

Inspection automatisée d'assemblages mécaniques : vers une approche couplée vision 2D / vision 3D

Ilisio Viana, Rémi Parlouar, Florian Bugarin, Jean-José Orteu, Ludovic Brèthes

▶ **To cite this version:**

Ilisio Viana, Rémi Parlouar, Florian Bugarin, Jean-José Orteu, Ludovic Brèthes. Inspection automatisée d'assemblages mécaniques : vers une approche couplée vision 2D / vision 3D. [Rapport de recherche] IMT Mines Albi. 2013. hal-00839887

HAL Id: hal-00839887
https://hal.science/hal-00839887

Submitted on 1 Jul 2013

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.

Inspection automatisée d'assemblages mécaniques : vers une approche couplée vision 2D / vision 3D

I. Viana[1] R. Parlouar [2] F. Bugarin[1] J.-J. Orteu[1] L. Brèthes[2]

[1] Université de Toulouse, Mines Albi, ICA (Institut Clement Ader), Campus Jarlard, F-81013 Albi France
[2] NOOMEO, 425 rue Jean Rostand, F-31672 Labège, France

ilisio.viana@mines-albi.fr
remi.parlouar@noomeo.eu
florian.bugarin@mines-albi.fr
jean-jose.orteu@mines-albi.fr
ludovic.brethes@noomeo.eu

Résumé

Cet article propose une méthodologie d'inspection automatisée d'assemblages mécaniques basée sur l'utilisation d'un bras manipulateur, équipé en tête d'effecteur d'un capteur de vision artificielle. La méthodologie d'inspection proposée dans cet article se base sur le couplage d'informations 2D et 3D, pour tirer profit de la rapidité de l'analyse en 2D et de la complétude des données 3D.

Mots Clef

Inspection Automatisée, Robotique, Vision Artificielle, Analyse d'Images

Abstract

This paper proposes a methodology for automated inspection of mechanical assemblies based on the use of a robot arm, equipped with end effector sensor vision. Inspection methodology proposed in this paper is based on combining 2D and 3D information, to take advantage of accelerated images processing thechniques and the completeness of 3D data.

Keywords

Automated Inspection, Robotics, Computer Vision, Image Processing

1 Introduction

Dans l'industrie, la plupart du temps, l'inspection est réalisée par des opérateurs humains, en prélevant des échantillons d'un ensemble de production. Les caractéristiques des échantillons sont généralisées sur l'ensemble de la population, ce qui implique un risque statistique autant pour le fournisseur que pour le consommateur [1].

De surcroît l'inspection peut être une tâche difficile voire irréalisable par des humains. Sont inclus dans cette catégorie, des tâches exigeant un contrôle rigoureux qui ne peut pas être réalisé par l'Homme compte tenu des limitations du système visuel humain, en terme de perception spatio-temporelle, l'inspection dans des environnements hostiles, l'inspection de volumes de production importants. Certaines applications dans l'aérospatiale, dans la médecine, etc, ne tolèrent aucun défaut. Inspecter une grande quantité de pièces peut être épuisant tant au plan physique que mental pour un Homme. Lorsque c'est le cas, cela peut avoir un impact négatif sur la fiabilité, l'uniformité et donc sur la qualité de l'inspection [1, 2, 3, 4].

Dans le but de contrôler automatiquement une quantité importante d'assemblages mécaniques, la société Noomeo a mis au point un système d'inspection automatisée, basé sur un bras manipulateur portant un capteur basé sur la stéréovision en tête d'effecteur (cf. figure 1).



FIGURE 1 – Robot portant le capteur.

Le capteur embarqué par le robot est composé de deux têtes stéréo, un champ large et un champ proche et de deux projecteurs de lumière (cf. figure 2). Il fournit des paires d'images stéréo et des nuages de points 3D.



FIGURE 2 – Capteur : deux têtes stéréos, grand et petit champs.

Le modèle CAO de la pièce qui contient le modèle CAO des éléments à contrôler est fourni (cf. figure 3).



FIGURE 3 – Exemple de modèle CAO d'un assemblage mécanique.

Les objectifs recherchés dans cette application sont les suivants :
- *décider la présence/absence de l'élément à contrôler,*
- *vérifier le positionnement lorsque qu'un élément à inspecter est présent,*
- *vérifier son exactitude géométrique par rapport à l'élément idéal décrit dans le modèle CAO.*

Pour contrôler un élément, il faut le sélectionner dans la maquette numérique. Les trajectoires menant à la position de cet élément ainsi que les points d'acquisition correspondants sont calculés hors ligne. Le capteur stéréo utilisé est calibré en amont, de façon précise, par une procédure hors ligne. Connaissant le modèle géométrique direct du robot, la position du capteur peut être estimée pour chaque point d'acquisition.

Ensuite, une reconstruction 3D dense est effectuée pour produire un nuage de points 3D de l'élément à contrôler, suivi d'un recalage avec le modèle CAO du même élément. La mesure de l'erreur entre les données reconstruites et les données théoriques permet de prendre la décision par rapport à la conformité ou non de l'élément inspecté. L'inspection d'un élément (cf. figure 4) par cette méthode dure près de 10 secondes en utilisant un ordinateur équipé d'un Processeur Intel Xeon quad core, de 3 GHz et de 12 Go de RAM.

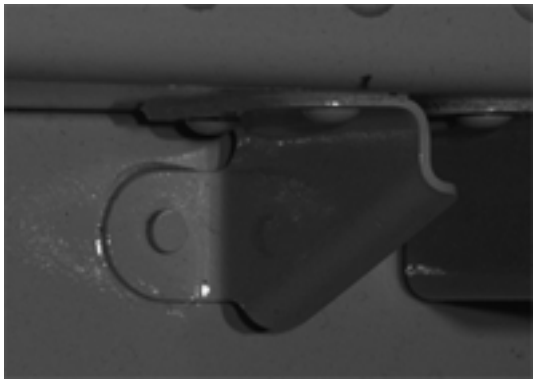


FIGURE 4 – Image réelle d'un élément à contrôler.

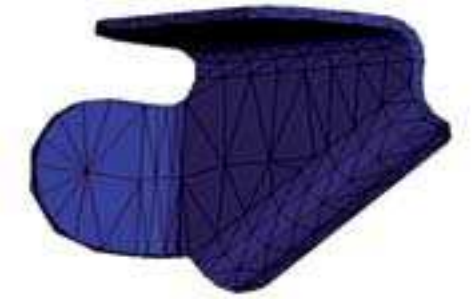


FIGURE 5 – Modèle CAO d'un élément à contrôler.

Cependant, pour certaines applications industrielles ce temps de calcul n'est pas satisfaisant. Par exemple, contrôler 1000 éléments prend plus de 2h30 de calcul. L'approche 3D nécessite une étape de calibration et de reconstruction. Cela ralentit le système et le rend sensible à la qualité du calibrage. Ces raisons ont conduit à la recherche d'une autre méthode d'inspection automatisée, qui puisse être complémentaire à l'approche 3D pure, tout en étant aussi fiable que cette méthode 3D mais plus rapide.

Afin d'accélérer l'inspection, la méthode proposée dans cet article se base sur l'utilisation des techniques d'analyse d'images 2D (en faisant l'hypothèse que ces types de traitements sont plus rapides que les traitements 3D).

Dans certains cas, une approche purement 2D peut ne pas être suffisante. Le but est alors de combiner la 2D avec la 3D pour faire un usage optimal des données disponibles.

1.1 Travaux antérieurs

Au cours des années, l'intérêt pour l'inspection augmente aussi bien au sein des laboratoires de recherche que dans le milieu industriel. Au début de l'inspection par vision artificielle, les approches 2D étaient privilégiées en raison de la faible puissance des calculateurs. Malgré cela, ces approches 2D ont pu faire leurs preuves.

Modayur et al. [5] ont présenté l'inspection visuelle de pièces manufacturées. Dans ce travail, ils décrivent un système automatisé de vision artificielle pour l'inspection dimensionnelle, basé sur la CAO. Ils ont défini des tolérances géométriques précises et ont développé des algorithmes de mesure correspondant à ces définitions de tolérances géométriques.

Newman et Jain [1] ont fait une étude sur les systèmes d'inspection visuelle parus dans la littérature entre 1988 et 1993. Leur travail aborde les progrès de l'inspection visuelle et le rôle grandissant de la CAO dans les systèmes d'inspection visuelle automatisée.

Edinbarough et al. [6] ont présenté des modèles mathématiques ainsi que des algorithmes permettant de déterminer l'erreur de circularité dans les éléments de forme circulaire. Dans l'approche qu'ils ont proposée, un modèle de circularité est comparé à l'élément à contrôler grâce à une comparaison basée sur les contours.

L'évolution de l'électronique et l'augmentation de la puissance de calcul ont permis de simplifier l'inspection [7, 8], notamment en rendant possible ou plus performante l'utilisation des techniques basées sur la 3D. Etudier la 3D est très intéressant car elle décrit les objets à inspecter de manière plus exhaustive que la 2D. L'enjeu autour de la vision 3D est de plus en plus important, toutefois, le temps de calcul émerge comme une difficulté importante à contourner. Plusieurs travaux intéressants d'inspection utilisant des techniques de vision 3D ont été réalisés y compris celui qui a été présenté dans la section précédente, concernant la société Noomeo.

Newman et Jain dans un autre travail [9], ont proposé un système utilisant la 3D, pour l'inspection basée sur la CAO, en utilisant les images de profondeur. L'apport de l'utilisation des images de profondeur et l'utilité du contrôle automatisé y sont clairement décrits. L'objectif visé de leur système consistait en l'inspection de défauts dans les surfaces de moulage ainsi qu'en l'inspection des tolérances dimensionnelles. Tout d'abord, ils ont segmenté la surface de l'élément à contrôler, en faisant une classification des surfaces et de certaines caractéristiques. Ils ont ensuite fait une mise en correspondance des résultats de cette classification avec des modèles présents dans une base des données, pour détecter la configuration de l'élément, la présence d'un artefact ou des parties manquantes dans l'élément à contrôler. Après ils ont décliné une approche descendante de détection des défauts, en partant des défauts les plus importants pour arriver aux plus fins.

Tarbox et Gottschlich [10] ont présenté ce qui à leur époque était une approche innovante d'inspection visuelle automatisée, basée sur la comparaison de modèles volumiques d'un objet de référence et d'un objet réel créé itérativement à partir des données d'un capteur. L'apport de cette approche était le fait de ne pas avoir besoin d'identifier des caractéristiques particulières car cela n'est pas toujours une tâche aisée.

Une autre étude bibliographique, plus récente, a été présentée par Malamas et al. [8]. Cette étude fait l'état de l'art de l'inspection par vision artificielle et présente les outils (logiciels et matériels) disponibles pour le développement de systèmes de vision industrielle.

Flavio Prieto [4] a travaillé sur l'inspection automatisée de pièces manufacturées en utilisant la CAO et des nuages de points 3D de la pièce à contrôler. Dans son travail, deux points importants sont à signaler : le recalage entre les données 3D et le modèle CAO, suivi d'une segmentation utilisée pour vérifier les tolérances géométriques et dimensionnelles de l'élément à inspecter. De plus, il a présenté un état de l'art très intéressant à propos de l'inspection visuelle.

Boutarfa et al. [11] ont présenté une méthode de contrôle automatique de pièces manufacturées. Ils ont utilisé les nuages de points 3D de la pièce à inspecter ainsi que le modèle CAO correspondant, dans les formats STL et IGES. D'abord, ils procèdent à un recalage entre ces deux entités, suivi d'une segmentation du nuage des points, en calculant la distance minimale entre celui-ci et les surfaces présentes dans le modèle CAO, format STL. Ensuite, ils ont comparé quelques caractéristiques géométriques du nuage de points 3D avec le point le plus proche dans le modèle. Puisqu'ils ont travaillé aussi avec un autre mode sous le format IGES, dans lequel les surfaces sont définies comme des surfaces paramétriques NURBS, ils ont su extraire des caractéristiques géométriques telles que la normale à la surface, les courbures Gaussienne et moyenne, pour les comparer au modèle CAO.

Récemment, Christian Wohler [12] a présenté un travail sur des systèmes d'inspection basés 3D. Ce travail aborde l'inspection des pièces rigides et non rigides ainsi que de pièces métalliques. Différentes méthodologies d'inspection applicables à chaque type de pièce sont présentées dans ce travail.

2 Inspection automatisée par couplage de techniques 2D/3D

Afin de répondre aux objectifs fixés (vérifier la présence/absence d'un élément que l'on souhaite contrôler, la conformité de son positionnement et son exactitude géométrique par rapport à un élément idéal décrit dans un modèle CAO), l'approche présentée dans cet article se base sur une mise en correspondance entre les données 3D du modèle CAO (cf. figure 5) projetées dans une image 2D (cf. figure 7) avec l'image 2D de l'élément réel (cf. figure 4), en utilisant des contours (cf. figures 8, 9). Cela permet d'éviter l'étape de reconstruction 3D de l'élément réel, économisant ainsi le temps de calcul.

L'entité représentant les deux types de données mis en jeu est un contour 2D. La méthode cherche à optimiser la position et l'orientation du contour 2D représentant l'objet de référence par rapport au contour 2D représentant l'élément réel.

La méthode proposée dans cet article est résumée sur la figure 6.

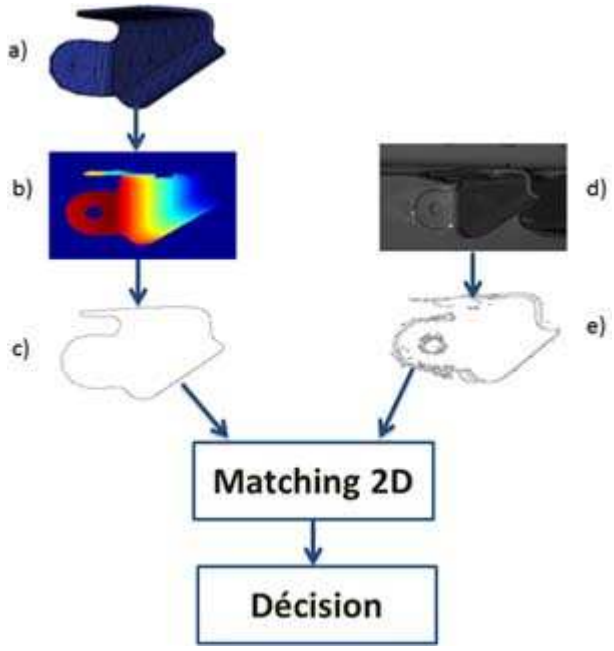


FIGURE 6 – Schéma de la méthode proposée. a) modèle CAO de l'élément, b) image de profondeur issue du modèle CAO, c) contour 2D issu de l'image de profondeur, d) image en niveau de gris de l'élément réel, e) contour extrait de l'image réelle.

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

2.1 Explique comment obtenir un contour 2D à partir d'un modèle CAO en 3D ;

2.2 Décrit la fonction de mesure qui sera employée lors du processus d'optimisation ;

2.3 Traite de la méthode d'optimisation utilisée ;

2.4 Tranche sur la convergence de l'algorithme d'optimisation et décide si l'élément a été bien contrôlé ou non.

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

Afin de récupérer des contours 2D à partir du modèle CAO de l'élément à contrôler, une rastérisation (conversion des données vectorielles, ici des triangles, en données matricielles, ici une image de profondeur) est faite, puis elle est suivie d'une extraction de contours. La matricialisation est effectuée à l'aide d'OpenGL. L'algorithme de matricialisation utilise les paramètres de calibration des caméras, aussi bien les paramètres intrinsèques qu'extrinsèques et renvoie l'image de profondeur correspondante. Les paramètres de calibrage des caméras sont requis lors de cette étape parce qu'ils permettent de calculer les matrices de projection et de vue (cela inclut les caractéristiques du capteur ainsi que la position de l'élément observé). L'image de profondeur est extraite directement de la carte graphique, du buffer de profondeur connu sous le nom de Z-buffer. L'utilisation des données 2D est un point clé de cette approche, notamment en termes d'économie en temps de calcul. Plus l'élément réel se rapproche des spécifications définies par le modèle théorique, plus l'image rastérisée sera proche de l'image de l'élément réel. Cela est une condition nécessaire pour prétendre avoir une méthode d'inspection automatisée qui soit rapide et fiable.

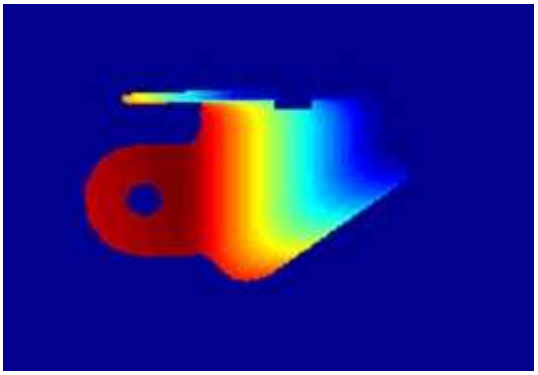


FIGURE 7 – Image obtenue par rastérisation de la CAO.

Dès lors que les deux images représentant le modèle 3D de base et l'élément réel sont disponibles, une extraction de contours est effectuée, en utilisant le détecteur de contours de Canny [13, 14]. Cette technique très courante en vision artificielle, donne des résultats satisfaisants et est disponible dans les libraires les plus populaires du traitement des images, notamment Matlab, OpenCV. La technique de détection de contours de Canny utilise le seuillage

[15, 16] par hystérésis. Cette technique de seuillage est adaptée car elle offre la possibilité d'ajuster les paramètres de façon à éviter éventuellement le bruit et/ou les faux contours. Plusieurs méthodes de filtrage et de préservation des contours [17] existent dans la littérature. Afin d'obtenir une meilleure qualité de contours, le détecteur de Canny peut être combiné avec une technique de filtrage pour éliminer les bruits dus notamment aux réflexions spéculaires.

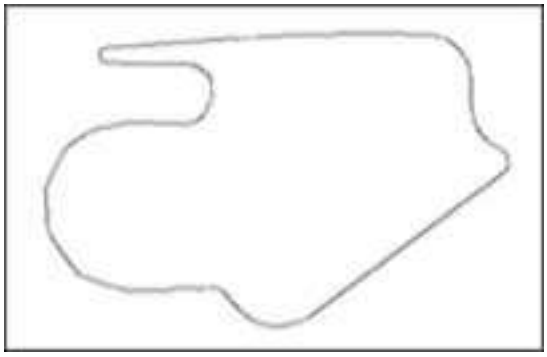


FIGURE 8 – Contours extraits de l'image de profondeur.

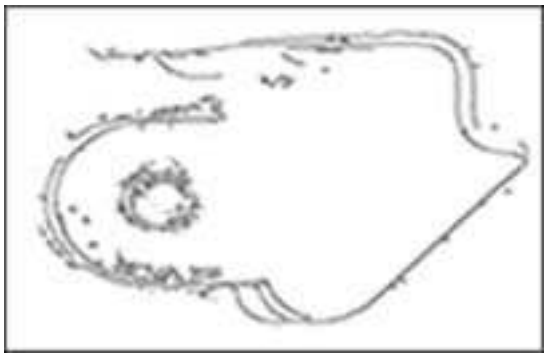


FIGURE 9 – Contours extraits de l'image de l'élément réel.

2.2 Construction de la fonction de mesure

Afin de comparer le contour 2D de l'élément réel et le contour 2D construit à partir du modèle CAO, une fonction de mesure est construite en deux étapes. La première phase consiste à effectuer une transformée en distance [18] d'une image de contours représentant l'élément réel que l'on souhaite inspecter. La transformée en distance (cf. figure 10) est une représentation d'une image numérique dans laquelle les pixels sont étiquetés avec la distance au plus proche obstacle, souvent la frontière ou un contour. La métrique [19, 20] utilisée pour le calcul de la transformée en distance est la distance euclidienne, pour des raisons de précision. L'avantage de la transformée en distance réside dans le fait qu'elle peut être calculée préalablement à l'étape d'optimisation puis appelée lors du calcul des itérés. Cela permet d'éviter un processus d'évaluation de la fonction objectif coûteux en temps de calcul.

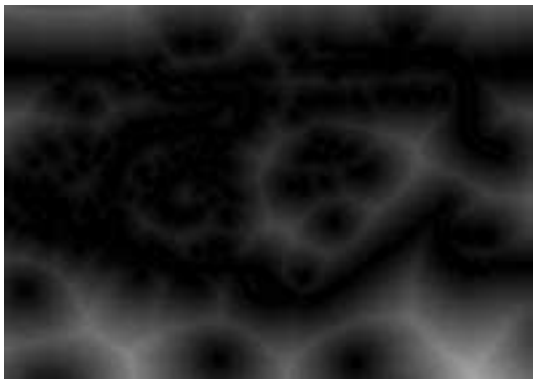


FIGURE 10 – Image de transformée en distance des contours d'un élément réel.

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

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

où
- C_T désigne l'ensemble des pixels du contour théorique issu du modèle CAO et C_R l'ensemble des pixels du contour réel.
- p désigne un pixel du contour du modèle CAO.
- $\mathbf{R}(\theta)$ est une matrice de rotation [21, 22] construite à partir de l'angle θ appliqué au contour théorique C_T.
- $\mathbf{t} = (t_x, t_y)$ est un vecteur de translation [21, 22] appliqué au contour théorique C_T.
- N_T désigne le nombre de pixels du contour théorique C_T
- $I_{DT}(C_R, p)$ est la transformée en distance du pixel p par rapport au contour réel C_R.

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

2.3 Algorithme d'optimisation utilisé

La méthode d'optimisation utilisée dans ce travail est la descente de gradient. C'est une méthode d'optimisation du premier ordre, utilisée pour trouver le minimum local d'une fonction. Nous utilisons cet algorithme d'abord sur la variable $\mathbf{t}$ pour une rotation nulle. Nous obtenons ainsi une translation optimale $\mathbf{t}^\star$. Puis nous optimisons la fonction $\theta \rightarrow S(\mathbf{t}^\star, \theta)$ et obtenons finalement un candidat $(\mathbf{t}^\star, \theta^\star)$ à l'optimalité. Pour ces deux optimisations successives,

compte tenu du fait que les dérivées partielles de la fonction objectif ne peuvent être décrites analytiquement, elles ont été estimées en utilisant les différences finies [23] centrées. Par exemple, les dérivées partielles de S selon les variables t_x et t_y sont données par :

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

et

$$\frac{\partial S(\theta,t_x,t_y)}{\partial t_y} = \frac{S(\theta,t_x,t_y+1)-S(\theta,t_x,t_y-1)}{2} \quad (3)$$

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

$$t_x^{i+1} = -K_x^i G_{t_x^i} + t_x^i \quad (4)$$

et

$$t_y^{i+1} = -K_y^i G_{t_y^i} + t_y^i \quad (5)$$

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

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

$$\theta^{i+1} = -K_{\theta^i} G_{\theta^i} + \theta^i \quad (6)$$

La méthode est initialisée à partir de $\theta = 0$ et le pas est initialement fixé à $\frac{1}{248}$. Cette dernière valeur a été déterminée par essai-erreur à l'aide de nos différentes expériences.

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

2.4 Description de la fonction de décision

Pour pouvoir comparer les résultats de la méthode proposée dans cet article et la méthode décrite dans la section 1, un critère de décision a été défini permettant de statuer sur la convergence de l'algorithme d'optimisation.

Ce critère de décision se base sur le taux de recouvrement entre le contour du modèle CAO et le contour de l'élément réel. Il ne tient pas seulement compte du nombre de pixels du modèle CAO qui se superposent aux pixels du contour de l'élément réel mais aussi des contours parasites à l'intérieur du modèle CAO pendant la phase de recouvrement. La décision se fait donc par seuillage du taux de recouvrement entre les contours du modèle CAO et celui de l'élément à contrôler et du taux de superposition des contours parasites de l'élément réel avec le contour du modèle CAO. Le premier taux doit dans le pire des cas être au moins supérieur ou égal à 50 % et le deuxième ne peut en aucun cas atteindre les 10 %.

3 Résultats préliminaires

La méthode proposée dans cet article a été développée en C++ avec la librairie OpenCV [24] et testée sur un ordinateur équipé du processeur Intel Xeon quad core, de 3 GHz et de 12 Go de RAM. Quelques résultats sont présentés dans les figures 11 à 15 ci-dessous.

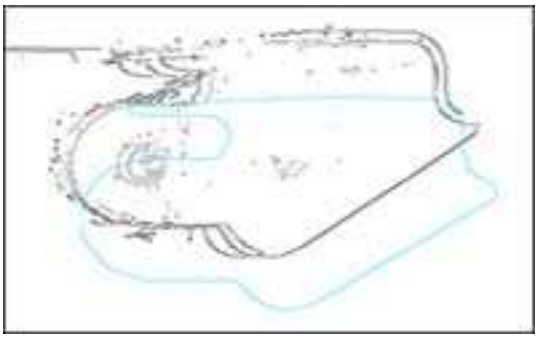


FIGURE 11 – Position initiale d'un élément réel par rapport à un élément de référence.

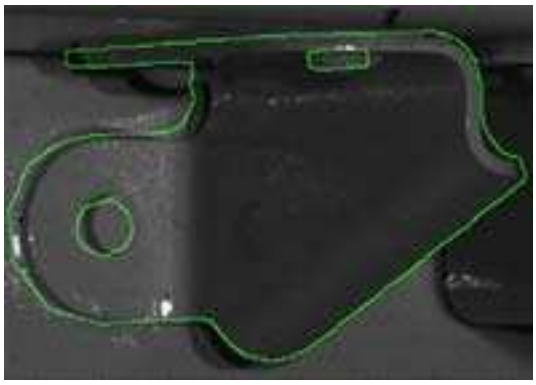


FIGURE 12 – Position finale d'un élément réel par rapport à un élément de référence.

Après l'optimisation, le taux de recouvrement entre le contour du modèle CAO et le contour de l'élément réel pour cet élément est de 52 % avec un taux de bruit interne de 3 %. L'élément inspecté est accepté. Un autre exemple est montré dans la figure 13, où les deux seuils fixés sont respectés.

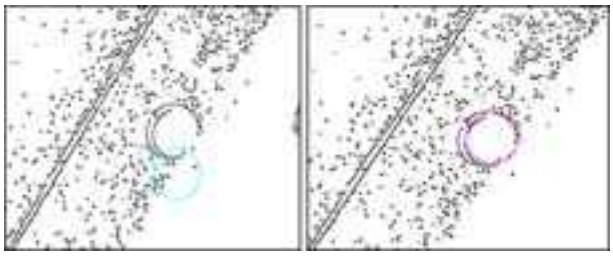


FIGURE 13 – A gauche, état initial avant optimisation, à droite, état final après optimisation.

Dans cette exemple, après l'optimisation, le taux de recouvrement entre le contour du modèle CAO et le contour de l'élément réel est de 85 %, avec un taux de bruit intérieur presque nul. En revanche, si l'un des deux seuils n'est pas respecté alors l'élément inspecté est refusé (cf. figures 14, 15). L'inspection d'un élément différent de celui attendu est détectée dans figure 14.

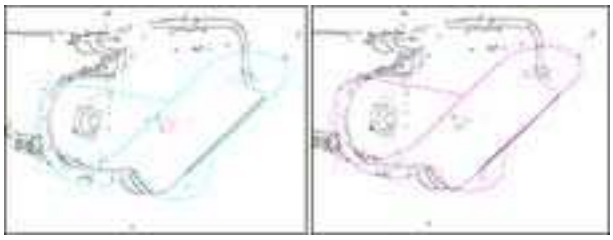


FIGURE 14 – A gauche, état initial avant optimisation, à droite, état final après optimisation.

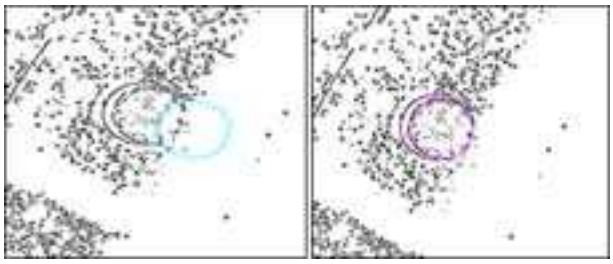


FIGURE 15 – A gauche, état initial avant optimisation, à droite, état final après optimisation.

Dans cette exemple, après l'optimisation, le taux de recouvrement entre le contour du modèle CAO et le contour de l'élément réel est de 97 %. Toutefois cet élément est refusé car le taux de recouvrement des contours parasites à l'intérieur du modèle CAO égal à 15 % dépasse le seuil des 10 % fixés. C'est un cas de fausse alarme qu'il faudra régler par la suite. Même si l'élément est bon, avec un taux de recouvrement de bruit de 15 % il ne serait pas fiable d'accepter cet élément.

En termes de comparaison avec la méthode présentée dans la section 1, les tests ont montré que la méthode présentée dans cet article prend en moyenne une seconde pour inspecter un élément alors que la méthode basée sur la 3D pure prend 10 secondes pour inspecter le même élément. En termes des vrais positifs, la méthode proposée est aussi performante que la méthode basée sur la 3D pure, avec l'avantage d'être dix fois plus rapide. Parfois, à cause d'une mauvaise initialisation, la descente de gradient peut rester bloquée dans un minimum local indésirable. En conséquence quelques cas de faux négatifs ont été constatés. L'utilisation d'un algorithme d'optimisation plus robuste, une approche probabiliste par exemple au lieu d'utiliser la descente de gradient pourrait résoudre ce problème.

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

4 Conclusion

Dans cet article, une nouvelle méthode d'inspection automatisée a été présentée, basée sur l'utilisation de contraintes 3D pour faire de l'inspection en utilisant des techniques d'analyse d'images 2D. Les expériences ont montré que cette méthode est plus rapide qu'une méthode basée sur la pure 3D, tout en étant fiable.

Actuellement, la méthode proposée dans cet article est capable de détecter la rotation 1D et la translation 2D d'un élément réel, en prenant un élément idéal comme référence. Des résultats encore meilleurs pourraient être atteints si les six degrés de liberté (3 rotations + 3 translations, selon les axes X, Y, Z), étaient pris en compte, en générant directement des contours théoriques à partir des différentes positions du modèle CAO en 3D, au lieu d'effectuer des rotations 1D et des translations 2D.

La détection des contours peut être améliorée afin de mieux traiter les images très bruitées. Une méthode d'optimisation plus robuste peut être envisagée afin de prendre en compte la présence des minima locaux dans la fonction optimisée.

Rajouter une étape de pré-traitement, pour vérifier la présence d'un élément, avant de lancer l'optimisation pourrait être utile pour la robustesse de la méthode, voire économique en temps de calculs sur certains cas. A ce propos, des travaux préliminaires, basés sur la transformée de Hough généralisée sont en cours, avec des résultats encourageants.

A l'avenir, il est envisagé d'utiliser les deux techniques, soit l'inspection basée sur la 2D pour les cas les plus favorables, pour des raisons de temps de calcul, et la 3D lorsqu'il n'est pas possible d'utiliser l'approche basée sur le 2D. Le couplage de ce deux types de techniques permettra une utilisation optimale des données disponibles pour faire de l'inspection fiable et rapide.

5 Remerciements

Ce travail fait partie du projet CAAMVis (AEROSAT 2011). Nous tenons à remercier l'État français (Fonds National pour l'Aménagement du Territoire et du Développement - FNADT) pour son soutien financier.

Références

[1] Timothy NEWMAN et Anil K. JAIN : A survey of automated visual inspection. *Computer Vision and Image Understanding*, volume 61, 1995.

[2] Jacques-Gérard POSTAIRE : *Les systèmes de vision : Chapitre 10. Vision industrielle et contrôle qualité*. Hermes Science Publications, décembre 2000.

[3] Véronique MORON : *Mise en correspondance de données 3D avec un modèle CAO : Application à l'inspection automatique*. Thèse de doctorat, INSA Lyon, 1996.

[4] Flavio PRIETO : *Métrologie Assisté par Ordinateur : Apport des capteurs 3D sans contact*. Thèse de doctorat, INSA Lyon, 1999.

[5] Bharath MODAYUR, Linda SHAPIRO et Robert HARALICK : Visual inspection of machined parts. *Computer Vision and Pattern Recognition*, juin 1992.

[6] A. EDINBAROUGH, R.NAINAR et P. RADHAKRISHNAN : Automated visual inspection of geometric tolerance in circular components. volume 10, pages 367–373, 1995.

[7] Jacob BARHAK : Utilizing computing power to simplify the inspection process of complex shapes. *Engineering Research Center for Reconfigurable Manufacturing Systems University of Michigan*, 2004.

[8] Elias MALAMAS, Euripedes PETRAKIS, Michalis ZERVAKIS, Laurent PETIT et Jean-Didier LEGAT : A survey on industrial vision systems , applications and tools. *Image and Vision Computing*, volume 21, pages 171-188, février 2003.

[9] Timothy NEWMAN et Anil K. JAIN : A system for 3D CAD-based inspection using range images. mars 1994.

[10] Glenn TARBOX et Susan GOTTSCHLICH : IVIS : an Integrated Volumetric Inspection System. *In Vision Workshop, 1994*, pages 220–227, février 1994.

[11] Abdelhalim BOUTARFA, Nour-Eddine BOUGUECHAL, Yassine ABDESSEMED et Redarce TANNEGUY : Pattern Recognition in Computer Integrated Manufacturing. *Journal of Electrical Engineering*, volume 57, pages 28-35, 2006.

[12] Christian WOHLER : *3D Computer Vision Efficient Methods and Applications : Part II Application Scenarios, 5 Applications to Industrial Quality Inspection*. Springer, 2009.

[13] John CANNY : A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8, 1987.

[14] Linda SHAPIRO et Georges STOCKMAN : *Computer Vision, 5.8 : The Canny Edge Detector, 10.3.2 :The Canny Edge Detector and Linker*. Prentice Hall, 2001.

[15] J. SAUVOLA et M. PIETIKAKINEN : Adaptive document image binarization. *the journal of the Pattern Recognition*, 2000.

[16] Faisal SHAFAIT, Daniel KEYSERS et Thomas M. BREUEL : Efficient implementation of local adaptive thresholding techniques using integral images. *Document Recognition and Retrieval XV*, 6815, janvier 2008.

[17] Carsten GARNICA, Frank BOOCHS et Marek TWARDOCHLIB : A new approach to edge-preserving smoothing for edge extraction and image segmentation. *IAPRS*, XXXIII, 2000.

[18] A. ROSENFELD et J.L. PFALTZ : Distance functions on digital pictures. *Pattern Recognition*, volume 1, pages 33-61, 1968.

[19] Antoni MOORE : The case for approximate distance transforms. *The 14th Annual Colloquium of the Spatial Information Research Centre, University of Otago, Dunedin, New Zealand*, décembre 2002.

[20] Jufu Feng LIWEI WANG, Yan Zhang : On the euclidean distance of images. *IEEE Transactions, Pattern Analysis and Machine Intelligence*, 2005.

[21] Linda SHAPIRO et Georges STOCKMAN : *Computer Vision, chapter 11 : Matching in 2D, chapter 13 : Sensing and objet pose computation*. Prentice Hall, 2001.

[22] Diane LINGRAND : *Introduction au traitement d'images : chapitre 5, transformations géométriques*. Vuibert, 2010.

[23] David JACOBS : Image gradients class notes for CMSC 426, 2005. *http ://www.cs.umd.edu/ djacobs/CMSC426/ImageGradients.pdf, consulté 21 janvier 2012*.

[24] Robert LAGANIÈRE : *OpenCV 2 Computer Vision Application Programming Cookbook*. Packt Publishing, mai 2011.