Evaluation of Shape Description Metrics applied to Human Silhouette Tracking

Olivier St-Martin Cormier, Frank P. Ferrie
Dept. Electrical and Computer Engineering &
McGill Centre for Intelligent Machines
Montreal, CANADA
Email: {olivier,ferrie}@cim.mcgill.ca

***Abstract*—Many computer vision applications compare the shapes of objects, but few papers provide meaningful comparisons between different shape distance metrics. This paper will begin by summarily describing six metrics that are widely used in the literature. Then a set of criteria to evaluate metrics will be described and a methodology to test the performance of these metrics will be presented. Finally, experimental results, based on the task of tracking articulated human posture from silhouettes, will be used to determine which metric is best suited to the purpose of human tracking. We find that most of the metrics evaluated herein are valid and perform properly to some extent, but that two of them present more desirable behaviors and robustness to noise.**

***Keywords*-Shape description metrics, human posture tracking, silhouettes, contours**

I. Introduction

A large number of computer vision applications require a similarity measure between object shapes. For example, object detection and tracking algorithms rely on silhouette as the main feature. Silhouettes, or visual contours, are known to carry a lot of information about the curvature and surface geometry of the generating object [1]. Relying on shape can also reduce the influence of varying texture and thus can make systems more robust to variations of intensities. Silhouettes are also fairly simple to extract from images via various methods including optical flow and statistical background modeling.

While this paper will concentrate on the application of shape matching to the task of human posture tracking over image sequences, the evaluation of the image similarity metrics should be useful for any application that requires comparisons of object outlines.

Most human tracking and human pose extraction papers describe the metric they use to measure distance between observed data and expected data, but few provide the rationale behind the choice of one metric over others and even fewer provide comparisons to other metrics. This paper aims to provide a comparison of widely used metrics and provide a better understanding of which types of metrics are more appropriate for the task of comparing human silhouettes. The intention is thus to implement these metrics and compare them, from both theoretical and practical standpoints.

A survey of some shape comparison metrics is provided in [2]. While there is some overlap between the metrics presented here and those in [2], the current paper goes beyond simply describing the selected metrics, but also systematically evaluates them.

When comparing metrics, many methods employ a specific dataset to measure the metric's performance. A good example of this practice in an other domain can be found in [3]. The application of shape matching to human tracking involves a variety of other components that can skew results; we will thus aim to evaluate metrics without relying on a specific dataset.

II. Evaluated Metrics

This section will present an overview of the studied metrics, providing a general description of each. A more in depth look at each metric can be found in the papers referenced.

Many of the methods presented by the original authors are neither translation nor scale invariant; we will attempt to alleviate these problems by cropping the silhouettes and scaling the resulting images to a uniform size. For methods that rely on a chain-coded representation of the silhouette edge, we will resample the chain code to a fixed number of points. These constraints should remove some notion of scale and translation from the metrics, thus simplifying comparison between silhouettes.

A. Hu Moments[4], [5]

Hu moments are a set of image descriptors that are simple to compute and represent many aspects of the evaluated shape such as area, centroid, and orientation. The authors of [4] mention that image moments can cause a lot of ambiguity, but demonstrate that they can be used to successfully match silhouettes with classes of known pose. They estimate pose by using an expectation-maximization algorithm to cluster the Hu moments for a set of pose classes. The main advantage of this metric is that negative space within the main shape blob is properly identified as the metric is computed on the pixel values instead of on the border pixels. Figure 1(b) provides an example of two large negative space regions inside a silhouette. Once the Hu moments are extracted, the distance between two silhouettes is computed as the Euclidean distance between the moment vectors.

B. Pixel Count[6], [7]

Another simple metric involves the computation of the number of pixels that differ between two silhouettes. This method requires good camera calibration or image matching, as both silhouettes need to be aligned. Cropping the bounding box of the silhouette and scaling to a common size, as stated previously, also helps in aligning silhouettes. This

metric can also handle negative space within silhouettes, as discussed in the description of the Hu moments. For two silhouettes S_1 and S_2 represented by binary images of dimensions m by n, we can compute the distance with Equation 1, where $\oplus$ is the exclusive or (xor) operation, and $S[x, y]$ is the value at pixel location (x, y).

$$D(S_1, S_2) = \sum_{x=0}^{n} \sum_{y=0}^{m} S_1[x, y] \oplus S_2[x, y] \tag{1}$$

C. Chamfer Distance[8]

This metric is computed solely from the contour of the silhouette. To do so, a chain code representation of the silhouette edge is extracted by traveling along the edge of the silhouette and recording the visited pixels. The distance between two chain code sets S_1 and S_2 can be computed via the modified Haussdorf distance presented in Equation 2, where $d(p, q)$ is the Euclidean distance between two points in the chain codes. A real Hausdorff distance would take the maximum instead of the sum.

$$D(S_1, S_2) = \sum_{p \in S_1} \min_{q \in S_2} d(p, q) \tag{2}$$

D. Turning Angle[8]

This metric requires the computation of a turning angle diagram, which is generated by recording the angle between successive points in the chain code. This diagram is said to better encode the shape of the silhouette than simply using the chain code point locations. The distance between two silhouettes can be computed as the sum of squared differences between the turning angle representations of each silhouettes.

E. Distance Signal[9]

The chain code representation of the contour is used for this metric, but an extra piece of information is added. Instead of simply looking at the position of the contour points, this metric considers the distance between each point and the center of mass of the silhouette. The center of mass $(\bar{x}, \bar{y})$ of a silhouette represented by a binary image S_i can be computed either with Equation 3 or via the central Hu moment.

$$\bar{x} = \frac{\sum_{x=0}^{n} \sum_{y=0}^{m} xS[x, y]}{\sum_{x=0}^{n} \sum_{y=0}^{m} S[x, y]}, \bar{y} = \frac{\sum_{x=0}^{n} \sum_{y=0}^{m} yS[x, y]}{\sum_{x=0}^{n} \sum_{y=0}^{m} S[x, y]} \tag{3}$$

The distance between two shapes S_1 and S_2 with precomputed distance signals DS_1 and DS_2 is computed as the sum of absolute differences, as shown in Equation 4, where n is the number of points in the chain code.

$$D(S_1, S_2) = D(DS_1, DS_2) = \sum_{i=0}^{n} \left| DS_1[i] - DS_2[i] \right| \tag{4}$$

F. Shape Contexts[10], [11]

Shape contexts are 2D histograms that can be computed at any point along the edge of a silhouette. These histograms encode the relation between the point at which it is computed and all other points of the chain code by recording the angle and distance to the other points. The shape context implicitly encodes the local curvature and shape of the silhouette. This metric is translation invariant by design and can be made scale invariant[12] by scaling the silhouettes and resampling the chain codes to a fixed number of points. The "distance" between two shape contexts p and q is computed by the use of the following χ^2 test:

$$d(p, q) = \frac{1}{2} \sum_{k=1}^{K} \frac{[h_p(k) - h_q(k)]^2}{h_p(k) + h_q(k)} \tag{5}$$

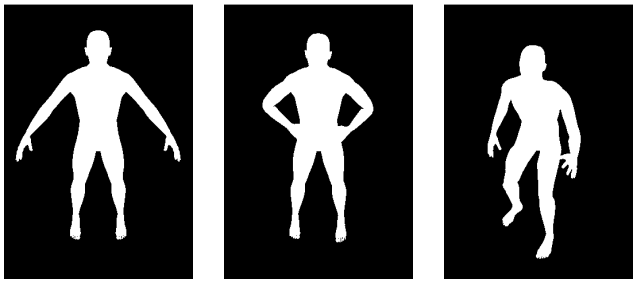
where $h_i(k)$ is the value of the k^{th} bin of the histogram at point i.

To compute a distance between two silhouettes, the authors of [10] suggest finding a one to one mapping between the points on each silhouette that minimizes the sum of distances between the two sets. This type of matching is a full bipartite graph combinatorial optimization problem that can be solved via the KuhnMunkres algorithm [13].

A computationally simpler method of computing the distance between two chain code sets is to use Equation 2. This alleviates the need to run the graph optimization step at the cost of not obtaining a one to one mapping. We will refer to this simplified version as a greedy set matching strategy.

III. Methodology

To make meaningful measurements and test the metrics previously described, we devised a framework that allows us to compute all metrics in a fully controlled manner. To do so, we use an articulated 3D model of a human which can be rendered in any pose. This framework relies on an OpenGL rendering system with simple shaders to directly render silhouettes. Figure 1 shows a few silhouettes generated from this system.



(a) (b) (c)

Figure 1. Sample rendered silhouettes

We opted to use a human model with a skeletal structure comprised of $n = 37$ degrees of freedom (DOF). Table I describes the skeletal structure.

Table I
PROPOSED SKELETAL STRUCTURE

Index	Parent	Name	DOF
00	–	Root / Pelvis	6
01	00	Abdomen	1
02	01	Thorax	3
03	02	Head	3
04	00	Right Hip	0
05	04	Right Thigh	3
06	05	Right Shin	1
07	06	Right Foot	1
08	00	Left Hip	0
09	08	Left Thigh	3
10	09	Left Shin	1
11	10	Left Foot	1
12	02	Right Shoulder	0
13	12	Right Upper Arm	3
14	13	Right Forearm	2
15	14	Right Hand	2
16	02	Left Shoulder	0
17	16	Left Upper Arm	3
18	17	Left Forearm	2
19	18	Left Hand	2
Total DOF			37

To represent the posture of the model, we define a posture vector of the form:

$$\mathbf{P} = [\theta_0, \theta_1, \cdots, \theta_{n-1}, \theta_n], \theta_i \in [0, 1] \quad (6)$$

Each DOF of the skeleton is configured as an axis of rotation and the rotation limits. Defining the minimum and maximum angle limits is what allows us to write the pose vectors with all elements in the range $[0, 1]$. The exact definition of the skeletal structure used for testing is presented in Table I. Given a variation δ_i along each DOF, we can define a pose difference by:

$$\mathbf{\Delta_{12}} = \mathbf{P_1} - \mathbf{P_2} = [\delta_0, \delta_1, \cdots, \delta_{n-1}, \delta_n], \delta_i \in [-1, 1] \quad (7)$$

We represent the projection of the pose vector $\mathbf{P}$ into a 2D silhouette via the rendering system by $S = \mathcal{R}(\mathbf{P})$. This formulation allows us to define two distances. The first is a distance in pose space, as shown in Equation 8, and the second is the metric similarity measure $D(S_1, S_2)$ between rendered silhouettes, as discussed earlier.

$$D(\mathbf{P_1}, \mathbf{P_2}) = D(\mathbf{P_1}, \mathbf{P_1} + \mathbf{\Delta_{21}}) = \|\mathbf{\Delta_{21}}\| \quad (8)$$

IV. METRIC EVALUATION

Before comparing each of the metrics, one must first determine what the characteristics of a good metric are. The author of [2] presents the following three desirable properties for shape matching metrics:

- Metric Property: The distance between any two silhouettes should always be positive and a minimum distance should only be obtained when comparing a given silhouette to itself. The distance metric should also respect the triangle inequality, as presented in Equation 9.

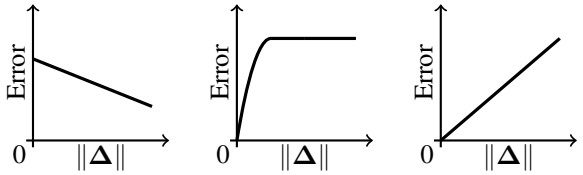
$$D(S_1, S_2) + D(S_2, S_3) \geq D(S_1, S_3) \quad (9)$$

- Robustness Property: Distance measures should be robust to the kind of noise that are often encountered in computer vision problems. Such sources of noise include discretization errors, blur, and occlusion.
- Invariance Property: The metric should be invariant to classes of transformations that are expected to be encountered.

Note that the invariance property is very important for image matching, but less so for the task of human posture tracking.

A stronger criterion can be added to the Metric Property by requiring metrics to increase monotonically as we move away from the correct solution; However, it is not expected that all metrics will do so over the entire pose space. We can sample the error manifold to determine the size of the neighborhood around the correct solution where the metric behaves monotonically. To test this, we will sample a given number of points at a distance $\|\mathbf{\Delta}\|$ and compute statistics of the recorded errors. By gradually increasing $\|\mathbf{\Delta}\|$, we will obtain a curve of how each metric behaves as a function of distance in pose space. From this curve, we will be able to determine if each error respects the Metric Property.

We expect to see three basic metric behavior types, as represented by example plots in Figure 2. The first (Figure 2(a)) is a metric that does not respect the Metric Property as the error does not increase with $\|\mathbf{\Delta}\|$, and is thus not an applicable metric. As all metrics that are tested herein have already been shown to work by the original authors, none of the results should resemble Figure 2(a). The second and third types of metrics (Figures 2(b) and 2(c)) are both valid with respect to the properties listed previously, but the metric in 2(c) is superior in cases where a gradient descent method is used to find the solution as the error increases linearly over the range of $\|\mathbf{\Delta}\|$.



(a) Decreasing (b) Plateauing (c) Linear

Figure 2. Three types of possible metric behaviors

Interestingly, the plot shown in Figure 2(c) is simply a linear plot of the distance in pose space (Error=$\|\mathbf{\Delta}\|$). This allows us to compare metrics by measuring if the metric values correlate well with the distance in pose space. As we are rendering the poses, we know the value of $\mathbf{\Delta_{21}}$ exactly. This makes it possible to directly compute the Pearson product-moment correlation coefficient between $D(S_1, S_2)$ and $D(\mathbf{P_1}, \mathbf{P_2})$.

The Robustness Property will be investigated by testing the metrics with the addition of noise. We will be simulating two common problems found in human tracking applications: ground shadows and camera noise. These sources of noise will change the overall shape of the silhouette and may cause a reduction in the performance of some metrics. Figure 3 shows the silhouettes for six test cases that will be considered.

- Ground Shadows (Figures 3(b) and 3(e)): While there exists a multitude of algorithms to segment the silhouette from an image, most of them suffer from difficulties dealing with shadows [14]. A blob will thus be added below the feet of the silhouette to simulate shadows cast on the ground.
- Camera Noise (Figures 3(c) and 3(f)): Camera noise may also affect the accuracy of the silhouette segmentation. This type of noise will also affect the segmentation and may cause mis-labeled pixels that will result in variations along the edge of the silhouette. To simulate this, we can randomly displace pixels along the edge.

The invariance property can be explored by testing the metrics under different scene configurations. We will run two sets of tests, one with the 3D human model facing the camera (Figures 3(a), 3(b), and 3(c)) and the other with the camera looking at the model sideways (Figures 3(d), 3(e), and 3(f)).

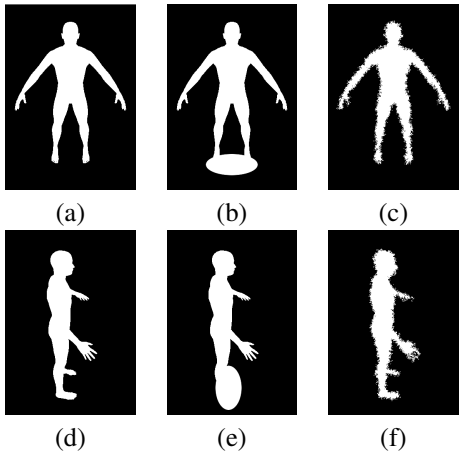


Figure 3. Reference silhouettes used for the tests, viewed from the front (a,b,c) and from the side (d,e,f). Shadows (b,e) and noise (c,f) are added to test robustness.

V. Experimental Methodology

As stated earlier, the initial experiment involved recording the values of the metrics as we move away from the correct solution. The absolute maximum distance between two poses occurs when $\mathbf{P_1} = [0, 0, ..., 0]$ and $\mathbf{P_2} = [1, 1, ..., 1]$, when we obtain $\|\mathbf{\Delta}\| = D(\mathbf{P_1}, \mathbf{P_2}) = 6.08$. To simplify testing procedure, we kept the values of the first 3 DOFs constant at zero. This choice eliminates the effect of the translation components, which have little to no effect on the rendered silhouettes because of the scaling and cropping discussed earlier. For the first few tests, we set $\mathbf{P_1} = [0.5, 0.5, ..., 0.5]$ as a reference pose and vary $\mathbf{P_2}$, which means that the maximum distance between $\mathbf{P_1}$ and $\mathbf{P_2}$ is halved to 3.04.

Metric Values are thus recorded as $\|\mathbf{\Delta}\|$ is increased from 0 to 3.04. For each increment step of $\|\mathbf{\Delta}\|$, a few hundred sample measurements of the metrics are computed in order to calculate the minimum and maximum values of the error as well as average value and standard deviation. These values are presented in Figure 4, with the average as a black line, error bars representing the standard deviation and a gray-shaded region to show the range of metric values between the minimum and maximum. From these results, it is possible to determine the region in which each metric is monotonically increasing and record it in Table II. The Pearson product-moment correlation coefficient between the error function and $\|\mathbf{\Delta}\|$ is computed to further describe the behavior of the metric. To do so, the error manifold of each metric is sampled at a few hundred randomly selected locations within $\|\mathbf{\Delta}\| \in [0, M]$ to generate a set of measurements X. For each of these measurements, the value of $\|\mathbf{\Delta}\|$ is recorded to obtain a set Y. The correlation between these two sets is computed according to Equation 10. The correlation coefficients are also recorded in Table II.

$$\rho_{X,Y} = \frac{\text{cov}(X, Y)}{\sigma_X \sigma_Y} \tag{10}$$

To obtain fair results, the value of M is selected as the largest monotonically increasing region of the metrics in the given test. For example, the correlation coefficients reported for the front of the clean case (column 3) of Table II are computed with $M = \max(0.53, 0.61, 0.76) = 0.76$.

The other measure that is recorded to compare the metrics is the time required to compute the distance between two silhouettes. Similar to the other tests, the distance between a few thousand silhouettes is computed and the execution time is recorded. This allows the computation of the average time and standard deviation. The absolute values alone have little meaning as they are mostly dependent on the hardware used, but the relation between them provides an indication of performance. These results are also be reported in Table II.

To get a more precise understanding of how the metrics behave, we also conduct a set of experiments that records the error values as we move a distance δ from the reference pose along each DOF independently. As these experiments result in a large number of plots, they will not be included within this paper because of space constraints, but can be found at http://cim.mcgill.ca/~olivier/crv2016/, along with all other results.

The final test, which acts as a validation, involves once again the measurement of the metric values as a function of distance in pose space. For this experiment, both poses are selected randomly, instead of keeping the reference pose constant. This should demonstrate how the metrics behave on arbitrary poses and thus confirm wether the results obtained in the previous tests are a good representation of the overall behavior of each metric.

VI. Results & Conclusions

Figure 4 shows the plots of the error versus the pose-space distance for each metric, when applied to the first tested case with the reference silhouette shown in Figure 3(a). The other plots have not been included because of space constraints, but can be found on the aforementioned website. The relevant information from the plots are recorded and summarized in Table II.

As stated earlier, the goal here was not to determine whether the metrics are valid in general cases, as this has

already been done by the original authors, but to systematically determine how appropriate each metric is to the specific task of articulated human posture tracking from silhouettes.

The first result that stands out from Table II is that Hu moments only increase monotonically over a fairly small region of the pose space. While they may be good metrics for matching between shape classes, they are not appropriate for matching slight variations in the pose of humans. The monotonic region measure also allows us to notice that the turning angle metric fails in the presence of noise or shadows. Similarly, the distance signal is seen not to be robust in the presence of shadows. Disregarding these three metrics, we are left with the pixel count, the chamfer distance, and the shape contexts. The chamfer distance is monotonic over a larger region than the other two, especially when the human is not facing the camera. However, the pixel count metric has a higher correlation coefficient in all tested cases.

The last element in Table II that can be used to compare the metrics is the runtime. The first interesting result to note here is that while the type of matching for shape contexts has little to no effect on monotonicity, and only a marginal effect on the correlation to pose-space distance, there is a 20 percent difference in the run times between the two methods. The trade-off between the performance of the metrics and the execution time difference leads us to believe that the added complexity of full bipartite graph matching is not necessary to obtain adequate results. There is a sharp contrast between the run times of the pixel count metric and that of the chamfer distance. The fact that the chamfer distance has a large run time was expected as it is the only one of the algorithms with an $\mathcal{O}(n^2)$ complexity with n being the number of points in the chain code. The complexity of the pixel count is $\mathcal{O}(n)$ with n the number of pixels in the silhouette. The difference in run time is further explained by the fact that the pixel count only requires integer and bitwise operations, whereas the chamfer distance relies on floating point distance computation.

By looking at the results from the test where we move along each DOF independently and comparing the plots for the pixel count and chamfer distances, we can see that the pixel count distance varies more smoothly with δ and presents less small variations (that appear as noise on the curves). This further indicates that the pixel distance is more appropriate for our purposes.

To conclude, the key finding of this paper is that all of the metrics discussed perform reasonably well in ideal conditions, but only the pixel count metric, the chamfer distance, and the shape contexts are robust to the types of noise that are commonly encountered in human posture tracking applications and are thus appropriate for such applications. Furthermore, with a higher correlation to pose-space distance and a lower computation time, the pixel count distance is deemed to be slightly superior to the other metrics.

These conclusions were arrived at by producing a systematically varied dataset under experimental control and probing the strengths and weaknesses of the metrics, rather than relying on a particular existing (and limited) dataset.

ACKNOWLEDGMENT

We thank Prasun Lala for support and constructive comments that helped in the writing of this document.

REFERENCES

[1] J. J. Koenderink, "What does the occluding contour tell us about solid shape?" *Perception*, vol. 13, no. 3, pp. 321–330, 1984.

[2] R. C. Veltkamp, "Shape matching: similarity measures and algorithms," in *Shape Modeling and Applications, SMI 2001 International Conference on.* IEEE, 2001, pp. 188–197.

[3] W. W. Cohen, P. D. Ravikumar, S. E. Fienberg *et al.*, "A comparison of string distance metrics for name-matching tasks." in *IIWeb*, vol. 2003, 2003, pp. 73–78.

[4] R. Rosales and S. Sclaroff, "Inferring body pose without tracking body parts," in *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, vol. 2. IEEE, 2000, pp. 721–727.

[5] M.-K. Hu, "Visual pattern recognition by moment invariants," *Information Theory, IRE Transactions on*, vol. 8, no. 2, pp. 179–187, 1962.

[6] Y. Wang, J. Min, J. Zhang, Y. Liu, F. Xu, Q. Dai, and J. Chai, "Video-based hand manipulation capture through composite motion control," *ACM Transactions on Graphics*, vol. 32, no. 4, p. 43, 2013.

[7] U. Güdükbay, I. Demir, and Y. Dedeoğlu, "Motion capture and human pose reconstruction from a single-view video sequence," *Digital Signal Processing*, vol. 23, no. 5, pp. 1441–1450, 2013.

[8] N. R. Howe, "Silhouette lookup for automatic pose tracking," in *Computer Vision and Pattern Recognition Workshop, 2004. CVPRW'04. Conference on*. IEEE, 2004, pp. 15–22.

[9] Y. Dedeoğlu, B. U. Töreyin, U. Güdükbay, and A. E. Çetin, "Silhouette-based method for object classification and human action recognition in video," in *Computer Vision in Human-Computer Interaction*. Springer, 2006, pp. 64–77.

[10] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 4, pp. 509–522, 2002.

[11] G. Mori and J. Malik, "Recovering 3d human body configurations using shape contexts," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 28, no. 7, pp. 1052–1062, 2006.

[12] A. Agarwal and B. Triggs, "Recovering 3d human pose from monocular images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 1, pp. 44–58, 2006.

[13] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval research logistics quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.

[14] T. B. Moeslund, A. Hilton, and V. Krüger, "A survey of advances in vision-based human motion capture and analysis," *Computer vision and image understanding*, vol. 104, no. 2, pp. 90–126, 2006.

Table II
RESULTS OF THE TESTS

Metric	Clean				Floor Shadow				Noise				Run Time (μs)	
	Monotonic Region		Correlation Coefficient		Monotonic Region		Correlation Coefficient		Monotonic Region		Correlation Coefficient			
	Front	Side	Front	Side	Front	Side	Front	Side	Front	Side	Front	Side	Average	Std.Dev.
Hu Moments	0.53	0.38	0.168	0.337	0.08	0.30	0.117	0.201	0.08	0.38	0.039	0.338	93.70	19.60
Pixel Count	0.61	0.46	0.730	0.746	0.61	0.46	0.747	0.741	0.61	0.76	0.727	0.748	0.59	0.08
Chamfer Distance	0.61	0.76	0.688	0.718	0.61	0.76	0.686	0.709	0.61	0.76	0.681	0.714	4300.88	1071.97
Turning Angle	0.61	1.37	0.672	0.700	0.00	0.61	0.415	0.666	0.08	0.76	0.099	0.572	37.22	6.28
Distance Signal	0.76	0.76	0.796	0.743	0.00	0.76	0.208	0.658	0.76	0.76	0.797	0.718	0.42	0.06
Shape Contexts														
Greedy Matching	0.61	0.46	0.714	0.684	0.61	0.46	0.691	0.659	0.61	0.46	0.703	0.681	30.75	6.86
Bipartite Matching	0.61	0.46	0.744	0.692	0.61	0.46	0.728	0.664	0.61	0.46	0.726	0.677	38.07	6.73

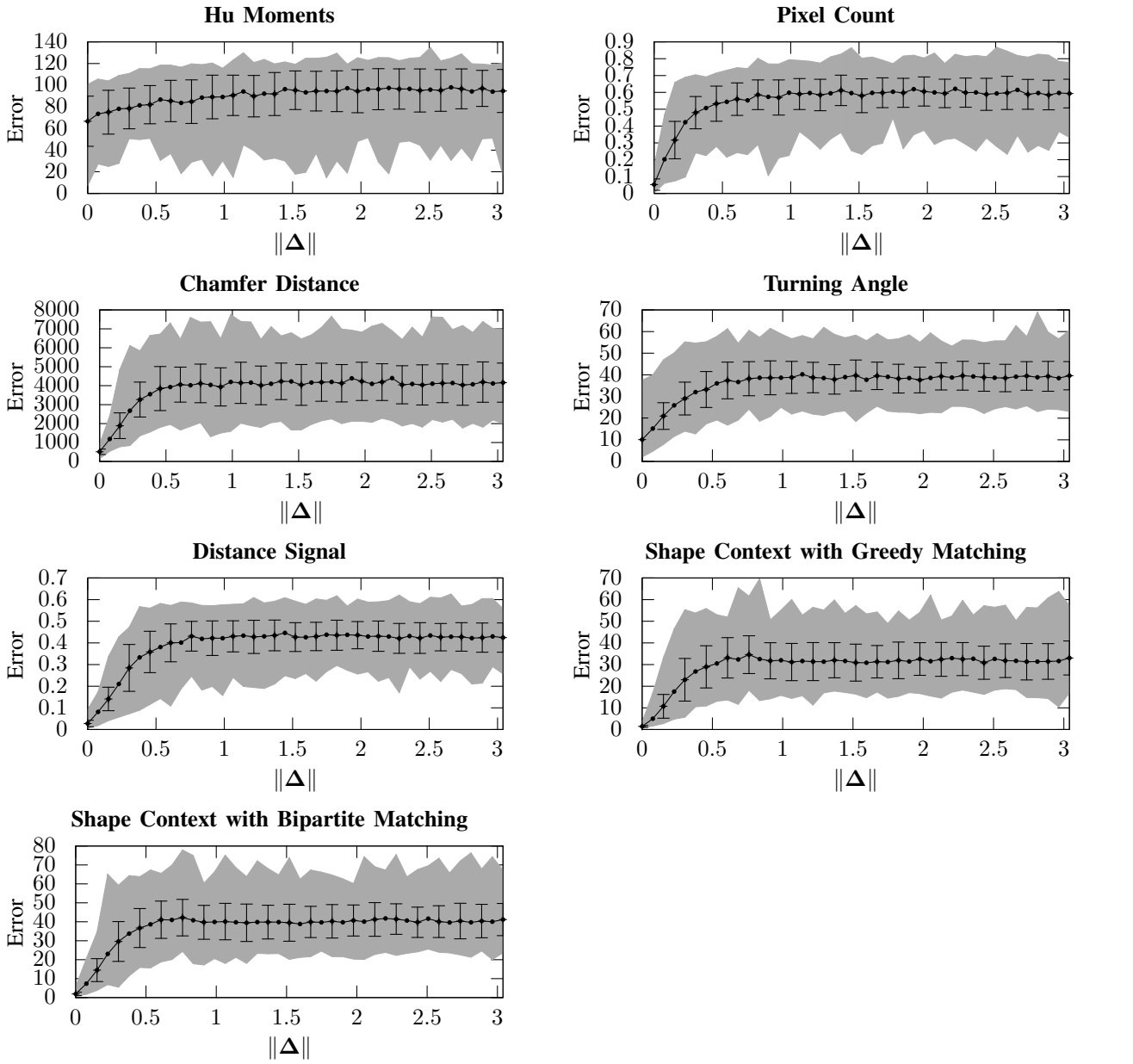


Figure 4. Results for the basic case observed from the front