Augmented reality (AR) is a technology that overlays digital content (graphics, text, audio, and video) on real-world images using mobile devices, PCs or glasses, which have an integrated camera and specialized software [9].

Every AR systems detect the presence of one or more objects of interest, estimate their positions, and continuously track their movements [11]. Object tracking plays a crucial role in making the augmented scene look more natural. Tracking continuously computes and provides the position of objects, regardless of environment conditions [10]. Without accurate tracking, the coexistence and coherence of the virtual world and the real world are disturbed [9].

Object tracking is a challenging problem in computer vision applications, affected by factors like lighting, camera movement, occlusion, and other aspects inherent to the image acquisition system and the environment [8, 9]. However, the occlusion problem is considered one of the most critical issues [8, 19]. Occlusion occurs when one object overlaps another, partially or entirely, causing loss of information that makes the user’s interaction with the system troublesome [8, 19]. This inaccuracy breaks the illusion of the augmented scene and can confuse user perception, nullifying system fluidity, and the basic principle of AR [8].

In order to solve the occlusion problems in tracking objects in AR systems, a possible approach is through the inclusion of recent artificial intelligence technologies as machine learning [9, 13, 16].

AR is an area that involves different techniques and technologies, which includes a variety of hardware and software systems during its operation [5, 10]. So, for its proper functioning, this must be characterized mainly by a) combining real and virtual objects, b) being interactive in real-time, and c) aligning 3D objects [1].

To accomplish this, AR systems must accurately align virtual objects with real objects and continuously track their positions in real-time. Thus, tracking techniques are essential elements of every AR system [1, 10].

However, several problems may occur during object tracking. These problems change the appearance of the object in the real scene, such as: occlusion (partial or total obstruction in the actual environment), lighting (variations and/or light effects), motion blur (noise image or fast motion blur), scale or distance (limited range between camera and actual object) and latency (system delays) [1, 5].

The problem of occlusion is considered relevant, since, in the case of incorrect treatment of occlusions, the whole fundamental principle of AR is disturbed. Objects will not be correctly aligned, tracking fluidity will be lost, depth perception may turn out wrong, and all illusion of coexistence between the real world and the virtual world will be nullified [8, 9].

The literature on object tracking is extensive; thus, this work focus only on recent works on machine learning-based approaches to tackle occlusion problems.

In [18], a machine learning-based tracking algorithm using Random Forests is presented to handle partial occlusions, illumination changes, and motion blur. In [15], the authors present a method based on convolutional neural networks (CNN) to 3D object detection and pose estimation of an object. In [3], an algorithm for estimating the pose of a rigid object in real-time under challenging conditions such as cluttered and changing environments, or large occlusions, is presented. Finally, the authors in [6] present a 6 degrees-of-freedom
(6DoF) tracking method which leverages deep learning to achieve state-of-the-art performance on challenging real-world datasets. The presented method is both, more accurate and more robust to deal with occlusions than the existing best performing approaches while maintaining real-time performance.

Another approach is presented in [20] for face recognition with partial occlusions. The authors proposed a robust LSTM-Autoencoder model to restore partially occluded faces, demonstrating efficacy in removing different types of facial occlusion.

3. Proposal
A possible approach to minimizing problems in objects tracking is the inclusion of emerging technologies such as wearable computing, the Internet of Things, or machine learning, that can help solve existing challenges and limitations in AR systems [9, 13, 16].

Machine learning provides algorithms that learn how to perform new tasks when exposed to new data without being explicitly reprogrammed [9]. So, this technology provides a promising prospect, since the literature still presents a few studies that include new machine learning-based methods for AR systems.

Our approach aims to develop a computational method for object tracking that handles occlusion based on machine learning. One of the methods to be initially explored are Generative Adversarial Nets (GANs). GANs are a framework proposed in 2014 composed of two competing artificial neural networks, a generating network, and a discriminating network. The architecture of GANs is illustrated in Figure 1, where the Generator aims to generate new data similar to the expected one, and the Discriminator aims to recognize if an input data is “real” belonging to the original data or if it is “fake” generated by a forger [7].

\[
\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_{\text{data}}(x)}[\log D(x)] + \\
\mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(z)))]
\] (1)

where, \(D\) seeks to maximize objective such that \(D(x)\) is close to 1 for real data, and \(D(G(z))\) is close to 0 to fake data, and \(G\) seeks to minimize objective such that \(D(G(z))\) is close to 1 [7, 14].

GANs are generative models, so the first application is for data generation. Some of the most well-known applications are in the area of computer vision, Natural Language Processing, medicine, and security [14].

Our goal is to harness the potential of data generation techniques, to restore or reconstruct partially occluded object for tracking in AR systems using GAN architectures.

The literature presents approaches to the reconstruction of images based on GANs. In [4], the author presents a Context-Conditional Generative Adversarial Networks (CC-GANs), where the Generator is trained to fill in a missing image patch, and the Generator and Discriminator are conditioned to the surrounding pixels. In [2], the author implements this model (CC-GANs) to the reconstruction of the partially occluded object, as shown in Figure 2.

4. Partial Results
For the first study, we chose the Vuforia SDK\(^1\) (software development kit) and we developed an evaluation demo prototype shown in Figure 3. This prototype allowed us to explore the performance and potential of markerless tracking, even under conditions where the object of interest was not fully visible [17].

In a second study, feature detection and extraction techniques were applied to the development of a basic markerless AR system. This system was able to detect a reference image and track it, as shown in Figure 4. The prototype allowed us to explore that, in conditions where the reference image was not completely visible, feature matching and tracking in the AR system failed.

\(^1\)Vuforia Engine: https://developer.vuforia.com/
As a third study, machine learning techniques were applied to minimize feature matching problems through classification techniques. A basic markerless AR system was developed to detect and track a reference image, as shown in Figure 5. The prototype allowed us to explore that through classification techniques, features matching problems diminished, but under conditions where the reference image was partially visible, tracking in the AR system failed.

To implement our proposal, several requirements are necessary, such as training and testing data, objects with occlusion, RGB data sets, and ground truth tracking data. Therefore, datasets with different occlusion levels (partial or high), RGB and RGBD data, and with pose 6DoF or bounding box were evaluated [12].

Finally, the dataset available in [6] was selected. This dataset is composed of four objects, captured by a Kinect v2, under different levels of occlusion (10%–40% of the object occluded). Also, this work presents an accurate, real-time temporal 6DoF object tracking method, applying a CNN, which is more robust to occlusions than existing state-of-the-art algorithms.

Currently, we are reproducing their work, seeking to reach the same results reported in [6]. The first stage was to generate the entire set of data for training, as shown in Figure 6, through a rendering-based method to generate the data necessary to train a deep network.

The second stage consisted of train the network with the dataset of 250,000 sample images generated. Figure 7 shows the loss function evolution for every training step for both, the training and the validation datasets. The training was performed on the CCES-UNICAMP Cluster Kahuna 2, which features an Intel Xeon E5-2670 v2 2.50GHz with 40 cores HT, 64G memory and an Nvidia Tesla K20M.

5. Future Work

Once the results will be obtained, we use the dataset [6] to compare the average error quantitatively in translation and rotation in three different situations: tracking without occlusion, tracking with occlusion, and tracking with occlusion (reconstruct occluded image – our proposal).

GANs are promissory for reconstructing occluded images. We will use the dataset images to implement the approach explained in Section 3, and to validate [3]http://cces.unicamp.br/computing-resources/
our proposal to reconstruct images under occlusion conditions, the images of the object will be partially occluded. Afterward, it is expected that the proposed system can be able to reconstruct the occluded areas of the object. Finally, these reconstructed images will be used for the pose estimation of the reference object and its corresponding tracking.

References


