Classification of Shapes via Explainable AI

Vincent Granville, Ph.D.
vincentg@MLTechniques.com
www.MLTechniques.com
Version 1.0, April 2022

Abstract

I define the mathematical concept of shape and shape signature in two dimensions, using parametric polar equations. The signature uniquely characterizes the shape, up to a translation or scale factor. In practical applications, the data set consists of points or pixels located on the shape, rather than the curve itself. If these points are not properly sampled - if they are not uniformly distributed on the curve - they need to be re-weighted to compute a meaningful centroid of the shape, and to perform shape comparisons. I discuss the weights, and then introduce metrics to compare shapes (observed as sets of points or pixels in an image). These metrics are related to the Hausdorff distance. I also introduce a correlation distance between two shapes. Equipped with these metrics, one can perform shape recognition or classification using training sets of arbitrary sizes. I use synthetic data in the applications. It allows you to see how the classifier performs, to discriminate between two very similar shapes, or in the presence of noise. Rotation-invariant metrics are also discussed.

1 Introduction

A central problem in computer vision is to compare shapes and assess how similar they are. This is used for instance in text recognition. Modern techniques involve neural networks. Here, I revisit a methodology developed before computer even existed. With modern technology, it leads to an efficient, automated AI algorithm. The benefit is that the decision process made by this black-box system, can be easily explained, and thus easily controlled.

To the contrary, neural networks use millions of weights that are impossible to interpret, potentially leading to over-fitting. Why they work very well on some data and no so well on other data is a mystery. My "old-fashioned" classifier, adapted to modern data and computer architectures, lead to full control of the parameters. In other words, you know beforehand how fine-tuning the parameters will impact the output. Thus the word explainable AI [Wiki].

In an ideal world, one would want to blend both methods, to benefit from their respective strengths, and minimize their respective drawbacks. Such blending is referred to as ensemble methods [Wiki]. Also, since we are dealing with sampled points located on a curve (the "shape"), the same methodology also applies to sound recognition.

2 Mathematical foundations

In this section, we are concerned with the mathematical concept of shape. Later on, I apply the methodology to shapes represented by sets of points or pixels, observed through a rectangular window – a digital image. The center of the window is the origin of the coordinate system. Shapes can be anything: they may represent a letter, an hieroglyph, or a combination of symbols. They may consist of multiple, non-connected curves. We are only interested in the contour that defines the shape. It may or may not correspond to the boundary of a domain; the contour may not be closed and could consist of disconnected segments.

Furthermore there is no color or gray scale involved. The mathematical shape model can be viewed as black on a white background, with no thickness. In practical applications, the rectangular image is centered around the shape, and the noise has been filtered out. See example in Figure 1, comparing two shapes.

It is convenient, for illustrations purposes, to define a 2-D mathematical shape using a parametric polar equation, as follows:

$$r_t = g(t), \quad \theta_t = h(t), \quad \text{with} \quad t \in T, \quad r_t \geq 0, \quad 0 \leq \theta_t \leq 2\pi.$$

Here g, h are real-valued functions, and T is the index domain. An example with $n = 20$ points is as follows:

$$\theta_t = (t + \eta) \bmod 2\pi, \quad r_t = c + d\sin(at)\sin(2\pi b - bt), \quad t = 2\pi k/n \text{ with } k = 0, \ldots, n-1. \tag{1}$$

This example is pictured in Figure 1. The parameter η controls the rotation angle or orientation of the shape. By definition, $\alpha \bmod \beta = \alpha - \beta\lfloor \alpha/\beta \rfloor$ where the brackets represent the integer part function. A more simple

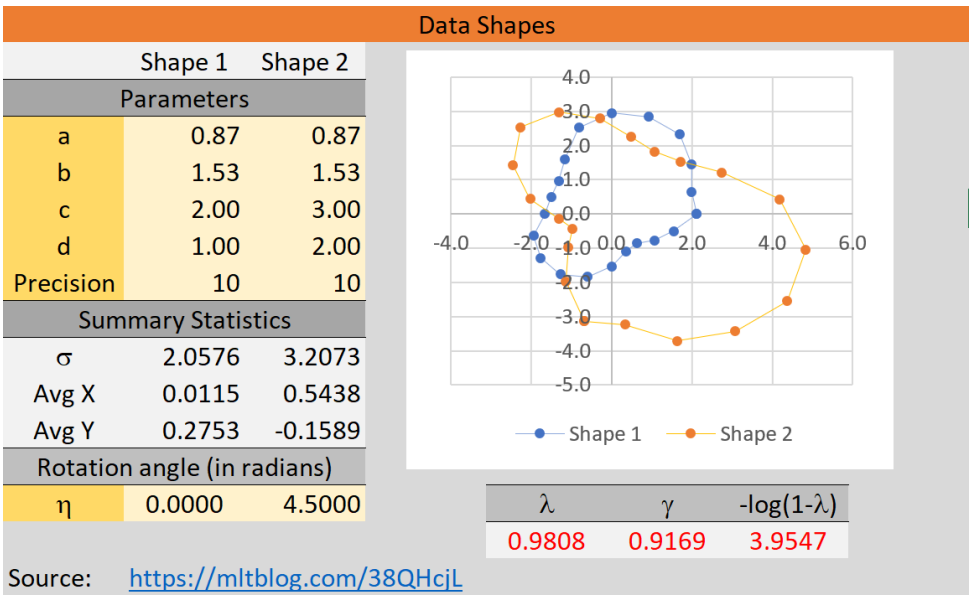

Figure 1: Comparing two shapes

example, corresponding to an elliptic arc, is

$$r_t = \frac{p}{1 - \epsilon \cos t}, \quad \theta_t = t, \quad 0 \leq t \leq t_0$$

where $0 < \epsilon < 1$, $p > 0$ and $0 < t_0 < 2\pi$ are the parameters. The parameter ϵ is the eccentricity. Since the functions $g(t)$ and $f(t)$ are arbitrary, can be discontinuous, and may contain infinitely many parameters (for instance, the coefficient of a Taylor or Fourier series), it covers all the possible shapes that exist in the universe.

3 Shape signature

The concept of shape signature is not new, see [1, 4]. Each shape (or set of points) is uniquely described by a normalized set called signature. In our context, this set can be a curve, a set of points, multiple broken curves, or a combination of these elements. The signature does not depend on the location or center of gravity of the shape. It depends on the orientation, though it is easy to generalize the definition to make it rotation-invariant, or to handle 3D shapes. The first step is to use the center of gravity (centroid) for the origin, and then rescale the shape by standardizing the variance of the radius r_t.

The centroid is the weighted average of the points located on the shape. Typically, the weight is constant. However, if the points are not uniformly distributed on the shape, you may use appropriate weights to correct for this artifact. This is illustrated in Figure 2. I now dive into the details of the reweighting procedure.

3.1 Weighted centroid

Let (x_t, y_t) be the standard coordinates of the observed points on the shape. In other words, $x_t = r_t \cos\theta_t$ and $y_t = r_t \sin\theta_t$. The centroid is defined as (G_x, G_y) with

$$G_x = \frac{1}{\mu}\int_T w_t x_t dt, \quad G_y = \frac{1}{\mu}\int_T w_t y_t dt, \quad \text{with } \mu = \int_T w_t dt. \tag{2}$$

Here $w_t > 0$ is the weight function, with $t \in T$. If t is discrete (for instance, the shape consists of observed data points), then the integrals are replaced by sums.

In most cases, the points are not evenly distributed on the curve. On a real data set, it translates by a curve that appears darker or thicker in locations with high point density: see Figure 2. If this is not a desirable feature, it can be eliminated by proper reweighting. To get the points evenly distributed on the curve, when computing the centroid, proceed as follows. Using notations from infinitesimal calculus, you want Δs_t, the length of an infinitesimal curve segment encompassing (x_t, y_t), to be proportional to w_t. Since the proportion factor does not matter, we must have

$$\Delta s_t = \sqrt{(\Delta x_t)^2 + (\Delta y_t)^2} = w_t \Delta t.$$

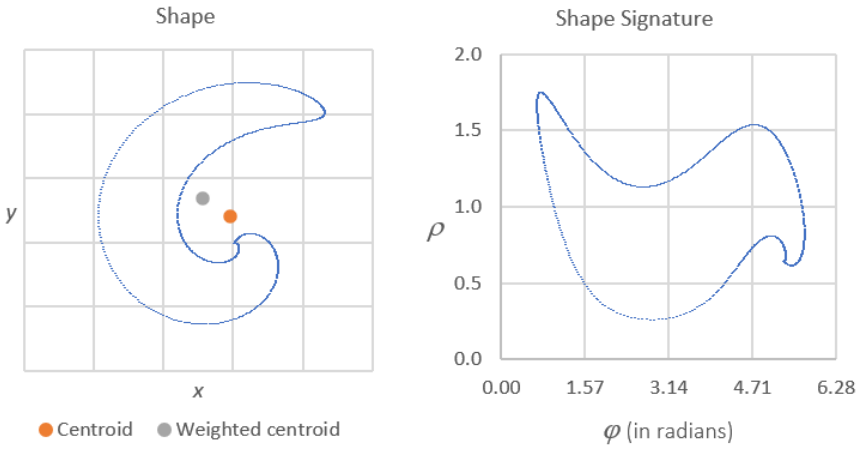

Figure 2: Weighted centroid, shape signature

This leads to

$$w_t = \sqrt{\left(\frac{dx_t}{dt}\right)^2 + \left(\frac{dy_t}{dt}\right)^2}.$$

It can be re-written using polar coordinates as

$$w_t = \sqrt{\left(\frac{dr_t}{dt}\right)^2 + r_t^2\left(\frac{d\theta_t}{dt}\right)^2}.$$

The formula assumes differentiability of the functions involved. In many cases, there are points (values of t) where the functions are either left- or right-differentiable [Wiki], but not both. Use the left or right derivative for these points.

3.2 Computing the signature

We want a mathematical object, easy to compute, that uniquely characterizes a shape, up to a translation vector and scaling factor. The set of all polar coordinates (r_t, θ_t) with $t \in T$, uniquely characterizes the shape. But it is not scale or translation invariant. To fix this problem, you first need to change the coordinate system to make the centroid (G_x, G_y) defined by formula (2), the origin. You may use the weight function w_t discussed in section 3.1. Then, you need to rescale by a factor σ. Eventually, the new coordinates are

$$u_t = \sigma^{-1} \cdot (x_t - G_x) = \rho_t \cos\varphi_t,$$
$$v_t = \sigma^{-1} \cdot (y_y - G_y) = \rho_t \sin\varphi_t.$$

Here u_t, v_t are the new Cartesian coordinates replacing x_t, y_t, and ρ_t, φ_t the new polar coordinates replacing r_t, θ_t. For reasons that will become obvious when comparing two shapes in section 4, the scaling factor is chosen as follows:

$$\sigma = \sqrt{\int_T (x_t - G_x)^2 + (y_t - G_y)^2 dt}.$$

It follows immediately that

$$\rho_t = \sigma^{-1} \cdot \sqrt{(x_t - G_x)^2 + (y_t - G_y)^2}, \quad \text{and} \quad \int_T \rho_t^2 dt = 1. \tag{3}$$

Now the signature is defined as the set of all (ρ_t, φ_t) with $t \in T$. By construction, $0 \leq \varphi_t \leq 2\pi$. When plotting the signature, to keep it bounded on $[0, 2\pi] \times [0, 1]$ regardless of the shape, one can use $\rho_t/(1+\rho_t)$ instead of ρ_t on the vertical axis. An example of signature is shown in Figure 2 (right plot).

3.3 Example

The shape illustrated in Figures 2 and 3 is different from that defined by (1). This time, it is defined by the parametric polar equation

$$\theta_t = (2\pi + 2\pi \sin(ct) + \eta) \bmod 2\pi, \quad r_t = t^a(1-t)^b, \quad t \in T = [0, 1]. \tag{4}$$

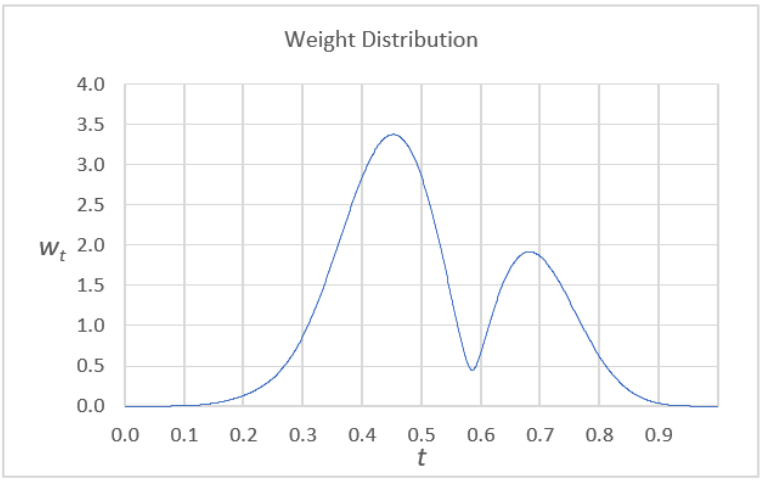

Figure 3: Weight function used in Figure 2

Again, η is the angle determining the orientation of the shape. The point density, visible to the naked eye, is much higher on the right side of the shape on the left plot in Figure 2. This is even more pronounced on the lower part. As a result, the centroid (orange dot) gets attracted to the dense area of the curve. Once this effect is corrected by the weight function, the new centroid (gray dot) now appears well "centered". Note that the weight function w_t, pictured in Figure 3, is bimodal. It was chosen to integrate to one, thus it represents a probability distribution on $T = [0, 1]$.

4 Shape Comparison

I start with a correlation metric based on the mathematical theory developed so far. I then discuss its strengths and weaknesses, and how to improve it. A full implementation on a real data set is investigated in section 5. I use the notation ρ_t, φ_t for the first shape, and ρ'_t, φ'_t for the second one. Assuming the parametric polar equations are such that (possibly after an appropriate transformation) $\varphi_t = \varphi'_t$ for $t \in T$, then define

$$\gamma = \int_T (\rho_t - \rho'_t)^2 dt = \int_T \rho_t^2 dt + \int \rho_t'^2 dt - 2\int_T \rho_t \rho'_t dt = 2 - 2\lambda,$$

where

$$\lambda = \int_T \rho_t \rho'_t dt.$$

It follows from (3) that $0 \leq \lambda \leq 1$. Furthermore, the two shapes are identical (up to a scaling factor and translation vector) if and only if $\lambda = 1$. The correlation λ measures how close the two shapes are from each other. It relies on the fact that $\varphi_t = \varphi'_t$. It this assumption is mildly violated, the classifier may still work on simple data sets, for instance to recognize the letters of the alphabet. But it may fail is the discrepancy between φ_t and φ'_t is significant.

It is not always possible to satisfy $\varphi_t = \varphi'_t$ for complicated shapes consisting of multiple arcs. But it can always be done for closed, convex shapes. Also, if the shapes are identical but rotated, usually $\lambda \neq 1$. The coefficient λ depends on the orientation angles η, η' of each shape, illustrated in formula (1). For this reason, λ is also denoted as $\lambda(\eta, \eta')$.

To circumvent this problem, one can use

$$\lambda^* = \min_{\eta,\eta'} \lambda(\eta, \eta').$$

Then the two shapes are identical, up to the scaling factors, translation vectors, and orientations, if and only if $\lambda^* = 1$. Due to symmetry, one can set $\eta = 0$. In practice, the metric $-\log(1 - \lambda)$ or $-\log(1 - \lambda^*)$ is used instead. See [5] for a general reference on shape correlation.

Of course, it is always possible to compare two shapes by comparing their signatures, see [3]. One way to do it is as follows. For each point P_t on the first shape signature, find its closest neighbor Q_t on the second shape signature, and compute the distance D_t between these two points. Then compute

$$D = \int_T D_t dt.$$

Repeat the operation by swapping the roles of the first and second shape: For each point Q'_t on the second shape signature, find its closest neighbor P'_t on the first shape signature, and compute the distance D'_t between these two points. Then compute

$$D' = \int_T D'_t dt.$$

If $D = 0$, the first shape is a subset of the second shape. If $D' = 0$, the second shape is a subset of the first shape. If $D = D' = 0$, the shapes are identical. Thus $\delta = \min(D, D')$ is a metric measuring shape similarity. It is closely related to the Hausdorff distance [Wiki], albeit less sensitive to outliers.

4.1 Shape classification

Now that we have a metric to compare two shapes, we can use it as a similarity measure to perform shape classification. If shapes S_1 and S_2 have $-\log(1-\lambda) > \alpha$, we may write $S_1 \sim S_2$ to mean that they are equivalent (very similar). In this case, $\alpha = 8$ is a good threshold. In character recognition, if you have a training set with thousands of hieroglyphs, you can use this technique to classify any new hieroglyph as equivalent to one or more in your training set, or as yet uncategorized (a brand new one, or one so poorly written that it is not recognizable).

5 Application

In all the cases investigated, including the mathematical ones, the computations were performed using sample points on the shape, corresponding to evenly spaced values of t. I used sums rather than integrals, and derivatives such as dx_t/dt were replaced by differences between successive values of x_t. You can find the computations in my spreadsheet `Shapes4.xlsx`, located on my GitHub repository, here.

Spreadsheet and data

The two main examples are:

- 20 points for the case pictured in Figure 1. Here I tested 8 pairs of shapes; you can find the summary in the animated Gif, posted here. In addition, I introduced various levels of noise to test the discriminating power of the classifier. The amount of noise is controlled by the parameter `Precision`.
- 1,000 points for the case pictured in Figure 2. Details are in the `Shape_Signature` tab in the spreadsheet. In the same tab, you will find the computation of the weight function, the weighted centroid, and the computations related to the new coordinate system ρ_t, φ_t.

My simulations rely on synthetic data [Wiki]. In other words, I use mathematically-generated shapes. The benefit is that you can generate a large class of shapes (actually, infinitely many), mimicking any existing shape, and compare the performance of various shape classifiers. In particular, you can assess how well a specific metric can detect different yet very similar shapes, or how it performs when various levels of noise are introduced. Modern methods combine real observations with synthetic data to further enrich training sets. This is known as augmented data [Wiki].

Several other machine learning techniques, tested on synthetic data and accompanied by professional summary spreadsheets, are available (along with the data sets and source code), in my book 'Stochastic Processes and Simulations" [2].

6 Exercises

The first exercise has an elegant solution. The second one is an application of the principles discussed.

Exercise 1 Find the weight function satisfying $\Delta(w_t x_t) = \Delta(w_t y_t)$. This is related to the material presented in section 3.1. It uses the same notations.

Solution

You need to find w_t satisfying $x_t \Delta w_t + w_t \Delta x_t = y_t \Delta w_t + w_t \Delta y_t$. Dividing by Δt, and letting $\Delta t \to 0$, we get

$$x_t \frac{dw_t}{dt} + w_t \frac{dx_t}{dt} = y_t \frac{dw_t}{dt} + w_t \frac{dy_t}{dt},$$

that is,

$$(x_t - y_t)\frac{dw_t}{dt} = \Big(\frac{dy_t}{dt} - \frac{dx_t}{dt}\Big) w_t.$$

This is successively equivalent to

$$\frac{d}{dt}(\log w_t) = \frac{1}{w_t}\frac{dw_t}{dt} = \frac{1}{x_t - y_t}\Big(\frac{dy_t}{dt} - \frac{dx_t}{dt}\Big),$$
$$\log w_t = \int \frac{1}{x_t - y_t}\Big(\frac{dy_t}{dt} - \frac{dx_t}{dt}\Big)dt + C,$$
$$w_t = C' \exp\Big[-\int_0^t \frac{1}{y_\tau - x_\tau}\Big(\frac{dy_\tau}{d\tau} - \frac{dx_\tau}{d\tau}\Big)d\tau\Big],$$

where C, C' are constants, with $C' > 0$. The value of C' is unimportant when using formula (2). However, you can choose it so that the weight function integrates to one. Or you can use $C' = 1$. I assumed, without loss of generality, that the integration domain T is an interval containing the origin $\tau = 0$.

You can test this formula on a line segment, defined by $x_t = t, y_t = a + bt$, with $t \in [0, 1]$. This is the most basic shape other than a finite set of points! A slightly more difficult exercise is to find the weight function satisfying $|\Delta(w_t x_t)| = |\Delta(w_t y_t)|$. Exercise 1 is a starting point to solve this problem.

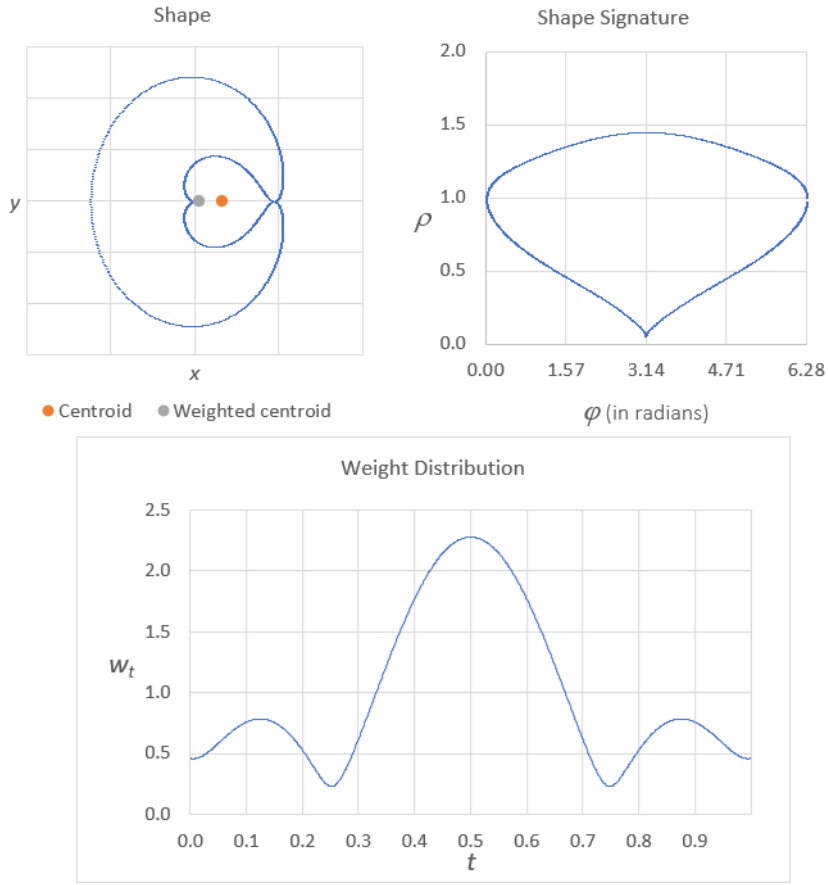

Figure 4: Another interesting shape

Exercise 2 Compare the shape in Figure 2 with that of Figure 4, using the metrics presented in this article (D and λ). These shapes correspond to equation (4). The first one has parameters $a = 7, b = 6, c = 8$. The second one has parameters $a = b = 1, c = 2\pi$. In both cases, $\eta = 0$. Use 1,000 sample points on each shape for comparison purposes. Order the points according to t, then according to φ_t, to see the impact on λ. Set $\eta = 0$, and choose η' (the orientation of the second shape) to maximize the similarity between the two shapes.

Solution

A starting point is my spreadsheet `Shapes4.xlsx`, available on my GitHub repository, here. This type of shape is analyzed in the `Shape_Signature` tab.

References

[1] Stamatia Giannarou and Tania Stathaki. Shape signature matching for object identification invariant to image transformations and occlusion. 2007. ResearchGate [Link]. 2

[2] Vincent Granville. *Off the Beaten Path Tutorial: Stochastic Processes and Simulations, Volume 1*. MLTechniques.com, 2022. [Link]. 5

[3] Kristen Grauman. Shape matching. 2008. University of Texas, Austin [Link]. 4

[4] Fred Park. Shape descriptor / feature extraction techniques. 2011. UCI iCAMP 2011 [Link]. 2

[5] Yu Vizilter and Sergey Zheltov. Geometrical correlation and matching of 2D image shapes. 2012. ResearchGate [Link]. 4