

# Robustness of Shape Similarity Retrieval under Affine Transformation

Sadegh Abbasi and Farzin Mokhtarian

Centre for Vision, Speech and Signal Processing, University of Surrey

Guildford, Surrey, GU2 5XH, England

Tel: +44-1483-879842

Fax: +44-1483-876031

S.Abbasi@surrey.ac.uk

## Abstract

The application of Curvature Scale Space representation in shape similarity retrieval under affine transformation is addressed in this paper. The maxima of Curvature Scale Space (CSS) image have already been used to represent 2-D shapes in different applications. The representation has shown robustness under the similarity transformations. Scaling, orientation changes, translation and even noise can be easily handled by the representation and its associated matching algorithm. In this paper, we also consider *shear* and examine the performance of the representation under affine transformations. It is observed that the performance of the method is promising even under severe deformations caused by shear. The method is tested on a very large database of shapes and also evaluated objectively through a classified database. The performance of the method is compared with the performance of two well-known methods, namely Fourier descriptors and moment invariants. We also observe that global parameters such as eccentricity and circularity are no longer useful in an affine transform environment.

## 1 Introduction

Considerable amount of information exists in two dimensional boundaries of 3D objects which enables us to recognise objects without using further information. However, despite great effort, the problem of shape representation in computer vision is still a very difficult one. A shape is originally defined by  $x$  and  $y$  coordinates of its boundary points which are subject to transformation if the position of the camera with respect to the object changes. The *similarity transformation* includes translation, uniform scaling and orientation changes. For example, if the distance between camera and object changes, the size of the boundary contour also changes (uniform scaling).

If the camera is allowed to change its viewpoint with respect to the object, the resulting boundary of the object will be deformed. For example, a circle will be converted to an ellipse (see also Figure 2 for a more complicated example). The deformation can be approximated mathematically by affine transformation when in addition to similarity transformation, shapes are also subject to shear.

It is also believed that a 3D object might be represented by a number of standard views eg back, front, above, etc. The possibility of changes in camera viewpoint will lead to deformation on either of these views and as a result, affine transformation is inevitable in such applications.

A number of shape representations have been proposed to recognise shapes under affine transformation. Some of them are the extensions of well-known methods such as Fourier descriptors [3] and moment invariants [5, 14]. The methods are then tested on a small number of objects for the purpose of object recognition. In both methods, the basic idea is to use a parametrisation which is robust with respect to affine transformation. The arc length representation is not transformed linearly under shear and therefore is replaced by *affine length* [6]. The shortcomings of the affine length include the need for higher order derivatives which results in inaccuracy, and inefficiency as a result of com-

putation complexity. Moreover, it can be shown [9] that although the arc length is not preserved, it does not change dramatically under affine transformation.

Affine invariant scale space is reviewed in [12]. This curve evolution method is proven to have similar properties as curvature evolution [8], as well as being affine-invariant. However, an explicit shape representation has yet to be introduced based on the theory of affine invariant scale space. The prospective shape representation might be computationally complex as the definition of the affine curvature involves higher order derivatives.

We have already used the maxima of Curvature Scale Space (CSS) image to represent shapes of boundaries in similarity retrieval applications [11, 10]. The representation is proved to be robust under similarity transformation which include translation, scaling and changes in orientation. In this paper, we examine the robustness of the representation under general affine transformation which also includes *shear*. As a result of shear, the shape is deformed and therefore the resulting representation may change. We will show that the performance of the method is promising even in the case of severe deformations.

The following is the organisation of the remainder of this paper. In section 2, CSS image is introduced and the CSS matching is briefly explained. Section 3 is about the affine transformation and the way we create our large databases. In section 4, we show that the conventional global parameters are not useful and introduce a new global parameter which can be used to narrow down the range of search. In section 5, we evaluate the performance of the method in two different ways. The results are then compared to the results of other well-known methods in section 6. The concluding remarks are presented in section 7.

## 2 Curvature Scale Space image and the CSS matching

Consider a parametric vector equation for a curve:

$$\vec{r}(u) = (x(u), y(u))$$

where  $u$  is an arbitrary parameter. The formula for computing the curvature function can be expressed as:

$$\kappa(u) = \frac{\dot{x}(u)\ddot{y}(u) - \ddot{x}(u)\dot{y}(u)}{(\dot{x}^2(u) + \dot{y}^2(u))^{3/2}} . \quad (1)$$

If  $g(u, \sigma)$ , a 1-D Gaussian kernel of width  $\sigma$ , is convolved with each component of the curve, then  $X(u, \sigma)$  and  $Y(u, \sigma)$  represent the components of the resulting curve,  $\Gamma_\sigma$ :

$$X(u, \sigma) = x(u) * g(u, \sigma)$$

$$Y(u, \sigma) = y(u) * g(u, \sigma)$$

It can be shown [10] that the curvature of  $\Gamma_\sigma$  is given by:

$$\kappa(u, \sigma) = \frac{X_u(u, \sigma)Y_{uu}(u, \sigma) - X_{uu}(u, \sigma)Y_u(u, \sigma)}{(X_u(u, \sigma)^2 + Y_u(u, \sigma)^2)^{3/2}} \quad (2)$$

As  $\sigma$  increases, the shape of  $\Gamma_\sigma$  changes. This process of generating ordered sequences of curves is referred to as the evolution of  $\Gamma$  (Figure 1a). If we calculate the curvature zero crossings of  $\Gamma_\sigma$  during evolution, we can display the resulting points in  $(u, \sigma)$  plane, where  $u$  is the normalised arc length and  $\sigma$  is the width of Gaussian kernel. The result is called the *Curvature Scale Space image* of the shape. For every  $\sigma$  we have a certain curve  $\Gamma_\sigma$  which in turn, has some curvature zero crossing points. As  $\sigma$  increases,  $\Gamma_\sigma$  becomes smoother and the number of zero crossings decreases. When  $\sigma$  becomes sufficiently high,  $\Gamma_\sigma$  will be a convex curve with no curvature zero crossing, and we terminate the process of evolution.

In order to construct the CSS image of a curve, we first re-sample it by 200 equally distant points. These points are numbered from 1 to 200; which are the values of  $u$ . Each coordinate of these points are convolved with a Gaussian function. The width of the Gaussian,  $\sigma$ , is gradually increased and at each level the locations of curvature zero crossings are determined and registered. The result of this process is represented by a binary image called CSS image of the curve (see Figure 1). The number of columns of this image is equal to the number of equidistant samples of the

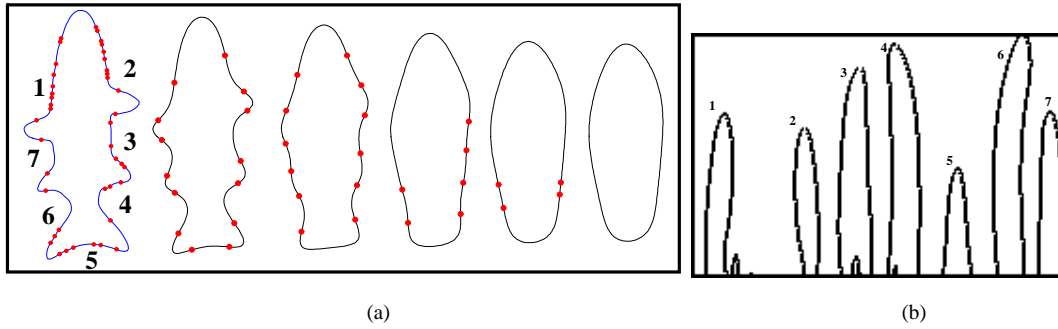


Figure 1: a) Shrinkage and smoothing of the curve and decreasing of the number of curvature zero crossings during the evolution, from left:  $\sigma = 1, 4, 7, 10, 12, 14$ . b) The CSS image of the shape.

curve. In fact, each column relates to a point on the resampled curve. Each row of the CSS image is related to a value of  $\sigma$ , starting from  $\sigma = 1.0$  for the bottom row and increased by 0.1 at each level. The intersection of every horizontal line with the contours in this image indicates the locations of curvature zero crossings on the corresponding evolved curve  $\Gamma_\sigma$ .

The CSS image contains a number of contours, each related to a segment of the shape. This is clearly shown in Figure 1. The curve is finally represented by locations of the maxima of its CSS image contours. For example, the shape in Figure 1 is represented by seven pairs of integer values corresponding to seven contours of the CSS image.

**Curvature Scale Space Matching** The algorithm used for comparing two sets of maxima, one from the input (also called *image*) and the other from one of the models, has been described in [10] and [11]. The algorithm first finds any possible changes in orientation which may have been occurred in one of the two shapes. A circular shift then is applied to one of the image maxima to compensate the effects of change in orientation. The summation of the Euclidean distances between the relevant pairs of maxima is then defined to be the matching value between the two CSS images.

The following is a condensed version of the algorithm which includes the basic concepts.

- Apply a circular shift to all image maxima so that the *u-coordinates* of the largest maxima, one from the image and the other from the model become identical.
- Starting from the second largest maximum of the image, determine the nearest maximum of the model to each maximum of the image.
- Consider the cost of the match as the summation of the Euclidean distances between the corresponding maxima.
- If the number of the image and the model maxima differs, add the  $\sigma$  coordinates of the unmatched maxima to the matching cost.

### 3 Pure shear transform and our databases

The general affine transformation can be represented mathematically with the following equation.

$$\begin{cases} x_a(t) = ax(t) + by(t) + e \\ y_a(t) = cx(t) + dy(t) + f \end{cases} \quad (3)$$

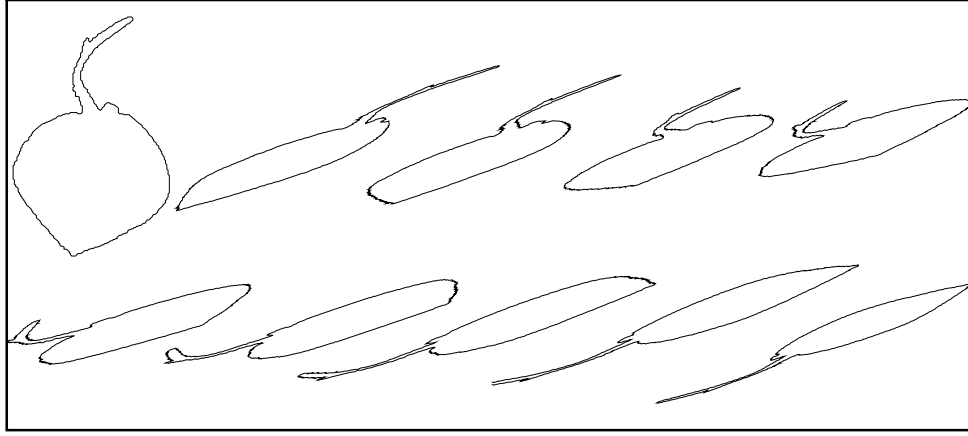


Figure 2: The deformation of shapes is considerable even with  $k = 1$  in shear transform. The original shape is presented in top left. Others represent transformation with  $k = 1$  and  $\theta = 20^\circ, 40^\circ, \dots, 160^\circ, 180^\circ$ .

where  $x_a(t)$  and  $y_a(t)$  represent the coordinates of the transformed shape. There are six degrees of freedom in this transformation. Translation and uniform scaling are represented by two degrees of freedom, while change in orientation needs just one parameter. The remaining parameter is related to *shear*. The general affine transformation contains all these transformations. However, it is possible to apply only one of them at a time. Scaling, change in orientation and shear are represented by the following matrices.

$$A_{scaling} = \begin{pmatrix} S_x & 0 \\ 0 & S_y \end{pmatrix} \quad A_{rotation} = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \quad A_{shear} = \begin{pmatrix} 1 & k \\ 0 & 1 \end{pmatrix}$$

If  $S_x$  is equal to  $S_y$ ,  $A_{scaling}$  represents a uniform scaling. A shape is not deformed under rotation, uniform scaling and translation. However, nonuniform scaling and shear contribute to the shape deformation under general affine transformation. In this paper, we examine the performance of the CSS representation under shear transform. The measure of shape deformation depends on the parameter  $k$ , *shear ratio*, in the matrix  $A_{shear}$ . In the present form of the matrix  $A_{shear}$ ,  $x$  axis is called *shear axis*, as the shape is pulled toward this direction.

Figure 2 shows the effects of affine transformation on shape deformation. In this Figure, shear ratio is selected as  $k = 1$ . In order to achieve different shear axes, we have changed the orientation of the original shape prior to applying the pure shear transformation. The values of  $\theta$  range from  $20^\circ$  to  $180^\circ$ , with  $20^\circ$  intervals. As this Figure shows, the deformation is severe for  $k = 1$ . For larger values of  $k$ , e.g. 1.5 and 2, the deformation is much more severe.

In order to create four different databases, we choose four different values for shear ratio, 0.5, 1.0, 1.5 and 2.0. We then apply the transformation on a database of 500 original object contours. From every original objects, we obtain 9 transformed shapes with different values of  $\theta$ . Therefore, each database consists of 500 original and 4500 transformed shapes. We then carry out a series of experiments on these databases to verify the robustness of the CSS image representation under affine transformations. The following sections are concerned with these experiments.

## 4 Global parameters

We have already used the CSS representation for shape similarity retrieval in similarity transformed environment [10][11]. Since the global shape of a closed curve is preserved under similarity transformation, we used two global parameters, namely eccentricity and circularity, to filter out dissimilar shapes prior to the CSS matching. As it is obvious from Figure 2, the global appearance of a shape is severely changed under shear. In this section we show that eccentricity and circularity can not be used as shape descriptors under shear. We introduce another global parameter which is extracted from the CSS image.

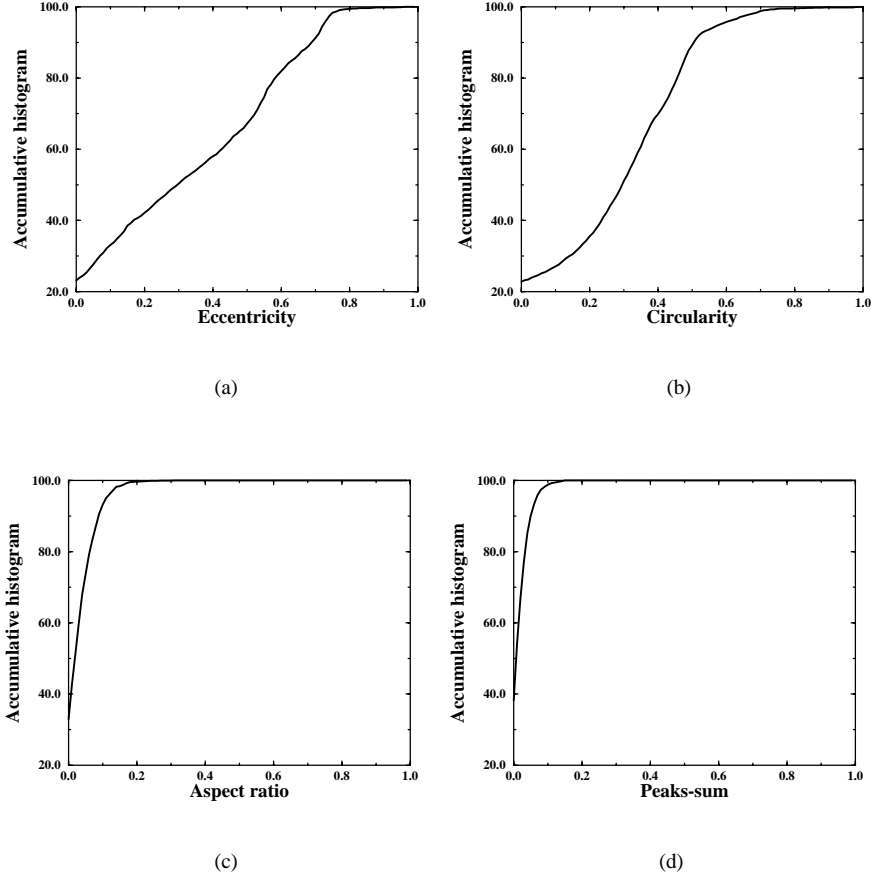


Figure 3: a) and b) The original shapes are not similar to their affine transformed versions with respect to eccentricity and circularity. c) and d) The CSS-based global parameters show better performance. (see section 4)

In order to reject dissimilar models prior to CSS matching, we used to compute  $\alpha_e$  and  $\alpha_c$ , and  $\alpha_a$  as follows [1].

$$\alpha_e = \frac{|e_i - e_m|}{\max(e_i, e_m)} \quad \alpha_c = \frac{|c_i - c_m|}{\max(c_i, c_m)} \quad \alpha_a = \frac{|a_i - a_m|}{\max(a_i, a_m)}$$

where  $e$  and  $c$  represent the eccentricity and circularity of the boundary and  $a$  represents the aspect ratio of the CSS image, while  $i$  and  $m$  stand for image and model respectively. According to their definition,  $\alpha_e$ ,  $\alpha_c$  and  $\alpha_a$  are between zero and one. Our experiences show that for globally similar shapes these values are not more than 0.4.

We considered each original shape of our database as the input shape and its transformed versions as the models. We then computed these parameters for all original images and their associated transformed shapes. For each parameter, we obtained 4500 values between zero and one. The accumulative histograms of different parameters for a shear ratio  $k = 1.5$  are presented in Figure 3.

As Figure 3a shows, eccentricity can not be used as a shape descriptor under affine transformation, as there is a large difference in this parameter between the original and transformed shapes. In more than 45% of cases,  $\alpha_e$  has been higher than 0.4. Though this figure is 30% for circularity, it is still much higher than it should be. The aspect ratio of the CSS image, however, remains in a reasonable range under affine transformation. Figure 3c shows that  $\alpha_a$  is always less than 0.2 when an original shape is used as the input and its transformed version as a model. We also

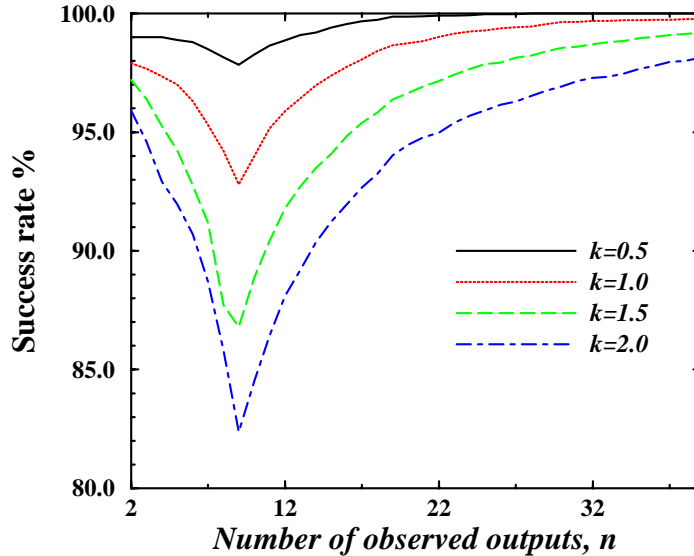


Figure 4: Identifying transformed versions of the input query.  $k$  is the shear ratio and represents the measure of deformation. (see section 5)

examined another global parameter called *peaks\_sum*, which is the summation of the heights of the CSS maxima. The same histogram for this parameter is presented in Figure 3d. Again, in all cases, the original and transformed shapes have almost the same *peaks\_sum*.

The main advantage of global parameters is that they avoid exhaustive comparison of the input query with every model of the database.

## 5 Evaluation

We examined the performance of the representation through two different experiments. The first one was performed on the databases of section 3. Every original shape was selected as the input query and the first  $n$  outputs of the system were observed to see if the transformed versions of the query are retrieved by the system.

We found out that for  $k = 0.5$ , almost all transformed versions of a shape appear in the first 20 outputs of the system. For  $k = 1$ , on average 99% of transformed shapes of the input query are among the first 20 shapes retrieved by the system as the similar shapes to the input query. This figure is 96% and 94% for  $k = 1.5$  and  $k = 2$  respectively. The results for different values of  $k$  and for different numbers of output images are presented in Figure 4. As this Figure shows, the first outputs of the system, always include a large portion of the transformed versions of the input query. These results show that the representation can be used in affine transformed environment. This fact is also verified by the following evaluation method.

### 5.1 Objective evaluation with classified database

We have already used the idea of using classified databases to evaluate a shape representation method in shape similarity retrieval [2]. A subset of our large database is usually used for this test. Here we have chosen 76 shapes and classified them in 10 different groups as presented in Figure 5. These are our original shapes and we produce 9 transformed shapes from each original one. As a result, a group with 8 members will have 80 members after adding the

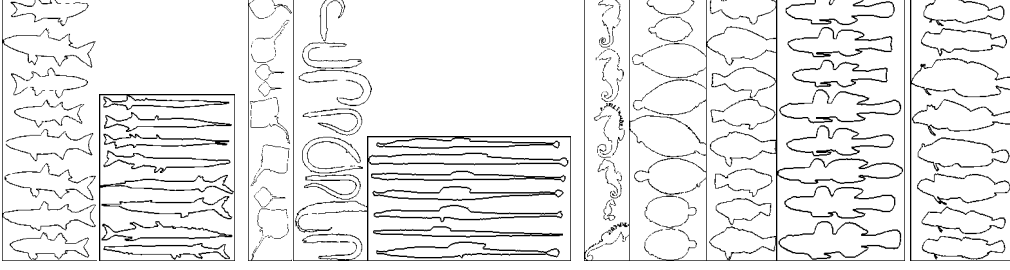


Figure 5: Classified database used for objective evaluation consists of 10 different classes. Note that for each original shape, nine transformed versions are also generated. Therefore, the actual size of the database is 10 times larger than the size of this database.

transformed shapes. The whole database then will have 760 shapes in 10 different groups.

In order to assign a performance measure to the method, we choose every member of each group as the input query and ask the system find the best  $n$  similar shapes to the input from the database. We then observe the number of outputs,  $m$ , which are from the same group as the input. The *success rate* of the system for this particular input is defined as follows.

$$\text{Success rate for an input query} = \frac{m}{m_{max}} \times 100$$

where  $m_{max}$  is the maximum possible value of  $m$ . Note that  $m_{max}$  is equal to  $n$  if  $n$  is less than the number of group members; if not,  $m_{max}$  will be equal to the number of group members. The success rate of the system for the whole database will be the average of the success rates for each input query. We chose different values for  $n$  and for each case, computed the success rate of the method. The results are presented in Figure 6a. As this Figure shows, for lower values of  $n$ , the success rate of the system is considerably large. For example, in the case of  $k = 1.5$  and  $n = 10$ , it is 98%. In other words, on average 9.8 out of the first 10 outputs of the system belong to the same group as the input. This is as a result of the fact that the system ranks the outputs based on their similarity measure to the input, and there is a strong possibility that the most similar models be in the same group as the input query. As  $n$  increases, the similarity between the bottom ranked outputs and the input query decreases and models from other groups may also appear as the output. At the same time,  $m_{max}$  is equal to  $n$  for  $n \leq 80$  and increases with  $n$ . For  $n \geq 80$ ,  $m_{max}$  does not increase anymore and is fixed at 80, however  $m$  always increases with  $n$  and therefore this part of each plot has a positive slope.

## 5.2 Effects of global parameters

In section 4 we argued that eccentricity and circularity are not appropriate in affine environment. However, the other two parameters, namely *aspect ratio* and *peaks\_sum* may be helpful. We carried out a series of experiments with different threshold values for each of these parameters and observed that although the improvement on the performance measure of the system was not significant (2 – 3%), the number of rejected candidates was considerably large (more than 70% of the database population). As a result, using global parameters increases the speed of the system significantly.

## 6 Comparison with other methods

Fourier descriptors [13] and Moment invariants [4, 7] have been widely used as shape descriptors in similarity transform environment. Both methods represent the global appearance of the shape in their most important components. For example, the largest magnitude component of Fourier descriptors represents the dimensions of the best fitted

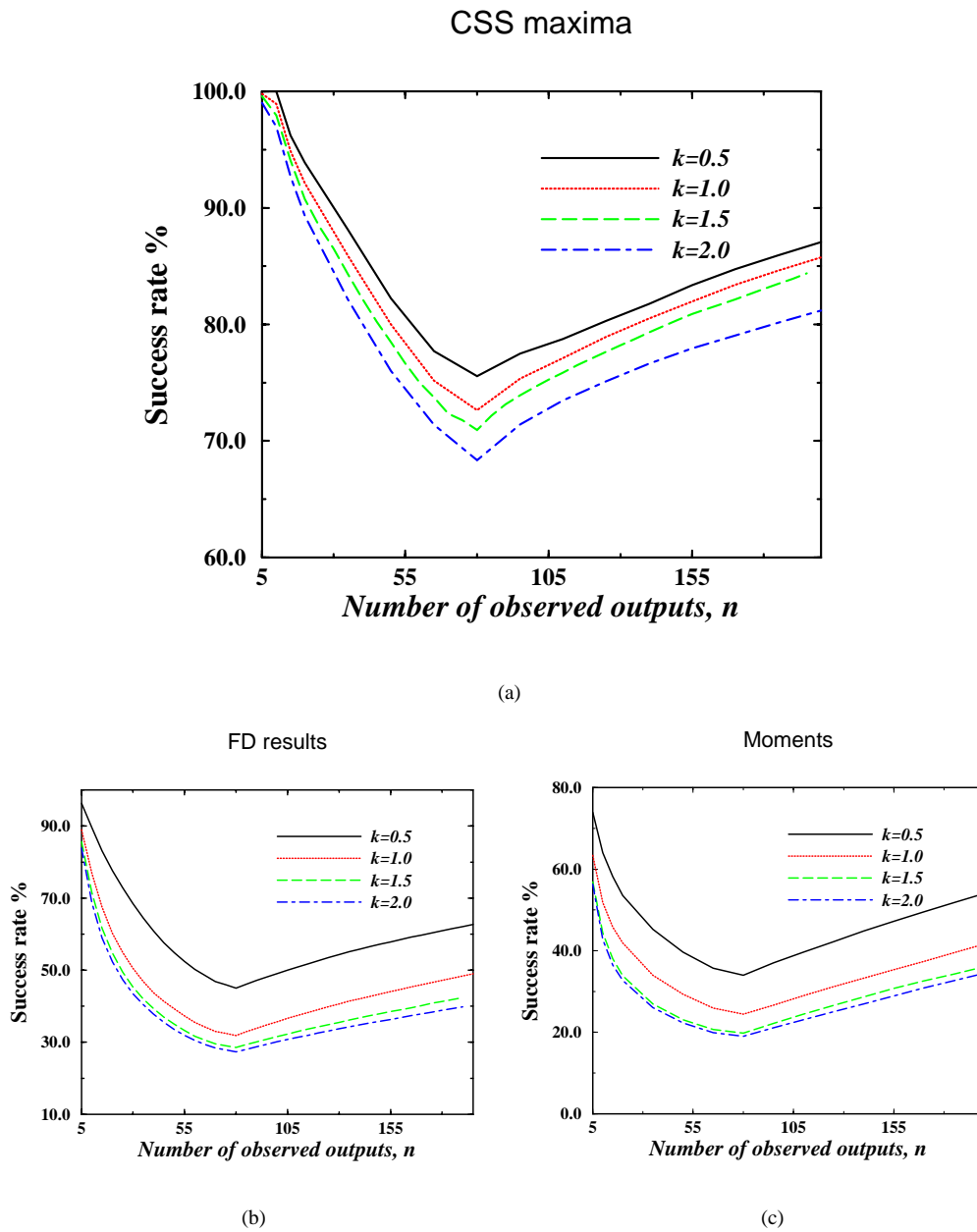


Figure 6: The results of objective evaluation for different approaches. All terms are defined in section 5.1

ellipse of the shape. Since affine transformation changes the global appearance of a shape, it is expected that the performance of these methods is negatively affected under the transformation.

The modified versions of these methods have been introduced to deal with affine transformation [3, 5, 14]. They are used in object recognition application and clustering a small number of shapes. However, it has been shown that the improvements in modified versions are not very significant in comparison to the conventional versions. Considering the fact that the implementation of the modified versions is not a straight forward task, we decided to examine the conventional versions of these methods and compare the results with the results of our method. We will observe that the difference between the performance measure of our method and the performance measure of each of these methods is very large. Even if we consider 10 – 15% improvement for the modified versions of the methods, their performance are still well behind the performance of the CSS representation.

We implemented the method described in [13] to represent a shape with its normalised Fourier descriptors. Every original shape and its transformed was presented by their first 20 components. The Euclidean distance was used to measure the similarity between the two representations.

The results for different values of  $k$  are presented in Figure 6b. The minimum of the performance measure for different values of  $k$  is around 30%, comparing to 70% of the CSS method. At the same time, the slope of the plots after the minimum points is not sharp which means that most of the missing models are not ranked even among the first 200 outputs of the system.

For moment invariant, each object is represented by a 12 dimensional feature vector, including two sets of normalised moment invariants [4], one from object boundary and the other from solid silhouette. The Euclidean distance is used to measure the similarity between different shapes. The results are presented in Figure 6c. Comparing this plot with Figure 6a, a large difference between the performance of the CSS representation and this method is observed.

## 7 Conclusion

In this paper we demonstrated that the maxima of the Curvature Scale Space image can be used for shape representation in similarity retrieval applications even under affine transformation. We observed that the method can identify the transformed versions of input query among a large number of database models. The results of evaluation of the method on a classified database of original and transformed shapes showed that the performance of the method is promising even under severe deformation of the shapes as a result of affine transformation. The method was compared with two well-known methods, namely Fourier descriptors and moment invariants. The results showed the superiority of our method over these methods. We also showed that global parameters such as eccentricity and circularity can not be used in an affine transformed environment. Consequently, we introduced another CSS-based global parameter and used it to filter out a large fraction of dissimilar shapes prior to CSS matching.

**Acknowledgements** Sadegh Abbasi is on leave from the University of Guilan, Rasht, Iran. He is grateful to the Ministry of Culture and Higher Education of Iran for its financial support during his research studies.

## References

- [1] Abbasi S, Mokhtarian F, Kittler J. Reliable Classification of Chrysanthemum Leaves through Curvature Scale Space. In: Proceedings of the Scale-Space'97 Conference, Utrecht, 1997, pp284-295
- [2] Abbasi S, Mokhtarian F, Kittler J. Shape similarity retrieval using a height adjusted curvature scale space image. In: Proceedings of 2nd International Conference on Visual Information Systems, San Diego, 1997, pp173-180
- [3] Arbter K, Snyder WE, Burkhardt H, Hirsinger G. Applications of affine-invariant Fourier descriptors to recognition of 3-D objects. *IEEE Trans. Pattern Analysis and Machine Intelligence* 1990; 12:640-646
- [4] Dudani S, Breeding K, Mcghee RB. Aircraft identification by moment invariants. *IEEE Transactions on Computers* 1977; C-26:39-45
- [5] Flusser J. Pattern recognition by affine moment invariants. *Pattern Recognition* 1993; 26:167-174

- [6] Guggenheimer HW. Differential Geometry. MCGraw-Hill, New York, 1963
- [7] Hu MK. Visual pattern recognition by moments invariants. IRE Trans. Information Theory. 1962; IT-8:179-187
- [8] Kimia BB, Siddiqi K. Geometric heat equation and nonlinear diffusion of shapes and images. Computer Vision and Image Understanding 1996; 64:305-332
- [9] Mokhtarian F, Abbasi S. Shape similarity retrieval under affine transforms through curvature scale space. Submitted to IEEE International Conference on Multimedia Computing and Systems, Florence, 1999
- [10] Mokhtarian F, Abbasi S, Kittler J. Robust and efficient shape indexing through curvature scale space. In: Proceedings of the sixth British Machine Vision Conference, BMVC'96, Edinburgh, 1996, pp53-62
- [11] Mokhtarian F, Abbasi S, Kittler J. Efficient and robust retrieval by shape content through curvature scale space. In: Proceedings of the First International Workshop on Image Database and Multimedia Search, Amsterdam, 1996, pp35-42
- [12] Sapiro G, Tannenbaum A. Affine invariant scale space. International Journal of Computer Vision 1993; 11:25-44
- [13] Wallace TP, Wintz P. An efficient three-dimensional aircraft recognition algorithm using normalised Fourier Descriptors. Computer Graphics and Image Processing 1980; 13:99-126
- [14] Zhao A, Chen J. Affine curve moment invariants for shape recognition. Pattern Recognition 1997; 30:895-901