Towards Industrial IoT-AR Systems using Deep Learning-Based Object Pose Estimation

Yongbin Sun*, Sai Nithin Reddy Kantareddy*, Joshua Siegel†
Alexandre Armengol-Urpi*, Xiaoyu Wu‡, Hongyu Wang§ and Sanjay Sarma*
*Auto-ID Lab, Department of Mechanical Engineering, Massachusetts Institute of Technology
†Department of Computer Science and Engineering, Michigan State University
‡Department of Computer Science, Boston University
§Department of Industry Engineering, University of Miami
Email: yb_sun@mit.edu, nithin@mit.edu, jsiegel@msu.edu

***Abstract*—Augmented Reality (AR) is known to enhance user experience, however, it remains under-adopted in industry. We present an AR interaction system improving human-machine coordination in Internet of Things (IoT) and Industry 4.0 applications including manufacturing and assembly, maintenance and safety, and other highly-interactive functions. A driver of slow adoption is the computational complexity and inaccuracy in localization and rendering digital content. AR systems may render digital content close to the associated physical objects, but traditional object recognition and localization modules perform poorly when tracking texture-less objects and complex shapes, presenting a need for robust and efficient digital content rendering techniques. We propose a method of improving IoT-AR by integrating Deep Learning with AR to increase accuracy and robustness of the target object localization module, taking both color and depth images as input and outputting the target's pose parameters. Quantitative and qualitative experiments prove this system's efficacy and show potential for fusing these emerging technologies in real-world applications.**

***Index Terms*—Internet of Things, Augmented Reality, Industry 4.0, Object Pose Estimation, Visualization, User Interaction**

I. Introduction

Today's cost-sensitive consumers demand custom products requiring on-demand production. Flexible, elastic manufacturing has become a critical differentiator, with IoT enabling such capabilities. Hyperconnectivity supports seamless machine interaction, data-driven decision making, richer sensing, and real-time monitoring and control. IoT makes sensing ubiquitous and data more accessible, combining with advanced sensor and semantic web technologies to underpin Big Data analytics. AR compliments this human-machine coordination by increasing process visibility. Firms can leverage these improvements for rapid design modifications and swift maintenance to meet dynamic production and reduce production downtime.

AR is an interactive technology that enriches user experience by superimposing virtual digital content into the real world. Existing rendering techniques and devices have enabled AR's adoption in domains including entertainment [1] and commerce [2]. Yet, there is more value in AR than simply displaying information – for example, by seamlessly blending real-time information into a user's perception of surrounding environment. AR systems have recently started to fuse heterogeneous technologies (e.g. computer vision and image segmentation) to better understand geometric and semantic properties of a working environment, so that they can render information at the corresponding position of a detected target object. This creates a more satisfying visualization and interaction experience, making AR more useful across domains including manufacturing.

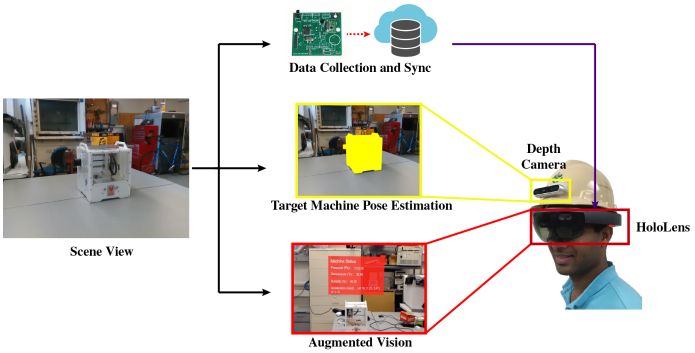


Fig. 1. Our system collects the target machine's information in real-time and estimates its pose via a helmet-mounted depth camera. The operator can visualize accurately and precisely superimposed augmented digital information through the HoloLens.

Integrating IoT with AR is a natural next step in unlocking the greatest potential of the manufacturing industry. Specifically, combining IoT and AR can reduce a manufacturer's downtime by enabling real-time, data-informed condition monitoring of machines and providing intelligent assistance in case of repairs, allowing unskilled employees to conduct basic maintenance. Currently, *machine status monitoring* and *maintenance assistance* receive the most research and industrial attention, and reducing their costs and enhancing associated efficiency are key to smart factory applications.

Fusing IoT and AR is not entirely novel, and has been demonstrated in [3]–[5]. However, AR has not been adopted to the fullest in a manufacturing context. One reason is that current object localization algorithms are insufficiently robust to handle industrial requirements, such as objects without distinguishable visual patterns (denoted as textureless objects), so rendered contents cannot reliably be accurately displayed in the correct position, resulting in a poor user experience. There

978-1-7281-1025-7/19/$31.00 ©2019 IEEE

therefore remains significant demand for accurate object pose estimation, motivating this paper's goal of building an accurate target object pose estimation system to support joint IoT-AR implementations.

Many contemporary existing AR systems use fiducial markers, such as QR codes, to recognize and localize target objects. This marker-based method requires additional preconfiguration processes and is not applicable to certain shapes (e.g. thin structures). Deriving object pose estimates directly from unmodified appearances is preferable. Ongoing work uses computer vision to extract handcrafted visual and geometric descriptors for target object pose (rotation and translation) estimation [6], [7], but these methods usually suffer from high variability under changing conditions (e.g. lighting) or with sensor noise. They also require detailed surface textures, and cannot handle complex geometric shapes. At the same time, advances in Artificial Intelligence (AI) has shown promise for effective pose estimation [8], [9]. We propose taking advantage of enhancements in AI and building upon existing techniques to improve object pose estimation accuracy, supporting IoT-AR in industrial applications.

In summary, we show how to improve an IoT-AR system by integrating state-of-the-art deep learning models with computer vision, and demonstrate the hybrid system with two use cases: machine status monitoring (Fig. 1) and maintenance assistance. More specifically, we make following contributions:

- Retrofitting existing commercial fabrication setups to create custom AR support and monitoring tools with available general purpose sensors
- Fast and efficient object pose estimation methodology
- Improved visualization to create accurate projections enhancing the overall user experience
- Scalable methodology to gather multi-sensor data for future maintenance and support requirements

This paper is organized as follows: Section II briefly summarizes relevant work; Section III introduces the pipeline and technical details of the proposed system; Section IV evaluates our system quantitatively and qualitatively, followed by our conclusion in Section V.

II. Related Work

Since this paper focus on enhancing machine monitoring and maintenance experience in AR environments by increasing pose estimation accuracy, we summarized relevant work under these topics.

A. AR for Manufacturing

AR for manufacturing has been previously explored as mobile maintenance support tool for remote maintenance assistance. Tools can be integrated with existing factory scheduling systems and increase the responsiveness in the event of downtime [10]. In a more recent study, AR application for industrial guidance and training is explored by creating a low-cost and easily scalable application development platform [11]. Another study focused on creating technical documentation and manuals for such industrial guidance [12]. The study demonstrated the methodology to create a maintenance manual for hydraulic breakers, The AR-based manuals create a future of immersive training and maintenance experiences for factory operators and engineers. Another study demonstrated a prototype of a methodology to help operators go through a sequential maintenance task. For repairing critical and expensive machinery such as well turbines, repair time and quality of execution are important. During such tasks, operators can get guided support through interactive manuals [13]. Human and AR tool interaction can be further enhanced by creating a feedback loop between operator's actions and AR visualization using sensing technologies. In one study, operator's wrist tracking is used to ensure the sequence of steps is followed to provide adapative user support further reducing the barriers between users (operators) and AR guiding tools [14]. Although these applications demonstrate a promising potential for AR in enabling industry 4.0 and some of the applications are already in use at commercial/trial stage, some of the challenges still exist. One of the key enabler for these tools is the real-time object identification and visualization. Complex object shapes, clutter, and background affect how accurately and how quickly the information can be processed and presented to the user. In this paper, we build up on some of the applications present in the literature and make improvements.

B. Object Pose Estimation

1) Fiducial Marker-based Method: Fiducial markers, such as QR codes, are commonly used in AR systems for detection and pose estimation purpose [15]. Usually, fiducial markers are either attached to target objects to dynamically infer objects' poses [16], or fixed positions in environments as static landmarks to estimate the camera's pose [17], [18]. Despite many easy-to-use tools and Application Programming Interfaces (APIs), such as ARToolKit [19], ARToolKit+ [20], and ARTag [21], have been developed, this approach requires additional preconfiguration processes and is not universally applicable, thus it is more desired to estimate objects' poses directly from their natural appearances.

2) 2D Visual Feature-based Method: One of the earliest practice to estimate objects' poses without additional artificial markers is to detect and match 2D visual features between scene images and known object models [22]–[25] through PnP algorithm [26]. However, the performance of these methods degenerate for low-texture or low-resolution inputs.

3) 3D Geometric Feature-based Method: 3D geometric local features provide another way to estimate objects' poses based on their geometric properties. This method works on 3D point clouds. Descriptors [27] from local point sets are extracted to match point pairs between two point clouds, and matched point pairs are used to compute the optimal pose. Iterative closest point (ICP) algorithm and its variants [28], [29] are usually used as a post-processing step to increase accuracy.

4) Deep Learning-based Method: Existing deep learning methods have explored various input sources for object pose estimation. 2D convolution neural network (CNN) can be

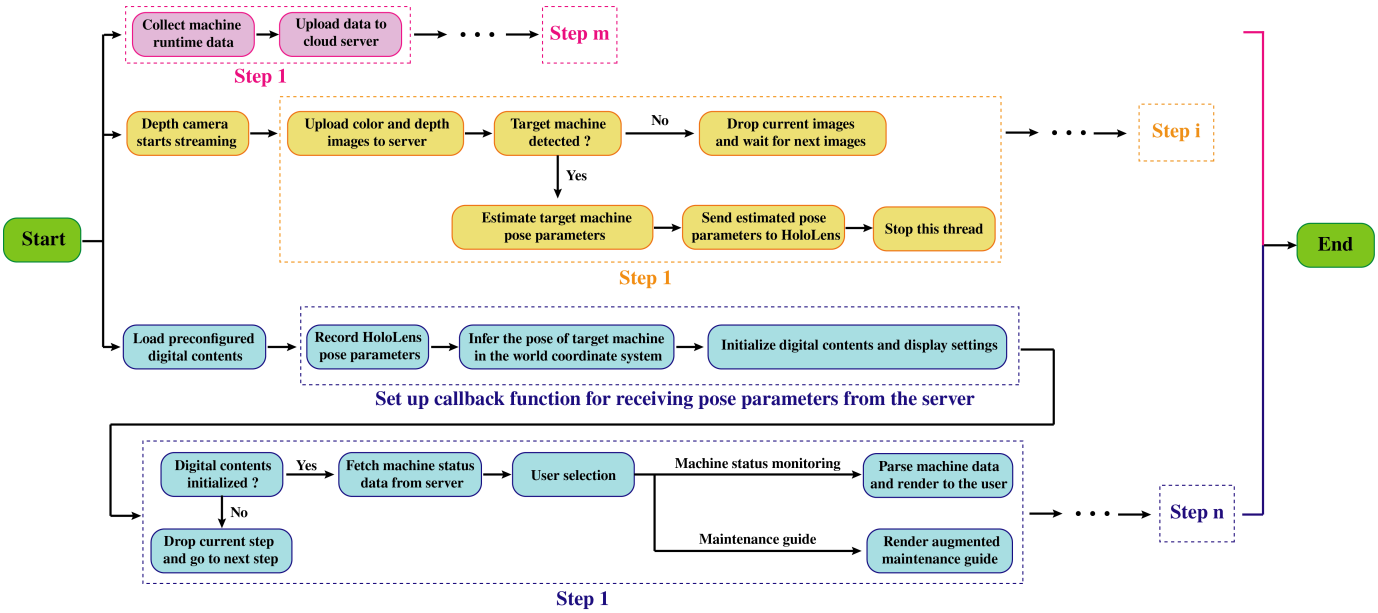


Fig. 2. Workflow of the proposed system.

directly adopted to estimate object pose parameters from RGB images [30]–[32]. Standard 2D convolution operations can also be extended to process 3D data for pose estimation. For example, 3D data can be voxelized and fed as input to 3D CNN models [33], [34], or directly passed in point cloud format to PointNet-based [35] models [36]. However, using either color or geometric information alone lacks full knowledge of the environment for pose estimation. Therefore, inspired by [37]–[39], we use information from both sources.

III. System

The proposed system include three main components: machine status data collection, target machine localization, and augmented visualization, as shown in Fig. 1. They work collaboratively to obtain machine status information and render it to the user through the head mounted display (HMD) at corresponding positions. In this section, we describe technical and design details of our prototype system.

A. System Workflow

Our system integrates heterogeneous technologies, including IoT, AI and AR, into a fully functional system. Fig. 2 shows the overall workflow with respect to the steps required to achieve the system's functionality. Its three components work in individual threads, and communicate to a central server independently. The first component (pink block) is deployed on the machine to collect and upload machine status data to the server at each step, and the server shares them to other components of the system. Since this thread serves as a data provider, it begins as the system starts and stops till the system ends. The second component (yellow block) estimates pose parameters of the target machine (e.g. its relative position and orientation to the camera), given streamed color and depth images captured from a depth camera mounted on the helmet. Since the computation load is heavy, this task is performed on the server using Graphics Processing Unit (GPU). The server sends machine pose parameters to the HoloLens if they are successfully estimated, otherwise, current images will be dropped and this thread enters the next step. This thread terminates when the target machine is correctly localized, because HoloLens put digital content into the fixed world coordinate system and only need to be set once. The third component (blue block) takes care of visualizing preconfigured digital content (e.g. machine status panels and augmented maintenance guide) to the user through HoloLens. When the system starts, this thread first loads preconfigured digital content, and then sets up a callback function for placing previously loaded digital content according to received pose parameters from the server. This thread displays digital content only after it receives estimated machine pose parameters. In this work, we demonstrate two cases: displaying machine status on virtual panels and showing augmented machine maintenance guide. Similar to the first thread, this visualization thread also terminates till the system ends.

B. Machine Status Synchronization

We used a desktop PCB milling machine by Bantam Tools to demonstrate the functionality of our prototype. Machine has a frame dimension of 140 x 114 x 34.3 mm and capable of cutting PCBs at 2,600 mm/min with spindle speeds raning from 8,500 - 26,000 rpm [40]. We used a battery-powered SensiBLE IoT module to capture physical attributes from the machine in real-time. SensiBLE module is an aggregate of sensors including accelerometer, gyro, magnetometer, temperature, humidity, light, pressure, microphone, proximity and ranging sensors. Although the IoT module is capable of reporting multi-modal sensor data, we used only temperature, humidity, pressure and accelerometer data in building the prototype presented in this paper. SensiBLE connects to a mobile software app over BLE and subsequently streams data

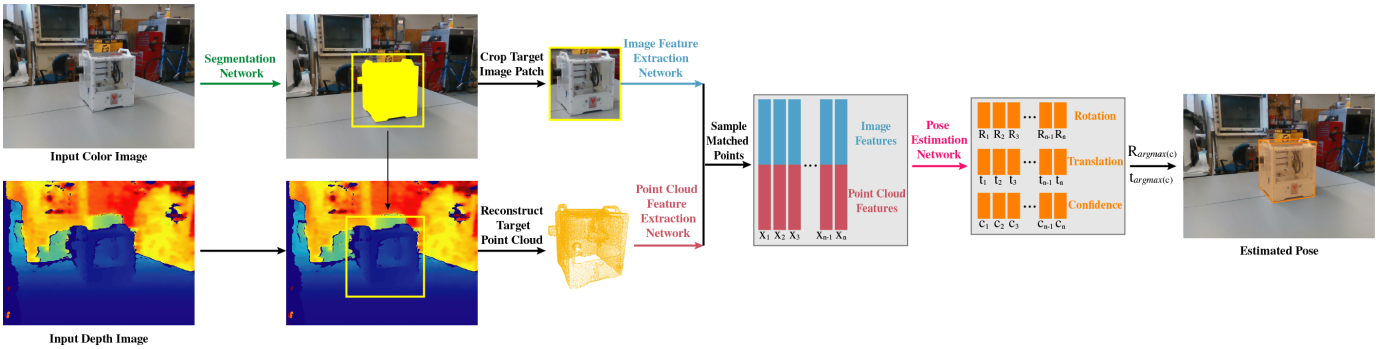


Fig. 3. Pipeline for target machine pose estimation given an input RGB-D image pair.

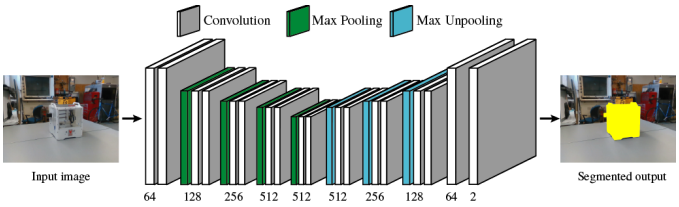


Fig. 4. Machine region segmentation network. The number below each layer indicates the dimension of corresponding feature maps.

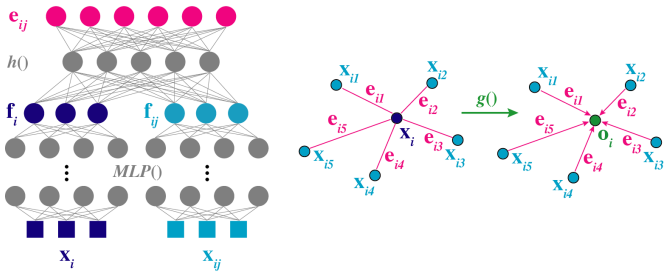


Fig. 5. Left: Compute the pair feature between a central point $\mathbf{x}_i$ and a neighboring $\mathbf{x}_{ij}$. Right: Compute the local geometric feature $\mathbf{o}_i$ for $\mathbf{x}_i$.

to a Mosquitto server over MQTT protocol. Data gleaned from the module is then used in creating visual content through the Hololens.

C. Target Machine Localization

Our pose estimation pipeline includes four deep neural networks, takes as input both color and depth images, and outputs pose parameters for the machine presented in the image. The complete pipeline is shown in Fig. 3.

1) Segmentation Network: To facilitate target machine pose estimation and reduce region of interest within the input image, the target machine is first segmented from the input color image. An encoder-decoder CNN model is implemented to support this purpose. Given an input image containing the target machine, the segmentation network first reduces the spatial dimension of intermediate feature maps via pooling (encoder), and then expands it via unpooling (decoder) to generate a binary output mask of the same size as the input image indicating foreground and background. At each layer, 2D convolution operations are used to aggregate information. The network architecture is shown in Fig. 4.

2) Image Feature Extraction Network: The segmented target machine region contains all the necessary visual information for its pose estimation, and the goal of this part is to extract this information. Specifically, the image feature extraction network takes as input an image patch containing the segmented machine region, and generates embedded visual features for all the pixels. Similarly to the segmentation network, this network also follows an encoder-decoder architecture, and ResNet18 [41] is used here to embed latent correlation between machine appearance and its pose parameters. Formally, given the input image patch of shape $H \times W \times 3$, the model outputs an embedding space of shape $H \times W \times d_{color}$, where d_{color} is the dimension of this visual embedding space.

3) Point Cloud Feature Extraction Network: The machine region in the depth camera contains additional geometric cues correlated to the machine pose parameters (e.g. viewing the machine from different viewpoints and distances would result in different positions, scales and geometric structures presented in the depth image), and it is beneficial to include such information. One direct way to extract this geometric information is to keep using 2D CNN on the cropped depth image patch, but this approach fails to fully discover the intrinsic 3D structure. On the other hand, 3D point clouds describe 3D properties in a more straightforward way, and the accuracy can be improved when operating on the reconstructed 3D point cloud [42]. We follow this direction, and further explore *point-to-point* relationship to enhance overall accuracy and stability of our system. A point cloud of the target machine can be reconstructed from the segmented foreground mask and the depth image, given known transformation parameters between color and depth images (bottom left of Fig. 3).

The goal here is to compute a geometric feature describing local geometric property for each reconstructed point. To begin with, for each point, $\mathbf{x}_i$, we first detect its k nearest neighboring points, $\{\mathbf{x}_{i1}, \mathbf{x}_{i2}, ..., \mathbf{x}_{ij}, ..., \mathbf{x}_{ik}\}$, and obtain their embedded feature vectors using multilayer perceptron (MLP) as in PointNet [35]. This step projects 3D point coordinates of a point, $\mathbf{x}$, to a latent feature embedding, $\mathbf{f}$, expressed as $\mathbf{f}_{ij} = MLP(\mathbf{x}_{ij})$. Next, we form point pairs between $\mathbf{x}_i$ and each of its neighbors $\mathbf{x}_{ij}$, and generate deeper point pair features encoding inter-point relationship. This step can be written as $\mathbf{e}_{ij} = h(\mathbf{f}_i, \mathbf{f}_{ij})$, with $h()$ and $\mathbf{e}_{ij}$ being a learnable feature fusion operation and the point pair feature, respectively.

Finally, we further fuse all pair features around the central point to capture their local geometric property. This step can be expressed as $\mathbf{o}_i = g(\mathbf{e}_{i1}, ..., \mathbf{e}_{ij}, ..., \mathbf{e}_{ik})$, with $g()$ and $\mathbf{o}_i$ being another learnable operation and the output point cloud feature of $\mathbf{x}_i$, respectively. Relevant operations are demonstrated in Fig. 5.

4) Pose Estimation Network: So far, we have obtained both visual and geometric features for segmented machine region, and those information now need to be fused for machine pose estimation. Since the correspondence between each pixel and point is available, features from both sources can be easily fused by concatenating them for matched pixel-point pairs (the middle part of Fig. 3). However, due to occlusion, noise and error in previous processes, different fused features contain valid information to different degrees. Therefore, to potentially find the correct machine pose parameters, as suggested in [42], we generate a set of pose parameters, together with a confidence score, for each fused feature. Ideally, a higher confidence score indicates the estimated pose parameters are closer to the ground truth ones. Note that the fused feature sets share the same format as point cloud spatial coordinates, thus similar MLP structures are reused here to regress pose parameters and confidence scores. As demonstrated in [43], we also generate a global feature vector to obtain global context information via max pooling point features across each feature dimension, and attach it to each individual point feature to enhance information flow along the forward propagation process. Finally, the network predicts rotation parameters (4-element quaternion vector), translation parameters (3-element vector in X, Y and Z direction), and a scalar confidence score. During testing time, the rotation and translation parameters with the highest confidence score is selected. The pose estimation network architecture is shown in Fig. 6.

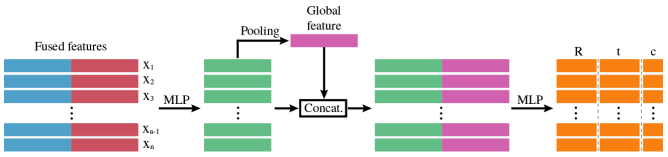


Fig. 6. Pose estimation network.

D. Visualization

After obtaining the target machine's pose, the system needs to superimpose digital content onto the real machine through HoloLens, as in Fig. 1. Since the machine's pose is estimated in the depth camera coordinate system, a pose correction procedure is required for HoloLens visualization.

First, let us define the estimated target machine pose matrix in the depth camera coordinate system as $M_{pose}^{dep\_cam}$, the pose transformation matrix from depth camera coordinate system to the HoloLens virtual camera coordinate system as $T_{dep\_cam}^{HoloLens}$, and the the pose transformation matrix from the HoloLens virtual camera coordinate system to the world coordinate system as $T_{HoloLens}^{world}$. These transformation matrices contain rotation and translation parameters and share the same format:

$$\mathbf{M} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix}$$

, where $\mathbf{R}$ is a 3×3 rotation matrix which can be converted from a quaternion vector, and $\mathbf{t}$ is a 3×1 vector representing translation. Then, the correct machine pose matrix in the world coordinate system, M_{pose}^{world}, which is required for the HoloLens rendering system [1], can be inferred according to:

$$M_{pose}^{world} = T_{HoloLens}^{world} \times T_{dep\_cam}^{HoloLens} \times M_{pose}^{dep\_cam} \quad (1)$$

IV. Evaluation

In this section, we evaluate and demonstrate our system from two main perspectives. First, at the core of our system is a deep learning-based pose estimation model, which shows how to use state-of-the-art AI techniques to improve AR systems, and we evaluate this model quantitatively and qualitatively. Second, the goal of the proposed system is to enhance user experience and working efficacy during manufacturing and relevant processes, so we will also demonstrate the real-time performance of our prototype.

A. Pose Estimation Model Evaluation

1) Dataset: As a well-known fact, the deep learning method is a data driven approach, and training a deep model requires a large amount of labeled samples for supervised learning tasks, including object pose estimation. Since our model takes as input color and depth images and outputs pose parameters, the collected training samples are ⟨color&depth images, target machine pose⟩ pairs. In this work, we collected 1,125 color and depth images covering different backgrounds from various viewpoints. To obtain ground truth pose parameters, we first reconstruct a point cloud from a color and depth image pair, and then manually align the machine's 3D model to its corresponding partial counterpart in the reconstructed scene point cloud. To facilitate this process, we build an aligning tool, which allows us to select matched points between the 3D machine model and scene point cloud. With at least 3 matched point pairs selected, a ground truth transformation matrix can be computed correctly. We split all the manually aligned samples into training and testing sets according to a split ratio of 0.9/0.1.

2) Competing Methods: We include another two common pose estimation methods in existing AR systems for comparison.

Geometric Method We implement the Fast Point Feature Histograms (FPFH) algorithm [44], a popular 3D feature-based method. To calculate the pose that transforms a 3D model to its corresponding position in the scene, FPFH features are computed for both model and scene point clouds, and based on these features the Sample Consensus Initial Alignment (SAC-IA) is used to determine the best target machine pose parameters.

Fiducial Marker Method Fiducial markers are well designed, and their patterns can be used to estimate their poses. If the relative pose between a marker and an object is known, the object's pose can also be easily inferred. To implement this competing method, we put 4 different QR codes around the

[1] https://docs.microsoft.com/en-us/windows/mixed-reality

target machine, and their relative poses to the machine are carefully measured. The machine's pose is estimated from visible QR codes to the camera. This method is expected to work well, but it lacks flexibility and is not universally applicable.

3) Evaluation Metrics: We adopt the average distance as the metric [32] to evaluate pose estimation accuracy. Given the ground truth rotation $\mathbf{R}$ and translation $\mathbf{t}$ and the estimated rotation $\tilde{\mathbf{R}}$ and translation $\tilde{\mathbf{t}}$, this metric computes the mean of the pairwise distances between the 3D model points transformed according to the ground truth pose and the estimated pose:

$$Distance\_error = \frac{1}{M}\sum_{j}||(\mathbf{R}\mathbf{x}_j + \mathbf{t}) - (\tilde{\mathbf{R}}\mathbf{x}_j - \tilde{\mathbf{t}})|| \quad (2)$$

, where M is the number of points considered.

4) Implementation Details: All the four neural networks in the system are implemented using PyTorch library[2]. We use one Nvidia Titan X GPU with 12 GB memory to accelerate neural network training and testing processes. The server machine is equipped with an Intel Core i7-6850K 6-Core 3.60-GHz CPU to test competing methods. The weighted distance error using predicted confidence scores is used as the objective loss function for training our model.

TABLE I
MODEL COMPARISON

	Geometric Method (FPFH + SAC-IA)	**Fiducial Marker** (QR-Code)	**Deep Learning** (Ours)
Run time (*sec.*)	4.732	**0.012**	0.015
Distance error (*m.*)	0.542	0.062	**0.010**

5) Model Performance Evaluation: We evaluate our pose estimation model from two perspectives: performance speed and accuracy. Both competing methods and our model are evaluated on the same testing set for fair comparison. We record the average run time and distance error for all three methods, and report their results in Table I. As can be observed, the geometric method performs significantly slower than fiducial marker and our deep learning methods, due to the fact that FPFH algorithm needs to exhaustively search the entire data structure to find neighboring points to compute local descriptors. Even though our model consists of complex deep learning architecture, it achieves similar run time as the simple fiducial marker method thanks to the speed boost of GPU. Moreover, our model achieves the least average distance error, and outperforms the other two methods by a large margin, proving the effectiveness of our deep learning model.

To further investigate details of the evaluation process, we count the percentage of testing samples with their distance errors less than predefined thresholds. We choose 10 different thresholds from 0.01m to 0.1m with an interval of 0.01m, and plot their corresponding percentages in Fig. 7. From this plot, we can clearly see how the distance error is distributed among testing samples for different methods. For example, about 50% of testing samples have distance errors less than 0.01m for our method, the geometric method completely fails to estimate reasonable pose parameters possibly because of the severe noise in captured point cloud data, and the number of samples is roughly uniformly distributed within each error range for marker-based method. The pose estimation results are also qualitatively compared in Fig. 8. We show both the 2D projection and 3D point cloud of the transformed machine model. The qualitative results match the quantitative results.

[2]https://pytorch.org

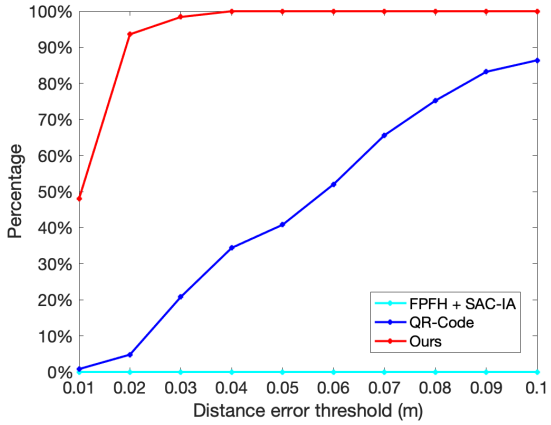


Fig. 7. The percentage of testing samples with their average distances less than different thresholds.

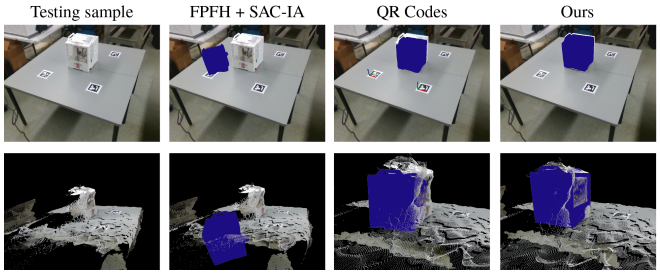
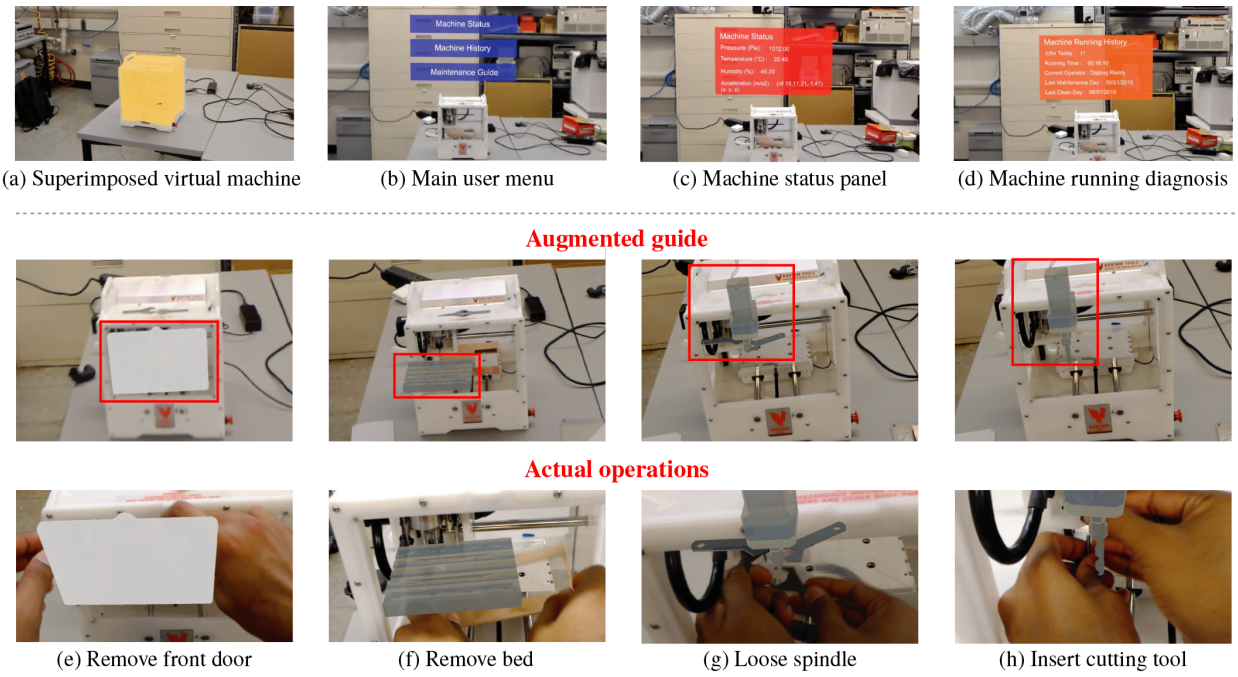


Fig. 8. Visualize pose estimation results (Top row: Project transformed the machine model onto scene images; Bottom row: Transform the machine model into scene point clouds). Note that QR codes shown in the images are only used for marker-based method.

B. System Demonstration

We demonstrate two use cases for our system: machine status monitoring and augmented maintenance guide.

1) Machine Status Monitoring: Sensor module we installed on the machine continuously transmits measured temperature, pressure, humidity, accelerometer data to our server. Visualizations through AR lens are updated in real-time with the new sensor information. Operators can read the data in real-time and notice any deviations from the normal mode of function. Fig. 9 (a) shows superimporsed visual projection of the machine over the real object in user's field of view. This helps the user know that the machine of interest is selected by the Hololens. User then sees options in the field of view to either view machine's status, history or maintenance guide



(a) Superimposed virtual machine (b) Main user menu (c) Machine status panel (d) Machine running diagnosis

(e) Remove front door (f) Remove bed (g) Loose spindle (h) Insert cutting tool

Fig. 9. Top row: Machine information monitoring and visualization; Bottom two rows: Augmented maintenance guide.

(Fig. 9 (b)). Fig. 9 (c)-(d) show machine's state and diagnosis information from the recorded sensor data to the operator to take further actions.

2) Augmented Maintenance Guide: Another functionality in our prototype is augmented maintenance assistance to the user. Traditionally, user go through the technical manual to train themselves on repairing a machine. Augmented maintenance guide accelerates this process by providing immersive visual sequences of the maintenance procedure to the user. We used CAD models of the spare parts to create a prototype version of augmented maintenance guide for our desktop mill. Fig. 9 (e) - (h) show a sequence of steps to change the tool. For example, the AR-maintenance guide projects safety covers on to the actual object and indicates a remove motion to let the user know that the next step is to remove the safety covers. This follows by indication to remove the work piece from the bed. Next steps shows where the wrench should be held against the spindle to correctly remove the tool and replace it with a new one. Although a physical technical documentation can layout such instructions in step-by-step manner, the advantage of AR-based maintenance guide is presenting these instructions through the use of visual overlays, CAD models and motion graphics so that the users learn with clarity and quickly.

V. Conclusion and Discussion

In this work, we presented an IoT-AR system that integrates advanced deep learning methods for accurate object pose estimation. We showed that the proposed pose estimation model achieves much better accuracy than commonly used methods in existing AR systems, and also satisfies real-time performance requirement. We further demonstrated our prototype on two use cases: machine status monitoring and augmented maintenance guide. Evaluation results prove the effectiveness of our system, and show the trend of fusing cutting-edge technologies for smart factory applications.

In this work, we only demonstrate the prototype system on one milling machine. But the pose estimation framework can be extended for multiple objects by including an additional classification branch and estimating the pose for each potential candidate. Yet, when more target machines are considered, more manual work is required to obtain ground truth poses. To resolve the scalability issue for industry development, an automatic way for pose learning will be a potential research direction.

References

[1] Daniel Wagner and Dieter Schmalstieg. Design aspects of handheld augmented reality games, 2018.

[2] Francesca Bonetti, Gary Warnaby, and Lee Quinn. Augmented reality and virtual reality in physical and online retailing: A review, synthesis and research agenda. In *Augmented reality and virtual reality*, pages 119–132. Springer, 2018.

[3] Yongbin Sun, Sai Nithin R Kantareddy, Rahul Bhattacharyya, and Sanjay E Sarma. X-vision: An augmented vision tool with real-time sensing ability in tagged environments. In *2018 IEEE International Conference on RFID Technology & Application (RFID-TA)*, pages 1–6. IEEE, 2018.

[4] G. Pappas, J. Siegel, and K. Politopoulos. VirtualCar: Virtual Mirroring of IoT-Enabled Avacars in AR, VR and Desktop Applications. In Tony Huang, Mai Otsuki, Myriam Servires, Arindam Dey, Yuta Sugiura, Domna Banakou, and Despina Michael-Grigoriou, editors, *ICAT-EGVE 2018 - International Conference on Artificial Reality and Telexistence and Eurographics Symposium on Virtual Environments - Posters and Demos*. The Eurographics Association, 2018.

[5] Yongbin Sun, Alexandre Armengol-Urpi, Sai Nithin Reddy Kantareddy, Joshua Siegel, and Sanjay Sarma. Magichand: Interact with iot devices in augmented reality environment. In *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pages 1738–1743. IEEE, 2019.

[6] Eric Marchand, Hideaki Uchiyama, and Fabien Spindler. Pose estimation for augmented reality: a hands-on survey. *IEEE transactions on visualization and computer graphics*, 22(12):2633–2651, 2015.

[7] Rafael Radkowski. A point cloud-based method for object alignment verification for augmented reality applications. In *ASME 2015 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, pages V01BT02A059–V01BT02A059. American Society of Mechanical Engineers, 2015.

[8] Martin Sundermeyer, Zoltan-Csaba Marton, Maximilian Durner, Manuel Brucker, and Rudolph Triebel. Implicit 3d orientation learning for 6d object detection from rgb images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 699–715, 2018.

[9] Fabian Manhardt, Wadim Kehl, Nassir Navab, and Federico Tombari. Deep model-based 6d pose refinement in rgb. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 800–815, 2018.

[10] Dimitris Mourtzis, Ekaterini Vlachou, Vasilios Zogopoulos, and Xanthi Fotini. Integrated production and maintenance scheduling through machine monitoring and augmented reality: An industry 4.0 approach. In *IFIP International Conference on Advances in Production Management Systems*, pages 354–362. Springer, 2017.

[11] Evangelos Tzimas, George-Christopher Vosniakos, and Elias Matsas. Machine tool setup instructions in the smart factory using augmented reality: a system construction perspective. *International Journal on Interactive Design and Manufacturing (IJIDeM)*, 13(1):121–136, 2019.

[12] Michele Gattullo, Giulia Wally Scurati, Michele Fiorentino, Antonio Emmanuele Uva, Francesco Ferrise, and Monica Bordegoni. Towards augmented reality manuals for industry 4.0: A methodology. *Robotics and Computer-Integrated Manufacturing*, 56:276–286, 2019.

[13] Mario José Castellanos and Andrés A Navarro-Newball. Prototyping an augmented reality maintenance and repairing system for a deep well vertical turbine pump. In *2019 International Conference on Electronics, Communications and Computers (CONIELECOMP)*, pages 36–40. IEEE, 2019.

[14] CY Siew, SK Ong, and AYC Nee. A practical augmented reality-assisted maintenance system framework for adaptive user support. *Robotics and Computer-Integrated Manufacturing*, 59:115–129, 2019.

[15] Francisco J Romero-Ramirez, Rafael Muñoz-Salinas, and Rafael Medina-Carnicer. Speeded up detection of squared fiducial markers. *Image and Vision Computing*, 76:38–47, 2018.

[16] Valentin Markus Josef Heun. *The reality editor: an open and universal tool for understanding and controlling the physical world*. PhD thesis, Massachusetts Institute of Technology, 2017.

[17] Bruce Thomas, Benjamin Close, John Donoghue, John Squires, Phillip De Bondi, Michael Morris, and Wayne Piekarski. Arquake: An outdoor/indoor augmented reality first person application. In *Digest of Papers. Fourth International Symposium on Wearable Computers*, pages 139–146. IEEE, 2000.

[18] Zulqarnain Rashid, Joan Melià-Seguí, Rafael Pous, and Enric Peig. Using augmented reality and internet of things to improve accessibility of people with motor disabilities in the context of smart cities. *Future Generation Computer Systems*, 76:248–261, 2017.

[19] Hirokazu Kato and Mark Billinghurst. Marker tracking and hmd calibration for a video-based augmented reality conferencing system. In *Proceedings 2nd IEEE and ACM International Workshop on Augmented Reality (IWAR'99)*, pages 85–94. IEEE, 1999.

[20] Ronald Azuma, Yohan Baillot, Reinhold Behringer, Steven Feiner, Simon Julier, and Blair MacIntyre. Recent advances in augmented reality. *IEEE computer graphics and applications*, 21(6):34–47, 2001.

[21] Mark Fiala. Designing highly reliable fiducial markers. *IEEE Transactions on Pattern analysis and machine intelligence*, 32(7):1317–1324, 2009.

[22] Alvaro Collet, Manuel Martinez, and Siddhartha S Srinivasa. The moped framework: Object recognition and pose estimation for manipulation. *The International Journal of Robotics Research*, 30(10):1284–1306, 2011.

[23] Vittorio Ferrari, Tinne Tuytelaars, and Luc Van Gool. Simultaneous object recognition and segmentation from single or multiple model views. *International Journal of Computer Vision*, 67(2):159–188, 2006.

[24] Fred Rothganger, Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. 3d object modeling and recognition using local affine-invariant image descriptors and multi-view spatial constraints. *International Journal of Computer Vision*, 66(3):231–259, 2006.

[25] Menglong Zhu, Konstantinos G Derpanis, Yinfei Yang, Samarth Brahmbhatt, Mabel Zhang, Cody Phillips, Matthieu Lecce, and Kostas Daniilidis. Single image 3d object detection and pose estimation for grasping. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3936–3943. IEEE, 2014.

[26] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[27] Xian-Feng Hana, Jesse S Jin, Juan Xie, Ming-Jie Wang, and Wei Jiang. A comprehensive review of 3d point cloud descriptors. *arXiv preprint arXiv:1802.02297*, 2018.

[28] Paul J Besl and Neil D McKay. Method for registration of 3-d shapes. In *Sensor Fusion IV: Control Paradigms and Data Structures*, volume 1611, pages 586–607. International Society for Optics and Photonics, 1992.

[29] Szymon Rusinkiewicz and Marc Levoy. Efficient variants of the icp algorithm. In *3dim*, volume 1, pages 145–152, 2001.

[30] Shubham Tulsiani and Jitendra Malik. Viewpoints and keypoints. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1510–1519, 2015.

[31] Yu Xiang, Wongun Choi, Yuanqing Lin, and Silvio Savarese. Subcategory-aware convolutional neural networks for object proposals and detection. In *2017 IEEE winter conference on applications of computer vision (WACV)*, pages 924–933. IEEE, 2017.

[32] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. *arXiv preprint arXiv:1711.00199*, 2017.

[33] Shuran Song and Jianxiong Xiao. Sliding shapes for 3d object detection in depth images. In *European conference on computer vision*, pages 634–651. Springer, 2014.

[34] Shuran Song and Jianxiong Xiao. Deep sliding shapes for amodal 3d object detection in rgb-d images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 808–816, 2016.

[35] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 652–660, 2017.

[36] Charles R Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J Guibas. Frustum pointnets for 3d object detection from rgb-d data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 918–927, 2018.

[37] Stefan Hinterstoisser, Stefan Holzer, Cedric Cagniart, Slobodan Ilic, Kurt Konolige, Nassir Navab, and Vincent Lepetit. Multimodal templates for real-time detection of texture-less objects in heavily cluttered scenes. In *2011 international conference on computer vision*, pages 858–865. IEEE, 2011.

[38] Wadim Kehl, Fausto Milletari, Federico Tombari, Slobodan Ilic, and Nassir Navab. Deep learning of local rgb-d patches for 3d object detection and 6d pose estimation. In *European Conference on Computer Vision*, pages 205–220. Springer, 2016.

[39] Danfei Xu, Dragomir Anguelov, and Ashesh Jain. Pointfusion: Deep sensor fusion for 3d bounding box estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 244–253, 2018.

[40] Bantam Tools. Bantam tools desktop pcb milling machine specifications. *Othermill*, 2018.

[41] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[42] Chen Wang, Danfei Xu, Yuke Zhu, Roberto Martín-Martín, Cewu Lu, Li Fei-Fei, and Silvio Savarese. Densefusion: 6d object pose estimation by iterative dense fusion. *arXiv preprint arXiv:1901.04780*, 2019.

[43] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *arXiv preprint arXiv:1801.07829*, 2018.

[44] Radu Bogdan Rusu, Nico Blodow, and Michael Beetz. Fast point feature histograms (fpfh) for 3d registration. In *2009 IEEE International Conference on Robotics and Automation*, pages 3212–3217. IEEE, 2009.