Watermarking scheme for copyright of digital images

Sami Baba, Lala Krekor, Thawar Arif and Zyad Shaaban

Faculty of Information Technology, Applied Science University, Amman 11931, Jordan

Summary

Copyright protection of digital images received increasing attention in the last decade due to massive digital artwork distribution via internet, so digital watermarking was a potential solution to such problem. Invisible watermarking scheme has been applied in frequency domain, to embed a logo image inside a large original image. The bits of the logo image are embedded in random color components of the original image, as well as in random positions in each selected block, these positions are AC coefficients of the DCT matrix. The randomness are obtained from a Non-Linear Feedback Shift Register (NLFSR) pseudorandom bit generator that determines in which color component this logo image bits will embed, as well, the block number for hiding each bit has been chosen according to a (semi-random) function proposed in this work.

Key words:

Watermarking, DCT, pseudorandom bit sequence.

1. Introduction

Digital image watermarking has received increasing attention in the few last years due to rapid growth in the internet traffic. It is gaining popularity due its significance in content authentication and copyright protection for digital multimedia data [1]. During images transfer, data integrity is not really secure. Watermarking can be an answer to such problems. For applications dealing with images, the watermarking objective is to embed an invisible message inside the image data [2].

Watermarking (data hiding) is the process of embedding data into a multimedia element such as an image, audio or video file. This embedded data can later be extracted from, or detected in, the multimedia for security purposes [3]. In general, a digital watermark is a code that is embedded inside an image. It acts as a digital signature, giving the image a sense of ownership or authenticity [4].

Watermarking applications include copyright protection, authentication, embedded and hidden information. Firstly, watermarking systems that are intended for copyright protection require a very high degree of robustness. Then, watermarking process for authentication belongs to the fragile class of schemes. Slightest change in the image completely destroys the mark. Finally watermarking for embedding information requires resistance against moderate level of modification due to routine image processing such as compression or cropping [5].

Watermarking techniques developed for images are mainly classified into visible and invisible approaches. While the visible methods provide means for overt assertion of ownership with logos, the invisible methods provide covert protection of these rights [6].

In the classification of watermarking schemes, an important criterion is the type of information needed by the detector [3]:

- Non-blind schemes require both the original image and the secret key(s) for watermark embedding.
- Semi-blind schemes require the secret key(s) and the watermark bit sequence.
- Blind schemes require only the secret key(s).

Currently the digital watermarking technologies can be divided into two categories by the embedding position—spatial domain and frequency domain watermark. Spatial domain techniques developed earlier and is easier to implement, but is limited in robustness, while frequency domain techniques is more robust and compatible to popular image compression standards. Thus frequency domain watermarking obtains much more attention. To embed a watermark, a frequency transformation is applied to the host data. Then, modifications are made to the transform coefficients. Possible frequency image transformations include the Discrete Fourier Transform (DFT), (DCT) and others [7]. The DCT is a mathematical transformation that takes a signal and transforms it from spatial domain into frequency domain [8]. The DCT transformation is adopted in this paper.

Some perform content-based image watermarking scheme, e.g., The Harris-Laplace detector is adopted to extract feature points, which can survive a variety of attacks. The local characteristic regions (LCRs) are adaptively constructed based on scale-space theory. Then, the LCRs are mapped to geometrically invariant space by using image normalization technique. Finally, several copies of the digital watermark are embedded into the nonoverlapped LCRs by quantizing the magnitude vectors of (DFT) coefficients [9].

For authentication purposes, in addition to being imperceptible, the watermark has to be sensitive to the slightest modification. This is termed fragile watermarking and allows the detection of tampering attempts. In some cases, the watermark is required to be sensitive only to some attacks while not being affected by others, such as

Manuscript received April 5, 2009
Manuscript revised April 20, 2009

common processing techniques (semi-fragile watermarking) [10]. In order for a watermark to be useful it must be robust to a variety of possible attacks by pirates. These include robustness against compression such as JPEG, scaling and aspect ratio changes, rotation, cropping, row and column removal, addition of noise, filtering, cryptographic and statistical attacks, as well as insertion of other watermarks [11].

Watermarking consists of three main stages: insertion, detection and the removal of a watermark. The detection and removal are usually considered together. An embedding algorithm is used to embed the watermark in the image. The extraction algorithm recovers the watermark, which requires the same secret key that was used for watermark embedding. Extraction of a watermark can be separated into two phases, namely, watermark detection and watermark recovery [12]. The rest of the paper is organized as follows: In section two different types and methods are reviewed. In section three, the DCT method and its implementation in jpg images is illustrated. Section four illustrates the pseudorandom bits generation and the ways o test the randomness. Section five is the proposed watermarking method, and sections six and seven are the results and conclusion respectively.

2. Digital Watermark Types

Digital watermarks and their techniques can be subdivided and segmented into various categories; for example, they can be classified according to the application, source type (image watermarks, video watermarks, audio watermarks, text watermarks), human perception, and technique used. As watermarks can be applied in the spatial or frequency domain, different concepts, such as DFT, DCT, and wavelet transformation, or additionally, manipulations in the color domain and noise adding can be mentioned. Furthermore, digital watermarks can be subdivided on the basis of human perception. Digital watermarks can be invisible or visible. We see visible watermarks every day watching television, that is, TV station logos. They can be robust against operations or even fragile for use in copy control or authenticity applications. At least, digital watermarks can be subdivided into blind and non-blind detection techniques, which are strongly related to the decoding process [13].

In order to detect the watermark information, blind and non-blind techniques are used. If the detection of the digital watermark can be done without the original data, such techniques are called blind. Here, the source document is scanned and the watermark information is extracted. On the other hand, non-blind techniques use the original source to extract the watermark by simple comparison and correlation procedures. However, it turns out that blind techniques are more insecure than non-blind methods [13].

3. Discrete Cosine Transform

The DCT is a mathematical transformation that takes a signal and transforms it from spatial domain into frequency domain. Many digital image and video compression schemes use a block-based DCT, because this algorithm minimizes the amount of data needed to recreate a digitized image. In particular, JPEG and MPEG use the DCT to concentrate image information by removing spatial data redundancies in two-dimensional images [8].

In the standard JPEG encoding, the representation of the colors in the image is converted from RGB to YCbCr, then the image is decomposed in 8×8 blocks, these blocks are transformed from the spatial to the frequency domain by the DCT. Then, each DCT coefficient is divided by its corresponding constant in a standard quantization table and rounded down to the nearest integer. After this step, the DCT quantized coefficients are scanned in a predefined zigzag order to be used in the final step, the lossless compression as illustrated in fig. 1. In each block the 64 DCT coefficients are set up from the lowest upper left corner) to the highest frequencies (lower right corner) [14].

The most important visual characteristics of the image are placed in the low frequencies while the details are situated in the higher frequencies. The HVS (Human Visual System) is most sensitive to lower frequencies than to higher ones[15].

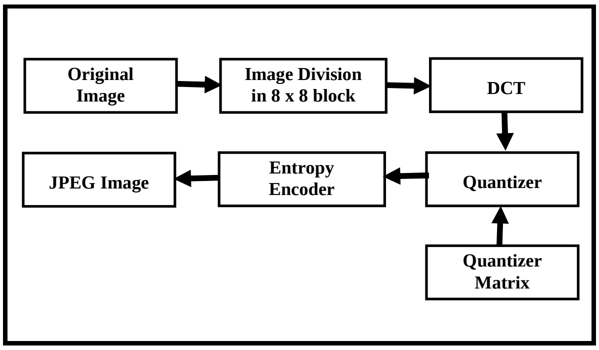


Fig. 1: JPEG Compression Algorithm

4. Pseudorandom Bits and Sequences

A (true) random bit generator requires a naturally occurring source of randomness. Designing a hardware device or software program to exploit this randomness and produce a bit sequence that is free of biases and correlations is a difficult task. Additionally, for most cryptographic applications, the generator must not be subject to observation or manipulation by an adversary

[16]. So pseudorandom bit generators are used to create a sequence of bits that appears to be random.

Linear feedback shift registers (LFSRs) are used in many of the keystream or bit sequence generators that have been proposed in the literature. There are several reasons for this [16]:

1. LFSRs are well-suited to hardware implementation;
2. They can produce sequences of large period;
3. They can produce sequences with good statistical properties; and
4. Because of their structure, they can be readily analyzed using algebraic techniques.

As an improvement, Non-Linear Feedback Shift Registers (NLFSRs) are used to overcome the linear complexity of the LFSR, as well as the basic statistical tests.

In this work, we used a NLFSR that consists of two 8-bits Shift Registers, with certain feedback functions, and the output of these two registers and XORed to give the pseudorandom bit sequence, as shown in fig. 2.

The proposed NLFSR pass four out of the five basic statistical tests to prove the randomness, these tests were:

1. Frequency test	(Passed).
2. Serial test	(Passed).
3. Poker test	(Not Passed).
4. Runs test	(Passed).
5. Autocorrelation test	(Passed).

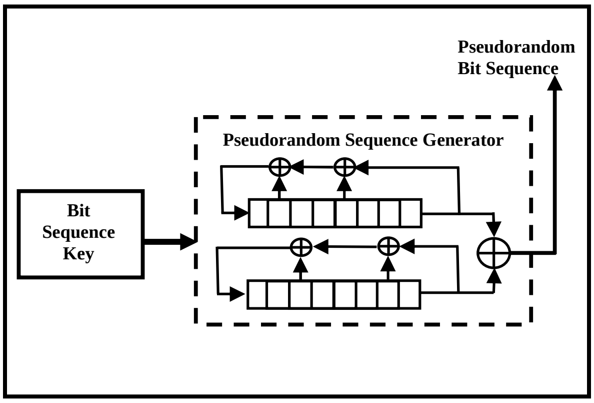


Fig. 2: NLFSR as a Pseudorandom Sequence Generator

5. Proposed Watermaking Method

The main idea of the proposed method is to take certain AC coefficient in the set of highest frequencies and embed a bit of the watermark image in the LSB of that AC coefficient. Then process of the JPEG encoding algorithm will continue as it is. The embedding process based on a pseudorandom sequence generator of binary bits that will use to decide in which color component (Y, Cb, Cr) the embedding will take place. An NLFSR has been implemented to generate the required pseudorandom bit sequence. The proposed method has been implemented by designing C#.NET software that have the ability to hide a logo inside a digital images and also the ability to check if the product is authorized or not by extracting the logo from the a **.JPEG** images.

In the proposed algorithm a digital signature (watermark) will be hidden in a true type **.BMP** color images where a logo of the company with size 40 × 40 greyscale image could be hidden in an image size of 1024 × 800 pixels or larger. The resulted image saved as a **JPEG** image file. Fig. 3 illustrates the block diagram of the proposed watermarking method. The embedding algorithm will be as follows:

Algorithm: Embedding Watermark

Input: **.BMP** image to apply the watermark to it, the logo of the company, the Key include the initial state of the NLFSR, and the step value.

Output: **.JPEG** image file.

Step 1: Open the true type **bmp** image, read the header of the image, the body of the image will be save in three separated arrays, the size of each array is as the image size, where the first array contains the red color component of the pixels, the second array contains the green color component, and the third array contains the blue color component of the pixels.

Step 2: Convert from Red, Green and Blue color space (**RGB)** to Luma component, blue-difference Chroma component and red-difference Chroma component **YCrCb.**

Step 3: Subtract 128 value from each resulted component, so the center will be around the zero.

Step 4: Each resulted array component of the three color spaces array will be decomposed into an 8 × 8 blocks and a Discrete Cosine Transformation (DCT) will be applied on each block.

Step 5: The resulted blocks are quantized using the quantization matrix Q50 with the quality level 50. The quantization process performed according to the following equation.
NewDCTCoefficient = round (OldDCT Coefficient / QuantizationM) for all elements in each block.

Step 6: The pseudorandom bit sequence generated from the NLFSR, will indicates in which color component the embedded bit will be hidden, hence the embedding process will be in the sequence of **Y**, **Cr** and **Cb** depending on the generated bit sequence, hence the existence of 1's in the pseudorandom sequence determines where to hide the embedded bit. For example, see the table below:

Bit sequence	: **1 1 0 1 0 0 1 1…**
Color Component	: **YCrCbYCrCbY…**
Embedding Sequence	: **YCrYYCr**

Step 7: This step determines in which block will hide the embedded bit, the embedded bits will be hidden.

A non-sequence of number is generated using a special algorithm (called Block Number Sequence Generator) used to generate a range of numbers which will accept two parameters, the width of the image and the step value, and return an array that contains numbers of different distribution, the same algorithm will applied to generate another non sequence numbers from the image height and the step.

Step8: Read the logo which will be embedded as a digital watermark converts the bytes of the logo to its corresponding bit value.

Step9: For all the bits of the logo do the following

- Get the first bit from the linear shift generator
- The bits of logo will be hidden in the Y, Cr, or Cb according to pseudorandom sequence as illustrated in step 6.
- Get the number of block form the BlockNumber function as illustrated in step 7.
- Select the AC coefficient [1,2] change the LSB of the selected coefficient to 0 or 1 depending on the bit to be hidden, if the bit is 1 change the LSB to 1 otherwise change it to 0, a hiding algorithm will be call.
- Apply Huffman code to save the resulted blocks.

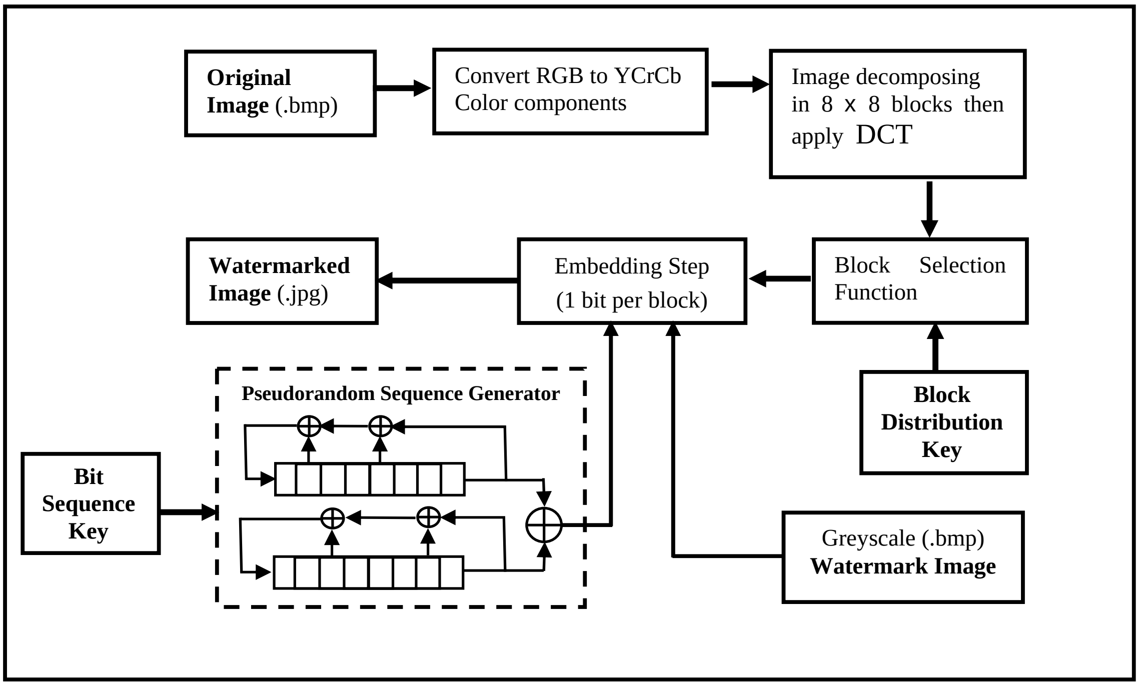

Original Image (.bmp)
Convert RGB to YCrCb Color components
Image decomposing in 8 x 8 blocks then apply DCT
Block Selection Function
Embedding Step (1 bit per block)
Watermarked Image (.jpg)
Pseudorandom Sequence Generator
Bit Sequence Key
Block Distribution Key
Greyscale (.bmp) Watermark Image

Fig. 3: A block diagram of the proposed water marking method.

Algorithm: Block Number Sequence Generator

Input : **Step Key, Max Range Number**

Output : Array **AllNum** Contain the generated Pseudorandom Numbers

Step1: Initialize the value of **x** to Max Range Number

Step2: Initialize integer variables I, start, startStep, RowN to zero

Step3: Define a two dimensional array NumGen, NumGen1

Step4:

```
For  n = step downto 0  do
Begin
        while  start <= x
        Begin
                NumGen[RowN, i ] = start
                start = start + step
                i = i +1
        End while
        RowN = RowN +1
        startStep = startStep +1
        start = startStep
        i = 0
End For
```

Step4: Initialize integer variable **Num** to zero and i = 0

Step5:

```
For  k=0  to step   do
Begin
        Num = k;
        while  Num  < NumGen[RowN,0].length()
        Begin
             For j = 0  to Step  do
            Begin
                NumGen1[j,Num] = NumGen[j, i];
            End For
            Num = Num + step;
            i= i +1
        End while
  End For
```

Step6: Initialize the initeger varaible TotalNum to zero

Step7:

```
  For  row = 0  to step  do
For k =0 to NumGen[RowN,0].length() do
     Begin
          AllNum[TotalNum] = NumGen1[row, k]
          TotalNum = TotalNum + 1
     EndFor
```

Algorithm: Extracting the Watermark

Input : the watermarked **.JPG** image, the Key that includes the initial state of NLFSR and the step value.

Output : .BMP greyscale logo image

Step 1: Read the Header of the **.jpg** and get the data of the stored image.

Step 2: Decode the .jpg image that already encoded by Huffman code and RLE.

Step 3: Use the Block Number algorithm that generates the sequence of blocks that contains the hidden logo bits.

Step 4: Use the NLFSR to generate the pseudorandom bit sequence that determines in which color components the bits of the logo were be hidden.

Step 5: Retrieve the required AC coefficient and extract the LSB of that coefficient, then save it in a string of bits

Step 6: Repeat Steps 4 and 5 until all bits of the logo are extracted.

Step 7: Divide the generated bit string in blocks of 8-bits to create the logo image.

6. Results

For all experiments, we have built a JPG image in the baseline sequential mode with quality factor 100% (lossless compression). The algorithms have been implemented using Visual C#.Net programming language. The .bmp images were or large sizes that enable us to hide the selected logo of size 40x40 greyscale bmp image. The first experiment used a bmp image of size 2304 x 1728, and a logo of size 40 x 40. Figs. 4 to 6 show the original, watermark logo and watermarked images of the first experiment. In the second experiment the original image was of smaller size (2048x1536), and the figs. 7 to 9 show the results of the second experiment.

7. Conclusion

The DCT have been applied successfully in digital image watermarking. In this paper we described a new approach based on DCT digital image water marking, which was done by embedding a watermark logo (image) in different color components as well as semi-random image blocks. The combination of NLFSR to generate a pseudorandom bit sequence enhances the watermark robustness against attempt to remove it.

Acknoledgment

This paper received financial support towards the cost of its publication from the Deanship of Research and Graduate Studies at Applied Science University, Amman, Jordan.

References

[1] B. Mohan and S. Kumar, "A robust image watermarking scheme using singular value decomposition, *J. Multimedia*, vol. 3, no. 1, May 2008.

[2] G. Lo-varco, W. Puech, and M. Dumas, "DCT-Based watermarking method using color components", *Second European Conference on Color in Graphics, Imaging and Vision*, Germany 2004.

[3] A. Sverdlov, S. Dexter, and A. Eskicioglu, "Robust DCT-SVD domain image watermarking for copyright protection: Embedding data in all frequencies", *International Multimedia Conference*, Germany, pp. 166-174, 2004.

[4] E. Fu, "Literature survey on digital image watermarking", Technical Report, EE381K-*Multidimensional Signal Processing*, 1998.

[5] G. Lo-varco, W. Puech, and M. Dumas, "Content Based watermarking for securing color images", *J. Imaging Science & Technology*, vol. 49, no. 6, 2005.

[6] S. Mohanty, P. Guturu, E. Kougianos, and N. Pati, "A novel invisible color image watermarking scheme using image adaptive watermark creation and robust insertion-extraction", 8th IEEE International Symposium on Multimedia, San Diego, USA, pp. 153-160, December 2006.

[7] L. Liu, *A survey on digital watermarking technologies*, Technical Report, Stony Brook University, New York, USA, 2005.

[8] C. Coconu, V. Stoica, F. Ionescu, and D. Profeta, "Distributed implementation of discrete cosine transform algorithm on a network of workstations", *Proceedings of the International Workshop Trends & Recent Achievements in Information Technology*, Romania, pp. 116-121, May 2002.

[9] Xiang-Yang Wang, Jun Wu, "A Feature-based Robust Digital Image Watermarking Against Desynchronization Attack", International Journal of AUTOMATION AND COMPUTING 2007 4 (4): 428-432.

[10] S. Bounkong, B. Toch, D. Saad, and D. Lowe, "ICA for watermarking digital images", *J. Machine Learning Research*, no. 4, pp. 1471-1498, December 2003.

[11] S. Pereira and T. Pun, "A framework for optimal adaptive DCT watermarks", European Signal Processing Conference, Finland, pp. 1669-1671, September 2006.

[12] A. Parthasarathy, *Improved Content Based Watermarking for images*, M.Sc. Thesis, Louisiana State University, August 2006.

[13] J. Seitz, *Digital watermarking for digital media*, Information Science Publishing, 2005.

[14] JPEG, jpeg.org.

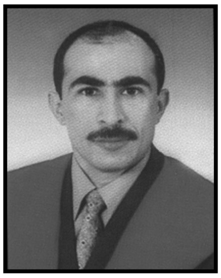
[15] W. Puech and J. Rodrigues, "Crypto-compression of medical images by selective encryption of DCT", *13th European Signal Processing Conference*, Turkey, September 2005.

[16] B. Schneier, *Applied Cryptography*, John Wiley and Sons, 1996.

Sami E. I. Baba got his BSc. Degree 1992 from Mansour University College, and his MSc. and PhD. From University of Technology (Baghdad-Iraq) in 1996 and 2000 respectively. He is now an Asst. Prof. at Computer Science Dept., Faculty of Information Technology in the Applied Science University, Amman Jordan. His research interests are Grammatical Inference, Adaptive Systems, image processing and Data Hiding.

Lala Z. Krikor got her BSc. Degree 1992 from Mansour University College, and her MSc. and PhD. From University of Technology (Baghdad-Iraq) in 1996 and 2000 respectively. She is now an Asst. Prof. at Computer Science Dept., Faculty of Information Technology in the Applied Science University, Amman Jordan. Her research interests are Image Processing and Information Hiding.

Thawar Arif is currently the head of Computer Science department at Applied Science University, Amman, Jordan. He has a Ph.D. from Baghdad University in Computer and Control Engineering, M.Sc. in Control and Instrumentation Engineering and B.Sc. in Control and Systems Engineering from University of Technology in Iraq. He is a member of the IEEE Computer Society. Also he is a professional member in the ACM. His research interests are: Image Retrieval, Watermarking, Adaptive Control systems and E-Government.



Zyad Shaaban was born in Jordan in 1969. He received a BSc. in Computer Science from Yarmouk University, Irbid, Jordan in 1992 and a Ph.D. in computer science from University of Technology, Johor Bahru, Malaysia in 1996. He is currently an assistant professor of computer science at the faculty of Information Technology, Applied Science University, Jordan. Dr. Zyad received a Fellowship from University of Technology, Malaysia and he was working on handwritten text recognition project. His research interests are: Handwritten Character Recognition, Moments Invariants, Neural Networks, Face Recognition, Image Retrieval and Arabic Text Recognition.

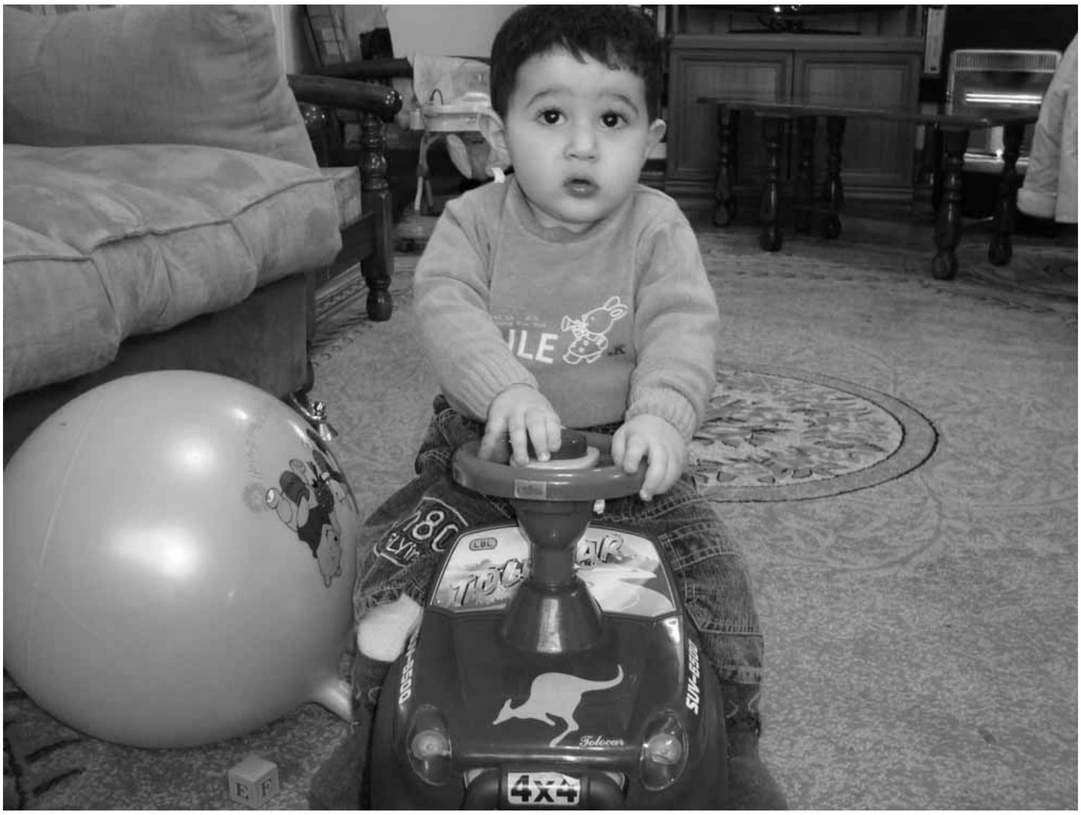


Fig. 4: Original Image (Kids Image) (.BMP) of size 2304 x 1728

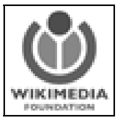


Fig. 5: Logo 1 (.BMP) Greyscale image of size 40 x 40

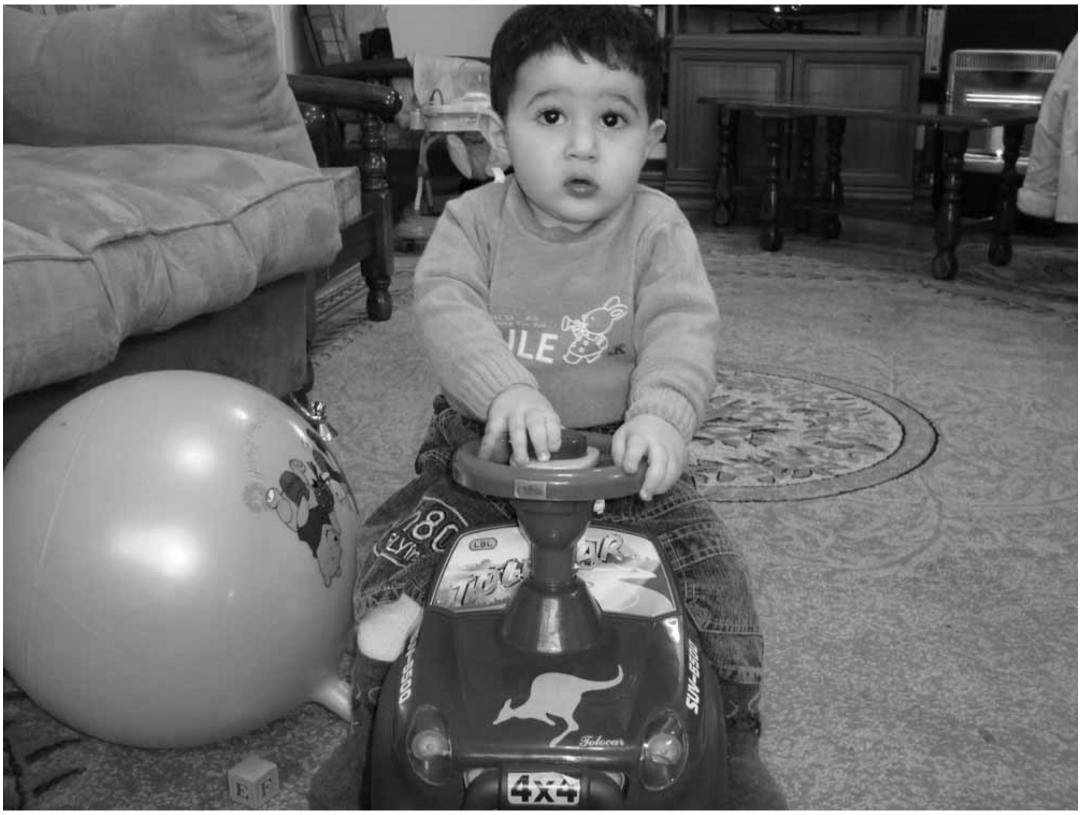


Fig. 6: Watermarked Image (.JPG) of size 2304 x 1728

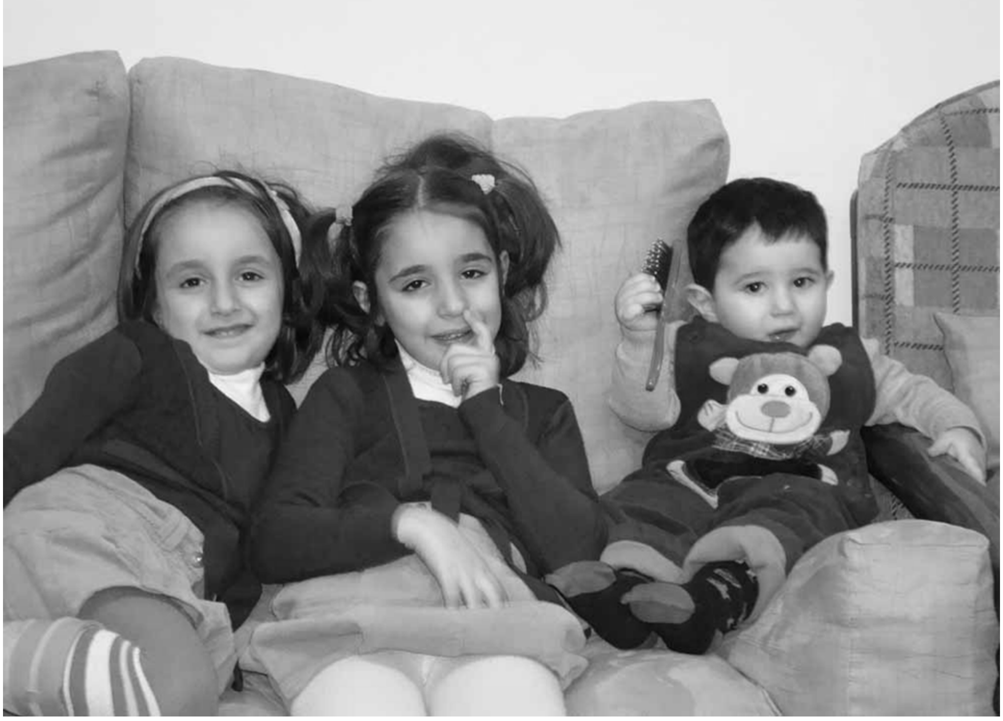


Fig. 7: Original Image (.BMP) of size 2048 x 1536

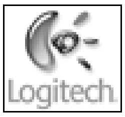


Fig. 8: Logo 2 (.BMP) Greyscale image of size 40 x 40

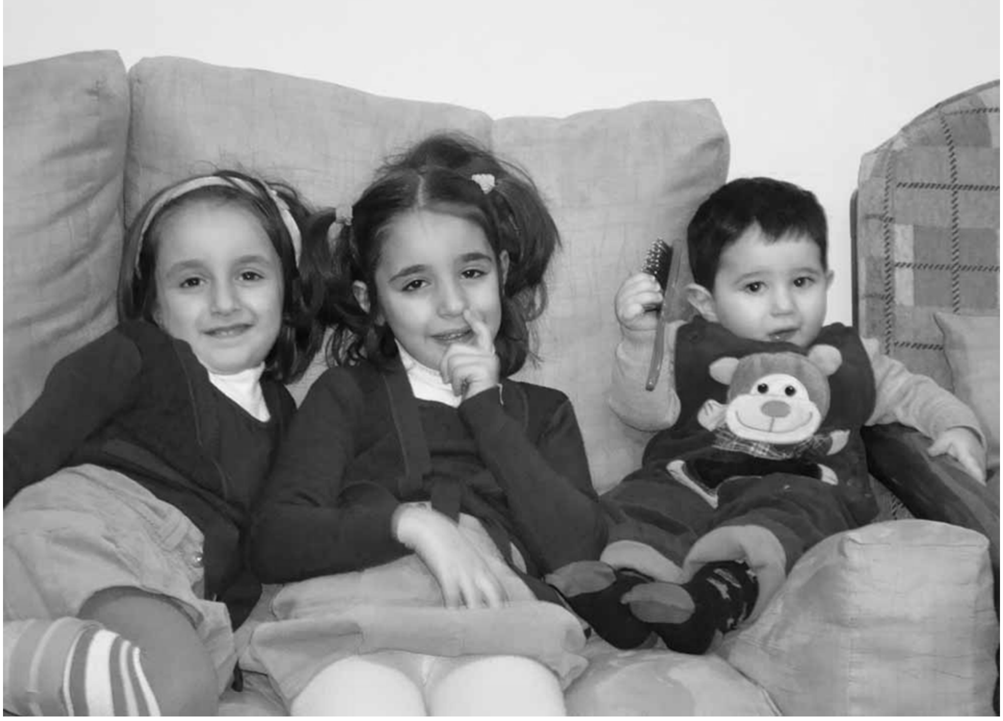


Fig. 9: Watermarked Image (.JPG) of size 2048 x 1536.