# Cuneiform Symbols Recognition Using Intensity Curves

Hilal Yousif[1], Abdul Munim Rahma[2], and Haithem Alani[3]
[1]Al-Rafidian University College, Iraq
[2]Computer Science Department, University of Technology, Iraq
[3]College of Science, University of Alnahrain, Iraq

**Abstract:** *The cuneiform symbol recognition is a vital step in automated reading and understanding the contents of tens of thousands cuneiform tablets available all over the world. Reading cuneiform records depends largely on hand written copies of cuneiform tablets for their data. A suggested method for cuneiform symbol recognition from a hand written images of cuneiform text. The method make use of the fact that there are a finite number of images for the symbols and trying to differentiate between them making use of the intensity profile curves which represent the intensities of selected pixels in the images.*

**Keywords:** *Cuneiform, intensity, recognition, image, symbol.*

*Received January 12, 2004; accepted May 162005*

## 1. Introduction

One of the greatest contribution of early civilization was writing. The first efforts at writing were pictures. The Sumerians developed the first writing known system. The Babylonians and Assyrians employed a system of writing to which the term cuneiform has been applied, cuneiform, from the Latin cuneus, meaning wedge, is the term applied to a mode of writing which is used as wedge-shaped stylus to make impressions on a clay surface and also on stone, metal and wax. Most clay tablets were sun-baked, making surviving tablets very fragile [4].

The form of the characters employed consists of various combinations of the upright wedge, the horizontal wedge, and the diagonal wedge, which frequently interchanges with slopping wedge. The characters are written left to right. This system of writing uses symbols for ideas rather than sounds, and is used to record business activities, but also tablets containing medical texts and other subjects have also been found. Cuneiform eventually developed into syllabic alphabet under the Assyrians and Babylonians who come to dominate the area about the middle of third millennium. Cuneiform script was used to represent texts in several languages in the ancient near East from the end of the fourth millennium until the first millennium BC.

Tens of thousands of clay tablets have been discovered, the researchers have a great attention in studying these tablets to discover there contents and the information they hold. A great number of symbols have been used, historians suggested that number came to 800 in mid 3000 BC.

Cuneiform writing is a mixture of figurative method and phonetic logical method. Because of the difficulty of cuneiform writing and because of the great number of symbols, those who dealt with cuneiform writing used lists contain the cuneiform signs and their figurative and phonetic signs [6]. Astrologists and cuneiform experts, write faithful copies of the original cuneiform text (from the tablets or inscriptions) by hand, they do not use any fonts in their papers and books for presenting the original source [2].

A large part of cuneiform tablets have never been studied or even read by scholars. Copying, deciphering, and publishing their contents, and checking that work, is slow and difficult because among 6.3 billion people in the world, may be only 300 read cuneiform, and their expert work is not the relatively straightforward process [3].

## 2. Reading Cuneiform Documents

The information recorded in cuneiform documents can be studied either directly, through first hand examination of originals (which is almost always not available for most people), or indirectly, by means of second-hand representations, at present time, such representations follow one of two approaches:

- Pictorial (photographs and hand copies) or
- Interpretative (translations and transcriptions).

At first, reading cuneiform record, depended largely on hand copies of cuneiform tablets for their data. Computerized cuneiform symbol recognition is an important matter in the subject of reading and

understanding the contents of the cuneiform tablets. As there are a great number of cuneiform tablets which has been copied by hand (hand written) the concern will constraint on a cuneiform sign recognition for hand written images [7].

In order to automate the recognition of the cuneiform sign, we need to define a definite form for it. As the cuneiform sign developed for more than 3000 years and the set has been modified and changed over years for many reasons out of our concern in this research. Although the method we developed has been applied to the final form of the cuneiform symbols (which is the Assyrians) but it can be applied to other set of cuneiform writing in the same way.

There is a similarity between the hand written sign on paper copied from inscribed one on the clay tablets that can be used in future to develop the method to be applied on images of inscribed sign. Figure 1 shows a cuneiform sign copied and its original image on the clay [1].

(a) Drawing of a sign.

(b) The way it looks in clay.

Figure 1. An image of a cuneiform sign and its copy.

## 3. Cuneiform Symbols Recognition Algorithms

The suggested method is built making use of the fact that there are a limited number of cuneiform symbols. Each cuneiform sign according to the suggested method has its own finger print which is a sequence of integers calculated by the method making use of its image characteristics, the finger print of all signs are stored in a signs data base which is used in sign recognition.

Two algorithms were used for the suggested method, these are:

- The image finger print calculation algorithm.
- The image search algorithm.

### 3.1. The Image Finger Print Calculation Algorithm

The finger print of the image consists of two vectors (V1 and V2) is calculated by the following steps:

1. Draw the intensity curve for the image in level y1, which is the curve of intensities of all pixels of the cuneiform symbol image at height y1, where the vertical axis represents the intensity value of the pixel and the horizontal axis represents the location of the pixel (the column number). Figure 2 shows an example of symbol image, Figure 3 shows the intensity curve at the specified level (y1).
2. Record locations of local minimum in the intensity curve (points a1, a2…).
3. Normalize the values a1, a2, .. by dividing the horizontal range of the image into m divisions each interval is represented by a number in vector V1.
4. Add one to the corresponding entry in V1 for each (a) value.
5. Repeated steps 1 to 4 for different levels y2, y3, ...
6. Numbers in vector V2 is calculated in the same way explained in steps (1 to 5) but for vertical lines x1, x2, ...

   The following diagram in Figure 4 shows the part of the algorithm for calculating the V1 vector, a similar diagram is used for calculating V2 vector.

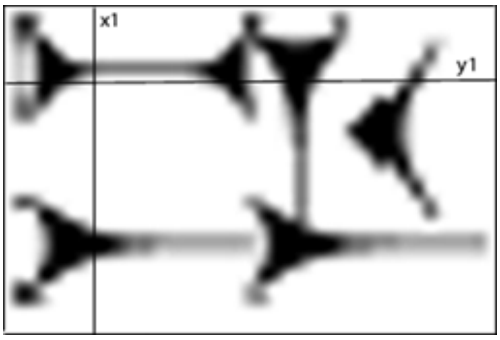


Figure 2. An image of cuneiform symbol.

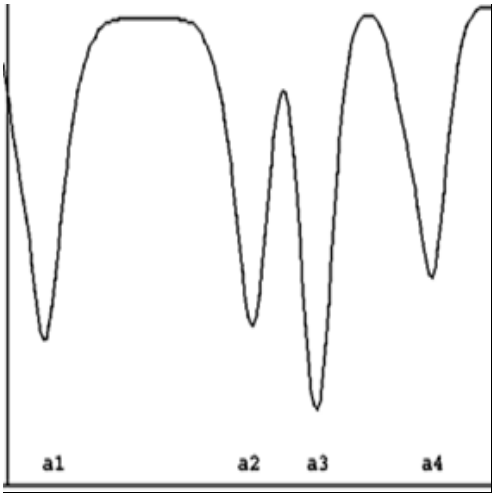


Figure 3. The intensity curve at the specified level (y1).

### 3.2. Cuneiform Symbol Image Recognition Algorithm

Applying the algorithm in 3.1 above on all the cuneiform symbols images, and storing the results in a two dimensional array (B), each row in the array contains V1 and V2 of finger print of a specific cuneiform symbol image. Identifying the symbol in a

given cuneiform symbol image is done by the following steps:

1. Producing the finger print of the given image using steps explained in 3.1 above, store V1 and V2 of the finger print in array A.
2. Calculating the total of the absolute difference between each entry in A and the corresponding entry in the first row in B. store the result in C [1].
3. Repeating step 2 for each row in B. i. e.,

$$[i] = \sum_{j=0}^{\text{no of entry-1}} \text{abs}\ (B[i][j] - A[j])$$

Where:

i: The row number, which contains the finger print of a cuneiform symbol image.
j: An entry in finger print.
Abs: An absolute difference function.

4. The index of entry of C which contains the smallest value is the number of image that best fit the given image.

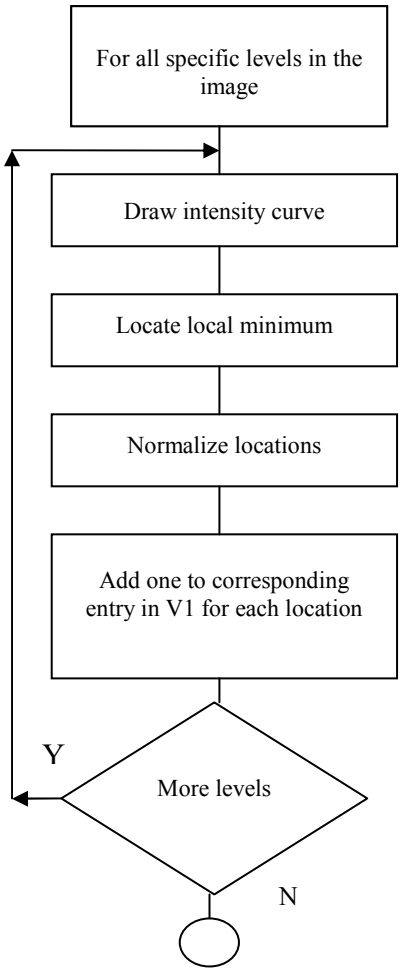


Figure 4. Calculating V1 steps.

The table in Figure 5 shows an example of the implementation of the suggested algorithm, the third entry in array c which contains the minimum value, signify that the third image is the most fit image to image in array A.

The calculation of the c[2] in the example is as follows:

c[2] = abs (3 - 2) + abs (1 - 0) + abs (1 - 1) + abs (1 - 1) + abs (0 - 1).
= 1 + 1 + 0 + 0 +1
= 3

| | | | | |
|---|---|---|---|---|
| **2** | **0** | **1** | **1** | **1** |

| | | | | | |
|---|---|---|---|---|---|
| **0** | **1** | **2** | **2** | **0** | **6** |
| **3** | **1** | **1** | **1** | **0** | **3** |
| **2** | **0** | **0** | **1** | **1** | **1** |
| **3** | **1** | **0** | **0** | **2** | **5** |
| **0** | **0** | **1** | **4** | **1** | **5** |
| **1** | **1** | **0** | **2** | **1** | **4** |

Figure 5. An example of implementing the algorithm.

Note that step (3) of calculating the finger print of the image (normalization step) minimizes the effect of changing the size of an image values in its finger print. (i. e., the same finger prints (or almost the same) would be produced for two images of the same symbol with different sizes).

## 4. Experimental Results

A total of 500 handwritten cuneiform symbols extracted from images of cuneiform tablets copied in different books especially cuneiform signs used in LABAT manual [5] was used to construct the database. The suggested system could successfully recognize slightly above 90% of the symbols.

A relatively small effort was spent to study the effect of noise in the images although the suggested system shows a good percent of success, especially with a salt and pepper noise and the success percent increased simply by increasing number of levels at which intensity curves is to be drawn (both in database construction and symbols recognition). The results achieved in recognition of cuneiform symbols can be improved if additional characteristics of the image symbol is used together with intensity curves (e. g., ratio of length and width, counting the number of changing the color.. etc.).

The following tables are the result of implementing the algorithm in cases as explained below:

1. Table in Figure 6 shows the results of implementing the algorithm in searching for complete identical image with the stored images1.
2. Table in Figure 7 shows the results of implementing the algorithm in searching for complete identical image but magnified with the stored images.
3. Table in Figure 8 shows the results of implementing the algorithm in searching for complete identical image but reduced size with the stored images.

4. Table in Figure 9 shows the results of implementing the algorithm in searching for distorted image with the stored images.

| A | 1 | 3 | 2 | 1 | 2 | 2 | 3 | 1 | 3 | 1 | 1 | 2 | 1 | 0 | 3 | 0 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | | | | | | | | C |
| 1 | 0 | 0 | 3 | 0 | 0 | 1 | 2 | 0 | 2 | 1 | 2 | 0 | 2 | 0 | 1 | 1 | | 19 |
| 2 | 0 | 1 | 0 | 2 | 2 | 3 | 3 | 0 | 2 | 1 | 0 | 2 | 1 | 1 | 0 | 0 | | 14 |
| 3 | 1 | 2 | 2 | 2 | 1 | 3 | 3 | 0 | 2 | 2 | 0 | 0 | 2 | 2 | 0 | 0 | | 16 |
| 4 | 0 | 2 | 3 | 2 | 3 | 2 | 1 | 3 | 0 | 0 | 2 | 2 | 1 | 1 | 3 | 1 | | 16 |
| 5 | 0 | 3 | 2 | 2 | 0 | 3 | 4 | 4 | 0 | 0 | 3 | 2 | 3 | 0 | 1 | 1 | | 20 |
| 6 | 1 | 2 | 2 | 2 | 1 | 0 | 4 | 3 | 0 | 0 | 1 | 3 | 1 | 1 | 3 | 1 | | 15 |
| 7 | 0 | 1 | 1 | 3 | 1 | 3 | 0 | 0 | 1 | 2 | 0 | 2 | 0 | 0 | 0 | 0 | | 20 |
| 8 | 0 | 1 | 1 | 3 | 3 | 1 | 3 | 0 | 2 | 3 | 0 | 1 | 0 | 1 | 0 | 0 | | 19 |
| 9 | 0 | 3 | 1 | 1 | 1 | 0 | 3 | 0 | 0 | 0 | 4 | 0 | 0 | 3 | 0 | 0 | | 22 |
| 10 | 1 | 1 | 0 | 4 | 3 | 2 | 0 | 2 | 0 | 2 | 1 | 1 | 1 | 3 | 0 | 0 | | 23 |
| ≈ | | | | | | | | | | | | | | | | | | ≈ |
| 65 | 0 | 2 | 1 | 2 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 2 | 2 | 1 | 1 | 0 | | 18 |
| 66 | 1 | 1 | 2 | 2 | 0 | 3 | 4 | 4 | 0 | 1 | 1 | 0 | 2 | 2 | 1 | 1 | | 21 |
| 67 | 0 | 3 | 1 | 2 | 0 | 1 | 3 | 0 | 0 | 3 | 0 | 1 | 2 | 0 | 3 | 0 | | 15 |
| 68 | 1 | 3 | 2 | 1 | 2 | 2 | 3 | 1 | 3 | 1 | 1 | 2 | 1 | 0 | 3 | 0 | | 0 |
| 69 | 0 | 2 | 2 | 3 | 1 | 3 | 1 | 4 | 3 | 2 | 0 | 1 | 0 | 1 | 3 | 0 | | 16 |
| 70 | 1 | 3 | 0 | 1 | 3 | 1 | 3 | 0 | 0 | 2 | 0 | 2 | 1 | 1 | 1 | 0 | | 13 |
| ≈ | | | | | | | | | | | | | | | | | | ≈ |
| …. | 4 | 1 | 2 | 3 | 1 | 4 | 1 | 2 | 3 | 0 | 3 | 1 | 1 | 1 | 0 | 0 | | 21 |
| …. | 1 | 2 | 1 | 2 | 0 | 2 | 3 | 3 | 0 | 1 | 2 | 1 | 3 | 1 | 2 | 2 | | 18 |
| …. | 1 | 1 | 1 | 3 | 0 | 1 | 3 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | | 18 |
| …. | 1 | 3 | 3 | 1 | 0 | 4 | 0 | 0 | 1 | 1 | 1 | 1 | 2 | 1 | 1 | 1 | | 17 |

Figure 6. Results of implementing the algorithm in searching for complete identical image.

| A | 1 | 3 | 2 | 1 | 1 | 3 | 3 | 1 | 2 | 2 | 1 | 2 | 1 | 0 | 3 | 0 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | | | | | | | | C |
| 1 | 0 | 0 | 3 | 0 | 0 | 1 | 2 | 0 | 2 | 1 | 2 | 0 | 2 | 0 | 1 | 1 | | 19 |
| 2 | 0 | 1 | 0 | 2 | 2 | 3 | 3 | 0 | 2 | 1 | 0 | 2 | 1 | 1 | 0 | 0 | | 14 |
| 3 | 1 | 2 | 2 | 2 | 1 | 3 | 3 | 0 | 2 | 2 | 0 | 0 | 2 | 2 | 0 | 0 | | 12 |
| 4 | 0 | 2 | 3 | 2 | 3 | 2 | 1 | 3 | 0 | 0 | 2 | 2 | 1 | 1 | 3 | 1 | | 18 |
| 5 | 0 | 3 | 2 | 2 | 0 | 3 | 4 | 4 | 0 | 0 | 3 | 2 | 3 | 0 | 1 | 1 | | 18 |
| 6 | 1 | 2 | 2 | 2 | 1 | 0 | 4 | 3 | 0 | 0 | 1 | 3 | 1 | 1 | 3 | 1 | | 15 |
| 7 | 0 | 1 | 1 | 3 | 1 | 3 | 0 | 0 | 1 | 2 | 0 | 2 | 0 | 0 | 0 | 0 | | 16 |
| 8 | 0 | 1 | 1 | 3 | 3 | 1 | 3 | 0 | 2 | 3 | 0 | 1 | 0 | 1 | 0 | 0 | | 19 |
| 9 | 0 | 3 | 1 | 1 | 1 | 0 | 3 | 0 | 0 | 0 | 4 | 0 | 0 | 3 | 0 | 0 | | 22 |
| 10 | 1 | 1 | 0 | 4 | 3 | 2 | 0 | 2 | 0 | 2 | 1 | 1 | 1 | 3 | 0 | 0 | | 23 |
| ≈ | | | | | | | | | | | | | | | | | | ≈ |
| 65 | 0 | 2 | 1 | 2 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 2 | 2 | 1 | 1 | 0 | | 18 |
| 66 | 1 | 1 | 2 | 2 | 0 | 3 | 4 | 4 | 0 | 1 | 1 | 0 | 2 | 2 | 1 | 1 | | 19 |
| 67 | 0 | 3 | 1 | 2 | 0 | 1 | 3 | 0 | 0 | 3 | 0 | 1 | 2 | 0 | 3 | 0 | | 13 |
| 68 | 1 | 3 | 2 | 1 | 2 | 2 | 3 | 1 | 3 | 1 | 1 | 2 | 1 | 0 | 3 | 0 | | 4 |
| 69 | 0 | 2 | 2 | 3 | 1 | 3 | 1 | 4 | 3 | 2 | 0 | 1 | 0 | 1 | 3 | 0 | | 14 |
| 70 | 1 | 3 | 0 | 1 | 3 | 1 | 3 | 0 | 0 | 2 | 0 | 2 | 1 | 1 | 1 | 0 | | 13 |
| ≈ | | | | | | | | | | | | | | | | | | ≈ |
| …. | 4 | 1 | 2 | 3 | 1 | 4 | 1 | 2 | 3 | 0 | 3 | 1 | 1 | 1 | 0 | 0 | | 21 |
| …. | 1 | 2 | 1 | 2 | 0 | 2 | 3 | 3 | 0 | 1 | 2 | 1 | 3 | 1 | 2 | 2 | | 18 |
| …. | 1 | 1 | 1 | 3 | 0 | 1 | 3 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | | 18 |
| …. | 1 | 3 | 3 | 1 | 0 | 4 | 0 | 0 | 1 | 1 | 1 | 1 | 2 | 1 | 1 | 1 | | 15 |

Figure 7. Results of implementing the algorithm in searching for magnified image.

| | 1 | 3 | 2 | 1 | 1 | 2 | 3 | 1 | 3 | 1 | 1 | 2 | 1 | 0 | 3 | 0 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | | | | | | | | C |
| 1 | 0 | 0 | 3 | 0 | 0 | 1 | 2 | 0 | 2 | 1 | 2 | 0 | 2 | 0 | 1 | 1 | | 18 |
| 2 | 0 | 1 | 0 | 2 | 2 | 3 | 3 | 0 | 2 | 1 | 0 | 2 | 1 | 1 | 0 | 0 | | 15 |
| 3 | 1 | 2 | 2 | 2 | 1 | 3 | 3 | 0 | 2 | 2 | 0 | 0 | 2 | 2 | 0 | 0 | | 15 |
| 4 | 0 | 2 | 3 | 2 | 3 | 2 | 1 | 3 | 0 | 0 | 2 | 2 | 1 | 1 | 3 | 1 | | 17 |
| 5 | 0 | 3 | 2 | 2 | 0 | 3 | 4 | 4 | 0 | 0 | 3 | 2 | 3 | 0 | 1 | 1 | | 19 |
| 6 | 1 | 2 | 2 | 2 | 1 | 0 | 4 | 3 | 0 | 0 | 1 | 3 | 1 | 1 | 3 | 1 | | 14 |
| 7 | 0 | 1 | 1 | 3 | 1 | 3 | 0 | 0 | 1 | 2 | 0 | 2 | 0 | 0 | 0 | 0 | | 19 |
| 8 | 0 | 1 | 1 | 3 | 3 | 1 | 3 | 0 | 2 | 3 | 0 | 1 | 0 | 1 | 0 | 0 | | 20 |
| 9 | 0 | 3 | 1 | 1 | 1 | 0 | 3 | 0 | 0 | 0 | 4 | 0 | 0 | 3 | 0 | 0 | | 21 |
| 10 | 1 | 1 | 0 | 4 | 3 | 2 | 0 | 2 | 0 | 2 | 1 | 1 | 1 | 3 | 0 | 0 | | 24 |
| ≈ | | | | | | | | | | | | | | | | | | ≈ |
| 65 | 0 | 2 | 1 | 2 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 2 | 2 | 1 | 1 | 0 | | 17 |
| 66 | 1 | 1 | 2 | 2 | 0 | 3 | 4 | 4 | 0 | 1 | 1 | 0 | 2 | 2 | 1 | 1 | | 20 |
| 67 | 0 | 3 | 1 | 2 | 0 | 1 | 3 | 0 | 0 | 3 | 0 | 1 | 2 | 0 | 3 | 0 | | 14 |
| 68 | 1 | 3 | 2 | 1 | 2 | 2 | 3 | 1 | 3 | 1 | 1 | 2 | 1 | 0 | 3 | 0 | | 1 |
| 69 | 0 | 2 | 2 | 3 | 1 | 3 | 1 | 4 | 3 | 2 | 0 | 1 | 0 | 1 | 3 | 0 | | 15 |
| 70 | 1 | 3 | 0 | 1 | 3 | 1 | 3 | 0 | 0 | 2 | 0 | 2 | 1 | 1 | 1 | 0 | | 14 |
| ≈ | | | | | | | | | | | | | | | | | | ≈ |
| …. | 4 | 1 | 2 | 3 | 1 | 4 | 1 | 2 | 3 | 0 | 3 | 1 | 1 | 1 | 0 | 0 | | 20 |
| …. | 1 | 2 | 1 | 2 | 0 | 2 | 3 | 3 | 0 | 1 | 2 | 1 | 3 | 1 | 2 | 2 | | 17 |
| …. | 1 | 1 | 1 | 3 | 0 | 1 | 3 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | | 17 |
| …. | 1 | 3 | 3 | 1 | 0 | 4 | 0 | 0 | 1 | 1 | 1 | 1 | 2 | 1 | 1 | 1 | | 16 |

Figure 8. Results of implementing the algorithm in searching for reduced size image.

| A | 1 | 4 | 2 | 1 | 2 | 3 | 3 | 1 | 3 | 1 | 2 | 2 | 1 | 0 | 4 | 0 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | | | | | | | | C |
| 1 | 0 | 0 | 3 | 0 | 0 | 1 | 2 | 0 | 2 | 1 | 2 | 0 | 2 | 0 | 1 | 1 | | 21 |
| 2 | 0 | 1 | 0 | 2 | 2 | 3 | 3 | 0 | 2 | 1 | 0 | 2 | 1 | 1 | 0 | 0 | | 16 |
| 3 | 1 | 2 | 2 | 2 | 1 | 3 | 3 | 0 | 2 | 2 | 0 | 0 | 2 | 2 | 0 | 0 | | 18 |
| 4 | 0 | 2 | 3 | 2 | 3 | 2 | 1 | 3 | 0 | 0 | 2 | 2 | 1 | 1 | 3 | 1 | | 18 |
| 5 | 0 | 3 | 2 | 2 | 0 | 3 | 4 | 4 | 0 | 0 | 3 | 2 | 3 | 0 | 1 | 1 | | 20 |
| 6 | 1 | 2 | 2 | 2 | 1 | 0 | 4 | 3 | 0 | 0 | 1 | 3 | 1 | 1 | 3 | 1 | | 19 |
| 7 | 0 | 1 | 1 | 3 | 1 | 3 | 0 | 0 | 1 | 2 | 0 | 2 | 0 | 0 | 0 | 0 | | 22 |
| 8 | 0 | 1 | 1 | 3 | 3 | 1 | 3 | 0 | 2 | 3 | 0 | 1 | 0 | 1 | 0 | 0 | | 23 |
| 9 | 0 | 3 | 1 | 1 | 1 | 0 | 3 | 0 | 0 | 0 | 4 | 0 | 0 | 3 | 0 | 0 | | 24 |
| 10 | 1 | 1 | 0 | 4 | 3 | 2 | 0 | 2 | 0 | 2 | 1 | 1 | 1 | 3 | 0 | 0 | | 27 |
| ≈ | | | | | | | | | | | | | | | | | | ≈ |
| 65 | 0 | 2 | 1 | 2 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 2 | 2 | 1 | 1 | 0 | | 22 |
| 66 | 1 | 1 | 2 | 2 | 0 | 3 | 4 | 4 | 0 | 1 | 1 | 0 | 2 | 2 | 1 | 1 | | 23 |
| 67 | 0 | 3 | 1 | 2 | 0 | 1 | 3 | 0 | 0 | 3 | 0 | 1 | 2 | 0 | 3 | 0 | | 19 |
| 68 | 1 | 3 | 2 | 1 | 2 | 2 | 3 | 1 | 3 | 1 | 1 | 2 | 1 | 0 | 3 | 0 | | 4 |
| 69 | 0 | 2 | 2 | 3 | 1 | 3 | 1 | 4 | 3 | 2 | 0 | 1 | 0 | 1 | 3 | 0 | | 18 |
| 70 | 1 | 3 | 0 | 1 | 3 | 1 | 3 | 0 | 0 | 2 | 0 | 2 | 1 | 1 | 1 | 0 | | 17 |
| ≈ | | | | | | | | | | | | | | | | | | ≈ |
| …. | 4 | 1 | 2 | 3 | 1 | 4 | 1 | 2 | 3 | 0 | 3 | 1 | 1 | 1 | 0 | 0 | | 21 |
| …. | 1 | 2 | 1 | 2 | 0 | 2 | 3 | 3 | 0 | 1 | 2 | 1 | 3 | 1 | 2 | 2 | | 20 |
| …. | 1 | 1 | 1 | 3 | 0 | 1 | 3 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | | 22 |
| …. | 1 | 3 | 3 | 1 | 0 | 4 | 0 | 0 | 1 | 1 | 1 | 1 | 2 | 1 | 1 | 1 | | 19 |

Figure 9. Results of implementing the algorithm in searching for distorted image.

## 5. Conclusion

This paper proves to give good and fast ability to recognize the sign using the intensity curves in defining a unique characteristic of each cuneiform sign images. The short mathematical calculations required in the suggested method making it very fast. The limited number of cuneiform sign makes the method very efficient from the side of defining a unique finger print for each image. Different sizes of the same image would give same finger print because of a normalization part in the suggested method. The effect of small percentages of noise has a minor effect on the algorithm results.

## References

[1] Aravanitis T., Davis T., Livingstone A., Pinilla Dutoit J., and Woolly S., "The Digital Classification of Ancient Near Eastern Cuneiform Data," University of Birmingham, 2000.

[2] Institute of Physics, "Fonts for Neo-Assyrian Cuneiform Piska Karel," Academy of science, Czech Republic, 1999.

[3] Keiger D., "Clay, Paper, & Code," *Johns Hopkins Magazine*, 2003.

[4] King L. W., *Assyrians Language Easy Lessons in Cuneiform Inscriptions*, London Kegan Paul, Trench, Trubner and CO, 1901.

[5] Manual D'epigraphie Akkadienne, Labat Rene, Librairie Orientaliste Paul Geuthner, Paris S. A., 1976.

[6] National Geographical Society, *Sumerian Dictionary to Decipher Ancient Texts*, Faye Flam, 2002.

[7] University of Birmingham, "The Computer Representation of Cuneiform: Towards the Development of A Character Code," Rosemere, 2000.

**Hilal Yousif** received his MSc from London University, and his PhD from Loughborough University of Technology, United Kingdom, in 1977 and 1986, respectively. He has taught at Iraqi Committee of Computers and Informatics Institute for Postgraduate Studies and University of Technology, Computer Science Department from 1988. Currently, he is the dean of Al-Rafidian University College. He polished 52 articles in the field of computer science, and supervised 15 PhD and 50 MSc students. His research interests include image processing, and data base.

**Abdul Munim Rahma** received his MSc from Brunel University, and his PhD from Loughborough University of Technology, United Kingdom, in 1982 and 1985, respectively. He taught at Baghdad University Department of Computer Science and Military College of Engineering, Computer Engineering Department from 1986 to 2003. Currently, he is an assistant professor at the University of Technology, Computer Science Department. He published 43 articles in the field of computer science and supervised 11 PhD and 32 MSc students. His research interests include image processing and graphics.

**Haithem Alani** received his MSc from University of Technology, Computer Science Department from 1988. He taught at Baghdad University Department of computer Science from 1993 to 2003. Currently, he is working at the Department of Computer Science, Al-Nahreen University. His research interest includes image processing.