



HAL
open science

Modélisation de formes 3D par les graphes pour leur reconnaissance : application à la vision 3D en robotique dans des tâches de "Pick-and-Place"

Pierre Willaume

► To cite this version:

Pierre Willaume. Modélisation de formes 3D par les graphes pour leur reconnaissance : application à la vision 3D en robotique dans des tâches de "Pick-and-Place". Vision par ordinateur et reconnaissance de formes [cs.CV]. Université de Strasbourg, 2017. Français. NNT : 2017STRAD041 . tel-01735392

HAL Id: tel-01735392

<https://theses.hal.science/tel-01735392>

Submitted on 15 Mar 2018

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.

**ÉCOLE DOCTORALE MATHÉMATIQUES, SCIENCES DE L'INFORMATION ET DE
L'INGÉNIEUR**

Equipe "Complex System & Translational Bioinformatics"

– Laboratoire ICube – UMR 7357

THÈSE présentée par :

Pierre WILLAUME

soutenue le : 11 Décembre 2017

pour obtenir le grade de : **Docteur de l'université de Strasbourg**

Discipline/ Spécialité : Informatique

**Modélisation de formes 3D par les
graphes pour leur reconnaissance.
Application à la vision 3D en robotique
dans des tâches de "Pick-and-Place"**

THÈSE dirigée par :

Mme DERUYVER Aline

McF HDR, Université de Strasbourg, ICube, Strasbourg

et co-encadrée par :

M PARREND Pierre

Docteur, ECAM-Strasbourg-Europe, ICube, Strasbourg

RAPPORTEURS :

Mme TOUGNE Laure

Professeur, Université de Lyon, LIRIS, Lyon

M FONLUPT Cyril

Professeur, Université du Littoral, LISIC, Touquet

EXAMINATEURS :

M MIGUET Serge

Professeur, Université Lumière Lyon 2, LIRIS, Lyon

M DOIGNON Christophe

Professeur, Université de Strasbourg, ICube, Strasbourg

M VACAVANT Antoine

Docteur, Université d'Auvergne, ISIT, Clermont-Ferrand

INVITE :

M GANCEL Etienne

Industrial Engineering Director, Hager, Obernai

Remerciements

Avant de présenter ma thèse, je tiens à remercier :

Monsieur Etienne GANCEL pour m'avoir choisi et pour son accompagnement bienveillant au sein de l'entreprise Hager. Il a su m'aider professionnellement et personnellement grâce à sa richesse scientifique et technique.

Ma directrice de thèse, Madame Aline DERUYVER pour son intérêt et son soutien. Mon encadrant, Monsieur Pierre PARREND pour son assistance précieuse et pour son investissement dans ma thèse.

Messieurs Pierre COLLET, Olivier POCH et Bruno KAUFFMANN pour m'avoir accueilli dans leurs services.

Mesdames Gaëlle MOUTAUX, Sophie DAUBIGNE, Laetitia POIDEVIN, Julie THOMPSON, Anne NEY, Anne JEANNIN-GIRARDON, Odile LECOMPTE, Catherine GUTH et Audrey DEFOSSET pour leur collaboration et leur sympathie. Elles ont su m'apporter une aide avec professionnalisme et bonne humeur.

Messieurs Joël WEIL, Maxime MARCK, Raymond RIPP, Arnaud KRESS et Ludovic GALMICHE pour leurs aides tout au long de cette thèse.

Messieurs Julio NAVARRO, Yannis NEVERS, Kirsley CHENNEN, Isaac ABOTSI et Luc MOULINIER pour toutes nos aventures au labo.

Mes remerciements s'adressent bien évidemment à toute l'équipe du CSTB, d'Hager et de l'ECAM Strasbourg Europe et des intervenants professionnels pour avoir assuré la partie théorique.

Une ambiance de travail exceptionnelle a bien sûr contribué au bon déroulement de cette aventure.

Mes remerciements s'adressent ensuite au jury.

Je remercie ma famille et mes amis pour m'avoir supporté pendant ces trois années.

Table des matières

1	Introduction	1
1.1	Contexte	2
1.2	Sujet de thèse	3
1.3	Challenges	4
1.3.1	Enjeux industriels	4
1.3.2	Enjeux scientifiques	6
1.4	Contributions techniques	7
1.5	Contributions scientifiques	11
1.6	Collaborations	12
I	Partie 1 : Etat de l'art	13
2	"Pick-and-Place"	15
2.1	Introduction	16
2.2	Les définitions	16
2.3	Les méthodes d'aménagement	16
2.3.1	Le bol vibrant	17
2.3.2	Le "Pick-and-Place" : Avec un robot et un convoyeur	18
2.3.3	Le "Bin-picking" : Avec un robot, plusieurs prises images	19
2.3.4	Comparaison	22
2.4	Le "Pick-and-Place"	22
2.4.1	L'historique du "Pick-and-Place"	22
2.4.2	Le "Pick-and-Place" : Un domaine pluridisciplinaire	25
2.4.3	Le "Pick-and-Place" : les techniques actuelles	26
2.5	Les Problématiques d'identifications	27
2.5.1	La caméra laser	27

2.5.2	La stéréoscopie	28
2.6	Conclusion	29
3	Traitement d'images pour l'extraction de pièces	31
3.1	Introduction	32
3.2	Les types de pièces à détecter	33
3.2.1	La morphologie	34
3.2.2	Les difficultés de reconnaissance des objets dans une scène	35
3.3	Représentation du modèle d'entrée	35
3.4	Identifications en temps borné	36
3.5	Extraction de points d'intérêt	37
3.5.1	SIFT	37
3.5.2	PCA-SIFT	39
3.5.3	SURF	40
3.5.4	Comparaison	40
3.5.5	Conclusion	42
3.6	Approximation polygonale	43
3.6.1	Fonctionnement	44
3.6.2	Les algorithmes	44
3.6.3	Avantages	46
3.7	Discussion et conclusion	47
4	Algorithmes évolutionnaires	49
4.1	Introduction	50
4.2	Les familles d'algorithmes évolutionnaires	52
4.2.1	Stratégies d'évolution (SE)	52
4.2.2	Programmation évolutionnaire (PE)	53
4.2.3	Algorithmes génétiques (AG)	53
4.2.4	Programmation génétique (PG)	54

4.3	<i>Harmony Search</i>	54
4.3.1	L'algorithme	56
4.3.2	Variantes de la méthode	58
4.3.3	Pourquoi <i>Harmony Search</i> ?	60
4.3.4	Un algorithme remis en cause	61
4.4	Conclusion	61
II	Partie 2 : Contributions	63
5	Al-Thocb HSC : Un Algorithme d'optimisation pour l'étalonnage de caméras	65
5.1	Introduction	66
5.2	Challenges	68
5.2.1	Précision	69
5.2.2	Facilité de configuration	69
5.2.3	Performance	70
5.3	Le calibrage de caméras stéréoscopiques	70
5.3.1	Mécanisme d'étalonnage de caméra 3D	70
5.3.2	Les mires	71
5.4	Sélection d'images pour l'étalonnage	73
5.4.1	Nombre d'images : temps de traitement vs score d'étalonnage	73
5.4.2	Sélection des meilleures mires	74
5.5	Méthodologie	76
5.6	Algorithmes d'étalonnage	77
5.7	Al-Thocb : Harmony Search Calibration	77
5.7.1	Etymologie d'Al-Thocb	78
5.7.2	Sauvegarde des points de chaque mire	79
5.7.3	Algorithme Al-Thocb	79

5.8	Partie expérimentale	80
5.8.1	Mise en place	82
5.8.2	Précision	83
5.8.3	Performance	85
5.8.4	Durée appropriée de l'algorithme de traitement pour une ligne de production	85
5.8.5	Paramétrage	86
5.9	Parallélisation	87
5.10	Discussion	89
5.11	Conclusions	90
6	ThOR : Détection d'objets fins dans une scène par squelettisation et mise en correspondance de graphes en 3D	93
6.1	Challenges	94
6.2	Méthode proposée	95
6.2.1	Étapes du processus de détection de l'algorithme Thor	96
6.2.2	Prétraitement	97
6.2.3	Squelettisation	99
6.2.4	Mise en correspondance d'images par approche à base de graphes pour détecter les objets	100
6.2.5	Recherche d'objets	103
6.3	Expérimentation	104
6.4	Discussion	105
6.5	Conclusions	109
7	HAThOR : Méthodologie d'optimisation de la détection	111
7.1	Challenge	112
7.2	Critères de validation	113
7.3	<i>Harmony Search</i> pour la recherche d'objets	113

7.4	Expérimentation	116
7.4.1	Performance	116
7.4.2	Précision	120
7.4.3	Comparaison d'HAThOR avec des algorithmes d'extraction de points clés	126
7.5	Paramétrage	128
7.6	Discussion	128
7.7	Conclusions	130
8	Conclusion et perspectives	133
8.1	Conclusion technique	134
8.1.1	Les réalisations techniques	134
8.1.2	Les perspectives	136
8.2	Conclusion scientifique	136
8.2.1	Le calibrage	136
8.2.2	Détection d'objet	137
8.2.3	L'avantage des algorithmes évolutionnaires	137
8.2.4	Les perspectives	138
8.3	Les publications	139
	Bibliographie	141

Table des figures

1.1	"Bin-picking" : la solution envisagée	8
2.1	Le bol vibrant véhicule les objets dans une goulotte par vibrations	17
2.2	Les objets sont pré-placés sur un convoyeur et sont ensuite sélectionnés par le ou les robots (source : "Logistik Heute, Arbeit 4.0 : Wo bleibt der Mensch ?")	18
2.3	Bande dessinée pour la représentation d'un système de "bin-picking" avec un robot, des caméras, et une bassine d'objets	21
2.4	Systèmes de "Pick and Place"	28
3.1	Exemples d'images de pièces photographiées	34
3.2	Représentation d'un fichier CAO	36
3.3	Extraction de points clés à l'aide de l'algorithme SURF (120 ms)	37
3.4	Approximation polygonale sur un objet représenté sur un fichier CAO : Bleu = Contour, Rouge = Point dominant	45
4.1	Schéma fonctionnel d'un algorithme évolutionnaire	51
4.2	Schéma fonctionnel de l'algorithme <i>Harmony Search</i>	55
5.1	Schéma fonctionnel du procédé de "Pick-And-Place"	66
5.2	Al-Thocb HSC : Mire avec des motifs carrés (damier)	71
5.3	Al-Thocb HSC : Mire avec des motifs ronds	72
5.4	Al-Thocb HSC : Temps de traitement des étalonnages aléatoires par rapport au nombre d'images (100 itérations) ; Ligne bleue : temps de traitement moyen	73
5.5	Al-Thocb HSC : Score des étalonnages aléatoires par rapport au nombre d'images	74

5.6	Al-Thocb HSC : Exemple de sélection de mire pour 3 calibrages efficaces. 1 ^{ère} ligne, 2 ^{ème} ligne et 3 ^{ème} ligne. V : image efficiente, X : image non efficiente	75
5.7	Al-Thocb HSC : processus d'étalonnage pour l'utilisateur . . .	75
5.8	Al-Thocb HSC : Méthodologie générale	76
5.9	Al-Thocb HSC : Diagramme	78
5.10	Al-Thocb HSC : Fonction de coût pour 30 calibrages et 21 images. Une valeur faible implique un meilleur étalonnage . . .	83
5.11	Al-Thocb HSC : Taux d'amélioration pour 30 étalonnages et 21 images	84
5.12	Al-Thocb HSC : performance -> HMS=5, 200000 calibrages .	85
5.13	Al-Thocb HSC : Fonction de coût en fonction des paramètres <i>HMCR</i> , <i>PAR</i> , <i>HMS</i> et le nombre d'itérations. Bleu : bonne fonction de coût. Rouge : Mauvaise fonction de coût.	86
5.14	Parallélisation d'Al-Thocb HSC : Temps d'extraction des points appartenant aux mires en fonction du nombre de mires pour différentes tailles d'images	88
5.15	Parallélisation d'Al-Thocb HSC : Temps de traitement en fonction du nombre d'itérations	89
6.1	Schéma fonctionnel du procédé de "Pick-And-Place"	94
6.2	ThOR : Schéma fonctionnel	95
6.3	ThOR : Simplification de l'image d'entrée afin de récupérer le squelette des objets recherchés (Binarisation : $Color_{min}$: 100 et $Color_{max}$: 255)	98

6.4	ThOR : Mise en correspondance de graphes pour des images stéréoscopiques ($A : 11, B : 10, C : 2\dots$). Deux nœuds connus sont sélectionnés comme référents (exemple : ($F : 6, A : 11$)). L'angle et les distances entre les nœuds ABF et AKF sont déterminés pour savoir si B a comme correspondant 10 ou 1.	102
6.5	ThOR : Recherche d'objets fins. Segment : Lien entre deux nœuds faisant ressortir les objets. Texte : Position et angle de l'objet en fonction d'un repère.	106
6.6	ThOR : Recherche d'objets spécifiques	107
6.7	ThOR : Représentation des temps de calcul de la mise en correspondance des nœuds sur une image de 640x480px. Moyenne du nombre de nœuds sur les images droites et gauches en fonction du temps en milliseconde : 600 = Pas d'enchevêtrement ; 1000 = Enchevêtrement ; 1800 = Différents objets positionnés aléatoirement	107
6.8	Recherche des objets fins par l'algorithme SIFT (<i>Scale Invariant Feature Transform</i>). Temps d'exécution : 271ms	108
7.1	HAThOR : schéma fonctionnel	114
7.2	HAThOR : Temps de prétraitement nécessaire pour simplifier l'image en fonction de leurs tailles (1600x1200-84x63)	118
7.3	HAThOR : Temps de prétraitement nécessaire pour simplifier l'image en fonction de leurs tailles (petites tailles) (177x33 to 84x63)	118
7.4	HAThOR : Performance de la mise en correspondance de deux nœuds en fonction du nombre de nœuds	119
7.5	HAThOR : Temps total de traitement total en fonction de la taille de l'image et du nombre de nœuds ; bleu : bon temps de recherche, rouge : mauvais temps de recherche	120

7.6	HAThOR et son application à la détection d'objets : Colonne 1 - image d'entrée; Colonne 2 - application de la segmentation par ligne de partage des eaux; Colonne 3 - application de la squelettisation; Colonne 4 - Application de la recherche d'objet. Segment bleu (colonne 4) : segment entre 2 nœuds correspondant à l'objet désiré	122
7.7	HAThOR : Identification d'objets placés dans une bassine . . .	123
7.8	HAThOR : Evaluation sur des objets spécifiques	124
7.9	HAThOR : Détection de clés avec des luminosités différentes .	125
7.10	HAThOR : Performance à détecter des objets lorsque l'image est floutée, Gaussien Noyau (6 objets * 23 (X :poids) * 9 (Y :Noyau) = 1242 images)	126
7.11	HAThOR : Détection d'objets pour différents niveaux de bruits ajoutés, représentée par un pourcentage de pixels blancs et noirs ajoutés au hasard dans l'image	127
7.12	HAThOR : Performance en fonction des paramètres Pitch Adjusting Rate (<i>PAR</i>) et <i>Harmony Memory Considering Rate (HMCR)</i> pour un pas de 0.1 : recherche de deux nœuds spécifiques parmi 40	129
7.13	HAThOR : Performance en fonction du nombre de nœuds et du paramètre <i>Harmony Memory Size (HMS)</i> pour un pas de 0.1	129

Liste des tableaux

2.1	Récapitulation des méthodes d'aménagement avec leurs avantages	22
3.1	Récapitulatif des avantages et inconvénients des méthodes d'ex- tractions de points clés [Juan 2009]	40
4.1	Exemple d'initialisation aléatoire de la population de l'algo- rithme <i>Harmony Search</i>	57
4.2	Création d'une nouvelle solution de l'algorithme <i>Harmony Search</i>	58
4.3	Création d'une nouvelle génération de l'algorithme <i>Harmony Search</i>	58
5.1	Ensemble de mires auquel appartient un ensemble de points : G = gauche, D = droite	79
5.2	Ensemble des coordonnées des pixels pour une mire photogra- phiée par deux caméras	80
5.3	Scores et temps de calibrages effectués par un opérateur	84
6.1	Performance de l'algorithme ThOR. O = Oui, N = Non	108

Introduction

Sommaire

1.1	Contexte	2
1.2	Sujet de thèse	3
1.3	Challenges	4
1.4	Contributions techniques	7
1.5	Contributions scientifiques	11
1.6	Collaborations	12

“ Un robot n'est pas tout à fait une machine. Un robot est une machine fabriquée pour imiter de son mieux l'être humain. ”

Isaac Asimov

1.1 Contexte

La mondialisation est de plus en plus présente dans notre vie quotidienne, nous sommes dans une société où le client veut rapidement de nouveaux produits personnalisés (domotique, téléphone portable, ordinateur portable ...). Pour produire plus rapidement et rester compétitives, les industries ont l'obligation de rendre flexibles leurs lignes de production. Tous ces éléments conduisent à l'utilisation de machines et plus particulièrement de robots contenant de plus en plus d'intelligence (reconnaissance d'objets, coordination de mouvement, sécurité), notamment pour la saisie d'objets. Malheureusement, l'intelligence artificielle des robots ne permet pas actuellement de détecter différents objets (taille, forme, matière) disposés au hasard dans un conteneur. Dans ce contexte, le problème à résoudre est de modéliser ces objets dans l'espace à partir de plusieurs prises de vue. L'ensemble de ces images constitue des données brutes très volumineuses, actuellement difficile à traiter en un temps raisonnable par rapport aux contraintes industrielles (inférieur à 2 secondes). La complexité des images nécessite la mise en œuvre d'algorithmes puissants. Le but de ce travail est de concevoir un système automatique constitué d'une ou de plusieurs caméras capables de détecter en 3 dimensions plusieurs types d'objets disposés aléatoirement de façons très proches les uns des autres, voire enchevêtrées, dans un conteneur. Il n'existe pour le moment aucune solution qui permette de gérer automatiquement un tel chargeur d'objets différents (forme, taille) disposés en vrac et en 3 dimensions. Certaines solutions permettent de gérer en 2 ou 3 dimensions un dispositif d'alimentation d'objets aux formes simples, d'une taille de quelques centimètres et positionnées de manière à pouvoir les séparer très facilement (plateau vibrant, conditions d'éclairages optimales, objets facilement reconnaissables. Dans le domaine de l'extraction d'objets dans les images, de nombreux travaux s'intéressent actuellement à la définition de méthodes orientées "objet" pour récupérer des éléments parti-

culiers. Dans ce cadre, chaque "objet" représente un ensemble de pixels homogènes construit sur les caractéristiques élémentaires des objets (spectrale, géométrique) et les relations spatiales entre eux (texture et relations contextuelles) [Henricsson 1998, Hofmann 2001, Sellaouti 2014]. Tous ces travaux s'accordent à dire que cette extraction ne peut être faite de manière totalement non supervisée, c'est-à-dire sans utilisation de connaissances du domaine sur les objets à extraire. Elles peuvent prendre plusieurs formes : ontologies du domaine, contraintes, relations inter-objets, etc.

1.2 Sujet de thèse

Pour cette thèse, le matériel mis à disposition lors des manipulations est un ensemble robot, PC et caméras fixes. Les caméras transmettent l'image au PC qui détecte et localise les objets pour transmettre au robot leurs positions. Le travail est décomposé en trois parties :

1. trouver une modélisation adaptée des objets recherchés en fonction de leurs topologies
2. comparer la pertinence et la performance des algorithmes pour la détection d'objets en fonction de leurs :
 - tailles
 - formes
 - matériaux (plastique transparent, métal...)
3. proposer et développer une approche algorithmique originale basée sur des algorithmes évolutionnaires afin d'optimiser les temps de recherche.

Ce travail de thèse doit permettre la mise en place d'un système d'aménagement pour différents types d'objets. Nous avons développé dans un premier cas un système muni d'un robot qui saisit des objets photographiés par des caméras. Les images sont détectées et localisées par un PC. Un ensemble d'outils inspiré

de l'état de l'art actuel a été développé pour la recherche d'objets avec des formes et des caractéristiques morphologiques qui ressortent facilement de l'image. Nous avons ensuite développé un algorithme de recherche d'objets fins ou l'objet ne ressort pas autant sur l'image. Nous avons ensuite optimisé la détection pour effectuer une recherche dans un temps borné.

1.3 Challenges

La mise en place d'un système autonome de reconnaissance et de localisation d'objets disposés aléatoirement dans une scène¹ nécessite :

- une simplification et une automatisation des systèmes de calibrage des caméras utilisées pour le calcul de la position de l'objet à sélectionner
- une représentation simple des données du modèle de l'objet à comparer avec les objets présents dans la scène à traiter
- d'optimiser les temps de recherche de l'objet dans la scène afin d'avoir une recherche inférieure à deux secondes.

1.3.1 Enjeux industriels

Les entreprises cherchent des solutions pour automatiser des postes d'assemblages de petits composants, présentés en vrac dans des bassines, que l'on ne sait traiter en automatique actuellement, qu'après présentation de ces composants sur un plateau lisse ou alvéolé, un convoyeur ou en sortie de sélection par bol vibrant. Deux enjeux industriels ressortent de cette thèse, le besoin d'automatiser les systèmes de calibrage et la façon dont les données sont représentés.

1. Scène : Espace photographié par les caméras

1.3.1.1 Besoin de calibrage

L'une des problématiques de l'utilisation de machines de vision dans un système industriel est la nécessité d'avoir des connaissances dans le domaine, notamment pour la phase de calibrage. Cette étape primordiale dans le système permet de mettre en relation le ou les capteurs en fonction de la scène et permet le calcul des coordonnées X, Y et Z de l'objet en sus de son orientation. Le calibrage est un procédé qui nécessite un ensemble de connaissances en optique et en traitement d'images. Cependant un opérateur n'a pas le temps d'avoir ces connaissances. Il est donc nécessaire d'avoir un système de calibrage autonome supprimant les prérequis. Ce système doit répondre à l'amélioration des critères suivants :

- la durée : Le temps de calibrage doit être court afin d'optimiser la production de la machine
- la qualité : Le calibrage récupéré doit répondre à une norme de qualité afin d'être en cohérence avec la prise d'objet
- les coûts : définis par le temps passé par la maintenance à résoudre les problèmes liés aux calibrages.

1.3.1.2 Comment représenter les données

Le calibrage effectué, il est nécessaire d'identifier les objets à sélectionner. Pour ceci, une reconnaissance d'objet doit être mise en place. La représentation des objets doit être transformée en un ensemble de données compréhensibles par le programme et comparables avec un modèle quel qu'il soit (image, CAO², caractéristiques). Plusieurs critères de qualité sont nécessaires quant au bon fonctionnement d'un système de reconnaissance d'objets par traitement d'images. Les données en entrée et en sortie du système doivent être compréhensibles par l'opérateur. Le programme de reconnaissance doit être

2. conception assistée par ordinateur

compréhensible par le personnel de maintenance. Le temps d'identification d'un objet dans une scène doit être restreint.

1.3.2 Enjeux scientifiques

L'objectif primaire de cette thèse est de proposer une méthodologie d'extraction et de localisation d'objets sur des images qui représentent des objets enchevêtrés, disposés aléatoirement et ceci dans un temps borné. Nous définissons l'extraction comme le procédé de reconnaissance d'un objet positionné dans une scène. La localisation est définie comme la définition de l'emplacement de l'objet en fonction d'un point de référence. Il est nécessaire de trouver un algorithme et un procédé de reconnaissance capable de localiser l'objet.

1.3.2.1 La reconnaissance d'objet en 3D

Beaucoup de méthodes existent pour la reconnaissance d'objets en 3D. Il est nécessaire de mettre en avant les plus cohérentes et de déterminer les facteurs de satisfaction de l'algorithme. Ceux-ci peuvent être liés à la flexibilité de l'algorithme à détecter des objets :

- fins/épais
- avec/sans enchevêtrement
- avec/sans transparence
- avec/sans bruits dans l'image

1.3.2.2 Comment représenter les objets fins

De nombreux algorithmes ont été proposés dans la littérature pour retrouver des objets disposés dans une scène en traitement d'image. Cependant, peu d'entre eux traitent de la reconnaissance d'objet quand celui-ci est fin et sans spécificités de contours où de formes. La problématique est de trouver un moyen de détecter ce type d'objet dans une scène quelconque. Les critères

de satisfaction de l'algorithme de détection dépendent de :

- sa qualité à trouver l'objet recherché dans une scène
- son temps de traitement nécessaire pour la détection d'objet
- son temps de mise en place dans une ligne de production
- son temps d'adaptation lors d'un changement de référence.

Les méthodes structurales basées sur la représentation des images par les graphes montrent depuis quelques années leur intérêt pour l'interprétation d'images [Keselman 2005, Deruyver 1997, Conte 2004, Bauckhage 2004, Messmer 1998]. Le formalisme des graphes est effectivement un moyen efficace pour modéliser des objets complexes constitués de plusieurs parties organisées spatialement. Cette approche est utilisée avec succès dans l'équipe CSTB pour interpréter des images cérébrales anatomiques [Deruyver 2009].

1.3.2.3 Recherche dans un temps borné

La reconnaissance d'objets dans une scène peut s'avérer longue. Il est nécessaire d'avoir un algorithme optimisé et de lier deux univers différents. Le premier étant celui du traitement d'images et le deuxième celui des algorithmes évolutionnaires. L'objectif de ces algorithmes est de proposer une solution proche à un problème donné. En reprenant notre cas, l'entreprise s'est fixé un temps de reconnaissance et de localisation à moins de 2 secondes.

1.4 Contributions techniques

Cette partie introduit les contributions techniques nécessaires pour le bon fonctionnement d'un système d'aménagement. Nous commençons par définir le besoin de l'entreprise d'automatiser les systèmes de production. Nous continuons par définir le système d'aménagement utilisé au cours de cette thèse. Nous spécifions ensuite le besoin de calibrage nécessaire pour localiser les objets. Nous déterminons ensuite la solution mise en place pour la représentation des don-

nées. Nous terminons par détailler le besoin d'avoir une recherche dans un temps borné.

Automatisation des systèmes La solution de l'entreprise est de se concentrer sur un système ayant pour but de localiser des objets dans une scène dans le but d'automatiser au maximum les postes de production. Pour ce faire, il est nécessaire d'avoir un système autonome, adaptable et rapide.

La prise d'objets dans une bassine Comme représenté dans la Figure 1.1, le système de prise d'objets dans une bassine (Eng : Bin-picking) est effectué à l'aide d'un ensemble robot-caméras. Le système "caméras" sera préalablement calibré afin de pouvoir localiser les objets dans une scène ayant des coordonnées en adéquation avec le robot. Nous utilisons ensuite un robot 6 axes pour pouvoir prendre les objets. Le PC récupère les images des caméras, effectue la recherche et la localisation de l'objet, et transmet les coordonnées de celui-ci au robot.

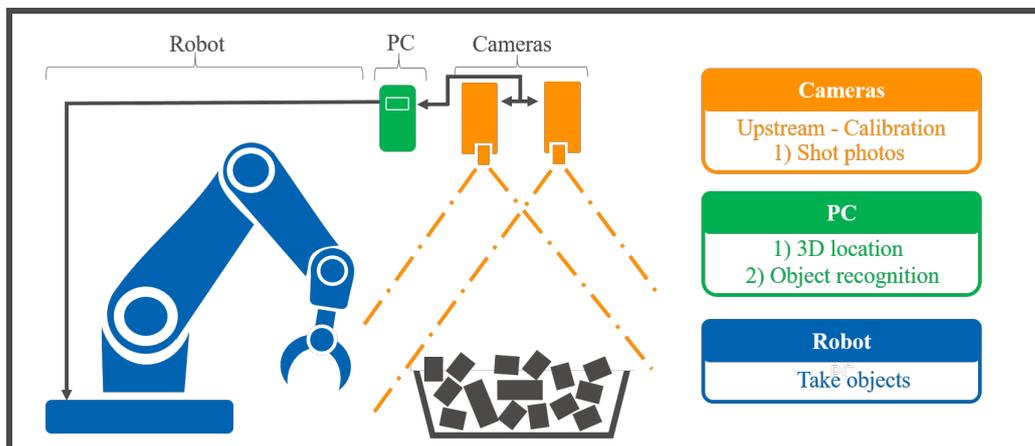


FIGURE 1.1 – "Bin-picking" : la solution envisagée

Caméra vs capteur de vision : Il existe beaucoup de types de capteurs notamment dans le domaine de l'imagerie. Nous distinguons un capteur de vision d'une caméra. La différence se situe dans l'intelligence embarquée issue

des deux éléments. Un capteur de vision ne possède pas ou très peu de mémoire embarquée et nécessite un traitement notamment à l'aide d'un PC. À l'inverse, une caméra possède un ensemble de fonctions permettant de localiser certains composants. Sur certaines caméras, il est possible d'identifier la position X, Y de l'objet ainsi que l'angle d'orientation. Cependant, le traitement d'images associé à ces caméras n'est pas optimisé pour des recherches d'objets dans une bassine et ne permet pas de récupérer la profondeur de l'objet. Ces données peuvent être transmises au robot. Nous utilisons dans notre cas un capteur de vision. Un autre désavantage des caméras est l'obligation d'utiliser le logiciel fourni par le fabricant de cette caméra. À tout changement de marque, il est nécessaire de réapprendre le langage de la caméra. Dans le cas d'un capteur de vision, le programme est sur le PC, il n'y a donc aucune dépendance du fournisseur.

Le robot : Il est la partie physique du système. Il sert à aller chercher l'objet en fonction des coordonnées transmises par le PC. Celui-ci doit être armé d'un préhenseur afin de récupérer l'objet. Il existe deux types de robots présents dans les entreprises. Il y a les robots Scara (Eng : Selective Compliance Articulated Robot Arm) et les robots 6 axes. La différence entre les deux se situe sur les degrés de liberté que peut avoir le robot. Un robot 6 axes possède toutes les possibilités contrairement à un robot Scara qui est limité à 3 degrés de liberté en rotation et une seule en translation. Le choix d'un robot dépend principalement de ses degrés de liberté, de la place qu'il prend et de son prix. Un robot 6 axes peut orienter sa pince en fonction de l'angle et de l'inclinaison de l'objet. Les objets étant positionnés aléatoirement dans une bassine, nous avons choisi un robot 6 axes.

Le PC : Il est le cerveau du système, il permet la coordination entre la caméra et le robot. Il va en un premier lieu positionner le robot à une coordonnée spécifique dite initial. Ensuite le PC va recevoir un ordre de prise d'un ensemble d'images afin de pouvoir identifier un objet et de calculer sa

localisation X , Y et Z ainsi que son orientation. Ces coordonnées sont ensuite comparées avec la position réelle du robot afin de calculer la trajectoire à donner au robot. Un ensemble de données nécessaire pour la saisie de l'objet est transmis au robot tel que :

- la position X , Y et Z de l'objet en fonction de la position du robot. Cette position permet d'avoir le point de saisie de l'objet.
- l'angle et l'inclinaison de l'objet. Ces données permettent d'orienter correctement la pince du robot.

Besoin de calibrage Le calibrage d'une ou plusieurs caméras permet de récupérer une information sur un objet situé dans l'image. Dans notre exemple trois informations sont nécessaires pour chercher un objet :

- sa position X , Y et Z
- son orientation
- son inclinaison.

Cette typologie d'informations est calculée au travers d'une ou plusieurs caméras calibrées. Le système de calibrage de caméras nécessite des connaissances issues du domaine du traitement d'images, et peut nécessiter un apprentissage pour l'opérateur. À cette fin, nous proposons d'utiliser un algorithme d'optimisation pour automatiser l'étalonnage des caméras. L'opérateur devra fournir un ensemble d'informations et d'images définies pour le bon fonctionnement de l'algorithme. Celui-ci sera optimisé pour sélectionner les meilleures images candidates pour récupérer les informations les plus pertinentes de la position, l'orientation et l'inclinaison.

Comment représenter les objets fins Dans le but d'identifier et localiser des objets fins dans une scène prise sous deux angles différents, nous proposons d'utiliser un ensemble de filtres afin de mettre en exergue les objets à récupérer. Après la simplification de l'image, nous suggérons d'utiliser un algorithme

de squelettisation pour récupérer la ou les courbes des objets à saisir. L'étape suivante consiste à sauvegarder les points extrema et de jonction de chaque entité. L'objectif final est d'effectuer une mise en correspondance de graphes dans le but de mettre en relation le squelette de l'image de droite et celle de l'image de gauche.

Recherche dans un temps borné Afin d'optimiser le temps de recherche, un ensemble d'algorithmes d'optimisation a été étudié. Nous proposons dans cette thèse d'utiliser les algorithmes évolutionnaires pour améliorer les temps de calibrage et de recherche d'objet (fin ou non). Afin d'avoir un programme utilisable en entreprise par un grand nombre de personnes, nous avons employé un algorithme simple et qui tend rapidement vers une valeur optimale. Proposé par Geem, Harmony Search est un algorithme d'optimisation s'inspirant du processus d'improvisation des musiciens de jazz. Il a l'avantage de contrôler simplement le ratio entre l'exploration et l'exploitation à l'aide de ses différents paramètres.

1.5 Contributions scientifiques

Trois contributions scientifiques ressortent de cette thèse. La première d'entre elles est l'automatisation et l'optimisation des systèmes de calibrage de caméras stéréoscopiques. Les procédés actuels proposés dans la littérature s'effectuent principalement de façon manuelle [Zhang 2000]. Nous proposons d'automatiser ces systèmes avec une approche par optimisation en adaptant un algorithme évolutionnaire dans le contexte de l'étalonnage de caméras. Notre deuxième contribution scientifique est l'identification des objets fins. L'ensemble des contributions scientifiques exposé dans la littérature effectue une recherche d'objets dans une scène où ceux-ci ont des formes [Ulrich 2012] et/ou des points d'intérêt ressortant facilement sur l'image [Piccinini 2012].

Ces algorithmes utilisent l'extraction et la description de points clés entre un modèle image et l'image de la scène à traiter. Nous proposons une approche différente basée sur la mise en correspondance de graphes entre deux images stéréoscopiques. Le modèle d'apprentissage différent d'un modèle image s'effectue à partir de données de l'objet telles que sa taille. Pour beaucoup de domaines (industriels, applications mobiles), les temps de traitement d'une image doivent être rapides (proche du temps réel). Pour ceci, notre troisième contribution est d'optimiser les temps de recherches en adaptant le même algorithme évolutionnaire utilisé pour l'étalonnage de caméras. Dans de bonnes conditions, nos temps de recherches nous permettent d'avoir une recherche dans un temps inférieur à la seconde.

1.6 Collaborations

Ces travaux s'effectuent sous l'encadrement d'Etienne Gancel dans le cadre de l'entreprise Hager, entreprise commanditaire, ainsi qu'en collaboration avec le laboratoire Icube et l'équipe CSTB sous l'encadrement d'Aline Deruyver (directrice de thèse), et L'ECAM Strasbourg-Europe et plus spécifiquement Pierre Parrend (Enseignant-Chercheur ECAM Strasbourg-Europe, chercheur membre de l'équipe CSTB du laboratoire Icube).

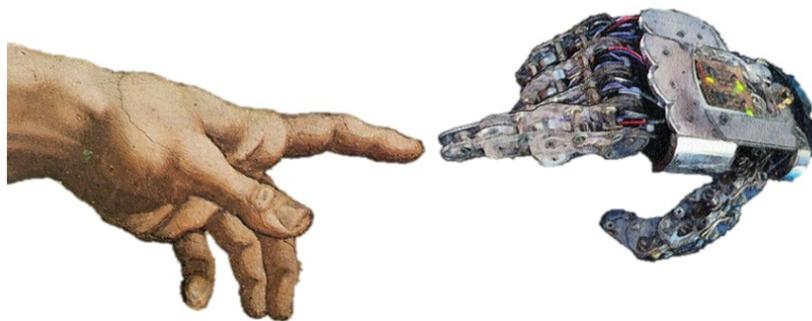
Première partie

Partie 1 : Etat de l'art

"Pick-and-Place"

Sommaire

2.1	Introduction	16
2.2	Les définitions	16
2.3	Les méthodes d'aménagement	16
2.4	Le "Pick-and-Place"	22
2.5	Les Problématiques d'identifications	27
2.6	Conclusion	29



2.1 Introduction

Le "Pick-and-Place" est un procédé d'automatisation qui permet d'amener un objet à un emplacement désiré. Une des définitions utilisées dans [Buchholz 2016] par PLM Engineering¹ est "l'automatisation est le transfert d'un travail humain vers des automates, réalisé avec l'aide de machines.". Ce procédé en amélioration continue permet de répondre aux besoins des entreprises en systèmes automatisés adaptables à l'environnement de travail [Berger 2014].

2.2 Les définitions

Amenage : Dans une machine-outil, l'amenage est l'opération qui met à portée de l'outil la pièce à travailler².

"Pick-and-Place" : C'est l'action de prendre un ou plusieurs objets. Cette action comporte plusieurs phases, la détection, la localisation, le placement de l'objet à une coordonnée X, Y et Z.

Préhenseur : C'est un objet capable de réaliser l'action de saisir, d'agripper ou de retenir un support, un objet, une proie, etc.³. Dans le domaine du "Pick-and-Place", un préhenseur désigne souvent une pince qui vient saisir un objet.

2.3 Les méthodes d'amenage

Différentes méthodes d'amenages sont utilisées en entreprise. Nous définissons les trois principales. Premièrement, nous présentons les bols vibrants. Deuxièmement, nous présentons les méthodes de "Pick-and-Place" avec un robot et une prise image. Troisièmement, nous présentons le système étudié

1. PLM Engineering. Betriebsautomation. <http://plm-engineering.com/leistungen/-betriebsautomation/index.php>, November 2013

2. <http://www.larousse.fr/dictionnaires/francais/amenage/2775>

3. <http://www.larousse.fr/dictionnaires/francais/prehenseur/63501>

avec un robot et plusieurs prises d'images. Plusieurs critères définissent quel type de méthodes doit être utilisé. Les plus importants sont :

- le coût de la machine
- le temps de la mise en place
- l'adaptabilité de la machine à détecter différents types d'objets
- le rendement
- la répétabilité de l'opération.

2.3.1 Le bol vibrant

Comme représenté sur la Figure 2.1, un bol vibrant ([Eng : Bowl feeder](#)) permet de transporter des objets à une position définie. Il existe deux types de bols vibrants :

- le bol vibrant circulaire
- le transporteur vibrant linéaire.



FIGURE 2.1 – Le bol vibrant véhicule les objets dans une goulotte par vibrations

Le bol vibrant est un modèle très courant en entreprise qui permet d'aligner des objets sur un rail pour se diriger vers un poste de production. Il se caractérise par :

- le contenu possible (volume ou la longueur du linéaire)
- le dispositif de sélection et de présentation des composants

Il est nécessaire que la machine soit adaptée au type d'objet. Ceci rend difficile l'adaptation d'une machine pour amener des objets différents. De

nombreux brevets ont été proposés afin d'améliorer les systèmes d'aménages [Lipe 1972]. Le procédé des bols vibrants a un rendement quasi constant, tous les objets se suivent les uns après les autres. De plus, il est très rapide. L'acquisition de ce type de machine s'effectue la plupart du temps par l'achat auprès d'un fournisseur qui doit adapter la machine au type d'objet à saisir ce qui est un désavantage conséquent pour l'adaptabilité, le coût, et la répétabilité.

2.3.2 Le "Pick-and-Place" : Avec un robot et un convoyeur

Un autre mécanisme d'aménagement consiste à utiliser la reconnaissance d'image en deux dimensions. Comme indiqué dans la Figure 2.2, les objets sont positionnés sur un convoyeur. Une caméra prend une photo et identifie les objets sur le convoyeur.

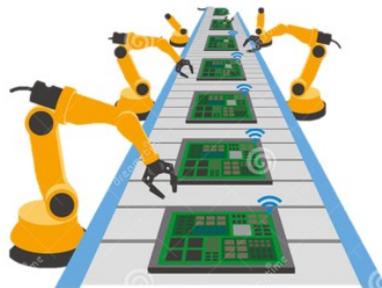


FIGURE 2.2 – Les objets sont pré-placés sur un convoyeur et sont ensuite sélectionnés par le ou les robots (source : "Logistik Heute, Arbeit 4.0 : Wo bleibt der Mensch?")

La position en X et Y et l'orientation de l'objet est déterminée et retransmise au robot. La profondeur étant unique, le robot peut aller directement chercher l'objet. L'avantage de cette méthode est la rapidité d'adaptation. Il est nécessaire d'apprendre le nouvel objet à saisir. Présentés dans le chapitre suivant, de nombreux algorithmes ont été proposés dans ce domaine [Ikeuchi 1987, Piccinini 2012, Buchholz 2016]. L'inconvénient de cette

méthode est le pré-positionnement des objets, coûteux en temps et en place car le système nécessite un convoyeur. L'adaptabilité est le plus gros avantage de ce type de système. Les méthodes de détections actuelles permettent de repérer un grand ensemble d'objets. Le pré-positionnement des objets les uns après les autres sur un convoyeur est le principal inconvénient de cette méthode en 2D, qui nécessite des opérations de préparation humaines, au préalable. Ce type de système prend beaucoup de place et peut être onéreux selon le type de convoyeur utilisé. La répétabilité dépend de la qualité de l'algorithme pour détecter des objets et du pré-placement de ceux-ci.

2.3.3 Le "Bin-picking" : Avec un robot, plusieurs prises images

Une des solutions consiste à utiliser un ensemble d'images avec une ou plusieurs caméras calibrées et de calculer la position et la localisation de l'objet à saisir.

Comme expliqué dans la petite bande dessinée en Figure 2.3, le procédé est divisé en un ensemble d'étapes préétablies :

1. En amont : les caméras sont calibrées et le système reçoit en entrée un modèle d'objet à chercher
2. Le système robot/caméra est à l'arrêt
3. Le système est mis sous tension (le démarrage peut s'effectuer soit par un opérateur ou par une machine préprogrammée)
4. Les caméras recherchent l'objet à saisir dans la scène à l'aide d'une base de données
5. Un ensemble d'informations est récupéré pour l'objet sélectionné (taille, angle...)

6. Le système calcule les coordonnées 3D de l'objet, ainsi que l'orientation de la pince du robot qui lui permettra de saisir l'objet
7. Le robot attrape l'objet
8. Le robot dépose l'objet à l'emplacement défini.

Une fois installé et configuré, ce système est plus adaptable qu'un bol vibrant car aucune modification physique n'est nécessaire pour un changement d'objet. Le système nécessite uniquement un ensemble PC, caméra et robot. Avec la tendance actuelle d'améliorer les systèmes de détection d'objets dans une scène, ce système est le plus efficace pour travailler avec différents types d'objets. La répétabilité et le rendement vont dépendre du type d'algorithme de reconnaissance utilisé et de la morphologie des objets à saisir.

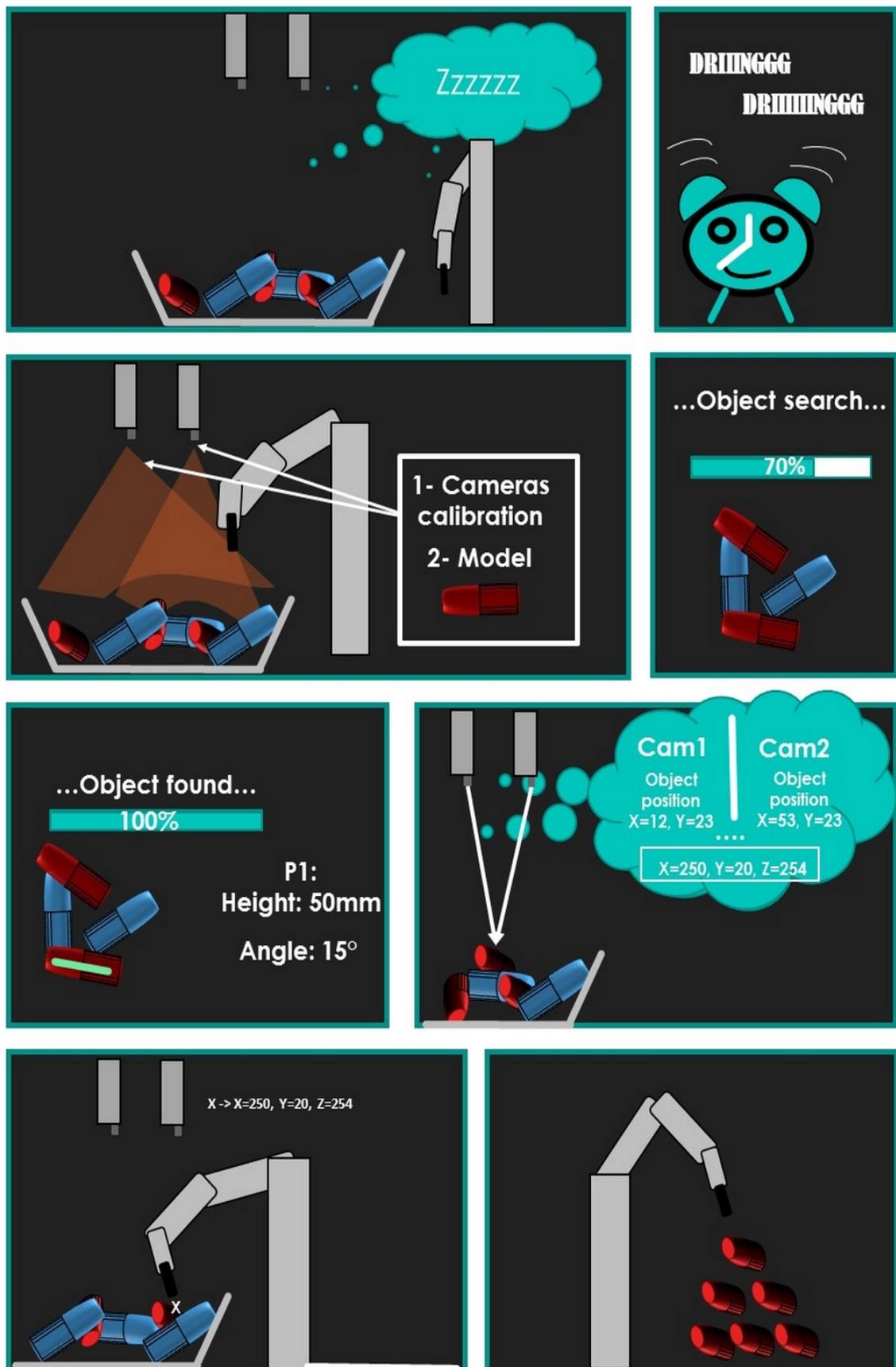


FIGURE 2.3 – Bande dessinée pour la représentation d'un système de "bin-picking" avec un robot, des caméras, et une bassin d'objets

2.3.4 Comparaison

La Table 2.1 synthétise les avantages et inconvénients des procédés étudiés. Le "bin-picking" est une des méthodes qui s'avère la plus avantageuse. Elle est très adaptable, ce qui est une des caractéristiques les plus importantes pour les systèmes d'amenage. Malgré de nombreux avantages, elle n'est pas forcément la plus efficace. Avec une cadence élevée, l'objet est difficilement saisissable par un robot.

Machine	Bol vibrant	Convoyeur	Bin-picking
Coût	Mauvais	Moyen	Bon
Installation	Mauvais	Moyen	Bon
Adaptabilité	Mauvais	Bon	Bon
Rendement	Bon	Mauvais	Moyen
Répétabilité	Bon	Moyen	Moyen
Bilan	3/5	3.33/5	4.33/5

TABLE 2.1 – Récapitulation des méthodes d'amenage avec leurs avantages

2.4 Le "Pick-and-Place"

Dans cette partie, nous allons étudier le domaine du "Pick-and-Place" et plus particulièrement le traitement adéquat pour la détection d'objets dans une scène.

2.4.1 L'historique du "Pick-and-Place"

Le terme robot a été introduit en 1920 par Karel Capek [Capek 1920] dans la pièce de théâtre "Rossum's universal robots" [Buchholz 2016]. Robot est la traduction tchèque du mot travail. Le terme a été ensuite appliqué à de nombreux dispositifs mécaniques tels que les téléopérateurs ou encore les véhicules

sous-marins... Tout ce qui opère avec de l'autonomie et sous le contrôle d'un ordinateur peut être appelé un robot. Suite à la Seconde Guerre mondiale, la robotique a commencé à voir le jour grâce à l'action de manipulateurs mécaniques télécommandés pour des matériaux radioactifs. Le système fonctionne sur le principe maître esclave (où le maître est le manipulateur "utilisateur") et où l'esclave effectue l'ensemble des actions. La robotique a été traversée par un certain nombre de points importants comme :

- 1947 : le développement du premier téléopérateur
- 1954 : la conception du premier robot programmable
- 1963 : le développement du premier système de vision pour robot
- 1973 : le développement d'un langage de programmation pour robot (wave)
- 1976 : l'utilisation d'un bras robotique sur les sondes spatiales Viking I et II (mars)
- 1979 : la conception du robot Scara
- 1986 : le robot sous-marin Jason explore l'épave du Titanic
- 1997 : le premier concours de robots qui jouent au football
- 2001 : la première opération réalisée à distance où les chirurgiens sont à New York et la patiente est à Strasbourg (France)
- 2001 : l'utilisation de robots pour rechercher des victimes sur le site du World Trade Center

L'un des objectifs de la robotique est de pouvoir saisir des objets avec un préhenseur. Buchholz effectue une étude approfondie des systèmes de "Pick-And-Places" dans son livre "Bin-picking - 5 decades of research" [Buchholz 2016]. Il fait référence à une des premières publications dans le contexte d'analyse de scène apparue en 1963 dans la thèse de Lawrence Gilman Roberts pour l'extraction d'objets [Roberts 1963]. Cette thèse intègre un état de l'art sur les méthodes de reconnaissance de formes (Eng : Pattern Recognition). L'objectif était de détecter des objets polyédriques dans une scène. Roberts extrait

les bords des objets pour en détecter les polygones et analyse pour ne générer que des polyèdres 3D. La scène 3D est ensuite générée et visualisée à partir de différentes vues.

Dans les années 60, Forsen [Forsen 1967] eut l'idée de projeter des points de lumières sur une scène pour calculer la profondeur. C'est en 1971 que Shirai et Suwa améliorèrent cette idée et proposent un des premiers algorithmes pour la détection d'objets 3D en développant un système utilisant un projecteur équipé d'une fente verticale pour projeter de la lumière sur la scène à analyser [Shirai 1971]. Avec la pose connue du projecteur par rapport à la caméra, les coordonnées des points 3D peuvent être calculées géométriquement le long de la ligne visible.

C'est en 1975 qu'apparaît une des premières applications industrielles, dans le but de classer et localiser des objets sans occlusion dans une pile d'autres objets [Tsuji 1975]. Pour ceci, une partie de l'objet doit être visible. Pour détecter les objets, des formes simples (arcs, les ellipses...) sont détectées en sauvegardant les points appartenant au gradient de l'image. Les points sont ensuite comparés à une base de données du modèle.

En 1977, Baird [Baird 1977] propose de manipuler des objets positionnés sur des convoyeurs. Cette méthode a été améliorée par Bolles [Bolles 1983] développant une méthode similaire pour détecter des objets plats posés sur une table. L'algorithme détecte les bords de l'objet. Les objets doivent avoir des motifs. Ces motifs peuvent être représentés par des trous ou des coins facilement reconnaissables.

En 1978, Perkins [Perkins 1978] propose de détecter les bords et les courbes fermées de l'image d'entrée et de les comparer à une base de données modèles des objets pouvant ainsi déterminer la pose de l'objet en calculant le centroïde des courbes correspondantes. Pour effectuer la mise en correspondance entre les courbes, 11 différentes caractéristiques ont été décrites comme la longueur totale, le rayon, la magnitude du changement angulaire...

En 1984, Winkelbach [Winkelbach 2002] propose une méthode pour estimer l'orientation de l'objet par rapport à la caméra en utilisant le principe de stéréo photométrie et des images de gaussiennes étendues (Eng : [Extended Gaussian Images \(EGI\)](#)). Le principe de la stéréo photométrie est de reconstruire une image 3D à partir d'une seule prise de vue et un ensemble d'images prises sous différentes luminosités. Plus il y a d'images, plus la précision est importante. Bolles et Horaud ont décrit une des premières approches pour identifier l'emplacement d'objets dans des bassines utilisant des capteurs 3D et des modèles d'objets [Bolles 1987].

2.4.2 Le "Pick-and-Place" : Un domaine pluridisciplinaire

D'après le livre "Bin-Picking : 5 decades of research" [Buchholz 2016], le "Pick-and-Place" est une méta-discipline qui combine plusieurs sous-disciplines telles que :

- l'analyse de scène et la reconnaissance d'objets dans le but d'identifier un objet parmi plusieurs dans une scène
- la localisation d'objets en déterminant les coordonnées réelles de celui-ci en fonction d'un point référencé dans l'espace
- le positionnement du préhenseur dans le but de déterminer l'inclinaison que doit avoir le préhenseur pour saisir correctement l'objet
- le chemin de prise du robot pour déterminer comment l'objet doit être saisi. Si le chemin n'est pas correct, le robot peut heurter l'objet sans le saisir.

Dans les débuts de la vision, les robots n'étaient pas aussi présents qu'actuellement, raison pour laquelle de nombreuses publications ne mentionnent qu'une utilisation possible des approches de la robotique. Par conséquent, les développements proposés dans la littérature traitent souvent des sujets

uniques pouvant être appliqués à d'autres modèles. Nous nous focalisons ici sur la reconnaissance d'objets dans une scène afin de récupérer les coordonnées X, Y, Z et l'orientation de l'objet à saisir avec le robot.

2.4.3 Le "Pick-and-Place" : les techniques actuelles

Buchholz [Buchholz 2016] sépare les techniques de localisation d'objets en deux catégories. Technique basée sur le vote et technique basée sur la correspondance.

technique basé sur le vote La technique basée sur le vote ([Eng : voting based technic](#)) utilise un système ou chacune des données vote pour toutes les solutions possibles. Par exemple, la méthode la plus connue utilise la transformée de Hough [Ballard 1981, Buchholz 2016]. Sur la base d'un modèle de l'objet à localiser, l'idée de la transformée de Hough est que chaque point de données détecté vote pour toutes les positions possibles du point de référence prédéfini du modèle. Le point dans l'espace de Hough qui obtient le plus grand nombre de voix donne les coordonnées du modèle par rapport au capteur. La dimension de l'espace Hough est définie par les degrés possibles de liberté de l'objet, par exemple six si la pose n'est pas restreinte du tout. La résolution des espaces de Hough définit la précision possible de la localisation. Pour chaque point de la numérisation, l'ensemble complet des points du modèle doit être accumulé.

La technique basée sur la correspondance La technique basée sur la correspondance ([Eng : correspondence based technic](#)) utilise des méthodes basées sur des descripteurs principalement utilisés pour le suivi ([Eng : tracking](#)) dans des systèmes d'images en 2D. Ces techniques adaptées aux données 3D peuvent être utilisées pour localiser des objets. Ces méthodes seront plus détaillées dans le chapitre suivant.

2.5 Les Problématiques d'identifications

La prise d'un objet nécessite la réalisation d'un ensemble de critères dans le but de récupérer la pose 6D de l'objet qui va permettre à un robot de le saisir. La pose 6D est définie par l'orientation 3D et la position 3D de l'objet dans la scène [Brachmann 2014]. Pour un objet non fin et non droit, il est nécessaire d'identifier un ensemble de points pour estimer la position et l'orientation de l'objet. L'objectif est de prendre des composants les plus diversifiés qu'ils soient, dans des situations et des positions difficiles à traiter comme l'enchevêtrement d'un objet par rapport à l'autre. Le temps défini par l'entreprise de détection et de saisie doit être inférieur à 2 secondes. Certaines solutions permettent déjà de gérer ce dispositif d'alimentation du robot avec des objets par comparaison avec un modèle image. Nous allons voir dans cette partie les différents systèmes permettant de détecter des objets tels que :

1. la caméra laser
2. la stéréoscopie

2.5.1 La caméra laser

La caméra laser ou encore appelée le scanner tridimensionnel utilise un faisceau laser pour déterminer les contours et avoir la profondeur. La technologie fonctionne en envoyant un faisceau laser infrarouge de faible puissance, qui balaie un objet ou une scène [Bernardini 2002, Varady 1997]. Lorsque la lumière est réfléchi, des photons sont dispersés dans des directions différentes. Un détecteur mesure le temps qu'il faut à une particule de lumière pour revenir à l'appareil et calcule la distance entre le système et l'objet. Cette technologie peut être très précise. Les capteurs 3D forment un nuage de points 3D nécessitant une analyse pour en détecter les objets à rechercher. L'objet est situé à l'intérieur du nuage de points. Pour le détecter, il est nécessaire de déterminer les 6 degrés de liberté de l'objet. Les degrés de liberté de l'objet

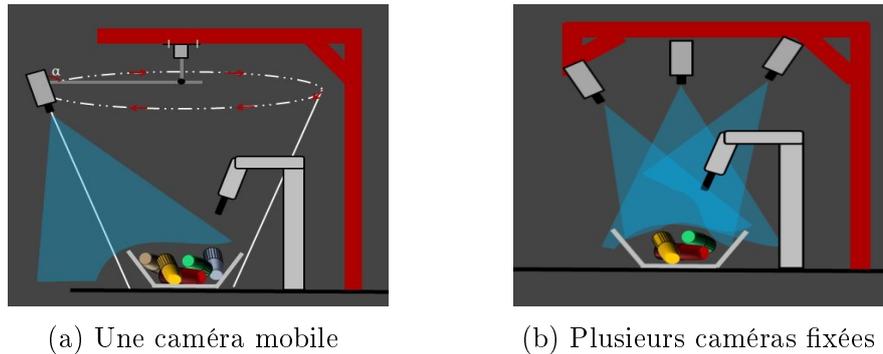


FIGURE 2.4 – Systèmes de "Pick and Place"

correspondent à sa position dans la scène. Les algorithmes qui déterminent la position des objets recherchent des caractéristiques souvent planes ou cylindriques dans le nuage de points. Cependant, ces caractéristiques ne sont pas toujours présentes sur les objets [Buchholz 2016].

2.5.2 La stéréoscopie

Cette méthode consiste à prendre plusieurs prises de vue de la scène à traiter. Le but est de pouvoir traiter ensuite les données pour déterminer la position et la localisation de l'objet. Cette méthode est assimilable à celle de la localisation et du calcul de distance par triangulation. En prenant plusieurs prises de la scène, par des algorithmes ou de "simples" calculs, nous pouvons retrouver les distances, en les comparant avec des données internes. En prenant plusieurs vues d'une scène, nous pouvons déterminer certaines dimensions de l'objet et ainsi les comparer à un modèle. La distance nous permet d'évaluer la distance de l'objet, par la différence de taille. À la différence, de la caméra laser, le traitement peut être effectué dans un espace en deux et trois dimensions. Dans notre cas, l'image en deux dimensions permet de simplifier l'image. Comme l'indique la Figure 2.4a et la Figure 2.4b, il existe plusieurs types de dispositifs de prises de vues : deux caméras positionnées à deux endroits différents ou une caméra mobile. Dans notre cas, nous utilisons

un ensemble de caméras à une position fixe.

2.6 Conclusion

Nous avons résumé dans cette partie les différents types d'aménagement les plus fréquents en entreprise et introduit les notions techniques nécessaires pour la compréhension de la suite de la thèse. Une des demandes de l'entreprise est d'utiliser un système stéréoscopique avec une ou plusieurs caméras embarquées et de détecter des objets pour calculer leurs positions et leurs orientations. Nous avons donc choisi d'utiliser le système d'aménagement avec plusieurs caméras fixes et une bassine. Cette méthode a été choisie car elle est adaptable et économique et en pleine expansion dans le monde de la recherche.

Traitement d'images pour l'extraction de pièces

Sommaire

3.1	Introduction	32
3.2	Les types de pièces à détecter	33
3.3	Représentation du modèle d'entrée	35
3.4	Identifications en temps borné	36
3.5	Extraction de points d'intérêt	37
3.6	Approximation polygonale	43
3.7	Discussion et conclusion	47

“ Certains peintres transforment le soleil en un point jaune ;
d'autres transforment un point jaune en soleil. ”

Pablo Picasso

3.1 Introduction

L'utilisation de caméras nécessite une étape de détection d'objets par traitement d'images. La détection d'objets est une tâche délicate car celui-ci peut être orienté et disposé de manière aléatoire dans la scène. L'enjeu est de détecter un objet dans une scène remplie de contraintes. Ces contraintes peuvent être dues à l'ajout d'autres objets similaires dans la scène, au bruit que peut avoir l'image et qui nuit à la comparaison entre les objets présents dans la scène et le modèle. Dans un système de "Pick-and-Place" doté d'un robot et de caméras fixes, il est nécessaire d'effectuer un traitement ou une analyse des images récupérées par la ou les caméras afin de récupérer les informations à transmettre au robot. L'objectif est d'identifier des objets dans une image. Le traitement d'images permet ainsi de simplifier l'image et de récupérer des informations pertinentes comme le contour d'un objet en utilisant des algorithmes tels que Canny [Canny 1986] ou Sobel [Sobel 2014]. De nombreux procédés de traitement d'images existent [Gonzales 2002, Acharya 2005, Jain 1989], notamment dans la recherche de pièces non fines dans une bassine pour les tâches de "Pick-and-Place" [Ikeuchi 1987, Kumar 2014, Piccinini 2012, Ulrich 2012]. Nous appelons un objet fin, une pièce avec une base avec très peu d'épaisseur. La largeur doit être représenté par une petite quantité de pixels. L'image récupérée par la caméra est souvent bruitée par la luminosité et par les autres pièces dans les cas d'enchevêtrement. À l'inverse, une image d'un fichier CAO pouvant être utilisée comme modèle ne contient que des informations non bruitées comme le contour de la pièce et la présence ou non de matière. De manière générale, deux critères sont pris en compte pour l'identification d'objets dans une scène. La première est le calcul de la fonction de coût (Eng : *Fitness function*) et la deuxième est le temps de traitement. La première évalue la pertinence de l'élément sélectionné en fonction d'un ensemble de critères tel que la mise en correspondance des contours ou des points

clés d'une image. Si le modèle de comparaison est un fichier CAO, alors une infinité de possibilités d'orientations du modèle est générable. Dans ce cas, il est nécessaire d'avoir un algorithme de comparaison rapide (quelques millisecondes). Dans un temps borné, plus l'algorithme est rapide, plus le nombre d'orientations du fichier CAO à générer est important. Ceci impacte directement la qualité de recherche de l'objet dans l'image. Certains algorithmes de comparaison tels que les algorithmes de comparaison de points d'intérêt correspondant à des discontinuités d'intensités dans l'image nécessitent un apprentissage à l'aide d'un modèle image pour un ensemble d'inclinaisons de la pièce. Cet apprentissage nécessite un temps de pré-traitement qui dépend de la précision et de l'inclinaison possible de la pièce que l'utilisateur souhaite avoir. Les algorithmes par comparaison de contours sont sensibles aux bruits, ceux-ci apparaissent par l'enchevêtrement d'une pièce par rapport à l'autre qui modifie la qualité de la segmentation de la pièce. Afin de comparer un maximum d'images modèles à l'image de la caméra, un algorithme rapide dérivé de SIFT appelé SURF [Bay 2006] a été proposé, cependant cet algorithme est sensible à l'orientation de la pièce et ne permet donc pas de retrouver l'orientation de la pièce en fonction du fichier CAO. Cependant et comme présenté dans la Section 3.5.3, cet algorithme est très rapide et peut comparer une image en une centaine de millisecondes. Nous présentons dans ce chapitre les différents types d'objets à détecter, le modèle d'entrée nécessaire, l'identification en un temps borné et les différents algorithmes avec leurs avantages et leurs inconvénients.

3.2 Les types de pièces à détecter

Pour avoir un système de "Pick-And-Place" performant, il est nécessaire d'identifier le type d'objet que l'on souhaite saisir. La morphologie, la matière et les motifs représentés sur l'image de l'objet vont avoir un impact sur le type

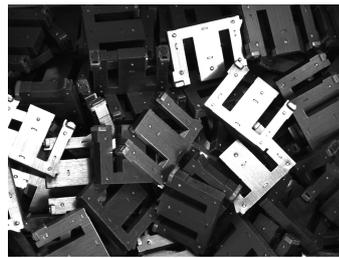
d'algorithme à utiliser.

3.2.1 La morphologie

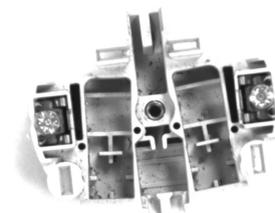
La morphologie de l'objet a une grande importance sur le type d'algorithme à utiliser. Comme représenté dans la Figure 3.1a, les pièces représentées peuvent être fines et difficilement séparables les unes des autres. Dans un autre cas, les objets représentés dans la Figure 3.1b peuvent être plus épais. Comme représenté dans la Figure 3.1c, l'objet peut être complexe et avoir plusieurs sous parties à la différence des précédentes pièces représentées par un seul bloc. Des caractéristiques telles que la transparence peuvent influencer le choix de l'algorithme.



(a) Image représentant des pièces fines enchevêtrés les unes sur les autres



(b) Image représentant des pièces non fines enchevêtrés les unes sur les autres



(c) Objet avec beaucoup de contraste

FIGURE 3.1 – Exemples d'images de pièces photographiées

La matière et les motifs présents sur l'objet ont un impact majeur sur sa recherche. Par exemple, le métal réfléchit les sources de lumières externes à la scène et peut engendrer une difficulté d'identification.

3.2.2 Les difficultés de reconnaissance des objets dans une scène

La facilité de détection d'un objet dans une scène varie en fonction de plusieurs critères. Premièrement et comme présenté précédemment, la pièce en elle-même joue un rôle majeur. Ceci est dû à ses caractéristiques photométriques. Par exemple et comme pour l'humain, un objet transparent ou métallique est plus compliqué à détecter car celui-ci peut être bruité par une source extérieure. La position et l'inclinaison de l'objet dans la pièce jouent aussi un rôle important. L'enchevêtrement d'une pièce par rapport à d'autres joue aussi un rôle important. Il est nécessaire pour ceci de pouvoir se détacher d'une recherche ou l'objet apparaît entièrement.

3.3 Représentation du modèle d'entrée

La représentation du modèle définit l'ensemble des caractéristiques qui seront comparées avec l'image à traiter. Il existe différentes façons de représenter un modèle. Premièrement, celui-ci peut être sous forme d'images. Celles-ci sont sauvegardées dans une base de données. Un ensemble d'informations sur ces images est comparé avec l'image à traiter. L'avantage de cette méthode résulte en sa facilité de mise en place. Elle nécessite une prise d'images cohérentes par l'opérateur où les données à ressortir sur l'image doivent être en adéquation avec l'image à traiter. La difficulté se situe dans la détermination de l'algorithme d'extraction des caractéristiques à comparer, car celui-ci doit s'adapter au type de pièce que l'on souhaite identifier. Deuxièmement et comme représenté dans la Figure 3.2, certains modèles peuvent être représentés sous forme de fichier CAO. La récupération d'un fichier CAO est un peu plus compliquée que de récupérer un ensemble d'images. Il est évident que la prise d'une image en la photographiant est plus simple que de devoir dessiner une pièce sur un or-

dinateur. Le fichier CAO est orienté dans différentes positions et transformé en image. Ce type de modèle d'entrée implique souvent l'utilisation d'algorithmes de comparaison de forme pour détecter l'objet. Troisièmement, un modèle peut être représenté sous forme de données. Par exemple, un opérateur peut donner des données telles que la taille d'une pièce, l'angle entre deux points distincts... Cette méthode est plus simple pour l'opérateur, car il doit juste donner des informations facilement récupérables.

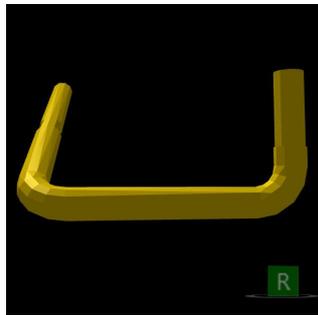


FIGURE 3.2 – Représentation d'un fichier CAO

3.4 Identifications en temps borné

Dans les milieux applicatifs (industriel, application mobile...), le traitement d'images nécessite souvent des algorithmes proches du temps réel. L'utilisateur veut avoir une réponse en un minimum de temps. Les algorithmes standard (SIFT, approximation polygonale) utilisés actuellement en entreprise sont rapides (moins de 500ms) mais détectent des objets avec des caractéristiques où les bords représentés par les images prises par les caméras font facilement ressortir l'objet. Le critère de temps est important et perfectible de différentes manières (parallélisation, algorithmes d'optimisation...). Il a été nécessaire dans cette thèse de proposer une solution innovante quant aux types d'objets à rechercher, mais aussi à l'optimisation des temps de traitements.

3.5 Extraction de points d'intérêt

Les algorithmes d'extractions de points d'intérêt calculent des "descripteurs" qui sont des informations indépendantes de l'image. L'objectif est de faire abstraction d'un ensemble de données de l'image comme la résolution du capteur, le cadrage, l'angle d'observation et l'exposition. Cette méthode est utilisée dans différents domaines [Gupta 2009] et notamment dans les domaines de la détection d'objets [Lowe 1999] et la robotique [Se 2001]. Les algorithmes ont notamment été adaptés pour les données en 3 dimensions [Flitton 2010] [Allaire 2008]. Nous détaillons dans ce chapitre les algorithmes les plus utilisés comme SIFT, PCA-SIFT et SURF et nous les comparons en deuxième partie. La troisième et quatrième partie présentent les avantages et les inconvénients de ces procédés. Comme représenté dans la Figure 3.3, des points clés sont détectés entre le modèle (partie gauche) et l'image à traiter (image droite) et mis en corrélation.

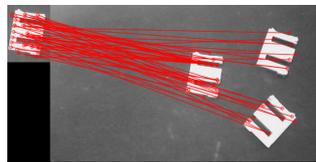


FIGURE 3.3 – Extraction de points clés à l'aide de l'algorithme SURF (120 ms)

3.5.1 SIFT

L'algorithme SIFT proposé par Lowe en 1999 est le premier algorithme d'extraction de caractéristiques [Lowe 1999, Lowe 2004]. Son nom traduit par "transformation de caractéristiques visuelles invariante à l'échelle" (Eng : [Scale-Invariant Feature Transform](#)) transforme une image en une collection de vecteurs. Les vecteurs doivent être distinctifs et invariants aux changements d'échelles, de rotations et de translations. Les vecteurs sont ensuite comparés

pour déterminer s'il y a une correspondance entre le modèle et l'image où la recherche de l'objet doit être effectuée. L'algorithme est séparé en plusieurs parties qui sont :

1. La création d'une différence de gaussienne (DOG) appliquée sur l'image. Elle est formée par la convolution de l'image originale avec des fonctions gaussiennes de largeur différente. La convolution permet d'estomper les petits détails présents dans l'image. L'équation de la différence de gaussienne est définie par $D(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma)$. Le coefficient k correspond à la finesse de la discrétisation de l'espace des échelles souhaitées. Le gradient du facteur d'échelle $L(x, y, \sigma) = G(x, y, \sigma) * I(x, y)$. Dans cette formule I est l'image, G le filtre gaussien de paramètre σ (facteur d'échelle).
2. La détection des points clés candidats provenant de la différence de gaussienne. Chaque point est comparé avec ses voisins. Si la valeur est minimale ou maximale, alors elle est considérée comme un point extrême. L'utilisation d'une pyramide permet d'optimiser le temps de calcul des images floutées sur un nombre élevé d'échelles différentes. La base de la pyramide représente l'image d'origine.
3. La détection des points extrema produit un grand nombre de points clés candidats. Il est nécessaire d'éliminer les candidats les plus faibles. Ceux-ci se démarquent par leurs faibles contrastes où la localisation est souvent située sur les arêtes.
4. L'assignation de l'orientation est effectuée à chaque point clé une ou plusieurs orientations. Elles sont basées sur la direction située dans le voisinage du point candidat. Un histogramme d'orientation est formé en fonction du gradient d'orientation des points d'échantillonnage qui se trouvent à l'intérieur d'une région située autour du point clé.
5. La distinction des candidats. Chaque point clé est assigné à un des-

cripteur pour la région locale de l'image. Une fonction de pondération gaussienne en fonction de l'échelle est utilisée comme valeur de magnitude. Les coordonnées du descripteur et le gradient de l'orientation sont déterminés en fonction de l'orientation du point clé. Ceci permet de pallier le problème de changement d'orientation de l'objet.

6. La mise en correspondance. Dans cette étape, les points clés et les descripteurs sont comparés en utilisant la distance euclidienne de chaque point. Plus le descripteur des deux points clés est proche, plus la mise en correspondance est bonne.

L'algorithme SIFT a l'avantage d'être invariant au changement d'échelle et de rotation et d'avoir des résultats cohérents sur des images floues. Il est le premier algorithme proposé et reste la référence dans l'extraction de points d'intérêts.

3.5.2 PCA-SIFT

La méthode PCA-SIFT [Ke 2004] est une technique similaire à SIFT avec une réduction de la dimension et de l'échantillonnage, ce qui rend l'algorithme plus vélocité et moins gourmand en termes de stockage d'informations des points clés. A la différence de SIFT, PCA-SIFT utilise un histogramme pour normaliser le gradient correctif. Sa représentation est sous forme vectorielle. Elle est plus restreinte que celle de SIFT. Pour effectuer la mise en correspondance, l'algorithme utilise la distance euclidienne entre les deux vecteurs où chaque vecteur représente un point clé dans l'image.

L'algorithme PCA-SIFT a l'avantage d'être plus rapide que SIFT mais est sensible aux images floues et au changement d'échelles.

3.5.3 SURF

L'algorithme SURF [Yang 2008, Bay 2006] est une technique similaire à SIFT mais diffère en termes de détection de points caractéristiques. SURF crée une pile sans échantillonnage 2 : 1 pour les niveaux élevés de la pyramide tandis que SIFT construit une image pyramidale en filtrant chaque couche avec une gaussienne. Il y a dans l'algorithme une notion d'empilement et de niveaux. Lors de la mise en correspondance des points clés, une distance euclidienne minimale est calculée entre le vecteur sélectionné et les plus proches voisins de celui-ci. Les distances sont ainsi comparées en fonction d'un ratio. L'algorithme SURF est le plus rapide. Plusieurs comparaisons des temps ont été effectuées dans la littérature [Juan 2009, Bay 2006]. En prenant un temps de référence de 1036 ms pour SIFT, l'algorithme SURF effectue la même recherche en 354 ms, soit un gain de 65% [Bay 2006]. Il est le moins sensible aux changements d'illumination. Cependant, il reste sensible au changement de rotation.

3.5.4 Comparaison

TABLE 3.1 – Récapitulatif des avantages et inconvénients des méthodes d'extractions de points clés [Juan 2009]

Méthode	Temps	Échelle	Rotation	Flou	Illumination
SIFT	Moyen	Meilleur	Meilleur	Meilleur	Moyen
PCA-SIFT	Bon	Moyen	Bon	Moyen	Bon
SURF	Meilleur	Bon	Moyen	Bon	Meilleur

Comme l'indique la Table 3.1, le choix de la méthode à utiliser dépend de l'application [Juan 2009]. L'algorithme SIFT va avoir de bons résultats en termes de reconnaissance d'objet avec une rotation, une variation d'échelle

et un bruit (flou) différent du modèle appris. Cependant, il reste sensible à l'illumination de l'image. Le temps d'exécution est plus long comparé aux algorithmes SURF et PCA-CIFT. L'algorithme PCA-SIFT est plus sensible aux déformations de l'image, mais pas à l'illumination. Son temps d'exécution est inférieur à SIFT. L'algorithme SURF est le plus rapide pour extraire les points clés. Il est efficace pour les cas d'illumination, mais reste sensible aux variations de l'image comme : l'échelle, la rotation et le flou comparé à l'algorithme SIFT.

3.5.4.1 Avantages

Ces méthodes sont excellentes pour la recherche d'objets dans une scène lorsque l'image modèle se retrouve dans l'image à traiter. L'apprentissage est simple car le système nécessite simplement en entrée des images du modèle à rechercher [Piccinini 2012]. Leur temps d'exécution est très rapide et permet de récupérer l'orientation d'un objet caché dans une image en un laps de temps court grâce à la mise en correspondance des points clés [Bay 2006].

3.5.4.2 Inconvénients

Ces algorithmes restent délicats pour comparer deux images avec des caractéristiques qui diffèrent. En prenant l'exemple d'un fichier CAO, il est difficile de mettre en correspondance une image d'un fichier pris par une caméra et celle d'un fichier CAO. Les seuls points de correspondance sont les contours et ceux-ci peuvent varier selon l'orientation de la pièce. Il en résulte des points clés difficiles à mettre en correspondance.

3.5.4.3 L'application pour les tâches de "Pick-And-Place"

En 2012 [Piccinini 2012], Piccinini propose une méthode basée sur SIFT pour effectuer des tâches de "Pick-And-Place". La procédure itère sur 4 phases

principales qui sont :

1. la détection en 2D des objets
2. la sélection d'un des objets souvent déterminé en fonction de l'optimisation de la trajectoire du robot pour saisir l'objet
3. la localisation 3D de l'objet en exploitant les paramètres de calibrage de la caméra pour obtenir les coordonnées réelles du point de saisie de l'objet sélectionné
4. la prise de l'objet en envoyant les coordonnées au robot

Piccinini utilise l'algorithme SIFT pour extraire des images (modèle et image de traitement). Il extrait les points clés et calcule les descripteurs. Il compare ensuite les points clés entre l'image acquise par la caméra et le modèle. Un schéma de vote est ensuite généré pour regrouper les points mis en correspondance entre les différents doublons. Ce processus implique une transformation euclidienne entre le modèle et l'image actuelle. En fonction de la mesure de similarité entre les deux images, les meilleures correspondances sont retenues. En fonction de l'ensemble des correspondances, l'algorithme détermine la meilleure localisation de l'objet détecté dans l'image actuelle.

3.5.5 Conclusion

Ces algorithmes utilisés dans le monde industriel et notamment dans les tâches de "Pick-and-Place" restent des standards. La comparaison d'un ensemble de points clés permet d'effectuer la recherche d'objets. Ils sont très efficaces en termes de corrélation d'éléments différents et permettent facilement d'avoir l'estimation de la pose 6D de l'objet par rapport à la caméra. Il est difficile de les utiliser pour la mise en correspondance des formes telles que pourrait avoir un fichier CAO car les points clés diffèrent d'un fichier CAO à une photo. Cependant, l'extraction de points clés peut être utilisée pour la

recherche rapide de pièces dans une bassine et ainsi effectuer la relation entre l'ensemble des objets et les modèles.

3.6 Approximation polygonale

Si l'objet ne possède pas de points clés, les contours de l'objet peuvent être comparés avec un modèle. Cette méthode très utile pour comparer des formes est importante dans le domaine de la reconnaissance d'objets [Stein 1992]. L'approximation polygonale ou polygonalisation est un algorithme de transformation d'une chaîne de points connexes en une liste successive de segments [Naouai 2013, Garnesson 1992]. Cette méthode est utilisée pour la reconnaissance d'objets où le modèle est aussi une image. Son objectif est de réduire le nombre d'informations de chaque chaîne de segment en fonction d'une tolérance. Les points sont souvent récupérés à la suite d'un algorithme de détection de contours. Les points de contours sont récupérés et transformés en droites. Les segments des droites sont comparées pour déterminer des zones d'appartenance entre le modèle et l'image à traiter. La difficulté réside dans la transformation d'un segment d'une droite en une série d'information comparable. De nombreuses méthodes et améliorations [Debled-Rennesson 2004] ont été proposées [Davis 1999] [Rosin 1997, Yin 2000, Douglas 1973]. Ce procédé a de nombreuses applications [Selvakumar 2013] comme la représentation des chiffres sur les plaques d'immatriculation des voitures et des avions [Kumar 2002], pour la compréhension des formes et la mise en correspondance d'images. Dans cette partie, nous commençons par présenter l'approximation polygonale. Nous présentons ensuite quelques algorithmes d'approximation polygonale. Nous poursuivons par les avantages et les inconvénients de l'approximation polygonale.

3.6.1 Fonctionnement

L'approximation polygonale transforme une suite de points connexes en un ensemble de segments. Pour ceci, une suite de points est transformée en un segment de droite caractérisée par des angles et des longueurs. Les caractéristiques sont ensuite comparées au modèle qui détermine une fonction de coût. La fonction de coût détermine la correspondance entre la droite de l'image et la droite du modèle.

3.6.2 Les algorithmes

Deux méthodes sont utilisées, la première par découpage récursif et la deuxième par "approximation itérative" [Pavlidis 1974, Kurozumi 1982]. L'algorithme de Ramer, aussi appelé méthode de la corde, est un des premiers à avoir été créé [Ramer 1972]. Un segment est testé pour savoir s'il répondra à une contrainte de norme particulière. Dans le cas contraire, la sélection d'un autre point est alors effectuée. Masood a proposé [Masood 2008] une méthode itérative qui utilise la polygonalisation inverse. Elle commence par extraire les points d'un contour (Exemple : l'algorithme Freeman). L'algorithme proposé par Masood supprime ensuite les points dominants qui sont déterminés en fonction de la valeur d'erreurs associées (VEA). Celle-ci est calculée en fonction des deux voisins dominants qui sont reliés par une ligne droite. L'algorithme de Masood [Masood 2008] est divisé en plusieurs étapes :

Premièrement la détection des points de rupture (Eng : [Break Point Detection](#)) est effectuée. Les points sont détectés à l'aide de l'algorithme *Freeman's chain-coding*. Chaque point est déterminé comme un point de rupture s'il n'est pas égal au précédent point traité. Les points dominants (DP) sont assignés aux points de fin des segments. La distance maximale perpendiculaire (carrée) de tous les points limites entre DP_{j-1} et DP_{j+1} de la ligne droite est appelée valeur d'erreur associée (AEV) des points dominants DP_j . Un score AEV est

Algorithm 1 Masood Algorithm**Require:**

Break Point Detection

Break points detection by using Freeman's chain-coding

Reverse Polygonization

Find break points from given boundary points

Mark detected break points as initial set of dominant points (DP).

Calculate AEV for all DPs.

DoFind dominant point (DP_j), having least AEV.Remove DP_j from the list of dominant points.Recalculate the AEV of neighboring dominant points (DP_{j-1} & DP_{j+1}).

Calculate MaxErr.

While(MaxErr < 0.9)

Leftover points make final set of dominant points

associé pour chaque point dominant. La deuxième partie supprime interactivement les points dominants de la table DP jusqu'à ce que l'approximation corresponde au résultat souhaité. Un point dominant avec un AEV minimum est éliminé à chaque itération. Après avoir supprimé un point dominant (DP) l'AEV de deux points dominants voisins (DP_{j-1}, DP_{j+1}) est recalculé et mis à jour dans la table DP. Nous avons développé un algorithme d'approximation polygonale dans le but de tester la robustesse de l'algorithme à trouver des pièces dans une bassine.

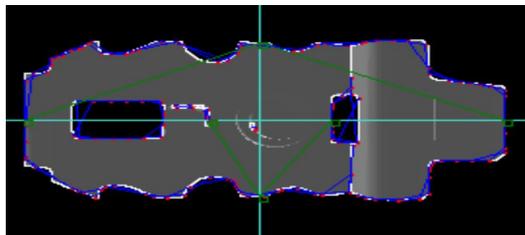


FIGURE 3.4 – Approximation polygonale sur un objet représenté sur un fichier CAO : Bleu = Contour, Rouge = Point dominant

Comme l'indique l'expérience de l'approche polygonale dans la Figure 3.4,

une approximation polygonale a été appliquée sur une image générée par un fichier CAO. Nous pouvons constater que les points de l'approximation polygonale sont bien mis en avant en rouge dans l'image. Ces points ont ensuite été comparés avec une image de l'objet prise dans une scène aléatoire dans le but de retrouver cet objet.

3.6.3 Avantages

Cette méthode est très utile pour comparer des objets avec des formes différentes et mises en avant dans l'image. Nous avons obtenu des temps de comparaison entre deux images proches du temps réel. Cette technique reste insensible au zoom et dépend de l'algorithme de segmentation utilisé auparavant.

3.6.3.1 Inconvénients

L'algorithme est sensible aux transformations géométriques de l'objet. Il est nécessaire d'avoir un modèle de comparaison de l'objet géométriquement identique à l'objet photographié dans la scène.

3.6.3.2 Conclusion

L'approximation polygonale permettant de convertir une suite de points en segments est un moyen rapide pour affiner la recherche du modèle en fonction de l'image. Si la segmentation est correctement effectuée, la comparaison des tailles des segments des deux images déterminera une fonction de coût qui permettra rapidement de comparer les deux images et d'orienter ainsi le modèle à trouver.

3.7 Discussion et conclusion

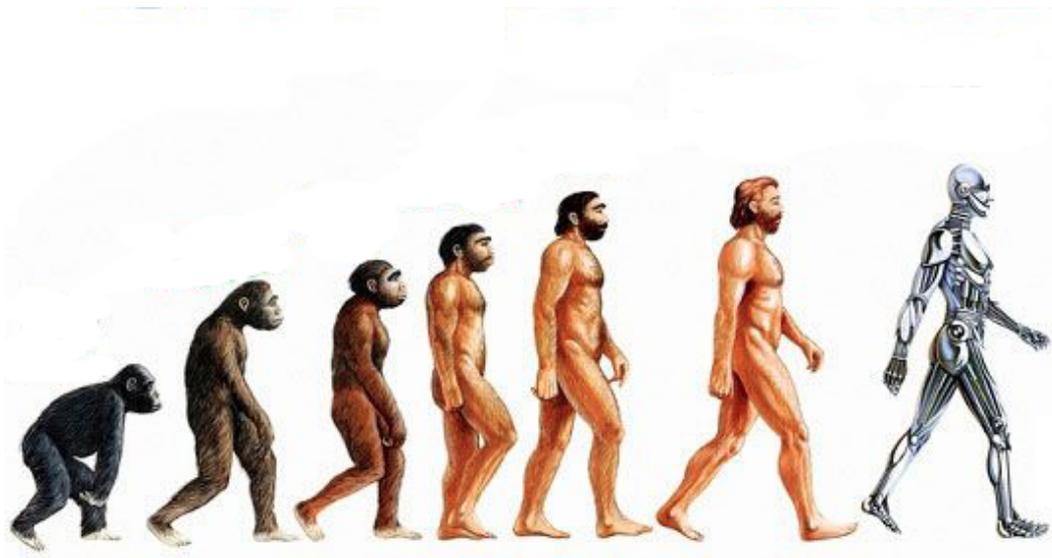
Tous les algorithmes proposés traitent des images où les objets sont imposants et/ou la topologie de l'objet est facilement reconnaissable. Ces algorithmes sont indispensables pour l'identification d'objet, notamment dans le domaine du "Pick-And-Place" où les objets diffèrent d'une application à une autre. Une méthode "universelle" n'est donc pas envisageable. Nous avons programmé un ensemble de ces procédés afin de proposer à l'entreprise un maximum de solutions d'identification. Nous avons programmé l'algorithme SIFT et SURF pour détecter des points d'intérêts dans une image et ainsi détecter des objets avec beaucoup de nuances. Nous avons aussi mis en place un algorithme d'approximation polygonale pour comparer des objets en fonction de leurs contours. Cependant, nous n'avons pas trouvé dans la littérature d'algorithme pour détecter des objets fins dans des temps inférieurs à une seconde. Aubert [Aubert 2005] a proposé une méthode pour détecter des structures fines dans des images en deux et trois dimensions. L'auteur propose un modèle mathématique pour identifier des singularités de codimension supérieure ou égale à deux dans une image, cependant l'algorithme met plusieurs dizaines de secondes pour effectuer le traitement. Une des possibilités serait d'utiliser les réseaux de neurones convolutifs (CNN). Ce sont des réseaux artificiels avec des interactions de connexion entre les neurones [LeCun 1998]. Ils consistent en une pile multicouche de perceptron, dont le but est de prétraiter de petites quantités d'informations. Ce succès est principalement dû aux grandes quantités de données disponibles (apprentissage) ainsi qu'à la rapidité des implémentations GPU. Ceci permet d'utiliser un grand ensemble de modèles non linéaires. Cette méthode est de plus en plus utilisée en particulier dans le domaine de la reconnaissance d'objets [Schwarz 2015, Schwarz 2016]. Il pourrait être un candidat efficient pour détecter n'importe quel type d'objet. Cependant, il nécessite un ensemble de données très important d'images.

Ce n'est pas le cas dans le monde industriel où les systèmes d'alimentation doivent être rapidement configurés. Concrètement, l'opérateur doit configurer la machine en prenant plusieurs dizaines d'images d'un objet à détecter, ce qui n'est pas le cas pour les CNN qui nécessitent un énorme ensemble d'images utilisées comme modèle. Nous devons donc proposer une nouvelle méthode d'extraction d'objets fins. Pour ceci, l'algorithme doit correctement détecter des objets fins et sans points clés. Afin de répondre aux exigences de temps inférieur à 2 secondes fixées par l'entreprise, nous proposons comme solution d'optimisation l'utilisation d'algorithmes évolutionnaires.

Algorithmes évolutionnaires

Sommaire

4.1	Introduction	50
4.2	Les familles d'algorithmes évolutionnaires	52
4.3	<i>Harmony Search</i>	54
4.4	Conclusion	61



4.1 Introduction

La recherche d'objets dans une bassine ou le calibrage de caméras sont des tâches qui peuvent être complexes. Quelle est la meilleure pièce à saisir dans une bassine ? Comment effectuer un calibrage optimisé ? Il est nécessaire de mettre en place des algorithmes d'optimisation. Apparus dans la fin des années 50, les algorithmes évolutionnaires (AE) proposent des méthodes d'optimisation stochastique simulant un ou plusieurs processus de l'évolution naturelle [Francisci 2002]. Plusieurs méthodes ont été proposées notamment dans les domaines des stratégies d'évolution, des algorithmes génétiques [Back 1997]. Les algorithmes évolutionnaires [Angeline 1998] visent à créer une intelligence autonome et à adapter leur comportement selon les circonstances [Eberhart 1998]. L'algorithme évolutionnaire est inspiré par la sélection naturelle, exposée par Darwin [Darwin 1872]. Il décrit un environnement où les personnes :

1. Évoluent
2. Survivent
3. Se reproduisent

Cette métaphore implique un environnement (*ie.* problème) où des individus (*ie.* candidats à une solution), évoluent jusqu'à un facteur de satisfaction. Celui-ci représente l'adaptation d'un candidat dans son environnement.

Les algorithmes évolutionnaires ont été une réussite dans différents domaines d'optimisation et d'apprentissage automatique (Eng : [machine learning](#)) [Al-salami 2009]. Ces algorithmes sont très puissants pour explorer des problèmes NP complet. De plus, l'avantage de cette méthode est la simplicité de conceptualisation du processus évolutionnaire qui est sous la forme de l'équation suivante :

$$x[t + 1] = s(v(x[t])) \tag{4.1}$$

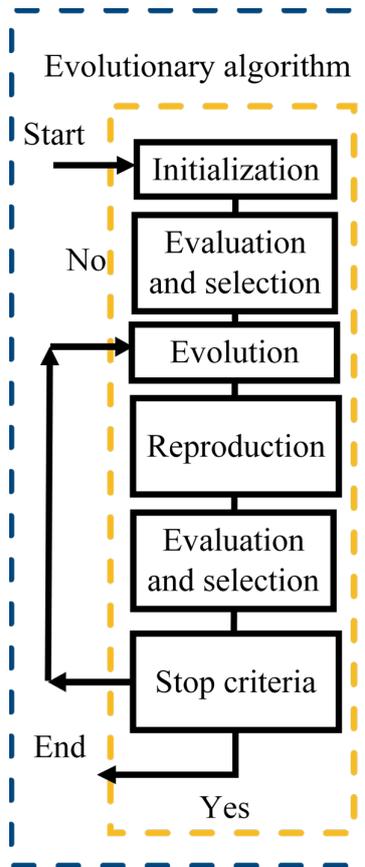


FIGURE 4.1 – Schéma fonctionnel d'un algorithme évolutionnaire

Où $x[t]$ est la population à un instant t , v est l'opérateur aléatoire de variation. Il est intéressant de mélanger différents algorithmes entre eux dans le but d'améliorer la qualité d'exploration des solutions. Ils sont intéressants pour couvrir un grand espace de recherche. Comme représenté dans la Figure 4.1, tous les types d'algorithmes évolutionnaires impliquent plusieurs phases spécifiques [Streichert 2002] :

- Étape 1 : Initialisation aléatoire de la population
- Étape 2 : Évaluation de tous les individus de la population
- Étape 3 : Sélection des individus de la population et génération de la population suivante
- Étape 4 : Vérification des critères d'arrêts de l'algorithme. Si les critères d'arrêts sont satisfaits, le programme s'arrête, sinon le programme

retourne à l'étape 2.

Il existe différents types d'algorithmes évolutionnaires, comme :

- les stratégies d'évolution
- les algorithmes génétiques
- la programmation évolutionnaire
- les programmations génétiques.

Ces algorithmes gardent le même schéma fonctionnel, mais diffèrent selon le domaine applicatif et le type de données à utiliser.

4.2 Les familles d'algorithmes évolutionnaires

4.2.1 Stratégies d'évolution (SE)

Spécificité : Individus représentés sous la forme de vecteurs ou de nombres réels. Les parents sont sélectionnés au hasard et les descendants les plus aptes sont utilisés pour la prochaine génération. Les individus sont autoadaptés. Apparue en 1965 [Schwefel 1965], les stratégies d'évolutions (SE) [Auger 2011] sont des algorithmes d'optimisation stochastique principalement utilisés pour l'optimisation de problèmes à boîte noire dans un espace de recherche continu. Par exemple, Mbikayi [Mbikayi 2013] l'utilise pour la transcription de musique polyphonique en un fichier MIDI. Ces algorithmes, arrivés dans les années 60 [Nikolaus 2013], reposent principalement sur l'application de la mutation, la combinaison et la sélection de solutions (candidats) dans les populations. Leurs principes prédominants sont l'impartialité et le contrôle adaptatif des paramètres de distribution. Les individus sont désignés comme parents ou descendants selon le contexte. Dans une procédure générale, un ou plusieurs parents sont choisis parmi la population et les nouvelles générations sont générées par la duplication et la combinaison de leurs parents. La population est modifiée en fonction de la nouvelle génération qui subit une mutation. La

sélection environnementale réduit la population à sa taille originale. Elle est appliquée comme une sélection par troncature et dépend de la fonction de coût des individus. Cet algorithme se différencie des autres par sa représentation des candidats sous forme de vecteurs de nombres réels.

4.2.2 Programmation évolutionnaire (PE)

Spécificités : Utilise la mutation et l'hybridation (reproduction à partir de plusieurs parents). Apparue en 1966 [Fogel 1966], la programmation évolutionnaire (PE) a été conçue pour faire évoluer des machines à état finies. La résolution d'un problème s'effectue par l'initialisation de la population où chaque individu génère ensuite un certain nombre de descendants en fonction de l'opérateur de mutation. L'opérateur de sélection permet de générer la prochaine génération à partir des parents et des enfants. La programmation évolutionnaire diffère des algorithmes évolutionnaires classiques, car il n'utilise pas une représentation spécifique des individus. La programmation évolutionnaire a été appliquée sur des problèmes tels que le problème du voyageur de commerce [Fogel 1993]. L'algorithme a aussi été amélioré en remplaçant la mutation sous forme de gaussienne par une mutation sous forme de Cauchy [Yao 1996].

4.2.3 Algorithmes génétiques (AG)

Spécificités : Utilise principalement la mutation. La sélection se fait en sélectionnant les parents en fonction de leur fonction de coût. Les individus sont donc évalués avant que la sélection ne soit effectuée. Apparue en 1975 [Holland 1975], les travaux de John Holland montrent comment un processus d'évolution peut être utilisé pour résoudre des problèmes au moyen d'une technique appelée algorithme génétique. Un algorithme génétique [Koza 1994] transforme une population d'objets individuels, chacun doté d'une valeur de

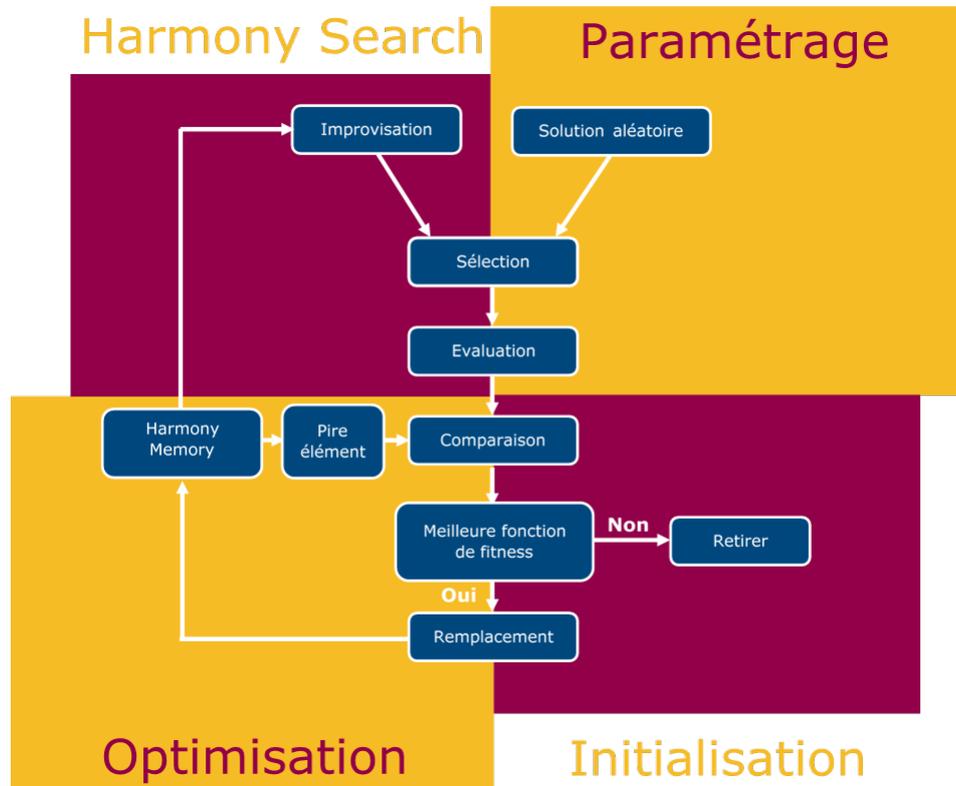
coût. L'algorithme se différencie des autres par sa représentation des candidats sous forme d'une chaîne binaire. À chaque nouvelle génération de la population, le système utilise le principe de survie et de reproduction des individus les plus forts. La population initiale évolue en concevant une nouvelle génération d'individus créée à l'aide d'un croisement de deux parents. L'objectif est d'obtenir une solution acceptable à une problématique d'optimisation où il n'existe pas de méthode dite exacte pour résoudre le problème en un temps raisonnable. Les algorithmes génétiques [Goldberg 1989] sont souvent utilisés dans le but de découvrir une solution avec peu d'information sur l'espace de recherche.

4.2.4 Programmation génétique (PG)

Spécificités : Faire évoluer des programmes. Apparue en 1992 [Koza 1992], la programmation génétique est inspirée des techniques algorithmiques génétiques dans le but de faire évoluer une population de programmes sous forme arborescente. La programmation génétique se différencie des algorithmes génétiques par la représentation des individus représentés par des programmes. Par exemple Zhang l'utilise pour la détection d'objet. Il divise l'algorithme en deux phases [Zhang 2009]. La première phase est l'apprentissage des données et la simplification de la fonction de coût. La deuxième phase est initialisée avec la première et utilise l'ensemble des données pour construire le programme de détection finale.

4.3 *Harmony Search*

Spécificité : Il est difficile de mettre *Harmony Search* dans une famille d'algorithme évolutionnaire. Il est souvent comparé dans la littérature avec les algorithmes à stratégies d'évolution mais il diffère par sa stratégie de re-

FIGURE 4.2 – Schéma fonctionnel de l'algorithme *Harmony Search*

cherche locale. On le compare aussi à un algorithme génétique mais diffère dans sa sélection de plusieurs valeurs dans sa population. Les algorithmes génétiques choisissent seulement deux candidats. L'algorithme *Harmony Search* (HS) est un algorithme heuristique inspiré par la musique qui a pour objectif de trouver la meilleure solution parmi des valeurs générées par un processus d'improvisation [Geem 2001]. Dans l'analogie, une solution est exprimée comme une harmonie, et la valeur comme une note jouée par les musiciens. Cet algorithme a été appliqué à divers problèmes d'optimisation tels que la résolution de Sudoku [Geem 2007], pour les problèmes d'optimisation de formes [Yildiz 2010] ou encore pour la conception du squelette de structure en acier [Saka 2009].

L'algorithme 2 et la Figure 4.2 indiquent la structure d'Harmony Search.

Il est divisé en 5 étapes qui sont :

- Étape 1 : Formulation du problème et définition des paramètres
- Étape 2 : Initialisation aléatoire de la population
- Étape 3 : Génération d'une nouvelle solution
- Étape 4 : Génération d'une nouvelle génération
- Étape 5 : Vérification des critères d'arrêt, continuation si ceux-ci ne sont pas satisfaits.

4.3.1 L'algorithme

4.3.1.1 Étape 1 : Formulation du problème et définition des paramètres

L'algorithme *Harmony Search* [Geem 2010b] commence par la formulation du problème. Cette partie permet ainsi de définir les paramètres tels que les éléments de satisfaction (fonction de coût) et les temps d'exécution. Pour exécuter l'algorithme, les paramètres suivants doivent être définis :

- *Harmony Memory Size* (HMS[1;+∞]) : Nombre de solutions à traiter dans l'algorithme
- *Harmony Memory Considering Rate* (HMCR[0;1]) : Taux auquel l'algorithme choisit une valeur aléatoire de la population existante (*Harmony Memory*)
- *Pitch Adjusting Rate* (PAR[0;1]) : Taux auquel l'algorithme choisit une valeur voisine à celle sélectionnée pour l'utiliser dans sa nouvelle solution

4.3.1.2 Étape 2 : Initialisation aléatoire de la population

Cette étape consiste à générer aléatoirement des solutions. Le nombre de solutions générées dépend du paramètre *Harmony Memory Size* (HMS). Un exemple de 6 solutions est présenté dans la Table 4.1 où chaque solution est

notée et sauvegardée dans l'*Harmony Memory* (HM).

TABLE 4.1 – Exemple d'initialisation aléatoire de la population de l'algorithme *Harmony Search*

Nombre	HM	Score
1	Solution 1	10
2	Solution 2	2
3	Solution 3	53
4	Solution 4	21
5	Solution 5	77
6	Solution 6	73

4.3.1.3 Étape 3 : Génération d'une nouvelle solution

Dans cette étape, et comme présenté dans la Table 4.2, une nouvelle solution est générée. Chaque élément de la solution X_{ij} peut être généré soit aléatoirement soit grâce à l'*Harmony Memory* (HM). Une valeur aléatoire est générée et comparée avec le paramètre *Harmony Memory Considering Rate* (HMCR : $0 \leq \text{HMCR} \leq 1$) dans le but de déterminer si la valeur est sélectionnée aléatoirement ou parmi les éléments de l'*Harmony Memory* (HM). Si la valeur est sélectionnée parmi l'*Harmony Memory*, une autre valeur aléatoire est générée. Elle est comparée avec le paramètre *Pitch adjusting Rate* (PAR : $0 \leq \text{PAR} \leq 1$) qui détermine si l'élément doit être muté en sélectionnant une valeur proche (voisine) de l'élément sélectionné. Le résultat final est un ensemble de valeurs correspondant à la nouvelle solution.

4.3.1.4 Étape 4 : Génération d'une nouvelle population

Si la nouvelle solution obtient un score supérieur au plus mauvais élément de la population de l'*Harmony Memory* (HM), alors l'élément est remplacé par la nouvelle solution. Comme montré dans la Table 4.3, le cinquième élément

TABLE 4.2 – Création d’une nouvelle solution de l’algorithme *Harmony Search*

Solution	Score
S{X1,...Xij}	17

TABLE 4.3 – Création d’une nouvelle génération de l’algorithme *Harmony Search*

Nombre	HM	Score
1	Solution 1	10
2	Solution 2	2
3	Solution 3	53
4	Solution 4	21
5	Solution 5	77
6	Solution 6	73
5	Solution 5	17

dans l’*Harmony Memory* (HM) a un score (fonction de coût) de 77. La solution au score de 77 est alors remplacée par la nouvelle solution qui a un score de 17.

4.3.1.5 Étape 5 : Vérification des critères d’arrêt

Représenté par *stop_alg()* dans l’algorithme 2, si un des critères d’arrêt est activé (nombre d’itérations, solution trouvée), l’algorithme se termine et peut ainsi retourner la meilleure solution. Dans le cas contraire, l’algorithme retourne à la troisième étape.

4.3.2 Variantes de la méthode

Plusieurs méthodes pour la sélection des valeurs des paramètres (*HMS*, *HMCR*, *PAR...*) ont été proposées. Plusieurs méthodes ont été soumises [Omran 2008, Taherinejad 2009, Yildiz 2010] dans l’objectif d’améliorer les résultats de convergence. Un aperçu des algorithmes a été réalisé [Moh’d Alia 2011].

Algorithm 2 *Harmony Search* : algorithme standard

Require: $HMS \in \mathbb{N}^+$, population Size
 $HMCR \in [0,1)$, Rate for choosing HM value
 $PAR \in [0,1)$, Rate for choosing neighbour value
Ensure: $s \in S$, S set of possible solutions
#INITIALIZATION#
 $HM \leftarrow \{s_0, s_1, \dots, s_{HMS-1}\}$
for stop_alg() \neq True **do**
#CREATE A NEW ELEMENT#
 $s_{new} = IMPROVISATION(HM)$
 $s_{worst} \leftarrow s_i \in HM | f(s_i) < f(s_j) \forall s_j \in HM, s_j \neq s_i$
#MEMORY UPDATE#
if $f(s_{new}) < f(s_{worst})$ **then**
REMOVE s_{worst} from HM
end if
end for
 $s \leftarrow s_i \in HM | f(s_i) > f(s_j) \forall s_j \in HM, s_j \neq s_i$
return s

Une proposition importante a été fournie par Madhavi [Mahdavi 2007] qui propose d'inclure un Pitch Adjusting Rate dynamique par l'équation :

$$PAR(gen) = PAR_{min} + \frac{(PAR_{max} - PAR_{min})}{NI} * gn \quad (4.2)$$

$$bw(gn) = bw_{max} \cdot e^{(c * gn)} \quad (4.3)$$

$$c = \frac{\ln\left(\frac{Bw_{min}}{Bw_{max}}\right)}{NI} \quad (4.4)$$

Où :

- PAR_{min}/PAR_{max} représentent les valeurs maximales et minimales du Pitch Adjusting Rate (PAR)
- NI est le nombre de solutions
- gn est le nombre de générations
- Bw_{min}/Bw_{max} représentent les valeurs maximales et minimales de la bande passante

Cette méthode permet de trouver une solution optimale plus rapidement,

car le paramètre PAR varie dynamiquement en fonction du nombre d'itérations. Wang et Juang ont proposé [Wang 2010] une autre méthode qui remplace et met à jour les paramètres PAR et bw (distance de la bande passante) en fonction de la valeur maximale et minimale dans HM . Les équations sont :

$$trial^i + [max(HM^i) - trial^i] \times ran[0, 1] \quad (4.5)$$

$$trial^i - [trial^i - min(HM^i)] \times ran[0, 1] \quad (4.6)$$

Où :

- $ran[0, 1] \Leftarrow$ nombre aléatoire $\in [0, 1)$
- $min(HM^i) \Leftarrow$ plus petite valeur de la i^{th} variable dans l'*Harmony Memory* (HM)
- $max(HM^i) \Leftarrow$ plus grande valeur de la i^{th} variable dans l'*Harmony Memory* (HM)
- $trial \Leftarrow$ *pitch* sélectionné dans l'*Harmony Memory* (HM)

Cette méthode de recherche est plus rapide, car les paramètres bw et PAR augmentent et diminuent en fonction des candidats extrema (meilleure et plus mauvaise valeur) de la population.

4.3.3 Pourquoi *Harmony Search* ?

L'algorithme *Harmony Search* à été sélectionné car :

- Il permet de répondre à des problématiques d'optimisations sur des sujet vastes [Wang 2015a] (systèmes de puissance, traitement d'image). Il n'est pas focalisé sur une problématique précise et est éclectique ce qui en fait un candidat polyvalent tel que le sont les problématiques industrielles de cette thèse (comment optimiser et automatiser le calibrage de caméras ? quelle est la meilleure pièce dans la bassine ?).

- C'est un algorithme évolutionnaire de type Pittsburgh, c'est-à-dire chaque individu est une solution. Il permet de contrôler simplement la recherche par le biais des paramètres : HMS qui définit la taille de la population, le paramètre HMCR qui définit le ratio entre exploration et exploitation, et par le paramètre PAR qui est un élément déterminant des stratégies de recherche par approche évolutionnaire.
- C'est un algorithme simple [Wang 2015b], car il ne possède pas de système de mutation, et n'a pas de reproduction à partir de plusieurs parents (Eng : crossover) ce qui lui permet une grande adaptabilité.

4.3.4 Un algorithme remis en cause

Weyland [Weyland 2012] a publié plusieurs articles remettant en cause l'algorithme, car celui-ci serait un cas particulier des algorithmes à stratégie d'évolution. Il publia aussi une critique sur la manière dont l'auteur [Geem 2007] utilise l'algorithme pour résoudre un sudoku [Weyland 2015]. En 2012, Padberg [Padberg 2012] prouva mathématiquement qu'*Harmony Search* est un cas particulier d'algorithme évolutionnaire pour une optimisation binaire. Depuis leur création, un grand nombre d'algorithmes évolutionnaires ont été proposés et beaucoup d'entre eux ont été sélectionnés pour leur aspect métaphorique [Sorensen 2015], qui permettent d'explorer d'autres stratégies d'évolution, et simple à comprendre.

4.4 Conclusion

Parmi l'ensemble des algorithmes évolutionnaires, *Harmony Search* a l'avantage d'être rapide pour converger, facile à mettre en place car il est simple et permet de trouver rapidement une solution, ce qui est important dans le domaine industriel. Les résultats obtenus par *Harmony Search* [Geem 2010a]

dans les différents domaines et problèmes d'optimisation font de lui un candidat efficient et robuste pour notre problématique d'optimisation de temps de recherche.

Deuxième partie

Partie 2 : Contributions

Al-Thocb HSC : Un Algorithme d'optimisation pour l'étalonnage de caméras

Sommaire

5.1	Introduction	66
5.2	Challenges	68
5.3	Le calibrage de caméras stéréoscopiques	70
5.4	Sélection d'images pour l'étalonnage	73
5.5	Méthodologie	76
5.6	Algorithmes d'étalonnage	77
5.7	Al-Thocb : Harmony Search Calibration	77
5.8	Partie expérimentale	80
5.9	Parallélisation	87
5.10	Discussion	89
5.11	Conclusions	90

" Il est aussi noble de tendre à l'équilibre qu'à la perfection ;
car c'est une perfection que de garder l'équilibre. "

Jean Grenier

5.1 Introduction

Ce chapitre est un pré-requis pour l'identification des pièces et donc du "Pick-And-Place" et présente un algorithme d'optimisation pour l'étalonnage de caméras. Comme représenté dans la Figure 5.1, le système a besoin d'être calibré avant d'être opérationnel. Ce chapitre se base sur l'article [Willaume 2017a]. Calibrer deux caméras permet de mesurer et de localiser des objets dans une scène tridimensionnelle.

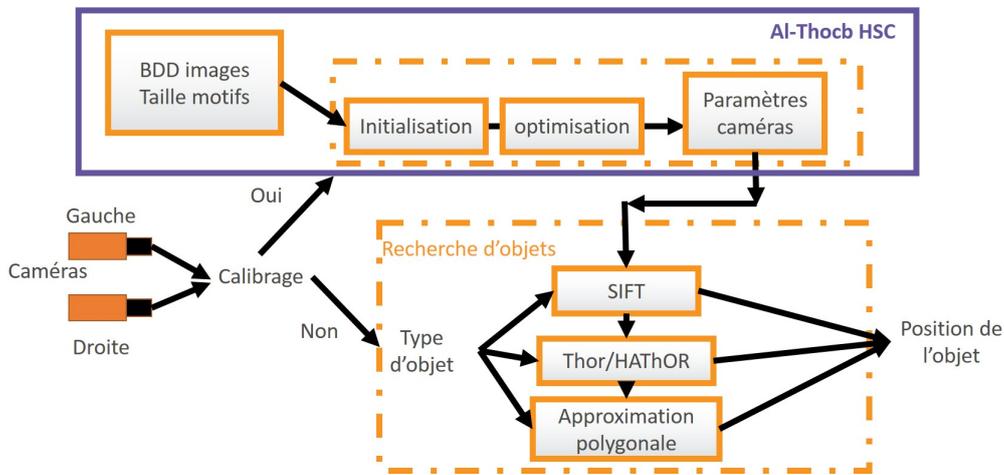


FIGURE 5.1 – Schéma fonctionnel du procédé de "Pick-And-Place"

Le calibrage revient à trouver la relation entre des pixels sélectionnés dans une image et les coordonnées spatiales de celui-ci. Le calibrage de caméra est appliqué dans de nombreux domaines comme la robotique [Murray 2000, Ayache 1991, Ni 2015], la détection d'obstacles pour les véhicules [Broggi 2005, Masaki 1998, Kim 2016] ou encore pour les applications industrielles [Aguilar 1996]. L'étalonnage dont l'objectif est d'établir une correspondance entre les coordonnées de l'image (pixels) et les coordonnées de l'objet (position de l'objet en fonction d'un point de référence) consiste à utiliser deux caméras pour prendre en photo plusieurs images d'un motif, appelé mire, qui permettent le calcul des paramètres internes (paramètres intrinsèques) et externes (paramètres extrinsèques) des caméras. Un étalonnage avec un trop petit nombre

d'images ne permet pas de récupérer des paramètres cohérents. En revanche, de trop nombreuses images impliquent un long temps de traitement. Dans le processus d'étalonnage standard, un opérateur doit créer une base de données d'images en prenant des photos avec les deux caméras. Cependant, cette tâche nécessite des connaissances spécialisées dans le traitement de l'image, ce qui n'est pas le cas dans l'industrie où les opérateurs sont souvent amenés à étalonner des caméras sans avoir nécessairement été formés. En outre, un étalonnage précis peut prendre beaucoup de temps et dépend de différents critères tels que l'expérience de l'opérateur, le nombre d'images pour effectuer le calibrage, la résolution des images... Un processus automatisé d'étalonnage nécessite un logiciel facile à configurer et facile à utiliser, tout en ayant une bonne convergence vers des paramètres intrinsèques et extrinsèques cohérents. Une solution a été proposée dans la littérature par Rupp [Rupp 2006] pour simplifier le processus d'étalonnage. Cette méthode utilise un algorithme Monte-Carlo amélioré en ajoutant une extension déterministe afin de déterminer automatiquement le meilleur sous-ensemble d'images qui évolue automatiquement si l'image récupérée est un calibrage optimal. Il introduit également une sélection des images d'entrée comme facteur d'amélioration. En outre, la comparaison de la fonction de coût avec un étalonnage voisin est gourmande en temps. Dans les contextes industriels, même si l'étalonnage est effectué en amont de la phase de production, la caméra doit être réglée rapidement pour éviter un temps d'arrêt prolongé de la ligne de production. Dans cette thèse, nous adaptons un algorithme évolutionnaire pour étalonner des caméras dans le but qu'un opérateur n'ait pas à s'inquiéter des données d'images d'entrée. L'algorithme d'étalonnage proposé est une adaptation de l'algorithme Harmony Search Optimization (HSO) [Geem 2001] présenté dans le chapitre 4. L'opérateur du processus d'étalonnage de la caméra doit simplement fournir une base de données d'images d'entrée cumulée à un ou plusieurs critères d'arrêt (seuil de coût, temps de traitement maximal). Le premier avantage

de la méthode proposée est l'automatisation d'un système de calibrage. Le deuxième avantage est l'approche par optimisation qui permet de récupérer des résultats pertinents dans un temps borné. Cette méthode permet d'optimiser le temps de recherche sans explorer toutes les solutions possibles. Le troisième avantage est la déclinaison d'un algorithme évolutionnaire dans le contexte du calibrage. Le quatrième avantage d'Al-Thocb est qu'il permet également d'aborder le problème d'étalonnage en amont en automatisant la sélection d'images plutôt que de modifier les paramètres d'étalonnage récupérés. À notre connaissance, il n'existe aucun autre processus automatisé d'étalonnage conforme aux contraintes de performance des environnements de production. Actuellement en industrie, les systèmes de calibrages se font sous-traiter ou nécessitent un personnel de maintenance avec des connaissances dans le domaine du traitement d'images. Ce chapitre est organisé comme suit. La Section 5.2 décrit le challenge du calibrage de caméras. La Section 5.3 introduit le domaine du calibrage. La Section 5.4 présente la sélection des images, une phase importante du processus d'étalonnage. La Section 5.5 présente la méthodologie utilisée. La Section 5.7 décrit la méthode d'étalonnage proposée Al-Thocb HSC. La Section 5.8 présente la partie expérimentale. La Section 5.10 définit les avantages et des inconvénients de la méthode proposée.

5.2 Challenges

L'automatisation d'un système d'étalonnage de caméra doit remplir un ensemble de critères de satisfaction :

- précision
- facilité de configuration
- performance

5.2.1 Précision

Ce critère détermine le comportement du système une fois qu'il est calibré. Si le résultat final renvoie une fonction de coût non cohérente, le système de "Pick-and-Place" ne sera pas efficace et le robot ne pourra pas effectuer correctement la saisie de la pièce. La précision est déterminée en fonction de l'erreur de reprojection qui est utilisée comme la fonction de coût du système. C'est l'erreur géométrique associée à la distance entre un point de référence et le point projeté. L'équation de l'erreur de reprojection est représentée sous la forme [Hartley 2003] :

$$\sum_i d(x_i, \hat{x}_i)^2 + d(x'_i, \hat{x}'_i)^2 \quad (5.1)$$

où $d(x_i, \hat{x}_i)$ correspond à la distance euclidienne entre les points de l'image représentés par les vecteurs x et \hat{x} .

5.2.2 Facilité de configuration

L'étape de configuration détermine les activités que l'opérateur doit effectuer sur la machine avant son fonctionnement. Ces actions doivent être limitées en termes de consommation du temps et de durée d'apprentissage. Un opérateur doit pouvoir prendre rapidement le système en main. L'étalonnage nécessite un certain nombre d'actions et de données d'entrée. Pour être efficace, un opérateur doit entrer uniquement des données facilement mises à disposition, telles que les images, la taille des motifs et le nombre de motifs sur les mires (voir 5.3.2). L'exigence principale pour une configuration facile réside dans un étalonnage qui doit être réalisé sans intervention humaine et qui doit fournir une solution optimale en fonction de la précision et de la rapidité requise par le système.

5.2.3 Performance

Sur la plupart des lignes de production, la qualité de l'algorithme influe sur le temps d'arrêt de la machine qui est lié au processus d'étalonnage. La principale exigence de performance est la suivante :

- l'algorithme doit converger rapidement vers une solution optimale sans explorer systématiquement tout l'espace de recherche. La machine doit s'arrêter si le résultat obtenu dépasse un seuil de satisfaction ou un temps de recherche trop long
- dans le cas contraire, l'application continue jusqu'à trouver une meilleure solution.

5.3 Le calibrage de caméras stéréoscopiques

5.3.1 Mécanisme d'étalonnage de caméra 3D

La construction d'une scène 3D implique le calcul d'un point de triangulation utilisant des points de correspondance entre les deux capteurs. Le point 3D est déterminé par l'intersection de deux lignes rétro projetées. Les points de correspondances proviennent du même point 3D de la scène. Il est nécessaire de trouver des pixels correspondants dans les deux images, plus précisément des pixels qui correspondent au même objet. Il faut ensuite calculer la triangulation du point 3D afin d'obtenir les coordonnées X , Y , Z de l'objet. Pour récupérer l'orientation et la position, les caméras doivent être modélisées ou calibrées au préalable. La modélisation des deux caméras s'effectue à l'aide de la géométrie épipolaire. C'est l'association des images dans chaque plan image pour le même point 3D, qui est nécessaire pour récupérer les coordonnées 3D d'un point dans l'image. Le calibrage est modélisé entre les coordonnées 2D des pixels et le point 3D du monde réel par la matrice de projection qui permet d'extraire les paramètres tels que la matrice de calibrage, la matrice de

rotation et la position de la caméra. Les paramètres idéaux [Guermeur 2003] d'un sténopé de caméra incluent des paramètres extrinsèques et intrinsèques. Les paramètres extrinsèques représentent l'emplacement et la rotation des caméras dans l'espace. Les paramètres intrinsèques représentent la focale et le centre optique des caméras.

5.3.2 Les mires

Plusieurs modèles de mires ont été proposés dans la littérature. Comme l'indique la Figure 5.2, un motif est proposé par [Tsai 1987] dans lequel les carrés sont représentés côte à côte avec deux couleurs distinctes. Une mire alternative est représentée dans la Figure 5.3 [Izaguirre 1983], qui représente des motifs circulaires. Dans les Figures 5.2 et 5.3, les deux couleurs utilisées sont le noir et le blanc. Cette distinction met en évidence les points d'intersection entre les motifs, généralement les carrés, avec une taille connue. Les points de référence sont alors les coins des carrés.

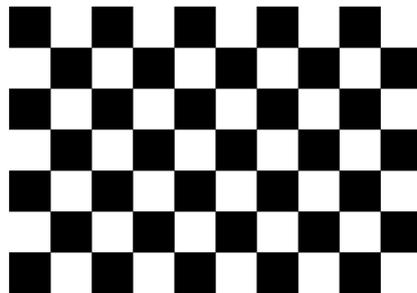


FIGURE 5.2 – Al-Thocb HSC : Mire avec des motifs carrés (damier)

Pour chacun des points du carré, l'algorithme d'étalonnage détecte les transitions verticales et horizontales à l'aide de l'algorithme de détection de coins proposé par Harris [Harris 1988]. La sortie est une liste de points appartenant à l'intersection carrée des deux images. Une régression linéaire est appliquée aux points afin d'obtenir les équations des droites. La distorsion géométrique de la lentille est très importante et limite la précision de détection. Cepen-

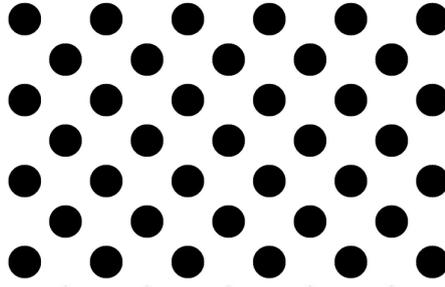


FIGURE 5.3 – Al-Thocb HSC : Mire avec des motifs ronds

dant, une réduction de la taille de ces motifs augmente le poids de l'erreur près des coins. Le modèle de damier avec les carrés est plus précis car les caractéristiques linéaires sont plus faciles à détecter à l'aide d'opérateur comme Canny ou Hough, ceci est dû à l'intersection des lignes. Dans le second cas, la plupart des détecteurs de points sont calculés avec la précision des pixels. Il correspond plus à nos attentes car il répond aux exigences critiques du nombre de points de référence, ainsi qu'à la détection locale proche du coin. C'est pourquoi nous avons choisi d'utiliser un modèle de type damier dans nos étalonnages. La précision de l'étalonnage a été mesurée à l'aide d'une fonction de coût appelée erreur de reprojection. Malheureusement, le modèle de reprojection n'est pas assez précis dans de nombreuses situations, particulièrement en raison de la distorsion. Il représente la déformation des images non linéaires et des imperfections dans les lentilles créées lors de la conception de celles-ci, mais aussi d'imperfections optiques et mécaniques (qualité du verre, courbure, etc.). Une autre problématique est l'identification de la mire à l'intérieur de l'image. Cette phase est la plus importante dans le processus d'étalonnage car elle détermine la qualité et le comportement du système. Si le système est mal calibré, une correction doit être effectuée en aval.

5.4 Sélection d'images pour l'étalonnage

La sélection d'un ensemble approprié d'images de mires est un processus important pour l'étalonnage de caméras. La précision de l'étalonnage dépend du nombre d'images utilisées et de leur qualité.

5.4.1 Nombre d'images : temps de traitement vs score d'étalonnage

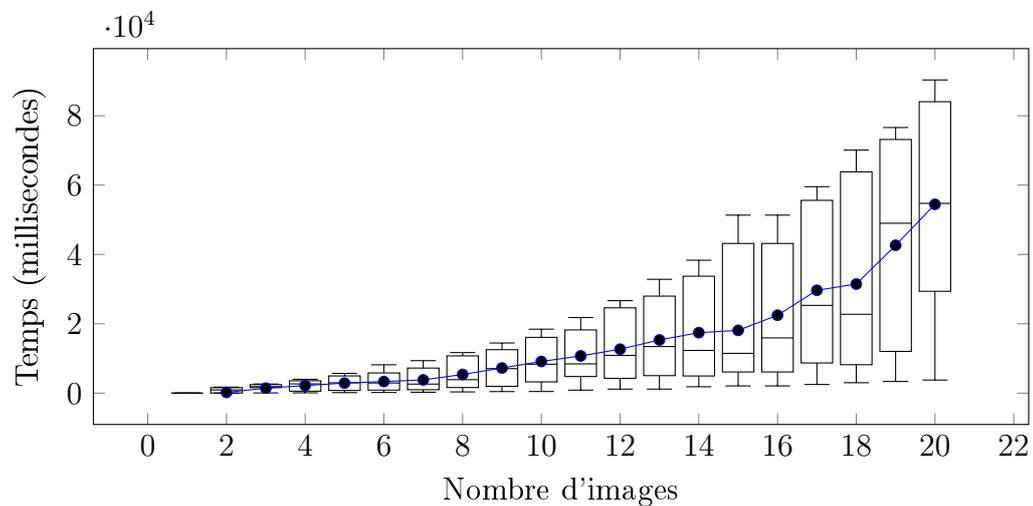


FIGURE 5.4 – Al-Thocb HSC : Temps de traitement des étalonnages aléatoires par rapport au nombre d'images (100 itérations); Ligne bleue : temps de traitement moyen

Le graphe de boîtes à moustache représentée dans la Figure 5.4 présente le temps requis pour calibrer deux caméras en fonction du nombre d'images fournie par l'opérateur en entrée. Chaque calibrage est effectué 100 fois avec un total de 21 images candidates, la moyenne des 100 mesures est utilisée comme valeur de performance. Nous pouvons remarquer que le temps de traitement dépend du nombre d'images. La Figure 5.4 représentant la courbe moyenne des performances met en avant une relation non-linéaire entre le nombre d'images et le temps de traitement. Comme l'indique la Figure 5.5, le nombre d'images

a également un impact sur le score d'étalonnage. Plus le nombre d'images est important, meilleur est le score. Néanmoins, le temps de traitement est un facteur crucial dans les lignes de production et un système d'étalonnage efficace doit donc trouver un équilibre entre le nombre d'images analysées, la précision et le temps de traitement requis.

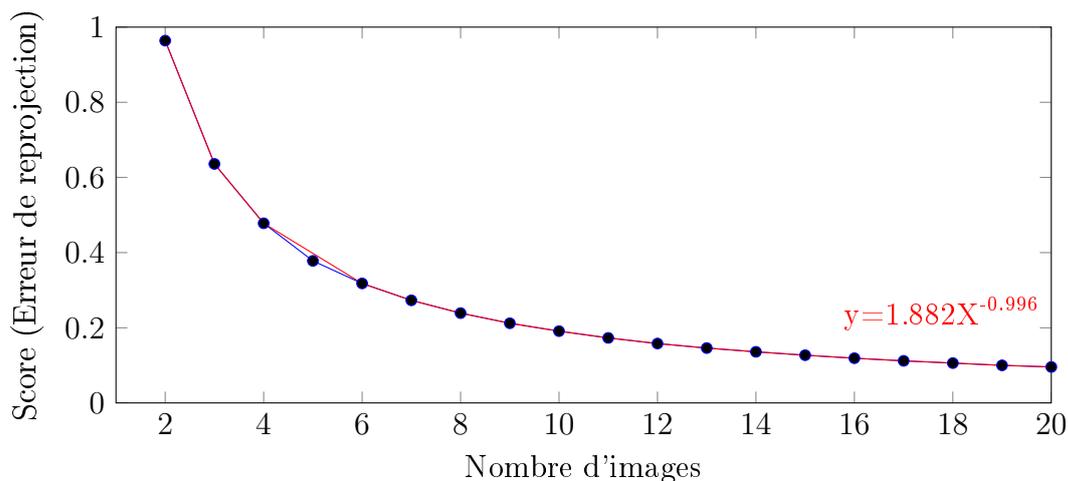


FIGURE 5.5 – Al-Thocb HSC : Score des étalonnages aléatoires par rapport au nombre d'images

5.4.2 Sélection des meilleures mires

Dans la Figure 5.6, la sélection des mires joue un rôle important dans le processus de calibrage. Les meilleures images récupérées pour l'étalonnage stéréoscopique sont marquées d'un V vert et les pires avec un X rouge sur l'image. Cependant, cette sélection est difficile pour un opérateur qui n'a pas les moyens et le temps de trouver les images pertinentes pour le système d'étalonnage.

Dans la Figure 5.7, lorsque les caméras sont étalonnées par des opérateurs humains, toutes les images d'entrée sont utilisées sans essayer d'optimiser la sélection d'images, car l'opérateur manque de temps pour les sélectionner. Actuellement, l'utilisateur donne un ensemble d'images comme entrée et les

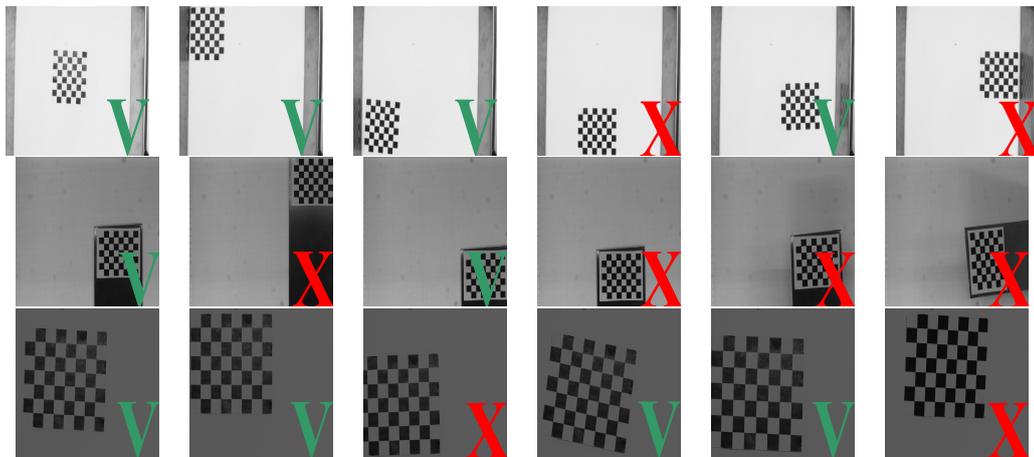


FIGURE 5.6 – Al-Thocb HSC : Exemple de sélection de mire pour 3 calibrages efficaces. 1^{ère} ligne, 2^{ème} ligne et 3^{ème} ligne. V : image efficace, X : image non efficace

données nécessaires pour étalonner les caméras (la taille du carré, et le nombre de points d'intersection entre les carrés sur les axes X, Y).

Dans ce chapitre, nous proposons d'améliorer cette phase en sélectionnant automatiquement les images afin de récupérer un étalonnage optimisé en termes de précision ou de réactivité.

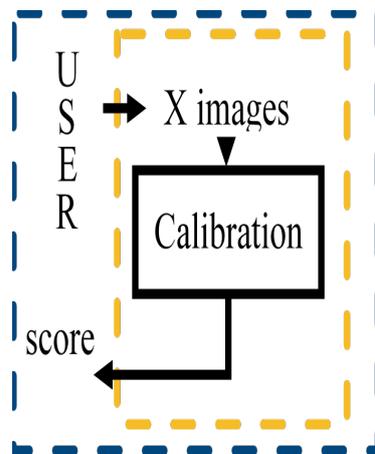


FIGURE 5.7 – Al-Thocb HSC : processus d'étalonnage pour l'utilisateur

5.5 Méthodologie

L'étalonnage de caméras est un domaine de recherche très dynamique, notamment en raison de l'évolution des lignes de production qui nécessitent des systèmes de vision de plus en plus précis. Ces systèmes ne sont pas adaptés aux lignes de production où les opérateurs n'ont pas le temps d'effectuer un étalonnage précis. Nous proposons l'algorithme Al-Thocb Harmony Search Calibration (HSC), basé sur un algorithme évolutionnaire, qui vise à simplifier la sélection de ces images. Comme l'indique la Figure 5.8, la méthode proposée optimise automatiquement la sélection des images d'entrée. Al-Thocb comprend deux phases : Une phase d'initialisation dans laquelle les images sont choisies aléatoirement dans la base de données. Une phase d'optimisation où les images sont soit sélectionnées parmi la population créée lors de la phase d'initialisation soit aléatoirement parmi les images de la base de données.

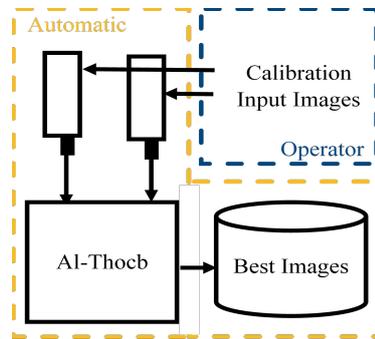


FIGURE 5.8 – Al-Thocb HSC : Méthodologie générale

Afin d'évaluer les capacités d'Al-Thocb, nous comparons dans la partie expérimentale les performances d'Al-Thocb HSC à d'autres algorithmes d'optimisation - un algorithme aléatoire amélioré et un algorithme de Monte-Carlo ainsi qu'aux performances réelles d'un opérateur humain. Nous définissons les différents paramètres nécessaires à la récupération d'un algorithme cohérent. Nous parallélisons également l'algorithme pour réduire le temps d'exécution et nous évaluons l'impact de cette parallélisation. Les avantages qualitatifs,

quantitatifs et les limites d'Al-Thocb sont évalués en fonction de la précision fournie par l'erreur de reprojection calculée à chaque génération de solutions.

5.6 Algorithmes d'étalonnage

De nombreuses méthodes ont été décrites dans la littérature pour étalonner des caméras. Par exemple, Heikkila [Heikkila 1997] propose une procédure d'étalonnage à quatre phases : L'algorithme commence par la transformation linéaire directe des coordonnées de l'objet aux coordonnées de l'image. Il continue ensuite par l'estimation des paramètres non linéaires. L'étape suivante est celle de la correction pour la projection asymétrique. La dernière étape est la résolution du problème de correction d'image. Sturm [Sturm 1999] propose un algorithme qui peut étalonner simultanément différentes vues d'une caméra avec des paramètres intrinsèques variables. Les projections de l'étalonnage planaire sur le plan de l'image sont calculées. Toutes contribuent à un système d'équations linéaires homogènes dans les paramètres intrinsèques, qui sont donc facilement déterminés. L'étalonnage peut être réalisé en résolvant des équations linéaires. L'algorithme d'étalonnage le plus connu est celui proposé par Zhang [Zhang 2000]. Cette méthode nécessite au moins deux images représentant un modèle d'étalonnage avec X points connus. Par exemple, et comme représentées dans la Figure 5.6, les caméras sont utilisées pour extraire la position des motifs des images à différents endroits et avec des inclinaisons différentes.

5.7 Al-Thocb : Harmony Search Calibration

La sélection des images est un dilemme important lors du calibrage d'un système. Nous proposons dans ce chapitre d'utiliser un algorithme évolutionnaire pour optimiser la sélection des images pour le calibrage. L'algorithme

effectue un ensemble de calibrages dans le but de trouver le paramétrage adapté pour l'utilisateur en fonction de ses critères de satisfaction (temps d'application de l'algorithme, qualité espérée des paramètres calculés). L'automatisation de la sélection qui s'effectue actuellement en majorité à l'aide d'un expert permet maintenant d'avoir un utilisateur moins expérimenté grâce à la simplification de la procédure. Cependant, cette méthode prendra plus de temps qu'un calibrage standard car le système doit effectuer plusieurs essais, afin de trouver le calibrage le plus cohérent possible. La Figure 5.9 représente le diagramme de fonctionnement d'Al-Thocb.

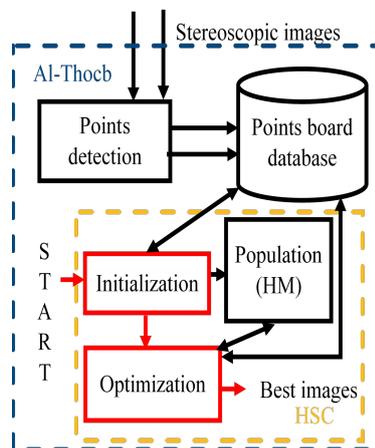


FIGURE 5.9 – Al-Thocb HSC : Diagramme

5.7.1 Etymologie d'Al-Thocb

Le mot Al-Thocb est la traduction de "le sténopé" en arabe. Cet appareil est une boîte dont l'une des faces est percée laissant ainsi passer la lumière. Le diamètre du trou est très faible (mm). L'image inversée se forme ainsi sous la surface opposée au trou. Ce nom est un hommage à Ibn al-Haytham, précurseur arabe en optique dans les années 1000. Il est le créateur de la chambre noire qui est un dérivé du fonctionnement du sténopé.

TABLE 5.1 – Ensemble de mires auquel appartient un ensemble de points : G = gauche, D = droite

Mire	Points
1	$P11(X_G, Y_G, X_D, Y_D), P12(X_G, Y_G, X_D, Y_D), P1m(X_G, Y_G, X_D, Y_D)$
2	$P21(X_G, Y_G, X_D, Y_D), P22(X_G, Y_G, X_D, Y_D), P2m(X_G, Y_G, X_D, Y_D)$
...	...
n	$Pn1(X_G, Y_G, X_D, Y_D), Pn2(X_G, Y_G, X_D, Y_D), Pnm(X_G, Y_G, X_D, Y_D)$

5.7.2 Sauvegarde des points de chaque mire

Avant d'effectuer l'algorithme de calibrage, il est nécessaire de récupérer les points des différentes mires prises par les caméras et données en entrée par l'opérateur. Cette étape de sauvegarde permet d'éviter une répétition de la détection de points sur les images à chaque calibrage car ceux-ci sont stockés dans une table. La Table 5.1 indique les points caractéristiques de l'ensemble des mires. La Table 5.2 illustre tous les points (coordonnées des pixels) des mires prises par les deux caméras. Ces points sont sauvegardés. L'exemple a été réalisé pour une paire de caméras stéréoscopiques où chaque point de la mire est récupéré de l'image des caméras droite et gauche. Ce procédé permet ainsi de gagner du temps sans avoir à répéter la recherche des points à chaque calibrage. Malheureusement, cette étape n'est pas naturelle car la plupart des calibrages sont effectués en une seule fois.

5.7.3 Algorithme Al-Thocb

L'algorithme 5 proposé est divisé en plusieurs étapes. En entrée, l'algorithme a besoin d'être paramétré et doit récupérer l'ensemble des points appartenant à chaque mire. La première étape appelée étape d'initialisation consiste à sélectionner aléatoirement les images fournies en entrée. Lors de la phase d'initialisation et donc d'exploration, le calibrage est effectué en prenant un nombre aléatoire d'images en entrée. La solution de la population est notée et sauvegardée dans l'Harmony Memory (*HM*). Nous avons utilisé l'erreur de

TABLE 5.2 – Ensemble des coordonnées des pixels pour une mire photographiée par deux caméras

P	X_G	Y_G	X_D	Y_D
1	426,3513	205,7727	545,2097	207,3947
2	424,7271	233,0224	543,7826	234,4729
3	423,5136	259,8424	542,668	261,1898
4	422,1038	286,4236	541,2261	287,7752
5	421,316	313,2817	540,3857	314,6167
6	419,8987	340,4086	539,1227	341,7628
7	418,5527	367,2238	537,7764	368,3676
8	399,0545	205,0854	518,1937	206,51
9	397,6282	231,9814	516,8386	233,3013
...
42	284,3722	361,0409	403,5888	362,3976

reprojection comme référence de qualité. C'est une valeur correspondant à la différence entre un point projeté et un point mesuré. Ce score permet d'évaluer la qualité du calibrage et correspond à la fonction de coût. La deuxième étape correspond à la partie optimisation qui est effectuée tant que le critère d'arrêt est insatisfaisant. Dans cette étape, les images sont sélectionnées soit aléatoirement soit dans la population de solutions candidates (Harmony Memory). Si l'image sélectionnée est issue de la population, une autre valeur détermine si le nombre d'images pour calibrer les caméras doit évoluer ou non. Si le nombre d'images doit évoluer, le nombre s'incrémente ou se décrémente en fonction du nombre sauvegardé d'images dans la population. Lorsque les images sont sélectionnées, une nouvelle solution est créée et testée. Si la nouvelle solution est meilleure que la plus mauvaise dans la population, alors elle est remplacée par la nouvelle.

5.8 Partie expérimentale

Nous avons évalué la capacité de l'algorithme Al-Thocb HSC à automatiser la sélection des images afin de calibrer les caméras. Les expériences ont été ef-

Algorithm 3 Al-Thocb HSC : Harmony Search Calibration Algorithm

Require: $HMS \in \mathbb{N}^+$, Number of elements in the population
 $HMCR \in [0,1)$, Rate to choose an image from the population
 $PAR \in [0,1)$, Rate to change the size of the number of images
 $I \leftarrow \{i_0, i_1, \dots, i_n\}$, Set of input images
 $HM \leftarrow \{s_0, \dots, s_{HMS}\}$, population of solutions
Calibration(), return boards and fitness function
 $Best() \leftarrow S_i \in HM | f(s_i) > f(S_j) \forall S_j \in HM, S_j \neq S_i$, best solution of HM
 $Worst() \leftarrow S_i \in HM | f(s_i) < f(S_j) \forall S_j \in HM, S_j \neq S_i$, worst solution of HM
stop_alg() \leftarrow Stop criteria, processing time || satisfying fitness function

Ensure: $S \leftarrow \{i_0, \dots, i_m\}$, best calibration

INITIALIZATION

for 0 to HMS **do**
 int $i = random(2, |I|)$
 for 0 to i **do**
 $S \leftarrow random(I - S)$
 end for
 $HM \leftarrow (Calibration(S))$
end for

IMPROVISATION

while stop_alg() \neq True **do**
 int $i = random(2, |I|)$
 for 0 to i **do**
 if $random[0, 1) \leq HMCR$ **then**
 int $x = random(0, |HM|)$
 $S_{new} \leftarrow random(HM[x] - S)$
 if $random[0, 1) \leq PAR$ **then**
 if $i > |HM[x]|$ **then**
 $i --$
 else
 $i ++$
 end if
 end if
 else
 $S \leftarrow random(I - S)$
 end if
 end for
 $HM \leftarrow (Calibration(S_{new}))$

MEMORY UPDATE

$HM = HM - Worst()$

end while

RETURN BEST SOLUTION

return $Best()$

fectuées sur un Intel (R) Core (TM) i7-5700HQ, CPU 2.70GHz. L'algorithme de Zhang [Zhang 2000] a été utilisé pour calculer le paramétrage des caméras. Il dépend directement des images sélectionnées par Al-Thocb. L'erreur de reprojection a été utilisée comme critère de satisfaction. C'est l'erreur géométrique associée à la distance entre un point de référence et le point projeté. Une erreur de reprojection plus faible implique un meilleur étalonnage. La Section 5.8.1 présente la phase d'initialisation nécessaire pour effectuer l'étalonnage. La Section 5.8.2 présente la précision obtenue par l'algorithme. La Section 5.8.3 présente les performances d'Al-Thocb. La Section 5.8.4 présente la durée appropriée de l'algorithme de traitement pour une ligne de production. La Section 5.8.5 décrit la phase de paramétrisation pour les variables *HMS*, *HMCR* et *PAR*. La Section 5.9 présente les améliorations de performance obtenues en parallélisant Al-Thocb HSC.

5.8.1 Mise en place

Avant d'exécuter l'algorithme, l'opérateur doit fournir :

- une base de données d'images
- la taille du damier ou la taille d'un carré ou la distance entre deux points si les motifs sont circulaires
- X , le nombre de points d'intersection entre les carrés horizontaux
- Y , le nombre de points d'intersection entre les carrés verticaux
- la valeur des paramètres *HMS*, *HMCR*, *PAR* (cf. chapitre 4.3.1)
- le nombre de calibrages à effectuer

En raison des facteurs externes de la machine comme la vibration dans les lignes de production, certaines machines doivent calibrer leurs caméras plusieurs fois par mois. La plupart des paramètres ne changent pas et ne doivent être définis qu'à la génération d'une nouvelle ligne de production. Exemple : *HMS* définit la taille de la mémoire, *PAR* détermine le pourcentage de sélec-

tion d'un voisin à l'élément sélectionné et $HMCR$ détermine le pourcentage de sélection d'un élément parmi la population actuelle ou parmi les données d'entrées.

5.8.2 Précision

La précision de l'algorithme a été comparée aux performances humaines et à celles deux autres méthodes :

- une méthode aléatoire optimisée
- une méthode de Monte-Carlo améliorée

Comme l'indiquent les résultats d'expériences données en Figure 5.10, l'algorithme Al-Thocb HSC converge plus rapidement vers une solution optimale que les autres méthodes.

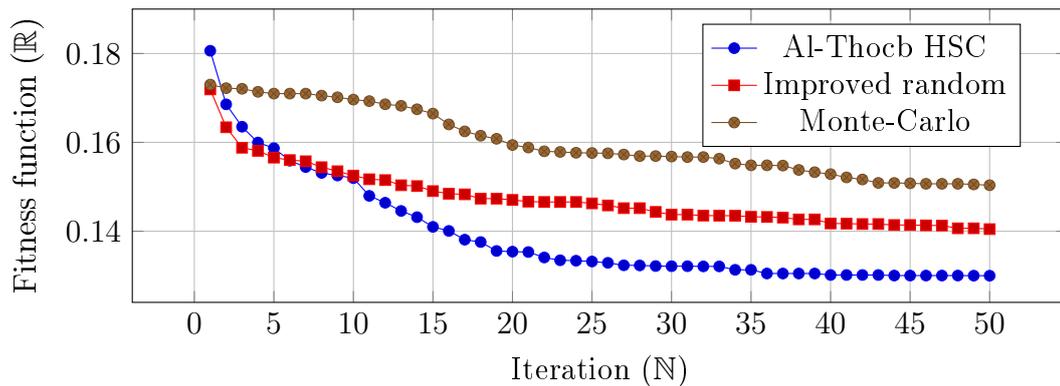


FIGURE 5.10 – Al-Thocb HSC : Fonction de coût pour 30 calibrages et 21 images. Une valeur faible implique un meilleur étalonnage

La Figure 5.11 indique les courbes de la fonction de coût pour les différentes méthodes, mettant en évidence une augmentation de performance de 20% par rapport à la méthode aléatoire améliorée et une augmentation de performance de 40% par rapport à la méthode de Monte-Carlo. Les différences sont dues à la phase d'optimisation. La Table 5.3 démontre une évaluation des performances effectuée par un humain sur 10 calibrages. Les sélections effectuées par l'opérateur sont insuffisamment cohérentes et les résultats obtenus

TABLE 5.3 – Scores et temps de calibrages effectués par un opérateur

Calibrage	Nombre d'image	Score	Temps
1	15	0.1822	01 : 29, 2
2	14	0.190	03 : 04, 4
3	17	0.1823	04 : 40, 4
4	21	0.140	07 : 31, 4
5	20	0.1466	09 : 25, 4
6	18	0.15	11 : 11, 3
7	18	0.1563	12 : 53, 9
8	19	0.1535	14 : 52, 6
9	19	0.1544	16 : 44, 4
10	20	0.1482	19 : 30, 2

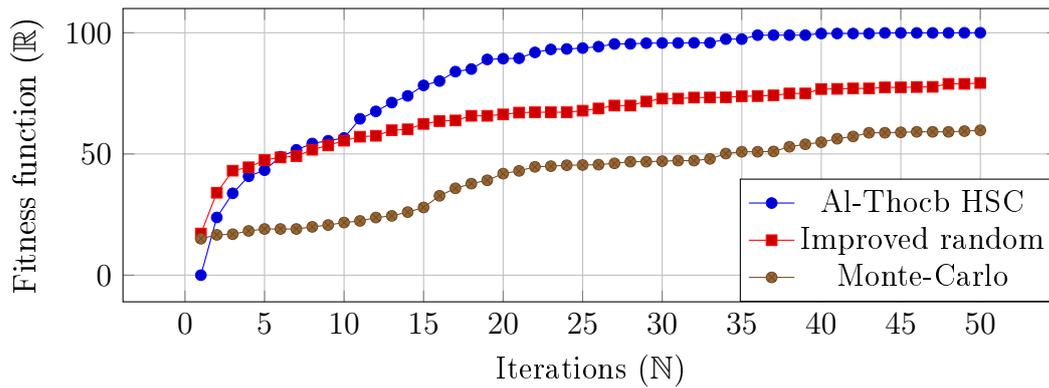


FIGURE 5.11 – Al-Thocb HSC : Taux d'amélioration pour 30 étalonnages et 21 images

présentent de grandes variations. Ceci s'explique par le fait que l'opérateur n'a pas de retour sur les étalonnages, contrairement à l'algorithme. Le meilleur calibrage effectué par l'opérateur obtient un score de 0.14. Il est supérieur et donc moins efficient que la performance d'Al-Thocb qui obtient un score 0.129. Le temps requis pour effectuer 10 étalonnages par l'opérateur est d'une vingtaine de minutes, ce temps est plus important que pour un algorithme automatisé qui prend moins de 3 minutes pour effectuer les 10 calibrages.

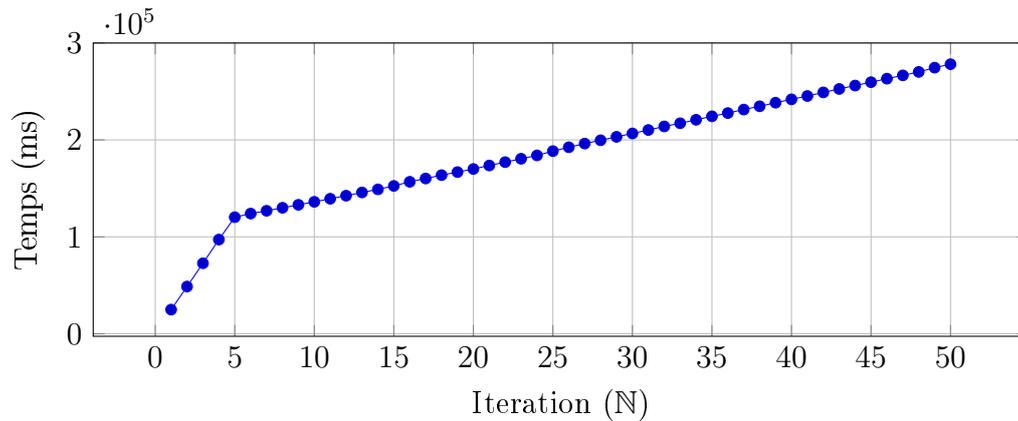


FIGURE 5.12 – Al-Thocb HSC : performance -> HMS=5, 200000 calibrages

5.8.3 Performance

La Figure 5.12 représente les résultats de l'algorithme Al-Thocb HSC qui prend 278 secondes, soit environ 5 minutes pour effectuer 50 étalonnages. Ce temps de traitement est acceptable par rapport au temps d'arrêt de la machine qui est généralement de quelques heures. De plus, le temps d'exécution peut être réglé par l'opérateur qui peut augmenter ou diminuer le nombre d'itérations que doit effectuer l'algorithme. Le temps d'exécution a un impact sur les performances de l'algorithme. Effectivement, il dépend directement du nombre d'étalonnages que l'algorithme peut effectuer dans le temps défini par l'opérateur.

5.8.4 Durée appropriée de l'algorithme de traitement pour une ligne de production

La Figure 5.12 donne le temps requis pour les phases combinées d'initialisation et d'optimisation de l'algorithme Al-Thocb. Une grande partie du temps de traitement est consacrée à la phase d'initialisation. Ceci est dû au paramètre de sélection du nombre d'images en entrée qui se stabilise seulement dans la partie optimisation. Néanmoins, la phase d'initialisation est essen-

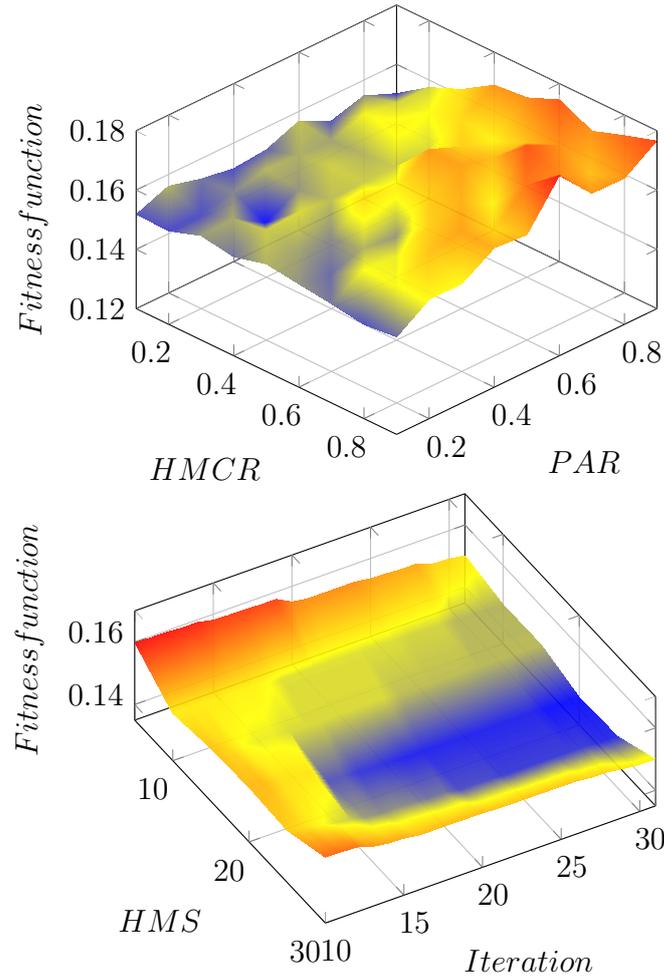


FIGURE 5.13 – Al-Thocb HSC : Fonction de coût en fonction des paramètres $HMCR$, PAR , HMS et le nombre d'itérations. Bleu : bonne fonction de coût. Rouge : Mauvaise fonction de coût.

tielle, car elle permet d'avoir une population solide et évolutive pour la phase d'optimisation. La courbe représente l'ensemble des calibrages effectués sur 50 itération avec les différents paramètres. L'algorithme a une complexité polynomiale, car le système utilise plusieurs briques imbriquées.

5.8.5 Paramétrage

Les Figures 5.13 représentent les résultats de performances d'un ensemble de calibrages en fonction des paramètres donnés à l'algorithme Al-Thocb. Elle

nous démontre l'importance de trouver les meilleurs paramètres pour obtenir des résultats cohérents. Les paramètres *HMS*, *HMCR* et *PAR* sont étudiés pour déterminer le meilleur comportement que peut avoir l'algorithme. Nous avons calculé le comportement de l'algorithme en fonction des paramètres *HMCR* et *PAR*. Le paramètre *PAR* détermine la variation du nombre d'images à sélectionner. Il doit être compris entre 0,1 et 0,3. Une valeur supérieure crée une dépendance sur la sélection du nombre d'images à sélectionner en fonction de la population HM. Le paramètre *HMCR* doit être compris entre 0,1 et 0,5. Une valeur plus élevée focalise la sélection des valeurs dans la population et limite ainsi l'exploration de nouvelles solutions. Une trop petite population empêche l'algorithme de sélectionner de bonnes images, car elle se concentre sur l'exploitation plutôt que sur l'exploration de l'espace de recherche. Nous avons testé le comportement de l'algorithme en fonction du nombre d'itérations et du paramètre *HMS*. Nous notons que le nombre d'itérations a un impact significatif sur la fonction de coût et donc sur la précision. La Figure 5.13 indique également que la taille de la population peut avoir un impact primordial sur les performances. Nous notons qu'un *HMS* optimal devrait être entre 15 et 25. Un *HMS* inférieur empêche Al-Thocb HSC de créer une population cohérente. Un *HMS* supérieur ne permet pas à l'algorithme d'effectuer l'optimisation puisque le processus d'initialisation requiert trop de temps.

5.9 Parallélisation

Enfin, l'algorithme Al-Thocb HSC a été parallélisé sur un ordinateur 4 cœurs dans le but de mesurer l'impact de la parallélisation sur le temps de recherche. Premièrement, comme représenté par les 3 courbes de la Figure 5.14, l'extraction des points sur les mires a été parallélisée. Chaque cœur représente une mire. En fonction de la taille de l'image, la parallélisation

permet un gain de temps considérable allant jusqu'à 70%. Celui-ci n'est pas de 75% car certaines actions doivent être effectuées de façon non parallélisée.

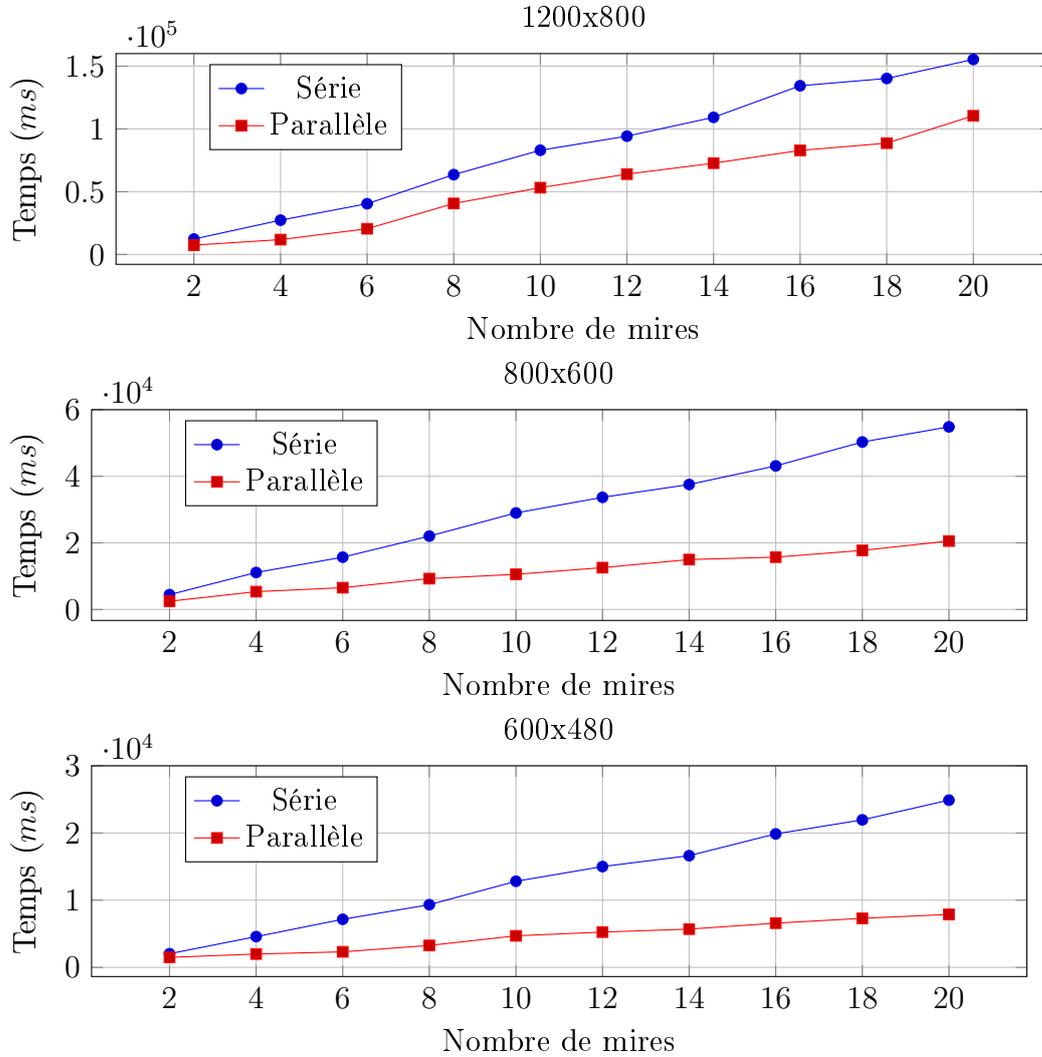


FIGURE 5.14 – Parallélisation d'Al-Thocb HSC : Temps d'extraction des points appartenant aux mires en fonction du nombre de mires pour différentes tailles d'images

Deuxièmement, et comme représenté dans la Figure 5.15, nous avons parallélisé l'algorithme Al-Thocb. Chaque cœur est associé à la création d'une nouvelle solution. Cette méthode réduit le temps de traitement d'environ 60%. Cette seconde phase de parallélisation est plus importante que la première car le calibrage est effectué tant que le critère d'arrêt n'est pas satisfait.

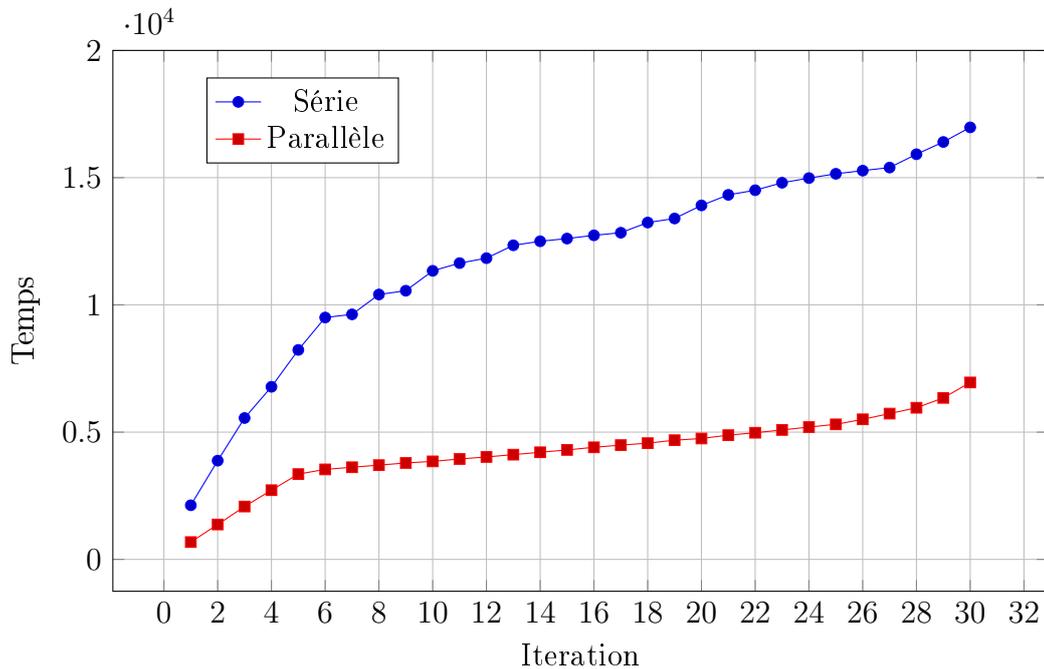


FIGURE 5.15 – Parallélisation d’Al-Thocb HSC : Temps de traitement en fonction du nombre d’itérations

5.10 Discussion

Les expériences décrites dans la partie précédente démontrent la capacité d’Al-Thocb à optimiser le calibrage de caméras stéréoscopiques pour un ensemble d’images en entrée. La sauvegarde des points sur la mire réduit le temps de recherche pour chaque calibrage. L’adaptation d’un algorithme évolutionnaire, en l’occurrence Harmony Search, permet de récupérer un étalonnage cohérent à sa phase d’exploration et d’exploitation.

Le paramétrage joue un rôle important dans l’optimisation de la recherche de bonnes images d’étalonnage. Des valeurs inadéquates pour les paramètres *HMS*, *PAR* et *HMCR* rendent l’algorithme inefficace. Une valeur inappropriée pour le nombre d’images d’entrée a un impact important sur le comportement du système d’étalonnage. Al-Thocb HSC permet une augmentation des performances d’environ 40% pour la méthode de Monte-Carlo et de 20% pour l’algorithme aléatoire amé-

lioré.

L'usabilité de l'algorithme est critique dans les situations industrielles.

L'avantage d'Al-Thocb est qu'après une paramétrisation générale, l'opérateur doit seulement fournir des informations simples, telles que la taille des carrés ou le nombre d'intersections entre eux.

L'efficacité du calibrage dépend du temps nécessaire à la machine pour être opérationnelle et dépend directement de la précision souhaitée. Il est nécessaire d'avoir un système rapide avec des temps inférieurs à la demi-heure. Al-Thocb répond ainsi parfaitement à la demande car il converge vers une solution optimale en quelques minutes grâce à la partie évolutionnaire de l'algorithme et à la parallélisation du code. La durée de l'algorithme dépend de la précision, mais elle est acceptable pour une utilisation de type "Pick-and-Place" dans une chaîne de production. La recherche prend beaucoup plus de temps avec un opérateur humain.

La Parallélisation permet une exécution beaucoup plus rapide de l'algorithme et peut être mise en place sur n'importe quel type de PC multicœurs, de plus en plus présent en bureautique.

5.11 Conclusions

Dans ce chapitre, nous avons proposé une méthode qui permet d'automatiser le calibrage de caméra stéréoscopique sans avoir de connaissances préalables dans la phase de sélection d'images. La méthode adapte un algorithme évolutionnaire appelé Harmony Search qui optimise et automatise la recherche d'images cohérentes à utiliser pour calibrer des caméras en fonction de l'erreur de reprojection récupérée pour chaque calibrage. Le système a besoin en entrée d'un ensemble d'images et de critères de satisfaction pour le calibrage des

caméras. Premièrement, chaque point présent sur l'échiquier de l'image est sauvegardé afin d'éviter de répéter le processus. Deuxièmement, le système initialise la population en sélectionnant aléatoirement les points appartenant aux mires sélectionnées. Pour chaque calibrage, un nombre est aléatoirement défini pour représenter le nombre de mires utilisées. L'erreur de reprojection est employée comme fonction de coût, elle détermine la qualité du calibrage. Troisièmement, une partie optimisation est effectuée en sélectionnant aléatoirement une mire à partir de la base de données d'entrée ou en sélectionnant une mire parmi la population créée pendant la phase d'initialisation. Si l'élément est choisi parmi la population, un autre critère détermine si le nombre de mires à utiliser pour effectuer le calibrage doit évoluer ou non. Si c'est le cas, le paramètre progresse en fonction du nombre d'éléments récupérés dans la population. L'algorithme a prouvé son efficacité de différentes manières. Premièrement, il converge rapidement vers une solution performante. Deuxièmement, l'algorithme a prouvé durant les essais réalisés son efficacité d'utilisation. Effectivement, l'algorithme a seulement besoin en entrée de l'ensemble d'images à utiliser et du critère d'arrêt du système (temps d'exécution de l'algorithme, erreur de reprojection à ne pas dépasser). Troisièmement, l'algorithme a prouvé son efficacité pour l'étape d'initialisation. Avec trois paramètres, les optimums peuvent être facilement récupérables en amont des calibrages. Lorsqu'ils sont connus, ils n'ont plus à être modifiés. Une comparaison avec d'autres méthodes a été réalisée dans le but de prouver l'efficacité de l'algorithme. Premièrement, l'algorithme est comparé aux performances de calibrages effectués par un opérateur. Cette comparaison prouve que l'algorithme peut aider l'opérateur à obtenir plus rapidement de meilleurs résultats. Deuxièmement, l'algorithme a été comparé avec une méthode aléatoire améliorée. L'algorithme sélectionne aléatoirement différentes mires en fonction des critères de satisfaction qui déterminent la chance de sélectionner une bonne ou une mauvaise image. Troisièmement, l'algorithme a été comparé avec une

méthode Monte-Carlo améliorée ou après un calibrage aléatoire. Le système sélectionne aléatoirement un calibrage voisin de celui effectué au préalable en ajoutant ou en supprimant à chaque fois une image. Dans le but de récupérer les meilleurs calibrages possible, l'algorithme a été testé pour chaque paramètre ($HMCR$, PAR) possible de l'algorithme Harmony Search ainsi que pour un grand nombre de tailles de population afin d'avoir un algorithme déjà paramétré.

ThOR : Détection d'objets fins dans une scène par squelettisation et mise en correspondance de graphes en 3D

Sommaire

6.1	Challenges	94
6.2	Méthode proposée	95
6.3	Expérimentation	104
6.4	Discussion	105
6.5	Conclusions	109



6.1 Challenges

Après avoir calibré les caméras, il est possible de localiser des objets dans une scène. Comme représenté dans la Figure 6.1, nous avons mis en place un ensemble d'algorithmes pour effectuer cette recherche. Cependant, la reconnaissance d'objets dans une bassine peut être une tâche difficile, notamment quand ceux-ci sont fins et sans motif.

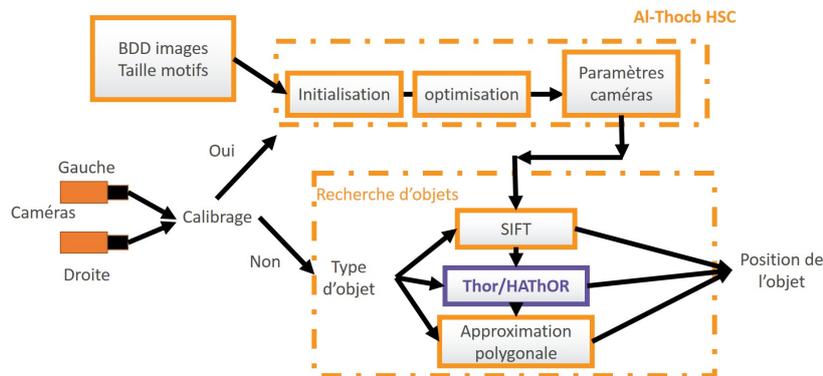


FIGURE 6.1 – Schéma fonctionnel du procédé de "Pick-And-Place"

La plupart des algorithmes sont centrés sur la reconnaissance de points clés dans l'image ou sur une approche polygonale en comparant la forme de l'objet. Ces méthodes sont très performantes pour reconnaître des objets à fort contraste ou avec de l'épaisseur. Cependant, leurs performances s'avèrent limitées pour la reconnaissance d'objets fins. L'objectif de ce chapitre est de proposer un algorithme simple d'utilisation dans le but de reconnaître des objets fins dans une scène tridimensionnelle à l'aide de deux caméras calibrées. L'algorithme doit être facilement utilisable afin de permettre à l'opérateur de se concentrer sur des tâches plus importantes. Pour lui, l'utilisation doit être focalisée sur l'apprentissage de l'objet dans son environnement. L'opérateur ne doit fournir qu'un ensemble de données facilement récupérables dans son environnement comme la taille de l'objet, les distances minimales et maximales auxquelles peuvent se situer les objets. Il est impératif d'éviter à l'opérateur

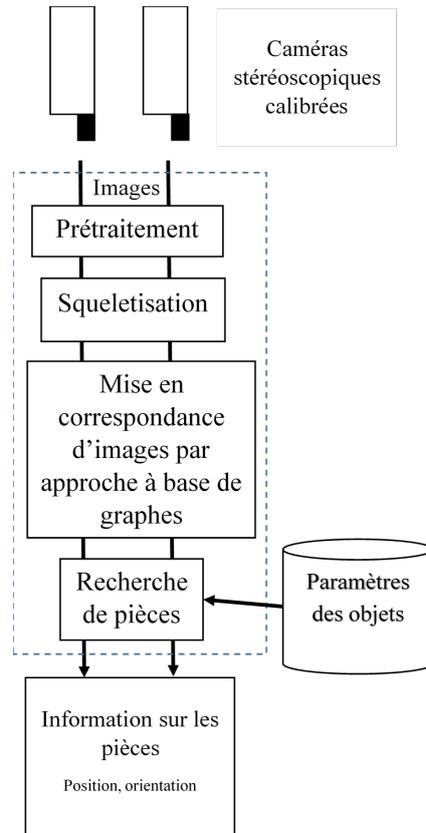


FIGURE 6.2 – ThOR : Schéma fonctionnel

de devoir programmer ou de renseigner des données difficiles à obtenir.

6.2 Méthode proposée

Dans cette partie, nous proposons l'approche ThOR (Eng : [THin Object Recognition by Graph Matching for Pick-and-Place Tasks](#)) basée sur la mise en correspondance de graphes 3D [Willaume 2017b]. L'objectif de l'algorithme est de détecter des objets fins sans spécificité de surface. Les graphes représentent les points de jonction et les points de terminaison appartenant aux squelettes des objets représentés sur les images stéréoscopiques. Ils sont utilisés pour récupérer la position et l'orientation des objets minces. Sur le schéma de la Figure 6.2, l'objectif est de récupérer les informations de localisation

des différents objets placés dans le champ de vision des caméras. Après le calibrage, une binarisation de l'image est appliquée pour sélectionner l'objet recherché. Ensuite, l'application d'un algorithme d'amincissement pour collecter toutes les courbes du squelette. Dans la prochaine étape, on effectue la corrélation entre les points de ces deux images, une mise en correspondance de graphes est appliquée. Les points de terminaison et les points de jonction des deux images sont récupérés. Ils représentent les nœuds des graphes. Une mise en correspondance entre les points de l'image de droite et celle de gauche est effectuée. La qualité de la synchronisation dépend de la distance entre les nœuds, de la cohérence de l'angle et de la profondeur récupérée à l'aide de la triangulation de ceux-ci. La dernière étape consiste à identifier les objets à l'aide de leurs tailles, les angles et les liens entre les nœuds du graphe. Le chapitre est organisé comme suit : Le paragraphe 6.2.1 présente l'ensemble des étapes du processus de l'algorithme Thor. Le paragraphe 6.2.2 décrit deux méthodes de prétraitement de l'image d'entrée. Le paragraphe 6.2.3 présente l'algorithme de squelettisation de l'image binarisé. Le paragraphe 6.2.4 définit le processus de mise en correspondance de graphe. La section 6.2.5 présente la méthode d'identification de l'objet à partir d'un modèle défini. Le paragraphe 6.3 explique l'expérimentation et expose les résultats obtenus. Le paragraphe 6.4 discute la méthode proposée.

6.2.1 Étapes du processus de détection de l'algorithme Thor

L'algorithme ThOR est divisé en 5 étapes. La première étape simplifie les images stéréoscopiques récupérées par les caméras dans le but de faire ressortir les objets à saisir. Cette simplification peut être appliquée en utilisant un algorithme de binarisation dans le but de faire ressortir tous les objets avec les mêmes caractéristiques de colorimétrie. Si la simplification nécessite de

séparer les objets les uns des autres, alors, l'algorithme de binarisation peut être remplacé par l'algorithme de ligne de partage des eaux qui permet ainsi de faire ressortir les objets selon leur taille, leurs couleurs... La simplification effectuée, un algorithme d'amincissement est appliqué en deuxième étape pour faire ressortir la courbe centrale des objets à saisir. Les points cohérents sur les deux images sont récupérés dans la troisième étape et mis en correspondance en fonction des paramètres de l'objet fourni par l'opérateur. L'objet est recherché dans la quatrième étape. Le calcul de la position et de l'orientation de l'objet peuvent être effectués dans la cinquième étape.

6.2.2 Prétraitement

Le prétraitement permet de simplifier l'image afin de mettre en avant les objets à saisir. Nous présentons dans cette partie deux solutions déployées. La première plus simpliste binarise l'image en noir et blanc. La deuxième plus complète utilise l'algorithme de ligne de partage des eaux pour mettre en avant des objets en fonction de leurs tailles, leurs couleurs...

6.2.2.1 Binarisation

Les algorithmes de binarisation permettent de convertir une image dont la valeur des pixels est supérieure à deux valeurs (par exemple : une image en couleurs possède pour chacun des pixels une valeur entre 0 et 255 pour les 3 couleurs Rouge-Vert-Bleu) en une image binaire. C'est-à-dire le blanc et le noir sont généralement utilisés pour transformer l'image. Afin d'éviter de comparer chaque pixel à un ensemble de couleurs, une image est souvent convertie au préalable en nuances de gris. Ceci permet ensuite une comparaison homogène des valeurs de gris souhaitées. La binarisation a pour objectif de supprimer l'élément de fond et de mettre en avant les objets souhaités. Un pixel est alors stocké dans un seul bit représenté par 'un' ou 'zéro'. Généralement, les

Chapitre 6. ThOR : Détection d'objets fins dans une scène par squelettisation et mise en correspondance de graphes en 3D

trois couleurs varient entre 0 et 255. Le choix du seuil dépend de la forme de l'histogramme (nombre de pixels par couleur). Comme représenté sur l'image de modèle 6.3a et sur les images binarisées 6.3b, la binarisation permet de mettre en avant l'objet. Une conversion de l'image RVB en niveaux de gris avec un seuil de 100 ($Color_{min}$) et 255 ($Color_{max}$) a été appliquée afin de faire ressortir l'objet. Si le pixel est en dehors de la plage, alors il a une valeur égale à 0 (noir). Inversement, le pixel est défini à une valeur égale 1 ou 255 (blanc). Cette méthode permet d'obtenir efficacement le squelette d'un objet avec l'objectif de simplifier l'image et récupérer uniquement des éléments indispensables pour rechercher les objets. Le choix du seuil dépend également du type d'objet à récupérer. Une binarisation va créer deux classes (présence ou non de l'objet).

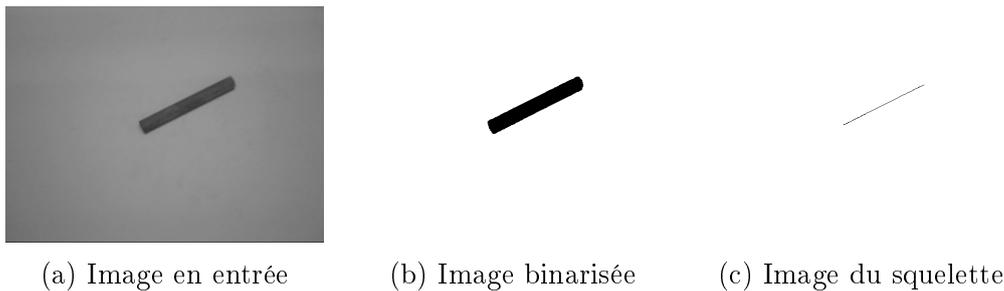


FIGURE 6.3 – ThOR : Simplification de l'image d'entrée afin de récupérer le squelette des objets recherchés (Binarisation : $Color_{min}$: 100 et $Color_{max}$: 255)

6.2.2.2 Ligne de partage des eaux

La méthode de ligne de partage des eaux (Eng : [Watershed](#)) proposée fin des années 70 [Beucher 1979], est un algorithme de segmentation simulant un procédé d'inondation. Les images sont divisées en surfaces dans lesquelles les pixels sont égaux. Les bassins sont séparés par des régions. Les régions sont détectées et labellisées. L'inondation est simulée avec une queue hiérarchique de type FIFO (premier entré - premier sorti). Chaque queue dépend de son niveau

de gris. Quand l'algorithme est fini, chaque pixel appartient à une région. Pour simplifier l'image, un étiquetage ([Eng : label](#)) est effectué pour transformer l'image en un ensemble d'éléments appelés objets où l'ensemble des pixels appartenant au même objet peuvent être récupérés indépendamment des autres pixels. Chaque objet est étiqueté avec différentes informations comme :

- l'identifiant : nom de l'objet
- la liste des pixels affiliée à l'objet
- le nombre de pixels
- la couleur moyenne de l'objet
- l'identifiant des objets à sa frontière

Cette méthode met en exergue les objets à saisir. En particulier, les éléments inutiles de la scène peuvent être supprimés dans le but de mettre l'objet en avant. La segmentation va créer plusieurs classes. Dans un cas idéal, une classe correspond à un objet à identifier.

6.2.3 Squelettisation

Les algorithmes de squelettisation ([Eng : Skeletonization](#)), encore appelés algorithmes d'amincissement ([Eng : Thinning Algorithm](#)) [[Gonzales 2002](#), [Cornea 2007](#)], sont des approches qui permettent de représenter la structure d'une région plane. L'algorithme réduit la forme de l'ensemble des courbes et se centre sur la forme originale afin d'obtenir le squelette de l'objet. Les topologies gardent leurs structures originales. Après avoir appliqué l'algorithme sur l'image [6.3a](#), les objets minces sont mis en évidence. Ils sont représentés sur la [Figure 6.3c](#). Zhang a proposé [[Zhang 1984](#)] une méthode rapide de squelettisation. Le procédé commence par retirer tous les points du bord de l'image excepté les points d'appartenance au squelette. Il préserve les connectivités du squelette. Les itérations sont divisées en 2 sous-itérations. La première supprime les points de la frontière sud-est et les coins situés dans le côté nord-

ouest de l'image, tandis que l'autre vise à supprimer les points de la frontière nord-ouest et les coins sud-est de l'image. L'application de cette méthode permet ainsi de répondre au challenge de simplification de l'image pour extraire les points importants du squelette.

6.2.4 Mise en correspondance d'images par approche à base de graphes pour détecter les objets

Les graphes sont des outils de représentation utilisés pour la reconnaissance de motif ([Eng : Pattern Recognition](#)) [[Conte 2004](#), [Biasotti 2008](#)]. Une image est transformée en un ou plusieurs graphes dans le but de représenter un ensemble de composants liés entre eux par un ensemble de relations. Plusieurs méthodes existent pour relier les graphes [[Cross 1997](#), [Bunke 1998](#), [Champin 2003](#)]. L'objectif est de créer un graphe de l'objet et de le mettre en correspondance avec un modèle. La mise en correspondance est divisée en deux catégories : la méthode d'appariement exacte et la méthode d'appariement approximée. Dans le premier cas, les objets et les graphes doivent correspondre parfaitement. Dans le deuxième cas, les objets et les graphes peuvent avoir une structure différente. Dans notre cas, nous souhaitons mettre en relation les images stéréoscopiques. Nous invalidons le calcul des critères de similarité, parce que ces graphes sont semblables. Chaque paire d'images stéréoscopiques est inscrite dans le même plan. L'objectif de l'approche est de réaliser la meilleure corrélation entre les nœuds pour faire ressortir l'objet à saisir. L'algorithme utilise les caractéristiques structurales du graphe. Nous le définissons comme 2 tuples $G = (V, E)$ où V est un ensemble V finit de sommets ([Eng : vertices](#)) $V = (v_1, v_2, v_3 \dots v_n)$ avec $n = |v|$. les sommets peuvent aussi être appelés nœuds ([Eng : nodes](#)). E est l'ensemble des arêtes ([Eng : Edges](#)) $E = e_1, e_2, e_3 \dots e_m$ avec $m = |E|$. Les arêtes peuvent aussi être appelés les arcs. Dans notre cas, les nœuds représentent les points extrêmes et

les points de connexions de l'image du squelette. Les arêtes sont déterminées en fonction des liens entre les nœuds. Ce chapitre définit l'approche par mise en correspondance de graphe (Eng : [graph matching](#)) pour définir le lien entre les nœuds. Une méthode de *backtracking* a été utilisée par [Brennecke 2004] et a démontré une grande efficacité pour mettre en correspondance la géométrie de modèle polygonal à l'aide des graphes représentés par le squelette de l'image. L'algorithme simplifie le squelette en retirant les nœuds insignifiants. Les nœuds sont choisis aléatoirement. Un processus de rétroaction est effectué afin de récupérer le graphe optimum commun entre les deux images, issue de la meilleure similitude mesurée entre les graphes. Montré dans l'algorithme 4, une image est qualifiée de référence quand tous les nœuds sont labellisés. Pour chaque nœud de l'image labellisée, la profondeur et la position sont prises en compte. Un nœud est sélectionné dans les deux images. Si les coordonnées 3D des deux nœuds mis en relation sont situées à l'intérieur d'une échelle définie par l'utilisateur, alors l'étiquette du premier nœud est ajoutée dans le deuxième nœud afin de l'identifier comme un candidat potentiel. Cette étape permet une pré-sélection. Un nœud peut avoir plusieurs candidats. Pour supprimer un nœud multi-candidats (ou plusieurs autres nœuds lui sont rattachés), deux nœuds étiquetés sont sélectionnés comme référents. Pour chaque nœud avec plusieurs candidats, une distance et un angle sont calculés entre les deux points sélectionnés et les candidats du nœuds. Dans la Figure 6.4a et 6.4b, le nœud B dans l'image de gauche a deux candidats dans la partie droite avec les possibilités 10 et 1.

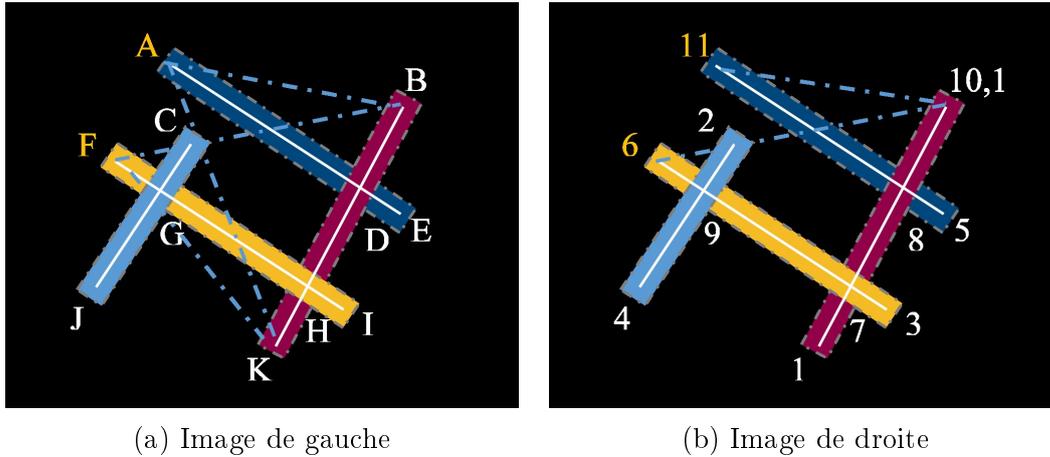


FIGURE 6.4 – ThOR : Mise en correspondance de graphes pour des images stéréoscopiques ($A : 11, B : 10, C : 2...$). Deux nœuds connus sont sélectionnés comme référents (exemple : $(F : 6, A : 11)$). L'angle et les distances entre les nœuds ABF et AKF sont déterminés pour savoir si B a comme correspondant 10 ou 1.

Deux points référents (" $A : 11$ " et " $F : 6$ ") sont sélectionnés. Dans l'image de droite, l'angle et les distances entre les nœuds sont déterminés entre les nœuds " 11 ", " $(10,1)$ " et " 6 ". Sur l'image de gauche, les nœuds " A " et " F " sont aussi traités en fonction de " B " et " K ". Les distances et l'angle entre " A, K, F " va être identique à " $11, 1, 6$ " et " A, B, F " va être identique à " $11, 10, 6$ ". Dans certain cas, le squelette de l'image n'est pas parfait et peut fournir certains nœuds inutiles. Pour éviter cette situation, une sélection du nœud est faite suivant la taille du segment. Une fonction de lissage permet de sortir les éléments déjà utilisés. Cette option est utilisée lorsqu'un nœud a été mis en correspondance. Elle parcourt toutes les possibilités et supprime le nœud déjà apparié. Toutefois, si un nœud de l'image de droite est mal assorti à celui de l'image de gauche, alors le graphe peut être erroné car le calcul des distances 2D et 3D donneront des résultats faux.

Algorithm 4 Graph matching**Require:** $(V(G1), V(G2))$: $G1$ and $G2$ graphs nodes set (D_{Min}, D_{Max}) : Depth interval (P_{Min}, P_{Max}) : Position interval**Ensure:** $(v.ID, w.ID)$: Graphs list labels**Integer** label = 0 ;**for all** node $v \in V(G1)$ **do**

label++ ;

v.ID.Add(label) ;

for all node $w \in V(G2)$ **do** $D \leftarrow D_calc(v, w)$; $P \leftarrow P_calc(v, w)$; **if** $D_{Min} < D < D_{Max}$ & $P_{Min} < P < P_{Max}$ **then** $w.ID.Add(label)$; **end if** **end for****end for****6.2.5 Recherche d'objets**

L'étape d'extraction des objets est appliquée par récupération des caractéristiques connues de l'objet. Premièrement, l'objectif est de lier les nœuds de l'image droite et de l'image gauche, qui appartiennent à un même objet. Deuxièmement, l'objet peut être trouvé à l'aide du calcul de la distance entre ses deux nœuds. La triangulation entre deux points des images stéréoscopiques permet de récupérer la position dans l'espace de l'objet. Pour chaque nœud dans l'image de droite correspondant au nœud de celle de gauche, une distance est calculée avec les autres nœuds. $G1$ et $G2$ sont les graphes représentant respectivement le graphe des images de gauche et de droite. $V(G1)$ et $V(G2)$ sont les ensembles de nœuds appartenant aux graphes $G1$ et $G2$. Les équations sont définies par :

$$3D = \sum_{i=1}^k \sum_{j=1}^l \sqrt{(Xi - Xj)^2 + (Yi - Yj)^2 + (Zi - Zj)^2} \quad (6.1)$$

$$2D = \sum_{i=1}^k \sum_{j=1}^l \sqrt{(X_i - X_j)^2 + (Y_i - Y_j)^2} \quad (6.2)$$

Où $k = |V(G1)|$ et $l = |V(G2)|$ sont les coordonnées 3D de chaque nœud. Pour chaque nœud $v \in V(G2)$, une distance est calculée entre v et chaque autre nœud du graphe. La distance peut être calculée à partir de coordonnées en 2D pour éviter que le système ne détecte des objets avec des orientations trop difficiles à attraper par le robot. Pour distinguer les objets avec plusieurs nœuds, les caractéristiques d'un nœud par rapport à un autre peuvent être calculés, comme le nombre de voisins, leur distance et leur angle afin de mettre en avant la spécificité de l'objet désiré.

6.3 Expérimentation

L'expérimentation appliquée sur un ensemble d'objets ayant chacun différentes orientations a pour but de localiser ces objets dans l'espace tridimensionnel. La partie expérimentale a été réalisée avec 2 types de capteurs. Deux webcams de 640*480px ont été utilisées dans une première expérimentation. Une deuxième expérimentation a été réalisée avec 2 caméras professionnelles Cognex IS-5603-10 de 800*600px avec une distance focale de 16 mm et une ouverture maximale du diaphragme de 1 : 1.4. Dans le premier cas, les deux caméras ont été placées à 400 mm de la scène. Dans le second cas, les caméras Cognex ont été placées à 700 mm de la scène. L'ordinateur utilisé est un Intel(R) Core(TM) i7-5700HQ CPU @ 2.70GHz, 2701 MHz. La calibration de l'ensemble des caméras a été réalisée à l'aide de la bibliothèque OpenCV (Algorithme Zhang), l'algorithme Al-Thocb ayant été finalisé dans une phase ultérieure de l'étude. Les images ont été obtenues sans ajout de lumière dans le but de prouver l'efficacité de l'algorithme en environnement réel. Différentes situations et plusieurs types d'objets ont été utilisés pour valider la méthode proposée. L'échelle de profondeur a été paramétrée entre 400 mm et 500 mm.

Elle dépend de la qualité de l'étalonnage. Comme représenté dans la Figure 6.5a, les objets ont été placés côte à côte avec différents angles afin de démontrer que l'algorithme peut trouver des objets dans des cas simples.

En second lieu et comme l'indique la Figure 6.5b, le programme a été testé avec un chevauchement des objets.

Cette situation prouve que l'algorithme peut identifier et labéliser chaque objet et donc les dissocier. Dans la Figure 6.5c, l'algorithme a été testé sur une image avec des chevauchements et une combinaison de différents types d'objets. Les objets ont été sélectionnés pour leurs couleurs et leurs formes. L'algorithme récupère correctement les composants, placés dans différentes positions. Les résultats présentés en Figure 6.5c indiquent que l'algorithme est capable de contourner le problème de deux objets clairs emmêlés (supérieure droite de l'image). Le temps de calcul présenté dans la Figure 6.7 indique que la durée de mise en correspondance entre le nœud de l'image de droite et le nœud correspondant de l'image de gauche dépend du nombre de nœuds. La Figure 6.6b permet de mettre en avant l'efficacité de l'algorithme pour trouver un objet avec transparence. Les Figures 6.6a et 6.6c mettent en avant l'efficacité de l'algorithme pour trouver des objets épais comme l'algorithme SIFT.

6.4 Discussion

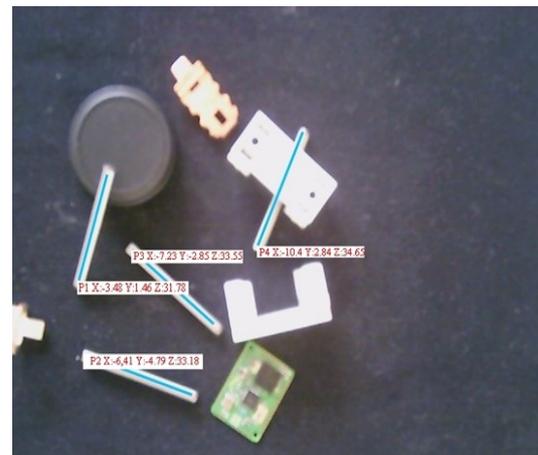
La partie expérimentale prouve que notre méthode est apte à trouver des objets épais et fins dans des positions aléatoires ainsi que des objets enchevêtrés. Le besoin est d'avoir un algorithme simple d'utilisation et performant pour la recherche d'objets fins. L'utilisation d'un algorithme d'amincissement permet de mettre en exergue l'objet à rechercher. La mise en correspondance de graphes entre les nœuds de l'image de droite et celle de gauche est apte à effectuer la récupération de l'objet. Les données transmises par l'opérateur



(a) ThOR : Recherche d'objets positionnés les uns à côté des autres.



(b) ThOR : Recherche d'objets positionnés avec un enchevêtrement.



(c) ThOR : Recherche d'un objet parmi un ensemble enchevêtré positionné aléatoirement.

FIGURE 6.5 – ThOR : Recherche d'objets fins. Segment : Lien entre deux nœuds faisant ressortir les objets. Texte : Position et angle de l'objet en fonction d'un repère.

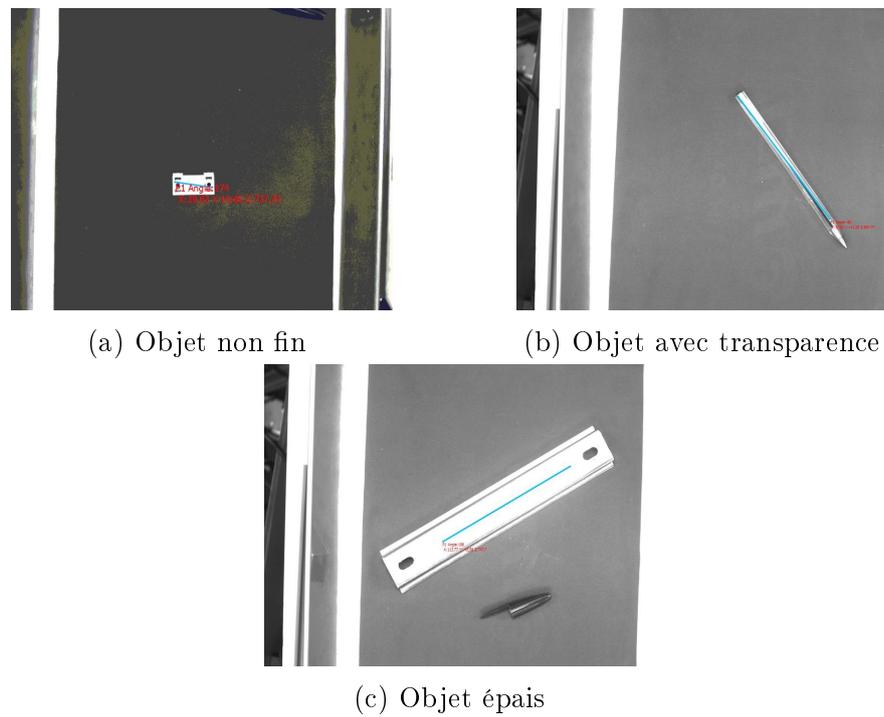


FIGURE 6.6 – ThOR : Recherche d'objets spécifiques

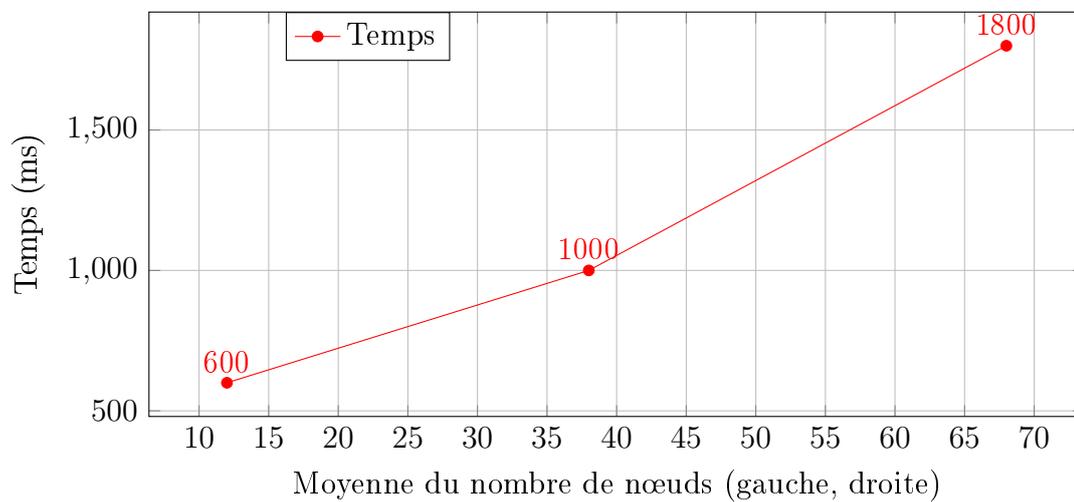


FIGURE 6.7 – ThOR : Représentation des temps de calcul de la mise en correspondance des nœuds sur une image de 640x480px. Moyenne du nombre de nœuds sur les images droites et gauches en fonction du temps en milliseconde : 600 = Pas d'enchèvement ; 1000 = Enchevêtrement ; 1800 = Différents objets positionnés aléatoirement

Image	A	B	C	D	E
Image					
Nombre d'objets	4	5	1	2	1
Nombre de candidats	4	5	1	1	1
Nombre de nœuds	8	11	313	283	85
Objet					
Fin	O	O	O	N	N
Position aléatoire	O	O	O	O	O
Enchevêtrement	N	O	N	N	N
Transparence	N	N	O	N	N
Seuil de Binarisation					
Min	186	186	170	185	170
Max	255	255	255	255	255
Score					
SIFT (%)	50	54	60	92	96
SURF (%)	49	56	74	59	71
Graph (%)	100	100	100	100	100

TABLE 6.1 – Performance de l'algorithme ThOR. O = Oui, N = Non

sont la distance entre les segments et les liens entre eux. Les résultats ont démontré l'efficacité de Thor de plusieurs points de vue. Premièrement, tous les objets ont été identifiés dans différentes situations (orientation, inclinaison, enchevêtrement, couleurs). Deuxièmement, le pré-traitement permet de récupérer l'objet grâce à ces caractéristiques.

La Table 6.1 indique les performances des différents algorithmes. Nous pouvons remarquer que ThOR a détecté l'ensemble des objets en fonction des

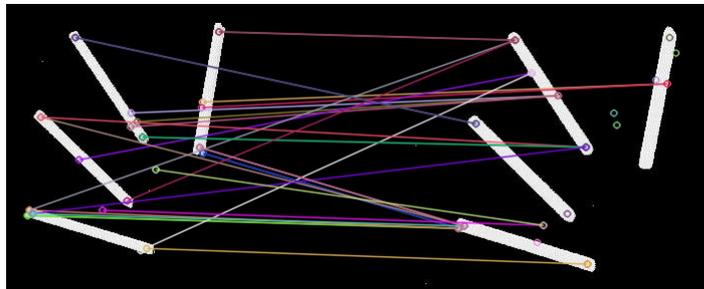


FIGURE 6.8 – Recherche des objets fins par l'algorithme SIFT (*Scale Invariant Feature Transform*). Temps d'exécution : 271ms

critères d'épaisseur de l'objet, de position de l'objet, d'enchevêtrement des objets entre-eux et de la transparence que peut avoir certains objets. L'algorithme a été comparé avec des algorithmes d'extraction de points d'intérêt. Nous pouvons remarquer que plus l'image est compliquée ou plus le nombre d'objets est important, alors plus le nombre de nœuds est important. Ceci est dû au pré-traitement qui a uniquement été réalisé par l'algorithme de binarisation. La représentation de l'algorithme SIFT est montrée sur la Figure 6.8. Nous pouvons remarquer que la mise en correspondance entre les deux images est difficile. Les scores de SIFT et SURF présentés dans la Table 6.1 ont été calculés en fonction du pourcentage du nombre de segments correctement mis en correspondance entre les deux images. Plusieurs points distinctifs ont été trouvés. Cependant, plusieurs mises en correspondance erronées sont observées. Les résultats erronés des algorithmes à détection de points d'intérêt sont dus à la complexité à corréliser des points clés issus d'objets trop fins, uniformes et sans spécificité. La méthode proposée est plus efficace que SIFT et SURF car les objets ont tous été trouvés. La durée d'exécution de l'algorithme proposé est correcte mais reste supérieure à la contrainte de 2 secondes. Cependant, le temps de simplification de l'image prend plusieurs centaines de millisecondes et doit être ajouté au temps de mise en correspondance des graphes qui dépasse la seconde. Ces temps sont trop importants pour être appliqué dans le domaine industriel. L'algorithme a une complexité polynomiale, car le système utilise plusieurs briques imbriquées. Ils peuvent être améliorés par l'utilisation d'algorithmes évolutionnaires permettant d'incrémenter une mise en correspondance de graphes optimisée.

6.5 Conclusions

Dans ce chapitre, nous avons proposé une approche pour reconnaître des objets fins en stéréovision pour effectuer des tâches d'aménages. Après la sim-

Chapitre 6. ThOR : Détection d'objets fins dans une scène par 110 squelettisation et mise en correspondance de graphes en 3D

plification de l'image pour en faire ressortir les objets à saisir, un squelette est construit pour chaque objet. Une récupération de chaque point de jonction et point extrêmes est effectuée pour résoudre la mise en correspondance de graphes en supprimant les points où la position et la profondeur ne coïncident pas. Les autres points sont identifiés grâce à l'analogie basée sur l'angle et la distance entre les points des images de droite et de gauche. Le procédé a été testé avec des objets placés aléatoirement, enchevêtrés les uns avec les autres. Tous les objets ont été trouvés. Les avantages de cette méthode sont en premier lieu la simplicité d'initialisation car les paramètres (forme, dimension, distance) sont connus par l'opérateur. De plus, l'algorithme est insensible aux problèmes de distorsion car il met en correspondance des données tridimensionnelles et n'est donc pas soumis à un seul plan. Le temps de détection supérieur à deux secondes est perfectible grâce à l'utilisation d'un algorithme évolutionnaire.

HAThOR : Méthodologie d'optimisation de la détection

Sommaire

7.1	Challenge	112
7.2	Critères de validation	113
7.3	<i>Harmony Search</i> pour la recherche d'objets	113
7.4	Expérimentation	116
7.5	Paramétrage	128
7.6	Discussion	128
7.7	Conclusions	130

“ Les espèces qui survivent ne sont pas les espèces les plus fortes, ni les plus intelligentes, mais celles qui s'adaptent le mieux aux changements. ”

Charles Darwin

7.1 Challenge

Comme expliqué dans le précédent chapitre, le temps de calcul est un facteur important dans l’industrie où la cadence de production doit être la moins perturbée. Ce chapitre se base sur la publication effectuée par Willaume [Willaume 2016]. Dans le chapitre précédent, les temps de recherche d’un objet sont supérieurs au standard de fabrication d’une entreprise. Pour un cas d’objets enchevêtrés, le prétraitement ajouté à la mise en correspondance de graphes dépassent les 2 secondes. Dans ce chapitre, nous essayons de résoudre cette problématique en optimisant la mise en correspondance de graphes. L’algorithme proposé ne doit pas réduire la simplicité d’utilisation. Le challenge est de trouver un algorithme qui soit facile à utiliser et performant. Nous proposons dans ce chapitre d’utiliser un algorithme évolutionnaire pour la mise en correspondance des graphes. L’objectif est d’adapter l’algorithme évolutionnaire pour effectuer une recherche d’objets dans un temps inférieur à 2 secondes. Nous proposons une approche appelée HAThOR (Eng : [Harmony Algorithm for THin Object Recognition](#)) basée sur l’optimisation de la mise en correspondance de graphes en 3 dimensions en utilisant un algorithme évolutionnaire, *Harmony Search*, pour trouver un objet placé aléatoirement dans un ensemble. Le nom HAThOR fait référence à la déesse Égyptienne de la musique, en rapport avec *Harmony Search* qui utilise la métaphore musicale pour définir une stratégie d’exploration originale, performante et simple d’utilisation. Les graphes sont extraits à partir du squelette de l’objet issu des images stéréoscopiques de la scène et de la mise en correspondance des caractéristiques du modèle objet. Ils sont utilisés pour récupérer la position d’un objet donné en 3 dimensions. La phase de recherche d’objet est optimisée à l’aide d’un algorithme évolutionnaire qui a pour objectif de réduire la durée de mise en correspondance des graphes. Cette méthode a pour but de fournir une solution optimale pour la recherche d’objets sans mettre en correspondance tous

les nœuds. Les caméras calibrées sont prises en compte [Zhang 2000]. Nous définissons dans cette partie l’algorithme de recherche d’objet. Ce chapitre a permis la publication d’un article dans une conférence à Athènes (Grèce) [Willaume 2016].

7.2 Critères de validation

L’algorithme proposé doit répondre à deux objectifs. Premièrement, l’algorithme proposé doit répondre aux exigences quantitatives. Le temps d’exécution de l’algorithme doit répondre aux exigences de l’entreprise. Les algorithmes de prétraitement et l’algorithme de mise en relation des graphes doivent être inférieurs à deux secondes. Deuxièmement, l’algorithme doit répondre aux exigences qualitatives. Il doit être capable de couvrir les cas réalistes de détection d’objets fins, non fins, avec ou sans motifs et avec ou sans transparence.

7.3 *Harmony Search* pour la recherche d’objets

L’algorithme d’optimisation est effectué après le prétraitement de l’image. Il permet de trouver la mise en correspondance des graphes des images stéréoscopiques. Le schéma fonctionnel en Figure 7.1 met en évidence l’application de l’algorithme évolutionnaire pour la mise en correspondance de graphes. *Harmony Search* est appliqué pour calculer la fonction de coût dans le but de trouver l’objet en mettant en correspondance les nœuds et en calculant la distance entre eux.

La fonction de coût est calculée en considérant les variables de décisions suivantes :

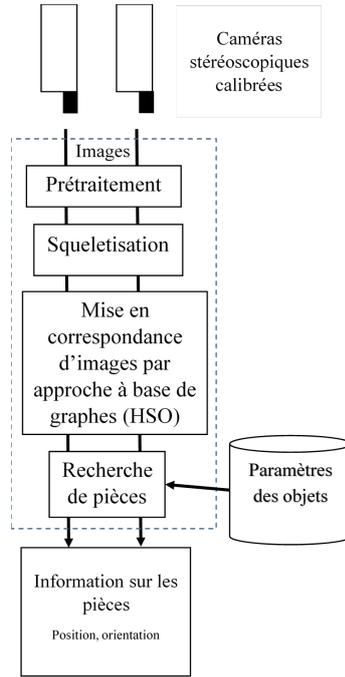


FIGURE 7.1 – HAThOR : schéma fonctionnel

— Les équations de la distance euclidienne en 2D et 3D :

$$3DS = \sqrt{(X_i - X_j)^2 + (Y_i - Y_j)^2 + (Z_i - Z_j)^2} \quad (7.1)$$

$$2DS = \sqrt{(X_i - X_j)^2 + (Y_i - Y_j)^2} \quad (7.2)$$

— Le pourcentage de pixels de $3DS$ appartenant à la partie binarisée ou labéllisée de l'image

L'algorithme 5 présente les différentes étapes à suivre pour optimiser la recherche d'objets. Il est nécessaire en entrée d'initialiser au moins une fois les paramètres HMS (*Harmony Memory Size*), $HMCR$ (*Harmony Memory Considering Rate*), PAR (*Pitch Adjusting Rate*). L'opérateur doit ensuite entrer la ou les tailles d'objets potentielles. Cette taille doit correspondre à des distances entre différents nœuds à mettre en correspondance. Plus simplement, elle correspond à la taille de l'objet. Celle-ci va être comparée aux solutions

créées tout au long de l'algorithme. La fonction de coût est calculée comme la différence entre la distance en 2D et 3D de la taille réelle de l'objet et de la distance calculée entre les nœuds du graphe. Pour la calculer, le système a besoin d'au minimum 4 nœuds (2 pour chaque image). Par exemple, pour détecter un stylo, il est nécessaire de trouver les deux embouts du stylo dans les deux images et de mettre en relation le nœud du capuchon et l'arrière du stylo. Si les caméras sont l'une à côté de l'autre, les nœuds à mettre en correspondance dans les deux images peuvent être prétraités en regardant la position du nœud dans l'image. Si nous prenons l'exemple du stylo, la position du nœud appartenant au capuchon sera différente (plus à droite ou à gauche, ou plus haut ou plus bas) que le nœud appartenant à l'arrière du stylo. La mise en correspondance de deux paires de nœuds va permettre de calculer la taille de l'objet. Plus la fonction de fitness est faible, plus les nœuds mis en correspondance représente l'objet à sélectionner.

HAThOR va commencer par l'étape d'initialisation. Elle est réalisée en fonction du paramètre *HMS*. Ce paramètre détermine la taille de la population (HM) et donc le nombre d'itérations à effectuer. Une itération correspond à une solution. Pour chaque solution, un nombre de nœuds est sélectionné aléatoirement dans l'image de droite et celle de gauche. La fonction de fitness va être calculée et sauvegardée dans la population (HM).

Après l'initialisation et tant que le critère d'arrêt n'est pas satisfaisant, une nouvelle solution est créée. La sélection des nœuds pour créer la solution dépend des paramètres *HMCR* et *PAR*. La mise en correspondance de deux nœuds est faite soit en sélectionnant les deux nœuds aléatoirement, ou en sélectionnant les nœuds depuis la population *HM* créée lors de l'initialisation. La variable *PAR* permet d'interchanger les nœuds déjà mis en correspondance dans *HM* par ceux sélectionnés afin de diminuer la partie optimisation. À chaque itération, une nouvelle solution est créée. Les nœuds mis en correspondance entre l'image de droite et celle de gauche créent une nouvelle solution.

Si elle est meilleure que la plus faible solution appartenant à la population HM , alors elle la remplace. Ceci permet de faire évoluer la population. Le critère d'arrêt correspond soit à un objet trouvé ou à un temps supérieur ou égal au temps souhaité pour la recherche de l'objet.

7.4 Expérimentation

L'objectif de l'expérimentation est de valider les performances de l'approche optimisée. L'expérimentation a été réalisée avec un Intel(R) Core(TM) i7-5700HQ, CPU 2.70GHz. L'algorithme AI-Thocb a été utilisé pour calibrer les caméras. Plusieurs cas ont été envisagés dans cette étude : les indicateurs de performances de l'algorithme sont la précision, les performances et la paramétrisation.

7.4.1 Performance

7.4.1.1 Temps de prétraitement

Le temps de prétraitement correspond au temps requis pour l'algorithme pour simplifier l'image. Comme l'indique la Figure 7.2, la durée dépend de la taille de l'image. Il est nécessaire de réduire l'image pour identifier l'objet dans un temps cohérent. L'algorithme peut aussi être appliqué à une partie (zone) de l'image afin de réduire les temps de traitement. Comme l'indique la Figure 7.3, l'une des étapes les plus longues est l'étiquetage de l'objet. Cette phase n'est pas nécessaire si les squelettes sont facilement formés, cependant elle peut être utile pour séparer les objets.

7.4.1.2 Performances de la mise en correspondance de graphes

Le temps de recherche dépend du nombre de nœuds requis pour représenter les objets ainsi que les paramètres donnés (HMS , PAR , $HMCR$) en

Algorithm 5 HAThOR : *Harmony Search* pour la recherche d'objets

Require: $HMS \in \mathbb{N}^+$, population Size
 $HMCR \in [0,1)$, Rate for choosing HM value
 $PAR \in [0,1)$, Rate for choosing neighbour value
 $V(G1), V(G2) \leftarrow$ Set of nodes of the graphs $G1$ and $G2$ corresponding to the left and the right images
 $stop_alg() \leftarrow$ Stop criteria, processing time || satisfying fitness function

Ensure: $s \leftarrow \{V_0, d_1, \dots, d_n\} \in S$, S set of possible solutions and n is a node belonging to an image
node $v[m] \in V(G1), w[m] \in V(G2)$, where m is the number of nodes to match between stereoscopic images
#INITIALIZATION#
for $i = 0$ **to** HMS **do**
 $v[m] \leftarrow$ random nodes $\in V(G1)$
 $w[m] \leftarrow$ random nodes $\in V(G2)$
 $HM[i] \leftarrow M[v[m], w[m]]$, Population HM where M is the set of nodes
end for
#IMPROVISATION#
while $stop_alg() \neq \text{True}$ **do**
 $s_{new} \leftarrow M[v[m], w[m]]$, Population HM where M is the set of nodes
 $x_i \leftarrow M[v[0], w[0]]$, Set of two nodes v and w
 for $i = 0$ **to** m **do**
 if $random[0, 1) \leq HMCR$ **then**
 $x_i \leftarrow$ Random $M[v[i], w[i]]$ from HM
 if $random[0, 1) \leq PAR$ **then**
 #Inversion of a node compared to a previous#
 $tempon \leftarrow v[i - 1]$
 $v[i - 1] \leftarrow v[i]$
 $v[i] \leftarrow tempon$
 $x_i \leftarrow M[v[i], w[i]]$
 end if
 else
 $v[i] \leftarrow$ random node $\in V(G1)$
 $w[i] \leftarrow$ random node $\in V(G2)$
 $x_i \leftarrow M[v[i], w[i]]$
 end if
 end for
 $s_{new} \leftarrow X\{0, 1, \dots, i\}$
 $s_{worst} \leftarrow s_i \in HM | f(s_i) < f(s_j) \forall s_j \in HM, s_j \neq s_i$
 #MEMORY UPDATE#
 if $f(s_{new}) < f(s_{worst})$ **then**
 REMOVE s_{worst} from HM
 end if
end while
 $s \leftarrow s_i \in HM | f(s_i) > f(s_j) \forall s_j \in HM, s_j \neq s_i$

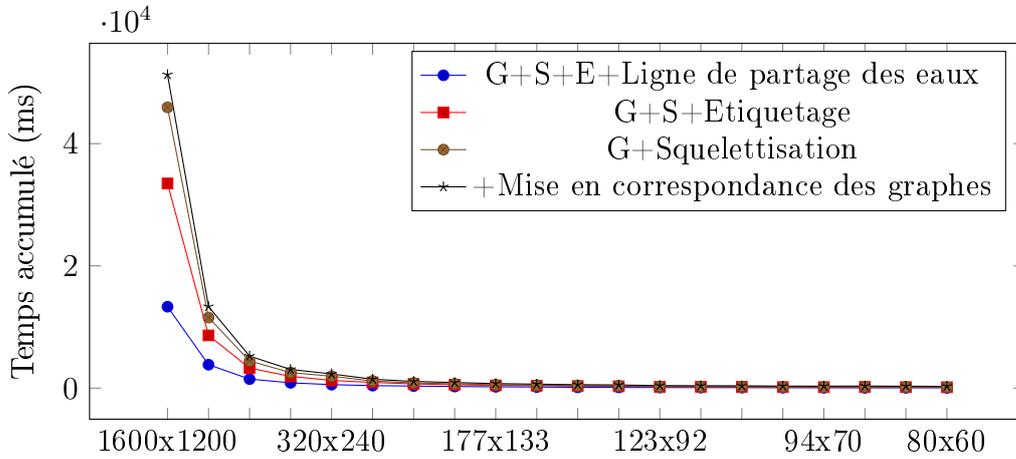


FIGURE 7.2 – HAThOR : Temps de prétraitement nécessaire pour simplifier l'image en fonction de leurs tailles (1600x1200-84x63)

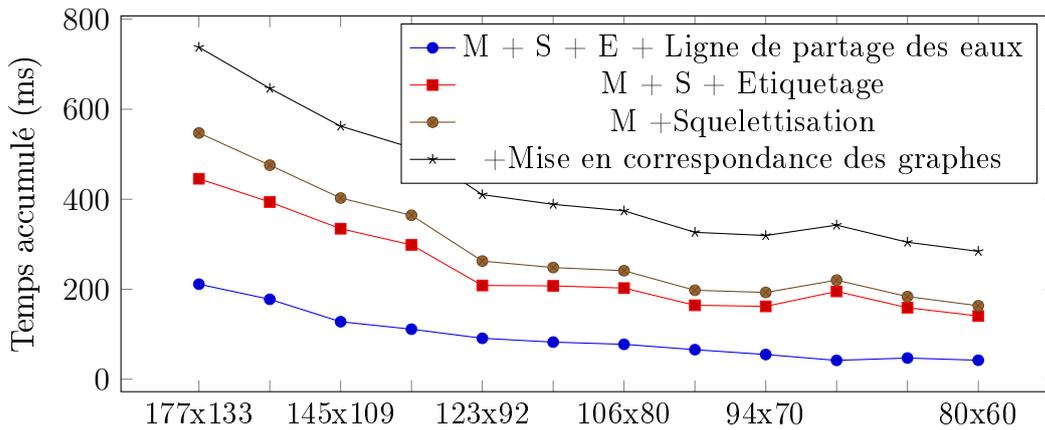


FIGURE 7.3 – HAThOR : Temps de prétraitement nécessaire pour simplifier l'image en fonction de leurs tailles (petites tailles) (177x33 to 84x63)

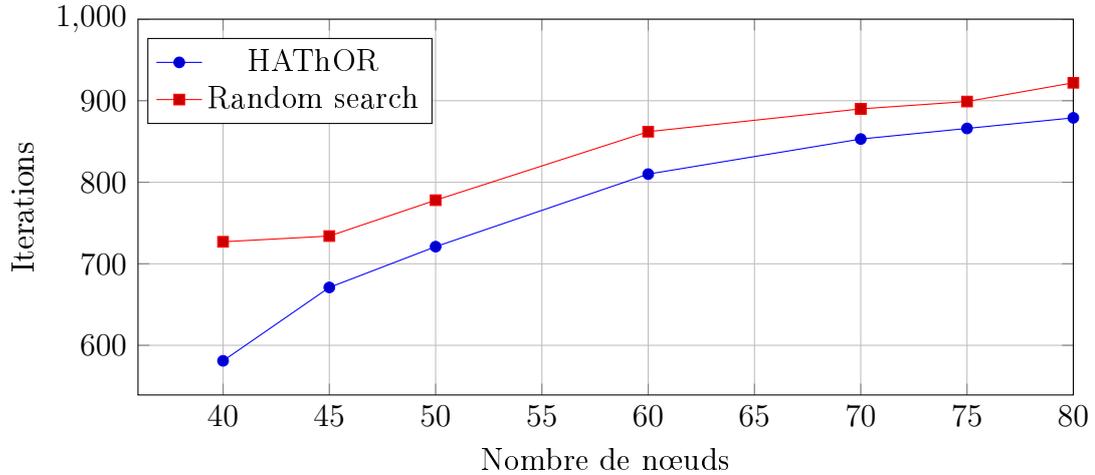


FIGURE 7.4 – HAThOR : Performance de la mise en correspondance de deux nœuds en fonction du nombre de nœuds

entrée. Nous avons simulé et comparé le nombre d'itérations requis pour deux algorithmes pour trouver deux nœuds parmi plusieurs. Les nœuds n'ont pas été récupérés d'une image mais générés par un programme afin d'avoir le nombre de nœuds exact à mettre en correspondance. Dans ce test, HAThOR a été comparé à une méthode de recherche aléatoire pour démontrer l'efficacité de l'algorithme évolutionnaire. Comme l'indique la Figure 7.4, le nombre d'itérations requis pour les deux algorithmes dépend du nombre de nœuds. HAThOR converge plus rapidement que la recherche aléatoire. Le prétraitement (segmentation par ligne de partage des eaux, étiquetage) a été exclu du temps de calcul afin de se concentrer sur le temps de traitement d'*Harmony Search*. Le temps de traitement comprend uniquement la phase de mise en correspondance de graphes. Il convient de noter que le nombre d'itérations détermine non seulement le temps de traitement, mais aussi la précision.

7.4.1.3 Performances totale de la recherche d'objets

Pour démontrer la performance d'HAThOR, nous avons déterminé le temps de traitement moyen que va avoir l'algorithme en fonction de la taille de

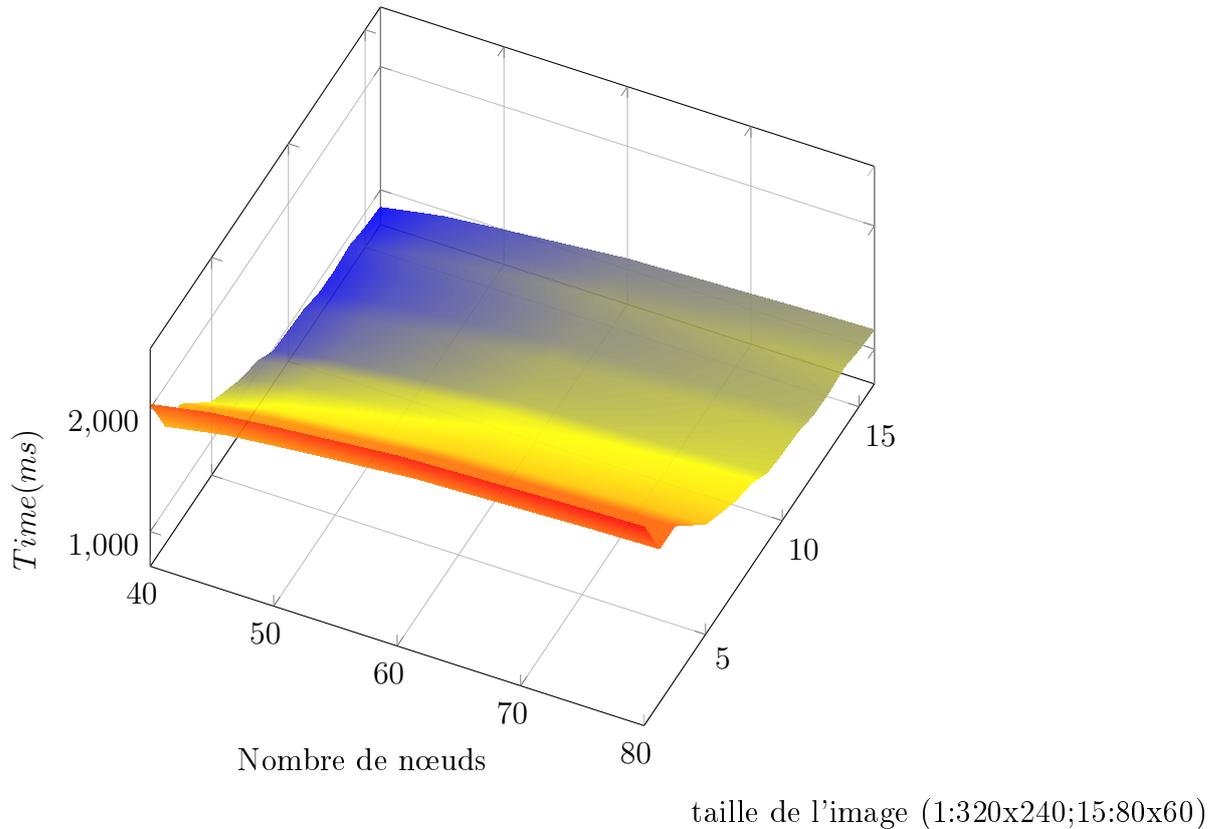


FIGURE 7.5 – HAThOR : Temps total de traitement total en fonction de la taille de l'image et du nombre de nœuds ; bleu : bon temps de recherche, rouge : mauvais temps de recherche

l'image et du nombre de nœuds présent dans l'image. Nous avons pour ceci concaténé le temps de prétraitement avec le temps de mise en correspondance des nœuds. Comme représenté dans la Figure 7.5, nous pouvons remarquer que la taille de l'image ne doit pas excéder 320x240 et que le nombre de nœuds joue un impact sur le temps. Une recherche d'objet pour une cinquantaine de nœuds peut s'effectuer facilement en moins d'une seconde.

7.4.2 Précision

La précision est l'un des aspects les plus importants de l'expérimentation. Il détermine la qualité de l'algorithme proposé. L'algorithme diffère de Thor

car il utilise un algorithme de segmentation pour le prétraitement de l'image dans le but de séparer les objets les uns des autres. Pour la détection d'objets fins, nous avons évalué de différentes manières la précision d'HAThOR. Tout d'abord, HAThOR doit correctement identifier des objets simples fins et sans motifs puis en deuxième détecter des objets placés dans une bassine. Enfin, l'algorithme doit détecter des objets plus épais, bruités et transparents.

7.4.2.1 Reconnaissance d'objets simples

Pour visualiser le comportement de l'algorithme, nous l'avons appliqué à la reconnaissance d'objets simples. Sur la Figure 7.6, nous avons utilisé l'algorithme pour reconnaître des objets avec peu d'enchevêtrements. Nous avons présenté dans le chapitre précédent deux méthodes de simplification d'images. Dans ce chapitre, nous avons utilisé l'algorithme de partage des eaux. Cet algorithme de segmentation permet de mieux séparer les objets tandis que l'algorithme de binarisation ne fait que ressortir les pixels situés entre deux ou plusieurs niveaux de gris. La segmentation permet de réunir les pixels communs à l'objet et d'effectuer un premier filtre sur les objets en fonction de leurs couleurs et le nombre de pixels appartenant à cet objet. Cette séparation permet de créer des squelettes significatifs. Un bon étalonnage permet d'identifier la distance entre les nœuds, puis d'identifier l'objet.

7.4.2.2 Recherche d'objets placés dans une bassine

Comme l'indique la Figure 7.7, nous avons utilisé l'algorithme pour identifier des objets placés dans une bassine. Les images 7.7a et 7.7b représentent des exemples de recherche d'objets enchevêtrés. Les objets avec une taille de pixel proche de l'objet recherché peuvent être mis en avant. Dans la Figure 7.7c, la recherche concerne des objets sans motif avec plusieurs nœuds. Pour pouvoir saisir cet objet, il est nécessaire d'avoir plus de deux nœuds mis en

Image	<i>Watershed</i>	Squelettisation	Objet

FIGURE 7.6 – HAThOR et son application à la détection d'objets : Colonne 1 - image d'entrée ; Colonne 2 - application de la segmentation par ligne de partage des eaux ; Colonne 3 - application de la squelettisation ; Colonne 4 - Application de la recherche d'objet. Segment bleu (colonne 4) : segment entre 2 nœuds correspondant à l'objet désiré

commun pour un objet afin de pouvoir récupérer les informations de rotation de l'objet. Les objets placés dans une bassine ont été détectés avec succès. Un test a également été effectué pour identifier un objet partiellement visible, comme le représente la Figure 7.7d. Lorsque l'opérateur entre la taille de l'objet, une échelle peut être sélectionnée. Si la dimension de l'objet est inférieure à cette échelle, l'objet peut être détecté.

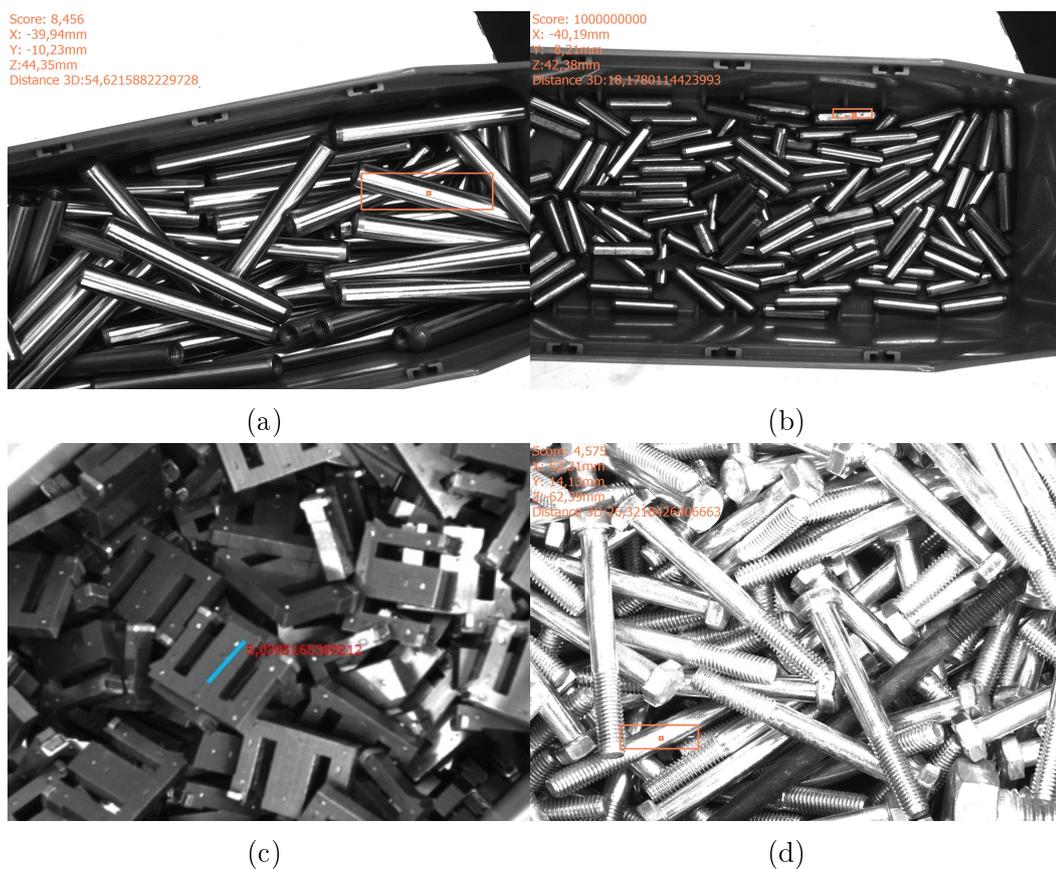


FIGURE 7.7 – HAThOR : Identification d'objets placés dans une bassine

7.4.2.3 Recherche d'objets inhabituels et variation de luminosité sur l'image

Par la suite, l'algorithme a été utilisé pour localiser des objets inhabituels (motifs, transparence..). Par exemple, nous avons appliqué notre algorithme

pour détecter des objets transparents, en particulier pour trouver des stylos, vue sur la Figure 7.8a.



(a) Objets transparents



(b) Objet avec des motifs



(c) Objets fins sans motif



(d) Objets ronds

FIGURE 7.8 – HAThOR : Evaluation sur des objets spécifiques

La recherche d'un objet transparent permet d'introduire une recherche d'objet plus complexe, car l'interprétation des images peut facilement mélanger les objets les uns aux autres. Comme l'indique la Figure 7.8b, nous avons également testé la capacité de notre algorithme à détecter des objets avec beaucoup de motifs. Ce test est effectué pour déterminer la limite de la recherche pour des objets avec des points clés. La segmentation par ligne de partage des eaux et l'étiquetage permettent de simplifier l'image, afin de récupérer uniquement le centre de l'objet. Un petit objet sans motif a également été détecté, Figure 7.8c. Cette recherche démontre l'efficacité de la simplification de l'image. Ensuite, nous avons appliqué notre algorithme pour rechercher un objet rond placé dans une bassine, comme l'indique la Figure 7.8d. Enfin, comme l'illustrent les exemples de la Figure 7.9, nous avons testé notre algorithme avec différents niveaux de luminosité. L'algorithme se comporte correctement grâce à la segmentation par ligne de partage des eaux qui sépare les objets de manière adéquate, en dépit de la différence de lumino-

sité. Pour être saisis par le robot, les objets représentés par les Figures 7.8b, 7.8c, 7.8d et 7.9 nécessitent l'identification de plusieurs nœuds appartenant à l'objet dans le but de déterminer la position de l'objet.

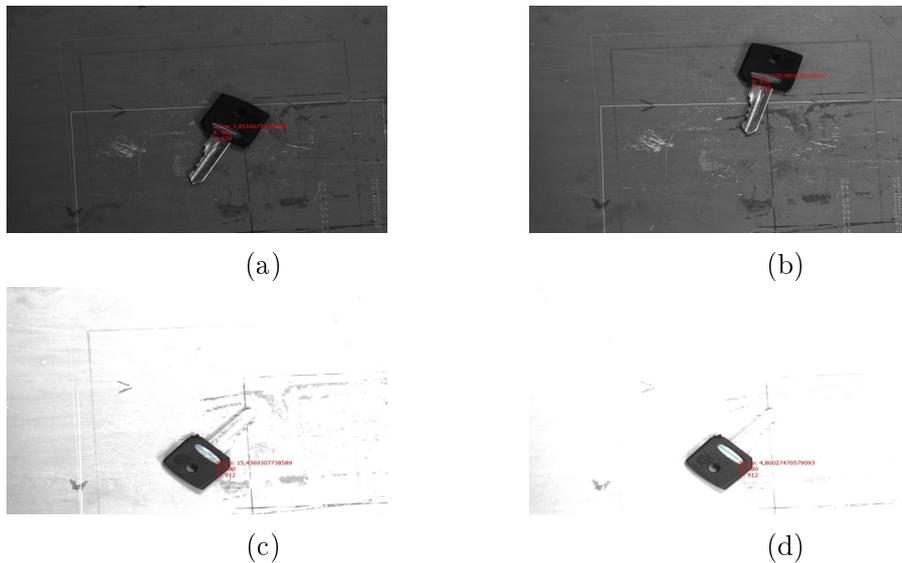


FIGURE 7.9 – HAThOR : Détection de clés avec des luminosités différentes

7.4.2.4 Effets du bruit sur la performance

L'algorithme a été testé avec différents types de bruits. Premièrement, et comme représenté sur la Figure 7.10, nous avons utilisé l'algorithme pour identifier des objets lorsque l'image est floue. 1242 images ont été générées pour prouver la précision de l'algorithme. Nous avons utilisé un flou Gaussien avec différentes tailles et poids. Pour chaque pixel, on attribue la moyenne de lui-même et de ses voisins en fonction d'une matrice de convolution de taille $M \times M$ (noyau). Des images représentant des objets dans une bassine ont principalement été utilisées. Nous avons employé un poids minimum de 55. Pour les poids inférieurs à cette valeur, le pourcentage d'objets trouvés est proche de 100%. Nous observons que les images floues ont un impact sur le comportement de l'algorithme. Les formes représentées sur l'image bruitée modifient le comportement de l'algorithme de segmentation par ligne de partage des eaux.

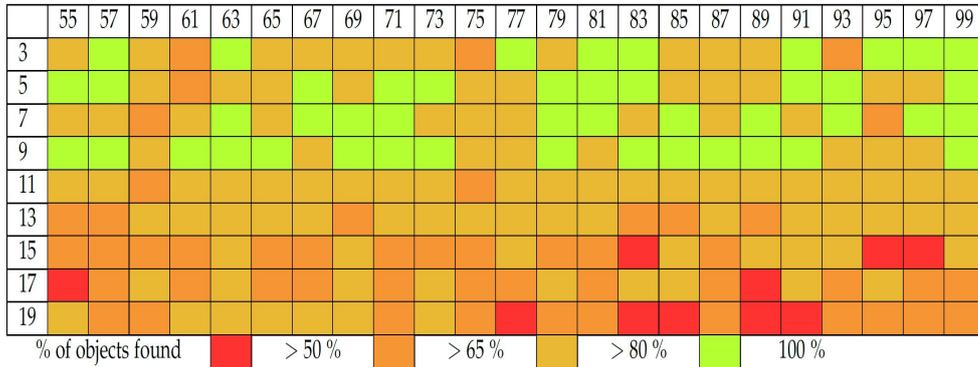


FIGURE 7.10 – HAThOR : Performance à détecter des objets lorsque l'image est floutée, Gaussien Noyau (6 objets * 23 (X :poids) * 9 (Y :Noyau) = 1242 images)

Ces images sont très saturées. Dans le pire des cas, le taux d'identification est supérieur à 50%. Avec un poids de 99 et une taille de noyau de 19, l'image est complètement floue. Nous pouvons remarquer qu'un noyau inférieur à 19 perturbe peu l'algorithme. La recherche se comporte correctement avec une détection comprise entre 65 et 80% jusqu'à un noyau de 15. La recherche est ensuite plus complexe et l'algorithme détecte entre 50 et 65% les objets proposés. Cependant, les objets sont correctement identifiés, ce qui confirme l'efficacité de l'algorithme lorsque les images sont bruitées.

Nous avons également testé notre algorithme en ajoutant au hasard des pixels blancs et noirs à l'intérieur de l'image. Nous avons généré 1000 images. Sur la Figure 7.11, entre 1 et 20% de bruit a été ajouté à l'image. Le niveau de bruit a un effet significatif sur la détection d'objet. Plus de 60% des objets sont trouvés.

7.4.3 Comparaison d'HAThOR avec des algorithmes d'extraction de points clés

L'algorithme HAThOR a été comparé aux algorithmes d'extraction de points clés en recherchant des objets à partir des images utilisées dans la

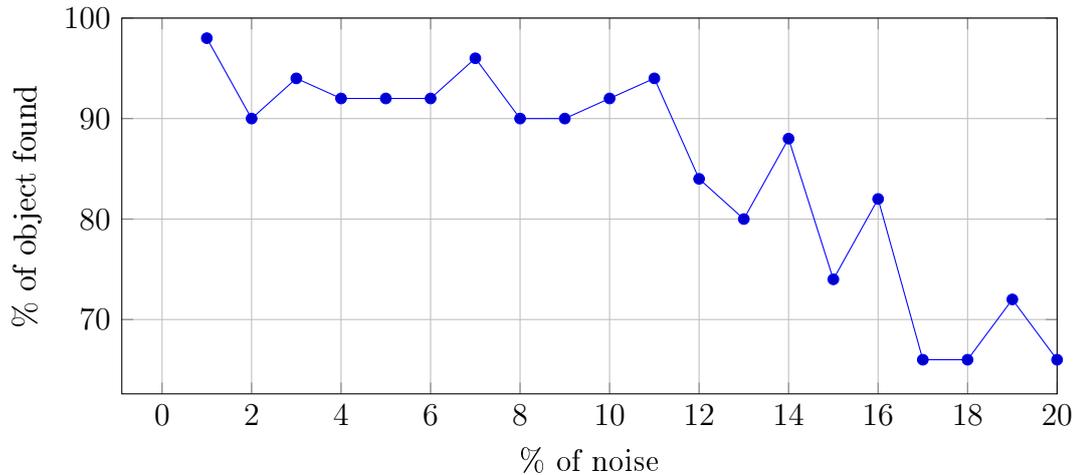


FIGURE 7.11 – HAThOR : Détection d’objets pour différents niveaux de bruits ajoutés, représentée par un pourcentage de pixels blancs et noirs ajoutés au hasard dans l’image

phase expérimentale. L’apprentissage est appliqué en prenant une image d’un seul des objets à identifier. L’objet est considéré comme étant trouvé si le nombre de points clés corrects est supérieur au nombre total de points clés. Afin de comparer notre algorithme avec les méthodes existantes, nous avons répété les tests décrits dans les sections précédentes à l’aide de la méthode SIFT, de la méthode SURF et d’une méthode d’extraction de coins (Harris) combinée à une corrélation maximale et à une estimation de l’homographie. Les algorithmes SIFT et SURF ont obtenu des résultats similaires, avec approximativement 65% d’objets trouvés pour les images utilisées à la Figure 7.6. Le descripteur de Harris a un taux plus élevé, avec 80% d’objets trouvés. Nous avons également testé les mêmes algorithmes en utilisant la base de données d’images bruitées. Les objets n’ont pas été détectés en raison de la sensibilité des algorithmes au bruit. Ces algorithmes sont moins efficaces car ils recherchent des points clés inexistantes dans ces images. HAThOR a de meilleurs résultats car il va utiliser la segmentation pour identifier les candidats potentiels. Cette première sélection permet ensuite d’extraire le squelette des objets souhaités. Le traitement d’images est ainsi effectué sur une zone

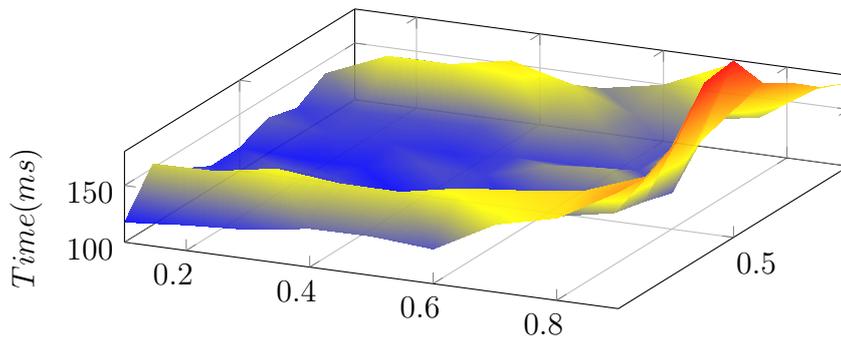
contrôlée en fonction de sa taille et de sa couleur, ce qui est un gain de temps et permet de rajouter du bruit sur les objets à détecter en retirant les zones inutiles.

7.5 Paramétrage

Pour évaluer l'impact des différents paramètres sur le processus de recherche d'objet et le temps de traitement, nous avons testé l'algorithme dans sa capacité à chercher deux nœuds spécifiques simulés parmi un ensemble de nœuds créés aléatoirement. Tout d'abord, nous montrons que les paramètres *PAR* et *HMCR* jouent un rôle primordial dans la phase d'improvisation de la recherche d'objet. Sur la Figure 7.12, le paramètre *HMCR* doit être compris entre 0,4 et 0,6 et *PAR* entre 0,1 et 0,3. Nous avons ensuite réglé ces paramètres sur *HMCR*=0,5 et *PAR*=0,3, afin d'étudier l'effet de *HMS* et du nombre de nœuds pris en compte. Comme l'indique la Figure 7.13, *HMS* et le nombre de nœuds ont un rôle important dans la recherche d'objet. *HMS* doit être réglé entre 20 et 70. Par exemple, un *HMS* de 37 obtient de bons résultats. En dessous, on augmente considérablement le nombre d'itérations et l'algorithme se comporte de manière aléatoire. Au-dessus de 70, le nombre d'itérations s'accroît également, bien que plus lentement. Le choix des valeurs appropriées de paramètres est crucial pour une recherche optimale d'objets.

7.6 Discussion

Les résultats obtenus lors de l'évaluation de l'algorithme d'optimisation prouvent l'efficacité d'HAThOR pour trouver des composants fins placés aléatoirement. Les graphes permettent de récupérer les points de jonction et extrêmes. L'algorithme HAThOR permet de trouver l'objet correspondant à un modèle. Le paramétrage joue un rôle important dans l'optimisation de la re-



Neighbor value rate selection (PAR)

Memory value selection rate ($HMCR$)

FIGURE 7.12 – HATHOR : Performance en fonction des paramètres Pitch Adjusting Rate (PAR) et *Harmony Memory Considering Rate* ($HMCR$) pour un pas de 0.1 : recherche de deux nœuds spécifiques parmi 40

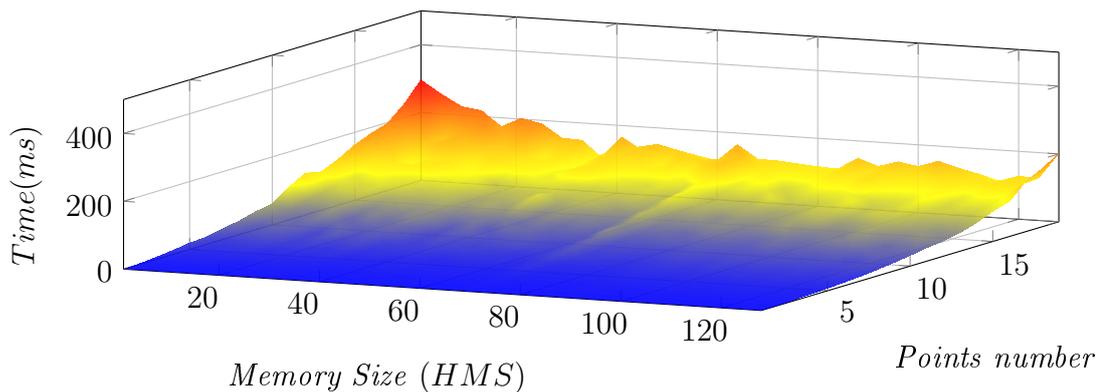


FIGURE 7.13 – HATHOR : Performance en fonction du nombre de nœuds et du paramètre *Harmony Memory Size* (HMS) pour un pas de 0.1

cherche de l'objet. La variable HMS doit être suffisante, sinon la recherche est alors très proche de l'aléatoire. Une population trop élevée rend l'algorithme inefficace parce que le système prend du temps à optimiser sa recherche. L'efficacité a été démontrée par la capacité de la méthode à trouver l'objet dans un temps tronqué d'environ 40% par rapport à une recherche aléatoire. Un autre avantage est la simplicité de la mise en place du système qui a seulement besoin de la valeur de la distance entre deux nœuds facilement mesurables par l'opérateur, ainsi que d'un ensemble de caractéristiques connues. Le temps de

traitement dépend du nombre de nœuds. Les temps de traitement peuvent être considérés comme acceptables pour une utilisation industrielle du "bin-picking" dans une ligne de production. L'algorithme a une complexité polynomiale, car le système utilise plusieurs briques imbriquées. La méthode s'avère efficace pour trouver des objets minces avec un petit nombre de nœuds.

7.7 Conclusions

Dans ce chapitre, une méthode d'amélioration a été proposée pour reconnaître des objets fins disposés aléatoirement dans un espace tridimensionnel en un minimum de temps en utilisant des images stéréoscopiques et un algorithme évolutionnaire. Un ensemble de filtres est appliqué pour chacune des images en deux dimensions dans le but de récupérer la localisation des objets désirés sur la scène. La simplification permet de calculer la taille des objets positionnés dans l'espace tridimensionnel. Pour chaque objet dans l'image, un squelette est construit et l'algorithme *Harmony Search* est appliqué pour effectuer la mise en correspondance des points de jonction et des points extrêmes récupérés par le biais des graphes. Les points sont aléatoirement mis en correspondance en deux et trois dimensions. Tous les objets ont été trouvés. Deux critères de validation ont été mis en place et approuvés pour prouver le bon fonctionnement de l'algorithme. Premièrement, HAThOR localise des objets en un temps inférieur à deux secondes grâce à l'utilisation d'un algorithme évolutionnaire appelé *Harmony Search*. Deuxièmement, l'algorithme prouve son efficacité pour retrouver différents types d'objets disposés de façon aléatoire et enchevêtrés les uns sur les autres. Nous avons poussé l'algorithme dans ses retranchements en identifiant des objets avec des points clés. L'algorithme fonctionne avec différents types de bruits. Le bruit peut être représenté par un changement de luminosité ou encore par une image floue. Le système est facilement initialisé car il requiert simplement les paramètres des dimensions

de l'objet connues par l'opérateur. A la différence de Thor, il peut récupérer en moins de 2 secondes la position d'un objet. Son prétraitement diffère aussi, Thor binarise tandis que HAThOR utilise un algorithme de segmentation. La binarisation va créer deux classes (objet ou non) tandis que la segmentation va en produire plusieurs. Ceci permet de séparer les objets les uns des autres.

Conclusion et perspectives

Sommaire

8.1	Conclusion technique	134
8.2	Conclusion scientifique	136
8.3	Les publications	139



8.1 Conclusion technique

Ce travail de thèse a traité deux problématiques, à savoir : le calibrage de caméras et la détection d'objets fins. La thèse est axée sur le développement d'un logiciel visant à faciliter la mise en place et l'utilisation d'un système de "Pick-and-Place". Ce travail n'est qu'un petit pas sur le domaine du "Pick-and-Place" qui est encore loin de la machine universelle qui pourra saisir n'importe quel type d'objet dans tout environnement. Afin d'avoir un système éclectique, nous avons proposé un maximum de fonctions pour détecter ces objets qui possèdent leurs avantages et leurs inconvénients. La plupart des méthodes proposées dans la littérature étant focalisées sur des objets avec des contours et des points clés qui ressortent facilement de l'image. Nous avons focalisé notre recherche sur une approche différente dans le but d'amener de nouvelles fonctions et de détecter des objets fins. Même si cette thèse est un sujet de développement pour des systèmes de production, elle permet d'ouvrir de nouveaux horizons industriels dans le domaine de l'intelligence artificielle et de l'usine hyper flexible du futur.

8.1.1 Les réalisations techniques

Une grande partie de la thèse a consisté à programmer un logiciel simple d'utilisation pour détecter des objets dans une bassine et d'envoyer des coordonnées 3D à un robot 6 axes. Plusieurs aspects du programme ont nécessité de longues réflexions et un grand nombre d'essais et d'erreurs. Le programme ne devait être utilisé que pour identifier des objets (ThOR/HAThOR) et calculer la position de ceux-ci. Cependant, nous nous sommes vite rendu compte de la complexité d'avoir un système stéréoscopique précis. Il a été nécessaire d'avoir une réflexion afin d'optimiser au maximum la méthode de calibrage pour proposer à l'utilisateur des données cohérentes en un minimum de temps et sans avoir à renseigner des informations difficilement récupérables (Al-Thocb). Un

grand nombre de fonctionnalités ont été mises en place dans le logiciel :

- interaction PC, Robot, Caméras : cette partie met en relation les trois composants. Le PC est le centre du système. Il va tout d'abord positionner le robot à l'étape initiale et va ensuite demander un ensemble d'images aux caméras. Celles-ci seront analysées dans le but de faire ressortir les objets à sélectionner. Le calcul de la coordonnée 3D et de l'orientation de l'objet permet d'envoyer ces données au robot pour qu'il puisse saisir l'objet.
- automatisation et optimisation du calibrage de caméras stéréoscopiques : Cette fonctionnalité a été réalisée pour simplifier la mise en place du système de calibrage par un opérateur. Il repose sur un algorithme évolutionnaire qui vient sélectionner les images les plus cohérentes pour le calibrage des caméras stéréoscopiques.
- détection d'objets fins disposés en vrac dans une bassine : Cette fonctionnalité a été mise en place pour identifier des objets sans motifs. La méthode repose sur un ensemble de fonctions de simplification de l'image bidimensionnelle pour mettre en avant sur l'image l'ensemble d'objets potentiels à saisir par le robot. Un squelette est ensuite construit pour récupérer la forme de l'objet. Les points d'extrémités et de jonction du squelette sont ensuite recherchés et mis en correspondance entre les deux images stéréoscopiques.
- mise en place d'algorithmes standard de détection d'objets (SIFT, SURF, Approximation polygonale). L'objectif est de proposer à l'entreprise plusieurs algorithmes qui ont leurs avantages et leurs inconvénients. Ces algorithmes permettent de détecter des objets non fins disposés en vrac dans une bassine.
- mise en place de fonction de traitement d'images : Un ensemble de fonctionnalités ont été ajoutées au système pour simplifier les images telles que la segmentation par ligne de partage des eaux, la binarisation

- d'une image, simplification par couleur, l'étiquetage des images...
- création d'un système de suivi des opérations effectuées par l'opérateur. En cas de débogage, ce système permet d'avoir un retour sur les manipulations effectuées par l'opérateur.
- les temps de traitement sont respectés
- mise en place d'un algorithme d'automatisation et d'optimisation de la mise en correspondance de graphes pour des images stéréoscopiques prétraitées.

8.1.2 Les perspectives

Du point de vue technique pour l'entreprise, il est nécessaire de récupérer à court terme les connaissances (théoriques et pratiques) pour faire perdurer le système. A moyen terme, il est nécessaire de transformer le code au standard de l'entreprise en respectant les règles mises en place par l'entreprise. Il sera nécessaire à long terme de standardiser un procédé de fabrication de "bin-picking" afin d'avoir une utilisation la plus simpliste qui soit. Standardiser cette méthode permettra une mise en place dans diverses zones de production de l'entreprise.

8.2 Conclusion scientifique

8.2.1 Le calibrage

Une des premières contributions scientifiques est l'automatisation d'un système de calibrage de caméras stéréoscopiques à l'aide d'un algorithme d'optimisation. Le système permet de sélectionner automatiquement les images les plus adéquates pour calibrer des caméras. Comme la plupart des algorithmes de calibrage, ce système a besoin d'une base de données d'images en entrée et de données d'entrée. Les données d'entrée sont la taille du motif représenté sur

l'image et le nombre d'itérations que l'algorithme doit effectuer. Le nombre d'itérations a un impact sur la précision du calibrage.

8.2.2 Détection d'objet

Nous avons au cours de cette thèse proposé un algorithme de reconnaissance d'objets fins. Nous avons programmé un algorithme de mise en correspondance de graphes où les graphes sont représentés par la forme centrale (squelette) de l'objet. L'objectif a été de proposer une méthode différente de celles présentes dans la littérature qui se concentrent sur un modèle d'apprentissage par image. Nous avons utilisé des données physiques de l'objet tel que sa taille. L'algorithme exploite les données 3D des points cohérents récupérés après avoir simplifié l'image. Ces données sont plus facilement récupérables et comparables à l'objet à rechercher tandis qu'une image comporte beaucoup trop de données non exploitables. La mise en correspondance des graphes diffère de la littérature car l'objectif n'est pas de comparer un graphe modèle avec l'image à traiter, mais de comparer deux graphes similaires pour trouver les pixels importants pour la détection de l'objet entre les images stéréoscopiques.

8.2.3 L'avantage des algorithmes évolutionnaires

Nous avons adapté tout au long de cette thèse un algorithme évolutionnaire dans le but d'optimiser les fonctions nécessaires quant au bon fonctionnement du système, et ceci dans un temps borné. Nous avons utilisé un algorithme appelé Harmony Search. Nous l'avons sélectionné car il répond à la demande d'effectuer des recherches dans un temps borné, est facilement adaptable grâce aux valeurs d'exploration et d'exploitation paramétrables. L'algorithme présente une bonne convergence vers une solution optimale. De plus, l'utilisation d'un seul algorithme d'optimisation pour deux applications dans un programme facilite la maintenance éventuelle du système. Dans cette

thèse, l'utilisation d'un algorithme évolutionnaire permet d'optimiser et d'automatiser le processus de calibrage de caméras stéréoscopiques. Il permet aussi d'optimiser la mise en correspondance de graphes provenant du squelette d'objets représentés sur des images stéréoscopiques.

8.2.4 Les perspectives

Sur le plan des perspectives liées aux travaux présentés dans cette thèse, il serait intéressant, à court terme, d'ajouter au programme HAThOR, une fonction d'apprentissage basée sur des images, ce qui permettrait de laisser le choix à l'opérateur entre un apprentissage avec des données physiques détaillées (taille, couleur) et un jeu d'images. Plusieurs pistes sont envisageables et notamment, l'application de l'algorithme de partage des eaux déjà implémenté avec différents seuils de segmentation sur les images stéréoscopiques du modèle, ce qui aurait l'avantage de fournir différents nœuds à des régularités différentes. Dès lors, l'algorithme pourrait effectuer un apprentissage des différents nœuds récupérés et les comparer avec les images à traiter. Une autre piste d'amélioration concerne l'étude de la stéréovision et notamment, de l'erreur de reprojection utilisée pour calculer la précision de l'algorithme de calibrage. Dans ce cadre, l'objectif serait, dans un premier temps, de déterminer si ajouter une seconde fonction de coût associée à un ensemble de distances peut améliorer la performance du calibrage. Une telle approche nécessiterait de multiplier les calculs de distances entre les points réels et les points dans l'image, mais permettrait sans doute d'obtenir une fonction de fitness plus robuste et plus cohérente. D'un point de vue général et avec l'émergence de l'industrie 4.0, il serait intéressant de continuer à introduire les approches d'intelligence artificielle dans les entreprises, notamment à l'aide des algorithmes évolutionnaires. En effet, comme nous l'avons vu dans ce manuscrit, pour la résolution de problèmes d'optimisation, ces algorithmes offrent un

bon compromis entre l'obtention d'une solution de qualité et les contraintes de temps. Cependant, leur champ d'application est beaucoup plus vaste et on peut penser, qu'à moyen terme, des applications dans le domaine du traitement d'images devraient s'imposer. De même, pour la détection d'objets, un algorithme basé sur l'apprentissage par "deep learning" pourrait être étudié. Actuellement, cette méthode devient de plus en plus incontournable et a déjà trouvé une application dans la lecture de chèques. Cependant, pour que ces approches se diffusent dans le milieu industriel, il sera impératif de trouver des méthodes d'apprentissage moins coûteuses en temps et en données d'entrée.

8.3 Les publications

Un ensemble de publications et de soumissions a été effectué durant ces trois années de thèse. Un article pour la détection d'objets fins a été présenté lors de la conférence internationale MVA (*Machine vision and application* - IAPR) à Nagoya (Japon) [Willaume 2017b]. L'utilisation de l'algorithme Harmony Search pour l'optimisation de la recherche a permis la présentation d'un article à la conférence SSCI (*Symposium Series on Computational Intelligence*) à Athènes (Grèce) [Willaume 2016]. L'utilisation d'algorithmes d'optimisation pour les tâches de "Pick-and-Place" a permis la publication d'un article à la conférence internationale ICPRS (*International Conference on Pattern Recognition Systems* - IAPR) [Willaume 2017a]. Actuellement, deux articles sont en soumissions. Premièrement, une soumission pour le journal *Machine-Vision and Application* a été proposée pour le calibrage de caméras. Deuxièmement, une soumission au journal TPAMI (*Transactions on Pattern Analysis and Machine Intelligence*) a été effectuée pour la reconnaissance d'objets fins dans une bassine.

Bibliographie

- [Acharya 2005] Tinku Acharya et Ajoy K Ray. Image processing : principles and applications. John Wiley & Sons, 2005. (Cité en page 32.)
- [Aguilar 1996] Juan-José Aguilar, F Torres et MA Lope. Stereo vision for 3D measurement : accuracy analysis, calibration and industrial applications. *Measurement*, vol. 18, no. 4, pages 193–200, 1996. (Cité en page 66.)
- [Al-salami 2009] Nada MA Al-salami. Evolutionary algorithm definition. *American J. of Engineering and Applied Sciences*, vol. 2, no. 4, pages 789–795, 2009. (Cité en page 50.)
- [Allaire 2008] Stéphane Allaire, John J Kim, Stephen L Breen, David A Jaffray et Vladimir Pekar. Full orientation invariance and improved feature selectivity of 3D SIFT with application to medical image analysis. In *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW'08. IEEE Computer Society Conference on*, pages 1–8. IEEE, 2008. (Cité en page 37.)
- [Angeline 1998] Peter J Angeline. A historical perspective on the evolution of executable structures. *Fundamenta Informaticae*, vol. 35, no. 1-4, pages 179–195, 1998. (Cité en page 50.)
- [Aubert 2005] Gilles Aubert, Jean-Francois Aujol et Laure Blanc-Feraud. Detecting CodimensionTwo Objects in an Image with Ginzburg-Landau Models. *International Journal of Computer Vision*, vol. 65, no. 1, pages 29–42, 2005. (Cité en page 47.)
- [Auger 2011] Anne Auger et Nikolaus Hansen. Theory of evolution strategies : a new perspective. *Theory of Randomized Search Heuristics : Foundations and Recent Developments*, vol. 1, pages 289–325, 2011. (Cité en page 52.)

- [Ayache 1991] Nicholas Ayache et Francis Lustman. Trinocular stereovision for robotics. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 13, no. 1, 1991. (Cité en page 66.)
- [Back 1997] Thomas Back, Ulrich Hammel et H-P Schwefel. Evolutionary computation : Comments on the history and current state. IEEE transactions on Evolutionary Computation, vol. 1, no. 1, pages 3–17, 1997. (Cité en page 50.)
- [Baird 1977] Michael L Baird. Image Segmentation Technique for Locating Automotive Parts on Belt Conveyors. In IJCAI, pages 694–695, 1977. (Cité en page 24.)
- [Ballard 1981] Dana H Ballard. Generalizing the Hough transform to detect arbitrary shapes. Pattern recognition, vol. 13, no. 2, pages 111–122, 1981. (Cité en page 26.)
- [Bauckhage 2004] Christian Bauckhage, Elke Braun et Gerhard Sagerer. From features to symbols and vice versa using graphs to loop data and model-driven processing in visual assembly recognition. International Journal of Pattern Recognition and Artificial Intelligence, vol. 18, no. 03, pages 497–517, 2004. (Cité en page 7.)
- [Bay 2006] Herbert Bay, Tinne Tuytelaars et Luc Van Gool. Surf : Speeded up robust features. In European conference on computer vision, pages 404–417. Springer, 2006. (Cité en pages 33, 40 et 41.)
- [Berger 2014] Roland Berger. Industrie 4.0 : The new industrial revolution. How Europe will succeed. Think Act, page 8, 2014. (Cité en page 16.)
- [Bernardini 2002] Fausto Bernardini et Holly Rushmeier. The 3D model acquisition pipeline. In Computer graphics forum, volume 21, pages 149–172. Wiley Online Library, 2002. (Cité en page 27.)

- [Beucher 1979] S Beucher et C Lantuejoul. International Workshop on Image Processing (Real-time Edge and Motion Detection/Estimation). 1979. (Cité en page 98.)
- [Biasotti 2008] Silvia Biasotti, Daniela Giorgi, Michela Spagnuolo et Bianca Falcidieno. Reeb graphs for shape analysis and applications. Theoretical Computer Science, vol. 392, no. 1, pages 5–22, 2008. (Cité en page 100.)
- [Bolles 1983] Robert C Bolles et Ronald A Cain. Recognising and locating partially visible objects : the local-feature-focus method. In Robot Vision, pages 43–82. Springer, 1983. (Cité en page 24.)
- [Bolles 1987] Robert C Bolles et Patrice Horaud. 3DPO : A three-dimensional part orientation system. In Three-dimensional machine vision, pages 399–450. Springer, 1987. (Cité en page 25.)
- [Brachmann 2014] Eric Brachmann, Alexander Krull, Frank Michel, Stefan Gumhold, Jamie Shotton et Carsten Rother. Learning 6d object pose estimation using 3d object coordinates. In European conference on computer vision, pages 536–551. Springer, 2014. (Cité en page 27.)
- [Brennecke 2004] Angela Brennecke et Tobias Isenberg. 3D Shape Matching Using Skeleton Graphs. In SimVis, pages 299–310. Citeseer, 2004. (Cité en page 101.)
- [Broggi 2005] Alberto Broggi, Claudio Caraffi, Rean Isabella Fedriga et Paolo Grisleri. Obstacle detection with stereo vision for off-road vehicle navigation. In 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)-Workshops, pages 65–65. IEEE, 2005. (Cité en page 66.)
- [Buchholz 2016] Dirk Buchholz. Bin-picking : New approaches for a classical problem, chapitre Bin-Picking—5 Decades of Research, pages 3–12.

- Springer International Publishing, Cham, 2016. (Cité en pages 16, 18, 22, 23, 25, 26 et 28.)
- [Bunke 1998] Horst Bunke et Kim Shearer. A graph distance metric based on the maximal common subgraph. *Pattern recognition letters*, vol. 19, no. 3, pages 255–259, 1998. (Cité en page 100.)
- [Canny 1986] John Canny. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, no. 6, pages 679–698, 1986. (Cité en page 32.)
- [Capek 1920] Karel Capek, N Playfair, P Selver et WA Landes. Rossum’s universal robots. Prague, CZ, vol. 1, 1920. (Cité en page 22.)
- [Champin 2003] Pierre-Antoine Champin et Christine Solnon. Measuring the similarity of labeled graphs. In *International Conference on Case-Based Reasoning*, pages 80–95. Springer, 2003. (Cité en page 100.)
- [Conte 2004] Donatello Conte, Pasquale Foggia, Carlo Sansone et Mario Vento. Thirty years of graph matching in pattern recognition. *International journal of pattern recognition and artificial intelligence*, vol. 18, no. 03, pages 265–298, 2004. (Cité en pages 7 et 100.)
- [Cornea 2007] Nicu D Cornea, Deborah Silver et Patrick Min. Curve-skeleton properties, applications, and algorithms. *IEEE Transactions on visualization and computer graphics*, vol. 13, no. 3, pages 0530–548, 2007. (Cité en page 99.)
- [Cross 1997] Andrew DJ Cross, Richard C Wilson et Edwin R Hancock. Inexact graph matching using genetic search. *Pattern Recognition*, vol. 30, no. 6, pages 953–970, 1997. (Cité en page 100.)
- [Darwin 1872] C. Darwin. *The origin of species*. Collins Classics. HarperCollins Publishers, 1872. (Cité en page 50.)
- [Davis 1999] Trevor J. Davis. Fast decomposition of digital curves into polygons using the Haar transform. *IEEE Transactions on Pattern*

- Analysis and Machine Intelligence, vol. 21, no. 8, pages 786–790, 1999. (Cité en page 43.)
- [Debled-Rennesson 2004] Isabelle Debled-Rennesson, Salvatore Tabbone et Laurent Wendling. Approximation polygonale à partir d'une analyse multi-ordres des points de contour. In Conférence Internationale Francophone sur l'Écrit et le Document (CIFED 04), 2004. (Cité en page 43.)
- [Deruyver 1997] Aline Deruyver et Yann Hodé. Constraint satisfaction problem with bilevel constraint : application to interpretation of over-segmented images. Artificial intelligence, vol. 93, no. 1, pages 321–335, 1997. (Cité en page 7.)
- [Deruyver 2009] Aline Deruyver, Yann Hodé et Luc Brun. Image interpretation with a conceptual graph : Labeling over-segmented images and detection of unexpected objects. Artificial Intelligence, vol. 173, no. 14, pages 1245–1265, 2009. (Cité en page 7.)
- [Douglas 1973] David H Douglas et Thomas K Peucker. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. Cartographica : The International Journal for Geographic Information and Geovisualization, vol. 10, no. 2, pages 112–122, 1973. (Cité en page 43.)
- [Eberhart 1998] Russell C Eberhart et Yuhui Shi. Comparison between genetic algorithms and particle swarm optimization. In International Conference on Evolutionary Programming, pages 611–616. Springer, 1998. (Cité en page 50.)
- [Flitton 2010] Gregory T Flitton, Toby P Breckon et Najla Megherbi Boualagu. Object Recognition using 3D SIFT in Complex CT Volumes. In BMVC, pages 1–12, 2010. (Cité en page 37.)

- [Fogel 1966] Lawrence J Fogel, Alvin J Owens et Michael J Walsh. Artificial intelligence through simulated evolution. 1966. (Cité en page 53.)
- [Fogel 1993] David B Fogel. Applying evolutionary programming to selected traveling salesman problems. Cybernetics and systems, vol. 24, no. 1, pages 27–36, 1993. (Cité en page 53.)
- [Forsen 1967] George E Forsen. Processing visual data with an automaton eye. Stanford Research Institute, 1967. (Cité en page 24.)
- [Francisci 2002] Dominique Francisci. Algorithmes évolutionnaires et optimisation multi-objectifs en data mining. 2002. (Cité en page 50.)
- [Garnesson 1992] Philippe Garnesson et Gerard Giraudon. L'Approximation polygonale : bilans et perspectives. PhD thesis, INRIA, 1992. (Cité en page 43.)
- [Geem 2001] Zong Woo Geem, Joong Hoon Kim et GV Loganathan. A new heuristic optimization algorithm : harmony search. Simulation, vol. 76, no. 2, pages 60–68, 2001. (Cité en pages 55 et 67.)
- [Geem 2007] Zong Woo Geem. Harmony search algorithm for solving sudoku. In International Conference on Knowledge-Based and Intelligent Information and Engineering Systems, pages 371–378. Springer, 2007. (Cité en pages 55 et 61.)
- [Geem 2010a] Zong Woo Geem. Research commentary : Survival of the fittest algorithm or the novelst algorithm ? International Journal of Applied Metaheuristic Computing (IJAMC), vol. 1, no. 4, pages 75–79, 2010. (Cité en page 61.)
- [Geem 2010b] Zong Woo Geem. State-of-the-art in the structure of harmony search algorithm. In Recent Advances In Harmony Search Algorithm, pages 1–10. Springer, 2010. (Cité en page 56.)

- [Goldberg 1989] David E Goldberg. Genetic algorithms in search, optimization, and machine learning, 1989. Reading : Addison-Wesley, 1989. (Cité en page 54.)
- [Gonzales 2002] Rafael C Gonzales et Richard Eugene Woods. Digital Image Processing. 2002. New Jersey : Prentice Hall, vol. 6, page 681, 2002. (Cité en pages 32 et 99.)
- [Guermeur 2003] Philippe Guermeur et Jean Louchet. An Evolutionary Algorithm for Camera Calibration. In Proceeding of International Conference on Robotics Distance Learning & Intelligence Communication Systems (ICRODIC'03), Rethymno, Greece, pages 799–804, 2003. (Cité en page 71.)
- [Gupta 2009] Om K Gupta et Ray A Jarvis. Using a Virtual World to Design a Simulation Platform for Vision and Robotic Systems. In International Symposium on Visual Computing, pages 233–242. Springer, 2009. (Cité en page 37.)
- [Harris 1988] Chris Harris et Mike Stephens. A combined corner and edge detector. In Alvey vision conference, volume 15, pages 10–5244. Manchester, UK, 1988. (Cité en page 71.)
- [Hartley 2003] Richard Hartley et Andrew Zisserman. Multiple view geometry in computer vision. Cambridge university press, 2003. (Cité en page 69.)
- [Heikkila 1997] Janne Heikkila et Olli Silvén. A four-step camera calibration procedure with implicit image correction. In Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on, pages 1106–1112. IEEE, 1997. (Cité en page 77.)
- [Henricsson 1998] Olof Henricsson. The role of color attributes and similarity grouping in 3-D building reconstruction. Computer vision and image understanding, vol. 72, no. 2, pages 163–184, 1998. (Cité en page 3.)

- [Hofmann 2001] PCRS Hofmann. Detecting urban features from IKONOS data using an object-oriented approach. In Proceedings of the first annual conference of the remote sensing & photogrammetry society, pages 79–91, 2001. (Cité en page 3.)
- [Holland 1975] John H Holland. Adaptation in natural and artificial systems. An introductory analysis with application to biology, control, and artificial intelligence. Ann Arbor, MI : University of Michigan Press, 1975. (Cité en page 53.)
- [Ikeuchi 1987] Katsushi Ikeuchi. Generating an interpretation tree from a CAD model for 3D-object recognition in bin-picking tasks. International Journal of Computer Vision, vol. 1, no. 2, pages 145–165, 1987. (Cité en pages 18 et 32.)
- [Izaguirre 1983] Alberto Izaguirre. Contribution à l'intégration de la vision dans la commande d'un robot. PhD thesis, 1983. (Cité en page 71.)
- [Jain 1989] Anil K Jain. Fundamentals of digital image processing. The Back-Projection Operator, vol. 445, 1989. (Cité en page 32.)
- [Juan 2009] Luo Juan et Oubong Gwun. A comparison of sift, pca-sift and surf. International Journal of Image Processing (IJIP), vol. 3, no. 4, pages 143–152, 2009. (Cité en pages xiii et 40.)
- [Ke 2004] Yan Ke et Rahul Sukthankar. PCA-SIFT : A more distinctive representation for local image descriptors. In Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on, volume 2, pages II–506. IEEE, 2004. (Cité en page 39.)
- [Keselman 2005] Yakov Keselman et Sven Dickinson. Generic model abstraction from examples. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 27, no. 7, pages 1141–1156, 2005. (Cité en page 7.)

- [Kim 2016] Byoung-Ik Kim, Jang-wook Choi, Kyung-jin Na et Soo-Young Ha. Calibration method Of Stereo Camera for Vehicle. In Proceedings of the International Conference on Image Processing, Computer Vision, and Pattern Recognition (IPCV), page 357. The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), 2016. (Cité en page 66.)
- [Koza 1992] John R Koza. Genetic programming : on the programming of computers by means of natural selection, volume 1. MIT press, 1992. (Cité en page 54.)
- [Koza 1994] John R Koza. Genetic programming II : Automatic discovery of reusable subprograms. Cambridge, MA, USA, 1994. (Cité en page 53.)
- [Kumar 2002] M Pawan Kumar, Saurabh Goyal, CV Jawahar et PJ Narayanan. Polygonal Approximation of Closed Curves across Multiple Views. In ICVGIP, 2002. (Cité en page 43.)
- [Kumar 2014] Rahul Kumar, Sanjesh Kumar, Sunil Lal et Praneel Chand. Object detection and recognition for a pick and place Robot. In Computer Science and Engineering (APWC on CSE), 2014 Asia-Pacific World Congress on, pages 1–7. IEEE, 2014. (Cité en page 32.)
- [Kurozumi 1982] Yoshisuke Kurozumi et Wayne A Davis. Polygonal approximation by the minimax method. Computer Graphics and Image Processing, vol. 19, no. 3, pages 248–264, 1982. (Cité en page 44.)
- [LeCun 1998] Yann LeCun, Léon Bottou, Yoshua Bengio et Patrick Haffner. Gradient-based learning applied to document recognition. Proceedings of the IEEE, vol. 86, no. 11, pages 2278–2324, 1998. (Cité en page 47.)
- [Lipe 1972] Lipeet al. Vibratory bowl feeder, Mars 21 1972. US Patent 3,650,367. (Cité en page 18.)

- [Lowe 1999] David G Lowe. Object recognition from local scale-invariant features. In Computer vision, 1999. The proceedings of the seventh IEEE international conference on, volume 2, pages 1150–1157. Ieee, 1999. (Cité en page 37.)
- [Lowe 2004] David G Lowe. Distinctive image features from scale-invariant keypoints. International journal of computer vision, vol. 60, no. 2, pages 91–110, 2004. (Cité en page 37.)
- [Mahdavi 2007] M Mahdavi, Mohammad Fesanghary et E Damangir. An improved harmony search algorithm for solving optimization problems. Applied mathematics and computation, vol. 188, no. 2, pages 1567–1579, 2007. (Cité en page 59.)
- [Masaki 1998] Ichiro Masaki. Machine-vision systems for intelligent transportation systems. IEEE Intelligent systems, vol. 13, no. 6, pages 24–31, 1998. (Cité en page 66.)
- [Masood 2008] Asif Masood. Dominant point detection by reverse polygonization of digital curves. Image and Vision Computing, vol. 26, no. 5, pages 702–715, 2008. (Cité en page 44.)
- [Mbikayi 2013] Herve Kabamba Mbikayi. Toward evolution strategies application in automatic polyphonic music transcription using electronic synthesis. arXiv preprint arXiv :1304.0969, 2013. (Cité en page 52.)
- [Messmer 1998] Bruno T Messmer et Horst Bunke. A new algorithm for error-tolerant subgraph isomorphism detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 20, no. 5, pages 493–504, 1998. (Cité en page 7.)
- [Moh'd Alia 2011] Osama Moh'd Alia et Rajeswari Mandava. The variants of the harmony search algorithm : an overview. Artificial Intelligence Review, vol. 36, no. 1, pages 49–68, 2011. (Cité en page 58.)

- [Murray 2000] Don Murray et James J Little. Using real-time stereo vision for mobile robot navigation. *Autonomous Robots*, vol. 8, no. 2, pages 161–171, 2000. (Cité en page 66.)
- [Naouai 2013] Mohamed Naouai. Localisation et reconstruction du réseau routier par vectorisation d'image THR et approximation des contraintes de type "NURBS". PhD thesis, Université de Strasbourg ; Université de Tunis-El Manar (Tunisie), 2013. (Cité en page 43.)
- [Ni 2015] Wei-Feng Ni, Shan-Chun Wei, Tao Lin et Shan-Ben Chen. A Self-calibration Algorithm with Chaos Particle Swarm Optimization for Autonomous Visual Guidance of Welding Robot. In *Robotic Welding, Intelligence and Automation*, pages 185–195. Springer, 2015. (Cité en page 66.)
- [Nikolaus 2013] Nikolaus. Evolution Strategies. 2013. (Cité en page 52.)
- [Omran 2008] Mahamed GH Omran et Mehrdad Mahdavi. Global-best harmony search. *Applied mathematics and computation*, vol. 198, no. 2, pages 643–656, 2008. (Cité en page 58.)
- [Padberg 2012] Miriam Padberg. Harmony Search Algorithms for binary optimization problems. In *Operations Research Proceedings 2011*, pages 343–348. Springer, 2012. (Cité en page 61.)
- [Pavlidis 1974] Theodosios Pavlidis et Steven L Horowitz. Segmentation of plane curves. *IEEE transactions on Computers*, vol. 23, no. 8, pages 860–870, 1974. (Cité en page 44.)
- [Perkins 1978] Walton A Perkins. A model-based vision system for industrial parts. *IEEE Transactions on Computers*, no. 2, pages 126–143, 1978. (Cité en page 24.)
- [Piccinini 2012] Paolo Piccinini, Andrea Prati et Rita Cucchiara. Real-time object detection and localization with SIFT-based clustering. *Image*

- and Vision Computing, vol. 30, no. 8, pages 573–587, 2012. (Cité en pages 11, 18, 32 et 41.)
- [Ramer 1972] Urs Ramer. An iterative procedure for the polygonal approximation of plane curves. Computer graphics and image processing, vol. 1, no. 3, pages 244–256, 1972. (Cité en page 44.)
- [Roberts 1963] Lawrence Gilman Roberts. Machine perception of three-dimensional solids. PhD thesis, Massachusetts Institute of Technology, 1963. (Cité en page 23.)
- [Rosin 1997] Paul L Rosin. Techniques for assessing polygonal approximations of curves. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 19, no. 6, pages 659–666, 1997. (Cité en page 43.)
- [Rupp 2006] Stephan Rupp, Matthias Elter, Michael Breitung, Walter Zink et Christian Kueblbeck. Robust camera calibration using discrete optimization. International Journal of Applied Science, vol. 13, pages 250–254, 2006. (Cité en page 67.)
- [Saka 2009] Mehmet Polat Saka. Optimum design of steel skeleton structures, pages 87–112. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009. (Cité en page 55.)
- [Schwarz 2015] Max Schwarz, Hannes Schulz et Sven Behnke. RGB-D object recognition and pose estimation based on pre-trained convolutional neural network features. In Robotics and Automation (ICRA), 2015 IEEE International Conference on, pages 1329–1335. IEEE, 2015. (Cité en page 47.)
- [Schwarz 2016] Max Schwarz, Anton Milan, Arul Selvam Periyasamy et Sven Behnke. RGB-D object detection and semantic segmentation for autonomous manipulation in clutter. The International Journal of Robotics Research, page 0278364917713117, 2016. (Cité en page 47.)

- [Schwefel 1965] H-P Schwefel. Kybernetische Evolution als Strategie der experimentellen Forschung in der Stromungsmechanik. Master's thesis, Technische Universitat Berlin, Hermann Fottinger Institut fuer Hydrodynamik, 1965. (Cité en page 52.)
- [Se 2001] Stephen Se, David Lowe et Jim Little. Vision-based mobile robot localization and mapping using scale-invariant features. In Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on, volume 2, pages 2051–2058. IEEE, 2001. (Cité en page 37.)
- [Sellaouti 2014] Aymen Sellaouti, Atef Hamouda, Aline Deruyver et Cedric Wemmert. Template-based hierarchical building extraction. IEEE Geoscience and Remote Sensing Letters, vol. 11, no. 3, pages 706–710, 2014. (Cité en page 3.)
- [Selvakumar 2013] Kalaivani Selvakumar et Bimal Kumar Ray. Survey on Polygonal Approximation Techniques for Digital Planar Curves. International Journal of Information Technology, Modeling and Computing (UITMC) Vol, vol. 1, 2013. (Cité en page 43.)
- [Shirai 1971] Yoshiaki Shirai et Motoi Suwa. Recognition of polyhedrons with a range finder. In IJCAI, pages 80–87, 1971. (Cité en page 24.)
- [Sobel 2014] Irwin Sobel. History and definition of the sobel operator. Retrieved from the World Wide Web, 2014. (Cité en page 32.)
- [Sorensen 2015] Kenneth Sorensen. Metaheuristics - the metaphor exposed. International Transactions in Operational Research, vol. 22, no. 1, pages 3–18, 2015. (Cité en page 61.)
- [Stein 1992] Fridtjof Stein et Gérard Medioni. Structural indexing : efficient 2D object recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 14, no. 12, pages 1198–1204, 1992. (Cité en page 43.)

- [Streichert 2002] Felix Streichert. Introduction to evolutionary algorithms. paper to be presented Apr, vol. 4, 2002. (Cité en page 51.)
- [Sturm 1999] Peter F Sturm et Stephen J Maybank. On plane-based camera calibration : A general algorithm, singularities, applications. In Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on., volume 1. IEEE, 1999. (Cité en page 77.)
- [Taherinejad 2009] Nima Taherinejad. Highly reliable harmony search algorithm. In Circuit Theory and Design, 2009. ECCTD 2009. European Conference on, pages 818–822. IEEE, 2009. (Cité en page 58.)
- [Tsai 1987] Roger Tsai. A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses. IEEE Journal on Robotics and Automation, vol. 3, no. 4, pages 323–344, 1987. (Cité en page 71.)
- [Tsuji 1975] Saburo Tsuji et Akira Nakamura. Recognition Of An Object in A Stack Of Industrial Parts. In IJCAI, pages 811–818, 1975. (Cité en page 24.)
- [Ulrich 2012] Markus Ulrich, Christian Wiedemann et Carsten Steger. Combining scale-space and similarity-based aspect graphs for fast 3D object recognition. IEEE transactions on pattern analysis and machine intelligence, vol. 34, no. 10, pages 1902–1914, 2012. (Cité en pages 11 et 32.)
- [Varady 1997] Tamas Varady, Ralph R Martin et Jordan Cox. Reverse engineering of geometric models : an introduction. Computer aided design, vol. 29, no. 4, pages 255–268, 1997. (Cité en page 27.)
- [Wang 2010] Chia-Ming Wang et Yin-Fu Huang. Self-adaptive harmony search algorithm for optimization. Expert Systems with Applications, vol. 37, no. 4, pages 2826–2837, 2010. (Cité en page 60.)

- [Wang 2015a] Xiaolei Wang, Xiao-Zhi Gao et Kai Zenger. An introduction to harmony search optimization method. Springer, 2015. (Cité en page 60.)
- [Wang 2015b] Xiaolei Wang, Xiao-Zhi Gao et Kai Zenger. The Overview of Harmony Search. In An Introduction to Harmony Search Optimization Method, pages 5–11. Springer, 2015. (Cité en page 61.)
- [Weyland 2012] Dennis Weyland. A rigorous analysis of the harmony search algorithm : how the research community can be. Modeling, Analysis, and Applications in Metaheuristic Computing : Advancements and Trends : Advancements and Trends, page 72, 2012. (Cité en page 61.)
- [Weyland 2015] Dennis Weyland. A critical analysis of the harmony search algorithm - How not to solve Sudoku. Operations Research Perspectives, vol. 2, pages 97–105, 2015. (Cité en page 61.)
- [Willaume 2016] Pierre Willaume, Pierre Parrend, Etienne Gancel et Aline Deruyver. The Graph Matching Optimization methodology for thin object recognition in pick and place tasks. In Computational Intelligence (SSCI), 2016 IEEE Symposium Series on, pages 1–8. IEEE, 2016. (Cité en pages 112, 113 et 139.)
- [Willaume 2017a] Pierre Willaume, Pierre Parrend, Etienne Gancel et Aline Deruyver. Optimisation of Pick and Place System by EvolutionaryAlgorithm to Facilitate its Implementation and its Use. In International Conference of Pattern Recognition Systems, pages 1–6, 2017. (Cité en pages 66 et 139.)
- [Willaume 2017b] Pierre Willaume, Pierre Parrend, Etienne Gancel et Aline Deruyver. Skeletonization and 3D Graph Approach for Thin Objects Recognition in Pick and Place Tasks. In IAPR International Conference on Machine Vision Applications, pages 1–4, 2017. (Cité en pages 95 et 139.)

- [Winkelbach 2002] Simon Winkelbach et Friedrich M Wahl. Shape from single stripe pattern illumination. In Joint Pattern Recognition Symposium, pages 240–247. Springer, 2002. (Cité en page 25.)
- [Yang 2008] Zhan-Long Yang et Bao-Long Guo. Image mosaic based on SIFT. In Intelligent Information Hiding and Multimedia Signal Processing, 2008. IHHMSP'08 International Conference on, pages 1422–1425. IEEE, 2008. (Cité en page 40.)
- [Yao 1996] Xin Yao et Yong Liu. Fast Evolutionary Programming. Evolutionary Programming, vol. 3, pages 451–460, 1996. (Cité en page 53.)
- [Yildiz 2010] Ali Rıza Yildiz et Ferruh Öztürk. Hybrid taguchi-harmony search approach for shape optimization. In Recent Advances In Harmony Search Algorithm, pages 89–98. Springer, 2010. (Cité en pages 55 et 58.)
- [Yin 2000] Peng-Yeng Yin. A tabu search approach to polygonal approximation of digital curves. International Journal of Pattern Recognition and Artificial Intelligence, vol. 14, no. 02, pages 243–255, 2000. (Cité en page 43.)
- [Zhang 1984] TY Zhang et Ching Y. Suen. A fast parallel algorithm for thinning digital patterns. Communications of the ACM, vol. 27, no. 3, pages 236–239, 1984. (Cité en page 99.)
- [Zhang 2000] Zhengyou Zhang. A flexible new technique for camera calibration. Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 22, no. 11, pages 1330–1334, 2000. (Cité en pages 11, 77, 82 et 113.)
- [Zhang 2009] Mengjie Zhang, Urvesh Bhowan et Bunna Ny. Genetic Programming for Object Detection : a Two-phase Approach with an Improved Fitness Function., 2009. (Cité en page 54.)

The Graph Matching Optimization Methodology for Thin Object Recognition in Pick and Place Tasks

Pierre Willaume^{*†‡§}, Pierre Parrend^{*†§}, Etienne Gancel[†]
and Aline Deruyver^{*§}

^{*}ICube Laboratory, Université de Strasbourg, France
4 Rue Blaise Pascal, 67081 Strasbourg, France

[†]ECAM Strasbourg-Europe, Schiltigheim, France
2 Rue de Madrid, 67300 Schiltigheim

[‡]Hager Group, Obernai, France

132 Boulevard de l'Europe, 67215 Obernai

[§]Complex System Digital Campus (UNESCO Unitwin) <http://unitwin-cs.org>
pierre.willaume@etu.unistra.fr

Abstract—Bin-picking emerges as a major interest in the industry. The aim is to replace current ‘pick and place’ systems, where one must place mechanical components in dedicated distribution devices such as bowl feeders for picking them up with a robot arm. A large number of image processing methods are available for recognizing these components. For instance, the stereovision approach provides fine results by comparing several images of the objects taken from different angles. However, when several types of components are available or for thin components, the identification remains a delicate task. We propose the Graph Matching Optimization methodology, which uses graph comparison with evolutionary algorithms between stereoscopic images and a model, in order to identify thin pieces in a constrained time frame. First, we extract characteristic component information by binarization and skeletonization of the images. Then, we retrieve the position of the objects in a 3 three-dimensional space through an evolutionary algorithm derived from Harmony Search Optimisation (HSO). Lastly, we extract and validate optimal parameter ranges for which the devised algorithm shows a high efficiency for representative component positions of randomly arranged thin objects.

I. INTRODUCTION

A topic of research for many years, ‘pick and place’ still generates a great interest especially in the industrialization domain [1]. It should enable robot arms to grasp one or several randomly placed objects and put them at a desired location. Many machines exist for simplifying the process, but currently no method permits to grasp any kind of components placed randomly in a given environment in three dimensions, *ie.* to support actual bin-picking. For example, bowl feeder requires a different device for each kind of objects, which makes the process expensive when assembling several components. Another method consists in taking the objects on a conveyor belt, but this system is cumbersome because the objects need to be preset on the conveyor. One of the current methods for recognizing and locating objects randomly arranged on top of each other is comparing the image of one or several cameras with a model of the object. The principal difficulty is the pick part because the system need to recognize and locate the pattern of the objects thanks to input images in



Fig. 1. Randomly placed objects

order to calculate the position in the space of the object (output). This system saves space, is cheaper and is adaptable to various object types. However, it requires a costly image pre-processing in order to recover the exact three dimensional position of the objects which can be hidden by others. As shown in figure 1, the aim is to recognize and locate a part of the desired object in order to pick it up. However, the objects can be entangled and do not necessarily stands out on the picture.

The purpose of this paper is to propose an algorithm for detecting randomly arranged thin objects and to optimize the time using an evolutionary algorithm. First, binarization of the image is performed on two stereoscopic images in order to highlight the target objects. Second, in order to highlight the curve of the components, a thinning algorithm is applied to the binary image. Third, the search of junction points and extremum point by graph theory is applied. Finally, an evolutionary algorithm of the Harmony Search family is defined for optimizing point matching between both images, and evaluated. This method allows to retrieve randomly arranged objects without a greedy pretreatment. Calibrated cameras allow to identify the position in the 3D space of the desired object and to calculate the length of the object.

This paper is organized as follows: Section II details the existing state-of-the-art of ‘pick and place’ methodologies. Section III presents Evolutionary Algorithms and introduces the Harmony Search algorithm family. Section IV presents the Graph Matching Optimization methodology. Section V describes experimental setup and results. Section VI discusses the advantages and the limitations of the proposed method. Section VII concludes this work.

II. IMAGE PROCESSING IN BIN-PICKING

Bin-picking is a method of choice for recovering disorganized objects [1] and continues to be the subject of significant research breakthroughs, for efficient storing of information [2] or for better discrimination of the component form using structure descriptors [3]. Different scientific fields such as vision, control technologies, information reduction [4] are required to support this process. The performance of the algorithms is usually calculated according to different criteria such as the setup time, which corresponds to the time to build the model, or the detection time, for actual location of the objects, according to the structure of the target components. The identification of an object having a particular structure requires a comparison model. Such a model can be represented by a CAD file, by an image, or by a set of descriptors or characteristics.

A. Detecting objects using key points

The Scale-Invariant Feature Transform (SIFT) method supports the extraction of objects in images [5] and can be leveraged for real-time location of these objects. It then looks for key points in the component images, and compare them with the different key points of the reference model to find similarities [6], as shown on figure 2. The key points are thus gathered using clustering in order to find the objects. Alternative solutions use derivation of this method such as Speeded Up Robust Features (SURF) [7] or PCA-SIFT [8] to improve computation time and the algorithm capability to find object with distortions. They allow to retrieve common graphic features that stand out, such as marking. The advantage of this approach is the flexibility to find hidden objects, but unfortunately it does not find thin objects because the key points do not stand out.

B. Detecting objects using feature extraction

Feature extraction enables to classify objects based on learnt characteristics of the images Kumar [9]. The process starts by converting the pixels into greyscale and then transforming the image into binary image. Edge detection is performed in order to highlight the objects shape. In order to preserve high value pixels, a double thresholding is performed. Edge tracking is also performed with hysteresis and followed by filling the dilated image. A blob analysis ends the algorithm, which allows to retrieve the components with a high contrast. However, and as in SIFT, the system is not suitable for the detection of thin objects because the threshold would not stand out the object to take.



Fig. 2. Sift Algorithm

C. Detecting objects using geometric models

Reference models are typically available in industrial environments as 3D CAO models. Such models are thus logically a standard reference for ‘pick and place’ tasks [10]. The 3D model of the object is generated for several orientations in order to retrieve the position of the object in the scene. The generated positions are classified according to the characteristics of the dominant visible faces of the object. This data is classified into an interpretation tree in order to optimize the search time, for instance using relationships between the neighbours. It is then compared to a set of CAD-model templates for retrieving the matching shapes and performing object identification. Recognition of the orientation and position of a 3D part is done using one [11] or several cameras, and a set of CAD files used for generating a hierarchical model of the object structure. These methods prove their efficiency in a large number of applications and are still used by companies. The advantage of the system is to have an easy setup which only requires a CAD file and a calibrated camera. However, the generation of the view of the CAD file in the preprocessing step is time-consuming because the system has to save a set of possible positions of the model.

Efficiency can be greatly improved by not using full images of CAD files as reference models, but abstracted models. Graph representation of object structure enable efficient processing, storage, and pre-treatment of the component image. We therefore propose to apply the graph matching approach to bin-picking for introducing a radical breakthrough in the performance of the object recognition process.

III. EVOLUTIONARY ALGORITHMS

A. Principles

Evolutionary algorithms [12] aim at creating an autonomous intelligence and to adapt its behaviour according to the circumstances [13]. It is inspired by natural selection, first exposed by Darwin [14]. These algorithms use metaphorical concepts inspired by natural evolution. It describes an environment where individuals:

- 1) Evolve
- 2) Survive
- 3) Reproduce

This metaphor implies an environment (*ie.* problem) where individuals (*ie.* candidate solutions) evolve until they build a satisfactory solution. A fitness threshold represents candidate’s adaptation to the environment.

Evolutionary Algorithms [15] have been used successfully in different domains like system optimization or machine

learning. Their global advantage is that there is a simple conceptualization of the evolutionary process by the equation:

$$x[t + 1] = s(v(x[t])) \quad (1)$$

where $x[t]$ is the population at time t , v is the random variation operator and s is the selection operator. It is interesting to mix different algorithms in order to improve the quality of solutions. The limitations are the lack of guarantee of finding the optimum solution, the expensive computing cost and the usual need for an expensive setting.

All types of Evolutionary Algorithms [16] entail several specific phases:

- Step 1: Randomly Initialize the Population
- Step 2: Evaluate all individuals from the population
- Step 3: Choose the best individuals from population to generate the next generation
- Step 4: Check termination criteria. Either the program stops or it continues to the step 2.

B. Harmony Search

Harmony Search (HS) [17] is a heuristic algorithm inspired by music whose aim is to find the best solution among value generated by an improvisation-like process. In the analogy, a solution is expressed as a harmony, and the value as the note played by the musicians. Its advantages are to be quick, easy to implement and with a fast convergence to quickly find a solution. These advantages are important in the industrial world. The results obtained by the Harmony search algorithm [18] applied to various optimisation problems makes it in any case a serious candidate for obtaining robust and efficient solutions. The algorithm 1 presents the structure of Harmony Search and is separated in five phases which are:

- Step 1: Formulate the problem and set the parameters
- Step 2: Randomly initialize the population
- Step 3: Generate a new solution
- Step 4: Generate the next generation
- Step 5: Check Stopping Criterion

1) *Step 1: Formulate the problem and set the parameters:* The Harmony Search algorithm [19] begins with problem formulation. It allows to define parameters such as the elements of satisfaction (fitness function) and time execution. To execute the algorithm, one has to set following parameters:

- Harmony Memory Size (HMS[1-X]) : Number of solutions handled simultaneously in the algorithm
- Harmony Memory Considering Rate (HMCR[0-1]) : Rate at which the algorithm picks one value randomly from the Harmony Memory
- Pitch Adjusting Rate (PAR[0-1]) : Rate at which the algorithm tweaks the value originally picked from memory

2) *Step 2: Randomly initialize the population:* It consists on randomly generated solutions. The number of solutions generated depends on the Harmony Memory Size (HMS). An example of six solutions is presented on Table I where each solution is scored and saved in the Harmony Memory HM.

TABLE I
RANDOMLY INITIALIZE THE POPULATION

Number	HM	Score
1	Solution 1	10
2	Solution 2	2
3	Solution 3	53
4	Solution 4	21
5	Solution 5	77
6	Solution 6	73

TABLE II
GENERATE A NEW SOLUTION

Solution	Score
S{X1,...Xij}	17

TABLE III
GENERATE THE NEXT GENERATION

Number	HM	Score
1	Solution 1	10
2	Solution 2	2
3	Solution 3	53
4	Solution 4	21
5	Solution 5	77
6	Solution 6	73
5	Solution 5	17

3) *Step 3: Generate a new solution:* In this part, and as shown on table II, a new solution is improvised. Each elements of the solution X_{ij} can be improved either randomly or thanks to the Harmony Memory (HM). A random value is generated and is compared with the parameter Harmony Memory Considering Rate (HMCR: $0 \leq HMCR \leq 1$) in order to determine whether the value is selected randomly or among the elements of the Harmony Memory (HM). If the value is selected among the Harmony Memory (HM), another random value is compared with the parameter named Pitch Adjusting Rate (PAR: $0 \leq PAR \leq 1$) in order to determine whether the value is mutated by selecting a value close (called neighbour) to that selected. The final result is a set of values which corresponds to a new solution.

4) *Step 4: Generate the next generation:* If the solution is better than the worst solution among the population, replace the element by the new one. As shown in table III, the fifth element in the Harmony Memory (HM) has the worst score with 77. This component is replaced by the new one which has a score inferior of 17.

5) *Step 5: Check Stopping Criterion:* Represented as *stop_alg()* on the algorithm 1, if one of the stopping criteria is matched (number of iterations, solution found), the algorithm returns the best solution among the element stored in the Harmony Memory. Otherwise the algorithm goes back to the third step.

Algorithm 1 Standard Harmony Search algorithm

Require: $HMS \in \mathbb{N}^+$, population Size
 $HMCR \in [0,1)$, Rate for choosing HM value
 $PAR \in [0,1)$, Rate for choosing neighbour value
Ensure: $s \in S$, S set of possible solutions
#INITIALIZATION#
 $HM \leftarrow \{s_0, s_1, \dots, s_{HMS-1}\}$
for stop_alg() \neq True **do**
 #CREATE A NEW ELEMENT#
 $s_{new} = \text{IMPROVISATION}(HM)$
 $s_{worst} \leftarrow s_i \in HM | f(s_i) < f(s_j) \forall s_j \in HM, s_j \neq s_i$
 #MEMORY UPDATE#
 if $f(s_{new}) < f(s_{worst})$ **then**
 REMOVE s_{worst} from HM
 REMOVE s_{worst} to HM
 end if
end for
 $s \leftarrow s_i \in HM | f(s_i) > f(s_j) \forall s_j \in HM, s_j \neq s_i$
return s

C. Variants of the Method

Different way for choosing parameter (HMS , $HMCR$, PAR ...) values has been proposed. Moreover, some methods have been proposed [20] [21] [22] for improving the results. An overview [23] of the proposed algorithms has been realised. A very important proposition has been done by Mahdavi [24] which propose to include a dynamic Pitch Adjusting Rate (PAR) by the equations:

$$PAR(gen) = PARmin + \frac{(PARmax - PARmin)}{NI} * gn \quad (2)$$

$$bw(gen) = bw_{max} \cdot e^{(c*gn)} \quad (3)$$

$$c = \frac{\ln(\frac{Bwmin}{Bwmax})}{NI} \quad (4)$$

Where :

- $PARmin/ PARmax$ are the minimum and maximum Pitch Adjusting Rate (PAR)
- NI is the number of solutions
- gn is the number of generation
- $Bwmin/ Bwmax$ are the minimum and maximum bandwidth

This method allows to find the optimum solution more quickly because the parameter PAR dynamically varies according to the number of iterations. Wang and Juang have proposed [25] another solution which replaces and updates the parameters PAR and bw (distance bandwidth) according to the maximal and the minimal values in HM . The equations are:

$$trial^i + [max(HM^i) - trial^i] \times ran[0, 1) \quad (5)$$

$$trial^i - [trial^i - min(HM^i)] \times ran[0, 1) \quad (6)$$

Where:

- $ran[0, 1) \leftarrow$ random number $\in [0, 1)$
- $min(HM^i) \leftarrow$ lowest values of the i^{th} variable in Harmony Memory (HM)
- $max(HM^i) \leftarrow$ highest values of the i^{th} variable in Harmony Memory (HM)
- $trial \leftarrow$ selected pitch from Harmony Memory (HM)

Thanks to this method, the search of solution is faster because the parameters bw and PAR are increasing and decreasing according to the worst and the best candidates of the population.

IV. THE GRAPH MATCHING OPTIMIZATION METHODOLOGY

This paper proposes an approach based on 3D graph matching using a Harmony Search algorithm in order to find an object from a set of randomly placed overlapping mechanical pieces. Graphs are extracted from the skeletons of objects in the stereo images of the scene and matched with the characteristic graph of the object model. They are used to recover the position of the object in 3 dimensions. The block diagram shown in figure 3 presents the different steps for recovering the object. The first step is the binarization of the image to highlight the desired object. The second step is the application of the thinning algorithm to retrieve the curves of the skeleton of the binarized images. The third step is the graph extraction in order to retrieve the end points and the junction points to match between stereo images. The search of an object is applied in the step four thanks to the evolutionary algorithm in order to reduce the duration of graph matching. This method will provide an optimal solution for the search of objects without matching all nodes. Calibrated cameras are taken into account [26]. This section is organized as followed: Part 1 presents the binarization of the image. Part 2 details the thinning algorithm. Part 3 describes the graph extraction method. Part 4 explains the objects retrieval method by using Harmony Search algorithm.

A. Binarization

Binarization (bi-level or two-level) is the process for converting a pixel image into a binary image, *ie.* an image with only two possible values (colours) for each pixel. The two colors used are typically the white and the black. The aim is to highlight the objects and to remove the background of the scene. The pixel can be stored in a single bit which is 0 or 1. Algorithm 2 presents the process of binarization. The color of all pixels of the image are compared with a model. A color is represented and is encoded by a system called RGB corresponding to the three primary colors which are Red, Green and Blue. The values are usually between 0 and 255. If the RGB present in the pixel is outside the range transmitted by the model, then the pixel is set to 0. Otherwise, the pixel is set at 1. As shown in figure 5, the binarization of the figure 4 with a threshold set between 100 and 255 brings out the desired object by setting all these pixels in black and all the background in white.

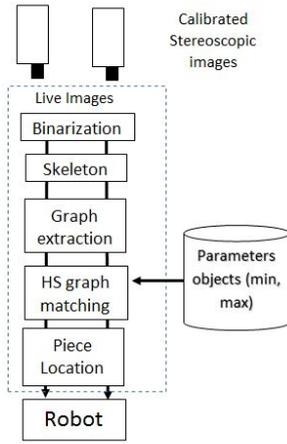


Fig. 3. Diagram of the different steps to find the objects

Algorithm 2 Binarization of the image

Require: *Image* and $Color_{min}$ and $Color_{max}$

#ITERATE THROUGH THE PIXELS#

for int $x=0$ to $x=Image.weight$ do

 for int $y=0$ to $y=Image.height$ do

 #COMPARE WITH THE THRESHOLD#

 if $Color_{min} < Image.Pixel.color(x,y) > Color_{max}$

 then

 #ASSIGN NEW VALUE#

 Image.Pixel.color(x,y) = white

 else

 Image.Pixel.color(x,y) = black

 end if

 end for

end for

B. Skeletonization

The algorithm of skeletonization [27] aims at representing the structural shape of a plane region by reducing the pattern to a set of curves named skeleton which are centred on the original shape. Topologies properties kept from the original shape are an advantage. Applying the thinning algorithm permits to let thin objects stand out as shown on figure 6. Performance improvement can be achieved by deleting all the points of the image edge with the exception of the skeleton points [28]. The iterations are divided in subiteration to preserve the connectivity of the skeleton. This part which is the foundation of the work is the most important because a poor skeletonization does not allow to find the links between different parts of the right image and the left image.

C. Graph transform: Extraction of the reference points

In image processing, a graph is a tool to represent an image by a component assembly with a set of relationships [29]. Graph are defined by a two-tuple $G=(V,E)$ where:

- V (nodes) \rightarrow finite set of vertices $V = \{v_1, v_2, v_3 \dots v_n\}$ ($n=Card(V)$)



Fig. 4. Input image

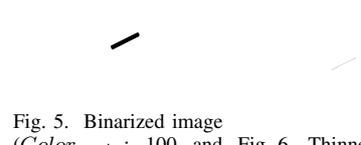


Fig. 5. Binarized image ($Color_{min}$: 100 and Fig. 6. Thinned image $Color_{max}$: 255)

- $E \rightarrow$ set of Edges $E = \{e_1, e_2, e_3 \dots e_m\}$ ($m=Card(E)$)

In our case, the graph represents the information of the set of shapes of the skeleton image where:

- Nodes \rightarrow endpoints and connection points of each part of the skeleton
- Edges \rightarrow link between the nodes

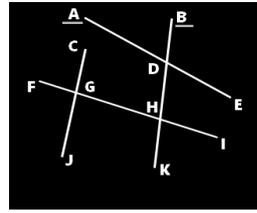


Fig. 7. Graph transform of left image: A and B are selected for object retrieval

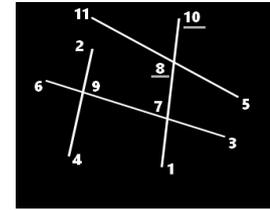


Fig. 8. Graph transform of right image: 8 and 10 are selected for object retrieval

The figures 7 and 8 show the nodes (endpoints, connection points). The recovery of the points is performed thanks to an algorithm that goes through each pixel and compares all pixel and neighbour with a model. The mapping will return whether or not the pixel belongs to a connection point, an endpoint or standard points in the image. This step allows to retrieve the nodes of the skeleton in order to do the graph matching and to search for an object in the next section.

D. Harmony Search for object retrieval

The extraction step of the object is applied by retrieving the known characteristics of the object. A link between nodes on the right and left image allows to calculate the 3D coordinates of the object and to allow the robot to pick the object. Harmony Search algorithm is applied to calculate the function fitness in order to find the object by matching nodes and by computing the distance between them. The fitness function is calculated according to decision variables which are:

- the equations of the euclidian distance in 2D and 3D :

$$3DS = \sqrt{(X_i - X_j)^2 + (Y_i - Y_j)^2 + (Z_i - Z_j)^2} \quad (7)$$

$$2DS = \sqrt{(X_i - X_j)^2 + (Y_i - Y_j)^2} \quad (8)$$

- the percentage of pixels of 3DS belonging to the setted binarized part

The algorithm 3 presents the different steps to follow to find the object. The algorithm needs the data input of HMS , $HMCR$, PAR and the distances of the different nodes

to match. The initialization is established according to the parameter HMS . For each iteration, a number of nodes is selected for the left and right image. A random matching of the node is performed in order to process the distance in two and three dimension. The figures 7 and 8 shows a random selection of the nodes to match (A and B for the left image and 8 and 10 for the right image). If the points between the right image and the left image does not match, then a penalty is added to the fitness function. It is defined according to the presence of the object between the nodes and according to the difference between experimental and theoretical distance in two and three dimensions. After the initialization and while the stop criteria are not met, a new solution is created. The selection of the nodes depends on the $HMCR$ and PAR parameters. For each node to match between both images, either both nodes are selected randomly or by selecting two nodes matched of the population HM . The PAR variable permits to interchange nodes already match in HM by those selected. All the nodes matched between the right image and the left create a new solution. If the new solution is better than the worst in the Harmony Memory (HM), then the worst object proposition is replaced by the new one. The stop criterion corresponds either to an object found according to the fitness function or to no object found in the required time.

V. EXPERIMENT

The experimentation has been realised with an Intel(R) Core(TM) i7-5700HQ, CPU 2.70GHz. The OpenCV library has been used to calibrate both webcams of 640x480 pixels. Several cases were studied for the search of objects. The use of the 3D position of the points to compute the distance between the nodes also allows the determination of the coordinates to provide to the robot so that it can grasp the object. The figure 9 and the figure 10 show the representation of the solution of Harmony Search. The detected object allows to retrieve the information of location and tilt in 3D space to give to the robot in order to take the object. The figure 9 shows the search of several thin objects, separated from each other on the same plane. The segment corresponds to the segments between both nodes. The text corresponds to the fitness function retrieved by the algorithm. This value is set according to the kind of skeleton and its environment needs to be under a threshold in order to be identified as an object. The figure 10 represents the search of a piece in context of cluttering and overlapping. In spite of the brightness of the pieces which tangle the ones on the others, the components are properly recovered. This image shows the advantage of the method to find objects that are not fixed in a same three dimensional space as could be bin-picking. The figure 11 represents the time needed by the algorithm to retrieve the object according to the number of nodes. The proposed method has been compare with a standard algorithm where each node is selected randomly. The higher the number of nodes, the longer the processing time. A minimum of 8 nodes was considered to avoid easy search test cases so that the object is not found in the initialization part. The algorithm was applied

Algorithm 3 Harmony Search algorithm for object retrieval

Require: $HMS \in \mathbb{N}^+$, population Size
 $HMCR \in [0,1)$, Rate for choosing HM value
 $PAR \in [0,1)$, Rate for choosing neighbour value
 $V(G1), V(G2) \leftarrow$ Set of nodes of the graph G1 and G2 corresponding to the left and the right images
Ensure: $s \leftarrow \{V_0, d_1, \dots, d_n\} \in S$, S set of possible solutions and n is a node belonging to an image
node $v[m] \in V(G1), w[m] \in V(G2)$, where m is the number of nodes to match between stereoscopic images
#INITIALIZATION#
for $i = 0$ **to** HMS **do**
 $v[m] \leftarrow$ random nodes $\in V(G1)$
 $w[m] \leftarrow$ random nodes $\in V(G2)$
 $HM[i] \leftarrow M[v[m], w[m]]$, Population HM where M is the set of nodes
end for
#IMPROVISATION#
while stop_alg() \neq True **do**
 $s_{new} \leftarrow M[v[m], w[m]]$, Population HM where M is the set of nodes
 $x_i \leftarrow M[v[0], w[0]]$, Set of two nodes v and w
 for $i = 0$ **to** m **do**
 if $random[0,1) \leq HMCR$ **then**
 $x_i \leftarrow$ Random $M[v[i], w[i]]$ from HM
 if $random[0,1) \leq PAR$ **then**
 #Inversion of a node compared to a previous#
 $tempon \leftarrow v[i-1]$
 $v[i-1] \leftarrow v[i]$
 $v[i] \leftarrow tempon$
 $x_i \leftarrow M[v[i], w[i]]$
 end if
 else
 $v[i] \leftarrow$ random node $\in V(G1)$
 $w[i] \leftarrow$ random node $\in V(G2)$
 $x_i \leftarrow M[v[i], w[i]]$
 end if
 end for
 $s_{new} \leftarrow X\{0, 1, \dots, i\}$
 $s_{worst} \leftarrow s_i \in HM | f(s_i) < f(s_j) \forall s_j \in HM, s_j \neq s_i$
 #MEMORY UPDATE#
 if $f(s_{new}) < f(s_{worst})$ **then**
 REMOVE s_{worst} from HM
 REMOVE s_{new} to HM
 end if
end while
 $s \leftarrow s_i \in HM | f(s_i) > f(s_j) \forall s_j \in HM, s_j \neq s_i$

100 times for each parameter change in order to recover a consistent average. An iteration takes about 1 millisecond. The pretreatment part (binarization, skeletonization) is not taken into account in the calculation time to avoid disrupting the time processing of harmony search part. The treatment time depends on several factors, in particular the parametrization of the search algorithm. In this case, the *HMS* parameter has been set to 20, *HMCR* has been set at 0.7 and *PAR* at 0.3. As shown in figure 12, the *HMS* parameter plays an important role in the search of objects. Figure 12 represents the number of iterations as a function of the *HMS* parameter and the number of nodes present in the image. In our case, *HMS* has to be set in the range of 20 to 70. If the number is lower, the number of iterations increases dramatically and the algorithm behaves randomly. The figure also shows that after the value 70, the curve of the number of iteration increases ever more. As demonstrated with the time curve (figure 11), the number of nodes impacts on the required number of iterations. *HMCR* and *PAR* settings play also an important role in improvisation of the search of the object. As shows the figure 13, the parameter *HMCR* need to be set between 0.5 and 0.8 and *PAR* between 0.1 and 0.3 to be in the optimal search configuration. The test has been applied in order to retrieve a component of two nodes in an image of 14 nodes. The *HMS* parameters has been set to 20.



Fig. 9. Search of objects separated the ones from the others. Blue: Segment between the nodes. Red: Fitness function of the object.



Fig. 10. Search of objects placed randomly with overlapping. Blue: Segment between the nodes. Red: Fitness function of the object.

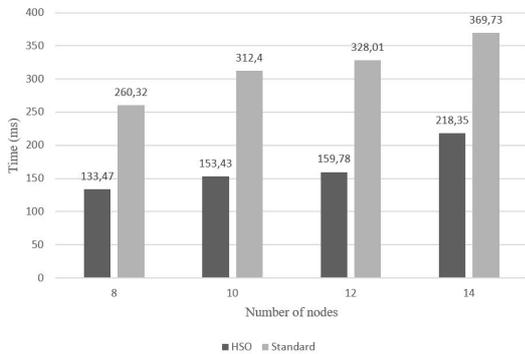


Fig. 11. Comparison of time algorithm between standard and HSO algorithm according to the number of nodes for objects with 2 nodes

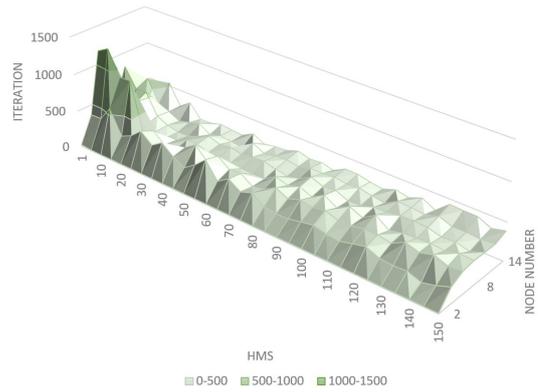


Fig. 12. Performance according to the number of nodes and the *HMS* parameter

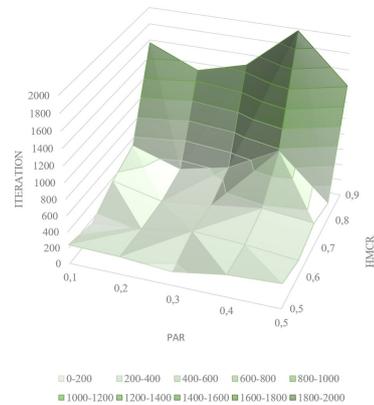


Fig. 13. Harmony Search performance according to the *PAR* and *HMCR* parameters

VI. DISCUSSION

The experimental part proves the efficiency of the proposed method to find thin components placed in a cluttered set. Binarization allows to stand out the objects in the scene. The thinning algorithm allows to retrieve the curve of each component. The graphs allow to retrieve the junction and extrema points. The Harmony Search allows to retrieve the object matching a reference object model. Parameterization plays an important role in optimizing the search for the object. *HMS* setting should not be too low because research is then very close to a random search. A population too high makes the algorithm less efficient because the system takes time to optimize its search. The efficiency has been demonstrated by the capability to the method to find the object in a time cut off by at least 40% wrt. the process without HSO. The second advantage is the efficiency in setting up the system which is lightweight because only distance needs to be calibrated. The difficulty depends on the number on nodes on the object. The matching duration depends on the number of nodes. However, the actual search duration can be considered as acceptable for operational use for bin-picking in a production line. The method proves its efficiency in finding

thin objects with little nodes. In a future work, an evaluation of the performance on different situation will be studied. The next step of improvement is to adapt the algorithm for any kind of objects and to perform tests with objects with many nodes. The system can be adapted by comparing the shape of the skeleton as well as the shapes of the object using a polygonal approach, which defines the curves between the different nodes of the graph. Comparing the graphs would then allow to compute the fitness function between the model and the images collected by the cameras.

VII. CONCLUSION AND PERSPECTIVES

This paper proposes a method to recognize thin objects randomly disposed in a 3D space in a constrained time frame by using stereo images and an evolutionary algorithm. First a set of filters is applied to each two-dimensional images in order to retrieve the location of the object on the scene. This simplification allows to calculate the size of the objects based on their location in 3D space. For each object on the image, a skeleton is built and Harmony search algorithm is applied to match the junction points and the extremum points. Points are randomly matched in two and three dimensions. The three dimensions enable the graph matching between nodes and the use of Harmony Search algorithm significantly improves its performance. The process has been tested with different objects placed close to each other, with cluttering and overlapping. All objects have been found. The algorithm works with different kinds of noises like brightness and luminosity. The system is quickly initialized because the parameters of the objects are known by the operator. The perspectives of our work is to adapt the system by analysing the edges objects by using a polygonal approximation.

ACKNOWLEDGMENT

We thank Icube laboratory for assistance with methodology and for comments that greatly improved the manuscript. This research was supported by Hager enterprise. We thank our colleagues who provided a great expertise and a huge insight that greatly assisted the research. We would also like to show our gratitude to the Ecam-Strasbourg Europe for sharing their pearls of wisdom with us during the course of this research.

REFERENCES

- [1] D. Buchholz, *Bin-Picking: New Approaches for a Classical Problem*. Cham: Springer International Publishing, 2016, ch. Bin-Picking—5 Decades of Research, pp. 3–12.
- [2] —, “Depth map based pose estimation,” in *Bin-Picking*. Springer, 2016, pp. 39–56.
- [3] C. M. Mateo, P. Gil, and F. Torres, “Visual perception for the 3d recognition of geometric pieces in robotic manipulation,” *The International Journal of Advanced Manufacturing Technology*, vol. 83, no. 9, pp. 1999–2013, 2016.
- [4] P. J. Sanz, A. P. Del Pobil, J. M. Inesta, and G. Recatala, “Vision-guided grasping of unknown objects for service robots,” in *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, vol. 4. IEEE, 1998, pp. 3018–3025.
- [5] D. G. Lowe, “Object recognition from local scale-invariant features,” in *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, vol. 2. IEEE, 1999, pp. 1150–1157.
- [6] P. Piccinini, A. Prati, and R. Cucchiara, “Real-time object detection and localization with sift-based clustering,” *Image and Vision Computing*, vol. 30, no. 8, pp. 573–587, 2012.
- [7] H. Bay, T. Tuytelaars, and L. Van Gool, “Surf: Speeded up robust features,” in *Computer vision—ECCV 2006*. Springer, 2006, pp. 404–417.
- [8] Y. Ke and R. Sukthankar, “Pca-sift: A more distinctive representation for local image descriptors,” in *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, vol. 2. IEEE, 2004, pp. II–506.
- [9] R. Kumar, S. Kumar, S. Lal, and P. Chand, “Object detection and recognition for a pick and place robot,” in *Computer Science and Engineering (APWC on CSE), 2014 Asia-Pacific World Congress on*. IEEE, 2014, pp. 1–7.
- [10] K. Ikeuchi, “Generating an interpretation tree from a cad model for 3d-object recognition in bin-picking tasks,” *International Journal of Computer Vision*, vol. 1, no. 2, pp. 145–165, 1987.
- [11] M. Ulrich, C. Wiedemann, and C. Steger, “Combining scale-space and similarity-based aspect graphs for fast 3d object recognition,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, no. 10, pp. 1902–1914, 2012.
- [12] P. J. Angelino, “A historical perspective on the evolution of executable structures,” *Fundamenta Informaticae*, vol. 35, no. 1-4, pp. 179–195, 1998.
- [13] R. C. Eberhart and Y. Shi, “Comparison between genetic algorithms and particle swarm optimization,” in *International Conference on Evolutionary Programming*. Springer, 1998, pp. 611–616.
- [14] C. Darwin, *The origin of species*. Lulu. com, 1872.
- [15] N. M. Al-salami, “Evolutionary algorithm definition,” *American J. of Engineering and Applied Sciences*, vol. 2, no. 4, pp. 789–795, 2009.
- [16] F. Streichert, “Introduction to evolutionary algorithms,” *paper to be presented Apr*, vol. 4, 2002.
- [17] Z. W. Geem, J. H. Kim, and G. Loganathan, “A new heuristic optimization algorithm: harmony search,” *Simulation*, vol. 76, no. 2, pp. 60–68, 2001.
- [18] Z. W. Geem, “Research commentary: Survival of the fittest algorithm or the novelst algorithm?” *International Journal of Applied Metaheuristic Computing (IJAMC)*, vol. 1, no. 4, pp. 75–79, 2010.
- [19] —, “State-of-the-art in the structure of harmony search algorithm,” in *Recent Advances In Harmony Search Algorithm*. Springer, 2010, pp. 1–10.
- [20] M. G. Omran and M. Mahdavi, “Global-best harmony search,” *Applied mathematics and computation*, vol. 198, no. 2, pp. 643–656, 2008.
- [21] N. Taherinejad, “Highly reliable harmony search algorithm,” in *Circuit Theory and Design, 2009. ECCTD 2009. European Conference on*. IEEE, 2009, pp. 818–822.
- [22] A. R. Yildiz and F. Öztürk, “Hybrid taguchi-harmony search approach for shape optimization,” in *Recent Advances In Harmony Search Algorithm*. Springer, 2010, pp. 89–98.
- [23] O. Moh’d Alia and R. Mandava, “The variants of the harmony search algorithm: an overview,” *Artificial Intelligence Review*, vol. 36, no. 1, pp. 49–68, 2011.
- [24] M. Mahdavi, M. Fesanghary, and E. Damangir, “An improved harmony search algorithm for solving optimization problems,” *Applied mathematics and computation*, vol. 188, no. 2, pp. 1567–1579, 2007.
- [25] C.-M. Wang and Y.-F. Huang, “Self-adaptive harmony search algorithm for optimization,” *Expert Systems with Applications*, vol. 37, no. 4, pp. 2826–2837, 2010.
- [26] Z. Zhang, “A flexible new technique for camera calibration,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 22, no. 11, pp. 1330–1334, 2000.
- [27] R. C. Gonzales and R. E. Woods, “Digital image processing. 2002,” *New Jersey: Prentice Hall*, vol. 6, p. 681, 2002.
- [28] T. Zhang and C. Y. Suen, “A fast parallel algorithm for thinning digital patterns,” *Communications of the ACM*, vol. 27, no. 3, pp. 236–239, 1984.
- [29] D. Conte, P. Foggia, C. Sansone, and M. Vento, “Thirty years of graph matching in pattern recognition,” *International journal of pattern recognition and artificial intelligence*, vol. 18, no. 03, pp. 265–298, 2004.

Optimisation of Pick and Place System by Evolutionary Algorithm to Facilitate its Implementation and its Use

Pierre Willaume, Pierre Parrend, Etienne Gancel, Aline Deruyver

ICube Laboratory, Université de Strasbourg, France
pierre.willaume@ecam-strasbourg.eu

Keywords: Industrial vision, optimization, evolutionary algorithm, Harmony Search

Abstract

Pick and Place is an important part of industrial processing. For example, it allows the assembly of elements by a robot. These systems working with cameras and a robot must be adaptable to the type of objects to identify as well as easy to use. An operator is not expected to have the skills to set and to use pick and place machines. In this paper, we propose an easily usable algorithm called HSP², Harmony Search for Pick and Place, to simplify bin-picking by using an evolutionary algorithm to set up cameras and identify the location of thin objects inside a scene so as to send their coordinates to the robot. The algorithm has been tested on camera calibration and the detection of thin objects.

1 Introduction

Globalization is increasingly present in our daily life. Industries need to automate their production lines to stay competitive. Moreover, the customer wants to quickly get new products which are more and more customized: this requires a great flexibility of the machine. However, a technician cannot change the machine program at each reference change. There is a strong need for the system to be adaptable. One of the most effective ways for assembling the components is to use a feeder system called bin-picking which performs pick and place tasks in order to position the elements with a robot. One method is to identify and locate the components by using one or several cameras and to send the coordinates to a robot. These very useful systems could be more suited to the operator who is not necessarily knowledgeable in pick and place operations. Several fields are required to maintain the system alive and adaptable. For example, to do stereovision, the system needs to be calibrated by using images of a pattern shot by both cameras. The detection of the pattern by the system enables finding its intrinsic and extrinsic parameters. The selection of the images determines the fitness function and the accuracy. However, an operator is not expected to have the skills to select the image adequately. Another example is the duration of

the identification of an object inside a scene. An operator should not need to have the skills to adapt an algorithm like SIFT to identify the kind of object that he wants to pick. In this paper, we propose an evolutionary algorithm which aims at simplifying the pick and place tasks related to the operator. The proposed algorithm called HSP² (Harmony Search for Pick and Place) allows to have an algorithm which is easy and efficient to use. It is based on the Harmony Search approach. The aim of this paper is to have an easily usable and installable system of pick and place to locate objects in a scene. We propose to use this algorithm to simplify both the calibration and the detection of thin objects in a scene. There are many advantages. Firstly, it allows the reduction of the workload of the operator. Secondly, it allows the simplification and the improvement of the system of pick and place by using a method of optimisation. Thirdly, it allows to have an adaptable system which can be easily usable from the operator point of view. Finally, it allows the optimization of the search time below the period of the robot production line. This algorithm is easy to understand and to set up: it is a good way to simplify operation and maintenance work. The optimization allows to quickly retrieve a coherent solution for both addressed problems. This paper is organized as follows: The section 2 describes the current state of the art of the setup and the detection of objects in pick and place. The section 3 details the Harmony Search algorithm. The section 4 presents HSP². The section 5 presents the experimentation setup and results. The section 6 discusses the advantages and the limitations of the proposed method. The section 7 states the conclusions.

2 State of the Art

The field of pick and place has been studied for several years [4] and still arouses a lot of interest. It allows the simplification of the tasks of an operator. The fields of cameras calibration, objects recognition or the way to grip the object are in constant improvement. This paper is focused on performance optimization for the calibration of the system and the detection of objects.

2.1 Stereo calibration

Camera calibration [6] has the aim to match the coordinate systems of the cameras with the real world. Firstly, it moves the origin of the world coordinates to the origin of the coordinate system of the camera. Secondly, the goal is to turn the world coordinate system until it coincides with that of the coordinates of the camera. Finally, the aim is to move the image plane sideways so that it goes in agreement with both coordinate systems. The ideal settings [10] of a camera pinhole is divided between intrinsic and extrinsic parameters. The extrinsic parameters represent the location and rotation of the camera in space. The intrinsic parameters refer to the internal model of the camera. The internal settings of a camera are the horizontal focal length, the vertical focal length and the scale factors. A method of calibration has been proposed by Zhang [26] in 2000 to easily calibrate cameras. The method requires several pictures of a planar pattern in different orientations. After retrieving the coordinates of the pattern, the system models the radial lens and the distortion. The method consists in having a closed-form solution and a nonlinear refinement based on the maximum likelihood criterion. The system can be improved upstream in order to simplify the calibration made by the operator. The aim is to let the machine decide the best image configuration in order to retrieve the best fitness function.

2.2 Thin object recognition

After calibrating, the system needs to identify the object by using a model. For example, the SIFT (Scale-Invariant Feature Transform) algorithm allows the extraction of objects in images [14]. The method detects key points in the component images, and then compares them with the set of key points retrieved by the reference model to find some similarities [18]. A clustering is used to find the objects on the scene. Once the object is detected, the system can calculate the coordinates to provide to the robot. Several alternative solutions have been proposed in the literature like PCA-SIFT [12] and SURF (Speeded Up Robust Features) [3] to improve capability to find distorted objects and to reduce the computation time. These algorithms of key points extraction are flexible and allows to find hidden objects. Another kind of object detection is used on pick and place by extracting the feature based on learnt characteristics of the images [13]. The algorithm converts the image into a binary image. Then, the detection of the edges is performed to highlight the shape of the objects. A double thresholding is performed to preserve high value pixels. Then, an edge tracker is performed by saving the edges with higher threshold values and by filling the dilated image. Finally, the analysis of the blob ends the algorithm to retrieve the components with a high contrast. However, these systems can be difficult to set up and cannot be adapted to the detection

of thin objects. In industries, the use of 3D CAD models can help the operator. To retrieve the position of the object, a 3D model [11] of the object is used. The different positions are then classified according to the dominant characteristics of the visible faces. An interpretation tree is used for classification in order to optimize the search time. To identify the objects, the image is compared to the set of CAD-model to retrieve the matching shapes. A solution has been proposed by using only one camera for recognition of the position and orientation [22] and a set of CAD files. The system has the advantage of having an easy setup by only requiring a CAD file and a calibrated camera which simplifies the operations. However, the pre-processing time is important because the system must generate the set of possible positions and the operator needs a CAD file. More and more solutions are proposed using evolutionary algorithms in order to solve optimization problems [2, 20].

3 Harmony Search

Inspired by Darwin [5], evolutionary algorithms [2, 20] allow to create autonomous intelligence [7]. A metaphorical concept inspired by natural evolution is used where individuals evolve, survive and reproduce. The success of evolutionary algorithms [1] is widely recognized by the variety of application domains. One of them, Harmony Search (HS) [9] is a heuristic algorithm which is inspired by music whose goal is to retrieve the best solution among value generated by an improvisation-like process. The specificity and the advantages of this approach are: 1) to be easy to implement because the algorithm has only 3 parameters; 2) to be quick; 3) to have a fast convergence to find a solution. These benefits are the most important in industry where a machine needs to be focused on the production. The results obtained by the different propositions [8] prove that HSO is an efficient candidate to use in industrial systems. We detail this approach in algorithm 1 which is composed of three steps:

3.0.1 Step 1: Randomly initialize the population

The harmony Search algorithm [8] starts by defining the different parameters such as the fitness function of the algorithm and the processing time. Then the values of the following parameters are defined. Firstly, the Harmony Memory Size ($HMS \in \mathbb{N}^+$) which is the number of solutions handled in the population. Then, the parameter Harmony Memory Considering Rate ($HMCR \in [0,1)$) which is the rate at which the algorithm selects a value randomly from the Harmony Memory. Finally, the Pitch Adjusting Rate ($PAR \in [0,1)$) which is the rate at which the algorithm tweaks the value. Once the set of parameters is defined, the next step consists of randomly generating new solutions. The parameter HMS defines the number of solutions to generate.

3.0.2 Step 2: Generate a new solution

To generate a new solution, each element of the current solution can be improved either by using the Harmony Memory (HM) or randomly. A random value is compared with the parameter $HMCR$ to determine whether the value is selected among the elements of the Harmony Memory (HM) or randomly. If the value is selected among HM, then another random value is generated and compared with PAR to determine whether the value is mutated by selecting a value in the neighborhood of the selected one. The set of values is the new solution.

3.0.3 Step 3: Generate the next generation

If the newly generated solution has a better fitness function than the worst solution among the population, then the worst solution is replaced by the new one. Represented as $stop_alg()$ on the algorithm 1, if one of the stopping criteria is satisfied (number of iterations, solution under the fitness function) then the algorithm returns the best solution. Otherwise the algorithm returns to the step 2.

Algorithm 1 Standard Harmony Search algorithm [24]

Require: $HMS \in \mathbb{N}^+$, cardinality of the population
 $HMCR \in [0,1)$, Rate for choosing an element from HM
 $PAR \in [0,1)$, Rate for choosing neighbour value from HM
 $HM \leftarrow \{s_1, s_2, \dots, s_{HMS}\}$
for $stop_alg() \neq \text{True}$ **do**
 $s_{new} = \text{IMPROVISATION}(HM)$
 if $f(s_{new}) < (f(s_{worst}) \text{ from HM})$ **then**
 REMOVE s_{worst} from HM
 $HM \leftarrow s_{new}$
 end if
end for
return s_{best} from HM

3.1 Variants of the Method

Different approaches for choosing the value of the parameters ($HMS, HMCR, PAR, \dots$) have been proposed. Several methods have been proposed [17, 21, 25] to improve the search performances of the algorithm. An overview [16] of the different proposed algorithms is proposed. For example, Mahdavi has proposed [15] to include a dynamic PAR with the following equations:

$$PAR(gen) = PAR_{min} + \frac{(PAR_{max} - PAR_{min})}{NI} * gn \quad (1)$$

$$bw(gn) = bw_{max} \cdot e^{(c * gn)} \quad (2)$$

$$c = \frac{\ln(\frac{Bw_{min}}{Bw_{max}})}{NI} \quad (3)$$

Where PAR_{min} and PAR_{max} corresponds respectively to the minimum and the maximum Pitch Adjusting Rate, NI represents the number of solutions, gn is the generation number and Bw_{min}/Bw_{max} correspond to the minimum and the maximum bandwidth. The variation of PAR according to the number of iterations allows to quickly find an optimum solution. Another solution proposed by Wang [23] replaces and updates the parameters PAR and bw (distance bandwidth) according to the minimal and maximal values in HM by the equations $trial^i + [max(HM^i) - trial^i] \times ran[0,1)$ and $trial^i - [trial^i - min(HM^i)] \times ran[0,1)$. Where the parameter $ran[0,1)$ is a random number $\in [0, 1)$, $min(HM^i)$ is the lowest value of the i^{th} variable in HM, $max(HM^i)$ is the highest value of the i^{th} variable in HM and $trial$ is the selected pitch from HM. This method allows faster to search a solution because the parameters bw and PAR are varying according to the extremum candidates inside the population.

4 Harmony Search for Pick and Place

The aim of this paper is to help an operator in setting up a system of pick and place by using evolutionary algorithms. In this paper, we are focused on camera calibration and the detection of thin objects. This chapter first describes the use of this approach for camera calibration and then describes the use of the same approach for the detection of thin objects. In each case, the stop criterion of the evolutionary algorithm can be defined by the duration of processing or/and according to the fitness function.

4.1 Camera calibration

To simplify the calibration part of the operator, we propose a method upstream of the calibration because the system requires a set of images in input. To know if the input images are satisfactory, we propose letting the algorithm decide which images are consistent compared to the others. The algorithm starts by the initialization part of the population by randomly selecting some of the image retrieved by the cameras. The number of images is randomly selected. The solutions of the population are scored and saved in the HM. The score used as fitness function is the reprojection error and allows the evaluation of the quality of the calibration. The second step which corresponds to the optimisation part is executed until the stop criteria is not satisfied. In this step, other solutions are generated. The set of images to use for a calibration is either selected randomly or among HM. If the image is selected inside HM, then another random value determines if the number of images to take needs to evolve or not. In this case, the number is incremented or decremented according to the number of images of the retrieved solution on HM. A new solution is then created and tested. If the new solution is better than the worst

solution in HM, then the new solution replaces the worst solution inside the population. The algorithm looks for another solution if the stop criterion is not satisfied.

4.2 Thin object recognition

Once the calibration is done, we need to identify some object to pick. The aim is to allow the operator to only provide the distance of the object as input parameter. To retrieve a thin object, we apply a thinning algorithm to retrieve the curve of the objects. Each junction point and end point pixel of the extremum curve of the image are retrieved in order to calculate the distance between them. The proposed algorithm starts by randomly calculating the distance between points. The points of both images are randomly selected. Three criteria are necessary to calculate the fitness function. The first one is the depth of two matched points. An operator can easily give the distance between the sensor and the bin of objects. The second criterion is the percentage of presence of the object in the image situated between both points. If the object is not straight, a polygonal approach can be performed. The last criterion is the 3D distance between the points. After applying randomly the initialization, the optimization part selects the points either randomly or among the HM according to the parameter $HMCR$. The PAR parameter determines if the point is selected among the neighbors. If the result is better, a new solution is created and replaces the worst solution inside the HM.

5 Experiment

The experimentation has been realised with an Intel(R) Core(TM) i7-5700HQ, CPU 2.70GHz, industrial cameras and webcams. Zhang [26] algorithm and OpenCV library have been used to calibrate cameras.

5.1 Cameras Calibration

In order to evaluate Harmony Search for Pick and place, Several dozens of calibrations have been made. Thanks to HSP^2 algorithm, the operator only needs to know parameters like the size of the board, the number of square intersections on X and Y on the board and the set of images. As shown on the Figure 1, the algorithm has been compared with several methods to prove its efficiency. Firstly, it has been compared with an improved random strategy where the algorithm randomly selects images. For each calibration, a fitness function is calculated and saved for each image. An average of all the calculated fitness functions is associated with each image. Then, the selection of an image is made randomly with chances to select a good or bad image. We have also compared the algorithm with an Improved Monte Carlo proposed by Rupp [19]. Firstly, the algorithm selects a random number of images. The fitness function is then calculated. The algorithm tests all the images by removing one after the other. The algorithm then adds another image to the initialized set

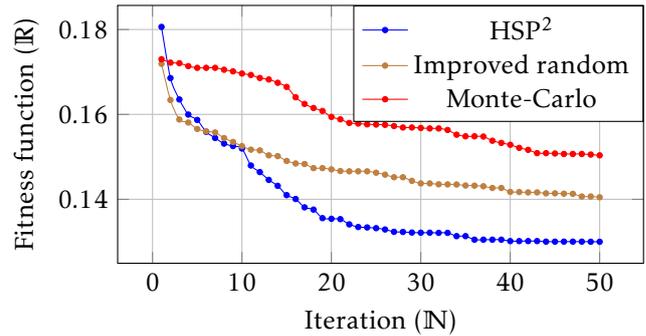


Figure 1: Score (30 calibrations, 50 iterations, 21 images)

among the remaining images. The set of images with the best fitness function is saved. Another cycle is performed by adding/removing all the images one by one.

5.2 Thin object recognition

In this part the operator only has to give the size of the object which corresponds to the distance between the points and a coherent depth scale between the camera and the bin. Several types of objects have been taken into account and placed in different positions. The algorithm was applied to thin objects with the same characteristics. As shown in Figure 2, one of the objects is recovered. It corresponds to the best element to choose according to the calculated criteria of our algorithm. This behavior prevents looking for all objects and reduces the processing time. Then, once the robot has picked the object, it is ready to retrieve another object. The performance of the algorithm was compared to a standard search where each point of the image is compared to the other. The Figure 3 shows the comparison of time processing between random search algorithm and HSP^2 algorithm for 100 search one of 2 points object according to other 2 points objects. The time processing is bigger in random search because there is no optimization process.

6 Discussion

The proposed method has shown its efficiency for system set up. The operator does not need to care about having knowledge in the areas of calibration and recognition. The system requires a good parametrization in order to get good results, but this parametrization is made upstream of the optimization process. The curve on the Figure 1 proves the efficiency of the proposed algorithm. As presented on the Figures 5 and 4 which represent the fitness function according to the number of objects found, the set of PAR , $HMCR$, and HMS as a big impact on the behavior of the system. To prove the importance of the parametrization we have applied a test to find one object among others. As can be read from the Figure 4, a good scale for HMS is between 20 and 70. If the value of HMS is too low, there is an increase of the number of

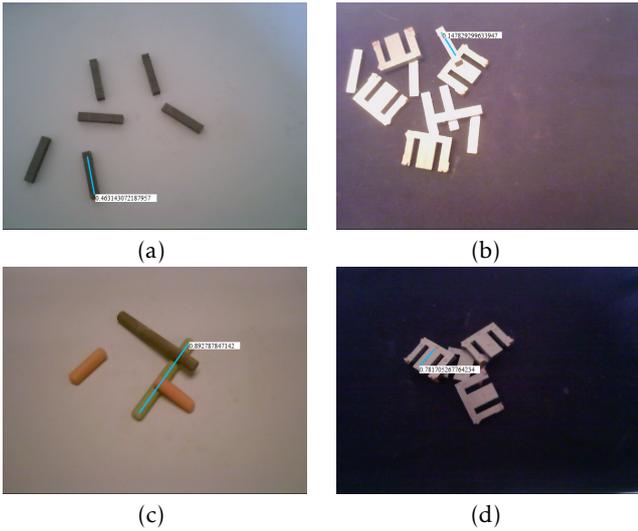


Figure 2: Blue: Objects detection. Text: score. [24]

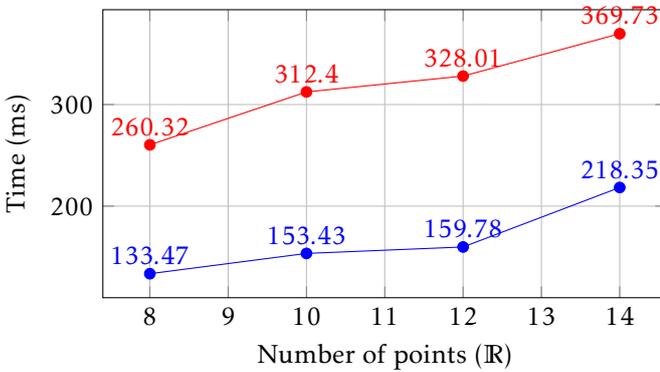


Figure 3: Comparison of HSO and a random algorithm, blue: HSP², red: Standard

iterations and the algorithm is behaving randomly. After a value of 70, the number of iterations is increasing too. An important number of input images or objects inside the scene can have an impact on the behavior of the system. As shows on the Figure 5, the bests parameter for *HMCR* is to be set between 0.5 and 0.8 and for *PAR* is to set between 0.1 and 0.3. These configurations allow to have an optimal search. The algorithm was run 100 times in order to prove the efficiency of the search of thin objects. A test of 7 objects with two end points has been applied in order to retrieve a component of two points. The limitation of the algorithm is the search of a good parametrization; however, this part is not carried out by the operator and must be performed beforehand by the administrator.

7 Conclusion

In this paper, we propose using an evolutionary algorithm to simplify operator pick and place tasks. We have applied the HSP² algorithm on the selection of input im-

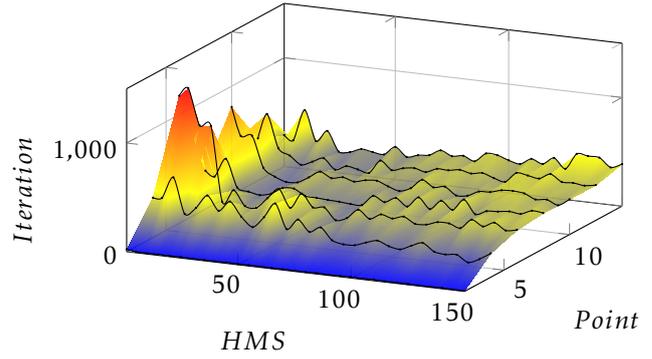


Figure 4: Performance: number of points, HMS

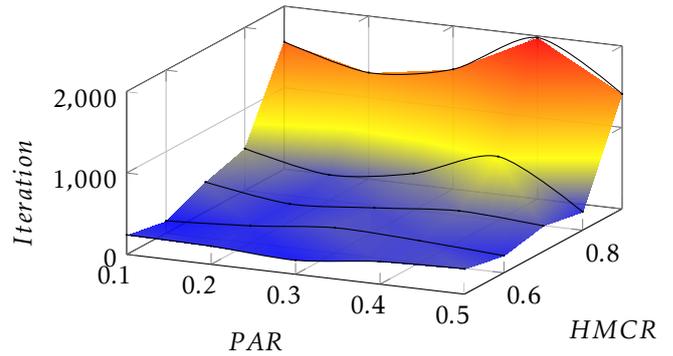


Figure 5: Performance: *PAR*, *HMCR*

ages for camera calibration. Then, we have adapted the algorithm to find thin objects in a scene with the cameras. We have proved the efficiency of the algorithm in several ways. Firstly, the optimization part of the algorithm allows to retrieve a good behaviour of the system. Secondly, the system is easy to set up and quickly retrieves a good fitness function. The next step is to adapt the algorithm with a polygonal approach in order to identify and to locate common objects.

Acknowledgements

This research is supported by the company Hager in France and we want to thank our colleagues for having provided their experience and their expertise. We would like to thank the laboratory ICube for their methodology which has improved the manuscript. Our thanks are also for the Engineering School Ecam-Strasbourg Europe.

References

- [1] Nada MA Al-salami. Evolutionary algorithm definition. *American J. of Engineering and Applied Sciences*, 2(4):789–795, 2009.
- [2] Peter J Angeline. A historical perspective on the evolution of executable structures. *Fundamenta Informaticae*, 35(1-4):179–195, 1998.

- [3] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In European conference on computer vision, pages 404–417. Springer, 2006.
- [4] Dirk Buchholz. Bin-Picking—5 Decades of Research, pages 3–12. Springer International Publishing, Cham, 2016.
- [5] C. Darwin. The Origin of Species. 1859.
- [6] E Roy Davies. Computer and machine vision: theory, algorithms, practicalities. Academic Press, 2012.
- [7] Russell C Eberhart and Yuhui Shi. Comparison between genetic algorithms and particle swarm optimization. In International Conference on Evolutionary Programming, pages 611–616. Springer, 1998.
- [8] Zong Woo Geem. State-of-the-art in the structure of harmony search algorithm. In Recent Advances In Harmony Search Algorithm, pages 1–10. Springer, 2010.
- [9] Zong Woo Geem, Joong Hoon Kim, and GV Loganathan. A new heuristic optimization algorithm: harmony search. Simulation, 76(2):60–68, 2001.
- [10] Philippe Guermeur and Jean Louchet. An evolutionary algorithm for camera calibration. In Proceeding of International Conference on Robotics Distance Learning & Intelligence Communication Systems (ICRODIC’03), Rethymno, Greece, pages 799–804, 2003.
- [11] Katsushi Ikeuchi. Generating an interpretation tree from a cad model for 3d-object recognition in bin-picking tasks. International Journal of Computer Vision, 1(2):145–165, 1987.
- [12] Yan Ke and Rahul Sukthankar. Pca-sift: A more distinctive representation for local image descriptors. In Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on, volume 2, pages II–506. IEEE, 2004.
- [13] Rahul Kumar, Sanjesh Kumar, Sunil Lal, and Pra-neel Chand. Object detection and recognition for a pick and place robot. In Computer Science and Engineering (APWC on CSE), 2014 Asia-Pacific World Congress on, pages 1–7. IEEE, 2014.
- [14] David G Lowe. Object recognition from local scale-invariant features. In Computer vision, 1999. The proceedings of the seventh IEEE international conference on, volume 2, pages 1150–1157. IEEE, 1999.
- [15] M Mahdavi, Mohammad Fesanghary, and E Daman-gir. An improved harmony search algorithm for solving optimization problems. Applied mathematics and computation, 188(2):1567–1579, 2007.
- [16] Osama Moh’d Alia and Rajeswari Mandava. The variants of the harmony search algorithm : an overview. Artificial Intelligence Review, 36(1):49–68, 2011.
- [17] Mahamed GH Omran and Mehrdad Mahdavi. Global-best harmony search. Applied mathematics and computation, 198(2):643–656, 2008.
- [18] Paolo Piccinini, Andrea Prati, and Rita Cucchiara. Real-time object detection and localization with sift-based clustering. Image and Vision Computing, 30(8):573–587, 2012.
- [19] Stephan Rupp, Matthias Elter, Michael Breitung, Walter Zink, and Christian Kueblbeck. Robust camera calibration using discrete optimization. International Journal of Applied Science, 13:250–254, 2006.
- [20] Felix Streichert. Introduction to evolutionary algorithms. 2002.
- [21] Nima Taherinejad. Highly reliable harmony search algorithm. In Circuit Theory and Design, 2009. ECCTD 2009. European Conference on, pages 818–822. IEEE, 2009.
- [22] Markus Ulrich, Christian Wiedemann, and Carsten Steger. Combining scale-space and similarity-based aspect graphs for fast 3d object recognition. IEEE transactions on pattern analysis and machine intelligence, 34(10):1902–1914, 2012.
- [23] Chia-Ming Wang and Yin-Fu Huang. Self-adaptive harmony search algorithm for optimization. Expert Systems with Applications, 37(4):2826–2837, 2010.
- [24] Pierre Willaume, Pierre Parrend, Etienne Gancel, and Aline Deruyver. The graph matching optimization methodology for thin object recognition in pick and place tasks. Symposium on Computational Intelligence in Feature Analysis, Selection, and Learning in Image and Pattern Recognition, 2016.
- [25] Ali Rıza Yildiz and Ferruh Öztürk. Hybrid taguchi-harmony search approach for shape optimization. In Recent Advances In Harmony Search Algorithm, pages 89–98. Springer, 2010.
- [26] Zhengyou Zhang. A flexible new technique for camera calibration. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 22(11):1330–1334, 2000.

Skeletonization and 3D Graph Approach for Thin Objects Recognition in Pick and Place Tasks

Pierre WILLAUME
Icube laboratory, Hager Group
pierre.willaume@ecam-strasbourg.eu

Pierre PARREND
Icube laboratory
Ecam Strasbourg-Europe

Etienne GANCEL
Hager Group

Aline DERUYVER
Icube laboratory
Unistra

Abstract

Objects retrieval is an important task in pick and place processes, especially when the objects are provided in a disorganized set. However, the methods proposed in the literature are not suitable for thin objects because only key points or edges of the shapes are studied, whereas in real cases, the objects can be thin and without special features. In this paper, a method based on graph comparison between stereoscopic images and a model is proposed to retrieve thin featureless objects and without having a long pre-processing time. Firstly, a thinning algorithm is applied to extract the skeleton of both shapes. Secondly, the skeleton is translated into a graph to extract the significant information of each part of the skeleton. Finally, the distances between the nodes are computed and the topography of the graph is analyzed to retrieve the desired objects. A comparison has been realized with a key-points extraction algorithm.

1 Instructions

Pick and place is a process that aims to bring an object to a selected location. This process is constantly improved with the desire of companies to use automated systems adaptable to the work environment [Berger, 2014]. When the starting position is not fixed, a method consisting in using a robot with one or several cameras is usually applied. The cameras detect objects and provide their positions to the robot to pick them up. This method is fast because the objects have not to be pre-placed by another system. However, these systems require image processing for detecting and locating the objects. Object detection difficulties mainly depend on the aspect and on the orientation of the object [Ikeuchi, 1987]. Currently, the proposed solutions require a large database because they are sensitive to the deformations and the method is not adapted for thin objects. The aim of this paper is to propose a method to randomly detect disposed thin objects in 3 dimensions without having a long pre-processing time. We propose a method based on graph matching. The first step is the application of a filter to reduce the noise in the images, which emphasizes the recovered objects. To distinguish the objects among the others and to bring out the thinnest ones, we apply a thinning algorithm to the images. This process allows to retrieve the skeleton of the set of objects but it does not allow to find the match between the representation of the left image and the right image. A solution is to

translate the shapes into a graph for performing the localisation of the different objects. This step permits to find the parts of the image belonging to the same skeleton. For each node of the graph, we match the extrema and the junction nodes. A set of rules based on depth, distance and angle between nodes is established to make the graph matching. Object recognition is performed according to the topology of the graph. The method stands out from the others by the non-necessity to have a CAD file and by its ability to manage objects without using key points features. This paper is organised as follows: The second section describes the state of the art. The third section details the proposed method. The fourth section presents the experimentation. The fifth section discusses the advantages and the limitations of our method. The last section states the conclusion.

2 Related Work

The field of bin-picking studied for decades [Buchholz, 2016, Ikeuchi, 1987, Ulrich et al., 2012, Willaume et al., 2016] still arouses interest mainly to recover disorganized objects. The objects in the bin must be compared with a model to recognize its shape in order to be able to take it. Algorithm performance is calculated according to three different criteria: 1) the computation time of image processing to detect the object is one of the most important criterion, which determines the computation time required to give the position of the object to the robot; 2) the kind of object to be processed (size, shape...); 3) the time to set up and build the model of the system. Some authors [Ikeuchi, 1987] proposed a method to generate 3D algorithms for recognizing objects with a geometric model for pick and place tasks. Apparent shape of the model is generated in various directions of the object. They are classified according to their position and based on a set of characteristics such as the dominant visible faces. An interpretation tree is generated according to the shape of the objects. This process has been improved [Ulrich et al., 2012] to recognize the orientation and position of a 3D part using a single camera and a CAD file. Only the geometric information from the CAD model is generated as a hierarchical model. Information of texture and reflectance of the surface of the object are not taken into account. The large number of applications using these methods in industry proves their efficiency. Indeed, the setup is easy because it only requires a calibration and a CAD file of the object that we look for. However, as the

pre-processing step of the CAD method is greedy in time, the system has to save the set of all the possible positions. Piccinini [Piccinini et al., 2012] proposed to use SIFT algorithm [Lowe, 1999] for the detection and the location of objects. The computation of key points and their description is made during the image acquisition step. The different models are saved in a database. The key points and the descriptions are compared with the models. A clustering is performed in order to detect if the key points are part of the objects. This method was applied to find parts with graphic features that stand out such as a marking. Its application works fine with hidden objects but does not fit for thin parts where the key points do not stand out. Detection of the object to look for depends on its characteristics (shapes, pattern...) and of the number of template images saved in the database. Kumar [Kumar et al., 2014] proposed an algorithm to recognize different objects. The algorithm starts by converting the image into greyscale in order to convert the image in a binary image. Edge detection is performed in order to retrieve the edge of the objects. A double thresholding is performed in order to preserve high value pixels. An edge tracking by hysteresis is performed followed by an image filling of the image dilatation. The algorithm ends by the BLOB analysis. This method allows to retrieve objects with high contrasts. However, the algorithm is not suitable for thin objects where the edge and the threshold would not put forward the object to take. In this paper, we propose a method to retrieve objects where the key points of the images are not present and where the CAD file is not provided due to the long pre-processing time.

3 The Proposed method

In this paper, we propose an approach based on the 3D graph matching. The graphs represent the skeletons of objects in the stereo images in order to recover the position and orientation of thin objects. The aim is to retrieve the location information of the different objects placed in the cameras field of view.

3.1 Binarization and Skeletonization

Algorithms of Binarization (bi-level or two-level) as represented on the image 1b allow to convert an image (1a) with different values of pixels into a binary image. It aims at removing the background elements and to highlight the target objects. The choice of the threshold is performed according to the shape of the histogram (number of pixels by color). The threshold is chosen according to the color of the object to retrieve and the color of the elements to remove like for example the background. Then, a Thinning or Skeletonization algorithm is applied in order to represent the structural shape of a plane region. It reduces the shape to a set of curves called skeleton which are centred to the original shape. Several propositions have been proposed in 3D matching [Cornea et al., 2005]. Topology properties are kept from the original shape. As shown on figure 1c, after having applied the thinning algorithm to the left image 1a, the thin objects stand out clearly on the right image. The fast method used [Zhang and Suen, 1984] remove all the points of

the image edge with the exception of the point belonging to the Skeleton. To preserve the connectivity of the skeleton, the iterations are divided in 2 subiterations. In the subiterations, the edge point is deleted according to a set of conditions (numbers of nonzero neighbours, number of pattern...).

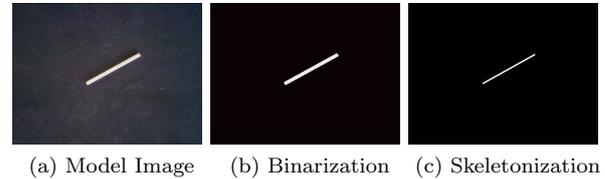


Figure 1: Image simplifications (threshold: [100,255])

3.2 Image Matching by using Graph approach

Graphs are representation tools and are widely used in the field of pattern recognition [Conte et al., 2004, Biasotti et al., 2008] and several methods have been proposed to relate them [Bunke and Shearer, 1998, Champin and Solnon, 2003]. However, in this method we do not attempt to calculate a similarity criterion because these graphs are similar, as it happens with every pair of stereoscopic images in the same plane. The approach has to focus on the best match by using the structural features of the graphs. We define a graph by a two-tuple $G=(V,E)$ where V is the finite set of vertices $V=\{v_1,v_2,v_3...v_n\}$ with $n=Card(V)$ (called also nodes) and E is a set of Edges $E=\{e_1,e_2, e_3...e_m\}$ with $m=Card(E)$. In our case, the nodes represent the end-points and the connection points of the skeleton image. The edges are defined by the link between the nodes. Firstly, and as shown in the algorithm 1, an image is defined as a reference and each node is labelled. For each node of the labelled picture, the depth and their position and those of the other image is processed. If the depth and the position is included inside the scale defined by the user, then the label of the first node is added to the second one. This selection allows to have a first pre-selection of the graph matching. However, this first selection is performed according to the accuracy defined by the user and a node can have several candidates. To remove the nodes with several candidates, two matched nodes are selected as referent. For each node with several candidates, a distance and an angle is computed between both selected points and the candidates of the nodes. In order not to reduce the method for planar configurations, a threshold can be added to the angle to permit the projective invariant. As shown in figure 2a and 2b, the node 5 in the left image 2a has two candidates in the right image 2b with the possibilities of 4 and 5. Two referents points are selected (7 and 8), in the right picture 2b, angle and distance node is determined by the points 7,8,"4,5" points, and on the left figure 2a both 4 and 5 nodes are processed. The skeleton of the picture is not perfect and can provide some useless node. To avoid this situation, a selection of the node is done by selecting them by the size of their segment. A smooth function can be added to pull out the element already used. This function is used when a node is matched. This function browses the element of matched nodes

and removes the taken nodes from the set of nodes. However, if a node is badly matched, then the graph matching will fail.

Algorithm 1: Graph matching

```

input : (V(G1),V(G2)): G1 and G2 graphs
         nodes set (DMin,DMax): Depth interval
         (PMin,PMax): Position interval
output: (v.ID,w.ID): Graphs list labels
Integer label = 0;
foreach node v ∈ V(G1) do
    label++;
    v.ID.Add(label);
    foreach node w ∈ V(G2) do
        D ← D.calc(v, w); P ← P.calc(v, w);
        if DMin<D<DMax & PMin<P<PMax then
            | w.ID.Add(label);
        end
    end
end
end

```

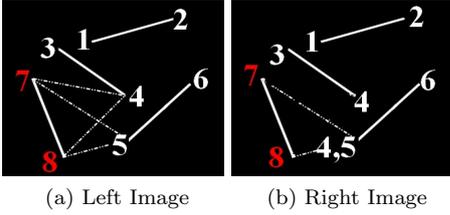


Figure 2: Graph matching between two pictures

3.3 Object retrieval

The link between right and left image allows to correctly calculate the coordinates of the 3D points to give to the robot so he can pick the object. The object extraction step is applied by retrieving known characteristics of the object. Firstly, we choose couples of linked nodes belonging to a same object. Secondly, the object can be found by processing the distance of the desired part. For each node on the right image and their match on the left image, a distance is calculated with the other nodes. We note G1 and G2 the graphs representing respectively the left and the right image. V(G1) and V(G2) are the sets of nodes of the graphs G1 and G2. The equation is $3D = \sum_{i=1}^k \sum_{j=1}^l \sqrt{(X_i - X_j)^2 + (Y_i - Y_j)^2 + (Z_i - Z_j)^2}$ where $k = \text{Card}(V(G1))$ and $l = \text{Card}(V(G2))$ and X Y Z are the 3D coordinates of each node. For each node $v \in V(G2)$, a distance is calculated between v and each other node of the graph. This distance can be calculated in 2D to avoid the system to detect an object with an orientation difficult to catch by the robot. To distinguish objects with several nodes, the characteristics like the number of neighbours with the distance and the angle between them can be processed to highlight specific nodes of the desired object.

4 Experimentation

In this paper, experimentation has been performed on several kinds of objects and with different orien-

tations. Experimental part has been realised with 2 webcams (640*480) placed at 400mm and 2 industrial cameras Cognex IS-5603-10 (800*600) with 16mm 1:1.4 lens placed at 700mm of the scene. The computer has an Intel(R) Core(TM) i7-5700HQ CPU @ 2.70GHz, 2701 MHz. The library OpenCV has been used to calibrate the cameras ([Zhang, 2000]). Zhang algorithm has been used for the skeletonization of the images. Images have been obtained in standard luminosity in order to prove the efficiency of the algorithm. Firstly, the objects have been placed side by side in different angles in order to show that the algorithm finds the object in an easy case. Secondly, the program has been tested with overlapped objects. This situation allows to show that the algorithm can dissociate each object. This situation is the best way to represent a pick and place task because the same object is available in several exemplars in a bin. Thirdly, the algorithm has been tested on an image with overlaps and a combination of different kinds of objects (figure 4). Objects were selected for their colours and shape. The algorithm correctly retrieves the components, placed at different position with overlaps. In the figure 4, it can be seen that the algorithm is able to bypass the problem of tangled objects. The algorithm also shows an efficiency to find non-stick objects (figure 3a) and with transparency (figure 3b). The processing time varies between 600 and 1800 milliseconds for images between 10 and 70 nodes.

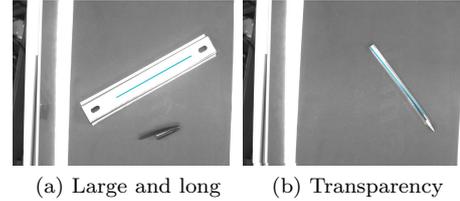


Figure 3: Looking for different objects



Figure 4: Looking for thin sticks placed randomly (Line: Segment of the objects, Text: Objects position)

5 Discussion

The experimental part allowed to prove that our method can find thin objects in random positions. The use of a thinning algorithm allowed to highlight the objects. Graph matching between the nodes of the right

Image	A	B	C	D	E
Image					
Objects number	4	5	1	2	1
Candidates number	4	5	1	1	1
Nodes number	8	11	313	283	85
Object					
Thin	Y	Y	Y	N	N
Random	Y	Y	Y	Y	Y
Cluttering	N	Y	N	N	N
Transparency	N	N	Y	N	N
Binarization Threshold					
Min	186	186	170	185	170
Max	255	255	255	255	255
Score					
SIFT (%)	50	54	60	92	96
SURF (%)	49	56	74	59	71
Graph (%)	100	100	100	100	100

Table 1: Algorithms performances. Y = Yes, N = No

image and the left image allows to perform the object retrieval step based on the data returned by the user as the distance of the segments and the links between them. The results have demonstrated their efficiencies from several points of views. First, all the objects have been identified under different conditions of orientation, tilt, tangle and colors. The second advantage of the method is to have no pre-treatment time because the algorithm directly retrieves the element part of the object to take. Lastly, the method permits to identify thin objects. A comparison with SIFT and SURF algorithm has been done on the images. Processing time in SIFT/SURF is under 300ms for each image, which is better than our method. However, it is negligible compared to the time taken in the pick and place task. Table 1 shows the percent of correct number of segments belonging to the sticks according to the total number of segments belonging to the sticks. Some features have been found, however some matches are wrong because looking for key points is difficult for this kind of image where the sticks are uniform and featureless.

6 Conclusion and Perspectives

This paper proposes an approach to recognize thin objects in stereovision for pick and place tasks. A skeleton has been built for each object and the mapping problem between the junction points and the extrema was solved by graph matching by removing points where position and depth cannot match. The other points are mapped thanks to the matching based on the angle and the distance between the points of the right image and the left image. The process has been tested with objects placed close to each other, with overlapping and cluttering. All objects have been found. The algorithm can work with different kinds of noises like luminosity and brightness. The advantage of our method is to be quickly initialized because the parameters (shape characteristics, dimensions) are known by the operator. There is also no problem of distortion because the extraction is made directly on the parts of the object of interest. The perspective of our work is to enhance the detection in real time of the

object by using for example an evolutionary algorithm.

References

- [Berger, 2014] Berger, R. (2014). Industrie 4.0: The new industrial revolution how europe will succeed. *Think Act*, page 8.
- [Biasotti et al., 2008] Biasotti, S., Giorgi, D., Spagnuolo, M., and Falcidieno, B. (2008). Reeb graphs for shape analysis and applications. *Theoretical Computer Science*, 392(1):5–22.
- [Buchholz, 2016] Buchholz, D. (2016). *Bin-Picking: New Approaches for a Classical Problem*, chapter Bin-Picking—5 Decades of Research, pages 3–12. Springer International Publishing, Cham.
- [Bunke and Shearer, 1998] Bunke, H. and Shearer, K. (1998). A graph distance metric based on the maximal common subgraph. *Pattern recognition letters*, 19(3):255–259.
- [Champin and Solnon, 2003] Champin, P.-A. and Solnon, C. (2003). Measuring the similarity of labeled graphs. In *International Conference on Case-Based Reasoning*, pages 80–95. Springer.
- [Conte et al., 2004] Conte, D., Foggia, P., Sansone, C., and Vento, M. (2004). Thirty years of graph matching in pattern recognition. *International journal of pattern recognition and artificial intelligence*, 18(03):265–298.
- [Cornea et al., 2005] Cornea, N. D., Demirci, M. F., Silver, D., Dickinson, S., Kantor, P., et al. (2005). 3d object retrieval using many-to-many matching of curve skeletons. In *International Conference on Shape Modeling and Applications 2005 (SMI'05)*, pages 366–371. IEEE.
- [Ikeuchi, 1987] Ikeuchi, K. (1987). Generating an interpretation tree from a cad model for 3d-object recognition in bin-picking tasks. *International Journal of Computer Vision*, 1(2):145–165.
- [Kumar et al., 2014] Kumar, R., Kumar, S., Lal, S., and Chand, P. (2014). Object detection and recognition for a pick and place robot. In *Computer Science and Engineering (APWC on CSE), 2014 Asia-Pacific World Congress on*, pages 1–7. IEEE.
- [Lowe, 1999] Lowe, D. G. (1999). Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, pages 1150–1157. Ieee.
- [Piccinini et al., 2012] Piccinini, P., Prati, A., and Cucchiara, R. (2012). Real-time object detection and localization with sift-based clustering. *Image and Vision Computing*, 30(8):573–587.
- [Ulrich et al., 2012] Ulrich, M., Wiedemann, C., and Steger, C. (2012). Combining scale-space and similarity-based aspect graphs for fast 3d object recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(10):1902–1914.
- [Willaume et al., 2016] Willaume, P., Parrend, P., Gancel, E., and Deruyver, A. (2016). The graph matching optimization methodology for thin object recognition in pick and place tasks. In *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1–8.
- [Zhang and Suen, 1984] Zhang, T. and Suen, C. Y. (1984). A fast parallel algorithm for thinning digital patterns. *Communications of the ACM*, 27(3):236–239.
- [Zhang, 2000] Zhang, Z. (2000). A flexible new technique for camera calibration. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(11):1330–1334.

Modélisation de formes 3D par les graphes pour leur reconnaissance. Application à la vision 3D en robotique dans des tâches de "Pick-and-Place"

Résumé

L'objectif de cette thèse est de concevoir un système automatique constitué d'une ou plusieurs caméras capables de détecter en trois dimensions un amalgame d'objets stockés dans un conteneur. Pour ceci, il est nécessaire de modéliser, de reconnaître et de localiser des formes dans une image. Dans un premier temps, Nous proposons une solution d'optimisation du calibrage de caméras. C'est une tâche essentielle pour récupérer des informations quantitatives sur les images capturées. Cette méthode nécessite des compétences spécifiques en matière de traitement d'image, ce qui n'est pas toujours le cas dans l'industrie. Nous proposons d'automatiser et d'optimiser le système d'étalonnage en éliminant la sélection des images par l'opérateur. Ensuite, nous proposons d'améliorer les systèmes de détection d'objets fins et sans motif. Enfin, nous proposons d'adapter des algorithmes évolutionnaires dans le but d'optimiser les temps de recherche.

Mots-clés : Reconnaissance d'objets, calibrage, algorithmes évolutionnaire, optimisation

Summary

The aim of this thesis is to design an automatic system involving one or several cameras capable of detecting in three dimensions a set of objects placed in a bin. To do this, we must model, recognize and locate shapes in an image. First, we propose a solution to optimize the camera calibration system. This is an essential task for the retrieval of quantitative information about the captured images. However, the current methods require specific skills in image processing, which are not always available in industry. We propose to automate and optimize the calibration system by eliminating the selection of images by the operator. Second, we propose to improve the detection systems for thin and featureless objects. Finally, we propose to adapt evolutionary algorithms to optimize search times.

Keywords: Object recognition, calibration, evolutionary algorithms, optimization