# Moments of Elliptic Fourier Descriptors

Octavian Soldea[1], Mustafa Unel[2], and Aytul Ercil[2]
[1]. *The Video Processing and Analysis Group, Philips Research, Eindhoven, The Netherlands*
[2] *Faculty of Engineering and Natural Sciences, Sabanci University, Istanbul, 34956 Turkey*
*E-mails: Octavian.Soldea@philips.com, {munel, aytulercil}@sabanciuniv.edu*

## Abstract

*This paper develops a recursive method for computing moments of 2D objects described by elliptic Fourier descriptors (EFD). Green's theorem is utilized to transform 2D surface integrals into 1D line integrals and EFD description is employed to derive recursions for moments computations. Experiments are performed to quantify the accuracy of our proposed method. Comparison with Bernstein-Bézier representations is also provided.*

## 1 Introduction

Moments of objects [10] are intrinsic to the shape [9], and therefore efficient computation of moments is a desirable feature for many practical tasks.

Moments of inertia are used in mechanical design and analysis. For example, in the design of aircrafts, ships, and automobiles the moments of inertia are employed to determine the dynamics of the vehicle [16]. In the medical domain, moments are used for automatic diagnosis and prognosis; for example they can be used in computing volumes of healthy and pathologic tissues [8].

Moment invariants are efficient tools in pattern recognition applications. In [18], the authors present a moment based pattern recognition application in agronomy and propose a measure for the analysis of the roundness of rose flowers. Another interesting application of moments can be found, for example, in optical character recognition systems such as [3]. Until recently, it was a common belief that projective moment invariants do not exist; however, their existence was proven in [15].

Although moments of objects in different forms have been widely studied in the literature, to the best of our knowledge, the moments of the elliptic Fourier descriptors (EFD) have not been explored until now. Since EFD representation is one of the most powerful boundary modeling tools, efficient computation of its moments may prove very useful in several model-based vision and pattern recognition applications. Motivated by this observation, in this work, we develop a computationally efficient recursive scheme for calculating moments of objects represented by elliptic Fourier descriptors. Several experiments are performed to quantify the accuracy of our proposed method and compare it with other representations such as Bernstein-Bézier representations [4].

## 2 Background on Elliptic Fourier Descriptors

Following [17], let $T$ be an arbitrary positive real number and let $C(t) : [0..T] \rightarrow R^2$,

$$C(t) = (x(t), y(t)) \tag{1}$$

be a planar curve parameterized by $t$, such that $C(t) \in C^{(2)}$. We can describe the curve in Equation $(1)$ using elliptic Fourier descriptors as follows:

$$\begin{pmatrix} x(t) \\ y(t) \end{pmatrix} = \sum_{i=0}^{\infty} \begin{pmatrix} a_i & b_i \\ c_i & d_i \end{pmatrix} \begin{pmatrix} \cos\left(\frac{2i\pi t}{T}\right) \\ \sin\left(\frac{2i\pi t}{T}\right) \end{pmatrix}, \tag{2}$$

where $a_0 = \frac{1}{T}\int_0^T x(t)\,dt$, $b_0 = c_0 = 0$, $d_0 = \frac{1}{T}\int_0^T y(t)\,dt$, $a_i = \frac{2}{T}\int_0^T x(t)\cos\left(\frac{2\pi it}{T}\right)dt$, $b_i = \frac{2}{T}\int_0^T x(t)\sin\left(\frac{2\pi it}{T}\right)dt$, $c_i = \frac{2}{T}\int_0^T y(t)\cos\left(\frac{2\pi it}{T}\right)dt$, $d_i = \frac{2}{T}\int_0^T y(t)\sin\left(\frac{2\pi it}{T}\right)dt$, for any $i \in N - \{0\}$. Since $\cos(\cdot)$ and $\sin(\cdot)$ are continuous functions, the integrability of $C(t)$ ensures existence of the above integrals.

## 3 Moments of 2D Shapes Represented by EFD

In this section, we develop a recursive scheme for computing moments of 2D shapes represented by EFD.

1051-4651/10 $26.00 © 2010 IEEE
DOI 10.1109/ICPR.2010.859

The recursive scheme provides efficient computation of moments.

We divide the computation of moment $m_{p,q}$, which is defined as a surface integral, into two components that are defined as 1D line integrals. The conversion from surface to line integral is achieved by the utilization of Green's theorem [5]. We outline the fundamental steps in deriving recursions and refer the reader to [13] for details of these derivations.

Consider a 2D shape $D \subseteq R^2$. The standard moment of order $p, q$ of $D$ is given as:

$$m_{p,q} = \int\int_D x^p y^q dx dy. \tag{3}$$

Green's theorem can be used to rewrite Equation (3) as

$$m_{p,q} = \frac{1}{2}\int_{t=0}^{T} x(t)^p\, y(t)^q \left[\frac{x(t)\, y'(t)}{p+1} - \frac{y(t)\, x'(t)}{q+1}\right] dt. \tag{4}$$

We define the following quantities:

$$\alpha_{i,j,p,q} = j\left(\frac{a_i d_j}{p+1} - \frac{b_j c_i}{q+1}\right), \tag{5}$$

$$\beta_{i,j,p,q} = j\left(\frac{-a_i c_j}{p+1} + \frac{c_i a_j}{q+1}\right) + i\left(\frac{d_i b_j}{p+1} - \frac{b_i d_j}{q+1}\right), \tag{6}$$

$$\gamma_{i,j,p,q} = i\left(\frac{-b_j c_i}{p+1} + \frac{d_j a_i}{q+1}\right), \tag{7}$$

$$m^c_{i,p,q} = \int_{t=0}^{T} x(t)^p\, y(t)^q \cos\left(\frac{2i\pi t}{T}\right) dt, \tag{8}$$

$$m^s_{i,p,q} = \int_{t=0}^{T} x(t)^p\, y(t)^q \sin\left(\frac{2i\pi t}{T}\right) dt. \tag{9}$$

Substituting Equation (2) into (4) and using Equations (5)-(9) we get

$$\begin{aligned} m_{p,q} = \frac{\pi}{2T}\sum_{i=0}^{\infty}\sum_{j=0}^{\infty}\Big\{ & \alpha_{i,j,p,q}\left(m^c_{|i-j|,p,q} + m^c_{i+j,p,q}\right) \\ & + \beta_{i,j,p,q}\left[m^s_{i+j,p,q} - \sigma(i-j)\, m^s_{|i-j|,p,q}\right] \\ & + \gamma_{i,j,p,q}\left(m^c_{|i-j|,p,q} - m^c_{i+j,p,q}\right)\Big\}. \end{aligned} \tag{10}$$

The computation of $m_{p,q}$ is based on the values $m^c_{i,p,q}$, and $m^s_{i,p,q}$, which will be computed recursively as shown below.

In the case $i > 0, p > 0$ and $q > 0$, we have

$$\begin{aligned} m^c_{i,p,q} = \frac{1}{2i}\sum_{k=0}^{\infty} k\Big\{ & p a_k\left(m^c_{|k-i|,p-1,q} - m^c_{k+i,p-1,q}\right) - \\ & p b_k\left[m^s_{k+i,p-1,q} - \sigma(k-i)\, m^s_{|k-i|,p-1,q}\right] + \\ & q c_k\left(m^c_{|k-i|,p,q-1} - m^c_{k+i,p,q-1}\right) - \\ & q d_k\left[m^s_{k+i,p,q-1} - \sigma(k-i)\, m^s_{|k-i|,p,q-1}\right]\Big\} \end{aligned}$$

and

$$\begin{aligned} m^s_{i,p,q} = \frac{1}{2i}\sum_{k=0}^{\infty} k\Big\{ & -p a_k\left[m^s_{k+i,p-1,q} + \sigma(k-i)\, m^s_{|k-i|,p-1,q}\right] \\ & + p b_k\left(m^c_{k+i,p-1,q} + m^c_{|k-i|,p-1,q}\right) \\ & - q c_k\left[m^s_{k+i,p,q-1} + \sigma(k-i)\, m^s_{|k-i|,p,q-1}\right] \\ & + q d_k\left(m^c_{k+i,p,q-1} + m^c_{|k-i|,p,q-1}\right)\Big\}. \end{aligned}$$

For details please see [13]. Next, we provide several simplified results for particular cases of interest.

- $i = 0, p = 0,$ and $q = 0$ :

$$m^c_{0,0,0} = T \text{ and } m^s_{0,0,0} = 0.$$

- $i = 0, p = 0,$ and $q > 0$ :

$$m^c_{0,0,q} = \sum_{i=0}^{\infty}\left(c_i m^c_{i,0,q-1} + d_i m^s_{i,0,q-1}\right) \text{ and } m^s_{0,0,q} = 0.$$

- $i = 0, p > 0,$ and $q = 0$ :

$$m^c_{0,p,0} = \sum_{i=0}^{\infty}\left(a_i m^c_{i,p-1,0} + b_i m^s_{i,p-1,0}\right) \text{ and } m^s_{0,p,0} = 0.$$

- $i = 0, p > 0,$ and $q > 0$ :

$$\begin{aligned} m^c_{0,p,q} = & \frac{1}{2}\sum_{i=0}^{\infty}\sum_{j=0}^{\infty}\Big[\left(a_i c_j + b_i d_j\right) m^c_{|i-j|,p-1,q-1} \\ & + \left(a_i c_j - b_i d_j\right) m^c_{i+j,p-1,q-1} \\ & + \left(-a_i d_j + b_i c_j\right)\sigma(i-j) m^s_{|i-j|,p-1,q-1} \\ & + \left(a_i d_j + b_i c_j\right) m^s_{i+j,p-1,q-1}\Big] \\ \text{and } m^s_{0,p,q} = & \ 0. \end{aligned}$$

- $i > 0, p = 0,$ and $q = 0$ :

$$m^c_{i,0,0} = m^s_{i,0,0} = 0.$$

- $i > 0, p = 0,$ and $q > 0$ :

$$\begin{aligned} m^c_{i,0,q} = & \frac{q}{2i}\sum_{k=0}^{\infty} k\Big[c_k\left(m^c_{|k-i|,0,q-1} - m^c_{k+i,0,q-1}\right) \\ & - d_k\left(m^s_{k+i,0,q-1} - \sigma(k-i)\, m^s_{|k-i|,0,q-1}\right)\Big] \end{aligned}$$

$$\begin{aligned} m^s_{i,0,q} = & \frac{q}{2i}\sum_{k=0}^{\infty} k\Big[-c_k\left(m^s_{k+i,0,q-1} + \sigma(k-i)\, m^s_{|k-i|,0,q-1}\right) \\ & + d_k\left(m^c_{k+i,0,q-1} + m^c_{|k-i|,0,q-1}\right)\Big]. \end{aligned}$$

- $i > 0, p > 0,$ and $q = 0$ :

$$\begin{aligned} m^c_{i,p,0} = & \frac{p}{2i}\sum_{k=0}^{\infty} k\Big[a_k\left(m^c_{|k-i|,p-1,0} - m^c_{k+i,p-1,0}\right) \\ & - b_k\left(m^s_{k+i,p-1,0} - \sigma(k-i)\, m^s_{|k-i|,p-1,0}\right)\Big] \end{aligned}$$

$$\begin{aligned} m^s_{i,p,0} = & \frac{p}{2i}\sum_{k=0}^{\infty} k\Big[-a_k\left(m^s_{k+i,p-1,0} + \sigma(k-i)\, m^s_{|k-i|,p-1,0}\right) \\ & + b_k\left(m^c_{|k-i|,p-1,0} + m^c_{k+i,p-1,0}\right)\Big]. \end{aligned}$$

# 4 Complexity Analysis

If schemes of computing moments can be converted into recursive algorithms [12], one can take advantage of *amortized times* of computations. In this context, usually, moments of higher degree depend on lower degrees.

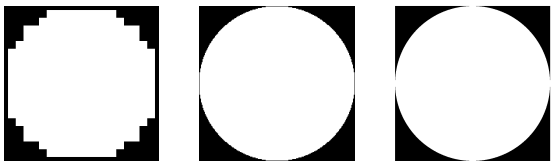
Let $M_r \triangleq \{m_{p,q} \,|p+q=r\}$ be the set of moments of order $r$. Moreover, let $\mathcal{M}_r \triangleq \bigcup_{i=0}^{r} M_i$ be the complete set of moments up to $r$. Suppose that the order, i.e. the upper limit of the summation in Equation (2), of the EFD used is $L$. Let $T_{m_{p,q}}$ be the time required for computing $m_{p,q}$. As detailed in [14], the amortized time for each moment is $O\left(L\left(p+q\right)^2\right)$ and $T_{m_{p,q}} = O\left(L^2 + L\left(p+q\right)^2\right)$.

In terms of memory consumption, our scheme needs $L^2\left(p+q+1\right)^4$ memory locations. For today's computers, this is a very small space requirement. All other schemes imply similar low memory requirements.

## 5 Experimental Results

We present experimental results related to the accuracy of our proposed computational scheme. We test our method on different shapes and provide several comparisons with other methods.

We first picked several shapes for which moments can easily be computed analytically. First, we use discrete moments computations method. Second, we use our proposed method. Third, as a ground truth, we use the analytical method where the surface integrals are transformed into line integrals and computed with Maple [6].



$(a)$ $(b)$ $(c)$

**Figure 1.** Circles with different radii are scaled for visual comparison: $(a)$ $R = 10$, $(b)$ $R = 100$, and $(c)$ $R = 1000$.

We selected three circles (disks) with radii $R = 10, 100,$ and $1000$, and centered at $(R, 0)$ , see Figure 1. Table 1 shows the relative errors for the discrete method of moment computations when compared to the ground truth provided by the analytical method. Our method, when compared to the ground truth, results in zero relative error up to fifteen significant decimal digits.

| | $m_{0,0}$ | $m_{2,0}$ | $m_{0,2}$ |
|---|---|---|---|
| $R = 10$ | $0.02915$ | $0.03473$ | $0.05704$ |
| $R = 100$ | $6.02450 \cdot 10^{-4}$ | $7.22542 \cdot 10^{-4}$ | $12.02910 \cdot 10^{-4}$ |
| $R = 1000$ | $2.28080 \cdot 10^{-5}$ | $2.73668 \cdot 10^{-5}$ | $4.56018 \cdot 10^{-5}$ |

**Table 1.** Relative errors of the discrete method.

We next compare the accuracy of moment computations for objects modeled by EFD and Bernstein-Bézier curves. In our experiments, we employ interpolation to approximate several point sets [4]. While efficient algorithms for computing moments of objects represented by Bernstein-Bézier curves can be found in [7], [12], and [11], our goal is to compare the accuracy of the moments computations for the Bernstein-Bézier and EFD representations. We evaluate Equation $(4)$ in Maple, symbolically, with a precision of one hundred digits.

A circle can be described exactly by EFD, but not necessarily with Bernstein-Bézier boundary curves. Since a circle can be approximated by zeroth and first harmonics, i.e. $n = 0,\ 1$ using EFD, i.e. six coefficients, we approximated the circle with a Bernstein-Bézier of degree five, which has six control points. The exact EFD representation of a circle implies higher accuracy when computing moments. In addition, we also present moments computed for a circle approximated with a Bernstein-Bézier of degree seven, which has eight control points. Table 2 illustrates accuracy of moments computations for a circle with radius $R = 1000$. Our method results in zero relative error up to fifteen significant decimal digits.

| Moments | Bernstein-Bézier | |
|---|---|---|
| | Degree 5 | Degree 7 |
| $m_{0,0}$ | 0.02070 | $12.97549 \cdot 10^{-4}$ |
| $m_{1,0}$ | 0.04019 | $25.57672 \cdot 10^{-4}$ |
| $m_{0,1}$ | 0.00000 | 0.00000 |
| $m_{2,0}$ | 0.05874 | $40.05980 \cdot 10^{-4}$ |
| $m_{1,1}$ | 0.00000 | 0.00000 |
| $m_{0,2}$ | 0.03110 | $4.12989 \cdot 10^{-4}$ |

**Table 2.** Relative errors of moments for a circle with radius $R = 1000$.

Finally, we consider 2D shapes from TOSCA benchmark database, see [2] and [1]. These shapes are modeled using EFD representation with different number of harmonics. In Figure 2, the first column presents original images whereas the remaining ones represent EFD approximations with $128$, $64$, $32$ and $16$ harmonics, respectively. Table 3 shows the computed moments up to order 3 for these objects.

| Moments | plier | scissor | horse | man |
|---|---|---|---|---|
| $m_{0,0}$ | 0.9999 | 1.0000 | -1.0000 | -0.9999 |
| $m_{1,0}$ | -0.0029 | -0.0013 | -0.0116 | -0.0008 |
| $m_{0,1}$ | -0.0135 | -0.0073 | 0.0013 | 0.0124 |
| $m_{2,0}$ | 0.2446 | 0.2206 | -0.1039 | -0.0827 |
| $m_{1,1}$ | -0.0004 | 0.0000 | -0.0151 | 0.0023 |
| $m_{0,2}$ | 0.5210 | 0.5041 | -0.1351 | -0.3193 |
| $m_{3,0}$ | -0.0020 | -0.0008 | -0.0053 | -0.0001 |
| $m_{2,1}$ | 0.1476 | -0.1066 | 0.0029 | -0.0113 |
| $m_{1,2}$ | -0.0011 | -0.0007 | -0.0001 | -0.0001 |
| $m_{0,3}$ | -0.0173 | 0.2278 | -0.0214 | -0.0491 |

**Table 3. Moment computation results for different objects from TOSCA database**

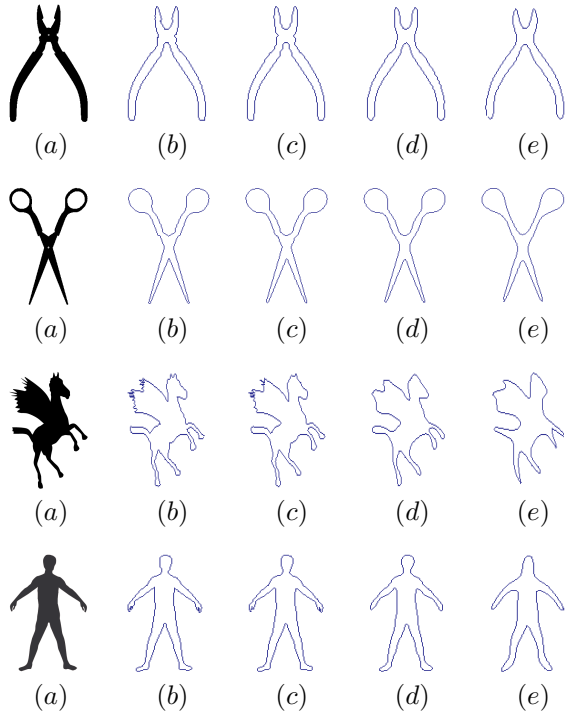



**Figure 2. Shapes from TOSCA benchmark database and EFD approximations with** $128, 64, 32,$ **and** $16$ **harmonics.**

## 6 Conclusions

We have outlined a computationally efficient scheme for calculating moments of objects represented by elliptic Fourier descriptors (EFD). The method is recursive and therefore implies fast computation of moments. We have experimented our scheme on 2D shapes modeled by EFD and provided measurement of accuracy of our method along with comparisons with some other techniques.

## References

[1] A. M. Bronstein, M. M. Bronstein, A. M. Bruckstein, and R. Kimmel. Analysis of two-dimensional non-rigid shapes. *International Journal of Computer Vision*, 78(1):67–88, 2008.

[2] A. M. Bronstein, M. M. Bronstein, and R. Kimmel. *Numerical Geometry of Non-Rigid Shapes*. Springer, 2008.

[3] L. G. Cash and M. Hatamian. Optical character recognition by the method of moments. *Computer Vision, Graphics, and Image Processing*, 39:291–310, 1987.

[4] E. Cohen, F. R. Riesenfeld, and G. Elber. *Geometric Modeling with Splines, An Introduction*. A. K. Peters, 2001.

[5] M. G. Fikhtengol'ts. *The Fundamentals of Mathematical Analysis*. Pergamon Press, New York, 1965.

[6] K. O. Geddes, K. M. Heal, G. Labahn, S. M. Vorkoetter, J. McCarron, P. DeMarco, and M. B. Monagan. *Maple 8 Advanced Programming Guide*. Waterloo Maple, Inc., ISBN-10: 189451128X, ISBN-13: 978-1894511285, 2002.

[7] C. Gonzales-Ochoa, S. McCamnon, and J. Peters. Computing moments of objects enclosed by piecewise polynomial surfaces. *ACM Transactions on Graphics*, 17(3):143–157, 1998.

[8] A. C. Jirapatnakul, A. P. Reeves, A. M. Biancardi, D. F. Yankelevitz, and C. I. Henschke. Identification of asymmetric pulmonary nodule growth using a moment-based algorithm. *Proceedings of the SPIE, Medical Imaging, Computer-Aided Diagnosis*, 7260:72602W–72602W–8, February 2009.

[9] L. J. Meriam. Dynamics. *John Wiley & Sons, Inc. New-York*, 1966.

[10] J. R. Prokop and P. A. Reeves. A survey of moment-based techniques for unoccluded object representation and recognition. *CVGIP: Graphical Models and Image Processing*, 54(5):438–460, 1992.

[11] A. S. Sheynin and V. A. Tuzikov. Moment computation for objects with spline curve boundary. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(10):1317–1322, 2003.

[12] O. Soldea, G. Elber, and E. Rivlin. Exact and efficient computation of moments of free-form surface and trivariate based geometry. *Computer-Aided Design*, 34(7):529–539, June 2000.

[13] O. Soldea, M. Unel, and A. Ercil. Appendix: Derivation details of equations. *http://www.cs.technion.ac.il/∼octavian/PUBLICATIONS/ICPR_app.pdf*, 2010.

[14] O. Soldea, M. Unel, and A. Ercil. Recursive computation of moments of 2d objects represented by elliptic fourier descriptors. *Pattern Recognition Letters, 10.1016/j.patrec.2010.02.009*, 2010.

[15] T. Suk and J. Flusser. Projective moment invariants. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(10):1364–1367, October 2004.

[16] R. Weinstein, J. Teran, and R. Fedkiw. Dynamic simulation of articulated rigid bodies with contact and collision. *IEEE Transactions on Visualization and Computer Graphics*, 12(3):265–374, 2006.

[17] K. K. R. Yip, K. P. Tam, and K. N. D. Leung. Application of elliptic fourier descriptors to symmetry detection under parallel projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(3):277–286, March 1994.

[18] M. Zhenjiang. Zernike moment-based image shape analysis and its application. *Pattern Recognition Letters*, 21(2):169–177, 2000.