Correspondence

A Unified Approach to the Change of Resolution: Space and Gray-Level

SHMUEL PELEG, MICHAEL WERMAN, AND HILLEL ROM

Abstract—Multiple resolution analysis of images is a current trend in computer vision. In most cases, only spatial resolution has been considered. However, image resolution has an additional aspect: gray level, or color, resolution. Color resolution has traditionally been considered in the area of computer graphics.

By defining a suitable measure for the comparison of images, changes in resolution can be treated with the same tools as changes in color resolution. A gray tone image, for example, can be compared to a halftone image having only two colors (black and white), but of higher spatial resolution.

An important application can be in pyramids, one of the most commonly used multiple (spatial) resolution schemes, where this approach provides a tool to change the color resolution as well. Increasing color resolution while reducing spatial resolution to retain more image details and prevent aliasing is an example of the possibility to find optimal combinations of resolution reduction, spatial and color, to best fit an application.

Index Terms—Computer graphics, computer vision, image digitization, image resolution.

I. INTRODUCTION

The process of image digitization: creating a digital picture from a continuous image, involves two stages. One stage is image sampling, which determines the spatial resolution of a picture, and the second is image quantization, which determines the gray-level resolution at each pixel [6], [7]. Each of these stages had been treated separately, and multiresolution approaches in computer vision [8] as well as in signal processing [2] consider only spatial resolution.

A method introduced earlier to measure distances between histograms [10] is the basis to a unified treatment of spatial and gray-level resolution. This measure has also been found useful for reduction of gray-level resolution [3], and in this paper we present its applicability to the change of spatial resolution as well.

The ability to manipulate both spatial and gray-level resolutions can be very useful in several computer vision applications. Many pyramid algorithms, for example, use the effect that objects arbitrarily far away in the image eventually become local at one of the pyramid's levels. But spatial resampling usually has aliasing effects which introduce information not in the original image. These aliasing effects can be greatly reduced by increasing the color resolution while decreasing the spatial resolution. In other applications, if resolution is being reduced for compression purposes, it is beneficial to consider for a given compression rate the best combination of resolution reduction in both space and gray level.

In Section II, we introduce a graph-theoretical definition of the basic measure used. This will be followed by application to spatial resampling in Section III, and gray-level requantization in Section IV. Section V presents a hybrid treatment of space and gray-level resolutions.

II. DISTANCE BETWEEN PICTURES

A *picture* will be represented by a set of *pixels* in the plane. Each pixel can have a gray level in the range $[0, \cdots, N]$, and is represented by N *pebbles*. When the gray-level of pixel i is g_i we assume that pixel i has g_i "white" pebbles and $N - g_i$ "black" pebbles. A picture is then regarded as a collection of pebbles, some white and some black, which are placed on the plane only at pixel locations, exactly N pebbles per pixel.

A *matching* θ_{AB} is a one to one mapping between the pictures A and B, i.e., each pebble of A is matched to a pebble of B having the same color. A matching is therefore defined only between pictures with same number of black (and white) pebbles, pictures with equal sums of gray level.

The *cost* of a matching θ_{AB}, $\phi : \theta_{AB} \rightarrow \ ^+$, is a function from matchings to the nonnegative real numbers. Examples of reasonable cost functions for matchings are $\Sigma_{(a_i, b_i) \in \theta_{AB}} |a_i - b_i|$; $\Sigma(a_i - b_i)^2$; $Max \{ |a_i - b_i| \}$ where $|a_i - b_i|$ is the distance between the locations of pebbles a_i and b_i. A matching θ_{AB} can be regarded as an algorithm to change picture A into picture B: for each $(a_i, b_i) \in \theta_{AB}$, move pebble $a_i \in A$ to the location of b_i. The cost of this single pebble move is $|a_i - b_i|$. The cost of the matching can be regarded as the cost of the entire transformation of A and B, and is computed from the costs of individual movements in one of the three ways described earlier (sum, sum of squares, or max).

Let Θ be the set of all matchings between pictures A and B, and let $\theta \in \Theta$ be of minimal cost. This minimal cost matching will be called the *transformation* of A and B, and the cost $\phi(\theta_{AB})$ will be called the *match-distance* between A and B and denoted by $\rho(A, B)$. The transformation is not always unique, and in such cases can be chosen arbitrarily from all minimal cost candidates. The match-distance is unique with respect to the cost function ϕ.

When two pictures have same number of pixels as well as pebbles, it is enough to consider the white pebbles only. Let (a_i, b_i) be a pair of white pebbles in the minimal matching, designating a move of one white pebble from a_i to B_i. As each pixel has N pebbles, there is a minimal matching with a black pebble from the location of b_i moved back to the location of a_i. $(a_i, b_i) \in \theta_{AB}$ can therefore be regarded as an exchange of a white pebble from the location of a_i and a black pebble from the location of b_i, leaving the total number of pebbles at each pixel unchanged.

Example: We will compute the distance $\rho = \Sigma |a_i - b_i|$ for the following three pictures:

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

The numbers at the above example represent the gray level, or the number of white pebbles, at a pixel.

The minimal matchings of white pebbles are

$$\theta_{f_1, f_2} = \left\{ \langle (1, 0), (0, 0) \rangle, \langle (1, 1), (1, 1) \rangle, \langle (1, 2), (2, 2) \rangle \right\}$$

$$\theta_{f_2, f_3} = \left\{ \langle (0, 0), (0, 1) \rangle, \langle (1, 1), (1, 1) \rangle, \langle (2, 2), (2, 1) \rangle \right\}$$

$$\theta_{f_1, f_3} = \left\{ \langle (1, 0), (0, 1) \rangle, \langle (1, 1), (1, 1) \rangle, \langle (1, 2), (2, 1) \rangle \right\}$$

θ_{f_1, f_2} specifies that in order to change f_1 into f_2 the white pebble at location $(1, 0)$ is exchanged with the black pebble at location $(0, 0)$, nothing happens at the center location $(1, 1)$, and the white pebble at location $(1, 2)$ is exchanged with the black pebble at

location $(2, 2)$. The cost of each exchange is 2 (moving two pebbles at distance of 1 each), yielding the total cost of $\rho(f_2, f_3) = 4$ and $\rho(f_1, f_3) = 4\sqrt{2}$. Using regular vector distance all three pictures are equally far apart, but using the match-distance yields larger distance between f_1 and f_3. This result is in better agreement with our intuitive notion of distance between pictures.

III. SPATIAL RESAMPLING

In spatial resampling, given a set $X = \{x_i\}$ of original pixels, and another set of pixels $Y = \{y_i\}$ (the target picture), distribute the gray levels ("pebbles") from the pixels of X to the pixels of Y, so that the picture Y is as similar as possible to the picture X. We move white as well as black pebbles, and as Y should be a picture too, each of its pixels receives equal number of pebbles.

As in all resampling methods, we assume that the gray level at a pixel y_j is a linear combination of the gray levels of pixels x_i, with weights w_{ij}

$$G(y_j) = \sum_i w_{ij} G(x_i).$$

The cost of this resampling transformation is computed as follows. As each pixel x_i has N pebbles, the cost of contributing w_{ij} of them to y_j is $N \cdot w_{ij} \cdot d_{ij}$ where d_{ij} is the distance between x_i and y_j. The total cost of the resampling transformation, under the sum cost, is

$$N \cdot \sum_i \cdot \sum_j w_{ij} \cdot d_{ij}.$$

We wish to minimize the above cost, while maintaining the following conditions:

1) The pixels in X contribute all their pebbles, i.e.,

$$\sum_j w_{ij} = 1.$$

2) All pixels of Y receive equal number of pebbles, $|X| \cdot N/|Y|$, i.e.,

$$\sum_i w_{ij} = \frac{|X|}{|Y|}$$

where $|X|$ is the number of pixels in the picture X.

3) The gray level of pixel y_j depends more on pixels x_i that are closer to it than on further ones.

We will find the coefficients of the transformation by using linear programming. As we move complete pebbles, we need the coefficients w_{ij} to be integers. We can get this by normalizing the coefficients and rewriting the linear program with coefficients r_{ij} as follows:

Minimize the cost $\sum_i \sum_j r_{ij} \cdot d_{ij}$

such that for each x_i $\quad \sum_j r_{ij} = |Y|$

and for each y_j $\quad \sum_i r_{ij} = |X|$.

Solving the above linear program for the r_{ij}'s, we can compute the white pebbles at Y (the gray levels) by $G(y_j) = \sum_i r_{ij} \cdot G(x_i)$. $G(y_j)$ is now in the range $[0, \cdots, N \cdot |X|]$. If we wish to normalize the gray levels of Y to be in the range of $[0, \cdots, N]$ as in X, we can normalize $G(y_j)$

$$G'(y_j) = \frac{\sum_i r_{ij} \cdot G(x_i)}{|X|}.$$

In Section V, we will discuss the case where we wish to change the gray-level range of the target image as well as its size.

For the pairwise distance d_{ij} between points x_i and y_j we can use different measures, like the city-block, chess-board, or Euclidean distances.

A property of this linear program is that there is always a minimum in which all the r_{ij} are integer. This follows from the fact that the matrix of the equalities is unimodular, as it describes a bipartite graph between the pebbles of X and Y [4].

An example of this method for the expansion and contraction of pictures is shown in Fig. 1. Although in Fig. 1 all images are on a rectangular grid, the change of scale is arbitrary, and is not restricted by integral multiplications. The method is also not restricted to regular grids, and is defined on any two sets of pixels. If can transform a square grid, for example, into an hexagonal grid, log polar, etc.

This resampling method can also accommodate varying weights on pixels, similarly to the method described in [5]. This approach gives higher weight to pixels on high information regions, like edges or other busy regions. A more elaborate treatment of weighted pixels can be found in [11].

IV. GRAY-LEVEL REQUANTIZATION

Reduction of the gray-level range of an image such that the resulting picture will be as similar as possible to the original picture is considered. We use the match-distance to measure the quality of reduction of gray-level resolution, enabling the comparison of gray-level resolution to spatial resolution as will be done in the next chapter.

This requantization method, first described in [3], can take an image in any gray-level range and transform it optimally to another image of the same size in any desired gray-level range. The benefit of this process is the cost it associates for the requantization. Such cost is unavailable with other requantization methods. The computational cost of arbitrary gray-level changes is high. More computationally efficient is the subproblem of the reduction of the gray-level range from $[0, \cdots, 2N]$ to $[0, \cdots, N]$. Repetition of this process will result in a binary halftone.

Given a picture with gray-level range of $[0, \cdots, 2N]$, reducing the gray-level range to $[0, \cdots, N]$ is equivalent to leaving the range $[0, \cdots, 2N]$, but having only even gray levels, i.e., $[0, 2, 4, \cdots, 2N]$. Changing from range of $[0, 2, 4, \cdots, 2N]$ to $[0, \cdots, N]$ is accomplished easily using a division by two. In order to change all gray levels to be even, only pixels with odd gray levels are considered. We pair together such pixels, and they exchange within each pair a white pebble for a black pebble. After such exchange, both pixels in the pair get an even gray level. We should remember that the gray level is also the number of white pebbles at a pixel.

The pairing of pixels with odd gray levels is done minimizing its cost: minimize the sum, over all pairs, of pairwise distances. As pairwise distances we consider the distance between the two pixels of the pair where the distance can be Euclidean, chessboard, city block, etc.

The algorithm for reducing the gray-level range from $[0, \cdots, 2N]$ to $[0, \cdots, N]$ is summarized as follows.

1) Optimally pair pixels with odd gray level. Algorithms for such pairing will be discussed later.

2) In all pairs, exchange one white pebble for a black pebble. The cost of such pairwise exchange is twice the distance between the pixels in the pair. After this step all gray levels are even.

3) Divide the gray levels by two.

Fig. 1. Examples for spatial resampling. The original image, size 128 × 128, is at the center. Other images are with sides of $1/2$, $2/3$, $3/2$, and 2 times the side of the original picture.

The following example can help to clarify the algorithm. Given the original picture:

$$\begin{bmatrix} 7 & 5 & 4 & 3 & 1 \\ 7 & 5 & 4 & 3 & 1 \\ 7 & 5 & 4 & 3 & 1 \\ 7 & 5 & 4 & 3 & 1 \\ 7 & 5 & 4 & 3 & 1 \end{bmatrix}.$$

We pair the pixels with odd gray levels

$$\begin{bmatrix} 7 & 5 & 4 & 3{-}1 \\ | & | & & \\ 7 & 5 & 4 & 3{-}1 \\ 7{-}5 & 4 & 3 & 1 \\ & & | & | \\ 7 & 5 & 4 & 3 & 1 \\ | & | & & \\ 7 & 5 & 4 & 3{-}1 \end{bmatrix}.$$

Within each pair we exchange a white for black pebble

$$\begin{bmatrix} 6 & 6 & 4 & 2{-}2 \\ | & | & & \\ 8 & 4 & 4 & 4{-}0 \\ 6{-}6 & 4 & 2 & 2 \\ & & | & | \\ 8 & 4 & 4 & 4 & 0 \\ | & | & & \\ 6 & 6 & 4 & 2{-}2 \end{bmatrix}.$$

We now divide the resulting gray level by two to get the final requantization

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

Step 1, which computes the minimal cost pairing, is the step with highest computational complexity. Good approximations to the minimal cost pairing can be found by heuristic methods, with linear time complexity [1], [9]. Exact proof that this pairing algorithm finds the closest picture in terms of the minimal match-distance can be found in [3]. Example of the gray-level requantization is shown in Fig. 2. The process is an iterative process, where at each iteration the upper bound of the gray-level range is reduced by half.

V. Hybrid Spatial and Gray-Level Resolution

The tool of match-distance enables us to perform and compare different combinations of reduction of resolution. We regard the *total resolution* of a picture as the number of bits used for its matrix representation. As an example, assume we wish to reduce by four the total resolution devoted to a picture of size 64 by 64 with 8 bits per pixel, denoted by $(64 \times 64 \times 8)$. Some of the possibilities include pure spatial resolution reduction: $(32 \times 32 \times 8)$; pure gray-level resolution reduction: $(64 \times 64 \times 2)$; and two hybrid combinations: $(37 \times 37 \times 6)$ and $(45 \times 45 \times 4)$. The hybrid combinations do not give an exact reduction by four, but come very close. Given a picture and a desired total resolution reduction, the match distance can give us a tool to select among the different options.

The hybrid resolution reduction can be formulated as follows. Given a set $X = \{x_i\}$ of original pixels, N pebbles in each pixel, and a target picture $Y = \{y_i\}$ with K pebbles in each pixel, find the picture Y that will most resemble X.
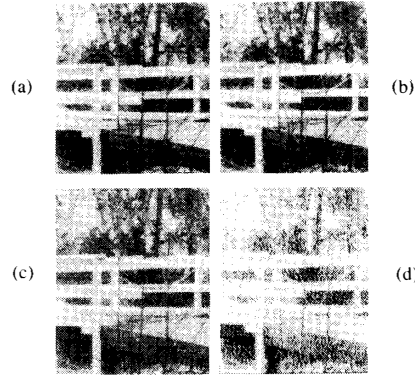

Fig. 2. Comparing the original image with gray-level range [0. . 256] (a) with the last three iterations of the requantization algorithm: gray-level range [0. .4] (b), gray-level range [0. .2] (c), and gray-level range [0. .1] (d).
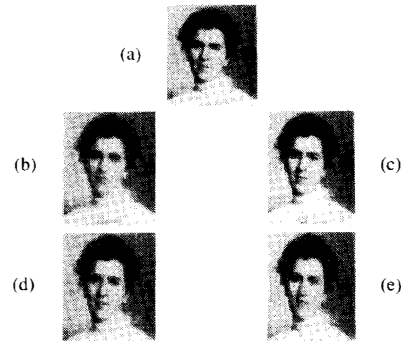

Fig. 3. Different combinations of same total resolution reduction. All images were spatially enlarged for display purposes to the size of the original picture. All reduced images need about the same number of bits for representation. (a) Original picture, size 144×144, gray-level range [0. .256]. (b) Pure spatial resolution reduction, size 48×48, gray-level range [0. .256]. (c) size 84×84, range [0. .4]. (d) size 72×72, range [0. .8]. (e) size 60×60, range [0. .16].

The number of pebbles in X is $|X| \cdot N$, and the number of pebbles in Y is $|Y| \cdot K$. As the number of pebbles in both pictures is different, $|X| \cdot N \neq |Y| \cdot K$, we use their *least common multiple*, and find the smallest α, β, such that $|X| \cdot N \cdot \alpha = |Y| \cdot K \cdot \beta$. We multiply the number of pebbles in each pixel of X by α (and have $N \cdot \alpha$ pebbles per pixel), and the number of pebbles in each pixels of Y by β (to have $K \cdot \beta$ pebbles per pixel). We then map (a minimal cost mapping) the pebbles of X and Y. We perform this mapping under the constraint that pixels in Y get only $k \cdot \beta$, ($k = 0, 1, \cdots, K$) white pebbles. With this constraint, simple division of the number of pebbles in Y by β will result in the desired range of K pebbles per pixel. Without this constraint the division would result in fractions of white and black pebbles.

The complexity of the algorithm to solve the mapping described above is very high, and at this moment we do not have an efficient algorithm to perform it completely. To get approximated results we use two stages. In the first stage, we change the spatial resolution using the linear program of Section III, and get $N \cdot |X|$ pebbles at each pixel of Y. In the second stage, we use the halftoning technique of Section IV to reduce the number of pebbles to the desired K. Actually, if $K \geq N$, simple division is satisfactory. Only when $K < N$ we have to use the halftoning technique.

Fig. 3-4 show combined application of space and gray level resolutions. Fig. 3 displays a portrait, while Fig. 4 is a texture picture.
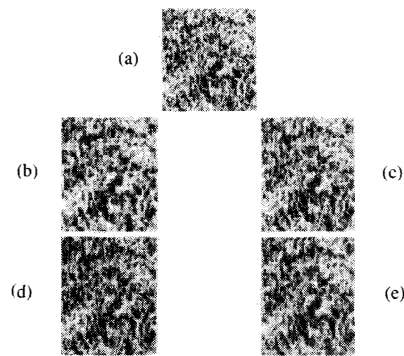
Fig. 4. Same as Fig. 3.

VI. Concluding Remarks

This paper presents a hybrid treatment of resolution, and the possibilities in trading spatial for gray-level resolution and vice versa. The theoretical foundation of the match-distance is strong, but it is computationally very expensive. As future directions, we see either the simplification of the match-distance to a measure more easily computed, or finding efficient and accurate estimates for the match-distance. We have no doubt, however, that this approach to the treatment of resolution will prove useful to the future generation of multiresolution methods.

References

[1] M. Ben-Or. "Lower bounds for algebraic computation trees," in *Proc. 15th ACM Symp. Theory Comput.*, Boston, MA, 1983, pp. 80–86.

[2] R. E. Crochiere and L. R. Rabiner, *Multirate Digital Signal Processing.* Englewood Cliffs, NJ: Prentice-Hall, 1983.

[3] M. Werman and S. Peleg, "Halftoning as optimal quantization," in *Proc. Eighth Int. Conf. Pattern Recog.*, Paris, France, Oct. 1986, pp. 1114–1116.

[4] C. K. Papadimitriou and K. Steiglitz. *Combinatorial Optimization: Algorithms and Complexity.* Englewood, Cliffs, NJ: Prentice-Hall, 1982.

[5] S. Peleg, O. Federbush. and R. A. Hummel, "Custom-Made Pyramids." in *Parallel Computer Vision.* New York: Academic, 1987, pp. 125–146.

[6] W. K. Pratt, *Digital Image Processing.* New York: Wiley-Interscience, 1978.

[7] A. Rosenfeld and A. C. Kak, *Digital Picture Processing (2nd ed.).* New York: Academic. 1982.

[8] A. Rosenfeld. *Multiresolution Image Processing and Applications.* New York: Springer Verlag, 1984.

[9] K. J. Supowit and E. M. Reingold, "Divide and conquer heuristics for minimum weighted matching," vol. 12, 1983.

[10] M. Werman. S. Peleg, and A. Rosenfeld, "A distance metric for multidimensional histograms." *Comput. Vision, Graph., Image Processing*, vol. 32, pp. 328–336, Dec. 1985.

[11] M. Werman. "A distance measure for the comparison of pictures and shapes." Ph.D. dissertation, The Hebrew Univ. Jerusalem, 1986.