

Article

Research on Assembly Method of Threaded Fasteners Based on Visual and Force Information

Yibang Zhou , Xiaoyong Wang and Lanzhu Zhang *

School of Mechanical and Power Engineering, East China University of Science and Technology, Shanghai 200237, China; y45190049@mail.ecust.edu.cn (Y.Z.); y84200007@mail.ecust.edu.cn (X.W.)

* Correspondence: lzzhang@ecust.edu.cn

Abstract: Threaded fastening operations are widely used in assembly and are typically time-consuming and costly. In low-volume, high-value manufacturing, fastening operations are carried out manually by skilled workers. The existing approaches are found to be less flexible and robust for performing assembly in a less structured industrial environment. This paper introduces a novel algorithm for detecting the position and orientation of threaded holes and a new method for tightening bolts. First, the elliptic arc fitting method and the three-point method are used to estimate the initial position and orientation of the threaded hole, and the force impact caused by switching from the free space to the constrained space during bolt tightening is solved. Second, by monitoring the deformation of passive compliance, the position information is introduced into the control process to better control the radial force between the bolt and the threaded hole in the tightening process. The constant force controller and orientation compliance controller are designed according to the adaptive control theory. A series of experiments are carried out. The results show that the proposed method can estimate the initial position and orientation of an M24 bolt with an average position error of 0.36 mm, 0.43 mm and 0.46 mm and an orientation error of 0.65° , 0.46° and 0.59° , and it can tighten the bolt with a success rate of 98.5%.

Keywords: threaded fasteners assembly; vision; force sensor; compliant control; position and orientation estimation



Citation: Zhou, Y.; Wang, X.; Zhang, L. Research on Assembly Method of Threaded Fasteners Based on Visual and Force Information. *Processes* **2023**, *11*, 1770. <https://doi.org/10.3390/pr11061770>

Academic Editors: Emad Abouel Nasr, Abdulrahman Al-Ahmari and Adham Ragab

Received: 19 May 2023
Revised: 4 June 2023
Accepted: 7 June 2023
Published: 10 June 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Industrial production is undergoing a transformation from mass general-purpose manufacturing to mass customization to meet the requirements of different scenarios. At present, most industrial applications achieve flexible production through specific end-effectors and efficient programming of robots. Moreover, in order to reduce the time and cost required to develop specific end-effectors, a flexible system should also be able to assemble different or the same series of products [1], and ensure minimal modifications to end-effectors and programs. More than a quarter of the assembly process in industrial manufacturing uses threaded fasteners [2]. Data show that nearly half of the parts used in the production process of an aircraft are threaded fasteners [3]. This shows that fasteners such as bolts and threaded holes play an important role in automotive, aerospace and other fields. However, threaded fastener assembly in some scenarios is easy for manual operation, but it is still very challenging to achieve automation. At present, many fastener assemblies are carried out in a semi-automatic manner, or simply with repeated automation on some assembly lines. Manual fastening enables the assembly of any variation in shape and position, but at the cost of reliability, such as consistency in applying tightening torque. Moreover, it has the disadvantages of low efficiency, high labor cost, high labor intensity, etc. In addition, under the harsh working conditions of high temperatures, high pressure, nuclear power and other environments, the manual nut tightening operation cannot be carried out. Therefore, there is an urgent need for a solution to realize the unstructured assembly of threaded fasteners, which can not only solve the problems of low efficiency

and poor reliability of traditional manual assembly, but also achieve high flexibility and adaptability. Such a solution requires both accurate estimation and orientation of the position of the threaded hole and monitoring of the bolt force and torque during the tightening process to achieve compliant assembly.

The purpose of this work is to develop an automated threaded fastener assembly solution that recognizes the position and orientation of threaded holes and that assembles fasteners (e.g., bolts) into threaded holes. The assembly platform of threaded fasteners is built in the laboratory environment. The test platform can carry out the real assembly process of threaded fasteners in different positions and angles, such as the mechanical assembly process commonly used in the automobile industry. At present, the work reported in this paper is still in the process of research. Its robustness and efficiency have been verified in the laboratory environment, and further work is needed before it can be applied to real industrial production.

2. Review of the Existing Solutions

The field of intelligent and compliant assembly has been developed for a long time, and some research results have been achieved. Compliant assembly technology is a kind of advanced assembly technology that can adapt to the requirements of rapid development and production, low cost manufacturing and modular recombination of equipment and tooling. Combined with digital technology and information technology, it forms a new field of automatic assembly technology. S. Huang [4] developed a method for direct admittance selection for force-guided assembly considering multiple-point contact. The fully compliant assembly system proposed by Finetto [5] and the hybrid flexible assembly system proposed by Rosati [6] solve the problem of frequent reconfiguration in small batch production. However, as described by Herakovic [7], the modern industrial assembly process requires the integration of target detection, pose estimation and advanced robot control methods to achieve the whole process.

At present, some progress has been made in compliant assembly based on 2D vision systems. Fei [8] developed a 2D automatic assembly method for puzzle assembly and successfully applied it in industry. The disadvantage of this approach is that it can only assemble objects of the same shape. Pitipong [9] proposed an automatic robotic screw fastening method. However, the location identified by this method is limited by the surrounding environment and mounting surface. Three dimensions have been used by visual systems to overcome the limitations of two-dimensional systems in a large number of studies. Bone [10] developed the visual guidance working unit for auto parts based on the robotic fixtureless assembly (RFA). Fleischer [11] developed a new approach based on the intrinsic markers of components, which is suitable for the spatial alignment of components. However, this is not an ideal solution for unstructured components.

Compliant control based on force feedback is another important technique in precision assembly. Skills such as insertion have force constraints so that the objects are not damaged during contact [12]. The force condition also determines the success of the skills such as grip [13]. For millimeter-sized objects, the influence of the adhesion force is not significant, and the force control mainly concerns the deformation and friction forces. The combined use of force, torque and vision sensors for tight-tolerance applications has also been investigated. Chen [14,15] developed a piston plug method based on 2D vision and a force–torque sensor. The piston was inserted into the valve body hole of an automobile’s transmission, and the force sensor was used to accurately control the piston insertion position. Lopez [16] proposed an intelligent assembly method using a 2D visual system and a learning system based on neural networks. The panel forming method proposed by Ilangovan [17] and the automatic polishing method proposed by Kalt [18] use visual and force sensors to correct the errors occurring in the manufacturing process. Mezouar et al. [19] developed the external/hybrid visual force control scheme to overcome the drawbacks of the hybrid and impedance-based vision–force schemes, for which they provided an exhaustive comparative analysis. In their approach, the external wrench is transformed into a displacement

of the image feature reference. Carelli et al. [20] proposed a hybrid force- and vision-based impedance controller to perform a peg-in-hole task. Posebased vs. (PBVS) is used to guide the end-effector toward the hole. Vision-based control methods are widely used in precision assembly systems. They are divided into three types, including position-based [21–24], image-based and hybrid schemes [25,26]. How to accurately estimate the orientation of small components is a major issue that needs to be explored in position-based methods. For image-based vision control systems, circular threaded holes are recognized as ellipses due to the distortion of the threaded hole image in the camera's field of view. Although many ellipse-fitting algorithms have been proposed, the structural characteristics of threaded holes can lead to more incorrect fitting during the ellipse fitting process. Therefore, the elliptic fitting scheme with arc support is a good choice. Hybrid force/position control methods are also often employed in precision assembly [27,28]. However, they are not suitable for our work because of the components sheltering each other, resulting in no available position information in the insertion process.

Although the aforementioned works present several paradigms, the following problems still remain:

- (1) The initial orientation of the threaded hole cannot be estimated. In the initial contact state between the bolt and the threaded hole, only the passive compliance control is used to realize the alignment of the initial orientation of the hole.
- (2) The initial position of threaded holes cannot be estimated. In [9,11], the threaded holes and assembled automobile parts are identified and detected, respectively, without modeling and fitting. Therefore, these two methods cannot be used for precision assembly in unknown and complex environments.
- (3) The allowed forces in assembly are small because the assembly precision of threaded holes is required to be higher. The criterion to determine the stiffness modulation factor in [29] is based on comparing the normalized variances of position and force, which might be improper when changes in position and force are both small.

The novelty of this paper is that it develops an efficient skill learning approach for precision assembly while considering and solving the above-mentioned problems.

- (1) A thread hole position and orientation estimation method based on image features is proposed to adjust the position and orientation of bolts when they initially contact threaded holes. The elliptic fitting method based on arc support and the three-point method are used to estimate the initial position and orientation of threaded holes, which solves the problem of force impact caused by switching from a free space to a constrained space during bolt tightening.
- (2) A compliant force controller composed of a constant force controller and an orientation controller is proposed to realize the compliant motion under the condition of a small contact force. First, the constant force control is realized in the X, Y and Z directions of the bolt, which can reduce the force impact at the moment of contact between the bolt and the threaded hole and improve the force tracking accuracy. Second, the orientation control is implemented at an Euler angle around the X, Y and Z directions of the bolt, so as to realize the robotic orientation compliance function in the assembly process, and finally to complete the compliant control in the tightening process.

3. System Configuration

The proposed tightening control strategy is discussed under the configuration of the basic assembly system shown in Figure 1, which consists of a workbench, a six-axis robot and its control system, a depth camera, a six-dimensional force sensor and a displacement sensor. The robot has six degrees of freedom and is responsible for inserting the bolt into the threaded hole. The force sensor is installed at the end of the robot and is used to detect the contact force and torque between the bolt and the threaded hole during tightening. The position and orientation of the bolts are adjusted in the X and Y directions based on force information to allow assembly with minimal resistance. There are four coil springs located between the force sensor and the clamp to play a passive compliance

role. The coil springs are chosen as a passive compliant device mainly because of their good designability, flexibility and linearity. For coil springs, their deformation can be estimated with displacement sensors. The displacement sensor is used to convert the deformation of the coil spring into the position information of the bolt during tightening. The orientation of the bolt and tightening tool is pre-adjusted so that their center axes are coaxial. The assembly system needs to be calibrated, including the internal parameter calibration of the camera, the hand-eye calibration of the camera coordinate system and the world coordinate system, and the rotation matrix from the force coordinate to the robot coordinate system. The contribution of this article is that it explores a precise assembly control strategy for threaded fasteners, mainly focusing on the alignment of threaded holes before insertion and the adjustment of bolt positions and orientations during the tightening stage. However, as discussed in the introduction, the significance of this paper is that it develops tightening control methods for precision assembly with strict requirements for threaded hole alignment. Thus, under the system configuration and previous basic assumptions, the assembly task is described as the incremental movement of the robot to progressively tighten the nut into the threaded hole based on visual and force information. In addition, the research of this paper is focused on the threaded fastener of threaded through holes, so it has a fixed threading sequence. The research and experiments in this paper follow the following assumptions:

- (1) The accuracy of the depth camera, displacement sensor and robot meet the requirements.
- (2) The required force and torque in the tightening process are within the range of the joint force and torque of the robot.
- (3) The role of the component thickness/depth of the threaded hole is ignored.
- (4) There is no relative displacement between the bolt and the fixture during tightening.

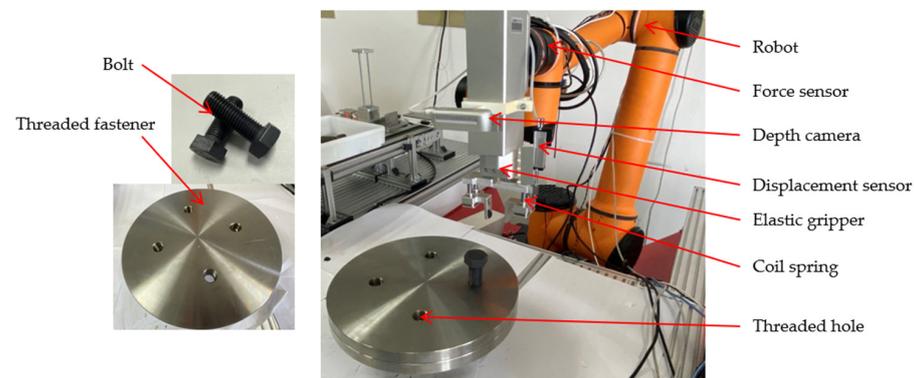


Figure 1. Test rig developed for automated assembly of fasteners.

4. Position and Orientation Estimation of Threaded Hole

Real threaded hole images are often affected by illumination, complex environments, camera positions and other factors. Edge information contains more noise and isolated points, and is often not the standard circle, which brings a lot of calculation and error information interference to the solution of the threaded hole position. At present, both the elliptic fitting method based on the Hough transform and the least squares method are realized under the premise of obtaining discrete edge points on the image. Therefore, an ellipse detection method based on arc segments is proposed. As early as 2012, Prasad et al. [30] proposed an ellipse detection algorithm combining arc segment extraction and the Hough transform, which improved the fitting efficiency of the Hough ellipse. In 2014, Fomaciari et al. [31] proposed an ellipse detection method based on the classification and screening of circular arc segments. This method first removes short arcs and straight lines in the set of circular arc segments, and then classifies them according to the concavity of arcs and the properties of gradient polarity, thus improving the efficiency of ellipse fitting. In 2018, Lu Changsheng et al. [32] proposed an elliptic detection method based

on arc supported line segments, which has high robustness and accuracy. In 2020, Meng Guo et al. [33] proposed an ellipse detection method based on edge connection, which was proposed to construct an adjacency matrix of an arc segment based on the AAM (Active Appearance Model) method of directed graphs and to screen out the final ellipse via clustering.

However, most ellipse detection methods based on arc segments have not improved ellipse detection methods for images containing multiple ellipse features under complex illumination. For such images, the extracted arc segments may come from the same ellipse, different ellipses or even incorrect edge curves (such as curve shadows caused by lighting on threaded fasteners). If the elliptic parameters of a single circular arc segment are fitted, the accuracy of the fitting results is low because a single circular arc segment cannot reflect the complete elliptic characteristics. If several arc segments are paired and combined for fitting, although the fitting accuracy is improved, it is easy to generate a lot of invalid ellipse information, which leads to the geometric verification algorithm of the ellipse occupying more time and resources, and it is not conducive to the implementation of real-time robot systems.

4.1. Improved Ellipse Fitting Algorithm for Arc Support Line Segments

The target of this article is threaded holes that contain multiple elliptical features (inner and outer loop of threaded holes, threads, etc.). Through research and analysis of the aforementioned ellipse detection methods, this article proposes an improved arc supported line segment fitting algorithm based on multiple elliptical feature images. The main workflow is as follows:

- (1) Arc segments are grouped according to the principle of continuity and convexity, the parameters of the ellipse are directly fitted to the arc support group with a large span, and the corresponding set of center points and center coordinates are obtained.
- (2) According to the distance between arc segments and the coordinates of the center of the circle, all sets of arc segments are classified, and subsets of arc segments of the same category are collected.
- (3) The classified arc segment set is combined and matched, and the parameters of the ellipse are fitted via the least squares method to obtain the initial ellipse information.
- (4) According to the geometric properties of ellipses, all candidate ellipses are verified to obtain the final candidate ellipses.

4.1.1. Extraction of Arc Support Line Segment

In the real image, the boundary of the ellipse consists of arc segments and a few discrete points. As the input source of ellipse parameters, the accuracy of arc segment extraction plays an important role in ellipse fitting. Therefore, arc segments were extracted using the Line Segment Detector (LSD) algorithm [34]. Based on the gradient direction of the image, the algorithm performs region segmentation, calculates the gradient direction of each pixel and forms a direction field. Pixels with similar gradients in the same direction field are then merged. By setting seed pixels and tolerance, the region growth algorithm is used to obtain a line and curve region in a gradient direction. The computational complexity of this method is only $O(n)$, and sub-pixel level line segments can be extracted within linear time (1 linear time). The curve edge after the removal of the line segments is an arc segment containing elliptic features, as shown in Figure 2, for specific steps.

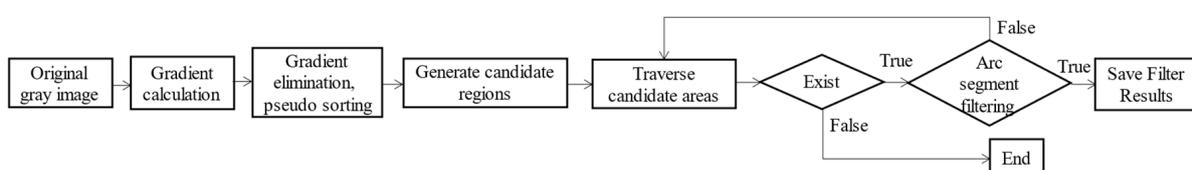


Figure 2. Flow chart of arc segment extraction algorithm.

When calculating the gradient, the gradient of each pixel in the image can be obtained through the 2×2 template in Table 1.

Table 1. Gradient calculation template.

$gray(x, y)$	$gray(x, y + 1)$
$gray(x + 1, y)$	$gray(x + 1, y + 1)$

$gray(x, y)$ represents the gray value of this pixel, and the gradient $g_x(x, y)$ of this point in the x direction is

$$g_x(x, y) = \frac{gray(x + 1, y) + gray(x + 1, y + 1) - gray(x, y) - gray(x, y + 1)}{gray(x, y + 1) + gray(x + 1, y + 1) - gray(x, y) - gray(x + 1, y)} \quad (1)$$

Therefore, the gradient $G(x, y)$ and gradient angle $level - line - angle(x, y)$ of this point are

$$\begin{cases} G(x, y) = \sqrt{g_x^2(x, y) + g_y^2(x, y)} \\ level - line - angle(x, y) = \left(\frac{g_x(x, y)}{g_y(x, y)} \right) \end{cases} \quad (2)$$

In the gradient diagram, the points whose gradient amplitudes are less than a certain threshold (set as 15% of the maximum gradient amplitude in this paper) are eliminated, and pseudo-sorting is performed first according to the magnitude of the gradient amplitude. The initial point (which is also the point with the most significant edge information) is selected in the gradient graph in the order of pseudo-sorting for region growth. The following formula is used to calculate the region angle of the candidate line segment region L_i .

$$Region\ Angle(L_i) = \arctan\left(\frac{\sum_j \sin(level - line - angle(p_j))}{\sum_j \cos(level - line - angle(p_j))}\right) \quad (3)$$

where p_j is the element in candidate region L_i . All candidate regions are traversed. When region L_i meets the following conditions, it is considered to be the candidate region of the arc support line segment:

- (1) The angle difference between two subregions L_i^1 and L_i^2 is within the angle tolerance, which is set as 22.5 in this paper.
- (2) The angles of candidate regions L_i, L_i^1 and L_i^2 are in the same direction.

Therefore, the image arc segment E_{arc} is extracted.

4.1.2. Grouping of Arc Support Line Segments

Arc segment e extracted in the previous step is an unconnected arc support segment, and the edge of the curve on the same ellipse may be composed of multiple arc support segments. Therefore, the purpose of arc segment grouping is to group the arc supporting line segments belonging to the same curve edge to form the arc supporting line segment group. The main steps are as follows:

- (1) Generate arc support line segment group

First, any arc segment in the E_{arc} set of arc segments is selected as the initial line segment L_{start} . The convexity of the edge of the curve is determined by the polarity of L_{start} , so the local neighborhood of the head and the local neighborhood of the tail of the L_{start} arc segment are determined. Then, the arc supporting line segment with the maximum number of pixels in the candidate region corresponding to the arc supporting line segment in the local neighborhood is counted. After the line segment is screened, the arc support line segment is connected with the initial line segment L_{start} . When all arc segments are connected, arc support segment group G_{arc} is obtained.

- (2) Calculation of crossing angle of arc support line segment group

In order to obtain the reflection degree of the elliptic features of the arc supported line segment group, the spanning angle k of each subset in G_{arc} is calculated. As the crossing angle becomes larger, the reflection degree of the ellipse features becomes higher, and the accuracy of the fitting of ellipse parameters becomes higher.

4.1.3. Elliptic Parameter Fitting

After the arc support line segment group G_{arc} is obtained via the previous step, the subset whose crossing angle of the connected arc support line segment in the group is greater than a certain threshold (set as 80 in this paper) is obtained. Then, the least squares elliptic fitting is carried out directly on it, and the elliptic parameter after fitting is $S = \{s_1, s_2, \dots, s_n\}$. k represents the number of fitted ovals, and s_i consists of five parameters of the ellipse (x_0, y_0, a, b, θ) .

For threaded fasteners, multiple ovals on the target are concentric circles. The initial center of symmetry of the ovals in the image can be obtained by clustering as (x_i, y_i) , and i represents the number of clustering centers. The subsets in the arc support line segment group G_{arc} are regrouped according to the distance from the center of symmetry (x_i, y_i) . After grouping, f arc support line segment groups $G_{new-arc}$ are obtained. The subsets in each group are grouped into five pairs, and the least squares method is used to fit several initial ellipses. Finally, the initial ellipse S is obtained.

4.1.4. Initial Ellipse Selection

Before the initial ellipses are screened via clustering, the initial ellipses are coarse eliminated according to the surface characteristics of the threaded fasteners studied in this paper. The inner and outer loop of the threaded hole and the inner thread on the surface of the threaded fastener are round. Except when the camera imaging angle is large, it is presented as an ellipse near a circle in the imaging plane. Therefore, the shape constraint is applied to the initial ellipse S first, and the ellipse subset that is too flat or too small is eliminated. The discriminant equation is as follows:

$$F = \begin{cases} 1, & \text{if } \begin{cases} \frac{a_i b_i}{\sqrt{a_i^2 + b_i^2}} > \frac{\sqrt{2}}{\sqrt{2}-1} \\ \sqrt{a_i b_i} > \frac{\sqrt{2}/2}{1-\cos(\theta_i)} \end{cases} \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

When F is 1, the subset is written into the new initial ellipse S^{init} . If F is 0, the subset is directly eliminated. After the initial ellipse is roughly eliminated, the ellipse class aggregation algorithm is used to perform cluster analysis on the initial ellipse set. The steps are as follows:

First, let the initial set of ellipses be S^{init} and the number of subsets be N^{init} . The mean-shift algorithm is used to cluster the elliptic centers of S^{init} , and n^{center} elliptic centers are obtained.

Then, for each ellipse subset in S^{init} , the distance between the center of the ellipse and each cluster center is calculated, and it is classified into the nearest cluster.

Then, each Ω_k is clustered according to the inclination direction of the ellipse, and the clustering center $\left\{ \varphi_1^c, \varphi_2^c, \dots, \varphi_{n_k^c}^c \right\}$ of n_k^Ω inclination angles is obtained.

Finally, each $\Omega_{(k,s)}$ is clustered according to the half-length axis, and $n_{(k,s)}^{axis}$ clustering centers $(a, b)_1^c, (a, b)_2^c, \dots, (a, b)_{n_{(k,s)}^{axis}}^c$ of the half-length axis of the ellipse are obtained.

4.1.5. Verification of Geometric Properties of Candidate Ellipses

For the candidate ellipses obtained in the previous step, it often needs to be determined whether they are true ellipses. Based on the geometric property of ellipses, an evaluation function $G_{evaluate}$ is introduced to verify it:

$$G_{evaluate}(e) = \sqrt{\frac{\#\{p_i : p_i \in SI(e)\}}{H} \cdot \frac{C}{360^\circ}} \quad (5)$$

where $\#\{p_i : p_i \in SI(e)\}$ represents the number of points within the edge of the ellipse, H approximates $\pi \left[\frac{3}{2}(a+b) - \sqrt{ab} \right]$ and C is related to the crossing angle (ellipse significance score) of the ellipse. After all the ellipses are evaluated, the results of $G_{evaluate}(e)$ less than the threshold value are eliminated, and then the ellipse is further fitted to obtain the final elliptic parametric equation.

4.2. Ellipse Detection Results Compared with the Method before Improvement

The camera captures a threaded hole image on a threaded fastener in a certain state in the same operating environment (Hardware: Intel Core i5-8700 CPU@3.20 GHz, 16 GB, manufacturer: Intel, sourced from Shanghai, China; Software environment: windows 10, Visual Studio code). The detection results and running time before and after the improvement of the ellipse detection method based on arc support line segments are shown in Figure 3 and Table 2. Figure 3a,b are schematic diagrams of the number of candidate ellipses and detected ellipses obtained before the improved algorithm. Figure 3c,d are schematic diagrams of the number of candidate ellipses and detected ellipses obtained after the improved algorithm. In the future, numerical optimization can be used to further shorten the detection time.

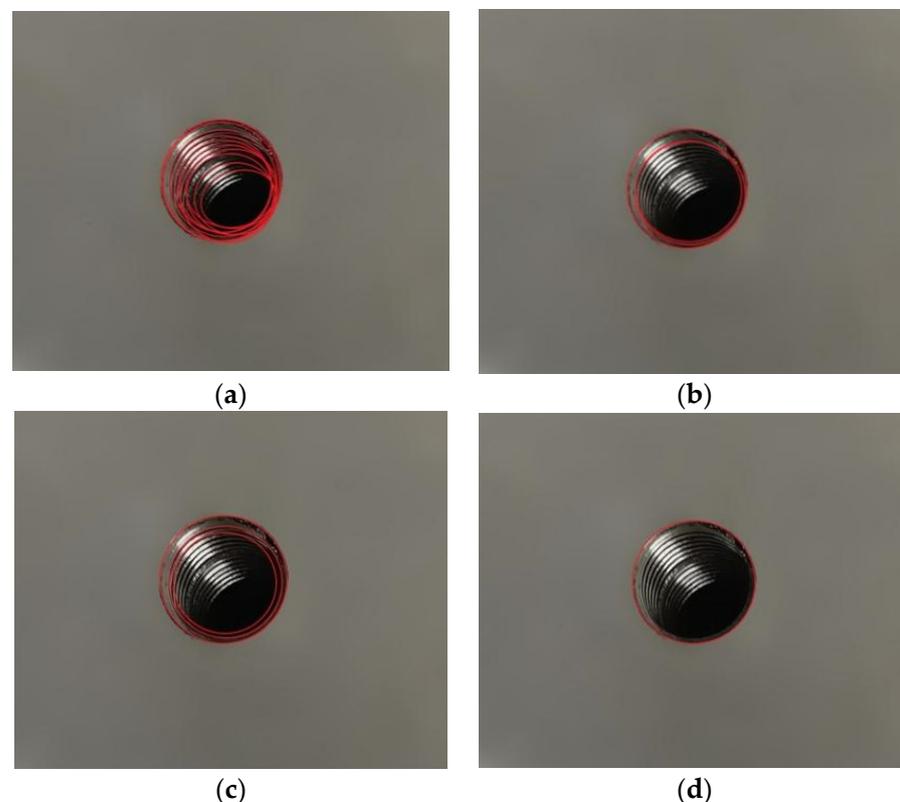


Figure 3. Schematic diagram of the number of candidate ellipses and detected ellipses obtained before and after the improved algorithm: (a) Schematic diagram of the number of candidate ellipses obtained before the improved algorithm. (b) Schematic diagram of the number of detected ellipses obtained before the improved algorithm. (c) Schematic diagram of the number of candidate ellipses

obtained after the improved algorithm. (d) Schematic diagram of the number of detected ellipses obtained after the improved algorithm.

Table 2. Statistics of ellipse detection results on threaded holes before and after improvement.

Algorithm	Number of Arc Segments	Initial Number of Ellipses	Number of Candidate Ellipses	Number of Detection Ellipses	Detection Time
Before improvement	100	13	8	3	0.232
After improvement	100	4	2	1	0.044

5. Estimation of Threaded Hole Orientation

By extracting the ellipse information of the thread hole, the relative position of the center of the thread hole is calculated. Then, the elliptic information of the threaded hole is used to solve its orientation. To ensure that the direction of the robot end fixture is perpendicular to the plane where the threaded hole is located during the tightening process of the bolt and threaded hole, this article proposes a threaded hole orientation estimation algorithm based on three-point sampling, which corrects the orientation of the robot end fixture by solving the world coordinates of the three points on the ellipse. The principle of thread hole orientation estimation and robot end orientation correction is as follows: First, the robot is moved above the threaded hole so that the coordinate system Z_t of the robot end fixture is parallel to the robot base coordinate system Z_W . The transformation matrix between the coordinate system Z_C of the follow-up camera and the coordinate system Z_t of the robot end fixture can be obtained through hand-eye calibration. Figure 4 shows the relationship between coordinate systems in the threaded hole orientation estimation process. The follower camera identifies the boundary ellipse of the threaded hole above the threaded hole, and the three pixel points on the ellipse are randomly sampled. The world coordinates A, B, C of the three sampling points are obtained by using a depth camera and the hand-eye relationship. First, the orientation correction angle of the robot end fixture is solved, and the plane equation and plane normal vector of the threaded hole can be calculated by using the world coordinates of A, B, C . Then, the normal vector is used as the Z -axis of the space rectangular coordinate system, the threaded hole coordinate system Z_b is established, and the orientation correction angle of the coordinate system Z_b to the base coordinate system Z_W is calculated. Finally, the coordinate system of the end fixture is adjusted according to the orientation correction angle to obtain the corrected robot end fixture orientation.

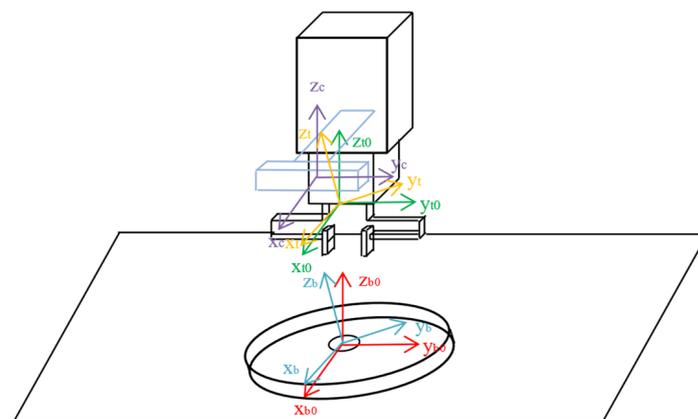


Figure 4. Relationship between coordinate systems in the process of estimating the orientation of threaded holes.

The calculation principle of threaded hole orientation estimation is as follows:
The plane equation determined by the coordinates of A, B, C on the threaded hole is

$$\begin{vmatrix} x & y & z & 1 \\ x_1 & y_1 & z_1 & 1 \\ x_2 & y_2 & z_2 & 1 \\ x_3 & y_3 & z_3 & 1 \end{vmatrix} = 0 \quad (6)$$

The equation of the threaded hole plane can be written as

$$ax + by + cz + d = 0 \quad (7)$$

where (a, b, c) is the normal vector of the plane where the threaded hole is located. By introducing the coordinates (x_1, y_1, z_1) , (x_2, y_2, z_2) , (x_3, y_3, z_3) of the three points A, B, C into Formula (7), we can obtain

$$\begin{cases} a = y_1z_2 - y_2z_3 - y_2z_1 + y_3z_1 + y_2z_3 - y_3z_2 \\ b = -x_1z_2 + x_1z_3 + x_2z_1 - x_3z_1 - x_2z_3 - x_3z_2 \\ c = x_1y_2 - x_1y_3 - x_2y_1 + x_3y_1 + x_2y_3 - x_3y_2 \\ d = -x_1y_2z_3 + x_1y_3z_2 + x_2y_1z_3 - x_3y_1z_2 - x_2y_3z_1 + x_3y_2z_1 \end{cases} \quad (8)$$

Therefore, the normal vector of the plane unit of the threaded hole is calculated:

$$e = (a, b, c) / \sqrt{a^2 + b^2 + c^2} \quad (9)$$

The orientation of the robot end is corrected by using the result of the orientation estimation of the threaded hole. During the orientation correction, the rotation matrix of the robot end fixture coordinate system to the base coordinate system is

$$\begin{bmatrix} \cos \beta \cos \gamma & \cos \gamma \sin \alpha \sin \beta - \cos \alpha \sin \gamma & \sin \alpha \sin \gamma + \cos \alpha \cos \gamma \sin \beta \\ \cos \beta \sin \gamma & \cos \alpha \cos \gamma + \sin \alpha \sin \beta \sin \gamma & \cos \alpha \sin \beta \sin \gamma - \cos \gamma \sin \alpha \\ -\sin \beta & \cos \beta \sin \alpha & \cos \alpha \cos \beta \end{bmatrix} \quad (10)$$

where α is the rotation angle about the X -axis, β is the rotation angle about the Y -axis and γ is the rotation angle about the Z -axis. The normal vector of the initial orientation of the robot end fixture is $m_0 = (0, 0, 1)$, and the orientation correction angle of the robot end fixture can be obtained from (9) and (10), as follows:

$$\begin{cases} a = \lambda(\sin \alpha \sin \gamma + \cos \alpha \cos \gamma \sin \beta) \\ b = \lambda(\cos \alpha \sin \beta \sin \gamma - \cos \gamma \sin \alpha) \\ c = \lambda \cos \alpha \cos \beta \end{cases} \quad (11)$$

where $\lambda = \sqrt{a^2 + b^2 + c^2}$. Because (11) is a set of nonlinear equations, a, b, c can be obtained through the world coordinates A, B, C on the threaded hole plane. The orientation correction angle $\alpha_e, \beta_e, \gamma_e$ of the robot end fixture can be obtained by solving the nonlinear equations using the Newton iteration method.

6. Design of Assembly Force Controller

6.1. Robot Impedance Control Based on Position

The servo motor used in existing industrial robots has a high precision for position control, so it can make full use of this advantage to realize the position and orientation control of the robot end. In this paper, the constrained environment is equivalent to the impedance system, then the end of the manipulator is the admittance system.

As shown in Figure 5, in the position-based impedance control outer loop, the collected force information is filtered and compensated first, and then the actual contact force F_e of the robot end fixture is decoupled. In the actual simulation process, the end contact force F_e

is obtained by multiplying the difference between the current end position of the robot X and the environmental position X_e by the environmental stiffness K_e . The robot controller can be regarded as a discrete microcomputer system in the time domain, and the relation between the position and orientation correction ΔX and the error e of the impedance control in a discrete form can be written as

$$\Delta x = \frac{e}{m_d s^2 + b_d s + k_d} \quad (12)$$

where m_d is the mass coefficient, b_d is the damping coefficient and k_d is the elastic coefficient.

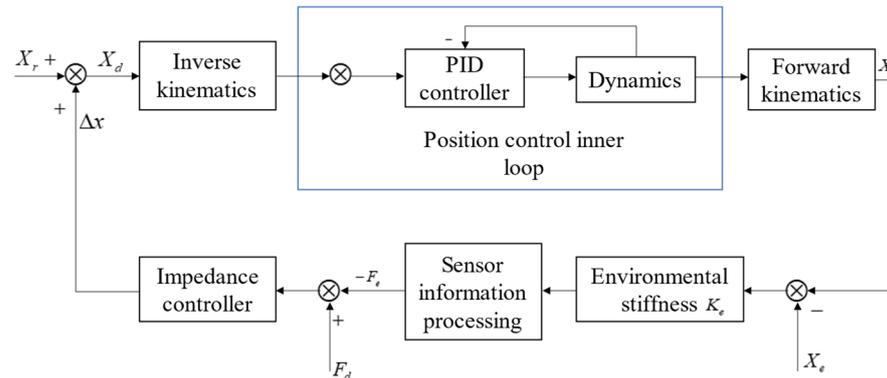


Figure 5. Impedance control block diagram based on position.

The force deviation e is calculated according to the set expected force F_d , and the impedance control Formula (12) is used to calculate the force deviation e and output the end-pose correction Δx to the robot. The reference pose X_r at the end of the robot and the correction amount are superimposed to generate the desired pose X_d of the robot, which is sent to the position controller. The main function of the position controller is kinematic transformation and a dynamic response. The advantage of position-based impedance control is that the design of the force controller does not need to consider robot dynamics. However, this control method is not available when the constrained environment is characterized by admittance characteristics or when the accuracy of the robot position controller is poor.

One of the biggest advantages of impedance control is that it can achieve the flexibility of the manipulator in both free space and constrained space. When the contact force f_e of the robot's end fixture is not 0, the impedance control of a single degree of freedom is

$$m_d(\ddot{x} - \ddot{x}_r) + b_d(\dot{x} - \dot{x}_r) + k_d(x - x_r) = -f_e \quad (13)$$

At this time, the robot is in the constrained space. According to whether the elastic coefficient k_d is zero, the robot displays two kinds of behaviors externally. When $k_d = 0$, the robot presents zero gravity externally, and the robot drag function is based on this behavior. When $k_d \neq 0$, the robot can also show compliance in the direction of impedance control, but the difference is that, when the external force is removed, the robot can automatically return to the original position. The amount of force required for the robot to deviate from the desired position is related to k_d , and as k_d becomes larger, the amount of force required becomes greater.

6.2. Design of Robot Compliance Force Controller

In the process of bolt assembly, the robot must not only adjust the lateral error of the bolt and threaded hole, but it must also control the angle error of the bolt and threaded hole. Therefore, the force controller should have the function of constant force tracking and orientation adjustment, so as to make the relative position and orientation of the bolt and threaded hole in the best state during the assembly process. Although impedance control

is often applied to robot force control, conventional impedance control cannot solve the problem of the force impact generated by the robot during the transition from free motion space to constrained space, as well as the tracking accuracy problem of the force controller during the tightening process. In view of the above problems, a compliant force controller composed of a constant force controller and an orientation controller is designed in this section, which is characterized by reducing the force impact at the moment of the bolt's contact with the threaded hole, improving force tracking accuracy in the assembly process, and realizing orientation adjustment of the bolt in the assembly process.

6.2.1. Constant Force Controller

Constant force tracking refers to the ability of the force controller to track the set expected force, which is the key to robot compliance. In this section, the design of the force tracking controller is mainly carried out to realize the constant force control problem in the X, Y and Z directions of the axis in the process of axle hole assembly. Impedance control based on an environment model and force error feedback are two kinds of control strategies for robot compliance. Based on the environmental model, the reference position x_r is calculated in real time according to the contact force and the actual position of the robot to achieve rapid convergence of the contact force. Based on force error feedback, the error feedback method (PID, etc.) is applied to impedance control. After feedback control, position correction is generated by the impedance function.

Industrial robots based on position control can only accept position instructions, but to realize force tracking, a control model that can accept force instructions and output position instructions is needed. Then, the single degree of freedom impedance control becomes

$$m_d \ddot{x} + b_d \dot{x} + k_d \Delta x = u \quad (14)$$

where $\Delta x = x - x_r$ is the position difference of the end of the manipulator and u is the input force.

It is assumed that the impedance system only has inertia. Under the action of the controlling force u , the system behaves as follows:

$$m_d \ddot{x} = u \quad (15)$$

Integrate against the above formula:

$$\Delta x = \iint \ddot{x} dt = \iint \frac{u}{m_d} dt \quad (16)$$

It is assumed that inertia, damping and elasticity exist in the impedance system at the same time. Under the action of the controlling force u , the system behaves as follows:

$$\ddot{x} = \frac{u - b_d \dot{x} - k_d x}{m_d} \quad (17)$$

Integrate against the above formula:

$$\Delta x = \iint \ddot{x} dt = \iint \frac{u - b_d \dot{x} - k_d x}{m_d} \quad (18)$$

By comparison with Equations (16) and (18), it can be seen that the introduced damping and elasticity can buffer and absorb energy in the system, making the motion of the robot more stable. PID control is the simplest method to achieve precise control of system input, using PID to generate control variables:

$$u = K_p e_f + k_i \int e_f + k_d \dot{e}_f \quad (19)$$

where $e_f = f_d - f_e$ is the deviation between the expected force and the actual force. The PID control method realizes the differential part of PID through the difference of error. However, the differential term has an amplification effect on noise, so the differential coefficient k_d parameter in the controller cannot be too large, which limits the performance of the PID and affects its application range. Therefore, the force controller needs to be redesigned.

In the assembly process, the robot moves at a constant speed, so the influence of the robot's end speed and acceleration on the contact force between the bolt and the threaded hole can be ignored. Moreover, the threaded fasteners remain static during robot operation. Therefore, the contact between the shaft and the hole workpiece environment can be defined as a linear spring.

Then, the error between the expected force and the contact force is

$$\begin{aligned} f_e &= k_e(x - x_e) \\ e &= f_d - f_e = f_d - k_e(x - x_e) \end{aligned} \quad (20)$$

The dynamic response differential equation of contact force deviation can be obtained by coupling (14), (20) and (21):

$$m_d \ddot{e} + b_d \dot{e} + (k_d + k_e)e = k_d(f_e + k_e x_e) - k_e k_d x_r \quad (21)$$

When the contact force reaches a steady state, the force tracking error is

$$e_{ss} = \frac{k_d}{k_d + k_e} [(f_d + k_e x_e) - k_e x_r] \quad (22)$$

Then, the contact force is

$$f_e = f_d - e_{ss} = \frac{k_e}{k_d + k_e} [f_d - k_e x_e + k_e x_r] \quad (23)$$

It can be seen from the above equation that the static force error is related to the reference force f_d and the reference position x_r . $e_{ss} = 0$ when the impedance parameter $k_d = 0$ or terminal reference position $x_r = x_e + f_d/k_e$. In general, k_d is not 0, so the reference position needs to be corrected in real time according to the force and current position of the robot end.

6.2.2. Orientation Compliance Controller

During the assembly process, the constant force controller is used to control the forces in the X, Y and Z directions of the bolt, and the orientation compliance controller adjusts the orientation of the bolt in real time according to the torque information to reduce the angle error between the bolt and threaded hole. Figure 6 only shows the schematic diagram of the bolt force in the threaded hole during robot assembly in the Y and Z plane. Figure 6a shows that the bolt is in a two-point contact state and subject to sliding friction f_1 and f_2 . In order to keep the bolt parallel to the threaded hole, the bolt needs to be adjusted clockwise around the Y-axis at the angle $-\theta_y$ in Figure 6a to reach the orientation shown in Figure 6b. Therefore, the bolt orientation can be adjusted according to the magnitude and direction of the torque received in the Y direction of the bolt.

The orientation angle is obtained by rewriting (14) into a frequency domain expression:

$$\pm\theta_y = \frac{\pm My}{Ms^2 + Bs + K} \quad (24)$$

Then, the orientation adjustment angle $\pm\theta_x$ of the bolt in the X direction is

$$\pm\theta_x = \frac{\pm Mx}{Ms^2 + Bs + K} \quad (25)$$

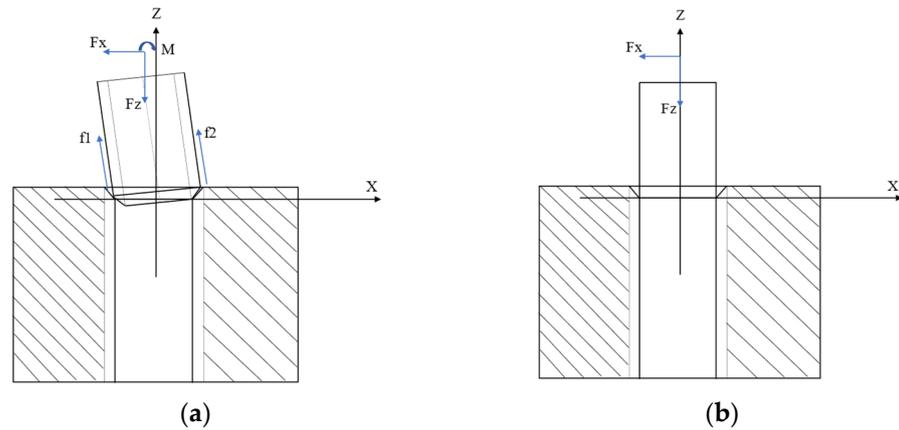


Figure 6. Force/torque diagram of bolt assembly under different orientations: (a) Bolt is in a two-point contact state. (b) Bolt is in a flat contact state.

6.3. General Control Scheme of Assembling Force Controller

As shown in Figure 7, the constant force controller and orientation controller are combined into a compliant force controller by adopting the force/position hybrid control method. The corresponding controller is selected according to the position error or angle error in each stage of assembly.

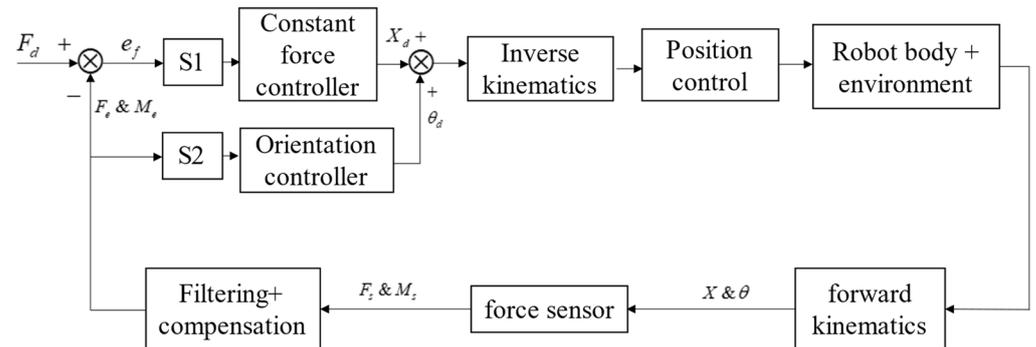


Figure 7. Compliant assembly force control block diagram.

The angle error of the bolt and force sensor during installation is ignored. The six-dimensional symmetric matrix S1 and S2 are designed, as shown in (27). All the elements of a symmetric matrix are 0 except the ones on the diagonal, which are 0 or 1. S1 indicates that the constant force controller is selected, and S2 indicates that the orientation controller is selected. If the bolt is detected to be subjected to a transverse binding force by the hole wall during assembly, S1 is selected, and the constant force controller adjusts the bolt to move in the direction where the axial contact force decreases based on the force in the X, Y and Z directions of the bolt. When extreme situations such as jamming and wedging occur in the assembly process, i.e., when a large torque is detected on the shaft, S2 is selected. The angle error between the bolt and the threaded hole can be reduced by adjusting the orientation of the bolt through the orientation controller.

$$S1 = \begin{bmatrix} 1 & & & & & \\ & 1 & & & & \\ & & 1 & & & \\ & & & 0 & & \\ & & & & 0 & \\ & & & & & 0 \end{bmatrix} \quad S2 = \begin{bmatrix} 0 & & & & & \\ & 0 & & & & \\ & & 0 & & & \\ & & & 1 & & \\ & & & & 1 & \\ & & & & & 1 \end{bmatrix} \quad (26)$$

As shown in Figure 7, the binding force/torque acquired by the sensor in real time is subtracted from the expected reference force after filtering and tool gravity/torque compensation, and the force error is input into the compliance controller to obtain the position correction and orientation correction. Then, the position and orientation correction is superimposed with the position and orientation of the robot in the previous cycle to generate the position and orientation that the robot needs to reach. Because the pose of the robot end fixture is expressed in a Cartesian space, an inverse kinematics transformation is required to obtain the optimal joint space angle and to send it to the position controller. It should be noted that the position and orientation correction generated by the compliance force controller is expressed in the tool coordinate system. The robot uses the base coordinate system to plan the trajectory to the pose point, so the position and orientation correction needs to be converted to the base coordinate system for representation.

$$\begin{bmatrix} {}^b\Delta x \\ {}^b\Delta y \\ {}^b\Delta z \end{bmatrix} = {}^bR_f^f R_t^f \begin{bmatrix} {}^t\Delta x \\ {}^t\Delta y \\ {}^t\Delta z \end{bmatrix} \quad (27)$$

where ${}^t\Delta x, {}^t\Delta y, {}^t\Delta z$ is the correction of the compliance force control in the tool coordinate system, ${}^b\Delta x, {}^b\Delta y, {}^b\Delta z$ is the position and orientation correction in the base coordinate system, which is sent to the servo motor of each axis after the inverse kinematics solution, ${}^fR_t^f$ is the transformation matrix from the robot end fixture coordinate system to the end-effector and ${}^bR_f^f$ is the transformation matrix from the end-effector to the robot base coordinate system.

7. Experiments and Results

Two series of laboratory experiments were carried out to evaluate the precision and repeatability of the proposed fastening processes in view of the industrial application requirements. According to the system configuration given in Section 3, the experimental system is established. In the experimental system, the Realsense D457 depth camera is located at the end of the robot. It has a depth of field of 58×87 and can shoot 90 frames per second with an image size of 1920×1080 pixels. The Marposs H50 is used as a displacement sensor with a measurement accuracy of 0.15 μm and a stroke of 25 mm. The six-dimensional force sensor is ATI Nano-43 with a measuring range of ± 18 N and a resolution of 1 mN after filtering along the X, Y and Z axes. The real-time data collected by the six-dimensional force sensor are processed with a Kalman filter. The robot is aubo-i10 with a translational resolution of 3 μm along the X, Y and Z axes.

The threaded fasteners to be assembled are shown in Figure 1. The material is carbon steel. The clearance between the bolt and the threaded hole is 0–10 μm . The bolt of M24 is used in the experiment. The lower surface of the bolt has a chamfer, its height is 60 μm , and it is inclined about 45° from the axial direction. The compression stiffness of the coil spring is 3.3 mN/ μm . We also tested the mechanical properties of the coil spring. The test results show that, when the compression deformation is less than 5 mm, the coil spring returns to its original shape. Moreover, the coil spring was deformed 100 times according to the above deformation scale, and no obvious fatigue effect was found. The end tightening tool is connected by four coil springs, and the displacement sensor is used to collect the coil spring deformation information and map it to the robot end coordinate.

7.1. Visual System Validation Experiment

Because the performance of the proposed method is largely dependent on the initial pose information obtained from vision, we conducted experiments to verify the accuracy and real-time performance of the vision system. To examine the geometrical precision, i.e., translational and rotational accuracies, four batches of twenty assembly tests were carried out on the threaded fasteners, and the time required for position and orientation estimation by the visual system was recorded for each experiment. For each batch, the

fasteners were tilted at four different angles from 5° to 30° and rotated randomly about their central axes. A series of metrology equipment was then used on the rig to measure the position of the threaded holes on the plate relative to the system's reference point within a $\pm 10\text{-}\mu\text{m}$ accuracy range. The information obtained was used as a benchmark to evaluate the accuracy of the coordination points calculated by the developed algorithms. The experiments were considered successful if the geometrical errors were within the defined application requirements, i.e., up to 1 mm for bolt sizes, and less than 1° rotational errors. The experimental results are shown in Table 3. It can be seen from the table that the average position errors of the system are 0.36 mm, 0.43 mm and 0.46 mm, and the average orientation errors are 0.65°, 0.46° and 0.59°, both of which meet the requirements.

Table 3. Experimental results of position and orientation estimation of visual system.

Tilt Angle	Trials	Mean Position Error (Degree)	Mean Position Error (mm)
5	20	(0.63, 0.44, 0.57)	(0.43, 0.35, 0.33)
10	20	(0.69, 0.50, 0.63)	(0.33, 0.48, 0.44)
20	20	(0.73, 0.55, 0.55)	(0.42, 0.36, 0.49)
30	20	(0.55, 0.35, 0.61)	(0.26, 0.53, 0.58)

7.2. Threaded Hole Assembly Experiment

The assembly process of threaded fasteners is shown in Figure 8. The robot uses the position and orientation estimation algorithm of the threaded hole introduced above for preliminary positioning. Then, according to the force and torque information in the assembly process, the flexibility force controller is used to adjust the position and orientation of the end of the robot to complete the threaded hole assembly task.

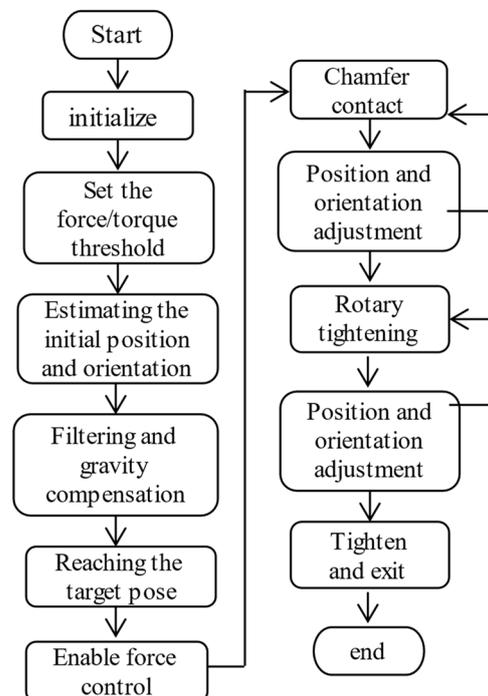


Figure 8. Threaded fastener assembly process.

After the initialization of the system and the robot, the initial position and orientation of the threaded hole can be obtained by the visual system, and the robot motion can be planned to make the bolt in the robot fixture move to the threaded hole. After reaching the specified point, the orientation compliance controller is used to adjust the contact state between the bolt chamfer and the threaded hole, and then the bolt begins to enter the

threaded hole. The relative position and orientation of the bolt is adjusted according to the size and direction of the contact force between the bolt and the threaded hole to avoid the phenomenon of sticking. Then, the tightening movement is carried out. The compliant force controller is used to correct the orientation in the tightening process. By setting the maximum torque threshold in the Y and Z axes, the maximum force threshold in the Z axis and the depth L of the bolt into the threaded hole, whether the bolt and the threaded hole are properly assembled is considered.

Figure 9 shows the assembly process of each stage of the threaded hole. Figure 9a shows the compliance movement of the bolt in the chamfer area, Figure 9b shows the bolt entering the chamfered hole, Figure 9c shows the bolt tightening and adjusting, and Figure 9d shows the completion of assembly and exiting the hole.

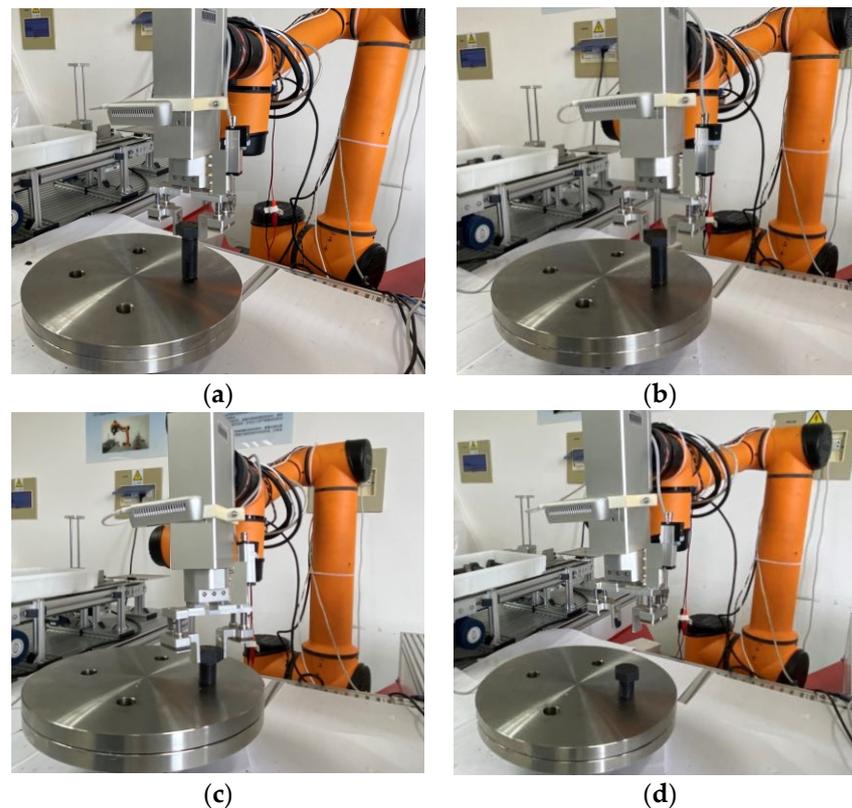


Figure 9. Each stage of the threaded hole assembly process: (a) Compliance movement of the bolt in the chamfer area. (b) Bolt entering the chamfered hole. (c) Bolt tightening and adjusting. (d) Completion of assembly and exiting the hole.

Figure 10 shows the changes in the force/torque, position and orientation of the bolts in each stage of threaded hole assembly. Figure 10a shows the force exerted on the threaded hole of the bolt in the assembly process, Figure 10b shows the torque exerted on the bolt, Figure 10c shows the change in the position of the bolt, and Figure 10d shows the change in the orientation of the bolt. In stage one, the bolt is close to the threaded hole. In this stage, the robot relies on high-precision position control to make the bolt move quickly to the anchor point of the threaded hole. In this stage, the force/torque on the bolt is 0. Stage two is the initial contact state between the bolt and the threaded hole. In this stage, the chamfer of the bolt begins to enter the threaded hole, and the robot performs compliance control along the contact direction. At this time, the Z and Y direction of the bolt is subject to a compliant motion force of about 10 N and 7 N and a small force in the X direction, respectively, and the X direction is subject to a large torque. At this time, the position and pose of the bolt are adjusted. In stage three, after adjusting the bolt position and orientation with the compliance force controller, the force on the bolt in the Z direction gradually

decreases to about 5 N. The force in the X and Y direction is basically 0, so the finding of the threaded hole can be considered successful. At this time, the robot stops the action and waits for the command of the bolt to continue moving down. Stage 4 is the tightening contact stage. When the bolt slides correctly into the threaded hole chamfer along the chamfer, the bolt begins to rotate and move down with the robot end. At the beginning of this stage, the bolt is adjusted in the threaded hole, and the X and Y direction of the bolt shows a trend of increasing first and then decreasing gradually under the force. This is also consistent with the phenomenon in which the contact force decreases along the X and Y direction during the assembly of threaded fasteners. After many adjustments, it can be seen that the contact force between the bolt and the threaded hole tends to a smaller value in the X and Y direction and a fixed value in the Z direction. Stage five is the successful assembly and exit stage. When it is detected that the bolt's movement in the Z direction reaches the set assembly distance, the robot stops moving.

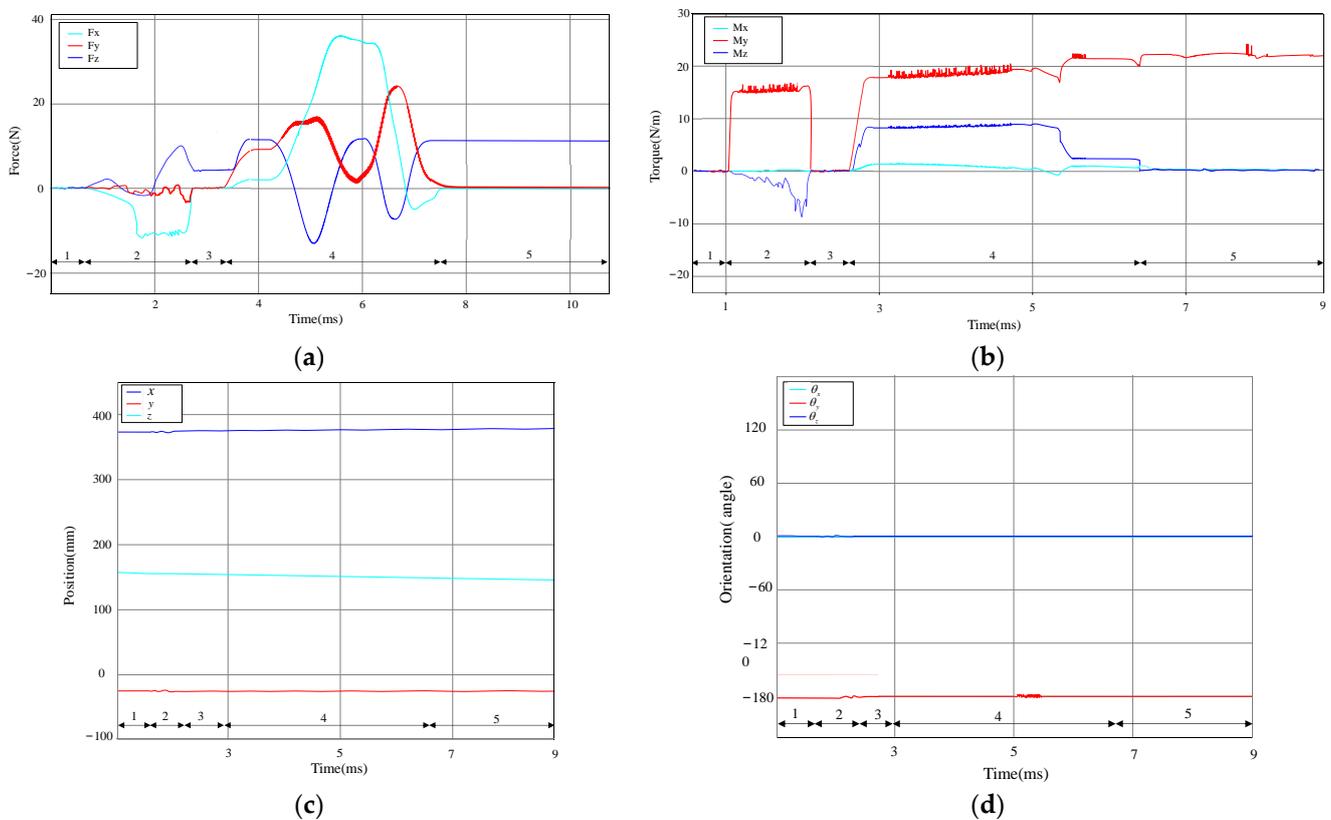


Figure 10. Changes in force/torque, position and orientation of bolts in each stage of threaded hole assembly. (a) Force exerted on the threaded hole of the bolt in the assembly process. (b) Torque exerted on the bolt. (c) Change in the position of the bolt. (d) Change in the orientation of the bolt.

In addition, during the experimental process, the coil spring operates within its elastic range and can be reused. Even the mechanical properties of the spring change due to fatigue effects, so replacing it with a new one is easy and has quite a low cost compared with the expensive RCC (Remote Center Compliance). This could be viewed as an advantage of choosing a coil spring as the compliant device.

This experiment was followed by a series of experiments on the actual fastening process of the bolt to evaluate the accuracy and success rate of the proposed fastening process. Four batches of fifty tests were carried out on the fastening of M24 bolts to the corresponding threaded holes on a randomly located plate. The plate was randomly rotated for each batch of the tests, and the force, torque, position and orientation of the fastening process were observed by the controller. This was to identify potential forceful assembly

or cross threading due to position and orientation errors. In this series of experiments, a test was considered successful if (a) the initial position and orientation of the threaded hole was estimated within the specified range, (b) the fastening process was completed without creating excessive force and torque and (c) the assembly process was completed within an acceptable time period, in accordance with the time available in an industrial application scenario. In total, of over 200 tests, 98.5% of the tests were considered successful. The experimental results are shown in Table 4, and compared with [26], the accuracy and success rates were improved.

Table 4. Experimental results of bolt tightening.

Tilt Angle	Trials	(a)	(b)	(c)	Success Rate
5	50	50	49	49	99%
10	50	50	49	49	98%
20	50	50	49	49	98%
30	50	50	50	50	100%

8. Conclusions

The assembly of threaded fasteners is an important part of the manufacturing process. In small-lot, high-value manufacturing, fasteners are usually assembled manually to allow flexibility and reduce the need for precision fixtures. A review of existing automated assembly techniques shows that current research methods are not fast and accurate enough for the assembly industry with threaded holes with chamfers. In this paper, the authors introduce a new threaded fastener assembly method in a semi-structured environment to solve the above problems.

A novel algorithm for detecting the position and orientation of threaded holes and a new method for tightening bolts are introduced in this paper. A demonstrator rig is presented, comprising a robot, a fastening tool, a set of threaded fasteners and a depth camera. The bolts and threaded holes are randomly located in a 3D space within a confined area. The novelty of this paper lies in the following aspects: First, the elliptic arc fitting method and the three-point method are used to estimate the initial position and orientation of the threaded hole, and the force impact caused by switching from the free space to the constrained space during bolt tightening is solved. Second, by monitoring the deformation of passive compliance, position information is introduced into the control process to better control the radial force on the bolt during insertion, which improves the robustness of the proposed method against alignment errors. The constant force controller and orientation compliance controller are designed according to the adaptive control theory of the assembly scheme, and experiments are carried out. The results of more than 160 tests show that the proposed method can estimate the initial position and orientation of M24 bolts with average position errors of 0.36 mm, 0.43 mm and 0.46 mm and orientation errors of 0.65°, 0.46° and 0.59°, and it can tighten the bolts with 98.5% accuracy. This is more accurate than the existing assembly method [26] (reported errors of 0.53 mm and 0.67° in the literature, with an accuracy of 98%).

The contribution of this paper is that it explores a combined visual and force method to monitor the bolt position and orientation during assembly and to monitor the measurable compliance of the spring to better control the radial forces between objects during insertion, which improves the robustness of the proposed method to alignment errors. Apart from the coil spring, we may also choose other types of compliant devices with mechanical properties similar to those of the coil spring, such as a rubber rod. In addition, the research of this paper is also applicable to the assembly process of shafts and holes.

Author Contributions: Conceptualization, L.Z. and Y.Z.; methodology, Y.Z. and X.W.; software Y.Z.; validation, L.Z. and X.W.; data curation, Y.Z. and X.W.; writing—original draft preparation Y.Z.; writing—review and editing, L.Z. and X.W.; funding acquisition, L.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China (Grant No. 51975213).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Michalos, G.; Makris, S.; Papakostas, N.; Mourtzis, D.; Chryssolouris, G. Automotive assembly technologies review: Challenges and outlook for a flexible and adaptive approach. *CIRP J. Manuf. Sci. Technol.* **2010**, *2*, 81–91. [CrossRef]
2. Klingajay, M.; Seneviratne, L.D.; Althoefer, K. Identification of threaded fastening parameters using the Newton Raphson Method. In Proceedings of the Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453), Las Vegas, NV, USA, 27–31 October 2003.
3. Holzer, M. Aircraft Fasteners. 2011. Available online: <http://www.aviationpros.com/article/10370291/aircraftfasteners> (accessed on 25 October 2016).
4. Huang, S.; Schimmels, J.M. Admittance selection conditions for frictionless force-guided assembly of polyhedral parts in two single-point principal contacts. *IEEE Trans. Robot.* **2008**, *24*, 461–468. [CrossRef]
5. Finetto, C.; Rosati, G.; Faccio, M.; Rossi, A. Implementation framework for a fully flexible assembly system (F-FAS). *Assem. Autom.* **2015**, *35*, 114–121. [CrossRef]
6. Rosati, G.; Faccio, M.; Barbazza, L.; Rossi, A. Hybrid flexible assembly systems (H-FAS): Bridging the gap between traditional and fully flexible assembly systems. *Int. J. Adv. Manuf. Technol.* **2015**, *81*, 1289–1301. [CrossRef]
7. Herakovic, N. *Robot Vision in Industrial Assembly and Quality Control Processes*; INTECH Open Access Publisher: London, UK, 2010.
8. Fei, N.; Zhuang, F.; Renqiang, L.; Qixin, C.; Yanzheng, Z. An image processing approach for jigsaw puzzle assembly. *Assem. Autom.* **2007**, *27*, 25–30. [CrossRef]
9. Pitipong, S.; Pornjit, P.; Watcharin, P. (2010) An automated fourDOF robot screw fastening using visual servo. In Proceedings of the 2010 IEEE/SICE International Symposium on System Integration, Sendai, Japan, 21–22 December 2010.
10. Bone, G.M.; Capson, D. Vision-guided fixtureless assembly of automotive components. *Robot. Comput. Integr. Manuf.* **2003**, *19*, 79–87. [CrossRef]
11. Fleischer, J.; Lanza, G.; Otter, M.; Elser, J. Spatial alignment of joining partners without fixtures, based on component-inherent markings. *J. Manuf. Syst.* **2013**, *32*, 489–497. [CrossRef]
12. Xing, D.; Liu, F.; Qin, F.; Xu, D. Coordinated insertion control for inclined precision assembly. *IEEE Trans. Ind. Electron.* **2016**, *63*, 2990–2999. [CrossRef]
13. Xu, Q. Precision position/force interaction control of a piezoelectric multimorph microgripper for microassembly. *IEEE Trans. Autom. Sci. Eng.* **2013**, *10*, 503–514.
14. Chen, H.; Zhang, G.; Zhang, H.; Fuhlbrigge, T.A. Integrated robotic system for high precision assembly in a semi-structured environment. *Assem. Autom.* **2007**, *27*, 247–252. [CrossRef]
15. Chen, H.; Wang, J.; Zhang, G.; Fuhlbrigge, T.; Kock, S. Highprecision assembly automation based on robot compliance. *Int. J. Adv. Manuf. Technol.* **2009**, *45*, 999–1006. [CrossRef]
16. Peña-Cabrera, M.; Lopez-Juarez, I.; Rios-Cabrera, R.; Corona-Castuera, J. Machine vision approach for robotic assembly. *Assem. Autom.* **2005**, *25*, 204–216. [CrossRef]
17. Ilangovan, B.; Monfared, R.P.; Jackson, M. An automated solution for fixtureless sheet metal forming. *Int. J. Adv. Manuf. Technol.* **2016**, *82*, 315–326. [CrossRef]
18. Kalt, E.; Monfared, R.; Jackson, M. Towards an automated polishing system—Capturing manual polishing operations. *Int. J. Res. Eng. Technol.* **2016**, *5*, 182–192.
19. Mezouar, Y.; Prats, M.; Martinet, P. External hybrid vision/force control. In Proceedings of the International Conference on Advanced Robotics, Jeju, Korea, 28 June–1 July 2007. Available online: https://www.researchgate.net/publication/266497261_External_Hybrid_VisionForce_Control (accessed on 25 October 2022).
20. Carelli, R.; Oliva, E.; Soria, C.; Nasisi, O. Combined force and visual control of an industrial robot. *Robotica* **2004**, *22*, 163–171. [CrossRef]
21. Tao, X.; Janabi-Sharifi, F.; Cho, H. An active zooming strategy for variable field of view and depth of field in vision-based microassembly. *IEEE Trans. Autom. Sci. Eng.* **2009**, *6*, 504–513.
22. Ren, L.; Wang, L.; Mills, J.K.; Sun, D. Vision-based 2-D automatic micrograsping using coarse-to-fine grasping strategy. *IEEE Trans. Ind. Electron.* **2008**, *55*, 3324–3331. [CrossRef]
23. Chu, H.K.; Mills, J.K.; Cleghorn, W.L. Fabrication of a microcoil through parallel microassembly. In Proceedings of the 2012 IEEE International Conference on Robotics and Automation, Saint Paul, MN, USA, 14–18 May 2012; pp. 5050–5055.

24. Tamadazte, B.; Arnould, T.; Dembele, S.; Le Fort-Piat, N.; Marchand, E. Real-time vision-based microassembly of 3D MEMS. In Proceedings of the 2009 IEEE/ASME International Conference on Advanced Intelligent Mechatronics, Singapore, 14–17 July 2009; pp. 88–93.
25. Corke, P.I.; Hutchinson, S.A. A new hybrid image-based visual servo control scheme. In Proceedings of the 39th IEEE Conference on Decision and Control, Sydney, NSW, Australia, 12–15 December 2000; Volume 3, pp. 2521–2526.
26. Dharmara, K.; Monfared, R.P.; Ogun, P.S.; Jackson, M.R. Robotic assembly of threaded fasteners in a non-structured environment. *Int. J. Adv. Manuf. Technol.* **2018**, *98*, 2093–2107. [[CrossRef](#)]
27. Rakotondrabe, M.; Ivan, I.A. Development and force/position control of a new hybrid thermo-piezoelectric microgripper dedicated to micromanipulation tasks. *IEEE Trans. Autom. Sci. Eng.* **2011**, *8*, 824–834. [[CrossRef](#)]
28. Rabenoroosa, K.; Cleve, C.; Lutz, P. Hybrid force/position control applied to automated guiding tasks at the microscale. In Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, Taipei, Taiwan, 18–22 October 2010; pp. 4366–4371.
29. Ureche, A.L.P.; Umezawa, K.; Nakamura, Y.; Billard, A. Task parameterization using continuous constraints extracted from human demonstrations. *IEEE Trans. Robot.* **2015**, *31*, 1458–1471. [[CrossRef](#)]
30. Prasad, D.K.; Leung, M.K.H.; Cho, S.Y. Edge curvature and convexity based ellipse detection method. *Pattern Recognit.* **2012**, *45*, 3204–3221. [[CrossRef](#)]
31. Fornaciari, M.; Prati, A.; Cucchiara, R. A fast and effective ellipse detector for embedded vision applications. *Pattern Recognit.* **2014**, *47*, 3693–3708. [[CrossRef](#)]
32. Lu, C.; Xia, S.; Shao, M.; Fu, Y. Arc-support Line Segments Revisited: An Efficient and High-quality Ellipse Detection. *IEEE Trans. Image Process.* **2019**, *29*, 768–781. [[CrossRef](#)] [[PubMed](#)]
33. Meng, C.; Li, Z.; Bai, X.; Zhou, F. Arcs Adjacency Matrix Based Fast Ellipse Detection. *IEEE Trans. Image Process.* **2020**, *29*, 4406–4420. [[CrossRef](#)] [[PubMed](#)]
34. Von Gioi, R.G.; Jakubowicz, J.; Morel, J.M.; Randall, G. LSD: A line segment detector. *Image Process. Line* **2012**, *2*, 35–55. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.