Artificial Neural Networks to Recognize ARToolKit Markers

Celso Camilo Jr
Federal University of Grande Dourados
Faculty of Exact Sciences and Technology
Km 12 Rodovia Dourados – Itahum, Dourados, MS.
+55 (67) 3411-3894
celsocamilo@gmail.com

Weinlton Luiz Gomes
Federal University of Uberlandia
Faculty of Electrical Engineering
Campus Santa Monica, Uberlandia
MG, Brazil
+55 (34) 3239-4165
wneiton@yahoo.com.br

Alexandre Cardoso
Federal University of Uberlandia
Faculty of Electrical Engineering
Campus Santa Monica, Uberlandia
MG, Brazil
+55 (34) 3239-4165
alexandre@ufu.br

Keiji Yamanaka
Federal University of Uberlandia
Faculty of Electrical Engineering
Campus Santa Monica, Uberlandia
MG, Brazil
+55 (34) 3239-4165
keiji@ufu.br

Edgard Lamounier
Federal University of Uberlandia
Faculty of Electrical Engineering
Campus Santa Monica, Uberlandia
MG, Brazil
+55 (34) 3239-4165
lamounier@ufu.br

ABSTRACT
The objective of this work is to present the implementation of an Artificial Neural Network associated to ARToolKit for the recognition of markers, providing a less complex algorithm in comparison to the traditional algorithm used in the ARToolKit system.

Keywords
Pattern Recognition, Artificial Neural Network, Augmented Reality, ARToolKit.

1 INTRODUCTION
There are plenty application problems where an ANN can be applied, but they are mainly used to pattern classification and optimization [1].

Augmented Reality is defined as a combination of a vision of the real environment associated to a virtual one [4, 5, 6]. In this technique, a video image is processed in such way that it is “augmented” with graphic scenes generated by computer [7]. Most of the low cost AR projects are developed using a set of computer libraries known as ARToolKit [8].

Associated to the capacity offered by ANNs, this work has the objective to improve ARToolKit’s ability to recognize markers (patterns) in a graphic image. Since the capture of a new scene and the insertion of new virtual objects must be performed in real time for AR techniques, there is a constant need to reduce computations time.

Thus, this paper proposes an implementation of a different ARToolKit library where ANNs are used to provide a better performance during the pattern recognition process.

This work is organized as follows: Section 2 presents basic concepts of the ARToolKit system and how marker recognition takes place. Section 3 introduces ANN concepts plus its time complexity algorithms. The conclusions of this work are presented in Section 4.

2 THE ARToolKit SYSTEM
The ARToolKit system is a C library developed at Washington University [1]. Through computer vision techniques, the system calculates the position and orientation of markers or patterns printed in cards. This position and orientation are captured by a digital camera and when the image is reprocessed, virtual objects are placed on these markers in the new scene (see Figure 1) [1].

![Figure 1. ARToolKit main flow [8].](image)

2.1 ARToolKit’s Valid Marker Recognition
The entire ARToolKit’s recognition process of valid markers is ranged from the identification of the picture inside the square border to the association of virtual objects with these pictures. All steps to set a marker is valid can be seen in Figure 2.

![Figure 2. Markers Recognition Process [8].](image)
3 USE ANN IN THE ARToolKit's MARKERS RECOGNITION

ARToolKit has a stage (see step 2 in 2.1) for valid patterns recognition. However, this work proposes the substitution of the used algorithm, for recognition, by a ANN (Figure 3), propitiating an acting earnings.

![Diagram of steps](image)

**Figure 3.** Match (ARToolKit) substitution for ANN for valid markers recognition.

As example, a scenery was created where was used three markers: Trace, Kaije and Hiro. This way, the net has 256 entrance neurons and three exit neurons. The technique used for training was the delta rule, the $\alpha$ used is 0.0039 and the stop condition is Quadratic Error or equal to $0.1$. The $\alpha$ is the learning rate used in the net, that most of the time is randomly defined. The $\alpha$ defines the step size for the net learning. The Quadratic Error is the errors addition of each exit neuron.

In the pre-execution phase the markers in the Figure 4 were presented to Mk-Patt that created a file for each marker with the 16x16 matrices. These three files formed the training base that was presented to the ANN for learning. The ANN got the Quadratic Error of 0.09998 after 1745 cycles.

![Markers](image)

**Figure 4.** Markers used in the ANN training.

3.1 Results of the algorithms complexity analysis

The algorithm analysis is an important part of the algorithm complexity theory, that provides theoretical estimates off necessary resources for any computation problem. These estimates impel an algorithms efficiency improvement [9,10,11].

It was made the complexity analyses of the Match and ANN algorithms, and the execution time analysis of each line. The Table 1.0 displays the execution time calculations of the Match algorithm.

![Code snippet](image)

**Figure 5.** ANN algorithm for ARToolKit markers.

The table 2.0 displays the execution time calculations of the ANN algorithm.

![Table](image)
### Table

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>(257 \times m)</th>
<th>(514 \times m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>2</td>
<td>(256 \times m)</td>
<td>(512 \times m)</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>(m)</td>
<td>(1 \times m)</td>
</tr>
<tr>
<td><strong>Total execution time</strong></td>
<td></td>
<td>((1029 \times m) + 2)</td>
<td></td>
</tr>
</tbody>
</table>

The tables 1.0 and 2.0 are formed by 4 columns:
- **Line**: determinates the algorithm line that is being analyzed;
- **Line Cost**: specifies the cost based on the executed operations number in the line. And, each operation (addition, subtraction, multiplication, division, comparison) have cost equal to one. For example, the line 11 of the Figure 9, where an addition exists (\(\text{ave} + =\)) and a subtraction (255 - data[i]), have two costs. Reminding that, the attribution doesn't have cost;
- **Repetition**: amount of times that the line is executed;
- **Total Line Time**: it is the result of the equation Line Cost \(\times\) Repetition.

### 4 CONCLUSIONS AND FUTURE WORK

Based on the obtained results, we concluded that the algorithm using artificial neural networks recognizes the markers just as the recognition module used by ARToolKit. However, in spite of both algorithms present a time complexity of \(O(m)\), where \(m\) represents the number of markers, the proposed algorithm is faster (execution time) than the ARToolKit’s Match algorithm. Thus, the proposed algorithm provides a better performance in the recognition process of the markers, which is very important for real time AR applications.

As future work tests involving a larger number of markers will be taking into account in order to measure the performance of both algorithms in these situations. Furthermore, tests where the AR markers present imperfect conditions will be considered to evaluate the algorithm quality in the recognition process.

### 5 REFERENCES


