SANDIA REPORT

SAND2004-4447
Unlimited Release
Printed October 2004

A Novel Window Based Method for Approximating the Hausdorff in 3D Range Imagery

Mark W. Koch

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia is a multiprogram laboratory operated by Sandia Corporation,
a Lockheed Martin Company, for the United States Department of Energy's
National Nuclear Security Administration under Contract DE-AC04-94AL85000.

Sandia National Laboratories

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

NOTICE: This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government, nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, make any warranty, express or implied, or assume any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represent that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, any agency thereof, or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof, or any of their contractors.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865)576-8401
Facsimile: (865)576-5728
E-Mail: reports@adonis.osti.gov
Online ordering: http://www.osti.gov/bridge

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd
Springfield, VA 22161

Telephone: (800)553-6847
Facsimile: (703)605-6900
E-Mail: orders@ntis.fedworld.gov
Online order: http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online

SAND2004-4447
Unlimited Release
Printed October 2004

A Novel Window Based Method for Approximating the Hausdorff in 3D Range Imagery

Mark W. Koch
Sensor Exploitation Applications
Sandia National Laboratories
P.O. Box 5800
Albuquerque, NM 87185-1163

Abstract

Matching a set of 3D points to another set of 3D points is an important part of any 3D object recognition system. The Hausdorff distance is known for it robustness in the face of obscuration, clutter, and noise. We show how to approximate the 3D Hausdorff fraction with linear time complexity and quadratic space complexity. We empirically demonstrate that the approximation is very good when compared to actual Hausdorff distances.

Intentionally Left Blank

Contents

Figures

1 Introduction

In 3D image understanding, we would like to recognize objects such as faces or vehicles in 3D range imagery. One approach uses a template from a database of objects and matches it to the probe image containing the unknown. Mean square error can determine the goodness of the match [2], but it fails for an obscured object or if the probe image has excess clutter. The Hausdorff distance can measure the goodness of a match in the presence of occlusion, clutter, and noise [5][6]. In this paper we discuss a new and novel method on how to efficiently, in time and memory, compute the Hausdorff distance for 3D range imagery.

1.1 Hausdorff Distance

The Hausdorff distance was originally designed to match binary edge images [5]. Modifications to the Hausdorff distance permit it to handle not only noisy edge positions, but also missing edges from occlusion, and spurious edges from clutter noise [5][6]. In edge image matching, we can conceptualize an edge image as a list of 2D points, where the points indicate the location of an edge. Thus, we can generalize this concept to 3D datasets where a list of 3D points is available.

For instance, let $A = \{a_1, \ldots, a_p\}$ represent the set of p points for the template. For edge images, a_i is a 2x1 column vector representing the (x, y) coordinate of the i^{th} edge point in the template, and for 3D images a_i represents a 3x1 column vector corresponding to 3D points located on the object. Similarly, let $B = \{b_1, \ldots, b_q\}$ represent q points for the probe. The following equation gives the directed Hausdorff distance $h(A, B)$:

$$h(A,B) = \max_{a \in A} \min_{b \in B} \|a - b\| \tag{1}$$

where $\|\cdot\|$ represents the distance between points a and b measured by some norm, for example the Euclidean norm L_2. Figure 1a shows a simple 2D example. The circles represent points from the template *A*, and the squares represent points from the probe *B*. In Figure 1b we show the directed Hausdorff $h(A,B)$.The first step finds for each point in *A* the closest point in *B*. The dotted ellipses shown in Figure 1b denote this pairing. The second step looks at the distance for each pairing and determines the largest distance. This largest distance is the directed Hausdorff $h(A,B)$.

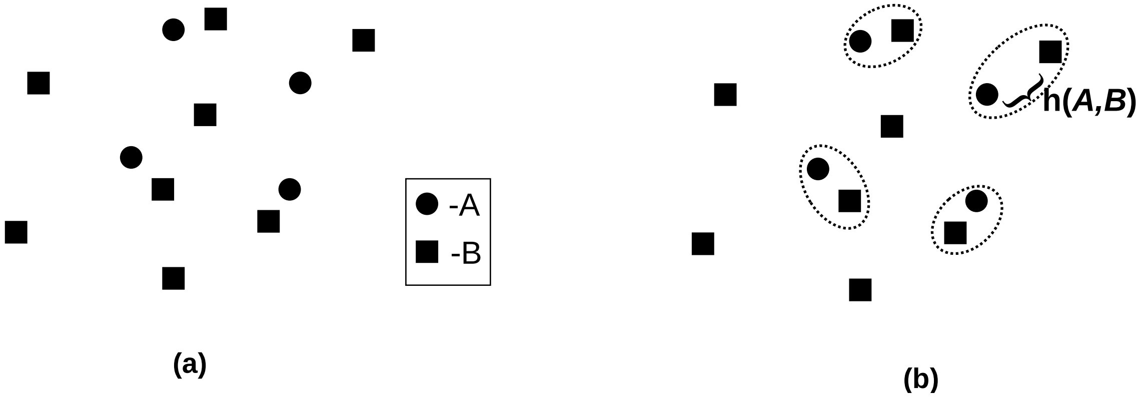

Figure 1. Hausdorff Computation Example. (a) A 2D example. Circles represent points from the template *A* and squares represent points from the probe *B*. (b) Computing the directed Hausdorff distance *h(A,B)*. See text for details.

1.2 Hausdorff Modifications

The following equation gives a modified Hausdorff distance for handling partial obscurations and clutter [5]:

$$h(A,B) = \underset{a\in A}{K^{th}} \min_{b\in B} \|a-b\|. \tag{2}$$

This equation uses the K^{th} rank pairing instead of the largest one. This allows a certain number of points from the template *A* to be obscured and not have a matching point from the probe *B*. An alternative formulation to (2) that does not require a partial sort computes the directed Hausdorff fraction $\phi(A,B)$ [5][6]:

$$\phi(A,B) = \frac{\left|\min_{b \in B} \|a-b\| < \tau\right|}{|A|}. \qquad \textbf{(3)}$$

Here, τ represents the largest acceptable Hausdorff distance and $|\cdot|$ represents cardinality or the counting operation. Thus, for each point in A we find the closest point in B and count the number of distances less than the threshold τ. We then divide that count by the number of points in the set *A*. This gives the fraction of points from *A* within a certain distance of points from *B*.

The directed Hausdorff distance $h(A,B)$ is not symmetrical, thus $h(A,B)$ does not always equal $h(B,A)$. One can view $h(A,B)$ as a hypothesis generation step and by computing $h(B,A)$ over a limited extent, $h(B,A)$ becomes a verification step. We usually take the limited extent to be a bounding box around the points in *A*. This reduces mismatches caused by other objects or clutter in the image containing the probe *B*. The same arguments apply to the directed Hausdorff fraction $\phi(A,B)$. Using a limited extent, the undirected Hausdorff distance $H(A,B)$ is defined as $\max(h(A,B),h(B,A))$. Thus, we require small distances for both the hypothesis and verification for point set *A* to match point set *B*. In a similar way, we can define the undirected Hausdorff fraction $\Phi(A,B)$ as $\min(\phi(A,B),\phi(B,A))$.

2 3D Sensors and Data

To ground the discussion of 3D representations and Hausdorff calculation we use the Minolta Vivid 910 ranger scanner. The Minolta Vivid 910 is commercially available and produces a 640x480 range image and a corresponding 640x480 color intensity image. The Minolta uses a projected laser stripe to acquire triangulation-based range data and has an optimal range depth of field of at most 1.2 m with a range resolution of 0.1 mm.

We verify our approach using a 3D database collected with the Minolta Vivid 900 range scanner by the University of Notre Dame. The data comes from Collection D of their biometric database [3][4] and contains images of 275 subjects acquired over a thirteen week period. Of the 275 subjects, 200 participated in more than one session allowing the first image to be used as template and subsequent acquisitions to be used as a probe image (test image against the template). The faces have an average of 80,000 3D points.

3 3D Data Representation and Space Complexity

Representation of 3D data is usually determined by the types of algorithms used to process the data and memory requirements for the data structures. We will discuss three data representations and their impact on computing the Hausdorff distance. The data representations are: 1) point cloud, 2) range image, and 3) voxel.

The point cloud representation creates a list of the 3D points and their (x, y, z) coordinates. Assuming 4 bytes to represent the floating point coordinate and 640x480 points, the point cloud representation would require at most 3.5 MB. We can reduce the number of points by segmenting out the object of interest or not including invalid points (points too far away), and thus reducing the amount of memory needed in the point cloud representation. The point cloud representation allows us to represent the 3D coordinates at the accuracy of the sensor using quadratic space complexity $O(n^2)$. Note, the “big-Oh” function $O(\cdot)$ or *order* function represents the asymptotic space or time complexity of the data representation or algorithm. For the point cloud representation, we implicitly assume that the height and width of image array grows at the same rate. For example, if we double the resolution of the sensor then 640x480 goes to 1280x960.

The range image representation stores the depth of each point in a 2D image array. Each (x, y) coordinate in 3D space is mapped to an integer (i, j) index. The mapping requires some quantization of the (x, y) coordinates, and will typically depend on the parameters of the matching function and the resulting range image size. By quantizing the Minolta data to 640x480 we do not lose much accuracy, since the Minolta scans to a 640x480 array. The Minolta scan pattern is not rectangular, so we lose some accuracy when quantizing. If we use 4 bytes to represent a range value, then a range image requires 1.17 MB. Thus, for a slight loss in accuracy, we gain a reduced memory representation, but still at $O(n^2)$ space complexity. For either the point cloud or range image representation we can reduce the memory requirements using a fixed point representation, if we can constrain the range of the data or normalize to the distance of the object from the 3D sensor.

The voxel representation stores a 3D image using a 3D array. Here, each (x, y, z) coordinate in 3D space maps to an integer (i, j, k) index. Here, the voxel representation has a cubic space complexity $O(n^3)$. Each array element contains a 1 or a 0 indicating whether there exists an (x, y, z) coordinate that maps to the corresponding (i, j, k) index or not, respectively. Here, one could represent each voxel with one bit. To represent the full resolution of the Minolta sensor (1.2 M range with 0.1 mm accuracy and 640x480 spatial size) would require 12000 ranges bins or 3500 MB to store the 3D array. To reduce the memory requirements, we can decrease the range accuracy to 1mm and require 350 MB.

4 Hausdorff Computation and Time Complexity

The speed of computing the Hausdorff distance depends on the representation used to store the 3D data. Using the point cloud representation, the Hausdorff fraction (2) can be computed with quadratic time complexity $O(n^2)$. Here, we implicitly assume p and q (the number of points in A and B) grow at the same rate. The quadratic time complexity results from having to find the closest point in B for every for point in A.

We can obtain a faster Hausdorff distance computation of $O(n)$ time by using the voxel representation. The reduced time complexity requires using a distance transform (DT) that can be computed in linear $O(n)$ time [7]. A DT of a voxel image is a new voxel image where every voxel gives the distance to the closest voxel containing a 3D point or a "1." Figure 2 shows an example of a 2D image and also the corresponding DT. In the DT image the numbers correspond to *square* L_2 norm values. To compute the Hausdorff fraction, we can first compute a DT of the probe image *B*. Using the coordinates of *A* as an index into the DT of *B* we can find the distance to the closest voxel in *B*. We then count the number of the distances less than the threshold τ^2 and divide by the total number of points in *A* to get the Hausdorff fraction $\phi(A,B)$. This approach gives the overall Hausdorff calculation a complexity of $O(n)$.

0	**0**	**0**	**1**	**1**	**1**	**0**	**0**	**0**
0	**1**	**1**	**0**	**0**	**0**	**1**	**1**	**0**
0	**1**	**0**	**0**	**0**	**0**	**0**	**1**	**0**
1	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**
1	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**
1	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**
0	**1**	**0**	**0**	**0**	**0**	**0**	**1**	**0**
0	**1**	**1**	**0**	**0**	**0**	**1**	**1**	**0**
0	**0**	**0**	**1**	**1**	**1**	**0**	**0**	**0**

(a)

2	**1**	**1**	**0**	**0**	**0**	**1**	**1**	**2**
1	**0**	**0**	**1**	**1**	**1**	**0**	**0**	**1**
1	**0**	**1**	**2**	**4**	**2**	**1**	**0**	**1**
0	**1**	**2**	**5**	**8**	**5**	**2**	**1**	**0**
0	**1**	**4**	**8**	**13**	**8**	**4**	**1**	**0**
0	**1**	**2**	**5**	**8**	**5**	**2**	**1**	**0**
1	**0**	**1**	**2**	**4**	**2**	**1**	**0**	**1**
1	**0**	**0**	**1**	**1**	**1**	**0**	**0**	**1**
2	**1**	**1**	**0**	**0**	**0**	**1**	**1**	**2**

(b)

Figure 2. 2D “voxel” image and its distance transform. (A) Example of a 2D “voxel” image. (B) The corresponding distance transform using the square of the L_2 norm.

Unfortunately, a voxel distance transform image has a large memory requirement. To store the distances as a 4 byte float would require 11250 MB or 11 GB for a 640x480 image at 1mm resolution and a 1.2 M depth of field. Here, the cubic space complexity $O(n^3)$ causes a large increase in memory requirements as we increase the number of bits per voxel.

To obtain reasonable speeds at low memory requirements we use the range image representation to approximate the Hausdorff distance. Here, we can get linear time complexity along with reduced memory requirements. To accomplish this we use the observation that for smooth objects, close points in the (x, y) plane also tend to be close in (x, y, z) space. Thus for a range image, to find the closest point in *B* to $A(i, j)$, we need to compute distances within a window of size *w* centered around $B(i,j)$. To accomplish this we use what we call a *distance kernel*. The distance kernel for a window of size *w*=5 is shown if Figure 3. Each entry of the distance kernel is the square L_2 norm to that entry’s position from the center of the distance kernel. To find the closest point to (i, j) in the template image *A*, we take the range value stored at (i, j) of the template $z_t(i, j)$ and center the distance kernel on top of the (i, j) point in the probe image *B*. Let $k(u,v)$ represent the value in the distance kernel for row u and column v and similarly we have the corresponding values in the probe image $z_p(u,v)$. To compute $\min_{b \in B} \|a-b\|$ in the Hausdorff fraction we compute:

$$\min_{u=1}^{w} \min_{v=1}^{w} [(z_t(i, j) - z_p(u,v))^2 + k(u,v)]. \tag{4}$$

This operation has constant time complexity $O(1)$, since $s = w^2$ is constant and independent of the number of points *B*. The quantity $\min_{b \in B} \|a-b\|$ has linear time complexity $O(n)$ and grows at the rate of the number of points in *A*. Thus, the time complexity for the Hausdorff fraction using the range image representation and the distance kernel is linear $O(n)$.

8	**5**	**4**	**5**	**8**
5	**2**	**1**	**2**	**5**
4	**1**	**0**	**1**	**4**
5	**2**	**1**	**2**	**5**
8	**5**	**4**	**5**	**8**

Figure 3. Example of a distance kernel for a window size of *w*=5.

The window based Hausdorff approximates the Hausdorff fraction. We can empirically estimate the approximation error by matching 3D faces using the window based Hausdorff for range imagery and computing the Hausdorff exactly from the point cloud representation. We demonstrate the performance of this approach using the University of Notre Dame's 3D face imagery from a 100 people with 2 images per face. We pick one face as the template and the other as the probe image. For the Hausdorff fraction (3), we use $\tau = \sqrt{3}$ and a window size of

5x5 as shown in Figure 3. Using Parzen density estimation, Figure 4 shows the distribution of differences between the window-based approximation of the Hausdorff and the actual Hausdorff fraction. The errors are very small with the largest being 0.021. The errors are always positive indicating that the window based Hausdorff always over estimates the true Hausdorff fraction. In theory, we could subtract the mean of the distribution ~0.01 from the window based Hausdorff to give a maximum error of ± 0.01.

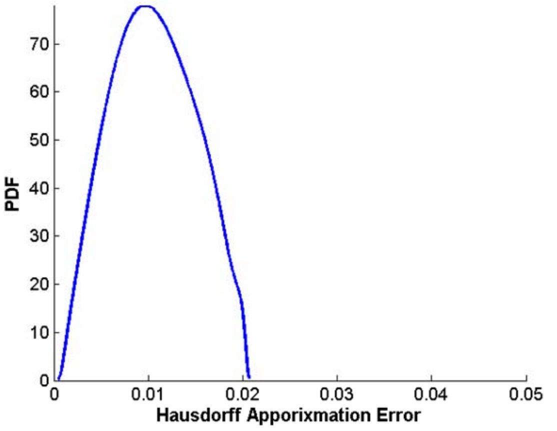

Figure 4. The Hausdorff approximation error using the window based Hausdorff for range imagery.

5 Conclusion

The 3D Hausdorff fraction can measure the goodness of match in the presence of occlusion, clutter, and noise. This paper describes how to efficiently approximate the 3D Hausdorff fraction in time and space. The window based Hausdorff has a time complexity of linear $O(n)$ and a space complexity of quadratic $O(n^2)$. Using a distance transform, we can get an exact Hausdorff distance in $O(n)$ time complexity, but the space complexity increases to cubic $O(n^3)$. A $O(n^3)$ space complexity is impractical for all, but the smallest of 3D images. Using a point cloud representation we can achieve an $O(n^2)$ space complexity, but the time complexity is also

$O(n^2)$. Even though the window based Hausdorff is an approximation, we empirically show that the approximation is very good.

6 References

[1] Achermann, B., and H. Bunke, "Classifying Range Images of Human Faces with Hausdorff Distance", *International Conference on Pattern Recognition*, 2: 809-813, September 2000.

[2] Besl, P. J., and N. D. McKay, "A Method for Registration of 3-D Shapes," IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 14(2): 239-256, February 1992.

[3] Chang, K., K. W. Bowyer, and P. J. Flynn, "Face Recognition using 2D and 3D Facial Data", *ACM Workshop on Multimodal User Authentication*, 25-32, December 2003.

[4] Flynn P. J., K. W. Bowyer, and P. J. Phillips, "Assessment of Time Dependency in Face Recognition: An Initial Study," *Audio and Video-Based Biometric Person Authentication*, 44-51, 2003.

[5] Huttenlocher, D. P., G. A. Kanderman, and W. J. Rucklidge, "Comparing Images using the Hausdorff Distance," IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 15(9): 850-863, September 1993.

[6] Huttenlocher, D. P., R. H. Lilien, and C. F. Olson, "View-Based Recognition using an Eigenspace Approximation to the Hausdorff Measure," IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 21(9): 951-955, September 1999.

[7] Maurer, C. R., "A Linear Time Algorithm for Computing Exact Euclidean Distance Transforms of Binary Images in Arbitrary Dimension," IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 25(2): 265-270, February 2003.

Distribution:

1	MS 0759	Tommy D. Woodall, 04145
1	MS 0780	Stephen Oritz, 04138
1	MS 0780	Trina D. Russ, 04138
4	MS 1163	Mark W. Koch, 15433
1	MS 9018	Central Technical Files, 8945-1
2	MS 0899	Technical Library, 9616