Line Feature-Based Recognition Using Hausdorff Distance*

Xilin Yi[1] and Octavia I. Camps[1,2]
[1]Department of Electrical Engineering
[2]Department of Computer Science and Engineering
The Pennsylvania State University
University Park, PA, 16802

Abstract

A point-based (edge pixels) correlational method using the Hausdorff distance to determine if there is any model pattern in a given image was proposed by Huttenlocher et al [3]. While this approach works well and it is computationally efficient in the presence of model translation in the image, it is significantly time consuming when the model has been rotated and scaled. In this paper, we propose a line-feature based approach for model based recognition using the Hausdorff distance. This new approach reduces the problem of finding the rotation and scaling to the problem of finding two translations, therefore exploiting the efficiency of the algorithm proposed in [3].

The use of line features separates the rotation, scaling and translation so that each of them can be handled individually. The line features in the original domain are first transformed into a new 2-D domain consisting of the orientation and the logarithm of the length of the line. In this way, rotation and scaling in the original domain correspond to a translation in the new domain and the Hausdorff point-based matching is used to find it. Next, the model is rotated and scaled using the result from first matching and second Hausdorff distance matching is performed to determine the model translation.

The method performance and sensitivity to segmentation problems are characterized using and experimental protocol with simulated data. It was found that the algorithm performs well, degrading nicely as the segmentation problems increase. The algorithm was tested with real images as well.

KEY WORDS: Hausdorff distance, object recognition, pattern recognition, line-features.

1 Introduction

A correlational method using the Hausdorff distance to determine if there is any model pattern in a given image was proposed by Huttenlocher et al [3]. The Hausdorff distance approach is robust in the presence of uncertainty and it can be made robust to outliers. So far, only point features (edge pixels) have been used for recognition with this approach. The technique has successfully recognized translated, scaled and rotated models in images. Efficient implementations of the algorithm have been developed, some of them exploiting specialized computer graphics hardware [3]. However, the computational time is dramatically increased when there is scaling and rotation with respect to the translation only case [4]. In this paper, we propose a line-feature based recognition approach that overcomes this problem.

The proposed approach uses line segments as features. Each segment has associated the coordinates of its mid-point (x, y), the logarithm of its length $\log l$, and its orientation θ. Thus, both model and image segments can be represented as points in the four dimensional space $(x, y, \log l, \theta)$. Then, the matching algorithm consists on applying twice the *point-based* Hausdorff matching algorithm to determine two translation vectors. The first matching is run on the $(\log l, \theta)$ plane between the projections on this plane of the four dimensional points representing the image and model segments. This matching will give us the rotation angle and the scaling factor. The second matching provides the translation factor and also serves as a verification stage for the previous step. It uses the results from the first matching to rotate and scale the model, and does the image and updated model point-based (edges) matching.

An extensive system evaluation is performed using simulated data. Random model and image line features are generated with a few probability distributions and perturbed to simulate different types of segmentation problems. The probabilities of success for the first matching and the second matching are then obtained. The relations between the maximum average fraction of matching segments and the line length and orientation uncertainties are also found. The approach is also tested using several real images. The processing time is recorded and compared with previous approaches.

2 The Hausdorff Distance

We leave the detail of the Hausdorff distance theory behind, since it has been addressed very thoroughly by Huttenlocher's group[1, 2, 3, 4]. However, for the sake of completeness, we give a brief introduction.

Given two point sets A and B, the Hausdorff distance between A and B is defined as

$$H(A, B) = \max(h(A, B), h(B, A)) ,$$

where

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

*This work was supported in part by NSF grant IRI9309100.

and $\|.\|$ denotes some norm defined on the plane.

To handle outliers, partial distances $h^K(B,A)$ and $h^L(B,A)$ are used instead, where

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

$K^{th}_{b\in B}$ denotes the K^{th} ranked distance, and $h^L(B,A)$ is defined in a similar way. The partial bidirectional Hausdorff distance is now defined as

$$H^{LK}(A,t(B)) = max(h^L(A,t(B)), h^K(t(B),A))$$

where $t(.)$ denotes a transformation of a set. The partial distance therefore measures the difference between a portion of the model and the image.

If a transformation t is applied to the model point set, we are interested in the Hausdorff distance as a function of the transformation of the set B, or

$$d(t) = H^{LK}(A,t(B)).$$

Therefore, to recognize a model-like image A, we can apply different transformations to the model B. Then, if $d(t)$ is less than a certain threshold we are convinced that a match between the model and the image exists.

The process of finding $d(t) = H^{LK}(A,t(B))$ is called *forward* matching. If we set a certain Hausdorff distance as threshold to this process, we may be able to find a fraction of model points satisfying this threshold for a certain transformation t. This fraction $f(t)$ is called the forward matching fraction. On the other hand, if we transform A instead of B, this will be called the *reverse* matching process. Accordingly, there is a reverse Hausdorff distance threshold and a reverse matching fraction $g(t)$.

Huttenlocher et al [3] presented an efficient algorithm to search for the transformation t that minimizes the Hausdorff distance. For pure translation, a 360×240 size image and a 115×199 size model, they reported a computation time of approximately 20 seconds on a Sun-4 (SPARCstation 2). However, with rotation and a $640X480$ image the algorithm took 44 minutes and 47 seconds [4]. We assume that the computational time will be even longer when scaling is involved. Due to this reason, an alternative method using line features is proposed for matching in the presence of translation, as well as rotation and scaling.

3 Proposed New Approach

It can be observed that the orientation angles of the image line segments get increased or decreased by a constant when the model is rotated, but remain unaffected under translation and scaling. On the other hand, the lengths of the segments get multiplied by a constant factor when the model is scaled but they are invariant under rotation and scaling. In this section it will be shown that these properties can be used to find the rotation and scaling transformations efficiently.

Model and image line segments can be represented by four dimensional points with coordinates equal to their mid-point coordinates, the logarithm of their length and their orientation. Let s_m be a model segment with coordinates $(x_m, y_m, \log l_m, \theta_m)$. Consider an ideal image s_i of s_m obtained by rotating s_m by an angle θ_{Rot} and scaling it by a factor S and the projections of the corresponding four dimensional points to s_m and s_i onto the plane $(\log l, \theta)$. The logarithm of the length and the orientation of the image segment s_i are given by

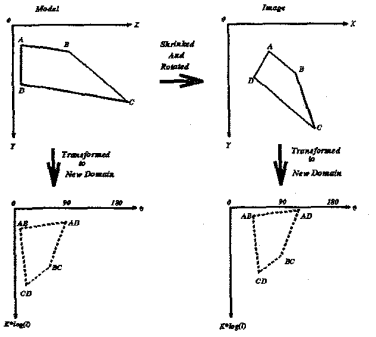

Figure 1: Illustration of transforming the image and model to the $(\log l, \theta)$ domain.

$$\begin{aligned} \log l_i &= \log l_m + \log S \\ \theta_i &= \theta_m + \theta_{Rot} \end{aligned}$$

Thus, it is seen that rotating and scaling a model segment is equivalent to translating the projection of the corresponding four dimensional point on the plane $(\log l, \theta)$ by an amount $t = (\log S, \theta_{Rot})$. This translation can be found very efficiently using the algorithm given in [3]. In the sequel, the plane $(\log l, \theta)$ will also be referred as the *new domain* or the *transformed domain*.

To simply illustrate the above idea, consider the following example. Suppose, we have two polygons, where one is a rotated and scaled version of the other as shown on the upper half of Figure 1. Matching these polygons is equivalent to find the rotation and scaling transformation that transforms one into the other. The representation of the given polygons in the $(\log l, \theta)$ plane is also shown in the lower half of Figure 1. One can see that two new polygons are formed in this plane, with their vertices corresponding to the sides of the original polygons representations in the new domain. The new polygons have identical shape and dimensions, and differ only in a translation corresponding to the rotation and scaling factors.

If we do not make any further modifications, the image and model in the new domain will have a range in the θ axis from 0° to 180°. However, applying the translational Hausdorff distance to the transformed segments will not always give the correct rotation angle θ_{Rot}. For example, if the model is rotated 110°, the model segment AD that has an orientation of 90° will have a corresponding image segment with 20° and the new polygons in the $(\log l, \theta)$ plane will not have the same shape. Fortunately, this problem can be solved by extending the image orientation range from 0° to 360°. The image pattern from 180° to 360° is an exact copy of the pattern from 0° to 180°. There is no need to modify the model pattern in the new domain. Figure 2 shows the modified approach. As we can see,

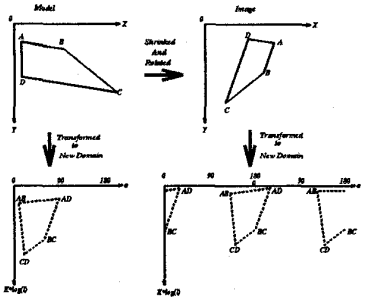

Figure 2: Illustration of the modified version of transforming the image and model to the new domain

in this modified version, the image will still be able to keep the same pattern in the new domain with spurious points in the lower and upper range of θ.

The above approach works well for rotation angles between 0° and 180°. However, if the rotation angle is between 180° and 360°, say 180° + ϕ, then the Hausdorff distance matching will give ϕ as the rotation angle, since the length and orientation of the lines will match even if the image is rotated by 180°. This is not a problem: when we perform the second matching we can simply rotate the model by ϕ and 180° + ϕ separately, and do the matching twice to see if there is any match.

Once the rotation and scaling parameters are obtained from the first matching, the model in the original domain is transformed using these parameters. Then, the updated model is matched with the image using the usual point-based Hausdorff distance to search for the translation. In this stage, false alarms from the previous stage can be eliminated.

There are some important issues that need to be addressed:

1. In the transformed domain, $\log l$ and θ are measured using different units. While applying Hausdorff matching, distances are computed. Thus, compatible measurements in both axes are needed. Adding a scaling factor to one of the axes is one way of solving this problem. In our approach, we use a K factor for the $\log l$ axis. This factor K is also related to the resolution of the $(\log l, \theta)$ plane in a discrete implementation. If a line has uncertainty ρ for its length, we would like to have this line still fall into the same "bin" as the ideal line without uncertainty would. Thus, K must satisfy:

$$|K*log(l+l*\rho) - K*log(l)| = |K*log(1+\rho)| < 1$$

2. If a line is broken into two or more parts, the Hausdorff distance using the $\log l$ and θ parameters will be incorrect. This is because the original line splits into lines with smaller lengths that have no relation to the original length. The sensitivity of the algorithm to this problem is characterized during our experiments.

3. In a complex image with a model-like sub-image, some lines from other objects can fall in the same area in the new domain that the lines from the model-like object fall. Also, lines from some other objects may form the same pattern that the model-like object forms in the new domain. As a result, the matching fraction in the new domain may be low and we may get some false alarms. However, these false alarms will be eliminated in the second stage for most of the cases as it is shown by our experiments.

4 Experimental Protocol

The experimental protocol consists of: (1) Performance characterization using simulated data and (2) testing with real data.

4.1 Simulated Data

Model Generation: A set of lines (30) with random orientations, lengths and locations are generated for the model, such that the coordinates of their midpoints are uniformly distributed within a 512 × 512 image, their orientations are uniformly distributed between 0° and 180°, and their lengths are normally distributed with a mean of 70 pixels and a variance of 45 pixels.

Image Generation: Two types of images are generated: images containing a model and clutter and images containing clutter only. The images with a model are generated by rotating, scaling, and translating the model segments to a known location. There is no restriction of picking the rotation angle and the scaling factor. In our experiments we used a rotation angle of 30° and a scaling factor of 0.8. Different types of noise perturbations are then added to the resulting set in order to generate the image line segments. The clutter only images are generated by generating random line segments uniformly distributed as described below.

Noise Perturbations: The following noise perturbations were used to simulate segmentation problems:

(1) **Clutter:** A fixed percentage (100%) of random line segments are added to the image as outliers. Lengths of these lines are uniformly distributed between 0 and 128. Orientations are also distributed uniformly between 0° and 180°. Locations of the midpoints are uniformly distributed in the image except for the area where the model is expected to lie. A fixed smaller percentage (2%) of lines are uniformly added to that area later. (2) **Length uncertainty:** A Rayleigh distribution with parameter σ_l is used to model the uncertainty in the length of the lines. (3) **Broken lines:** For modeling broken lines, the following method is used: The number of points where a line is broken is chosen using a Poisson distribution with parameter p_b. The points where the line is to be broken are then obtained using a uniform distribution along its length. The length of the gap to be created is obtained using a Rayleigh distribution. (4) **Orientation uncertainty:** The orientation of the line is perturbed using a Gaussian distribution with zero mean and standard deviation of σ_t/l where l is the length of the line. This is done since longer lines are expected to have less angular perturbation than shorter lines.

Performance Characterization:

1. **Determination of factor K:** The probability of correctly detecting the rotation angle, scaling factor, and translation are obtained for $\sigma_t = \sigma_l = 0.5$ and varying K. The factor K should be selected such that these probabilities are maximum.

2. **First Matching:** The Hausdorff distance is computed between the model and the image for all translations in the new domain corresponding to rotation and scaling in the original domain. Let $f(\theta_1, L_1)$, $f(\theta_2, L_2)$ ... $f(\theta_n, L_n)$ be the matching fractions from the forward or reverse Hausdorff matching results when noise is added to the image, where $\theta_1, \theta_2 ... \theta_n$ are the searched rotation angles, and L_1, L_2 ... L_n are the logarithms of the scale factors S times K. The pair (θ_i, L_i), such that $f(\theta_i, L_i)$ is maximum is the one selected.

   The following observations are made:

   (a) The selected forward matching fraction is averaged for 20 simulations and plotted against the rotation perturbation σ_θ and the length perturbation σ_l.

   (b) The probability of the selected forward matching fraction being greater than a threshold was found and this was plotted against the noise parameters σ_t.

   (c) A plot was made for the probability of successful detection of the rotation angle against σ_t. A rotational angle is said to be successfully detected if the difference between the estimated angle θ and the ground truth θ_{true} is less than a threshold.

   (d) A plot was obtained for the probability of successful detection of the scaling against σ_l. A scaling factor is said to be successfully detected if the difference between the estimated scaling and the ground truth is less than a threshold.

   (e) The probability of a false match is also studied by matching models against clutter only images. The ratio between the number of lines in the model and the clutter image is takes as 1, 2, and 3 for this experiment. A false match is said to happen if the matching fraction is found to be greater than a given threshold. The results of the experiments are average for 20 simulations and plotted as a function of the threshold.

3. **Second matching:** The model and image segments are rasterized and the image edge pixels locations are perturbed with using a Gaussian distribution with zero mean and variance 0.5. The rotation and scaling obtained from the first Hausdorff distance matching is then applied to the simulated model. Next, the second Hausdorff distance matching is run between the rotated and scaled model and the noisy image. The best forward matching fraction $f(x, y)$ and the corresponding translation (t_x, t_y) is selected from each matching result.

   The following observations are made:

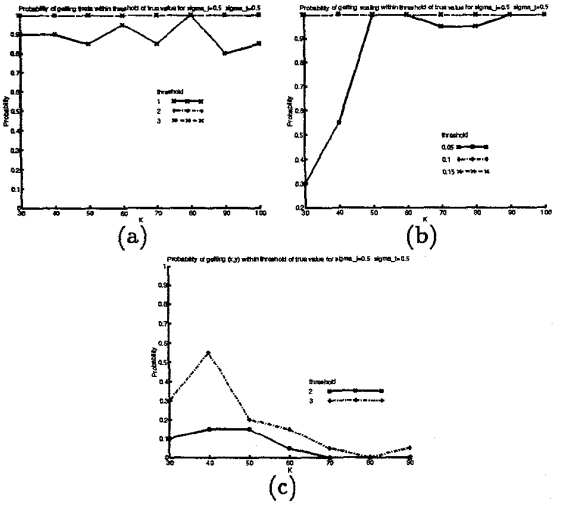

Figure 3: The probabilities for correctly detecting (a) rotation angle, (b) scaling factor, and (c) translation

   (a) The probability of the matching fraction being greater than a threshold is found and it is plotted against the noise parameter σ_t.

   (b) A plot is made for the probability of successful detection of the true translation against σ_l. The detection is successful when $|\max f(x_i, y_i) - f(x_{true}, y_{true})| < threshold$, where x_{true} and y_{true} are the ground truth translation coordinates.

4.2 Real Data

Real objects were used to test the effectiveness of the method. The 512 × 480 images were taken in our vision laboratory. Rotation and translation were modeled by twisting and translating the camera with a robot arm and scaling was modeled by zooming the camera. Matchings were done using first and second Hausdorff distance and the resulting parameters of rotation, scaling were compared with the ground truth values.

5 Results

5.1 Simulated Data

1. **Determination of K.** The plots for the probabilities of correctly detecting the rotation, scaling and translation factors are shown in Figure 3. As we can see there is no single K that can maximize all three probabilities. Furthermore, it is certain that if we use different σ_t and σ_l values, the results will be different. In our experiment, we chose K as 60 as a good compromise.

2. **First Matching.** All the plotted results from the simulation for the first matching are shown in Figure 4. It is observed that (1) the matching fraction averaged for the 20 simulations plotted against the rotation perturbation and the length

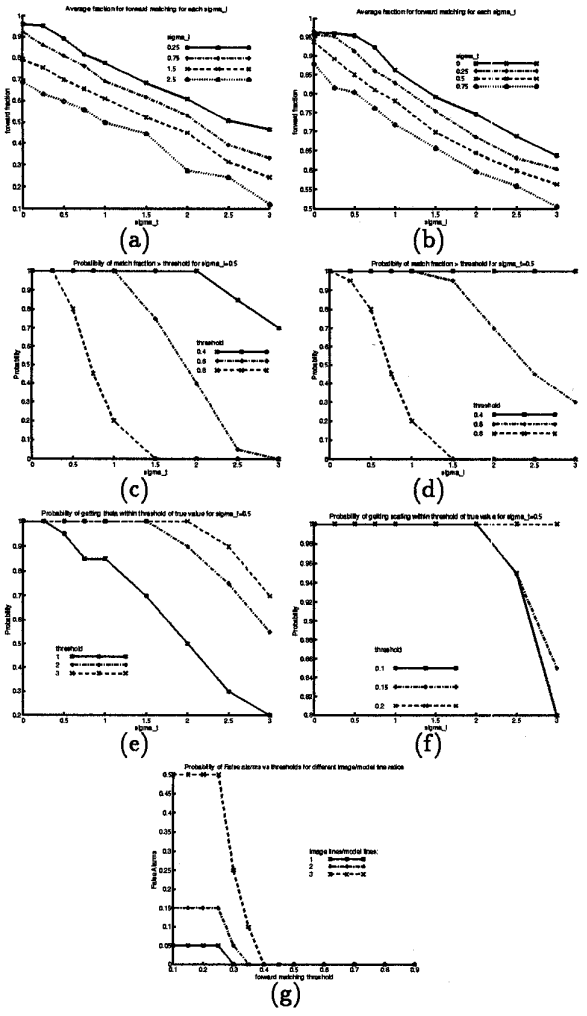

Figure 4: The simulation results from the first matching(a to g)

perturbation σ_l in Figure 4 (a) and (b) go down as the noise deviations σ_l and σ_t increase. (2) the probability of the matching fraction being greater than a threshold plotted against the noise parameters σ_t and σ_l are shown in Figure 4 (c) and (d). It is seen that the probability decreases as the noise deviation σ_t increase. However, the probability decreases faster with respect to σ_t. This is expected because if a line is deviated from the ideal angle, we may expect most of the line points deviate away from the ideal positions. However, if we have some uncertainty for the line length, we may still have a large number of points staying at the ideal positions. (3) Figure 4 (e) and (f) show the relation between the probability of θ being within threshold of the true rotation angle against σ_t and the relation between the probability of getting the scaling within the threshold of the true scaling

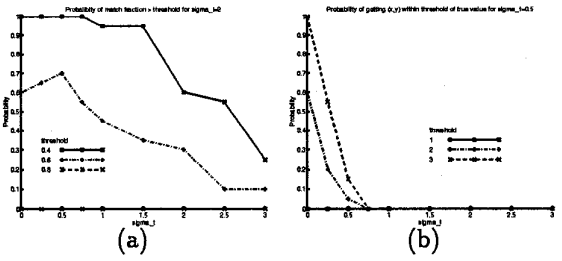

Figure 5: The matching results from the second matching.

value and σ_l, respectively. As we can see both probabilities decrease with the increase of σ_t and σ_l. (4) Figure 4 (g) shows the plots for the false matches when the ratio between the number of segments in the image and the model were 1, 2, and 3 (for images without a model). It is seen that the probability of a false match decreases as the ratio of number of lines and the forward matching threshold increase.

3. **Second Matching.** The optimal scaling and rotation was applied to the second matching. The probability of the matching fraction being greater than a threshold versus σ_t is plotted in Figure 3 (a). The probability of the translation being within threshold of the true translation against σ_l is also plotted in Figure 3 (b). It is seen that the matching fraction decreases dramatically with respect to σ_t. Also the probability for correctly detecting the translation is very sensitive to σ_l, and decreases when it increases.

5.2 Real Images

A few images and models were tested. Figure 5.2 shows two of our tested examples. When we performed the tests, we simply selected the results for rotation and scaling from the first matching, then rotated and scaled the model for the second matching. The models and the images were segmented using the Canny edge operator and linked using the ORT package. The rotation angle and scaling factor of Figure 5.2 (a) are 31° and 0.75 respectively with respect to the model in Figure 5.2 (c). The rotation angle and scaling factor of Figure 5.2 (e) are 12° and 0.76 respectively with respect to the model in Figure 5.2 (f) . The final matching results are shown in Figure 5.2 (i) and (j). The CPU time on a SPARCstation 5 in both cases is less than 7 seconds for the first matching and less than 50 seconds for the second matching.

6 Conclusions

In this paper, a method to use line features for Hausdorff distance matching was presented. The proposed method separates rotation, scaling and translation into three parts and associates them with the line's orientation, length and individual points, respectively. The original image was transformed into a new domain where the rotation and scaling in the original domain is

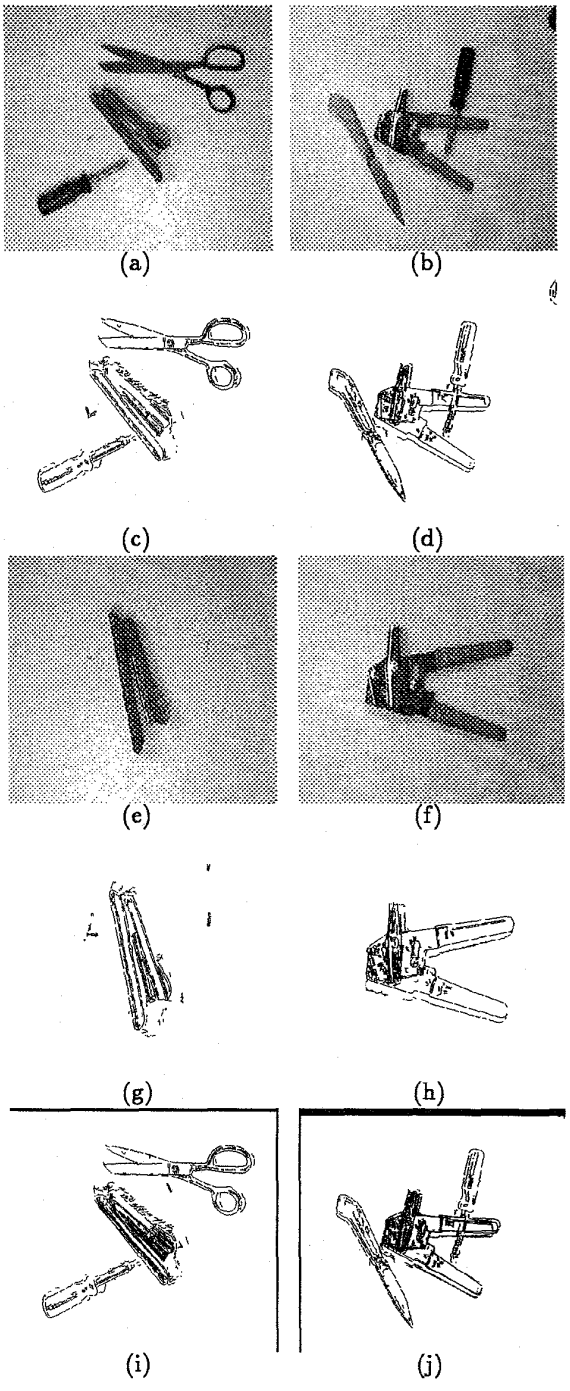

Figure 6: The original image and the rotated/scaled stapler model, their edge detected images, and the matching result.

mapped into a translation. The Hausdorff transform was then used to estimate the translation and hence the parameters of rotation and scaling. Since there exist efficient algorithms for computing translations, the computational time has been reduced significantly from previous approaches that try to compute the rotation and translation at the same time.

The performance of the method and its sensitivity to segmentation problems were characterized by using a rigorous experimental protocol modeling the most common types of errors in segment extraction. The algorithm performs well, degrading smoothly as the perturbations increase. The method was also tested on real images with and without partial occlusion.

7 Acknowledgment

The authors would like to thank Dr. Huttenlocher for providing the software for point based Hausdorff distance matching and Tarak Gandhi for his help on implementing some of the source code and designing the experimental protocol.

References

[1] D. P. Huttenlocher, G. Klanderman, and W. Rucklidge. *Comparing Images Using the Hausdorff Distance under translation.* Technical Report 1211, Cornell University, Department of Computer Science, 1991.

[2] D. P. Huttenlocher, G. Klanderman, and W. Rucklidge. Comparing images using the hausdorff distance under translation. *Proc. IEEE Computer Vision and Pattern Recognition*, pages 654–656, 1992.

[3] D. P. Huttenlocher, G. Klanderman, and W. Rucklidge. Comparing images using the hausdorff distance. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 15:850–863, September 1993.

[4] W. Rucklidge. *Efficient Computation of the Minimum Hausdorff Distance for Visual Recognition.* Technical Report, Cornell University,Department of Computer Science, 1995.