Journal of Computational and Applied Mathematics 306 (2016) 53–68

Contents lists available at ScienceDirect

Journal of Computational and Applied Mathematics

journal homepage: www.elsevier.com/locate/cam

Numerical problems with the Pascal triangle in moment computation

Jaroslav Kautsky [a], Jan Flusser [b,*]

[a] The Flinders University of South Australia, CSEM, GPO Box 2100, Adelaide, SA 5001, Australia

[b] Institute of Information Theory and Automation of the AS CR, Pod Vodárenskou věží 4, 182 08 Prague 8, Czech Republic

ARTICLE INFO

Article history:
Received 1 February 2015
Received in revised form 23 February 2016

Keywords:
Stable calculations
Polynomials
Moments
Pascal triangle
Orthogonal moments
Moment invariants

ABSTRACT

Moments are important characteristics of digital signals and images and are commonly used for their description and classification. When calculating the moments and their derived functions numerically, we face, among other numerical problems studied in the literature, certain instabilities which are connected with the properties of Pascal triangle. The Pascal triangle appears in moment computation in various forms whenever we have to deal with binomial powers. In this paper, we investigate the reasons for these instabilities in three particular cases—central moments, complex moments, and moment blur invariants. While in the first two cases this phenomenon is tolerable, in the third one it causes serious numerical problems. We analyze these problems and show that they can be partially overcome by choosing an appropriate polynomial basis.

1. Introduction

Moments are scalar quantities which have been used for more than hundred years to characterize a (possibly multidimensional) function and to capture its significant features. They have been widely used in statistics for the description of the shape of a probability density function and in classic rigid-body mechanics to measure the mass distribution of a body. From the mathematical point of view, moments are projections of a function onto a polynomial basis (in the same way that the Fourier transform is a projection onto a basis of harmonic functions).

Moments were first introduced to the pattern recognition and image processing community in 1962, when Hu [1] employed the results of the classical theory of algebraic invariants [2–5] and derived the first moment-based characteristics (features) suitable for object description and recognition. Since that time, this field has undergone significant development. The study of moments has formed a well-established area of image recognition with thousands of relevant papers and several survey monographs [6–10] and has become one of the most frequently used features in image analysis.

A general definition of a moment in d dimensions is as follows. Let $\{b_{\mathbf{k}}(\mathbf{x})\}$ be a d-variable polynomial basis of the space of image functions defined on $D \subset \mathbb{R}^d$ and let $\mathbf{k} = (k_1, \ldots, k_d)$ be a multi-index of non-negative integers which show the highest power of the respective variables in $b_{\mathbf{k}}(\mathbf{x})$. Then the *general moment* $M_{\mathbf{k}}^{(f)}$ of image f is defined as

$$M_{\mathbf{k}}^{(f)} = \int_D b_{\mathbf{k}}(\mathbf{x}) f(\mathbf{x}) \mathrm{d}\mathbf{x}. \tag{1}$$

The number $|\mathbf{k}| = \sum_{j=1}^{d} k_j$ is called the *order* of the moment. We omit the superscript (f) when there is no ambiguity.

[*] Corresponding author.
E-mail addresses: Jerry.Kautsky@flinders.edu.au (J. Kautsky), flusser@utia.cas.cz (J. Flusser).

http://dx.doi.org/10.1016/j.cam.2016.03.033
0377-0427/© 2016 Elsevier B.V. All rights reserved.

The properties of the moments depend on the choice of the basis $\{b_{\mathbf{k}}(\mathbf{x})\}$. The most common choice is a standard power basis

$$b_{\mathbf{k}}(\mathbf{x}) = \mathbf{x}^{\mathbf{k}} = x_1^{k_1} \dots x_d^{k_d}$$

which leads to *geometric moments*

$$m_{\mathbf{k}} = \int_D \mathbf{x}^{\mathbf{k}} f(\mathbf{x}) \mathrm{d}\mathbf{x}. \tag{2}$$

In addition to that, numerous orthogonal bases have been used in the literature (see [8] for a survey).

The moments themselves are rarely used directly for object recognition because they depend on the position, orientation, size, and many other variations of the object. To overcome this, one can create certain functions of moments which stay constant under certain group of transformations. These functions are called *moment invariants*. A number of various moment invariants have been reported in the literature—rotation invariants, similarity invariants, affine invariants, and blur invariants are the most important ones (see [8] for a survey and detailed forms of individual invariants).

Although the theory of all moment invariants has been developed in a continuous domain because of the comfortable mathematical tools available, our practical interest is in the domain of discrete (digital) signals and images. The transition from the continuous to the discrete domain entails an approximation of the integral in (1) by a sum. In the simplest case, a zero-order approximation is used and (1) turns to the form (we show a 1D case for simplicity)

$$M_k = \sum_{j=1}^{N} b_k(x_j) f(x_j) \tag{3}$$

where the evaluation points are commonly chosen $x_j = j$. Regardless of what approximation scheme has been applied, this transition always induces errors of several kinds. Some of them are firmly connected with the sampling and quantization errors of the image and have been thoroughly analyzed for instance in [7].

Another group of errors is connected with the numerical evaluation of a sum such as that in (3) and originates from the finite precision of the computer arithmetic. Although they might be by several orders higher than the sampling errors (especially if an unstable numerical algorithm has been applied), they have not been systematically studied and fully explored yet. Most papers on numerical moment calculations have been focused on fast algorithms rather than on error analysis [11,12]. If the authors had observed some instability in the experiments, they mostly explained it as a moment sensitivity to additive noise in the image or as a consequence of using very large values when working with the powers. They tried, with partial success, to overcome these problems by changing the polynomial basis to Legendre [13], Zernike [13,14], Pseudo-Zernike [15], Gauss–Hermite [16], Chebyshev [17], Krawtchouk [18], and other special bases [19,20].

In this paper we investigate a specific source of errors: the poor condition of a Pascal matrix (11). We show that the Pascal matrix (and forms derived from it) often appears when evaluating polynomial moments and moment invariants. We demonstrate that in some cases it introduces serious numerical errors while in other, seemingly very similar, cases its impact is much less significant. This aspect of moment calculations has never been investigated and errors of this kind have, in the past, been misinterpreted.

The paper is organized as follows. In Section 2 we recall three important classes of moments, and their invariants, where Pascal-like matrices appear in their evaluation. Section 3 presents a general overview of numerical algebra on the given problem. In Section 4 we briefly discuss the importance of choosing the suitable domain for the calculation of geometric moments (that is, choosing x_j in (3)), then we study how the Pascal triangle influences the transformations between geometric, central and complex moments. We describe the effect of combining the Pascal triangle with a Toeplitz matrix carrying the centroid information and, finally, we show that to calculate complex moments from the geometric moments, a 2D version of Pascal triangle leads to better numerical accuracy than might be expected. The stability of all these calculations can be studied by elementary means, that is through the numerical condition of the linear transforms involved. On the other hand, the evaluation of *blur invariants* leads to a numerical misbehavior of such magnitude that cannot be explained just by the numerical condition of the underlying linear system. In Section 5 we demonstrate, analyze and explain this instability, caused inherently by the Pascal triangle. We demonstrate that the error of this kind is actually very serious because it renders higher order moments useless for the purpose of discriminating between blurred images. We show how it can be partially overcome. This is the main contribution of the paper. Section 6 contains a summary and discussion.

2. Central moments, complex moments and blur invariants

In this section, we introduce three examples of derived moments, where various versions of the Pascal triangle appear. The influence of the Pascal triangle on the numerical calculations will be investigated later in the paper. We limit ourselves to the 1D and 2D cases for the sake of simplicity. The extension into 3D or even to higher dimension (in case of blur invariants) is, in principle, straightforward.

The most common moments are those with respect to the basis composed of the power monomials $\{x^p\}$ in 1D and $\{x^p y^q\}$ in 2D. They are called *geometric moments* and in the 2D continuous domain are defined as

$$m_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q f(x, y) dxdy. \tag{4}$$

In signal analysis, they are not used directly because they do not exhibit any desirable invariant property. However, numerous invariants can easily be derived from them (see [8] for a detailed survey). The invariance to translation is readily achieved by shifting the power basis into the "centroid" of the signal, which leads to so-called *central moments*

$$\mu_{pq} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} (x - x_c)^p (y - y_c)^q f(x, y)dxdy, \tag{5}$$

where

$$x_c = m_{10}/m_{00}, \qquad y_c = m_{01}/m_{00}$$

are the coordinates of the centroid. The central moments can easily be expressed in terms of geometric moments as

$$\mu_{pq} = \sum_{k=0}^{p}\sum_{j=0}^{q} \binom{p}{k}\binom{q}{j} (-1)^{k+j} x_c^k y_c^j m_{p-k,q-j}.$$

This relation is sometimes used when calculating central moments by means of a fast algorithm designed to calculate geometric moments. Note that the summation ranges over geometric moments of all pairs of indices $k \le p, j \le q$.

Achieving rotational invariance is more difficult. Among several published methods, probably the most straightforward is the one using *complex moments* [21], which are projections of the signal onto the basis $\pi_{kj}(x, y) = (x + iy)^k (x - iy)^j$ (complex moments do not exist in 1D), where i is the imaginary unit

$$c_{pq} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} (x + iy)^p (x - iy)^q f(x, y)dxdy. \tag{6}$$

Geometric moments and complex moments carry the same amount of information in the following sense. Each complex moment can be expressed in terms of geometric moments *of the same order* (which is now the sum of the indices) as

$$c_{pq} = \sum_{k=0}^{p}\sum_{j=0}^{q} \binom{p}{k}\binom{q}{j} (-1)^{q-j} \cdot i^{p+q-k-j} \cdot m_{k+j,p+q-k-j} \tag{7}$$

and *vice versa*

$$m_{pq} = \frac{1}{2^{p+q} i^q} \sum_{k=0}^{p}\sum_{j=0}^{q} \binom{p}{k}\binom{q}{j} (-1)^{q-j} \cdot c_{k+j,p+q-k-j}. \tag{8}$$

Under image rotation, complex moments preserve their magnitude and change only the phase as can be seen from their transformation into polar coordinates

$$c_{pq} = \int_{0}^{\infty}\int_{0}^{2\pi} r^{p+q+1} e^{i(p-q)\theta} f(r, \theta)drd\theta. \tag{9}$$

Although (9) might seem more convenient for numerical evaluation, one often uses (7) due to the availability of fast algorithms for geometric moment calculation. Hence, this leads to dealing again with a kind of Pascal triangle.

The last of our examples where the Pascal triangle appears is the set of invariants with respect to a convolution of a signal with a symmetric kernel function. These *blur invariants* have been used in signal analysis to identify blurred signals without the need for deconvolution. Various blur invariants have been published in the literature. Particular form of the invariant depends on the kind of the symmetry of the kernel [22]. However, the Pascal-like matrices appear in all of them and their impact does not depend on the particular form of the invariant. For the sake of simplicity, we limit our analysis to the 1D case, where the only existing symmetry of the kernel is the central symmetry.

The first formula for moment blur invariants in 1D was published in [23]

$$S_p = \mu_p - \frac{1}{\mu_0} \sum_{n=1}^{(p-1)/2} \binom{p}{2n} S_{p-2n} \cdot \mu_{2n} \tag{10}$$

(p being odd). In [24] we showed that blur invariants are in fact a solution of a linear algebraic system depending on the moments and derived the relation between blur invariants with respect to two different polynomial bases, particularly between standard powers (leading to geometric moments) and orthogonal polynomials. We also gave general advice on how to implement such transformation to obtain the benefits of the more numerically stable orthogonal polynomials. In this case, the Pascal triangle enters the computation in a way different from that of the two previous examples.

It is interesting to observe that the level of instability caused by the Pascal triangle depends significantly on the particular form in which it appears in the calculations. While in the case of central and complex moments this instability does not affect the calculated values significantly (Section 4), it leads to serious problems in the case of blur invariants (Section 5).

3. Numerical linear algebra viewpoint

As we pointed out in the previous section, calculation with discrete moments may lead to large linear systems and care should be exercised in solving them. It is well understood, for example, that when solving the linear algebraic system $A\boldsymbol{x} = \boldsymbol{b}$, the relative error of the solution $\boldsymbol{x}$ is bounded by the relative error of the data ($\boldsymbol{b}$, A and the errors of calculation) multiplied by the *condition of the matrix*, $\mathrm{cond}(A) = \|A\|\,\|A^{-1}\|$ ($\|A\|$ is the matrix norm induced by the l_2 vector norm used in this paper). As this condition is the same for the matrix inverse the bound applies not only to *solving* of the system but to applying the linear transform as well (calculating $\boldsymbol{b}$ for a given $\boldsymbol{x}$). While the bound may, in particular cases, be pessimistic the matrix condition is still an important indicator of possible numerical problems despite a sometimes popular belief that calculations with explicit formulae, such as forward- or back-substitution, are inherently stable. Another source of numerical instability, which may appear in moment computation, is calculation with polynomials represented through their power expansions. This is because the powers of higher degrees are numerically nearly linearly dependent regardless of the fact that they are actually linearly independent. In discretized situations we can form a matrix of these basis functions at, say, equidistant points and observe the condition of such a matrix. Replacing the power basis by orthogonal polynomials and, possibly, choosing suitable points for the evaluation would significantly improve the condition of the discrete approximations. Polynomials have the remarkable (but obvious) property that a polynomial of a sum $x + y$ is again a polynomial, either in x (with coefficients depending on y) or in y (with coefficients depending on x). Whenever this is exploited (as in the calculation of certain moments derived from the basic geometric moments, in the evaluation of some invariants, etc.) it leads to linear algebraic systems involving binomial coefficients which can be gathered into a lower triangular matrix, the Pascal triangle

$$P_n = \left(\binom{j-1}{k-1}\right)_{j,k=1}^{n} = \begin{pmatrix} 1 & 0 & 0 & 0 & \\ 1 & 1 & 0 & 0 & \vdots \\ 1 & 2 & 1 & 0 & \\ 1 & 3 & 3 & 1 & \\ & & \dots & & \ddots \end{pmatrix}. \tag{11}$$

The condition of P increases rapidly with its size – see Fig. 2 – which also leads to numerical instability of calculations. It would appear that making sure the condition of any matrix involved in our calculations and choosing the right polynomial basis (if relevant) should lead to numerically safe methods. We will demonstrate that this is true in some cases but not in others.

4. Numerical aspects of calculating geometric, central and complex moments

4.1. Geometric moments

Regardless of what interpretation we associate with a discretized image we approximate the integrals defining the moments using a midpoint quadrature formula with equidistant knots $x_1, x_2, \ldots$. The evaluation of geometric moments (4) can be written as

$$M_{\boldsymbol{g}} = \gamma Z_{\boldsymbol{x}} F Z_{\boldsymbol{y}}^T \tag{12}$$

where $M_{\boldsymbol{g}}$ is a $r \times r$ matrix of moments m_{pq} of orders at most r, F is an $n_x \times n_y$ matrix of the discretized image and, with powers as the polynomial basis,

$$Z_{\boldsymbol{x}} = \left(x_k^{j-1}\right)_{j=1,k=1}^{j=r,k=n_x}$$

is a $r \times n_x$ matrix and similarly for $Z_{\boldsymbol{y}}$.

The scaling factor γ depends on the interpretation of the physical aspects of the discretization and is essentially irrelevant. What is more relevant is the choice of the points x_k which influences the numerical condition of matrices $Z_{\boldsymbol{x}}$ and $Z_{\boldsymbol{y}}$.

It is not unusual to use simply $x_k = k$. This may lead to overflow for higher degrees of powers but it also leads to extremely badly conditioned matrices $Z_{\boldsymbol{x}}$ as shown in Fig. 1 where the testing image was an image with $n_x = n_y = 256$. Shifting the origin improves the situation a little, as explained in the next section. The reason for the very bad conditioning is that we are using the powers for very large arguments where they are indeed computationally nearly linearly dependent.

A much better strategy is, as when using orthogonal polynomials, to use a symmetric fixed length interval $[-a, a]$ for some $a > 0$. The choice $a = 1$ is standard here but we propose $a = 2$ as a good compromise between underflow and overflow for very high degrees. The general formula for the knots of mid-point quadrature with equal weights is

$$x_k = \frac{2k - n - 1}{n} a, \quad k = 1, 2, \ldots, n.$$

The conditions of matrices $Z_{\boldsymbol{x}}$ are shown in Fig. 2.

We do not claim that using the power basis over the suggested domain resolves all the problems of numerical instability but wish only to point out the drawbacks of an approach that is, unfortunately, often used.

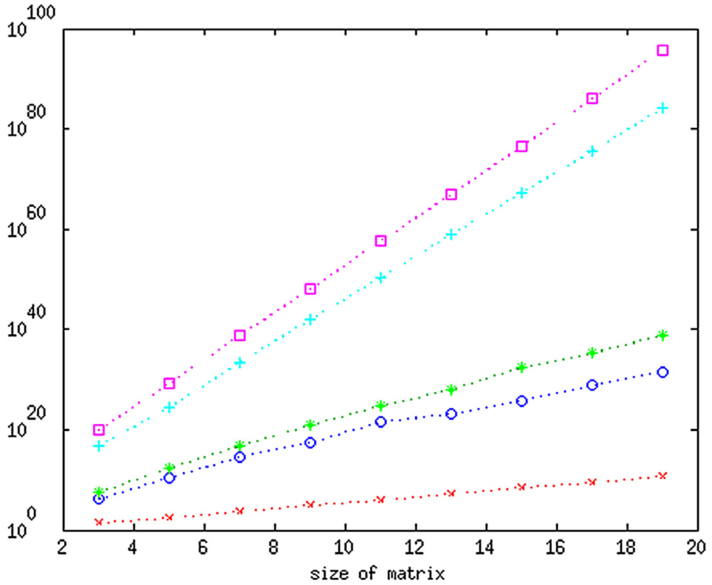

Fig. 1. Conditions of some matrices of size r, $2 < r < 20$: discretized powers Z_x on $[1, 256]$ (green $*$), same shifted to centroid origin (blue $\circ$) and Pascal P_r (red $\times$). Also $\|M_c\|$ (cyan $+$) and $\|M_g\|$ (mauve $\square$). The centroid was $[137, 124]$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

4.2. Central geometric moments

As discussed in Section 2, it is important to be able to calculate geometric moments with respect to a given coordinate origin. Regardless of how that origin is chosen we are interested in the numerical properties of such a shift.

Given the new origin $[x_c, y_c]$ the *central* moments are like those in (4) with (x, y) replaced by $[x - x_c, y - y_c]$. In the discretized form this leads to

$$\mu_{pq} = \sum_{k=0}^{p} \binom{p}{k} (-x_c)^k \sum_{j=0}^{q} \binom{q}{j} (-y_c)^j m_{p-k,q-j}. \tag{13}$$

To study the numerical properties of such a conversion we will express it in matrix form. We note that the summation of the j-factors acts on the rows of the matrix $M_{\boldsymbol{g}}$ of the geometric moments m_{pq} and the other sum acts on its columns.

By $T(\boldsymbol{v})$ we denote the lower triangular Toeplitz matrix with the vector $\boldsymbol{v}$ as its first column, that is

$$T(\boldsymbol{v}) = \begin{pmatrix} v_1 & 0 & 0 & 0 & \\ v_2 & v_1 & 0 & 0 & \vdots \\ v_3 & v_2 & v_1 & 0 & \\ v_4 & v_3 & v_2 & v_1 & \\ & & \dots & & \ddots \end{pmatrix}.$$

We also denote

$$G_{\boldsymbol{x}} = P_r . * T\left(\left(x_c^0 \quad x_c^1 \quad x_c^2 \quad \cdots \quad x_c^{r-1}\right)\right)$$

the Hadamard (element-wise) product of the Pascal matrix and the Toeplitz matrix with the powers of x_c, and similarly for $G_{\boldsymbol{y}}$. We then have, for the matrix of central moments,

$$M_{\boldsymbol{c}} = G_{\boldsymbol{x}} M_{\boldsymbol{g}} G_{\boldsymbol{y}}^T. \tag{14}$$

The numerical accuracy of this conversion is thus governed by the conditions of the matrices $G_{\boldsymbol{x}}$ and $G_{\boldsymbol{y}}$ which now depend on the data through the Toeplitz matrices T with powers of the coordinates of the new origin. The smaller these coordinates are, the closer $G_{\boldsymbol{x}}$ and $G_{\boldsymbol{y}}$ will be to P_r, the condition of which is shown in Figs. 1 and 2.

The ratio of the relative error of the solution of a linear system to the relative error of the input is bounded by the condition number of the matrix of the transformation (see, e.g., Theorem 2.5-1 in [25]).

To demonstrate this we calculated both geometric $M_{\boldsymbol{g}}$ and central $M_{\boldsymbol{c}}$ moments from first principal and then calculated

$$\widetilde{M}_{\boldsymbol{c}} = G_{\boldsymbol{x}} M_{\boldsymbol{g}} G_{\boldsymbol{y}}^T \quad \text{and} \quad \widetilde{M}_{\boldsymbol{g}} = G_{\boldsymbol{x}}^{-1} M_{\boldsymbol{c}} (G_{\boldsymbol{y}}^T)^{-1}$$

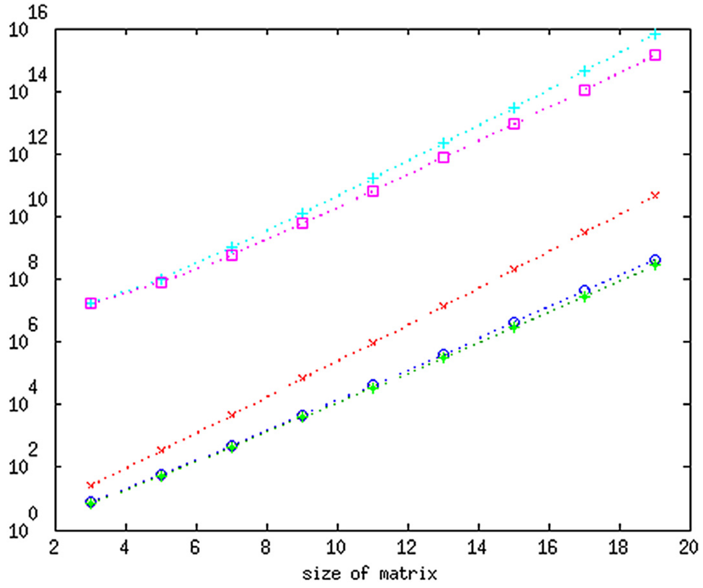

Fig. 2. Conditions of some matrices of size r, $2 < r < 20$: discretized powers Z_x on $[-2, 2]$ (green $*$), same shifted to centroid origin (blue $\circ$) and Pascal P_r (red $\times$). Also $\|M_c\|$ (cyan $+$) and $\|M_g\|$ (mauve $\Box$). The centroid was $[0.14, -0.0674]$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

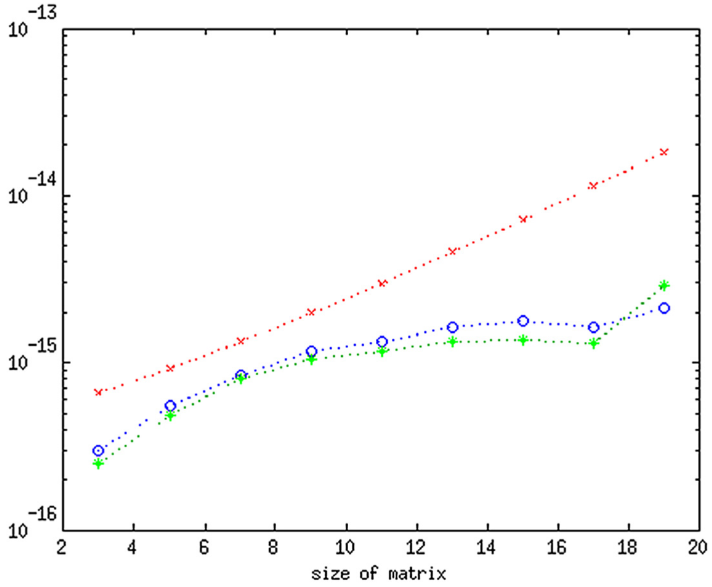

Fig. 3. Conditioning of the conversion between geometric and complex moments. Relative errors in calculating M_c from M_g (blue $\circ$) and *vice versa* (green $*$). Also the error bound $(\text{cond}(G_x) + \text{cond}(G_y))\varepsilon$ (red $\times$), ε is the machine precision. The magnitudes of $\|M_c\|$ and $\|M_g\|$, which divide the relative errors, can be observed in Fig. 2. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

using (14) both ways. The relative errors

$$\frac{\|\widetilde{M}_{\boldsymbol{c}} - M_{\boldsymbol{c}}\|}{\|M_{\boldsymbol{c}}\|} \quad \text{and} \quad \frac{\|\widetilde{M}_{\boldsymbol{g}} - M_{\boldsymbol{g}}\|}{\|M_{\boldsymbol{g}}\|}$$

are shown in Fig. 3, together with their upper bound

$$(\text{cond}(G_{\boldsymbol{x}}) + \text{cond}(G_{\boldsymbol{y}}))\varepsilon$$

which, according to the theorem mentioned above, bounds the calculated relative errors of the solutions because $\varepsilon = 10^{-16}$, the machine precision, represents the relative error of the input.

Note that it would be misleading to conclude here that the calculations have the same accuracy regardless of the size of the systems—it is the *relative* accuracy we are observing here; the quantities we are calculating depend very much on the size of the system as shown in Fig. 2 for comparison.

4.3. Complex moments

The Pascal triangle is hidden in (7) in two interlacing ways. We now need to characterize the linear transforms involved in this transformation from geometric to complex moments.

Defining the sum of powers $r = p + q$ to be the *order* of the moments m_{pq} and c_{pq} we see, from (7) and (8), that complex moments of order r are fully determined by the geometric moments of order r and *vice versa*. This mapping is described in the following lemma.

Lemma 4.1. *Denote* $\mathbf{c}_r = \begin{pmatrix} c_{0r} & c_{1r-1} & \cdots & c_{r0} \end{pmatrix}^T$ *the vector of complex moments of order r and similarly* $\mathbf{m}_r$ *for geometric moments. Then*

$$\mathbf{c}_r = A_r \mathbf{m}_r \tag{15}$$

where

$$A_r = D_{pm} B_r D_{pm} D_i.$$

Here $D_{pm} = \operatorname{diag}\left(\begin{pmatrix} 1 & -1 & 1 & \cdots \end{pmatrix}\right)$, $D_i = \operatorname{diag}\left(\begin{pmatrix} i^r & i^{r-1} & \cdots & i^0 \end{pmatrix}\right)$ *and*

$$(B_r)_{pj} = \sum_{k=\max(0,p+j-r)}^{\min(p,j)} (-1)^k \binom{p}{k} \binom{r-p}{j-k}.$$

Proof. In (7), shift j by $-k$, switch the order of summations and use $q = r - p$ for $p = 0, 1, \ldots, r$. □

Matrices B_r have the same condition as A_r (the diagonal matrices D_{pm} and D_i are unitary) and are (unlike A_r) real and have a simpler sign pattern. For example, $B_0 = 1$, $B_1 = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$,

$$B_2 = \begin{pmatrix} 1 & 2 & 1 \\ 1 & 0 & -1 \\ 1 & -2 & 1 \end{pmatrix}, \qquad B_3 = \begin{pmatrix} 1 & 3 & 3 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -3 & 3 & -1 \end{pmatrix},$$

$$B_4 = \begin{pmatrix} 1 & 4 & 6 & 4 & 1 \\ 1 & 2 & 0 & -2 & -1 \\ 1 & 0 & -2 & 0 & 1 \\ 1 & -2 & 0 & 2 & -1 \\ 1 & -4 & 6 & -4 & 1 \end{pmatrix}$$

and so on. It is not difficult to observe that the first r rows of B_r can be obtained from the rows of B_{r-1} by the same summation rule as their first rows satisfy in the Pascal triangle. The last row of B_r is just the first one with alternating signs. So $B_0, B_1, B_2, \ldots$ form a *Pascal pyramid*.

Numerical calculation shows that for larger r (greater than 6) we have

$$\operatorname{cond}(A_r) = \operatorname{cond}(B_r) \approx 0.588 \cdot 10^{0.287r},$$

obtained by a least-square fit to the data. We suggest that this is an acceptable growth, particularly in comparison with the condition of the Pascal triangle—see Fig. 4 to compare them. For example, $\operatorname{cond}(B_{20}) \approx 13 \cdot 10^3$ suggesting that in IEEE double precision about 10 decimal places in the solution will be reliable.

To confirm these bounds on particular cases is not straightforward because exact results are not easily available. Calculating complex moments numerically involves, both in Cartesian or polar coordinates, the same powers causing inaccuracies. Both geometric and complex moments were evaluated for a simple binary image (X shape rotated by 45°, see Fig. 5) using explicit formulae and a high accuracy quadrature. It appears that the error of using (15) to calculate complex moments from the geometric ones is, in this case, in fact negligible.

5. Numerical instability of blur invariants

5.1. Background

Denote by $\boldsymbol{w}(t) = \begin{pmatrix} 1 & t & t^2 & \cdots & t^{n-1} \end{pmatrix}^T$ the vector of standard powers and also recall (11), the lower triangular Pascal matrix P with rows containing the binomial coefficients. The geometric moments

$$\boldsymbol{\nu} = \begin{pmatrix} \nu_0 & \nu_1 & \nu_2 & \cdots & \nu_{n-1} \end{pmatrix}^T = \int g(t)\boldsymbol{w}(t)dt$$

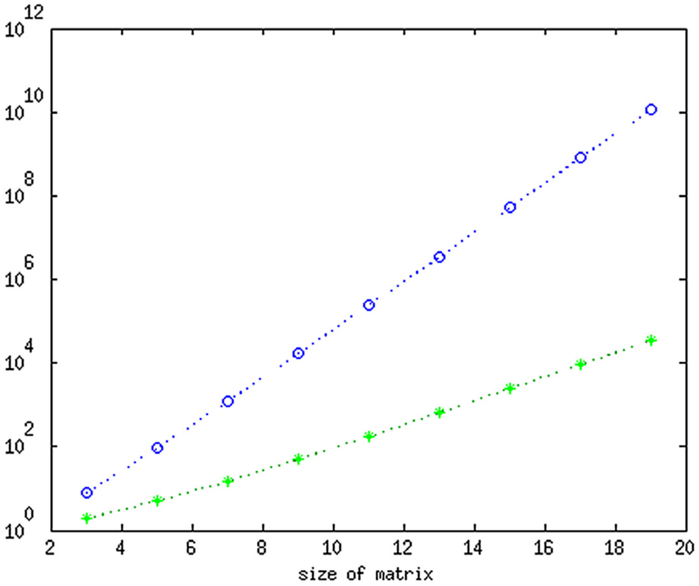

Fig. 4. Conditions of the Pascal matrix P_r (blue ∘) and of the layers of the Pascal pyramid A_r (green ∗). The smaller condition of the latter confirms the reasonable stability of the transform between geometric and complex moments. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

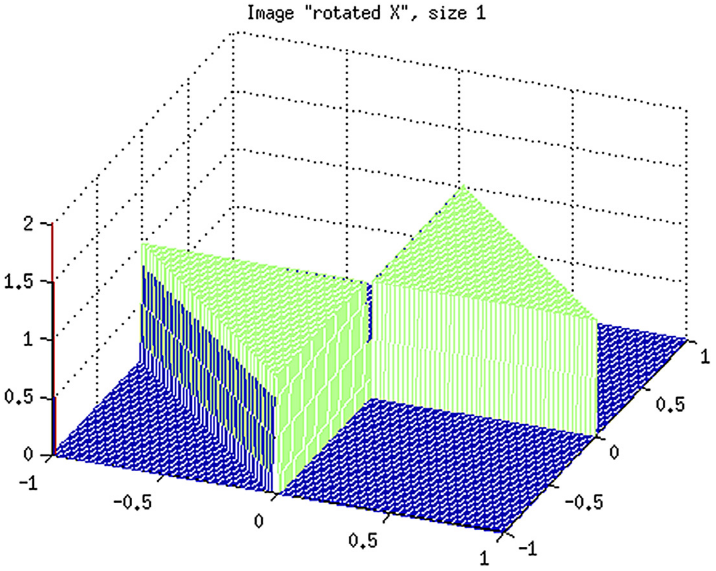

Fig. 5. Binary image “rotated X” for which the geometric and complex moments were calculated from first principles.

of the function

$$g(t) = (f * h)(t) = \int f(\tau)h(t-\tau)d\tau$$

which is a convolution of functions $f(t)$ and $h(t)$ satisfy

$$M(\boldsymbol{\mu})\boldsymbol{\pi} = \boldsymbol{\nu} \tag{16}$$

where

$$\boldsymbol{\mu} = \int f(\tau)\boldsymbol{w}(\tau)d\tau \quad \text{and} \quad \boldsymbol{\pi} = \int h(\tau)\boldsymbol{w}(\tau)d\tau$$

are the vectors of moments of the functions f and h, and the matrix

$$M(\boldsymbol{\mu}) = \left(\mu_{j-k}\binom{j-1}{k-1}\right)_{j,k=1}^{n} = \begin{pmatrix} \mu_0 & 0 & 0 & \\ \mu_1 & \mu_0 & 0 & \vdots \\ \mu_2 & 2\mu_1 & \mu_0 & \\ & \dots & & \ddots \end{pmatrix} = P. * T(\boldsymbol{\mu}) \tag{17}$$

is the Hadamard product of the Pascal matrix P and a lower triangular Toeplitz matrix $T(\boldsymbol{\mu})$ with first column $\boldsymbol{\mu}$.

Assume that n is even and denote by $\boldsymbol{w}_e(t)$ the vector of even powers, by $\boldsymbol{w}_d(t)$ the vector of odd powers and similarly for the vectors of moments. Consider now the linear system of $n/2$ equations in the unknown $\boldsymbol{\beta}(\boldsymbol{\mu})$

$$C(\boldsymbol{\mu}_e)\boldsymbol{\beta}(\boldsymbol{\mu}) = \mu_0\boldsymbol{\mu}_d \tag{18}$$

where the matrix

$$C(\boldsymbol{\mu}_e) = \left(\mu_{2(j-k)}\binom{2j-1}{2k-1}\right)_{j,k=1}^{n/2} = \begin{pmatrix} \mu_0 & 0 & 0 & \\ 3\mu_2 & \mu_0 & 0 & \vdots \\ 5\mu_4 & 10\mu_2 & \mu_0 & \\ & \dots & & \ddots \end{pmatrix} = P_{dd}. * T(\boldsymbol{\mu}_e)$$

is the Hadamard product of P_{dd}, the even numbered (that is, corresponding to odd powers) rows and columns of the Pascal matrix P, and a lower triangular Toeplitz matrix $T(\boldsymbol{\mu}_e)$ with first column $\boldsymbol{\mu}_e$. If the filter $h(t)$ is symmetric (so that $\boldsymbol{\pi}_d = \mathbf{0}$) it has been shown [23] that the solution $\boldsymbol{\beta}(\boldsymbol{\mu})$ of (18) is the vector of blur invariants so that

$$\boldsymbol{\beta}(\boldsymbol{\mu}) = \boldsymbol{\beta}(\boldsymbol{\nu})$$

for any symmetric PSF $h(t)$. This can be also shown by a fairly straightforward matrix calculation from (16).

In [24] it was also shown how to derive similar results for moments with respect to any polynomial basis $\boldsymbol{p}(t) = L\boldsymbol{w}(t)$, L a non-singular, usually lower triangular, matrix. Assuming the new basis is symmetric (L is checkerboard) and denoting $\tilde{\boldsymbol{\mu}} = L\boldsymbol{\mu}$ the moments with respect to the new basis (and similarly for $\tilde{\boldsymbol{\nu}}$, $\tilde{\boldsymbol{\pi}}$), the equations determining the invariants becomes

$$\widetilde{C}(\tilde{\boldsymbol{\mu}}_e)\tilde{\boldsymbol{\beta}}(\tilde{\boldsymbol{\mu}}) = \tilde{\mu}_0\tilde{\boldsymbol{\mu}}_d \tag{19}$$

where

$$\widetilde{C}(\tilde{\boldsymbol{\mu}}_e) = L_{dd}C(L_{ee}^{-1}\tilde{\boldsymbol{\mu}}_e).$$

5.2. A numerical instability observed

The blurring of the signal is assumed to be due to the convolution of the original signal with an unknown filter. This leads to an algebraic relation between the digitized signals and filter which in turn gives a similar relation between the moments of this data with respect to the chosen functional basis, (16) for geometric moments. The invariants are then obtained by eliminating the moments of the unknown filter from this relation using some assumed property of the filter. Assuming its symmetry, for example, leads to (17).

An instability has been observed in the following numerical experiments. For 1D signals, we calculated moments of two rows of a digitized image and, to check the invariance of the solution of (17) we blurred them by three different filters (box, hat and the second order B-spline). Having confirmed the invariance, we show the blur invariants in Fig. 6 in logarithmic scale. The fact, that the values of higher order invariants steeply increase should be suspicious but, even so, we were able to distinguish between the two signals. The higher order invariants are needed when we want to distinguish between signals which have the same first few moments. What happens in such a situation is shown in Figs. 7 and 8—we see that when 8 moments coincide, resulting in the first four blur invariants being identical, the higher invariants are also not different. This experiment was performed by combining low order moments of one signal with higher order moments of the other signal.

This problem is more serious than the growth mentioned above because it implies that the higher order moments, and higher order invariants, do not contribute to the discriminative power of the blur invariants.

We also ran the experiment with orthogonal polynomials because they are known to resolve problems associated with the powers basis. f The results are summarized in Fig. 9 where we show only the differences between blur invariants with certain numbers of common low order moments. Note that the discriminative power has increased but not beyond somewhere between invariants of order 8 and 12.

For reason discussed towards the end of Section 5.3 we performed yet another experiment. We used a basis of corrected powers

$$\boldsymbol{w}_\gamma(t) = \begin{pmatrix}1 & t & t^2+\gamma & t^3 & t^4+\gamma\cdots\end{pmatrix}^T \tag{20}$$

for some constant γ. In Fig. 10 we see that this basis performs better than the orthogonal invariants in Fig. 9 but that improvement is limited, too.

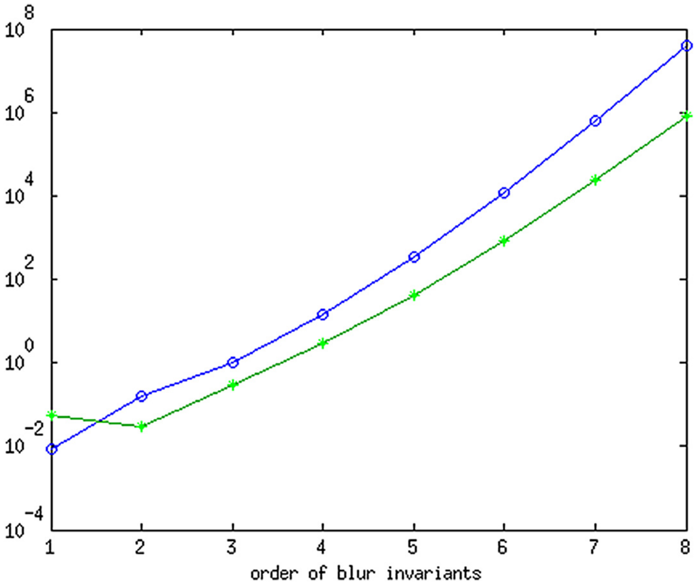

Fig. 6. Geometric blur invariants of two signals with all moments different. Blurred and original rows 12 and 44 of the 256 × 256 Lena image were used as signal data. Their l_2 norm difference was 858. Blurring by three different filters showed the invariance was less than $3.49e-10$.

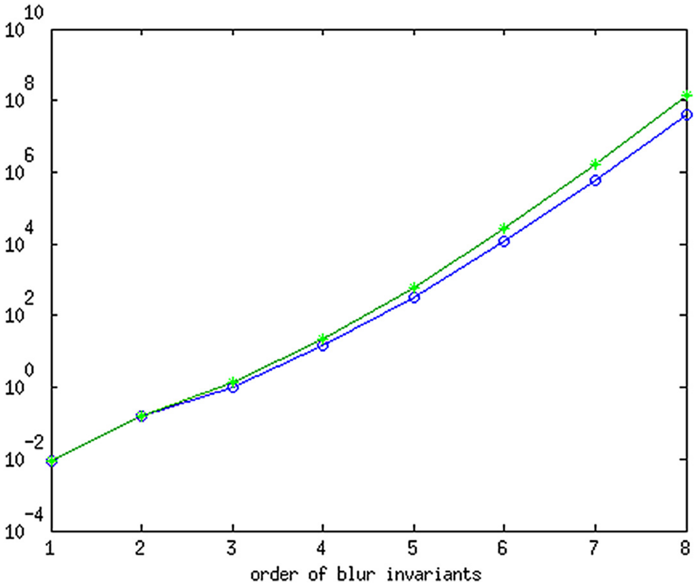

Fig. 7. Geometric blur invariants of two signals as in Fig. 6 but with first 4 moments equal; consequently, the first two invariants are equal as well.

For comparison, we show in Fig. 11 the blur invariants up to order 25 using real trigonometric (Fourier) "moments" (see Section 5.4 for definition and details). We observe that they are of reasonable magnitude and the change in higher order moments is well captured by the invariants.

We can now formulate the following conclusions from these experiments.

(a) Calculation of blur invariants using a polynomial basis suffers from severe instability resulting in higher moments not contributing to the ability to discriminate between signals with common lower order moments.

(b) For geometric moments this limitation starts at very low degree (moment order 8/invariants order 4).

(c) Although using an orthogonal basis improves things the same remedy is achieved by using corrected powers where a constant is added to the even powers. This increases the discriminative power to moment order 20/invariants order 10, probably more than necessary for practical purposes.

The suggested explicit limits are based on our particular experiments; they may slightly change depending on details of the implementation but not significantly.

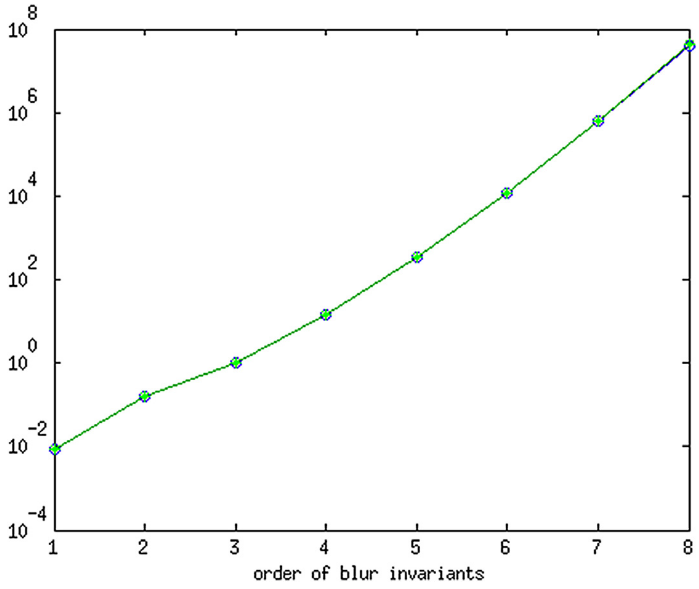

Fig. 8. Geometric blur invariants of two signals with first 8 moments equal; invariants of order 5 and higher should differ but do not!.

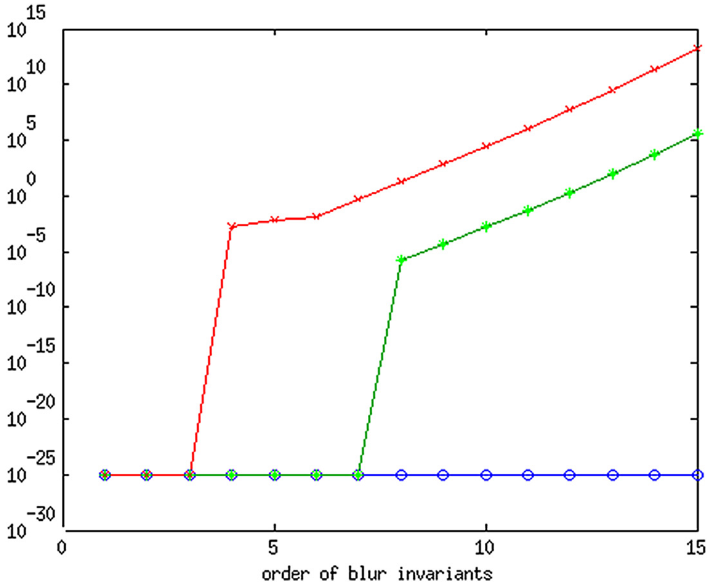

Fig. 9. Difference between blur invariants of two signals with same first 6 (red $\times$), 15 (green $+$) and 24 (blue $\circ$) Legendre orthogonal moments polynomials (to control the logarithmic display small values are replaced by 10^{-25}). Signals in which the first 24 moments are equal are not recognized as different. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

The roots of this instability should, however, be properly investigated and understood. This is the aim of the next section.

5.3. Analysis of the numerical instability

To study the reasons for the ineffectiveness of the higher order invariants is difficult; to analyze the invariants as we have no means, independent of solving (18), for obtaining correct results (a comment in passing, substituting into explicit formulae for the invariants is numerically equivalent to solving the system, or possibly even worse).

However, when we use blur invariants, we are implicitly using the moments of the unknown filter the values of which we can obtain independently. Therefore studying their evaluation should give us information about the blur invariants as well. Thus, in the experiments where we know the filter and its moments we can compare what we are evaluating with the exact results.

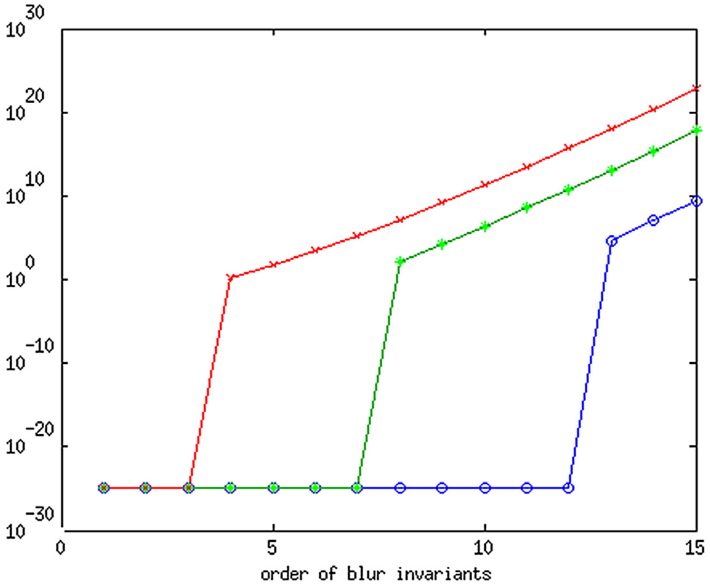

Fig. 10. Difference between blur invariants of two signals as in Fig. 9 but using the extended powers. Signals in which the first 24 moments are equal are recognized as different (only minor improvement).

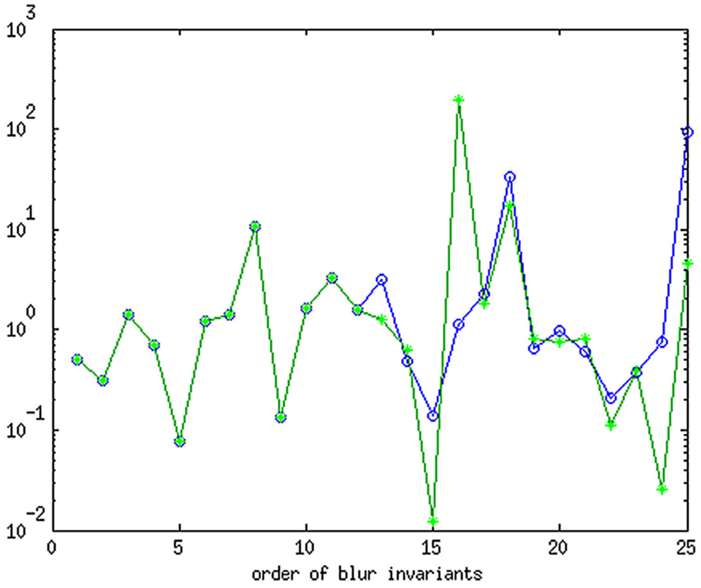

Fig. 11. Blur invariants using trigonometric moments, two signals with the same first 25 moments. Higher order invariants are bounded and show the difference well. Using three different filters the blur invariants differ by less than 3.49*e*−10.

In Fig. 12 we show the results of numerical calculations of the filter moments. Using the same quadrature as for the moments of other signals results in the blue graph: the odd order moments are, because of the symmetry, theoretically zeros, but computationally about 15 orders of magnitude smaller than the neighboring moments of even order. The other graphs were obtained by solving the system (16) which depends on the moments of the original signal and with the right-hand side generated by the moments of the blurred signal. The green graph used a wrong blurred signal, the purple the correct one and the two in between used a blur signal with increasing number of equal moments to the original signal.

The critical observation is in the purple graph. Up to order 8 the nonzero (even) moments of the filter are calculated quite accurately while the accuracy of the odd moments, which should be zero, gradually decreases. Furthermore, for orders higher than 8, the calculated moments steadily increase and have no resemblance of the correct moment values. It is therefore not surprising that the geometric blur invariants could not distinguish between two signals having the same first 8 moments.

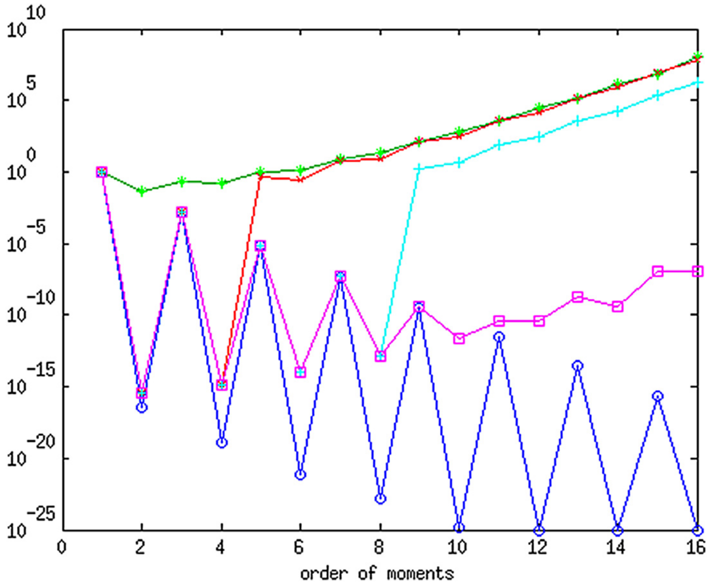

Fig. 12. Calculation of geometric moments of a known symmetric filter (blue ○), by solving (16) with the correct r.h.s. (purple □). Also while using r.h.s. with 6 (red ×) and 12 (cyan +) first moments as the original signal. Blurred and original rows 12 and 44 of the 256 × 256 Lena image were used as signal data. Their l_2 norm difference was 858. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

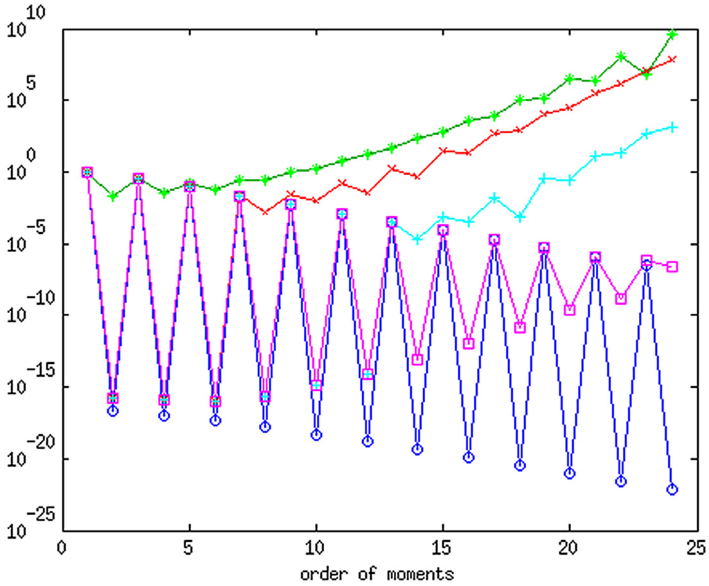

Fig. 13. Same calculation of symmetric filter moments as in Fig. 12 but using Legendre orthogonal polynomial moments.

In Fig. 13 we show similar results using orthogonal instead of geometric moments. We see that there is an essential improvement – the even (non-zero) filter moments are reasonably well evaluated up to order 22 rather than 8 – but then the same instability creeps in.

The reason for the instability is explained by the following lemma, theorem and corollary, the proofs of which are straightforward. First we need the following result which is essential as it replaces the Hadamard product in the definition of $M(\boldsymbol{\mu})$ by a straightforward multiplication of matrices.

Lemma 5.1. *Let* $D_{\boldsymbol{f}} = \mathrm{diag}((1, 1, 1/2!, 1/3!, \ldots, 1/(n-1)!))$ *be a diagonal matrix with the indicated elements,* ($\cdot_{\boldsymbol{f}}$ *is a mnemonic for factorial). Then the matrices* P *and* $M(\boldsymbol{\mu})$ *can be factorized as*

$$P = D_{\boldsymbol{f}}^{-1} T(D_{\boldsymbol{f}} \mathbf{1}) D_{\boldsymbol{f}} \quad \text{and} \quad M(\boldsymbol{\mu}) = D_{\boldsymbol{f}}^{-1} T(D_{\boldsymbol{f}} \boldsymbol{\mu}) D_{\boldsymbol{f}}.$$

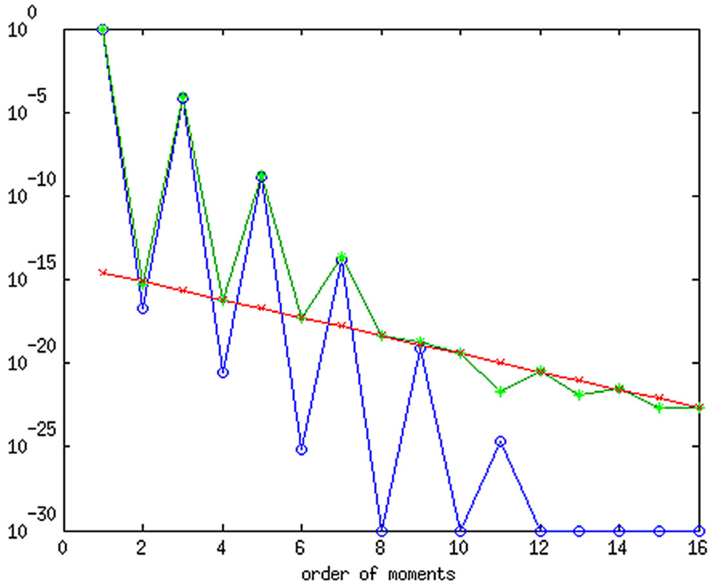

Fig. 14. Solutions of the reduced system (21) for geometric moments: calculated (green $*$) and exact (blue o). The slope of the spurious solution (red x) is given by the maximal root of the polynomial determined by the rows of the system matrix (mean 0.289, st. dev. 0.00017) and fits the calculated zeros. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

We can now formulate the main observation clarifying the problems of solving the system (16).

Theorem 5.2. *The system (16) for evaluating the filter moments $\boldsymbol{\pi}$ is equivalent to solving*

$$T(D_{\boldsymbol{f}}\boldsymbol{\mu})\boldsymbol{z} = D_{\boldsymbol{f}}\boldsymbol{\nu} \quad \text{and then} \quad \boldsymbol{\pi} = D_{\boldsymbol{f}}^{-1}\boldsymbol{z}. \tag{21}$$

If the basis functions are reasonably normalized then the moments are bounded. Thus the elements of the right-hand side, of the solution and also of the first column of the Toeplitz matrix of the system all decrease as $1/n!$ (the diagonal elements in $D_{\boldsymbol{f}}$), which is faster than any power. The higher order moments have limited influence on the result because their contribution is scaled down by the factorial. The recovery of $\boldsymbol{\pi}$ from $\boldsymbol{z}$, although trivial, is still numerically unstable. Even worse, the higher order moments of the filter cannot be calculated at all because of the following result which thus explains the observed instability.

Corollary 5.3. *If $\mu_j = 0$ for $j > j_0$ then there exist a numerical solution $\tilde{\pi}_j$ to (16) which is approximately*

$$\tilde{\pi}_j = c\sigma^j, \quad c \text{ a constant,}$$

where σ is the largest (in absolute value) root of the polynomial

$$\sum_{k=0}^{j_0} \binom{j_0}{k} \mu_k \sigma^k. \tag{22}$$

We have observed that the dominant root is almost independent of j_0; see Fig. 14 for the fit of the spurious solution to the calculated solution and also the standard deviation of the dominant root over the range of j_0's. The actual value of this root depends on the signal but, due to the facts mentioned above, the spurious solution will always prevail over the desired one.

A similar analysis can be done for the evaluation of the blur invariants, that is the solution of (18). After scaling to a form similar to (21), the rate of the decrease or growth of the spurious solution can be established and compared to the calculated solution.

One question of interest to answer is why do orthogonal polynomials perform better than the standard powers? Comparing Figs. 12 and 13 we observe that the spurious solution behaves similarly in both. On the other hand, the exact values of the geometric moments of the filter decrease quickly while those for orthogonal moments are almost constant. Thus the spurious (or parasitic) solution begins to dominate the correct values later, that is, only for higher order moments. This lead us to introduce the "corrected powers" (20) by adding a constant to the even powers, as mentioned in Section 5.2. This appears to suffice; orthogonal polynomials are not necessary to obtain a reasonable performance of blur invariants.

Note that the observed instability caused by a spurious solution of the forward process prevails whether we actually solve (21) by forward elimination or by other means, such as a solver using pivoting. One exception might be, as in other such situations, to run the elimination backwards; one estimates the tail of the solution, solves the intermediate equations and tries to fit the top end of the system. Such a shooting technique appears very insensitive for this problem and leads to no improvement. This may be due to the presence of very small roots of the polynomial in (22) which generate rapidly growing spurious solutions of the reversed process.

Finally, we investigated the possibility of exploiting the HRA (High Relative Accuracy) method of calculating with TN (Totally Nonnegative) matrices [26,27]. Recall that a totally nonnegative matrix is a matrix whose minors are all nonnegative. Although our lower triangular Pascal matrix is TN, the Toeplitz matrix in the critical system (21) is generally not. Numerical tests showed that even if it is forced to be one, by an artificial choice of the moments, the HRA method still calculates the spurious solution; its existence is not caused by the loss of relative accuracy.

5.4. Why are polynomials not convenient in the construction of blur invariants?

Another way to ask this question is: Why do Pascal triangles arise in this problem?

Let $\boldsymbol{p}(t) = \big(p_1(t), p_2(t), \ldots, p_n(t)\big)^T$ be a vector of some basis functions. Evaluating moments of the blurred signal $g = f * h$ with respect to this functional basis leads to

$$\boldsymbol{v} = \int g(x)\boldsymbol{p}(x)dx = \iint f(t)h(x-t)\boldsymbol{p}(x)dtdx = \iint f(t)h(x)\boldsymbol{p}(x+t)dxdt.$$

To involve moments of f and h we need to break the $\boldsymbol{p}(x+t)$ to separate the dependence on x and t. Generally, we would like to have

$$\boldsymbol{p}(x+t) = M(\boldsymbol{p}(t))\boldsymbol{p}(x) \tag{23}$$

where the matrix $M(\boldsymbol{a})$ may or may not be linear in $\boldsymbol{a}$,

$$M(\boldsymbol{a}) = \sum a_j M_j$$

for some constant matrices M_j. Using powers $\boldsymbol{w}(t)$ for $\boldsymbol{p}(t)$ brings in the binomial coefficients of the Pascal triangle into $M(\boldsymbol{w}(t))$.

It appears that the only other functions with the property (23) are those based on exponentials;

$$\boldsymbol{p}(t) = \begin{pmatrix}1 & \sin t & \cos t & \sin 2t & \cos 2t & \cdots\end{pmatrix}^T$$

is a good symmetric/asymmetric real choice with $M(\boldsymbol{p}(t))$ block diagonal. This basis was used in the calculations leading to Fig. 11. The projections onto this basis should in fact not be called moments because the basis is not formed by polynomials. This basis is closely related to the discrete Fourier transform and the moments are essentially the Fourier coefficients. The property (23) implies the well-known fact that the Fourier transform turns a convolution into a product. In signal processing, this property is called *Fourier Convolution Theorem*. Blur invariants in the Fourier domain are then derived from the ratio of a signal spectrum and its symmetric projection (see [22] for details). It is interesting to note that Fourier-domain invariants and moment blur invariants are closely related via the Taylor expansion of the Fourier spectrum, as was discovered in [28] and generalized in [22].

6. Conclusions

In this paper, we studied the influence of the Pascal triangle on the numerical evaluation of geometric moments and their derived form—central moments, complex moments, and moment blur invariants. Although these moment functions have been used extensively in signal and image analysis, this aspect has not been investigated before. We showed that the Pascal-type matrices entering moment calculations are extremely badly conditioned. While in the case of central and complex moments the impact on the numerical computation is almost negligible, in the case of blur invariants a spurious solution appears which dominates the correct solution for higher orders. Since the blur invariants are typically used to distinguish between two (blurred or non-blurred) signals, the consequence of our observation is that two signals are either distinguishable by low-order invariants up to certain order ρ or are not distinguishable at all. This effect is observable for any polynomial basis but the ρ depends on the particular choice of basis but not in any significant way on the signal itself. While for the power basis this threshold is about $\rho = 4$, by introducing the "corrected powers" or orthogonal polynomials we achieve higher threshold about $\rho = 10$ in both cases. This not only explains limited recognition power of the invariants reported in the literature (see for instance [8] for some examples and other relevant references) but also shows that introducing an orthogonal basis (which may be advantageous for other reasons) may here be replaced by a simple modification of the power basis with the same results.

As a historical remark, we note that the instability explanation presented in this paper is an echo from the 1960s when Dahlquist's resolution of instabilities of multi-step methods for ODEs [29] has lead to a stable application of such algorithms.

Acknowledgments

We acknowledge constructive suggestions by an anonymous referee. This work was supported by the Czech Science Foundation under the Grant No. GA15-16928S. The work was accomplished when J. Kautsky was visiting the Institute of Information Theory and Automation, Prague, Czech Republic.

References

[1] M.-K. Hu, Visual pattern recognition by moment invariants, IRE Trans. Inf. Theory 8 (2) (1962) 179–187.
[2] D. Hilbert, Theory of Algebraic Invariants, Cambridge University Press, Cambridge, UK, 1993.
[3] J.H. Grace, A. Young, The Algebra of Invariants, Cambridge University Press, 1903.
[4] G.B. Gurevich, Foundations of the Theory of Algebraic Invariants, Nordhoff, Groningen, The Netherlands, 1964.
[5] I. Schur, Vorlesungen über Invariantentheorie, Springer, Berlin, Germany, 1968, (in German).
[6] R. Mukundan, K.R. Ramakrishnan, Moment Functions in Image Analysis, World Scientific, Singapore, 1998.
[7] M. Pawlak, Image Analysis by Moments: Reconstruction and Computational Aspects, Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław, Poland, 2006.
[8] J. Flusser, T. Suk, B. Zitová, Moments and Moment Invariants in Pattern Recognition, Wiley, 2009.
[9] J. Flusser, T. Suk, B. Zitová, 2D and 3D Image Analysis by Moments, Wiley, 2016, in press.
[10] G.A. Papakostas (Ed.), Moments and Moment Invariants—Theory and Applications, Science Gate Publishing, 2014.
[11] G. Papakostas, D. Koulouriotis, E. Karakasisl, Computation strategies of orthogonal image moments: A comparative study, Appl. Math. Comput. 216 (1) (2010) 1–17.
[12] T. Suk, J. Flusser, C. Hoschl, Decomposition of binary images—A survey and comparison, Pattern Recognit. 45 (12) (2012) 4279–4291.
[13] C.-H. Teh, R.T. Chin, On image analysis by the method of moments, IEEE Trans. Pattern Anal. Mach. Intell. 10 (4) (1988) 496–513.
[14] M.R. Teague, Image analysis via the general theory of moments, J. Opt. Soc. Amer. 70 (8) (1980) 920–930.
[15] S.O. Belkasim, M. Shridhar, M. Ahmadi, Pattern recognition with moment invariants: A comparative study and new results, Pattern Recognit. 24 (12) (1991) 1117–1138.
[16] B. Yang, M. Dai, Image analysis by Gaussian–Hermite moments, Signal Process. 91 (10) (2011) 2290–2303.
[17] R. Mukundan, S.H. Ong, P.A. Lee, Image analysis by Tchebichef moments, IEEE Trans. Image Process. 10 (9) (2001) 1357–1364.
[18] P.-T. Yap, R. Paramesran, S.-H. Ong, Image analysis by Krawtchouk moments, IEEE Trans. Image Process. 12 (11) (2003) 1367–1377.
[19] T.S. Chihara, An Introduction to Orthogonal Polynomials, Gordon and Breach, New York, USA, 1978.
[20] M. Abramowitz, I.A. Stegun, Handbook of Mathematical Functions with Formulas, Graphs and Mathematical Tables, National Bureau of Standards, Washington, DC, USA, 1964.
[21] J. Flusser, On the independence of rotation moment invariants, Pattern Recognit. 33 (9) (2000) 1405–1410.
[22] J. Flusser, T. Suk, J. Boldyš, B. Zitová, Projection operators and moment invariants to image blurring, IEEE Trans. Pattern Anal. Mach. Intell. 37 (4) (2015) 786–802.
[23] J. Flusser, T. Suk, Classification of degraded signals by the method of invariants, Signal Process. 60 (2) (1997) 243–249.
[24] J. Kautsky, J. Flusser, Blur invariants constructed from arbitrary moments, IEEE Trans. Image Process. 20 (12) (2011) 3606–3611.
[25] G.H. Golub, C.F.V. Loan, Matrix Computations, North Oxford Academic, Oxford, 1983.
[26] P. Alonso, J. Delgado, R. Gallego, J.M. Peña, Conditioning and accurate computations with pascal matrices, J. Comput. Appl. Math. 252 (2013) 21–26.
[27] P. Koev, Accurate computations with totally nonnegative matrices, SIAM J. Matrix Anal. Appl. 29 (2007) 731–751.
[28] J. Flusser, T. Suk, Degraded image analysis: An invariant approach, IEEE Trans. Pattern Anal. Mach. Intell. 20 (6) (1998) 590–603.
[29] G. Dahlquist, Stability and error bounds in the numerical integration of ordinary differential equations (Ph.D. thesis), Royal Inst. of Technology, Stockholm, Sweden, 1958.