On the Calculation of Arbitrary Moments of Polygons

Carsten Steger

Technical Report FGBV–96–05

October 1996

Forschungsgruppe Bildverstehen (FG BV), Informatik IX
Technische Universität München

Technical Report FGBV–96–05

Address: Carsten Steger
Forschungsgruppe Bildverstehen, Informatik IX
Technische Universität München
Orleansstraße 34
81667 München
Germany
WWW: http://wwwradig.informatik.tu-muenchen.de/forschung/fgbv/
E-mail: stegerc@informatik.tu-muenchen.de
fgbv@informatik.tu-muenchen.de

On the Calculation of Arbitrary Moments of Polygons

Carsten Steger

Abstract—*This paper describes a method to calculate moments of arbitrary order from polygons that enclose a region of two-dimensional space. These include various and often used shape features, such as area, center of gravity, and second order moments. Contrary to most approaches, which compute these features from discrete pixel data, here they are calculated by using only the points of the enclosing polygon, which may be sub-pixel accurate. This has two main advantages. Firstly, since the precise points of the boundary of the shape are used, the resulting features will be computed with maximum accuracy. Furthermore, since the boundary typically consists of fewer points than the whole region, the procedure is computationally more efficient.*

1 Introduction

A problem that frequently occurs in image processing tasks is the calculation of area, centroid (center of gravity), and second order moments (moments of inertia) of a region R in an image. These are important shape features in themselves. For example, they can be used to fit ellipses to extracted contours [3]. Often they are also used to compute the dominant directions and approximate diameters of a region. These in turn are important for camera calibration tasks, for example, where they can be used to approximate values for some parameters of the exterior orientation to thus obtain good starting values for a calibration procedure [8]. Additionally, recent work on moment based fitting of elliptic curve segments [13] and super-quadrics [12] requires higher-order moments to be calculated.

The moment of order (p, q) of an arbitrary region R is given by

$$\nu_{p,q} = \iint\limits_R x^p y^q \, dx \, dy \ . \tag{1}$$

If $p = q = 0$ we obtain the area a of R. The moments are usually normalized by the area a of R:

$$\alpha_{p,q} = \frac{1}{a} \iint\limits_R x^p y^q \, dx \, dy \ . \tag{2}$$

Thus, we have $\alpha_{0,0} = 1$. Finally, for $p + q \geq 2$, one is usually only interested in the normalized central moments of R:

$$\mu_{p,q} = \frac{1}{a} \iint\limits_R (x - \alpha_{1,0})^p (y - \alpha_{0,1})^q \, dx \, dy \ . \tag{3}$$

This equation, of course, also holds for $p + q = 1$ with $\mu_{1,0} = \mu_{0,1} = 0$. The central moments $\mu_{p,q}$ can be calculated from the moments $\alpha_{i,j}$ in the following way:

$$\mu_{p,q} = \sum_{i=0}^{p} \sum_{j=0}^{q} \binom{p}{i} \binom{q}{j} (-1)^{p+q-i-j} \alpha_{1,0}^{p-i} \alpha_{0,1}^{q-j} \alpha_{i,j} \quad . \tag{4}$$

Usually, the region R will be discrete, i.e., consist of a set of pixels, each of which has an area of 1. The integrals in the above equations can then simply be calculated by summation over the region R. However, if regions are extracted by a sub-pixel precise feature extraction algorithm, e.g., by the line detector given in [10], only the closed boundary b of the region is known, usually as a sub-pixel precise contour, which can be regarded as a polygon. Therefore, the above equations cannot be applied to calculate the moments. There are two obvious solutions to this problem. The first is to discretize the region R to the pixel raster, which is undesirable since sub-pixel accuracy is lost. Alternatively, one may triangulate the polygon, and calculate the moments by computing them for each triangle, which can be done easily, and then to finally add up the results [14, 11, 9]. However, triangulation is a costly operation, and therefore a scheme that uses only the points on the polygon p to compute the moments is highly desirable. It is interesting to note that, using a certain way of triangulating a polygon, formulas similar to the ones derived in this paper can be obtained [9]. However, the derivation given there lacks a rigorous proof of how these formulas were obtained.

In contrast to these approaches, one can apply Green's theorem to reduce the area integral over R in (1) to a curve integral along its border b [1]. Various authors [4, 14] have used this approach to derive recursive formulas to calculate the moments, but no general closed form has been given. In order to calculate $\nu_{p,q}$ these recursive formulas require all moments of order $p' + q' \leq p + q$ to be calculated. Furthermore, the recursion formulas can lead to numerical instabilities [14]. These problems are particularly undesirable in applications where only very few higher order moments are required, e.g., moment-based fitting methods [13, 12]. In these cases an efficient, minimal closed form solution for moments of arbitrary order is highly desirable.

2 Mathematical Tools

As described above, a way to calculate the moments is to apply the powerful result of Green's theorem, sometimes also referred to as the Green formula, the Gauss formula, or the Stokes formula. This theorem allows one to compute the integral of a function over a sub-domain R of the two-dimensional space by reducing it to a curve integral over the border b of R [1, Section 3.1.13.1]. More formally, it can be stated as follows: Let $P(x, y)$ and $Q(x, y)$ be two continuously differentiable functions on the two-dimensional region R, and let $b(t)$ be the boundary of R. If b is piecewise differentiable and oriented such that it is traversed in positive direction (counterclockwise), an integral over the region R can be reduced to a curve integral over the boundary b of R in the following manner:

$$\iint_R \frac{\partial Q}{\partial x} - \frac{\partial P}{\partial y} \, dx \, dy = \int_b P \, dx + Q \, dy \quad . \tag{5}$$

It is obvious that the computation of an integral of an arbitrary function $F(x, y)$ over R, as is the case for the moments, where $F(x, y) = x^p y^q$, $F(x, y)$ must somehow be decomposed into $\partial P/\partial y$ and $\partial Q/\partial x$. It is also obvious that this decomposition cannot be unique, and therefore the choice of the decomposition is essentially arbitrary.

The integral on the right side of (5) is a general curve integral. It is defined as follows [1, Section 3.1.8.4]: Let $b(t) = (x(t), y(t)),\ t \in [t_1, t_2]$ be a curve, and let $f(x, y)$ be a continuous function. Then the following two curve integrals exist and can be transformed to the definite integrals:

$$\int\limits_b f(x,y)\,dx \;=\; \int\limits_{t_1}^{t_2} f(x(t),y(t))x'(t)\,dt \tag{6}$$

$$\int\limits_b f(x,y)\,dy \;=\; \int\limits_{t_1}^{t_2} f(x(t),y(t))y'(t)\,dt \ \ . \tag{7}$$

Furthermore, let $P(x, y)$ and $Q(x, y)$ be two continous functions. The general curve integral is then given by [1, Section 3.1.8.3]:

$$\int\limits_b P(x,y)\,dx + Q(x,y)\,dy = \int\limits_b P(x,y)\,dx + \int\limits_b Q(x,y)\,dy \ \ . \tag{8}$$

Curve integrals have some important properties which will be useful in later sections: Let $b_1(t),\ t \in [t_1, t_2]$ and $b_2(t),\ t \in [t_2, t_3]$ be two curves with $b_1(t_2) = b_2(t_2)$, and let $b = b_1 \cup b_2$. The curve integral over b can then be calculated as follows:

$$\int\limits_b f(x,y)\,dx = \int\limits_{b_1} f(x,y)\,dx + \int\limits_{b_2} f(x,y)\,dx \ \ . \tag{9}$$

Furthermore, if the direction of the curve is reversed, i.e., if b' is the reverse of b, the sign of the integral changes:

$$\int\limits_b f(x,y)\,dx = -\int\limits_{b'} f(x,y)\,dx \ \ . \tag{10}$$

3 Application to Closed Polygons

We now have all the tools at hand to compute the moments of R by only using the points on its border b. Before we can apply these tools, however, we need to consider how we can parameterize $b(t)$.

A closed polygon p with points $p_i = (x_i, y_i),\ i \in \{0, \ldots, n\}$, and $p_0 = p_n$ bounding a region R in the two-dimensional plane, can be regarded as a piecewise linear curve b, which in turn can be regarded as the union of n line segments

$$b(t) = \bigcup_{i=1}^{n} b_i(t) \ \ , \tag{11}$$

where $b_i(t),\ t \in [0, 1]$ is given by

$$b_i(t) = tp_i + (1-t)p_{i-1} \ \ . \tag{12}$$

Hence it follows that the coordinate functions and their derivatives needed to calculate the curve integral are given by

$$\begin{aligned} x_i(t) &= tx_i + (1-t)x_{i-1} &(13)\\ y_i(t) &= ty_i + (1-t)y_{i-1} &(14)\\ x_i'(t) &= x_i - x_{i-1} &(15)\\ y_i'(t) &= y_i - y_{i-1} \ . &(16)\end{aligned}$$

Therefore, any general curve integral along a polygon $p = b(t)$ can be calculated in the following manner:

$$\int\limits_b P\,dx + Q\,dy = \sum_{i=1}^{n} \int\limits_{b_i} P\,dx + Q\,dy \ . \tag{17}$$

Please note that this parametrization of the boundary p of R is more general than the one used in other approaches [4, 9] since it is valid for arbitrary polygons, and therefore does not require different cases to be treated specially.

4 Calculation of Moments

An unnormalized moment of arbitrary order of a region R is given by (1). In order to apply (5) we need to decompose $x^p y^q$ into $\partial Q/\partial x$ and $\partial P/\partial y$. For reasons of simplicity we choose

$$\frac{\partial Q}{\partial x} = x^p y^q \quad \text{and} \quad \frac{\partial P}{\partial y} = 0 \ . \tag{18}$$

Hence

$$P(x,y) = 0 \quad \text{and} \quad Q(x,y) = \frac{1}{p+1} x^{p+1} y^q \ . \tag{19}$$

Therefore, the moment $\nu_{p,q}$ of a region R can be calculated as follows:

$$\nu_{p,q} = \iint\limits_R x^p y^q\,dx\,dy = \int\limits_b \frac{1}{p+1} x^{p+1} y^q\,dy \ . \tag{20}$$

By (17), the integral in (20) can be calculated as the sum over the curve integrals along the line segments of the polygon. Each term of the sum is given by:

$$\begin{aligned} &\int\limits_{b_i} \frac{1}{p+1} x^{p+1} y^q\,dy \\ &= \frac{1}{p+1} \int\limits_0^1 x_i(t)^{p+1} y_i(t)^q y_i'(t)\,dt \\ &= \frac{1}{p+1} \int\limits_0^1 (tx_i + (1-t)x_{i-1})^{p+1} (ty_i + (1-t)y_{i-1})^q (y_i - y_{i-1})\,dt \\ &= \frac{1}{p+1} (y_i - y_{i-1}) \int\limits_0^1 \Big(\sum_{k=0}^{p+1} \binom{p+1}{k} x_i^k x_{i-1}^{p+1-k} t^k (1-t)^{p+1-k} \Big) \times \end{aligned}$$

$$
\begin{aligned}
& \Big(\sum_{l=0}^{q}\binom{q}{l} y_i^l y_{i-1}^{q-l} t^l (1-t)^{q-l}\Big)\,dt \\
= \; & \frac{1}{p+1}(y_i - y_{i-1}) \int_0^1 \sum_{k=0}^{p+1}\sum_{l=0}^{q}\binom{p+1}{k}\binom{q}{l} x_i^k x_{i-1}^{p+1-k} y_i^l y_{i-1}^{q-l} t^{k+l}(1-t)^{p+q+1-k-l}\,dt \\
= \; & \frac{1}{p+1}(y_i - y_{i-1}) \sum_{k=0}^{p+1}\sum_{l=0}^{q}\binom{p+1}{k}\binom{q}{l} x_i^k x_{i-1}^{p+1-k} y_i^l y_{i-1}^{q-l} \int_0^1 t^{k+l}(1-t)^{p+q+1-k-l}\,dt \\
= \; & \frac{1}{p+1}(y_i - y_{i-1}) \sum_{k=0}^{p+1}\sum_{l=0}^{q}\binom{p+1}{k}\binom{q}{l} x_i^k x_{i-1}^{p+1-k} y_i^l y_{i-1}^{q-l} B(k+l+1, p+q+2-k-l) \\
= \; & (y_i - y_{i-1}) \sum_{k=0}^{p+1}\sum_{l=0}^{q} a_{k,l}^{p+1,q} x_i^k x_{i-1}^{p+1-k} y_i^l y_{i-1}^{q-l} \;, \qquad (21)
\end{aligned}
$$

where $a_{k,l}^{p+1,q}$ is a coefficient given by

$$
a_{k,l}^{p+1,q} = \frac{1}{(p+q+2)(p+1)} \cdot \frac{\binom{p+1}{k}\binom{q}{l}}{\binom{p+q+1}{k+l}} \quad . \qquad (22)
$$

Here, we have used the upper indices $p+1$ and q to denote the range of integers for k and l. Therefore, for arbitrary p and q the unnormalized moments can be calculated as

$$
\nu_{p,q} = \sum_{i=1}^{n}(y_i - y_{i-1}) \sum_{k=0}^{p+1}\sum_{l=0}^{q} a_{k,l}^{p+1,q} x_i^k x_{i-1}^{p+1-k} y_i^l y_{i-1}^{q-l} \quad . \qquad (23)
$$

However, this equation is not satisfactory for two reasons. Firstly, it doesn't reflect the inherent symmetry of the problem. If we exchange x and y, we expect to get the same formulae. Secondly, if we expand the case $p = q = 0$ we obtain:

$$
\nu_{0,0} = \sum_{i=1}^{n}(y_i - y_{i-1})(\frac{1}{2}x_{i-1} + \frac{1}{2}x_i) \quad . \qquad (24)
$$

A closer inspection reveals that the terms for $x_{i-1}y_{i-1}$ and $x_i y_i$ will telescope, i.e., cancel in successive terms of the sum. Therefore, the formula reduces to

$$
\nu_{0,0} = \frac{1}{2}\sum_{i=1}^{n} x_{i-1}y_i - x_i y_{i-1} \quad . \qquad (25)
$$

For higher order moments even more subtle cancellations occur. Therefore, the question arises whether there is a canonical formula for calculating the moments by using a mimimum number of terms.

Proposition 1 *It is always possible to transform (21), i.e., each term $\nu_{p,q,i}$ of (23) into the canonical form*

$$
\nu_{p,q,i} = (x_{i-1}y_i - x_i y_{i-1}) \sum_{k=0}^{p}\sum_{l=0}^{q} c_{k,l}^{p,q} x_i^k x_{i-1}^{p-k} y_i^l y_{i-1}^{q-l} \;, \qquad (26)
$$

where $c_{k,l}^{p,q}$ is a suitably chosen coefficient.

Proof: See Appendix A. □

The formula for computing the moments given in (26) is minimal since no terms can be factored out of the double sum.

In the proof of Proposition 1, the coefficients $c_{k,l}^{p,q}$ are defined by a sum over a range of $a_{i,j}^{p+1,q}$. The next proposition establishes that there exists a simple, symmetric closed form for these coefficients.

Proposition 2 *The coefficients $c_{k,l}^{p,q}$ are given by*

$$c_{k,l}^{p,q} = \frac{1}{(p+q+2)(p+q+1)\binom{p+q}{p}} \binom{k+l}{l} \binom{p+q-k-l}{q-l} \ . \tag{27}$$

Proof: See Appendix B. □

Therefore, the unnormalized moments can be calculated by

$$\begin{aligned} \nu_{p,q} \quad = \quad & \frac{1}{(p+q+2)(p+q+1)\binom{p+q}{p}} \times \\ & \sum_{i=1}^{n} (x_{i-1}y_i - x_i y_{i-1}) \sum_{k=0}^{p} \sum_{l=0}^{q} \binom{k+l}{l} \binom{p+q-k-l}{q-l} x_i^k x_{i-1}^{p-k} y_i^l y_{i-1}^{q-l} \ . \end{aligned} \tag{28}$$

The normalized moments can, of course, be obtained by dividing the result by $a = \nu_{0,0}$.

For reference purposes we list the most commonly used normalized and central moments up to order 2:

$$a = \frac{1}{2} \sum_{i=1}^{n} x_{i-1}y_i - x_i y_{i-1} \tag{29}$$

$$\alpha_{1,0} = \frac{1}{6a} \sum_{i=1}^{n} (x_{i-1}y_i - x_i y_{i-1})(x_{i-1} + x_i) \tag{30}$$

$$\alpha_{0,1} = \frac{1}{6a} \sum_{i=1}^{n} (x_{i-1}y_i - x_i y_{i-1})(y_{i-1} + y_i) \tag{31}$$

$$\alpha_{2,0} = \frac{1}{12a} \sum_{i=1}^{n} (x_{i-1}y_i - x_i y_{i-1})(x_{i-1}^2 + x_{i-1}x_i + x_i^2) \tag{32}$$

$$\alpha_{1,1} = \frac{1}{24a} \sum_{i=1}^{n} (x_{i-1}y_i - x_i y_{i-1})(2x_{i-1}y_{i-1} + x_{i-1}y_i + x_i y_{i-1} + 2x_i y_i) \tag{33}$$

$$\alpha_{0,2} = \frac{1}{12a} \sum_{i=1}^{n} (x_{i-1}y_i - x_i y_{i-1})(y_{i-1}^2 + y_{i-1}y_i + y_i^2) \tag{34}$$

$$\mu_{2,0} = \alpha_{2,0} - \alpha_{1,0}^2 \tag{35}$$

$$\mu_{1,1} = \alpha_{1,1} - \alpha_{1,0}\alpha_{0,1} \tag{36}$$

$$\mu_{0,2} = \alpha_{0,2} - \alpha_{0,1}^2 \ . \tag{37}$$

It should be noted that (28) only holds if the polygon p encloses the region R counterclockwise. However, from (10) it is obvious that only the sign of the area changes if p encloses R clockwise. Hence we have a simple criterion for the decision of whether the result of (28) is valid, namely the sign of a. If a is negative, every calculated moment needs to be multiplied by -1. Furthermore, the calculation of central moments according to (4) is only efficient for the second order moments. For higher order moments it is more efficient to plug $x_i - \alpha_{1,0}$ and $y_i - \alpha_{0,1}$ directly into (28).

5 Examples

Let us now demonstrate that the moments calculated by considering only the points p_i of the enclosing polygon yield correct results. Consider the rectangle given by $p_0 = (2,0)$, $p_1 = (10,4)$, $p_2 = (8,8)$, and $p_3 = (0,4)$. In order to be able to compute the moments by the formulae derived above, we need to introduce an additional point $p_4 = p_0$ to close the polygon. Figure 1 displays this rectangle.

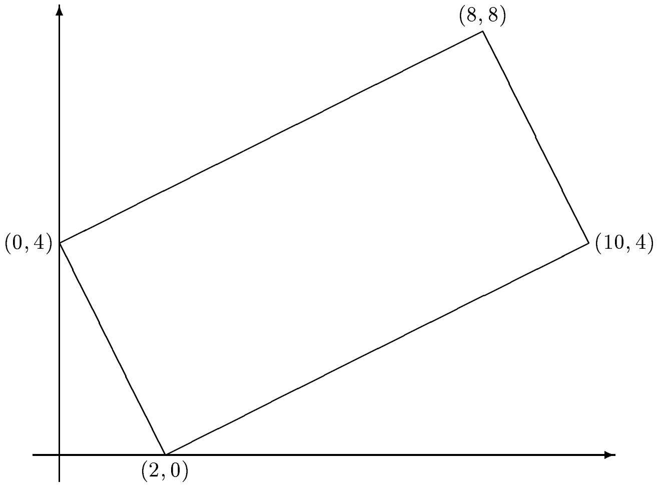

Figure 1: The rectangle used in the first example

From the geometry of this rectangle it is obvious that $a = 40$, and that the centroid is $(\alpha_{1,0}, \alpha_{0,1}) = (5,4)$. Hence, we will not derive these values by integration. The second moment $\alpha_{2,0}$ can be calculated as follows:

$$\begin{aligned}
\alpha_{2,0} &= \frac{1}{a} \iint\limits_R x^2\,dx\,dy \\
&= \frac{1}{a}\Big(\int\limits_0^2 \int\limits_{4-2x}^{x/2+4} x^2\,dx\,dy + \int\limits_2^8 \int\limits_{x/2-1}^{x/2+4} x^2\,dx\,dy + \int\limits_8^{10} \int\limits_{x/2-1}^{24-2x} x^2\,dx\,dy \Big) \\
&= \frac{1}{40}\Big(\int\limits_0^2 \Big(x^2 y \Big|_{y=4-2x}^{x/2+4} \Big)\,dx + \int\limits_2^8 \Big(x^2 y \Big|_{y=x/2-1}^{x/2+4} \Big)\,dx + \int\limits_8^{10} \Big(x^2 y \Big|_{y=x/2-1}^{24-2x} \Big)\,dx \Big)
\end{aligned}$$

$$
\begin{aligned}
&= \frac{1}{40}\Big(\int_0^2 x^2(\frac{1}{2}x+4) - x^2(4-2x)\,dx + \int_2^8 x^2(\frac{1}{2}x+4) - x^2(\frac{1}{2}x-1)\,dx \\
&\qquad + \int_8^{10} x^2(24-2x) - x^2(\frac{1}{2}x-1)\,dx\Big) \\
&= \frac{1}{40}\Big(\Big(\frac{5}{8}x^4\Big|_0^2\Big) + \Big(\frac{5}{3}x^3\Big|_2^8\Big) + \Big(-\frac{5}{8}x^4 + \frac{25}{3}x^3\Big|_8^{10}\Big)\Big) \\
&= 30\frac{2}{3} \ . \qquad (38)
\end{aligned}
$$

Similarly, we obtain

$$
\begin{aligned}
\alpha_{1,1} &= 22 \qquad (39)\\
\alpha_{0,2} &= 18\frac{2}{3} \ . \qquad (40)
\end{aligned}
$$

We now calculate a by using (29):

$$
a = \frac{1}{2}(2\cdot 4 - 10\cdot 0 + 10\cdot 8 - 8\cdot 4 + 8\cdot 4 - 0\cdot 8 + 0\cdot 0 - 2\cdot 4) = 40 \ . \qquad (41)
$$

Therefore, the area computed using (29) yields the correct result. Furthermore, we can see that the rectangle is indeed oriented counterclockwise since $a > 0$.

For $\alpha_{1,0}$ and $\alpha_{0,1}$, by (30) and (31) we have

$$
\begin{aligned}
\alpha_{1,0} &= \frac{1}{6a}((2+10)\cdot(2\cdot 4 - 10\cdot 0) + (10+8)\cdot(10\cdot 8 - 8\cdot 4) + \\
&\qquad (8+0)\cdot(8\cdot 4 - 0\cdot 8) + (0+2)\cdot(0\cdot 0 - 2\cdot 4)) \\
&= 5 \qquad (42)\\
\alpha_{0,1} &= \cdots \\
&= 4 \ . \qquad (43)
\end{aligned}
$$

Again, (30) and (31) yield the correct results with much less computational burden.

To calculate $\alpha_{2,0}$ and $\alpha_{0,2}$ we use (32) and (34):

$$
\begin{aligned}
\alpha_{2,0} &= \frac{1}{12a}((2^2 + 2\cdot 10 + 10^2)\cdot(2\cdot 4 - 10\cdot 0) + (10^2 + 10\cdot 8 + 8^2)\cdot(10\cdot 8 - 8\cdot 4) + \\
&\qquad (8^2 + 8\cdot 0 + 0^2)\cdot(8\cdot 4 - 0\cdot 8) + (0^2 + 0\cdot 2 + 2^2)\cdot(0\cdot 0 - 2\cdot 4)) \\
&= 30\frac{2}{3} \qquad (44)\\
\alpha_{0,2} &= \cdots \\
&= 18\frac{2}{3} \ . \qquad (45)
\end{aligned}
$$

For $\alpha_{1,1}$ we obtain by (33):

$$
\alpha_{1,1} = \frac{1}{24a}((2\cdot 2\cdot 0 + 2\cdot 4 + 10\cdot 0 + 2\cdot 10\cdot 4)\cdot(2\cdot 4 - 10\cdot 0) +
$$

$$
\begin{aligned}
& (2\cdot 10\cdot 4 + 10\cdot 8 + 8\cdot 4 + 2\cdot 8\cdot 8)\cdot(10\cdot 8 - 8\cdot 4) + \\
& (2\cdot 8\cdot 8 + 8\cdot 4 + 0\cdot 8 + 2\cdot 0\cdot 4)\cdot(8\cdot 4 - 0\cdot 8) + \\
& (2\cdot 0\cdot 4 + 0\cdot 0 + 2\cdot 4 + 2\cdot 2\cdot 0)\cdot(0\cdot 0 - 2\cdot 4)) \\
= \; & 22 \; . \qquad (46)
\end{aligned}
$$

Again, for the second order moments, (32), (34), and (33) yield the correct results. The central second order moments are given by:

$$\mu_{2,0} = \alpha_{2,0} - \alpha_{1,0}^2 = 30\frac{2}{3} - 25 = \frac{17}{3} \qquad (47)$$

$$\mu_{1,1} = \alpha_{1,1} - \alpha_{1,0}\alpha_{0,1} = 22 - 20 = 2 \qquad (48)$$

$$\mu_{0,2} = \alpha_{0,2} - \alpha_{0,1}^2 = 18\frac{2}{3} - 16 = \frac{8}{3} \; . \qquad (49)$$

From these three values, according to [3, Appendix A] we can compute the parameters of an ellipse with the same second order moments by calculating the eigenvalues and eigenvectors of the following matrix:

$$\frac{1}{4(\mu_{2,0}\mu_{0,2} - \mu_{1,1}^2)} \begin{pmatrix} \mu_{0,2} & -\mu_{1,1} \\ -\mu_{1,1} & \mu_{2,0} \end{pmatrix} = \frac{3}{400} \begin{pmatrix} 8 & -6 \\ -6 & 17 \end{pmatrix} \; . \qquad (50)$$

The eigenvalues are given by the solutions of

$$\begin{vmatrix} 8-\lambda & -6 \\ -6 & 17-\lambda \end{vmatrix} = (8-\lambda)(17-\lambda) - 36 = \lambda^2 - 25\lambda + 100 \; . \qquad (51)$$

Hence, $\lambda_1 = 5$ and $\lambda_2 = 20$. Therefore, the corresponding major axes of the ellipse have the following lengths: $a = 2/(\sqrt{5\cdot 3/400}) = 8\sqrt{5/3}$ and $b = 2/(\sqrt{20\cdot 3/400}) = 4\sqrt{5/3}$. The directions of the major axes a and b are given by $(2,1)$ and $(1,-2)$, respectively, as is easily obtainable by calculating the corresponding eigenvectors. Thus, the dominant directions of this region were obtained correctly. Obviously, this can be done much easier for a rectangle, but the approach is also valid for arbitrary shapes, as the next example shows.

Figure 2(a) displays a calibration target, and Fig. 2(b) the upper-rightmost calibration mark [8]. From this mark, edges were extracted with sub-pixel precision by extracting bright lines in the gradient image [10]. Figure 3(a) displays the resulting edges. In this example, 103 edge points were found, leading to a closed polygon with 102 line segments. From these edge points, the moments of the extracted shape can be calculated as

$$
\begin{aligned}
a &= 566.32474 \\
\alpha_r &= 26.40761 \\
\alpha_c &= 28.17205 \\
\alpha_{rr} &= 770.99165 \\
\alpha_{rc} &= 729.00953 \\
\alpha_{cc} &= 824.30414 \\
\mu_{rr} &= 73.62978 \\
\mu_{rc} &= -14.94700 \\
\mu_{cc} &= 30.63971 \; ,
\end{aligned}
$$

where r and c denote the row and column axes, respectively. According to [3, Appendix A], the corresponding ellipse with the same moments has a major axis of length $a = 35.39849$ and a minor axis of $b = 20.37789$, with the angle of the major axis to the column axis given by $\varphi = 72.59321°$. Figure 3(b) displays this ellipse superimposed onto the extracted edge points. As can be seen, the difference is hardly noticeable.

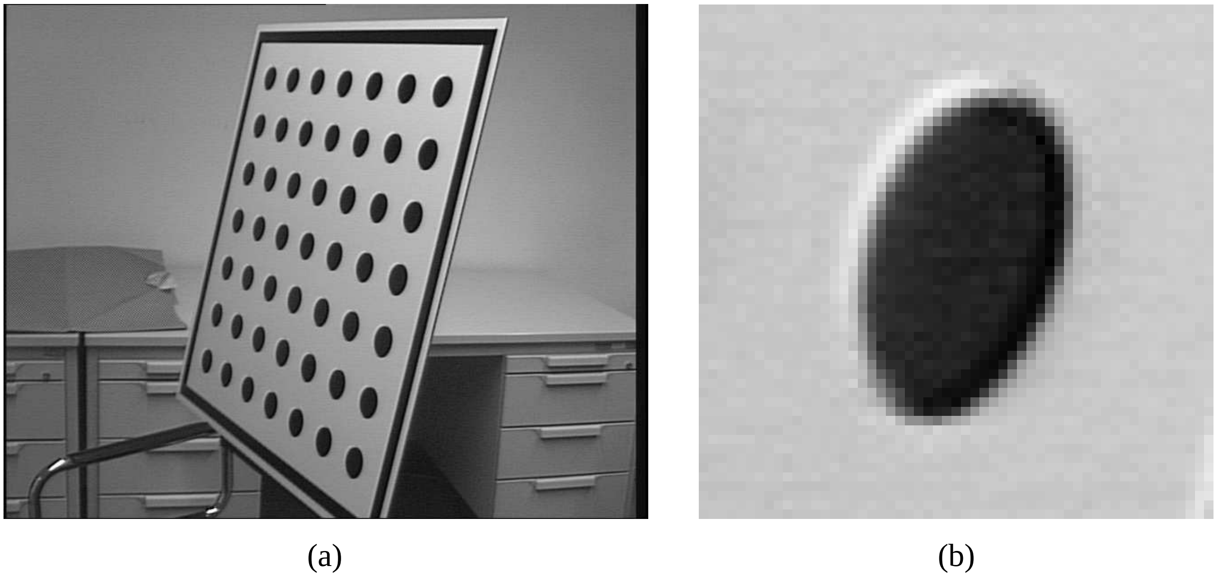

Figure 2: A calibration target (a) and one calibration mark (b).

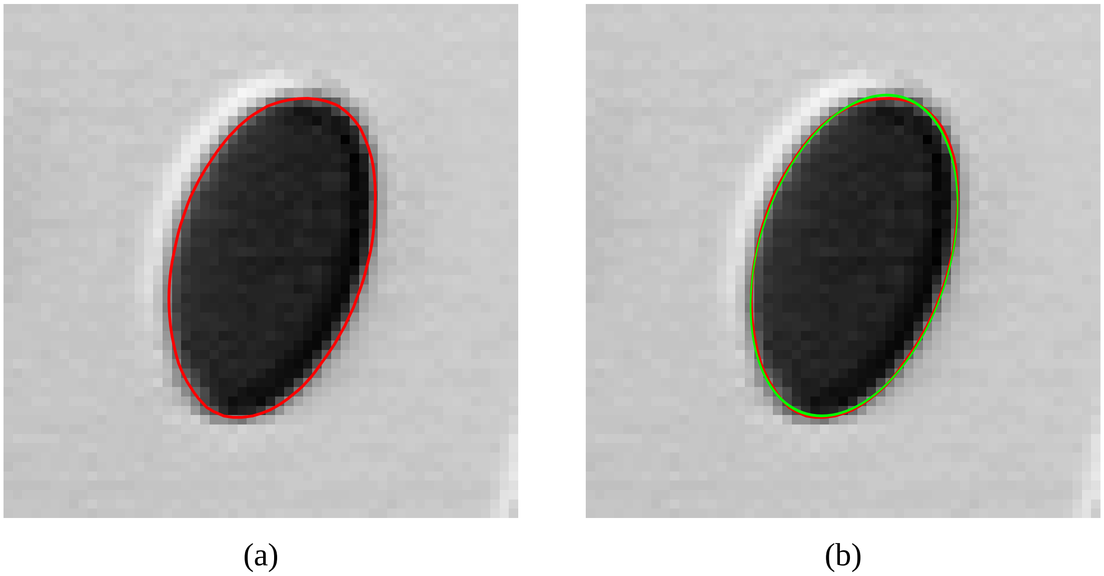

Figure 3: Extracted edges (a) and ellipse with the same moments (b).

6 Conclusions

This paper has presented an explicit method for the calculation of moments of arbitrary closed polygons. Contrary to most implementations, which obtain the moments from discrete pixel data, this approach calculates moments by using only the border of a region. Moments of subpixel precise features may thus be computed without loss of accuracy. Furthermore, since no explicit region needs to be constructed, and because the border of a region usually consists of significantly fewer points than the entire region, the approach is very efficient. The presented algorithm will be used in the vision system described in [5, 6, 7] to approximate edges by straight lines and ellipse segments using an approach similar to [13].

A Proof of Proposition 1

We will calculate the coefficients $b_{k,l}^{p+1,q+1}$ of $x_i^k x_{i-1}^{p+1-k} y_i^l y_{i-1}^{q+1-l}$ if we expand (26) and (21). In the first case we have:

$$\begin{aligned}
b_{k,l}^{p+1,q+1} &= c_{k,l-1}^{p,q} - c_{k-1,l}^{p,q} && \text{for} \quad 1 \le k \le p \wedge 1 \le l \le q && (52)\\
b_{0,l}^{p+1,q+1} &= c_{0,l-1}^{p,q} && \text{for} \quad 1 \le l \le q && (53)\\
b_{p+1,l}^{p+1,q+1} &= -c_{p,l}^{p,q} && \text{for} \quad 0 \le l \le q && (54)\\
b_{k,0}^{p+1,q+1} &= -c_{k-1,0}^{p,q} && \text{for} \quad 1 \le k \le p && (55)\\
b_{k,q+1}^{p+1,q+1} &= c_{k,q}^{p,q} && \text{for} \quad 0 \le k \le p && (56)\\
b_{0,0}^{p+1,q+1} &= b_{p+1,q+1}^{p+1,q+1} = 0 \ . && && (57)
\end{aligned}$$

In the second case we obtain:

$$\begin{aligned}
b_{k,l}^{p+1,q+1} &= a_{k,l-1}^{p+1,q} - a_{k,l}^{p+1,q} && \text{for} \quad 0 \le k \le p+1 \wedge 1 \le l \le q && (58)\\
b_{k,0}^{p+1,q+1} &= -a_{k,0}^{p+1,q} && \text{for} \quad 1 \le k \le p+1 && (59)\\
b_{k,q+1}^{p+1,q+1} &= a_{k,q}^{p+1,q} && \text{for} \quad 0 \le k \le p && (60)\\
b_{0,0}^{p+1,q+1} &= b_{p+1,q+1}^{p+1,q+1} = 0 \ . && && (61)
\end{aligned}$$

The last equation holds because $a_{0,0}^{p+1,q} = a_{p+1,q}^{p+1,q}$, and therefore these terms will telescope.

If we regard a, b, and c as vectors, the above equations define a set of linear equations

$$\mathbf{D}c = \mathbf{E}a \ , \qquad (62)$$

where $\mathbf{D}$ is a $((p+2)(q+2)-2) \times ((p+1)(q+1))$ matrix, and $\mathbf{E}$ is a $((p+2)(q+2)-2) \times ((p+2)(q+1))$ matrix. The right hand side of (62) is completely known from (58)–(60). Hence (62) defines an overdetermined system of linear equations for c. Therefore, Proposition 1 reduces to the question whether the extra $p+q+1$ equations by which (62) is overdetermined are automatically fulfilled.

In order to show this, we can pick any $(p+1)(q+1)$ equations from (62) to calculate $c_{k,l}^{p,q}$. For example, we can choose:

$$c_{k,l}^{p,q} = c_{k-1,l+1}^{p,q} + b_{k,l+1}^{p+1,q+1} \quad \text{for} \quad 1 \le k \le p \wedge 0 \le l \le q-1 \qquad (63)$$

$$
\begin{aligned}
c^{p,q}_{0,l} &= b^{p+1,q+1}_{0,l+1} && \text{for} \quad 0 \le l \le q-1 && (64)\\
c^{p,q}_{k,q} &= b^{p+1,q+1}_{k,q+1} && \text{for} \quad 0 \le k \le p \ . && (65)
\end{aligned}
$$

The first of these equations defines a recursion on previous values of $b^{p+1,q+1}_{k,l}$, for which the last two equations give the starting values. By expanding the recursion and plugging in the starting values we obtain:

$$
c^{p,q}_{k,l} = \begin{cases} \displaystyle\sum_{i=0}^{k} b^{p+1,q+1}_{k-i,l+1+i} & \text{for} \quad k < q-l \\ \displaystyle\sum_{i=0}^{q-l} b^{p+1,q+1}_{k-i,l+1+i} & \text{for} \quad k \ge q-l \ . \end{cases} \tag{66}
$$

By substituting $b^{p+1,q+1}_{k-i,l+1+i}$ with its definition according to (58)–(60), this can be transformed to:

$$
c^{p,q}_{k,l} = \begin{cases} \displaystyle\sum_{i=0}^{k} a^{p+1,q}_{k-i,l+i} - \sum_{i=0}^{k} a^{p+1,q}_{k-i,l+1+i} & \text{for} \quad k < q-l \\ \displaystyle\sum_{i=0}^{q-l} a^{p+1,q}_{k-i,l+i} - \sum_{i=0}^{q-l-1} a^{p+1,q}_{k-i,l+1+i} & \text{for} \quad k < q-l \ . \end{cases} \tag{67}
$$

In order to check whether (62) is solvable, we need to compare the values of $c^{p,q}_{k,l}$ computed according to (67) with those obtained from (55) and (54):

$$
\begin{aligned}
c^{p,q}_{k,0} &= a^{p+1,q}_{k+1,0} && \text{for} \quad 0 \le k \le p && (68)\\
c^{p,q}_{p,l} &= a^{p+1,q}_{p+1,l} - a^{p+1,q}_{p+1,l-1} && \text{for} \quad 1 \le l \le q \ . && (69)
\end{aligned}
$$

The key for this proof is to recognize that each of the two sums in (67) forms an almost complete Vandermonde convolution [2, Chapter 5]:

$$
\sum_{\substack{0 \le n-k \le r \\ 0 \le m+k \le s}} \binom{r}{n-k} \binom{s}{m+k} = \binom{r+s}{m+n} \ . \tag{70}
$$

For $c^{p,q}_{k,0}$ we need to add $-a^{p+1,q}_{k+1,0}$ and get:

$$
\begin{aligned}
c^{p,q}_{k,0} &= \sum_{i=0}^{k} a^{p+1,q}_{k-i,l+i} - \sum_{i=-1}^{k} a^{p+1,q}_{k-i,l+1+i} + a^{p+1,q}_{k+1,0} \\
&= \frac{1}{(p+1)(p+q+2)} \cdot \frac{1}{\binom{p+q+1}{k}} \sum_{i=0}^{k} \binom{p+1}{k-i} \binom{q}{i} \\
&\quad - \frac{1}{(p+1)(p+q+2)} \cdot \frac{1}{\binom{p+q+1}{k+1}} \sum_{i=-1}^{k} \binom{p+1}{k-i} \binom{q}{i+1} + a^{p+1,q}_{k+1,0} \\
&= \frac{1}{(p+1)(p+q+2)} \left(\frac{\binom{p+q+1}{k}}{\binom{p+q+1}{k}} - \frac{\binom{p+q+1}{k+1}}{\binom{p+q+1}{k+1}} \right) + a^{p+1,q}_{k+1,0} \\
&= a^{p+1,q}_{k+1,0} \ .
\end{aligned} \tag{71}
$$

This is the same value as obtained from (68). The proof for $c^{p,q}_{p,l}$ is analogous. This time, however, we need to add $a^{p+1,q}_{p+1,l-1} - a^{p+1,q}_{p+1,l}$ to get a full Vandermonde convolution. This concludes the proof of Proposition 1.

B Proof of Proposition 2

The structure of (63)–(65) suggests a proof by induction. The basis of the induction will be for $c^{p,q}_{0,l}$ $(0 \leq l \leq q-1)$ and $c^{p,q}_{k,q}$ $(0 \leq k \leq p)$. For $c^{p,q}_{k,q}$ we have:

$$\begin{aligned}
c^{p,q}_{k,q} &= a^{p+1,q}_{k,q} = \frac{1}{(p+q+2)(p+1)} \cdot \frac{\binom{p+1}{k}\binom{q}{q}}{\binom{p+q+1}{k+q}} \\
&= \frac{1}{(p+q+2)(p+1)} \cdot \frac{\frac{(p+1)!}{k!\,(p+1-k)!} \cdot \frac{q!}{0!\,q!}}{\frac{(p+q+1)!}{(k+q)!\,(p+1-k)!}} \\
&= \frac{1}{(p+q+2)(p+q+1)} \cdot \frac{p!\,q!\,(k+q)!}{(p+q)!\,k!\,q!} \\
&= \frac{1}{(p+q+2)(p+q+1)\binom{p+q}{p}} \binom{k+q}{q}\binom{p-k}{0} \quad . \qquad (72)
\end{aligned}$$

Similarly, for $c^{p,q}_{l,0}$ we have:

$$\begin{aligned}
c^{p,q}_{0,l} &= a^{p+1,q}_{0,l} - a^{p+1,q}_{0,l+1} \\
&= \frac{1}{(p+q+2)(p+1)} \cdot \frac{\binom{p+1}{0}\binom{q}{l}}{\binom{p+q+1}{l}} - \frac{1}{(p+q+2)(p+1)} \cdot \frac{\binom{p+1}{0}\binom{q}{l+1}}{\binom{p+q+1}{l+1}} \\
&= \frac{1}{(p+q+2)(p+1)} \left(\frac{\frac{(p+1)!}{0!\,(p+1)!} \cdot \frac{q!}{l!\,(q-l)!}}{\frac{(p+q+1)!}{l!\,(p+q+1-l)!}} - \frac{\frac{(p+1)!}{0!\,(p+1)!} \cdot \frac{q!}{(l+1)!\,(q-l-1)!}}{\frac{(p+q+1)!}{(l+1)!\,(p+q-l)!}} \right) \\
&= \frac{1}{(p+q+2)(p+q+1)} \left(\frac{p!\,q!\,l!\,(p+q+1-l)!}{(p+q)!\,l!\,(p+1)!\,(q-l)!} \right. \\
&\qquad \left. - \frac{p!\,q!\,(l+1)!\,(p+q-l)!}{(p+q)!\,(l+1)!\,(p+1)!\,(q-l-1)!} \right) \\
&= \frac{1}{(p+q+2)(p+q+1)\binom{p+q}{p}} \left(\frac{(p+q+1-l)!}{(p+1)!\,(q-l)!} - \frac{(p+q-l)!}{(p+1)!\,(q-l-1)!} \right)
\end{aligned}$$

$$
\begin{aligned}
&= \frac{1}{(p+q+2)(p+q+1)\binom{p+q}{p}} \cdot \frac{(p+q+1-l)! - (q-l)(p+q+l)!}{(p+1)!\,(q-l)!} \\
&= \frac{1}{(p+q+2)(p+q+1)\binom{p+q}{p}} \cdot \frac{((p+q+1-l)-(q-l))(p+q-l)!}{(p+1)p!\,(q-l)!} \\
&= \frac{1}{(p+q+2)(p+q+1)\binom{p+q}{p}} \binom{l}{l}\binom{p+q-l}{q-l}\ . \qquad (73)
\end{aligned}
$$

Now assume that the identity holds for $k-1$ and $l+1$. Then we have:

$$
\begin{aligned}
c_{k,l}^{p,q} &= c_{k-1,l+1}^{p,q} + a_{k,l}^{p+1,q} - a_{k,l+1}^{p+1,q} \\
&= \frac{1}{(p+q+2)(p+q+1)\binom{p+q}{p}} \binom{k+l}{l+1}\binom{p+q-k-l}{q-l-1} \\
&\quad + \frac{1}{(p+q+2)(p+1)} \left(\frac{\binom{p+1}{k}\binom{q}{l}}{\binom{p+q+1}{k+l}} - \frac{\binom{p+1}{k}\binom{q}{l+1}}{\binom{p+q+1}{k+l+1}} \right) \\
&= \frac{1}{p+q+2} \left(\frac{1}{p+q+1} \frac{\frac{(k+l)!}{(l+1)!\,(k-1)!} \cdot \frac{(p+q-k-l)!}{(q-l-1)!\,(p+1-k)!}}{\frac{(p+q)!}{p!\,q!}} \right. \\
&\qquad\qquad + \frac{1}{p+1} \frac{\frac{(p+1)!}{k!\,(p+1-k)!} \cdot \frac{q!}{l!\,(q-l)!}}{\frac{(p+q+1)!}{(k+l)!\,(p+q+1-k-l)!}} \\
&\qquad\qquad \left. - \frac{1}{p+1} \frac{\frac{(p+1)!}{k!\,(p+1-k)!} \cdot \frac{q!}{(l+1)!\,(q-l-1)!}}{\frac{(p+k+1)!}{(k+l+1)!\,(p+q-k-l)!}} \right) \\
&= \frac{1}{(p+q+2)(p+q+1)} \left(\frac{p!\,q!\,(k+l)!\,(p+q-k-l)!}{(p+q)!\,(l+1)!\,(k-1)!\,(q-l-1)!\,(p+1-k)!} \right. \\
&\qquad\qquad + \frac{p!\,q!\,(k+l)!\,(p+q+1-k-l)!}{(p+q)!\,k!\,(p+1-k)!\,l!\,(q-l)!} \\
&\qquad\qquad \left. - \frac{p!\,q!\,(k+l+1)!\,(p+q-k-l)!}{(p+q)!\,k!\,(p+1-k)!\,(l+1)!\,(q-l-1)!} \right) \\
&= \frac{1}{(p+q+2)(p+q+1)\binom{p+q}{p}} \left(\frac{k(q-l)(k+l)!\,(p+q-k-l)!}{(l+1)!\,k!\,(q-l)!\,(p+1-k)!} \right.
\end{aligned}
$$

$$
\begin{aligned}
&\qquad\qquad + \frac{(l+1)(k+l)!\,(p+q+1-k-l)!}{(l+1)!\,k!\,(q-l)!\,(p+1-k)!} \\
&\qquad\qquad - \frac{(q-l)(k+l+1)!\,(p+q-k-l)!}{(l+1)!\,k!\,(q-l)!\,(p+1-k)!}\Bigg) \\
= \quad & \frac{1}{(p+q+2)(p+q+1)\binom{p+q}{p}} \times \\
& \Bigg(\frac{k(q-l)+(l+1)(p+q+1-k-l)-(q-l)(k+l+1)}{(l+1)(p+1-k)} \times \\
& \frac{(k+l)!\,(p+q-k-l)!}{k!\,l!\,(q-l)!\,(p-k)!}\Bigg) \\
= \quad & \frac{1}{(p+q+2)(p+q+1)\binom{p+q}{p}} \binom{k+l}{l} \binom{p+q-k-l}{q-l} \ . \qquad (74)
\end{aligned}
$$

References

[1] I. N. Bronstein and K. A. Semendjajew. *Taschenbuch der Mathematik*. B. G. Teubner Verlagsgesellschaft, Leipzig, 22nd edition, 1985.

[2] Ronald L. Graham, Donald E. Knuth, and Oren Patashnik. *Concrete Mathematics*. Addison-Wesley Publishing Company, Reading, MA, USA, 2nd edition, 1994.

[3] Robert M. Haralick and Linda G. Shapiro. *Computer and Robot Vision*, volume I. Addison-Wesley Publishing Company, Reading, MA, USA, 1992.

[4] X. Y. Jiang and H. Bunke. Simple and fast computation of moments. *Pattern Recognition*, 24(8):801–806, 1991.

[5] Stefan Lanser, Olaf Munkelt, and Christoph Zierl. Robust video-based object recognition using CAD models. In U. Rembold, R. Dillmann, L.O. Hertzberger, and T. Kanade, editors, *Intelligent Autonomous Systems IAS-4*, pages 529–536. IOS Press, 1995.

[6] Stefan Lanser and Christoph Zierl. MORAL: Ein System zur videobasierten Objekterkennung im Kontext autonomer, mobiler Systeme. In G. Schmidt and F. Freyberger, editors, *Autonome Mobile Systeme*, Informatik aktuell. Springer-Verlag, 1996.

[7] Stefan Lanser and Christoph Zierl. On the use of topological constraints within object recognition tasks. In *13th International Conference on Pattern Recognition*, 1996.

[8] Stefan Lanser, Christoph Zierl, and Roland Beutlhauser. Multibildkalibrierung einer CCD-Kamera. In G. Sagerer, S. Posch, and F. Kummert, editors, *Mustererkennung 1995*, Informatik aktuell, pages 481–491. Springer-Verlag, 1995.

[9] Mark H. Singer. A general approach to moment calculation for polygons and line segments. *Pattern Recognition*, 26(7):1019–1028, 1993.

[10] Carsten Steger. Extracting curvilinear structures: A differential geometric approach. In Bernard Buxton and Roberto Cipolla, editors, *Fourth European Conference on Computer*

Vision, volume 1064 of *Lecture Notes in Computer Science*, pages 630–641. Springer-Verlag, 1996.

[11] N. J. C. Strachan, P. Nesvadba, and A. R. Allen. A method for working out the moments of a polygon using an integration technique. *Pattern Recognition Letters*, 11:351–354, May 1990.

[12] Herbert Suesse and Klaus Voss. Fitting von Objekten durch Super-Qadriken. In Bernd Jähne, Peter Geißler, Horst Haußecker, and Frank Hering, editors, *Mustererkennung*, Informatik aktuell, pages 29–36. Springer-Verlag, 1996.

[13] K. Voss, H. Suesse, and R. Neubauer. Moment-based invariant fitting of elliptical segments. In Václav Hlaváč and Radim Šára, editors, *6th International Conference on Computer Analysis of Images and Patterns*, volume 970 of *Lecture Notes in Computer Science*, pages 562–567. Springer-Verlag, 1995.

[14] Klauss Voss and Herbert Süße. *Adaptive Modelle und Invarianten für zweidimensionale Bilder*. Verlag Shaker, Aachen, 1995.