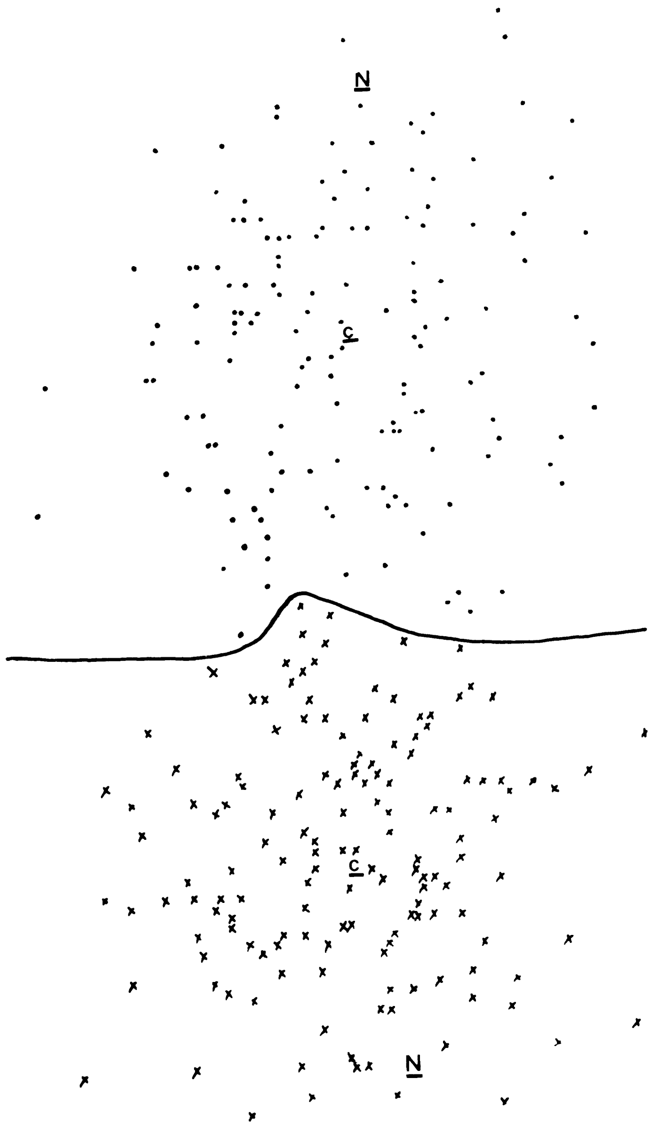

Fig. 11. Example.

random value (parameters are $(N-1)$, P_x) and P_x is the probability of a given point x to be in the parallelotope V_x centered in x:

$$E(P_x) = \sum_{i_1 \ldots i_n} P_{i_1 \ldots i_n}^2$$

$$E(\text{Card (E)}) = E\left(\sum_{x=1}^{N} N_x\right) = \sum_{x=1}^{N} E((N-1)P_x)$$

$$= \sum_{x=1}^{N} (N-1) \sum_{i_1 \ldots i_n} P_{i_1 \ldots i_n}^2$$

$$= N(N-1) \sum_{i_1 \ldots i_n} P_{i_1 \ldots i_n}^2.$$

Let us calculate $E(\eta)$. $N_{i_1 \ldots i_n}$ is a binomial random value (parameters are N and $P_{i_1 \ldots i_n}$). So

$$E(N_{i_1 \ldots i_n}) = NP_{i_1 \ldots i_n}$$

and

$$E(N_{i_1 \ldots i_n}^2) = N(N-1)P_{i_1 \ldots i_n}^2 + NP_{i_1 \ldots i_n}$$

$$E(\eta) = E\left(\sum_{i_1 \ldots i_n} N_{i_1 \ldots i_n}[N_{i_1 \ldots i_n} - 1]\right)$$

$$= \sum_{i_1 \ldots i_n} [E(N_{i_1 \ldots i_n}^2) - E(N_{i_1 \ldots i_n})]$$

$$= \sum_{i_1 \ldots i_n} [N(N-1)P_{i_1 \ldots i_n}^2 + NP_{i_1 \ldots i_n} - NP_{i_1 \ldots i_n}]$$

$$= N(N-1) \sum_{i_1 \ldots i_n} P_{i_1 \ldots i_n}^2.$$

η is equal to Card (E) in mean value.

References

[1] K. Fukunaga, *Introduction to Statistical Pattern Recognition*. Academic Press, 1972.
[2] M. R. Anderberg, *Cluster Analysis for Applications*. Academic Press, 1973.
[3] S. Watanabe, "A unified view of the clustering algorithm," IFIP Long. 1971 Ijubiana Booklet TA-2.
[4] E. Diday, "Optimization in non hierarchical clustering," *Pattern Recognition*, vol. 6, pp. 17–33, 1974.
[5] D. J. Hall and G. H. Ball, "ISODATA: A novel method of data analysis and pattern classification" technical report 5R1—Projet 5533, Stanford Research Institute, Menlo Park, CA, 1965.
[6] E. Diday, "La méthode des nuées dynamiques et la reconnaissance des formes, *Cahier de l'I.R.I.A.*, Rocquencourt.
[7] R. R. Sokal and P. M. Sneath, *Principles of Numerical Taxonomy*. San Francisco: Freeman, 1963.

Coarse-Fine Template Matching

AZRIEL ROSENFELD, FELLOW, IEEE, AND GORDON J. VANDERBRUG

***Abstract*—The computational cost of template matching can be reduced by using block averaging to decrease the spatial resolution of the template and the input picture, applying the low-resolution ("coarse") template to the low-resolution picture, and using the full-resolution ("fine") template only when the coarse template's degree of mismatch with the picture is below a given threshold. This correspondence discusses the degree of resolution reduction that should be used, under various conditions, in order to minimize the expected computational cost.**

I. Introduction

In [1], a method of reducing the computational cost of template matching was described. This method applies a subtemplate to the input picture and uses the rest of the template only at positions where the subtemplate's degree of mismatch with the picture is below a given threshold. A probabilistic analysis of this approach was given, with emphasis on the choice of subtemplate size to minimize the expected computational cost. The analysis assumed that the average absolute difference in gray level between template points and corresponding picture points is used as the mismatch measure.

This correspondence treats an analogous method of two-stage template matching in which a reduced-resolution template, rather than a subtemplate, is used for the initial "screening" of the picture. The template is divided into blocks, which we assume

Manuscript received July 8, 1976; revised September 9, 1976. This work was supported by the National Science Foundation under Grant MCS-72-03610.
The authors are with the Computer Science Center, University of Maryland, College Park, MD 20742.

here to be of equal size (let us say containing m pixels each), and the average of each block is computed. The picture is averaged over an m-pixel neighborhood, of the same shape as the blocks, at each point. We compute the average absolute difference between the template block averages and the picture neighborhood averages at the corresponding picture points, for a given position of the template. If this average is below a certain threshold, a possible match between template and picture has been detected in the given position. In this case, we match the full-resolution template and picture at that position, by computing the average absolute difference of their individual gray levels at corresponding points, in order to investigate whether the detected match is a true match or a false alarm.

In this two-stage template matching process, the expected computational cost at each point of the picture (ignoring the cost of the averaging operations; see the next paragraph) is of the form $c + pd$, where

- c cost of applying the "coarse" (i.e., reduced-resolution) template,
- p probability of a below-threshold mismatch to the coarse template,
- d cost of applying the "fine" (i.e., full-resolution) template.

We may assume, as in [1], that the costs c and d are proportional to the numbers of pixels in the coarse and fine templates (so that $c = d/m$). If the coarse template has very few points, c will be small; but under some circumstances, this will lead to a higher probability of below-threshold mismatch, so that pd may be large. We can now pose an optimization problem analogous to that in [1]: Is there a degree of coarseness for which the expected computational cost is a minimum?

The cost of block-averaging the template can be ignored in our analysis, since this need only be done once, and when we are calculating the total computational cost for every position of the template relative to the picture, the contribution due to averaging the template becomes negligible. It is harder to justify ignoring the cost of averaging the picture, since this must be done in a neighborhood of every picture point. However, if we expect to match many different templates with the same picture, we can share the cost of the picture averaging among these costs, since the picture need be averaged only once.[1]

It should be pointed out that the analysis in this correspondence considers only the computational cost of the template-matching process, and not the costs of errors (false alarms or dismissals).

II. Probabilistic Analysis

As in [1], we assume that the picture consists mostly of background (i.e., that points where the template matches the picture are rare), so that the situation at background points dominates the expected computational cost. Let us suppose that the gray levels of background points have independent, identical normal distributions with mean μ and standard deviation σ. Then [2] the average of a block of m background points is normally distributed with mean μ and standard deviation $\sigma' = \sigma/\sqrt{m}$.

Let the average gray level of the ith block of the template be q_i. Then the (signed) difference between q_i and the average of a block of background points is normally distributed with mean $\mu_i = \mu - q_i$ and standard deviation σ'. The absolute value of this difference is not normally distributed; but as derived in [1], it has mean

$$v_i = \frac{2\sigma'}{\sqrt{2\pi}} e^{-\mu_i^2/2\sigma'^2} + \mu_i[2\Phi(\mu_i/\sigma') - 1]$$

and variance

$$\sigma_i^2 = \sigma'^2 + \mu_i^2 - v_i^2.$$

If there are sufficiently many blocks, we can assume (by the central limit theorem [2]) that the average absolute difference is normally distributed; for n blocks, the mean and standard deviation of this average are given by

$$\bar{\mu} = \frac{1}{n}\sum_{i=1}^{n} v_i \quad \text{and} \quad \bar{\sigma} = \frac{1}{n}\sqrt{\sum_{i=1}^{n} \sigma_i^2}.$$

These results are exactly the same as in the subtemplate case of [1], except that here v_i and σ_i are functions of $\sigma' = \sigma/\sqrt{m}$ rather than of the original σ.

Based on this analysis, it is straightforward to compute numerically the probability that the average absolute difference is less than some given threshold; this is the false alarm probability p of Section I. By doing this for various m's (i.e., various degrees of coarseness), we can determine—for any given template—the expected computational cost as a function of m, and can find the m for which this cost is a minimum. Note that for a given template, the number of blocks (n) varies inversely with the block size (m), since their product is the total number of points in the template.

III. Qualitative Discussion

In Section IV we shall compute the false alarm probability and the expected computational cost for cases involving a picture background composed of normally distributed noise, and a template having constant gray level, either equal or unequal to the mean of the noise. The following qualitative remarks can be made about these cases.

Let the template gray level z be different from the mean μ of the picture background. The coarser we make the template, the lower the standard deviation of the averaged picture becomes. Thus the expected absolute difference between a template block average (which is still z) and a picture neighborhood average will tend toward the nonzero value $|\mu - z|$. If this value is sufficiently high relative to the threshold t, the probability of the average absolute difference being below t—i.e., the false alarm probability p—should decrease. On the other hand, the coarser we make the template, the fewer blocks we have, so that the variability of the average absolute difference becomes greater, and this may increase p. Coarsening the template makes c decrease; but since there are two factors affecting p in opposite ways, the expected cost $c + pd$ may turn out to have a nontrivial minimum.

Analogous remarks can be made for the case in which $z = \mu$. Here, as the template gets coarser, the expected absolute difference between a template block average and a picture neighborhood average tends toward zero, so that for any threshold t, the false alarm probability p should increase. The increased variability in the average absolute difference may also tend to increase p. On the other hand, coarsening the template decreases c; thus the expected cost ought to have a nontrivial minimum even for low thresholds.

For the subtemplate problem, decreasing the size of the subtemplate decreases the cost c, but it increases the false alarm probability, since the average absolute difference has greater variability; and as seen in [1], this gives rise to nontrivial minima

[1] One could also argue that averaging the picture can be done at low cost by special-purpose hardware, or optically; but if this is true, then the template matching itself could presumably also be done that way.

in the expected computational cost. Note that for both problems, if we make the subtemplate too small or the coarse template too coarse (i.e., the number n of blocks too small), the central limit theorem no longer applies, so that our analysis becomes unreliable.

Our examples of templates having constant gray level are not very realistic; but the same analysis can also be used in more realistic cases. Suppose that the template consists of two kinds of points, light and dark, which have gray levels z and w, respectively, where $z - \mu = \mu - w$; in other words, these levels differ by equal amounts, but in opposite directions, from the picture mean μ. When we divide the template into blocks, let us do this in such a way that each block consists entirely of light points or entirely of dark points, so that the average of each block is either z or w. (We have nowhere assumed that the blocks must have simple shapes; they can be arbitrary sets of template points—and corresponding sets of picture points—as long as they have the specified size m.) Thus the absolute difference between a template block average and a picture "neighborhood" average has expected value $|\mu - z| = |\mu - w|$, and our discussion of the constant gray-level case applies in this case too.

An interesting modification of our optimization problem is obtained if we allow the threshold to vary with the amount of averaging, e.g., we use $t/\sqrt{m}$ when the block size is m, to compensate for the fact that the variability of the absolute differences has been reduced. This idea will not be investigated further in this correspondence.

IV. Experiments

A program similar to that in [1] was used to compute false alarm probabilities and expected computational costs for coarse-fine template matching, using the equations in Sections I and II.

Let the picture background mean be $\mu = 32$, and its standard deviation be $\sigma = 10$. Let the template have 64 points, each of gray level 32. Then the false alarm probabilities p, for various values of the block size m and threshold t, are shown in Table I. The corresponding expected computational costs, assuming that c and d are equal to the numbers of points in the coarse and fine templates, respectively, are shown in Table II. It is seen that nontrivial optima, between $m = 4$ and $m = 16$, do indeed exist.

Tables III and IV give analogous results for the case where the template has constant gray level 28 rather than 32. In this case there are nontrivial cost minima only for relatively large values of t. For smaller t's, false alarms are rare even for large m, since the expected absolute difference between template and picture is large; thus for small t's, the expected cost keeps decreasing as m increases, as discussed in Section II.

In these examples, we have used block sizes m that evenly divide the number of points in the template. One could also use "blocks" containing noninteger numbers of points (i.e., one could divide the template into n equal-area regions and compute the average gray level in each region by resampling; the same would have to be done in computing the neighborhood averages in the picture.) This would allow the data in Tables I–IV to be refined, so that the minima could be located more exactly. On the other hand, it would not normally be desirable to make the number of blocks (n) a noninteger, unless there is some special reason for using blocks of unequal sizes (e.g., we want to preserve the detail of some parts of the template, but we are interested only in the gross structure of other parts).

V. Concluding Remarks

Coarse-fine template matching provides a potential speedup of the template matching process by checking the gross structure of the template against a correspondingly coarsened picture, which permits rapid rejection of mismatch positions. This concept is a special case of the "planning" approach to scene analysis discussed by Kelly [3], where edges detected in a coarsened picture are used as clues to guide the tracking of these edges in the original picture.

Sampling (i.e., using subtemplates) and coarsening are two approaches to reducing the cost of template matching by a preliminary screening process. In general, we could compute the values of an arbitrary set of properties of the template (not necessarily sample gray levels, or block average gray levels), and

TABLE I
False Alarm Probabilities for Template Having Gray Level Equal to Background Mean

t \ m =	2	4	8	16	32
.01	0	0	.002	.035	.150
.02	0	.00	.019	.165	.421
.03	0	.003	.108	.445	.738
.04	.00	.026	.345	.757	.930
.05	.00	.133	.669	.937	1

TABLE II
Expected Computational Costs for Case in Table I*

t \ m =	2	4	8	16	32
.01	32	16	8.116	6.244	11.609
.02	32	16.009	9.226	14.544	28.934
.03	32	16.171	14.932	32.462	49.215
.04	32.001	17.637	30.066	52.456	61.491
.05	32.030	24.488	50.804	63.993	66

* The minima are underlined.

TABLE III
False Alarm Probabilities for Template Having Constant Gray Level Different from Background Mean

t \ m =	2	4	8	16	32
.03	0	.000	.007	.027	.042
.04	0	.002	.031	.083	.112
.05	.000	.014	.105	.201	.240
.06	.001	.064	.258	.385	.424

TABLE IV
Expected Computational Costs for Case in Table III*

t \ m =	2	4	8	16	32
.03	32	16.012	8.434	5.704	4.665
.04	32	16.130	10.002	9.306	9.139
.05	32.003	16.893	14.689	16.850	17.392
.06	32.044	20.083	24.498	28.652	29.150

* For $t = .03$ and .04, the cost continues to decrease even at the highest degree of coarseness. For $t = .05$ and .06, the minima are underlined.

match these values against the same property values measured at the given position in the picture; if a below-threshold mismatch of the property values occurred, we would investigate further by matching the template to the picture in that position. (Of course, this assumes that the cost of computing the property values can be ignored; see the end of Section I.) These remarks define a large class of methods of reducing template matching cost. The specific method appropriate to a given situation will, in general, depend on the nature of the picture and the template.

A general discussion of methods of detecting positions in which a template and a picture are likely to match can be found in Barnea and Silverman [4]. The specific technique of block averaging, as applied to a variety of picture processing operations, has been investigated by Riseman [5] and Tanimoto [6]. The chief contribution of the present paper is its *quantitative* treatment of the block-averaging approach, demonstrating that for specific amounts of averaging, the expected computational cost of the template-matching process is minimized.

We have seen that in some cases, the expected computational cost of coarse-fine template matching should be minimized for a large amount of coarsening (Section III). When the template and picture mean gray levels are equal, however, it is nontrivial to determine the degree of coarsening that yields the minimum computational cost; but the minimum can be found numerically (Section IV). In summary, the results of this correspondence confirm the value of coarse-fine matching as a method of reducing the computational cost of template matching operations.

References

[1] G. J. VanderBrug and A. Rosenfeld, "Two-stage template matching," *IEEE Trans. Computers*, vol. C-26, 1977, in press.
[2] E. Parzen, *Modern Probability Theory and Its Applications.* New York: Wiley, 1960, pp. 371–372.
[3] M. D. Kelly, "Edge detection in pictures using planning," in B. Meltzer and D. Michie, Eds., *Machine Intelligence*, vol. 6. Edinburgh University Press, 1971, pp. 397–409.
[4] D. I. Barnea and H. F. Silverman, "A class of algorithms for fast digital image registration," *IEEE Trans. Computers*, vol. C-21, pp. 179–186, 1972.
[5] A. R. Hanson and E. F. Riseman, "Preprocessing cones: A computational structure for scene analysis," COINS Technical Report C-7, Univ. of Mass., Amherst, MA, Sept. 1974.
[6] S. Tanimoto and T. Pavlidis, "A hierarchical data structure for picture processing," *Computer Graphics Image Processing*, vol. 4, pp. 104–119, 1975.

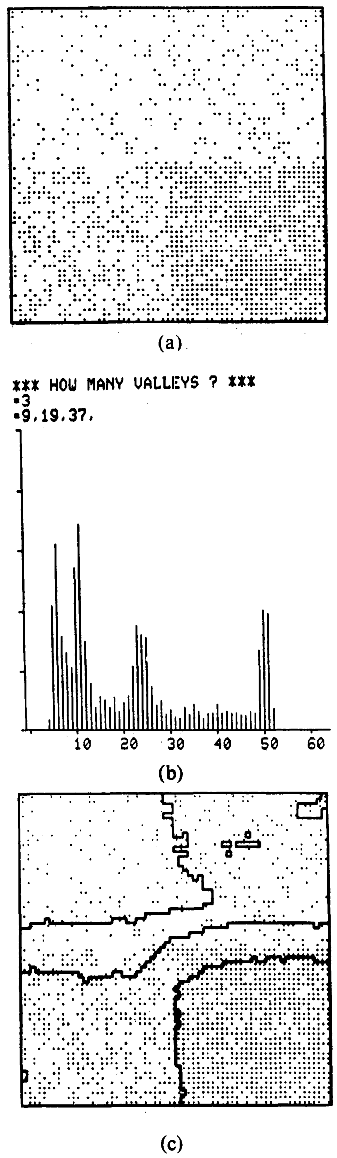

Fig. 1. (a) Picture containing four regions of different average gray level; probabilities of black points are 0.1, 0.2, 0.4, and 0.8, respectively. (b) Gray-level histogram of picture obtained by local averaging (a), using 16-by-16 neighborhood. (c) Regions obtained by thresholding locally averaged picture.

Extraction of Multiple Regions by Smoothing in Selected Neighborhoods

FUMIAKI TOMITA AND SABURO TSUJI

***Abstract*—A fixed smoothing method for extracting regions in a picture does not work if it has too many regions. An improved smoothing method which averages gray values in a selected neighborhood around each point in the picture is presented. This filtering process is iterated until an easily distinguishable picture is obtained.**

I. Introduction

Recently, L. S. Davis *et al.* discussed simple methods of extracting regions from a picture by averaging the picture and then thresholding it [1]. If the picture contains the regions of which the average gray levels lie in two disjoint ranges, a fixed smoothing method can separate the two region types. Unfortunately, this method does not work if there are three or more ranges, and regions in the high and low ranges are adjacent one another.

This correspondence presents an alternative averaging method to overcome the difficulty by selecting a homogeneous area around each point in the picture and averaging the gray values in it. This filtering process is iterated until one can obtain a picture with easily distinguishable regions.

II. Selection of Neighborhoods for Local Averaging

The procedures of detecting the regions by the fixed smoothing method are as follows.

1) Generate a new averaged picture in which the gray level of each point is the result of local averaging gray levels in a fixed size neighborhood of the corresponding point in the input picture.
2) Select thresholds by inspecting the gray level histogram of the averaged picture.
3) Threshold the averaged picture to obtain regions.

Manuscript received July 19, 1976; revised September 27, 1976.
The authors are with the Department of Control Engineering, Osaka University, Toyonaka, Osaka 560, Japan.