

On Hausdorff Distance Measures

Michael D. Shapiro
Palace of Dreams
120 Garfield Pl.
Brooklyn, NY 11215, USA
email: sifka@earthlink.net

Matthew B. Blaschko
Computer Vision Laboratory
University of Massachusetts
Amherst, MA 01003, USA
email: blaschko@cs.umass.edu

UM-CS-2004-071

ABSTRACT

A number of Hausdorff-based algorithms have been proposed for finding objects in images. We evaluate different measures and argue that the Hausdorff Average distance measure outperforms other variants for model detection. This method has improved robustness properties with respect to noise. We discuss the algorithms with respect to typical classes of noise, and we illustrate their relative performances through an example edge-based matching task. We show that this method produces a maximum a posteriori estimate. Furthermore, we argue for improved computational efficiency by tree-like subdivisions of the model and transformation spaces.

KEY WORDS

Hausdorff, Object Recognition, Spatially Coherent Matching

1 Introduction

Object detection often relies on metrics that describe the degree of difference between two shapes. The one-sided Hausdorff distance, $h(A, B)$, in this context is a measure between the set of feature points defining a model, A , and the set of points defining a target image, B , where

$$h(A, B) = \max_{a \in A} \min_{b \in B} \|a - b\|$$

and $\|\cdot\|$ is a norm of the points of A and B . Image matching using Hausdorff-based distances has been applied to many domains including astronomy [10], face detection [5][13], and word matching [8].

Several variations of this set metric have been proposed as alternatives to the max of the min approach in traditional Hausdorff matching that are less prone to outliers in the data. These include Hausdorff fraction, Hausdorff quantile [4], Spatially Coherent Matching [1], and Hausdorff average [3]. We analyze the form of these metrics and argue about their behavior under several classes of noise. Specifically, we determine whether there are discontinuities of the measure with respect to minor perturbations

of the point sets. We show analytically and experimentally that Hausdorff average has superior performance under these conditions as in [12].

2 A General Schema

Our general schema is as follows. We have an image I and a model M . We wish to discover if the object represented by M occurs in I , and if so where. Our concept of “where” is encapsulated in a set of transformations $\mathcal{T} = \{T\}$ which we think of as positioning a copy $T \oplus M$ of M in I . For each such position, we compute a *score* $s(T \oplus M)$. Among all transformations, we pick the one \hat{T} with the “best” score. If this score is sufficiently “good”, we announce that M is at position T in I . Otherwise, we announce that M is not in I .

3 Notation

Let us fix notation. Our model M consists of a vector of features,

$$M = (S_1, \dots, S_{N_M}).$$

After application of a transformation $T \in \mathcal{T}$ each feature S_i acquires a position

$$(x_i, y_i)_T = (x, y)(T \oplus S_i).$$

We will drop the T subscript when this is clear from context. Strictly speaking S_i itself does not have a position. It acquires one if we imagine an identity transformation placing it at some predetermined position in I . In general, \mathcal{T} need not act on the image plane. For example, if M is a three-dimensional model, \mathcal{T} may consist of rigid motions followed by projection. We will be interested in the case where the transformations are translations in the plane. Then

$$(x, y)(T \oplus S_i) = T((x, y)(S_i)).$$

If we like, we may assume that I consists of a similar vector of features, but this is not necessary. We only need to enquire whether a feature S_i of M is *matched* at a position (x, y) in I . When I and M are both sets of edge pixels, every feature of M is matched at every feature of I .

For each feature S_i , the *match locus*, $\mathcal{M}_I(S_i)$ is the set of positions at which S_i is matched in I .

$$\mathcal{M}_I(S_i) = \{(x, y) \mid S_i \text{ is matched at } (x, y)\}.$$

We will drop the subscript I when there is no ambiguity and that neither I nor M is empty.

When I and M are point sets (for example, pixels found by an edge detector), $\mathcal{M}(S_i)$ consists of all points of I . However, there may be more than one type of feature. For example, one type may consist of edge pixels, whereas another type may consist of points where a filter exceeds some threshold. If the features S_i and S_j are both edge

pixels, then $\mathcal{M}(S_i) = \mathcal{M}(S_j)$. Likewise if S_p and S_q are features for the same filter then $\mathcal{M}(S_p) = \mathcal{M}(S_q)$. However, we should not expect that $\mathcal{M}(S_i) = \mathcal{M}(S_p)$. Thus we may build different aspects of the object into our model and attempt to match them in the image.

We choose a distance function $d((x_1, y_1), (x_2, y_2))$ in the plane. Then the match set induces a *distance* for each translated feature,

$$d_{T,i} = d(T \oplus S_i) = \min\{d((x_i, y_i)_T, (x, y)) \mid (x, y) \in \mathcal{M}(S_i)\}.$$

For each T this gives us a vector of distances,

$$d_T = (d_{T,1}, \dots, d_{T,N_M}) = (d(T \oplus S_1), \dots, d(T \oplus S_{N_M})).$$

Each $d_{T,i}$ tells us how far the feature S_i is from its nearest match when M is placed at position T . Each instance of our schema can be described in terms of this vector. In each instance, we assume a fixed model and image.

4 Distance measures

4.1 One-sided Hausdorff distance

For each T , the *Hausdorff score*, $s_H(T)$, is $\max_i\{d_{T,i}\}$. We pick \hat{T} to minimize this score. We accept it if $s_H(\hat{T})$ is less than some threshold value, s_0 .

4.2 Hausdorff fraction

We fix a minimum acceptable distance k . For each T , *Hausdorff fraction score*, $s_{HF}(T)$ is cardinality

$$s_{HF}(T) = \frac{\#\{S_i \mid d_{T,i} < k\}}{N_M}.$$

We pick \hat{T} to maximize $s_{HF}(T)$. We accept \hat{T} if the fraction $s_{HF}(\hat{T})$ exceeds a threshold value s_0 .

4.3 Hausdorff quantile

We fix a quantile q . For each T , the *Hausdorff quantile score*, $s_{HQ}(T)$ is the q th quantile value of distances in d_T . We choose \hat{T} to minimize $s_{HQ}(T)$ and accept \hat{T} if $s_{HQ}(\hat{T})$ is less than some fixed value s_0 .

4.4 Spatially Coherent Matching

To define Spatially Coherent Matching, we need to define a *neighbor* relation on the features of M . We take the features of M to be the vertices of a graph. Two features are neighbors if there is an edge between them in this graph. We are interested in the case where the features are edge pixels. We will consider pixels to be neighbors if they are adjacent, horizontally or vertically. We fix a minimum acceptable distance k . For

each T , the *Spatially Coherent Matching score*, $s_{SCM}(T)$ is defined as follows. We take $G(T) = \{S_i \mid d_{T,i} < k\}$. We take $B(T)$ to be the set of edges with one vertex in $G(T)$ and one vertex not in $G(T)$. This counts the number of edges on the “boundary” of $G(T)$. We take

$$s_{SCM}(T) = \frac{\#G(T) - \#B(T)}{N_M}.$$

We choose \hat{T} to maximize $s_{SCM}(T)$. We accept \hat{T} if $s_{SCM}(\hat{T})$ exceeds some fixed value s_0 .

4.5 Hausdorff Average

For each T , the *Hausdorff average score*, $s_{HA}(T)$ is

$$s_{HA}(T) = \frac{\sum_i d_{T,i}}{N_M}$$

We choose \hat{T} to minimize $s_{HA}(T)$. We accept T if $s_{HA}(\hat{T})$ is less than some fixed value s_0 .

We will want to generalize this a little bit. Given a function f , we will want to consider a *Hausdorff functional average score*, $s_{HA,f}(T)$,

$$s_{HA,f}(T) = \frac{1}{N_M} \sum_i f(d_{T,i})$$

Note that Hausdorff average score is a special case of Hausdorff functional average score with $f(d_{T,i}) = d_{T,i}$.

5 Maximum a Posteriori Estimate

The Hausdorff functional average algorithm has a simple justification as a maximum a posteriori estimate. We follow the argument of [1] in a simplified form. We have a collection of hypotheses and must choose the one which maximizes the probability

$$p(h \mid I) = \frac{p(h)p(I \mid h)}{p(I)}.$$

Since I is fixed, we seek

$$h = \operatorname{argmax}_h p(h)p(I \mid h).$$

We take as our hypothesis space $\mathcal{T} \cup \emptyset$, where each hypothesis $T \in \mathcal{T}$ corresponds to the assertion that the object modeled by M occurs at position T in I , and the hypothesis \emptyset is the assertion that the object does not occur in I . Our prior knowledge consists of the assumption that $p(\emptyset) = \rho$ and that all positions T have equal probability C_1 . We assume that given a position T , the likelihood of I given T is

$$p(I \mid T) = C_2 \prod_i e^{-f(d_{T,i})}$$

We assume that when the object is not there, all images have equal probability, C_3 . This gives

$$p(h)p(I|h) = \begin{cases} (1-\rho)C_1C_2 \prod_i e^{-f(d_{T,i})} & \text{if } h = T \in \mathcal{T} \\ \rho C_3 & \text{if } h = \emptyset \end{cases}$$

In order for the presence of the object at position T to be more likely than the absence of the object, we need

$$(1-\rho)C_1C_2 \prod_i e^{-f(d_{T,i})} > \rho C_3$$

Since logarithms are monotonic, the argmax of this function is the argmax of its logarithm. Taking negative logarithms reverses the inequality, turns the argmax into argmin, and yields the inequality

$$-\ln(1-\rho) - \ln(C_1) - \ln(C_2) + \sum_i f(d_{T,i}) < -\ln(\rho) - \ln(C_3)$$

The best position \hat{T} is the one minimizing $s_{HA,f}(T)$, and this represents the best hypothesis if that value is less than

$$s_0 = \frac{\ln(1-\rho) + \ln(C_1) + \ln(C_2) - \ln(\rho) - \ln(C_3)}{N_M}.$$

6 Noise and stability:

We would like to investigate the behavior of these measures when there is noise. We will deal with two classes of noise, *distortion* and *stray features*, typically, stray pixels.

By distortion we mean disturbance of the coordinates of a feature or the points of its match set. Thus, we may have M with features (S_1, \dots, S_{N_M}) and M' with features (S'_1, \dots, S'_{N_M}) . The distortion is captured in the fact that for each T , we have $(x_i, y_i) = (x, y)(T(S_i))$ and $(x'_i, y'_i) = (x, y)(T(S'_i))$. We capture distortion between images I and I' by assuming that there is a function g so that for each i $\mathcal{M}_{I'}(S_i) = g(\mathcal{M}_I(S_i))$.

Stray pixels show up as extra features in the model or the image. Thus, we may have M with features (S_1, \dots, S_{N_M}) and M' with features $(S_1, \dots, S_{N_M}, S_{N_M+1}, \dots, S_{N_M+n})$. We assume that for each $j > N_M$ there is $i \leq N_M$ so that $\mathcal{M}(S_j) = \mathcal{M}(S_i)$. Thus we are not throwing in any new types of features, and none of these have empty match sets.

Stray pixels in the image show up as extra points in the match set. Thus, we may have an image I with $\mathcal{M}_I(S_i)$ and an image I' with match set $\mathcal{M}_{I'}(S_i) = \mathcal{M}_I(S_i) \cup X_i$.

We would like to know how robust our scores are with respect to noise. A robust score should only change a small amount if we introduce a small amount of noise. What we want is for the score to be a continuous function of noise. Since position is typically discrete, we will take a less purist approach to this. We will act as if noise can be made arbitrarily small, though this may not be the case. What is a small amount of noise? A distortion is small when for each pair of points (x, y) and (x', y') as above, the distance $d((x, y), (x', y'))$ is small. One might choose a stricter definition here. One might ask that the support of the distortion is small, that is, that only a small number of pairs of points have $d((x, y), (x', y')) \neq 0$. We will argue that this is too restrictive. We will say that the set of stray pixels is small when they are few in number compared to

the total number of features or pixels in the match set. We will restrict to stray pixels lying a fixed bounded distance from the remaining pixels. This is another way of saying that our image has a fixed size and the stray pixels lie in that image.

These are realistic classes of noise. If the model is derived from measurements, these are subject to inaccuracy. If the model comes from a photograph, distortion could result from inaccuracy of the photographic setup, for example, as to angle or distance to the object. This sort of difference corresponds to taking a photograph, applying a rigid motion to the object and taking a second photograph. When that motion is small, the resulting difference between the two photographs is a small distortion. This distortion extends to the entire model. This is why we reject restricting the support of a distortion as a requirement for it being small.

Differing edge detectors are likely to disagree about exact placement of edge, producing another source of distortion. Likewise a model that is the result of extracting edges from an image can easily acquire stray pixels.

Stability with respect to distortion carries over to stability with respect to change of transformation T . If T and T' are close then so are $(x,y)(T(S_i))$ and $(x,y)(T'(S_i))$, and thus, so are d_T and $d_{T'}$.

7 Stability results

Let us fix notation. We are given an image I , and a model M . Suppose that I' and M' differ from these by distortion, and that I'' and M'' differ from these by stray pixels.

For each fixed T have

$$\begin{aligned} d_T &= d_T(I, M) &= (d_{T,1}, \dots, d_{T,N_M}) \\ d'_T &= d_T(I', M') &= (d'_{T,1}, \dots, d'_{T,N_M}) \\ d''_T &= d_T(I, M'') &= (d_{T,1}, \dots, d_{T,N_M}, d_{T_{N_M+1}}, \dots, d_{T_{N_M+n}}) \\ d'''_T &= d_T(I'', M) &= (d''_{T,1}, \dots, d''_{T,N_M}) \end{aligned}$$

To investigate stability, we must observe how these vectors differ and the differing results that our scores (and hence our algorithms) produce as a result.

There are several sources of instability that apply to all of our methods above. The first is that even a small change in the score $s(\hat{T})$ can push it above or below the threshold value s_0 . The second is that the operations argmax and argmin are discontinuous. That is, if $s(T)$ and $s(T')$ are near optimal and close in value, then either can be promoted to \hat{T} by a small change in $s(\cdot)$, even though T and T' may be widely separated.

Third, all these scores are highly sensitive to stray pixels in the image. The reason here is that the addition of a single point to I may cause d'''_T to differ from d_T by a large amount in a large number of coordinates. This occurs when there are many features S_i so that $(x,y)T(S_i)$ is closer to the new point than any of the points of I . This happens when T fails to match many features of I in M . Since every pixel which is not part of the object is in some sense a stray pixel, we must hope that this does not occur too often for optimal values of T .

However, the requirement that we have a small amount of noise does place meaningful restrictions on the differences between d_T , d'_T and d''_T and we can trace these to their potential impact on our algorithms. The restriction that we have a small amount of distortion ensures that d'_T and d_T differ at most by a small amount in each coordinate.

The requirement that we have a small number of stray pixels in M' ensures that $\frac{n}{N_M}$ is small.

7.1 One-sided Hausdorff distance

The score $s_H(T)$ is stable with respect to distortion. This is because a small distortion results in small differences in the coordinates of d_T and d'_T , and thus a small difference between $\max_i(\{d_{T,i}\})$ and $\max_i(\{d'_{T,i}\})$.

One-sided Hausdorff distance is unstable with respect to stray pixels in the model. This is because a single stray pixel S_{N_M+1} may produce $d_{T,N_M+1} > \max_i(\{d_{T,i}\})$.

7.2 Hausdorff fraction

The score s_{HF} is unstable with respect to distortion. This is because of the discontinuity involved in thresholding. Here is a simple example. Suppose that M is the boundary of a square of side n and I is the boundary of a square of side $n + 2k - 2$. Suppose that T places M squarely in the center of I . Then all of $T \oplus M$ is within k of I . If M' is the boundary of a square of side $n - 2$, then none of $T \oplus M'$ is within k of I . Indeed, for the optimal transformation \hat{T} only half of $\hat{T} \oplus M'$ is within k of I .

Hausdorff fraction is stable with respect to stray pixels in the model. Assume we have matched m pixels, so that $s_{HF}(T) = \frac{m}{N_M}$. After the addition of stray pixels, we have the score $\frac{m+j}{N_M+n}$ with $j \leq n$. Since $\frac{n}{N_M}$ is small, so is the difference between these two scores.

7.3 Hausdorff quantile

The score S_{HQ} is stable with respect to distortion. This follows from the fact that the coordinates of d_T and d'_T are close and that the quantile operation is continuous when restricted to a space of fixed dimension.

Hausdorff quantile is unstable with respect to stray pixels in the model. This is because the quantile operator is discontinuous with respect to additional data points. To see this, consider the situation in which there is a large gap in the values of our data. Suppose also that the quantile we seek is represented by the datum at the bottom of this gap. Adding an additional data point above this gap causes the quantile measure to jump up to the top of this gap.

7.4 Spatially Coherent Matching

The score $s_{SCM}(T)$ is unstable with respect to distortion for the same reasons as Hausdorff fraction.

Spatially coherent matching is stable with respect to stray pixels. To see this, apply the argument for Hausdorff fraction separately to both the matching set $G(T)$ and the boundary set $B(T)$.

7.5 Hausdorff average

The score $s_{HA}(T)$ is stable with respect to distortion. To see this observe that d_T and d'_T are close and the operation of taking the average is continuous.

Hausdorff average is stable with respect to stray pixels in the model. To see this, recall that we have bounded the distance of our stray pixels. This ensures that for $i > N_M$ we have $d_{T,i} < d_{\max}$ for some fixed bound d_{\max} . Before the addition of the stray pixels, our score is

$$\frac{\sum_{i=1}^{N_M} d_{T,i}}{N_M}$$

After the addition of the stray pixels, the score is now

$$\frac{\sum_{i=1}^{N_M} d_{T,i} + \sum_{i=N_M+1}^{N_M+k} d_{T,i}}{N_M + n}$$

The latter summands are bounded, so when $\frac{n}{N_M}$ is sufficiently small, these scores are as close as we like.

These theoretical results can be seen experimentally as the presence or absence of noise in the measure. See figures 9-14 and the explanation in section 9 below.

8 Algorithmic advantages of stability

All of the previous methods search for an optimum value $\hat{T} \in \mathcal{T}$. The most naive implementation of this search would iterate through all values of $T \in \mathcal{T}$. For each such T , it would iterate through each feature S_i of M to compute $(x,y)(T \oplus S_i)$. For each S_i it must then compute the distance from $(x,y)(T \oplus S_i)$ to $\mathcal{M}(S_i)$ for which it would iterate through the points of $\mathcal{M}(S_i)$. Thus, if M consists of N_M edge pixels, \mathcal{T} consist of N_T transformations and I consists of N_I edge pixels, this requires $N_M N_T N_I$ individual distance computations. What is worse, the images are two-dimensional and \mathcal{T} is at least two-dimensions (and could easily be four or five dimensional). This contemplates a computation which is at least $\mathcal{O}(n^6)$. Fortunately, we need not be so naive!

8.1 Efficiencies in the image

As Huttenlocher points out [4], the iteration over the points of I can be done once and recorded in the *Voronoi function*. If I represents an image of width w_I and height h_I , the Voronoi function is the map from $[0, w_I - 1] \times [0, h_I - 1]$ which gives the distance from each pixel in the domain of I to the nearest (actual) pixel of I . If M has f_M different types of features, then there are f_M different match loci in I and we need f_M different Voronoi functions.

8.2 Efficiencies in \mathcal{T}

For s_H , Huttenlocher shows that one can improve the efficiency of the search for \hat{T} by subdividing \mathcal{T} into regions. Given a region R , one picks $T_R \in R$ (perhaps at its

center) and computes $s_H(T_R)$. One is also able to compute a bound, ε_R so that if $T \in R$, $\|s_H(T) - s_H(T_R)\| \leq \varepsilon_R$. Thus, if $s_H(T_R)$ is sufficiently bad (say $s_H(T_R) > s_0 + \varepsilon_R$), one can remove from further consideration not only T_R , but all transformations $T \in R$. If we are seeking \hat{T} and we have already computed $s_H(T_{R'})$, then we can reject all $T \in R$ if $s_H(T) > s_H(T_{R'}) + \varepsilon_R$. Regions which are not eliminated from consideration are subdivided and the process repeats. The process ends by either discovering that all transformations have been eliminated (in which case there is no transformation T with $s_H(T) < s_0$) or by finding \hat{T} with $s_H(\hat{T}) < s_0$.

This process depends on the existence of ε_R . The existence of this bound is guaranteed by stability with respect to distortion. To see this in general, chose the region R to be compact. For each feature, S_i , $(x,y)T \oplus S_i$ is continuous with respect to T . Hence, the vector d_T is continuous with respect to T . Finally, stability with respect to distortion ensures that $s(T)$ is continuous with respect to T . This ensures that transformations T which are close to T_R have scores $s(T)$ which are close to $s(T_R)$. In particular the set of scores $s(T)$ with $T \in R$ is bounded.

In practice, of course, we need not only the existence of ε_R but an effective way to compute it. This is feasible for many reasonable choices of \mathcal{S} and R . In the case that s is s_H , s_{HQ} or s_{HA} , \mathcal{S} consists of translations, R is a rectangle, and R_T is its center, then ε_R is one half the diagonal of R . The most natural way to subdivide \mathcal{S} in this case is by using a quad tree [11].

8.3 Efficiencies in the model

To further speed up the computation, Huttenlocher suggests quick Monte Carlo elimination of regions by computing the distance for randomly chosen points of M . This may fail to eliminate a region which would have been eliminated if the distance were computed for all points of M . However, all of its sub-regions will eventually be eliminated.

This works only in the case of s_H . It works because $s_H(T) = \max(d_{T,i})$. Hence T can be eliminated on the basis of a single coordinate of d_T .

We propose an additional efficiency which applies to any measure which has stability with respect to distortion. It applies most easily to s_H , s_{HQ} and s_{HA} where the error bounds are once again the lengths of half-diagonals.

We require the additional assumption that each feature S_i of M has a position $(x,y)(S_i)$ and that for each T , $(x,y)T \oplus S_i$ is a continuous function of $(x,y)(S_i)$. (These methods can be applied to three dimensional models undergoing out-of-plane rotation, but we will exposit them in the case where M is two dimensional.)

Suppose now, that we impose a quad tree T_M on our model M . At each level ℓ , M is composed into 4^ℓ regions, $R_{\ell 1}, \dots, R_{\ell 4^\ell}$. Let $c_{\ell 1}, \dots, c_{\ell 4^\ell}$ be their centers. Suppose that these regions contain $n_{\ell 1}, \dots, n_{\ell 4^\ell}$ features of M . We may replace our model M with an approximation M_ℓ consisting of the points $c_{\ell 1}, \dots, c_{\ell 4^\ell}$ with multiplicities $n_{\ell 1}, \dots, n_{\ell 4^\ell}$. Of course we will omit any points of multiplicity 0.

Now, for a given transformation T , we may compute the vector

$$d_{\ell T} = (d(T \oplus c_{\ell 1}), d(T \oplus c_{\ell 1}), \dots, d(T \oplus c_{\ell 4^\ell}), d(T \oplus c_{\ell 4^\ell}), \dots),$$

where each coordinate $d(T \oplus c_{\ell_i})$ is repeated n_{ℓ_i} times. Now each of these coordinate is within ε_ℓ of the corresponding coordinate of d_T . (Here ε_ℓ is the half-diagonal these regions.) Notice also that s_H, s_{HQ} and s_{HA} are invariant under permutation of coordinates. It follows that for each of these,

$$\|s(T) - s(d_{\ell T})\| \leq \varepsilon_\ell.$$

This observation combines with that of the previous subsection, so that if R is a region of the subdivision of \mathcal{T} and ℓ is a level of the subdivision of our model, and $T \in R$ then

$$\|s(T) - s(d_{\ell R})\| \leq \varepsilon_R + \varepsilon_\ell.$$

Notice that computing $s(d_{\ell T})$ requires at most 4^ℓ distance lookups in the Voronoi diagram rather than the full N_M of them. In the case where we are computing $s_H(d_{\ell T})$ or $s_{HA}(d_{\ell T})$ we need not even write out multiple copies of the value $d(T \oplus c_{\ell_i})$ since we are computing either the max of these or a weighted average of them.

8.4 Implementation issues

The tree-like nature of the decomposition of \mathcal{T} into regions raises questions as to what is the best way to search this tree. Introducing the levels ℓ raises a question as to which level is best for querying a given region R . We conjecture that the optimum strategy is to keep ε_R and ε_ℓ as similar in magnitude as possible once the regions R become small enough.

9 Experimental results

For our experiments we synthesized data using 11 arbitrary backgrounds with a cropped object from the COIL 20 dataset [9] superimposed. We synthesized 100 target images in total by randomly choosing a foreground object and randomly placing it within the background. We then sampled 10 models for each target including the object known to be in the scene. We used the results of matching these models to estimate Receiver Operating Characteristic (ROC) curves for each measure as in [1].

Each target image was processed using the Canny edge detector with MatLab default parameters for threshold and sigma [2]. The COIL images were also processed using the same edge detection algorithm. We refer to the results of edge detection on the COIL images as the model set.

Detection consisted of selecting the maximum value and transform into a target for a given model of the distance measure for the Hausdorff Fraction and Spatially Coherent Matching algorithms, and of selecting the minimum value of the distance measure for the One-sided Hausdorff, Hausdorff Quantile, Hausdorff Average, and Hausdorff Functional Average distance measures. The transform space was limited in these experiments to translation in x and y by one pixel increments. In the case that there were multiple points in the transformation space that tied for the best match, we detected 8-connected regions and took the centroid of the largest such region. In the case that there were multiple regions with the same size, we chose one at random. Thus



Figure 1: An example target image and the results of Canny edge detection

we arrive at a single maximally matching transform for which to evaluate a match in a target, model pair.

ROC curves were constructed by varying the parameter s_0 and calculating tuples of the form $(p(d), p(f))$ measuring the probability of detection vs. the probability of a false positive. Detection is said to have taken place if the reported location of a known object is within a radius of n pixels from its actual placement in the target. We report results for $n = 5$ in figures 2-8.

The Hausdorff fraction score requires the parameter for the minimal acceptable distance, k to be set. It seems that a value of $k = 5$ is reasonable given the scale of the objects we are dealing with. We would not be happy if a match was determined based on a match locus that is more than 5 pixels away from our model feature, but a 5 pixel tolerance gives flexibility for noise from imaging, edge detection, and transformations not accounted by our transformation space, such as minor rotation.

For the Hausdorff quantile distance, we used the 90th quantile. This is the case where we can have approximately 10% occlusion. Hausdorff Quantile is particularly useful in the case where a significant component of noise comes from occlusion. A value of 10% noise is realistic for this experimental setup, as large portions of the objects being detected are not occluded.

Spatially Coherent Matching requires a threshold for which to measure where the neighbor relation crosses the boundary of a circle with that radius. This penalizes the number of pixels that match within the radius by the number of edges to neighbors outside the radius. We wish to select a value that allows for displacement of pixels being matched up to a point, but penalizes for further distortion. We choose a radius of 5 as this accounts for the noise in the edge detection process, but should eliminate to a certain degree false positives from erroneous matching with pixels with a large displacement.

For Hausdorff functional average matching, we chose a function of x^2 as this is analogous to sum of squared distance correlation. We see in the results, however, that this performed worse than Hausdorff Average matching. Hausdorff average was by far the most accurate measure in terms of performance for the task of detecting models in a set of target images as one can see in figure 6.

An interesting visualization of the measures is the surface formed by plotting the

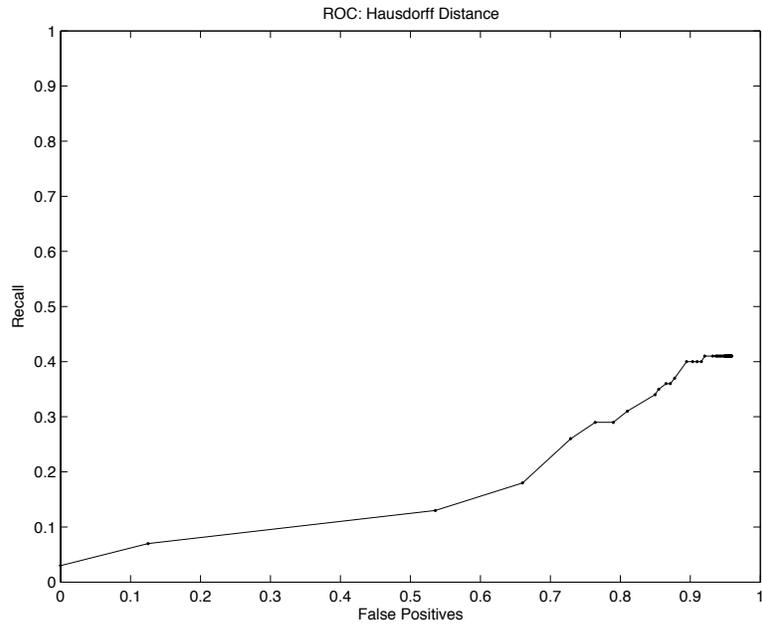


Figure 2: ROC curve for one-sided Hausdorff distance.

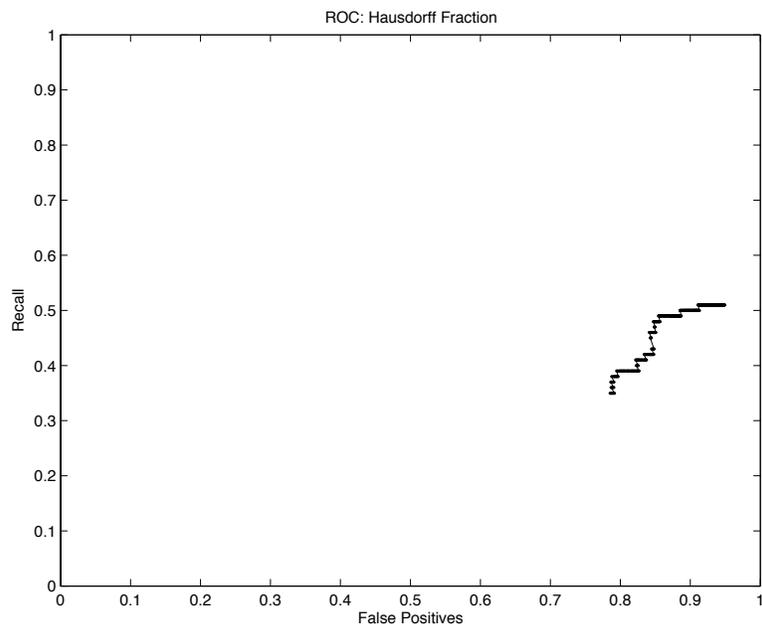


Figure 3: ROC curve for Hausdorff fraction.

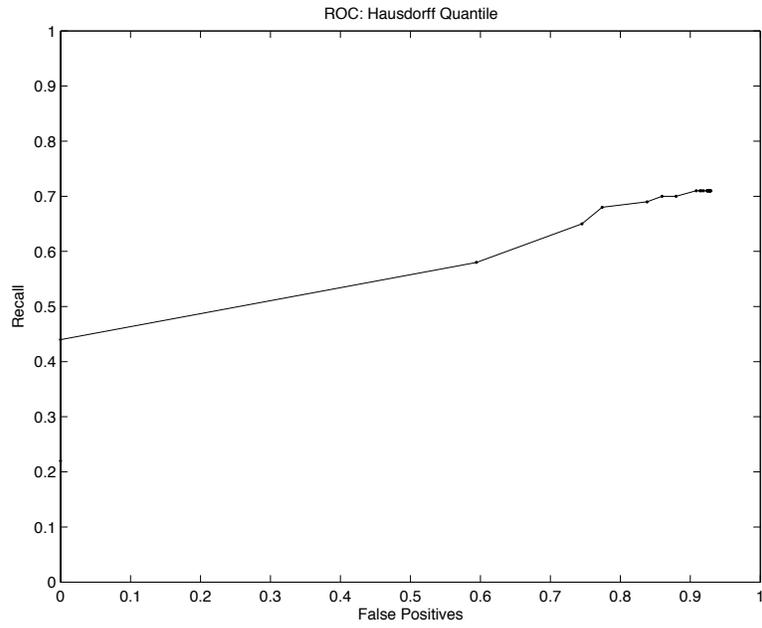


Figure 4: ROC curve for Hausdorff quantile.

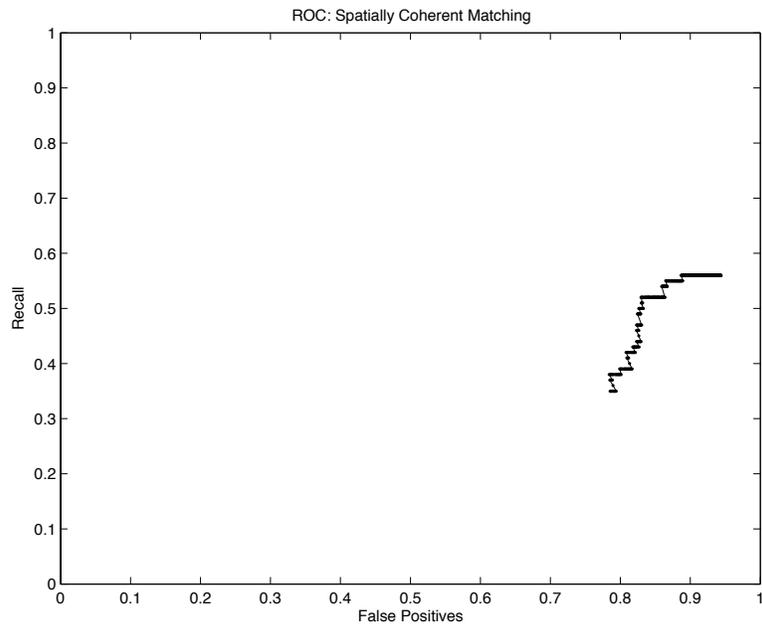


Figure 5: ROC curve for Spatially Coherent Matching.

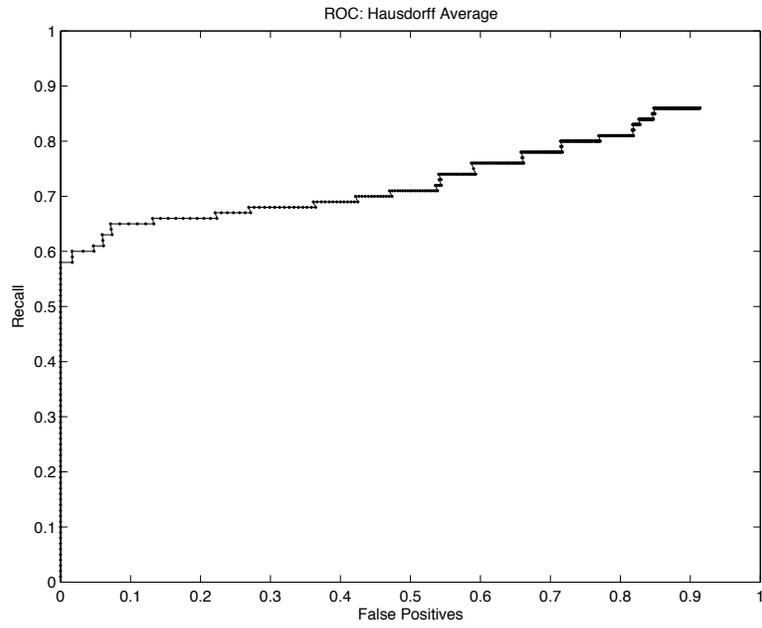


Figure 6: ROC curve for Hausdorff average.

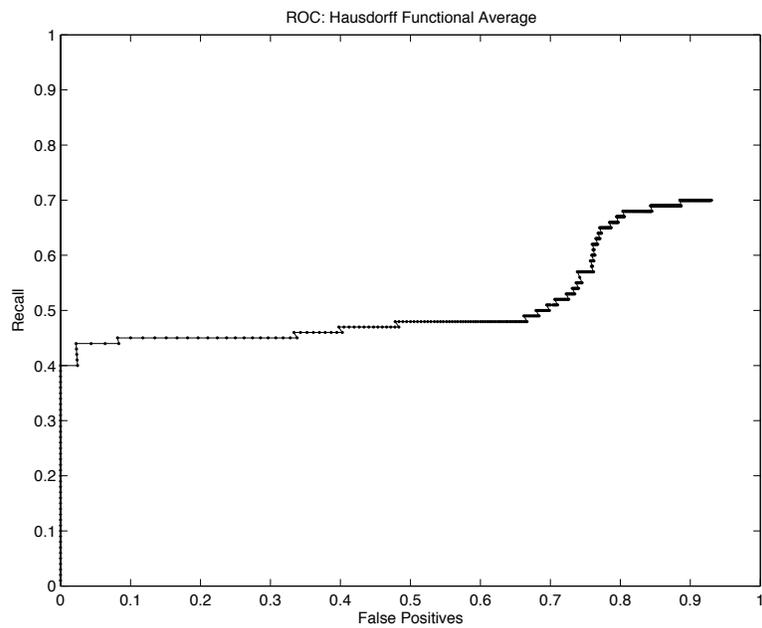


Figure 7: ROC curve for Hausdorff functional average.

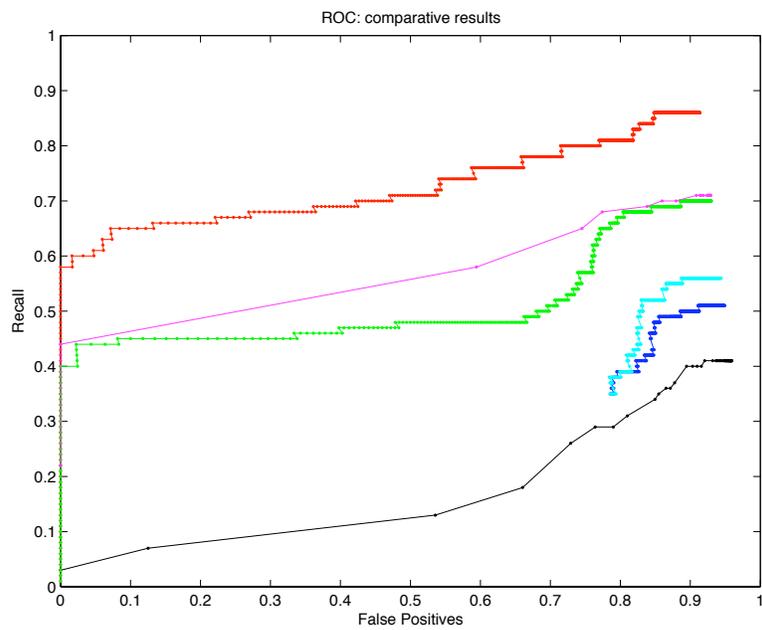


Figure 8: ROC curves. From top to bottom: Hausdorff average is shown in red, Hausdorff quantile in magenta, Hausdorff functional average in green, Spatially Coherent Matching in cyan, Hausdorff fraction in blue, and One-sided Hausdorff in black.

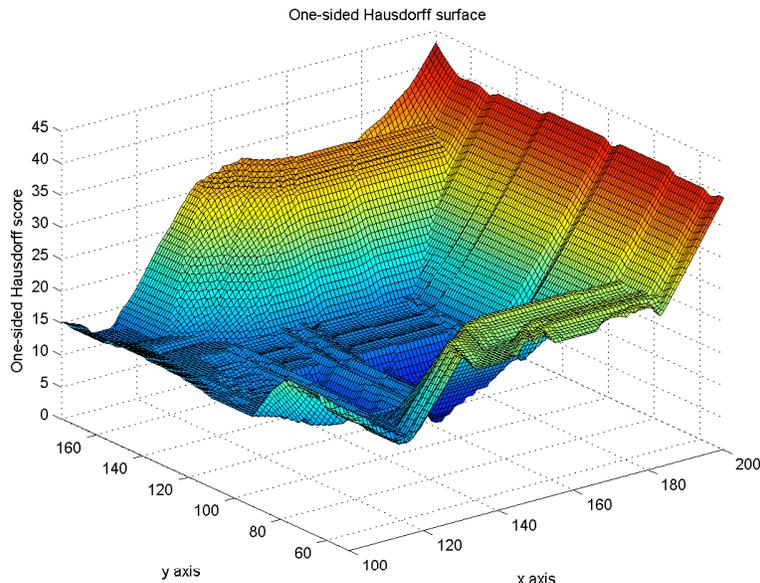


Figure 9: This surface shows $s_H(T)$ as a function of the transformation space.

score as a function of the transformation space. Figures 9-14 show this surface for each of the six measures for the matching task in the data of figure 1. Each measure correctly shows an optimal point in the vicinity of $(152, 100)$ in the transformation space, which indeed corresponds to the actual location of the object in the image. As we noted in section 7, $s_H(T)$, $s_{HQ}(T)$, $s_{HA}(T)$, and $s_{HA,f}(T)$ all show local stability, as is shown in figures 9, 11, 13, and 14, respectively. The plots of $s_{HF}(T)$ and $s_{SCM}(T)$ show noise as predicted in sections 7.2 and 7.4. Hausdorff quantile (figure 11) appears to show similar noise, but the score only differs by at most one between adjacent points in the transformation space. This is an artifact of the scale at which the graph is displayed.

10 Conclusion

We have shown that Hausdorff Average distance measure outperforms other variants for model detection in an edge matching experimental setup. As an additional advantage over other techniques, there are no other parameters than the match threshold, unlike Hausdorff fraction, Hausdorff quantile, and Spatially Coherent Matching. Hausdorff average has continuity with respect to noise in the form of both distortion and stray pixels, a property not held by the other measures. We have also shown that this method produces a maximum a posteriori estimate and described a scheme for improved computational efficiency.

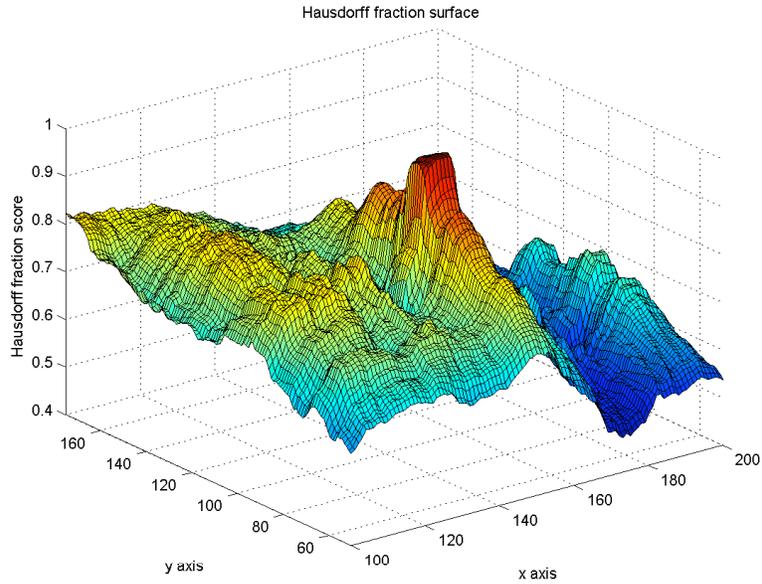


Figure 10: This surface shows $s_{HF}(T)$ as a function of the transformation space.

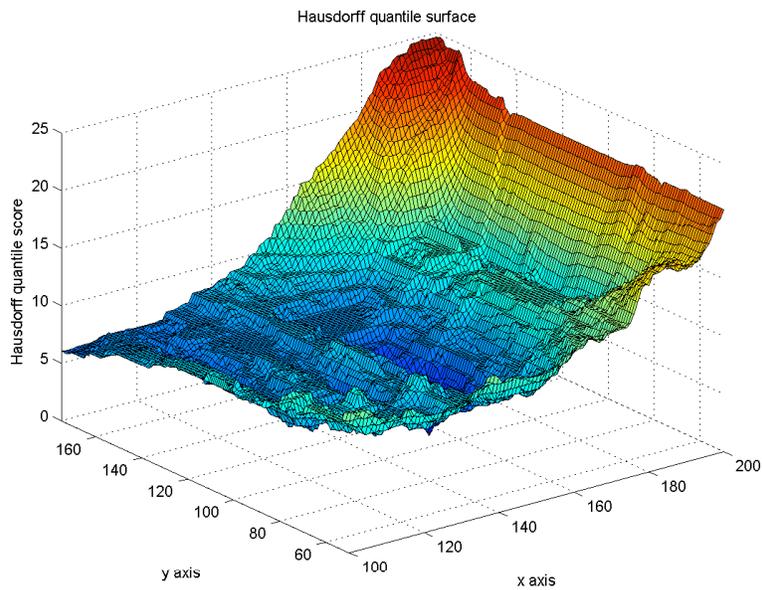


Figure 11: This surface shows $s_{HQ}(T)$ as a function of the transformation space.

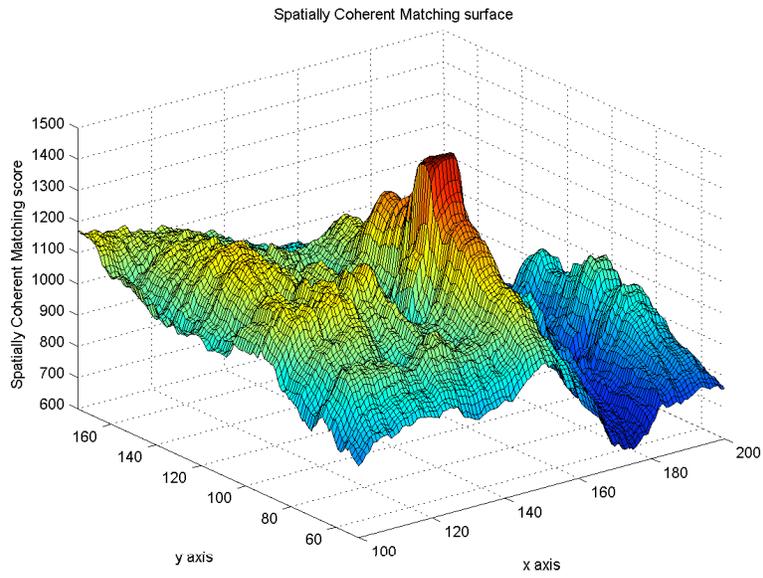


Figure 12: This surface shows $s_{SCM}(T)$ as a function of the transformation space.

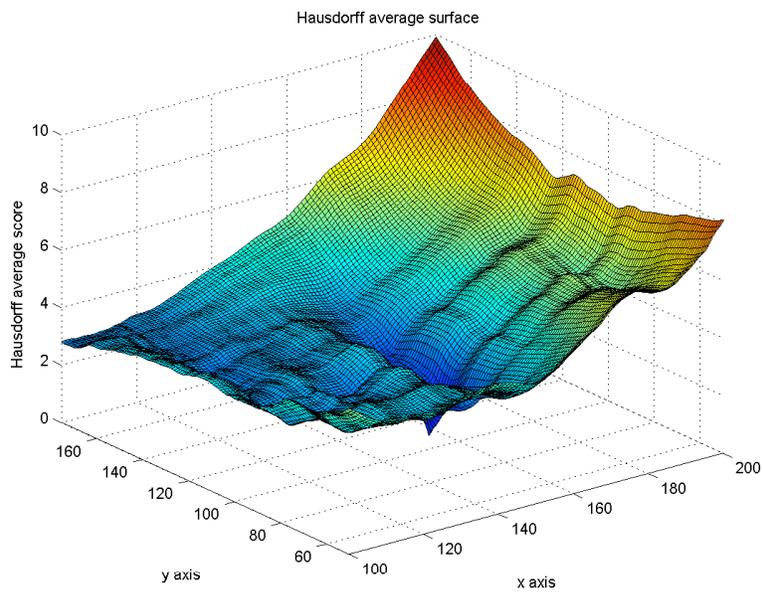


Figure 13: This surface shows $s_{HA}(T)$ as a function of the transformation space.

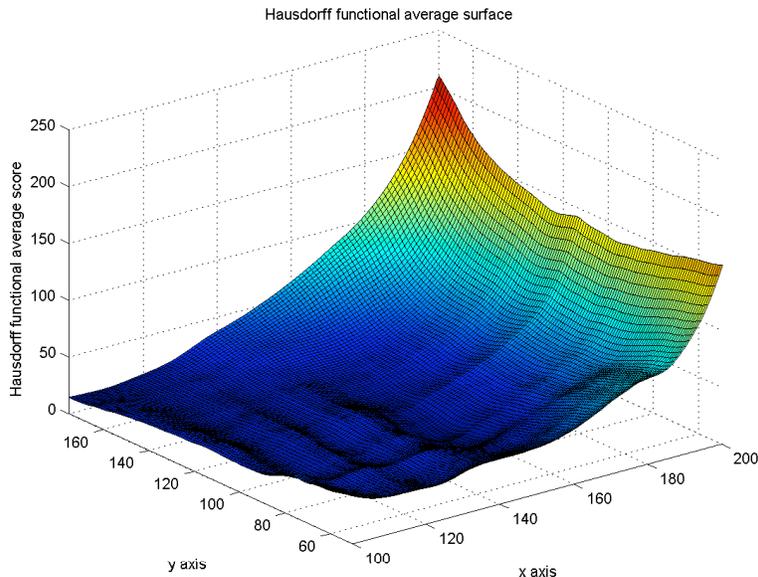


Figure 14: This surface shows $s_{HA,f}(T)$ as a function of the transformation space.

10.1 Future directions

We have only applied relatively simple features here. Any complete exploration of image matching using a Hausdorff-based distance measure will likely include several different feature detectors. Interest point detectors such as those described in [7] may be promising.

There are situations where the nature of a feature in addition to its location is subject to transformation. For example, the frequency or orientation of a texture often changes with out of plane rotation thus making it more sensitive to a different filter. The stability results of this paper do not necessarily hold in such a case.

In producing the results of this paper, we did not implement the results of section 8 because they do not apply to all the measures studied here. The usability of $s_{HA}(T)$ depends on the increases in efficiency suggested by section 8. This deserves empirical study.

The family of measures, $s_{HA,f}(T)$, invites the question: are there choices for f that are appropriate to domain specific applications.

Finally, Ed Riseman has proposed that review of the existing literature on statistical robustness may suggest new Hausdorff-based measures (personal communication). For example, see [6].

Acknowledgements

The first author wishes to thank Mary Kerins without whom this paper would not have been possible. The second author would like to thank Ed Riseman for his helpful suggestions. The second author is generously funded by the National Science Foundation (NSF ITR-0325937).

References

- [1] Y. Boykov, D. Huttenlocher, A New Bayesian Approach to Object Recognition, *Proceedings of IEEE CVPR*, 1999, 517-523.
- [2] J. Canny, A Computational Approach to Edge Detection, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1986. Vol. PAMI-8, No. 6, pp. 679-698.
- [3] M. P. Dubuisson, and A. K. Jain, A modified Hausdorff distance for object matching. *ICPR94*, pp. A:566-568, Jerusalem, Israel, 1994
- [4] D. Huttenlocher, G. Klanderman, and W. Rucklidge, Comparing Images Using the Hausdorff Distance, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 15, no. 9, pp. 850-863, 1993
- [5] O. Jesorsky, K. Kirchberg, R. Frischholz, Robust Face Detection Using the Hausdorff Distance, *Audio and Video-Based Biometric Person Authentication*, 3rd International Conference, AVBPA 2001, Halmstad, Sweden, June 2001.
- [6] I.S. Kharin, Y. Kharin, *Robustness in Statistical Pattern Recognition*, Kluwer, 1996.
- [7] D.G. Lowe, Distinctive image features from scale-invariant keypoints *International Journal of Computer Vision*, 60, 2 (2004), pp. 91-110
- [8] Y Lu, C L Tan, W Huang and L Fan, An approach to word image matching based on weighted Hausdorff distance, *6th International Conference on Document Analysis and Recognition ICDAR2001*, 10-13 Sept 2001
- [9] H. Murase, S. Nayar, Visual Learning and Recognition of 3-D Objects from Appearance. *Int. J. Computer Vision*. 14(1): 5-24, Jan. 1995
- [10] J. Paumard, E. Aubourg (1997). Adjusting astronomical images using a censored Hausdorff distance. *Proc. of 4th IEEE Int. Conf. on Image Processing (ICIP '97*, Santa Barbara, CA), v. III, pp. 232-xxx.
- [11] H. Samet, *The Design and Analysis of Spatial Data Structures*, Addison-Wesley, Reading, MA, 1990
- [12] M. Shapiro, M. Blaschko, Stability of Hausdorff-based Distance Measures, *Proceedings of IASTED Visualization, Imaging, and Image Processing*, Marbella, Spain, Sept. 6-8, 2004.

- [13] S. Srisuk, W. Kurutach, New Robust Hausdorff Distance-based Face Detection, *ICIP01*(I: 1022-1025).