Efficient Real-time Contour Matching

VARGA Robert, COSTEA Arthur, SZAKATS Istvan, NEDEVSCHI Sergiu
Technical University of Cluj Napoca
Cluj Napoca
Telephone: (800) 555–1212
Fax: (888) 555–1212

***Abstract*—In this paper we present certain shape descriptors from the literature that enable real-time contour matching. We propose slight modifications to these descriptors and more importantly provide fast and efficient matching techniques that return distances between shapes in the order of milliseconds. We show that applying these instead of more time consuming matching algorithms matching accuracy remains the same but the speed is at least 10 times faster. We show results on the MPEG7 benchmark. Using our matching we improve some scores from the literature, this is mainly due to the aligning of the shapes and the usage of the $L1$ metric.**

***Index Terms*—Image processing; shape descriptors; contour matching; object classification; MPEG7 benchmark; shape context descriptor.**

I. Introduction

Determining whether or not two different shapes are the representations of a single object is an important task in computer vision. This can be handled by shape matching. Applications range from content-based annotation/retrieval to medical imaging, tracking, etc. Throughout the years solutions to this problem gave rise to numerous shape descriptors and countless matching techniques. Shape or contour descriptors characterize the boundary of an object by a vector which can be easily matched and also possesses certain important properties such as invariance to translation, scaling, rotation and deformations.

In this paper we focus on selecting and appropriately modifying available contour descriptors. More importantly, we focus on providing matching techniques that perform fairly well and require small execution time. A very fast matching technique is a necessity when KNN retrieval is executed on large shape databases and also for real-time shape matching applications. Even though there are numerous proposed methods both for descriptors and matching algorithms, most of them need about one second for matching two shapes. This is clearly not feasible for real-time contour matching where thousands of shape comparisons must take place in a fraction of a second.

II. Related Work

The technical literature is abundant with methods that tackle the shape matching problem. Three main approaches exist.

The first category characterizes the entire contour by a global descriptor. Elliptic Fourier[1], [2] descriptors, Contour Points Distribution Histogram[3], invariant moments[4], etc. Global descriptors are robust to noise but fine details of the shape contour are lost.

In the second category descriptors are based on a hierarchical representations of the object[5], [6], [7], [8]. These approaches usually require some prior knowledge such as the number of parts and rely on statistical methods to describe the relationships between the parts. This is learned from multiple examples. They can achieve better matching performance than global descriptors, however, for tasks with limited training samples are available or complex shapes they are hard to apply.

The third category methods associate to the parts of the shape [9], [10] or to each contour point a separate descriptor - local shape descriptor[11], [12]. We will focus on this type in the paper. Shape context is an important contribution. It was defined by Belongie et al.[13]. It describes the distribution of the other contour points around a single point. Inner distance shape context[14] extends the previous method by providing a more useful alternative for distance and angle calculations. Other approaches use curvature[15] which measures how fast the unit tangent vector to a curve rotates but necessary adjustments need to be made for the case where the contour is given by a set of discrete points. Flexibility[16] measures the bendable potential of a point. This identifies whether or not the point belonging to the object could move in other possible articulations. Paper [17] presents a method for populating the shape space with ghost-points. The points are introduced in such a way as to make the original metric space more dense while maintaining correct structure (metric space embedding). This improves the accuracy of the matching algorithms.

One can consider a global descriptor by concatenating the local descriptors to form a large vector. This is particularly useful for fast matching techniques. However, it is known that higher retrieval rate is assured by devising specific matching techniques based on the pair-wise similarities of local descriptors between two different contours. Such matching methods are: assignment cost minimization using dynamic programming[13],[14], dynamic time warping[16], earth mover's distance[3]. These aggregate local descriptor distances in a global similarity. In most cases these similarity measures arise as solutions to optimization problems.

Further improvement is possible on the final similarity matrix defined between contours. This is possible by using different meta-similarity measures which correct the similarity matrix based on KNN information [18], [19], [20], [17].

Other works aiming at obtaining fast contour matching

include [21], but there the focus is on retrieval time reduction by pruning. In their approach they have used $L2$ distance for histogram matching instead of the original χ^2.

III. Shape descriptors

This section describes the contour descriptor types utilized in this work. Most of the descriptors are taken from the literature. We additionally provide a simple alternative which behaves relatively well. For every descriptor type we state clearly what are our contributions or modifications if there are any. These are necessary to enable the fast matching that is our goal.

The shape descriptor extraction procedure requires a list of 2D points which represent the boundary points of an object. The result of the procedure is a vector which characterizes the contour. The length of this vector is dependent on the method used as well as specific parameters, but for the same algorithm the vector length is constant.

Shape descriptor extraction is commonly preceded by a sampling phase where only a subset of the contour points is retained. This is done here in a uniform way although one can grant more importance to points where the contour changes orientation - as in [16]. The sampling ensures that all resulting contours will have the same number of points.

In the subsequent sections the following notations are used. Landmarks (samples) from the contour points are notated with $x_i \in R^2$. The fixed number of samples or landmarks obtained through sampling is N. Descriptors for two arbitrary contours are termed p and q. $p, q \in R^{ND}$, where D is the dimensionality of the local shape descriptor. More specifically, p is the global descriptor, while p_i is the local descriptor corresponding to landmark x_i, which in turn can be a vector (e.g. in the case of the inner distance shape context). Summation notation over global descriptor vectors presumes $i \in \overline{1, N}$.

A. Polar Contour Points - PCP

We propose an elementary shape descriptor that is basically the representation of the contour samples in a polar space. This means that dimension of the feature vector is N and $D = 1$. This transformation is achieved by finding the center of mass of the samples, denoted by c. Afterwards, the points are translated so that the center of mass is the origin and we retain only the radii values. This operation achieves *translation invariance*. A normalization of the obtained values grants *scale invariance* to the descriptor. This normalization is in most cases achieved by dividing with the maximum radius. Another possibility is to divide all values by the $L2$ norm of the global vector. This transformation produces a one dimensional vector from a list of 2D points. The advantage of this representation is that it ensures the before mentioned properties and it is compact.

The descriptor values for landmarks x_i are given by the following equation:

$$p_i = \frac{||x_i - c||}{max_j\{||x_j - c||\}}, i \in \overline{1, N} \tag{1}$$

B. Contour Points Distribution Histogram - CPDH

Shu et al. in [3] define a histogram as the global shape descriptor vector. The steps of the extraction algorithm are presented in the following. Firstly, the minimum circumscribed circle is found for the given shape. Afterwards, the area of disk enclosed by circle is partitioned based on the angle and the distance from the center. Let n_d be the number of distance partitions (or bins) and n_θ be the number of angle partitions. The histogram has $n_d \cdot n_\theta$ bins and the histogram value for each bin is equal to the number of points from its corresponding partition. This is essentially the discrete distribution of the contour points in the polar space. For this global descriptor the dimension is $n_d \cdot n_\theta$, which is more compact than for the polar contour points, but information is lost. This dimension remains the same regardless of the number of points used, this is why the entire set of contour points is employed for histogram calculation.

C. Inner Distance Shape Context - IDSC

Shape context - SC - defined in [13] describes the relative distribution of the contour points around a selected individual contour point. More precisely, it is given by the distribution of the vectors to the other points in log-polar space. For point i the histogram is defined as:

$$h_i(k) = |\{x_j : j \neq i, x_i - x_j \in bin(k)\}| \tag{2}$$

The histogram has n_d distance bins and n_θ angle bins. We subtract the angle of the tangent at p_i from the angle of the vector $x_i - x_j$ when determining which bin it belongs to. This is similar to the inner distance and the reason for it is to achieve rotation invariance. However, in some cases this effect is not desirable. The contour will be characterized by a vector of dimension $N \cdot D, D = n_d \cdot n_\theta$, where each point from the shape is described by its corresponding shape context. This descriptor is significantly larger than the preceding ones.

The inner distance variant of the shape descriptor is given in [14]. Instead of binning the vectors based on the $L2$ norm and their angle the inner distance and inner angle is employed. Inner distance between two contour points is defined as the shortest path between the two points that is entirely inside the shape. This path can be obtained by considering the contour points as vertices of a graph and running a shortest path algorithm (Floyd-Warshall, Bellman-Ford). The inner angle at a contour point to another one is the angle enclosed by the direction of the first segment from the shortest path between them and the tangent at the contour point. The inner distance has the important property of being insensitive to articulations, meaning that it remains constant if the object's parts move relative to each other around certain common junctions. Inner distance shape context descriptor extraction requires $O(N^3)$ operations because of the shortest-path algorithm, however this does not affect the matching time.

IV. Real-time contour matching

In order to enable fast matching the distance calculation between two global descriptors must take place in less than

$O(N^2 \cdot D)$. This practically excludes popularly used matching algorithms such as: Dynamic Programming based matching is $O(N^2 \cdot D), D \geq 1$ [13], [14], [22] because of the pair-wise distance calculations, Earth Mover Distance[3] - at least $O(N^2 \cdot D)$, chamfer distance $O(N^2 \cdot D)$[23], [24], [25] and other matching techniques based on optimization problems.

A. Distance types

In this section we enumerate different metrics employed in global shape descriptor matching. Two of the most common Lp metrics provide the baseline. These are applied on the global shape descriptor which is the concatenation of the local shape descriptors.

$$d_{L1} = \sum_i |p_i - q_i| \tag{3}$$

$$d_{L2} = \sqrt{\sum_i (p_i - q_i)^2} \tag{4}$$

We tested three additional distance types which are especially useful for histograms: χ^2, histogram intersection and Kullback-Leibler divergence (in order to avoid 0 and ∞ we add $\epsilon = 10^-5$ to the histograms).

$$d_{\chi^2} = 0.5 \cdot \sum_{k=1}^{D} \frac{(p_k - q_k)^2}{p_k + q_k} \tag{5}$$

$$d_{min} = 1 - \frac{\sum_{k=1}^{D} min\{p_k, q_k\}}{min\{\sum_{k=1}^{D} p_k, \sum_{k=1}^{D} q_k\}} \tag{6}$$

$$d_{KL} = \sum_{k=1}^{D} p_k log \frac{p_k}{q_k} \tag{7}$$

By considering the descriptor as a continuous function we can define a distance based on an inner-product. We notate two arbitrary descriptors as functions f and g respectively, which are periodic of period N. We use the canonical inner product which is given by:

$$< f, g >= \int_0^N f(t)\overline{g(t)}dt \tag{8}$$

One can consider the angle between the two functions as a similarity measure. This is sometimes called the Procrustes distance. It can be derived using the inner product:

$$\theta = cos^{-1} \frac{< f, g >}{||f|| \cdot ||g||} \tag{9}$$

In our case several simplifications can be made. Firstly, since the descriptor is a finite vector the integrals translate to sums. Secondly, if we normalize the descriptor vectors at extraction time by their L2 norms the denominator becomes one and vanishes from (9). Lastly, in order to obtain a distance there is no need to compute the inverse cosine, it suffices to invert the dot product and add one to assure positivity. This gives us the inner product based distance:

$$d_{ip} = 1 - p^T \overline{q} = 1 - \sum_i p_i \cdot \overline{q_i} \tag{10}$$

This is applicable in cases where vector elements are scalars. It is also valid for 2D points if we consider the elements p_i to be complex numbers.

B. Rotational invariance and mirroring

In our framework rotational and mirroring invariance is achieved at the matching phase. There is a possibility to align the descriptors based on the minimum momentum at extraction time, however this is not explored since it is prone to errors. To define a metric between two shapes that ignores rotational transformations we consider several orientations of the second shape. The minimal distance of the first shape and the rotated second shape is considered as the distance. Rotating practically implies translating the index of the second descriptor in the distance sums. Of course, the periodicity forces this translation to be taken in mod N. Since all distances defined require N operations the number of different rotations considered increases the time complexity by a factor of K, K being the number orientations taken into consideration. We provide the formula for the $L1$ distance, which can be easily generalized for the other distance types as well. The *K rotation-invariant distance* between descriptors p and q is given by:

$$\rho(p, q) = min_k\{\rho_k(p, q)\} := min_k\{\sum_i |p_i - q_{(i+k) mod N}|\} \tag{11}$$

In the last equation $k = j\frac{N}{K}, j \in \overline{1, K}$. Note, that ρ is symmetric only if $k|N$, because in this case rotating contour q by k is equivalent to rotating p by $N - k$.

In the case of the CPDH descriptor adjustments have to made to equation (11) to rotate the descriptors only in the angle bins. When using the SC or IDSC p_i and q_i are themselves vectors so the absolute value function is replaced by some norm of the difference ($L1$, χ^2). Since the inner angle is invariant to rotation, it is not necessary to rotate the local IDSC descriptors. However, local descriptors may not be in the same order forcing the verification of different alignments for optimal matching. The same remarks hold for mirrored matching.

Mirrored matching is obtained by considering the second descriptor in reverse order. This includes both the horizontal and the vertical flip of the contour. The *K mirrored rotation-invariant distance* between descriptors p and q is given by:

$$\mu(p, q) = min_k\{\mu_k(p, q)\} := min_k\{\sum_i |p_i - q_{(-i+k) mod N}|\} \tag{12}$$

We encompass the results from the previous metrics in a single distance that accounts for both rotation and mirroring:

$$d_{RM}(p, q) = min\{\rho(p, q), \mu(p, q)\} \tag{13}$$

C. Heuristic for distances

The resulting matching technique from the previous section has the complexity $O(K \cdot N \cdot D)$. In order to further reduce the complexity we propose a heuristic. At summations we consider some subset points sampled equidistantly from original landmarks. The reduced number of samples is $M < N$. We find the angles (indexes) which minimize the following sums:

$$\alpha = argmin_k\{\sum_{m=0}^{M} |p_{mh} - q_{(mh+k) mod N}|\} \tag{14}$$

$$\beta = argmin_k\{\sum_{m=0}^{M} |p_{mh} - q_{(-mh+k) mod N}|\} \tag{15}$$

Where $k = j\frac{N}{K}, j \in \overline{1,K}$ and $h = \frac{N}{M}$. The final distance makes use of the optimal angles to find the distance calculated with all contour landmarks:

$$d^*_{RM}(p,q) = min\{\rho_\alpha(p,q), \mu_\beta(p,q)\} \tag{16}$$

This matching algorithm has the time complexity, $max\{O(K \cdot M \cdot D), O(N \cdot D)\}$, but usually $K \cdot M > N$. The advantage here is that we align the shapes based on some reduced number of points and in most cases this is sufficient to obtain to good guess. At the end we use the full descriptor to find the true distance.

D. Flexible matching - invariance to small deformations

Flexible matching entails relaxing the one-to-one correspondence between the feature vector elements at distance calculation. More precisely, descriptor element p_i from the first vector is matched to the closest descriptor in a limited neighborhood of q_i. In this sense the $\rho_k L1$ distance is modified as (the form for μ^*_k is analogous):

$$\rho^*_k(p,q) = \sum_i \min_l\{|p_i - q_{(i+k+l) mod N}|\}, l \in [-\delta, \delta] \tag{17}$$

This technique is applicable to every distance type defined and is especially useful for more descriptive features such as the IDSC. The operations would multiply the matching complexity with a factor of $\Delta = 2\delta + 1$ if it were applied for every angle. However, this can be used only at the optimal angle to avoid this effect. More specifically, we use (14) and (15) to find the optimal alignment angles and calculate the final distance using flexible matching. By considering a small neighborhood we loose the symmetry of the matching function but we gain in robustness.

$$d^*_{RM}(p,q) = min\{\rho^*_\alpha(p,q), \mu^*_\beta(p,q)\} \tag{18}$$

σ(i)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
i	16	8	4	9	2	10	5	11	1	12	6	13	3	14	7	15

Fig. 1: Mapping order for greedy matching for $N = 16$

E. Layered Greedy Matching

We propose a greedy matching scheme. This associates local descriptor p_i to the closest local descriptor q_j. Matching is exclusive, meaning that each q_j can only be used once. To avoid unrealistic alignment of the first descriptors, local descriptors from p are considered in a specific order (Fig. (1)). This order is given by the following fractions of the descriptor size N:

$$\sigma = [\frac{1}{2}, \frac{1}{4}, \frac{3}{4}, \frac{1}{8}, \frac{3}{8}, \frac{5}{8}, \frac{7}{8}, ..., \frac{2^{l-1}-1}{2^l}] \cdot N + 1 \tag{19}$$

l is the number of levels considered, if N is not divisible by 2^l the integer part function is applied. One can view this as matching the descriptors at a coarse level than refining the score using more and more points. This ensures a good global matching. An even more efficient strategy limits the maximum match cost between two local descriptors by a factor τ_G.

By stopping before all the indexes are exhausted - $\frac{2^{l-1}-1}{2^l} = L < N$ - matching complexity remains below $O(N^2 \cdot D)$ but this is the most time consuming of the techniques presented here. This is related to the chamfer(21) distance. However, here the matching is exclusive and it is done in a specific order. Also chamfer distance is applied directly on the contour points using distance transform.

$$d_G = \sum_{i=1}^{L} min_{j \in J_i} |p_{\sigma(i)} - q_j|, \tag{20}$$

$$J_1 = \overline{1,N}, J_{n+1} = J_n - argmin_j |p_{\sigma(n)} - q_j|$$

$$d_{ch}(p,q) = \frac{1}{N} \sum_i min_j\{|p_i - q_j|\} \tag{21}$$

F. Assignment cost minimization

This method was used in [13], [14]. It is presented here to show speed differences. It involves minimizing the total assignment cost which is efficiently solved through dynamic programming:

$$d_{DP} = min_\pi \sum_i c(p_i, q_{\pi(i)}) \tag{22}$$

where π is the mapping of indexes from the first descriptor to the second. It is increasing so that the order of the landmarks is preserved. $c(p_i, q_j)$ is the cost of the assignment and takes the value of the distance $d(p_i, q_j)$ or τ if p_i is not assigned. Superior performance of the DP matching can be justified by the fact that it allows flexible matching and also it is possible to not assign some noisy descriptors at all. The time complexity of the algorithm is $O(N^2 \cdot D)$. If we consider again several

different orientations for the second descriptor the complexity is multiplied by K.

V. Implementation details

All test times provided are obtained on a machine with Intel 2.4GHz two core processor and 8GB RAM. Contour extraction starts from the leftmost and highest point and proceeds by having the interior of the contour on the left. We have implemented the algorithms in C and have used multiple threads for execution.

All rotation and modulo operations can be carried out efficiently by copying the second descriptor after itself in the memory. This enables iterating through the sums without calculating $mod N$ every time. The mirroring is obtained by creating a copy in inverse order of the second descriptor. Further optimization is possible by using Streaming SIMD Extensions which greatly increase the speed of floating operations on vectors. Speed boost is obtained by compiling our source code with Intel C++ Composer XE 2011 compiler.

VI. Experimental Results

We have performed two types of experiments in order to validate the exposed methods. It is a well accepted procedure to test contour descriptors on benchmarks such as the MPEG7, Swedish Leaf or Kimia dataset. We provide retrieval rate results for all of the methods on the MPEG7 dataset. This dataset contains 70 types of objects with 20 images each. During retrieval the 40 most similar shapes must be returned for every shape from the database. Retrieval rate represents the sum of the number of correct retrievals (that share the same class as the query contour) divided by the possible maximum number of correct answers (1400x20). It is also called bull's eye score.

Best bull's eye scores from the literature are in the range 75-89% (shape context 75%[13], IDSC 85%[14], flexibility 89%[16]). These include only methods that do not use meta-similarity measures. Meta shape similarities are applied after the distances between shapes are found as a postprocessing step and can boost the recall rate with 4-8%.

In Table (IV) we provide the recall rates on the MPEG7 database using different descriptor types (see Table (I) for descriptor parameters) and matching techniques (see Table (II)). Our implementation results are close those reported in [14]. It is important to note that we obtain above 80% with SC with our matching as opposed to the original 76%. This is due to rotational matching and the $L1$ metric. The speed gain is obvious and significant. A single shape comparison takes place in 0.3 seconds as reported in [14] while here, we obtain 1400x700 comparisons in 4700 seconds in the case of the same DP matching (4ms for one comparison) and an even lower 500 seconds (0.5ms for one comparison) using the proposed matching technique d^*_{RM}. Our matching technique obtains approximately the same recall value for the same underlying features but is at least 10 times faster compared to DP matching. Greedy matching is both slower than d^*_{RM} and weaker with 1-2% but still performs well. Without the specific ordering greedy matching performs worse when L approaches N, e.g. on a simple test 77.8% as opposed to 79.3%. Also note, that the naive PCP method performs relatively well while being much faster both at extraction time and matching time.

TABLE I: Shape descriptor parameters

Descriptor	**N**	n_d	n_θ	**time**	**exec**
fourier	200	-	-	$O(N)$	-
CPDH	-	5	80	$O(N)$	0.1
PCP	500	-	-	$O(N)$	0.02
SC	300	8	12	$O(N^2)$	40.3
IDSC	300	8	12	$O(N^3)$	123.2

N is the number of contour samples, n_d the number of distance bins, n_θ the number of angle bins (where applicable). Column time gives the time complexity of the descriptor extraction procedure. Execution time refers to global run time for obtaining all 1400 descriptors.

TABLE II: Time complexities of different matching algorithms

Method	**compl.**	**Method**	**compl.**
Lp	$O(N \cdot D)$	d_{ip}, d^2_χ	$O(N \cdot D)$
d_{RM}	$O(K \cdot N \cdot D)$	d^*_{RM}	$O(K \cdot M \cdot D)$
d_G	$O(L \cdot N \cdot D)$	d_{dp}	$O(K \cdot N^2 \cdot D)$
chamfer	$O(N^2 \cdot D)$	EMD	$\Omega(N^2 \cdot D)$

Since local descriptors may be histograms of length D every complexity contains D. Different implementations of EMD exist but all are above N^2.

TABLE III: Effect of distance type when comparing histograms

	L1	**L2**	χ^2	d_{min}	KL	d_{ip}
score	78.1%	68.7%	76%	73.8%	65.6%	47.4%
time	1	1.14	5.42	1.5	9.69	1.07

We have tested all the distance functions enumerated. In almost all cases $L1$ distance performs best while being also the fastest. Very similar results are obtained using χ^2. General observation is that $L2$ is poor for contour matching. Typical score values are given in Table (III) using SC with $N = 50, K = 1, n_\theta = 6, n_d = 4$. The second line indicates relative execution times obtained by dividing with $L1$ case. d_{ip} is well suited for global descriptors and not for histograms. PCP with d_{ip} achieves a score that is 0.3% lower than with $L1$.

Another ad hoc experiment that is undertaken uses a more realistic dataset and aims at utilizing the fastness of the matching methods. We apply a retrieval based shape classification on contours that represent human body postures. We have worked with 9 pose classes: *standing, walking, running, bowing-down, jumping, crouching, waving, kicking, punching*(see Figure (2)). The contour database is obtained from the depth images provided by Microsoft Kinect motion sensing input device. The depth sensor consists of an infrared laser projector combined

(a) walking (b) kicking (c) punching (d) jumping

Fig. 2: Sample contours obtained from Kinect depth images

TABLE IV: Retrieval rates from the MPEG7 dataset

Method	Score	K/L	M	Match comp.	exec
fourier + $L1$	55%	-	-	$O(1)$	-
PCP + d_{ip}	73.5%	200	-	$O(N \cdot K)$	58s
PCP + d_{RM}	73.7%	200	-	$O(N \cdot K)$	40s
CPDH + d_{RM}	74.0%	80	-	$O(n_d \cdot n_\theta \cdot K)$	35s
IDSC + greedy	77.9%	128	-	$O(N \cdot D \cdot L)$	1360s
IDSC + d^*_{RM}	77.9%	50	20	$O(M \cdot D \cdot K)$	144s
IDSC + DP	80.2%	8	-	$O(N^2 \cdot D)$	4700s
SC + $greedy$	80.8%	128	-	$O(N \cdot D \cdot L)$	257s
SC + d^*_{RM}	81.3%	100	100	$O(N^2 \cdot D)$	500s
SC + DP	82.5%	8	-	$O(N^2 \cdot D)$	10120s

The first column indicates the descriptor type and the distance used for matching. K is the number of rotations considered $K < D$. L denotes the number of points used for greedy matching with $L < D$. The last column indicates the time necessary to compute the pairwise distances of all 1400 shapes (essentially 1400x700 matches).

with a monochrome CMOS sensor, which captures video data in 3D under any ambient light conditions. Shape context descriptors are extracted from the contours provided in real-time and the closest contour from the training database is retrieved. This means that both extraction and retrieval must be done between two frames. Using SC+d^*_{RM} with $N = 100$ we can classify 30 per second for every frame provided by the device while DP matching enables only 15 or less. The contour is then classified as being in the same class as the returned shape from the prototype database. We have obtained a classification accuracy of 91% for our noisy pose contours with SC+d^*_{RM}.

VII. Conclusion

The aim of this work was to establish a handful of shape matching methods that perform well and are also fast. This is necessary for real-time applications. Even though the technical literature is abundant with existing methods that perform excellent on synthetic databases they are usually very slow. Our goal was to find a compromise between shape descriptor and matching complexity and fastness. To this end we have made use of several well established contour descriptors and implemented them along with some proposed modifications.

We have proposed a matching technique that relies on checking different contour orientations at match time. This is done efficiently while maintaining matching accuracy. This claim is demonstrated by the experimental results which show that this technique gives the same results as some more time-consuming variants. This proves that it is excellent for fast shape matching.

References

[1] L. Yang, P. Meer, and D. J. Foran, "Multiple class segmentation using A unified framework over mean-shift patches," in *CVPR*, 2007, pp. 1–8.

[2] D. Comaniciu, P. Meer, and D. J. Foran, "Image-guided decision support system for pathology," *Machine Vision and Applications*, vol. 11, no. 4, pp. 213–224, 1999.

[3] X. Shu and X. jun Wu, "A novel contour descriptor for 2D shape matching and its application to image retrieval," *Image Vision Comput*, vol. 29, no. 4, 2011.

[4] P. L. Rosin, "Shape description by bending invariant moments."

[5] P. F. Felzenszwalb and J. D. Schwartz, "Hierarchical matching of deformable shapes," in *CVPR*, 2007, pp. 1–8.

[6] P. F. Felzenszwalb, "Object recognition with pictorial structures," in *MIT AI-TR*, 2001.

[7] S. Agarwal, A. Awan, and D. Roth, "Learning to detect objects in images via a sparse, part-based representation," *IEEE Trans. Pattern Anal. Mach. Intell*, vol. 26, no. 11, 2004.

[8] R. Fergus, P. Perona, and A. Zisserman, "Object class recognition by unsupervised scale-invariant learning," in *CVPR*, 2003, pp. II: 264–271.

[9] R. Basri, L. Costa, D. Geiger, and D. W. Jacobs, "Determining the similarity of deformable shapes," in *Physics Based Modeling Workshop in Computer Vision*, 1995, p. SESSION 5.

[10] A. I. Comport, E. Marchand, and F. Chaumette, "Object-based visual 3D tracking of articulated objects via kinematic sets," in *Workshop on Articulated and Non-Rigid Motion*, 2004, p. 2.

[11] B. B. Kimia, A. R. Tannenbaum, and S. W. Zucker, "Shapes, shocks, and deformations I: the components of two-dimensional shape and the reaction-diffusion space," *International Journal of Computer Vision*, vol. 15, no. 3, pp. 189–224, 1995.

[12] T. B. Sebastian, P. Klein, and B. B. Kimia, "Recognition of shapes by editing their shock graphs," *ieeetopami*, vol. 26, pp. 550–571, 2004.

[13] S. Belongie, J. Malik, and J. Puzicha, "Shape context: A new descriptor for shape matching and object recognition." MIT Press, 2000.

[14] H. B. Ling and D. W. Jacobs, "Shape classification using the inner-distance," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 29, no. 2, pp. 286–299, Feb. 2007.

[15] M. A. Fischler and H. C. Wolf, "Locating perceptually salient points on planar curves," *IEEE Trans. Pattern Anal. Mach. Intell*, vol. 16, no. 2, pp. 113–129, 1994.

[16] C. Xu, J. Liu, and X. Tang, "2D shape matching by contour flexibility," *IEEE Trans. Pattern Anal. Mach. Intell*, vol. 31, no. 1, pp. 180–186, 2009.

[17] X. W. Yang, S. K. Tezel, and L. J. Latecki, "Locally constrained diffusion process on locally densified distance spaces with applications to shape retrieval," in *CVPR*, 2009, pp. 357–364.

[18] A. Egozi, Y. Keller, and H. Guterman, "Improving shape retrieval by spectral matching and meta similarity," *IEEE Transactions on Image Processing*, vol. 19, no. 5, pp. 1319–1327, 2010.

[19] P. Kontschieder, M. Donoser, and H. Bischof, "Beyond pairwise shape similarity analysis," in *ACCV (3)*, ser. Lecture Notes in Computer Science, H. Zha, R. ichiro Taniguchi, and S. J. Maybank, Eds., vol. 5996. Springer, 2009, pp. 655–666.

[20] X. W. Yang, X. Bai, L. J. Latecki, and Z. W. Tu, "Improving shape retrieval by learning graph transduction," in *ECCV*, 2008, pp. IV: 788–801.

[21] G. Mori, S. J. Belongie, and J. Malik, "Efficient shape matching using shape contexts," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 27, no. 11, pp. 1832–1837, Nov. 2005.

[22] E. Petrakis, A. Diplaros, and E. Milios, "Matching and retrieval of distorted and occluded shapes using dynamic programming," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 11, pp. 1501–1516, Nov. 2002.

[23] M.-Y. L. 0001, O. Tuzel, A. Veeraraghavan, and R. Chellappa, "Fast directional chamfer matching." IEEE, 2010.

[24] G. Borgefors, "Hierarchical chamfer matching: a parametric edge matching algorithm," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 10, pp. 849–865, 1988.

[25] A. Thayananthan, B. Stenger, P. H. S. Torr, and R. Cipolla, "Shape context and chamfer matching in cluttered scenes," in *CVPR*, 2003, pp. I: 127–133.