Robust 3D/2D hybrid registration for integrating a priori CAD model into X-ray Computed Tomography

Victor Bussy, Caroline Vienne

Robust 3D/2D Hybrid Registration for Integrating a priori CAD model into X-ray Computed Tomography

V. Bussy C. Vienne

Université Paris-Saclay, CEA, LIST, F-91190, Palaiseau, France
victor.bussy@cea.fr

Résumé

Le positionnement d'un modèle CAO d'une pièce est effectué à partir de projections expérimentales par rayons X (RX) afin de pouvoir l'utiliser comme information a priori d'une reconstruction tomographique RX. La méthode proposée est un asservissement visuel hybride en points ainsi qu'en moments d'images. La robustesse et la précision de la méthode sont testées sur six objets de formes variées en considérant à chaque fois une centaine d'initialisations différentes. Enfin, un pas vers l'automatisation complète du procédé est envisagé.

Mots Clef

Rayons X, Recalage 2D/3D, Estimation de Pose, Moments d'images, Asservissement Visuel

Abstract

In this paper, a 2D/3D image registration is performed between X-ray projections of an object and its CAD model to match the world and the CAD coordinate framework. This operation is needed to apply a mask on voxels during the tomographic reconstruction. The method presented is based on visual servoing, whose features are points and image moments. The robustness and precision of the algorithm are tested on six objects. Finally, a step towards the complete automation of the process is envisaged.

Keywords

X-ray, Image Registration 2D/3D, Pose Estimation, Image Moments, Visual Servo Control

1 Introduction

X-ray Computed Tomography (CT) is now a standard non-destructive technique to control and inspect objects in three dimensions. However, this technique still suffers from a very long acquisition time since more than one hour is typically needed for acquiring the hundreds of radiographic images (or projections) used in the reconstruction process. This work describes the first step of a more general methodology, whose aim is to adress the specific challenge of the deployment of CT in the production lines by reducing the number of acquired projections by a factor 10. To achieve this goal, we propose to use in the reconstruction algorithm a prior information coming from the geometrical model (the so-called CAD model) of the object and use it as a mask. In other words, instead of considering a cubic matrix of voxels, we propose to use the CAD information, when available, as a mask inside of which the reconstruction is performed. The number of unknowns is therefore highly reduced, which allows a good quality of reconstruction from few projections. [1]. A prerequisite for applying the mask is to perfectly position the CAD model according to the real object's position. The CAD coordinate framework should match the world framework so that the projection of the CAD's edges are superimposed to the ones of the experimental image. Many studies have already addressed this *Pose Estimation problem*, but lots use extrinsic markers, which is an important constraint.

This paper focuses on solving a pose estimation problem without fiducials and using an optimiser based on visual features. We propose to adapt existing approaches developed for cameras to our case due to the high similarity between X-ray images and visual images formation principle. In section 2, we briefly review the existing methods for solving the pose estimation problem. Section 3 details the visual servo-control, and section 4 describes our approach to the problem. Finally, in section 5, we present the performance of our strategy.

2 Methodology

2.1 Pose estimation

Pose estimation problems early appeared in many areas. The need to match the world coordinate frame and the simulation frame came out like an inevitable issue. In medical robotics, image registration enables image-guided radiation therapies, and surgeries from its 2D radio images [2]. In augmented reality, this technology allows the insertion of virtual objects into the world frame [3].

This paper focuses on matching a given CAD model of any object to its position in real life. This problem means finding a rigid transformation $\mathbf{T}$ between the world and CAD coordinate frameworks, which aligns the given experimental X-ray projection of an object to its simulated projection. Figure 1 illustrates the pose estimation problem in the particular case of this study.

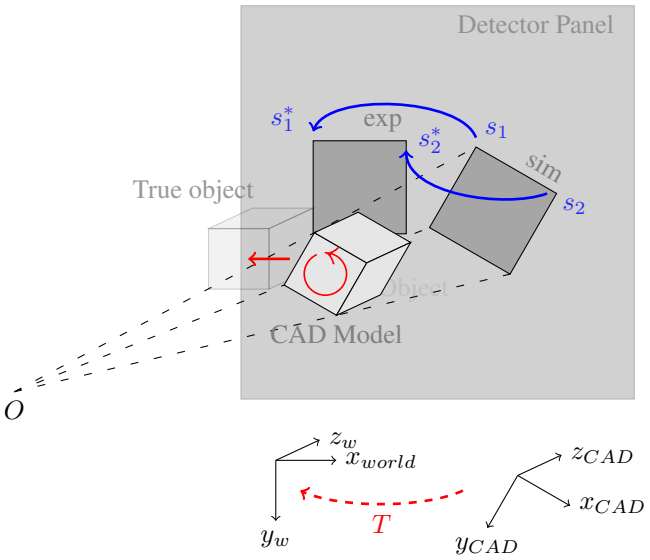


Figure 1: Illustration of the Pose Estimation Problem (*exp* corresponds to the experimental X-ray image of the object. *sim* stands for the simulated projection of the CAD model. Red arrows represent the **T** transformation, made of a rotation and a translation. The blue arrows illustrate the pairing between points from simulated and target projection.)

When a 3D model of the object is available, *Direct Linear Transform* and *Perspective-n-Point* (PnP) methods are the most common solutions [3]. From a set of *n* correspondences between 3D points and their projection, one can find the camera pose (i.e. the 6 degrees of freedom: roll, pitch, yaw and three translations). PnP algorithms have many variants. Some do not require prior correspondences but are too computationally expensive for complex objects [4]. Another kind of method to match a CAD model onto reality is *iterative inverse perspective matching* [5]. 2D points from the image projection and CAD 3D points are paired, and the cost function to minimise is defined as the distances between 3D points and projection rays. This algorithm is inspired by Iterative Closest Points methods. Unfortunately, as the CAD model can be arbitrarily complex, this method can be quickly computationally too expensive.

Markerless solutions have emerged, tracking methods based on edges-detection or features-detection have been widely developed to achieve good performance in real-time [6]. The main drawback of those methods is that they demand a pre-processing of the CAD as they deal with features (lines, planes,...) and thereby lack generality.

2.2 Visual Servo Control

Though classically used to perform positioning or tracking tasks, the *visual servo-control* strategy is also well-fitted for pose estimation when combined with a reliable simulation environment. Those methods have the advantages of being robust thanks to the closed control loop and can be easily generalised to any object.

Our study aims to propose a robust method, applicable to all types of objects, fast, and with limited user intervention. As this process can be automated, the choice of visual servoing was made. Visual servo-control refers to a method in which a visual sensor is used to control the movements of a dynamic system with a closed-loop approach. Most of the time, a set of features called $\mathbf{s}^*$ is extracted from an experimental image and compared to the same set of features, called $\mathbf{s}(\mathbf{x})$, from the projection of the 3D model. The vector $\mathbf{x} = (x, y, z, \omega_x, \omega_y, \omega_z)$ represents the pose of the camera along the axis O_x, O_y, and O_z. Visual servo-control aims to minimise the error $\boldsymbol{e}$:

$$\mathbf{e} = (\mathbf{s} - \mathbf{s}^*) \tag{1}$$

The principle of visual control is that each feature is associated with an *interaction matrix* $\mathbf{L_c} = \frac{\partial \mathbf{s}}{\partial \mathbf{x}}$, also called *feature Jacobian*, first introduced by [7]. This matrix describes the variation of the features $\mathbf{s}$ according to the camera movement and is, along with features' choice, the keystone to efficiently pilot the system. It allows determining the camera kinematic screw $\mathbf{v}$ to apply to reduce the visual error.

$$\begin{aligned} \frac{d\mathbf{s}}{dt} &= \frac{\partial \mathbf{s}}{\partial \mathbf{x}} \frac{d\mathbf{x}}{dt} + \frac{\partial \mathbf{s}}{\partial t} \\ \frac{d\mathbf{s}}{dt} &= \mathbf{L_c v} + \frac{\partial \mathbf{s}}{\partial t} \end{aligned} \tag{2}$$

As the object is motionless, the last term is zero. If an exponential decrease of the error is imposed,

$$\frac{d\mathbf{e}}{dt} = -\lambda \mathbf{e} \tag{3}$$

$$\frac{d\mathbf{e}}{dt} = \frac{\partial \mathbf{e}}{\partial \mathbf{s}} \frac{\partial \mathbf{s}}{\partial \mathbf{x}} \frac{d\mathbf{x}}{dt} = \mathbf{L_c v} \tag{4}$$

Where λ is a positive constant. Then, the camera kinematic screw can be expressed as:

$$\mathbf{v} = -\lambda \mathbf{L_c}^{+} (\mathbf{s} - \mathbf{s}^*) \tag{5}$$

Where $\mathbf{L_c}^{+}$ is the *Moore–Penrose inverse* of the interaction matrix. A closed-loop servoing is used to minimise the error of the selected features. The vector $\mathbf{v}$ is calculated to lead the camera and progressively match world and CAD coordinate frameworks at each iteration.

3 Choice of visual features

3.1 Contour Points as Features

As the features used in servoing are not iconic, it is possible to work with binary images. This strategy is

all the more advantageous because it is simple. Indeed, producing realistic simulated projections of a CAD object which can be compared to the experimental ones is tricky. It requires several parameters to adjust [8]. The considered approach consists of projecting the points of the CAD model onto the detector plane. Then, the simulated hull of the object is compared to the experimental image after binarization. Therefore, contours points have been perceived as the most natural and straightforward features. Moreover, in our considered application case of in-line CT inspection, the object is moving on a conveyor belt and is all alone in the field of view of the detector, which makes this binarization step easy to perform.

The interaction matrix for a point (x,y) on the projection corresponding to the point (X,Y,Z) in the object is given by :

$$\mathbf{L_X} = \begin{pmatrix} -\frac{d}{Z} & 0 & \frac{x}{Z} & \frac{xy}{d} & -\frac{d^2+x^2}{d} & y \\ 0 & -\frac{d}{Z} & \frac{y}{Z} & \frac{d^2+y^2}{d} & -\frac{xy}{d} & -x \end{pmatrix} \quad (6)$$

Where d is the distance between the source and the detector. A rough approximation of Z is often enough to achieve good convergence.

The matching between feature points $\mathbf{s}^*$ extracted from the experimental image and the current feature point $\mathbf{s}$, as illustrated by Figure 1, is done by the Iterative Closest Point algorithm (ICP) [9].

However, even if contours points are often numerous, the interaction matrix may become singular [10]. Moreover, detecting contours can be difficult for some objects, and matching points when the object has been rotated around O_y is often impossible. Thereby, the idea of using other features like lines or ellipses has been explored. Those approaches are sometimes more efficient , but prior knowledge of the studied object is needed. More generic features like image moments have finally been considered [11].

3.2 Image Moments as Features

In discrete space, image moments of order (i+j) are defined as:

$$M_{ij} = \sum_x \sum_y x^i y^j I(x,y) \quad (7)$$

Where $I(x,y)$ represents the pixel intensities. Given H the projection of the object on the detection image, we choose:

$$I(x,y) = \begin{cases} 1 \text{ if (x,y)} \in H \\ 0 \text{ else} \end{cases}$$

Image moments are computationally fast to calculate and can have physical interpretations. The fact that image moments rely on the whole image makes those features less sensitive to noise and removes the need for a matching algorithm. The biggest challenge with image moments servoing is to choose them. A well-known sufficient condition to achieve stability in the system is the interaction matrix's positive definiteness, but this condition is hard to use in practice [12]. Most of the time, studies focus their features choice according to the decoupling of the interaction matrix. A full rank, decoupled feature jacobian with a low condition number would lead to optimal movements.
Commonly, the area and centre of gravity (moments of order zero and one) are used to control translations and the orientation $\alpha = \frac{1}{2}\arctan\left(\frac{\mu_{11}}{\mu_{20}-\mu_{02}}\right)$ (combination of moments of order two) to control the rotation around O_z (see [13] for more details).
Howbeit, at least two other features are needed to control the rotations around O_x and O_y. To have an interaction matrix of rank six, one needs to use moments of order higher than three since symmetries cannot be properly handled by moments of order two. Features invariant to scale, translation or rotation, can be used to decouple $\mathbf{L_c}$.

4 Our method: A hybrid approach for increased robustness

4.1 Combining points and moments features in a dual-view approach

In this work, we propose to use both contours points and image moments. Those features are convenient for most objects. Nonetheless, symmetries can lead to wrong image registrations as the same projection can correspond to different positioning. More visual information is needed to avoid those issues. Thereby, two perpendicular views are now used for pose estimation. Multicamera systems are already commonly used for image-based visual servoing. The interaction matrix of the transverse view is expressed in the frame of the first view with:

$${}^1L_{c_2} = {}^2L_{c_2} \cdot {}^2W_1 \quad (8)$$

$${}^2W_1 = \begin{pmatrix} {}^2R_1 & [{}^2t_1]_\times \, {}^2R_1 \\ 0 & {}^2R_1 \end{pmatrix} \quad (9)$$

Where $[\mathbf{t}]_\times$ is the the skew-symmetric matrix associated with the vector $\mathbf{t}$ and $({}^2\mathbf{R}_1, {}^2\,\mathbf{t}_1)$ is the rigid-body transformation between the coordinates frames, ${}^1L_{c_2}$ and ${}^2L_{c_2}$ are the interaction matrices of the second view in the front and its own frames, respectively.

Our proposed algorithm (see 1) uses successively contour points and image moments. The control with points being much more robust, the first iterations are done with points as features, then once a position close to the final position has been roughly approached, image moments are used. A visual error is calculated between the simulated and target projections to measure how the registration performs and to know when the preliminary rough convergence is reached. The visual error represents the percentage of pixels mismatched between the two images.

A new approach is proposed for the choice of moment features. Usually, as described above, the user seeks to decou-

ple the interaction matrix. In this algorithm, translations are managed thanks to the centre of gravity of the object shape in the two views. In addition, to direct the rotations, the most discriminating features are used. Among a previously defined set of image moments (area, centroid, central normalized moments and Hu's moments of order three and less), the features which have converged the least are used for the servoing. Those features are considered more informative and discriminant, thus allowing a more precise registration. If the interaction matrix is singular or ill-conditioned, more features are added. Usually, each view uses the centroid coordinates and four additional features.

4.2 Improving local minima handling

As stated by [10], some configurations such as a pure rotation of π around O_z can lead to an infinite camera retreat. Convergence to local minima also occurs. Two solutions have been used. The first one inspired by [14] and [15] decouples rotations and translations to avoid an infinite camera retreat. In this algorithm, calculated movements are not applied to the camera, but the inverse movements are applied to the object in the framework centred on itself. The most significant advantage of this method is that rotations are now decoupled from translations. Worst initial positions can now be handled as more significant rotations can be applied. Moreover, this method tends to keep the object in the field of view of the camera.

Another improvement to handle poor initial positions is to test different viewpoints before starting the control loop. The algorithm simulates some projections from views around the initial position and then starts the servo-control from the best one. There are many merit functions to find the best view [2]. Here, the best view is not chosen according to the moments but according to the ICP algorithm's ability to establish correspondences. This metric does not quantify the resemblance of two images but rather if the points to be matched by the ICP are present on the two images. Indirectly, it transcribes the correctness of the rotation around O_y because rotations around this axis hide points.

As the initialisation step improved the behaviour significantly, the same strategy was implemented during the control loop. The algorithm compares the current point of view with two other ones rotated along O_y and picks the best view every ten iterations. This process, further mention as *simulated annealing*, allows to explore positions and avoid local minima.

5 Results and Analysis

5.1 Overall performance

The method has been tested on six objects with different geometries: a screw, a car, an engine, a fox, a curved grate and the statue of the Nike of Samothrace (see Annexe). The magnification parameter is about 1.5 in every situation. For each object, 100 tests were made with random initial positions ranging in $\pm\pi/$ 3 radians around each axis real position and an initial error on translations of $\pm$ 2 cm along each axis as well. A translation of 2 cm corresponds, in our case, to a translation of 77 pixels. As the field of view is relatively small, it represents a significant error. The results are shown in Table 1. For each object, the visual error and its standard deviation are given for both views. The success rate is the frequency at which the procedure resulted in an accurate registration. For each object, a criterion on the visual error has been previously chosen so that a successful registration corresponds to an almost perfect superposition of the CAD and the experimental image.

Table 1: Results and Performance

Objects	Visual error Front view	Visual error Transverse view	Success Rate[%]
Fox	1.30 ± 8.81	0.60 ± 2.06	98
Engine	0.58 ± 1.27	0.39 ± 0.73	80
Screw	6.67 ± 16.01	3.65 ± 9.84	81
Grate	2.37 ± 4.52	3.46 ± 4.80	89
Car	2.57 ± 3.77	2.25 ± 3.01	88
Statue	0.77 ± 0.51	0.87 ± 0.69	100

The method shows good performances on all the studied configurations despite the wide range of initialisations. The differences observed can be explained by the presence of angular contours or symmetries in the projections. Fox and statue show a higher success rate because they are asymmetrical and of approximately constant thickness. To control the system with images moments, the hypothesis that the object is plane has been made (see [11]). This assumption is not verified in any of the cases, especially for the grate and the car. Most objects have projections with symmetries, leading to a more difficult servoing with image moments, especially when the object is square-shaped (α is not defined). The car and the grate have internal contours, which allow better points matching with the ICP. The screw is a particular case as the object has symmetry around its main axis. Therefore, the transformation $\mathbf{T}$ sought is not unique. Moreover, the thread of the screw makes the pose estimation difficult.

In addition to being robust, the algorithm allows accurate registration. Table 2 shows the average errors on position and orientation for the six studied objects when convergence is achieved. Despite the large standard deviations, one can see that the pose estimation error on translations is always below the millimetre and the orientation error is under a degree.

To illustrate the operation of the method, the results on a particular case are detailed. Figure 2 shows the movement

Table 2: Errors at convergence

Objects	Error on Translations [mm]	Error on Rotations [°]
Fox	0.47 ± 0.10	0.115 ± 0.286
Engine	0.60 ± 0.43	0.859 ± 0.802
Grate	0.73 ± 0.59	0.343 ± 0.171
Car	0.83 ± 0.55	0.859 ± 0.630
Statue	0.69 ± 0.36	0.343 ± 0.229

of the fox between its initial point and its target position. First, the initialisation procedure has found a better starting point than the initial (see Figure 7 and 8). Then, at the tenth iteration, the simulated annealing procedure rotates around O_y, bringing the fox near its final position. The effects of the simulated annealing can also be seen in Figures 3 and 4. These figures represent the convergence of features from the front and transverse views. At the 21^{st} iteration, the layering score was low enough to approximate convergence with points, so image moments-based control started. This change is visible on the slopes of figure 2. Note that at the 50^{th} and 60^{th} iterations, the simulated annealing changed the path of the object. This change brought the object slightly closer to its target position. The features ϕ_1 and hu_1 in the front view got better while they worsen in the transverse view. Even if the criterion for the choice of views during the simulated annealing considers the two projections, one may be favoured.

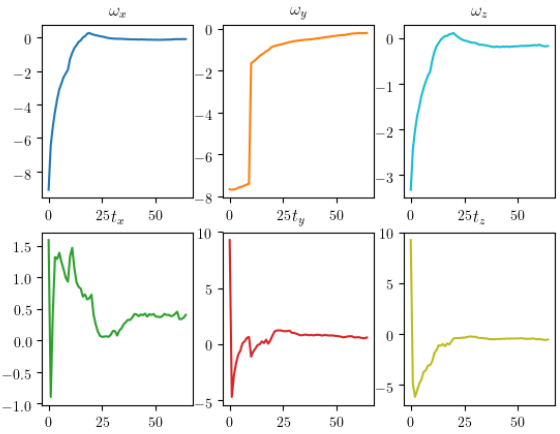


Figure 2: Object's path. The initial point is ($\omega_x = -42.9°$, $\omega_y = 26.9°$, $\omega_z = 31.5°$, $t_x = 1.59$mm $t_y = 9.28$mm, $t_z = 9.22$mm) and the target position is ($\omega_x = \omega_y = \omega_z = 0.00°$, $t_x = t_y = t_z = 0.00$mm)

5.2 Purpose of image moments as features

In all of the examples, good convergence can be achieved with contour-based control. However, this technique rarely allows for optimal convergence. ICP algorithm often results in mismatches that help the first few iterations to get the object roughly to its target orientation but does not achieve the desired final accuracy. Nevertheless, these mismatches are helpful because they allow the servo-control loop to be robust and handle positions where the image moments only would fail to converge.

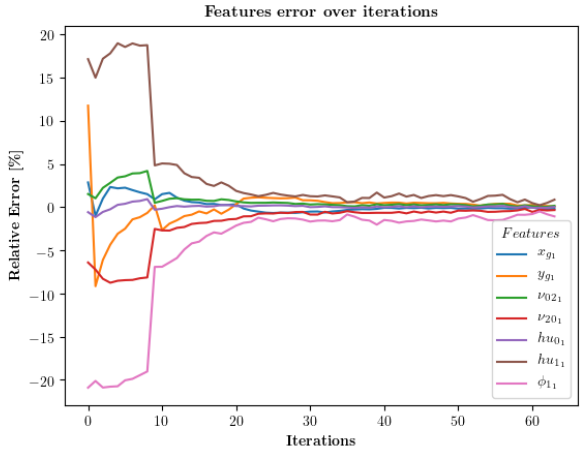


Figure 3: Features error over iterations from the front view

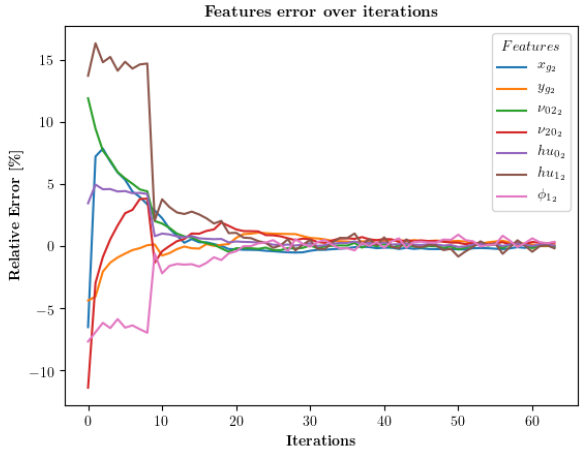


Figure 4: Features error over iterations from the transverse view

To illustrate the interest of combining the two types of features, Figure 5 shows the final layerings between experimental projections and simulated ones when only contours are used. The registration is satisfactory but not optimal. Figure 6 shows the final layerings when both points and image moments are used. The results are better with the hybrid method. Contours match almost perfectly.

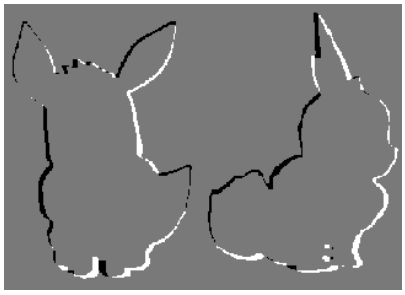


Figure 5: Final layerings with points as features (70 iterations)

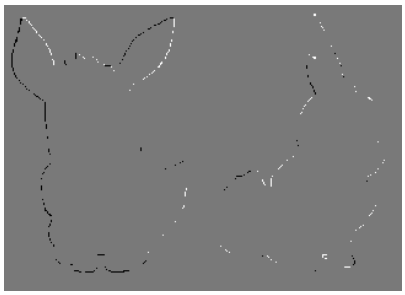


Figure 6: Final layerings with points then moments as features (70 iterations)

Figure 7: Initial position ($\omega_x = -42.9$ °, $\omega_y = 26.9$ °, $\omega_z = 31.5°$, $t_x = 1.59$mm $t_y = 9.28$mm, $t_z = 9.22$mm)

Figure 8: Position after initialization step ($\omega_x = -8.59°$, $\omega_y = -7.66°$, $\omega_z = -3.32°$, $t_x = 1.59$mm $t_y = 9.28$mm, $t_z = 9.22$mm)

5.3 Initialisation and Simulated annealing

As stated above, to choose the best starting point or for the simulated annealing, the algorithm picks the view which minimises the distance between the matched points for the ICP algorithm. This choice is made to promote the algorithm's robustness at the expense of the speed of convergence. A low distance indicates that the pairing of points has been done correctly, so the error vector $\boldsymbol{e}$ is well-calculated and that the object will be precisely positioned. This distance does not give information on the relevance of the current position. For example, a rotation around O_z shows the same distance as the target position with this criterion. To illustrate the usefulness of the initialisation step, Figures 7 and 8 show the layerings before and after initialisation for the same configuration as in 5.1. Initialisation greatly improves the image registration process.

To study the relevance of this choice, 100 tests carried out on each object to check whether the initialisation step has picked the best-envisaged position. The initialisation compares 27 different views, each made at either -35°,0°or +35°, along each axis of rotation around the initial position. This choice was made to cover the 120° range described earlier. Table 3 lists the initialisation step's failure rate, i.e. the frequency of an initial point choice that is not the closest point to the target. Then, the frequency of convergence of the points whose initialisation is not optimal is measured. The convergence rates are chiefly lower than those observed in Table 1. A better merit function for the next position choice could be found. However, it would be too quick to conclude that bad initialisations are the only ones responsible for convergence rate degradation. The points with a failed initialisation are often particularly far from the solution so that the ICP give incorrect results, and the method does not succeed.

Table 3: Initialisation tests

Objects	Failure rate [%]	Convergence rate [%]
Fox	9	77.78
Engine	12	75.00
Screw	42	71.42
Grate	39	89.74
Car	19	68.42
Statue	11	100

5.4 Automatic pose estimation

A new initialisation has been developed to perform without human intervention. Since the proposed method is generally functional over a wide range of initial positions, the objective is to position the object in this range before starting the servo-control loop. For this, the centroid of the CAD model is placed between the source and the detector automatically. Then, a search for the best starting position by dichotomy is done. Namely, the initialisation presented

above is used recursively, each time reducing the angle of aperture on each axis by two. When the aperture is below $\pi/3$, the recursion stops. Results are shown in Table 4.

Table 4: Automation of the process

Object	Visual error Front view	Visual error Transverse view	Success[%]
Fox	0.88 ± 2.34	1.03 ± 2.94	90

6 Conclusion

In this article, a 2D / 3D registration method has been proposed between X-ray projections and a CAD model using visual servoing. A first robust but rough registration is performed using contours points, then features based on image moments are considered to achieve the required accuracy. Increased robustness has been obtained thanks to a pose initialisation step to avoid local minimum. These improvements put together have resulted in a reliable, automatic registration method applicable to a wide variety of objects. This approach has been applied on simulated data and future works will consist in considering real experimental images for computing the desired features. This step should be straightforward since the chosen features are all computed from the object contour, which will be easy to detect in the radiographic images.

References

[1] Justin Park, Hao Zhang, Yunmei Chen, Qiyong Fan, Darren Kahler, Chihray Liu, and Bo Lu. Priori mask guided image reconstruction (p-mgir) for ultra-low dose cone-beam computed tomography. *Physics in medicine and biology*, 60:8505–8524, 10 2015.

[2] P Markelj, D Tomaževič, B Likar, and F Pernuš. A review of 3d/2d registration methods for image guided interventions. *Medical image analysis*, 16:642–61, 04 2010.

[3] E. Marchand, H. Uchiyama, and F. Spindler. Pose estimation for augmented reality: A hands-on survey. *IEEE Transactions on Visualization and Computer Graphics*, 22(12):2633–2651, 2016.

[4] Philip David, Daniel DeMenthon, Ramani Duraiswami, and Hanan Samet. Softposit: Simultaneous pose and correspondence determination. *International Journal of Computer Vision*, 59, 09 2003.

[5] P. Wunsch and G. Hirzinger. Registration of cad-models to images by iterative inverse perspective matching. *Proceedings of 13th International Conference on Pattern Recognition*, 1:78–83 vol.1, 1996.

[6] Tom Drummond and Roberto Cipolla. Real-time visual tracking of complex structures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):932–946, July 2002.

[7] L. Weiss, A. Sanderson, and C. Neuman. Dynamic sensor-based control of robots with visual feedback. *IEEE Journal on Robotics and Automation*, 3(5):404–417, 1987.

[8] Árpád Marinovszki, Jan De Beenhouwer, and Jan Sijbers. An efficient cad projector for x-ray projection based 3d inspection with the astra toolbox. NDT, 2018.

[9] P.J. Besl and Neil D. McKay. A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992.

[10] François Chaumette. Potential problems of stability and convergence in image-based and position-based visual servoing. *Robotics, IEEE Transactions on*, 237, 01 1998.

[11] François Chaumette. Image moments: A general and useful set of features for visual servoing. *Robotics, IEEE Transactions on*, 20:713 – 723, 09 2004.

[12] C. Samson and B. Espiau. Application of the task-function approach to sensor-based control of robot manipulators. *IFAC Proceedings Volumes*, 23(8, Part 5):269–274, 1990. 11th IFAC World Congress on Automatic Control, Tallinn, 1990 - Volume 5, Tallinn, Finland.

[13] Aurélien Yeremou Tamtsia. *Nouvelles contributions à l'application des moments en asservissement visuel*. Theses, Université Blaise Pascal - Clermont-Ferrand II, October 2013.

[14] Peter Corke and Seth Hutchinson. A new partitioned approach to image-based visual servo control. *Robotics and Automation, IEEE Transactions on*, 17:507 – 515, 09 2001.

[15] K. Deguchi. Optimal motion control for image-based visual servoing by decoupling translation and rotation. *Proceedings. 1998 IEEE/RSJ International Conference on Intelligent Robots and Systems. Applications (Cat. No.98CH36190)*, 2:705–711 vol.2, 1998.

Acknowledgement

This research was supported by DIMOFAC project (grant agreement No 870092, European Union's Horizon 2020 research and innovation programme).

Annexe

Algorithm 1: Hybrid method

Data: CAD file, 2 orthogonal experimental projections I_1 and I_2, distance d between the x ray source and the detector, detector size in pixels, initial position $\mathbf{T_0}$, threshold ϵ, step λ

Result: $\mathbf{T}$

compute the projections P_1 and P_2 at $\mathbf{T_0}$

$\mathbf{T}$, P_1, $P_2 \leftarrow \mathbf{T_0}$, Proj$_1(\mathbf{T_0})$, Proj$_2(\mathbf{T_0})$

$\mathbf{s_1^*}, \mathbf{s_2^*}$ = contours points from I_1 and I_2

$\mathbf{s_1}, \mathbf{s_2}$ = contours points from P_1 and P_2

$\mathbf{s}^*, \mathbf{s} = \begin{bmatrix}\mathbf{s_1^*} & \mathbf{s_2^*}\end{bmatrix}, \begin{bmatrix}\mathbf{s_1} & \mathbf{s_2}\end{bmatrix}$

iteration = 0

while *visual error* $> \epsilon$ **do**

- $L_c = \begin{bmatrix} L_{c1} \\ L_{c2}W \end{bmatrix}$
- $\mathbf{v} = -\lambda L_c^+(\mathbf{s} - \mathbf{s}^*)$
- $\mathbf{T} = \mathbf{T} + \mathbf{v} \cdot \Delta t$
- recompute P_1,P_2 and $\mathbf{s}$
- **if** *iteration % 10 ==0* **then**
  - update $\mathbf{T}$ with simulated annealing
- **end**
- iteration = iteration +1

end

$\mathbf{s_1^*}, \mathbf{s_2^*}$ = moments features from I_1 and I_2

$\mathbf{s_1}, \mathbf{s_2}$ = moments features from P_1 and P_2

$\mathbf{s}^*, \mathbf{s} = \begin{bmatrix}\mathbf{s_1^*} & \mathbf{s_2^*}\end{bmatrix}, \begin{bmatrix}\mathbf{s_1} & \mathbf{s_2}\end{bmatrix}$

while *visual error keeps decreasing* **do**

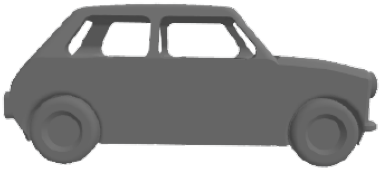
- $L_c = \begin{bmatrix} L_{c1} \\ L_{c2}W \end{bmatrix}$
- $\mathbf{v} = -\lambda L_c^+(\mathbf{s} - \mathbf{s}^*)$
- $\mathbf{T} = \mathbf{T} + \mathbf{v} \cdot \Delta t$
- $\epsilon = ||\mathbf{s} - \mathbf{s}^*||$
- recompute P_1,P_2 and $\mathbf{s}$
- **if** *iteration % 10 ==0* **then**
  - update $\mathbf{T}$ with simulated annealing
- **end**
- iteration = iteration +1

end

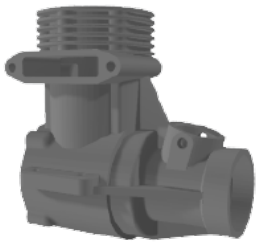
return $\mathbf{T}$

CAD Model of a curved grate from DIMOFAC project

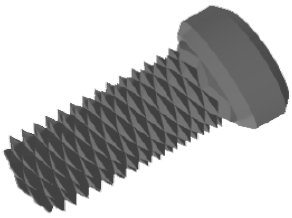
CAD Model of the fox Eevee



CAD Model of a car



CAD Model of a motorcycle engine



CAD Model of a screw

CAD Model of the Nike of Samothrace