AFFINE INVARIANT CURVE MATCHING USING NORMALIZATION AND CURVATURE SCALE-SPACE

V. Giannekou, P. Tzouveli, Y. Avrithis and S.Kollias

Image Video and Multimedia Systems Laboratory,
School of Electrical and Computer Engineering,
National Technical University of Athens,
Iroon Polytexneiou 9, 15773 Zografou, Greece
{vickyg,tpar,iavr,stefanos@image.ntua.gr}

ABSTRACT

In this paper, an affine invariant curve matching method using curvature scale-space and normalization is proposed. Prior to curve matching, curve normalization with respect to affine transformations is applied, allowing a lossless affine invariant curve representation. The maxima points of the curvature scale-space (CSS) image are then used to represent the normalized curve, while retaining the local properties of the curve. The matching algorithm that follows, matches the maxima sets of CSS images and the resulting matching cost provides a measure of similarity. The method's performance and robustness is evaluated through a variety of curves and affine transformations, obtaining precise shape similarity and retrieval.

1. INTRODUCTION

Shape is known as one of the most important visual features, since it enables us to recognize objects without using further information. Consequently, shape representation, recognition and matching are considered to be an important element in image processing field. Many methods have been proposed in the literature for efficient representation and analysis of boundary curves extracted from real objects. Shape representation methods focus either on shape boundaries (boundary-based methods) which use curvature [1]-[4] or on interior region of shape (region based methods) which use moments [5]-[8] and Fourier descriptors [9]-[12].

Many approaches have taken under consideration the problem of drastic shape change due to perspective transformations when objects are far from the camera. Therefore, invariance to viewpoints is a desirable property in many shape recognition systems. These deformations can be modeled approximately by a general affine transformation. As a result, affine invariance in shape representation has proved to be of critical importance and several methods have been proposed in order to deal with the problem of affine transformations.

A number of shape representations have been proposed to recognize shapes even under affine transformation [6],[7],[10]], affine invariant scale space [1]-[3] and affine curvature [4] have also been explored. A number of shape representation techniques are based on level-set methods [13].

Mokhatarian et al [2] proposed an affine invariant shape representation of closed curves based on multi scale theory and curve evolution called CSS shape representation, which has been set as one of MPEG-7 descriptors. The shape representation is computed by convolving the curve with a Gaussian function at different levels of scale and extracting the location of inflection points of the resulting curves in order to construct the curvature scale space (CSS) image. Next, to perform the matching process, the maxima of the CSS image [2] are extracted and matched using a proposed matching algorithm, which compares two sets of maxima and assigns a matching value to them. This method ensures affine invariance through affine length parametrization of the curve. As a result, the curve is being normalized while, on the other hand, preserving all of its local properties.

Even though the CSS method has been proved to be robust under affine transformations, affine length parametrization provides affine invariance in case of translation, scaling, rotation and shear but doesn't apply in case of reflection and starting point transformation. For this reason, the CSS matching algorithm attempts to apply a circular shift to the CSS image equivalent to the change in orientation of the corresponding object, but its performance depends on a hypothesis for the required shift that may not lead to the best match.

In this paper we attempt to improve the CSS method's efficiency, replacing the affine length normalization method with an affine invariant curve normalization method proposed by Avrithis et al [3]. The alternative normalization method, which is based on moments and Fourier coefficients, seems to improve affine robustness of transformed curves without any loss of information. The matching scheme of the proposed method is based on calculating the Euclidean distances between two maxima sets of CSS images. The measure of similarity is provided by a resulting matching cost which is in fact a summation of minimum distances between maxima sets. The processes of affine invariant representation and matching are completely decoupled in this approach.

The paper is organized as follows Section 2 refers to the CSS shape representation using affine length parametrization while Section 3 presents the CSS shape representation using affine invariant curve normalization. The performance of the proposed method is examined in experimental field as it is presented in Section 4.

2. CSS SHAPE REPRESENTATION AND MATCHING USING AFFINE LENGTH PARAMETRIZATION

2.1 Curve parametrization

Curvature scale space representation finds its roots in multi–scale shape representation theory which handles image structures at different levels of scale. It is a method mainly used in the problem of shape similarity retrieval and it has proved to be efficient and robust under general affine transformations. The CSS representation of an object results to the construction of its CSS image, following a procedure of steps that also ensures affine invariance.

In the first step, the equation curve is being parameterized by an arbitrary parameter *u*. If u is the arc length parameter, the resulting representation is called the *natural* representation of the curve and seems to be robust to Euclidean transformations. However, arc length is not preserved under affine transformation, which led to the use of *affine length* parameter. Therefore, a resampling based on affine length parametrization follows, in order to achieve affine invariance but also preserve the curve's local characteristics.

2.2 CSS image construction

After applying the affine length parametrization, the curve is being smoothed by convolving each of its coordinates with a *Gaussian* function, while increasing the width of the Gaussian kernel. The resulting process creates ordered sequences of curves and is referred to as the *evolution* of the curve. Following curve smoothing, that reduces the effect of noise, the curvature function is calculated in order to find the locations of the curvature's *zero crossing* points.

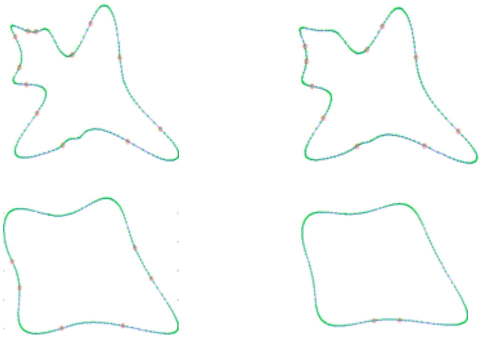


Figure 1: Curve evolution

The equation is shown below:

$$\kappa(u,\sigma) = 0 \qquad (1)$$

where *u* is an approximation of the affine length parameter and σ is the width of the Gaussian kernel. The resulting locations are depicted on Fig.1 during the curve evolution.

Through the curve evolution, and as *s* increases, the curve becomes smoother and it is observed that zero crossing points tend to join in pairs whenever a concave or convex part of the curve is being smoothed. Finally, when the last pair is joined, the evolution comes to an end and the resulting curve contains no concavities or convexities. Zero crossing points seem to define each curve and carry valuable information of its form. Therefore, zero crossing point locations are displayed below in a (u, σ) plane, where *u* is an approximation of the affine length parameter and σ is the width of the Gaussian kernel. The result of this process can be represented as a binary image called the *regular* CSS image of the curve (Fig. 2).

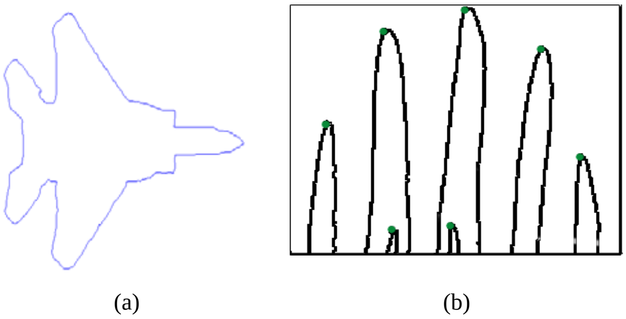


Figure 2: (a) Boundary curve (b) Regular CSS image of the curve and its maxima points

The intersection of every horizontal line with the contours in the image shown in Fig. 2 indicates the locations of curvature zero crossings of the corresponding evolved curve. Each concavity of the original curve is represented in both curve evolution (Fig. 1) as well as in regular CSS image (Fig. 2). In fact, there are two zero crossings of the curvature on every concave part of the shape, and as the curve becomes smoother, these points approach each other and finally join, when the concavity is filled. The resulting point represents the maximum of the relevant contour; basically it represents an actual segment of the shape. Therefore, each contour *maxima* in the CSS image is being extracted – shown on Fig. 2(b) – and their locations are then used to represent every image boundary.

Unfortunately, the locations of maxima are not readily available and must be extracted from the CSS image. Notice that the CSS contours are usually connected everywhere except in a neighborhood of their maxima. Therefore, in order to find the maxima locations, the peaks of both branches of a contour in the CSS image are found and the midpoint of the line segment joining the pair is considered as maxima of the CSS image. Small contours of the CSS image are related to noise and are not included in the final representation. So, if a maximum is less than a certain threshold *t*, it is considered as noise. An indicative value for the threshold could be the 0.2 of the largest maximum of the CSS image.

2.3. CSS matching

Following the CSS maxima extraction, the CSS *matching algorithm* compares two sets of maxima and assigns a matching value to them. The matching value represents the similarity measure between the actual boundaries of image contours.

The first step of the matching algorithm deals with the problem of different starting points of matching CSS images. A solution could be applying a circular shift to one of the two sets of image maxima, so that the effect of randomly selected starting point is compensated. Since the exact value of required shift is not available, it is assumed that the *best shift* is the one that shifts one CSS image so that its major maximum equals the major maximum of the other CSS image. Each maximum is then matched according to the minimum Euclidean distance·the final *total cost* of the match is the summation of Euclidean distances between the matched maxima [2].

Basically, the above matching algorithm attempts to normalize possible reflection and starting point transformation of the image through a circular shift. The hypothesis that the major maxima shift is considered as the best shift does not always lead to the best match, so alternative hypotheses for all possible shifts need to be examined. For each case, the matching cost should be determined and its lowest value will then represent the best match, a time consuming and error-prone process. In the proposed approach, on the other hand, starting point normalization is curried out before matching, making such search process unnecessary. Normalization in this case also includes reflection, apart from any affine transformation.

3. CSS SHAPE REPRESENTATION AND MATCHING USING AFFINE INVARIANT NORMALIZATION

As shown above, CSS representation using affine length parametrization seems to be efficient and robust under affine transformations. Nevertheless, its efficiency depends on successful affine normalization and matching algorithm implementation, procedures that can both include error possibility. Affine length parametrization can ensure affine invariance in case of translation, scaling, rotation and shear of the original curve but doesn't apply in the case of reflection and starting point transformations.

In order to improve the CSS method's efficiency, we replace the affine length normalization method with an alternative normalization method [3].

Consequently, a new curvature shape representation (N-CSS) is proposed here, aiming at minimizing the effect of affine transformation in shape representation. N-CSS shape representation is based on the existing CSS representation method and attempts to improve the method's performance by introducing alternative affine invariant curve normalization.

3.1 Curve normalization

Assume that the parametric vector equation for a curve is considered as:

$$s(u) = (x(u), y(u)) \qquad (2)$$

where *u* is an arbitrary parameter.

At first, in order to obtain the discrete points of the curve, a uniform sampling is applied and the object's boundary is represented by a number of equally distant points. Then, the boundary curve is being normalized in order to become robust to general affine transformation. The normalization method is considered to ensure affine invariance with respect to any affine transformation, including translation, scaling, rotation, shear, starting point and reflection transformations. *Affine invariant curve normalization* for shape based retrieval is a method for two–dimensional curve normalization that can be applied as a preprocessing step to any shape representation, classification, recognition or retrieval technique, without any actual loss of information of the original curve. In particular, the curve is being normalized in several stages in order to eliminate translation, scaling, rotation and shear as well as starting point and reflection transformations. Normalization is based on a combination of curve features including moments and Fourier descriptors.

The first stage of normalization includes curve *orthogonalization*, a procedure which effectively manages to normalize the original curve with respect to translation, scaling and shear transformation. In fact, affine transformations are reduced to orthogonal ones based on curve *moments*. For each curve *s*, the (*p,q*)-order moments

$$m_{pq}(s) = \frac{1}{N}\sum_{i=0}^{N-1} x_i^p y_i^q \qquad (3)$$

of order up to two are used for the construction of the corresponding *normalized curve* $n_a(s)$.

Below, the individual orthogonalization steps are concisely presented:

1. The center of gravity of the curve is normalized so as to coincide with the origin:

   $$x_1 = x - m_{10}(s), \qquad y_1 = y - m_{01}(s) \qquad (4)$$

   where $\mu_x = m_{10}(s), \mu_y = m_{01}(s)$.
2. The curve is scaled horizontally and vertically so that its second-order moments become equal to one:

   $$x_2 = \sigma_x x_{1,} \qquad y_2 = \sigma_y y_1 \qquad (5)$$

   where $\sigma_x = 1/\sqrt{m_{20}(s_1)}, \sigma_y = 1/\sqrt{m_{02}(s_1)}$.
3. The curve is rotated counterclockwise by $\theta_0 = \pi/4$:

   $$s_3 = R_{\pi/4}s_2 = \frac{1}{\sqrt{2}}\begin{bmatrix} x_2 - y_2 \\ x_2 + y_2 \end{bmatrix} \qquad (6)$$

   where R_θ is a 2 x 2 matrix corresponding to a counterclockwise rotation by θ radiants.
4. Finally the curve is scaled again, exactly as in step 2:

   $$x_4 = \tau_x x_{3,} \qquad y_4 = \tau_y y_3 \qquad (7)$$

   where $\tau_x = 1/\sqrt{m_{20}(s_3)}, \tau_y = 1/\sqrt{m_{02}(s_3)}$.

The normalized curve $n_a(s) \equiv s_4$ can also be written as:

$$\begin{aligned} n_a(s) &= N(s)(s - \mu(s)) \\ &= \frac{1}{\sqrt{2}}\begin{bmatrix} \tau_x & 0 \\ 0 & \tau_x \end{bmatrix}\begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}\begin{bmatrix} \sigma_x & 0 \\ 0 & \sigma_x \end{bmatrix}\left(\begin{bmatrix} x - \mu_x \\ y - \mu_y \end{bmatrix}\right) \end{aligned} \qquad (8)$$

where $\mu(s) = [m_{10}(s) m_{01}(s)]^{\mathrm{T}}$ and *N(s)* denotes the 2 x 2 *normalization matrix* of *s*. The resulting curve may contain only rotation and/or reflection transformations that are examined in the following normalization stage.

The second normalization stage applies *starting point* normalization for closed curves. As a result, rotation and reflection normalization follows, since they are affine transformations depended on the starting point of the curve. The normalization procedure with respect to the starting point is based on discrete Fourier transform of the complex vector z representing a curve:

$$u_k = \sum_{i=0}^{N-1} z_i \omega^{-ki}, k = 0,1,\ldots,N-1 \qquad (9)$$

For each element of the Fourier transform we will employ its primary argument, or phase, defined as $\alpha_k = \arg u_k = \theta \in [0, 2\pi) : u_k = re^{j\theta}, r \in \Re^+$.

The corresponding phase vector is then $a = \arg u = [a_0 \quad a_1 \quad \cdots \quad a_{N-1}]$.

Then, this normalization stage is completed by applying a circular shift, if needed, to the starting point. Therefore, we estimate a *standard* circular shift for each curve, based on the difference between the first and last Fourier phases:

$$p(z) = \left[\frac{N}{4\pi}(a_1 - a_{N-1})\right] \bmod N/2 \qquad (10)$$

and then apply the opposite shift in order to normalize the curve:

$$n_p(z) = S_{-p(z)}(z). \qquad (11)$$

In the final normalization stage, the curve is being normalized with respect to *rotation* and *reflection* transformation. As in the starting point case, normalization is based on the first and last Fourier phase a_1 and a_{N-1} of **z**, and consists of the following steps:

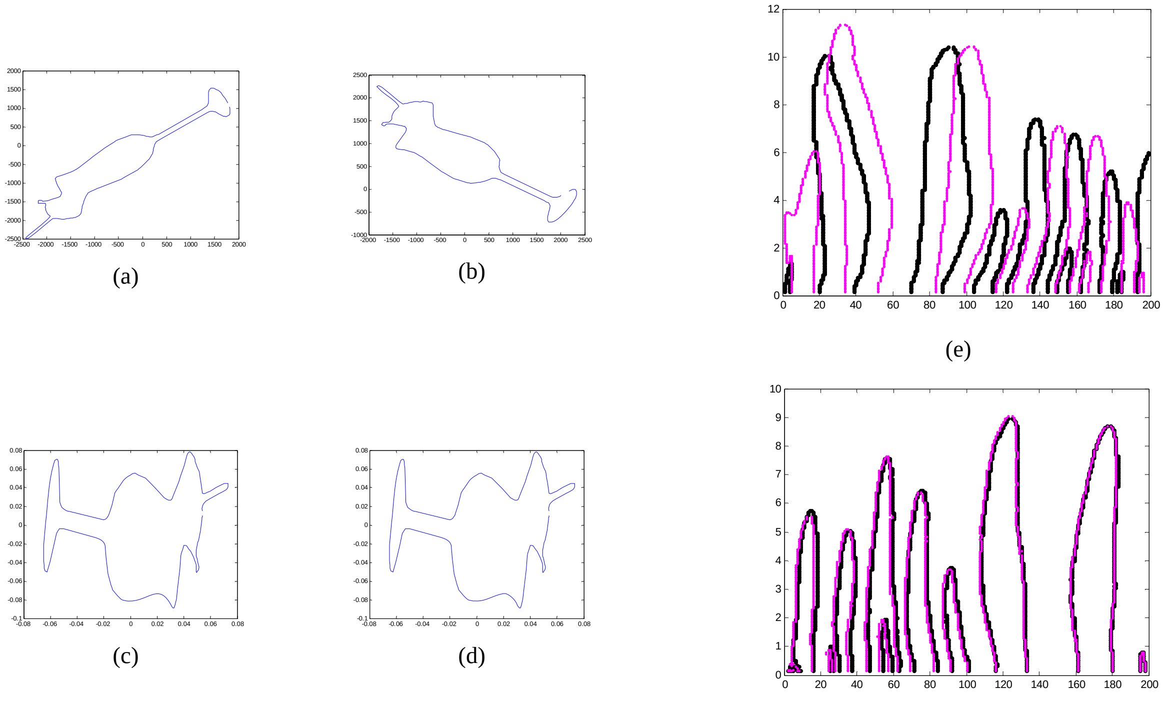
1. The rotation of the initial curve **z** is normalized according to the average value of a_1 and a_{N-1}:

   $$\begin{aligned} r(z) &= \left(\frac{1}{2}(a_1 + a_{N-1})\right) \bmod \pi \\ z_1 &= ze^{-jr(z)} \end{aligned} \qquad (12)$$
2. The horizontal and vertical reflection is normalized according to the third-order moments of z_1:

   $$\begin{aligned} v(z_1) &= v_x(z_1) + jv_y(z_1) \\ &= \operatorname{sgn} m_{12}(z_1) + j \operatorname{sgn} m_{21}(z_1) \\ n_r(z) &= z_2 = v_x(z_1)x_1 + jv_y(z_1)y_1 \end{aligned} \qquad (13)$$

where sgn denotes the signum function

Two final steps are required after rotation and reflection normalization. First, the starting point ambiguity of *N*/2 that may have derived from the starting point normalization is resolved by applying an additional circular shift of *N*/2 samples, if $x_0 < 0$; otherwise the normalized curve is left intact. Secondly, the curve orientation is normalized to being counterclockwise.



(f)

Figure 3: Normalized curves with affine length parametrization (a),(b) (CSS method), with affine normalization (c), (d) (N-CSS method) and corresponding CSS images (e) and (f)

3.2 Normalized CSS matching

After applying curve normalization, the proposed method follows respectively the exact steps of the CSS method, which include curve smoothing, construction of the CSS image and finally, the matching procedure.

So, initially, the resulting normalized curve is being smoothed using the Gaussian function. Each coordinate of the boundary curve is convolved with a Gaussian function while increasing the width of the Gaussian kernel σ.

We continue by constructing the CSS image of the curve and extracting the locations of CSS image maxima. As mentioned in Section 2, maxima locations are indicative of each CSS image and are used in the consequent matching procedure. In N-CSS method we apply the same matching algorithm as in CSS method, in order to compare two sets of CSS maxima and calculate their matching cost. However, the matching procedure is indifferent to the required best shift hypothesis, since starting point normalization has already been applied. That constitutes our method's main advantage and indicative results are shown in Fig. 3, where affine normalization seems to be capable of a more precise matching.

As shown in Fig. 3(e) and Fig. 3(f), the normalization applied in N-CSS method places the two CSS images in similar position with respect to their starting point, so that no additional shift is required. This improvement seems to be quite valuable whereas the number of the concave parts of the boundary curve increases. In such cases more hypotheses for the required best shift need to be considered, since there are respectively more contour maxima that can perform the best shift match. Therefore, N-CSS normalization fully succeeds in minimizing error possibility included in both affine length normalization and matching procedure and, as a result, effectively improves the performance of CSS method.

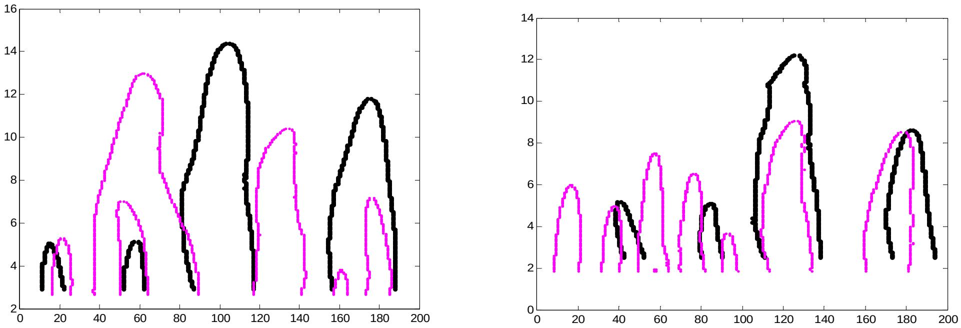


Figure 5: CSS images of different shapes of same shape category (a) in CSS method (b) in N-CSS method

4. EXPERIMENTAL RESULTS

In this section, we examine the performance of our proposed shape representation method in comparison with the CSS method and the results of our experiments are presented and discussed. In order to achieve that, we have created a database of real object's contours categorized by shape, such as cars, planes, drinks etc. Our experimental procedure consists of three different types of experiments involving general affine transformations. More specifically, the first experiment deals with affine transformation of the same boundary curve of the same shape category. Therefore, we have created a number of random affine transformations for each boundary curve in the database and calculated the matching cost of the original and affine transformed curve, using both CSS and N-CSS method.

For the whole experimental procedure, it is assumed that the contour shape of an object is available and represented by a set of ordered points forming a 2-D planar and closed curve. This set of sampled points is obtained from image data by means of manual or automatic segmentation.

As already mentioned, our proposed method seems to be quite efficient in case of matching affine transformed curves of the same shape, where affine invariant normalization contributes to effective curve matching. Equivalently, we continue by evaluating the method's performance in case of affine transformations of different shapes but in the same shape category. For example, we match two different types of glasses and their random affine transforms (Fig. 4). In Fig. 5 we can observe two indicative CSS images for both CSS and N-CSS method. Comparing the two CSS images (Fig. 5(a) and (b)) we come to the conclusion that N-CSS normalization also provides starting point affine invariance and therefore more accurate matching. More specifically, the CSS images approximately match the size of the corresponding contours, placing each contour in proportional position.

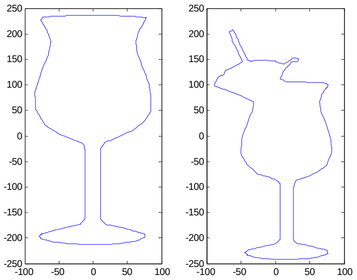


Figure 4: Contours of different shapes of the same shape category

The resulting CSS images do not match in every contour shape as in Fig. 3, which is expected due to the fact that, even in the same shape category, the boundary shape slightly alters.

Finally, we examine the method's performance in case of affine transformations of different shapes of different shape category, including their random affine transforms. The results of these experiments have also improved CSS image representation as well as its following matching procedure. However, since the shape of the examined boundary curves completely alters, matching cost value remains respectively high.

To verify the above theory, we conducted a number of experiments including a variety of curves from different shape categories. Curve matching for each curve and its affine transformations was applied, resulting to a matching cost value. More specifically, our database consists of 5 shape categories (cars, fish, glasses, hammers, planes).

Firstly, we apply 100 random affine transformations to each curve and calculate the matching cost between the original (non-affine) and its affine transformed curves (see Fig.3). The average matching cost values are depicted on Table 1. Observing the matching cost values, it becomes clear that affine curve normalization used in N-CSS method, leads to a more efficient curve matching, regarding random affine transforms of the same curve.

Methods	Shape category				
	cars	fish	glasses	hammers	planes
CSS	46,73	56,17	42,10	60,81	63,36
N-CSS	8,91	14,48	7,84	6,87	18,67

Table 1: Average matching cost values for the same curve of the same shape category estimated by CSS and N-CSS method

In fact, the matching cost values in case of N-CSS method seem to approach the desired minimum cost. The results also indicate the success rate of affine invariant normalization used in our proposed N-CSS method.

Next, we apply a random affine transformation to each curve of the same shape category (Fig.4), and calculate the matching cost between the affine transformed curves (Fig.5). The average matching cost values are depicted on Table 2.

Methods	Shape category				
	cars	fish	glasses	hammers	planes
CSS	40,37	58,67	54,49	52,63	60,70
N-CSS	30,99	54,12	46,36	38,72	47,91

Table 2: Average matching cost values for different curves of the same shape category estimated by CSS and N-CSS method

The above experimental results show that our proposed method (N-CSS) improves the performance of the existing method (CSS), but not in an extended rate as in the previous experiment. Such outcome is normally expected, since the normalization of N-CSS method provides affine invariance but is depended on the characteristics of each curve. So, as the shape of the curve alters, the matching cost of N-CSS method evenly increases. Therefore, N-CSS method has similar performance in case of different curves of different shape category, as shown in Table 3 and 4.

CSS method	cars	fish	glasses
cars	40,37	52,77	57,15
fish	52,77	58,67	60,73
glasses	57,15	60,73	54,48

Table 3: Average matching cost values for different curves of different shape category estimated by CSS method

N-CSS method	cars	fish	glasses
cars	30,99	52,21	36,49
fish	52,21	54,12	46,30
glasses	36,49	46,30	46,36

Table 4: Average matching cost values for different curves of different shape category estimated N-CSS method

To sum up, average matching cost results of both CSS and proposed N-CSS method are shown in Table 5.

Curve contours	Matching cost	
	CSS method	**N-CSS method**
Same shape/same category	53,83	11,354
Different shape/same category	53,37	43,62
Different shape/different category	54,98	44,61

Table 5: Average matching cost values estimated by CSS and N-CSS method

Observing the resulting matching costs, we come to the conclusion that our proposed method can lead to an improved shape representation and consequently to a more reliable curve matching. Moreover, curve normalization prior to curve matching seems to fully compensate possible insufficiency of affine length parametrization and matching algorithm. The resulting matching values can be considered not only as a measure of similarity between curves but also as a distinctive measure to classify curves into different shape categories, whereas the existing method proves to be inadequate.

Furthermore, depending on the average matching cost values, we can establish criteria for shape similarity retrieval and even more for successful image classification.

5. CONCLUSION – FURTHER WORK

Affine invariant curve matching using curvature scale-space and normalization has proved to be quite efficient in the problem of shape similarity retrieval in an affine transformed environment.

In this paper, the improvement is achieved through affine invariant curve normalization, applied as a preprocessing stage to CSS image construction and matching procedure. As a result, our proposed method is able to ensure affine invariance with respect to both parameter and coordinate affine transformations, while retaining all the local characteristics of the curve. The matching algorithm which compares two sets of representations and assigns a matching value as a measure of similarity has also proved to be simple and fast. In addition, every contour of the CSS image corresponds to a concavity or a convexity of the curve, providing important information regarding the shape of the image.

The proposed affine invariant curve matching method can be submitted to further enhancement and possible improvement by re-evaluating its individual stages. A first approach could deal with the improvement of the matching algorithm, where alternative hypotheses for the best match between maxima can be applied. Furthermore, one could re-examine convolution based on the Gaussian function, proposing alternative curve smoothing methods. Finally, the CSS maxima extraction could be handled by alternative algorithms, aiming at a more accurate detection of the maxima points of the curve.

6. REFERENCES

[1] F. Mokhtarian and S. Abbasi, “Affine curvature scale space with affine length parametrization,” Pattern Analysis. Application, vol. I5, no. 2, pp 143–158, 2001.

[2] F. Mokhtarian and A. Sadegh, “Shape similarity retrieval under affine transforms,” Pattern Recognition, no. 35, pp. 31–41, 2002

[3] Y. Avrithis, Y. Xirouhakis and S. Kollias, “Affine-invariant curve normalization for object shape representation, classification, and retrieval”, Machine Vision and Applications (2001) 13: 80–94.

[4] A.C. Jalba, M. Wilkinson, J. Roedink, “Shape representation and recognition through morphological curvature scale spaces”, IEEE Trans. On Image Processing.

[5] M. R. Teague, “Image analysis via the general theory of moments,” J. Opt. Soc. Amer., vol. 70, pp. 920-930, Aug. 1980.

[6] Z. Huang and F. Cohen, “Affine-invariant B-SPLINE moment for curve matching,” in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1994, pp. 490–495.

[7] A. Zhao and J. Chen, “Affine curve moment invariants for shape recognition,” Pattern Recognition, vol. 6, no. 30, pp. 895–901, 1997.

[8] J. Flusser and T. Suk, “Pattern recognition by affine moment invariants,” Pattern Recognition, vol. 1, pp. 167–174, 1993.

[9] Persoon E, Fu KS (1986) “Shape discrimination using Fourier descriptors” IEEE Trans Pattern Anal Mach Intell 8: 388–397

[10] Pratt I (1996) “Shape representation using Fourier coefficients of the sinusoidal transform” Technical Report Series UMCS-96-7-1. University of Manchester, U.K.

[11] Z. Ping, Y. Sheng, S. Deschenes, and H. Arsenault, “Fourier-Mellin descriptor and interpolated feature space trajectories for 3-D object recognition”, Opt. Eng., vol. 39, no. 5, pp. 1260–1266, 2000.

[12] D. Zhang, G. Lu, “Shape-based image retrieval using generic Fourier descriptor”, Gippsland School of Computing, Monash University, Vic. 3842, Australia, July 2002

[13] R. Malladi and J. A. Sethian, “Level set and fast marching methods in image processing and computer vision,” in Proc. IEEE Int. Conf. Computer Vision, vol. 1, 1996, pp. 489–492.