applied sciences

MDPI

*Article*

# MANTRA: An Effective System Based on Augmented Reality and Infrared Thermography for Industrial Maintenance

**Mario Ortega [1], Eugenio Ivorra [1,*], Alejandro Juan [1], Pablo Venegas [2] and Jorge Martínez [3] and Mariano Alcañiz [1]**

[1] Institute for Research and Innovation in Bioengineering, Polytechnic University of Valencia, 46022 Valencia, Spain; maorpre@i3b.upv.es (M.O.); aljuasin@i3b.upv.es (A.J.); malcaniz@i3b.upv.es (M.A.)
[2] Aeronautical Technologies Centre (CTA), 01510 Miñano, Spain; pablo.venegas@cta.aero
[3] SEGULA Technologies, 1003 Vitoria, Spain; jmartinez@segula.es
* Correspondence: euivmar@upvnet.upv.es or euivmar@i3b.upv.es

**Featured Application: MANTRA system has been validated in laboratory condition with an electronic board maintenance task, but it has been developed to work for almost any industrial maintenance task of electromechanical devices such as electric motors, power control panels or special machinery and custom automation systems.**

**Abstract:** In recent years, the benefits of both Augmented Reality (AR) technology and infrared thermography (IRT) have been demonstrated in the industrial maintenance sector, allowing maintenance operations to be carried out in a safer, faster, and more efficient manner. However, there still exists no solution that optimally combines both technologies. In this work, we propose a new AR system—MANTRA—with specific application to industrial maintenance. The system can automatically align virtual information and temperature on any 3D object, in real time. This is achieved through the joint use of an RGB-D sensor and an IRT camera, leading to high accuracy and robustness. To achieve this objective, a pose estimation method that combines a deep-learning-based object detection method, YOLOV4, together with the template-based LINEMOD pose estimation method, as well as a model-based 6DOF pose tracking technique, was developed. The MANTRA system is validated both quantitatively and qualitatively through a real use-case, demonstrating the effectiveness of the system compared to traditional methods and those using only AR.

**Keywords:** Augmented Reality; thermography; maintenance; tracking; Industry 4.0

check for updates

**Citation:** Ortega, M.; Ivorra, E.; Juan, A.; Venegas, P.; Martínez, J.; Alcañiz, M. MANTRA: An Effective System Based on Augmented Reality and Infrared Thermography for Industrial Maintenance. *Appl. Sci.* **2021**, *11*, 385. https://doi.org/10.3390/app11010385

Received: 14 December 2020
Accepted: 29 December 2020
Published: 3 January 2021

## 1. Introduction

A new industrial revolution, called Industry 4.0, is considered to have recently begun. In this context, industrial maintenance is required to be fast and effective, to optimize production systems. In this regard, the Augmented Reality (AR) technology is key in the Industry 4.0 paradigm, as it allows for maintenance operations to be carried out safely, quickly, and in a very effective manner [1,2]. Likewise, it has also been proved that the inclusion of advanced sensors, such as infrared thermography (IRT) cameras, in the maintenance sector contribute to this optimization [3–5].

First, it has been shown that AR technology can optimally guide maintenance operators in complex tasks. Specifically, maintenance operators can work faster, with fewer mistakes, and with less cognitive load during AR-assisted operations [1,2]. AR has been tested successfully in assembly, repair, and predictive maintenance tasks. Consequently, the introduction of AR systems allows for a direct reduction of operations costs in the industrial maintenance sector, giving a higher percentage of return on investment. This can provide a key difference in this competitive sector for companies in the near future. For example, costs associated with maintenance and repair expenses are estimated to be between 15% and 70% of the total production cost [6]. Moreover, in sectors such as aeronautics, maintenance and reparation tasks can increase to up to 80% of the total product cost during its life cycle [1].

An AR system geometrically aligns virtual and real objects in the real world in real time, to "augment" the available information [2]. This alignment requires the estimation of the scene-relative position and orientation of the camera. This process is called tracking [1,2]. In particular, when the position and orientation of a specific object in the scene are required to be known, it is called object pose estimation. There are multiple ways of solving tracking problems, which can mostly be divided into two types: Using external markers or without them (markerless tracking). On one hand, marker-based tracking techniques employ tags placed manually over the scene to estimate the camera pose. On the other hand, markerless tracking techniques use the natural features of the scene to estimate it.

Industrial maintenance objects tend to have the following attributes: They are complex and textureless 3D objects. Moreover, they usually have partial occlusion, and their appearance can change over time (e.g., due to rusting or grease stains). In addition, the working environment may have adverse conditions, such as not having controlled light conditions or reduced physical space [7]. Most current AR systems are not able to robustly detect and estimate the 3D object pose in real time under such conditions. Consequently, AR has not been widely integrated into the production sector, because the tracking ability to align accurate 3D virtual information over the physical objects in real time is a critical factor [1,2].

Secondly, IRT is a developed technology which has widely expanded in recent years. Specifically, IRT is a technology that measures and analyses temperature information using a non-contact device [8]. IRT obtains this information from electromagnetic radiation called infrared radiation, whose wavelength is longer than the visible light spectrum. Although the infrared radiation theoretically covers a range of wavelengths $\lambda$ from 0.7 µm to 1000 µm, for practical purposes the ranges used by the infrared sensors covers values of $\lambda$ from 1 µm to 15 µm; these ranges being delimited by the atmospheric transmission windows. The physical basis of IRT has been explained in terms of thermal radiation theory [9]. Its specific characteristics of non-contact measurement and harmless radiation make IRT a useful tool in numerous industrial and scientific utilities. Its main recent applications include the inspection and monitoring of electronic, mechanical, or electromechanical components [3–5]; although IRT has also been applied in other sectors, such as for medical trials [10] or thermic isolation checking in civil engineering [11].

Although the numerous benefits of AR and IRT technologies in the industrial maintenance sector have been widely studied, there are few works in the scientific literature that have combined both [12–14] in other sectors. To the best of our knowledge, there does not exist any AR system applied to the industrial maintenance sector using IRT sensors with comparable capabilities to the one presented in this work. Specifically, the developed system can align 3D virtual information over different industrial physical objects accurately, robustly, and in real time, together with their temperature information.

*Proposals*

In this work, a new AR system combined with IRT called MANTRA is proposed, which can be applied to industrial maintenance. In more detail, the main contributions of this work are:

- To our knowledge, it is the first time that an AR system has been used in combination with IRT in the field of industrial maintenance.
- We develop a new AR system using a combination of multiple methods that can align virtual content on 3D objects automatically, precisely, robustly, and in real time. Specifically, the 3D object detection and pose estimation method is based on a modified LINEMOD method [15] with a built-in deep-learning method called YOLOV4 [16]. In addition, a 6 Degree of Freedom (6DOF) pose tracking method, based on the work of Tjaden et al. [17], was also integrated. Finally, it is important to note that the

YOLOV4 detection method was trained only with photorealistic synthetic images generated automatically by BlenderProc [18].

- The MANTRA system allows for obtaining the temperature information of a 3D object precisely and in real time, as well as that of its components, using the information of the previously calculated object pose.
- A new calibration pattern is designed, to spatially calibrate RGB-D and IRT cameras.
- The 3D object detection and pose estimation method used by the MANTRA system is validated on different public data sets.
- Both quantitative and qualitative validation of the MANTRA system, as applied to industrial maintenance in a real use-case, demonstrate its effectiveness, compared to traditional maintenance methods and those using only AR.

## 2. Related Work

Multiple works demonstrating the benefits of AR technology as a tool for maintenance in the industrial sector have been published [1,2,19].

For example, Fiorentino et al. [20] showed that using AR can reduce errors by 92%, while maintenance tasks were performed 79% faster. Within the aeronautical sector, Ceruti et al. [21] estimated a 27% reduction in time when carrying out the assembly and the disassembly of an airplane within a maintenance operation using AR. In [22], the authors showed how AR technology can be also used to train technicians to carry out maintenance tasks. Most of these studies have highlighted the importance of the tracking method chosen as one of the determining factors to guarantee the success of the AR system applied to maintenance [2,7]. Traditionally, the use of fiducial external 2D markers, such as ARToolkit or ArUco, has been the tracking method most used by the scientific community to add virtual elements in AR systems applied to maintenance, assembly, and repair in the industrial sector [23]. This is because it is relatively easy to quickly and accurately identify and estimate the pose of these markers using computer vision techniques [7,24]. Despite its extensive use in the industry, this type of tracking technique has some important limitations [2,7,20,24]: The technician must spend a lot of time manually setting the physical fiducial markers, to correctly align the virtual content of the scene. At the same time, multiple markers must be used and related to each other, such that the AR system is able to estimate the pose from any point of view of an object [24]. Secondly, the technician's ability to work in real situations is reduced, as the markers may interfere with access to objects of interest [7,24]. Moreover, these types of markers are critically affected when the marker is occluded or under non-controlled light conditions [7,24]. The aforementioned technical limitations make it difficult to apply these AR systems to maintenance in the industrial sector and, in particular, for performing on-site maintenance tasks.

As an alternative to marker-based tracking systems, markerless AR systems do not need any kind of additional marker scene to accurately add and align virtual information about any element of the scene [24]. These methods use the natural features of the scene, such as texture (e.g., a color histogram) or geometric (e.g., edges) information to perform tracking. In Table 1 are summarized the main AR methods for AR tracking. For example, some of the most common tracking methods in AR use Simultaneous Localization and Mapping (SLAM) techniques, such as the PTAM [25] method. These methods simultaneously estimate the pose of the camera in real time, with respect to an unknown scene, while performing the 3D reconstruction of it. The main limitation of this kind of method is that they usually produce failures when the scene is dynamic, and it is required that the scene is sufficiently textured to estimate key points. Thus, they are generally not suitable for maintenance tasks, as the environment may be dynamic and industrial environments are usually not sufficiently textured [7].

**Table 1.** Main AR tracking methods.

| | **Markerless** | | |
|---|---|---|---|
| | | Model-based | |
| **Marker-based** [20] | SLAM-based [25] | Tracking by detection<br>• Local appearance features [26]<br>• Local 3D geometric descriptors [27]<br>• Template-based [15]<br>• Deep-learning methods [31–34] | 6DOF pose tracking [17,28–30] |

Another type of tracking method is those known as model-based. These methods estimate the pose of a real 3D object from information obtained from a 3D or CAD model of the object. These methods can, in turn, be divided into two types: On one hand, tracking by detection methods detect and estimate the pose of the 3D object independently in each frame of a video sequence; on the other hand, 6DOF pose tracking methods start from an initial pose of the object and update its pose throughout the successive frames of the video sequence (i.e., frame-to-frame tracking). Specifically, 6DOF pose tracking methods minimize the error between the contour of the observed object and its model using an iterative optimization approach [17,28–30]. Although these methods are usually fast, the main limitation is that they require an initial pose of the object [29]. Moreover, these methods tend to accumulate error over time (drift) and face difficulties when the background is cluttered. Various authors have proposed the incorporation of the color information (e.g., tcl-histogram data), as well as new optimization techniques; among these, the PW3P3D method [30] or the work by Tjaden et al. [17] stand out.

Some of the most common tracking by detection methods used by the scientific community use the local appearance features of an image, such as the SIFT method [26]. As previously mentioned, this tracking technique is not recommended in industrial environments, as the objects are not usually textured. Another way to estimate the pose of a 3D object is by using an RGB-D or depth camera and calculate the local 3D geometric descriptors of the 3D model, such as Point Pair Features (PPFs) [27]. The main problem with these methods is that they tend to be very computationally expensive [15,35], preventing their use in AR systems. Another alternative is using template matching techniques, such as the LINEMOD method [15]. The main limitation of these kind of methods is that are occlusion-sensitive [31].

Recently, methods for detecting and estimating the pose of object based on deep-learning techniques have been introduced. Such methods have gained great popularity in recent years, mainly due to their promising results. Some of these current methods are BB8 [31], SSD-6D [32], or the most recent DPOD [33]. Despite their promising results, these methods have not yet been used in an AR system applied to industrial maintenance [33], except for a few exceptions, such as the work of [34] who proposed an AR system applied to assembly. These new methods have mostly only been tested on pre-established public data sets, such as LINEMOD [15] or T-LESS [36]. This is mainly because they need many real images in which the real pose of the different objects of interest has been labeled or an accurate 3D model of the object. Moreover, some of these methods suffer when the tracked target is occluded, and may not be fast enough for an AR system [31–34], greatly limiting their real application in the industry.

At present, IRT is a mature technology, which can be used to monitor thermal conditions in real time and without contact. Thermography allows for the detection of equipment failures during operation, thus providing a significant reduction in downtime and associated maintenance costs [37]. Its use has been extended to different applications, including the monitoring of civil structures, electrical installations, machinery, and equipment; as well as in different industries, such as nuclear, aerospace, or food [38]. IRT has been successfully applied to the maintenance of electrical installations—mainly for the detection of faults in static machines, such as power transformers—although it has also been applied to the predictive maintenance of rotating electrical machines. The authors of [38] proposed a monitoring methodology based on IRT, to detect failures in induction motors, such as misalignments, cooling problems, imbalances between phases, or bearing damage. Its effectiveness was validated through the analysis of several practical cases. The results obtained in these investigations demonstrated an accuracy of 91% in the detection of failures and operating anomalies of the equipment monitored with infrared thermography. Regarding economic benefits, the work of Hakimollahi et al. [39] calculated a net annual benefit of 66,100 USD using IRT in maintenance activities of electrical equipment in a medium-sized distribution station

The proposed MANTRA system, explained in detail below, is based on a model-based method that joins together a template matching technique with 6DOF pose tracking to make use of the best attributes of each method. It also takes advantage of IRT technology, to increase the operator efficiency, easily showing the temperature over physical objects with virtual information.

## 3. Materials and Methods

In this section, the MANTRA system is described and decomposed into its different software modules: The 3D object detection and pose estimation method, the 6DOF pose tracking module, and the method for integrating the IRT information into the AR system. A diagram explaining the procedure can be seen in Figure 1.

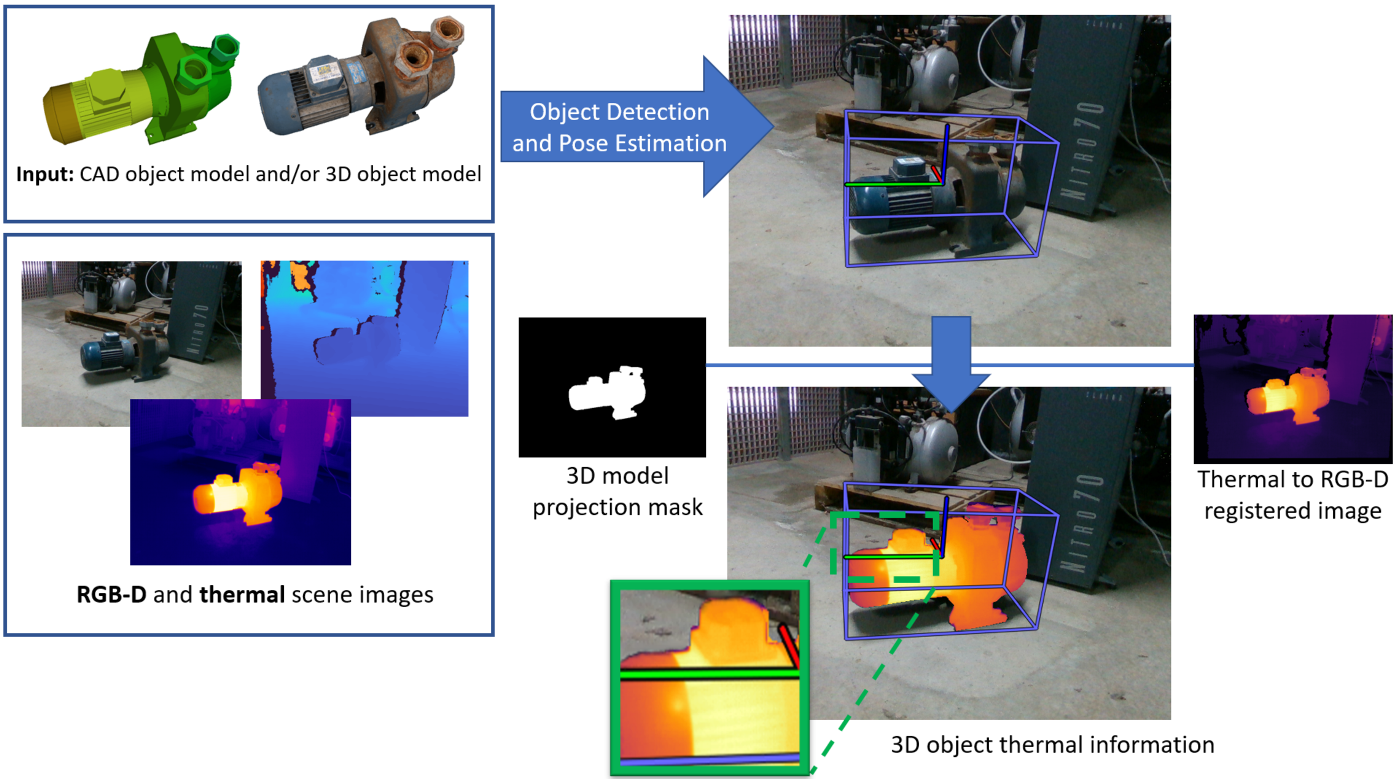
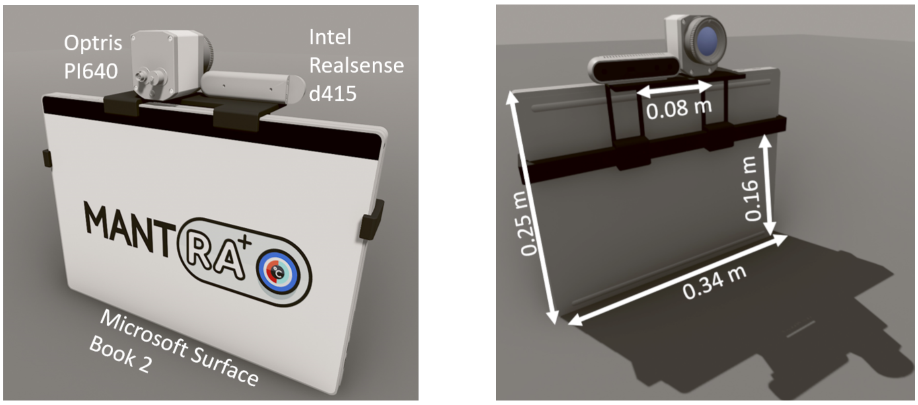

Input: CAD object model and/or 3D object model
RGB-D and thermal scene images
Object Detection and Pose Estimation
3D model projection mask
Thermal to RGB-D registered image
3D object thermal information

**Figure 1.** Process diagram of the object detection and pose estimation for combining IRT information.

### 3.1. MANTRA System Description

MANTRA is a handheld AR system composed of a Microsoft Surface Book 2 and two small cameras, as shown in Figure 2. These cameras are an Intel RealSense d415 RGB-D camera (Intel Corp., Santa Clara, CA, USA) with up to 1920 × 1080 color resolution and an Optris PI640 thermal imaging camera (Optris GmbH, Portsmouth, NH, USA) with 640 × 480 resolution horizontally aligned with each other. PI640 has a temperature measurement range of −20 °C to 90 °C with a spectral range of 8 to 14 µm. These cameras are attached onto the Surface display with a 3D designed support on top. Both cameras are powered directly from the USB port, making the whole system usable with the Surface battery. These cameras were chosen due its small weight and size with 72 g. the RGB-D camera and 320 g. (lens included) for the IRT camera. Another decision factor is that d415 camera is based on an active Stereo technology to obtain the 3D information that works both indoor and outdoor with a range up to 10 m and a minimum range of 0.16 m.



(**a**) Front view. (**b**) Back view.

**Figure 2.** Render model of the MANTRA hardware system.

A graphical user interface was created that allows for the dynamic loading of maintenance and repair tasks by reading XML files. This allows for the easy creation of use-cases by non-expert users. The interface was programmed with the Unity graphics engine, to ensure high visual quality. In addition, the depth information of the camera was used to superimpose the virtual models in a realistic way by using the z-buffering technique [40]. Z-buffering, also known as depth buffering, is a method to make the virtual content more realistic by taking into account real physical occlusions. Specifically, it uses the depth information of the RGB-D camera to render only virtual content that is closer to the camera than the physical scene.

The MANTRA software system is designed to be modular, such it can work with or without the IRT module and it is also future-proof, as each module can be updated separately. The following sections explain each of the modules that allow accurate virtual and thermal information to be placed over the physical object, as shown in Figure 1.

### *3.2. 3D Object Detection and Pose Estimation Module*

The main objective of the AR system is to automatically align the associated virtual content precisely onto any type of real object, for which the technician needs to obtain additional information about the task they are performing, in real time. To achieve this, it is necessary to detect and estimate the pose of 3D objects using computer vision techniques. The method used in this work consists of two parts: The first part is responsible for detecting and estimating the initial pose of the 3D object, while the second one is the 6DOF pose tracking module, which updates the pose of the object over time. The combination of both methods allows the system to work in real time more robustly. Specifically, it is robust to significant changes in the scale of the object, as well as changes in the lighting conditions of the scene and the object, allowing large occlusions of the object to occur; for instance, when the object is partially seen by the camera or when it is occluded by another object. The system is also robust to motion blur, can works with cluttered backgrounds, and has negligible jitter. The Figure 3 represents the 3D object detection and pose estimation workflow.

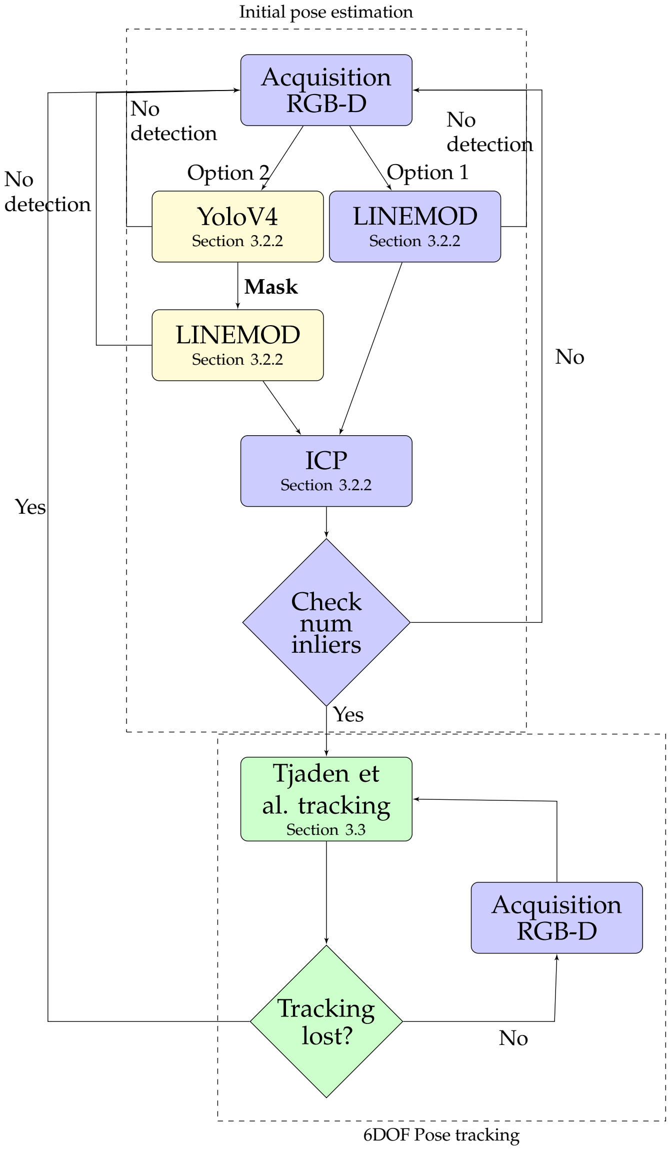


**Figure 3.** Flowchart of AR tracking method.

### 3.2.1. Foundation

In this section, the pose estimation problem is mathematically described, as well as the notation used throughout the work. A camera pinhole model is assumed, where $K_C \in \mathbb{R}^{3\times3}$ is the intrinsic matrix of the RGB camera, previously obtained in the calibration phase. In addition, we also assume that the depth image is registered to the color one and that the radial distortion has also been corrected. On one hand, let $\mathbf{x} = [x, y]^T \in \Omega \subseteq \mathbb{R}^2$ be a pixel of the image captured by the RGB camera, $I_{\text{rgb}}$, where $\mathbf{y} = I_{rgb}(\mathbf{x}) \in \mathbb{R}^3$ is the vector that represents the color of the image in the pixel $\mathbf{x}$. On the other hand, let $\mathbf{X}_o = [X_o, Y_o, Z_o]^T \in \mathbb{R}^3$ be one of the vertices of the 3D object in its coordinate system, whose representation in homogeneous coordinates is $\tilde{\mathbf{X}}_o = [X_o, Y_o, Z_o, 1]^T \in \mathbb{R}^4$.

Mathematically, the problem of estimating the pose of a 3D object consists of determining a matrix $G \in \mathbb{SE}(3)$ that relates the coordinate system of the object with the coordinate system of the camera, where

$$\mathbb{SE}(3) = \left\{ G = \begin{bmatrix} R & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \in \mathbb{R}^{4\times4} | R \in \mathbb{SO}(3), \mathbf{t} \in \mathbb{R}^3 \right\} \tag{1}$$

is the special Euclidean group, being a collection of rigid transform matrices, and

$$\mathbb{SO}(3) = \left\{ R \in \mathbb{R}^{3\times 3} | RR^T = I, det(R) = 1 \right\} \tag{2}$$

is the special Orthogonal group. In this way, we have that:

$$\tilde{\mathbf{X}}_c = G\tilde{\mathbf{X}}_o, \tag{3}$$

where $\tilde{\mathbf{X}}_c = [X_c, Y_c, Z_c, 1]^T \in \mathbb{R}^4$ is a point in the camera coordinate system.

Moreover, any point of a 3D model can be projected onto a 2D image plane using:

$$\mathbf{x} = \pi(K_C(G\tilde{\mathbf{X}}_o)_{3\times 1}), \tag{4}$$

where $\pi(\mathbf{X}) = [X/Z, Y/Z]^T$.

### 3.2.2. Method for Initial 3D Object Detection and Pose Estimation

An initial 3D object detection and pose estimation method was constructed, based on the work of Hinterstoisser et al. [15]. This method uses the LINEMOD method [41] to detect the 2D location of the 3D object in the image captured by the camera. Using this location, the initial pose of the object is inferred from the associated poses of the most similar templates. Finally, the initial pose is refined using both the depth information from the camera through an ICP algorithm [42], as well as the color information through HSV color space decomposition.

In detail, given an RGB-D image, $\mathrm{I}_{model}$ (obtained after rendering the 3D model from a certain point of view of the virtual camera), whose relative pose is $G_{\text{cam}} \in \mathbb{SE}(3)$, the template associated with this RGB-D image can be defined as $T = (\{O_m\}_{m\in\Lambda}, P)$, where $O$ is a set of template features (orientation of the gradient or orientation of the surface normals) and $\Lambda$ represents the modalities of the image (RGB image or depth image), while $P$ is a list of $r$ locations in $I_{model}$. Then, through a sliding-window approach, a template is compared with the captured RGB-D image $I_{cam}$ at location $c$, based on a similarity measure on its $L$ neighbors:

$$\epsilon_s(I_{cam}, T, c) = \sum_{r\in P} \max_{t\in L(c+r)} f_m(O_m(r), I_m(t)), \tag{5}$$

where $f_m(O_m(r), I_m(t))$ measures the similarity between the gradient orientations and the surface normals. A template is found in $I_{cam}$ if the similarity score, $\epsilon_s$, is greater than a certain threshold (generally $\geqq 80$). For more details of this method, see [41].

An important contribution is our optimization of the original LINEMOD method. Although the original work has low computational cost—thanks largely to the use of SSE instructions in the search for the most similar template in the image—in this work, various optimizations were also made, such as those proposed by Ivorra et al. [35], in such a way that the computational cost of the detection method was almost halved. Specifically, feature extractions were performed in separate threads for each color and depth images and numerous loops were parallelized.

One of the main limitations of the pose estimation method of Hinterstoisser et al. [41] is the precision obtained by the LINEMOD detection method, as it tends to produce both many false positives and false negatives on certain occasions when detecting the 3D object [43] in cluttered backgrounds (see Figure 4). This fact depends mainly on the characteristics of the 3D object to be identified in the RGB-D image. Specifically, 3D objects with simpler geometries are most likely to be affected by this phenomenon, due to two reasons: The first reason is that simple 3D objects generate templates with regular and simple geometric contour shapes, such as rectangles, circles, or ellipses, which match with the projections of many other 3D objects within the scene, causing false positives [43]. The second reason is due to possible ambiguities in relating a template to a pose when two or more virtual views generally have the same template; for example, in the case of symmetrical objects. Another important factor, when obtaining false positives, is the number of templates used to represent the 3D object: As this number increases, it is more likely that a greater number of false positives will be obtained when detecting the object in a cluttered environment [43]. Finally, another critical factor is the similarity score used, as a low value may produce false positives and a too high similarity score would make it such that the object will not be detected in the image.

Various methods can be found in the literature which attempt to solve this problem using post-processing techniques, such as the use of color information and the ICP algorithm [15] or template clustering [43]. The solution proposed in this work is to employ an object detection method based on a deep-learning technique, which improves the accuracy of the method and, thus, reduces both the number of false positives and false negatives in the LINEMOD 3D object detection phase. This approach is similar to that described in [43], in which the SSD algorithm [44] was employed; however, we use the new state-of-the-art object detector, YOLOV4 [45]. The real-time detector algorithm YOLOV4 had a 10% higher average precision (AP) on the COCO [46] data set and was also 12% faster than the previous version YOLOV3 [47]. Moreover, YOLOV4 had 14.7% more AP than SSD on the COCO data set employed by [43]. YOLOV4 overpasses in speed and AP also the object detection RetinaNet method [48] with a 2.4% more AP and on COCO data set. Finally, the popular object detection Faster-RCNN [49] similarly scored 3.7% AP below YOLOV4 on COCO data set. Even more, RetinaNet and Faster-RCNN methods are not fast enough for an AR system. Therefore, YOLOV4 is recognized as one of the better and faster object detection method on COCO dataset available currently. For a more detailed comparison please check the work of [45]. It is important to note that the LINEMOD method cannot be fully replaced by YOLOV4, as it is still needed to provide the relative 3D object pose. This combination of methods is called LINEYOLO in this work.

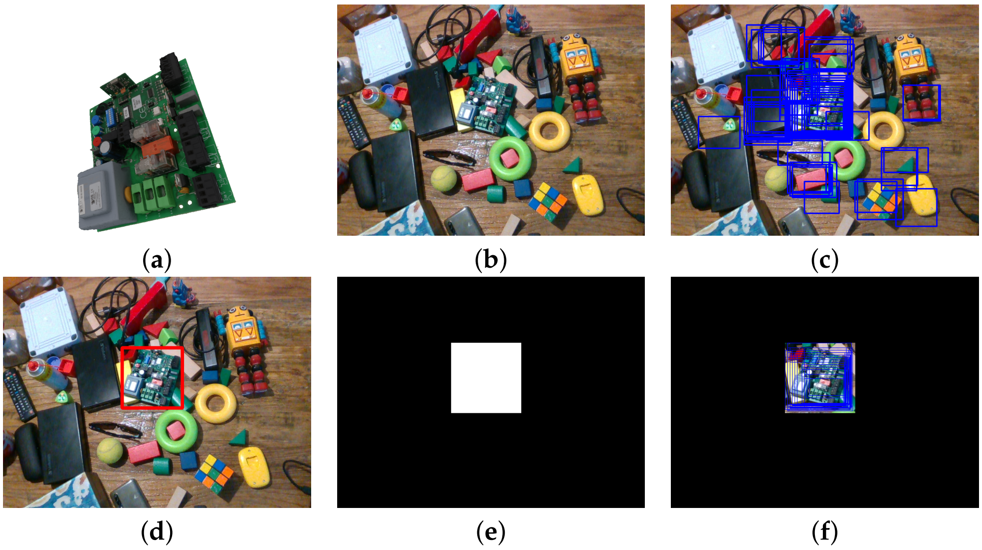


**Figure 4.** LINEYOLO method—LINEMOD with YOLOV4: (**a**) 3D model; (**b**) Raw cluttered scenario image; (**c**) Detection results using the LINEMOD method; (**d**) Detection results using YOLOV4; (**e**) Extracted mask from the result of the YOLOV4 algorithm; and (**f**) LINEMOD detection method results after using YOLOV4 algorithm ROI selection.

The LINEYOLO procedure is summarized in Figure 4. Specifically, in Figure 4c, the LINEMOD method gave many false positives when detecting the 3D object in the camera's image (due the simple object characteristics previously explained). On the contrary, when YOLOV4 [45] was used to detect the object, it can be observed that it detected it univocally and precisely (Figure 4d). Consequently, prior to the 2D location phase of the object by the LINEMOD method, the information extracted by the YOLOV4 algorithm is used. This information is determined by a mask extracted from the 2D location of the object using

YOLOV4 (Figure 4e). This mask determines a ROI on which to apply the LINEMOD detection method. On one hand, the LINEMOD features are extracted only in that region of the RGB-D image; on the other hand, the search for the most similar template is only carried out in this region of the image (Figure 4f). This reduces the number of similar templates when searching and, consequently, the method of estimating the pose is more accurate. In our case, the ROI determined by YOLOV4 was enlarged by 10%, as indicated in the work of [43].

The procedure used to carry out the post-processing and obtain both the most similar template and the final pose of the 3D object is similar to that described in [15]; however, in our case, the color check step is not performed. We made this decision to be able to work with CAD models. In addition, unlike [15], to determine the most similar template, the ICP algorithm is performed for each of the candidate templates in parallel, using independent threads. Finally, the most similar template is determined, according to the higher number of inliers obtained by each ICP. For efficiency reasons, the ICP algorithm is carried out only in the four most similar templates.

Finally, it should be noted that despite the optimizations in LINEYOLO, it still has a series of important limitations due to the use of the LINEMOD method. Specifically, the LINEMOD method suffers from the so-called occlusion problem [31], either because the object is partially out of the camera's field of view or is partially covered by another object. Another limitation of this method is that it is not capable of detecting the object when there are significant changes to its scale, as the LINEMOD method is not scale-invariant. In turn, despite the various optimizations made in the LINEMOD method, the pose of the object is obtained at an average rate of 8 fps Section 4, mainly due to the use of the ICP algorithm. Another aspect to highlight is the jitter produced when estimating the pose, as this is calculated independently in each frame. Therefore, to overcome these limitations, we decided to incorporate a 6DOF pose tracking model to update the initial pose of the 3D object over time.

### 3.2.3. Training Detection Model

One of the biggest drawbacks of detection methods based on deep-learning techniques is the need to create a large enough labeled image data set of the objects to detect prior to training the deep-learning model. This is problematic due the difficulty of finding enough representative images of a particular instance of an object and labeling them, which is very time-consuming. Moreover, the labeling task is usually done manually, consequently entailing that the user takes a long time to annotate many images; in addition to the fact that human errors may occur, even though it is done by an expert user.

There are several alternatives that allow for considerable reduction of the previously mentioned limitations, using methods that allow for the generation of synthetic images to carry out the training of the object detection model. This saves a great deal of time, both in labeling and collecting the information, as this is done automatically; additionally, no mistakes are made in the labeling process. Although [50] showed that using synthetic images in conjunction with real images can improve the performance of detection methods, this was not the case when only synthetic images were used to train the model. In this case, object detector methods trained only with synthetic data sets do not transfer well to real scenes, given the domain gap between real and synthetic data and, consequently, a significant loss of performance can occur [51]. To solve this problem, various alternatives have been proposed, such as the use of methods that allow for the generation of photorealistic synthetic images [52], as well as the use of domain adaptation methods [51]. Specifically, in this work, BlenderProc [18] was used; in particular, its variant known as BlenderProc4BOP. In Figure 5 different image examples generated by BlenderProc4BOP can be seen. Employing this automatic procedure, an unlimited number of images, both color and depth, which are rich in variability can be obtained, with precise associated information about object poses, masks, regions of interest, and labels.

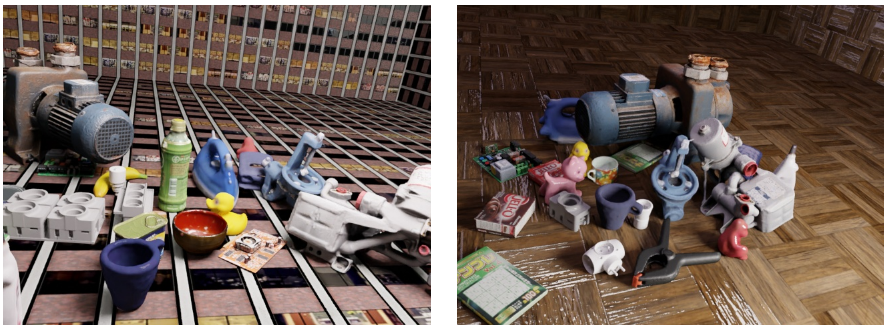


**Figure 5.** Photorealistic synthetic images generated using BlenderProc4BOP [18] with some of the 3D models used in our experiments, where other 3D models of different data sets, such as LINEMOD [15] or T-LESS [36], were used as distractors.

Darknet [16] was used as a framework to train the detection model based on YOLOV4. To carry out the different experiments, around 20,000 synthetic images were generated for each of the 3D objects split into 80% for training and 20% for testing. To train the YOLOV4 network, the initial weights calculated on the COCO data set [46] were first used. Subsequently, the network was fine-tuned using only the synthetic images generated by BlenderProc4BOP. Table 2 lists the main training parameters employed.

**Table 2.** Main parameters used to train YOLOV4.

| Input Images Size | Batch Size | Subdivisions | Momentum | Initial Learning Rate | Decay | Max Batches |
|---|---|---|---|---|---|---|
| 416 × 416 | 64 | 8 | 0.949 | 0.0003 | 0.0005 | 40k |

### *3.3. 6DOF Pose Tracking Module*

In this section, the 6DOF pose tracking method used in the AR system is described. The 6DOF pose tracking method used is the one proposed by Tjaden et al. [17], as it meets all the requirements to solve the technical limitations of the method of estimating the pose based on LINEMOD. Basically, this tracking method consists of iteratively solving a non-linear optimization problem of the parameters that define the rigid transformation of an object between two consecutive frames. Consequently, the first step to take is to determine a simpler representation of the object pose by reducing the number of parameters from 12 (9 for rotation and 3 for translation) to 6 by using a Lie Algebra. More specifically, to represent the rigid movement of an object between two consecutive frames, a twist vector $\xi = [t_x, t_y, t_z, w_x, w_y, w_z]^T \in \mathbb{R}^6$ is used, whose representation by the Lie algebra is

$$\widehat{\xi} = \begin{bmatrix} 0 & -w_x & w_y & t_x \\ w_z & 0 & w_x & t_y \\ -w_y & w_x & 0 & t_z \\ 0 & 0 & 0 & 0 \end{bmatrix} \in se(3), \tag{6}$$

where $se(3)$ is the Lie algebra associated with the Lie group $\mathbb{SE}(3)$ [17]. On the other hand, a twist vector can be converted into an element of the Lie group $\mathbb{SE}(3)$ by an exponential function $\exp(\cdot)$ using the Rodrigues formula [53]. Once the new parametrization of the pose has been carried out, the next step of the tracking method consists of projecting the 3D model onto the 2D plane of the image captured by the camera using the initially pose estimated (see Equation (4)). The 2D contour of the 3D object projected onto the 2D plane of the image, which we denote as $C$, divides the image into two disjoint regions: $\Omega_f \in \Omega$ (foreground) and $\Omega_b = \Omega \backslash \Omega_f$ (background) (see Figure 6).

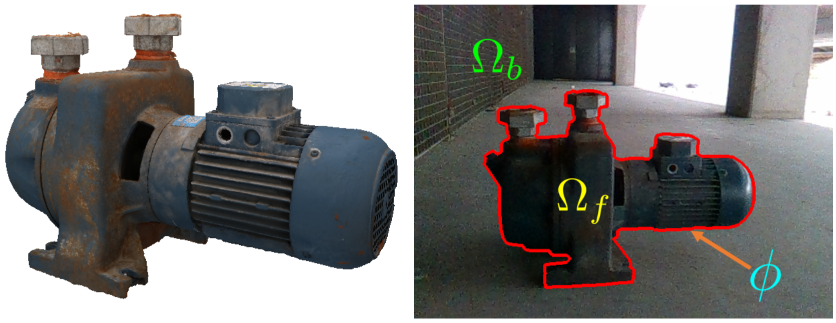


**Figure 6.** Left Image: 3D model. Right Image: Initialization of the tracking method by projecting the 3D model on the 2D image captured by the camera for a certain pose $G$ (see Equation (4)). The projected object contour determines the background, $\Omega_b$, and the foreground, $\Omega_f$.

On the other hand, $C$ can be represented by a level set function, such that $C = \{\mathbf{x}|\phi(\mathbf{x}) = 0\}$, where $\phi(\mathbf{x})$ is defined as:

$$\phi(\mathbf{x}) = \begin{cases} -d(\mathbf{x}, C) & \forall \mathbf{x} \in \Omega_f \\ d(\mathbf{x}, C) & \forall \mathbf{x} \in \Omega_b \end{cases} \tag{7}$$

and $d(\mathbf{x}, C) = \min_{c \in C} |c - \mathbf{x}|$. The next step of the method is to determine a function that maximizes the discrepancies between the appearance of the background and the foreground with respect to its pose. Specifically, the functional measures the separation between the background and the foreground with respect to the 2D shape determined by $\phi$.

Each region has its own statistical model of appearance: $P(\mathbf{y}|M_b)$ for the background and $P(\mathbf{y}|M_f)$ for the foreground, where $M_b$ and $M_f$ are the parameters of the background and foreground models, respectively. Specifically, $P(\mathbf{y}|M_f)$ and $P(\mathbf{y}|M_b)$ represent the probabilities that a pixel with color $\mathbf{y} = I_{rgb}(\mathbf{x})$ belongs to the foreground or background region, based on the information obtained from the color histogram. Then, taking into account the work of [17] and assuming that the pixels are independent of each other, the resolution of the 6DOF pose tracking problem is based on the minimization of the following energy function with respect to the pose parameters $\xi$:

$$E(\xi) = -\sum_{\mathbf{x} \in \Omega} \log(H_e(\phi(\mathbf{x}))P_f(\mathbf{x}) + (1 - H_e(\phi(\mathbf{x}))P_b(\mathbf{x})), \tag{8}$$

where $H_e$ is the smoothed Heaviside function, and $P_f$ and $P_g$ are the posterior probabilities of each pixel belonging to the foreground or background, respectively, such that:

$$P_m(\mathbf{x}) = \frac{P(\mathbf{y}|M_m)}{\tau_f P(\mathbf{y}|M_f) + \tau_b P(\mathbf{y}|M_b)} \quad m \in \{b, f\}, \tag{9}$$

$\tau_f$ and $\tau_g$ being the respective areas of $\Omega_f$ and $\Omega_g$.

To make the object pose method more robust against cluttered backgrounds and dynamic changes in the appearance of the scene, the appearance is analyzed only locally in $N_p$ regions, $\{\Omega_n\}_{n=1}^{N_p}$, selected randomly around the object's contour. Specifically, the color histogram is analyzed in each region of the image $\Omega_n := \{\mathbf{x} \text{ with } |\mathbf{x} - \mathbf{x}_n| < r\}$, where $\mathbf{x}_n$ is a pixel that belongs to the contour. In this way, each $\Omega_n$ is divided into two regions: $\Omega_{f_n} \subseteq \Omega_n$ for the foreground and $\Omega_{b_n} \subseteq \Omega_n$ for the background. Then, according to the work of [17], the energy function to optimize, in which only the appearance information is incorporated in a region around the object's contour, is as follows:

$$E(\xi) = -\sum_{\mathbf{x} \in \Omega} \log(H_e(\phi(\mathbf{x}))\overline{P}_f(\mathbf{x}) + (1 - H_e(\phi(\mathbf{x}))\overline{P}_b(\mathbf{x})), \tag{10}$$

where

$$\overline{P}_m(\mathbf{x}) = \frac{1}{\sum_{i=1}^{N_p} B_i(\mathbf{x})} \sum_{i=1}^{N_p} P_{m_i}(\mathbf{x}) B_i(\mathbf{x}) \quad m \in \{b, f\} \tag{11}$$

and

$$B_i(\mathbf{x}) = \begin{cases} 1 & \forall \mathbf{x} \in \Omega_i \\ 0 & \forall \mathbf{x} \notin \Omega_i \end{cases} \tag{12}$$

To solve this non-linear optimization problem, there are several alternatives. The proposal made by Tjaden et al. [17], which is followed in this work, consists of using a second-order Gauss–Newton method. This approach requires reformulation of the optimization problem as a non-linear least squares weights estimation problem. Finally, in each iteration of the Gauss–Newton method, the object pose is updated from the solutions obtained in the least squares problem:

$$\Delta\xi = -\left( \sum_{\mathbf{x} \in \Omega} H(\mathbf{x}) \right)^{-1} \sum_{\mathbf{x} \in \Omega} J(\mathbf{x})^T, \tag{13}$$

using Cholesky decomposition, where $J$ is the Jacobian and $H$ is the Hessian at pixel $\mathbf{x}$. Consequently, if $G^t \in \mathbb{SE}(3)$ is the pose of the 3D object at time $t$, then the pose of the 3D object at time $t+1$, $G^{t+1} \in \mathbb{SE}(3)$ is defined by

$$G^{t+1} = \exp(\widehat{\Delta\xi}) G^t, \tag{14}$$

where $G^0 \in \mathbb{SE}(3)$ is the initial pose of the 3D object obtained by the LINEMOD method at time $t = 0$.

On the other hand, to improve the motion blur, as well as to reduce the computational cost, the update procedure of the 3D object's pose is performed using an image multi-scale decomposition. Finally, once the pose optimization process is finished, the tracking method detects that it has lost the object if $\frac{E}{|\Omega|} > \lambda$ with $\lambda \in [0.5, 0.6]$. In this case, the AR system relocates the 3D object in the image by calling the method based on LINEMOD for detecting and estimating the pose of 3D objects again.

### *3.4. RGB-D and IRT Images Fusion Module*

The objective of this module is to integrate IRT information into the AR system. In this way, it is possible to show the temperature of an isolated 3D object, as well as that of its various components, on top of the color image with associated virtual information, both precisely and in real time. To achieve this objective, it is necessary to calibrate each of the sensors, place them in the same coordinate system (extrinsic parameters), as well as overlap the registered images using the information of the previously calculated 3D object pose.

#### 3.4.1. Sensor Calibration

It is important to have a good camera calibration of the different sensors, to be able to merge their information. The used sensors were placed in a horizontal parallel configuration, as shown in Figure 7. For this reason, stereo calibration is carried out to obtain accurate intrinsic and extrinsic parameters that relate the local coordinate systems of the RGB-D camera and with the IRT camera, assuming a pinhole camera model for both. Furthermore, with the stereo calibration, the fourth-order coefficients of the radial and tangential distortion for each of the cameras are calculated. The distortion model employed is Brown's model [54]. In addition to radial distortion, it also models tangential distortion, which occurs when the lens is not aligned with the sensor. These distortions are compensated with the following equations:

$$\begin{aligned} x_u &= x_d + \overline{x}(\kappa_1 r^2 + \kappa_2 r^4 + \kappa_3 r^6 + \cdots) + \\ &\quad [\rho_1(r^2 + 2\overline{x}^2) + 2\rho_2 \overline{x}\,\overline{y}](1 + \rho_3 r^2 + \cdots) \\ y_u &= y_d + \overline{y}(\kappa_1 r^2 + \kappa_2 r^4 + \kappa_3 r^6 + \cdots) + \\ &\quad [\rho_2(r^2 + 2\overline{y}^2) + 2\rho_1 \overline{x}\,\overline{y}](1 + \rho_3 r^2 + \cdots) \end{aligned} \tag{15}$$

where $[x_u, y_u]^T$ is the undistorted image point, $[x_d, y_d]^T$ is the distorted image point, $[x_c, y_c]^T$ is the center of distortion, $\overline{x} = x_d - x_c$, $\overline{y} = y_d - y_c$, $r^2 = \overline{x}^2 + \overline{y}^2$, $\kappa_n$ is the $n$th radial distortion coefficient and where $\rho_n$ is the $n$th tangential distortion coefficient.

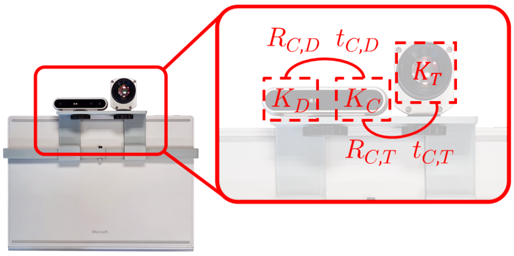


**Figure 7.** Used setup.

The factory calibration of the RGB-D camera is used to have the depth sensor registered on the color sensor, such that it is only necessary to calculate the intrinsic parameters of the thermal camera $K_T$ and the extrinsic parameters $R_{C,T}$ and $t_{C,T}$ seen in Figure 7. The extrinsic parameters are constant, due to the fixed positioning of the sensors relative to each other. To perform the stereo calibration, the method of Zhang et al. [55] is used. In detail, the calibration is calculated using multiple images from different points of view and scales of the calibration target pattern, whose 3D points are known and constant, and performing a RANSAC [56] optimization to minimize the reprojection error.

Although a traditional calibration target, such as a chessboard, is usually used to calibrate a color camera, to calibrate the thermal camera, it was necessary to design a special calibration target; this was because the contrast obtained in the thermal image was not enough to calculate accurate correspondences between points precisely.

### 3.4.2. Calibration Target Used to Calibrate the IRT Camera

Standard calibration patterns, such as a printed chessboard, are not suitable for IRT calibration due to their near-uniform temperature. To solve this problem, the calibration target boards must be made of materials with two different emissivities. Several techniques and pattern designs have been proposed to be able to calibrate thermal cameras. In [57], a very clear calibration target classification was made; they can be classified based on working principle (active or passive), on the markers employed (e.g., corners, circles, Aruco, and so on), and the kind of pattern that the markers follow (structured or unstructured). The most interesting target classification for IRT is that based on the working principle. Passive targets do not use any external energy sources to be detectable by the sensor, while active ones need external energy sources to be detectable. The target board designed in this work was an active evolution of the passive asymmetric circle board constructed in [57]. Specifically, circles in a 9 × 3 asymmetric pattern were cut out of white stainless steel plate with a CNC laser cutter machine. It had the same size as an A4 paper sheet and a thickness of 1 mm. To be able to heat the target board, several nichrome wires (with 0.2 mm diameter) were attached across the back of the plate, surrounding the circles of the board to ensure that there was enough temperature contrast to detect the circle contours accurately. Then, the target board was screwed into a black-painted wood plank, leaving a 2 cm gap between the board and the wood base, to isolate the nichrome wires from the wood base. For convenience and portability, a 5 V/2.1 A powerbank powered the nichrome circuit. The calibration pattern reached a temperature which was safe to be handled but high enough to obtain high-contrast IRT images (see Figure 8).

The reprojection error obtained with the geometric calibration of the IRT camera, using 50 images of varying orientation and distance to the camera, gave a result of 0.332 ± 0.09 pixels. This error was similar to that obtained in [58] with custom-made calibration algorithms. A reprojection error greater than 0.8 pixel would indicate to a MANTRA user that the calibration process is not good enough and it has to be repeated.

The developed calibration pattern is effective for long periods of time (depending on the powerbank capacity), accurate for RGB-D and IRT cameras, can be employed with optimized conventional algorithms, is portable, and is not difficult to manufacture.

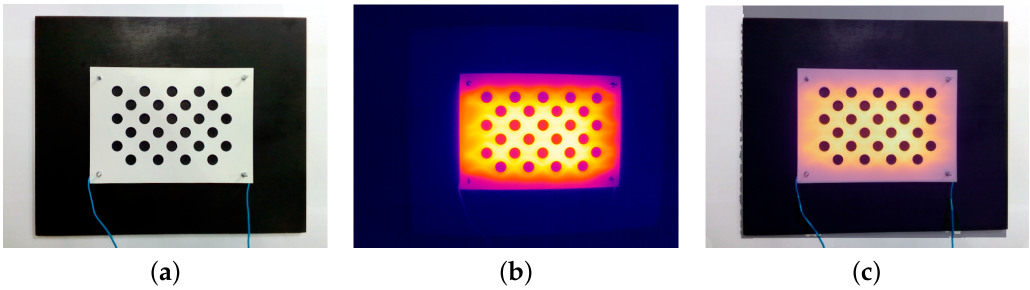


**Figure 8.** Calibration target board: (**a**) Seen with the color sensor; (**b**) Seen with the IRT sensor; and (**c**) After blending the color and thermal images.

### 3.4.3. Thermal and RGB-D Image Fusion

A one-to-one pixel correspondence between sensors is needed, to be able to overlap the thermal view with the RGB one. The process to obtain this correspondence is known as registration between sensors. The color and depth sensors of the RGB-D camera are already registered, such that only registering the IRT camera with the RGB-D one is required. The registration process can be performed as follows: First, the 3D point cloud is calculated from the depth image ($I_{depth}$) using the RGB camera intrinsic parameter; then, this 3D point cloud is transformed using the rigid transformation defined by the extrinsic parameters obtained in the calibration $[R_{C,T}, t_{C,T}]$ (see Figure 7); finally, this 3D point cloud is projected, using Equation (4), onto the IRT image plane using the IRT camera intrinsic parameters. The 3D projected points give the correspondence between IRT pixels and RGB pixels, obtaining a registered IRT image. Then, a more explainable image can be obtained by performing a blending procedure to mix the color image $I_{\text{rgb}}$ and the registered IRT image $I_{\text{thermal}}$ using alpha channel mixing.

In addition, knowing the object pose allows us to show only the temperature of the object and not that of the whole scene. Moreover, it is accurate enough to even extract the temperature of each object component. This segmentation is performed by calculating a mask $I_{mask} = I_{model\text{-}depth} > 0$ and applying it onto the registered IRT image of $I_{\text{thermal-rgb}}$, such that we have:

$$I_{compose} = \alpha \cdot I_{\text{thermal-rgb}} \cdot I_{mask} + (1 - \alpha) \cdot I_{\text{rgb}}. \tag{16}$$

In Figures 1 and 8c, two examples of the fusion of RGB and IRT information with different $\alpha$ values can be seen.

## 4. Results

In this section, the quantitative results of the pose detection and estimation method are described, as well as a detailed analysis of the time efficiency of the different methods. Finally, the efficiency of MANTRA in a real maintenance use-case is detailed.

### *4.1. Quantitative Validation of the Results of the 3D Object Pose Detection and Estimation Method*

To quantitatively evaluate the influence of the incorporation of the YOLOV4 algorithm to the detection and estimation method of the initial pose of 3D objects, the LINEMOD (LM) [15] and LINEMOD-OCCLUDED (LM-O) [59] data sets were used. On one hand, the LM data set consists of 15 non-textured 3D objects, for which 15 color 3D models are available. For each of the models, a sequence of RGB-D test images in which the object of interest appears in a cluttered environment with the real pose annotated ("ground truth") is available. On the other hand, the LM-O data set consists of 8 objects extracted from the LM data set, in which these 8 models appear simultaneously, with different levels of occlusion and where the actual pose of each object has also been annotated. To evaluate the performance of the method, the average distance of the model points (ADD) [15] and the 2D projection error (Proj.2D) [60] were used as metrics. The pose is generally considered correct if the ADD is less than 10% of the object's diameter while Proj.2D metric considers that a pose is correct if it is less than 5 pixels.

Before performing the quantitative validation of the initial pose estimation method, the precision of the YOLOV4 detection algorithm was validated on both the LM and LM-O data sets, as the precision of this directly influences the object pose estimation method. For this, 50,000 synthetic images were generated using BlenderProcBop Section 3.2.3 for training the YOLOV4 algorithm, in which the different LM objects appear (at least 8 LM different models), as well as distractors from other data sets. This algorithm was tested against the LM and LM-O RGB-D real test images. The results obtained, as measured with the mAP@.5 IOU and mAP@.75 IOU metrics, are shown in Table 3.

**Table 3.** Accuracy obtained by the YOLOV4 detection method in the LM and LM-O data sets.

| Dataset | mAP@.5 IOU | mAP@.75 IOU |
|---|---|---|
| LM | 99.01% | 94.24% |
| LM-O | 91.46% | 72.55% |

In Table 4, the LINEMOD and LINEYOLO pose estimation results on the LM and LM-O data sets can be seen. It is important to highlight that unlike the work of [15], the same parameters were used for all the objects to estimate the pose; specifically, $\epsilon_s = 80$ was used as the similarity score of the LINEMOD method. Also, as post-processing, the ICP algorithm was only performed on the four most similar templates; using, in this case, the same parameters for the ICP algorithm in all models.

Taking into account the results obtained (see Table 4), it can be seen that the incorporation of the YOLOV4 detection algorithm improved the results by 7.89% when using ADD as a metric and by 7.88% when using Proj.2D, in terms of the results obtained by the pose estimation method in the LM data set. This difference was even more significant in the LM-O data set, the improvement in this case being 13.18% using ADD metric. This indicates that the use of YOLOV4 improved the results of estimating the object pose, especially when occlusion occurred over the object. Song et al. [61] recently reported the results of the most significant pose estimation methods based on deep-learning techniques from the scientific literature on different data sets. Using this report as a benchmark, LINEYOLO used in this work improved upon the results of all the evaluated methods in the LM and LM-O data sets with the ADD measure, except for the DPOD method [33], which obtained an ADD of 95.2% in the LM data set. This fact indicates that LINEMOD methods is still a competitive object pose estimation method.

Despite the results obtained (see Table 4) with LINEYOLO, this method still suffered when there was an occlusion. This fact is a problem as, for example, the user will need to manipulate the 3D object on many occasions and, consequently, the implemented method will not be able to estimate its pose. Therefore, to solve this and similar problems, a 6DOF pose tracking method was incorporated into the AR system.

**Table 4.** Results of the LINEMOD-based pose detection and estimation system on the LM [15] and LM-O [59] data sets using different metrics with LINEYOLO or with LINEMOD. The percentage is calculated as the number of times the pose was correctly estimated with respect to the total number of images for each of the sequences.

| Model | LM | | | | LM-O | | | |
|---|---|---|---|---|---|---|---|---|
| | LINEYOLO | | LINEMOD | | LINEYOLO | | LINEMOD | |
| | ADD | Proj.2D | ADD | Proj.2D | ADD | Proj.2D | ADD | Proj.2D |
| Ape | 97.97% | 98.62% | 94.17% | 94.41% | 63.24% | 68.11% | 58.80% | 62.73% |
| Bench Vise | 99.01% | 98.84% | 97.94% | 97.04% | × | × | × | × |
| Bowl | 96.48% | 93.34% | 95.13% | 94.32% | × | × | × | × |
| Cam | 90.00% | 89.67% | 89.34% | 89.34% | × | × | × | × |
| Can | 92.05% | 95.31% | 89.54% | 93.06% | × | × | × | × |
| Cat | 99.15% | 99.32% | 96.52% | 96.52% | 63.19% | 64.66% | 49.18% | 50.90% |
| Cup | 95.72% | 79.75% | 91.12% | 75.32% | 17.42% | 17.69% | 12.58% | 12.58% |
| Driller | 88.55% | 86.36% | 81.73% | 78.11% | 36.88% | 34.77% | 27.81% | 26.32% |
| Duck | 96.73% | 98.96% | 95.77% | 96.65% | 60.28% | 69.29% | 54.08% | 60.38% |
| Box | 98.48% | 98.16% | 98.96% | 98.80% | 29.57% | 25.72% | 18.30% | 17.14% |
| Glue | 91.55% | 88.85% | 56.39% | 54.67% | 33.33% | 27.92% | 8.67% | 8.30% |
| Hole Punch | 97.00% | 97.49% | 74.13% | 74.13% | 90.05% | 90.68% | 59.06% | 59.19% |
| Iron | 96.78% | 96.70% | 92.79% | 92.62% | × | × | × | × |
| Lamp | 92.99% | 91.76% | 84.59% | 83.86% | × | × | × | × |
| Phone | 91.15% | 90.82% | 67.09% | 66.93% | × | × | × | × |
| Average | 94.90% | 93.59% | 87.01% | 85.71% | 49.24% | 49.85% | 36.06% | 37.19% |

Finally, the 6DOF pose tracking method used in this work uses the original implementation of the method by Tjaden et al. [17]. This method was quantitatively evaluated by the authors, obtaining results that significantly improved upon those of methods such as PWP3D [30]. In addition, the validation performed showed that the 6DOF pose tracking system is robust against occlusion and significant changes in the scale of the object, as well as dynamic changes in the lighting of the scene, among other factors. Some qualitative results of the MANTRA-implemented 6DOF pose tracking can been seen, in Figure 9, for different 3D objects (i.e., water pump and fuel filter of an airplane motor). These images show that the system can obtain the pose of the 3D object precisely against significant changes in object scales, occlusion, and blurring.

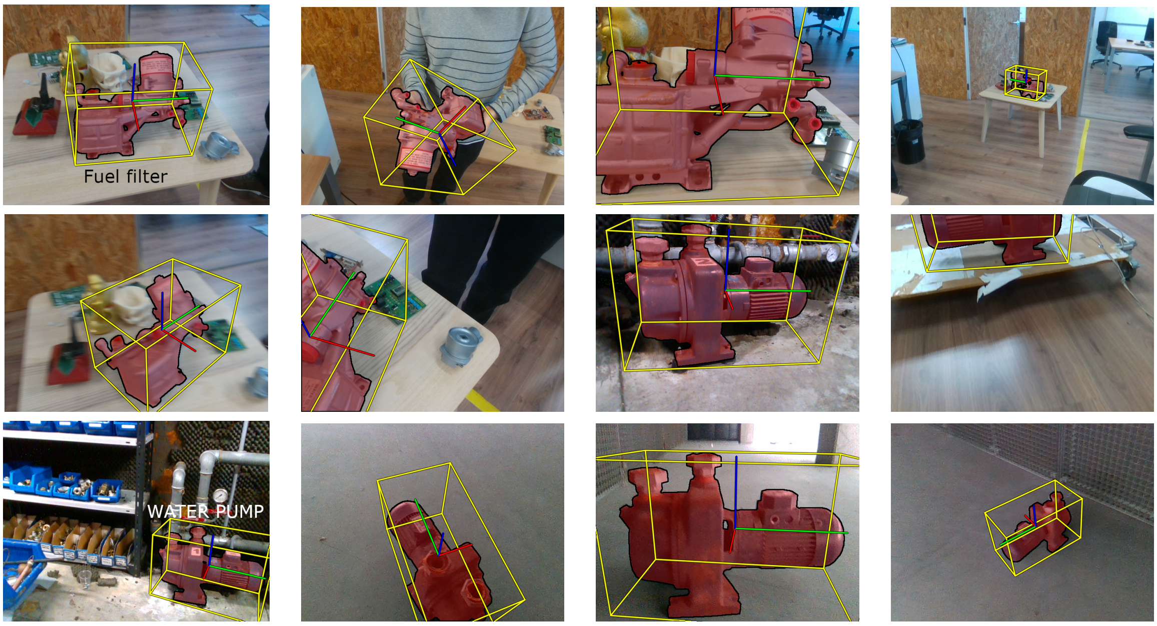


**Figure 9.** Qualitative results of the pose estimation method implemented in this work for different 3D models. The red color of the 3D model indicates the projection of the 3D model on the 2D image with the obtained pose.

### *4.2. Time Efficiency*

In this section, the time efficiency of the main modules of the AR system is determined: Detection and estimation of the initial pose based on LINEMOD (with and without YOLOV4), the 6DOF pose tracking method, and registration of the IRT image with the RGB-D one. To quantify the algorithm execution times, the hardware device used was a Microsoft Surface Book 2 with a 1.90 GHZ i7 processor, 16 gigabytes of RAM, and an NVIDIA Geforce GTX 1060 graphics card. The images from the RGB-D camera were captured at 60 fps with a resolution of 640 × 480. The LINEMOD 3D model data set was used to carry out the tests, as well as three 3D models generated in this work: An electronic board, an airplane combustible filter, and a water pump. A series of observations must be made regarding the different modules. First, for the initial pose estimation method, the same number of templates per 3D model was used as those cited by Hintertoisser et al. [15]; in particular, 1235 templates. Second, the same similarity score was used in all cases, $\epsilon_s = 80$. It should be noted that the higher the similarity score, the faster the algorithm works to find the most similar LINEMOD template, as the search is reduced to a smaller number of candidates. The average time execution values are compiled in Table 5. The time efficiency of the optimized LINEMOD method [35] without using the ICP algorithm was 0.069 s (14.49 fps). On the other hand, as mentioned in Section 3.2.2, incorporation of the YOLOV4 detection algorithm into the preprocessing phase sped up the LINEMOD calculation to 0.035 s but, to this time, the YOLOV4 execution time of 0.036 should be added, such that the LINEYOLO total execution time was 0.071 s. In summary, the overall initial detector and pose estimation phase execution time was 0.122 s (8.19 fps), including the Parallel ICP using LINEYOLO (option 2), or 0.12 s without YOLOV4 (option 1). As can be seen, use of the YOLOV4 algorithm did not greatly influence the computational cost, as a similar performance was obtained in both cases.

Moreover, regarding the 6DOF pose tracking module, an average rate of 43.4 fps was obtained. It should be noted that this time depends on several factors, including the number of polygons of the 3D object and the size that the object takes up in the image captured by the camera. On the other hand, the registration of the IRT and the RGB-D images was performed at 83.3 fps. Taking into account that most of the time the AR system will only use the 6DOF pose tracking and the registration modules, one of the main objectives set out in this work is met: The proposed AR system works in real time.

**Table 5.** Average time (in seconds) of the main methods used in the AR module.

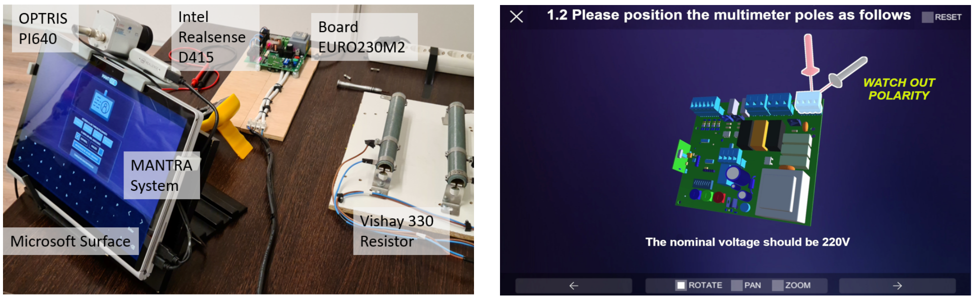
| PHASE | PARTS | | TIME (s) |
|---|---|---|---|
| Initial detector and pose estimation | Option 1: | LINEMOD | 0.069 |
| | Option 2:<br>LINEYOLO | YOLOV4<br>LINEMOD | 0.036<br>0.035 |
| | Parallel ICP | | 0.051 |
| 6DOF pose tracking | Tjaden et al. method [17] | | 0.023 |
| IRT Fusion | IRT to RGB-D Registration | | 0.012 |

### *4.3. Maintenance Task Use-Case Efficiency Validation*

A maintenance task of an electronic board was chosen as a use-case for testing the MANTRA system. Electronic boards are low-power devices that have generally fairly low temperature levels, in which the highest temperature points can be easily located. Specifically, the electronic board selected was a VDS EURO230M2 control board. This small control board can control two motors of up to 550 W each and is powered by a 220 V alternating current. An output load (typically the motors) was simulated using a Vishay 330 Ω high-power variable resistor. The setup used is shown in Figure 10.

Five usual non-destructive faults suffered by these kinds of electronic boards were chosen and simulated. Specifically, the electronic board could be in any of the following conditions:

1. Reference state: This is the case where the electronic board works correctly. In this state, the board is programmed to complete the electromechanical process in 5 s by pressing the operation button.
2. Blown fuse: This consists of the detection and replacement of one or more fuses on the board. To simulate this failure, a blown fuse was installed.
3. Bad input connections: This occurs when a wire has come loose from the power input terminals or has burned out and does not conduct electricity.
4. Relay malfunction: Relays control the activation of the outputs. If the mechanism is not making full contact or is stressed due to higher inrush current, carbon build-up can happen, due to excessive electrical arcing. This could lead to blockage, such that they cannot open or close the circuit.
5. Bad contact in load outputs: This occurs when the wires have come loose from the output terminals and are not making good contact. This failure, which can lead to more severe failures and even lead to breakage of the device, consists of the appearance of bad contacts in the load connections where the different drives are powered.
6. Overload/underload in the board drive: This fault occurs when the current to be supplied to the actuators is not as expected, which may be due to the interference of some type of external factor on the drive itself, which requires a higher power demand.



1.2 Please position the multimeter poles as follows

The nominal voltage should be 220V

(**a**) (**b**)

**Figure 10.** (**a**) Use-Case setup. (**b**) GUI Example.

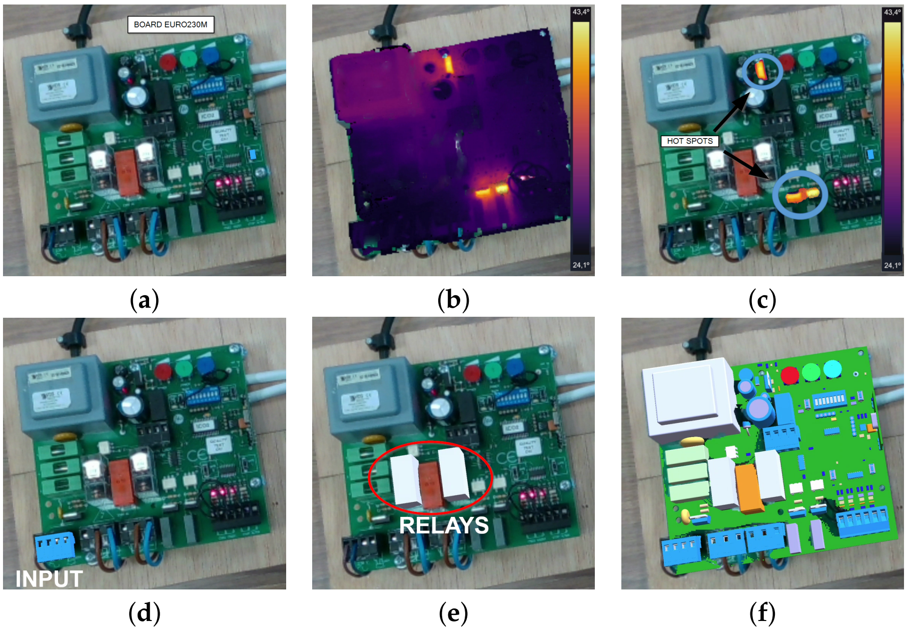
The task to be carried out by the users was to correctly diagnose the electronic board state and, in case of failure, to repair it. To perform this task, the participants were divided into three groups, depending on the system used to address this task:

1. Manual (*M*): This is the standard procedure, in which they have a manual, the equipment specification sheets, and an electrical diagram.
2. MANTRA with AR (*AR*): They used the MANTRA system, but without the thermal camera.
3. MANTRA with AR and IRT (*AR+IRT*): They used the full MANTRA system.

The objective of this use-case was to check whether the AR system (with or without IRT) improved the usual procedure carried out for the maintenance and repair of electrical circuits. To do so, the diagnosis time and repair time were analyzed for each of the users, usability questionnaires were carried out with SUS Test [62], and cognitive load was assessed with NASA-TLX test [63]. The NASA TLX questionnaire is a multidimensional assessment test that collects the cost of cognitive load that users incur when using a system. It uses six factors that are relevant in the subjective experience of workload, using a post-hoc Likert scale questionnaire. The usability of the system was also studied by performing the Spanish SUS version [62] test, as all the subjects were Spanish. This test contains 10 questions that collect the opinions of users regarding the usability of the system.

In the current use-case, a review task was performed to diagnose the failure and three repair tasks to fix each of the possible failures. Examples of different interface screens are shown in Figure 10b. The workflow of this interface can be summarized as first selecting the user profile and the task to perform; then, subtasks are shown sequentially (in order of execution). For each of these subtasks, there is a graphical instruction display mode, using animations and 3D models; a second AR mode, where 3D models and instructions superimposed on the real model are shown; and a third mode that allows the user to visualize the temperature mixed with the real image for ease of interpretation. In addition, when using the IRT display mode, it is possible to show only the temperature of the component to be analyzed, using the known electronic board pose.

Figure 11 shows several working examples of the revision task applied to the board, showing the location of various components, such as relays, power connections, or hot spots to be checked.



(**a**) (**b**) (**c**)

(**d**) (**e**) (**f**)

**Figure 11.** Examples of the AR and IRT display modes for the electronic board: (**a**) RGB view; (**b**) IRT view; (**c**) IRT view—hot spots; (**d**) AR view—Input detection; (**e**) AR view—Relays detection; and (**f**) AR view—Board tracking.

### 4.3.1. Subjects

A total of 31 people participated in the MANTRA system study. Of these, 24 were men and 7 were women with a mean age of 35 ± 15 years. In the Figure 12 can be seen two users testing the MANTRA system. All participants signed an informed consent before conducting the study. The sample population was studied, taking into account their experience with AR (Figure 13) and their experience with repairing electronic equipment such as that contemplated in this use-case. Specifically, 39% had extensive experience repairing electronic equipment (more than 10 times) and 35% had never carried out a repair of this type. Of these users, 83.9% had a higher education level and 9.7% had a doctorate.

The population was divided into two groups of 10 people and a group of 11 people. These groups were divided in a way such that there was an even distribution of electronic board repair experts and people with more experience using AR-based tools, to avoid deviations caused by previous experience.

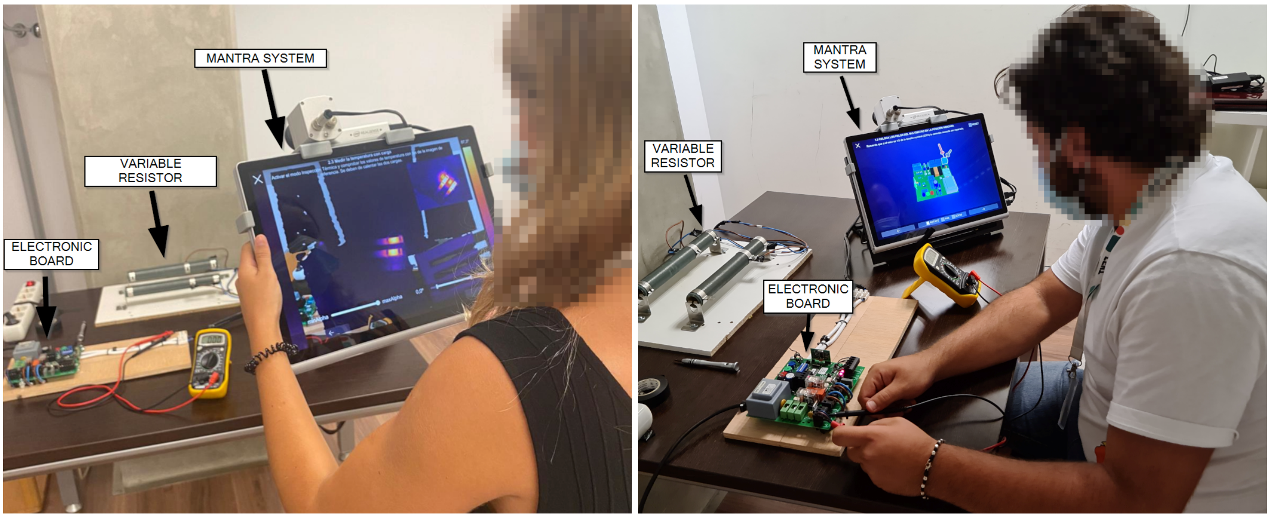


**Figure 12.** Users testing the MANTRA system.

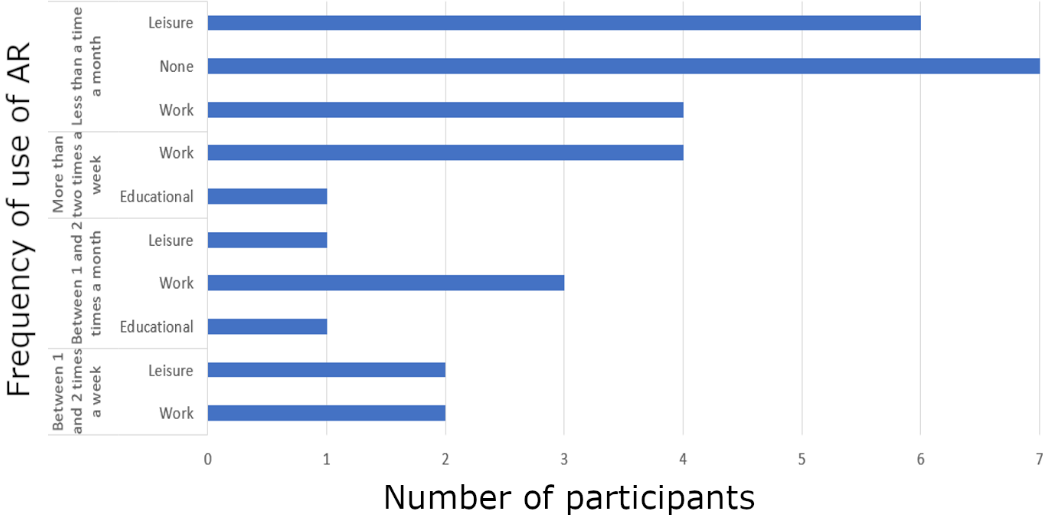


**Figure 13.** User experience with AR.

### 4.3.2. Results

For the MANTRA system analysis use-case, both the subjective appreciation of the participants (through questionnaires) and the objective measurement of efficiency were studied by measuring times and errors committed. Table 6 shows the efficiency results, in terms of time and the number of errors made. As can be seen, there was a significant mean reduction (11%) in time when using AR in the diagnosis phase, a value that improved to 17% when using IRT. This was due to the fact that having IRT images allowed users to carry out some of the diagnostic steps in a faster way by being able to check the operating temperatures of the different elements without having to measure the voltage and/or current with a multimeter. This reason was also reflected in a reduction in errors made. We understand errors as misconfiguration of the multimeter, measuring voltages or currents at the wrong points, getting confused about the tool to use, misdiagnosing the simulated fault, or performing an incorrect repair. IRT made it possible to greatly reduce the use of a multimeter and, therefore, the number of errors made. In the case of repair, detailed and sequential guidance with AR and/or IRT also reduced the time spent, especially among people with little knowledge of electronics. After the repair task, a board functional check was carried out which, in the case of using IRT, was simplified to verify that the temperature image coincides with its normal operation.

**Table 6.** Efficiency results of the different industrial maintenance systems.

| | Metric | Diagnosis Task | | | Repair Task | | |
|---|---|---|---|---|---|---|---|
| | | M | AR | AR + IRT | M | AR | AR + IRT |
| Time (Minutes) | Mean value | 6.74 | 6.00 | 5.60 | 5.01 | 4.07 | 3.91 |
| | Standard deviation | 5.45 | 3.21 | 2.35 | 2.58 | 1.34 | 2.00 |
| | Reduction (%) | | −10.98 | −16.88 | | −18.81 | −21.84 |
| Errors (Quantity) | Mean value | 3.60 | 1.33 | 1.00 | 3.1 | 1.4 | 0.8 |
| | Standard deviation | 2.27 | | | | | |
| | Reduction (%) | | −62.96 | −72.22 | | −54.84 | −74.19 |

Regarding subjective measures, Figure 14 shows the accumulated results of both the NASA TLX test and the SUS test. In the SUS test, depending on the score obtained, a system can be classified as usable (from 68), good (around 70), or excellent (from 85), with 100 being the highest possible score. According to this classification, the manual system was usable, while the systems with AR and IRT were good. Moreover, with the addition of the IRT, this score increased. This may be due to the fact that, thanks to the use of IRT, the user had to physically perform fewer operations with the board. In the case of the NASA test (Figure 14b), the system with IRT was also the best-rated, followed by the AR system and, finally, the manual system.

Figure 15 shows the NASA test results as box plots for each of the aspects that imply cognitive load for an operator. Several conclusions can be deducted from these graphs: The first is that there were no significant differences in physical demand as, in this use-case, the physical work was minimal. The second is that when using the manual system, there was a greater variance in all aspects, this may have been due to the fact that the population with great technical experience required little effort, due their familiarity with working with electrical diagrams and data sheets, while the participants with little or no experience had to exert much more significant effort. This was also reflected in the mental demand graph. When using the MANTRA system, the variance was reduced and, in addition, better mean values were obtained in all aspects; being especially significant in the graphs of frustration, effort, and mental demand. Finally, from the performance graph, it can be seen that subjectively better results are obtained with AR + TE, then with AR; these results were coherent with those obtained quantitatively, in terms of the least number of errors and the least time spent.

In reference [64], an automatic detection system of faults in electric impact drills is proposed, using a thermographic measurement classification system based on machine learning techniques (Nearest neighbors and backpropagation neural network). The results and conclusions reached in [64] can complement the current study, so future actions will try to use similar techniques to automate the detection of faults from thermographic measurements and take advantage of the potential of AR to guide the maintenance and repair tasks.

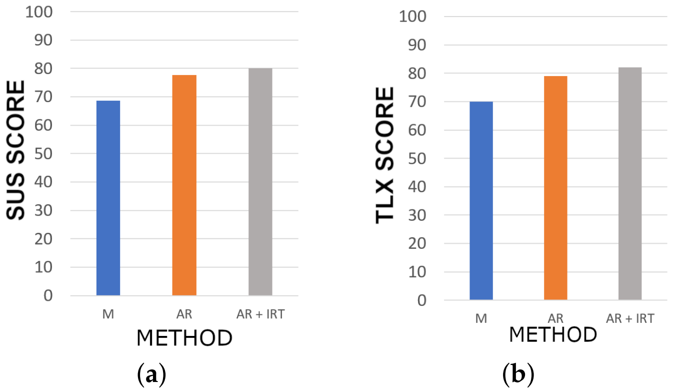


**Figure 14.** Questionnaires results carried out after the maintenance task: (**a**) SUS results; and (**b**) NASA results.

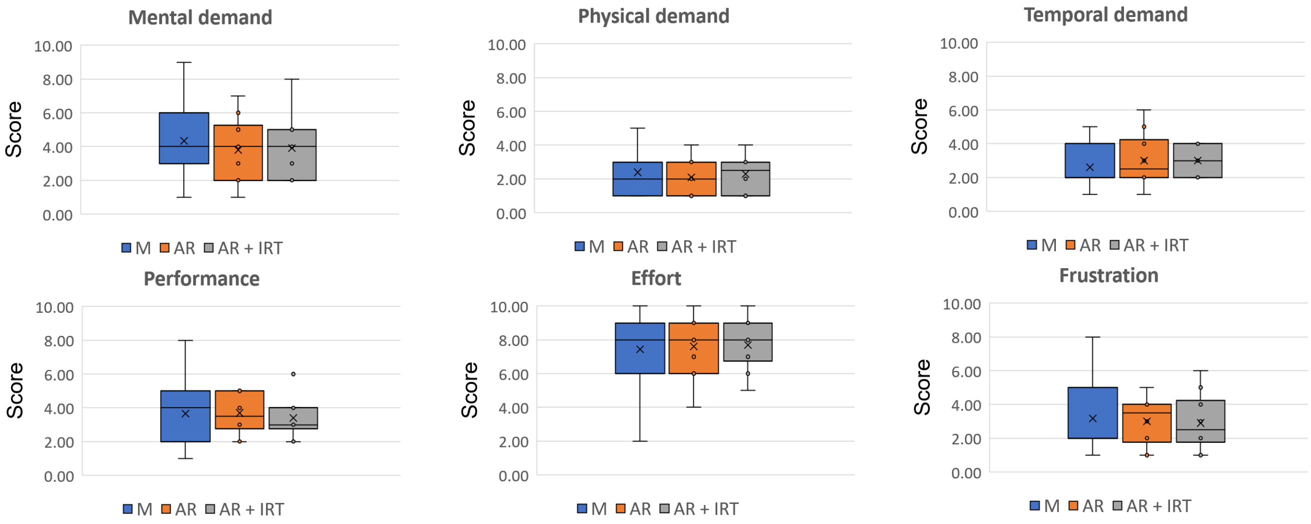


**Figure 15.** NASA Results.

## 5. Conclusions

MANTRA is a novel AR system that successfully combines IRT and AR technologies for use in the industrial maintenance sector. This system was both tested quantitatively and qualitatively validated, showing excellent results due the new combination of methods for detection and pose estimation. Specifically, MANTRA can automatically align virtual information and temperature on any 3D object, in real time. MANTRA was designed as a modular system, so the Hardware and Software components can be replaced or upgraded (e.g., using another RGB-D camera or object detector method). It was also demonstrated the benefits of adding the IRT technology to an AR system to increase the effectiveness in maintenance operations. In addition, the IRT - RGB-D registration procedure explained with our new calibration pattern would be very helpful for popularizing the IRT and AR combination. Moreover, MANTRA has a lot of potential in other sectors, such as manufacturing or quality inspection.

As a future work, we plan to replace the RGB-D camera by a RGB camera and replace the visualization display with a head-mounted display, such as the Hololens 2. Another area for improvement is to replace LINEMOD method with future object detection and pose estimation method based on deep-learning that would surpass it. It is also planned to perform a field-scale experiment with another industrial maintenance use-case such as water pump repair operations.

**Author Contributions:** Conceptualization, M.O. and E.I.; methodology, M.O. and E.I.; software, M.O. and E.I.; validation, P.V., A.J., and M.O.; formal analysis, M.O. and E.I.; investigation, M.O., E.I., and P.V.; resources, M.A.; data curation, M.O. and E.I.; writing—original draft preparation, M.O., E.I., and A.J.; writing—review and editing, M.O., E.I., and P.V.; visualization, M.A.; supervision, J.M.; project administration, J.M.; funding acquisition, M.A. All authors have read and agreed to the published version of the manuscript.

**Funding:** The MANTRA project was funded by the Spanish Ministry of Economy and Competitiveness, through its program Retos-Colaboración with grant number RTC-2017-6312-7. In addition, GPU servers for deep-learning funded by the European Union through the ERDF (European Regional Development Fund) program of the Valencian Community 2014-2020 (IDIFEDER/2018/029) were employed for training some of the algorithms applied in this work.

**Institutional Review Board Statement:** The study was conducted according to the guidelines of the Declaration of Helsinki, and approved by the Ethics Committee of Universitat Politècnica de València (P2 _ 11 _ 01_ 18, 11 of January 2018).

**Informed Consent Statement:** Written informed consent has been obtained from the user subjects to publish this paper.

**Data Availability Statement:** LINEMOD and LINEMOD-OCCLUDED dataset can be obtained at https://bop.felk.cvut.cz/datasets/.

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

## Abbreviations

The following abbreviations are used in this manuscript:

| | |
|---|---|
| AR | Augmented Reality |
| IRT | Infrared Radiation Thermography |
| 6DOF | 6 Degree of Freedom |
| LM | LINEMOD dataset |
| LM-O | LINEMOD-OCCLUDED dataset |

## References

1. Lamberti, F.; Manuri, F.; Sanna, A.; Paravati, G.; Pezzolla, P.; Montuschi, P. Challenges, opportunities, and future trends of emerging techniques for augmented reality-based maintenance. *IEEE Trans. Emerg. Top. Comput.* **2014**, *2*, 411–421. [CrossRef]
2. Palmarini, R.; Erkoyuncu, J.A.; Roy, R.; Torabmostaedi, H. A systematic review of augmented reality applications in maintenance. *Robot. Comput.-Integr. Manuf.* **2018**, *49*, 215–228. [CrossRef]
3. Lim, G.M.; Bae, D.M.; Kim, J.H. Fault diagnosis of rotating machine by thermography method on support vector machine. *J. Mech. Sci. Technol.* **2014**, *28*, 2947–2952. [CrossRef]
4. Bagavathiappan, S.; Lahiri, B.; Saravanan, T.; Philip, J.; Jayakumar, T. Infrared thermography for condition monitoring—A review. *Infrared Phys. Technol.* **2013**, *60*, 35–55. [CrossRef]
5. Jadin, M.S.; Taib, S. Recent progress in diagnosing the reliability of electrical equipment by using infrared thermography. *Infrared Phys. Technol.* **2012**, *55*, 236–245. [CrossRef]
6. You, M.Y.; Liu, F.; Wang, W.; Meng, G. Statistically planned and individually improved predictive maintenance management for continuously monitored degrading systems. *IEEE Trans. Reliab.* **2010**, *59*, 744–753. [CrossRef]
7. Zubizarreta, J.; Aguinaga, I.; Amundarain, A. A framework for augmented reality guidance in industry. *Int. J. Adv. Manuf. Technol.* **2019**, *102*, 4095–4108. [CrossRef]
8. Maldague, X. Theory and practice of infrared technology for nondestructive testing. In *Wiley Series in Microwave and Optical Engineering*; Wiley: New York, NY, TX, USA, 2001.
9. Rytov, S.M. *Theory of Electric Fluctuations and Thermal Radiation*; Technical Report; Air force Cambridge Research Lab Hanscom: Arlington, USA, 1959.
10. Diakides, M.; Bronzino, J.D.; Peterson, D.R. *Medical Infrared Imaging: Principles and Practices*; CRC Press: Boca Raton, FL, USA, 2012.
11. Balaras, C.A.; Argiriou, A. Infrared thermography for building diagnostics. *Energy Build.* **2002**, *34*, 171–183. [CrossRef]
12. Fukuda, T.; Yokoi, K.; Yabuki, N.; Motamedi, A. An indoor thermal environment design system for renovation using augmented reality. *J. Comput. Des. Eng.* **2019**, *6*, 179–188. [CrossRef]
13. Kurz, D. Thermal touch: Thermography-enabled everywhere touch interfaces for mobile augmented reality applications. In Proceedings of the 2014 IEEE International Symposium on Mixed and Augmented Reality (ISMAR), Munich, Germany, 10–12 September 2014; pp. 9–16.

14. Cifuentes, I.J.; Dagnino, B.L.; Salisbury, M.C.; Perez, M.E.; Ortega, C.; Maldonado, D. Augmented reality and dynamic infrared thermography for perforator mapping in the anterolateral thigh. *Arch. Plast. Surg.* **2018**, *45*, 284. [CrossRef] [PubMed]
15. Hinterstoisser, S.; Lepetit, V.; Ilic, S.; Holzer, S.; Bradski, G.; Konolige, K.; Navab, N. Model based training, detection and pose estimation of texture-less 3D objects in heavily cluttered scenes. In Proceedings of the Asian Conference on Computer Vision, Daejeon, Korea, 5–9 November 2012; pp. 548–562.
16. Alexey, A. YOLOv4—Neural Networks for Object Detection (Windows and Linux Version of Darknet). 2020. Available online: https://github.com/AlexeyAB/darknet (accessed on 3 November 2020).
17. Tjaden, H.; Schwanecke, U.; Schomer, E. Real-time monocular pose estimation of 3D objects using temporally consistent local color histograms. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 124–132.
18. Denninger, M.; Sundermeyer, M.; Winkelbauer, D.; Zidan, Y.; Olefir, D.; Elbadrawy, M.; Lodhi, A.; Katam, H. BlenderProc. *arXiv* **2019**, arXiv:1911.01911.
19. Masood, T.; Egger, J. Augmented reality in support of Industry 4.0—Implementation challenges and success factors. *Robot. Comput.-Integr. Manuf.* **2019**, *58*, 181–195. [CrossRef]
20. Fiorentino, M.; Uva, A.E.; Gattullo, M.; Debernardis, S.; Monno, G. Augmented reality on large screen for interactive maintenance instructions. *Comput. Ind.* **2014**, *65*, 270–278. [CrossRef]
21. Ceruti, A.; Marzocca, P.; Liverani, A.; Bil, C. Maintenance in aeronautics in an Industry 4.0 context: The role of Augmented Reality and Additive Manufacturing. *J. Comput. Des. Eng.* **2019**, *6*, 516–526. [CrossRef]
22. Webel, S.; Bockholt, U.; Engelke, T.; Gavish, N.; Olbrich, M.; Preusche, C. An augmented reality training platform for assembly and maintenance skills. *Robot. Auton. Syst.* **2013**, *61*, 398–403. [CrossRef]
23. Castellanos, M.J.; Navarro-Newball, A.A. Prototyping an Augmented Reality Maintenance and Repairing System for a Deep Well Vertical Turbine Pump. In Proceedings of the 2019 International Conference on Electronics, Communications and Computers (CONIELECOMP), Cholula, Mexico, 27 February–1 March 2019; pp. 36–40.
24. Alvarez, H.; Aguinaga, I.; Borro, D. Providing guidance for maintenance operations using automatic markerless augmented reality system. In Proceedings of the 2011 10th IEEE International Symposium on Mixed and Augmented Reality, Basel, Switzerland, 26–29 October 2011; pp. 181–190.
25. Klein, G.; Murray, D. Parallel tracking and mapping for small AR workspaces. In Proceedings of the 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality, Nara, Japan, 13–16 November 2007; pp. 225–234.
26. Lowe, D.G. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.* **2004**, *60*, 91–110. [CrossRef]
27. Drost, B.; Ulrich, M.; Navab, N.; Ilic, S. Model globally, match locally: Efficient and robust 3D object recognition. In Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 13–18 June 2010; pp. 998–1005.
28. Drummond, T.; Cipolla, R. Real-time visual tracking of complex structures. *IEEE Trans. Pattern Anal. Mach. Intell.* **2002**, *24*, 932–946. [CrossRef]
29. Choi, C.; Christensen, H.I. Real-time 3D model-based tracking using edge and keypoint features for robotic manipulation. In Proceedings of the 2010 IEEE International Conference on Robotics and Automation, Anchorage, AK, USA, 3–8 May 2010; pp. 4048–4055.
30. Prisacariu, V.A.; Reid, I.D. PWP3D: Real-time segmentation and tracking of 3D objects. *Int. J. Comput. Vis.* **2012**, *98*, 335–354. [CrossRef]
31. Rad, M.; Lepetit, V. BB8: A scalable, accurate, robust to partial occlusion method for predicting the 3D poses of challenging objects without using depth. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 3828–3836.
32. Kehl, W.; Manhardt, F.; Tombari, F.; Ilic, S.; Navab, N. Ssd-6D: Making RGB-based 3D detection and 6D pose estimation great again. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 1521–1529.
33. Zakharov, S.; Shugurov, I.; Ilic, S. Dpod: 6D pose object detector and refiner. In Proceedings of the IEEE International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019; pp. 1941–1950.
34. Su, Y.; Rambach, J.; Minaskan, N.; Lesur, P.; Pagani, A.; Stricker, D. Deep Multi-state Object Pose Estimation for Augmented Reality Assembly. In Proceedings of the 2019 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct), Beijing, China, 14–18 October 2019; pp. 222–227.
35. Ivorra, E.; Ortega, M.; Catalán, J.; Ezquerro, S.; Lledó, L.; Garcia-Aracil, N.; Alcañiz, M. Intelligent Multimodal Framework for Human Assistive Robotics Based on Computer Vision Algorithms. *Sensors* **2018**, *18*, 2408. [CrossRef]
36. Hodan, T.; Haluza, P.; Obdržálek, Š.; Matas, J.; Lourakis, M.; Zabulis, X. T-LESS: An RGB-D dataset for 6D pose estimation of texture-less objects. In Proceedings of the 2017 IEEE Winter Conference on Applications of Computer Vision (WACV), Santa Rosa, CA, USA, 24–31 March 2017; pp. 880–888.
37. Du, B.; He, Y.; He, Y.; Zhang, C. Progress and trends in fault diagnosis for renewable and sustainable energy system based on infrared thermography: A review. *Infrared Phys. Technol.* **2020**, *109*, 103383. [CrossRef]
38. Lopez-Perez, D.; Antonino-Daviu, J. Application of infrared thermography to failure detection in industrial induction motors: Case stories. *IEEE Trans. Ind. Appl.* **2017**, *53*, 1901–1908. [CrossRef]

39. Hakimollahi, H.; Zamani, D.; Hosseini, S.H.; Rahimi, R.; Abbasi, M. Evaluation of thermography inspections effects on costs and power losses reduction in Alborz Province Power Distribution Co. In Proceedings of the 2016 21st Conference on Electrical Power Distribution Networks Conference (EPDC), Karaj, Iran, 26–27 April 2016; pp. 222–226.
40. Leal-Meléndrez, J.A.; Altamirano-Robles, L.; Gonzalez, J.A. Occlusion handling in video-based augmented reality using the kinect sensor for indoor registration. In *Iberoamerican Congress on Pattern Recognition*; Springer: Berlin, Germany, 2013; pp. 447–454.
41. Hinterstoisser, S.; Cagniart, C.; Ilic, S.; Sturm, P.; Navab, N.; Fua, P.; Lepetit, V. Gradient response maps for real-time detection of textureless objects. *IEEE Trans. Pattern Anal. Mach. Intell.* **2011**, *34*, 876–888. [CrossRef] [PubMed]
42. Zhang, Z. Iterative point matching for registration of free-form curves and surfaces. *Int. J. Comput. Vis.* **1994**, *13*, 119–152. [CrossRef]
43. Chen, S.; Hong, J.; Liu, X.; Li, J.; Zhang, T.; Wang, D.; Guan, Y. A Framework for 3D Object Detection and Pose Estimation in Unstructured Environment Using Single Shot Detector and Refined LineMOD Template Matching. In Proceedings of the 2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), Zaragoza, Spain, 10–13 September 2019; pp. 499–504.
44. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. Ssd: Single shot multibox detector. In *European Conference on Computer Vision*; Springer: Berlin, Germany, 2016; pp. 21–37.
45. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv* **2020**, arXiv:2004.10934.
46. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*; Springer: Berlin, Germany, 2014; pp. 740–755.
47. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
48. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal loss for dense object detection. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2980–2988.
49. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *39*, 1137–1149. [CrossRef]
50. Dwibedi, D.; Misra, I.; Hebert, M. Cut, Paste and Learn: Surprisingly Easy Synthesis for Instance Detection. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 1310–1319.
51. Tremblay, J.; Prakash, A.; Acuna, D.; Brophy, M.; Jampani, V.; Anil, C.; To, T.; Cameracci, E.; Boochoon, S.; Birchfield, S. Training deep networks with synthetic data: Bridging the reality gap by domain randomization. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Salt Lake City, UT, USA, 18–22 June 2018; pp. 969–977.
52. Hodaň, T.; Vineet, V.; Gal, R.; Shalev, E.; Hanzelka, J.; Connell, T.; Urbina, P.; Sinha, S.N.; Guenter, B. Photorealistic Image Synthesis for Object Instance Detection. In Proceedings of the 2019 IEEE International Conference on Image Processing (ICIP), Taipei, Taiwan, 22–25 September 2019; pp. 66–70.
53. Dai, J.S. Euler–Rodrigues formula variations, quaternion conjugation and intrinsic connections. *Mech. Mach. Theory* **2015**, *92*, 144–152. [CrossRef]
54. Brown, D.C. Decentering distortion of lenses. *Photogramm. Eng. Remote. Sens.* **1966**,*32*, 444–462.
55. Zhang, Z. A flexible new technique for camera calibration. *IEEE Trans. Pattern Anal. Mach. Intell.* **2000**, *22*, 1330–1334. [CrossRef]
56. Fischler, M.; Bolles, R. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM* **1981**, *24*, 381–395. [CrossRef]
57. Rangel, J.; Soldan, S.; Kroll, A. 3D thermal imaging: Fusion of thermography and depth cameras. In Proceedings of the International Conference on Quantitative InfraRed Thermography, Bordeaux, France, 7–11 July 2014; Volume 3.
58. Vidas, S.; Lakemond, R.; Denman, S.; Fookes, C.; Sridharan, S.; Wark, T. A mask-based approach for the geometric calibration of thermal-infrared cameras. *IEEE Trans. Instrum. Meas.* **2012**, *61*, 1625–1635. [CrossRef]
59. Brachmann, E.; Krull, A.; Michel, F.; Gumhold, S.; Shotton, J.; Rother, C. Learning 6D object pose estimation using 3D object coordinates. In *European Conference on Computer Vision*; Springer: Berlin, Germany, 2014; pp. 536–551.
60. Brachmann, E.; Michel, F.; Krull, A.; Ying Yang, M.; Gumhold, S. Uncertainty-driven 6D pose estimation of objects and scenes from a single rgb image. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 3364–3372.
61. Song, C.; Song, J.; Huang, Q. Hybridpose: 6D object pose estimation under hybrid representations. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 431–440.
62. Aguilar, M.I.H.; Villegas, A.A.G. Análisis comparativo de la Escala de Usabilidad del Sistema (EUS) en dos versiones/Comparative analysis of the System Usability Scale (SUS) in two versions. *RECI Rev. Iberoam. Las Cienc. Comput. Inform.* **2016**, *5*, 44–58.
63. Martinetti, A.; Rajabalinejad, M.; Van Dongen, L. Shaping the future maintenance operations: Reflections on the adoptions of Augmented Reality through problems and opportunities. *Procedia CIRP* **2017**, *59*, 14–17. [CrossRef]
64. Glowacz, A. Fault diagnosis of electric impact drills using thermal imaging. *Measurement* **2021**, *171*, 108815. [CrossRef]