Pose initialization method of mixed reality system for inspection using convolutional neural network

Yong Hwi KIM* and Kwan H. LEE*
*School of Mechanical Engineering, Gwangju Institute of Science and Technology (GIST)
123 Cheomdangwagiro, Bukgu, Gwangju, Korea
E-mail: khlee@gist.ac.kr

Received: 2 March 2019; Revised: 22 April 2019; Accepted: 22 May 2019

Abstract

The Mixed Reality (MR) has become a trend in industrial applications such as inspection and maintenance thanks to the benefit of technological advances in computer vision. Simultaneous Localization And Mapping (SLAM) is a key component of the MR system which augments the CAD model of a target object in the live stream. However, the existing SLAM-based systems rely on a manual handling or a marker-based registration between the model coordinate and the global coordinate. In this paper, we present a non-marker based registration method which automatically performs both the target object detection in the live stream and its initial 3D pose estimation. We exploit two Convolutional Neural Networks (CNNs) to align the CAD model in a global map, and to detect the target object in keyframes of the SLAM system. Since manual preparation of training data is very laborious, we also propose a data augmentation method for the industrial application. The data augmentation method generates a synthesized dataset consisting of pairs of the RGB image and the corresponding camera pose using the object's CAD model. Two CNNs for the object detection in keyframes and the initial pose estimation are trained with the synthesized dataset, respectively. Our result shows that this method can robustly find the target object's initial pose without a dense point cloud or other features detected by hand-crafted descriptors.

Keywords: Mixed reality, Pose initialization, Deep learning, Simultaneous localization and mapping (SLAM), Process inspection

1. Introduction

The Mixed Reality (MR) has become a trend in industrial applications. Recent technological advances in the computer vision technology enable us to retrieve virtual contents in real captured images through various types of displays such as Head-Mounted Display (HMD), and mobile devices. Thanks to mobility and real-time performance of these displays, the MR system has been changing the paradigm of the model inspection and maintenance (Schall et al., 2009; Eschen et al., 2018).

In the MR system, initial registration between the system coordinate and the virtual environment is essential in order to find the relevant pose of a virtual object in the system coordinate. Earlier work has performed it by using fiducial markers (Bleser et al., 2005; Wang et al., 2005; Fiorentino et al., 2014), or descriptors oriented by the characteristics of the target environment (Carozza et al., 2014). However, they require a laborious task to accurately position physical markers or to understand the underlying scene attributes. Both procedures are undesirable for the industrial construction complex which changes quickly. By utilizing the Simultaneous Localization And Mapping (SLAM) on object tracking, markerless-based methods have been proposed. Mostly, they use active sensors acquiring a rich information of the target environment, such as a RGBD sensor (Macedo et al., 2013), a 3D rider sensor (Bosche et al., 2010), or a sensor-embedded system (Schall et al. 2009; Martin et al. 2015; Greenhalgh et al. 2016). These sensors are robust to the scale-ambiguity problem (Mur-Artal et al., 2017) commonly encountered in a monocular-based SALM, but it requires a heavy computation for the registration of dense point clouds in real-time application.

In this paper, we present a markerless-based registration method which automatically performs both the target object detection in the live stream and the estimation of its initial pose. Our method aims not to use any active sensor so that it
can be run in a commercial mobile device (e.g., smart-phone, or table PC). We solve the scale ambiguity problem by utilizing the CNN-based pose estimator which is trained by the synthesized dataset. Manual preparation of the training data manually is very laborious since, in general, it requires hundred-thousand images to train such a CNN for the pose estimation. Therefore, we propose a data augmentation process which automatically generates still images of the target object at the simulated camera pose. The training dataset generated by the data augmentation process consists of possible pairs of a camera motion and the corresponding 2D RGB image in reference to the target object's CAD model.

In contrast to the existing method which exploits hand-craft features for the initial registration of the target object (Carozza et al., 2014; Fiorentino et al., 2014), our method can efficiently estimate the initial pose of the target object in the system coordinates by selecting Region-of-Interests (ROIs) in keyframes, which are automatically selected by another CNN for the object detection. We also perform the fine tuning process that further optimizes the initial pose of the target object using the Iterative Closest Point (ICP) algorithm. For demonstration, we have built a proof-concept SLAM system that tracks a spool-like industrial object in real time. Our results show that our method can robustly find the target object's initial pose without a dense point cloud or other features detected by hand-crafted descriptors. In summary, our contributions are as follows:

- We propose a markerless initial registration method for a monocular-based SLAM using CNN-based pose estimator.
- We propose a data augmentation process to efficiently generate the training dataset.

2. Related work

2.1 Offline calibration

Since the target object is generally pre-defined for inspection, the offline calibration has been widely used in the initial pose estimation. From the knowledge of the scene surrounding the target object or object’s attributes such as geometry, previous work designed a hand-crafted descriptor which is suitable for a given environment (Carozza et al., 2014; Alvarez et al., 2011). However, the performance of the hand-crafted descriptor is limited to a similar environment for which they are designed. Marker-based systems can greatly reduce the complexity of the registration process by adding known features on the scene (Bleser et al., 2005; Fiorentino et al., 2014). However, they require accurate manual positioning of physical markers. Furthermore, it is not desirable to install physical markers in the construction site in which continuous changes of a scene occurs. In contrast to the aforementioned methods, our method does not require prior assumptions for the environment and does not rely on physical markers as well.

2.2 Online calibration

The majority of markerless approaches uses a depth sensor for the pose estimation. Macedo et al. (2013) performed 3D face tracking using a RGBD sensor to augment virtual contents on the frame. Bosche et al. (2010) proposed an offline registration method which merges partial observations from a 3D radar sensor, into a global coordinate. Greenhalgh et al. (2016) developed a HMD-based system which displays a 3D CAD model at the current head position in real-time. These methods can capture a dense 3D information of the target environment but they require a special equipment, and a high performance computing system due to the registration between dense point clouds in the system coordinates and the CAD model. The monocular-based SLAM system is somewhat free to these limitations since it estimates 3D points in the system coordinates directly from RGB-frames. Several methods have exploited a monocular based SLAM for registration. Petit et al. (2013) improved the stability of the monocular SLAM by detecting color edges and combines them with the geometry of a target object. Alvarez et al. (2011) also improved the quality of camera pose estimation by hashing geometric information derived from object's CAD model. Similarly, our method uses the given CAD model to supply scene information for the registration process. However, it does not rely on a prior shape of a feature detector when estimating the initial pose of the target object. It allows us to perform a robust pose estimation for the object which has a variety of variation in the surface texture.

2.3 CNN-based pose estimation

The 3D pose estimation directly from an image of the target object is an active research field in CNN-based deep learnings (Mane et al., 2017). Mehta et al. (2017) estimated human's position from a single 2D image using a multi-modal 3D pose prediction model. Miao et al. (2016) employed a set of CNN regressors extracting local features from a X-ray...
image and performed the 2D-3D matching process. Similarly, Tian et al. (2017) estimated the GPS location of a user from a single street view by a CNN-based local matching technique. Massa et al. (2016) improved the accuracy of the 2D-3D matching by exploiting adaptive features between a real image dataset and a synthetic image set generated from object's CAD model.

Although our CNN-based pose estimation performs a similar task for estimating the 3D position of the reference camera from a query 2D image, we have designed a novel approach to employ CNNs for the initial pose estimation of the target object in a real-time SLAM system.

3. Overview

Our method uses two CNNs as illustrated in Fig. 1: one for the target object detection in a 2D image and one another for the initial pose estimation of the CAD model. Our method is divided into two major processes named as the ‘offline’ and the ‘online’. In the offline process, Two CNNs are trained using a dataset generated by the data augmentation process (Sec. 4). The online process indicates the real-time process of the MR system augmenting the virtual contents in the image frame. In the online process, we use the pre-trained CNNs to detect the target object in keyframes of the SLAM (Sec. 5), and to estimate the initial 3D position of the target object (Sec. 6). If the object has been detected in the keyframe by the trained CNN, then the MR system marks the bounding box, as the Region-of-Interest (ROI), where the object is located in, and segments it on the 2D frame. After the object detection process, the pose estimation CNN finds an initial registration between the object coordinate and the system coordinate using point clouds estimated from the segmented ROI image. The prior distribution for the camera motion in the training dataset is not perfectly matched with the actual
camera motion. Therefore, we further optimize the estimated initial registration using a standard Iterative Closest Point (ICP) algorithm.

### 3.1 Assumptions

We assume that the 3D CAD model of the target object is given. This assumption is based on the common scenario in industrial applications such as inspection and maintenance, in which an operator wants to query the CAD model of the work environment. Another assumption is that the camera is specified in advance so that we have the intrinsicities of the camera used in the MR system. The target object is presumed to be a rigid body. In that case, the registration process can be simplified as finding a rigid-body transformation matrix which is expressed as a $4 \times 4$ matrix. The object should be non-symmetric. If the object is symmetric (e.g. perfect rectangle), there are ambiguities to estimate a unique camera position from a 2D image as a projection of the 3D CAD model.

### 3.2 Camera model

We adopt a simple pinhole model as the projection model:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x & 0 \\ 0 & f_y & c_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} r_x x' + t_x \\ r_y y' + t_y \\ r_z z' + t_z \end{bmatrix} = \begin{bmatrix} r_x x + t_x \\ r_y y + t_y \\ r_z z + t_z \end{bmatrix}$$

where $(x',y')$ is the corresponding 2D pixel point in the image coordinate, $(x,y,z)$ is a 3D point in the system coordinate, $f_x,f_y$ are the focal length of the camera in x- and y-coordinates, and $c_x,c_y$ are the camera center locations in the image coordinate. The transformation matrix which is the second $4 \times 4$ matrix in the Eq. (1) contains the 3×3 rotation matrix denoted as $r$ and the 3×1 translation vector denoted as $t$. Assuming that the camera is specified, $f_x,f_y$ and $c_x,c_y$ can be given in advance using a feature-based calibration method (Bouguet et al. 2004). In our method, the MR system finds the rotation matrix as well as the translation vector for each image frame. In order to successfully plot a virtual object according to the motion of the camera, we first need to estimate the initial registration matrix.

A 3D point in the CAD model of the target object can be projected onto the image frame by the following equation:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x & 0 \\ 0 & f_y & c_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \hat{r}_x x + \hat{t}_x \\ \hat{r}_y y + \hat{t}_y \\ \hat{r}_z z + \hat{t}_z \end{bmatrix} = \begin{bmatrix} \hat{r}_x x + \hat{t}_x \\ \hat{r}_y y + \hat{t}_y \\ \hat{r}_z z + \hat{t}_z \end{bmatrix}$$

where $\hat{r}, \hat{t}$ denote the element of the rotation matrix and the translation vector in the registration matrix which is the third matrix in the right hand side of Eq. (2). For simplicity, we denote the transformation matrix as $T_{trans}$, and the registration matrix as $T_{reg}$ in the remaining sections.

### 4. Data augmentation

#### 4.1 Motion simulation

To generate the training data, we simulate the camera motion in the hemispherical domain for a given range of the distance between object's center and the camera position. Without the loss of generality, we further restrict the orientation of the camera as the forward axis points against to the object's center. The orientation of the camera can be fixed as:

$$\begin{align*}
\hat{w} &= \text{norm}(\{x_c,y_c,z_c\}^T) \\
\hat{u} &= \text{cross}(\{0,0,1\}^T, \hat{w}) \\
\hat{v} &= \text{norm}(\text{cross}(\hat{w}, \hat{u}))
\end{align*}$$

(3)
where \( \vec{w}, \vec{u}, \) and \( \vec{v} \) are the forward, right, and up vector of the camera coordinate, and \([x_c, y_c, z_c]^T\) is the vector of camera center defined in the system coordinate.

### 4.2 Parameter sampling and data representation

The domain of the camera motion defined in the radial coordinate \((d, \theta, \phi)\) where \(x_c = dsin(\theta)cos(\phi)\), \(y_c = dsin(\theta)sin(\phi)\), and \(z_c = dcos(\theta)\) is randomly sampled. Table 1 shows the parameter domain of the camera motion.

From the pair of the synthesized image and the corresponding camera pose, we generate the input tensor \( I(n \times w \times h) \) and the ground truth camera position matrix \( O(n \times 6) \) where \( n \) is the number of generated images, \( w, h \) are the width and height of the image. Each row of \( O \) contains the three Euler angles of the rotation matrix, and the translation vector of the generated camera motion. Since the absolute magnitude of Euler angles in \( O \) is relatively small to the absolute magnitude of the translation values \( x_c, y_c, z_c \) in \( O \), we perform a normalization of each column space of \( O \) so that each column space has zero-mean and unit-variance. This normalization step prevents from producing a biased solution which ignores the variation of Euler angles in camera motion.

### 4.3 Data augmentation

The data augmentation process has two major steps. At first, we generate a virtual image at each simulated camera motion using a renderer-based simulator (Fig. 2 left). Generated images are, of course, not sufficient to train CNNs since its color distribution is very distinct from a real images of the target object. We secondly perform the data augmentation process using the Cycle GAN (Zhu et al. 2017). The Cycle GAN allows us to transfer the color distribution of a target dataset to a reference distribution. To create the reference distribution, we use a video sequence of the target object which is taken by the same camera with the one used in the MR system. In experiments, we observed that about one-minute video captured around the target object is sufficient to transfer the style of images generated by the renderer.

Figure 2 shows the result of the data augmentation process. By the forward and the inverse mapping function between

---

**Table 1 Parameter domain of the camera motion. Parameters are sampled by a uniform distribution \( \Omega \) for each parameter.**

| Parameter           | Domain          |
|---------------------|-----------------|
| Camera distance (m) | \(d \sim \Omega(1.5, 2.5)\) |
| Elevation angle (rad)| \(\theta \sim \Omega(0, \pi/4)\) |
| Azimuth angle (rad) | \(\phi \sim \Omega(0, 2\pi)\) |
the synthetic dataset and the video sequence, we can make the generated images more realistically using the reference distribution of the video sequence. Training time for the Cycle GAN is shown in Table 2.

5. Object detection using CNN

In this method, the objection detection indicates the starting point of the registration process. The registration must be performed right after the target object is appeared in a live stream. Because the performance of the registration between two point clouds is crucially dependent on the structural closeness of two point clouds in terms of the density, the geometrical similarity, and its scale. If we can determine which frame the target object appears, the registration process can be much easier than the case that uses the entire point clouds acquired by the MR system. Obviously, it is much simple to estimate the transformation matrix between a partial point cloud in a specific area and the target CAD model rather than between the whole point cloud acquired from all RGB frames and the target CAD model.

Therefore, we adopt the Mobilenet-based CNN for object detection in a RGB frame. Our choice of the CNN is based on two characteristics of the MR system: First, we have the specific target object so that we do not need a complex training process for a massive dataset containing a variety of different objects. Second, the object detection has to be performed within a short period of time since the tracking in the MR system runs in real-time. Considering these facts, we selected a less-weighted CNN model designed to perform in real-time on a mobile device.

As shown in Fig. 1, the Mobilenet-based CNN is trained using our training dataset. One benefit of our dataset is that we can also have the ground truth bounding box indicating where the object is located in the image coordinate (Fig. 3 (a)). From the synthesized dataset and its ground truth mask map, we can train the network and apply them to the MR system. For every keyframe selected by the MR system, the trained model estimates the pixel location of the target object. If it is detected with a high probability, then the pose initialization CNN is activated to estimate the initial pose of the CAD model.

Figure 3 (right) shows an example of object detection. It predicts not only the probability of the existence of the target object in a frame, but the location of the object. From this information, we can select a partial point cloud which only contains the points corresponding to the surface of the target object. 3D points matched with a 2D pixel within the selected region are separately stored in the tracking system.

6. Pose initialization using CNN

Figure 4 illustrates the structure of the CNN for the pose estimation. The network has three convolutional layers and three dense layers attached to the last convolutional layer. For each convolutional layer, the 'ReLU' activation layer and the 'Max-pooling' layer are sequentially followed. The size of the input is gradually decreased while being passed through the network. Features from the last convolution layer are flattened to be passed to the dense layer. We also attach the 'Dropout' layer for each dense layer. The size of the last dense layer is identical to the DOF of the camera motion. We use
the 'Adam' optimizer with the mean square error between $O$ and its outputs.

Using the predicted camera motion $T'_{\text{trans}}$ by the pose estimation CNN, we can estimate $T_{\text{reg}}$ by the following equation:

$$T_{\text{reg}} \approx sT^{-1}_{\text{trans}}T'_{\text{trans}}$$

(4)

where $T^{-1}_{\text{trans}}$ is the inverse of the actual camera motion of the MR system, and $s$ is the scale factor which reflects the scale difference between CAD model and point cloud in the system coordinate. We can adjust $s$ by the characteristic of our training dataset: All the images in the dataset are defined in the absolute scale since we utilize its actual CAD model in the data augmentation process as mentioned in Sec. 4. Using this characteristic, we define $s$ as the difference between the distance between the absolute point and the camera center where $s \approx d_c/d_c'$. Figure 5 shows the initial pose estimation result using the ROI selection in a point cloud. After the ROI selection (Fig. 5 (b)), we can further perform fine registration using the ICP in order to refine $T_{\text{reg}}$. The ICP optimizes the following objective function:

$$R, \tilde{t} = \arg \min_{R, \tilde{t}} \sum_{p \in \Omega_{\text{reg}}} \|p - Rp' - \tilde{t}\|$$

(5)

where $p'$ is a 3D point in the object coordinate, $p$ is the corresponding 3D point in the system coordinate, and $R$, $\tilde{t}$ are the rotation matrix and the translation vector for fine registration. Let $T_{\text{ICP}}$ denote the transformation matrix from the ICP algorithm. $T_{\text{reg}}$ is now re-defined as

$$T_{\text{reg}} \approx sT_{\text{ICP}}T^{-1}_{\text{trans}}T'_{\text{trans}}$$

(6)
In general, the performance of the ICP algorithm is highly dependent on the initial pose of a target model. However, our initial position estimation using the pose estimation CNN leads to a globally-optimal solution with an estimated camera position. A simple nearest-neighborhood selection can be adapted in this case where the target model is matched with a partial point cloud.

7. Results

7.1 Test environment

For validation, we collected three structures in a test environment as shown in Fig. 6. The ‘Spool’ object was installed on an indoor location and its 3D CAD model was drawn using a CAD tool. The ‘Tank’ and ‘Control panel’ objects were installed on a mechanical facility and their CAD models were acquired in advance by a hand-held IR-based scanner. The scale of the CAD models is identical to the actual size of the corresponding objects so that we can retrieve object's scale in the monocular SLAM by performing the pose estimation described in Sec. 6.

We generated 18000 images of the CAD model with pre-calibrated camera intrinsics. The data augmentation task was performed before training two networks. For data augmentation, we captured a video sequence (640 × 480) for the target object using a commercial tablet (Samsung Galaxy Tab3). For computational efficiency, we resized the images as (320 × 240), only when training the pose estimation CNN. The batch size and the training rate were set to 24, 0.004, respectively. The number of iterations of two CNNs was empirically chosen as $2 \times 10^4$ for object detection and $7.5 \times 10^4$ for pose estimation, where both CNNs converged sufficiently. The computation time of each process is summarized in Table 2. Although the offline process requires heavy training of CNNs, the performance of online processes is fast enough to be used in a real-time MR system. By the initial registration from the pose estimation CNN, we can reduce a wrong neighborhood selection in the ICP. As a result, the computation time of the ICP was greatly reduced to near 1 second.

![Image of the target objects](a) Spool (b) Tank (c) Control panel

![Graphs](a) Euclidean distance between the estimated camera center and the ground truth camera center in meters according to the number of training images (m), (b) mean radian difference of three Euler angles (deg), (c) training time (min).
7.2 Reliability of the pose estimation CNN

We tested the prediction error of the pose estimation CNN according to the number of synthesized images for the ‘Spool’ object. In this test, we evaluated three metrics on the test dataset whose images in the dataset are not included in the training set. Three evaluation metrics are defined as the mean Euclidean distance between the estimated camera center and the actual camera center generated from the uniform distribution (Table 1), the mean angle difference between each coordinate, and the training time. The test dataset contains 2000 images generated by the same method for constructing the training dataset.

The training time linearly increases as the number of images in the training dataset increases (Fig. 7 (c)). Whereas the distance and the angle difference decreases according to the number of images, but they tend to be converged to a certain error level. Considering the computation efficiency, it is desired to find a sweet-spot between the average errors and the training time. In this paper, we used 18000 images for training both CNNs.

7.3 MR system

We fully accommodated the monocular version of the ‘ORB SLAM2’ (Mur-Artal et al., 2017) for a real-time tracking and a localization (Fig. 8). In this system, the pose initialization step is computed as follows. Once a keyframe has been selected in a runtime, the object detection model is activated to find the target object. If the target object is appeared in the keyframe at the first time, the system runs the pose initialization model in parallel with the SLAM algorithm. While performing the initial pose estimation, the actual camera transformation matrix is stacked into a $4 \times 4$ matrix so that we can instantly map the 3D model after the pose initialization has been finished. As described in Sec. 6, the fine registration process is performed with a selected point cloud. By employing two independent CNNs to the SLAM system, we can resolve two major problems in the monocular SLAM: scale-ambiguity and ROI selection. As shown in Fig. 9, our method robustly projects the 3D CAD model to the image frame even in the monocular case. Although we observed a small drift of the CAD model to the reference object, we believe that it is originated from the design error between the physical model and the 3D CAD model.

---

**Table 2 Computation time of each process.** All the processes were performed on a commercial PC (i7-8700k, 8GB RAM, GTX 1080Ti).

| Type   | Process                  | Time (s) |
|--------|--------------------------|----------|
|        |                          | Spool    | Tank    | Control panel |
| offline| Image generation         | 508.1    | 454.5   | 485.7         |
|        | Data augmentation        | 63122.2  | 49057.4 | 22814.6       |
|        | Training object detection CNN | 6640.3  | 6563.1  | 6625.1       |
|        | Training pose estimation CNN | 3621.7  | 3664.4  | 3611.3       |
| online | Object detection         | 0.3      | 0.4     | 0.3           |
|        | Pose estimation          | 0.1      | 0.1     | 0.1           |
|        | Fine registration        | 1.1      | 0.8     | 0.7           |

---

**Fig. 8** Diagram of the MR system. For demonstration, we have employed the feature-based SLAM on the MR system. Our pose initialization step is performed in each keyframe selected by the MR system. Once it finds a robust registration of the target object, the viewer displays virtual contents in the RGB frame.

[DOI: 10.1299/jamdsm.2019jamdsm0093] © 2019 The Japan Society of Mechanical Engineers
7.4 Limitations

There are several limitations in our method. First, it is impossible to render the exactly same image on the real environment, the distribution generated by synthesized images can be quite far from actual camera motions. In that case, our system becomes fragile. Second, the activation function of the object detection CNN can be biased by the background of the object, especially for a thin object as shown in Fig. 6. Lastly, we use an approximation solution for refining the scale factor $s$ which is not derived from a full analysis of the distribution of point cloud. It could fail to estimate an exact scale between the point cloud and the CAD model when there are many outliers or irrelevant feature points in the point cloud. An improved ICP method regarding the scale difference (Zinsser et al. 2005) can be applied to our pose estimation pipeline.

8. Conclusion

In this paper, we have presented a novel framework which automatically estimates the initial pose of a target object. We have employed two CNN-based models for pose estimation and object detection. We have also proposed a data augmentation process which can generate a fair amounts of training images using the Cycle GAN. In our demonstration of the MR system, we have shown that our method robustly aligns a virtual object to a real-time RGB frame without intervention of the user and physical markers.

In future work, we would like to seek geometric features which are invariant to the change of illuminations or materials of the object. This work will reduce the complexity of image-based object detection as well as the pose estimation by transforming the input data into a much smaller data space.

Acknowledgement

We would like to thank to reviewers for constructive feedback. This work was supported by GIST Research Institute (GRI) grant funded by the GIST in 2018.
References

Alvarez H., Aguinaga I., Borro D., Providing guidance for maintenance operations using automatic markerless augmented reality system, In Proceedings of the IEEE International Symposium on Mixed and Augmented Reality (2011), pp.181-190.

Blaser G., Pastarmov V., Stricker D., Real-time 3d camera tracking for industrial augmented reality applications (2005), UNION Agency.

Bosche F., Automated recognition of 3D CAD model objects in laser scans and calculation of as-built dimensions for dimensional compliance control in construction, Advanced engineering informations, Vol.24 (2010), No.1, pp.107-118.

Bouguet JY., Camera calibration toolbox for matlab (online), available from <http://www.vision.caltech.edu/bouguetj/calib_doc/index.html>, (accessed on 28 February, 2019).

Carozza L., Tingdahl D., Bosche F., Van Gool L., Markerless vision-based augmented reality for urban planning, Computer-Aided Civil and Infrastructure Engineering, Vol.29 (2014), No.1, pp.2-17.

Eschen H., Kotter T., Rodeck R., Harnisch M., Schuppstuhl T., Augmented and virtual reality for inspection and maintenance processes in the aviation industry, Procedia manufacturing, Vol.19 (2018), pp.156-163.

Fiorentino M., Uva AE., Gattullo M., Debernardis S., Monno G., Augmented reality on large screen for interactive maintenance instructions, Computers in Industry, Vol.65 (2014), No.2, pp.270-278.

Greenhalgh P., Mullins B., Grunnet-Jepsen A., Bhowmik AK., Industrial deployment of a full-featured head-mounted augmented-reality system and the incorporation of a 3D-sensing platform (2016).

Macedo MCdF., Apolinario Junior AL., Souza ACdS., KinectFusion for faces: real-time 3D face tracking and modeling using a kinect camera for a markerless AR system (2013), SBC.

Mane D., Kulkarni U., A survey on supervised convolutional neural network and its major applications, International Journal of Rough Sets and Data Analysis (IJRSDA), Vol.4 (2017), No.3, pp.71-82.

Martin-Gutierrez J., Fabiani P., Benesova W., Meneses MD., Mora CE., Augmented reality to promote collaborative and autonomous learning in higher education, Computers in Human Behavior, Vol.51 (2015), pp.752-761.

Massa F., Russell BC., Aubry M., Deep exemplar 2d-3d detection by adapting from real to rendered views, In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2016), pp.6024-6033.

Mehta D., Rhodin H., Casas D., Fua P., Sotnychenko O., Xu W., et al. Monocular 3d human pose estimation in the wild using improved cnn supervision, International Conference on 3D Vision (3DV) (2017), pp.506-516.

Miao S., Wang ZJ., Liao R., A CNN regression approach for real-time 2D/3D registration, IEEE transactions on medical imaging, Vol.35 (2016), No.5, pp.1352-1363.

Mur-Artal R., Tardos JD., Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d Cameras, IEEE Transactions on Robotics, Vol.33 (2017), No.5, pp.1255-1262.

Petit A., Marchand E., Kanani K., Augmenting markerless complex 3D objects by combining geometrical and color edge information, In Proceedings of the IEEE International Symposium on Mixed and Augmented Reality (2013), pp.287-288.

Schall G., Mendez E., Kruij E., Veas E., Junghanns S., Reitinger B., et al., Handheld augmented reality for underground infrastructure visualization, Personal and ubiquitous computing, Vol.13, No. 4 (2009), pp.281-291.

Tian Y., Chen C., Shah M., Cross-view image matching for geo-localization in urban environments, In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2017), pp.3608-3616.

Wang X., Kotranza A., Quares J., Lok B., Allen BD., A pipeline for rapidly incorporating real objects into a mixed environment, In Proceedings of the 4th IEEE/ACM International Symposium on Mixed and Augmented Reality, IEEE Computer Society (2005), pp.170-173.

Zhu JY., Park T., Isola P., Efros AA., Unpaired image-to-image translation using cycle-consistent adversarial networks, In Proceedings of the IEEE International Conference on Computer Vision (2017), pp.2223-2232.

Zinber T., Schmidt J., Niemann H., Point set registration with integrated scale estimation, International conference on pattern recognition and image processing (2005), pp.116-119.