Revised version submitted to J. Elect. Imaging, Feb. 15, 2000

Pattern Matching using the Blur Hit-Miss Transform

Dan S. Bloomberg and Luc Vincent

Xerox Palo Alto Research Center, Palo Alto, CA 94304

Abstract

The usefulness of the hit-miss transform (HMT) and related transforms for pattern matching in document image applications is examined. Although the HMT is sensitive to the types of noise found in scanned images, including both boundary and random noise, a simple extension, the Blur HMT, is relatively robust. The noise immunity of the Blur HMT derives from its ability to treat both types of noise together, and to remove them by appropriate dilations.

In analogy with the Hausdorff metric for the distance between two sets, metric generalizations for special cases of the Blur HMT are derived. Whereas Hausdorff uses both directions of the directed distances between two sets, a metric derived from a special case of the Blur HMT uses just one direction of the directed distances between foreground and background components of two sets. For both foreground and background, the template is always the first of the directed sets. A less restrictive metric generalization, where the disjoint foreground and background components of the template need not be be set complements, is also derived. For images with a random component of noise, the Blur HMT is sensitive only to the size of the noise, whereas Hausdorff matching is sensitive to its location. It is also shown how these metric functions can be derived from the distance functions of the foreground (FG) and background (BG) of an image, using dilation by the appropriate templates.

The Blur HMT can be used as a fast heuristic to avoid more expensive integer-based matching techniques, and it is implemented efficiently with boolean image operations. The FG and BG images are dilated with structuring elements that depend on image noise and pattern variability, and the results are then eroded with templates derived from patterns to be matched. Subsampling the patterns on a regular grid can improve speed and maintain match quality, and examples are given that indicate how to explore the parameter space. Truncated matches give the same result as full erosions, are much faster, and for some applications can be performed at a restricted set of locations.

Keywords: pattern matching, scanned image, hit-miss transform, Hausdorff distance, blur hit-miss, image morphology, OCR

1 Introduction

Pattern matching techniques are critical for all aspects of the analysis of document images. Documents are typically scanned into a binary image, and many of the operations subsequently performed, both for page segmentation and character identification, use the pattern matching techniques (e.g., *erosion* and its dual, *dilation*) of binary image morphology. There are several reasons: they are implemented by fast boolean operations; they can be used either for extracting or extending pixel aggregations, both for direct use in later image processing and for subsequent analysis; they are translationally invariant; they can be used to treat both foreground (FG) and background (BG) simultaneously; and they can be used without regard to connected component analysis. Further, there exist a variety of methods for controlling the noise immunity of these operations.

One of the most important uses of pattern matching is in the analysis of character shapes. For binary input, the result of image processing can be either binary or gray (integer value) images. Binary results are much faster to compute, but they contain less information. Even for binary output, the internal operations can be integer or boolean. For integer operations, such as convolution and thresholded convolution (rank order filters[8]), some level of noise immunity is achieved, but at the price of doing expensive arithmetic operations on each pixel.

We use the term *template* to refer to the pattern of FG and BG pixels that are to be matched in the image. The *hit-miss transform* (HMT) is a faster boolean operation that performs translationally-invariant matching between both the FG and BG of template and image sets. However, it is prone to error from noise because exact matches are required between image and template in both the FG and BG. Because of the simplicity and power of the HMT, there have been many attempts to use it for pattern matching. The usual approach is to choose a subset of the template pixels, typically sparse. We cite a few examples.

Zhao and Daut[3] gained noise immunity, relative to a HMT, by using either boundary pixels of eroded FG and BG templates, or skeletons of these templates, as structuring elements for the HMT. Wilson[15] automated the design of the structuring elements through a training process that searched for the smallest subset of pixels that would attain the desired level of discrimination. Kraus and Dougherty[5] generated a sparse set of structuring elements by thresholding a single grayscale instance of each character. Appropriate choice of thresholds is the critical element: if chosen too conservatively, the subset is too sparse and lacks discriminatory ability; if chosen too tightly, instances with atypical variation are missed. Gillies[4] took a somewhat different approach, accumulating statistics from instances of each character, and thresholding the aggregates to generate non-sparse structuring elements. From these, multi-pixel features were extracted and used to train a classifier for character discrimination.

One characteristic that these methods have in common is an attempt to compensate for *image noise* by altering the *template*, and leaving the image alone. We argue here that although it is useful to choose a subset of template pixels, it is important to alter the image before performing the HMT. The *blur hit-miss transform* (BHMT) has been introduced to do precisely this[1]. Unlike the HMT, the BHMT performs the match between template and image, for both FG and BG, with a variable degree of tolerance to alignment of image and template pixels. The "blur" parameter specifies the maximum distance allowed between a template pixel and the nearest image pixel, in order to constitute a match for that template pixel. Stated this way, there is an interesting relation between the BHMT and the Hausdorff metric for the distance between two sets, but the differences are important for their uses in applications.

To understand the usefulness of the BHMT, it is necessary to consider the origin of noise in scanned document images. We postulate a simple model, where variability between instances in the image is caused by two different processes. One type is *boundary noise*, caused by the binarization process along the edge of an object. Depending on the sub-pixel alignment of scanned objects with scanner pixels, considerable edge variation occurs. This boundary noise is typically restricted to a width of two pixels, including both FG and BG boundary pixels. The second type is *random noise*, either generated in printing or due to scanner defects such as dirt on the platten. This is assumed to occur independently of the pixels in the scanned object, and is most often observed as isolated FG "pepper" pixels surrounded by BG. It should be noted that both types of noise occur in *boundary pixels*, defined to be pixels of either FG or BG that are adjacent to a pixel of the opposite type. Thus, operations that treat boundary pixels appropriately will influence both types of noise.

Even without random noise, boundary noise will defeat an HMT that uses boundary pixels in the template. Therefore, when computing matches, it is necessary to give little or no weight to the boundary pixels. On the other hand, the non-boundary pixels, because of their high correlation between template and image, are critical for matches. Differences occurring between non-boundary pixels in image and template, although relatively rare, will defeat both a matching technique like HMT, that requires an exact match of all pixels, and a metric such as Hausdorff, that is sensitive to such "outliers".

The paper is organized as follows. In Sec. 2.1 the BHMT is defined, and the method in which it provides immunity to both types of noise is described qualitatively. In Sec. 2.2, the Hausdorff metric is introduced, and the connection between this measure on sets and operations using morphology is made. Also, an illustration is given to show why the Hausdorff metric is not appropriate for matching templates to noisy images. In Sec. 2.3, two metric functions are constructed, that are related to special cases of the BHMT. The same example is then used to show how the BHMT succeeds in matching templates to noisy images. Then in Sec. 2.4, the BHMT metric functions are again derived, this time from dilations by the template of the distance function for the image. In Sec. 3, several methods for efficiently implementing the BHMT are described, including subsampling the template. Some experimental results are given in Sec. 4 to illustrate the use of the BHMT in identifying characters, and the major findings of the paper are summarized in Sec. 5.

We end this section with an illustration, in Fig. 1, of the family of rank and blur template matching operations. The HMT generalizes the erosion to operations that match in both FG and

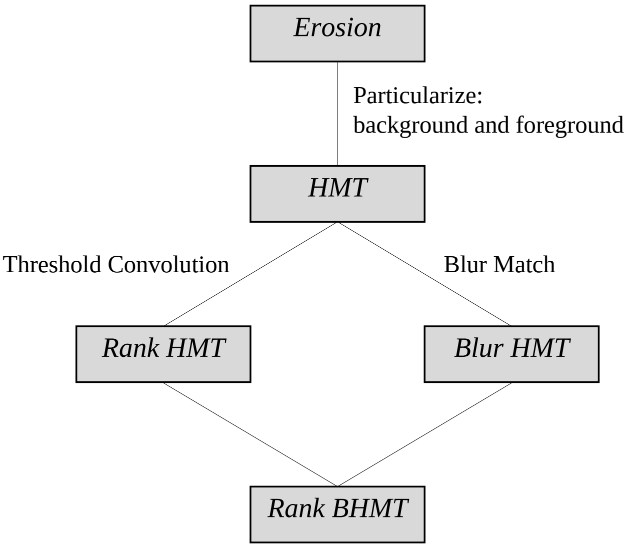


Figure 1: Family of rank and blur template matching operations.

BG. The rank operations take thresholds on convolutions, whereas the blur operations remove boundary pixels appropriately before doing strict matching. The *rank* HMT, a relatively expensive integer operation, requires co-location of image and template pixels, but eases the constraint on the number of matches. The rank HMT and the BHMT can also be combined into the *rank* BHMT, in which a match is accepted if only a given number of template pixels are within a given distance of the nearest image pixel. In the sequel, we concentrate on the BHMT, but we give one example of the use of the *rank* BHMT.

2 Blur HMT

2.1 Basic definitions

We are strictly interested in the discrete case of sets and functions defined on $\mathbb{Z}^2$, although extensions can be made to the continuous case or higher dimensions. The basic morphological operations are *erosion* and *dilation*. The erosion of a binary image X by a structuring element (SE) B is the set operation defined by

$$X \ominus B \;=\; \bigcap_{b \in B} X_{-b} \;=\; \{x \in \mathbb{Z}^2 \mid B_x \subseteq X\} \tag{1}$$

where X_{-b} is the translation of image X by $-b$. The second definition states that erosion generates a set with a non-empty result at every location where the translate of B fits entirely within X. The dilation of an image X is defined

$$X \oplus B \;=\; \bigcup_{b \in B} X_b \;=\; \bigcup_{x \in X} B_x \;=\; \{x + b \in \mathbb{Z}^2 \mid x \in X, b \in B\} \tag{2}$$

The first and second definitions state that dilation generates a set composed of the union of translations of X by elements in B, and *v.v.* Note that $\ominus$ and $\oplus$ are *not* the original definitions of Minkowski subtraction and addition, respectively, which require an inversion of the SE about its center[12].

The HMT is a morphological template matcher whose definition is based on the erosion operator[12]. The HMT of a binary image X by a disjoint pair (τ_f, τ_b) of SEs is defined as the set transformation

$$X \otimes (\tau_f, \tau_b) \quad = \quad (X \ominus \tau_f) \;\cap\; (X^C \ominus \tau_b) \tag{3}$$

where X^C is the set of BG pixels of X. The HMT generates a set with non-empty result at every location where both the FG SE τ_f fits entirely within X and the BG SE τ_b fits entirely within X^C, the complement of X. It is common to speak of the elements in τ_f as *hits*, of elements in τ_b as *misses*, and elements not in their union as *don't-cares*.

There are several methods for reducing the sensitivity to boundary noise. We can erode the template SEs by the blur SEs, or dilate the image by the blur SEs, all prior to the HMT. Eroding the template removes its boundary pixels from consideration, whereas dilating the image removes the image boundary pixels. The random noise pixels are also affected: eroding the template opens up FG and BG holes, so they are not included in the match; dilating the image closes up holes (i.e., removes salt and pepper) in FG and BG. The results of these operations (followed by the HMT) differ, and the choice must be made based on the statistics of expected noise. For document images, the template is expected to be free of salt and pepper noise in situations where it can be generated by averaging a large set of instances. However, this is *not* true for the image. Salt and pepper noise in the image will prevent matches between FG and BG of the template, respectively. Generally, it is imperative to remove noise pixels from both the template and image before doing the HMT. For situations where random noise is more frequent in the image than in the template, we thus define the BHMT of a binary image X using the SE pair (τ_f, τ_b) for the template and the SE pair (β_f, β_b) for blur as follows (Also see [1]):

$$X \otimes (\tau_f, \tau_b; \beta_f, \beta_b) \quad = \quad (X \oplus \beta_f) \ominus \tau_f \;\cap\; (X^C \oplus \beta_b) \ominus \tau_b \tag{4}$$

It can be noted that there is no requirement that the SEs used for blur are symmetric about their center. Translation of the center of a SE simply results in translation of the dilated image. Also note that τ_f and τ_b are typically non-overlapping. Take for example the case where $\beta_f = o$, $\beta_b = o$ and $\tau_f \cap \tau_b \neq \emptyset$: it corresponds to a traditional HMT (no blur), using an overlapping pair of SEs. In such a case, for any set X, one can easily verify that $X \otimes (\tau_f, \tau_b) = \emptyset$: it is indeed impossible to translate (τ_f, τ_b) in such a way that the translated τ_f is included in X *and* the translated τ_g is included in X^C. With some blur, that is when sets β_f and β_b are not reduced to a single pixel, it is possible to obtain a non-empty BHMT results even when $\tau_f \cap \tau_b \neq \emptyset$. However the practical interest of using overlapping τ_f and τ_b is extremely limited. In the sequel, we typically assume that $\tau_f \cap \tau_b = \emptyset$

2.2 Relation between Hausdorff metric and morphology

The Hausdorff metric is a distance between sets that allows one to define a topology on the set of all possible sets in the plane[9]. In image terminology, it is a distance between the FG of two images. The relation between the Hausdorff metric and a pair of blurred template matches has been noted previously[2], and we present the connection here.

Define the distance function[11] from a point p to the nearest point in a set X to be $d(p, X)$. If $p \in X$, then $d(p, X) = 0$. For two sets T and I, define the *directed* Hausdorff distance[6] from $T \Rightarrow I$ as the maximum over the pixels in the set T of the distance from the pixel in T to its nearest pixel in I:

$$D(T, I) \quad = \quad sup_{t \in T} d(t, I) \tag{5}$$

For applications to document images, consider T to be a FG template and consider I to be a *windowed* subset of the image X with support equal to that of the template. Then for each position in the plane represented by X, there exists a windowed subset $I \subset X$ and a directed Hausdorff distance $D(T, I)$ between T and the co-located I. Suppose this distance is δ. If the set I is dilated by a disk of radius δ, the distance between the dilated set and T will be zero. Consequently, an erosion of the dilated I by T will give a non-empty result.

The Hausdorff metric D_H is formed symmetrically between T and I, as the maximum of the two directed Hausdorff distances:

$$D_H(T, I) \quad = \quad max\{D(T, I), D(I, T)\} \tag{6}$$

Its relation to the blurred match between template and windowed image subset is[2]

$$D_H(T, I) \quad = \quad inf\{\rho \geq 0 \ \mid \ T \subseteq (I \oplus \rho B) \text{ and } I \subseteq (T \oplus \rho B)\} \tag{7}$$

where B is the unit disk SE. This is a symmetrical relation between FG sets. If I and T are very similar, small dilations act only to reduce the distance contributions from boundary pixels. Thus, a single non-boundary noise pixel in either I or T can render the Hausdorff distance quite large.

The effects of noise are illustrated by the two sets shown in Fig. 2. Call the sets on the left and right T and I, respectively. We have chosen the template to be less noisy and slightly eroded with respect to the image. The directed Hausdorff distances generated by these sets are shown in Fig. 3. The left frame is the directed distance $D(T_{(x,y)}, I)$ from $T \Rightarrow I$, evaluated at each possible location

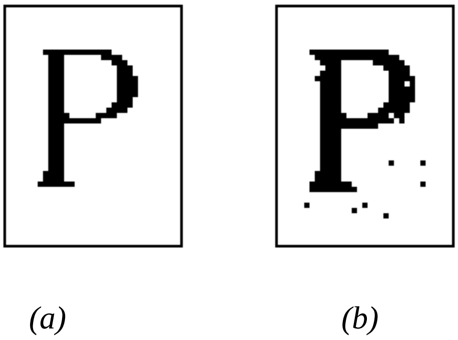


Figure 2: *Two sets used to illustrate effect of noise in Hausdorff distance. One percent of random noise was added to the set on the right.*

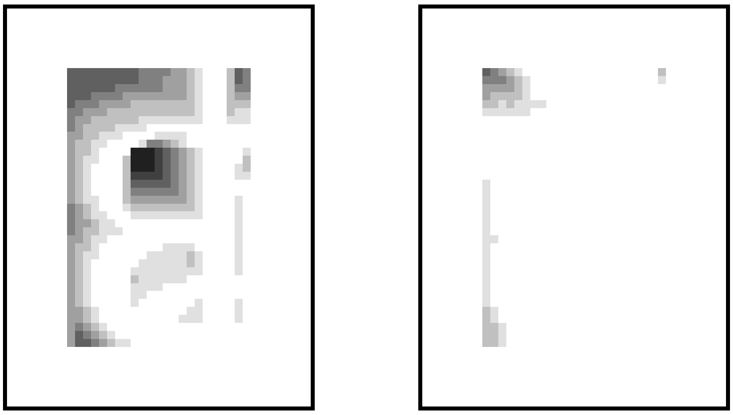


Figure 3: *Directed Hausdorff distances generated from sets in Fig. 2. The left and right frames are the directed distances $D(T_{(x,y)}, I)$ and $D(I_{(x,y)}, T)$, respectively.*

of T with respect to I. Darker values represent shorter distances. The best match, with a distance of 0, is from the dark region near the center. However, because of the noise in I, the distance $D(I_{(x,y)}, T)$ from $I \Rightarrow T$, shown in the right frame, has a very large value at that location. In fact, the smallest distances in $D(I_{(x,y)}, T)$ are found near the boundaries, due to clipping. This clipping effect is another complication of using a windowed directed Hausdorff distance from large image to a small template. For this example, the Hausdorff distance $D_H(T, I)$ is identical to the directed distance $D(I_{(x,y)}, T)$ for every translate of I, because the match between the two sets is entirely obscured by the noise in I.

2.3 Blur HMT metric

The BHMT produces a binary image representing locations of a match with a given amount of FG and BG blurring. In this section, we construct two BHMT-related distance metrics, in analogy with the Hausdorff metric. These are a generalized distance between image and template that, when thresholded, produce a BHMT for the value of FG and BG blur equal to the threshold.

When the template SE T is located on some windowed subset I of X, its center falls on coordinates (x, y). Label each subset I of X by this location (x, y). Then form a FG/BG metric D_{FB}, in analogy to D_H, that measures the directed distance between the FG and BG parts of T and I. The obvious choice for direction is $T \Rightarrow I$. Indeed, since the template is supposed to have been carefully chosen, it should exhibit little boundary pixel noise and no salt-and-pepper noise. Since the main purpose of the dilation operations used as part of the BHMT operation is to eliminate such noise before matching (See Eg. 4), it would not make sense to use direction $I \Rightarrow T$. For the same reason, using Hausdorff distance D_H typically does not work as well as this directed blur HMT metric because of adverse effects of salt-and-pepper type noise. For example, mainly because of pepper noise, the Hausdorff distance between the two sets of Fig. 2 would be very large. On the contrary, the *directed* Blur HMT distance from Fig. 2a to Fig. 2b would be dramatically smaller, which reflects the fact that these two sets simply are different instances of the same letter P.

Accordingly, we now dilate the full image X and use the (x, y) translate of T, $T_{(x,y)}$, to compare T with each subset I of X in computing the metric:

$$\begin{aligned} D_{FB}(T_{(x,y)}, X) &= inf\{\rho \geq 0 \mid T_{(x,y)} \subseteq (X \oplus \rho B) \text{ and } T^C_{(x,y)} \subseteq (X^C \oplus \rho B)\} \quad (8) \\ &= max\{D(T_{(x,y)}, X), D(T^C_{(x,y)}, X^C)\} \quad (9) \end{aligned}$$

To understand the relation between D_{FB} and the BHMT, consider the BHMT in its most simple form, with two disk SEs of equal radius for the blur and two SEs for the templates that are set complements. Setting $\tau_f = T$ and $\tau_b = T^C$, the BHMT is found by thresholding D_{FB}:

$$X \otimes (T, T^C; rB, rB) = \{(x, y) \in \mathbb{Z}^2 \mid D_{FB}(T_{(x,y)}, X) \leq r\} \quad (10)$$

The restriction in D_{FB} that the two SEs for the templates are set complements can easily be relaxed to the disjoint constraint for SEs in the HMT, by requiring only that $\tau_f \subseteq T$ and $\tau_b \subseteq T^C$. Then the metric D_{FB} is generalized to

$$D_{BHMT}(\tau_{f_{(x,y)}}, \tau_{b_{(x,y)}}, X) = inf\{\rho \geq 0 \mid \tau_{f_{(x,y)}} \subseteq (X \oplus \rho B) \text{ and } \tau_{b_{(x,y)}} \subseteq (X^C \oplus \rho B)\} \quad (11)$$

$$= max\{D(\tau_{f_{(x,y)}}, X), D(\tau_{b_{(x,y)}}, X^C)\} \quad (12)$$

with the thresholding relation

$$X \otimes (\tau_f, \tau_b; rB, rB) = \{(x,y) \in \mathbb{Z}^2 \mid D_{BHMT}(\tau_{f_{(x,y)}}, \tau_{b_{(x,y)}}, X) \leq r\} \quad (13)$$

We use the notation "BHMT*" to indicate the special case where the same dilation operator is used for both FG and BG. For the general case there are two BHMT distance metrics, one for FG and one for BG, and the BHMT is derived from from them by thresholding each separately and AND-ing the results.

For reasons of both efficiency and effectiveness we are usually interested in BHMT where $\tau_b \neq \tau_f^C$ and $\beta_f \neq \beta_b$. Examples will be given in Sec. 4. In Sec. 2.4, we arrive at a distance metric generalization for the BHMT by a different route.

Whereas D_H has bi-directional symmetry between two FG sets, and ignores the BG, D_{FB} has FG/BG symmetry but imposes a directionality on the relation between the two sets T and X. Unlike the Hausdorff metric, D_{FB} and D_{BHMT} are relatively immune to salt and pepper noise pixels in X. However, they are sensitive to noise in the themplate T, so in practice, one must ensure that the FG and BG of template T is free of salt and pepper noise.

Referring to Fig. 4, illustrating the BHMT for the same example as previously shown for Hausdorff, the directed distance $D(T, I)$ in the first frame is identical to the directed Hausdorff $D(T, I)$ in Fig. 3. (Although we now omit the (x, y) label on T, remember that these functions are defined over the set $(x, y) \in \mathbb{Z}^2$ of translates of T.) However, the second frame gives the directed distance for the BG, $D(T^C, I^C)$. The effect of noise on this distance is small: it is determined by the *size* of the noise in I, rather than the *distance* from the noise to T. The contribution from the BG does affect the overall match to a small extent, as shown by D_{FB} in the third frame. This is the special case of the BHMT metric D_{BHMT} where the FG and BG templates are set complements. Because the BHMT distances are always directed $T \Rightarrow X$ from the template to the large image, boundary effects occur only on the boundary of the image X.

Geometrically, the BHMT as we have defined it has the following interpretation: the "blur dilations" of image X and its complement X^C create a halo of pixels near image boundaries. This halo can be regarded as the "don't care" region of image X, and one of its nice characteristics is that it typically contains all the salt and pepper pixels. In a BMHT operation based on template T, a match is obtained when the FG of the template is located on FG pixels of the image or on "halo pixels", and when the BG of the template is located on BG pixels of the image or on "halo pixels".

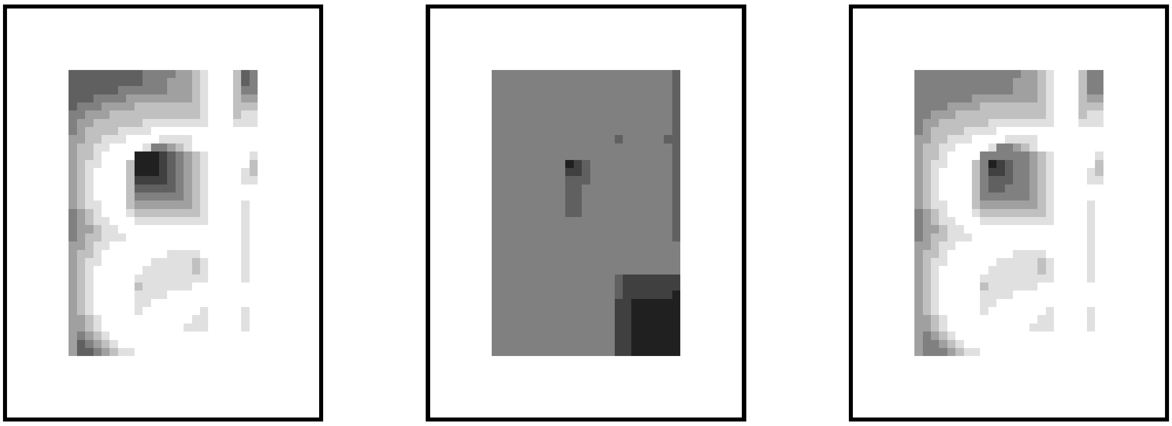


Figure 4: *BHMT distances generated from sets in Fig. 2. The first and second frames are the directed distances $D(T, I)$ and $D(T^C, I^C)$, respectively. The third frame is D_{FB}, the maximum of the two directed distances.*

In other words, this BHMT operation introduces a way to use “don’t care” pixels in the image as well as in the template used for matching. As such, it is a natural extension of the traditional hit-miss operation.

2.4 Blur HMT metric derived from distance function

Let us now consider a template T and a binary image X. For each translation $T_{(x,y)}$ of this template we are interested in computing the directed Hausdorff distance $D(T_{(x,y)}, X)$ between the translated template and X. This Hausdorff distance is equal to the smallest isotropic dilation size of X such that $T_{(x,y)}$ is included in this dilated image. Specifically, if B represents the 3×3 isotropic ball of the 8-connected distance function (3×3 square), we can write

$$D(T_{(x,y)}, X) = \min\{n \geq 0 \mid T_{(x,y)} \subseteq (X \oplus nB)\} \tag{14}$$

where $X \oplus nB$ represents the n-fold dilation of X by B, or equivalently the dilation of X by a $(2n + 1) \times (2n + 1)$ square structuring element, whose center is located on its geometric center.

Consider now for any pixel (x, y) the following distance function:

$$d_X(x, y) = \min\{n \geq 0 \mid (x, y) \in (X \oplus nB)\} \tag{15}$$

This distance d_X is simply a traditional distance function computed on the background of X: it assigns to each pixel its 8-connected distance to the nearest pixel of X. Obviously, any pixel (x, y)

included in X is given a value of 0 by this distance function. See [11, 13] for more information on distance funtions and their use in morphology.

Putting together the previous two equations we can write:

$$D(T_{(x,y)}, X) = \max\{d_X(i,j) \mid (i,j) \in T_{(x,y)}\} \tag{16}$$

In other words, the directed Hausdorff distance from $T_{(x,y)}$ to X can be obtained by extracting the maximal value of distance function d_X over pixels (i,j) belonging to the translated template $T_{(x,y)}$. Therefore:

$$D(T_{(x,y)}, X) = (d_X \oplus T)(x,y), \tag{17}$$

that is, the directed Hausdorff distance between template T translated to pixel (x,y) is equal to the value of the grayscale dilation of distance function d_X by template T at pixel (x,y).

The benefits of Equation (17) are numerous. First, it provides us with a computationally attractive method to compute a map of the quality of directed Hausdorff match at each pixel location: in this map, the pixels with value 0 correspond to locations where the translated template exactly fits inside image X, pixels with value 1 are the locations where the template fits inside a dilation of size 1 of X, etc. Second, looking at this map as a grayscale image, a number of techniques can now be used to extract its local minima, which provide us with the location of the local best matches of the template. In addition, the same method can be used with T^C and X^C, thereby providing a map of the matches between template complement and image complement.

We can one step further: to improve the "granularity" of this metric and speed up the algorithm significantly, we propose to use an asymmetric distance function in equation (17). Instead of defining it based on a 3×3 structuring element B, use a 2×2 square with the center of the structuring element at the upper-left. Using S for distance functions has two main advantages:

- The distance function based on S can be computed in a single raster-order pass through image X instead of the 2-passes required by traditional distance functions. See [14] for more details on this asymmetric distance and its use in fast morphological algorithms.

- The granularity of Hausdorff distance measurements is improved by a factor of 2. In the "match map" obtained through application of equation (17) using this asymmetric distance, all the pixels with an odd value $2p-1$ correspond to cases where a dilation with a $(2p) \times (2p)$ square provided the Hausdorff match. Using the distance based on 3×3 element B, one would not be able to differentiate between these matches and the lower-quality matches where a dilation by $(2p+1) \times (2p+1)$ was required for the match.

However, one caution in using such a non-symmetric dilation is that the results are shifted by one pixel with respect to the ones obtained using a symmetric dilation. The left and right frames of Fig. 5 show this distance function for the two sets in Fig. 2.

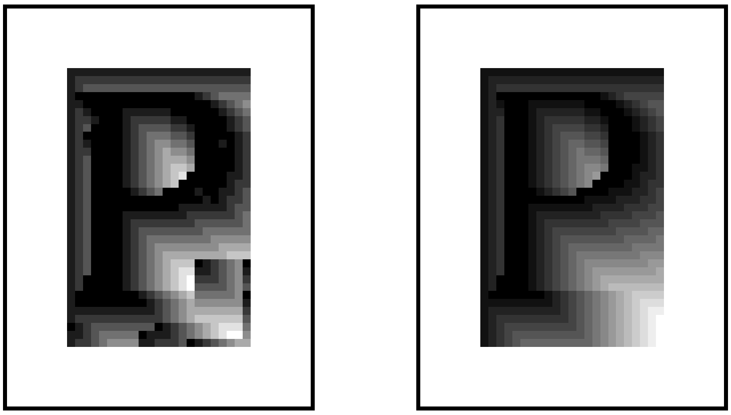


Figure 5: Distance function generated from the left and right sets in Fig. 2, respectively

.

Now, distance D_{BHMT} requires taking the maximum of two such directed distances, one computed for the FG and one for the BG. Because the distance function is asymmetric, the *location* of the result is translated to the SouthEast, relative to X, by an amount equal to the distance function itself. Thresholding D_{BHMT} with some value r generates the identical set as using blur dilation with an $r + 1 \times r + 1$ SE on BG and FG before the HMT.

This set of relations is illustrated in Fig. 6, which shows the sequence of operations that generate BHMT distance metrics and BHMT images. We start with the image X and FG and BG templates. Grid spacings of 2 in x and y directions are used for generating the FG and BG templates. In the FG, the distance function is found for X, and dilated with the FG template, giving the FG directed distance metric. The dual process in the BG yields the BG directed distance metric. The maximum of these gives the BHMT* metric, and for this example it is possible to find a threshold that yields a single match in the BHMT* set. The same result can be derived using the threshold individually on the FG and BG metrics, and AND-ing the result. With the threshold chosen, it should be noted that the thresholded FG metric yields matches in two locations, one of which is between the template and the “g” in the image. This match was not seen in the BG, which removed it from the BHMT. Details of the BHMT* metric are shown in Fig. 7.

In the general case, one would choose different threshold values (blur SEs) in FG and BG, in order to make the matching process more robust. This is the difference between the BHMT and the BHMT*. When an asymmetric distance function is used, which is analogous to the use of different asymmetric SEs for FG and BG blur, the thresholded binary images, which have different translations of the match with respect to the image, must be re-aligned before being AND-ed.

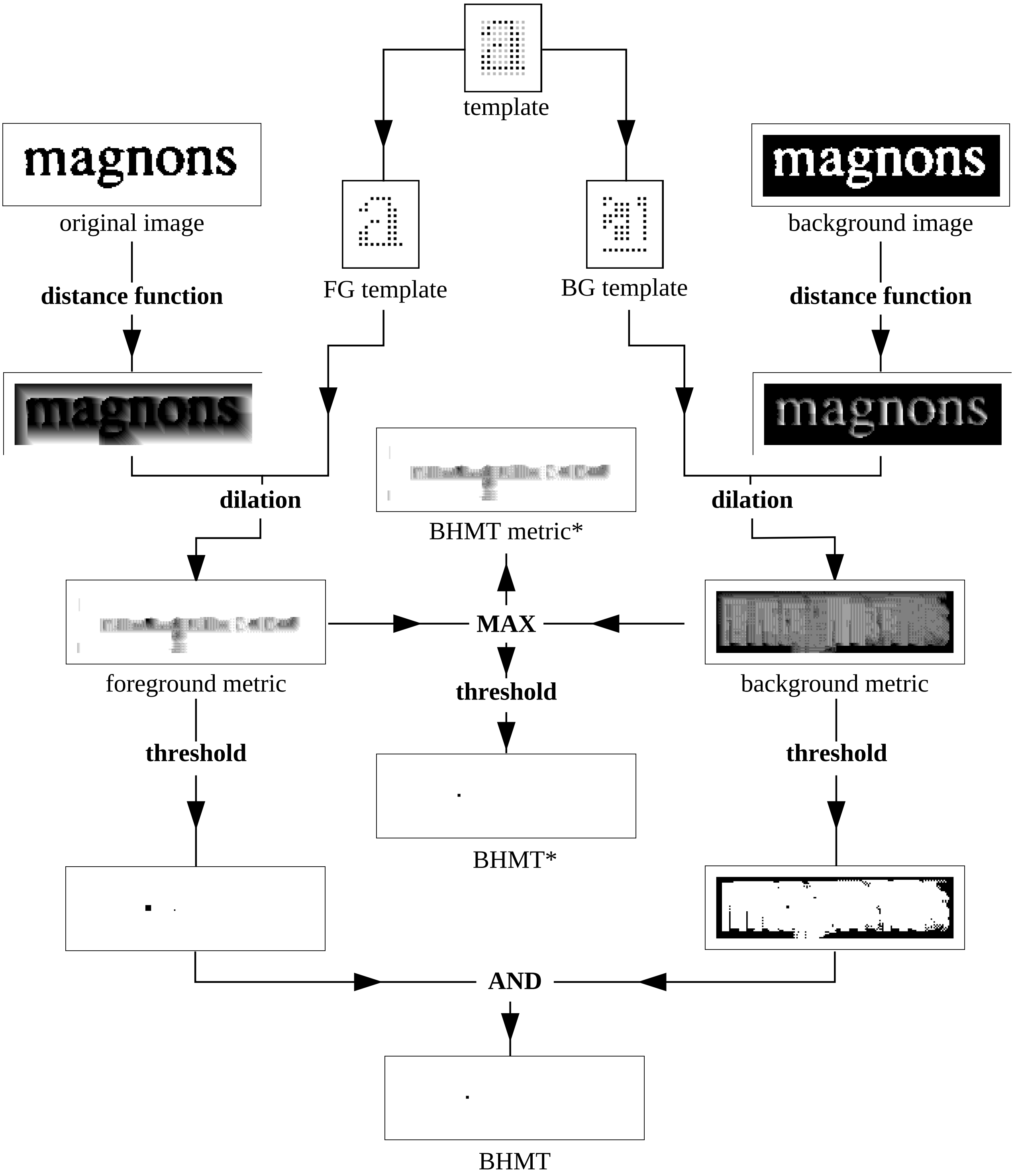


Figure 6: Sequence of operations that generate BHMT distance metrics and BHMT images.

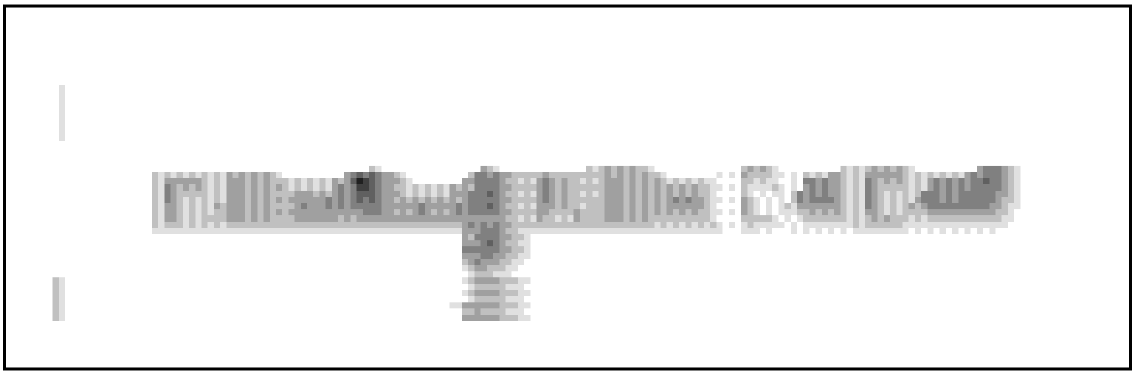


Figure 7: Detail of BHMT metric for example in Fig. 6*

3 Efficient implementations

Our interest is in finding relatively efficient implementations of the BHMT that are effective at locating matches without a large number of false positives. For character recognition, for example, the purpose is not to use the best possible pattern matchers, such as those used to estimate probabilities for templates in a maximum likelihood calculation[7]. Instead, we might want information that is good enough to be used as a heuristic for narrowing the search space for more computationally-intensive methods that do a better job of identifying characters. The rank operations, such as the rank BHMT, are less efficient than the BHMT because they require two (integer) convolutions by SEs, followed by thresholding. The BHMT uses only boolean operations.

We now give two approaches to the efficient use of the BHMT for identifying text characters in an image given a template.

3.1 Subsampled BHMT

To improve the efficiency in a direct implementation of the BHMT, it is possible to

- *Scale down both image and template.* Because we match all template pixels at each image position, the total number of pixels to be matched varies as the fourth power of the scaling parameter. The actual reduction in computation will be between the second power and the fourth power, depending on the implementation.

- *Subsample the template.* Suppose each template is subsampled by imposing a regular grid, with subsampling factors n_x and n_y. This has two effects. First, it decreases the computation required. The reduction varies from approximately n_y to the product $n_x \times n_y$, depending on the implementation. The second effect is that the subsampling tends to reduce the overall template dimensions, effectively augmenting the blur in the image.

The template can also be subsampled by choosing a random subset of template pixels, rather than a rectangular grid, but for a given number of template pixels chosen, the matching is significantly more accurate when a rectangular grid is chosen. Results using rectangular subsampled gridding of the template are given in Sec. 4, where it will be seen that some choices of subsampling greatly improve the results.

The first step is to perform the blur dilations on both the FG and BG of the image. This can then be used for a multiplicity of templates. The BHMT can be implemented in several ways. For fully parallel methods, each erosion can be formed separately by the usual set of translations and ANDs, where the unit of operation can be anything from the pixel to the entire image. The BHMT results by taking the intersection of the FG and BG results.

Matches of a template to a character in the image typically succeed at more than one (x, y) location. So that we do not over-count the matches, after the BHMT it is necessary to identify the matches by labelling the 8-connected components in the image. For the sparse BHMT images, this is relatively fast compared to the BHMT itself.

3.2 Truncated BHMT

There is another implementation of the BHMT that is faster, somewhat serialized, and largely circumvents the labelling process itself. The idea is to truncate the matching process in each location at the first instance of failure. Each template can be composed of an array of words, with each word representing the pixels in a template row. Suppose both FG and BG templates are to be tested at some location (x,y). Choose the FG template and align its first row with the image to test for a match. (The test requires only three boolean operations: AND between template and image; XOR between this result and the template; test for 0.) If a line match is found, proceed to the next template row. Whenever a line match is not found, quit the process at (x,y) and move to the next image location. If a full FG match is found, repeat with the BG template. If both matches succeed, record the location (the labelling process).

Matches to a single image feature typically occur contiguously within some region that is comparable to or smaller than the blur size. The serial aspect is required to avoid recording multiple positions for an image feature that has already been matched. When a match occurs, move several pixels away before looking for the next match. For the same reason, when scanning successive image lines, it is useful to avoid regions where a match was found proximally on lines above. Truncation of the matching process reduces the computation time by a factor proportional to the number of lines in the FG and BG templates that have ON pixels.

The rank BHMT defined earlier in the paper can provide a finer control over the matching. The approach described in the previous paragraph also provides an efficient implementation of the rank BHMT. Rather than testing for zero, count the ON pixels in the template line that are OFF in the image. Accumulate this sum over succesive template lines until either the rank threshold for that template is exceeded or the full template is matched. Do this for both FG and BG, which generally have different thresholds. As in the BHMT, avoid searching for matches in the vicinity of any location where a full match has already been found.

The efficiency of the truncated approach to the BHMT can be estimated within a factor of two or so, depending on implementation details and the hardware. For most locations of the template(s), the match will fail on the first line, requiring about 10 machine instructions (MIs). Suppose on average that 20 MIs are required for each (x, y) location. A 400 MIPS machine can then match 20 million positions/second. For a document image where the vertical location of text baselines is known within ± 2 pixels, and where the textline width is 2000 pixels, matches are required at 10,000 positions for each textline. For a full page with 50 textlines, the matching time is about 25 milliseconds.

3.3 Use of truncated rank BHMT

The matching operation of the rank BHMT, performed in a truncated fashion and only on a small subset of image locations, can be used to build an efficient JBIG2 encoder for binary images. JBIG2 is a lossy encoding where similarly-shaped connected components are replaced by a single representative template and a set of locations in the image where this template is to appear. The JBIG2 standard specifies the file format, but not the encoding method.

The basis of the JBIG2 encoder is an unsupervised clustering procedure, including the shape-matching algorithm. Consider a two-pass method, where the image is pre-segmented into 8-connected components. In the first pass, each component is examined sequentially to determine if it is sufficiently similar to the representative of an existing class, and if not, it becomes the representative of a new class. The comparison is done using a truncated version of the rank BHMT, where both the template and the image component to be compared are of comparable size. To reduce computation, it is preferable to evaluate the match at just one relative location of template and image. This location can be chosen by aligning the centroids of the two images[10]. With the rank BHMT, the FG and BG of the image are dilated and the FG and BG of each template of similar dimensions are tested in a truncated way, as described in Sec. 3.2. Two thresholds, for FG and BG outliers, are set for each template. A template is considered to match an image component if both the FG and BG outliers fall below their thresholds. If more than one template matches an image component, the one with the smallest number of outlier pixels (i.e., the best fit) should be used.

Once the initial clustering is made, the instances within each cluster are combined, by again aligning the centroids, to make a less noisy template. These templates are then used for the second pass. All image components are sequentially compared with the new templates, and the best fit is chosen. If an image component does not match any template, it is used as the template for a new class, as in the first pass. The amount of image distortion, produced by the substitution of the templates for each instance in the class, is controlled by the blur size and the thresholds. The advantage of the rank BHMT over the rank Hausdorff is evidenced in the second pass, where the template noise is much less than the noise in the image components. As we have seen previously, small salt and pepper noise in the image components is removed by the dilations, so the thresholds can be set lower for rank BHMT than for rank Hausdorff.

4 Illustrative results for BHMT

In this section, the use of the BHMT for matching image characters is briefly explored. A number of parameters can be varied independently: the FG and BG blur of the image, the x and y grid subsampling of the template, and scale reduction of both image and template.

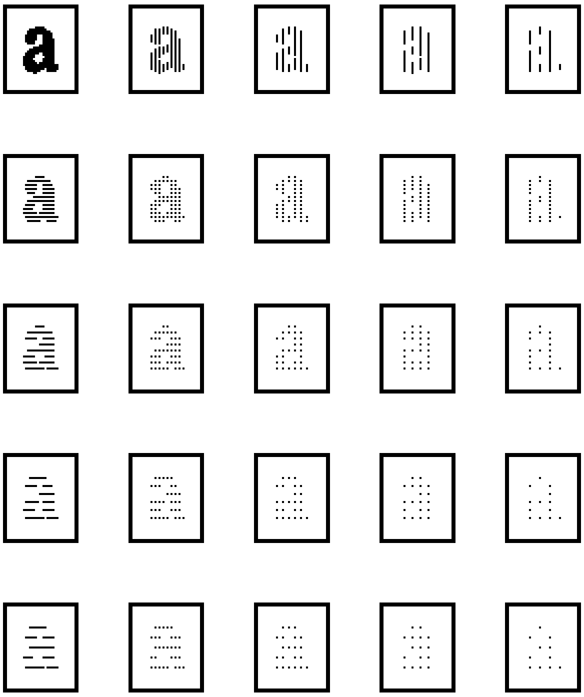
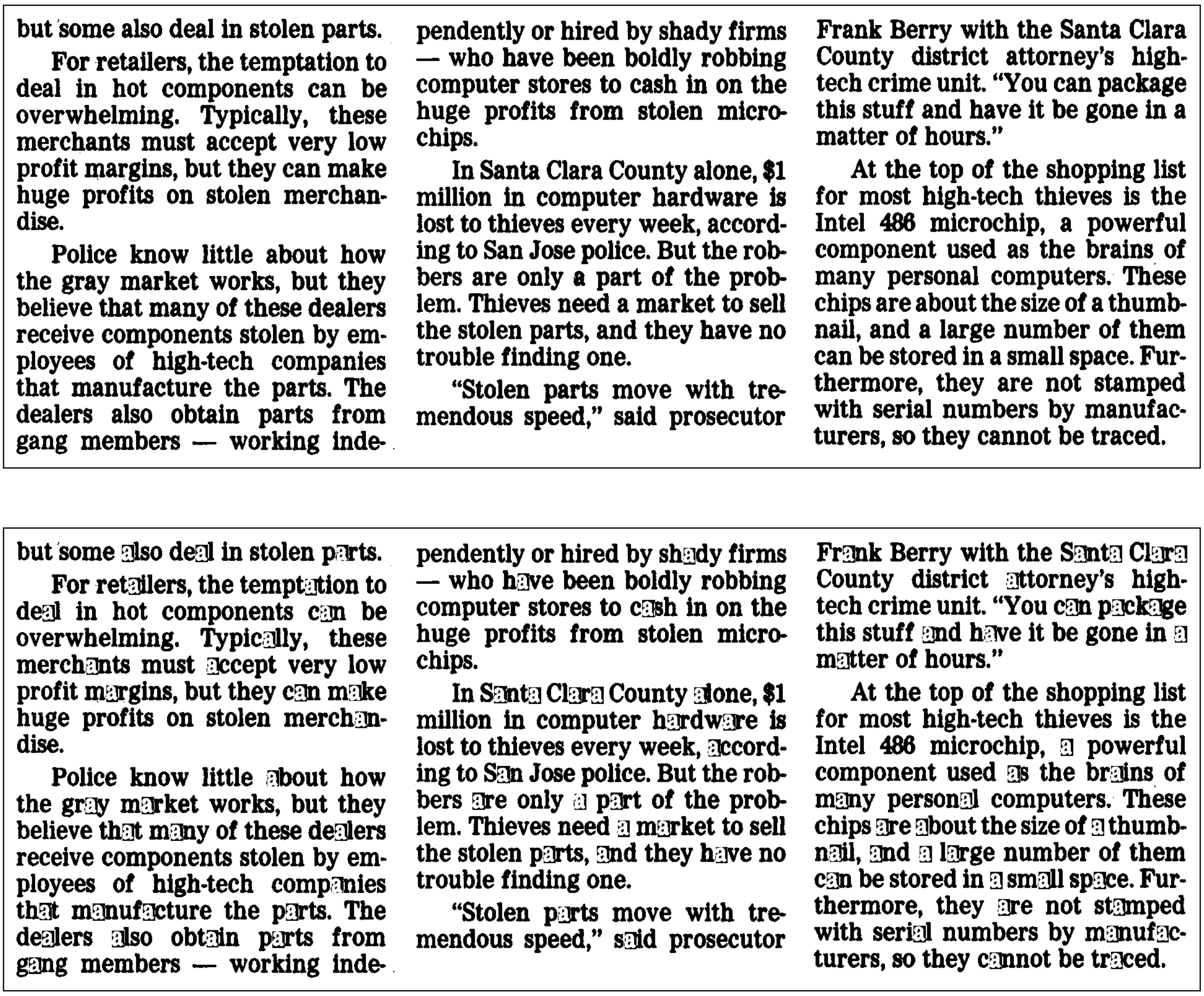


Figure 8: *Gridded FG templates, for (n_x, n_y) varying from (1,1) in the upper-left template to (5,5) in the lower-right template. Grid spacing n_x increases to right; n_y increases downward.*

To demonstrate the effect of blur and template gridding, an instance of the character “a” is chosen at random from the image at the top of Fig. 9, and is subsampled on a regular grid. A set of 25 gridded FG templates is shown in Fig. 8, where the n_x subsampling increases to the right (from 1 to 5), and the n_y subsampling increases downward. There is another set (not shown) for



but some also deal in stolen parts.

For retailers, the temptation to deal in hot components can be overwhelming. Typically, these merchants must accept very low profit margins, but they can make huge profits on stolen merchandise.

Police know little about how the gray market works, but they believe that many of these dealers receive components stolen by employees of high-tech companies that manufacture the parts. The dealers also obtain parts from gang members — working independently or hired by shady firms — who have been boldly robbing computer stores to cash in on the huge profits from stolen microchips.

In Santa Clara County alone, $1 million in computer hardware is lost to thieves every week, according to San Jose police. But the robbers are only a part of the problem. Thieves need a market to sell the stolen parts, and they have no trouble finding one.

"Stolen parts move with tremendous speed," said prosecutor Frank Berry with the Santa Clara County district attorney's high-tech crime unit. "You can package this stuff and have it be gone in a matter of hours."

At the top of the shopping list for most high-tech thieves is the Intel 486 microchip, a powerful component used as the brains of many personal computers. These chips are about the size of a thumbnail, and a large number of them can be stored in a small space. Furthermore, they are not stamped with serial numbers by manufacturers, so they cannot be traced.

but some also deal in stolen parts.

For retailers, the temptation to deal in hot components can be overwhelming. Typically, these merchants must accept very low profit margins, but they can make huge profits on stolen merchandise.

Police know little about how the gray market works, but they believe that many of these dealers receive components stolen by employees of high-tech companies that manufacture the parts. The dealers also obtain parts from gang members — working independently or hired by shady firms — who have been boldly robbing computer stores to cash in on the huge profits from stolen microchips.

In Santa Clara County alone, $1 million in computer hardware is lost to thieves every week, according to San Jose police. But the robbers are only a part of the problem. Thieves need a market to sell the stolen parts, and they have no trouble finding one.

"Stolen parts move with tremendous speed," said prosecutor Frank Berry with the Santa Clara County district attorney's high-tech crime unit. "You can package this stuff and have it be gone in a matter of hours."

At the top of the shopping list for most high-tech thieves is the Intel 486 microchip, a powerful component used as the brains of many personal computers. These chips are about the size of a thumbnail, and a large number of them can be stored in a small space. Furthermore, they are not stamped with serial numbers by manufacturers, so they cannot be traced.

Figure 9: Top: Example image. Bottom: characters matched by the BHMT for the templates in Fig. 8 (and the BG templates as well) have been highlighted.

		Blur 2,2		Blur 3,3		Blur 4,4		Blur 2,4		Blur 2,5		Blur 4,2	
n_x	n_y	*nmiss*	*nfp*	*nmiss*	*nfp*	*nmiss*	*nfp*	*nmiss*	*nfp*	*nmiss*	*nfs*	*nmiss*	*nfp*
1	1	84	0	19	0	4	0	8	0	4	0	83	0
1	2	80	0	10	0	0	32	1	0	1	0	77	0
1	3	71	0	16	0	4	0	6	0	2	0	68	0
1	4	65	0	16	0	4	1	4	0	0	0	64	0
2	1	80	0	5	0	0	1	0	0	0	0	79	0
2	2	70	0	1	1	0	74	0	0	0	0	69	0
2	3	54	0	4	0	0	11	0	0	0	0	53	0
2	4	46	0	4	0	0	54	0	0	0	0	46	0
3	1	65	0	1	0	0	15	3	0	3	0	61	0
3	2	47	0	0	0	2	130	0	0	0	0	43	0
3	3	55	0	0	1	0	150	0	0	0	5	51	0
3	4	14	0	0	31	13	328	0	0	0	33	14	16
4	1	39	0	0	0	-	-	4	0	4	0	34	0
4	2	0	0	0	32	-	-	0	0	0	3	0	168
4	3	16	0	0	21	-	-	1	0	1	6	9	100
4	4	1	0	6	193	-	-	0	0	1	45	11	609

Table 1: *Use of BHMT to identify 88 instances of the character "a", using a templated derived by subsampling one of those instances. For text of this size and thickness, the most stable region for matches is with with parameters near FG and BG blurs $(\beta_f, \beta_b) = (2, 4)$ and and grid spacings $(n_x, n_y) = (2, 2)$. For each set of FG/BG blurs, the numbers of misses and false positives are given for different grid spacings.*

the BG templates. In the upper image in Fig. 9, there are 88 instances of the character "a". These are highlighted in the bottom image, having been identified by a BHMT using the blur parameters $(\beta_f, \beta_b) = (2, 4)$ and grid spacings $(n_x, n_y) = (2, 2)$. With this combination, as with several others, all instances were identified and no false positives were found.

Results for five combinations of FG/BG blur factors and sixteen grid subsamplings are shown in Table 1. For each blur factor pair, the numbers of misses and false positives are given. For example, the two columns labeled "Blur2,4" used $(\beta_f, \beta_b) = (2, 4)$. With strict matching parameters (fine gridding, small dilation) there are no false positives and a significant number of misses. A few false positives are seen for Blur3,3, particularly for larger n_x. With Blur4,4, the number of false positives is significantly increased; this amount of dilation removes the differentiation of the "a"s from other similar characters. A few intermediate combinations succeeded in finding all occurrences without any false positives. As seen from Table 1, the optimal working region is near the parameters $(\beta_f, \beta_b) = (2, 4)$ and $(n_x, n_y) = (2, 2)$.

When FG and BG blur are not equal, there is an asymmetry in the results. Table 1 shows particular combinations, where the BG blur is larger than the FG blur, that give good matches. When FG blur is larger, as for Blur4,2, there are typically more misses and more false positives. The asymmetry exists because the image has large solid white areas that can give false positive matches to all BG templates; consequently, excessive FG dilation contributes significantly to these errors.

When matching multiple characters in a font and size, an optimum pair of FG and BG blur SEs can be chosen, and the template grid spacings can be individually optimized for each character, including use of different grid spacings for FG and BG templates. The BHMT exhibits significant immunity to random noise. For example, when random noise at the one percent level shown in (b) of Fig. 2 is added to the image, the numbers of missed and false positive characters from the BHMT are not significantly changed. The BHMT is typically more computationally efficient with coarser grid spacing, particularly in the y-direction.

5 Summary

Translationally invariant methods for pattern matching in scanned document images have no dependencies on pixel connectedness in either the image or template. We have focused on the most efficient techniques, that require only boolean operations. The basic operation, the HMT (which should be called the Hit-*And*-Miss transform!), is maximally sensitive to noise in *both* FG and BG.

Fortunately, there are ways to increase the noise immunity of the HMT. For extensions that use only boolean operations (as opposed to linear convolution and rank order filters), and considering the nature of binary pixel noise in both images and templates, we have argued that the BHMT is the best choice.

Considerable attention was devoted to distance metrics that can be derived from (special cases of) the BHMT. We began with the well-known relation between the Hausdorff metric and blurred template matches, and derived similar metrics for the BHMT. *This distance provides a measure of the goodness of fit of the template at every location in the image.* Comparing the Hausdorff and BHMT mechanisms of action on noisy document images, we showed why (1) the bi-directional symmetry of Hausdorff is problematic and (2) the uni-directional but FG/BG-symmetric BHMT provides immunity to both boundary and random pixel noise. An intuitive presentation of these differences is a primary goal of this paper.

We also showed how the BHMT* metric can be derived in the grayscale regime starting with distance functions for the FG and BG image. For most sensitivity, we choose an 8-connected asymmetric function that is generated in one raster scan and increments the distance to the South and East. These distance functions are then dilated with the FG and BG templates, and combined using the pixelwise Max operator.

The BHMT is useful for binary document image pattern matching tasks. We have shown the results of an experiment on pattern matching for characters, to illustrate the effects of FG and BG blur, and of regular subsamplings of the templates. Regular gridding of the templates gives far better results than using random subsets, for the same number of elements chosen. We also discussed methods for truncating the matches; this is much more efficient than using full erosions at every location. These truncation methods are also applicable to rank operations, which can be designed to have fewer matching failures than the BHMT itself.

References

[1] D. S. Bloomberg and P. Maragos, "Generalized hit-miss operations", *SPIE Conf. Image Algebra and Morphological Image Processing, Vol. 1350,* San Diego, CA, July 1990, pp. 116-128.

[2] H. J. A. M. Heijmans, *Morphological Image Operators*, Acad. Press, 1994, see p. 296.

[3] D. Zhao and D. G. Daut, "Shape recognition using morphological transformation", *J. Visual Comm. and Image Rep.* **2**, pp. 230-243, Sept 1991.

[4] A. M. Gillies, "Automatic gemeration of morphological template features", *SPIE Conf. Image Algebra and Morphological Image Processing, Vol. 1350,* San Diego, CA, July 1990, pp. 252-261.

[5] E. Kraus and E. Doughterty, "Segmentation free morphological character recognition", *SPIE Conf. Document Recognition, Vol. 2181*, San Jose, CA, Feb 1994, pp. 14-23.

[6] D. P. Huttenlocher, G. A. Klanderman and W. J. Rucklidge, "Comparing images using the Hausdorff distance", *IEEE Trans. PAMI* **15**, pp. 850-863, Sept 1993.

[7] G. Kopec and P. Chou, "Document image decoding using Markov source models", *IEEE Trans. PAMI* **16**, pp. 602-618, June 1994.

[8] P. Maragos and R. W. Schafer, "Morphological filters - Part II: their relations to median, order-statistic, and stack filters," *IEEE Trans. Acoust. Speech Signal Process.*, ASSP-35, pp. 1170-1184, Aug. 1987.

[9] G. Matheron, *Random Sets and Integral Geometry*, J. Wiley and Sons, NY, 1975.

[10] Kris Popat, private communication.

[11] A. Rosenfeld and J. L. Pfalz, "Distance functions on digital pictures", *Pattern Recognition* **1**, pp. 33-61, 1968.

[12] J. Serra, *Image Analysis and Mathematical Morphology*, Acad. Press, 1982.

[13] L. Vincent, “New trends in morphological algorithms”, *SPIE Conf. Nonlinear Image Processing II, Vol. 1451*, San Jose, CA, Feb 1991, pp. 158-169.

[14] L. Vincent, “Fast Opening Functions and Morphological Granulometries”, *Proc. SPIE Vol. 2300, Image Algebra and Morphological Image Processing V,* San Diego, CA, Jul 1994, pp. 253–267.

[15] S. S. Wilson, ”Training structuring elements in morphological networks”, *Mathematical Morphology in Image Processing*, ed. E. R. Dougherty, Marcel Dekker, NY 1992.