FAST IMAGE SEARCH USING A MULTISCALE STOCHASTIC MODEL

Srinivas Sista, Charles A. Bouman and Jan P. Allebach

School of Electrical Engineering
1285 Electrical Engineering Building
Purdue University
West Lafayette, IN 47907-1285

ABSTRACT

Searching an image for the occurrence of a pattern or a template is an essential step in a number of image processing applications. We propose a new multiresolution matching criterion based on the generalized log likelihood ratio. We also developed a multiscale search technique which facilitates finding the best solution by searching a small subset of the entire set of possible template locations. The search technique is designed to keep the amount of computation at each resolution approximately the same. The results obtained on our example images demonstrate the robustness and accuracy of the matching criterion along with a speed-up of over two orders of magnitude by the search technique.

1. INTRODUCTION

In the current work, we study the application of multiscale stochastic techniques to the problem of searching an image for the occurrence of desired objects specified either as example images or manual sketches. Our approach to the search is based on a multiscale stochastic image model. These models have recently been applied to tasks such as image segmentation [1], regularization [2], and automated object inspection [3].

Multiscale techniques can both improve performance and reduce computation. In fact, Burt has shown that the computational advantage is particularly great for search applications because the computation scales as the fourth power of the resolution [4]. Hence, decreasing the sampling frequency by a factor of 2 decreases the computation by a factor of 16.

The multiscale stochastic matching technique developed in this paper is based on formal detection theory. We compute the generalized log likelihood ratio as a measure of similarity, between the image and template data. We further show that this is the same as maximizing the mutual information between the image and template. Our criterion requires that the template and image be similar but not exactly the same. This gives a few advantages over existing fixed scale deterministic criteria such as Hausdorff distance [5], which are either insensitive to high resolution detail or have fixed sensitivity.

This work was supported by a grant from the Xerox Corporation.

Appeared in the *IEEE Int'l Conf. on Image Proc.*, Washington, DC, Oct. 22-25, 1995.

2. MULTISCALE MODEL

The image and template data are represented as multiscale trees as shown in Fig. 1. Each level of the data tree represents a different scale in the image. The image tree is assumed to be Markov in scale. Hence each node in the image tree is dependent on the previous node at a coarser scale. The dependence relationship is shown by arrows in the figure. The image tree node is also assumed to be dependent on the template tree node at the same scale wherever the template is present. The template, Y, is usually much smaller than the image, X, and is shifted to find the matching location. At any given location of the template, all those image pixels at scale i that fall within a selected window of the template's range are denoted by the set $T^{(i)}$. Only the pixels in $T^{(i)}$ participate in the matching criterion computation developed in Sec. 3. The model is parameterized by the transition probabilities in the tree. These parameters are given by

$$\begin{aligned} \theta_{jkl}^{(i)} &\triangleq P\{X_s^{(i)} = j \mid X_{\partial s}^{(i+1)} = k, Y_s^{(i)} = l\} \\ \theta_{jk}^{(i)} &\triangleq P\{X_s^{(i)} = j \mid X_{\partial s}^{(i+1)} = k\} \end{aligned} \quad (1)$$

where ∂s is the parent node of s.

3. MATCHING CRITERION

The problem of matching is formulated as a detection problem. The two hypotheses are "template" and "no template". If $p_\theta(x \mid y)$ and $p_\theta(x)$ are the corresponding probability mass functions (pmfs), the generalized log

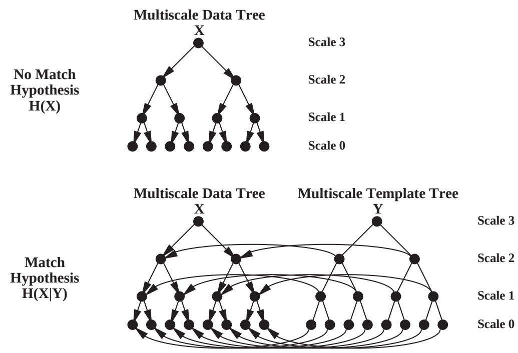


Figure 1: Multiscale data and template trees.

likelihood ratio is given by

$$L(x,y) = \log\left(\frac{\max_\theta p_\theta(x \mid y)}{\max_\theta p_\theta(x)}\right) \quad (2)$$

Since the log likelihood ratio is zero outside the window occupied by the template, it need be computed only over the extent of the template, $T^{(i)}$. It can be shown that under the model of Sec. 2, the log likelihood has the form

$$\begin{aligned} L(x,y) &= \hat{I}(x,y) \\ &= \hat{H}(x) - \hat{H}(x \mid y) \end{aligned} \quad (3)$$

where $\hat{H}(x)$ and $\hat{H}(x \mid y)$ are the estimated entropy and conditional entropy computed using the multiscale model.

The entropies can be estimated efficiently from a knowledge of the transition rates at each node of the image tree. To compute these, we first generate a histogram of the data and template trees. The histogram has an entry for each distinct value of the parent node, child node and corresponding template node. If $h_{jkl}^{(i)}$ is the entry for the child node value j, parent node value k and template node value l at resolution i, then

$$h_{jkl}^{(i)} = \sum_{s \in T^{(i)}} \delta(X_s^{(i)} = j, X_{\partial s}^{(i+1)} = k, Y_s^{(i)} = l) \quad (4)$$

where δ is an indicator function becoming 1 whenever all the conditions are satisfied. If the marginal histograms are obtained by summing the joint histograms along appropriate indices such as,

$$h_{jk}^{(i)} = \sum_l h_{jkl}^{(i)}, \quad (5)$$

then the final form of the mutual information at scale r is given by

$$\hat{I}(X^{(r)}, Y^{(r)}) = \textstyle\sum_{i=r}^{L-1}$$
$$\left[\textstyle\sum_{jkl} h_{jkl}^{(i)} \log\left(\frac{h_{jkl}^{(i)}}{\sum_j h_{jkl}^{(i)}}\right) \; - \; \sum_{jk} h_{jk}^{(i)} \log\left(\frac{h_{jk}^{(i)}}{\sum_j h_{jk}^{(i)}}\right)\right] \quad (6)$$

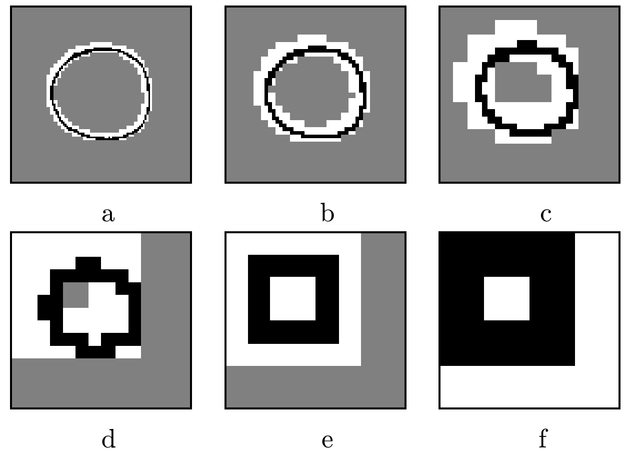


Figure 2: a)-f) Windowing with increasing scale. Data pixels corresponding to the grey regions are not included in the histogram computation.

The indices j,k and l are either 0 or 1. Hence the summations in (6) at any i contain at most 8 terms. So the computation of the matching criterion $\hat{I}(.,.)$ is dominated by the histogram computation at each resolution given by (4).

4. COMPUTATIONAL ISSUES

To generate the multiscale data tree the image is converted to an edgemap. The parent pixels for any resolution are generated by computing the logical "or" of the four children. Different resolutions of the template can be constructed in a similar way. It should be noted that there is no stochastic model assumed for the template.

Let the nodes corresponding to edge pixels be represented by 1 and those that are not by 0. Then, in the data tree, whenever a parent node is 0, all its children are also 0's. The entropy of such a configuration is zero, and hence does not contribute to the mutual information metric. Therefore, the computation required to match at any given scale is of the order of the number of edge pixels in $T^{(i)}$.

5. WINDOWING

If the window $T^{(i)}$ is too large, then, the matching criterion becomes sensitive to spurious edges and occlusion. This is especially true when the template has large areas of no detail. To avoid such behaviour, we restrict $T^{(i)}$ to only those data pixels that lie in the viscinity of the edge pixels of the template. This fact is illustrated in Fig. 2 for a specific choice of windowing. The data pixels corresponding to the black and white regions belong to $T^{(i)}$. We have observed that windowing significantly improves the performance of the matching criterion.

6. MULTISCALE SEARCH

In order to locate the template in the image, we must determine the template shift which maximizes the matching criterion. Ideally, we could compute the matching criterion for all possible shifts at the finest resolution; however, such a full search approach is computationally very expensive. Alternatively, we propose a course-to-fine search heuristic which evaluates the criterion only at small number of high resolution shifts, but effectively finds the maximum in most cases.

More specifically, let $C(i, N_i)$ denote the cost of computing the matching criterion for N_i shifts of the template at resolution i. Then, since the number of template pixels increases by 4 with each resolution, we may assume

$$C(i, N_i) = \alpha N_i 4^{-i}$$

where α is a constant. At each resolution, we need only search the template shifts corresponding to the locations of the data pixels. Therefore, the number of distinct template shifts at resolution i is proportional to 4^{-i}. Consequently, the total computation required for a full search at resolution i is

$$C(i, 4^{-i}) = \alpha 16^{-i} \ .$$

This rapid exponential growth in computation is highly undesirable.

A variety of techniques have been proposed for approximately solving such problems using search heuristics [6]. One reasonable approach is the best-first search method which determines the M_i best candidates at resolution i, and then, at resolution $i-1$, computes matches only at the candidate shift and its 8 neighboring shifts. Using this approach, the total computation at resolution i is given by

$$C(i, 9M_{i+1}) = \alpha 9 M_{i+1} 4^{-i} \ .$$

The performance of the best-first search will still depend heavily on the choice of M_i. Our approach is to balance the total computation at each resolution. This insures that an excessive amount of computation is not expended at fine resolutions. Specifically, we chose

$$M_i = k(4\rho)^{i-1}$$

where ρ is a constant which is generally greater than 1. Larger values of ρ allocate greater computation to coarser resolutions, and therefore improve the robustness of the search. Notice that $M_1 = 1$, insures that minimum computation is done at the finest resolution. However, setting $k \geq 2$ searches for the k largest values of the matching criterion facilitating the detection of the template occuring at more than one location. Both ρ and M_1 are to be chosen to get good results. We also note that for sufficiently large i, M_i will exceed the maximum number of possible shifts. In this case, a full search is performed.

7. RESULTS

The proposed method is applied on two image-template pairs shown in Figs. 3 and 5. The mutual information computed exhaustively at each resolution is shown in Fig. 4 for Image 1. A choice of $M_1 = 1$ and $\rho = 1$ yielded the correct solution. For Image 2, we used $M_1 = 2$ to detect both the eyes. For this choice of M_1 we had to use $\rho = 2$ to obtain correct detection of the template positions. The positions of the template at which mutual information takes its highest and second higest values are highlighted in Fig. 5. The times taken for the full search procedure (incuding the time taken to build the multiresolution pyramids) and for our partial search procedure are shown in Table. 1. The speed-up obtained is about 143 for Image 1 and about 312 for Image 2 which is significant. The match in Image 1 inspite of corrupting lines shows the advantage of windowing.

Image	Image Size	Template Size	Full Search	Partial Search
1	196x220	98x100	54.48	0.38
2	500x758	128x176	1095.68	3.51

Table 1: Run time of the algorithm in seconds for full search and partial search on two different images.

8. REFERENCES

[1] C. A. Bouman and M. Shapiro, "A Multiscale Random Field Model for Bayesian Image Segmentation," *IEEE Trans. on Image Processing*, vol. 3, no. 2, pp. 162-177, March 1994.

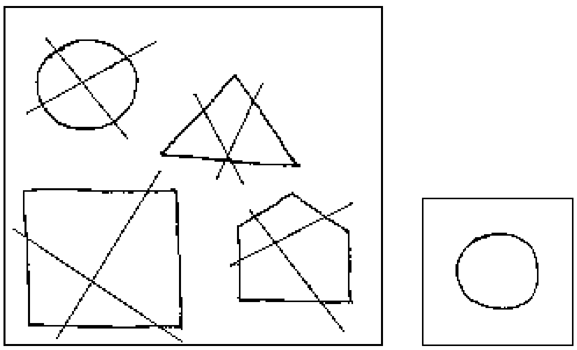
[2] M. R. Luettgen, W. C. Karl, and A. S. Willsky, "Efficient Multiscale Regularization with Applications to the Computation of Optical Flow," *IEEE Trans. on Image Processing*, vol. 3, no. 1, pp. 41-64, Jan. 1994.

[3] D. Tretter, C. A. Bouman, K. W. Khawaja, and A. A. Maciejewski, "A Multiscale Stochastic Image Model for Automated Inspection," to appear in *IEEE Trans. on Image Processing.*

[4] P.J. Burt, "Smart Sensing within a Pyramid Vision Machine," *Proceedings of the IEEE,* vol. 76, no. 8, pp. 1006-1015, August 1988.

[5] D. P. Huttenlocher, G. A. Klanderman, and W. J. Rucklidge, "Comparing Images Using the Hausdorff Distance," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. 15, no. 9, pp. 850-863, Sep. 1993.

[6] J. Pearl, Heuristics: Intelligent Search Strategies for Computer Problem Solving, Addison-Wesley, 1984.



a b

Figure 3: a) Image 1 b) Template.

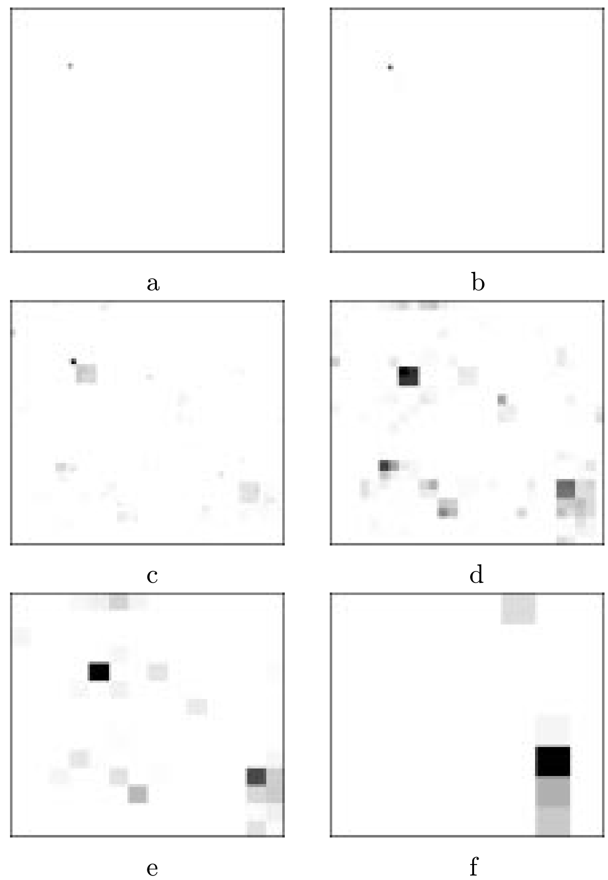
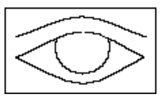


Figure 4: a)-f) Mutual information at different scales when the shapes in the image are corrupted with lines.

a



b

c

d

Figure 5: a) Image 2 b) Template c) Template location corresponding to highest mutual information d) Template location corresponding to second highest mutual information.