projet CAAMVis. L'objectif général étant de combiner l'utilisation d'un modèle CAO avec un capteur à double tête stéréo afin de réaliser l'inspection d'un assemblage mécanique aéronautique. Nous commençons le chapitre en rappelant les objectifs du projet et en expliquant comment le modèle CAO est utilisé dans ce projet, cf. section 3.1. Ensuite, dans la section 3.2, nous introduirons la méthode d'inspection développée dans le cadre du projet Lynx © qui utilise une seule caméra. La méthode sera par la suite détaillée dans le **chapitre 4**.

Sommaire

3.1	Projet CAAMVis : inspection à partir d'un capteur stéréo porté par un robot	44
3.1.1	Rappel des objectifs et stratégie d'inspection	44
3.1.2	Rappel de la méthode développée par la société Noomeo, comparaison 3D /3D	45
3.1.3	Inspection basée sur la comparaison 2D/2D via la transformée en distance	46
3.1.4	Résultats préliminaires et discussions	52
3.1.5	Bilan du projet CAAMVis : améliorations envisagées	54
3.2	Introduction au projet Lynx et à l'inspection à partir d'une caméra PTZ .	56
3.2.1	Rappel des objectifs	56
3.2.2	Les deux stratégies d'utilisation du modèle CAO dans le projet Lynx©	57

3.1 Projet CAAMVis : inspection à partir d'un capteur stéréo porté par un robot

3.1.1 Rappel des objectifs et stratégie d'inspection

3.1.1.1 Rappel des objectifs

Les objectifs recherchés dans cette application sont les suivants :

- ◇ *décider de la présence/absence de l'élément à contrôler*
- ◇ *vérifier le positionnement lorsque qu'un élément à inspecter est présent*

- ◇ *vérifier son exactitude géométrique par rapport à l'élément idéal de référence décrit dans le modèle CAO*

3.1.1.2 Stratégie d'inspection

Afin de contrôler un élément, il faut d'abord le sélectionner dans la maquette numérique. Les trajectoires menant à la position de cet élément ainsi que les points d'acquisition correspondants sont calculés hors ligne. Le capteur stéréo utilisé est calibré en amont, de façon précise, par une procédure hors ligne. Le système étant calibré, la transformation entre les deux caméras de chaque paire stéréo a pu être déterminée pendant la phase de calibrage. Connaissant le modèle géométrique direct du robot, la position du capteur peut être estimée pour chaque point de vue souhaité. En résumé, l'étape de localisation consiste à appliquer au modèle CAO une première fois la pose grand champ pour que la structure réelle et la structure virtuelle des assemblages soient alignées et ensuite appliquer la pose petit champ à l'élément CAO d'intérêt, la même pose petit champ ayant servi à faire l'acquisition de ce même élément avec le capteur physique. À ce stade, on effectue l'acquisition des données nécessaires à l'inspection à l'aide du capteur de stéréo-vision. Ces données sont ensuite utilisées pour mettre en place une méthode d'inspection qui permet de comparer un élément du modèle CAO avec un élément réel, soit en termes de comparaison 3D/3D directement (cf. section 3.1.2), soit en comparant l'élément issu du modèle CAO avec un élément 2D d'une image réelle (cf. section 3.1.3). Dans ce projet, le travail de calibrage et d'estimation de pose a été réalisé par la société Noomeo. Nous contribuons à ce projet en développant la méthode d'inspection qui combine l'utilisation du modèle CAO aux techniques d'analyse d'images. Nous le verrons plus tard, le but de notre contribution est de réduire le temps de calcul constaté lors du développement de la méthode reposant sur la comparaison directe 3D/3D développée par la société Noomeo et que nous présentons dans la section 3.1.2 ci-dessous.

3.1.2 Rappel de la méthode développée par la société Noomeo, comparaison 3D /3D

Initialement, la société Nooméo qui porte le projet CAAMVis (cf. **chapitre 1**) a développé une méthode d'inspection automatisée basée sur la comparaison directe d'un élément du modèle 3D avec un élément d'un nuage de points obtenu par reconstruction, en utilisant les deux images de la caméra petit champ du capteur à double tête stéréo que nous avons présenté dans la section 1.2.1. Les étapes majeures de la méthode mise au point par la société Noomeo sont les suivantes :

D'abord, pour le point de vue souhaité, deux images sont prises avec les deux caméras petit champ du capteur stéréo. Ces deux images permettent d'effectuer la reconstruction 3D dense de l'objet d'intérêt. Ensuite, le nuage de points 3D de l'objet à évaluer est comparé

au nuage de points théorique issu de la CAO via une procédure de recalage de type ICP [Besl and McKay, 1992].

Pour finir, un seuillage est réalisé sur l'erreur de recalage afin de prendre une décision quant à la conformité de l'élément à tester.

En moyenne, l'inspection d'un élément (cf. figure 3.1b) par cette méthode dure près de 10 secondes en utilisant un ordinateur équipé d'un Processeur Intel Xeon quad core, de 3 GHz et de 12 Go de RAM.

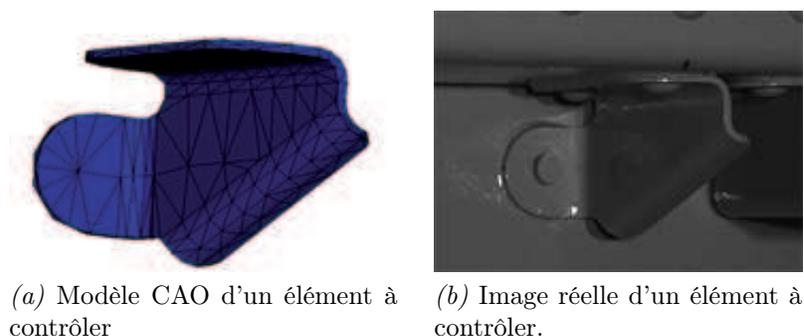


Figure 3.1 – A gauche : modèle CAO d'un élément à contrôler, à droite : image de luminance de l'élément réel

3.1.2.1 Constat : Le problème du temps de calcul

La méthode développée par la société Noomeo a été éprouvée de façon industrielle et elle est fiable et robuste. Cependant, pour certaines applications industrielles ce temps de calcul de 10 secondes par élément n'est pas satisfaisant. Par exemple, contrôler 1000 éléments prend plus de 2h30. L'approche 3D nécessite une étape de reconstruction. Cela ralentit le système et le rend sensible à la qualité de la reconstruction. Ces raisons ont conduit à la recherche d'une autre méthode d'inspection automatisée, qui puisse être complémentaire à l'approche 3D pure, tout en étant aussi fiable que cette méthode 3D mais plus rapide.

3.1.3 Inspection basée sur la comparaison 2D/2D via la transformation en distance

Afin d'accélérer l'inspection, nous proposons une autre approche d'inspection, qui se base sur l'utilisation des techniques d'analyse d'images 2D. L'hypothèse consiste à dire que les traitements à réaliser pour l'inspection prendront moins de temps si on exploite les images 2D qu'en utilisant une stratégie 3D pure.

Afin de répondre aux objectifs fixés, de vérifier la présence/absence d'un élément que l'on souhaite contrôler, la conformité de son positionnement et son exactitude géométrique par rapport à un élément idéal décrit dans un modèle CAO, l'approche que nous présentons se base sur une mise en correspondance entre les données 3D du modèle CAO (cf. figure

3.1a) projetées dans une image 2D (cf. figure 3.3) avec l'image 2D de l'élément réel (cf. figure 3.1b), en utilisant des contours (cf. figures 3.4a, 3.4b). Cela permet d'éviter l'étape de reconstruction 3D de l'élément réel, économisant ainsi le temps de calcul.

Nous avons choisi d'utiliser les contours pour représenter les deux jeux de données (théoriques et réelles) utilisés. En effet, puisque nous utilisons un modèle CAO et une image réelle, les seules informations communes aux données sont de nature géométrique. Notre méthode cherche à optimiser la position et l'orientation du contour 2D représentant l'objet de référence par rapport au contour 2D représentant l'élément réel, via l'utilisation d'une carte de distances pour accélérer les calculs.

Notre méthode 2D /2D peut être résumée par le schéma de la figure 3.2.

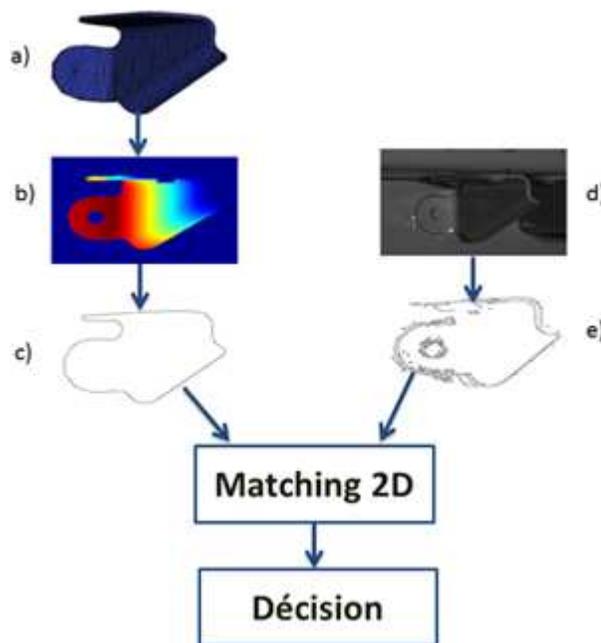


Figure 3.2 – Schéma de la méthode proposée. a) modèle CAO de l'élément, b) image de profondeur issue du modèle CAO, c) contour 2D issu de l'image de profondeur, d) image en niveau de gris de l'élément réel, e) contour extrait de l'image réelle.

Cette méthode opère en 4 étapes faisant l'objet des 4 sections suivantes :

- ◇ La section 3.1.3.1 explique comment obtenir un contour 2D à partir d'un modèle CAO en 3D ;
- ◇ La section 3.1.3.2 décrit la fonction de mesure qui sera employée lors du processus d'optimisation ;
- ◇ La section 3.1.3.3 traite de la méthode d'optimisation utilisée ;
- ◇ La section 3.1.3.4 tranche sur la convergence de l'algorithme d'optimisation et permet de classer l'objet contrôlé en défectueux ou non défectueux.

3.1.3.1 Obtenir des contours 2D à partir d'un modèle 3D

Afin de récupérer des contours 2D à partir du modèle CAO de l'élément à contrôler, nous commençons par effectuer une rasterisation, conversion des données vectorielles, ici des triangles, en données matricielles, ici une image de profondeur. S'en suit une étape d'extraction de contours. L'algorithme de rasterisation utilise les paramètres de calibrage de la caméra, aussi bien les paramètres intrinsèques qu'extrinsèques et renvoie l'image de profondeur correspondante. Les paramètres de calibrage de la caméra sont requis lors de cette étape parce qu'ils permettent de calculer les matrices de projection et de vue (cela inclut les caractéristiques du capteur ainsi que la position de l'élément observé). L'image de profondeur est extraite directement de la carte graphique, du buffer de profondeur connu sous le nom de Z-buffer.

L'utilisation des données 2D est un point clé de cette approche, notamment en termes d'économie en temps de calcul. Plus l'élément réel se rapproche des spécifications définies par le modèle théorique, plus l'image rasterisée sera proche de l'image de l'élément réel.

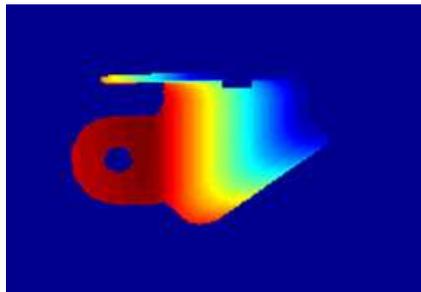
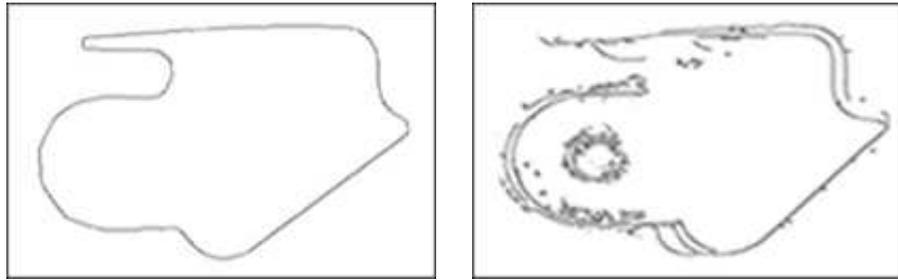


Figure 3.3 – Image de profondeur obtenue par rasterisation de la CAO.

Dès lors que les deux images représentant le modèle 3D de base et l'élément réel sont disponibles, une extraction de contours est effectuée, en utilisant le détecteur de contours de Canny. La technique de détection de contours de Canny utilise le seuillage et est adaptée car elle offre la possibilité d'ajuster les paramètres de façon à éviter éventuellement le bruit et/ou les faux contours.



(a) Contours extraits de l'image de profondeur

(b) Contours extraits de l'image réelle

Figure 3.4 – Contours extraits de l'image synthétique et de l'image réelle

3.1.3.2 Construction de la fonction de mesure basée sur l'utilisation d'une carte de distances

Afin de comparer le contour 2D de l'élément réel et le contour 2D construit à partir du modèle CAO, une fonction de mesure est construite en deux étapes. La première phase consiste à effectuer une transformée en distance [Rosenfeld and Pfaltz, 1968] d'une image de contours représentant l'élément réel que l'on souhaite inspecter. La transformée en distance (cf. figure 3.5) est une représentation d'une image numérique dans laquelle les pixels sont étiquetés avec la distance au plus proche obstacle, souvent la frontière ou un contour. La métrique [Moore, 2002, Liwei Wang et al., 2005] utilisée pour le calcul de la transformée en distance est la distance euclidienne. L'avantage de la transformée en distance réside dans le fait qu'elle peut être calculée préalablement à l'étape d'optimisation puis utilisée à chaque itération. Cela permet d'éviter un processus d'évaluation de la fonction objectif coûteux en temps de calcul.

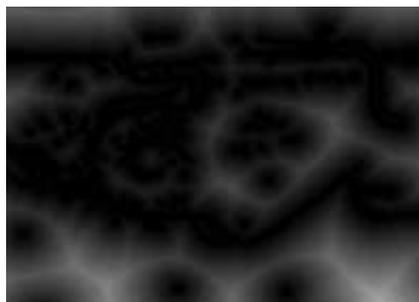


Figure 3.5 – Image de transformée en distance des contours d'un élément réel.

La deuxième phase consiste à utiliser la transformée en distance pour construire la fonction de mesure (un score) qui sera optimisée. Le score à minimiser est optimal lorsque l'élément réel et l'élément théorique sont superposés. Le score est la somme des pixels du contour dans l'image de la transformée en distance de l'élément réel, pondéré par le nombre de points appartenant au contour théorique.

Mathématiquement il est défini par :

$$S(\mathbf{R}(\theta), \mathbf{t}) = \frac{1}{N_T} \sum_{p \in f(C_T, \mathbf{R}(\theta), \mathbf{t})} I_{DT}(C_R, p) \quad (3.1)$$

où

- ◇ C_T désigne l'ensemble des pixels du contour théorique issu du modèle CAO et C_R l'ensemble des pixels du contour réel.
- ◇ p désigne un pixel du contour du modèle CAO.
- ◇ $\mathbf{R}(\theta)$ est une matrice de rotation [Shapiro and Stockman, 2001, Lingrand, 2010] construite à partir de l'angle θ appliqué au contour théorique C_T .
- ◇ $\mathbf{t} = (t_x, t_y)$ est un vecteur de translation [Shapiro and Stockman, 2001, Lingrand, 2010] appliqué au contour théorique C_T .
- ◇ N_T désigne le nombre de pixels du contour théorique C_T .
- ◇ $I_{DT}(C_R, p)$ est la valeur du pixel p par rapport à la transformée en distance du contour réel C_R .

Des expériences ont été réalisées et ont confirmé qu'à l'optimum $(\mathbf{R}^*, \mathbf{t}^*)$ les deux contours coïncident. De plus, ces expériences ont montré que cette fonction objectif est lisse et donc particulièrement adaptée à un algorithme de minimisation par descente de gradient. La mise en œuvre de ce dernier est présentée dans la section suivante.

3.1.3.3 Algorithme d'optimisation utilisé

La méthode d'optimisation utilisée dans ce travail est la descente de gradient. C'est une méthode d'optimisation du premier ordre, utilisée pour trouver un candidat au minimum local d'une fonction. Nous effectuons une optimisation paramétrique, c'est-à-dire que nous utilisons cet algorithme d'abord sur la variable \mathbf{t} pour une rotation nulle. Nous obtenons ainsi une translation optimale \mathbf{t}^* . Puis nous optimisons la fonction $\theta \rightarrow S(\mathbf{t}^*, \theta)$ et obtenons finalement un candidat (\mathbf{t}^*, θ^*) à l'optimalité. Pour ces deux optimisations successives, compte tenu du fait que les dérivées partielles de la fonction objectif ne peuvent être décrites analytiquement, elles ont été estimées en utilisant les différences finies centrées. Par exemple, les dérivées partielles de S selon les variables t_x et t_y sont données par :

$$\frac{\partial S(\theta, t_x, t_y)}{\partial t_x} = \frac{S(\theta, t_x + 1, t_y) - S(\theta, t_x - 1, t_y)}{2} \quad (3.2)$$

et

$$\frac{\partial S(\theta, t_x, t_y)}{\partial t_y} = \frac{S(\theta, t_x, t_y + 1) - S(\theta, t_x, t_y - 1)}{2} \quad (3.3)$$

Ainsi, les itérés successifs, selon les variables t_x et t_y , de l'algorithme de descente de gradient sont donnés par :

$$t_x^{i+1} = -K_x^i G_{t_x}^i + t_x^i \quad (3.4)$$

et

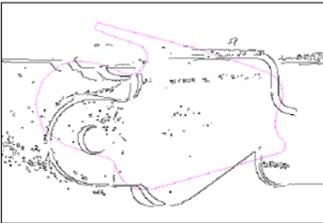
$$t_y^{i+1} = -K_y^i G_{t_y}^i + t_y^i \quad (3.5)$$

où $G_{t_x}^i$ et $G_{t_y}^i$ représentent le vecteur de dérivées partielles selon la variable \mathbf{t} (calculées par différences finies) évaluées aux itérés t_x^i et t_y^i . K_x^i et K_y^i représentent la longueur de pas à l'itéré i dans la direction considérée. L'initialisation se fait en $(t_x, t_y) = (0, 0)$. Les longueurs de pas K_x et K_y ont été initialement fixées à 10.

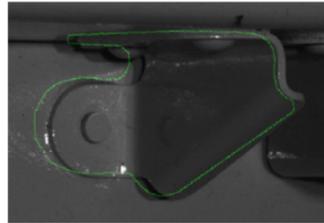
Par analogie, les itérés de la descente de gradient selon la variable θ sont donnés par

$$\theta^{i+1} = -K_{\theta^i} G_{\theta^i} + \theta^i \quad (3.6)$$

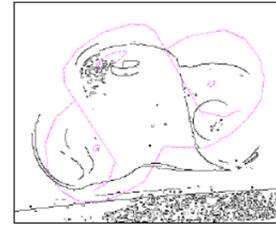
La méthode est initialisée à partir de $\theta = 0$ et le pas est initialement fixé à $\frac{1}{248}^\circ$. Cette dernière valeur a été déterminée par essai-erreur à l'aide de nos différentes expériences. La figure 3.6 montre des exemples de résultats obtenus par cette méthode. À la fin du chapitre nous verrons d'autres exemples.



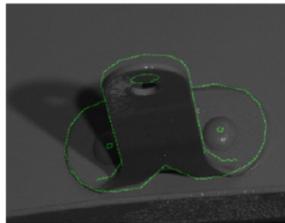
(a) Position initiale d'un élément réel par rapport à un élément de référence



(b) Position finale d'un élément réel par rapport à un élément de référence



(c) Position initiale d'un élément réel par rapport à un élément de référence



(d) Position finale d'un élément réel par rapport à un élément de référence

Figure 3.6 – Exemple de résultat obtenu par notre méthode d'appariement 2D/3D basée sur recalage de contour grâce à la transformée en distance

L'algorithme présenté ici, utilisant la descente de gradient de manière successive sur chacune des variables, est une méthode simple fonctionnant sous certaines hypothèses. Cependant, comme la méthode est appliquée de manière successive, rien n'assure que le minimum obtenu soit un minimum absolu. Ainsi, il pourrait être utile d'avoir recours à une méthode d'optimisation plus robuste car notre approche peut converger vers un minimum indésirable, notamment en cas de mauvaise initialisation.

3.1.3.4 Description de la fonction de décision

Pour pouvoir décider de la conformité ou non d'un élément à évaluer, un critère de décision a été défini. Il permet de statuer sur la convergence de l'algorithme d'optimisation. Ce critère de décision se base sur le taux de recouvrement entre le contour du modèle CAO et le contour de l'élément réel. Il ne tient pas seulement compte du nombre de pixels du modèle CAO qui se superposent aux pixels du contour de l'élément réel mais aussi des contours parasites à l'intérieur du modèle CAO pendant la phase de recouvrement. La décision se fait donc par seuillage du taux de recouvrement entre les contours du modèle CAO et celui de l'élément à contrôler et du taux de superposition des contours parasites de l'élément réel avec le contour du modèle CAO. Le premier taux doit dans le pire des cas être au moins supérieur ou égal à 50 % et le deuxième ne peut, en aucun cas, atteindre les 10 %. Ces valeurs ont été obtenues de manière empirique.

3.1.4 Résultats préliminaires et discussions

La méthode décrite dans ce chapitre a été prototypée en Matlab © puis développée en C++ avec la librairie OpenCV [Laganière, 2011, Baggio et al., 2012, Dawson-Howe, 2014] et testée sur un ordinateur équipé du processeur Intel Xeon quad core, de 3 GHz et de 12 Go de RAM. Ci-dessous nous montrons quelques résultats obtenus avec notre méthode. Avant de montrer nos résultats, nous allons introduire quelques concepts élémentaires dans la classification des tests cf. [Dawson-Howe, 2014] :

- ◇ Vrai Négatif (VN) : l'objet est non défectueux et le test dit qu'il est non défectueux.
- ◇ Faux Négatif (FN) : l'objet est défectueux et le test dit qu'il est non défectueux (situation critique).
- ◇ Vrai Positif (VP) : l'objet est défectueux et le test dit qu'il est défectueux.
- ◇ Faux Positif (FP) : l'objet est non défectueux et le test dit qu'il est défectueux (fausse alarme).

Ces notions sont résumées dans le tableau 3.1 suivant :

Vocabulaire utilisé en analyse de performances		
	Réalité	
Test	Non-défectueux	Défectueux
Non-défectueux	VN	FN
Défectueux	FP	VP

Tableau 3.1 – Vocabulaire utilisé en test

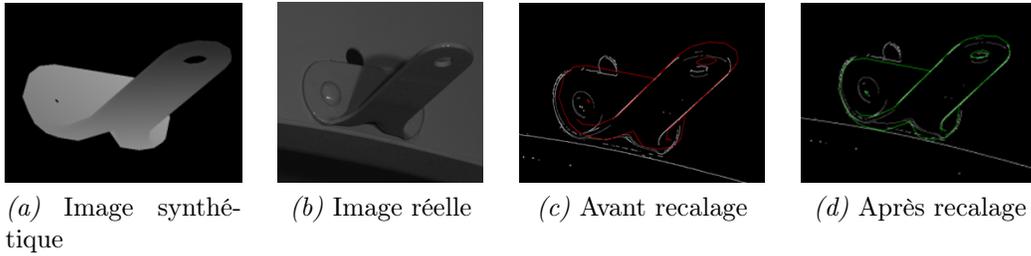


Figure 3.7 – Test de la méthode sur l'objet réel 3.7b

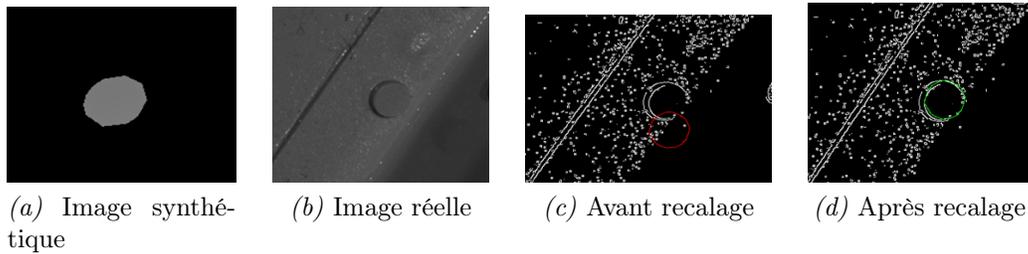


Figure 3.8 – Test de la méthode sur l'objet réel 3.8b

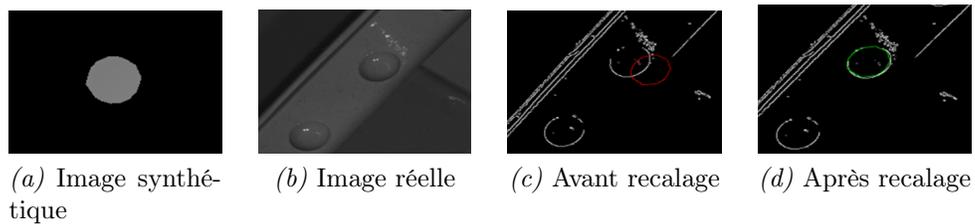


Figure 3.9 – Test de la méthode sur l'objet réel 3.9b

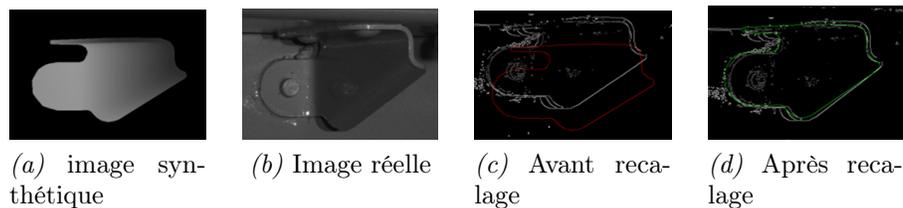


Figure 3.10 – Test de la méthode sur l'objet réel 3.10b

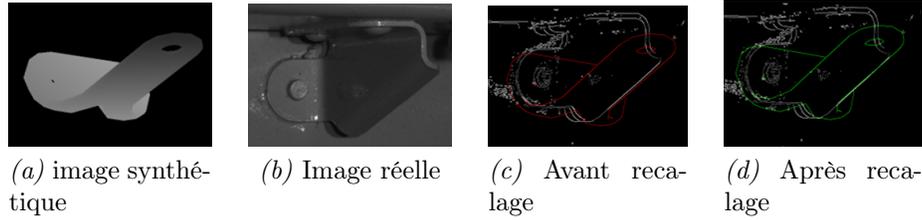


Figure 3.11 – Test de la méthode sur l'objet réel 3.11b

Résultats de la méthode CAAMVis			
	Taux de recouvrement (en %)	Taux de bruit intérieur (en %)	Décision
Objet 3.7	53	2	Conforme
Objet 3.8	85	0	Conforme
Objet 3.9	75	1	Conforme
Objet 3.10	52	3	Conforme
Objet 3.11	30	3	Non Conforme

Tableau 3.2 – Résultats obtenus avec notre la méthode d'inspection décrite dans ce chapitre

En termes de comparaison avec la méthode 3D pure, les tests ont montré que notre méthode 2D/3D exploitant la projection du modèle CAO en 2D et les images acquises par le capteur réel prend en moyenne une seconde pour inspecter un élément alors que la méthode basée sur la 3D pure prend 10 secondes pour inspecter le même élément. En termes de vrais positifs, la méthode proposée est aussi performante que la méthode basée sur la 3D pure, avec l'avantage d'être dix fois plus rapide. Parfois, à cause d'une mauvaise initialisation, la descente de gradient peut rester bloquée dans un minimum local indésirable. En conséquence quelques cas de faux positifs ont été constatés. L'utilisation d'un algorithme d'optimisation plus robuste pourrait résoudre ce problème.

Par ailleurs, la méthode basée 2D ne nécessite pas de reconstruction (de l'élément réel), elle est donc capable de contrôler des pièces que la méthode basée purement sur la 3D serait incapable de contrôler à cause d'un échec lors de la phase de reconstruction. Selon les points des vues, il est possible que certaines images contiennent de la réflexion spéculaire. La flexibilité de la méthode 2D permet d'utiliser l'une des deux images du capteur stéréo alors que l'approche basée sur de la 3D pure échoue inévitablement en cas de reconstruction d'un élément présentant de la réflexion spéculaire.

3.1.5 Bilan du projet CAAMVis : améliorations envisagées

La méthode décrite dans la section 3.1 est prometteuse mais présente quelques points faibles. On constate que le résidu de l'optimisation n'est pas utilisé. Nous pensons que ce résidu pourrait être utilisé comme coefficient de pondération dans la fonction de décision.

L'utilisation d'une métrique plus adaptée pourrait rendre la méthode plus robuste.

Un autre point à améliorer est le recalage par descente de gradient. Ce processus d'optimisation est décomposé deux parties, l'optimisation en translation précède l'optimisation en rotation. Or, si en pratique cela fonctionne plutôt bien, réaliser l'optimisation de cette manière n'est pas une approche classique. Conventionnellement, on applique la rotation et la translation en un seul mouvement.

En outre, il serait souhaitable de réaliser l'optimisation directement en 3D, en exploitant les 6 degrés de liberté de l'espace avant de projeter le modèle sur le plan 2D pour effectuer la comparaison en 2D. D'autres approches de comparaison de formes telles que l'exploitation de la théorie des moments invariants, histogrammes de gradients orientés, exploitation de la topologie par le squelette pourraient être envisagées.

Par ailleurs, nous avons fortement envisagé la possibilité de mettre en œuvre une méthode d'inspection basée à la fois sur l'analyse topologique et dimensionnelle, afin de retrouver la structure de la scène comportant l'élément à évaluer ainsi que les dimensions des éléments compris dans la scène. La topologie pourrait être obtenue par utilisation de graphes en exploitant notamment le concept de polymorphisme de graphe et les dimensions seraient quant à elles récupérées par des descripteurs tels que l'aire, la longueur, le nombre de pixels appartenant à une composante connexe, la circularité, l'excentricité, etc. Concernant le travail sur la topologie de la scène nous avons travaillé sur une procédure qui permettait de colorier (avec une couleur différente) chaque catégorie d'objet (support, rivet, etc) présent dans le modèle CAO lors de sa projection sur le plan image, (cf. figure 3.12).

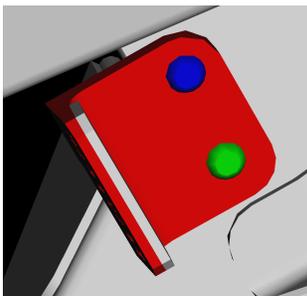


Figure 3.12 – Exemple de projection d'un support contenant trois classes d'objets. L'objet à contrôler en bleu, le support en rouge, un objet de même type que l'objet à contrôler en vert.

Le but était d'analyser l'image issue du modèle CAO pour trouver une correspondance avec la scène réelle. Le coloriage faciliterait cette analyse scénique dans le sens où les étiquettes (classe et couleur de chaque objet) seraient connues avant la segmentation. Cela rendrait l'étape de segmentation de l'image synthétique plus facile à réaliser et pourrait

même guider l'étape d'extraction d'indices visuels sur l'image réelle.

Néanmoins, les pistes de recherche que nous évoquons ici n'ont pas pu être explorées pleinement. En effet, la cessation d'activités de la société Noomeo a compromis ces directions de recherche car elle mettait fin à l'accès aux données réelles qui pouvaient être utilisées pour la mise en place des améliorations envisagées. Par la suite, nous avons commencé une collaboration avec la société G2Metric avec un but similaire qui est celui de réaliser le contrôle automatisé d'assemblages aéronautiques en exploitant au mieux le modèle CAO mais avec une méthodologie différente. Cela nous a alors conduit à réfléchir sur une méthode que nous détaillons dans le **chapitre 4**.

Avant de décrire notre deuxième méthode d'inspection réalisée dans le cadre du projet Lynx© (cf. section 1.2.2), nous allons revenir sur la problématique d'utilisation de la CAO.

3.2 Introduction au projet Lynx et à l'inspection à partir d'une caméra PTZ

Avec la cessation d'activités de la société Noomeo, nous avons entamé une nouvelle collaboration avec la société G2Metric. Dans le cadre de cette collaboration nous avons développé une nouvelle méthode d'inspection que nous décrivons dans le **chapitre 4**. Avant de décrire cette méthode d'inspection, il nous semble opportun de rappeler maintenant les objectifs de cette collaboration ainsi que la problématique concernant l'utilisation du modèle CAO et d'expliquer comment nous avons traité cette problématique pour ensuite ne nous concentrer que sur la méthode lors du **chapitre 4**.

3.2.1 Rappel des objectifs

La version initiale de Lynx© se base sur la comparaison d'une image réelle contenant l'objet test à une image de référence de ce même objet contenue dans une banque d'images de référence, construite lors d'une phase amont de collecte d'images de référence.

L'enjeu dans le projet Lynx© consiste à remplacer la base d'images de référence par le modèle CAO pour avoir plus de souplesse et de fiabilité au niveau de contrôle. En effet, lors de la construction d'une banque d'images de référence il se pose aussi la question de savoir "qu'est-ce qu'une bonne image de référence?", car l'image de référence peut elle-même changer selon les conditions d'acquisition. L'utilisation du modèle CAO peut aider à s'affranchir de ce problème, en apportant notamment une image de référence plus stable et indépendante des conditions ambiantes. À l'instar des objectifs recherchés dans le projet CAAMVis, dans le projet Lynx© on s'intéresse à savoir si un élément est présent ou

absent ou si son placement est correct ou encore s'il y a les bonnes dimensions. Un autre objectif de Lynx© (en tant que produit) est de vérifier la conformité en termes de couleurs d'un objet. Néanmoins, cet objectif ne rentre pas dans le cadre de notre collaboration.

La question que l'on se pose dès le départ est comment exploiter au mieux le modèle CAO pour faire l'inspection. Dans ce chapitre nous avons vu comment procéder dans le cadre du projet CAAMVis. Dans la section suivante nous allons aborder les deux méthodes envisagées pour obtenir les données synthétiques et réelles servant à l'inspection présentée dans le **chapitre 4**. Les détails de ces deux façons différentes d'exploiter le modèle CAO afin d'obtenir les données utilisées lors de la comparaison entre l'assemblage réel et le modèle sont dans l'annexe D.

3.2.2 Les deux stratégies d'utilisation du modèle CAO dans le projet Lynx©

Dans le contexte du projet Lynx© nous avons envisagé deux stratégies différentes (déjà évoquées dans 1.2.2.3) pour obtenir les données utilisées par la méthode de comparaison entre données du modèle et données réelles. Ces deux stratégies sont les suivantes :

- ◇ 1. Positionner d'abord la caméra virtuelle dans la maquette numérique puis positionner la caméra réelle dans l'environnement réel.

Dans cette stratégie l'inspection est préparée hors-ligne au niveau de la maquette numérique, en positionnant une caméra virtuelle qui observe chacun des éléments à inspecter. La gamme d'inspection est ensuite réalisée en orientant la caméra réelle vers chacun des éléments à inspecter (on rejoue dans l'environnement réel la séquence d'inspection préparée en virtuel).

- ◇ 2. Positionner d'abord la caméra réelle dans l'environnement réel, la calibrer et estimer sa pose et réutiliser ces éléments pour projeter le modèle CAO sur le plan image. Dans cette stratégie on utilise donc des données 2D du modèle CAO projeté.

Comme nous l'avons dit précédemment, l'annexe D contient plus de détails sur la façon dont ces deux stratégies ont été mises en œuvre. Dorénavant, seront employés, de façon indifférenciée, les termes image théorique, ou image de synthèse générée à partir du modèle CAO de l'objet à contrôler afin de parler des données synthétiques que nous utilisons. La méthode d'inspection i.e. l'algorithme permettant de comparer une image réelle et une image synthétique générée à partir du modèle CAO sera présentée dans le **chapitre 4**.

4

Appariement de primitives géométriques par la théorie des graphes et mesure de ressemblance

Ce chapitre traite de la mise en correspondance entre une image réelle acquise par une caméra et une image synthétique générée à partir du modèle CAO.

Sommaire

4.1	Concepts de base en théorie des graphes	59
4.1.1	Graphe et notion de degré	59
4.1.2	Incidence et adjacence, prédécesseurs et successeurs	60
4.1.3	Isomorphisme de graphe	62
4.1.4	Graphes et sous-graphes	63
4.1.5	Bipartisme de graphe	64
4.1.6	Graphe attribué relationnel (ARG en anglais pour Attributed Relational Graph)	65
4.1.7	Graphes et le problème de voisinage	65
4.1.8	Graphe de Voisinage Relatif RNG en anglais pour Relative Neighbourhood Graph	65
4.1.9	Le graphe de Gabriel	66
4.1.10	Autres graphes de voisinage	66
4.1.11	Mise en correspondance de deux graphes	66
4.2	Schéma fonctionnel de la méthode	67
4.3	Développement de la méthode d'appariement de primitives en exploitant des images de synthèse	69
4.3.1	Segmentation des sept images de la figure 4.7	70
4.3.2	Recalage d'images	73
4.3.3	Extraction de primitives géométriques	75
4.3.4	Appariement des primitives par combinaison de graphes - fonction de ressemblance	81
4.3.5	Recherche du meilleur couple d'appariement à l'aide d'un graphe biparti	87

4.4	Prise de décision	90
4.5	Conclusion	91

4.1 Concepts de base en théorie des graphes

Afin de mieux aborder l'appariement de graphes, réalisé dans ce chapitre, il est indispensable de poser quelques concepts basiques de la théorie des graphes pour mieux comprendre notre démarche, notamment les raisons qui nous ont amenées à proposer la méthode développée dans ce chapitre plutôt qu'une recherche classique d'appariement de sous-graphes communs. Ces concepts de base nous permettront aussi de rappeler quelques termes utilisés en théorie des graphes.

4.1.1 Graphe et notion de degré

Soit un graphe $G = (V, E)$, avec V et E l'ensemble des nœuds et d'arêtes du graphe G , l'ensemble $N_G(v) = \{u \in V(G) | uv \in E\}$ désigne le voisinage du nœud v . Le degré du nœud v est $|N_G(v)|$ et peut s'écrire $d_G(v)$. Si $d_G(v) = 0$ alors le nœud v est dit isolé. Dans un graphe orienté, le degré entrant, $d_G^-(v)$ d'un nœud v est donné par le nombre d'arêtes ayant une flèche vers ce nœud v . Le degré sortant $d_G^+(v)$ d'un nœud v est donné par le nombre d'arêtes sortant de ce nœud v , [Panda and Pradhan, 2013]. Le concept de degré dans la théorie des graphes est un concept assez proche du concept d'incidence des nœuds, cf. section 4.1.2. D'ailleurs, uniquement les arêtes incidentes à un nœud participent au calcul de son degré.

4.1.1.1 Exemple

Soit le graphe de la figure 4.1 ci-dessous.

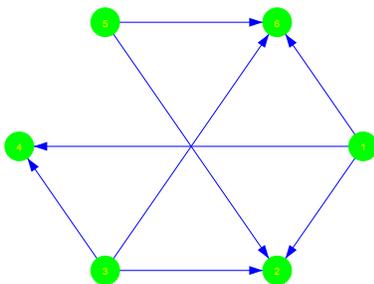


Figure 4.1 – Exemple d'un graphe orienté

si l'on considère le nœud étiqueté 4, le nombre de flèches qui entrent dans ce nœud est de 2, alors le degré entrant de ce nœud est 2. En revanche, aucune flèche ne sort de ce nœud, le degré sortant de ce nœud est donc de 0. Si on réitère cet exercice sur l'ensemble des sommets, on a alors le tableau 4.1 ci-dessous.

Graphe orienté et notion de degré			
nœuds	degré entrant (d_G^-)	degré sortant (d_G^+)	degré (d_G)
1	0	3	3
2	3	0	3
3	0	3	3
4	2	0	2
5	0	2	2
6	3	0	3

Tableau 4.1 – Tableau montrant les degrés entrant et sortant du graphe orienté de la figure 4.1

4.1.2 Incidence et adjacence, prédécesseurs et successeurs

L'incidence d'une arête à un nœud est un concept proche du concept de degré des nœuds, vu au point 4.1.1. Dans le cas d'un graphe non orienté, une arête est incidente à un sommet v s'il y a une liaison entre celle-ci et ce sommet. Deux nœuds ayant une arête incidente commune sont dits adjacents.

Une liste d'adjacence est une liste qui associe chaque nœud du graphe à une collection d'autres nœuds qui lui sont adjacents (qui ont une arête commune).

Une matrice d'adjacence est une matrice carrée A_d de dimensions $(|V| \times |V|)$ dont chaque élément $A_{d,i,j}$ est égal au nombre d'arêtes incidentes aux nœuds i et j , avec $|V|$ égal au nombre de nœuds du graphe. Autrement dit, la matrice d'adjacence permet de repérer rapidement dans un graphe les couples de sommets adjacents.

Un nœud u est prédécesseur d'un autre nœud v s'il existe une arête (u, v) connectant u et v . Par opposition, s'il existe une arête (v, u) connectant v et u , il est dit que u est un successeur de v .

4.1.2.1 Exemple

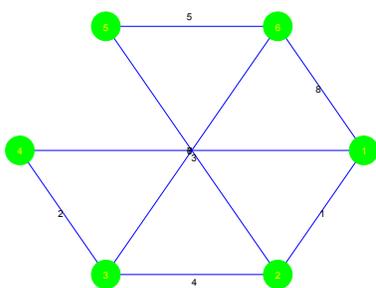


Figure 4.2 – Exemple d'un graphe non orienté avec les arêtes numérotées pour illustrer le concept d'incidence

Si l'on considère le graphe de la figure 4.2, on peut dire que le nœud étiqueté 1 est adjacent aux nœuds $\{2, 4, 6\}$. En réitérant le processus, on arrive au tableau 4.2 ci-dessous, où nous montrons les arêtes incidentes à chacun des nœuds.

Liste d'incidence						
Nœuds de référence	1	2	3	4	5	6
Arêtes incidentes	{1, 3, 8}	{1, 4, 7}	4, 2, 6	{3, 2}	{7, 5}	{8, 6, 5}

Tableau 4.2 – Tableau montrant la liste d'incidence du graphe de la figure 4.2

On peut déduire du graphe de la figure 4.2 la liste d'adjacence que comme son nom l'indique associe à un nœud du graphe les nœuds qui lui sont adjacents.

Liste d'adjacence						
Nœuds de référence	1	2	3	4	5	6
Nœuds adjacents	{2, 4, 6}	{1, 3, 5}	4, 2, 6	{1, 3}	{2, 6}	{1, 3, 5}

Tableau 4.3 – Tableau montrant la liste d'adjacence du graphe de la figure 4.2

Dans le tableau 4.3 ci-dessus, on constate que l'adjacence dans la théorie des graphes est un concept simple mais qui encapsule à la fois le concept de prédécesseur et de successeur. Toujours en prenant comme référence la figure 4.2, on peut déduire la liste des prédécesseurs et successeurs ci-dessous.

Liste de prédécesseurs et successeurs						
Nœuds de référence	1	2	3	4	5	6
Nœuds prédécesseurs	{ \emptyset }	{1, 3, 5}	{ \emptyset }	{1, 3}	{ \emptyset }	{1, 3, 5}
Nœuds successeurs	{2, 4, 6}	{ \emptyset }	{4, 2, 6}	{ \emptyset }	{2, 6}	{ \emptyset }

Tableau 4.4 – Tableau montrant la liste des prédécesseurs et des successeurs de du graphe de la figure 4.2

À partir de cette liste d'adjacence donnée par le tableau 4.3 on peut déduire la matrice d'adjacence suivante :

Matrice d'adjacence						
	1	2	3	4	5	6
1	-1	1	-1	3	-1	8
2	1	-1	4	-1	7	-1
3	-1	4	-1	2	-1	6
4	3	-1	2	-1	-1	-1
5	-1	7	-1	-1	-1	5
6	8	-1	6	-1	5	-1

Tableau 4.5 – Matrice d'adjacence du graphe de la figure 4.2

A noter que la matrice d'adjacence peut être remplie différemment. En effet une autre convention utilise des zéros à la place de -1 , et les numéros d'arêtes sont remplacés par des 1.

4.1.3 Isomorphisme de graphe

L'isomorphisme f est une structure qui préserve la bijection, $f : V(G) \rightarrow V(H)$. Il s'agit d'un concept qui met en jeu au moins deux graphes. La bijection entre les sommets des deux graphes isomorphes permet de préserver les arêtes. En conséquence, deux graphes isomorphes ont la même structure. Formellement, deux graphes G et H sont dits isomorphes si une paire des nœuds $\{u, v\}$ du graphe G est une arête de G , et que $\{f(u), f(v)\}$ est une arête de H . Une autre définition, peut être plus simple à comprendre, dit que deux graphes sont isomorphes s'ils ont le même nombre de sommets et sont connectés de la même façon. En pratique, cela veut dire qu'il suffit de déplacer les sommets du graphe G pour obtenir le graphe H et réciproquement. On écrit que $G \simeq H$.

L'isomorphisme de graphe préserve la topologie des graphes même si leur tracé peut sembler tout à fait différent. On comprend donc que l'isomorphisme de graphe soit très recherché notamment dans la mise en correspondance des problèmes liés à la topologie. L'isomorphisme de graphe serait donc un outil adapté à la reconnaissance des formes. Dans l'exemple suivant nous montrons deux graphes isomorphes aux tracés complètement différents.

4.1.3.1 Exemple

La figure 4.3 ci-dessous illustre isomorphisme de graphes à travers trois graphes dont deux sont isomorphes.

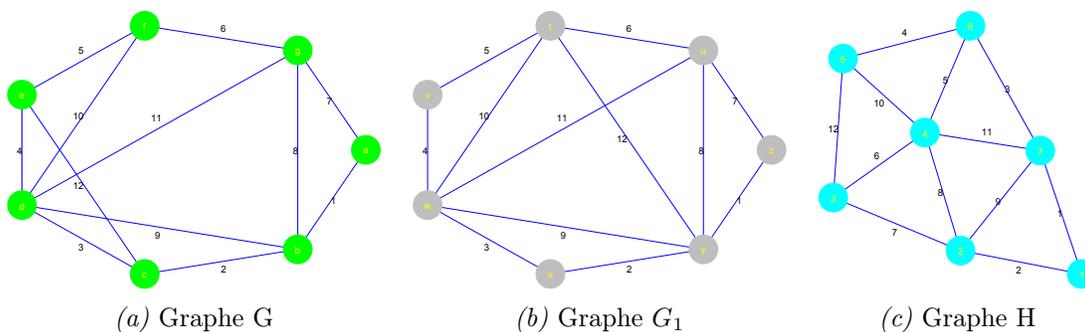


Figure 4.3 – Illustration d'isomorphisme

Les graphes G et H sont isomorphes tandis que même si G_1 a un dessin assez proche de G , ils ne sont pas isomorphes. En effet leur topologie est différente. Compte-tenu des définitions ci-dessus, les graphes G et H sont isomorphes. Le tableau 4.6 ci-dessous montre la bijection existante entre ces deux graphes. À noter qu'ici l'isomorphisme n'est pas défini par rapport aux poids des arêtes. D'ailleurs, les étiquettes sur les arêtes sont simplement les numéros représentant l'ordre de création de celles-ci.

Isomorphisme de graphes G et H				
nœuds Graphe G			nœuds Graphe H	
a		\leftrightarrow		1
b		\leftrightarrow		2
c		\leftrightarrow		3
d		\leftrightarrow		4
e		\leftrightarrow		5
f		\leftrightarrow		6
g		\leftrightarrow		7

Tableau 4.6 – Correspondance entre les graphes G et H

4.1.4 Graphes et sous-graphes

Soit un graphe $G = (V, E)$, le graphe $G' = (V', E')$ est un sous-graphe du graphe G si $V' \subseteq V$ et $E' \subseteq E$. Dit autrement, le graphe G' est obtenu à partir du graphe G , en enlevant un ou plusieurs nœuds du graphe G ainsi que les arêtes incidentes à ces nœuds. Cette opération peut conduire à d'autres notions de graphes, notamment le graphe partiel, si le sous-graphe de G' est composé de parties du graphe G complètement séparées à cause de la suppression d'une ou de plusieurs arêtes, ou de clique (sous-graphe induit complet, c'est-à-dire graphe tel que quelque soit le couple de nœuds pris au hasard ils sont adjacents).

4.1.4.1 Exemple de graphe et sous-graphes

Considérons le graphe G de la figure 4.4a ci-dessous. Ce graphe est composé de l'ensemble des sommets $\{V1, V2, V3, V4, V5, V6, V7\}$, et des arêtes $\{e1, e2, e3, e4, e5, e6, e7, e8, e9, e10\}$. Si l'on considère que ce graphe est notre graphe de départ, on peut à partir de ce graphe, déduire d'autres graphes, en procédant à des opérations de suppression de nœuds de notre graphe G , ainsi que des arêtes incidentes à ces nœuds. Par exemple, en enlevant le nœud $V3$ du graphe, on crée un graphe formé de deux parties, $\{V1, V2, V5\}$ et $\{V4, V6, V7\}$, cf. figure 4.4b, ce graphe est un sous-graphe du graphe G . On l'appellera G_1 . Les deux parties du graphe G_1 ont la particularité d'être aussi des cliques de G , cf. section 4.1.4.

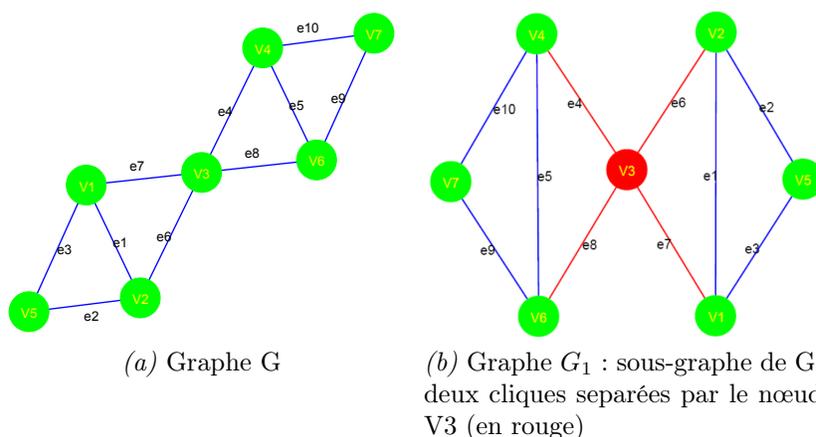


Figure 4.4 – Illustration de graphes et sous-graphes

4.1.5 Bipartisme de graphe

Un graphe $G = (V, E)$ est dit biparti si l'ensemble de ses nœuds $V(G)$ peut être partitionné en deux ensembles disjoints X et Y tels que chaque arête $e \in E$ puisse connecter un nœud de type X à un nœud de type Y . La partition (X, Y) de V appelée une bipartition. Un graphe biparti avec la bipartition (X, Y) de V est noté $G = (X, Y, E)$, [Panda and Pradhan, 2013].

4.1.5.1 Exemple

Si l'on observe le graphe de la figure 4.5, on constate que l'ensemble des nœuds de ce graphe peut être séparé en deux sous-ensembles distincts, tel que le premier groupe comporte tous les nombres impairs de 1 à 6 compris et le deuxième groupe comporte tous les nombres pairs de 1 à 6. Ce graphe est donc biparti. Formellement, on peut écrire que $X \subset V$, $Y \subset V$, et que $V = X \cup Y$ avec $X \neq Y$, où $X = \{1, 3, 5\}$ et $Y = \{2, 4, 6\}$

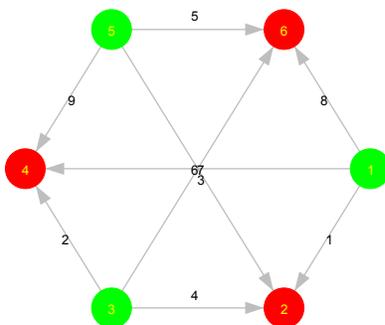


Figure 4.5 – Exemple d'un graphe biparti, formé de l'ensemble de nombre impairs et pairs de 1 à 6

À titre informatif, le graphe de la figure 4.1 est aussi un graphe biparti.

4.1.6 Graphe attribué relationnel (ARG en anglais pour **Attributed Relational Graph**)

Un graphe attribué relationnel est un graphe $G(V, E)$ dont l'ensemble des nœuds et ou d'arêtes sont étiquetés avec des attributs. Les nœuds possèdent des étiquettes de description tandis que les arêtes matérialisent une certaine relation entre les nœuds. Par exemple, dans une application de segmentation d'images en plusieurs régions, les nœuds peuvent être les caractéristiques de ces régions (taille de la région, son aire, son périmètre, sa forme, son centre, sa convexité, ses moments statistiques, etc) et les arêtes peuvent être une pondération en termes de distance de deux régions ayant certaines caractéristiques. Nous reviendrons sur la construction de graphes attribués dans la section 4.3.4.1, où nous montrerons notamment quelques exemples de graphes attribués relationnels.

4.1.7 Graphes et le problème de voisinage

Un graphe est un puissant outil de représentation. Le problème de voisinage de graphe, consiste à utiliser le graphe pour donner un sens à un ensemble de points distribués dans l'espace. Ce type d'utilisation de graphe est classique et très présent dans les Systèmes d'Informations Géographiques (SIG), où l'on cherche à connecter des points représentant par exemple des villes, ou encore dans le problème de reconnaissance de forme où l'on cherche à connecter les objets d'une scène de telle sorte que la liaison entre deux objets ait une signification. Utiliser le graphe pour donner un sens, une structure à un ensemble de points remonte à très longtemps. On peut en ce sens citer l'utilisation de la triangulation de Delaunay, ou encore l'utilisation des graphes de Gabriel, ou de Graphe de voisinage relatif de Toussaint, parmi tant d'autres types de représentations. Dans nos travaux, l'objectif est représenter nos ensembles de primitives, segments de droite et ellipses représentant les entités de test et de référence que nous cherchons à apparier. Pour ce faire nous allons de façon succincte aborder quelques-uns des graphes de voisinage les plus connus.

4.1.8 Graphe de Voisinage Relatif RNG en anglais pour **Relative Neighbourhood Graph**

Soit un ensemble des points distribués dans l'espace, le graphe de voisinage relatif introduit par Toussaint, en 1980, est une amélioration du graphe de voisinage "relativement proche" de Lankford, défini 1959 qui permet de régler le problème de dégénérescence du graphe de voisinage de Lankford. En effet, selon Lankford si on considère un ensemble de points P , formé de n points distincts dans le plan, $P = \{p_1, p_2, \dots, p_n\}$ deux points p_i et $p_j, \in P$ sont voisins si et seulement si :

$$d(p_i, p_j) < \max[d(p_i, p_k), d(p_j, p_k)] \quad \forall k = 1, \dots, n, k \neq i, j. \quad (4.1)$$

Or, cette définition amène selon Toussaint a une dégénérescence. En effet, si on prend 3 points équidistants, avec cette relation 4.1 on ne peut relier aucun de ces trois points à un autre point appartenant à l'ensemble des trois points. La relation de voisinage RNG, dans la définition de Toussaint est la suivante. Si on considère le même ensemble des points P ci-dessus, deux points p_i et p_j , $\in P$ sont voisins si et seulement si :

$$d(p_i, p_j) \leq \max[d(p_i, p_k), d(p_j, p_k)] \quad \forall k = 1, \dots, n, k \neq i, j. \quad (4.2)$$

4.1.9 Le graphe de Gabriel

Dans le graphe de Gabriel, deux points p_i et p_j , $\in P$ sont voisins si et seulement si :

$$d(p_i, p_j)^2 \leq d(p_i, p_k)^2 + d(p_j, p_k)^2, \quad \forall k = 1, \dots, n, k \neq i, j \quad (4.3)$$

A l'instar de la relaxation de la contrainte de voisinage effectuée par Toussaint à la définition initiale de voisinage relatif, la définition ci-dessus, cf. éq. (4.3) est une modification apportée à la définition de graphe de Gabriel de Matula et Sokal, 1980, cf. [Cimikowski, 1992].

4.1.10 Autres graphes de voisinage

Les graphes de voisinage présentés dans les sections 4.1.8, 4.1.9 ne sont pas les seuls existants. On pourraient encore citer le graphe d'Urquhart, le β -Skeleton ($\beta \leq 1$) le MST et bien d'autres. Néanmoins, il nous a paru opportun de décrire ces graphes car ils sont en quelque sorte la référence en matière de graphe de voisinage permettant de respecter la structure des données qu'ils représentent, tout en évitant les cas extrêmes de la triangulation de Delaunay ou de l'arbre couvrant minimal (Minimum Spanning Tree en anglais, MST).

4.1.11 Mise en correspondance de deux graphes

Dans l'appariement de graphes, l'usage, lorsque cela est possible, consiste à effectuer des appariements exacts à travers l'isomorphisme de graphe, ou la recherche des plus grands sous-graphes communs. Parfois l'utilisation de graphes inexacts est nécessaire [Roberto M. Cesar-Jr. et al., 2005].

Selon Horst Bunke [Bunke, 2000] et Shokoufandeh [Shokoufandeh and Dickinson, 1999] dans des applications réelles, il n'est pas toujours possible d'utiliser les concepts tels que l'isomorphisme de graphe ou la recherche du plus grand graphe ou sous-graphe commun car les données du monde réel ne sont pas parfaites. Par conséquent, une solution possible consiste à utiliser une méthode qui calcule la similarité entre les deux graphes. L'approche décrite dans ce manuscrit cherche à apparier deux graphes via une fonction de mesure

de similarité, décrite ultérieurement, basée sur les caractéristiques des nœuds des deux graphes attribués relationnels (cf. section 4.12).

4.2 Schéma fonctionnel de la méthode

La mise en correspondance de primitives extraites de deux images requière plusieurs étapes, dès l'acquisition de l'image jusqu'à la vérification du respect de la contrainte d'unicité de l'appariement. Le schéma de la figure 4.6 ci-dessous montre le lien entre les différentes étapes nécessaires à la réalisation de la mise en correspondance entre deux ensembles de primitives extraites de deux images dont une primitive de référence et l'autre de test.

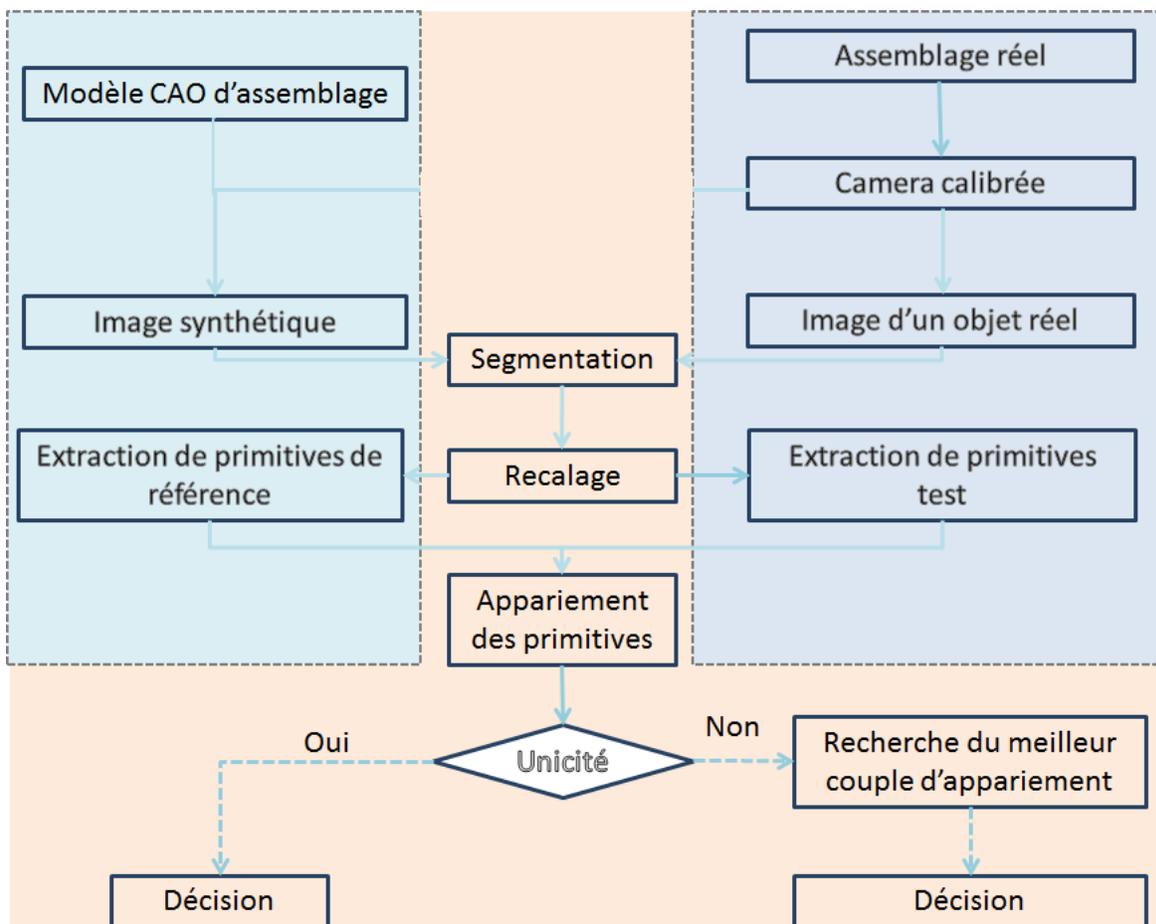


Figure 4.6 – Schéma fonctionnel de notre méthode d'appariement de primitives

Le but étant d'inspecter un assemblage réel, on part d'une image de cet assemblage, pris avec une caméra calibrée. La pose utilisée pour acquérir l'objet réel est appliquée au modèle CAO, permettant ainsi d'avoir le modèle CAO et l'assemblage réel pris tous les deux selon le même point de vue. Cela permet de disposer de deux images, une réelle et une synthétique. S'en suit alors une étape de segmentation afin de récupérer l'objet d'intérêt.

Rappelons-le, l'information commune au modèle CAO et à l'objet réel est la géométrie de l'objet à inspecter. Puisque cet objet peut avoir été légèrement déplacé, nous prévoyons une étape de recalage rigide avant de procéder à l'extraction de primitives utilisées lors de la mise en correspondance. Pendant la phase d'appariement on s'assure que l'unicité de la mise en correspondance est vérifiée avant de juger de la conformité de l'objet inspecté.

4.3 Développement de la méthode d'appariement de primitives en exploitant des images de synthèse

Dans cette section nous allons nous intéresser d'abord à l'appariement de deux images synthétiques, A et B, où l'image A est l'image de référence et l'image B est l'image test. Cette image test B est une déformation ou transformation de l'image de référence A, avec l'objectif de générer des défauts pour mettre en relief un aspect particulier de la mise en correspondance, lorsque l'on enlève, rajoute, ou déplace un ou plusieurs éléments de l'image A. Ces images de synthèse ont été générées au format vectoriel SVG puis converties au format matriciel ".tiff" pour les traitements de segmentation réalisés avec Matlab ©. L'un des avantages du format vectoriel SVG par rapport à la génération de l'image au format matriciel ".bmp" réside dans le fait que celui-ci permet de générer uniquement la géométrie des éléments. L'avantage d'utiliser des images de synthèse pour mettre en place la méthode réside dans le fait que l'on connaît en amont et avec certitude les défauts existant dans l'image test. De plus l'utilisation de ces images rend plus simple les traitements.

La figure 4.7 montre un ensemble de sept images synthétiques. Nous allons utiliser l'image 4.7b pour expliquer notre méthode d'appariement de primitive à travers un cas. Dans le chapitre 5 nous montrerons des résultats sur plusieurs cas synthétiques et réels.

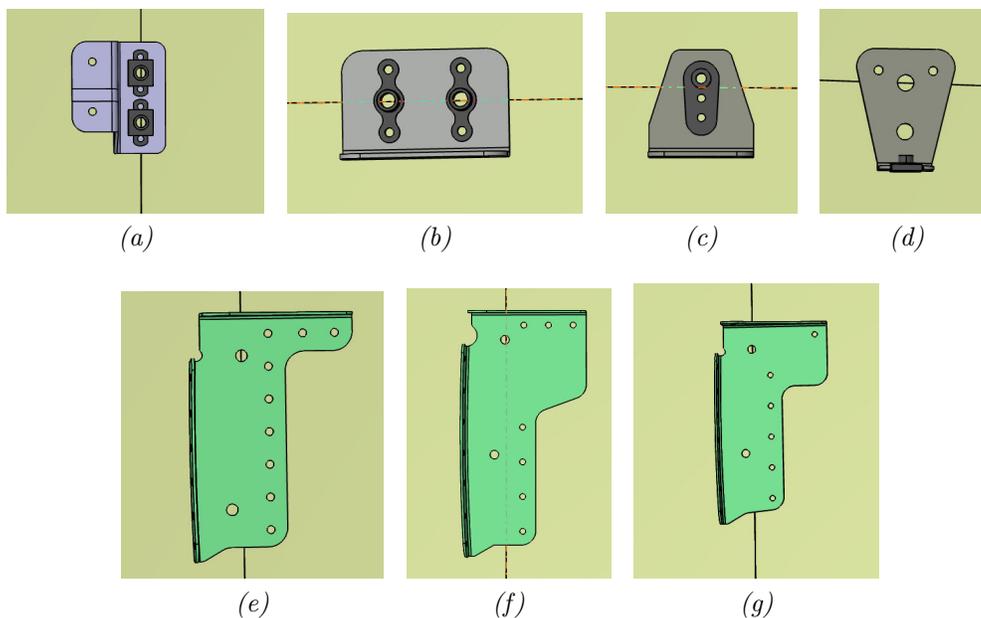


Figure 4.7 – Images générées à partir du modèle CAO de l'assemblage, en format ".bmp"

Bien que la méthode sera détaillée en prenant comme exemple l'image synthétique de la figure 4.7b, nous présentons cet ensemble de sept images afin d'illustrer les traitements à effectuer sur les images, notamment la segmentation (cf. section 4.3.1) en vue de l'ex-

traction des primitives présentée dans la section 4.3.3. Chaque bloc du schéma fonctionnel présenté dans la section 4.2 sera décrit ultérieurement dans les sections suivantes :

- ◇ La section 4.3.1 explique comment obtenir des contours à partir des images théoriques (cf. figure 4.7), en décrivant la série de pré-traitements réalisés sur les images synthétiques.
- ◇ La section 4.3.2 décrit le processus d'alignement de deux images à appairer.
- ◇ La section 4.3.3 explique comment on procède à l'extraction des primitives à appairer dans les deux images.
- ◇ La section 4.3.4 traite de l'appariement des primitives (cf. section 4.3.3) en s'appuyant sur la théorie des graphes.
- ◇ La section 4.3.5 emploie l'utilisation d'un graphe biparti afin de vérifier la contrainte d'unicité de l'appariement (cf. section 4.3.4).

4.3.1 Segmentation des sept images de la figure 4.7

L'objectif de la segmentation est d'isoler l'information importante de l'information qui n'est pas utile pour la suite des traitements. Partant de l'image au format ".bmp" (cf. figure 4.7), après le chargement et conversion d'une image couleur originale en image en niveau de gris, l'image est binarisée. On constate que l'image binarisée contient des trous. Afin de pouvoir isoler les éléments à l'intérieur de l'objet de la partie extérieure de l'objet on va procéder en plusieurs étapes. Pour commencer, partant de l'image binarisée (que l'on appellera ci-après *Imb*) on effectue une recherche de composantes connexes, en prenant en compte les 8 plus proches voisins. Pour l'ensemble des régions obtenues grâce à la recherche des composantes connexes, on calcule les attributs suivantes :

- ◇ l'aire,
- ◇ La liste ordonnée des pixels formant la région,
- ◇ les coordonnées du centre de la région,
- ◇ le nombre d'Euler (différence entre nombre d'objets et nombre de trous dans l'image),
- ◇ l'excentricité,
- ◇ la boîte englobante.

Partant toujours de l'image binarisée *Imb*, on procède au remplissage des trous de l'image binarisée en appliquant d'abord une fermeture morphologique avec un disque de taille 15 (taille adaptée pour nos images, trouvée par essai-erreur) puis en faisant appel à l'algorithme *food fill*. On obtient avec ces opérations une image où la zone représentant l'objet d'intérêt devient complètement uniforme (on l'appellera *Imf*). Avec cette opération de

food fill, il devient facile d'extraire le contour extérieur de la forme en appliquant une opération d'extraction de périmètre sur l'image *Imf*. Puisqu'il peut éventuellement y avoir des objets ou artefacts dans le voisinage de notre objet d'intérêt, on effectuera quelques opérations supplémentaires. Partant de l'image *Imf* après recherche des composantes connexes (au sens des 8 plus proches voisins), on calcule les informations suivantes pour chaque région :

- ◇ l'aire,
- ◇ La liste ordonnée des pixels formant la région,
- ◇ le centre de la région,
- ◇ le nombre d'Euler,
- ◇ la taille de l'axe majeur de la région,
- ◇ boîte englobante,

Partant des composantes connexes, on recherche ensuite la région d'aire maximum, en faisant l'hypothèse qu'il n'y a pas d'autres objets plus gros que l'objet qui nous intéresse dans l'image, en prenant soin d'élaguer cette région des éventuels pixels formant des branches indésirables grâce à l'application d'un filtre morphologique qui permet d'amincir le contour de la région pour obtenir une épaisseur égale à un pixel. Si l'on suppose que la plus grande région peut avoir une branche connexe à sa partie externe, on peut appliquer un algorithme de croissance de région sur celle-ci, en partant du centre de la région. L'algorithme s'arrêtant à la frontière, on récupère bien une région avec une délimitation plus souhaitable. C'est pour cette raison que les coordonnées du centre sont calculées pour chaque composante connexe.

À ce stade on connaît le périmètre de notre objet d'intérêt et comme on a calculé le nombre d'Euler pour l'image binaire de départ on sait combien de régions il y a à l'intérieur de notre objet d'intérêt.

On peut maintenant extraire ces petites régions afin de poursuivre vers une segmentation complète de notre image. Puisqu'au début nous avons effectué une recherche des composantes connexes de l'image binaire, maintenant que nous connaissons la région délimitant notre objet (sa boîte englobante), on va pouvoir extraire les régions à l'intérieur de l'objet en nous servant à la fois de la boîte englobante de l'objet et de l'ensemble des composantes connexes calculées à partir de l'image binaire.

La stratégie, cette fois, consiste à chercher tous les objets inclus dans la boîte englobante de la région principale, à une tolérance arbitraire d'au moins 10 pixels. Pour chacune des ces régions, après une opération de chaînage des contours et de remplissage des trous, on récupère la liste des pixels des objets inclus dans l'objet d'intérêt. Pour compléter

la segmentation, on amincit les contours des différentes régions pour que leur épaisseur soit d'un pixel maximum. Une simple opération d'addition logique permet d'obtenir une image segmentée où on a une forme extérieure ainsi que les objets inclus dans celle-ci. La segmentation en régions a été réalisée pour l'ensemble des sept images synthétiques de la figure 4.7. Les résultats de cette segmentation sont disponibles dans l'annexe B.

À partir des images segmentées, il devient plus simple de procéder à l'extraction de caractéristiques particulières dites primitives pour réaliser la mise en correspondance entre une image de contour prise pour base de comparaison avec une image test.

Comme nous l'avons dit auparavant, on va prendre pour référence une des images synthétiques de la figure 4.7 pour décrire la méthode. La méthode compare une image de référence à une image test. Ici l'image test sera une transformation de l'image synthétique de référence. Nous avons choisi de travailler sur l'image 4.7b. Ci-dessous nous montrons quelques exemples d'images test (segmentées) générées à image de référence (cf. figure 4.8). Les primitives à appairer seront extraites de ce type d'images.

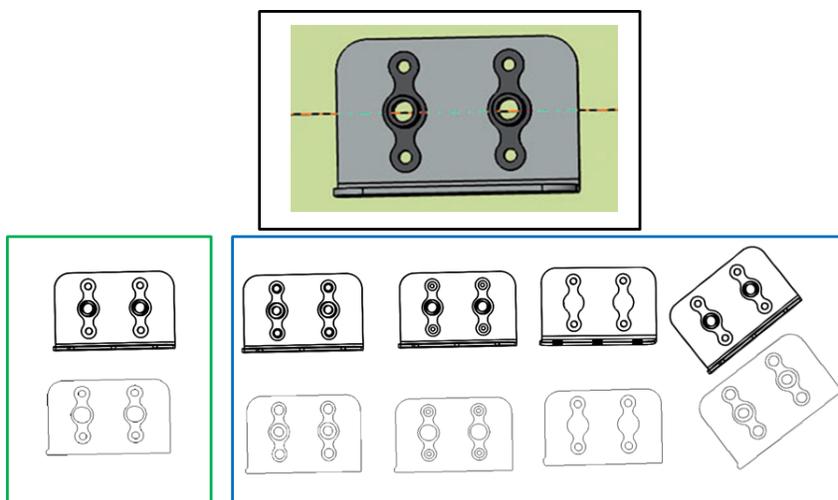


Figure 4.8 – Première ligne : copie d'écran du modèle CAO. Deuxième ligne, à gauche : image générée à partir du modèle CAO, au format vectoriel et son contour, extrait après le passage du format vectoriel au format matriciel (".tiff") ; à droite : images du modèle CAO modifiées, au format vectoriel (SVG) et leur contours respectifs après la conversion au format matriciel (".tiff").

Avant de rentrer dans des considérations liées à l'appariement, il est important de se rappeler que dans une application d'inspection, comme celle que l'on envisage dans ce projet, où en pratique les objets peuvent être montés incorrectement, c'est-à-dire avec une translation rotation ou même être de dimensions plus grandes ou plus petites comparativement à ce qui a été modélisé. Ainsi, il est primordial de mettre en place une procédure de recalage rigide afin d'avoir et l'objet de référence et l'objet test dans le même repère avant de poursuivre vers l'extraction des primitives qui seront utilisées lors de l'appariement référence/test réalisé ultérieurement. Pour cette raison, la section suivante aborde cette

question d’alignement de l’image de référence avec l’image test.

4.3.2 Recalage d’images

Soient deux images (ou plus) représentant la même scène (prise à différents instants, point de vues ou avec différents capteurs) le recalage d’image [Dubey et al., 2012, Zitová and Flusser, 2003] (ici de contours) est un processus qui permet de ramener ces images au même référentiel, de telle sorte qu’elles soient alignées [Zitová and Flusser, 2003].

Comme nous l’avons expliqué précédemment, l’objet à évaluer peut avoir été assemblé de façon incorrecte par l’opérateur, en conséquence, une étape de recalage 2D doit être envisagée avant d’effectuer l’extraction et l’appariement des primitives de l’image test avec l’image de référence. Cette situation est illustrée par la figure 4.9, où nous simulons une translation de l’objet test par rapport à l’objet de référence (cf. figure 4.7b). Les prochains paragraphes expliqueront comment ce recalage est effectué, en utilisant les contours. En

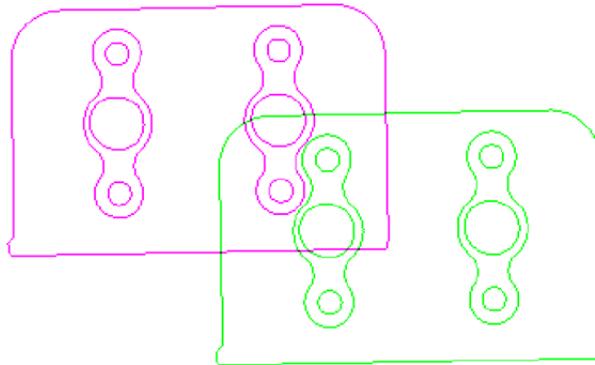


Figure 4.9 – Deux contours avant recalage : contour de l’image de référence en vert et contour de l’image test en magenta

observant la figure 4.9 il paraît évident que les deux contours (référence et test) nécessitent d’être recalés si on souhaite les apparier. Le processus de recalage permettra d’obtenir la matrice décrivant la transformation rigide à partir de laquelle on peut calculer les paramètres du recalage 2D (angle de rotation et le vecteur de translation) qui serviront à superposer les deux contours (référence en vert et test en magenta, cf. figure 4.9).

Le recalage d’image a été traité aussi bien par la communauté d’analyse d’image que par celle de vision artificielle. En effet, le recalage est une étape préalable à plusieurs applications de ces deux domaines, à savoir, la reconstruction 3D, l’analyse d’images médicales, l’inspection, et bien d’autres sujets. Les méthodes de recalage sont nombreuses et spécifiques aux types de données que l’on souhaite recalcr. On peut citer des méthodes à base de descripteurs locaux (cf. **chapitre 2**), corrélation de phase, optimisation par la descente de gradient (exploitant la transformée en distance comme dans le **chapitre 3**) qui ont pu être employées dans différentes applications.

4.3.2.1 Notre choix

En ce qui concerne notre application, nos images ont été recalées en utilisant un algorithme classique de la descente de gradient cherchant à minimiser l'erreur quadratique moyenne entre les deux images d'entrée ou à maximiser l'information mutuelle [Mattes et al., 2001, Smriti et al., 2005]. Certaines de nos images ont pu être recalées en utilisant l'algorithme de corrélation de phase [Reddy and Chatterji, 1996, Kokila and Thangavel, 2013]. Le résultat de la figure 4.10 a été obtenu en appliquant un algorithme classique de la descente en gradient cherchant à minimiser l'erreur quadratique moyenne entre les deux images de la figure 4.9.

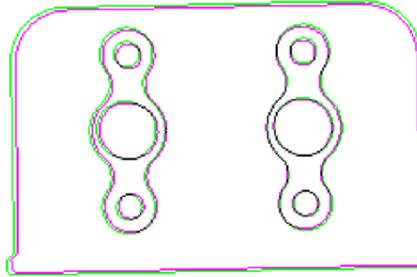


Figure 4.10 – Deux contours recalés : contour de l'image de référence en vert et contour de l'image test en magenta. Les pixels noirs concernent les endroits où les pixels verts et magenta se superposent.

Pour une transformation rigide 2D, la matrice de transformation (*note TM dans suite*) est composée d'une matrice de rotation $R(\theta)$ et d'un vecteur de translation T . Cette matrice de transformation rigide (TM) s'écrit :

$$TM = \begin{pmatrix} \cos(\theta) & -\sin(\theta) & t_x \\ \sin(\theta) & \cos(\theta) & t_y \\ 0 & 0 & 1 \end{pmatrix} \quad (4.4)$$

Le résultat de la figure 4.10 est obtenu en appliquant la matrice de transformation rigide (4.5) à l'image de contour de référence pour que celui-ci s'aligne avec le contour de l'objet test. Les paramètres de recalage (l'angle de rotation $\theta_1 = 0.2^\circ$ et le vecteur de translation $T_1 = [-127.804, -65.295]^t$) sont obtenus à partir de cette matrice de transformation rigide (4.5) ci-dessous :

$$TM_1 = \begin{pmatrix} 0.999 & -0.0004 & -127.804 \\ 0.0004 & 0.999 & -65.295 \\ 0 & 0 & 1 \end{pmatrix} \quad (4.5)$$

4.3.3 Extraction de primitives géométriques

L'étape de recalage étant accomplie, il est possible d'avancer un pas de plus vers l'appariement des images. L'étape suivante est celle d'extraction des primitives que nous allons apparier ultérieurement.

La méthode que nous présentons dans ce chapitre est envisagée pour le contrôle d'assemblages contenant plusieurs éléments de formes ellipsoïdales et/ou des éléments pouvant se modéliser par des formes géométriques simples de telle sorte qu'on peut les modéliser totalement ou partiellement par un ensemble de segments de droite. De ce fait, les ellipses et les segments de droite sont les primitives que nous chercherons à extraire de l'image réelle et des données de la CAO pour effectuer une mise en correspondance entre les données de l'image de référence et celles de l'image test afin de vérifier la conformité des éléments d'un assemblage. L'extraction de ces primitives est un problème connu en vision. Il existe principalement deux grandes familles de méthodes d'extraction de ces types de primitives (ellipses et segments de droite).

4.3.3.1 Méthodes à base de système de votes et d'ajustement de courbes aux données

4.3.3.1.1 Méthodes de type Transformée de Hough

La première grande famille de méthodes d'extraction permettant d'obtenir les ellipses et les segments de droite dans une image dérive soit de la transformée de Hough [Duda and Hart, 1972] qui utilise un système de vote ou soit d'une approche qui consiste en l'ajustement d'une équation mathématique aux données via un processus d'optimisation de type moindres carrés. Il a été montré que cette catégorie de méthodes peut être utilisée aussi bien pour trouver des segments de droite ou des ellipses dans une image. La transformée de Hough classique [Duda and Hart, 1972] est une référence en la matière de détection de segments de droite dans une image. Elle a été généralisée pour résoudre les problèmes d'autres types de courbes paramétriques telle que les ellipses. À titre d'exemple on peut citer la transformée de Hough Probabiliste [McLaughlin, 1998] qui a pu être utilisée avec succès afin d'extraire les segments de droite et les ellipses dans une image.

4.3.3.1.2 Méthodes d'ajustement de type Prasad

Prasad et al. [Prasad et al., 2013] ont présenté un algorithme d'ajustement d'ellipses aux données d'une image permettant une extraction efficace d'ellipses. Une implémentation de cet algorithme rendu disponible par les auteurs permet en effet une bonne extraction d'ellipses, après segmentation et décomposition en composantes connexes.

4.3.3.2 Méthodes basées sur le regroupement de contours

La deuxième catégorie des méthodes d'extraction de primitives dans une image concerne les méthodes basées sur le regroupement de contours [Wang et al., 2014, Nguyen et al.,

2009]. Un algorithme qui est en train de devenir une référence en la matière d'extraction de segments de droite dans une image est celui présenté par van Gioi dans l'article [Rafael Grompone von Gioi et al., 2012].

Les méthodes appartenant à cette catégorie ont pu être utilisées pour la détection d'ellipses et d'arcs d'ellipses.

4.3.3.3 Autres méthodes

Il est important de noter qu'un autre classement de méthodes d'extraction de primitives peut être établi et que la décomposition en deux classes n'est pas la seule catégorisation possible. En effet, Wang et al. [Wang et al., 2014] ont fait une classification décomposant les méthodes d'extraction de primitives en trois catégories, en incluant dans leur classification les méthodes à base d'algorithmes génétiques.

Dans la référence [Patraucean et al., 2012], Patraucean et al. ont présenté une méthode d'extraction d'ellipses et segments de droite " *Ellipse and Line Segment Detector*" (ELSD, en anglais) qui fonctionne sans que l'utilisateur ait à régler un quelconque paramètre. Cette méthode pourrait être classifiée comme une méthode hybride car elle combine à la fois les caractéristiques des méthodes présentées aux sections 4.3.3.1 et 4.3.3.2.

4.3.3.4 Notre choix

L'objectif des travaux présentés dans cette thèse n'est pas de développer un nouveau algorithme d'extraction de primitives mais plutôt d'utiliser un ou plusieurs algorithmes déjà existants afin d'extraire les primitives qui seront appariées ultérieurement. L'algorithme de Prasad [Prasad et al., 2013] a été testé sur nos images et fonctionne correctement dès lors que les images ont été préalablement segmentées et décomposées en composantes connexes. Actuellement, la segmentation, sur les images présentées dans cette section, est faite en combinant la morphologie mathématique et des techniques de décomposition en composantes connexes. L'algorithme ELSD [Patraucean et al., 2012] a également été testé. L'avantage de cet algorithme réside dans le fait qu'il peut être utilisé seul pour extraire à la fois les ellipses et les segments de droite. En revanche, même en utilisant des images bien segmentées, il est nécessaire de filtrer le résultat obtenu. Pour cette raison, dans notre approche il a été choisi d'utiliser l'algorithme de Prasad et al. [Prasad et al., 2013] pour l'extraction des cercles et la transformée de Hough classique pour l'extraction des segments de droite.

4.3.3.4.1 Extraction d'ellipses par la méthode de Prasad et leur caractérisation

Notre objectif consiste à utiliser l'algorithme de Prasad, [Prasad et al., 2013] afin de détecter des ellipses par une méthode d'ajustement, qui pour un ensemble de points des données, cherche la meilleure ellipse s'ajustant à ces points, en déterminant ses paramètres, à savoir, les coordonnées du centre de l'ellipse, les deux demi-longueurs des deux axes et l'orientation de l'ellipse.

Cette méthode de détection est appliquée sur les deux images théoriques, l'image synthétique servant de référence et l'image test, après la phase de segmentation et de recalage. L'algorithme d'extraction fonctionne de la manière suivante :

Pour chaque composante connexe r de l'image d'entrée, on récupère la liste des pixels appartenant à cette région r . On applique l'algorithme de Prasad d'ajustement d'ellipses par moindres carrés sur l'ensemble des points de cette région, et on obtient ainsi les paramètres de l'ellipse représentant la région ainsi que le résidu d'optimisation qui permet de connaître la précision de l'ajustement de l'ellipse aux points des données. A noter qu'un résidu proche de zéro signifie que l'ellipse s'ajuste bien aux points des données. Partant des informations données par l'algorithme de Prasad, on construit une "matrice d'ellipses candidates", où chaque ligne correspond aux paramètres d'une ellipse (y compris le résidu), et le numéro d'étiquette de la composante connexe. Dans cette matrice, une ligne correspondant à une ellipse contient aussi l'aire de l'ellipse. Une fois que la matrice d'ellipses candidates est remplie, on sélectionne les ellipses pour lesquelles le résidu obtenu pendant le processus d'ajustement est plus faible qu'un seuil prédéfini. Afin de lever toute ambiguïté par rapport à la sélection du cadre extérieur d'un objet, on ne sélectionne que les ellipses dont la longueur de l'axe majeur est inférieur à la longueur du grand axe de l'ellipse représentant le contour extérieur de l'objet duquel on souhaite extraire les ellipses. Un exemple de l'utilisation de la méthode de Prasad sur une de nos images synthétiques est illustré par la figure 4.11.

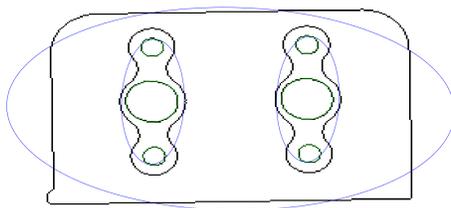


Figure 4.11 – Extraction d'ellipses par la méthode de Prasad : les ellipses considérées correctes sont dessinées en vert et les candidates non retenues sont dessinées en bleu. Le contour est affiché en noir sur fond blanc pour plus de clarté

Comme nous l'avons dit précédemment, l'algorithme de Prasad, permet de récupérer

les paramètres décrivant une ellipse, en plus d'un résidu d'optimisation permettant de connaître la précision de l'ajustement de la courbe dessinée par l'ellipse sur les points en entrée de l'algorithme. En vue de l'appariement ultérieur, nous avons construit un tableau résumant les caractéristiques utiles à l'appariement (cf. tableau 4.7).

Tableau descriptif de ellipse						
	cx_e	cy_e	l_1	l_2	orientation θ_e ($^\circ$)	$\pi \times l_1 \times l_2$
1	797.8	490.9	16.74	16.02	-0.35	842.7
2	797.9	448.2	7.20	6.97	-0.16	157.8
3	799.4	534.2	7.19	6.99	-0.06	157.9
4	897.1	489	16.69	15.96	-0.43	837.2
5	897	446.2	7.23	6.98	-0.47	158.9
6	898	532.2	7.12	7.004	-0.78	156.8

Tableau 4.7 – Tableau décrivant les primitives de type ellipse, cf. figure 4.11

4.3.3.4.2 Extraction de segments de droite par la transformée de Hough

Notre objectif est d'extraire aussi bien les segments de droite que les ellipses. Ainsi, après l'extraction des ellipses, on enlève les composantes connexes constituant les ellipses sélectionnées dans l'étape d'extraction d'ellipses et on procède à l'extraction des segments de droite par la transformée de Hough standard. En choisissant les quatre pics les plus importants de l'accumulateur de Hough, on obtient les quatre segments de droite (en vert) de la figure 4.12

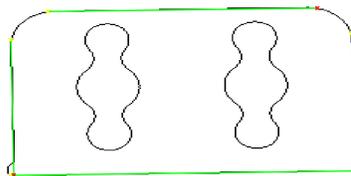


Figure 4.12 – Extraction de segments de droite par la transformée de Hough standard. Les quatre segments de droite correspondant aux quatre pics les plus importants de l'accumulateur de Hough sont dessinés en vert. Les débuts des segments de droite sont dessinés en jaune et leur fin en rouge.

En réalité, les quatre segments des droite sont représentés par leurs deux points terminaux. A partir des deux points terminaux d'un segment de droite, on déduit leurs caractéristiques utilisées dans la mise en correspondance, résumées par le tableau 4.8 suivant.

Tableau descriptif du segment de droite								
	cx_s	cy_s	L_s (norme L_2)	orientation θ_s ($^\circ$)	x_1	x_2	y_1	y_2
1	848	570	230	-0.99	733	963	572	568
2	848.5	420.5	181	-0.94	758	939	422	419
3	734	510	124	89	733	735	448	572
4	963	504.5	125	89	9.62	964	442	567

Tableau 4.8 – Tableau décrivant les primitives de type segments de droite cf. figure 4.12

Les coordonnées des points terminaux x_1 , x_2 , y_1 , y_2 , d'un segment sont obtenues par la transformée de Hough standard. Ils nous permettent de déduire les coordonnées du point milieu du segment :

$$cx_s = \frac{(x_2 - x_1)}{2}. \quad (4.6)$$

$$cy_s = \frac{(y_2 - y_1)}{2}. \quad (4.7)$$

La longueur L_s d'un segment est déduite en appliquant la norme L_2 sur les coordonnées points terminaux.

$$L_s = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}. \quad (4.8)$$

L'orientation θ_s est obtenu par :

$$\theta_s = \arctan \frac{(y_2 - y_1)}{(x_2 - x_1)}. \quad (4.9)$$

Dans l'équation (4.9) on doit bien évidemment prendre les précautions nécessaires au cas limite $x_2 = x_1$.

4.3.3.4.3 Résultat de l'extraction des primitives

Dans les paragraphes précédents, nous avons décrit l'extraction d'ellipses, illustrée par la figure 4.11 et de segments de droite, illustrée par la figure 4.12. La figure 4.13 ci-dessous montre l'ensemble de ces primitives sur une même image.

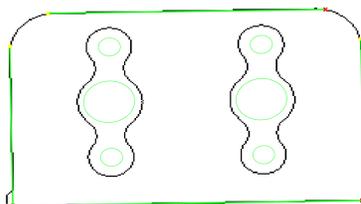


Figure 4.13 – Extraction d'ellipses et de segments de droite : affichage de l'ensemble des primitives sélectionnées (en vert)

Notre algorithme d'appariement d'une image de référence avec une image test fonctionne sur ces types de primitives dont les caractéristiques sont résumées dans les deux tableaux 4.7 et 4.8, vus précédemment. Après le recalage de l'image test avec l'image de référence et l'extraction de primitives, on obtient deux jeux de primitives qu'il convient d'apparier afin de décider de la conformité des objets présents dans l'image test.

La figure 4.14 ci-dessous montre un exemple de deux ensembles de primitives extraites de l'image de référence et de l'image test après le recalage des contours. Ces deux ensembles constituent l'entrée de l'algorithme d'appariement que nous avons développé, basé sur la théorie des graphes. L'algorithme sera présenté dans les prochaines sections.

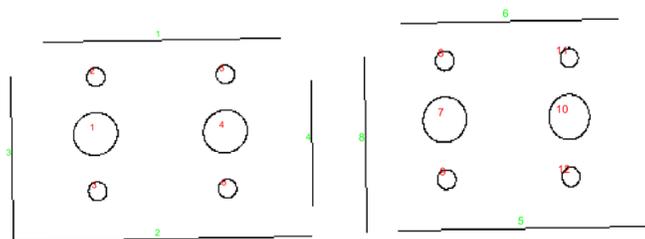


Figure 4.14 – Ensemble de primitives (segments de droite et ellipses) extraits de l'image de référence (gauche) et de l'image test (droite), après recalage de contour.

Dans l'image de référence, les ellipses sont numérotées de 1 à P , avec $P = 6$ pour l'exemple ci-dessus. En conséquence, les étiquettes des ellipses dans l'image test démarrent à partir de $P + 1$. Parallèlement, les segments de droite sont numérotés de 1 à L , avec $L = 4$ dans ce cas. Ainsi, dans l'image test, les étiquettes des segments de droite commencent de $L + 1$. Cette façon de numérotter les primitives reste valable pour les nœuds des graphes attribués, illustrés par la figure 4.17. Depuis le début du chapitre, deux types de primitives sont considérés, à savoir les ellipses et les segments de droite. Chaque type de primitive est défini par un ensemble d'attributs. Par exemple, une ellipse est définie par cinq paramètres, les deux coordonnées du centre, l'orientation et les deux demi-longueurs d'axes. Un segment de droite est quant à lui décrit par ses deux points terminaux.

4.3.4 Appariement des primitives par combinaison de graphes - fonction de ressemblance

Tout au long de cette section nous allons décrire notre méthode d'appariement de segments de droite et d'ellipses extraits de deux images, en exploitant les concepts de la théorie des graphes vus précédemment. Parmi les graphes présentés dans la section 4.1.7 notre choix s'est porté pour le graphe de voisinage relatif RNG. En effet, Toussaint a montré que le graphe RNG impose moins de structure aux données qu'il représente. Il a montré notamment qu'en prenant 4 points disposés dans l'espace comme un carré, le graphe RNG permettait de retrouver la forme de carré alors que cela n'était possible ni avec la triangulation de Delaunay, ni avec le MST. Nous allons construire des graphes attribués relationnels (au sens RNG) représentant les primitives que nous allons appairier ultérieurement.

Appairier les segments de droite [McIntosh and Mutch, 1988] est un sujet étudié depuis quelques décennies car la mise en correspondance de segments de droite entre deux images (ou plus) peut être utile dans des applications diverses telles que la reconstruction 3D [Bay et al., 2005] ou la reconnaissance d'objets, l'inspection, etc. Il en va de même pour des primitives telles que les ellipses [Hutter and Brewer, 2009], utilisées notamment dans l'inspection, l'estimation de pose, le calibrage, etc.

Dans cette partie, nous expliquerons notre méthode d'appariement de primitives, en étudiant d'abord le cas de la figure 4.14. D'autres cas seront discutés plus tard, dans le **chapitre 5**. La section 4.3.4.1 explique comment nous construisons les graphes attribués relationnels.

4.3.4.1 Construction des deux graphes en vue de l'appariement

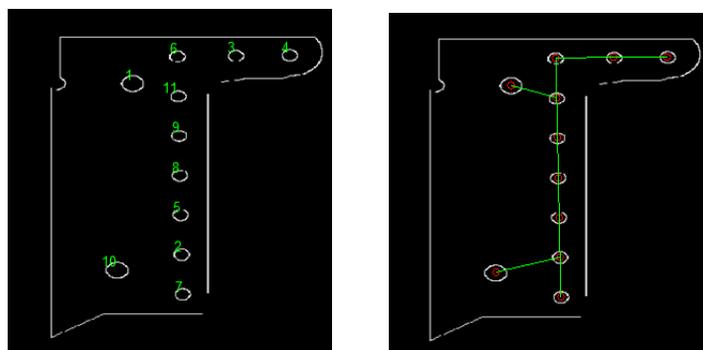
Soit deux ensembles de primitives X et Y , avec X représentant les primitives dans l'image de référence et Y représentant celles de l'image de l'objet test, on s'attache à montrer comment le score de ressemblance $s(X, Y)$ représentant la similarité entre ces deux ensembles peut être obtenu. Pour commencer, nous allons construire un graphe attribué par type de primitive (ellipse ou segment de droite) pour l'image de référence. Ensuite on construit également un graphe attribué par type de primitive pour l'image test. Nous allons illustrer de façon progressive la manière dont nous construisons un graphe attribué en utilisant la relation de voisinage relatif comme moyen de connecter deux primitives. L'objectif est de construire un graphe attribué relationnel où les nœuds du graphe représentent le centre d'une primitive, avec un vecteur d'attributs associé à chaque nœud. Une arête de ce graphe lie deux primitives proches au sens de la relation de voisinage relatif, cf. section 4.1.7, [Correa and Lindstrom, 2012].

Considérons le cas des ellipses, à partir des attributs d'un tableau descriptif du type ta-

bleau 4.7, on récupère les coordonnées du centre d'une ellipse et pour chaque centre d'une ellipse on cherche son plus proche voisin (au sens RNG, cf. section 4.1.8) et on relie ces deux points. On réitère ce processus sur l'ensemble des centres d'ellipses et on obtient un graphe relationnel. On associe chaque nœud du graphe construit avec le vecteur d'attribut décrivant l'ellipse. Ce vecteur d'attributs correspond à une ligne du tableau descriptif de l'ellipse, cf. section 4.7. Le graphe représentant l'élément de référence est composé de N nœuds (pour les N primitives de référence) et celui représentant l'élément test contient K nœuds (pour les K primitives extraites de l'image de test). En ce qui concerne les segments de droite, la procédure est la même. À partir du tableau descriptif d'un segment, on récupère les coordonnées du point milieu et pour ce point on cherche son plus proche voisin, toujours au sens 4.1.8 et on réitère jusqu'à ce que tous les segments soient représentés.

Il est important de préciser qu'une autre relation de voisinage (cf. section 4.1.7) pourrait aussi être utilisée, car toutes les informations exploitées pour l'appariement des primitives, proviennent du vecteur d'attributs associé aux nœuds. Les graphes attribués ayant leurs arêtes reliant les centres de primitives par une relation de proximité est une sorte de représentation du squelette de la forme. En conséquence, on peut dire que les arêtes des graphes attribués portent en elles l'information sur la topologie des objets.

Afin de mieux appréhender ce processus de construction de graphe attribué relationnel, nous allons utiliser l'objet de l'image 4.7e. Considérons, la figure 4.15, et plus particulièrement 4.15a contenant les ellipses extraites de l'objet de l'image 4.7e, numérotées de 1 à 11. Sur la figure 4.15b on peut observer un graphe topologique mettant en évidence ces ellipses.



(a) Ellipses extraites de l'image 4.7e.

(b) Chemin entre les ellipses de la figure 4.7e, obtenu en utilisant le voisinage relatif RNG, (cf. section 4.1.8).

Figure 4.15 – Passage de primitives (ellipses) au graphe relationnel.

Le chemin illustré dans la figure 4.15b met en évidence le graphe relationnel, utilisant la relation de voisinage relatif, de type RNG (cf. section 4.1.8). Ce chemin est l'équivalent

du graphe de la figure 4.16 ci-dessous.

En associant chaque nœud à son vecteur descriptif, on obtient le graphe attribué relationnel pour cet ensemble d'ellipses. Comme nous l'avons dit précédemment, le vecteur d'attributs provient du tableau descriptif de la primitive.

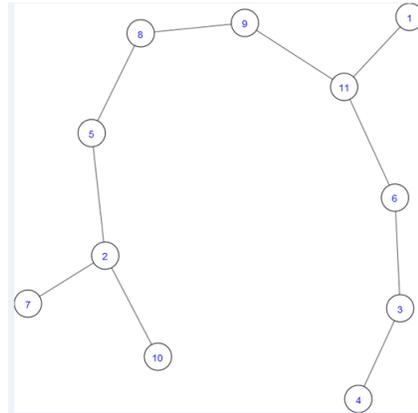


Figure 4.16 – Graphe relationnel, au sens 4.1.8, représentant les ellipses de la figure 4.7e.

Cet exemple a été choisi pour illustrer la construction du graphe relationnel car la disposition des ellipses dans l'image 4.7e, permet de comprendre de façon aisée le processus de construction du graphe relationnel attribué.

La figure 4.17 montre quatre graphes attribués, correspondant aux deux ensembles de primitives (ellipse et segments de droite) présentés dans la figure 4.14.

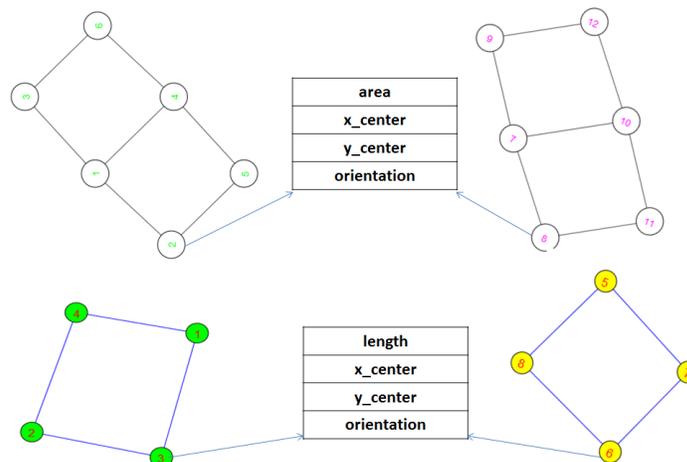


Figure 4.17 – Graphes attribués pour les deux ensembles de primitives de la figure 4.14. Première ligne : graphe correspondant à l'ensemble d'ellipses extraites de l'image de référence (gauche) ; graphe correspondant à l'ensemble d'ellipses extraites de l'image de l'objet à évaluer (droite). Deuxième ligne : graphe correspondant à l'ensemble des segments de droite extraits de l'image de référence (gauche) ; graphe correspondant à l'ensemble des segments de droite de l'image test (droite).

Les graphes attribués ont chacun leurs nœuds associés à un vecteur d'attributs. Dans la

figure 4.17, à titre d'exemple, cette association est matérialisée sur un nœud du graphe de référence et un autre nœud du graphe représentant une primitive de l'élément à évaluer.

4.3.4.2 Graphes attribués et mesure de similarité versus graphe topologique

L'objectif de l'approche présentée ici consiste à mettre en correspondance les primitives de deux images avec pour but de constater ou non la présence d'un élément, son bon placement par rapport à l'image de référence mais aussi certifier que les dimensions des objets sont adéquates. C'est pour cette raison que les informations exploitées pour l'appariement sont celles se trouvant sur les nœuds, car les informations portées par le vecteur d'attributs associé aux nœuds permettent de rendre compte des dimensions d'objets. Dans une application de recherche de similarité entre deux objets au sens reconnaissance d'objet, l'information de proximité portée par les arêtes (information d'ordre topologique) aurait pu être exploitée également, comme dans [Bai and Latecki, 2008] où le squelette (la topologie) de deux animaux sont comparés au dépens de leur dimensions pour effectuer la reconnaissance. Dans nos travaux il aurait été possible d'utiliser aussi l'information topologique en procédant en deux temps. Par exemple certifier, que la topologie des objets est conforme avant d'effectuer des calculs supplémentaires pour certifier que le placement et les dimensions des objets sont cohérents. Dans notre approche, les vecteurs d'attributs choisis permettent une analyse aisée et plus complète que l'exploitation seule de la topologie d'objet. Mais se pose à l'avenir la question de comment combiner intelligemment et les informations portées par les nœuds du graphe attribué et les informations portées par ses arêtes.

L'utilisation d'un graphe attribué n'est d'ailleurs pas une obligation au sens où les vecteurs d'attributs peuvent être exploités sans passer par les graphes attribués mais l'utilisation de ce type de graphe permet une formulation élégante du problème et le graphe attribué comme le dit Bengoetxea, [Bengoetxea, 2002] est un moyen efficace de représenter les objets, standardisé et facile à manipuler. De plus, on peut assigner aux nœuds et aux arêtes autant d'attributs que l'on souhaite.

4.3.4.3 Fonction de ressemblance

Dans la section 4.1.11 nous avons expliqué pourquoi cela n'est pas toujours possible d'avoir recours à des techniques classiques d'appariement de graphes. L'utilisation d'une fonction de mesure de similarité entre les deux graphes peut dans de telles circonstances être utilisée comme une alternative à l'appariement classique.

Nous allons de ce fait utiliser une fonction de mesure pour apparier les graphes attribués que nous avons construits dans la section 4.3.4.1.

La fonction de ressemblance est calculée de la façon suivante. On commence par diviser les attributs en deux classes ($c = [c_1, c_2]$). Les attributs tels que l'aire de l'ellipse ou la

longueur de segment de droite appartiennent à la classe c_1 car il est possible de calculer le rapport "attribut de référence" vs "attribut de l'élément à évaluer" cf. eq. (4.10). En revanche, cela n'a pas de sens de calculer le ratio pour des attributs tels que les coordonnées du centre (ellipse ou segment) ou l'orientation. En conséquence, on dit que ces attributs appartiennent à la classe c_2 . Pour ces attributs, le score de ressemblance est calculé en utilisant la différence en valeur absolue de l'attribut appartenant à la primitive de référence et à la primitive test, cf. eq. (4.11). Ce calcul incorpore l'écart maximum δ_i accepté entre les deux attributs (référence et test), afin de normaliser le calcul du score pour les attributs de la classe c_2 .

En résumé, pour les ellipses, la classe c_1 contient l'aire de l'ellipse aussi bien pour l'ellipse de l'objet de référence que l'aire de ellipse de l'élément à tester. Pour ces attributs appartenant à la classe c_1 le ratio est calculé et normalisé pour être compris entre $[0, 1]$. La classe c_2 contient l'orientation de l'ellipse ainsi que son centre pour les deux types de primitives (référence/test). Les centres sont utilisés pour calculer la distance euclidienne entre les deux primitives. Le score de similarité obtenu pour les attributs de la classe c_2 est dans l'intervalle $[-\infty, 1]$. Pour l'appariement, seuls les scores dans l'intervalle $[0, 1]$, sont conservés, aussi bien pour la classe c_1 que pour la classe c_2 . Considérons maintenant le cas des segments de droite : dans ce cas, la classe c_1 contient les longueurs des deux segments de droite et la classe c_2 contient les deux centres des segments de droite ainsi que les deux orientations. Le score de ressemblance entre une primitive de l'élément de référence et une primitive de l'élément test est la somme des tous les scores pour chacun des attributs pris individuellement. Cette fonction de ressemblance a été proposé par McIntosh et Mutch [McIntosh and Mutch, 1988] pour l'appariement de segments de droite. Nous étendons son usage aux cas des ellipses.

$$s_i = \begin{cases} \frac{\min(X_{k_i}, Y_{m_i})}{\max(X_{k_i}, Y_{m_i})}, & \text{if } i \in c_1 \\ \frac{\delta_i - \text{abs}(X_{k_i} - Y_{m_i})}{\delta_i}, & \text{if } i \in c_2 \end{cases} \quad (4.10)$$

$$s_i = \begin{cases} \frac{\min(X_{k_i}, Y_{m_i})}{\max(X_{k_i}, Y_{m_i})}, & \text{if } i \in c_1 \\ \frac{\delta_i - \text{abs}(X_{k_i} - Y_{m_i})}{\delta_i}, & \text{if } i \in c_2 \end{cases} \quad (4.11)$$

$$s(X_k, Y_m) = MF(X_k, Y_m) = \sum_i^N s_i w_i; \quad (4.12)$$

où X_k représente une primitive extraite de l'image de référence tandis que Y_m représente une primitive de l'image test et i est l'*i*ème attribut pour une primitive donnée. δ_i est l'écart maximal pour l'*i*ème attribut de la classe c_2 . Le scalaire s_i est le score de ressemblance entre une primitive de référence et une primitive de l'élément à évaluer, pour *i*ème

attribut. Puisqu'il est possible d'attribuer plus d'importance à un paramètre plutôt qu'à un autre, [Roberto M. Cesar-Jr. et al., 2005] une fonction qui définit cette importance peut être requise. L'approche décrite dans ce manuscrit utilise la fonction de pondération de McIntosh et Mutch, [McIntosh and Mutch, 1988]. Celle-ci est représentée dans l'équation (4.12) par w_i . Pour chaque attribut pour lequel il faut calculer un score de ressemblance référence/test, une priorité P_i est définie, ayant une valeur entre 1 et 10. Une variable $Q_i = 11 - P_i$ est alors créée. Ensuite, on définit $T = \sum Q_i$ et $w_i = \frac{Q_i}{T}$. Le point important dans cette fonction de pondération est de savoir comment définir P_i . Donner une faible valeur de priorité à une variable, revient à lui accorder un fort poids de pondération. Dans le cas où il n'est pas possible de définir quel poids donner à chacun des attributs, il est alors convenable de leur donner à tous un poids w_i égal. Cette technique de McIntosh et Mutch garantit une valeur de pondération toujours dans l'intervalle $[0, 1]$.

Enfin, $s(X_k, Y_m)$ est le score de mesure de ressemblance entre X_k et Y_m .

L'algorithme de mise en correspondance est réalisé en parcourant tous les nœuds du graphe de référence et en cherchant son meilleur vis-à-vis dans le graphe représentant les primitives de l'élément à évaluer. Le critère de similarité utilisé pendant ce processus est la fonction de mesure qui a été préalablement décrite dans l'équation (4.12). Pendant ce processus de mise en correspondance une matrice d'appariement est construite. Cette matrice contient le score de similarité pour chaque élément de l'ensemble de référence X avec une primitive de l'ensemble Y (cf. tableau 4.9). Cette matrice d'appariement est très importante car c'est elle qui sera injectée dans le graphe biparti qui le cas échéant est optimisé afin de respecter la contrainte d'unicité de la mise en correspondance.

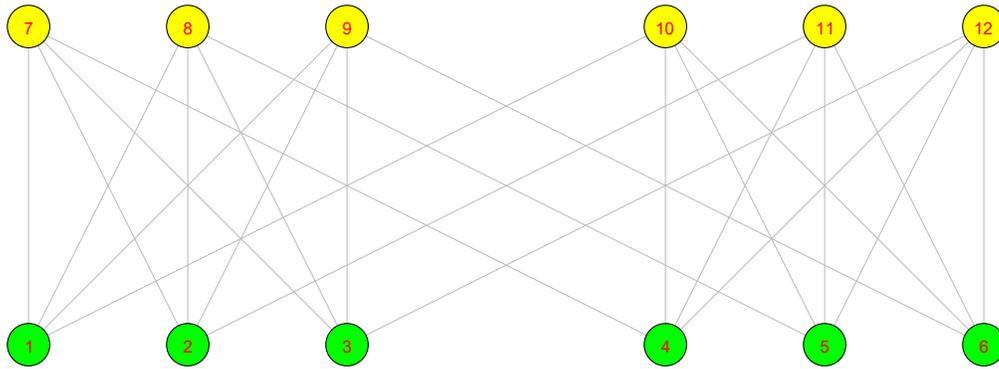
Tableau d'appariement d'ellipses					
0.97	0.37	0.36	0.16	-0.18	-0.18
0.37	0.97	0.27	-0.16	0.15	-0.10
0.36	0.28	0.98	-0.17	-0.12	0.17
0.17	-0.17	-0.17	0.98	0.38	0.36
-0.17	0.15	-0.11	0.37	0.97	0.27
-0.18	-0.12	0.15	0.36	0.27	0.97

Tableau 4.9 – Tableau d'appariement entre les ellipses de l'image test et celles de l'image de référence, montrées dans la figure 4.14.

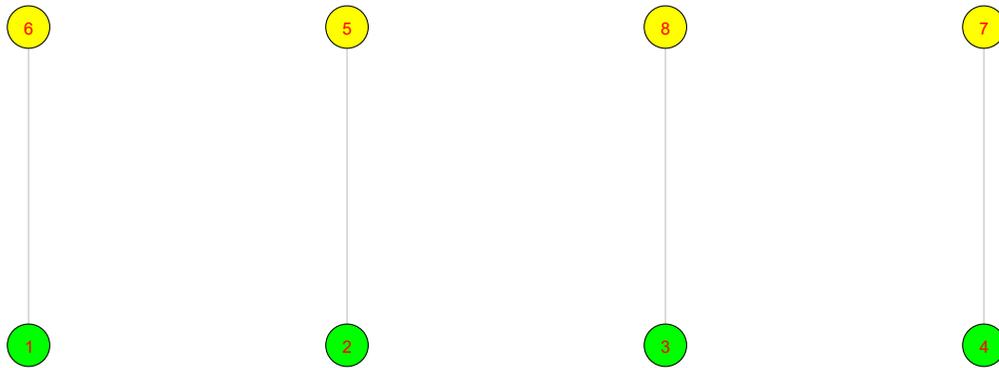
Tableau d'appariement de segments de droite			
-0.27	0.96	-0.84	-0.85
0.83	-0.33	-0.76	-0.72
-0.83	-0.88	-0.95	0.97
-0.69	-0.80	0.82	-1.04

Tableau 4.10 – Tableau d'appariement entre les segments de droite de l'image test et ceux de l'image de référence, montrées dans la figure 4.14.

Les deux tableaux d'appariement 4.9 et 4.10 conduisent aux graphes suivants :



(a) Appariement d'ellipses



(b) Appariement de segments de droite

Figure 4.18 – Graphe biparti montrant les candidats à l'appariement final pour le cas présenté dans la figure 4.14

Comme on le constate sur la figure 4.18, on doit sélectionner les meilleurs candidats afin de poursuivre vers l'inspection, car l'un des pré-requis de la mise en correspondance est de satisfaire la contrainte d'unicité. La section suivante explique comment cette recherche est effectuée via le graphe biparti.

4.3.5 Recherche du meilleur couple d'appariement à l'aide d'un graphe biparti

L'appariement de graphe biparti est un sujet connu avec une littérature abondante. En effet, le graphe biparti peut être utilisé dans différentes branches de la science notamment dans l'exploration de données "data mining", [Zha et al., 2001], dans les mathématiques [Panda and Pradhan, 2013], ou encore en vision artificielle [Fishkel et al., 2006, Shokoufandeh and Dickinson, 1999, Wang and Ma, 2014].

Le terme graphe biparti rend évident l'utilité de cette représentation dans un problème de

deux classes. Le graphe biparti est utilisé dans notre méthode de la façon suivante : En premier lieu, il s'agit de compter le nombre d'occurrences M qu'un score présent dans la matrice d'appariement satisfait un critère de seuillage préalablement défini. Ensuite un graphe biparti dirigé comportant M arêtes est créé. Ce graphe biparti porte en lui deux types de nœuds (X, Y), représentant les primitives de l'image de référence et celles de l'image test.

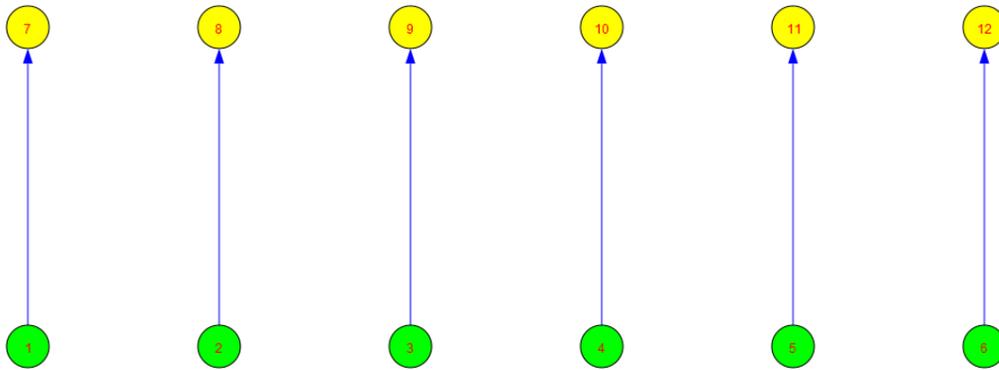
Un nœud de type X est incident à un nœud de type Y lors que le score de similarité entre les primitives représentées par ces nœuds satisfait la condition de seuillage prédéfinie. L'arête incidente à ces deux nœuds est pondérée par le score d'appariement correspondant. Sachant que le score d'appariement est compris dans l'intervalle $[0, 1]$, il en va de même pour la valeur de seuil choisie. Cette valeur doit être réglée en fonction du niveau de ressemblance voulu par l'utilisateur dans le cadre de son application. Par exemple, dans une application de contrôle, un utilisateur "A" décide que son élément à contrôler n'est acceptable que si le score d'appariement des primitives de son objet est quasi parfait (ex. score = 0.95). Pour cette application il est évident que la valeur du seuil peut atteindre une valeur numérique élevée, mais elle doit être inférieure à 0.95. En revanche s'il y a des primitives pour lesquelles un score plus faible peut être accepté (0.75 par exemple), alors il faut que la valeur du seuil soit suffisamment faible (inférieur à 0.75) pour ne pas filtrer de potentiels bons candidats à l'appariement final. On comprend alors que l'utilisation du seuil est optionnel. En effet si le seuil est fixé à zéro, le graphe biparti sera plus volumineux (beaucoup plus d'arêtes) car chaque nœud source (de référence) sera adjacent à un nœud cible (test). Dans le graphe biparti, un nœud de type X -type peut être adjacent à un ou plusieurs nœuds de type Y et réciproquement. En ce sens, une étape d'optimisation peut être nécessaire afin de supprimer les candidats ayant un score moins bon et garder uniquement les appariements pour lesquels le score de ressemblance est le plus élevé pour la paire référence-test.

En cas de multiples candidats à l'appariement on procède ainsi :

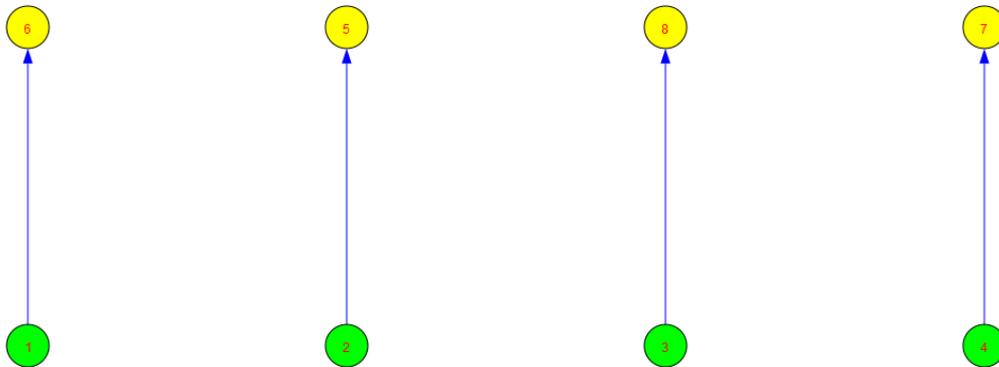
Premièrement, une recherche des meilleures scores dans le sens X vers Y est effectué. Celle-ci se déroule de la façon suivante. Soit k un nœud de type X , si le degré sortant de k est 1, alors l'arête reliant k à un nœud de type Y doit être sélectionnée pour la liste des bons candidats. En revanche, si le degré sortant de k est supérieur à 1, alors on effectue la recherche de l'arête ayant le meilleur score parmi toutes celles qui sortent du nœud k . Cette arête est alors sélectionnée pour la liste des bons candidats et les autres sont empilées dans la liste d'arêtes à élaguer. Puisque le graphe est orienté, cette première façon de procéder peut être considérée comme une recherche du meilleur score dans le sens source-cible. À la fin de ce processus s'il y a toujours plus d'une correspondance pour un quelconque nœud de type Y , alors on effectue la deuxième étape. Dans le cas contraire, la

mise en correspondance est terminée.

Lorsque qu'il y a des cas de conflits, la seconde étape de la recherche du meilleur score à appairer consiste en la recherche du meilleur score dans le sens cible-source, uniquement pour ces cas où il y a un conflit, en utilisant cette fois-ci le degré entrant des nœuds cibles. Seuls les meilleurs scores dans les deux sens sont à conserver, et toutes les autres arêtes sont éliminées du graphe. Cette recherche des meilleurs candidats à appairer permet de satisfaire la contrainte d'unicité qui doit être vérifiée dans une telle mise en correspondance. Pour le cas de la figure 4.18, après la recherche des meilleurs couples d'appariements, nous obtenons les deux graphes 4.19a et 4.19b suivants, pour les segments de droite et pour les ellipses respectivement.



(a) Appariement d'ellipses



(b) Appariement de segments de droite

Figure 4.19 – Graphe biparti après optimisation des graphes de la figure 4.18 pour respecter la contrainte d'unicité

À partir du moment où la contrainte d'unicité d'appariement est respectée, il devient alors possible de combiner les résultats d'appariement d'ellipses avec ceux des segments de droite afin d'obtenir un score global pour l'ensemble de l'objet à évaluer.

4.4 Prise de décision

Avant de combiner les résultats des différents appariements pour les ellipses et les segments de droite, il est important de rappeler que les images de contour utilisées en entrée du processus subissent un recalage rigide avant la phase d'appariement. En conséquence, les paramètres de la transformation rigide (angle de rotation et vecteur de translation) subie par les contours doivent être pris en compte dans le processus de décision. La prise en compte de ces valeurs trouvées lors du recalage doit être flexible car les tolérances données par les utilisateurs peuvent changer d'un utilisateur à un autre. Par exemple, un utilisateur 'A' peut décider qu'un élément de son assemblage n'est pas défectueux s'il est décalé d'une certaine valeur, qui sera refusée par un autre utilisateur 'B'. On envisage donc qu'en entrée l'utilisateur de l'application fournisse les tolérances qui pour lui sont acceptables et que l'algorithme compare ces tolérances avec les valeurs qu'il aura trouvé (cf. **chapitre 5**) afin de statuer sur la conformité de chaque objet à inspecter.

Comme il a été dit précédemment, lorsque la contrainte d'unicité de l'appariement est respectée pour les segments de droite et pour les ellipses il est possible de combiner les deux résultats de la mise en correspondance afin d'obtenir un seul score global de ressemblance entre l'objet de référence et l'objet à évaluer. Cela est effectué comme suit :

$$s_{e,l} = \frac{1}{2} \left[\frac{1}{P} \sum_{i=1}^{M_1} e_i + \frac{1}{L} \sum_{j=1}^{M_2} l_j \right] \quad (4.13)$$

Dans l'équation (4.13), $s_{e,l}$ représente le score global, prenant en considération l'ellipse e et le segment de droite l . P représente le nombre d'ellipses du modèle, tandis que L désigne le nombre de segments de droite du modèle. M_1 et M_2 sont les nombres d'ellipses et de segments de droite théoriques qui ont pu être appariés avec leur correspondant de l'image test.

e_i et l_j sont les scores respectifs pour l' i ème ellipse et le j ème segment de droite de l'image de référence. La définition d'un score global facilite la prise de décision. Si l'on prend en considération la matrice d'appariement pour les ellipses, cf. tableau 4.9 et celle des segments de droite, cf. tableau 4.10, le score global pour le cas test de la figure 4.9 est le suivant :

$$score_1 = \frac{1}{2} \left[\frac{0.980 + 0.971 + 0.980 + 0.983 + 0.973 + 0.975}{6} + \frac{0.969 + 0.863 + 0.859 + 0.976}{4} \right] = 0.9469$$

Nous avons expliqué pourquoi la prise en compte des paramètres de décision doit être faite de manière flexible, de sorte à permettre à l'application d'être utilisée par différents utilisateurs. Ainsi, le score global et les paramètres de la transformation rigide entre l'élément de référence et l'élément à évaluer (cf. matrice d'appariement (4.5)) sont les suivants :

- ◇ score : 0.9469
- ◇ translation : $t_{x_1} = 127.804 \text{ pixels} \rightarrow 25.4 \text{ mm}$, $t_{y_1} = 65.295 \text{ pixels} \rightarrow 13.0 \text{ mm}$
(taille de l'objet : $46 \text{ mm} \times 30 \text{ mm} \rightarrow 231 \text{ pixels} \times 150 \text{ pixels}$)
- ◇ angle de rotation : $\theta_1 = 0.2^\circ$.

Ces informations constituent un vecteur d'aide à la décision, où le score est renforcé par les paramètres de la transformation rigide et l'ensemble est à comparer avec les valeurs pré-définies par l'utilisateur. Si ces valeurs sont dans l'intervalle de tolérance fixé préalablement, alors l'élément à inspecter est dit conforme.

4.5 Conclusion

Dans ce chapitre nous avons vu comment le problème d'inspection peut être transformé en un problème de recherche opérationnelle pour en faciliter la résolution. Nous avons commencé le chapitre en introduisant les concepts de base de la théorie des graphes sur lesquels nous nous sommes appuyés pour effectuer l'appariement entre les primitives extraites d'une image de référence et celles d'une image test. Nous avons expliqué pourquoi il est nécessaire d'aligner les deux entités, référence et test à apparier, avant d'en extraire les primitives. Nous avons aussi justifié notre choix d'utiliser une fonction de mesure de similarité pour résoudre le problème d'appariement de graphes, au détriment d'une approche plus conventionnelle exploitant l'isomorphisme de graphe ou la recherche de sous-graphes. Puisque l'appariement doit respecter la contrainte d'unicité, nous avons vu comment la combinaison d'un graphe biparti avec une validation croisée permet de remplir cette condition. Enfin, nous avons établi et décrit de manière simple une procédure permettant de décider de la conformité d'un élément inspecté (cf. section 4.4).

Dans le **chapitre 5** nous verrons comment ce mécanisme d'aide à la décision peut être utilisé afin de statuer sur la conformité ou non d'un élément.

5 Résultats et discussion

Le but de ce chapitre est de présenter les résultats de notre méthode décrite dans le **chapitre 4**, qui consiste à appairer les primitives d'une image de référence avec celles d'une image test qui représente l'objet à évaluer. Nous allons d'abord présenter les différents tests réalisés sur des images synthétiques pour valider la méthode. Ensuite nous présentons des résultats réalisés sur des cas réels.

Sommaire

5.1	Résultat d'appariement entre deux images synthétiques	92
5.1.1	Cas où l'image test a moins d'éléments que l'image de référence . . .	93
5.1.2	Cas où l'image test a plus d'éléments que l'image de référence . . .	100
5.1.3	Combinaison de translation et éléments manquants dans l'image test	103
5.1.4	Combinaison de zoom + nombre d'éléments différents	106
5.1.5	Remarque concernant les objets tests ayant subi une rotation	115
5.2	Résultats d'appariement entre une image synthétique et une image réelle .	115
5.2.1	Tests réalisés sur les données du projet AirCOBOT	115
5.3	Conclusion	122

5.1 Résultat d'appariement entre deux images synthétiques

La méthode de mise en correspondance d'une image de référence avec une image test décrite dans ce manuscrit a été testée avec des données synthétiques et a prouvé son efficacité. Elle est capable d'appairer correctement les segments de droite et les ellipses présents dans les deux images. Les tests ont été réalisés notamment dans des cas où il manque certains éléments dans l'image test et dans des cas où il y a des éléments en plus dans l'image test que ceux attendus dans l'image de référence. Les cas d'éléments ayant des dimensions différentes, des éléments translétés ou ayant subi une rotation ont été également testés. Il en va de même pour la combinaison de ces cas. Dans l'ensemble des tests réalisés, le résultat d'appariement et d'inspection est satisfaisant.

Outre le cas test présenté dans la figure 4.9, représentant l'objet 4.7b pour illustrer l'approche, les sections suivantes présentent les résultats des tests réalisés en considérant d'autres cas de figure.

Nous avons présenté les tests avec un niveau de complexité croissant. Au départ nous allons montrer des cas où l'objet de référence et l'objet test sont au départ le même objet mais avec une modification mineure de l'objet test par rapport à l'objet de référence. La modification en question pourra être le manque d'une ou plusieurs primitives dans l'objet à évaluer et inversement nous allons montrer des cas où l'objet test a plus de primitives que l'objet de référence.

Ensuite, nous allons combiner les cas de figures susmentionnés avec une translation. Pour finir, afin de se rapprocher le plus possible de la réalité, nous allons présenter un ensemble de cas où l'objet test est différent de l'objet de référence et peut de ce fait avoir plus ou moins de primitives que l'objet de référence en plus d'avoir une taille différente. Ces tests simulent un effet de zoom.

5.1.1 Cas où l'image test a moins d'éléments que l'image de référence

5.1.1.1 Cas n° 1

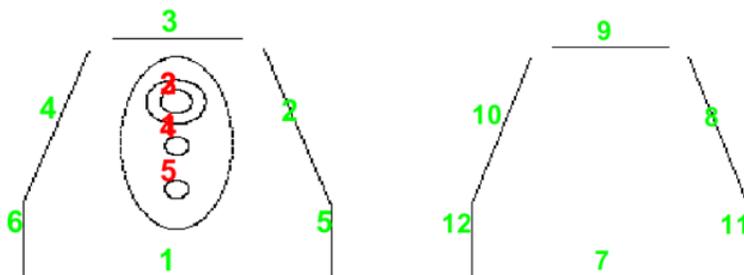


Figure 5.1 – Ensemble des primitives à appairer pour le cas de l'objet 4.7c versus objet 4.7c modifié

Tableau d'appariement de segments de droite					
1.00	-0.41	-0.54	-0.42	-0.25	-0.23
-0.41	1.00	-0.18	-0.36	-0.16	-0.82
-0.54	-0.18	1.00	-0.17	-0.72	-0.70
-0.42	-0.36	-0.17	1.00	-0.83	-0.15
-0.25	-0.16	-0.72	-0.83	1.00	-0.31
-0.23	-0.82	-0.70	-0.15	-0.31	1.00

Tableau 5.1 – Tableau d'appariement de segments de droite pour le couple référence-test de la figure 5.1

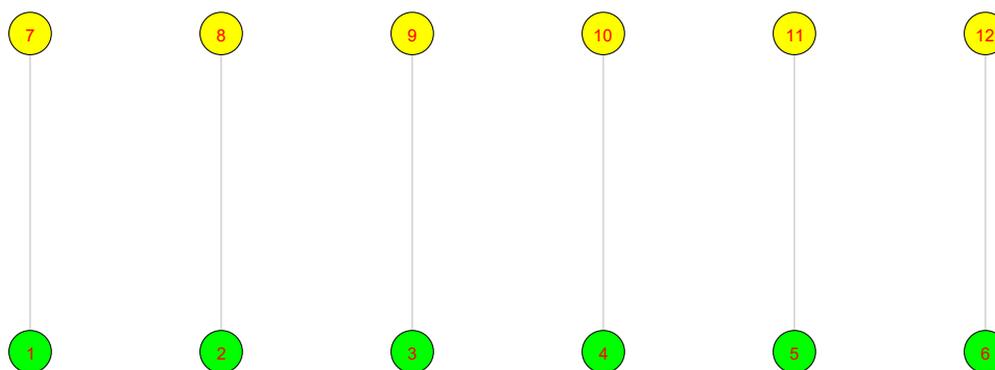


Figure 5.2 – Graphe biparti correspondant à l'appariement des primitives du cas test 4.7c versus test 4.7c modifié

Pour cet élément, le résultat rapporté est le suivant :

- ◇ score : 0.5.
- ◇ translation en x : 0 mm, en y : 0 mm.
- ◇ angle de rotation : 0° .

Ce test présente un cas de figure simple d'un support qui devrait contenir un autre petit support (contient des trous) mais qui n'est pas présent sur l'élément. Il y a un score parfait pour les segments de droite mais tous les éléments à l'intérieur du support, représentés par des ellipses sur l'objet de référence (cf. figure 5.1) ne sont pas présents. Aussi il n'y que la moitié des éléments qui participent au score. Les éléments d'aide à la décision cf. section 5.1.1.1 ci-dessus permettent ensuite moyennant le réglage d'un seuil (cf. section 4.4) de trancher sur la conformité de la pièce. En plus de donner un score, l'information consistant à dire qu'aucune ellipse n'a été trouvée, participe aussi à la prise de décision.

5.1.1.2 Cas n ° 2

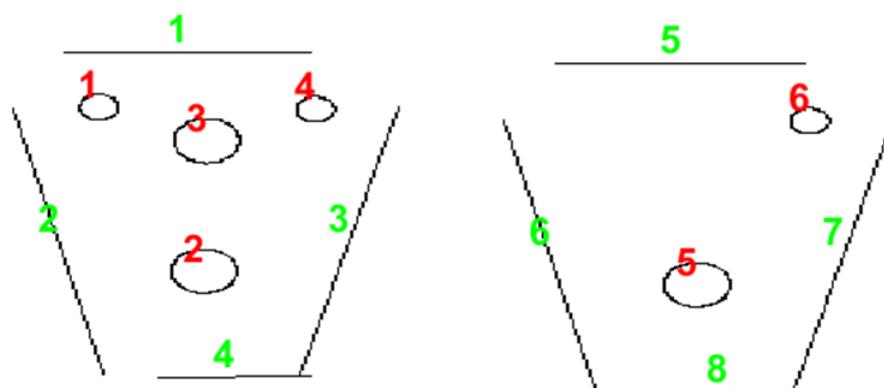


Figure 5.3 – Ensemble des primitives à appairer pour le cas de l'objet 4.7d versus objet 4.7d modifié

Tableau d'appariement de segments de droite

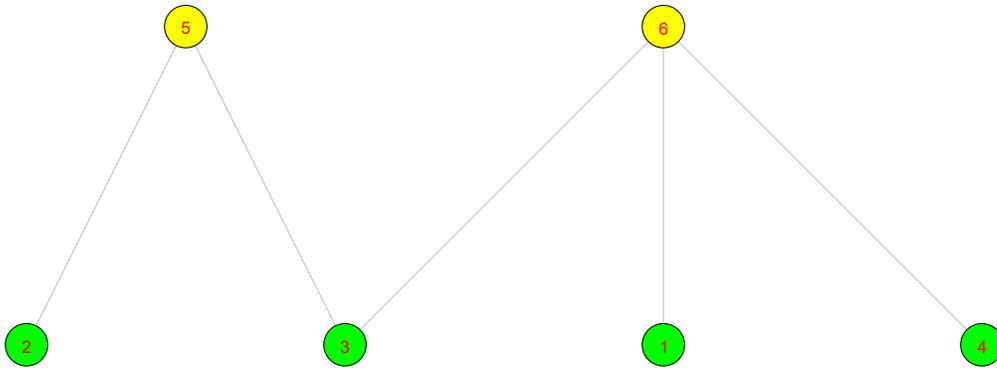
1.00	-0.41	-0.46	-0.65
-0.41	1.00	-0.23	-0.36
-0.46	-0.23	1.00	-0.25
-0.65	-0.36	-0.25	1.00

Tableau 5.2 – Tableau d'appariement de segments de droite pour le couple référence-test de la figure 5.3

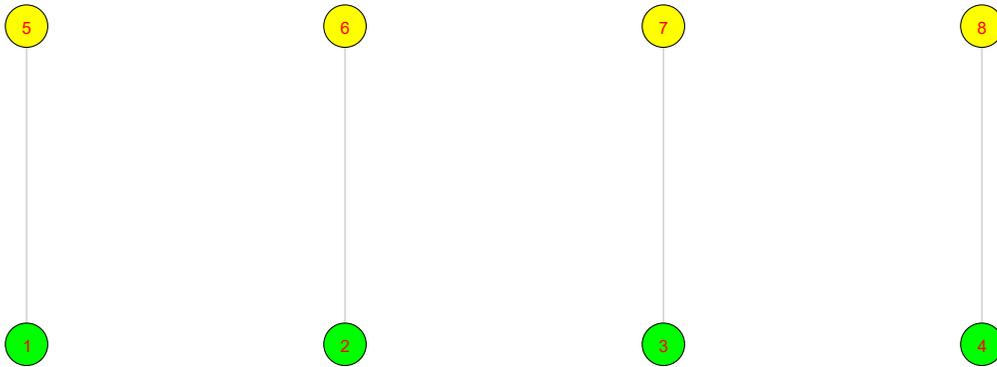
Tableau d'appariement d'ellipses

-0.11	0.29
1.00	-0.07
0.37	0.40
-0.07	1.00

Tableau 5.3 – Tableau d'appariement d'ellipses pour le couple référence-test de la figure 5.3



(a) Appariement d'ellipses

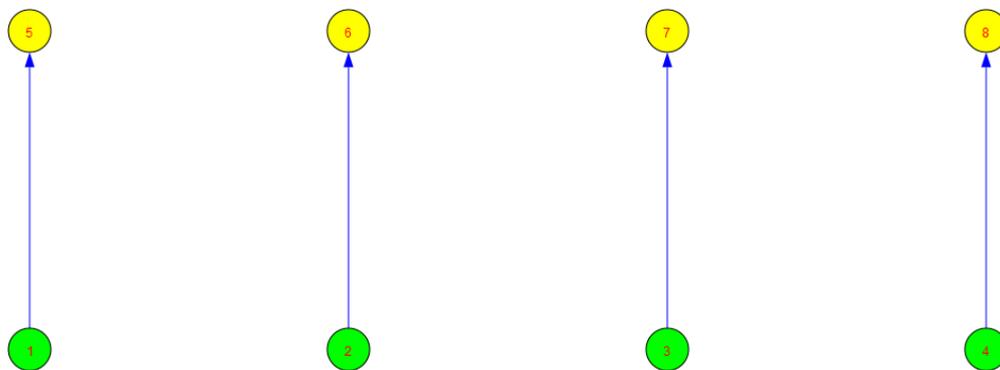


(b) Appariement de segments de droite

Figure 5.4 – Graphe biparti correspondant à l'appariement des primitives du cas test 4.7d versus 4.7d modifié



(a) Appariement d'ellipses



(b) Appariement de segments de droite

Figure 5.5 – Graphe biparti après optimisation des graphes de la figure 5.4 pour respecter la contrainte d'unicité

À la fin de la procédure, le résultat rapporté est le suivant :

- ◇ score : 0.75.
- ◇ translation en x : 0 mm, en y : 0 mm.
- ◇ angle de rotation : 0° .

Ce cas test de la figure 5.3 est un cas proche de celui de la figure 5.1. Il manque des éléments de fixation du support matérialisé par l'absence de quelques ellipses. Comme dans le cas précédent, on remonte les éléments d'aide à la décision et le nombre d'éléments manquants.

5.1.1.3 Cas n ° 3

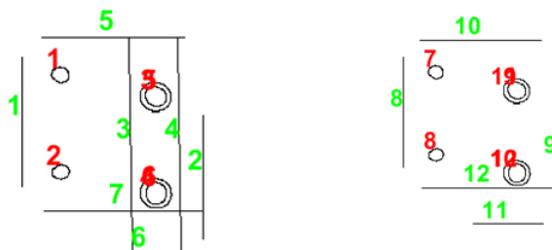


Figure 5.6 – Ensemble des primitives à appairer pour le cas de l'objet 4.7a versus objet 4.7a modifié

Tableau d'appariement de segments de droite

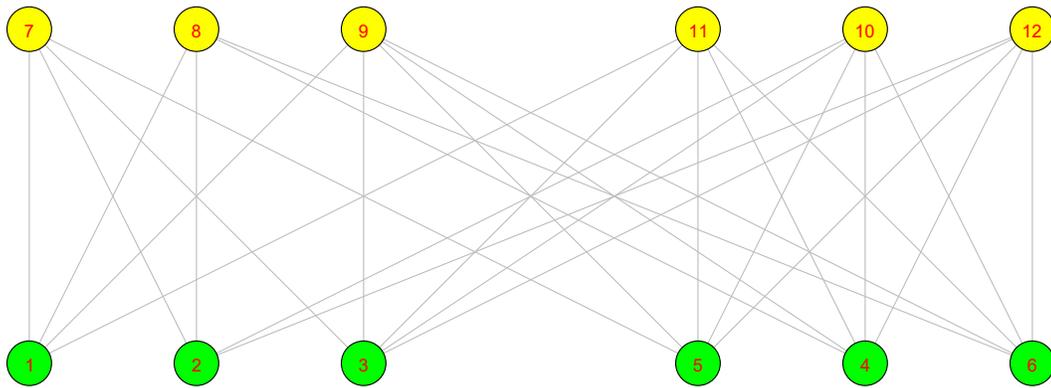
1.00	-0.33	-0.23	-0.71	-0.33
-0.33	1.00	-0.53	-0.10	-0.01
0.06	0.28	-0.21	-0.29	0.12
-0.27	0.55	-0.33	-0.32	-0.02
-0.23	-0.53	1.00	-0.66	-0.24
-0.71	-0.10	-0.66	1.00	0.51
-0.33	-0.01	-0.24	0.51	1.00

Tableau 5.4 – Tableau d'appariement de segments de droite pour le couple référence-test de la figure 5.6

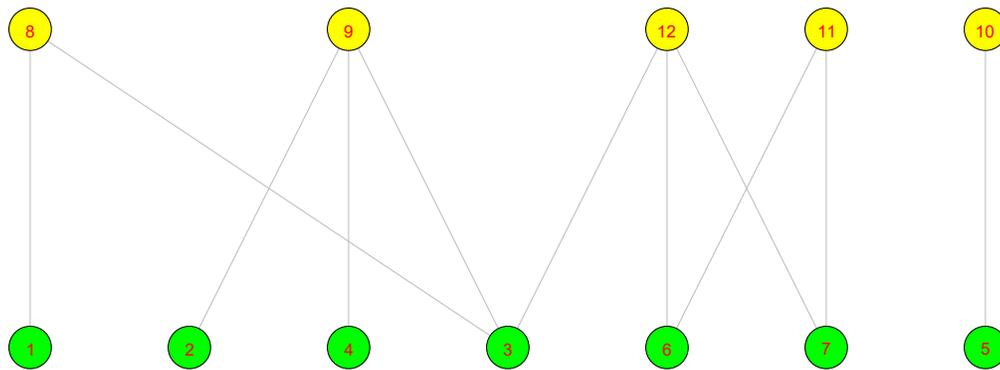
Tableau d'appariement d'ellipses

1.00	0.31	0.08	-0.30	0.17	-0.25
0.31	1.00	-0.10	0.06	-0.01	0.14
0.08	-0.10	1.00	0.33	0.81	0.08
-0.30	0.06	0.33	1.00	0.14	0.75
0.17	-0.01	0.81	0.14	1.00	0.24
-0.25	0.14	0.08	0.75	0.24	1.00

Tableau 5.5 – Tableau d'appariement d'ellipses pour le couple référence-test de la figure 5.6

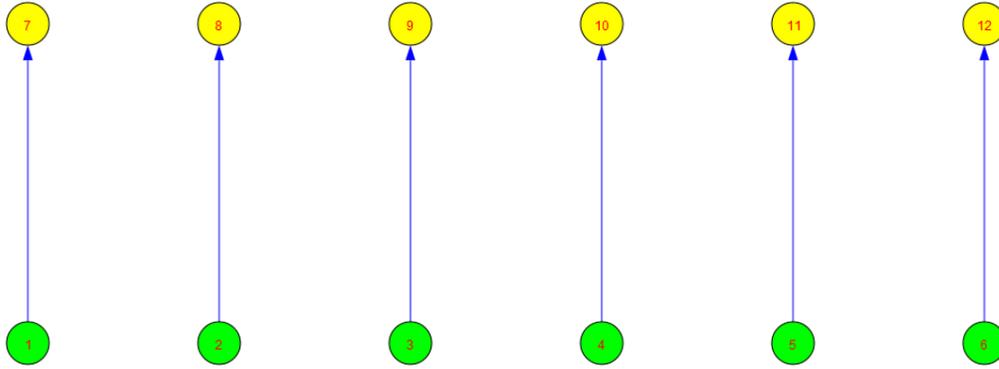


(a) Appariement d'ellipses

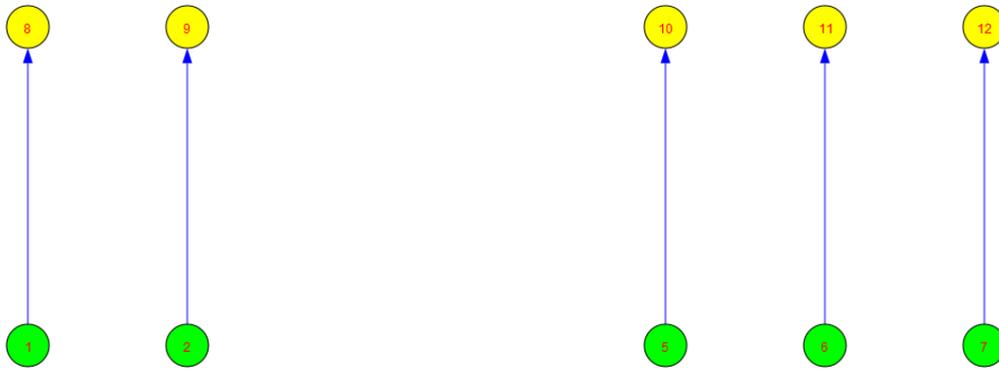


(b) Appariement de segments de droite

Figure 5.7 – Graphe biparti correspondant à l'appariement des primitives du cas au cas test 4.7a versus 4.7a modifié



(a) Appariement d'ellipses



(b) Appariement de segments de droite

Figure 5.8 – Graphe biparti après optimisation des graphes de la figure 5.7 pour respecter la contrainte d'unicité

- ◇ score : 0.8571.
- ◇ translation en x : 0 mm, en y : 0 mm.
- ◇ angle de rotation : 0° .

Ci-dessus nous simulons un autre cas où l'objet test présente moins de primitives que l'objet de référence. Cette fois-ci une partie du support est coupée où n'a pas pu être détectée. Comme pour les autres cas, on remonte l'information d'aide à la décision ci-dessus (cf. section 5.1.1.3) et le nombre de primitives manquant par rapport à la référence. Sur l'ensemble des cas tests présentant des éléments manquants, le comportement de la méthode est cohérent. Nous allons voir d'autres configurations.

5.1.2 Cas où l'image test a plus d'éléments que l'image de référence

Considérons l'image test de la figure 5.9. Dans cet exemple, l'image de référence a 6 ellipses tandis que l'image test en a 10. L'une des ellipses de référence (couleur verte) peut être mise en correspondance avec plus d'une ellipse test (couleur magenta) comme le montre la figure 5.9, deuxième ligne.

5.1.2.1 Cas n ° 1

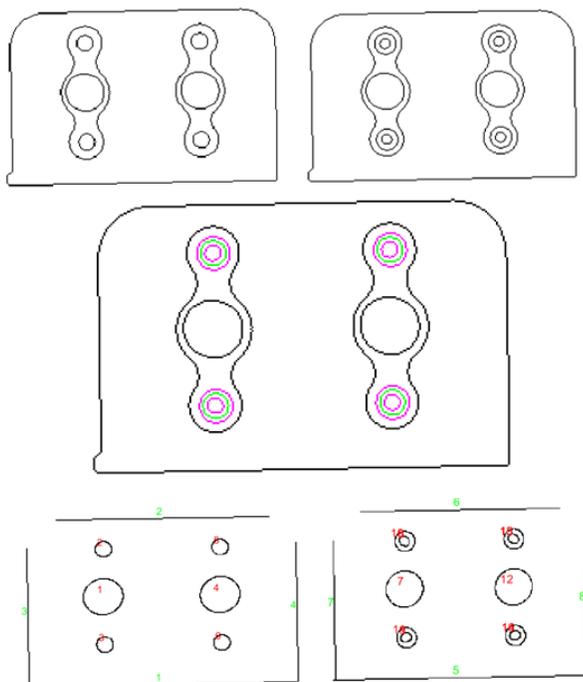


Figure 5.9 – Première ligne : contours des images de référence (gauche) et test(droite) avant recalage.
Deuxième ligne : Les deux contours recalés.
Troisième ligne : Primitives extraites des contours

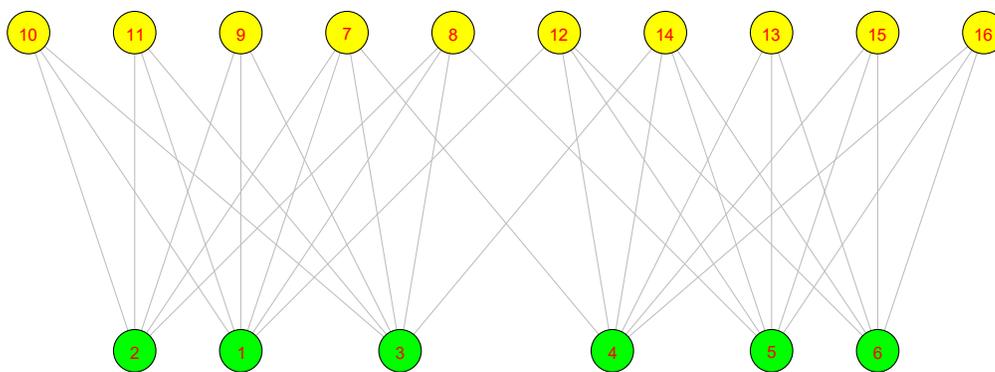
Tableau d'appariement de segments de droite

1.00	-0.32	-0.73	-0.76
-0.32	1.00	-0.87	-0.85
-0.73	-0.87	1.00	-0.91
-0.76	-0.85	-0.91	1.00

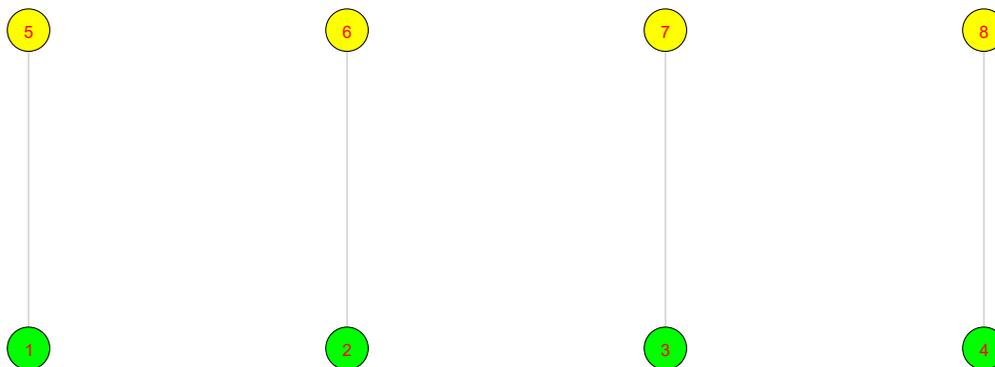
Tableau 5.6 – Tableau d'appariement de segments de droite pour le couple référence-test de la figure 5.9

Tableau d'appariement d'ellipses									
1.00	0.39	0.39	0.30	0.33	0.17	-0.17	-0.15	-0.24	-0.22
0.37	0.81	0.11	0.77	0.07	-0.17	-0.01	-0.26	-0.05	-0.30
0.36	0.10	0.83	0.06	0.78	-0.18	-0.28	0.00	-0.32	-0.04
0.17	-0.15	-0.15	-0.24	-0.21	1.00	0.39	0.40	0.30	0.33
-0.18	0.00	-0.27	-0.07	-0.31	0.37	0.82	0.13	0.76	0.08
-0.19	-0.26	-0.02	-0.34	-0.05	0.36	0.12	0.85	0.03	0.78

Tableau 5.7 – Tableau d'appariement d'ellipses pour le couple référence-test de la figure 5.9

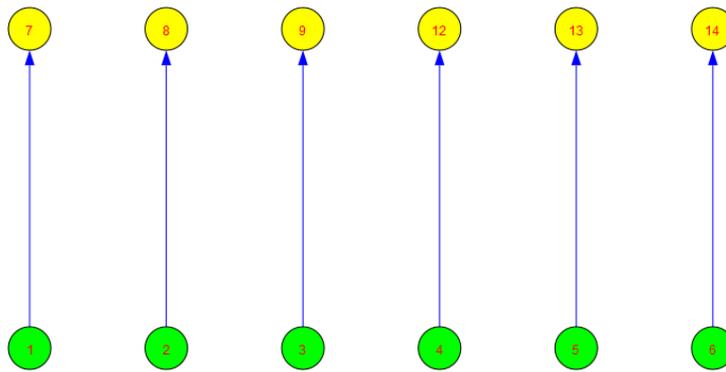


(a) Appariement d'ellipses

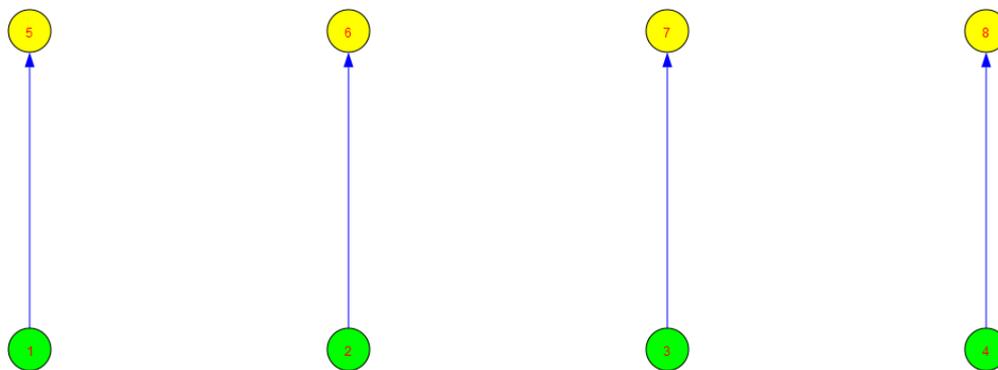


(b) Appariement de segments de droite

Figure 5.10 – Graphe biparti correspondant à l'appariement des primitives du cas au cas test 4.7b versus 4.7b modifié



(a) Appariement d'ellipses



(b) Appariement de segments de droite

Figure 5.11 – Graphe biparti après optimisation des graphes de la figure 5.10 pour respecter la contrainte d'unicité

La méthode a su trouver les bons appariements comme on peut le voir sur les graphes bipartis de la figure 5.11. Pour cette image, le résultat final rapporté par la méthode est le suivant :

- ◇ score : 0.94 . On remarque que le score (combinant l'appariement des ellipses et ceux des segments de droite) est élevé malgré le fait que l'image test a plus d'ellipses que l'image de référence. Cela est dû au fait que toutes les ellipses attendues dans le modèle se trouvent dans l'image test. Toutefois ayant connaissance du nombre d'ellipses attendues, en plus du score, la méthode lance également un avertissement qui précise que les appariements sont bons mais que 4 ellipses supplémentaires ont été trouvées dans l'image test.
- ◇ translation : $t_{x_2} \sim 0 \text{ mm}$, $t_{y_2} \sim 0 \text{ mm}$
- ◇ angle de rotation : $\theta_2 \sim 0^\circ$.

Ce résultat prend en compte la matrice d'appariement (5.7) pour les ellipses, et combine ces

résultats avec ceux de l'appariement des segments de droite (ici on a un score parfait pour les segments de droite). Il inclut également les valeurs des paramètres de la transformation rigide entre les deux contours mis en jeu. Ces valeurs de paramètres ont été obtenues en utilisant l'algorithme standard de la descente de gradient, minimisant l'écart quadratique moyen entre l'image de contour de l'objet test et l'image de contour de l'objet de référence.

5.1.3 Combinaison de translation et éléments manquants dans l'image test

5.1.3.1 Cas n° 1

Considérons maintenant les deux contours présentés dans la figure 5.12 ci-dessous. Dans cet exemple, l'image test est traduite et contient moins d'ellipses que l'image de référence. Le score est donc plus faible que dans le cas précédent. Les matrices d'appariement obtenues

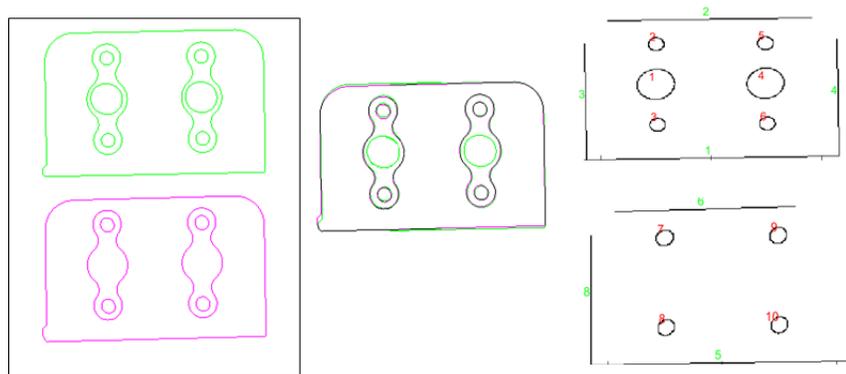


Figure 5.12 – Illustration d'un cas test contenant des éléments manquants. Première colonne : Avant le recalage, référence (vert) et test (magenta). Deuxième colonne : après le recalage. Troisième colonne : primitives extraites des deux contours après le recalage.

pour les deux ensembles de primitives, constitués de segments de droite et d'ellipses sont données par les deux tableaux 5.8 et 5.9 ci-après.

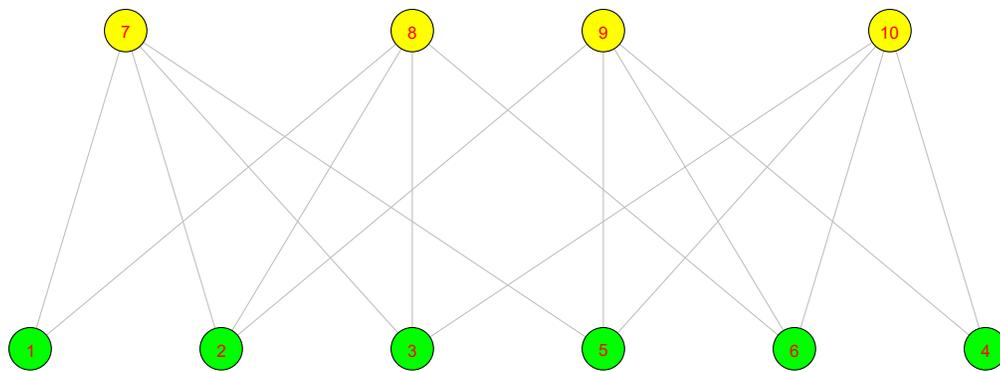
Tableau d'appariement de segments de droite			
0.99	-0.35	-0.76	-0.75
-0.30	0.84	-0.86	-0.86
-0.73	-0.76	-0.92	0.95
-0.76	-0.95	0.98	-0.96

Tableau 5.8 – Matrice d'appariement entre les segments de droite de l'image de référence et celles de l'image test (cf. figure 5.12).

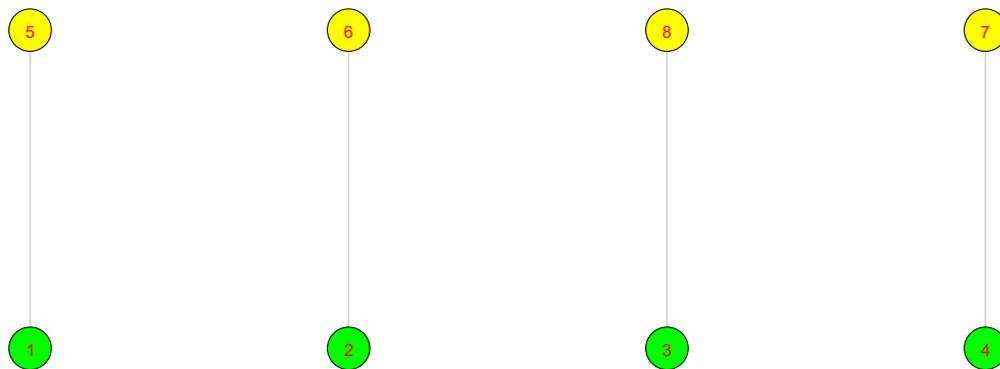
Tableau d'appariement d'ellipses

0.38	0.35	-0.19	-0.19
0.95	0.23	0.12	-0.14
0.26	0.95	-0.14	0.12
-0.17	-0.19	0.36	0.36
0.15	-0.13	0.96	0.25
-0.11	0.14	0.25	0.96

Tableau 5.9 – Matrice d'appariement entre les ellipses de l'image de référence et leurs correspondants de l'image de l'objet à inspecter, cf. Fig. 5.12.

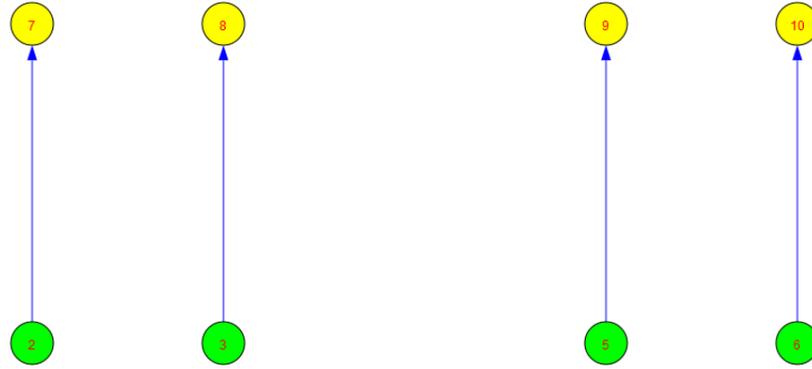


(a) Appariement d'ellipses

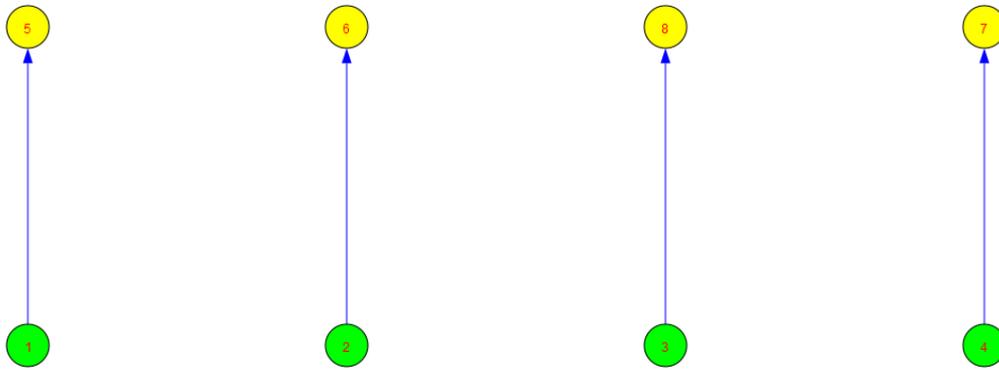


(b) Appariement de segments de droite

Figure 5.13 – Graphe biparti correspondant à l'appariement des primitives du cas test 4.7b versus 4.7b modifié



(a) Appariement d'ellipses



(b) Appariement de segments de droite

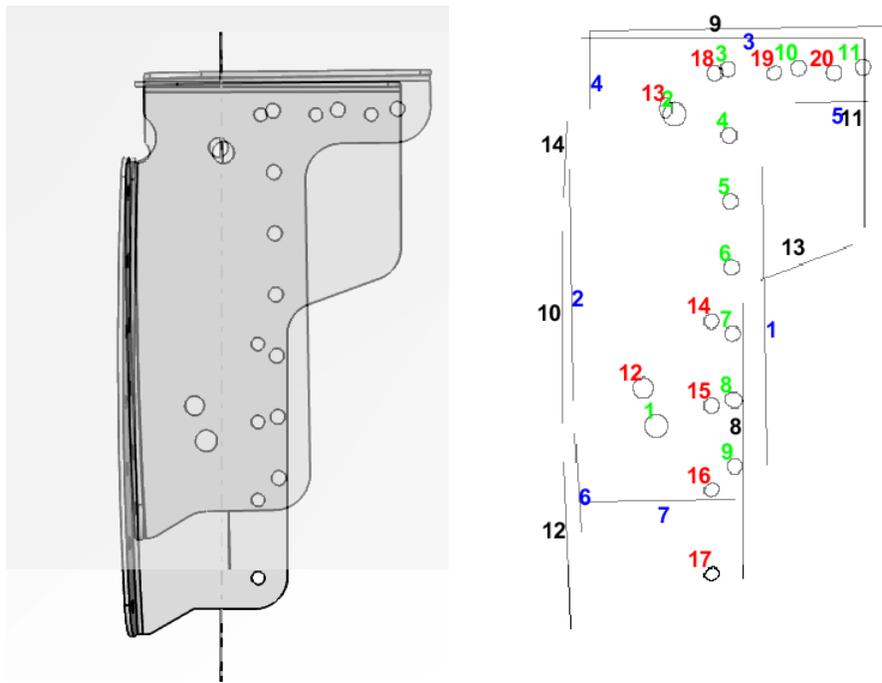
Figure 5.14 – Graphe biparti après optimisation des graphes de la figure 5.13 pour respecter la contrainte d'unicité

À la fin de la procédure, le résultat rapporté est le suivant :

- ◇ score : 0.7945. Il s'agit du score de ressemblance global, combinant les contributions des ellipses et des segments de droite. En plus de ce score, l'information concernant les éléments manquants est aussi rapportée.
- ◇ translation : $t_{x_2} \sim 1.7 \text{ px} \rightarrow 0.3 \text{ mm}$, $t_{y_2} \sim 172 \text{ px} \rightarrow 34.4 \text{ mm}$
- ◇ angle de rotation : $\theta_2 \sim 1^\circ$.

5.1.4 Combinaison de zoom + nombre d'éléments différents

5.1.4.1 Cas n ° 1



(a) Référence 4.7e versus test 4.7f

(b) Primitives de référence 4.7e superposée aux primitives test 4.7f

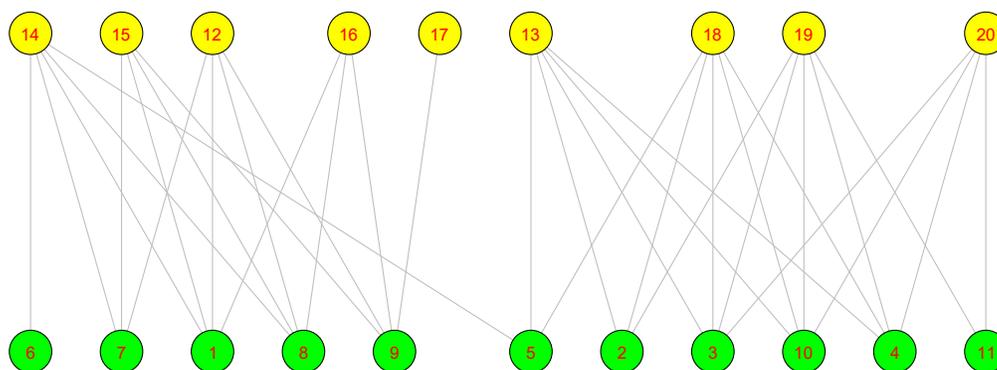
Figure 5.15 – Appariement de primitives en prenant comme référence 4.7e versus 4.7f

Tableau d'appariement de segments de droite						
0.13	-1.16	-0.46	-0.50	-1.15	0.01	-1.02
-0.58	-1.25	0.62	-1.22	-0.83	-1.05	-0.16
-2.03	0.79	-1.72	-0.50	-3.01	-1.13	-1.01
-1.83	-0.45	-0.87	-1.01	-2.34	-1.27	0.30
-1.84	-0.10	-1.81	0.17	-2.85	-0.47	-1.11
-0.44	-2.56	-0.29	-2.20	0.42	-1.38	-1.39
-0.15	-2.19	-0.67	-2.11	-0.00	-1.27	-1.79

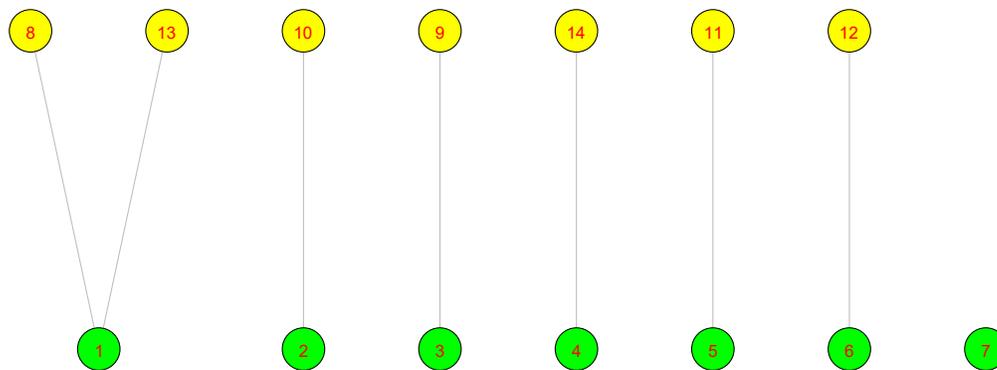
Tableau 5.10 – Tableau d'appariement pour le couple référence-test de la figure 5.15

Tableau d'appariement d'ellipses									
0.62	-1.25	0.01	0.36	0.22	-0.25	-1.52	-1.64	-1.77	
-0.93	0.76	-0.57	-1.18	-1.70	-2.24	0.39	0.03	-0.31	
-1.28	0.43	-0.77	-1.25	-1.81	-2.35	0.81	0.61	0.22	
-0.90	0.47	-0.31	-0.83	-1.36	-1.90	0.52	0.43	0.14	
-0.57	0.25	0.14	-0.47	-0.99	-1.54	0.06	-0.01	-0.18	
-0.21	-0.14	0.55	-0.04	-0.57	-1.11	-0.36	-0.42	-0.53	
0.12	-0.57	0.76	0.40	-0.12	-0.66	-0.76	-0.81	-0.89	
0.22	-0.98	0.37	0.74	0.30	-0.23	-1.18	-1.23	-1.29	
0.04	-1.43	-0.05	0.49	0.72	0.22	-1.60	-1.63	-1.68	
-1.53	0.08	-0.81	-1.39	-1.91	-2.44	0.36	0.71	0.66	
-1.70	-0.38	-1.00	-1.49	-1.97	-2.49	-0.02	0.35	0.75	

Tableau 5.11 – Tableau d'appariement pour le couple référence-test de la figure 5.15

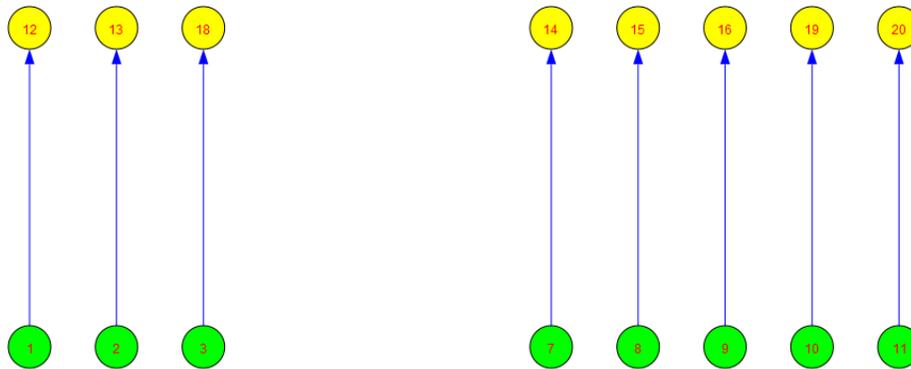


(a) Appariement d'ellipses

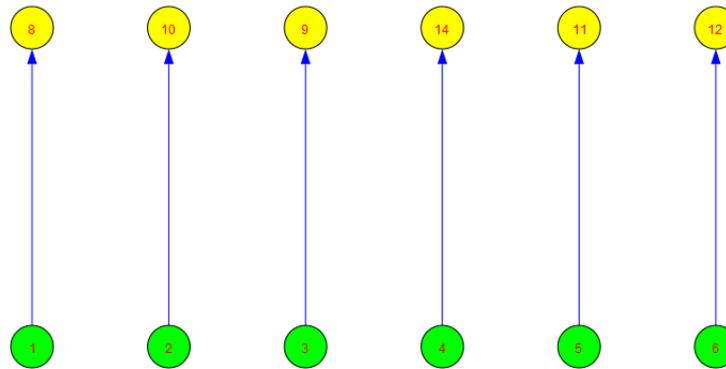


(b) Appariement de segments de droite

Figure 5.16 – Graphe biparti correspondant à l'appariement des primitives du cas test 4.7e versus 4.7f



(a) Appariement d'ellipses



(b) Appariement de segments de droite

Figure 5.17 – Graphe biparti après optimisation des graphes de la figure 5.16 pour respecter la contrainte d'unicité

À la fin de la procédure, le résultat rapporté est le suivant :

- ◇ score : 0.547.
- ◇ translation en x : 0 mm, en y : 0 mm.
- ◇ angle de rotation : 0° .

En observant les deux graphes biparti après l'optimisation pour satisfaire la contrainte d'unicité, on constate que, sur ce cas, qui pourrait être rencontré dans une situation réelle, la méthode fonctionne bien. Elle est capable de trouver les bons appariements. On remonte les informations d'aide à la décision, incluant le score et le fait qu'il n'y a ni de translation ni de rotation, cf. section 5.1.4.1. En plus, connaissant les informations de référence on peut remonter également le nombre de primitives n'ayant pas été appariées. On note cependant le couple d'appariement (5;11) pour les segments de droite, qui même mérite notre attention. On est dans un cas où il y a une petite contribution du score pour la longueur des segments et une assez forte contribution pour sa localisation, mais il y a une forte divergence en terme d'orientation. Au final les points de divergence et convergence

se compensent pour arriver à un score de 0.17. Pour un tel cas on peut avoir recours à ce que Lourakis et Manolis ([Lourakis, 1997]) appellent contrainte de conjonction. Cela veut dire que pour tous les attributs considérés en vue de la mise en correspondance s'il y a en au moins un qui diverge fortement d'un seuil prédéfini on rejette la primitive en question et elle ne sera pas appariée. Pour le moment nous ne procédons pas de la sorte, car nous trouvons cette contrainte assez rigide.

5.1.4.2 Cas n ° 2

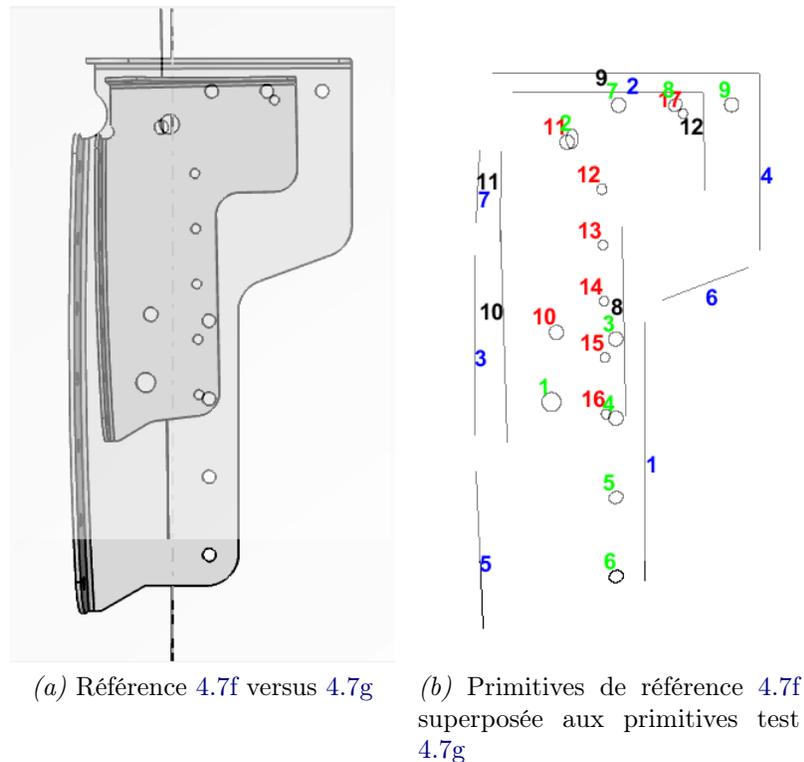


Figure 5.18 – Appariement de primitives en prenant comme référence 4.7f versus 4.7g

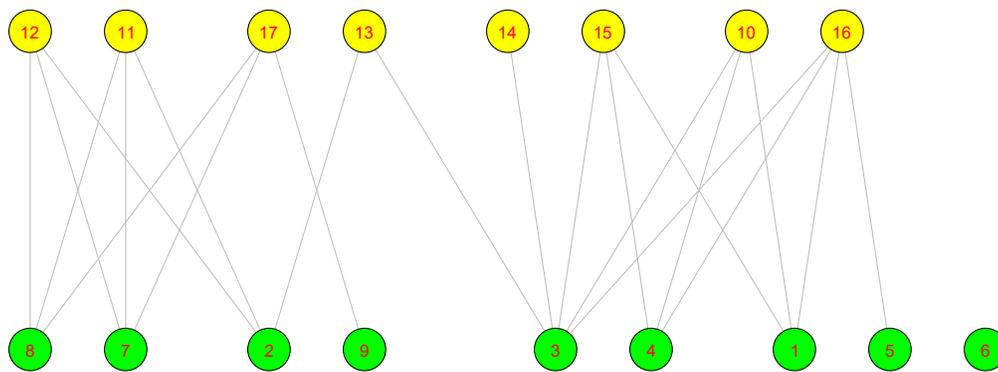
Tableau d'appariement de segments de droite

-0.03	-1.90	-0.39	-1.25	-1.40
-1.13	0.72	-1.31	-0.76	-0.26
-0.09	-1.31	0.63	-0.22	-1.28
-0.49	-0.50	-1.22	-0.98	0.42
-0.96	-2.65	-0.67	-1.68	-2.39
-0.12	-0.98	-0.94	-0.86	-0.33
-0.70	-0.64	-0.37	0.69	-0.79

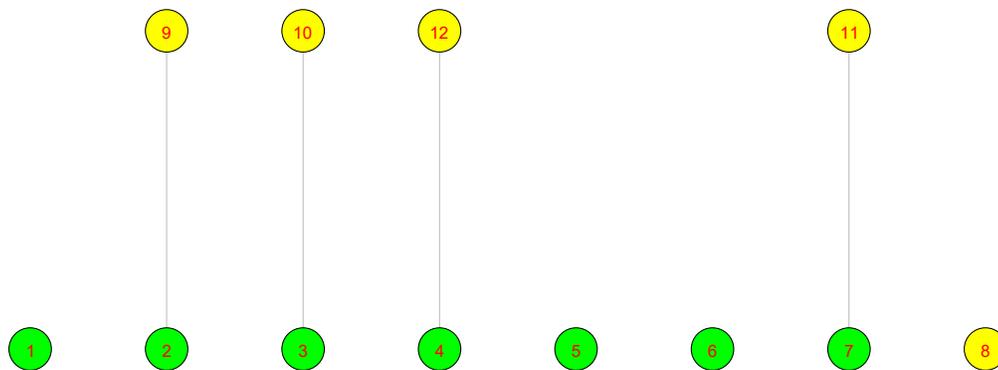
Tableau 5.12 – Tableau d'appariement de segments de droite pour le couple référence-test de la figure 5.18

Tableau d'appariement d'ellipses							
0.33	-1.01	-0.84	-0.45	-0.08	0.24	0.31	-1.47
-0.44	0.87	0.40	0.02	-0.39	-0.78	-1.13	-0.02
0.51	-0.45	-0.19	0.19	0.54	0.65	0.31	-0.79
0.22	-1.01	-0.82	-0.41	-0.01	0.38	0.72	-1.34
-0.25	-1.53	-1.33	-0.92	-0.52	-0.12	0.25	-1.84
-0.77	-2.07	-1.87	-1.46	-1.06	-0.66	-0.29	-2.37
-0.66	0.52	0.22	-0.14	-0.52	-0.91	-1.27	0.39
-0.81	0.14	0.02	-0.27	-0.61	-0.96	-1.34	0.75
-1.01	-0.20	-0.26	-0.48	-0.77	-1.11	-1.44	0.50

Tableau 5.13 – Tableau d'appariement d'ellipses pour le couple référence-test de la figure 5.18

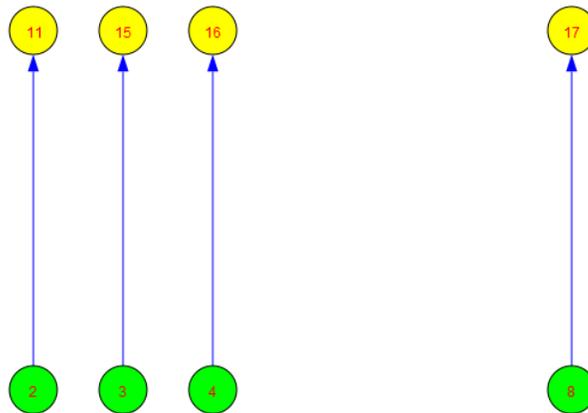


(a) Appariement d'ellipses

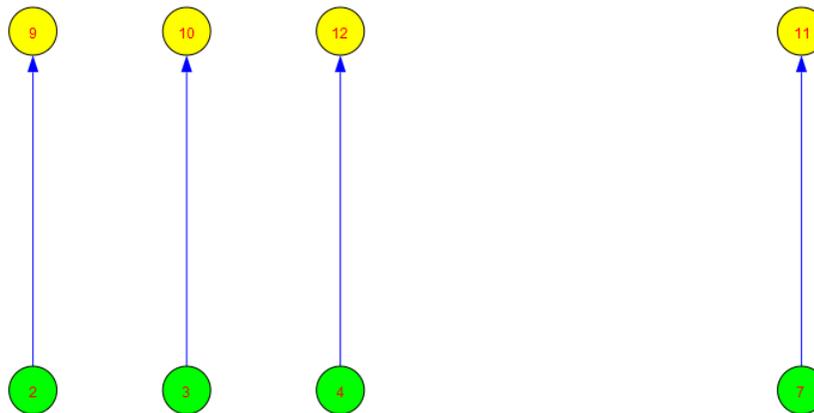


(b) Appariement de segments de droite

Figure 5.19 – Graphe biparti correspondant à l'appariement des primitives du cas 4.7e versus 4.7f



(a) Appariement d'ellipses



(b) Appariement de segments de droite

Figure 5.20 – Graphe biparti après optimisation des graphes de la figure 5.19 pour respecter la contrainte d'unicité

À la fin de la procédure, le résultat rapporté est le suivant :

- ◇ score : 0.3418.
- ◇ translation en x : 0 mm, en y : 0 mm.
- ◇ angle de rotation : 0° .

Le résultat pour ce cas test, comme on peut le constater sur les deux graphes bipartis optimisés 5.20a et 5.20b est cohérent et le faible score obtenu permettrait en pratique de fixer facilement un seuil qui aiderait à écarter l'objet testé.

5.1.4.3 Cas n ° 3

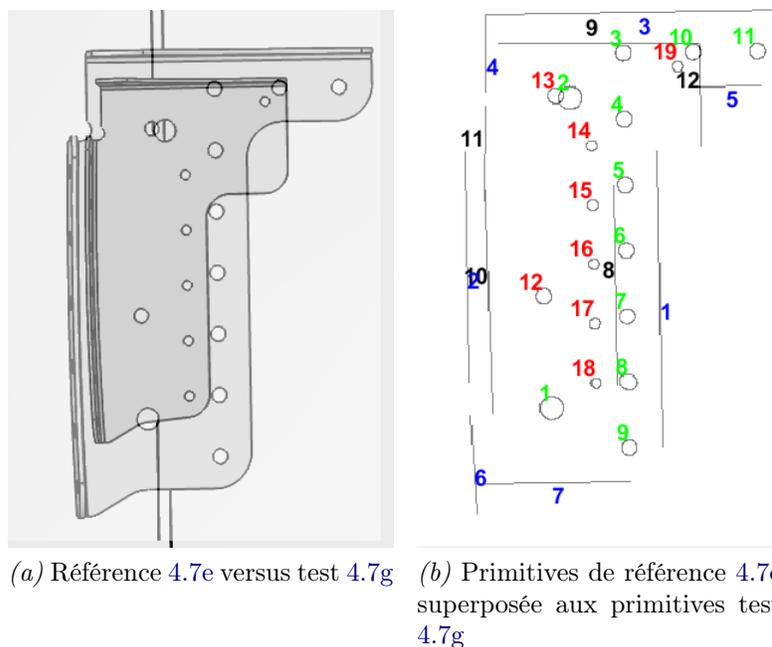


Figure 5.21 – Appariement de primitives en prenant comme référence 4.7e versus 4.7g

Tableau d'appariement de segments de droite

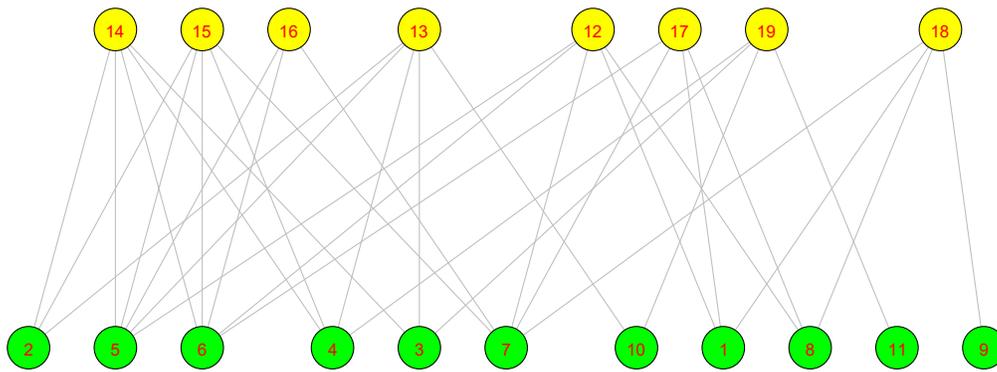
0.59	-1.15	-0.19	-0.76	-0.57
-0.02	-1.06	0.75	-0.00	-1.07
-1.22	0.53	-1.45	-0.91	-0.23
-0.96	-0.27	-0.83	0.26	-0.52
-1.03	-0.12	-1.60	-1.04	0.39
-0.74	-2.35	-0.39	-1.18	-1.92
-0.75	-1.97	-0.76	-1.64	-2.14

Tableau 5.14 – Tableau d'appariement de segments de droite pour le couple référence-test de la figure 5.21

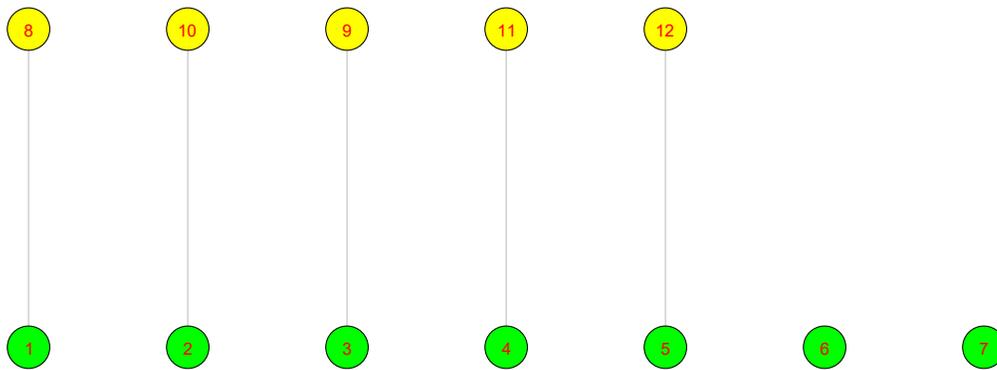
Tableau d'appariement d'ellipses

0.07	-1.22	-1.02	-0.62	-0.25	0.09	0.41	-1.64
-0.54	0.69	0.39	0.00	-0.40	-0.80	-1.15	-0.04
-0.72	0.40	0.10	-0.25	-0.61	-0.98	-1.38	0.41
-0.29	0.48	0.49	0.19	-0.16	-0.53	-0.93	0.32
0.02	0.21	0.47	0.54	0.20	-0.18	-0.52	-0.08
0.30	-0.16	0.09	0.42	0.53	0.23	-0.10	-0.48
0.39	-0.54	-0.34	0.05	0.38	0.56	0.33	-0.87
0.17	-0.96	-0.78	-0.38	-0.01	0.34	0.60	-1.28
-0.15	-1.38	-1.21	-0.81	-0.41	-0.03	0.33	-1.69
-0.94	0.01	-0.08	-0.40	-0.76	-1.12	-1.46	0.62
-1.14	-0.38	-0.47	-0.67	-0.93	-1.26	-1.59	0.28

Tableau 5.15 – Tableau d'appariement d'ellipses pour le couple référence-test de la figure 5.21

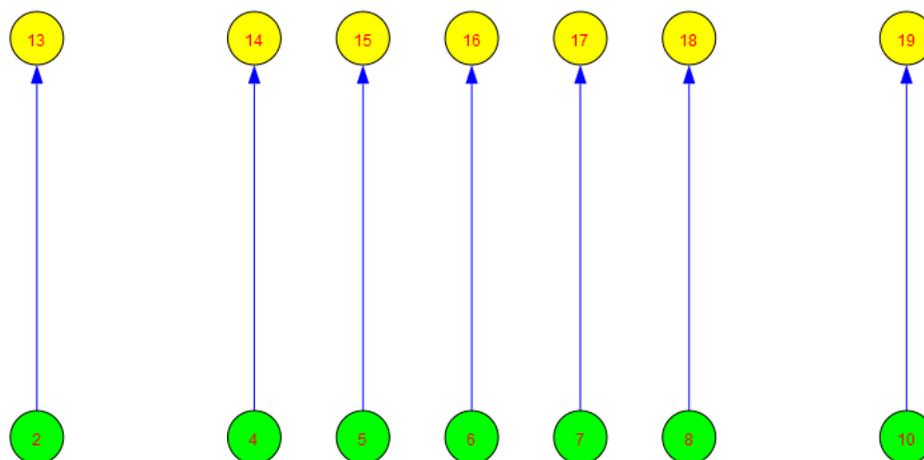


(a) Appariement d'ellipses

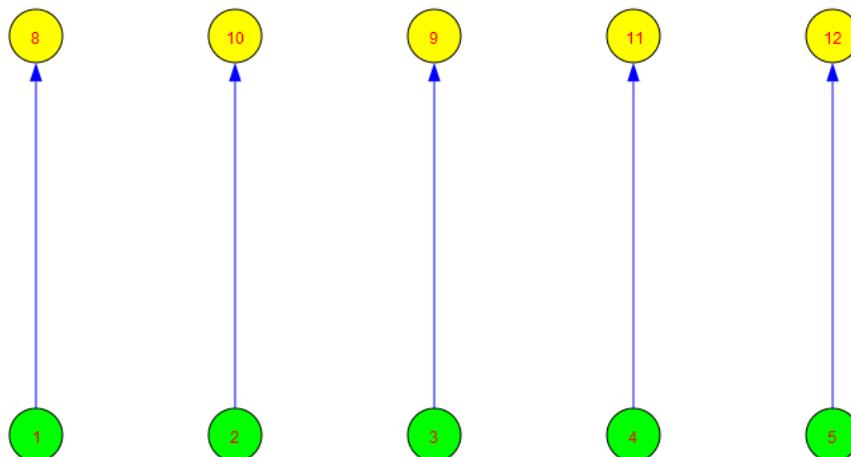


(b) Appariement de segments de droite

Figure 5.22 – Graphe biparti correspondant à l'appariement des primitives du cas 4.7e versus 4.7g



(a) Appariement d'ellipses



(b) Appariement de segments de droite

Figure 5.23 – Graphe biparti après optimisation des graphes de la figure 5.22 pour respecter la contrainte d'unicité

- ◇ score : 0.363.
- ◇ translation en x : 0 mm, en y : 0 mm.
- ◇ angle de rotation : 0° .

Sur ce cas test illustré par la figure 5.21 on attendait plus d'ellipses et de segments de droite que ceux trouvés dans l'image test. De plus ces primitives ne sont pas tout à fait distribuées spatialement de la même façon et n'ont pas les mêmes dimensions. Ces éléments conduisent à un score faible, ce qui est souhaitable pour le bon fonctionnement de la méthode.

5.1.5 Remarque concernant les objets tests ayant subi une rotation

Un point à souligner concerne le cas où l'élément test présente une rotation par rapport à l'élément de référence. Trouver la rotation entre les deux éléments ne constitue pas un problème dès lors que la rotation n'est pas trop importante (de l'ordre de 45° , avec les algorithmes que nous avons utilisés). D'ailleurs la figure 5.24 ci-dessous montre un tel exemple.

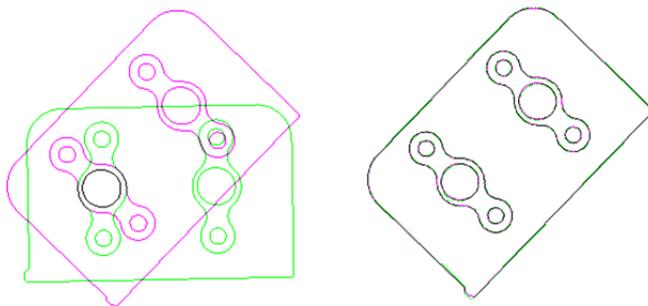


Figure 5.24 – Exemple de rotation (image de référence et image de l'objet à évaluer) avant et après recalage. À gauche : image de référence (green), image test (magenta), avant le recalage. À droite image de référence et image test après le recalage.

Néanmoins, se pose le problème d'extraction de primitives dans un tel cas. En effet pendant le recalage, l'objet à recaler subit une interpolation résultant en pertes de pixels. Extraire les primitives (notamment les ellipses) dans ces conditions en utilisant l'algorithme de Prasad (cf. 4.3.3.1.2) est difficile car le contour ayant subi le recalage présente des discontinuités, conduisant une ellipse à être fragmentée en plusieurs morceaux, ce qui rend l'ajustement à l'ellipse irréalizable.

Une solution possible consisterait à extraire d'abord les primitives, effectuer le recalage et appliquer les paramètres de recalage sur les primitives non recalées pour obtenir des primitives correctement alignées et effectuer ensuite la mise en correspondance.

5.2 Résultats d'appariement entre une image synthétique et une image réelle

5.2.1 Tests réalisés sur les données du projet AirCOBOT

Dans cette section, notre objectif est de montrer comment notre méthode d'appariement de primitives se comporte en présence de données réelles. Il s'agit d'apparier les primitives issues du modèle CAO avec des primitives extraites d'une image réelle acquise avec une caméra physique. Pour le projet Lynx© nous avons rencontré quelques difficultés pour

obtenir les jeux de données constitués d'une part par des images synthétiques et d'autre part par des images réelles comme nous l'expliquons et illustrons dans l'annexe D de ce manuscrit. Néanmoins, nous avons eu l'occasion de tester la méthode dans un autre contexte applicatif, avec un jeu de données équivalent. Il s'agit des tests réalisés lors d'une collaboration avec des membres de l'équipe projet AirCOBOT. Dans la section suivante nous allons présenter ce projet AirCOBOT et les résultats obtenus avec notre méthode sur des tests réalisés avec les données de ce projet.

5.2.1.1 Contexte

Pour tester notre méthode dans cette situation nous avons exploité les données d'une application de navigation d'un robot autour d'un aéronef, suivi d'inspection.

Cette application s'insère dans le projet AirCOBOT que nous décrivons ci-après. Le projet AirCOBOT, un projet collaboratif, entre plusieurs acteurs industriels (Akka Technologies, Airbus, Sterela, M3Systems, 2moro Solutions) et deux laboratoires de recherche (l'Institut Clément Ader et le LAAS-CNRS). Le but de ce projet est de déployer une solution complète pour l'inspection d'aéronefs au sol, avant vol, par un robot autonome et collaborant avec les systèmes d'information et les agents de maintenance. Parmi les nombreuses tâches à réaliser dans ce projet, il y a la tâche de navigation du robot, c'est-à-dire, le positionnement du robot devant les zones d'intérêt de l'aéronef à inspecter. Ensuite, une fois que le robot est positionné correctement la tâche suivante consiste à réaliser l'inspection. Un des éléments disponibles dans ce projet est le modèle CAO de l'aéronef à inspecter. Nous avons collaboré avec les équipes du projet AirCOBOT issues de l'Institut Clément Ader et du LAAS, pour mettre au point une stratégie de navigation du robot basée sur la mise en correspondance d'amers (ellipses et segments de droite) présents sur la partie extérieure de l'aéronef. Après cette localisation basée sur l'appariement de primitives, la même méthode est utilisée pour inspecter certains objets de l'avion qui peuvent se décomposer en ensemble d'ellipses ou de segments de droite.

Avant de montrer les résultats d'appariement d'amers obtenus lors de la navigation puis de l'inspection, nous allons présenter ci-dessous quelques vues du modèle CAO de l'un des avions à inspecter ainsi que quelques images acquises avec une caméra PTZ, montrant notamment les différentes zones d'intérêt.

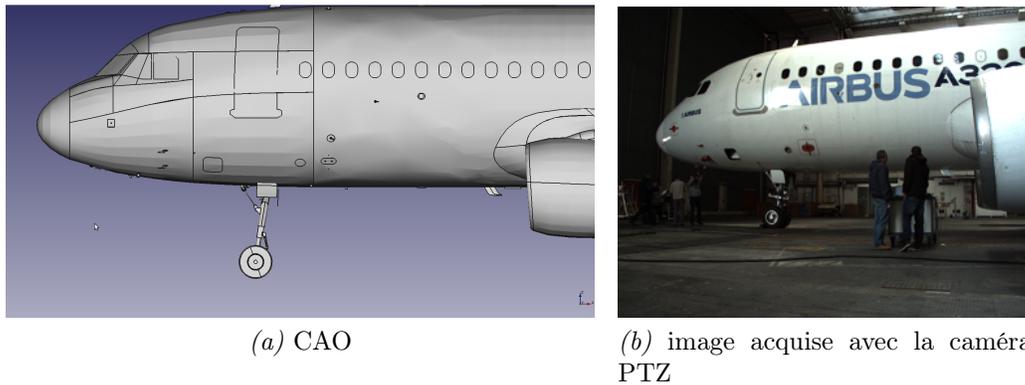


Figure 5.25 – CAO d’un avion à inspecter et son correspondant acquis avec la caméra PTZ, projet AirCOBOT

Dans la vue CAO (cf. figure 5.25a) ainsi que dans l’image réelle, on peut voir que l’avion que l’on cherche à inspecter présente des formes géométriques simples (segments de droite, ellipses, arcs d’ellipses), rendant ainsi possible l’utilisation de notre algorithme d’appariement de primitives basée sur l’exploitation des graphes.

Comme nous l’avons expliqué, dans l’application AirCOBOT, un robot se déplace autour de l’aéronef à inspecter. En effet, dans les travaux conjoints réalisés avec l’équipe projet d’AirCOBOT, notre algorithme a servi notamment à mettre en correspondance les ellipses du modèle CAO et celles de l’image acquise par la caméra PTZ afin d’aider à la navigation du robot. Nous verrons les résultats de cette mise en correspondance ultérieurement.

Observons maintenant l’image de la figure 5.33a et son vis-à-vis réel de la figure 5.33b. Ce cas correspond à un cas où le robot portant la caméra PTZ est déjà localisé face à l’avion et on souhaite réaliser l’inspection. On constate ici, la présence des segments de droite, rendant possible encore une fois l’utilisation de notre algorithme d’appariement de primitives, cette fois-ci pour réaliser l’inspection. Dorénavant, nous allons montrer les résultats d’appariement pour quelques éléments mis en correspondance par notre algorithme, soit lors de la navigation, soit lors de l’inspection. Avant de montrer ces résultats, deux points importants sont à noter :

- ◇ 1. L’extraction des primitives appariées dans cette section a été réalisée par nos collègues du LAAS et de l’Institut Clément Ader, en utilisant l’algorithme de Transformée de Hough Probabiliste pour extraire les segments de droite et l’algorithme de EDCircles pour l’extraction des ellipses dans les images acquises par la caméra PTZ. Les primitives issues du modèle CAO sont projetées sur l’image réelle en réalité augmentée. Cela est illustré par les images ci-après.
- ◇ 2. Systématiquement, nous allons montrer l’image réelle avec les primitives mises en relief par une couleur spécifique (rouge) ainsi que la projection des primitives du modèle CAO sur une copie de la même image réelle (en cyan) pour que le lecteur

puisse observer distinctement les deux ensembles de primitives à appairer. Ensuite, nous montrerons la matrice d'appariement (quand cela est possible, car parfois le nombre de primitives est trop important) pour l'ensemble de primitives théoriques et réelles, le graphe biparti construit en exploitant la matrice d'appariement puis pour finir le graphe biparti final, montrant à la fois toutes les correspondances, tout en mettant en évidence les correspondances qui respectent la contrainte d'unicité de l'appariement.

5.2.1.2 Cas n ° 1



(a) Projection des primitives CAO (ellipses) sur l'image (b) Ellipses détectées dans l'image acquise avec la caméra PTZ.

Figure 5.26 – Navigation : robot loin de l'avion.

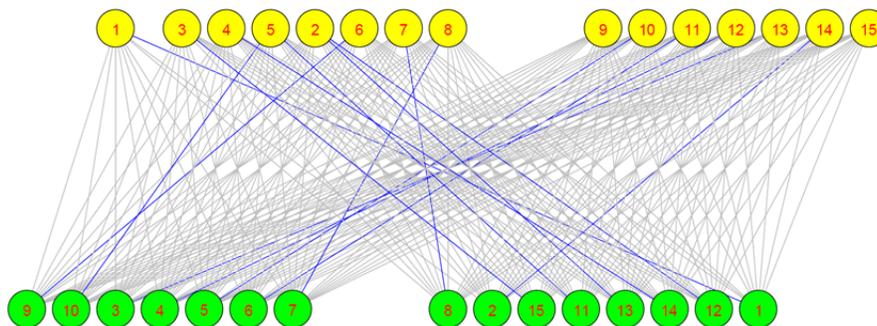


Figure 5.27 – Graphe biparti illustrant des cas de conflits (non respect de la contrainte d'unicité de l'appariement) lors de la mise en correspondance des éléments de la figure 5.26

Sur la figure 5.27, on voit par exemple que le nœud 2 en jaune est mis en correspondance avec les nœuds 12 et 13 en vert. Après l'optimisation du graphe, on constate sur la figure 5.28 que le nœud 2 en jaune n'est plus connecté qu'au seul nœud 13 en vert.

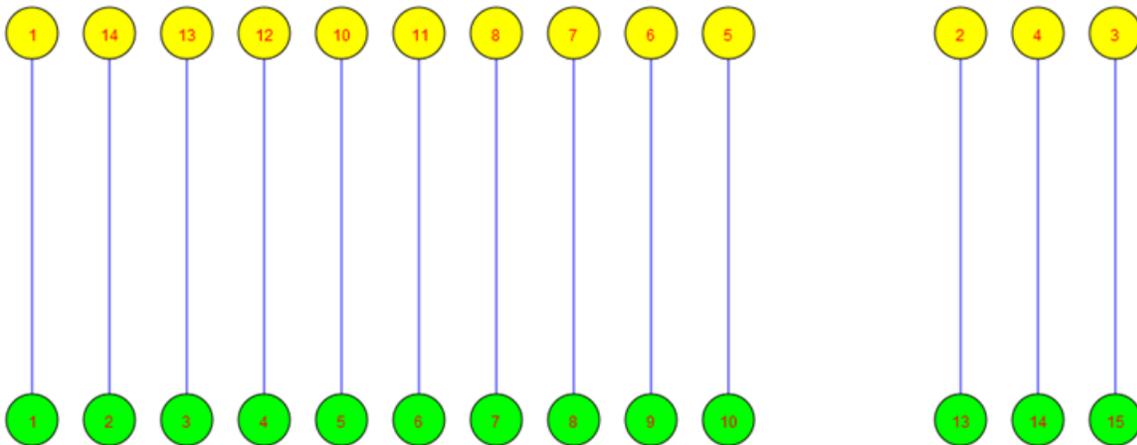


Figure 5.28 – Graphe biparti de la figure 5.27 élagué pour respecter l'unicité de l'appariement

5.2.1.3 Cas n° 2



(a) Projection des primitives CAO (ellipses) sur l'image acquise avec la caméra PTZ.



(b) Ellipses détectées dans l'image acquise avec la caméra PTZ.

Figure 5.29 – Navigation, robot proche de l'avion.

Tableau d'appariement d'ellipses

-0.407	0.204	-0.424	-0.180
-1.147	0.446	0.461	0.193
-0.565	0.120	-0.113	0.548

Tableau 5.16 – Matrice de scores pour le jeu de primitives de la figure 5.29

Le tableau de scores 5.16 est utilisé pour construire le graphe biparti de la figure 5.30 où nous montrons les meilleurs couple d'appariement en bleu.

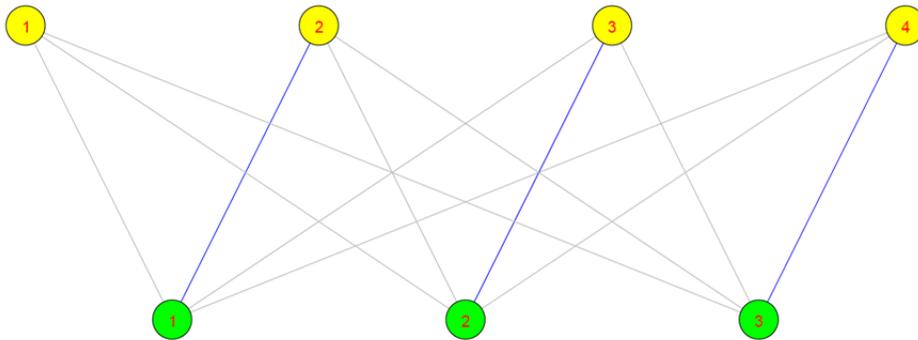
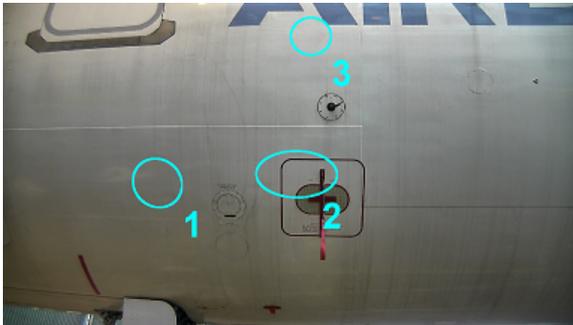
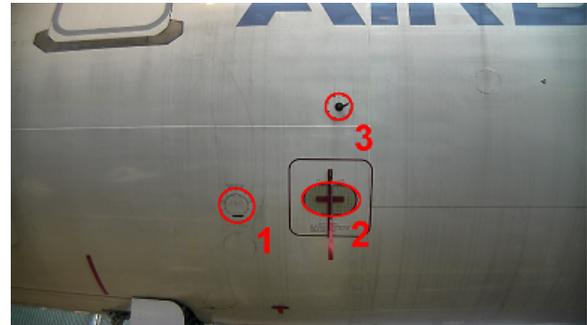


Figure 5.30 – Graphe biparti correspondant à la matrice de scores 5.16, avec les meilleurs appariements mis en évidence par la couleur bleue, dans la figure 5.16.

5.2.1.4 Cas n ° 3



(a) Projection des primitives CAO (ellipses) sur l'image acquise avec la caméra PTZ.



(b) Ellipses détectées dans l'image acquise avec la caméra PTZ.

Figure 5.31 – Robot à un point de contrôle pour inspecter.

Le tableau 5.17 montre les appariements obtenus pour les primitives de la figure 5.31.

Tableau d'appariement d'ellipses		
-0.882	-2.477	-3.079
-0.944	0.140	-1.011
-2.890	-2.164	-0.867

Tableau 5.17 – Matrice de scores pour le jeu de primitives de la figure 5.31

Après la sélection des meilleurs couples d'appariements du tableau 5.17, on obtient les appariements en bleu sur la figure 5.32

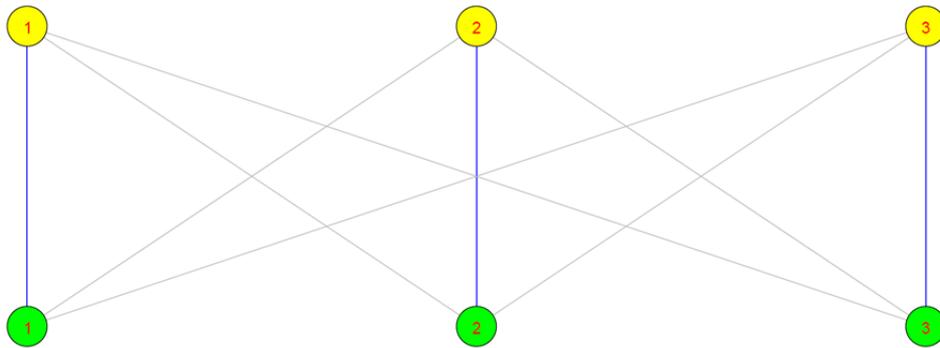
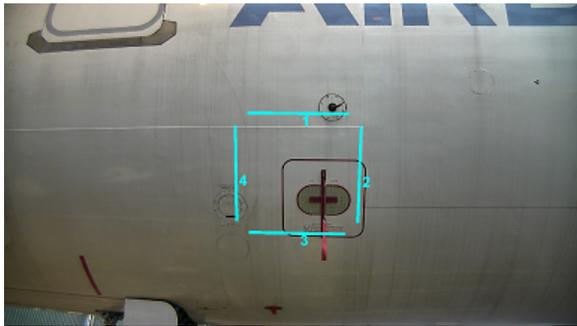
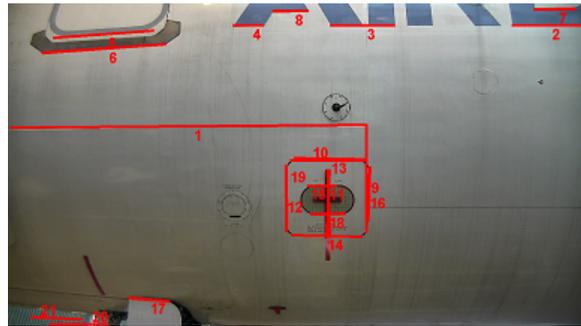


Figure 5.32 – Graphe biparti correspondant au tableau des scores 5.17, avec les meilleurs appariements mis en évidence par la couleur bleue.

5.2.1.5 Cas n° 4



(a) Projection des primitives CAO (segments de droite) sur l'image acquise avec la caméra PTZ.



(b) Segments de droite extraits de l'image acquise par la caméra PTZ

Figure 5.33 – Robot positionné à un point de contrôle, pour réaliser l'inspection

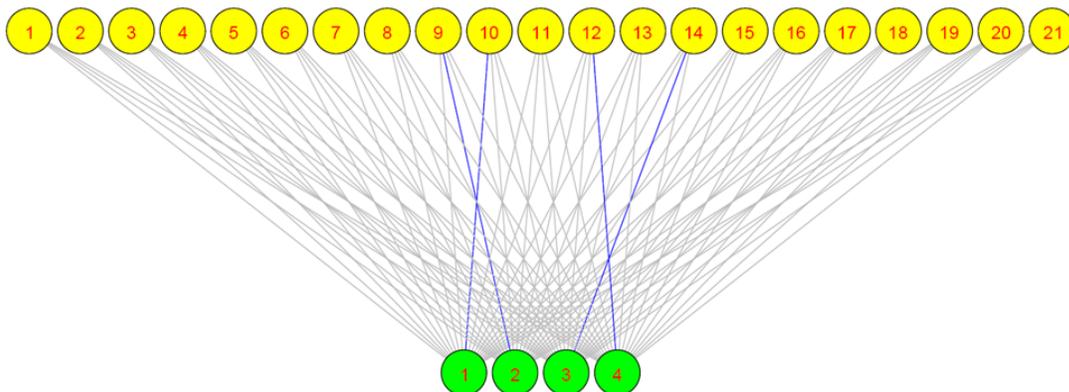


Figure 5.34 – Graphe biparti correspondant à la mise en correspondance des primitives de la figure 5.33 meilleurs appariements mis en évidence par la couleur bleue.

La figure 5.34 illustre la mise en correspondance des primitives de la figure 5.33.

5.3 Conclusion

Dans ce chapitre nous avons montré comment se comporte notre méthode, en réalisant des tests d’abord sur des données synthétiques, puis sur des données réelles. Les tests sur les données synthétiques concernent plusieurs cas différents, notamment cas d’objets translattés, ou pivotés, combinaison de plusieurs cas comme translation et objets manquants, ou objets en plus sur les données tests en comparaison avec les données de référence. Dans ce chapitre nous avons également illustré, sur des exemples concrets, la problématique de validation croisée que nous avons expliqué dans le **chapitre 4**. De même, nous avons aussi abordé l’importance du choix de l’extracteur des primitives qui peut conditionner la partie de mise en correspondance. Nous avons illustré cette difficulté lié à l’extracteur sur un cas de rotation.

Malgré le bon comportement de la méthode sur l’ensemble des tests réalisés, nous aurions souhaité effectuer plus de tests sur des données réelles. Pour la suite, nous souhaiterions augmenter le nombre de types de primitives utilisées dans la méthode et d’y inclure notamment les arcs d’ellipses. Nous estimons que l’utilisation d’ellipses, segments de droit et arcs d’ellipses permettent une décomposition plus complète des objets. Par ailleurs ayant construit des graphes attribués relationnels nous nous limitons à exploiter les attributs sur les nœuds de graphes.

À l’avenir il faudrait élargir la méthode afin qu’elle puisse combiner l’exploitation des mesures dimensionnelles à l’aspect topologique des graphes relationnels. Un autre élément qui pourrait être intégré à la méthode consiste en l’exploitation du concept multivues [Mery and Carrasco, 2006, Chawla and Deb, 2012]. Au lieu d’exploiter une seule vue du modèle et de l’image réelle, on pourrait agréger les scores de différentes vues dans le processus de prise de décision de la méthode. Pour finir, nous pensons qu’il serait utile de développer un cadre mathématique pour la projection directe des primitives 3D (ellipses, segments de droite, arcs d’ellipses) issues du modèle sur l’image réelle, en réalité augmentée.

6 Conclusion générale

Tout au long de ce document nous avons décrit nos travaux sur l'inspection automatisée d'assemblages mécaniques aéronautiques par vision artificielle, grâce à l'exploitation du modèle CAO de l'assemblage à inspecter. Dans le **chapitre 1**, nous avons précisé notre problématique qui consiste à utiliser au mieux le modèle CAO pour en faire la référence dans le processus d'inspection d'assemblage mécanique aéronautique. On ne peut pas traiter cette problématique sans ramener le modèle CAO et l'assemblage physique dans le même repère afin de disposer des données comparables. À travers deux projets, CAAMVis et Lynx©, nous avons vu comment ce problème pouvait être traité. La solution retenue consistant à poser d'abord la caméra devant l'assemblage réel, la calibrer et estimer sa pose et réappliquer cette pose pour générer les données synthétiques (cf. annexe D).

Au **chapitre 2**, nous nous sommes intéressés à faire un état de l'art sur l'inspection de manière générale, en commençant par définir ce que nous entendons par inspection. Nous avons par la suite fait le parallèle entre inspection et reconnaissance tout en mettant en exergue leur similitudes et différences. Ce qui nous a ensuite permis de décrire des méthodes de reconnaissances d'objets et de dire quand elles peuvent être adaptées au problème d'inspection. En lisant ce **chapitre 2**, on conclut avec aisance que nos méthodes d'inspection présentées dans les **chapitres 3 et 4** respectivement s'inscrivent dans le contexte des méthodes qui utilisent l'information géométrique, et sont de ce fait de la même famille que les méthodes de reconnaissance d'objets présentées dans la section 2.2.4, avec en plus l'inclusion d'informations d'ordre dimensionnel et/ou de tolérances afin de satisfaire les exigences de l'inspection. Toujours dans le **chapitre 2**, nous avons abordé les différents systèmes d'acquisition, utilisés en inspection et nous avons aussi catégorisé l'inspection en deux groupes, celui utilisant le modèle CAO et un autre plus classique qui ne se sert pas du modèle CAO. Cela nous a permis de souligner l'importance de l'inspection tant les domaines où elle apparaît sont variés.

Au **chapitre 3**, nous avons décrit notre méthode d'inspection développée dans le cadre du projet CAAMVis. La méthode exploite le modèle CAO d'un assemblage et des images provenant d'un capteur à double tête stéréo, calibré en amont et porté par un robot qui à chaque point de vue CAO prédéfini positionne le capteur afin d'obtenir les images. Les paramètres intrinsèques et extrinsèques obtenus lors du calibrage sont ensuite utilisés pour projeter le modèle CAO sur une image de même taille que l'image réelle et contenant uniquement l'objet CAO correspondant à l'objet réel à inspecter.

Par la suite, après la détection des contours dans les deux images, nous construisons la transformée en distance de l'objet réel et nous cherchons à optimiser la position et l'orientation du contour synthétique dans la transformée en distance de l'objet réel. L'algorithme d'optimisation utilisé est la descente de gradient, en procédant en deux temps, en translation puis en rotation. La décision quant à l'acceptation ou au rejet de l'objet évalué se base sur des notions de taux de recouvrement et de taux de bruit à l'intérieur des objets après la convergence. Les résultats obtenus avec cette méthode sont encourageants, avec néanmoins quelques pistes d'amélioration évoquées lors du bilan sur la méthode (cf. section 3.1.5). Dans cette section 3.1.5, nous expliquons par ailleurs que nos développements dans les projets CAAMVis se sont arrêtés de façon prématurée à cause de la cessation d'activité de la société Noomeo avec qui nous collaborions étroitement. Ce qui nous a conduit à une autre collaboration, avec la société G2Metric, afin de contribuer au projet Lynx©.

Dans le contexte du projet Lynx©, nous avons développé la méthode décrite dans le **chapitre 4**. Elle consiste à exploiter les données d'une caméra unique de type PTZ et le modèle CAO pour réaliser l'inspection. Nous avons proposé notre méthode d'appariement basée sur la comparaison d'une image réelle et d'une image synthétique issue du modèle CAO (cf. schéma présenté dans la figure 4.6). Après l'obtention d'une image synthétique et d'une image réelle, les deux ayant un référentiel commun, nous proposons de segmenter les deux images, de recalibrer leurs contours (la seule information commune aux deux images est la géométrie), d'en extraire les primitives (ellipses et segments de droite), puis de procéder à la mise en correspondance de ces primitives par l'exploitation d'éléments de la théorie des graphes (que nous avons pris soin d'introduire en début du **chapitre 4**) et d'une fonction de mesure qui étend les travaux présentés dans [McIntosh and Mutch, 1988]. La mise en correspondance devant respecter la contrainte d'unicité, nous avons proposé d'effectuer cette vérification à l'aide d'un graphe biparti. Enfin, nous avons défini une procédure d'aide à la décision qui permet de remonter des informations utiles à l'utilisateur de l'application sous forme d'un rapport. Cette procédure d'aide à la décision, se base sur les *a priori* du modèle CAO et du résultat de la mise en correspondance ainsi que des paramètres de la matrice de transformation rigide obtenue lors du recalage.

Les tests de la méthode présentés dans le **chapitre 5** ont montré que la méthode est performante, sur des données synthétiques et réelles. Et de plus nous avons vu que cette méthode peut être adaptée à d'autres applications car les ellipses et segments de droite sont très présents dans notre environnement quotidien. Ainsi la méthode a pu être testée sur une application de navigation d'un robot autour d'un aéronef pour aider à sa localisation grâce aux amers visuels de type ellipses et segments de droite [Jovancevic et al., 2016]

présents sur l'aéronef.

Nous aurions voulu traiter la méthode sur notre assemblage (cf. figures 1.1c et 1.1). Les difficultés mentionnées dans l'annexe D ne nous ont pas permis de réaliser ces tests. Un problème qui pour nous mérite d'être regardé avec attention à l'avenir est le choix des prises de vue. Ce choix peut conditionner l'extraction de primitives comme on le voit sur l'annexe D et limiter l'utilisation de la méthode qui aujourd'hui adresse mieux les objets plats que les objets pouvant avoir des parties saillantes qui selon l'angle de vue ne sont plus visibles sur l'image. Par ailleurs, il est indispensable à l'avenir de définir un cadre mathématique pour la projection des primitives, à l'instar de [Hutter and Brewer, 2009] pour pouvoir projeter uniquement les primitives du modèle qui nous intéressent et ne pas s'encombrer à projeter toute la surface d'un objet. Pour terminer, un autre point à investiguer est la collaboration de la méthode avec d'autres méthodes de type topologiques, comme par exemple la méthode de [Bai and Latecki, 2008] utilisant le squelette.

Annexe A

Assemblages mécaniques utilisés pour nos tests

L'objectif de cette annexe est de concentrer dans un seul endroit les différents assemblages et modèles que nous avons utilisés lors des tests réalisés avec nos deux méthodes d'inspection, décrites dans les **chapitres 3 et 4**.

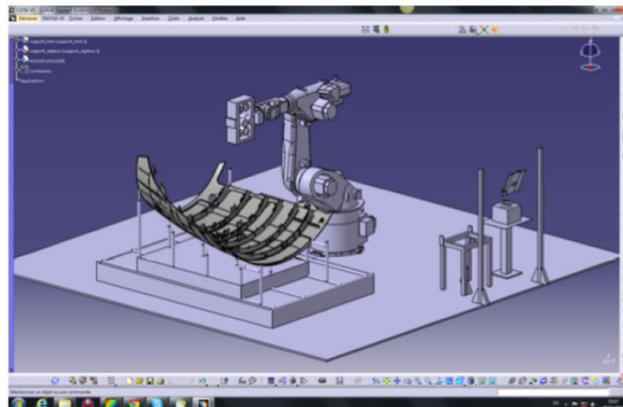
A.1 Assemblage et modèle CAO utilisés dans le projet CAAMVis



(a) Modèle CAO d'un assemblage 1



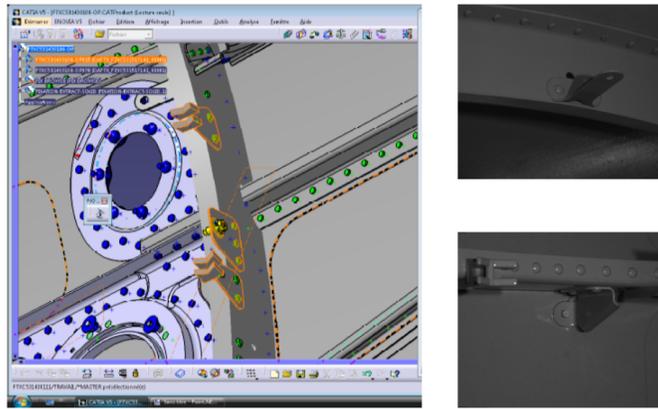
(b) Robot portant le capteur double tête stéréo en face de l'assemblage réel



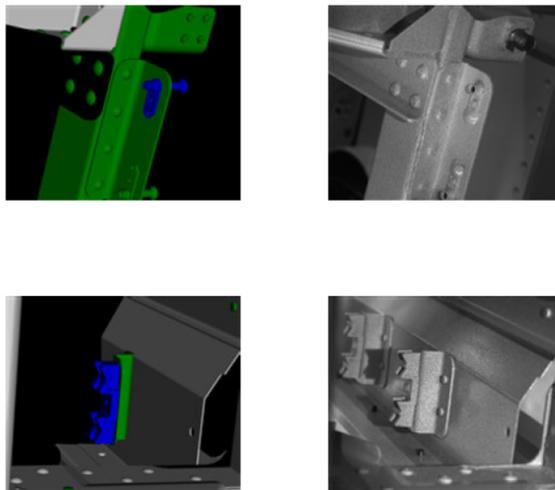
(c) Modèle CAO d'un assemblage 1, vue avec robot

Figure A.1 – CAAMVis : zoom sur modèle CAO et assemblage réel

A.1.1 Zoom sur quelques éléments du modèle CAO et de l'assemblage réel à contrôler



(a)



(b)

Figure A.2 – CAAMVis : zoom sur quelques éléments du modèle CAO, de l'assemblage réel

A.1.2 Exemple d'élément réel à contrôler reconstruit en 3D

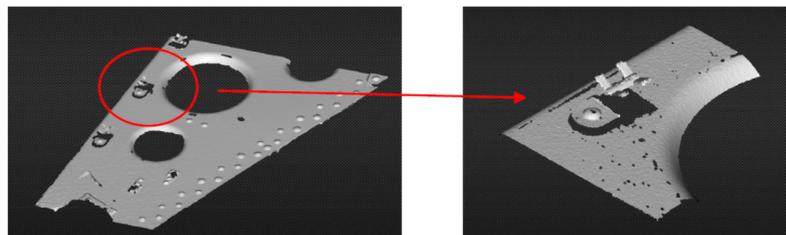
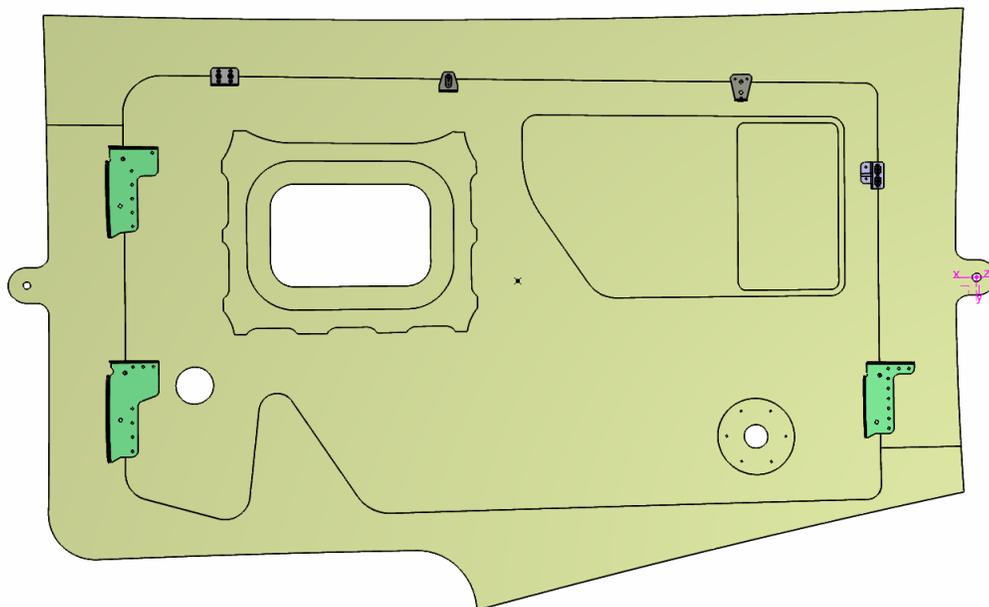


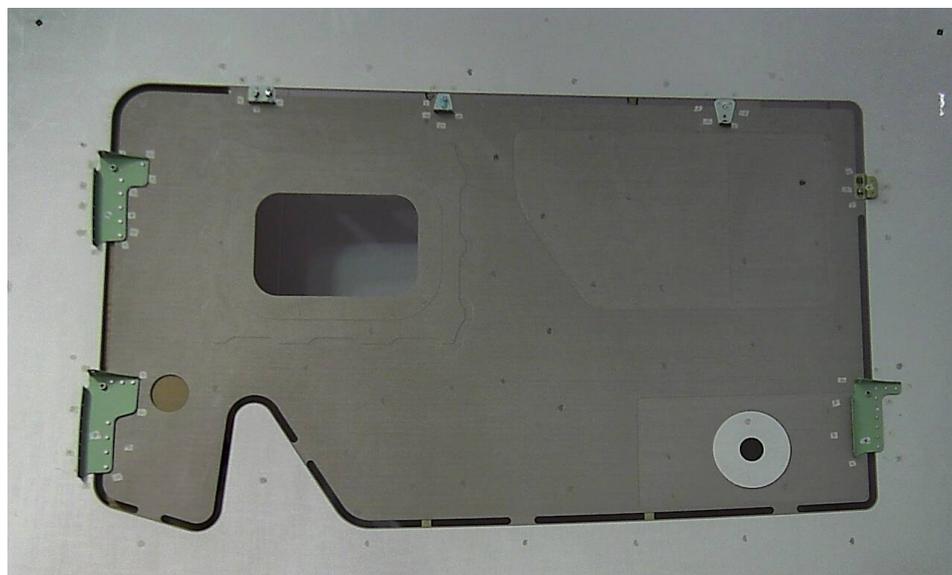
Figure A.3 – CAAMVis : zoom sur un élément CAO reconstruit en 3D

A.2 Assemblage et modèle CAO utilisés dans le projet Lynx©

A.2.1 Assemblage et modèle CAO fournis par G2Metric



(a) Modèle CAO de l'assemblage G2Metric



(b) Image de l'assemblage réel mis à disposition par G2Metric

Figure A.4 – Modèle CAO et assemblage réel mis à disposition par G2Metric dans le cadre du projet Lynx©

A.2.2 Zoom sur quelques éléments du modèle CAO, utilisés lors de tests avec données de synthèse

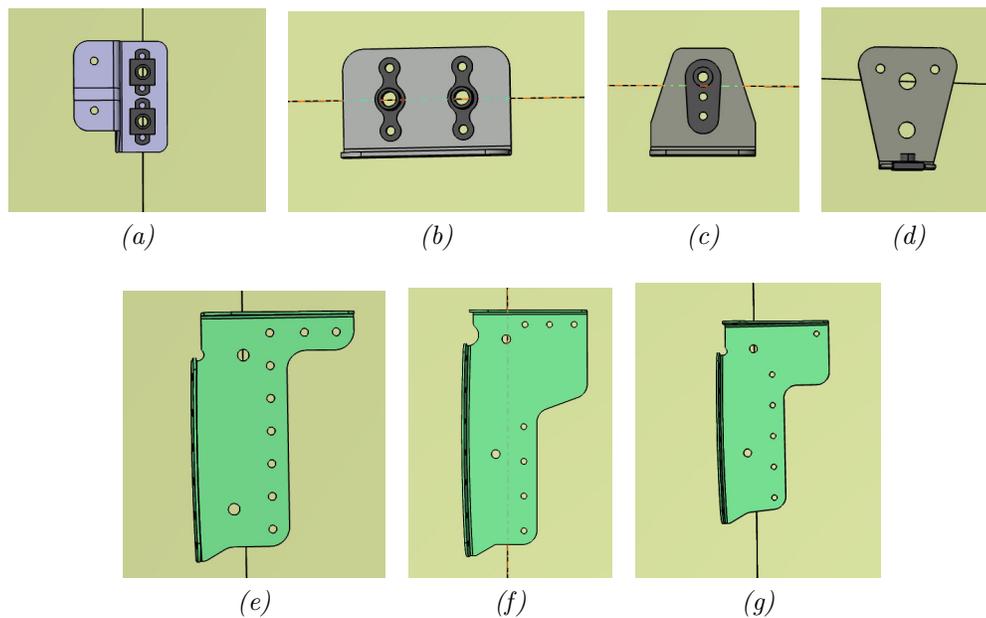
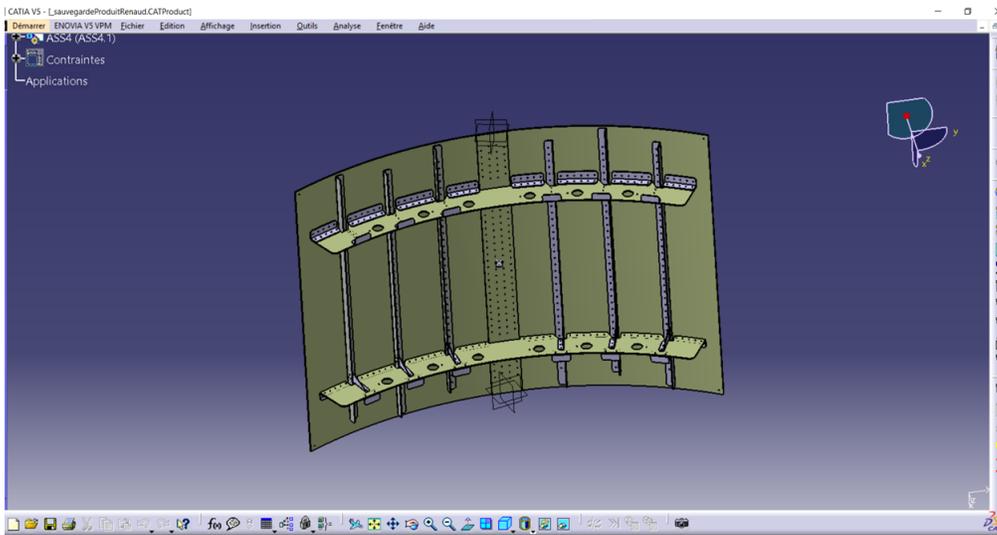


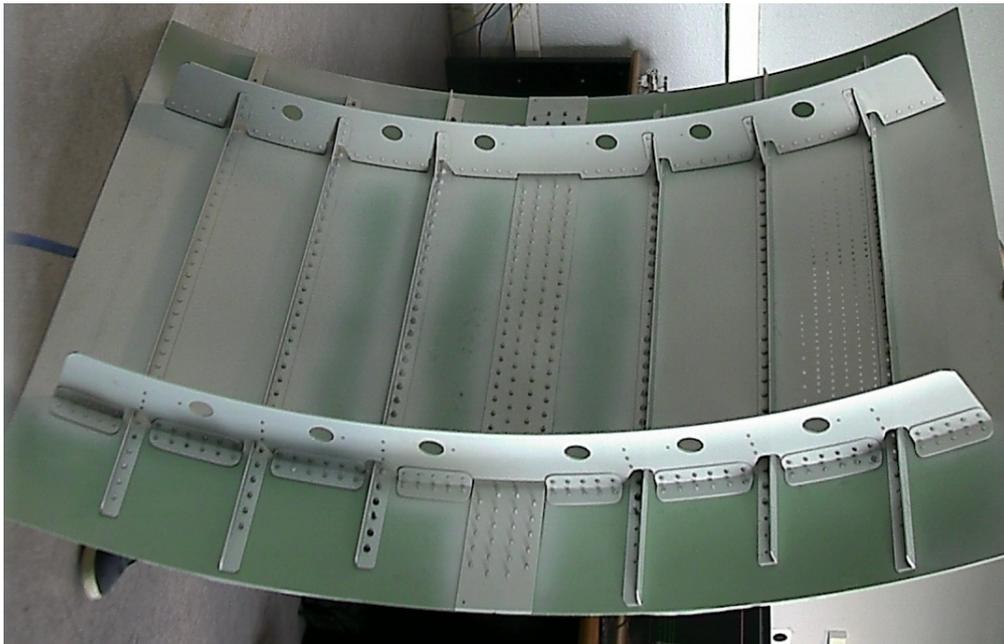
Figure A.5 – Images générées à partir du modèle CAO et utilisées lors de tests synthétiques

A.2.3 Assemblage et modèle CAO fournis par Aéroliia

Bien que mis à notre disposition dans le cadre du projet CAAMVis, les travaux sur cet assemblage n'ont pas pu être réalisés dans le contexte de ce projet (à cause de la cessation d'activités de la société Noomeo). Nous l'avons utilisé pour tester la méthode d'inspection développée dans le cadre du projet Lynx.



(a) Modèle CAO assemblage mis à disposition par Aérolia



(b) Assemblage réel 2

Figure A.6 – CAAMVis : Zoom sur modèle CAO et assemblage réel

A.2.3.1 Éléments de l'assemblage Aérolia



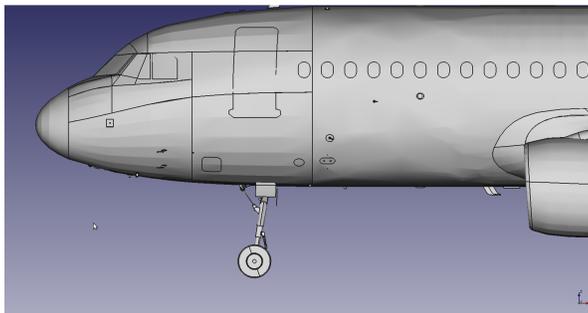
(a) zoom sur éléments de l'assemblage Aérolia, vue de face



(b) zoom sur éléments de l'assemblage Aérolia, vue de côté

Figure A.7 – CAAMVis : zoom sur quelques éléments de l'assemblage

A.3 Données utilisées pour tester notre méthode lors de la collaboration avec l'équipe projet AirCOBOT



(a) CAO d'un avion



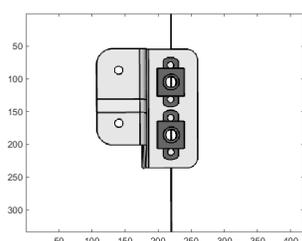
(b) Image réel de l'avion, acquise avec une caméra PTZ

Figure A.8 – Données du projet AirCOBOT utilisées pour tester notre méthode développée dans le cadre du projet Lynx©

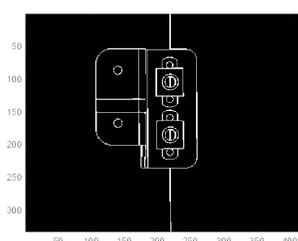
Annexe B

Résultats de la segmentation des images de la figure 4.7

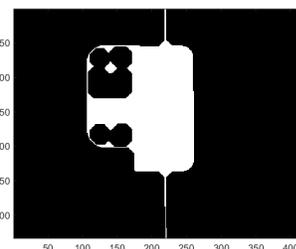
B.1 Résultat de segmentation de l'objet 4.7a



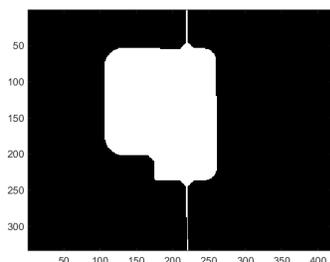
(a) 4.7a en niveaux de gris



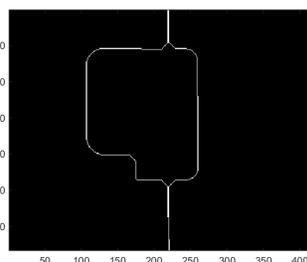
(b) 4.7a binarisée



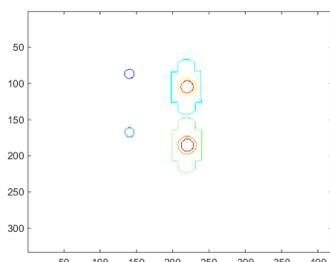
(c) 4.7a après fermeture avec un disque de taille 15



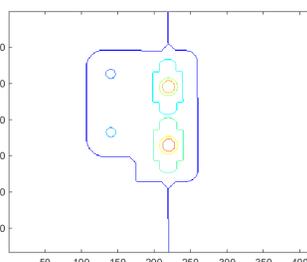
(d) 4.7a après remplissage des trous



(e) 4.7a contour extérieur de l'objet



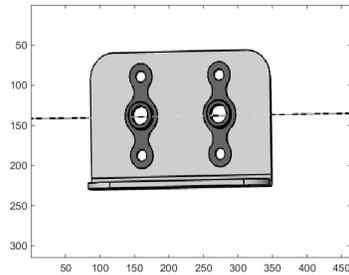
(f) 4.7a régions internes



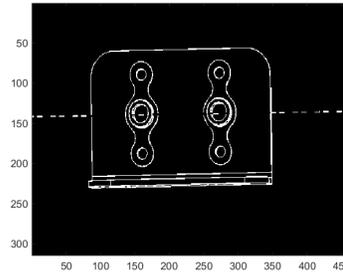
(g) 4.7a image finale segmentée

Figure B.1 – Image synthétique segmentée 4.7a par application de plusieurs pré-traitements successifs

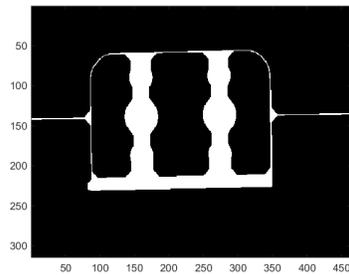
B.2 Résultat de segmentation de l'objet 4.7b



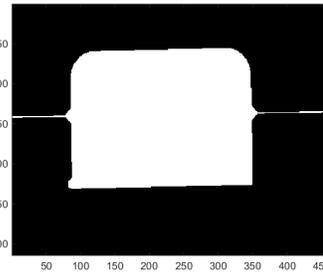
(a) 4.7b en niveaux de gris



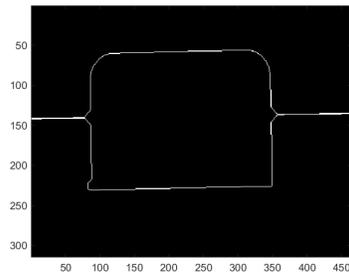
(b) 4.7b binarisée



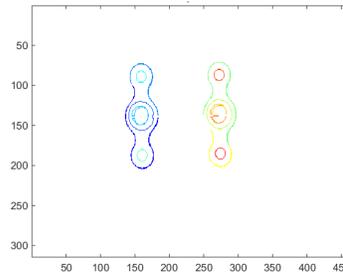
(c) 4.7b après fermeture avec un disque de taille 15



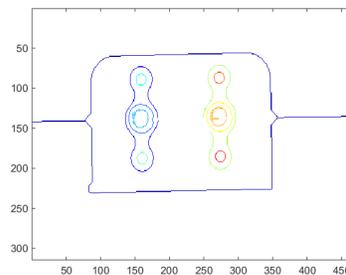
(d) 4.7b après remplissage des trous



(e) 4.7b contour extérieur de l'objet



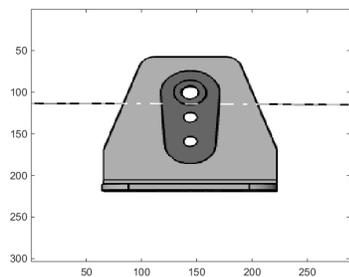
(f) 4.7b régions internes



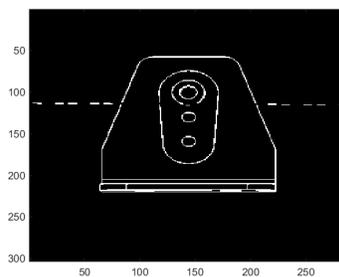
(g) 4.7b image finale segmentée

Figure B.2 – Image synthétique 4.7b segmentée par application de plusieurs pré-traitements successifs

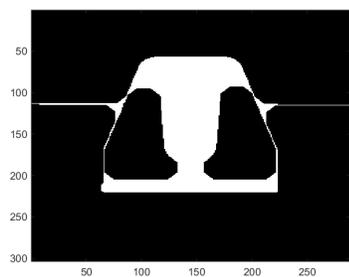
B.3 Résultat de segmentation de l'objet 4.7c



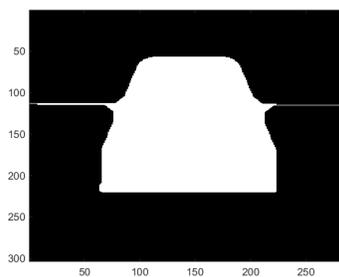
(a) 4.7c en niveaux de gris



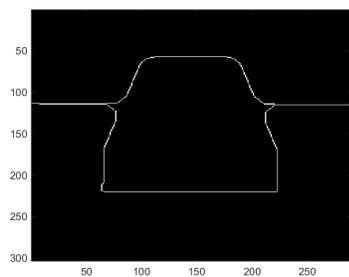
(b) 4.7c binarisée



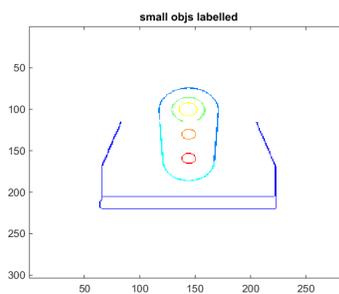
(c) 4.7c après fermeture avec un disque de taille 15



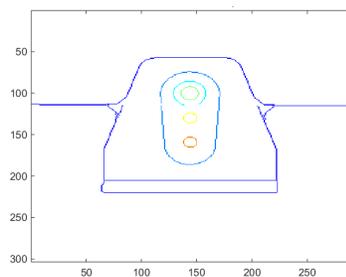
(d) 4.7c après remplissage des trous



(e) 4.7c contour extérieur de l'objet



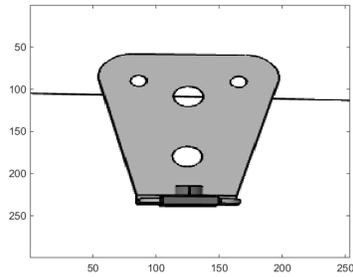
(f) 4.7c régions internes



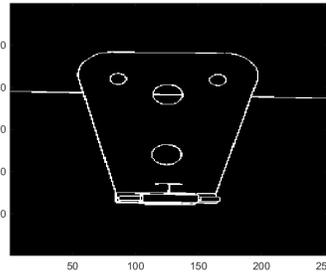
(g) 4.7c image finale segmentée

Figure B.3 – Image synthétique 4.7c segmentée par application de plusieurs pré-traitements successifs

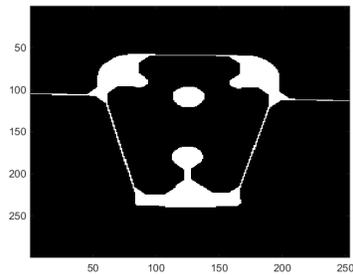
B.4 Résultat de segmentation de l'objet 4.7d



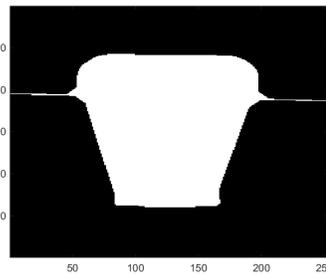
(a) 4.7d en niveaux de gris



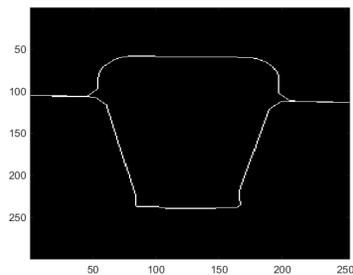
(b) 4.7d binarisée



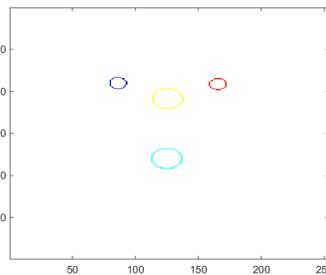
(c) 4.7d après fermeture avec un disque de taille 15



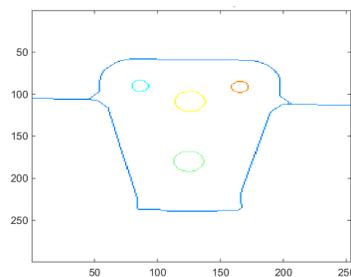
(d) 4.7d après remplissage des trous



(e) 4.7d contour extérieur de l'objet



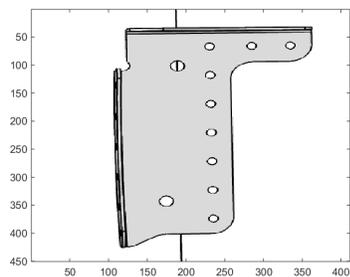
(f) 4.7d régions internes



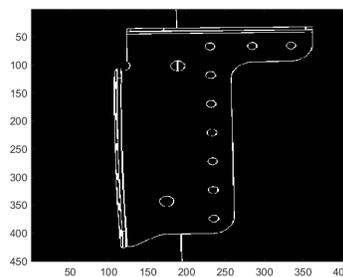
(g) 4.7d image finale segmentée

Figure B.4 – Image synthétique segmentée 4.7d par application de plusieurs pré-traitements successifs

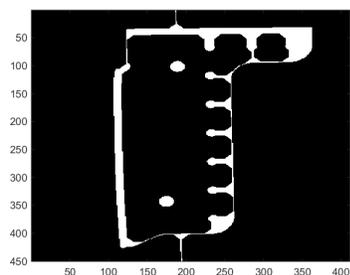
B.5 Résultat de segmentation de l'objet 4.7e



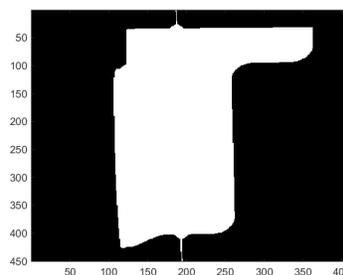
(a) 4.7e en niveaux de gris



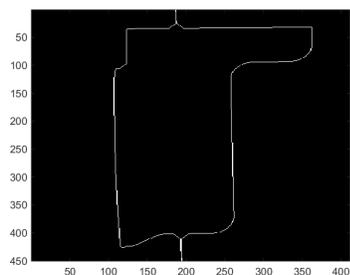
(b) 4.7e binarisée



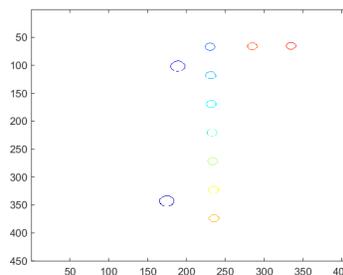
(c) 4.7e après fermeture avec un disque de taille 15



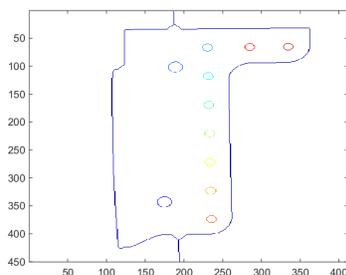
(d) 4.7e après remplissage des trous



(e) 4.7e contour extérieur de l'objet



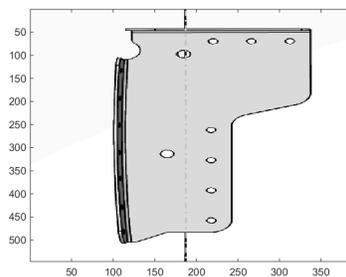
(f) 4.7e régions internes



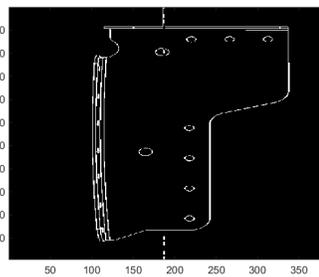
(g) 4.7e image finale segmentée

Figure B.5 – Image synthétique segmentée 4.7e par application de plusieurs pré-traitements successifs

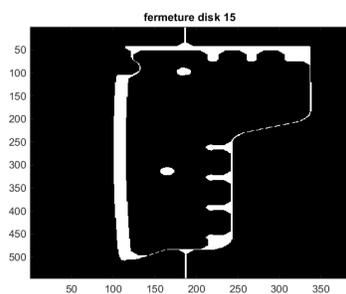
B.6 Résultat de segmentation de l'objet 4.7f



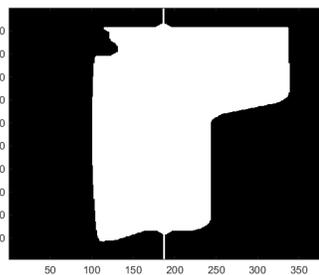
(a) 4.7f en niveaux de gris



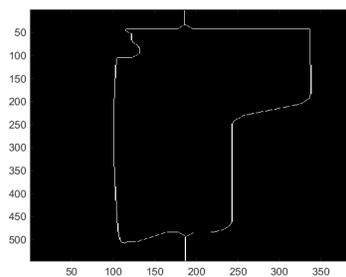
(b) 4.7f binarisée



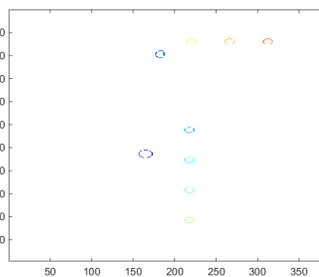
(c) 4.7f après fermeture avec un disque de taille 15



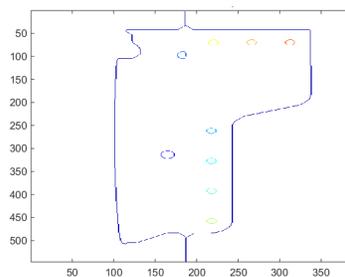
(d) 4.7f après remplissage des trous



(e) 4.7f contour extérieur de l'objet



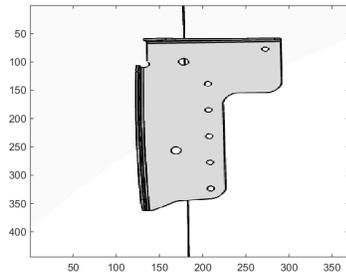
(f) 4.7f régions internes



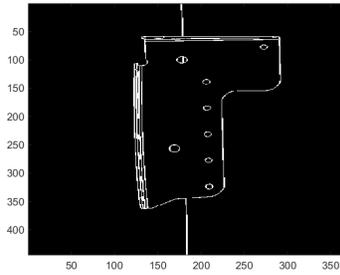
(g) 4.7f image finale segmentée

Figure B.6 – Image synthétique segmentée par application de plusieurs pré-traitements successifs

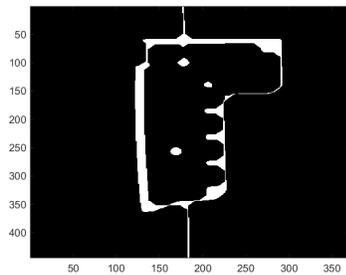
B.7 Résultat de segmentation de l'objet 4.7g



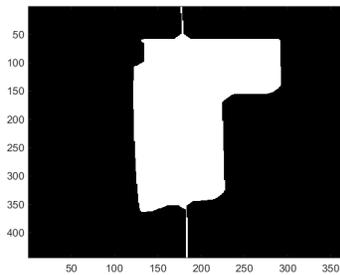
(a) 4.7g en niveaux de gris



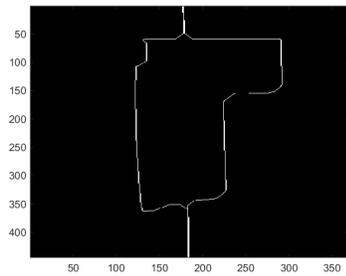
(b) 4.7g binarisée



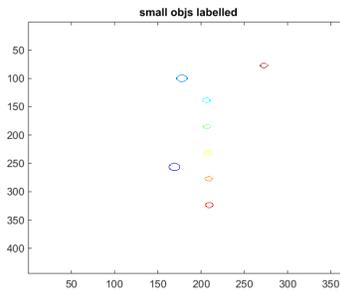
(c) 4.7g après fermeture avec un disque de taille 15



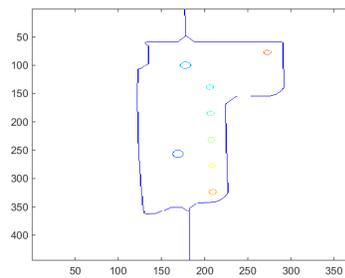
(d) 4.7g après remplissage des trous



(e) 4.7g contour extérieur de l'objet



(f) 4.7g régions internes



(g) 4.7g image finale segmentée

Figure B.7 – Image synthétique 4.7g segmentée par application de plusieurs pré-traitements successifs

Annexe C

Résultats d'extraction de primitives sur les images segmentées de l'annexe B

Pour une meilleure appréciation des résultats d'extraction des primitives, ellipses par l'algorithme de Prasad et segments de droite par la transformée de Hough standard nous montrons ci-dessous les images synthétiques originales. Pour extraire les primitives nous travaillons sur les images segmentées présentées dans l'annexe B. La figure ci-dessous C.1 est la même que la figure 4.7 que nous avons vue au **chapitre 4**.

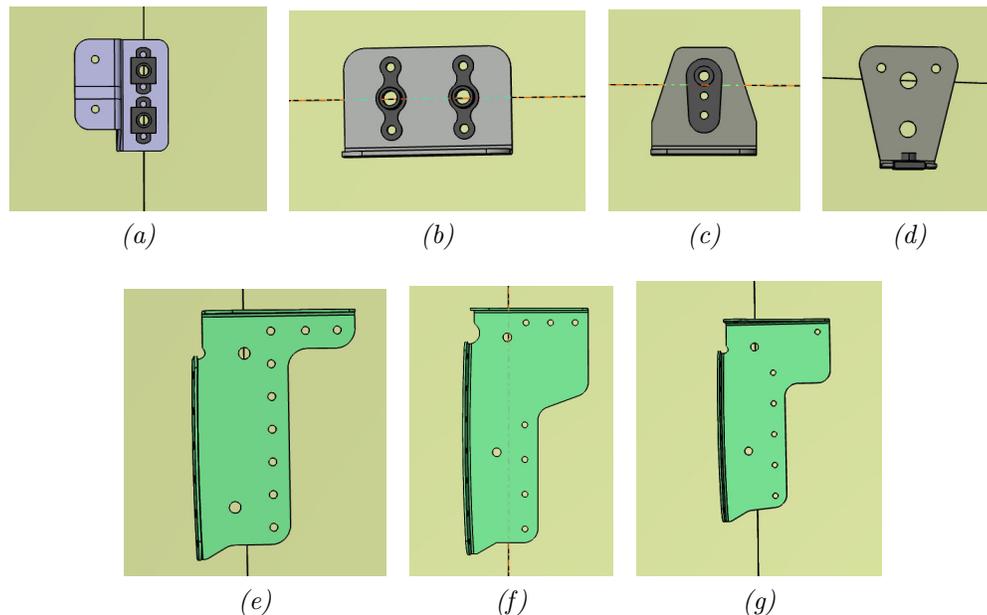
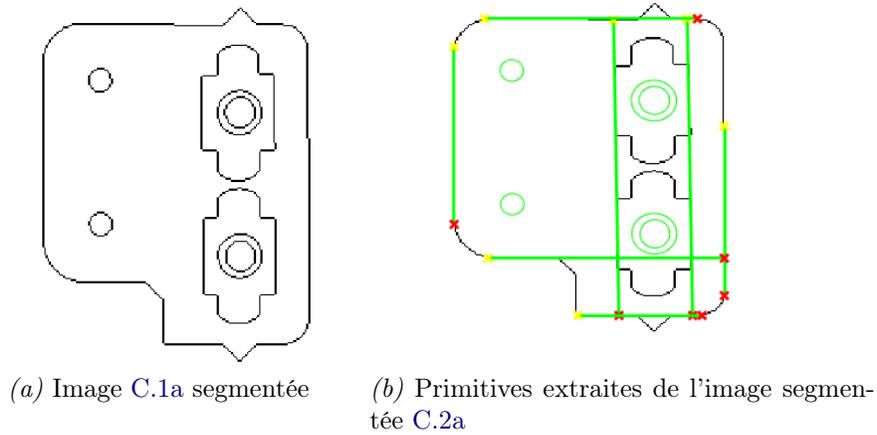


Figure C.1 – Images générées à partir du modèle CAO de l'assemblage, en format ".bmp"

Pour chaque objet considéré, nous présentons à la fois l'image segmentée sur laquelle nous avons procédé à l'extraction des primitives et les deux ensembles de primitives (ellipses et segments de droite). Les images segmentées sur lesquelles nous avons travaillé lors de l'extraction de primitives sont très proches des images segmentées montrées dans l'annexe B. La seule différence entre ces images réside dans le fait qu'avant l'extraction de primitives nous avons activé l'algorithme de croissance de région au centre de l'objet d'intérêt, ce qui a permis d'enlever les branches extérieures qui matérialisent la fixation des objets et qui pourraient être confondues avec des segments de droite.

C.1 Extraction de primitives de l'image C.1a



(c) Primitives numérotées obtenues de C.1a

Figure C.2 – Extraction d'ellipses et de segments de droite (en vert) de l'image segmentée C.2a

C.2 Extraction de primitives de l'image C.1b

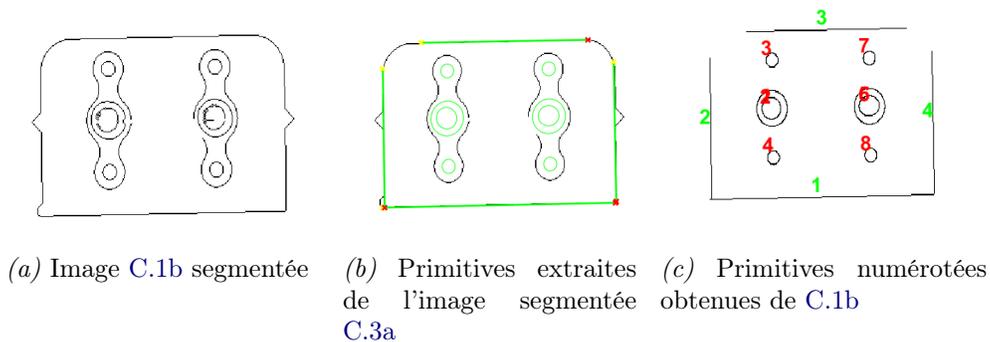
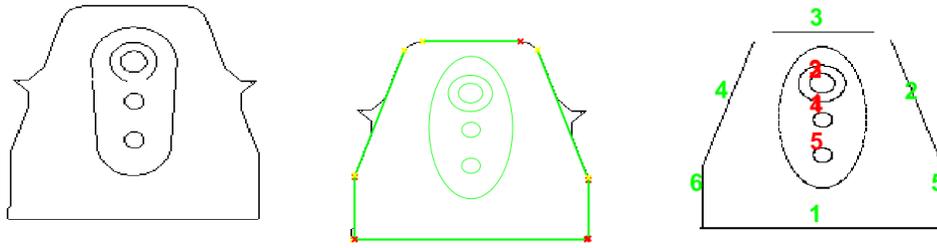


Figure C.3 – Extraction d'ellipses et de segments de droite (en vert) de l'image segmentée C.3a

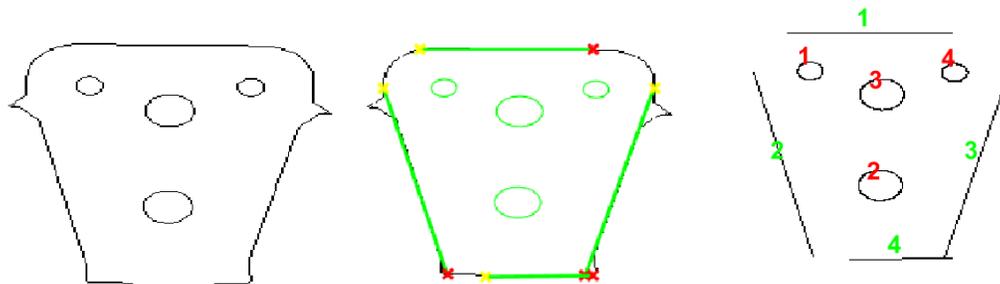
C.3 Extraction de primitives de l'image C.1c



(a) Image C.1c segmentée (b) Primitives extraites de l'image segmentée C.4a (c) Primitives numérotées obtenues de C.1c

Figure C.4 – Extraction d'ellipses et de segments de droite (en vert) de l'image segmentée C.4a

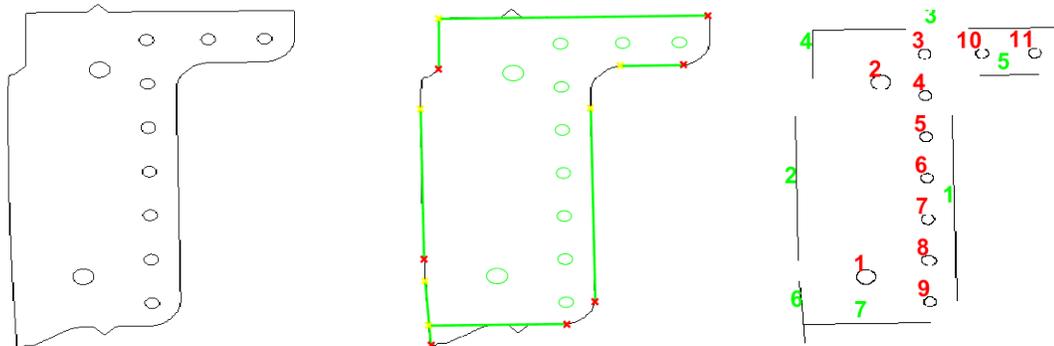
C.4 Extraction de primitives de l'image C.1d



(a) Image C.1d segmentée (b) Primitives extraites de l'image segmentée C.5a (c) Primitives numérotées obtenues de C.1d

Figure C.5 – Extraction d'ellipses et de segments de droite (en vert) de l'image segmentée C.5a

C.5 Extraction de primitives de l'image C.1e



(a) Image C.1e segmentée (b) Primitives extraites de l'image segmentée C.6a (c) Primitives numérotées obtenues de C.1e

Figure C.6 – Extraction d'ellipses et de segments de droite (en vert) de l'image segmentée C.6a

C.6 Extraction de primitives de l'image C.1f

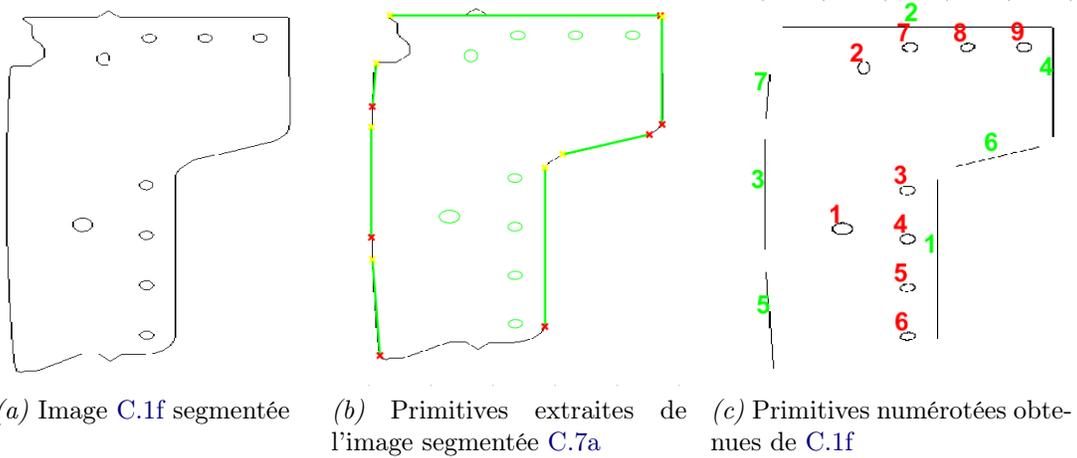


Figure C.7 – Extraction d'ellipses et de segments de droite (en vert) de l'image segmentée C.7a

C.7 Extraction de primitives de l'image C.1g

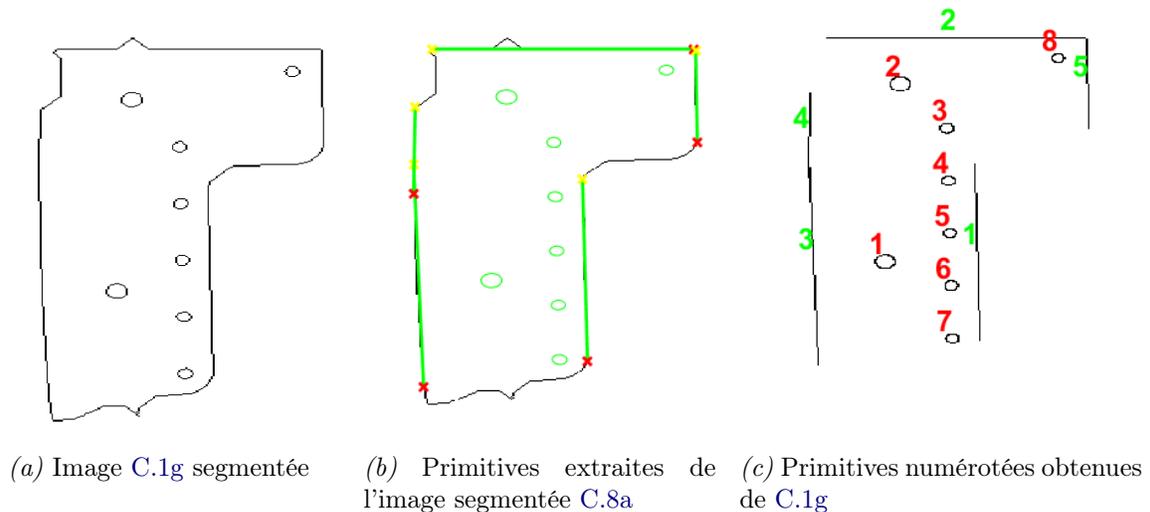


Figure C.8 – Extraction d'ellipses et de segments de droite (en vert) de l'image segmentée C.8a

Annexe D

Expérimentations réalisées pour obtenir les données réelles

Cette annexe décrit et illustre (de façon non exhaustive) les expérimentations réalisées en vue de l'obtention des données réelles pour le test de la méthode d'appariement des primitives que nous avons décrite dans le **chapitre 4**. On y décrit les deux stratégies d'obtention des données dans le contexte du projet Lynx@que nous avons déjà évoquées lors de l'introduction, dans la section 1.2.2.3 et dans la section 3.2.2.

D.1 Positionner d'abord la caméra virtuelle dans la maquette numérique puis positionner la caméra réelle dans l'environnement réel.

Afin de pouvoir comparer un objet issu d'un modèle théorique de conception et un objet acquis avec une caméra, il est incontournable que les deux objets soient dans le même repère. Parmi les deux stratégies possibles, le pilotage de la caméra réelle par le modèle CAO se déroule en deux étapes ci-dessous :

- D.1.1. Dans la maquette numérique, choisir un point de vue pertinent pour l'objet d'intérêt. Ce point de vue est atteint grâce notamment au placement d'une caméra virtuelle.
- D.1.2. Enregistrer les paramètres importants permettant de reproduire le point de vue théorique avec la caméra réelle. Ces paramètres sont notamment l'origine du repère choisi, la distance de la caméra à l'objet dans le monde virtuel et au moins deux autres points, par exemple la distance du centre de l'objet à deux points (un point par axe) sur les axes du repère choisi. Il est à noter que si la scène virtuelle contient des amers naturels susceptibles d'exister dans la scène réelle, ils peuvent être considérés pour faciliter cette étape de localisation.

Un point fondamental à considérer dans cette façon de procéder est la connaissance de la correspondance du facteur d'échelle entre la caméra virtuelle et la caméra réelle. Le bon facteur d'échelle doit être employé lors du réglage de la caméra réelle afin que les objets à comparer soient non seulement localisés au même endroit dans les deux scènes mais qu'ils aient également des dimensions comparables dans le plan image. Ce facteur d'échelle peut être déduit assez facilement de manière expérimentale. Dans notre cas, nous avons réalisé une table de correspondance donnant différentes valeurs de zoom à appliquer à la caméra réelle pour que l'objet dans l'image acquise avec la caméra PTZ ait est les mêmes dimensions que l'objet dans l'image générée à partir du modèle CAO. Et dans ce cas de caméra

PTZ les angles Pan et Tilt à appliquer à la caméra PTZ sont déduits à partir de relations trigonométriques issues des mesures effectuées à l'aide du logiciel de conception.

Une fois que la caméra théorique est placée dans la maquette numérique et que l'étape de la section D.1 est réalisée, la suite consiste à générer une image théorique à l'aide des différentes options du logiciel de conception. Pour obtenir l'image réelle, il faut placer la caméra réelle en respectant les contraintes fixées dans le point D.1.2. de cette même section D.1.

Le placement d'un objet de la CAO dans une scène virtuelle peut donner des indications intéressantes permettant de placer la caméra réelle lors d'une application d'inspection ou de comparaison entre un objet théorique issu du modèle CAO et un objet réel. Néanmoins, cette façon de procéder impose des contraintes physiques très fortes lors du placement de la caméra réelle. À savoir, la distance entre la caméra et la scène virtuelle doit être minutieusement respectée lorsque la caméra physique est positionnée. De surcroît, il faut établir une correspondance au niveau du facteur d'échelle entre les deux caméras, virtuelle et réelle. Si ce deuxième point, bien que potentiellement fastidieux n'est pas bloquant, le premier point pose problème car il est physiquement difficile de placer la caméra réelle au point déterminé par le placement de la caméra virtuelle et même d'atteindre une précision suffisante qui puisse permettre à un algorithme de recalage de compenser l'écart éventuel qu'il pourrait y avoir entre le placement des deux caméras. Notre tentative de mettre en place cette méthodologie de localisation n'a pas été concluante.

D.2 Positionner d'abord la caméra réelle calibrée dans l'environnement réel et estimer sa pose

L'objectif de la méthode étant de comparer l'objet d'intérêt dans l'image réelle acquise avec la caméra à son vis-à-vis théorique obtenu à partir du modèle CAO, il était nécessaire de résoudre le problème de localisation entre le modèle théorique et la caméra réelle. Les étapes pour obtenir les données synthétiques sont les suivantes :

D.2.1. Calibrage de la caméra.

Le but du calibrage est de trouver les différents paramètres intrinsèques de la caméra et de corriger les distorsions dues à l'optique qu'une image peut subir. Nous avons testé trois méthodes différentes de calibrage. Le calibrage de base présent dans la librairie OpenCV qui est une implémentation en C++ de la méthode de Bouguet [Bouguet,] et la méthode de calibrage itérative proposée par Ankur Datta dans [Datta et al., 2009]. Nous avons aussi testé une troisième méthode qui consiste en

un calibrage par le logiciel propriétaire Vic3D ©.

Pour la première méthode nous avons utilisé une mire en damier et une mire circulaire. Pour les tests avec la deuxième méthode nous avons utilisé une mire circulaire et une mire en anneau. Les résultats les plus encourageants ont été obtenus avec le logiciel Vic3D ©. C'est cette méthode qui a permis d'obtenir la plus faible erreur de reprojection. Les résultats obtenus en utilisant la mire en anneau, dans le cadre de la méthode itérative [Datta et al., 2009] montrent une erreur de reprojection plus élevée comparée au calibrage avec le système Vic3D ©. Néanmoins, cette méthode permet de mettre en place un système de calibrage plus pratique. Il s'agit ici d'un problème de simplicité face à la précision. Dans tous les cas, on ne peut vraiment juger cette étape qu'après la procédure d'estimation de pose, de projection du modèle et de génération de l'image synthétique car c'est seulement en fin de procédure que l'on peut comparer l'écart entre l'image théorique générée et l'image réelle capturée avec la caméra réelle calibrée.

D.2.2. Estimation de la pose.

Le problème de localisation est aussi bien une affaire de détermination de paramètres intrinsèques de la caméra mais c'est surtout une affaire de géométrie mettant en jeu la position de la caméra au moment de la prise de vue, par rapport à la scène 3D à étudier. Cette position de la caméra est donnée par les paramètres extrinsèques de la caméra, une matrice de rotation de taille 3 et un vecteur (colonne par convention) de translation, de taille 3×1 . De manière simplifiée, on peut assimiler l'estimation de pose à un processus de mise en correspondance entre les points 3D connus de la scène réelle (monde) avec les points 2D dans le plan image. Il a été démontré qu'il s'agit d'un problème d'optimisation non linéaire. Ce problème d'estimation de pose est souvent résolu par une méthode d'optimisation non linéaire, de type Levenberg-Marquardt. Pour obtenir les paramètres extrinsèques de la caméra, nous avons employé deux stratégies différentes selon la méthode de calibrage utilisée, soit le calibrage par le logiciel Vic3D © ou par la méthode itérative de [Datta et al., 2009]. En effet, pour calibrer la caméra, il y a déjà une mise en correspondance entre les points 3D et les points 2D de la scène qui peut permettre de récupérer directement les paramètres de localisation pour chaque mire plane utilisée lors du calibrage. Ceci est le cas notamment de la méthode itérative de [Datta et al., 2009]. Bien que cette méthode ne fournisse pas automatiquement les paramètres extrinsèques à la fin de la procédure de calibrage, elle les calcule pendant cette phase de calibrage. Nous avons donc pu extraire ces paramètres en indiquant au programme de les donner en fin de procédure, après la convergence de la méthode de calibrage. En revanche

ceci n'est pas possible avec le logiciel propriétaire Vic3D © dont le code source n'est pas libre. Il est néanmoins possible de récupérer les points 2D utilisés dans le calcul des paramètres intrinsèques. Afin d'obtenir les paramètres extrinsèques lorsque nous avons utilisé Vic3D ©, nous avons récupéré les points 2D fournis par le logiciel pour une position de mire jugée pertinente. Il s'agit ici d'une position facilement repérable sur le modèle CAO en vue de l'étape de projection future du modèle CAO sur le plan image. Les points 3D de la mire étant connus, nous avons pu effectuer l'optimisation souhaitée en utilisant l'implémentation d'OpenCV de la méthode d'optimisation de Levenberg-Marquadt et nous avons ainsi pu obtenir les paramètres extrinsèques.

D.2.3. Génération des données à partir du modèle CAO pour une mise en correspondance entre un objet théorique avec un objet réel

Solution 1 : génération d'une image synthétique.

Après l'obtention des paramètres intrinsèques et de la pose il devient alors possible d'utiliser ces données pour projeter le modèle CAO sur le plan image. Cette tâche est réalisée en deux phases. Pour commencer, dans la maquette numérique, on isole le modèle CAO de l'élément à évaluer. Ce modèle CAO est ensuite converti au format STL, c'est à dire en ensemble de triangles pour lesquels les coordonnées de chaque point sont connues. Pour finir, ces points, constituants les triangles, sont projetés dans le plan image en appliquant le modèle de projection perspective en tenant compte notamment des paramètres de distorsions optiques calculés pendant la phase de calibrage. Après la projection de l'élément du modèle CAO sur le plan image, l'ensemble des triangles vides projetés subissent une procédure de remplissage qui permet de générer la surface pour tous les triangles et de respecter les endroits où il doit y avoir un trou. A la fin, connaissant la taille de l'image réelle, on peut créer une image matricielle de même taille que l'image réelle et qui contient les triangles projetés et remplis, c'est à dire l'objet CAO.

Solution 2 : Projection directe des primitives CAO.

Cette solution consiste à projeter directement sur l'image réelle les éléments du modèle CAO qui nous intéresse, en réalité augmentée, sans passer par l'intermédiaire d'une image synthétique.

Afin d'obtenir les données pouvant servir au test de l'algorithme présenté dans le **chapitre 4**, nous avons utilisé dans un premier temps une caméra PTZ avec l'assemblage mis à notre disposition par notre collaborateur G2Metric puis une caméra CCD classique dans un deuxième temps, en utilisant cette fois-ci le panneau mis à disposition par la société Aéroliia dans le cadre du projet CAAMVis. Nous allons dans les sections suivantes donner

des détails qui sont propres à chacune de ces deux expérimentations.

D.3 Expérimentation avec la caméra PTZ et assemblage mis à disposition par G2Metric

Une première expérience a été réalisée avec notre caméra PTZ face à l'assemblage mis à disposition par G2Metric. Pour rappel, le modèle CAO de cet assemblage a été généré en rétro-conception et de ce fait nous avons remarqué quelques problèmes. Le modèle CAO et l'assemblage réel présentent des divergences notoires (formes, point de fixation différents), et on a pu le constater lors de la projection des éléments du modèle sur l'image réelle. Néanmoins cette expérimentation a été utile car en se focalisant sur un élément particulier lors de cette première expérimentation cf. figure D.1 on constate que pour aller plus loin, il est important de disposer de plusieurs vues du même élément.

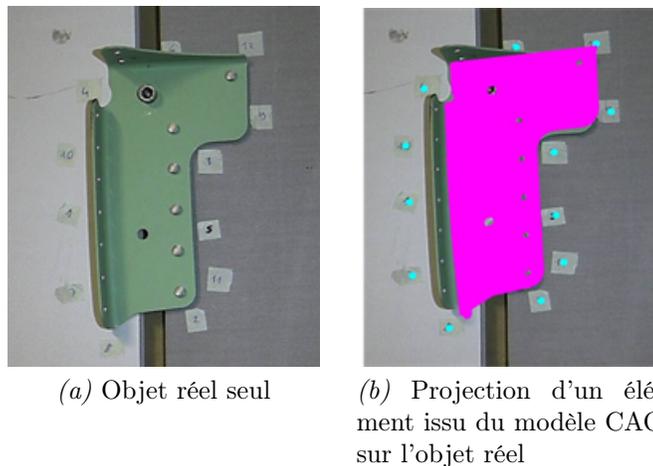


Figure D.1 – Projection d'un élément CAO sur l'image réelle. Illustration du problème de point de vue. Les détails (parties saillantes) de l'objet qui se trouvent dans l'orientation de la caméra sont écrasés pendant la projection

En effet, une projection faite à partir d'un certain angle ne nous donne que l'information possible pour cet angle de vue, et celle-ci peut s'avérer insuffisante.

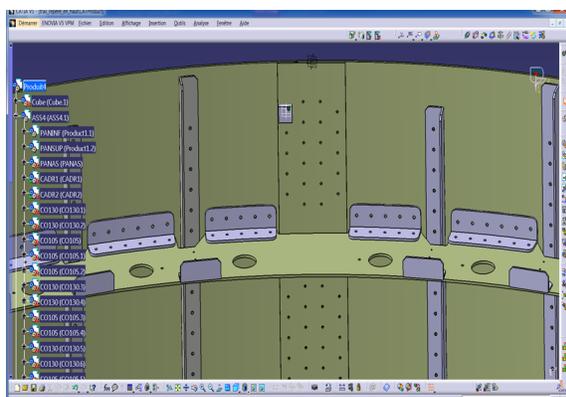
La caméra PTZ utilisée bien qu'elle puisse être dirigée horizontalement et verticalement n'est pas suffisamment maniable pour assurer des positionnements spécifiques qui pourraient permettre l'obtention d'une vue plus adaptée.

Ces constats nous ont conduit à utiliser un autre assemblage réel (celui qui nous a été mis à disposition par la société Aéroliia dans le cadre du projet CAAMVis, cf. figure D.2) lors des expérimentations futures que nous avons réalisées en utilisant des caméras standard montées sur trépied et donc plus maniables en comparaison avec la caméra PTZ.

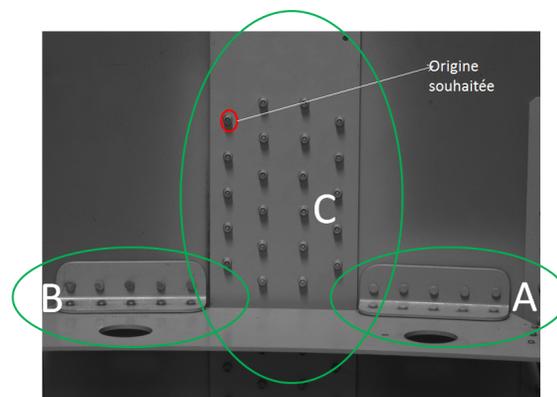
D.4 Expérimentation avec une caméra CCD standard et l'assemblage mis à disposition par Aéroliia

Dans cette section, nous allons illustrer l'expérimentation réalisée avec une caméra CCD standard dans le but d'obtenir les données réelles. Ici, nous illustrons la procédure de calibrage, d'estimation de pose et de projection des éléments du modèle CAO en utilisant la mire en anneau et la méthode de calibrage de [Datta et al., 2009].

Sur la figure D.2, on voit à gauche une image du modèle CAO d'un panneau et à droite une image du panneau réel en focalisant sur trois objets étiquetés "A", "B", "C". On va essayer de faire en sorte de projeter les objets du modèle CAO sur l'image réelle. Dans les deux cas on distingue aisément le centre du repère qui est clairement défini dans la figure D.3.



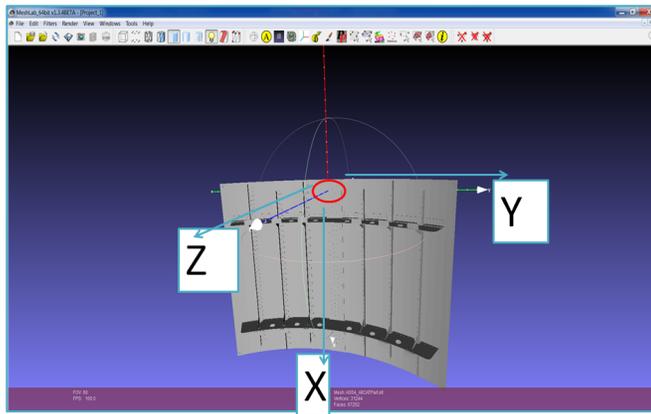
(a) Modèle CAO d'assemblage utilisé



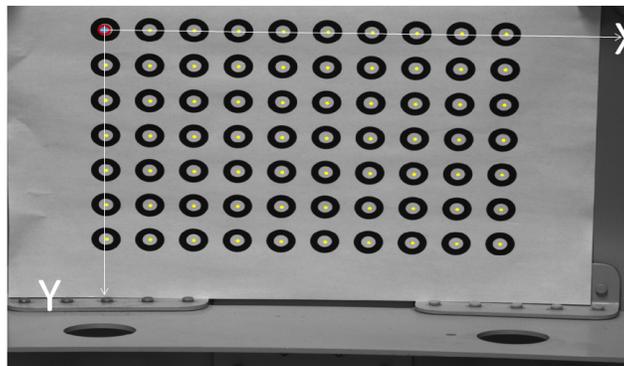
(b) Assemblage réel : zoom sur trois objets

Figure D.2 – CAO de l'assemblage et assemblage réel

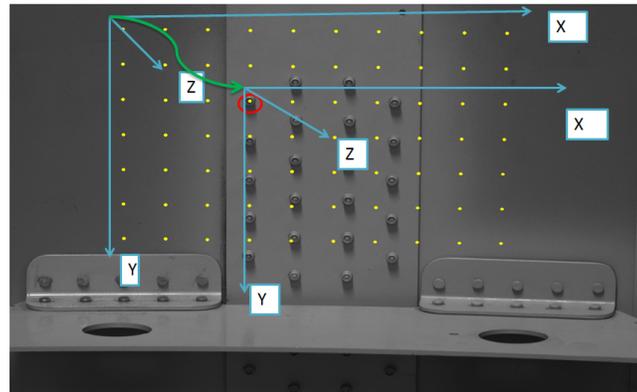
Sur la figure D.3, on précise les conventions prises pour définir les repères, et en CAO et sur le panneau réel ainsi que sur la mire en anneau utilisée pour le calibrage.



(a) Modèle CAO convention de repères



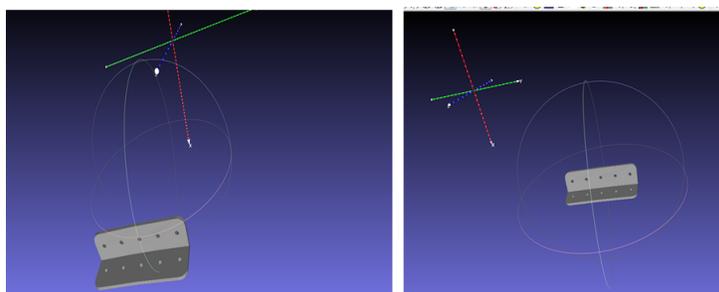
(b) Mire circulaire et convention de repères



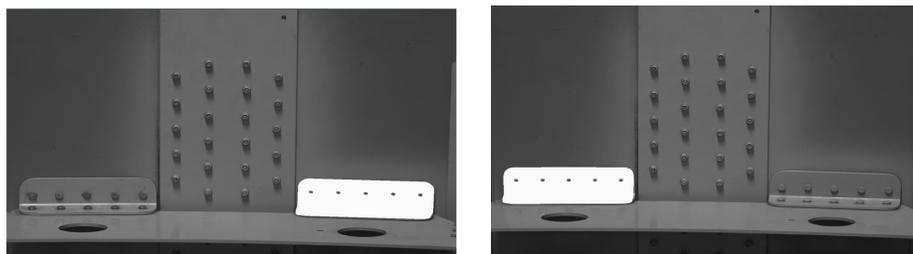
(c) Assemblage réel et convention des repères

Figure D.3 – Conventions de repères

Une fois les repères définis, nous avons déroulé la procédure de calibrage, estimation de pose et projection décrite dans l'annexe D, à la section D.2. Les résultats obtenus pour les objets "A" et "B" de la figure D.2b sont illustrés par les figures D.4b et D.4c



(a) Zoom sur deux éléments du modèle dans la maquette numérique



(b) Projection de l'élément "A" sur l'image réelle (c) Projection de l'élément "A" sur l'image réelle

Figure D.4 – Projections d'éléments de la CAO sur l'assemblage réel

On constate que la projection des objets "A" et "B" est perfectible, dans le sens où les trous sur la partie inférieure des éléments "A" et "B" ne sont pas visibles. En effet, elle est conditionnée par le choix du point de vue. À défaut d'avoir un aspect de l'objet qui nous permet d'obtenir toutes les primitives à partir d'une seule vue, nous pensons qu'une alternative envisageable serait de prendre un ensemble de vues représentatifs qui permettraient de projeter des parties individuelles de ces objets (par exemple deux vues dans ce cas) de faire l'appariement pour les parties séparées d'agréger les résultats de chaque vue afin de produire un résultat unique.

La projection des objets plats pose moins de problème, car il est plus facile d'avoir une vue (fronto-parallèle) qui permet de récupérer l'ensemble des primitives de l'objet du modèle CAO projeté (avec les paramètres de calibration et de la pose estimée au préalable). Sur la figure D.5, on voit que l'objet "C" est partiellement masqué par les équerres de fixation. Pour cette raison, pour avoir uniquement la partie supérieure de l'objet "C", il aurait fallu projeter également la première équerre de fixation du modèle (cf. figure D.2) de sorte à pouvoir ensuite couper l'objet "C" à son intersection avec cette équerre de fixation. Ces constats montrent que la projection du modèle sur le plan image peut, en pratique, poser plus de problèmes que ceux que l'on peut attendre en théorie.

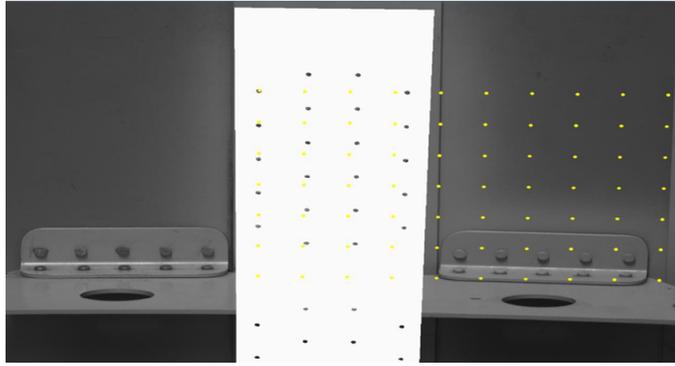


Figure D.5 – Projection du modèle CAO correspondant à l'objet réel "C"

Références

- [Alahi et al., 2012] A. Alahi, R. Ortiz, and P. Vandergheynst (June 2012). FREAK : Fast Retina Keypoint. *In Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference*, pages 510 – 517. (Cité page 26.)
- [Baggio et al., 2012] Daniel Lélis Baggio, Shervin Emami, David Millán Escrivá, Khvedchenia Ievgen, Naureen Mahmood, and Jason Saragih (December, 2012). *Mastering OpenCV with Practical Computer Vision Projects*. Packt Publishing. (Cité pages 27 et 52.)
- [Bai and Latecki, 2008] Xiang Bai and Longin Jan Latecki (2008). Path similarity skeleton graph matching. *Pattern Analysis and Machine Intelligence*, 30 :1282 – 1292. (Cité pages 35, 84 et 125.)
- [Bay et al., 2005] Herbert Bay, Vittorio Ferrari, and Luc Van Gool (2005). Wide-baseline stereo matching with line segments. *Proceedings of Computer Vision and Pattern Recognition*, 1 :329–336. (Cité page 81.)
- [Bay et al., 2006] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool (2006). SURF : Speeded Up Robust Features. *ECCV*, 3951 :404–417. (Cité page 24.)
- [Bengoetxea, 2002] E. Bengoetxea (2002). *Inexact Graph Matching Using Estimation of Distribution Algorithms*. PhD thesis, Ecole Nationale Supérieure des Télécommunications, Paris, France. (Cité page 84.)
- [Berndt, 2011] Dirk Berndt (2011). Worker assistance and quality inspection. (Cité pages 41 et 43.)
- [Besl and McKay, 1992] Paul Besl and Neil McKay (1992). A Method for Registration of 3-D Shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2) :239 –256. (Cité pages 41 et 46.)
- [Bilal et al., 2015] Karabagli Bilal, Simon Thierry, and Orteu Jean-José (2015). A new chain-processing-based computer vision system for automatic checking of machining set-up application for machine tools safety. *The International Journal of Advanced Manufacturing Technology*, pages 1–22. (Cité page 43.)
- [Bouguet,] Jean-Yves Bouguet. Camera calibration toolbox for matlab. (Cité page 145.)
- [Bourgeois, 2006] Steve Bourgeois (2006). *Alignement d’Objets Mécaniques Complexes par Vision Monoculaire*. PhD thesis, Clermont Ferrand University, France. (Cité page 40.)
- [Boutarfa et al., 2006] Abdelhalim Boutarfa, Nour-Eddine Bouguechal, Yassine Abdessemed, and Tanneguy Redarce (2006). Pattern Recognition in Computer Integrated Manufacturing. *Journal of Electrical Engineering*, 57 :28–35. (Cité pages 40 et 42.)
- [Briechle and Hanebeck, 2001] Kai Briechle and Uwe D. Hanebeck (2001). Template matching using fast normalized cross correlation. *Optical Pattern Recognition XII*. (Cité page 21.)

-
- [Bruna et al., 2013] Arcangelo Bruna, Giovanni Maria Farinella, Giuseppe Claudio Guarnera, and Sebastiano Battiato (2013). Forgery detection and value identification of euro banknotes. *Sensors*, pages 1–15. (Cité page 36.)
- [Bukovec et al., 2007] M. Bukovec, Z. Spiclin, and F. Pernus (2007). Automated visual inspection of imprinted pharmaceutical tablets. *Meas. Sci.Tech.*, 18 :2921–2930. (Cité pages 6 et 37.)
- [Bunke, 2000] Horst Bunke (2000). Graph matching : Theoretical foundations, algorithms, and applications. In *International Conference on Vision Interface*. (Cité page 66.)
- [Calonder et al.,] M. Calonder, V. Lepetit, C. Strecha, and P. Fua. BRIEF : Binary Robust Independent Elementary Features. In *European Conference on Computer Vision, 2010*. (Cité page 25.)
- [Canclini et al., 2013] A. Canclini, M. Cesana, A. Redondi, M. Tagliasacchi, J. Ascenso, and R. Cilla (2013). Evaluation of low-complexity visual feature detectors and descriptors. *18th International Conference on Digital Signal Processing (DSP)*. (Cité page 26.)
- [Chawla and Deb, 2012] Palkush Rai Chawla and Sankha Deb (2012). A computer-aided inspection methodology for mechanical parts based on machine vision. *Proceedings of 4th Int. and 25th AIMTDR Conference, Jadavpur University, Kolkata, Dec 2012*. (Cité pages 40 et 122.)
- [Chuang et al., 2010] Shui-Fa Chuang, Wen-Tung Chang, Chih-Cheng Lin, and Yeong-Shin Tarn (2010). Misalignment inspection of multilayer pcbs with an automated x-ray machine vision system. *International Journal of Advanced Manufacturing and Technology*, 51 :995–1008. (Cité pages 19 et 37.)
- [Cimikowski, 1992] Robert J. Cimikowski (1992). Properties of some euclidean proximity graphs. *Pattern Recognition Letters*, 13 :417–423. (Cité page 66.)
- [Correa and Lindstrom, 2012] Carlos D. Correa and Peter Lindstrom (2012). Locally-scaled spectral clustering using empty region graphs. *ACM, Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1330–1338. (Cité page 81.)
- [Cubero et al., 2011] Sergio Cubero, Nuria Aleixos, Enrique Moltó, Juan Gómez-Sanchis, and Jose Blasco (2011). Advances in machine vision applications for automatic inspection and quality evaluation of fruits and vegetables. *Food Bioprocess Technol*, 4 :487 – 504. (Cité pages 18 et 19.)
- [Dalal and Triggs, 2005] Navneet Dalal and Bill Triggs (2005). Histograms of oriented gradients for human detection. *Computer Vision and Pattern Recognition*, 2. (Cité page 26.)
- [Datta et al., 2009] Ankur Datta, Jun-Sik Kim, and Takeo Kanade (2009). Accurate camera calibration using iterative refinement of control points. *12th International Conference on Computer Vision Workshops (ICCV Workshops)*, 4322 :1201 – 1208. (Cité pages 145, 146 et 149.)

-
- [Davies, 2012] E. R. Davies (2012). *Computer and Machine Vision : Theory, Algorithms, Practicalities*. Elsevier. (Cité pages 36 et 40.)
- [Dawson-Howe, 2014] Kenneth Dawson-Howe (2014). *A Practical Introduction to Computer Vision with OpenCV*. Wiley. (Cité pages 40 et 52.)
- [Dubey et al., 2012] Deepika Dubey, Abhishek Jain, and Uday Pratap Singh (2012). An overview on : Image alignment and open issues. *International Journal of Advanced Research in Computer Science and Software Engineering*, 2. (Cité page 73.)
- [Duda and Hart, 1972] Richard O. Duda and Peter E. Hart (1972). Use of the Hough Transformation to detect lines and curves in pictures. *Communications of ACM*, 15 :11–15. (Cité page 75.)
- [Excelitas, 2015] Excelitas (2015). Excelitas website. <http://www.excelitas.com/sitepages/downloads.aspx?Product=CCDLinearCameras>, consulté le 03/01/2015. (Cité page 19.)
- [Far, 2005] Aicha Beya Far (2005). *Analyse multi-images Application à l'extraction contrôlée d'indices images et à la détermination de descriptions scéniques*. PhD thesis, Université Louis Pasteur, France. (Cité pages 40 et 42.)
- [Ferreira, 2015] Manuel Joao Ferreira (2015). Visum summer school. Atmosfera-M, Porto, Portugal. (Cité page 38.)
- [Fishkel et al., 2006] F. Fishkel, A. Fischer, and S. Ar (2006). Verification of engineering models based on bipartite graph matching for inspection applications. *Springer-Verlag Berlin Heidelberg. LNCS 4077*, pages 485–499. (Cité pages 40, 42 et 87.)
- [Gangodkar et al., 2012] Durgaprasad Gangodkar, Sachin Gupta, Gurbinder Singh Gill, Padam Kumar, and Ankush Mittal (2012). Efficient variable size template matching using fast normalized cross correlation on multicore processors. *ADCONS, LNCS 7135*, page 218–227. (Cité pages 21 et 22.)
- [García et al., 2015] Gloria Bueno García, Oscar Deniz Suarez, José Luis Espinosa Aranda, Jesus Salido Tercero, Ismael Serrano Gracia, and Noelia Váñez Enano (March, 2015). *Learning Image Processing with OpenCV*. Packt Publishing. (Cité page 23.)
- [Golnabi and Asadpour, 2007] H. Golnabi and A. Asadpour (2007). Design and application of industrial machine vision systems. *Robotics and Computer-Integrated Manufacturing*, 23 :630–637. (Cité page 6.)
- [Goodfellow et al., 2016] Ian Goodfellow, Yoshua Bengio, and Aaron Courville (2016). Deep learning. Book in preparation for MIT Press. (Cité page 32.)
- [Hackeling, 2014] Gavin Hackeling (2014). *Mastering Machine Learning with scikit-learn*. Packt Publishing Ltd. (Cité pages vii, 29, 30 et 32.)
- [Haller, 2015] Stefan Haller (12 january 2015). An Update on Features : SURF, BRISK, ORB and FAST. *Technische Universität Dresden*, pages 1–69. (Cité pages 25 et 26.)
-

-
- [Harvent, 2010] Jacques Harvent (2010). *Mesure de formes par corrélation multi-images : application à l'inspection de pièces aéronautiques à l'aide d'un système multi-caméras*. PhD thesis, Université de Toulouse Paul Sabatier. (Cité page 39.)
- [Heinly et al., 2012] Jared Heinly, Enrique Dunn, and Jan-Michael Frahm (2012). Comparative evaluation of binary features. *ECCV 2012*. (Cité page 26.)
- [Hittawe et al., 2015] Mohamad Mazen Hittawe, Désiré Sidibé, and Fabrice Mériaudeau (2015). A machine vision based approach for timber knots detection. *Twelfth International Conference on Quality Control by Artificial Vision, Proc. of SPIE*, 9534 :95340L–1 – 95340L–8. (Cité page 40.)
- [Hong-Seok and Mani, 2014] Park Hong-Seok and Tuladhar Upendra Mani (2014). Development of an inspection system for defect detection in pressed parts using laser scanned data. *Elsevier, Procedia Engineering 69, 24th DAAAM International Symposium on Intelligent Manufacturing and Automation*, 69 :931 – 936. (Cité page 41.)
- [Hutter and Brewer, 2009] Marcus Hutter and Nathan Brewer (2009). Matching 2-D ellipses to 3-D circles with application to vehicle pose identification. *Proceedings of International Conference Image and Vision Computing*, pages 153 – 158. (Cité pages 81 et 125.)
- [Jovancevic et al., 2015] Igor Jovancevic, Jean-José Orteu, Thierry Sentenac, and Réemi Gilblas (2015). Automated visual inspection of an airplane exterior. 9534 :95340Y–1 – 95340Y–9. (Cité page 39.)
- [Jovancevic et al., 2016] Igor Jovancevic, Ilísio Viana, Jean-José Orteu, Thierry Sentenac, and Stanislas Larnier (2016). Matching CAD model and image features for robot navigation and inspection of an aircraft. *Fifth International Conference on Pattern Recognition Applications and Methods, Rome, Italy*. (Cité pages 14 et 124.)
- [Jung et al., 2000] Soo-Mok Jung, Sung-Chul Shin, Hyunki Baikand, and Myong-Soon Park (2000). New fast successive elimination algorithm. *Circuits and Systems, 2000. Proceedings of the 43rd IEEE Midwest Symposium*, 2 :616 – 619. (Cité page 23.)
- [Karabagli, 2013] Bilal Karabagli (2013). *Vérification automatique des montages d'usinage par vision : Application à la sécurisation de l'usinage*. PhD thesis, Toulouse University, France. (Cité pages 35 et 40.)
- [Klette, 2014] Reinhard Klette (2014). *Concise Computer Vision : An Introduction into Theory and Algorithms*. Springer London Heidelberg New York Dordrecht. (Cité pages 27 et 32.)
- [Kohandani et al., 2006] A. Kohandani, O. Basir, and M. Kamel (2006). A fast algorithm for template matching. *ICIAR, LNCS 4142*, pages 398–409,. (Cité pages 20, 21 et 22.)
- [Kokila and Thangavel, 2013] R. Kokila and P. Thangavel (2013). FFT based image registration using corner response. *International Journal of Computer Science and Electronics Engineering (IJCSEE)*, 1. (Cité page 74.)
-

-
- [Konishi et al., 2012] Yoshinori Konishi, Yasuyo Kotake, Yoshihisa Ijiri, , and Masato Kawade (2012). Fast and precise template matching based on oriented gradients. *ECCV Ws/Demos, Part III, LNCS 7585*, page 607–610. (Cité page 21.)
- [Laganière, 2011] Robert Laganière (2011). *OpenCV 2 Computer Vision Application Programming Cookbook*. Packt Publishing. (Cité page 52.)
- [Leutenegger et al., 2011] Stefan Leutenegger, Margarita Chli, and Roland Y. Siegwart (November 2011). BRISK : Binary Robust Invariant Scalable Keypoints. *In Computer Vision (ICCV), 2011 IEEE International Conference*, pages 2548 – 2555. (Cité page 25.)
- [Lingrand, 2010] Diane Lingrand (2010). *Introduction au traitement d'images : chapitre 5, transformations géométriques*. Vuibert. (Cité page 50.)
- [Liwei Wang et al., 2005] Liwei Wang, Yan Zhang, and Jufu Feng (2005). On the euclidean distance of images. *IEEE Transactions, Pattern Analysis and Machine Intelligence*. (Cité page 49.)
- [Lourakis, 1997] Manolis I. A. Lourakis (1997). Establishing straight line correspondence. *FORTH-ICS / TR-208 : http://users.ics.forth.gr/~lourakis/publ/1997_tr208.pdf*, pages 1–14. (Cité page 109.)
- [Lowe, 2004] D. G. Lowe (2004). Distinctive image features from scale-invariant keypoints. *IJCV*, 2. (Cité page 24.)
- [Lucas, 2010] Yves Lucas (2010). Habilitation à diriger les recherches "systèmes de vision avancés - conception et réglage". Présentée le 24 septembre 2010. (Cité page 41.)
- [Malamas et al., 2003] Elias Malamas, Euripedes Petrakis, Michalis Zervakis, Laurent Petit, and Jean-Didier Legat (2003). A survey on industrial vision systems , applications and tools. *Image and Vision Computing*, 21 :171–188. (Cité page 6.)
- [Malinsky et al., 2012] Milos Malinsky, Roman Peter, Erlend Hodneland, Astri J. Lundervold, Arvid Lundervold, and Jiri Jan (2012). Registration of fa and t1-weighted mri data of healthy human brain based on template matching and normalized cross-correlation. *Journal of Digital Imaging*. (Cité page 21.)
- [Martin and Crowley, 1995] Jerome Martin and James Crowley (1995). Comparison of correlation techniques. *Intelligent Autonomous Systels*. Karlsruhe. (Cité page 21.)
- [Matas et al., 2004] J. Matas, O. Chum, M. Urban, and T. Pajdla (2004). Robust wide-baseline stereo from maximally stable extremal regions. *Image Vision Comput*, 22 :761 – 767. (Cité page 26.)
- [Mattes et al., 2001] D. Mattes, D.R. Haynor, H. Vesselle, T. Lewellen, and W. Eubank (3 July 2001). Non-rigid multimodality image registration. *Proc. Medical Imaging 2001 : Image Processing. SPIE Publications*, 4322 :1609–1620. (Cité page 74.)
- [McIntosh and Mutch, 1988] James H McIntosh and Kathleen M. Mutch (1988). Matching straight lines. *Computer Vision, Graphics and Image Processing*, 43 :386–408. (Cité pages 81, 85, 86 et 124.)
-

- [McLaughlin, 1998] Robert A. McLaughlin (1998). Randomized Hough Transform : Improved ellipse detection with comparison. *Elsevier, Pattern Recognition Letters*, 19 :299–305. (Cité page 75.)
- [Mery and Carrasco, 2006] Domingo Mery and Miguel Carrasco (2006). Advances in image and video technology : Advances on automated multiple view inspection. *Lecture Notes in Computer Science*, Volume 4319 :513–522. (Cité page 122.)
- [Mikolajczyk and Schmid, 2005] Krystian Mikolajczyk and Cordelia Schmid (October 2005). A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27 :1615 – 1630. (Cité page 26.)
- [Miksik and Mikolajczyk, 2012] Ondrej Miksik and Krystian Mikolajczyk (2012). Evaluation of local detectors and descriptors for fast feature matching. *International Conference on Pattern Recognition*, <http://www.miksik.co.uk/papers/miksik2012icpr.pdf>, consulté le 08/10/2016. (Cité page 26.)
- [Modayur et al., 1992] Bharath Modayur, Linda Shapiro, and Robert Haralick (1992). Visual inspection of machined parts. *Computer Vision and Pattern Recognition*. (Cité page 6.)
- [Moore, 2002] Antoni Moore (2002). The case for approximate distance transforms. *The 14th Annual Colloquium of the Spatial Information Research Centre, University of Otago, Dunedin, New Zealand*. (Cité page 49.)
- [Moron, 1996] Véronique Moron (1996). *Mise en correspondance de données 3D avec un modèle CAO : Application à l'inspection automatique*. PhD thesis, INSA Lyon. (Cité page 17.)
- [Mozina et al., 2013] Miha Mozina, Dejan Tomazevic, Franjo Pernus, and Bostjan Likar (2013). Automated visual inspection of imprint quality of pharmaceutical tablets. *Machine Vision and Applications*, 24 :63–73. (Cité pages 6 et 37.)
- [Muja and Lowe, 2009] Marius Muja and David G. Lowe (2009). Fast approximate nearest neighbors with automatic algorithm configuration. *Conference on Computer Vision Theory and Application VISSAPP 09*, pages 331 – 340. (Cité page 26.)
- [Muja and Lowe, 2012] Marius Muja and David G. Lowe (2012). Fast Matching of Binary Features. *Ninth Conference on Computer and Robot Vision (CRV)*. (Cité page 26.)
- [Murty and Devi, 2011] M. Narasimha Murty and V. Susheela Devi (2011). *Pattern Recognition, An Algorithmic Approach*. Springer London Dordrecht Heidelberg New York. (Cité pages vii, 27, 28, 29, 30, 31 et 32.)
- [Musé, 2004] Pablo Musé (2004). On the definition and recognition of planar shapes in digital images. <https://tel.archives-ouvertes.fr/tel-00133648>, dernière consultation le 07/10/2016. (Cité page 34.)
- [Newman and Jain, 1994] Timothy Newman and Anil K. Jain (1994). A system for 3D CAD-based inspection using range images. (Cité page 42.)

-
- [Newman and Jain, 1995] Timothy Newman and Anil K. Jain (1995). A survey of automated visual inspection. *Computer Vision and Image Understanding*, 61. (Cité pages 6, 17, 36 et 40.)
- [Nguyen et al., 2009] Thanh Minh Nguyen, Siddahant Ahuja, and Q.M.Jonathan Wu (October, 2009). A real-time ellipse detection based on edge grouping. *IEEE International Conference on Systems, Man and Cybernetics*, pages 3280 – 3286. (Cité page 76.)
- [Nixon and Aguado, 2008] Mark S. Nixon and Alberto S. Aguado (2008). *Feature Extraction and Image Processing 2nd edition, 5.3 Template Matching*. Academic Press. (Cité page 20.)
- [Panda and Pradhan, 2013] B.S. Panda and D. Pradhan (2013). Minimum paired-dominating set in chordal bipartite graphs and perfect elimination bipartite graphs. *Journal of Combinatorial Optimization*, 26 :770 – 785. (Cité pages 59, 64 et 87.)
- [Patraucean et al., 2012] Viorica Patraucean, Pierre Gurdjos, and Rafael Grompone von Gioi (2012). A parameterless line segment and elliptical arc detector with enhanced ellipse fitting. *Proceedings of the 12th European Conference on Computer Vision, Part II* :572–585. (Cité page 76.)
- [Perng et al., 2011] Der-Baau Perng, Hsiao-Wei Liu, and Ching-Ching Chang (2011). Automated smd led inspection using machine vision. *International Journal of Advanced Manufacturing and Technology*, 57 :1065–1077. (Cité pages 18 et 37.)
- [Prasad et al., 2013] Dilip K. Prasad, Maylor K.H. Leung, and Chai Quek (2013). Ellifit : An unconstrained, non-iterative, least squares based geometric ellipse fitting method. *Pattern Recognition*, 46 :1449–1465. (Cité pages 75, 76 et 77.)
- [Prieto et al., 2002] F. Prieto, T. Redarce, R. Lepage, and P. Boulanger (2002). An automated inspection system. *International Journal of Advanced and Manufacturing Technologies*, page 917–925. (Cité pages 17 et 42.)
- [Rafael Grompone von Gioi et al., 2012] Rafael Grompone von Gioi, Jeremie Jakubowicz, Jean-Michel Morel, and Gregory Randall (2012). LSD : a Line Segment Detector. *Image Processing On Line* : <http://dx.doi.org/10.5201/ipol.2012.gjmr-lsd>. (Cité page 76.)
- [Reddy and Chatterji, 1996] B.S. Reddy and B.N. Chatterji (1996). An FFT-based technique for translation, rotation, and scale-invariant image registration. *IEEE Transactions on Image Processing*, 5 :1266 – 1271. (Cité page 74.)
- [Richert and Coelho, 2013] Willi Richert and Luis Pedro Coelho (2013). *Building Machine Learning Systems with Python*. Pack publishing. (Cité page 33.)
- [Roberto M. Cesar-Jr. et al., 2005] Roberto M. Cesar-Jr., Endika Bengoetxea, Isabelle Bloch, and Pedro Larrañaga (2005). Inexact graph matching for model-based recognition : Evaluation and comparison of optimization algorithms. *Elsevier, Pattern Recognition*, 38 :2099–2113. (Cité pages 66 et 86.)

-
- [Rosenfeld and Pfaltz, 1968] A. Rosenfeld and J.L. Pfaltz (1968). Distance functions on digital pictures. *Pattern Recognition*, 1 :33–61. (Cité page 49.)
- [Rosten and Drummond, 2006] E. Rosten and T. Drummond (2006). Machine learning for high speed corner detection. *In European Conference on Computer Vision*, 1. (Cité page 26.)
- [Rublee et al., 2011] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski (November 2011). ORB : An efficient alternative to SIFT or SURF. *In Computer Vision (ICCV), 2011 IEEE International Conference*. (Cité page 25.)
- [Shapiro and Stockman, 2001] Linda Shapiro and Georges Stockman (2001). *Computer Vision, chapter 11 : Matching in 2D, chapter 13 : Sensing and objet pose computation*. Prentice Hall. (Cité page 50.)
- [Shokoufandeh and Dickinson, 1999] Ali Shokoufandeh and Sven Dickinson (1999). Applications of bipartite matching to problems in object recognition. *ACM, Proceedings of ICCV on Graph Algorithm and Computer Vision*. (Cité pages 66 et 87.)
- [Smriti et al., 2005] Rahunathan Smriti, D. Stredney, P. Schmalbrock, and B.D. Clymer (2005). Image registration using rigid registration and maximization of mutual information. *Poster presented at : MMVR13. The 13th Annual Medicine Meets Virtual Reality Conference, Long Beach, CA*. (Cité page 74.)
- [Soto et al., 2015] Juan Soto, Guillermo Granda, Flavio Prieto, William Ipanaque, and Jorge Machacuay (2015). Cocoa bean quality assessment by using hyperspectral images and fuzzy logic techniques. *Twelfth International Conference on Quality Control by Artificial Vision, Proc. of SPIE*, 9534 :95340M–1 – 95340M–7. (Cité page 36.)
- [Stefano and Mattoccia, 2003] Luigi Di Stefano and Stefano Mattoccia (2003). Fast template matching using bounded partial correlation. *Machine Vision and Applications*, 13 :213–221. (Cité page 23.)
- [Stefano et al., 2005] Luigi Di Stefano, Stefano Mattoccia, and Frederico Tomabari (2005). ZNCC-based template matching using Bounded Partial Correlation. *Pattern Recognition Letters*, 26 :2129–2134. (Cité page 23.)
- [Stemmer, 2013] Stemmer (2013). The Imaging and vision handbook. *Stemmer Imaging*. (Cité pages 18 et 20.)
- [Teledyne, 2015] Teledyne (2015). Teledyne dalsa, catalogue 2015. http://info.teledynedalsa.com/acton/attachment/14932/f-0485/1/-/-/-/-/MachineVision_brochure_2015.pdf, consulté le 03/01/2015. (Cité page 19.)
- [Tola et al., 2010] Engin Tola, Pascal Fua, and Vincent Lepetit (2010). DAYSY : An efficient dense descriptor applied to wide-baseline stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32. (Cité page 26.)
-

- [Viana et al., 2015a] Ilísio Viana, Florian Bugarin, Nicolas Cornille, and Jean-José Orteu (2015.a). Application of bipartite graph to the matching of geometric primitives in vision based inspection. *International Summer School on VISion Understanding and Machine intelligence (VISUM), Porto-Portugal*. (Cité page 14.)
- [Viana et al., 2015b] Ilísio Viana, Florian Bugarin, Nicolas Cornille, and Jean-José Orteu (2015b). CAD-guided inspection of aeronautical mechanical parts using monocular vision. *International Conference on Quality Control by Artificial Vision, France, Proceedings of SPIE, 95340I*. (Cité page 14.)
- [Viana et al., 2014] Ilísio Viana, Florian Bugarin, Rémi Parlouar, and Jean-José Orteu (2014). Inspection d’assemblages mécaniques aéronautiques par vision artificielle. *Annual Conference of French Doctoral School, Systems (EDSYS)*. (Cité page 14.)
- [Viana et al., 2015c] Ilísio Viana, Jean-José Orteu, Nicolas Cornille, and Florian Bugarin (2015c). Inspection of aeronautical mechanical parts with a pan-tilt-zoom camera : an approach guided by the computer-aided design model. *Journal of Electronic Imaging*, 24(6)061118. (Cité page 14.)
- [Viana et al., 2013] Ilísio Viana, Rémi Parlouar, Florian Bugarin, Jean-José Orteu, and Ludovic Brèthes (2013). Inspection automatisée d’assemblages mécaniques : vers une approche couplée vision 2D/vision 3D. <http://hal.archives-ouvertes.fr/hal-00839887>. (Cité page 14.)
- [Viana et al., 2012] Ilísio Viana, Rémi Parlouar, Jean-José Orteu, and Ludovic Brèthes (2012). Fast Automated Inspection of Mechanical Assembly using a combined 2D/3D vision approach. *IUTAM Symposium on Advances of Optical Methods in Experimental Mechanics, Taipei (Taiwan), oral presentation only*. (Cité pages 13 et 42.)
- [Wang and Ma, 2014] Chen Wang and Kai-Kuang Ma (2014). Bipartite graph-based mismatch removal for wide-baseline image matching. *Journal of Visual Communication and Image Representation*, 25 :1416–1424. (Cité page 87.)
- [Wang et al., 2014] Gangyi Wang, Guanghui Ren, Zhilu Wu, Yaqin Zhao, and Lihui Jiang (2014). A fast and robust ellipse-detection method based on sorted merging. *Hindawi Publishing Corporation, The Scientific World Journal*. (Cité page 76.)
- [Webb, 2002] Andrew Webb (2002). *Statistical Pattern Recognition*. Springer London Heidelberg New York Dordrecht. (Cité pages 27, 28, 29, 32 et 33.)
- [Weiss et al., 2008] Y. Weiss, A. Torralba, and R. Fergus (2008). Spectral hashing. *In Advances in neural information*. (Cité page 26.)
- [Yang and Cheng, 2014] Xin Yang and Kwang-Ting Cheng (2014). Local difference binary for ultra-fast and distinctive feature description. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(1) :188–194. (Cité page 25.)
- [Zendjebil et al., 2008] I. M. Zendjebil, F. Ababsa, J-Y Didier, J. Vairon, L. Frauciel, M. Hachet, P. Guitton, and R. Delmont (2008). Outdoor augmented reality : State of the art

- and issues.
- 10th ACM/IEEE Virtual Reality International Conference (VRIC 2008), Laval, France.*
<https://hal.archives-ouvertes.fr/hal-00339480/document>, pages 177 – 187.
(Cité page 43.)
- [Zha et al., 2001] Hongyuan Zha, Xiaofeng He, Chris Ding, Horst Simon, and Ming Gu (2001). Bipartite graph partitioning and data clustering. *ACM, Proceedings of the tenth international conference on Information and knowledge management*, pages 25–32. (Cité page 87.)
- [Ziegler et al., 2012] Andrew Ziegler, Eric Christiansen, David Kriegman, and Serge Be-longie (2012). Locally uniform comparison image descriptor. *Advances in Neural Information Processing Systems 25 (NIPS)*. (Cité page 26.)
- [Zitová and Flusser, 2003] Barbara Zitová and Jan Flusser (2003). Image registration methods : a survey. *Image and Vision Computing*, 21 :977–1000. (Cité page 73.)

Author : Ilísio Viana do Espírito Santo

Title : "Automated inspection of mechanical parts by computer vision : An approach based on CAD model"

Abstract

The work presented in this manuscript deals with automated inspection of aeronautical mechanical parts using computer vision.

The goal is to decide whether a mechanical assembly has been assembled correctly i.e. if it is compliant with the specifications. This work was conducted within two industrial projects. On one hand the CAAMVis project, in which the inspection sensor consists of a dual stereoscopic head (stereovision) carried by a robot, on the other hand the Lynx© project, in which the inspection sensor is a single Pan/Tilt/Zoom camera (monocular vision). These two projects share the common objective of exploiting as much as possible the CAD model of the assembly (which provides the desired reference state) in the inspection task which is based on the analysis of the 2D images provided by the sensor.

The proposed method consists in comparing a 2D image acquired by the sensor (referred to as "real image") with a synthetic 2D image generated from the CAD model. The real and synthetic images are segmented and then decomposed into a set of 2D primitives. These primitives are then matched by exploiting concepts from the graph theory, namely the use of a bipartite graph to guarantee the respect of the uniqueness constraint required in such a matching process. The matching result allows to decide whether the assembly has been assembled correctly or not. The proposed approach was validated on both simulation data and real data acquired within the above-mentioned projects.

Keywords : Inspection, Graph matching, Primitive extraction, Bipartite graph, Distance transform, Image registration, CAD model.

Auteur : Ilísio Viana do Espírito Santo

Titre : "Inspection automatisée d'assemblages mécaniques aéronautiques par vision artificielle : une approche exploitant le modèle CAO"

Résumé

Les travaux présentés dans ce manuscrit s'inscrivent dans le contexte de l'inspection automatisée d'assemblages mécaniques aéronautiques par vision artificielle.

Il s'agit de décider si l'assemblage mécanique a été correctement réalisé (assemblage conforme). Les travaux ont été menés dans le cadre de deux projets industriels. Le projet CAAMVis d'une part, dans lequel le capteur d'inspection est constitué d'une double tête stéréoscopique portée par un robot, le projet Lynx© d'autre part, dans lequel le capteur d'inspection est une caméra Pan/Tilt/Zoom (vision monoculaire). Ces deux projets ont pour point commun la volonté d'exploiter au mieux le modèle CAO de l'assemblage (qui fournit l'état de référence souhaité) dans la tâche d'inspection qui est basée sur l'analyse de l'image ou des images 2D fournies par le capteur.

La méthode développée consiste à comparer une image 2D acquise par le capteur (désignée par « image réelle ») avec une image 2D synthétique, générée à partir du modèle CAO. Les images réelles et synthétiques sont segmentées puis décomposées en un ensemble de primitives 2D. Ces primitives sont ensuite appariées, 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. Le résultat de l'appariement permet de statuer sur la conformité ou la non-conformité de l'assemblage. L'approche proposée a été validée à la fois sur des données de simulation et sur des données réelles acquises dans le cadre des projets sus-cités.

Mots-clés : Inspection, Appariement de graphes, Extraction de primitives, Graphe biparti, Transformée en distance, Recalage d'images, Modèle CAO.