

Michael Werman
Shmuel Peleg

Department of Computer Science
The Hebrew University of Jerusalem
91904 Jerusalem, Israel

Abstract

An algorithm for reducing the number of gray levels of a picture with minimal degradation (in a mathematically defined sense) is described. This method can even produce a binary halftone of the picture, which is needed for many types of hard copy devices. The algorithm is based on a novel notion of distance between pictures that captures visual aspects.

1. Introduction

A metric between pictures is defined that takes into consideration visual aspects. This metric is used to quantize a picture into fewer gray levels with minimal degradation. In the limit the picture is represented by only two gray levels and a good binary halftone of the picture is generated.

Given a gray level picture, a picture with only half as many gray levels as the original picture is generated, which is in some mathematically defined sense the closest possible approximation to the original picture. The output picture appears to have more gray levels because the average area coverage varies with spatial gray level distribution, thereby providing the appearance of continuous shades of gray even when the output picture has only few gray levels. The limit of this process is the halftone representation of a gray level picture by a binary picture.

There have been many methods mentioned in the literature for computing a reduced quantizations of a picture (see [1]), and others for generating halftones (see [2] for a survey), each giving different results. The novelty in the approach described here is that it unifies the treatment of quantization with that of halftoning and treats them as an optimization problem which takes into consideration both the gray levels and their spatial arrangement.

2. A Metric for Pictures

Let P_1 and P_2 be two pictures, each having the same sum of gray levels: $\sum P_1(x, y) = \sum P_2(x, y)$, where $P_i(x, y)$ is the gray level of P_i at point (x, y) , $i = 1, 2$.

Definition 1:

$UF(P)$, the unfolding of P , is the multiset $\{(x, y)^{P(x, y)}\}$, where each element (x, y) appears $P(x, y)$ times.

Definition 2:

A matching (sometimes called a bipartite perfect matching) of $UF(P_1)$ and $UF(P_2)$ is a 1-1 pairing between elements from $UF(P_1)$ and elements from $UF(P_2)$.

Definition 3:

The weight of a matching is the sum of the distances μ between paired elements.

Definition 3 leaves open the choice of the pairwise "distance". In this paper we will use Euclidean distance, i.e. $\mu((x_1, y_1), (x_2, y_2)) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$.

Definition 4:

The match distance ρ between pictures P_1 and P_2 is the minimum weight for all possible matchings of $UF(P_1)$ and $UF(P_2)$.

The match distance ρ in Definition 4 is a metric, provided the distance μ between elements used in the weight computation (Definition 3) is a metric. For proofs see [3].

Example:

We will compute the match distance ρ for the following pictures, each having 3 as the sum of their gray levels.

$$f_1 = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}; f_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}; f_3 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 2 \end{bmatrix}.$$

The unfolded multisets are:

$$UF(f_1) = \{(1,1), (1,1), (2,2)\}; UF(f_2) = \{(1,1), (2,2), (3,3)\};$$

$$UF(f_3) = \{(2,2), (3,3), (3,3)\}.$$

The elements $(1,1)$ in $UF(f_1)$ and $(3,3)$ in $UF(f_3)$ appear twice, since $f_1(1,1)=2$ and $f_3(3,3)=2$. The match distances using minimal matching of the unfolded sets are:

$$\begin{aligned} \rho(f_1, f_2) &= \mu\{(1,1), (1,1)\} + \mu\{(1,1), (2,2)\} + \mu\{(2,2), (3,3)\} \\ &= 0 + \sqrt{2} + \sqrt{2} = 2\sqrt{2} \end{aligned}$$

$$\begin{aligned} \rho(f_2, f_3) &= \mu\{(1,1), (2,2)\} + \mu\{(2,2), (3,3)\} + \mu\{(3,3), (3,3)\} \\ &= \sqrt{2} + \sqrt{2} + 0 = 2\sqrt{2} \end{aligned}$$

$$\begin{aligned} \rho(f_1, f_3) &= \mu\{(1,1), (2,2)\} + \mu\{(1,1), (3,3)\} + \mu\{(2,2), (3,3)\} \\ &= \sqrt{2} + 2\sqrt{2} + \sqrt{2} = 4\sqrt{2} \end{aligned}$$

It can be easily verified that the above matchings are minimal.

We will compare the match distance ρ to the distance l_1 , which is the sum of absolute differences ($\sum |P_1(x, y) - P_2(x, y)|$).

This research has been supported by a grant from the Israeli NCRD and AFIRST. Part of this work has been carried out at the University of Maryland, where it was supported by NSF grant DCR-82-18408.

l_1 is the distance usually used between pictures. Let

$$g_1 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix}; g_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}; g_3 = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix}.$$

Using vector distance l_1 , all the g 's are equally far apart with distance 4. Using the match distance, however, yields $\rho(g_1, g_2) = \rho(g_2, g_3) = 2$ and $\rho(g_1, g_3) = 2\sqrt{2}$. This result is in better agreement with our intuitive notion of distance between these pictures.

3. Quantization as an Optimization Problem

We will first look at the problem of quantizing a picture into only half the original number of gray levels. Specifically, we consider the problem of reducing the gray level range from $[0, \dots, 2^n]$ to $[0, \dots, 2^{n-1}]$. This choice of reducing the number of gray levels from 2^n+1 to $2^{n-1}+1$ eliminates many problems that would arise if we used the more common numbers of gray levels 2^n (corresponding to the range $[0, \dots, 2^n-1]$) and 2^{n-1} (corresponding to the range $[0, \dots, 2^{n-1}-1]$).

Q is a quantization of P if

- 1) Q has half as many gray levels as P .
- 2) Q "looks like" P .

The sizes of P and Q are assumed equal.

If P and Q have the gray level ranges indicated above, Q will be said to "look like" P if $2Q$ (each gray level $Q(x, y)$ is changed to $2Q(x, y)$) is closest to P with respect to the match distance.

This problem can be solved as follows:

- 1) For each pixel (x, y) , set $Q(x, y)$ to $\left\lfloor \frac{P(x, y)}{2} \right\rfloor$.
- 2) Compute the minimal weight pairing on the set of all points (x, y) such that $P(x, y)$ is odd. For each pair $((x_1, y_1), (x_2, y_2))$ in this minimal weight pairing, add 1 to $Q(a, b)$ for some point (a, b) on the straight line segment connecting (x_1, y_1) and (x_2, y_2) .

Step (1) of the algorithm gives a perfect encoding of all the even gray levels, that is restored exactly by multiplying Q by 2. After this step there is an error of at most one gray level at each point. To compensate for this error we pair the odd pixels, and add one gray level to Q for each pair.

We now prove that this algorithm generates a picture Q such that $2Q$ is closest to P with respect to the match distance. When $P(x, y) = k$, it is treated as k points all at location (x, y) in P . Therefore, a point corresponds to one "gray level unit". Thus a point in Q is transformed into two points in $2Q$. Therefore in a matching between $2Q$ and P each point in Q is matched with two points in P . The optimization is achieved by minimizing the sum of distances between the points in Q and these pairs of points in P .

If a point of Q is matched with a pair of points of P it is optimal to place the point in Q at a location somewhere on the straight line segment connecting the pair of points of P . This is so, since for any pair of points p_1 and p_2 in the plane, the minimum of the sum of distances between these two points and any other point x is achieved when x is on the segment connecting p_1 and p_2 (triangle inequality). Moreover, this minimum sum

of distances is always the length of the straight line segment connecting p_1 and p_2 . Thus it is clear that to compute Q it is enough to compute a minimal weight pairing of the points in P .

When the distance between two points is zero they can always be paired in the minimal weight pairing (triangle inequality). This is used in the algorithm by dividing the gray levels by 2 at the first step. This part of the matching is done separately from Step (2) in order to save time.

Example:

We will compute the quantization Q into half the gray levels for the picture P :

$$P = \begin{bmatrix} 7 & 5 & 4 & 3 & 1 \\ 7 & 5 & 4 & 3 & 1 \\ 7 & 5 & 4 & 3 & 1 \\ 7 & 5 & 4 & 3 & 1 \\ 7 & 5 & 4 & 3 & 1 \end{bmatrix} \quad Q \text{ after Step (1)} = \begin{bmatrix} 3 & 2 & 2 & 1 & 0 \\ 3 & 2 & 2 & 1 & 0 \\ 3 & 2 & 2 & 1 & 0 \\ 3 & 2 & 2 & 1 & 0 \\ 3 & 2 & 2 & 1 & 0 \end{bmatrix}$$

The odd pixels of P are marked below with x s, and are paired in a minimal weight pairing:

$$\begin{bmatrix} x & x & . & x-x \\ x & x & . & x-x \\ x & x & . & x-x \\ x & x & . & x-x \\ x-x & . & x & x \end{bmatrix}$$

For each pair of odd pixels above we designate a location to which 1 will be added. This location is usually the location of one of the pixels in the pair, or a point on the straight line segment connecting the pair. The selected locations for our example are shown below:

$$\begin{bmatrix} 1 & . & . & . & 1 \\ . & 1 & . & 1 & . \\ 1 & . & . & . & 1 \\ . & 1 & . & 1 & . \\ 1 & . & . & . & 1 \end{bmatrix}$$

The final Q is the sum of the first Q and the points chosen from the connecting segments between elements of the minimal weight pairing:

$$\begin{bmatrix} 3 & 3 & 2 & 1 & 1 \\ 4 & 2 & 2 & 2 & 0 \\ 3 & 3 & 2 & 1 & 1 \\ 4 & 2 & 2 & 2 & 0 \\ 3 & 3 & 2 & 1 & 1 \end{bmatrix}$$

An unfortunate limitation of the algorithm is that Step (2) requires computation of a minimal pairing, which may take on the order of n^3 time, where n can be as large as the number of pixels in P . This is quite unacceptable, as the number of pixels in a picture is usually quite large. For minimal weight pairing in the plane, as needed for Step (2), there are some good heuristics that compute a pairing having close to minimal weight, and that take time no more than linear in the number of points to be matched [4,5,6].

To compute a picture with only half as many gray levels as a given picture we therefore use a heuristic which gives a pairing whose weight is close to optimal, rather than the the exact minimal weight pairing. The time complexity of this method is thus linear in the number of pixels in the input picture.

By carrying out this process k times we get a picture whose range of gray levels is reduced by a factor of 2^k from that of the original picture. Given a picture with gray levels in the range $[0, \dots, 2^k]$, after performing the algorithm k times, the resulting picture will only two gray levels and will be a halftone representation of the original picture. In the process, levels of quantization between the halftone and the original picture are also obtained. An example is shown in Figures 1, 2, and 3.

The computation of Q such that nQ is closest to P (for example, computing the halftone in one step) for arbitrary n is in the class NP-complete, which effectively means that it is not feasible for exact computation. We therefore use the iterative algorithm.

4. Concluding Remarks

A unifying method for quantization and half toning has been presented, based on a new definition of the distance between pictures.

A method of determining the location at which to add the extra point on the segment connecting two odd points was not given, as it does not seem to matter visually. In some cases, it may be desirable to use an endpoint, an in other cases the midpoint. In our example, we used a randomly chosen endpoint.

5. Acknowledgements

The authors wish to thank Ralph Hartley for suggesting halftoning as an optimization problem and Nati Lineal for helpful discussions.

REFERENCES

- [1] A. Rosenfeld and A.C. Kak, *Digital Picture Processing*, Academic Press, 1982, Ch. 4.
- [2] J.C. Stoffel and J.F. Moreland, A survey of techniques for pictorial reproduction, *IEEE Transactions on Communications*, Vol. COM-29, 1981, 1898-1925.
- [3] M. Werman, S. Peleg, and A. Rosenfeld, A distance metric for multidimensional histograms, *Computer Vision, Graphics and Image Processing*, Vol. 32, 1985, 328-336.
- [4] M.E. Dyer and A.M. Frieze, A partitioning algorithm for minimum weighted Euclidean matching, *Information Processing Letters*, Vol. 18, 1984, 59-62.
- [5] K.J. Supowit and E.M. Reingold, Divide and conquer heuristics for minimum weighted matching, *SIAM J. Computing* Vol. 12, 1983, 118-143.
- [6] K.J. Supowit, E.M. Reingold and D. Plaisted, The traveling salesman problem and minimum matching in the unit square for minimum weighted matching, *SIAM J. Computing*, Vol. 12, 1983, 144-156.



Figure 1: The iterations of the quantization algorithm starting from an original picture with 129 gray levels (A) and ending with a halftone with 2 gray levels (G).

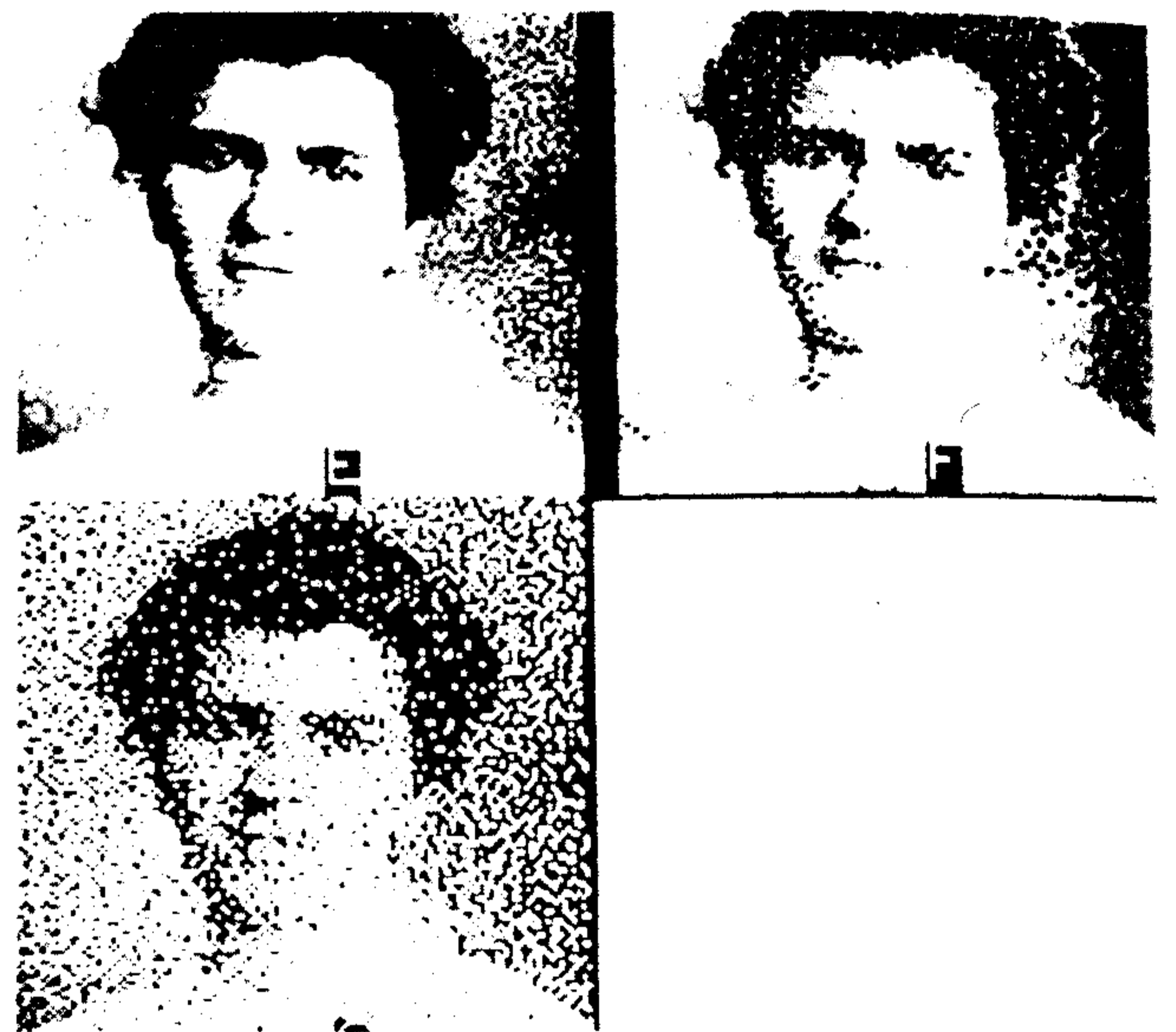


Figure 2: The last three pictures in Figure 1, with 5, 3 and 2 gray levels, enlarged.



Figure 3: Further enlargement of the picture with five grey levels.