Digital Assistance for Quality Assurance: Augmenting Workspaces using Deep Learning for Tracking Near-Symmetrical Objects

João Belo, Andreas Fender, Tiare Feuchtner, Kaj Grønbæk

Aarhus University, Denmark

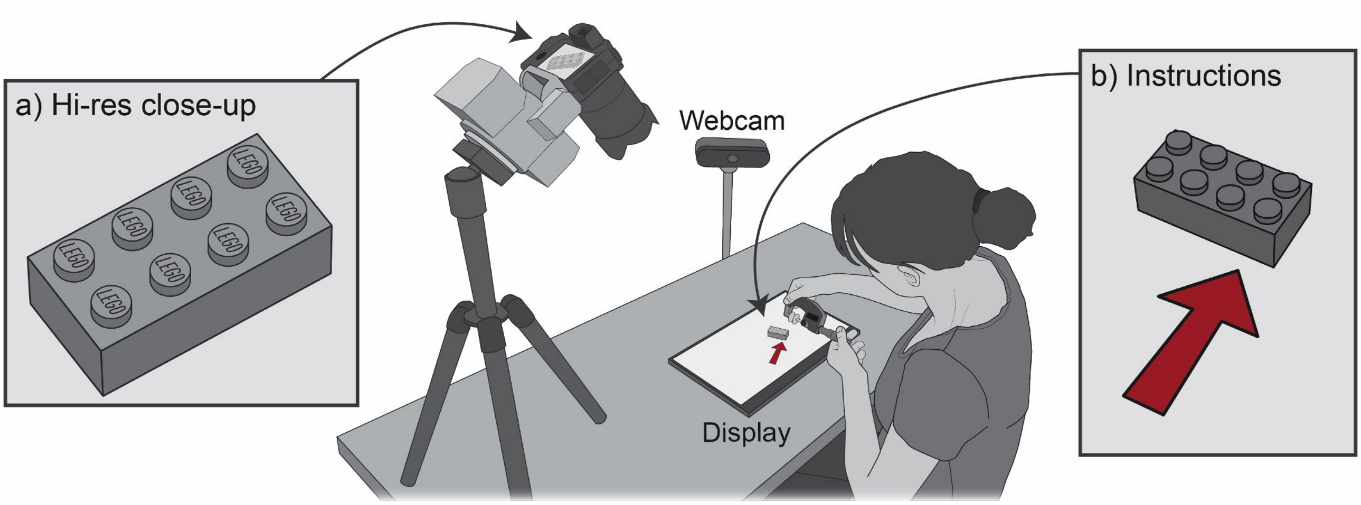


Figure 1. In the explored use-case, a worker needs to measure exact distances between different pre-defined points on a near-symmetrical LEGO brick. We present digital assistance for this metrology task by displaying situated step-by-step measurement guides on a tabletop-display. (a) While webcams locate the brick, a zoomed-in camera on a pan-tilt unit rotates towards the brick to identify its unique orientation based on fine-grained features (a LEGO logo in this case). (b) Based on the tracked unique orientation, situated guides can indicate the correct points to measure.

ABSTRACT

We present a digital assistance approach for applied metrology on near-symmetrical objects. In manufacturing, systematically measuring products for quality assurance is often a manual task, where a main challenge for the workers lies in accurately identifying positions to measure and correctly documenting these measurements. This paper focuses on a use-case, which involves metrology of small near-symmetrical objects, such as LEGO bricks. We aim to support this task through situated visual measurement guides. Aligning these guides poses a major challenge, since fine grained details, such as embossed logos, serve as the only feature by which to retrieve an object's unique orientation. We present a two-step approach, which consists of (1) locating and orienting the object based on its shape, and then (2) disambiguating the object's rotational symmetry based on small visual features. We apply and compare different deep learning approaches and discuss our guidance system in the context of our use case.

ISS '19, November 10–13, 2019, Daejeon, Republic of Korea.
Copyright © 2019 Association of Computing Machinery.
ACM ISBN 978-1-4503-6891-9/19/11 ...$15.00.
http://dx.doi.org/10.1145/3343055.3359699

Author Keywords

Metrology; Industry 4.0; Computer Vision; Augmented Reality; Focus+Context Tracking; Fine-grained Feature Recognition; Situated Instructions

CCS Concepts

•Human-centered computing → Interactive systems and tools; •Computing methodologies → Computer vision problems;

INTRODUCTION

In manufacturing, metrology is the activity of measuring objects as a part of standard Quality Assurance (QA) procedures. Nowadays, even though industrial metrology tasks are increasingly automated, many still require manual work. While robots can support the overall procedure, e.g., by presorting the objects, many of the actual measurements are conducted by workers, as was observed in real-world use cases in companies associated with the Manufacturing Academy of Denmark (MADE)[1]. The current procedure in these companies involves following electronic instruction manuals that are viewed on a desktop screen. Some measurement tools provide the capability of digitally transmitting data, whereas others require manual input of measurements to a computer

[1] Manufacturing Academy of Denmark: https://www.made.dk/

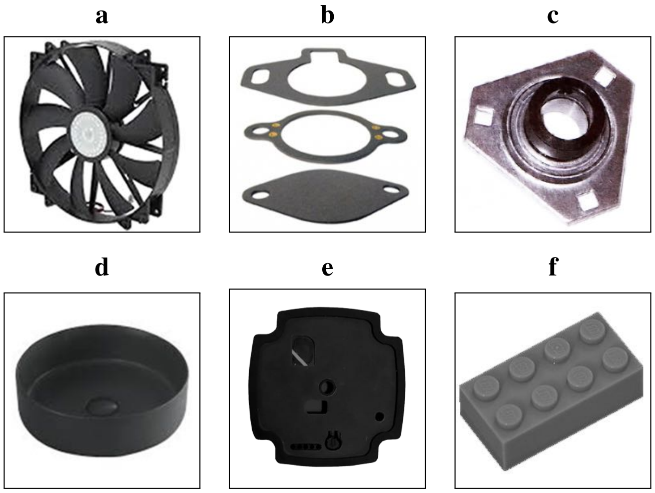


Figure 2. Examples of near-symmetrical objects, including components of fans (a), thermostats (b), pumps (c,e), plumbing (d), and a LEGO brick (f). The shape of each object has different degrees of rotational symmetry. Only a couple small visual features on each object allow to determine its unique orientation.

database. Conventional systems are not aware of what instruction is being followed, hence even when the measurement tool can send the value to the system, the worker must still specify what measurement position the value corresponds to. In other words, the worker must associate the schematic drawing in the instruction manual to the object being measured and determine the respective mental rotation to know which positions to measure. In particular, near-symmetrical objects pose challenges, since it is difficult to identify the measurement points quickly and accurately with the human eye. In this context, "near-symmetrical" means, that the overall shape of the object has rotational symmetry. Its unique orientation can be identified only by small visual features: either by their locations on the object, or by determining the orientation of a non-symmetrical feature. We have come across a multitude of such near-symmetrical objects in industrial manufacturing (for examples see Figure 2). Each of these has at least one visual feature that, through careful inspection, permits identification of the object's unique orientation. Such features may be a single adjustment screw on one side of a shaft, a notch or pin that prevents wrong insertion of a component, a serial number, etc. In the example of a 2x4 LEGO brick (Figure 2, f), the symmetry-breaking feature is the LEGO logo (Figure 1, a).

To better facilitate metrology of such small and near-symmetrical objects, we propose the digital guidance system shown in Figure 1, which dynamically provides step-by-step instructions for a given metrology task. We further propose to provide these measurement instructions as visual guides that are situated in close proximity to the measured object. This aims to prevent the repeated attention-switches between object and desktop display, which can increase the worker's time and energy demand [37, 17]. Furthermore, by aligning the guides with the object's current orientation, we strive to reduce workers' cognitive load, decreasing the need of mental rotations [6, 38].

In this paper we discuss the visualization of measurement guides situated in the task space and present a tracking technique for near-symmetrical objects that need to be measured. Our approach supports automatic detection of small symmetry-breaking features through computer vision and deep learning, which allows us to identify a object's unique orientation. We devised a two-step tracking solution that computes the position of the near-symmetrical object in the whole tracking area (context), and resolves the ambiguity of its rotation (focus). We refer to this as *Focus+Context tracking*, analogous to Focus+Context output [3].

We aim to support workers performing a metrology task by:

1. Providing assistance in disambiguating the object's orientation, which is challenging due to its near-symmetrical characteristics.

2. Displaying situated measurement guides superimposed on or in close proximity to the object, to reduce the frequency of switching between *information* and *workpiece* tasks [23].

3. Presenting the measurement guides corresponding to the object's current orientation, to reduce the cognitive load of applying mental transformations [48].

In the following sections, we elaborate on our Focus+Context pipeline and its application to the described real-world use case. We first present a pipeline for tracking a LEGO brick on a horizontal display to render co-located instructions. Thereafter, we present a generalized variation of the pipeline for tracking a handheld brick and discuss the deep learning techniques that both pipelines utilize. We test a number of hypotheses about training procedure refinements for orientation disambiguation, through ablation studies. Finally, we discuss the generalizability of our approach to objects with different shapes, sizes, and visual features, and present the tracking results for the pump component *e* in Figure 2 as an additional example.

RELATED WORK

The term "Focus+Context Tracking" used in this paper, is inspired by the work of Baudisch et al. [3], where a screen consists of low-resolution regions providing context and high-resolution regions for focus information. Focus+Context tracking can be seen as a metaphor of the same concept, applied to input devices used for tracking, instead of output devices.

The general approach of using multiple cameras to capture different levels of detail has already been investigated [1]. We follow an approach similar to the one used for marker tracking by Rekimoto et al. [34], where a fixed camera is responsible for tracking an entire tabletop surface and a high-resolution pan-tilt camera performs marker recognition. However, while markers are optimized for tracking, we tackle the more challenging problem of estimating the orientation of a marker-less, near-symmetrical object. In other words, we detect the position and orientation of an object based on its shape and small visual features in an image. Previous work has investigated detecting the 2D orientation of a texture, or parts of a texture, e.g., based on gradient vectors [4], or principal directions [16]. These techniques work for 2D rotations in image space, which implies that their applicability is limited, when trying to detect the orientation of a texture seen from an oblique angle. Furthermore, in our case the object features a slightly reflective material that causes view-dependent highlights in the image.

In the scenario presented in this paper, estimating a 2D rotation of the visual feature in image space is not sufficient. We therefore devised a solution with deep learning-based vision techniques.

Computer Vision and Deep Learning

To detect and identify the object's orientation, we apply deep learning in our tracking pipeline. In this regard, the work of Krizhevsky et al. [19] has led to significant breakthroughs in image recognition using Convolutional Neural Networks (CNNs). Since then, CNNs have proven to be highly successful in other image recognition tasks, such as object detection [32, 35], instance segmentation [11], and pose estimation [44]. The accuracy and efficiency of CNNs have increased substantially over the years, due in part to improvements in the architectures of these networks [12, 39]. Furthermore, techniques such as transfer learning [8, 31, 47] allow for improved generalization when the size of the dataset is small, and Kornblith et al. [18] found a strong correlation between accuracy on the ImageNet dataset [19] and transfer learning accuracy, when fine-tuning or using pre-trained networks as feature extractors.

Augmented Workspaces

Augmented environments that seamlessly combine the virtual and real world have been envisioned since the early 90's, exploring how everyday environments could be augmented to improve people's lives and the way they work [30, 45]. Since then, researchers have proposed systems like the *DigitalDesk* [46], where the user can interact with digital information that is superimposed on conventional paper. *Augmented Surfaces* [34] follows up on the idea of projecting virtual content onto a desk to augment a meeting room, allowing users to utilize their environment as an extension of their laptops and attach data to physical objects.

Even though our solution is technically not augmented reality (AR), there are many related AR systems with similar goals and characteristics [5, 24, 26, 33]. The effectiveness of AR in industry has become an active topic of research over the past few years. For example, Baird and Barfield [2] showed that workers using AR would complete assembly tasks faster and with fewer errors. A study on object assembly [42] provided additional evidence for this and demonstrated that AR can also reduce cognitive load of the worker performing the task. Henderson and Feiner [15] similarly demonstrated that AR assistance in a procedural task can increase the workers' performance and that co-located instructions lead to fewer head movements. Furthermore, they found similar benefits of using AR during maintenance tasks [14]. More recently, Uva et al. [43] conducted a study on the effectiveness of spatial augmented reality in manufacturing, providing evidence that co-located technical information greatly reduces the complexity of the tasks, improving completion times and lowering error rates, when compared to paper-based instructions. Finally, Polvi et al. [27] confirmed that an AR interface can also be beneficial in inspection tasks, resulting in lower completion times, fewer errors, fewer gaze shifts, and a lower subjective workload.

To our knowledge, there is no existing research on augmenting the workplace to specifically support metrology tasks. Assembly and maintenance tasks are related, in that most activities are performed in a predictable environment and are part of a procedural task. Furthermore, inspection tasks entail a similar step of information matching as in metrology. However, our use case of manual metrology poses the need for accurate pose estimation of near-symmetrical objects, which goes beyond related research.

USE CASE

In connection with the MADE project, we explore QA processes at multiple manufacturing companies, where workers manually conduct metrology on various near-symmetrical objects. In this paper, we focus on a single use-case of applied metrology at the LEGO Group. In the presented use-case, workers employ a range of specialized tools for measuring objects. Some of these tools are still analog and require manual input of numbers into the database. Digital measurement tools allow to directly transmit the measured values to the database. However, the worker still has to indicate which measurement step (i.e., which field in the database) the value corresponds to. Thus, to ensure correct recording of measurements, the workers currently measure certain positions, following a strict order. This order is indicated in an electronic instruction manual (i.e., pdf), which includes a schematic drawing of the object with numbered measurement positions. A computer is used to display this manual and the database with measurement entries. Mouse and keyboard serve as input devices for navigation and entry of measured values.

Since in this use-case most products are small and near-symmetrical, the worker must carefully inspect each object to correctly orient it, before being able to accurately identify the next position at which to take a measurement. Within the LEGO Group, we focus on a common near-symmetrical object that undergoes rigorous QA procedures - a 2x4 LEGO brick, which is simply referred to as *brick* in the remainder of this paper.

DIGITAL ASSISTANCE: USER INTERFACE

Our digital assistant displays situated instructions for metrologists. To ensure that the workers obtain all required information about the task at hand, the interface features an *overview panel* (see white panels in Figure 3, left). This contains textual information similar to the original instruction manual, i.e., describing the type of measurement to take and what tool to use. Furthermore, it communicates how many measurements are left in the current stage, and shows the last saved measurement. This panel further contains a schematic 3D representation of the object (e.g., the LEGO brick), which reflects the orientation of the tracked object. Measurement guides on this representation indicate which point currently needs to be measured.

The remainder of the screen surface is reserved for displaying co-located measurement guides when the object is placed on

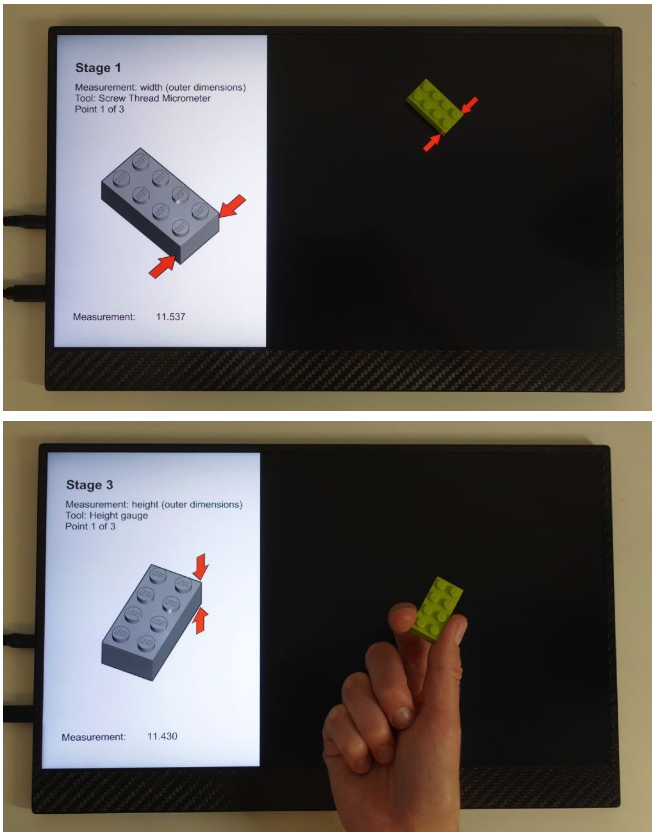

Stage 1
Measurement: width (outer dimensions)
Tool: Screw Thread Micrometer
Point 1 of 3
Measurement: 11.537
Stage 3
Measurement: height (outer dimensions)
Tool: Height gauge
Point 1 of 3
Measurement: 11.430

Figure 3. User interface of the digital assistant. The left panel shows textual instructions and an enlarged schematic representation of the object, which always reflects the orientation of the actual tracked object. Measurement guides in form of red arrows indicate the current points to measure. Top: In the right part of the screen, the guides are shown co-located with the physical object. Bottom: If the current instructions cannot be co-located or the object is handheld, the left panel still shows the instructions. In both cases, rotating the tracked object will rotate the schematic representation on the left.

the screen. In this manner they indicate measurement positions directly on the physical object (see Figure 3, top). Both, the co-located guides and the oriented schematic are only displayed when the system is certain about the actual orientation of the tracked physical object, since it is crucial for the instructions to always be displayed on the correct side.

Measurement guides consist of a pair of arrows. Whenever co-location of guides is not possible, the worker can instead refer to the guides on the schematic representation in the left part of the GUI. This occurs either when the brick is handheld, or when the current instructions would need to display arrows on top of, or underneath, the object (e.g., when measuring height). For instance, in Figure 3 (bottom), both of these conditions are met. We will elaborate on this in the *Handheld mode* section. In each step, only one pair of arrows is displayed at a time, indicating the measurement that should be taken. When measuring with an analog caliper, the worker can input the measured value with a keyboard, and hit Enter to save it. When using a digital caliper that is connected to the system, the current measurement is saved automatically upon pressing a button on the device. The system then automatically transitions to the next step showing guides for a new point to measure.

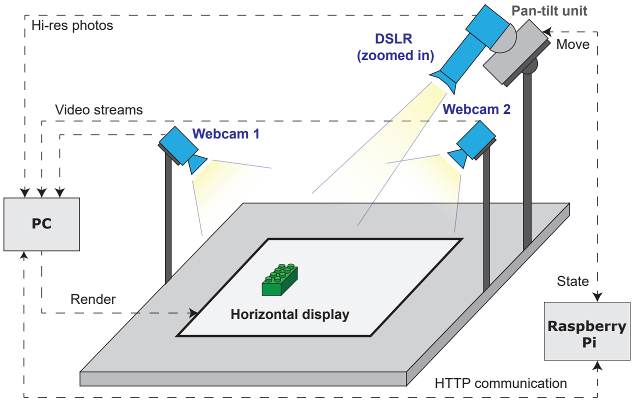


Figure 4. Overview of hardware and software components. Two webcams cover the entire tracking area. A zoomed-in DSLR camera provides high-resolution pictures of the tracked LEGO brick. The camera is mounted on a pan-tilt unit, which is controlled by a Raspberry Pi. This allows keeping the brick in focus even when it is moved. The PC controls the overall system flow and renders instructions on a horizontal display, so that they are co-located with the brick when it is placed on the screen.

ARCHITECTURE

In this section, we provide an overview of the hardware and software components of our prototype, and describe the interplay between these.

Overview of the tracking pipeline

An overview of our hardware and software components is provided in Figure 4. The video streams from two webcams are used to track the location of the brick on a tabletop screen surface. Furthermore, the brick's ambiguous orientation can be retrieved from these video streams: at this point the orientation can only be defined up to symmetry due to the 180° symmetrical shape of the brick. In a second step, the DSLR camera is oriented towards the brick's position with the help of an underlying pan-tilt unit. To do so, the main PC calculates the necessary rotation and forwards these values to a Raspberry Pi via network. This in turn controls the pan-tilt unit, to ensure continuous tracking of the brick. The overall camera setup can be described as a *master-slave configuration* [1, Ch.8.4], with two webcams as *master* and the DSLR camera as *slave*.

The DSLR camera periodically takes pictures of the brick. The zoom level and resolution of these pictures is sufficient to identify small symmetry-breaking features on the brick, such as a LEGO logo (see Figure 6). Such features allow to disambiguate the orientation of the brick. Once the 2D position and unique rotation of the object are known, measurement guides can be displayed accordingly.

The following section provides further details on the individual steps of our tracking pipeline. Additional information on the specific hardware and software components that we used may be found in the *System Implementation* section.

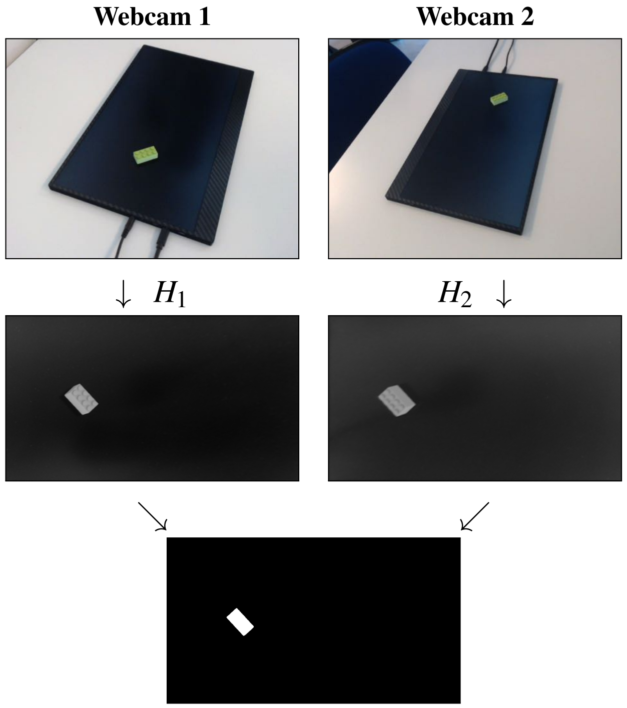


Figure 5. Our context tracking sub-pipeline is based on conventional image processing techniques. Polarizing filters make the screen contents appear black (top). Both video feeds are perspectively unwarped using homographies H_1 and H_2 (middle). This brings both feeds into the screen space of the horizontal display. The images are then thresholded and combined, to create the mask of the brick in image space (bottom). The output is the position and ambiguous orientation of the brick.

FOCUS+CONTEXT TRACKING

The idea of the pipeline shortly described above is to divide the tracking task into two separate steps: (1) The *context* step tracks the location and symmetric orientation of the brick continuously, based on a simple and fast approach using conventional computer vision techniques. (2) The *focus* step disambiguates the brick's orientation. It is triggered less frequently and is based on deep learning. This section explains each of these steps in detail.

Context tracking

To locate the brick, two context cameras stream their video feeds to the main PC (see Figure 5, top). We attached polarization filters to the cameras, so that all content on the tabletop screen appears black in the video feeds [29]. This way, co-located instructions will not interfere with the tracking. By applying (pre-calibrated) homographies to each stream, both video feeds are warped into screen space (see Figure 5, middle). The brick can then be segmented in each feed simply by binary thresholding. These thresholded images are combined with an AND-operation on each pixel. The resulting binary image is then searched for a mask that has 4 corners and a (rotated) rectangular shape, to exclude other potential objects on the screen. The center of this mask corresponds to the brick's position, and its rotation can be identified by averaging the angles of its two long edges. However, as mentioned before, the rotation is still ambiguous at this point, since the brick yields identical thresholded images when it is rotated by 180° (see "Context" image in Figure 7). Overall, this approach for context tracking requires very little computational power and can therefore output the position and orientation of the brick in real-time.

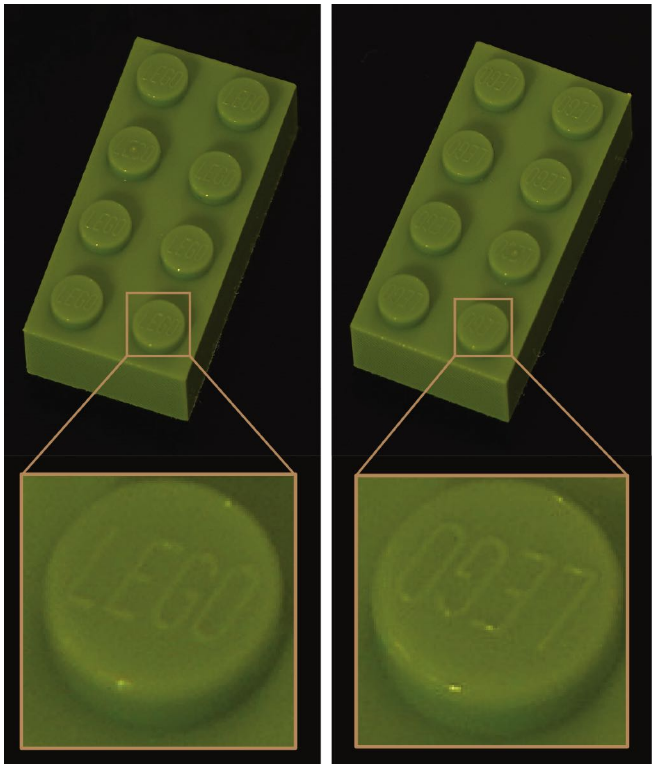


Figure 6. Raw pictures from the zoomed-in DSLR camera. For better print quality, we adjusted the aperture and increased the exposure time compared to the values we use at run-time. Furthermore, we cropped the resulting images. The left and right picture lead to the same orientation in the *context* tracking step. However, the upright LEGO logo on the left and the upside-down logo on the right allow disambiguation between both possible orientations in the *focus* step.

Focus tracking

Once the position of the brick is known, the pan-tilt unit can orient the DSLR camera towards it. This camera takes a picture every 2 seconds and transmits it to the main PC. Figure 6 shows two examples of raw images provided by the camera, of a 2x4 brick that is rotated by 180°. As with the context cameras, a polarization filter makes the screen beneath the brick appear black. The raw image is then passed to a Mask R-CNN [11] instance segmentation model. In contrast to simpler techniques such as chroma keying, this approach allows the system to effectively detect the brick even when other objects are in the picture, or when the object is partially occluded.

After performing instance segmentation, the picture is cropped and the values of all pixels outside the segmented mask are set to 0. The cropped picture is then fed into an additional CNN to disambiguate the object's orientation. This problem was solved using a 50-layer residual network architecture [12]. As shown in the "Focus" illustration in Figure 7, the classifier

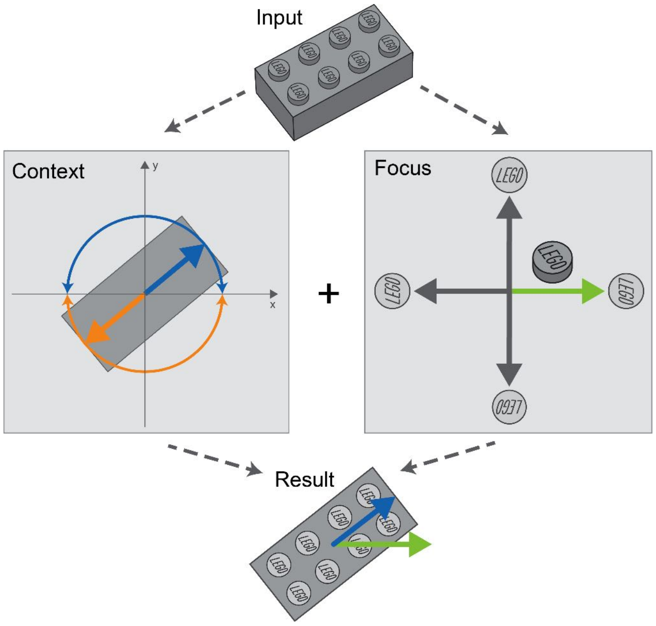


Figure 7. Combination of Focus+Context to retrieve the unique orientation of the brick. The output of the context pipeline is an ambiguous orientation: in the "Context" image the orientations marked by blue and orange arrows lead to equivalent results. The output of our focus classifier is one of four directions ("Focus" image). We then choose the direction from the *context* output that has a positive dot product with the *focus* output. In the illustrated example, the brick's orientation corresponds to the blue vector in the "Result" image.

returns one of 4 different classes, depending on the orientation of the LEGO logo: up, down, left, and right.

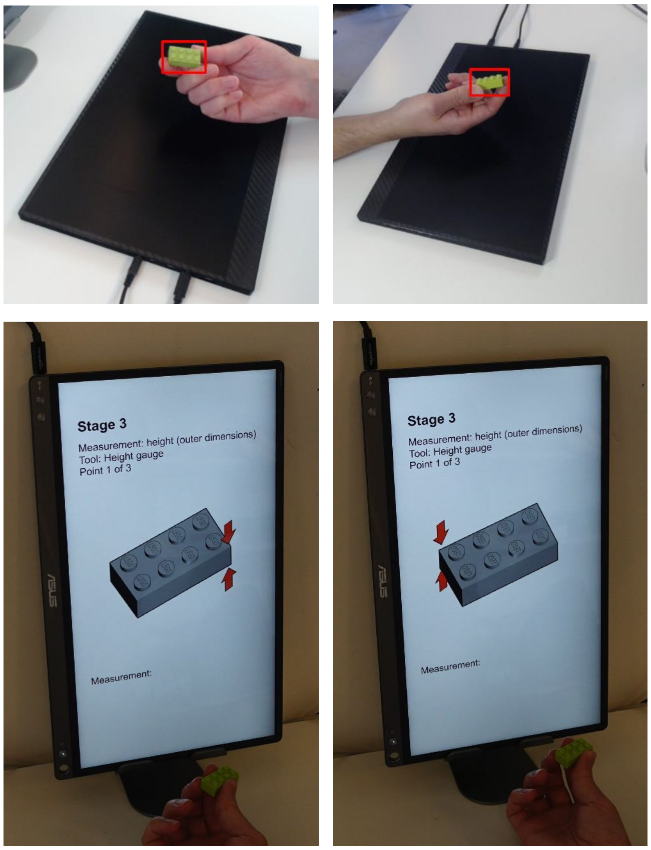
Combining focus and context

In the final step of our tracking pipeline, the outputs of *focus* and *context* are combined. While the position of the brick can be retrieved directly from the context tracking step, the orientation results from a combination of both sub-pipelines, as is illustrated in Figure 7. The output of the context tracking consists of two vectors, indicating two possible orientations (the blue and orange arrows in the "Context" image of Figure 7). The output of the focus tracking is one vector, indicating one of four main directions (see Figure 7, "Focus" image). We then form the dot product of each vector from context tracking and the single vector resulting from focus tracking, and we choose the context vector that results in a positive dot product (blue arrow in the "Result", Figure 7). Even in the rare case when the output of the context tracking is in between classes (e.g., exactly between pointing up-right and down-left) and the focus tracking is undecided between two classes, the end-result from the dot product will still be valid. We will elaborate on this in the *Discussion and future work* section.

The resulting direction vector is used to calculate the unique orientation of the brick, to properly align the measurement guides.

HANDHELD MODE

In the previous sections, we described an easy-to-replicate setup, which utilizes the polarized light from a horizontal display to segment the object from the background. This makes the context pipeline simple and computationally fast. However, an obvious limitation of this approach is that the object must always be placed on top of the display, in order to receive situated instructions. Furthermore, in a preliminary interview, workers were concerned that continuously gazing down at the tabletop throughout an entire work session might cause neck strain. With this limitation and the workers' concerns in mind, we created a variation of the pipeline that allows the object to be handheld and instructions to be displayed on a separate vertical screen.



Stage 3
Measurement: height (outer dimensions)
Tool: Height gauge
Point 1 of 3

Measurement:

Figure 8. Top: Alternative context tracking pipeline. When tracking the object based on image segmentation within each of the two webcam streams (left and right), the 3D position of the object can be estimated with the intrinsic and extrinsic parameters of the two webcams. Bottom: Alternative display setup. The system can alternatively be used with a vertical screen, or both a vertical and a horizontal screen simultaneously. In each arrangement, the displayed guides are always presented in accordance with the object's current orientation.

To achieve this, we generalized the context tracking step of the pipeline, by basing it on the Mask R-CNN instance segmentation model, instead of simple binary thresholding. This makes it possible to correctly detect the object in more challenging scenarios, e.g., when it is partially occluded due to being held by the worker, or being partly encompassed by a measuring tool. The same model, which we use for segmenting the object in the DSLR camera image, can directly be applied in real-time to the footage of the two webcams.

Figure 8 (top) shows tracking of a handheld object by combining the segmentation results from both webcams. Based on the camera positions, intrinsic parameters, and bounding boxes in each camera stream, we can estimate the position of the object. For each webcam, we create a ray from the camera's position through the center of the detected bounding box in the image plane. This creates skewed 3D rays, i.e., they are neither parallel, nor do they intersect, since the centers of the bounding boxes are rarely located at the exact same points on the brick. Based on the line equations, we then find the point with minimal distance to both rays. This gives us the object's position, which is forwarded to the pan-tilt unit controller for orientation of the DSLR camera.

The next step is to identify the ambiguous orientation of the object. In the image of the the zoomed-in DSLR camera, we approximate a polygon around the segmented object in image space and take the longest edge as orientation indicator. We then calculate the orientation in world space by making two assumptions: (1) Due to the fact that the DSLR camera is zooming in on the small object, we can assume an almost orthographic projection of the object in the segmentation. (2) With our chosen set of instruction steps, the orientation will vary only around the y-axis (up-axis). Based on these assumptions, we can simply transform the direction of the longest edge into world space, using the known extrinsic parameters of the DSLR camera. Finally, we resolve the near-symmetry as in the previously presented pipeline.

With this approach, the non-co-located instructions for a hand-held brick can be presented in correct orientation corresponding to that of the tracked object. This is illustrated in Figure 3 (bottom) and Figure 8 (bottom). Workers can switch between these modes as desired: they can trigger co-located guides by placing the brick on the horizontal screen, or they can look at the vertical screen to reduce neck strain.

SYSTEM IMPLEMENTATION

This section provides details about the frameworks, engines, and hardware that our particular implementation of the architecture is based on. While the pipeline is not bound to the specific set of software and hardware components described here, these choices were useful for an effective proof-of-concept setup.

The overall pipeline and the rendering is implemented in *Unity 3D*. The measurement guides are displayed on a 15.6" portable screen, which can be used as a horizontal tabletop display, or positioned vertically. The worker can input and save measurements in the system through a digital measuring device, such as the *Mitutoyo* micrometer (series 406), or a traditional keyboard. Pressing a button on the digital measuring device emulates keyboard inputs with the digits of the measurements followed by Enter. Alternatively, a foot pedal could be used to perform this button press.

We use the *Velt* Framework [10] to handle the data flow of the system and the communication between its various components. This node-based framework is a *Unity 3D* plugin and simplifies the creation and inspection of data flow pipelines, including pre-processing, network communication, etc. The context tracking is implemented as a specialized Velt extension, but also based on built-in nodes, e.g., nodes that wrap *OpenCV* functionalities.

We use a *Raspberry Pi Model 3 B* for receiving HTTP requests and for interfacing with a *Maxwell MPR-202* pan-tilt unit. This serves to correctly orient the attached focus camera, which is a *Sony RX10 II* DSLR camera. To trigger rotations of the DSLR camera, the Raspberry Pi controls relays, opening and closing circuits on the pan-tilt unit's DIN7 socket. Since the pan-tilt unit only supports relative movements and does not have a built-in sensor to provide its pan and tilt values, we attached an accelerometer (*MPU-9160*) to calculate its current orientation. Thus, when an absolute desired orientation is forwarded to the Raspberry Pi (based on the tracked object's position relative to the pan-tilt unit), it rotates the pan-tilt unit until the requested orientation is reached, so that the DSLR camera is oriented towards the tracked object. We take the high-resolution pictures with an ISO value of 640, an exposure time of 0.1 seconds, and an f-number of 3.2. These values only serve as an orientation, as the robustness of the pipeline does not heavily depend on the camera settings, as long as the symmetry-breaking features (e.g., LEGO logo) are visible in the picture. We then use the *Sony Imaging Edge Remote* tool [40] to automatically take pictures and periodically transmit them to our system via USB. Another specialized Velt node receives these pictures and triggers the focus part of our tracking pipeline.

All deep learning components are implemented in Python and the central pipeline communicates with these via HTTP. We use the *PyTorch* framework [28] to implement and train our models and we follow a training procedure inspired by He et al. [13]. All evaluations of our system were conducted on machines with two *Nvidia RTX2080ti* graphics cards.

TECHNICAL EVALUATION

In this section, we evaluate the accuracy of our deep learning models. To train our models we gathered two different datasets, which are described in the following subsections.

Instance Segmentation Model

We start by describing the training procedure for the Mask R-CNN model that was used for instance segmentation. For this problem we used a training dataset with 90 pictures and a validation dataset with 20 pictures, which were annotated using the VIA annotation tool [9]. Our Mask R-CNN model uses a Feature Pyramid Network [20] backbone architecture based on a 50-layer residual network [12]. We used a model that was pre-trained on the COCO dataset [21]. This model was trained over 50 epochs using stochastic gradient descent with momentum, at an initial learning rate of 0.005 divided by 10 every 13 epochs, a weight decay of 0.0005, and a batch size of 2. After training, our Mask R-CNN detector achieves a segmentation mAP of 88% and a mask mAP of 87% on the validation dataset, which is robust enough for our segmentation needs.

Orientation Model

The orientation model is responsible for disambiguating the orientation of the tracked object. For the orientation problem in our specific use case we had a training dataset with 400

	Classification				Regression			
	D=50 (120e)	D=100 (60e)	D=200 (30e)	D=400 (15e)	D=50 (120e)	D=100 (60e)	D=200 (30e)	D=400 (15e)
Baseline	0.45 ± 0.02	0.46 ± 0.03	0.45 ± 0.03	0.46 ± 0.03	0.35 ± 0.04	0.31 ± 0.02	0.31 ± 0.04	0.29 ± 0.01
+ Transfer learning	0.82 ± 0.06	0.96 ± 0.02	0.95 ± 0.02	0.96 ± 0.01	0.66 ± 0.10	0.85 ± 0.06	0.85 ± 0.04	0.82 ± 0.05
+ Rotation	0.91 ± 0.04	0.98 ± 0.01	0.97 ± 0.01	0.97 ± 0.01	0.79 ± 0.03	0.86 ± 0.06	0.90 ± 0.03	0.85 ± 0.05

Table 1. Evaluation results. D stands for size of the dataset, followed by the number of epochs. Each experiment was executed 5 times and we report the average accuracy. The best results were obtained when using a classifier and a training procedure using transfer learning and rotation as part of the augmentation techniques. Higher accuracies were obtained when the dataset had at least 100 samples.

images and a validation dataset with 192 images. Each image had the ground truth of the 2D pose of the brick. In this section we will test the following three hypotheses related to this model:

(H1) *With a small dataset, using transfer learning improves accuracy.*

Models pre-trained on ImageNet [7] tend to lead to improved performance for diverse image classification tasks [8, 31]. However, recent research [18] has demonstrated that, for some small fine-grained image classification datasets, the benefits of transfer learning are minimal.

(H2) *Augmenting data with random rotations leads to higher accuracy.*

Rotation in image space is an augmentation technique that has been used successfully in previous work [25]. We expect that such an augmentation is particularly beneficial when training a model that predicts the orientation of an object.

(H3) *Solving our problem using a regression loss function leads to better performance compared to a classification loss function.*

Since the goal of regression is to predict the exact orientation of the object, we expect it to be more accurate when comparing to solving the problem for classification.

Considering that classification alone would not be sufficient to get the orientation of circular objects, we solved the problem using regression to estimate the 2D rotation unit vector of the object. To test H3, we compared the accuracy of regression models to classification models by assigning a class from the rotation vector estimated through regression. This can be obtained by normalizing the output vector and assigning it to its corresponding class.

We performed various experiments to test our hypotheses (see Table 1). We tried different sizes of training datasets, since it is not only relevant to know how large the dataset has to be in order to solve the orientation problem, but also to explore the efficiency of the different refinements in the training procedure when the size of the dataset varies. We used stochastic gradient descent with momentum to train the orientation models and used a ResNet-50 architecture [12] in all the experiments, due to its simplicity and accuracy on the ImageNet dataset. For transfer learning, we used weights pre-trained on the ImageNet dataset [7]. The number of epochs was adjusted according to the size of the dataset.

Each experiment was executed 5 times, and we report the average accuracies in Table 1. In preliminary experiments we obtained the best results with a learning rate of 0.001, a batch size of 8, and a weight decay of 0.00004. Therefore, we used these hyperparameters for all further experiments. We did not decay the learning rate for the experiments in Table 1, since for the bigger dataset sizes the number of epochs is low. When solving our problem using regression, we used the mean squared error loss function. For the classification problem we used cross entropy loss. In all experiments, after cropping the image to the bounding box from our detector, we cropped the pictures with an aspect ratio randomly sampled in $[\frac{3}{4}, \frac{4}{3}]$ and an area distributed between 8% and 100%, finally resizing them to the input size of the network (224x224). This method has been used successfully in previous work [13, 41] and also worked well for the brick. However, this may be facilitated by the fact that the symmetry-breaking LEGO logo is present on most of the brick's surface. For objects where fine details are important it might be necessary to keep the image aspect ratio unchanged and add padding to the image, or make changes to the CNN architecture to support a larger input size.

To test H1 and H2, we conducted a baseline training experiment where we did not use transfer learning. We added each of the refinements incrementally, hence in the *Transfer learning* row of experiments in Table 1 we used a pre-trained model, and in the *Rotation* row we added rotation as a data augmentation technique. For the latter, we randomly rotated the image by an angle between $[-30°, 30°]$ and adjusted the ground truth accordingly. Intrigued by the lower accuracy obtained when solving the problem with a regression loss function, we decided to run additional experiments for longer with the larger training dataset (90 epochs). The learning rate was adjusted to 0.002, but decayed at a rate of 0.1 every 30 epochs. Results thereof are shown in Figure 9.

Discussion of results

The results in Table 1 are in line with H1 and H2. For this use case, transfer learning always resulted in substantial improvements in accuracy. Using rotation as an augmentation technique also resulted in better accuracy, in particular in cases when the dataset was small. These results provide evidence that it is possible to perform the rotation disambiguation with a very small dataset. Contrary to H3, our results indicate that classification always performed better than regression in this particular task. The graph of accuracies shown in Figure 9 also suggests that when solving the problem using a classifier, the

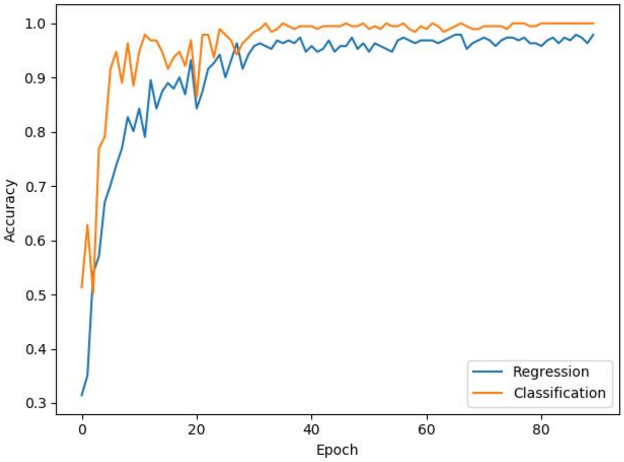


Figure 9. Accuracy comparison between a model using classification and a model using regression, trained with the complete training dataset containing 400 pictures.

model was able to learn faster than with regression. However, these findings are closely related to the choice of architecture, loss function and training procedures. Hence, further research is needed to understand why approaching the problem from a classification perspective results in better performance.

Other materials and shapes

In this paper we apply well established deep learning algorithms that have been used successfully to accomplish different visual recognition tasks [11, 12, 20] in a variety of complex datasets [7, 21]. Therefore, we speculate that our approach is generalizable for most near-symmetrical objects that require such QA procedures in industry. To support this argument, we further tested the orientation model with the pump component depicted in Figure 2 (e). This component has 4 degrees of symmetry and is composed of black plastic and metal. The zoomed in image in Figure 10 shows its symmetry-breaking features, which consist of several holes of different shapes and sizes, as well as a bright vertical element. The experiment was conducted using a training dataset with 100 images, trained over 60 epochs, with the same hyper-parameters as described in Table 1. We used 8 classes, spanning 45° each. The results of this experiment, given in Table 2, show similarly high accuracies as earlier experiments with the brick (see Table 1, column with classification, D=100, 60e). While these results support that our method is generalizable, further research is necessary to confirm this assumption.

	Classification (D=100, 60e)
Baseline	0.91 ± 0.05
+ Transfer learning	0.95 ± 0.02
+ Rotation	1.00 ± 0.00

Table 2. Results of experiments using object e from Figure 2. D stands for size of the dataset, followed by the number of epochs. Each experiment was executed 5 times and we report the average accuracy.

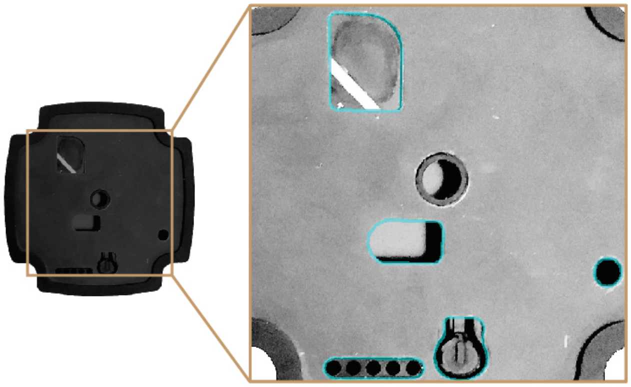


Figure 10. Near-symmetrical pump component made of black metal and plastic. It measures 8.5x8.5cm. The zoomed in image on the right is enhanced to highlight the symmetry-breaking features of the component (outlined in blue).

DISCUSSION AND FUTURE WORK

Our system is inspired by metrology practices in QA at several manufacturing companies associated with the MADE project. While such practices involve various types of objects and different measurement tools, we focus merely on a subset of a metrologist's task space. We hope that in the future the concepts presented in this paper can be applied to more varied measurement activities. In this section, we reflect on essential parts of our pipeline, discuss limitations and give considerations for future work.

Degrees of symmetry

In the presented work, we primarly focused on 180°-symmetric objects (e.g., a 2x4 LEGO brick). This means that after context tracking, there are two possible rotations to choose from (see "Context" in Figure 7). We then use four classes at 90° to each other (i.e., up, down, left, right), to resolve uncertainties, as is shown in Figure 7 ("Focus"). Even if the orientation is close to the boundaries between two classes and the classifier is undecided, the end-result remains valid. For instance, if the detected direction is exactly between "up" and "left", it does not matter whether the classifier outputs "up" or "left", since in both cases the resulting vector based on the dot product will be the same. We can therefore argue that for a 180° symmetrical object, the minimum number of classes for resolving ambiguities orientation is three, i.e., each spanning 120°. In our example we use four classes, to increase the stability and yield a more intuitive set of directions for output and training.

From this we can go on to surmise more generally, that the minimum number of classes to disambiguate orientation is the degree of rotational symmetry plus one (i.e., with 180° symmetry, a resulting vector can stem from exactly 2 different orientations, ergo $2+1=3$ classes). These classes must be evenly distributed around a full circle (360°). For instance, a 90°-symmetric object, such as a the pump component in Figure 2 (e), would require a minimum of five classes, spanning 72° each. This approach is limited regarding round shapes, like discs, since these have no discrete set of rotations to disambiguate from. For instance, for a round object with a small non-symmetric logo in the middle the context tracking pipeline in our setup could merely provide the object's location for the focus camera to orient towards, but all orientation information would need to be provided by focus tracking. This can be achieved with the orientation model that estimates the exact rotation, as was discussed in the *Technical Evaluation* section.

Limitations and alternatives

There is room for improvement in several parts of the Focus+Context tracking pipeline. Our current scope covers measurement steps when the object is oriented so that the symmetry breaking feature faces up towards the cameras. With small adjustments to the setup, the same principles may be used to cover further cases (e.g., a side-ways brick) and support a larger variety of measurement steps. In more general terms, in the future we intend to integrate our orientation model in the Mask R-CNN framework to explore real-time 3D pose estimation using deep learning. Other promising approaches could involve continuously tracking the object using information from the previous known pose, or designing a deep learning framework that uses the input of both context and focus cameras to improve accuracy. This could also help cope with the issue of occlusion, which persists in particular when measuring small objects. As of now, the object has to be visible to the focus camera so the system can provide instructions with the correct orientation. However, in these situations, the instruction could still be visualized in an initial default pose or the last known one.

Currently, although the deep learning algorithms run in real time, the system has some latency caused by the pan-tilt unit and DSLR camera. Rotating the unit and taking a picture takes some seconds before it is received by the main PC. One way to circumvent this practical limitation would be to use multiple focus cameras. A faster pan-tilt unit, or industrial cameras with zoom lenses and high resolution video streams would also reduce the system's latency.

Alternative solutions could also be explored in regards to the display technology. In our system, we currently use LCD screens to prevent worker instrumentation. However, projectors and head-mounted displays could allow co-location of measurement guides even in a hand-held tracking scenario. We aim to explore further display options and their trade-offs in the future.

Future long-term evaluation with experts

To assess the practical value of our proposed solution for digital assistance in applied metrology, a long-term evaluation of our system at manufacturing companies is required. Arguably our guidance system can lead to performance advantages in QA. In an unaided scenario, workers currently need to closely inspect an object to identify and adjust its orientation manually, before referring to the measurement instructions, and must then manually enter the values in the correct field. Our algorithm detects the object's orientation for them and presents the measurement guides accordingly, which removes the need for close scrutiny and mental rotations. Furthermore, our proposed approach entails that both the guides and the object are always visible in the worker's field of view, which reduces task complexity [36]. While a field study is beyond the scope of this paper, it would allow us to explore the efficacy of our approach, identify further limitations, and help us to better address the workers' needs. The results of such a study would also lead to further development of our system. For example, this could involve a step for verification of measurements - i.e., tracking the worker and the measurement tool to verify what position is measured, to ensure that it is measured correctly and allow automatic recording of values. By providing the technical details involved in tracking objects for digital assistance during applied metrology, this paper forms the ground work for further development and evaluation during deployment in the field.

Applications beyond metrology

The presented approach is aimed at industrial metrology tasks that are executed in a conventional work space (consisting of a desk, chair, measurement tools and a computer). This arguably makes our solution easily transferable to similar activities beyond metrology, e.g., in a play context. For instance, Miller et al. [22] track the building process of a colored brick construction to create a virtual replication thereof. This could be extended through our concept, by additionally tracking the unique orientation of each new brick whenever the user attaches it to the construction. This would add degrees of freedom to the building process without requiring specialized bricks: the orientation of a brick could alter the local appearance of a virtual texture that spans across the construction, or it could be used to define the *inside* and the *outside* of the construction.

CONCLUSION

In this paper, we present the basic concepts for providing digital assistance for metrology during quality assurance in manufacturing. In particular, we propose a two-step approach for pose estimation of near-symmetrical objects, which we call Focus+Context tracking. By combining (1) coarse-grained object recognition with context cameras and (2) precise pose estimation based on fine-grained features with a focus camera, we leverage (1) fast computer vision techniques and (2) accurate deep learning strategies. We describe the tracking pipeline we implemented and elaborate on how this was applied to a typical metrology scenario, using a 2x4 LEGO brick as an example use case. The results show that the fine-grained features on a brick are sufficient to successfully estimate its pose, including its unique orientation, with very high accuracy. We further apply our framework to the example of tracking a pump component and argue that the presented concept for pose-estimation can be extended to a wider range of applications with near-symmetric objects, or more generally, nearly identical objects with small distinguishing features.

ACKNOWLEDGEMENTS

Thanks are due to Marius Hográfer and Aïna Georges for their constructive feedback. This work was supported by the Innovation Fund Denmark (IFD) grant no. 6151-00006B, as part of the Manufacturing Academy of Denmark *MADE Digital* project.

REFERENCES

[1] H. Aghajan and A. Cavallaro. 2009. *Multi-Camera Networks: Principles and Applications*. Elsevier Science. https://books.google.dk/books?id=XA_6o2dhTGEC

[2] K. M. Baird and W. Barfield. 1999. Evaluating the effectiveness of augmented reality displays for a manual assembly task. *Virtual Reality* 4, 4 (01 Dec 1999), 250–259. DOI:http://dx.doi.org/10.1007/BF01421808

[3] Patrick Baudisch, Nathaniel Good, and Paul Stewart. 2001. Focus Plus Context Screens: Combining Display Technology with Visualization Techniques. In *Proceedings of the 14th Annual ACM Symposium on User Interface Software and Technology (UIST '01)*. ACM, New York, NY, USA, 31–40. DOI: http://dx.doi.org/10.1145/502348.502354

[4] Rein Van Den Boomgaard and Joost Van De Weijer. 2002. Robust Estimation of Orientation for Texture Analysis. (2002).

[5] T. P. Caudell and D. W. Mizell. 1992. Augmented reality: an application of heads-up display technology to manual manufacturing processes. In *Proceedings of the Twenty-Fifth Hawaii International Conference on System Sciences*, Vol. ii. IEEE, Kauai, HI, USA, USA, 659–669 vol.2. DOI:http://dx.doi.org/10.1109/HICSS.1992.183317

[6] Lynn A Cooper. 1975. Mental rotation of random two-dimensional shapes. *Cognitive Psychology* 7, 1 (1975), 20 – 43. DOI:http://dx.doi.org/https://doi.org/10.1016/0010-0285(75)90003-1

[7] J. Deng, W. Dong, R. Socher, L. Li, and and. 2009. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*. 248–255. DOI: http://dx.doi.org/10.1109/CVPR.2009.5206848

[8] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. 2014. DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition. In *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32 (ICML'14)*. JMLR.org, I–647–I–655. http://dl.acm.org/citation.cfm?id=3044805.3044879

[9] Abhishek Dutta and Andrew Zisserman. 2019. The VIA Annotation Software for Images, Audio and Video. (2019).

[10] Andreas Fender and Jörg Müller. 2018. Velt: A Framework for Multi RGB-D Camera Systems. In *Proceedings of the 2018 ACM International Conference on Interactive Surfaces and Spaces (ISS '18)*. ACM, New York, NY, USA, 73–83. DOI: http://dx.doi.org/10.1145/3279778.3279794

[11] K. He, G. Gkioxari, P. Dollár, and R. Girshick. 2017. Mask R-CNN. In *2017 IEEE International Conference on Computer Vision (ICCV)*. 2980–2988. DOI: http://dx.doi.org/10.1109/ICCV.2017.322

[12] K. He, X. Zhang, S. Ren, and J. Sun. 2016. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 770–778. DOI: http://dx.doi.org/10.1109/CVPR.2016.90

[13] Tong He, Zhi Zhang, Hang Zhang, Zhongyue Zhang, Junyuan Xie, and Mu Li. 2018. Bag of Tricks for Image Classification with Convolutional Neural Networks. (2018).

[14] S. Henderson and S. Feiner. 2011a. Exploring the Benefits of Augmented Reality Documentation for Maintenance and Repair. *IEEE Transactions on Visualization and Computer Graphics* 17, 10 (Oct 2011), 1355–1368. DOI: http://dx.doi.org/10.1109/TVCG.2010.245

[15] S. J. Henderson and S. K. Feiner. 2011b. Augmented reality in the psychomotor phase of a procedural task. In *2011 10th IEEE International Symposium on Mixed and Augmented Reality*. 191–200. DOI: http://dx.doi.org/10.1109/ISMAR.2011.6092386

[16] K. Jafari-Khouzani and H. Soltanian-Zadeh. 2005. Radon transform orientation estimation for rotation invariant texture analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27, 6 (June 2005), 1004–1008. DOI: http://dx.doi.org/10.1109/TPAMI.2005.126

[17] SeungJun Kim and Anind K. Dey. 2009. Simulated Augmented Reality Windshield Display As a Cognitive Mapping Aid for Elder Driver Navigation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '09)*. ACM, New York, NY, USA, 133–142. DOI: http://dx.doi.org/10.1145/1518701.1518724

[18] Simon Kornblith, Jonathon Shlens, and Quoc V. Le. 2019. Do Better ImageNet Models Transfer Better?. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[19] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger (Eds.). Curran Associates, Inc., 1097–1105. http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

[20] T. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. 2017. Feature Pyramid Networks for Object Detection. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 936–944. DOI: http://dx.doi.org/10.1109/CVPR.2017.106

[21] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. Microsoft COCO: Common Objects in Context. In *Computer Vision – ECCV 2014*, David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars (Eds.). Springer International Publishing, Cham, 740–755.

[22] A. Miller, B. White, E. Charbonneau, Z. Kanzler, and J. J. LaViola Jr. 2012. Interactive 3D Model Acquisition and Tracking of Building Block Structures. *IEEE Transactions on Visualization and Computer Graphics* 18, 4 (April 2012), 651–659. DOI: http://dx.doi.org/10.1109/TVCG.2012.48

[23] U. Neumann and A. Majoros. 1998. Cognitive, performance, and systems issues for augmented reality applications in manufacturing and maintenance. In *Proceedings. IEEE 1998 Virtual Reality Annual International Symposium (Cat. No.98CB36180)*. 4–11. DOI:http://dx.doi.org/10.1109/VRAIS.1998.658416

[24] V. Paelke. 2014. Augmented reality in the smart factory: Supporting workers in an industry 4.0. environment. In *Proceedings of the 2014 IEEE Emerging Technology and Factory Automation (ETFA)*. 1–4. DOI: http://dx.doi.org/10.1109/ETFA.2014.7005252

[25] Luis Perez and Jason Wang. 2017. The Effectiveness of Data Augmentation in Image Classification using Deep Learning. (2017).

[26] J. Platonov, H. Heibel, P. Meier, and B. Grollmann. 2006. A mobile markerless AR system for maintenance and repair. In *2006 IEEE/ACM International Symposium on Mixed and Augmented Reality*. 105–108. DOI: http://dx.doi.org/10.1109/ISMAR.2006.297800

[27] J. Polvi, T. Taketomi, A. Moteki, T. Yoshitake, T. Fukuoka, G. Yamamoto, C. Sandor, and H. Kato. 2018. Handheld Guides in Inspection Tasks: Augmented Reality versus Picture. *IEEE Transactions on Visualization and Computer Graphics* 24, 7 (July 2018), 2118–2128. DOI: http://dx.doi.org/10.1109/TVCG.2017.2709746

[28] PyTorch. 2019. PyTorch - An open source deep learning platform that provides a seamless path from research prototyping to production deployment. (2019). https://pytorch.org/ Accessed: 6/25/2019.

[29] Roman Rädle, Hans-Christian Jetter, Jonathan Fischer, Inti Gabriel, Clemens N. Klokmose, Harald Reiterer, and Christian Holz. 2018. PolarTrack: Optical Outside-In Device Tracking that Exploits Display Polarization. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (proceedings of the 2018 chi conference on human factors in computing systems ed.). ACM. https://www.microsoft.com/en-us/research/publication/polartrack-optical-outside-device-tracking-exploits-display-polarization/

[30] Ramesh Raskar, Greg Welch, Matt Cutts, Adam Lake, Lev Stesin, and Henry Fuchs. 1998. The Office of the Future: A Unified Approach to Image-based Modeling and Spatially Immersive Displays. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '98)*. ACM, New York, NY, USA, 179–188. DOI: http://dx.doi.org/10.1145/280814.280861

[31] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. 2014. CNN Features Off-the-Shelf: An Astounding Baseline for Recognition. In *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW '14)*. IEEE Computer Society, Washington, DC, USA, 512–519. DOI: http://dx.doi.org/10.1109/CVPRW.2014.131

[32] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. 2016. You Only Look Once: Unified, Real-Time Object Detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 779–788. DOI: http://dx.doi.org/10.1109/CVPR.2016.91

[33] Dirk Reiners, Didier Stricker, Gudrun Klinker, and Stefan Müller. 1999. Augmented Reality for Construction Tasks: Doorlock Assembly. In *Proceedings of the International Workshop on Augmented Reality : Placing Artificial Objects in Real Scenes: Placing Artificial Objects in Real Scenes (IWAR '98)*. A. K. Peters, Ltd., Natick, MA, USA, 31–46. http://dl.acm.org/citation.cfm?id=322690.322694

[34] Jun Rekimoto and Masanori Saitoh. 1999. Augmented Surfaces: A Spatially Continuous Work Space for Hybrid Computing Environments. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '99)*. ACM, New York, NY, USA, 378–385. DOI:http://dx.doi.org/10.1145/302979.303113

[35] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett (Eds.). Curran Associates, Inc., 91–99. http://papers.nips.cc/paper/5638-faster-r-cnn-towards-real-time-object-detection-with-region-proposal-networks.pdf

[36] C. M. Robertson, B. MacIntyre, and B. N. Walker. 2008. An evaluation of graphical context when the graphics are outside of the task area. In *2008 7th IEEE/ACM International Symposium on Mixed and Augmented Reality*. 73–76. DOI: http://dx.doi.org/10.1109/ISMAR.2008.4637328

[37] Robert D Rogers and Stephen Monsell. 1995. Costs of a predictible switch between simple cognitive tasks. *Journal of experimental psychology: General* 124, 2 (1995), 207.

[38] Roger N. Shepard and Jacqueline Metzler. 1971. Mental Rotation of Three-Dimensional Objects. *Science* 171, 3972 (1971), 701–703. DOI: http://dx.doi.org/10.1126/science.171.3972.701

[39] K. Simonyan and A. Zisserman. 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *International Conference on Learning Representations*.

[40] Sony. 2019. Imaging Edge Remote. (2019). https://imagingedge.sony.net/en-us/ie-desktop.html Accessed: 6/22/2019.

[41] C. Szegedy, Wei Liu, Yangqing Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. 2015. Going deeper with convolutions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 1–9. DOI: http://dx.doi.org/10.1109/CVPR.2015.7298594

[42] Arthur Tang, Charles Owen, Frank Biocca, and Weimin Mou. 2003. Comparative Effectiveness of Augmented Reality in Object Assembly. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '03)*. ACM, New York, NY, USA, 73–80. DOI: http://dx.doi.org/10.1145/642611.642626

[43] Antonio E. Uva, Michele Gattullo, Vito M. Manghisi, Daniele Spagnulo, Giuseppe L. Cascella, and Michele Fiorentino. 2018. Evaluating the effectiveness of spatial augmented reality in smart manufacturing: a solution for manual working stations. *The International Journal of Advanced Manufacturing Technology* 94, 1 (01 Jan 2018), 509–521. DOI: http://dx.doi.org/10.1007/s00170-017-0846-4

[44] S. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh. 2016. Convolutional Pose Machines. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 4724–4732. DOI: http://dx.doi.org/10.1109/CVPR.2016.511

[45] Pierre Wellner, Wendy Mackay, and Rich Gold. 1993. Back to the Real World. *Commun. ACM* 36, 7 (July 1993), 24–26. DOI: http://dx.doi.org/10.1145/159544.159555

[46] Pierre David Wellner. 1994. *Interacting with paper on the DigitalDesk*. Technical Report UCAM-CL-TR-330. University of Cambridge, Computer Laboratory. https://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-330.pdf

[47] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. 2014. How transferable are features in deep neural networks? In *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger (Eds.). Curran Associates, Inc., 3320–3328. http://papers.nips.cc/paper/5347-how-transferable-are-features-in-deep-neural-networks.pdf

[48] Guy W. Zimmerman, Dale Klopfer, G. Michael Poor, Julie Barnes, Laura Leventhal, and Samuel D. Jaffee. 2011. "How Do I Line Up?": Reducing Mental Transformations to Improve Performance. In *Proceedings of the 14th International Conference on Human-computer Interaction: Design and Development Approaches - Volume Part I (HCII'11)*. Springer-Verlag, Berlin, Heidelberg, 432–440. http://dl.acm.org/citation.cfm?id=2022384.2022435