Efficient Computation of Local Geometric Moments

Judit Martínez and Federico Thomas

Abstract— **Local moments have attracted attention as local features in applications such as edge detection and texture segmentation. The main reason for this is that they are inherently integral-based features, so that their use reduce the effect of uncorrelated noise. The computation of local moments, when viewed as a neighborhood operation, can be interpreted as a convolution of the image with a set of masks. Nevertheless, moments computed inside overlapping windows are not independent and convolution does not take this fact into account. By introducing a matrix formulation and the concept of accumulation moments, this paper presents an algorithm which is computationally much more efficient than convolving and yet as simple.**

EDICS: IP1.1 (coding), IP1.2 (filtering).

I. Introduction

Two-dimensional geometric moments, sometimes also called standard moments, have been widely used in image analysis. In addition to them, a number of other moments have been proposed. Some examples are Legendre moments, Zernique moments, rotational moments, complex moments, etc. Actually, any set of parameters obtained by projecting an image onto a 2D polynomial basis are called moments. Then, since different sets of polynomials up to the same order define the same subspace, any complete set of moments up to a given order can be obtained from any other set of moments up to the same order, at least in theory. In practice, numerical ill-conditionings may arise and this is why, roughly speaking, different sets of moments exhibit different degrees of noise sensibility, information redundancy, and discrimination power [28]. The reader is addressed to [22] for an up-to-date monograph on moments and their role in image analysis.

It can be distinguished between global and local moments of an image. While the former are computed to obtain sets of global characteristics of, in general, binary images – in applications such as invariant pattern recognition, object classification pose estimation, image coding and reconstruction –, the latter are of interest as local features in gray-level images – in applications such as edge detection or texture segmentation. The use of global moments is mainly motivated either by their straightforward geometrical interpretation, or by the fact that some nonlinear combinations of them yield invariant parameters to changes of scale, translation, rotation and reflection. In the case of local moments [5], their interest as image features has been motivated by the fact that they are inherently integral-based features, so that their use reduce the effect of uncorrelated noise. This is why, for example, moment-based edge detectors have been shown to have a superior noise robustness and higher accuracy than those based on derivative operators which are very sensitive to noise [8], [16]. They have also been successfully used in texture segmentation [3], [29], and optic flow computations [4].

The computation of a set of geometric moments directly from its definition leads to very long processing times. To provide a way around this difficulty, hardware and software schemes have been proposed. While several hardware architectures for the fast computation of geometric moments of gray-level images have been explored including optical, VLSI, and parallel ones [28], most of the efforts in the software side have been focused on the binary domain [10]. [13]. In this sense, two different approaches have been considered: (1) reducing the number of involved pixels and (2) using recursivity. The first approach is carried out either by decomposing the image into non-overlapping areas such as one-line-thick rectangles (run-lengths), triangles or trapezoids, or by transforming the summation over the area to the summation along its contour using the discrete version of the Green's theorem. The second approach uses geometric moments of lower order to compute those of higher order. Both approaches have been jointly considered and extended to the 3D case in [31]. The first software approach that avoided the direct evaluation of the definition in the gray-level domain was presented by M. Hatamian in [7]. He exploited the idea that a 2-D filter with separable impulse response $x^m y^n u(x)u(y)$ can be used to generate the (m,n)th order moment of an image. The obtained result was limited to third order moments and no generalization was provided. B. Li expanded on this idea in [14], providing a general approach using rather involved signal processing techniques. Later on, the same results were obtained by the authors in [20] using a matrix formulation and the introduction of the concept of accumulation moments. Also, by projecting the image in different directions, the computation of its moments can be decomposed to become a number of one-dimensional ones. This is the approach taken in [26], where a discrete version of the Radon transform is used to this end. Since only additions are used to perform the projections, the number of multiplications can be greatly reduced leading to computational costs comparable to that of [14] or [20].

While it has been an active interest in finding efficient algorithms for computing global moments, little has been done to efficiently compute moments over small windows. At first glance, there is no need to make such as distinction but, if windows overlap, the possibility that they share computations arise.

Expressing a set of local features in a sliding-window in terms of the features in the preceding location of the window can be attempted for any set of local features, but the way this is achieved heavily relies on the properties of the

J. Martínez is with the Computer Vision Center, Edifici O, Campus UAB, 08193 Bellaterra, Spain (e-mail:judit@cvc.uab.es).

F. Thomas is with the Industrial Robotics Institute (CSIC-UPC), Llorens Artigas 4-6, 2 planta, 08028 Barcelona, Spain (e-mail:fthomas@iri.upc.es).

chosen features and the ability of the researcher to obtain the lowest computational cost. In the context of linear features, advancing a window one step in a given direction entails subtracting the contribution of the region excluded from the window, shifting the result, and adding the contribution of the entering region. Following this idea, the updating of the coefficients resulting from discrete Fourier or Cosine Transforms was considered in [32] for the particular application of image restoration and enhancement. In [11], a CORDIC-based unified systolic architecture for sliding window applications of discrete linear transforms was presented. In the context of non-linear filtering some simplifications might be necessary, in general, to made this idea applicable. For example, an approximation of the median image filter using a sliding window was presented in [18] where the filter is decomposed into two one-dimensional filters so that the median of the medians is actually computed.

The computation of local moments, when viewed as a neighborhood operation, can be interpreted as a convolution of the image with a set of masks. This simple interpretation has probably limited the number of further insights into the problem in the idea that any algorithm using intermediate computations to reduce the overall complexity would be unnecessarily involved. A sliding-window technique for updating the local moments of a real one-dimensional signal was presented in [27]. The main goal was to formulate an updating algorithm well-suited for VLSI implementation based on processor arrays with elements having pipeline structure. Almost simultaneously, [21] presented a sliding-window technique for computing the local moments of an image.

We elaborate herein some of the ideas presented in [20], [21], and [19] providing a self-contained presentation of an alternative to the convolution approach for the computation of the local geometric moments of an image. The presented algorithm is computationally much more efficient than convolving with a set of masks, and yet as simple to implement. Actually, computational costs are in general reduced in more than one order of magnitude.

This paper is organized as follows. Section 2, after reformulating the problem of computing geometric moments in matrix terms, introduces the concept of accumulation moments and shows how these moments can be readily obtained from geometric moments and vice versa through matrix computations. This relationship is exploited in section 3 for reducing the cost of computing geometric moments in a given image window as it slides over the image. Finally, we offer concluding remarks in section 4. Two appendices expand on different theoretical aspects of accumulation moments.

II. Accumulation moments

Let us assume a signal, $f(t)$, time-limited to the interval $0 \le t \le T$. Then, using the Laplace transform [25], it can be proved that its n-th moment can be calculated as follows:

$$\int_0^T t^n f(t)dt = \frac{n!}{(-1)^{n+1}} \int_0^T \int_0^{t_n} \cdots \int_0^{t_2} \int_0^{t_1} f(T-t_0)dt_0 dt_1 \ldots dt_{n-1} dt_n. \quad (1)$$

Roughly speaking, this equation means that the result of integrating a signal $n+1$ times is directly related to its n-th geometric moment. Also, from this formulation, it seems possible to compute moments of higher order using the intermediate steps required to obtain those of lower order. Based on this possibility, a simple iterative algorithm for the efficient computation of grey-level image accumulation moments could be envisaged, in the same way it was done in the binary domain in [10].

This section is essentially devoted to obtaining the 2D discrete counterpart of (1). To this end, a matrix formulation of the problem of computing geometric moments, that will greatly simplify our task, is first introduced.

A. Geometric moment matrices

In what follows, capital bold letters denote matrices and we always use subscripts to denote their dimensions. For example, $\mathbf{Z}_{mn}$ denotes a matrix and $\mathbf{Z}_{mn}[k,l]$, its element (k,l), where k and l may range from 1 to m, and from 1 to n, respectively. For simplicity, square matrices will only have one subscript. Superscripts are also used to denote any parameter on which a matrix depends. If a parameter becomes zero for a particular matrix instance the corresponding superscript will be omitted. Two unary matrix operations are used: $(\cdot)^t$ denotes the transpose of a given matrix; and $(\cdot)^{-1}$, its inverse. To avoid confusions, matrices are always embraced by parenthesis when superscripts refer to power or transpose.

Definition 1 (Geometric moment matrix) The geometric moment of order (m,n) with respect to an arbitrary point (v,w) of a discrete image $\mathbf{I}_{ab}$ is defined as:

$$\mu_{mn}^{vw} = \sum_{x=1}^{a}\sum_{y=1}^{b}(x-v)^m(y-w)^n\mathbf{I}_{ab}[x,y]. \quad (2)$$

Then, the geometric moment matrix, $\mathbf{M}_{mn}^{vw}$, is defined as:

$$\mathbf{M}_{mn}^{vw}[k,l] = \mu_{k-1,l-1}^{vw}.$$

Lemma 1: It turns out that

$$\mathbf{M}_{mn}^{vw} = (\mathbf{T}_m^v)^t\ (\mathbf{V}_{am})^t\ \mathbf{I}_{ab}\,\mathbf{V}_{bn}\,\mathbf{T}_n^w, \quad (3)$$

where

$$\mathbf{T}_p^q[k,l] = \begin{cases} \binom{l-1}{k-1}(-q)^{l-k}, & \text{if } \ l \ge k, \\ 0, & \text{otherwise,} \end{cases}$$

and $\mathbf{V}_{pq}$ is a non-square Vandermode matrix whose general term is:

$$\mathbf{V}_{pq}[k,l] = k^{l-1}. \quad (4)$$

Corollary 1: The geometric moments with respect to the origin of a discrete image $\mathbf{I}_{ab}$ can be obtained from:

$$\mathbf{M}_{mn} = (\mathbf{V}_{am})^t \, \mathbf{I}_{ab} \, \mathbf{V}_{bn}. \tag{5}$$

Corollary 2: The geometric moments with respect to an arbitrary point can be computed from those obtained with respect to the origin, as follows:

$$\mathbf{M}^{vw}_{mn} = (\mathbf{T}^v_m)^t \, \mathbf{M}_{mn} \, \mathbf{T}^w_n. \tag{6}$$

This corollary implies that, since $\mathbf{T}^q_p$ are upper triangular matrices, a geometric moment of an image positioned at any arbitrary point can be obtained from the geometric moments of the same order and lower of the image positioned at the origin. Since all the elements of their diagonals are different from zero, they are invertible. Note also that each matrix $\mathbf{T}^q_p$ accounts for a translation along either x or y. As the reader might imagine, this will be useful later when dealing with sliding windows on an image.

Characterizing all the information contained in an image requires the computation of a geometric moment matrix of the same size of the image itself. Most of the times the goal is to select a subset of moment values that contain sufficient information to uniquely characterize the image for a given application.

Next, we show that the accumulation of the pixel values of an image in different directions can be interpreted as moments and hence they can be directly related to geometric moments. We also provide the directions of accumulation that lead to the best numerically conditioned relationship between both kinds of moments.

B. Direct accumulation moment matrices

Definition 2 (Direct accumulation moment matrix) The direct accumulation moment of order $(k-1, l-1)$ of matrix $\mathbf{I}_{ab}$ is the value of $\mathbf{I}_{ab}[a, b]$ after top-down accumulating its columns k times (i.e., after applying k times the assignment $\mathbf{I}_{ab}[i+1, j] \leftarrow \mathbf{I}_{ab}[i+1, j] + \mathbf{I}_{ab}[i, j]$, for $i = 1$ to $a-1$, and for $j = 1$ to b), and accumulating the resulting last row from left to right l times (i.e., after applying l times the assignment $\mathbf{I}_{ab}[a, j+1] \leftarrow \mathbf{I}_{ab}[a, j+1] + \mathbf{I}_{ab}[a, j]$, for $j = 1$ to $b-1$). The direct accumulation moment matrix is defined so that $\mathbf{L}_{mn}[k, l]$ is the direct accumulation moment of order $(k-1, l-1)$.

This definition is better understood through an example.

Example 1: Consider the matrix

$$\begin{pmatrix} 0 & 2 & 3 \\ 1 & 0 & 8 \\ 1 & 1 & 1 \end{pmatrix}. \tag{7}$$

According to the definition, its direct accumulation moment of order $(1, 2)$ is computed by top-down accumulation of its columns twice and then left-right accumulating the resulting last row three times. The first and second column accumulations lead to

$$\begin{pmatrix} 0 & 2 & 3 \\ 1 & 2 & 11 \\ 2 & 3 & 12 \end{pmatrix} \text{ and } \begin{pmatrix} 0 & 2 & 3 \\ 1 & 4 & 14 \\ 3 & 7 & 26 \end{pmatrix},$$

respectively, and the three accumulations of the last row lead to

$$\begin{pmatrix} 3 & 10 & 36 \end{pmatrix}, \begin{pmatrix} 3 & 13 & 49 \end{pmatrix} \text{ and } \begin{pmatrix} 3 & 16 & 65 \end{pmatrix}.$$

Thus, the direct accumulation moment of order $(1, 2)$ of matrix (7) is 65. Obviously, the same result would be obtained if the rows were accumulated from left to right three times and the resulting last column, top-down accumulated twice.

The set of coefficients that we have just introduced can also be expressed as the projection coefficients of the image onto a 2D polynomial basis (see Appendix A for details) and thus they can be properly called moments. Now, our aim is to directly relate accumulation moment matrices to geometric moment matrices.

By simply analyzing the particular form of the coefficients that weight each element of the image matrix in successive accumulations, it is possible to verify that the direct accumulation moment of order $(k-1, l-1)$ can be expressed as

$$\mathbf{L}_{mn}[k, l] = \sum_{r=1}^{a} \sum_{s=1}^{b} \binom{a-r+k-1}{a-r} \mathbf{I}_{ab}[r, s] \binom{b-s+l-1}{b-s}. \tag{8}$$

Then, we can write

$$\mathbf{L}_{mn} = (\mathbf{Q}_{am})^t \, \mathbf{I}_{ab} \, \mathbf{Q}_{bn}, \tag{9}$$

where $\mathbf{Q}_{pq}$ is an up-down-flipped non-square Pascal matrix whose general term is:

$$\mathbf{Q}_{pq}[k, l] = \binom{p-k+l-1}{p-k}. \tag{10}$$

Lemma 2: Matrix $\mathbf{Q}_{pq}$ can be factored as follows:

$$\mathbf{Q}_{pq} = \mathbf{V}_{pq} \, \mathbf{G}^p_q, \tag{11}$$

where $\mathbf{V}_{pq}$ is a Vandermonde matrix and $\mathbf{G}^p_q$, an upper triangular matrix.

Proof: The combinatorial number in (10) can be decomposed into its multiplicative terms as follows:

$$\begin{aligned} \binom{p-k+l-1}{p-k} &= \frac{(p-k+l-1)!}{(p-k)!(l-1)!} \\ &= \frac{(-1)^{l-1}}{(l-1)!} (k+\beta^p_{l-1})(k+\beta^p_{l-2}) \cdots (k+\beta^p_1), \end{aligned} \tag{12}$$

where $\beta^p_i = -p-i$, with $i = 1, \ldots, (l-1)$.

If this expression is considered as a polynomial in k, it can be directly expressed in terms of its coefficients instead of its roots (see [2] for details) leading to:

$$\begin{aligned} &(k+\beta^p_{l-1})(k+\beta^p_{l-2}) \cdots (k+\beta^p_1) \\ &\qquad = k^{l-1} + B^p_{1,l-1} k^{l-2} + \cdots + B^p_{l-1,l-1}, \end{aligned}$$

where

$$\begin{aligned} B^p_{1,l-1} &= \sum_{i_1=1}^{l-1} \beta^p_1 \\ B^p_{2,l-1} &= \sum_{\substack{i_1,i_2=1 \\ i_1<i_2}}^{l-1} \beta^p_1 \beta^p_2 \\ B^p_{3,l-1} &= \sum_{\substack{i_1,i_2,i_3=1 \\ i_1<i_2<i_3}}^{l-1} \beta^p_1 \beta^p_2 \beta^p_3 \\ &\vdots \\ B^p_{l-1,l-1} &= \beta^p_1 \beta^p_2 \cdots \beta^p_{l-1}, \end{aligned} \tag{13}$$

and $B^p_{r,s} = 1$, either if $r = 0$ or $s = 0$.

Then, in vector form,

$$\mathbf{Q}_{pq}[k,l] = \frac{(-1)^{l-1}}{(l-1)!} \begin{pmatrix} 1 & k & \dots & k^{l-1} \end{pmatrix} \begin{pmatrix} B^p_{l-1,l-1} \\ B^p_{l-2,l-1} \\ \vdots \\ B^p_{1,l-1} \\ 1 \end{pmatrix}.$$

Considering all the elements of matrix $\mathbf{Q}_{pq}$, the above vector product turns into the matrix product stated in equation (11), where

$$\mathbf{G}^p_q[k,l] = \begin{cases} \frac{(-1)^{l-1}}{(l-1)!} B^p_{l-k,l-1}, & \text{if } l \geq k, \\ 0, & \text{otherwise,} \end{cases} \tag{14}$$

is a square upper triangular matrix. Since all the elements of its diagonal are different from zero, it is non-singular. ■

Using lemma 2 and corollary 1, the following result is obtained.

Corollary 3: Direct accumulation moments can be expressed in terms of geometric moments and vice versa as:

$$\mathbf{L}_{mn} = (\mathbf{G}^a_m)^t \, \mathbf{M}_{mn} \, \mathbf{G}^b_n, \tag{15}$$

and

$$\mathbf{M}_{mn} = \left((\mathbf{G}^a_m)^{-1} \right)^t \, \mathbf{L}_{mn} \, \left(\mathbf{G}^b_n \right)^{-1}, \tag{16}$$

respectively. Since $\mathbf{G}^a_m$ and $\mathbf{G}^b_n$ are upper triangular matrices, the geometric moment of order (m,n) of an image can be obtained from the accumulation moments up to order (m,n) and vice versa.

Values for $(\mathbf{G}^a_n)^{-1}$ and $\left(\mathbf{G}^b_m\right)^{-1}$ can be obtained by numerically inverting $\mathbf{G}^a_n$ and $\mathbf{G}^b_m$, respectively.

Example 2:

$$\left(\mathbf{G}^{16}_4\right)^{-1} = \begin{pmatrix} 1 & 17 & 289 & 4913 \\ 0 & -1 & -35 & -919 \\ 0 & 0 & 2 & 108 \\ 0 & 0 & 0 & -6 \end{pmatrix},$$

$$\left(\mathbf{G}^{16}_5\right)^{-1} = \begin{pmatrix} 1 & 17 & 289 & 4913 & 83521 \\ 0 & -1 & -35 & -919 & -21455 \\ 0 & 0 & 2 & 108 & 3890 \\ 0 & 0 & 0 & -6 & -444 \\ 0 & 0 & 0 & 0 & 24 \end{pmatrix},$$

and

$$\left(\mathbf{G}^{17}_4\right)^{-1} = \begin{pmatrix} 1 & 18 & 324 & 5832 \\ 0 & -1 & -370 & -1027 \\ 0 & 0 & 2 & 114 \\ 0 & 0 & 0 & -6 \end{pmatrix}$$

Note the increasing dispersion in the elements of $\left(\mathbf{G}^p_q\right)^{-1}$, as p and q increase, in the above example. This suggests that an underlying numerical ill-conditioning might appear in the inversion due to roundoff errors. A measure of the ill-conditioning in the inversion of $\mathbf{G}^p_q$ is given by its condition number [6], i.e. the ratio between its largest and its smallest eigenvalue, for different values of p and q. The result is plotted in figure 1a and 1b. It confirms that, although $\mathbf{G}^p_q$ is non-singular, its inverse for high order moments or large images cannot be accurately computed because of an ill-conditioning.

Next, we will explore the possibility of accumulating image values in the direction opposite to the one we have just done in order to obtain a better conditioned relationship with geometric moments. This is encouraged by the fact that, while direct accumulation moments are more sensitive to variations of pixel values close to the origin, geometric moments referenced to the origin are more influenced by those far from it.

C. Reverse accumulation moment matrices

Definition 3 (Reverse accumulation moment matrix) The reverse accumulation moment of order $(k-1, l-1)$ of matrix $\mathbf{I}_{ab}$ is the value of $\mathbf{I}_{ab}[1,1]$ after bottom-up accumulating its columns k times (i.e., after applying k times the assignment $\mathbf{I}_{ab}[a-i,j] \leftarrow \mathbf{I}_{ab}[a-i,j] + \mathbf{I}_{ab}[a-i+1,j]$, for $i = 1$ to $a-1$, and for $j = 1$ to b), and accumulating the resulting first row from right to left l times (i.e., after applying l times the assignment $\mathbf{I}_{ab}[1,b-j] \leftarrow \mathbf{I}_{ab}[1,b-j] + \mathbf{I}_{ab}[1,b-j+1]$, for $j = 1$ to $b-1$). The reverse accumulation moment matrix is defined so that $\mathbf{R}_{mn}[k,l]$ is the reverse accumulation moment of order $(k-1, l-1)$.

Example 3: Using the same matrix as in example 1, the reverse accumulation moment of order $(1,2)$ requires two column accumulations, which leads to

$$\begin{pmatrix} 2 & 3 & 12 \\ 2 & 1 & 9 \\ 1 & 1 & 1 \end{pmatrix} \text{ and } \begin{pmatrix} 5 & 5 & 22 \\ 3 & 2 & 10 \\ 1 & 1 & 1 \end{pmatrix}$$

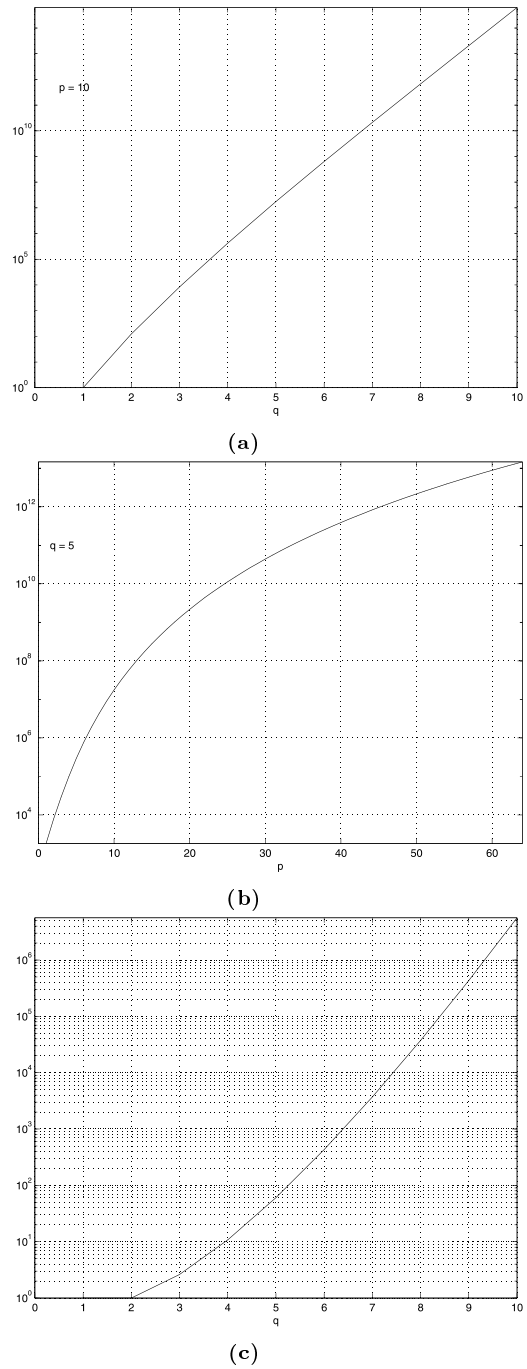


Fig. 1. **(a)** *Logarithmic plot of the condition number of* $\mathbf{G}_q^p$ *as a function of* q*, and* **(b)** *as a function of* p*.* **(c)** *The same for* $\mathbf{H}_q$ *as a function of* q*.*

respectively, and the three right-to-left accumulations of the first row to

$$\begin{pmatrix}32 & 27 & 22\end{pmatrix}, \begin{pmatrix}81 & 49 & 22\end{pmatrix} \text{ and } \begin{pmatrix}152 & 71 & 22\end{pmatrix}.$$

Then, it is said that the reverse accumulation moment of order $(1,2)$ of matrix (7) is 152.

It can be easily shown that the reverse accumulation moment of order $(k-1, l-1)$ can be expressed, in terms of the image, as:

$$\mathbf{R}_{mn}[k,l] = \sum_{r=1}^{a}\sum_{s=1}^{b} \binom{r+k-2}{r-1} \mathbf{I}_{ab}[r,s] \binom{s+l-2}{s-1}. \quad (17)$$

Then, we can write

$$\mathbf{R}_{mn} = (\mathbf{P}_{am})^t \, \mathbf{I}_{ab} \, \mathbf{P}_{bn}, \quad (18)$$

where $\mathbf{P}_{pq}$ is a non-square Pascal matrix whose general term is:

$$\mathbf{P}_{pq}[k,l] = \binom{k+l-2}{k-1}. \quad (19)$$

Lemma 3: Matrix $\mathbf{P}_{pq}$ can be factored as follows:

$$\mathbf{P}_{pq} = \mathbf{V}_{pq} \, \mathbf{H}_q, \quad (20)$$

where

$$\mathbf{H}_q[k,l] = \begin{cases} \frac{(-1)^{l-k}}{(l-1)!} B^{-1}_{l-k,l-1}, & \text{if } \ l \geq k \\ 0, & \text{otherwise,} \end{cases} \quad (21)$$

is a square upper triangular matrix. Since all the elements of its diagonal are different from zero, it is non-singular.

Using lemma 3 and corollary 1, the following result is obtained.

Corollary 4: Reverse accumulation moments can be expressed in terms of geometric moments and vice versa as:

$$\mathbf{R}_{mn} = (\mathbf{H}_m)^t \, \mathbf{M}_{mn} \, \mathbf{H}_n \quad (22)$$

and

$$\mathbf{M}_{mn} = \left((\mathbf{H}_m)^{-1}\right)^t \, \mathbf{R}_{mn} \, (\mathbf{H}_n)^{-1}, \quad (23)$$

respectively.

The condition number of $\mathbf{H}_p$ as a function of p is shown in figure 1c. Clearly, the relationship between reverse accumulation moments and geometric moments, besides being independent from the size of the image, is better conditioned as suspected.

Appendix C shows how direct and reverse accumulation moments are related through a simple matrix expression.

D. Numerical considerations

As a consequence of ill-conditioning, small perturbations in the input data generate large errors in the result. Even assuming that our data is not affected by measurement errors, roundoff errors are always present. Most computers use a floating point representation that takes 11 bits for the exponent, 52 bits for the mantissa, and 1 bit for the sign. Then, the exponent ranges from 1 to 2046, where $2046 = 2^{11} - 2$, but it is normalized from -1023 to 1022. As a consequence, the largest representable floating point number is $1.797693134862316 \times 10^{308}$ and the smallest positive floating point number is $2.225073858507201 \times 10^{-308}$. Since the mantissa is specified by 52 bits, the *computer integer resolution*, i.e. the maximum unsigned floating point integer, is $\eta = 2^{53} - 1 = 9.007199254740991 \times 10^{15}$. Therefore, any pair of numbers that have the same exponent but

their mantissa differ below η^{-1} have the same representation. This is how roundoff errors arise. Hence, the order in which additions are carried out is important since the associative law $(n_1 \oplus n_2) \oplus n_3 = n_1 \oplus (n_2 \oplus n_3)$, where the symbols $\oplus$ denotes computer addition, does not always hold.

Assuming a square image of size a and $0 \leq \mathbf{I}_a[k,l] \leq 255$, the maximum order $(m_{\max}, m_{\max})$ of the geometric or accumulation moment which can be computed without roundoff errors appear in Figure 2 [19]. For example, for an image of size 20×20, the maximum order of the geometric and accumulation moments (either direct or reverse) which can be computed without roundoff errors is $(6,6)$ and $(10,10)$, respectively.

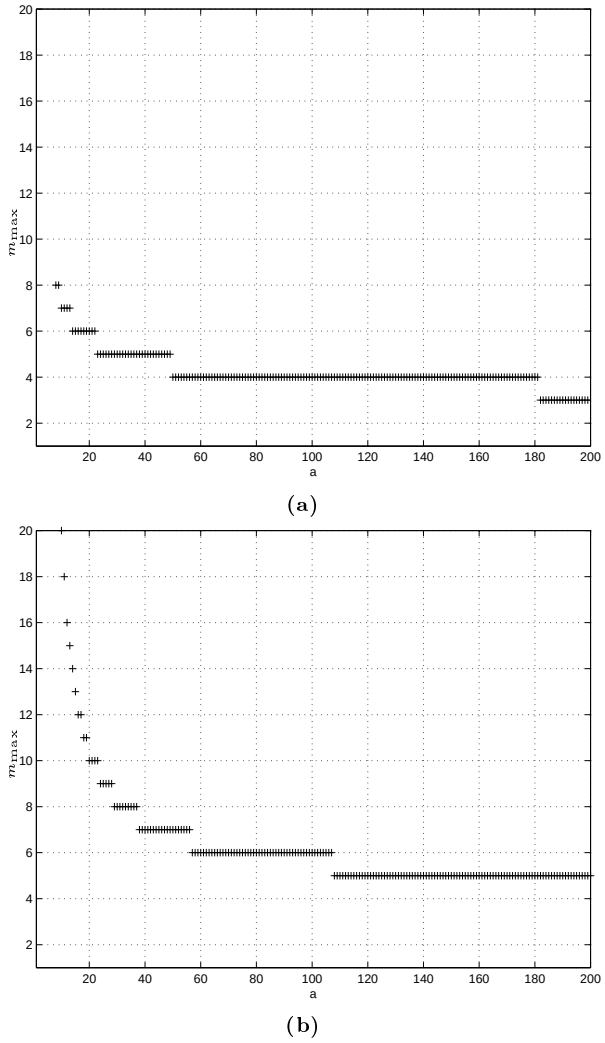


Fig. 2. *Maximum moment order, (m_{max}, m_{max}), that can be obtained without roundoff error in function of the size of an $a \times a$ image, when the computer integer resolution is $2^{53}-1$ for* **(a)** *geometric moments, and* **(b)** *accumulation moments.*

III. Computing local geometric moments

Using the matrix expressions obtained up to this point, it can be seen that, in terms of the number of products and additions, the complexity of computing geometric moments using (23), i.e. through accumulation moments, compares favorably to the direct evaluation of (5).

Let us consider a square image of size a for which all moments up to order (m,m) have to be computed. Then, the number of multiplications and additions required to compute (5) is $a^2m + am^2$, and $a(a-1)m + (a-1)m^2$, respectively. On the other hand, since $\mathbf{H}_m$ are triangular matrices, the evaluation of (23) requires $m^2(m+1)$ multiplications and $m(m-1)^2$ additions. Taking into account that the number of additions required to obtain $\mathbf{R}_m$ is $a(a-1)m + m^2(a-1)$, the total number of additions is $a(a-1)m + m^2(a-1) + m(m-1)^2$. In this case the number of multiplications is independent from the size of the image. Table I compiles these results.

Figure 3 shows the relative computational cost of computing geometric moments directly or through accumulation moments. The comparison is done by varying the maximum order of the required moments (fig. 3a), and the image size (fig. 3b). We have assumed that multiplication and addition operations require the same time, as it usually happens on most RISC processors where double (64 bits) multiplications and additions are computed in the same number of clock cycles (see Table I in [1]).

Thus, the advantage of using accumulation moments as an intermediate step for computing geometric moments is clear. The obtained computational complexity is the same as the one obtained in [14], but the mathematical deductions we used here to attain it are far simpler. The advantages of the used matrix formulation become even clearer in what follows.

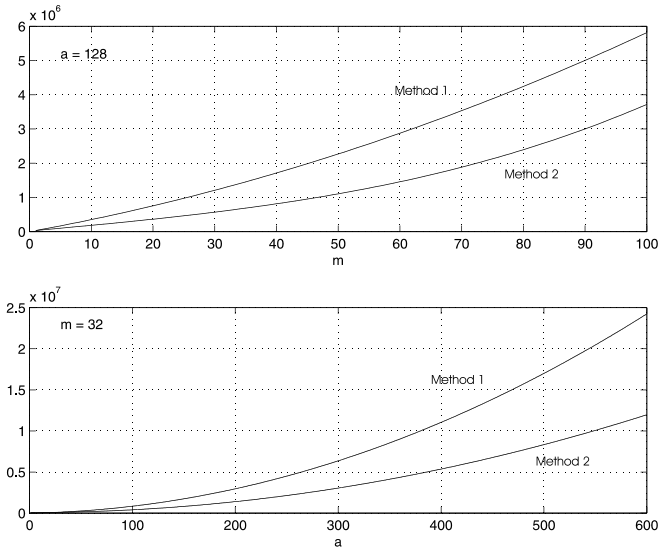


Fig. 3. *Relative computational costs of the two methods in Table I.*

Now, consider a window W of size $w \times w$ located at (r,s) on image $\mathbf{I}_{ab}$, as shown in figure 4(a). According to (2), the local geometric moment of order (m,n) in the local reference of W is:

$$\mu_{mn}^{rs} = \sum_{x=1}^{w}\sum_{y=1}^{w} x^m y^n \mathbf{I}_{ab}[x+r, y+s]. \qquad (24)$$

Method	*Additions*	*Multiplications*
1- Evaluation of (5)	$a(a-1)m+(a-1)m^2$	a^2m+am^2
2- Accumulations and evaluation of (23)	$a(a-1)m+(a-1)m^2+m(m-1)^2$	$m^2(m+1)$

TABLE I

NUMBER OF OPERATIONS REQUIRED TO COMPUTE GEOMETRIC MOMENTS DIRECTLY OR THROUGH ACCUMULATION MOMENTS.

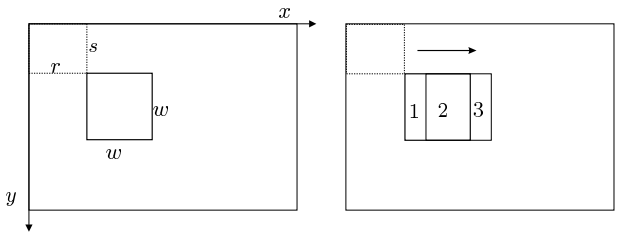


Fig. 4. **(a)** A window of size $w \times w$ located at (r, s), and **(b)** the three regions induced by the window as it slides from left to right.

When this is computed for every possible location of the window on the image, it can be interpreted as a convolution of the image with a mask. Nevertheless, it is clear that moments computed inside overlapping windows are not independent and a convolution does not take this fact into account.

In order to find an alternative to convolution, consider the case in which a sliding window is displaced all over the image. Then, the geometric moments obtained in a particular window location can be related to those in the previous location of the window as follows. Consider the three regions induced by the window sliding from left to right in two consecutive locations. According to figure 4(b), region 1 will be called the *outgoing region;* region 2, the *overlapped region;* and region 3, the *incoming region*. Let $\mathbf{M1}_{mn}$, $\mathbf{M2}_{mn}$ and $\mathbf{M3}_{mn}$ denote the geometric moments of each of these regions, and $\mathbf{M12}_{mn}$ and $\mathbf{M23}_{mn}$, the geometric moments obtained at two consecutive locations of the window. Then, if two consecutive locations of this sliding window reference differ in just one pixel, it can shown, using corollary 2, that

$$\mathbf{M23}_{mn} = \left(\mathbf{M12}_{mn} - \mathbf{M1}_{mn}\right)\mathbf{T}_n^1 + \mathbf{M3}_{mn}\,\mathbf{T}_n^{1-w}, \quad (25)$$

where $\mathbf{M1}_{mn}$ and $\mathbf{M3}_{mn}$ become the geometric moment matrices of column vectors. Therefore, their columns have all the same value and they can be expressed as:

$$\mathbf{M1}_{mn} = \mathbf{M1}_{mn}[:,1]\mathbf{s}_n$$

and

$$\mathbf{M3}_{mn} = \mathbf{M3}_{mn}[:,1]\mathbf{s}_n,$$

where $\mathbf{M1}_{mn}[:,1]$ and $\mathbf{M3}_{mn}[:,1]$ stand for the first column of $\mathbf{M1}_{mn}$ and $\mathbf{M3}_{mn}$, respectively; and $\mathbf{s}_n = \begin{pmatrix}1 & \dots & 1\end{pmatrix}$.

Since $\mathbf{T}^{1-w} = \mathbf{T}^1\mathbf{T}^{-w}$, the substitution of these expressions in (25) leads to:

$$\mathbf{M23}_{mn} = \mathbf{M12}_{mn}\,\mathbf{T}_n^1 - \mathbf{M1}_{mn}[:,1]\mathbf{u}_n + \mathbf{M3}_{mn}[:,1]\mathbf{u}_n\mathbf{T}^{-w}, \quad (26)$$

where $\mathbf{u}_n = \mathbf{s}_n\mathbf{T}^1 = \begin{pmatrix}1 & 0 & 0\dots & 0\end{pmatrix}$.

Moreover, it can be proved that $\mathbf{v}_n^w = \mathbf{u}_n\mathbf{T}_n^{-w}$ is a rowwise vector of the form

$$\mathbf{v}_n^w[k] = (w+1)^{k-1}.$$

The equivalent result to (25) for reverse accumulation moments is:

$$\mathbf{R23}_{mn} = \left(\mathbf{R12}_{mn} - \mathbf{R1}_{mn}\right)(\mathbf{U}_n)^{-1} + \mathbf{R3}_{mn}\,(\mathbf{U}_n)^{w-1}, \quad (27)$$

where the numbered matrices have their obvious meaning, and

$$\mathbf{U}_n[l,k] = \begin{cases}1, & \text{if } \ l \geq k,\\ 0, & \text{otherwise.}\end{cases} \quad (28)$$

To prove (27), note that the reverse accumulation moment matrix of the sliding window in its shifted location, $\mathbf{R23}_{mn}$, becomes $\mathbf{R23}_{mn}\mathbf{U}$ when expressed in the original reference. This can be derived by substituting s by $s+1$ in the second combinatorial number in (17). As a consequence, $\mathbf{R12}_{mn}$ becomes $\mathbf{R12}_{mn}(\mathbf{U})^{-1}$ when expressed in the shifted reference.

Since $\mathbf{R1}_{mn}$ and $\mathbf{R3}_{mn}$ are the accumulation moment matrices of column vectors, they can be expressed as:

$$\mathbf{R1}_{mn} = \mathbf{R1}_{mn}[:,1]\mathbf{s}_n$$

and

$$\mathbf{R3}_{mn} = \mathbf{R3}_{mn}[:,1]\mathbf{s}_n,$$

respectively. The substitution of these expressions in (27) leads to:

$$\mathbf{R23}_{mn} = \mathbf{R12}_{mn}(\mathbf{U}_n)^{-1} - \mathbf{R1}_{mn}[:,1]\mathbf{u}_n + \mathbf{R3}_{mn}[:,1]\mathbf{z}_n^w \quad (29)$$

where $\mathbf{z}_n^w$ is a rowwise vector of the form:

$$\mathbf{z}_n^w[k] = \binom{w+k-2}{k-1};$$

and

$$(\mathbf{U}_n)^{-1} = \begin{pmatrix} 1 & -1 & 0 & \cdots & 0 \\ 0 & 1 & -1 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & & & & \\ 0 & 0 & 0 & \cdots & -1 \\ 0 & 0 & 0 & \cdots & 1 \end{pmatrix}.$$

Then, as alternatives to convolution, the geometric moments in a sliding window can be computed using either (26), or (29) in conjunction with (23). Although both alternatives provide an efficient way to compute geometric and accumulation moments, in sliding window applications, the particular matrices involved in these equations make the overall computational cost lower for the latter.

Let us assume that the moment matrix be square of size m. Then, to evaluate $\mathbf{M12}_{mm}\mathbf{T}^1_m - \mathbf{M1}_{mm}[:,1]\mathbf{u}_m + \mathbf{M3}_{mm}[:,1]\mathbf{v}^w_m$, one has to bear in mind that:

- The cost of evaluating $\mathbf{M1}_{mm}$ or $\mathbf{M3}_{mm}$ entails $m(w-1)$ additions and $w(m-1)$ products.
- The cost of evaluating $\mathbf{M12}_{mm}\mathbf{T}^1_m$ entails $\frac{m(m-1)}{2}$ additions and $\frac{m(m+1)}{2}$ products.
- $\mathbf{M1}_{mm}[:,1]\mathbf{u}_m$ does not entail any product or addition.
- $\mathbf{M3}_{mm}[:,1]\mathbf{v}^w_m$ requires m^2 products.
- Adding up the three terms requires $2m^2$ additions.

As a consequence, the total cost is $2m(w-1)+\frac{5m^2}{2}-\frac{m}{2}$ additions and $2w(m-1)+\frac{3m^2}{2}+\frac{m}{2}$ products.

On the other hand, to evaluate $\mathbf{R12}_{mm}(\mathbf{U}_m)^{-1} - \mathbf{R1}_{mm}[:,1]\mathbf{u}_m + \mathbf{R3}_{mm}[:,1]\mathbf{z}^w_m$, one has to bear in mind that:

- The cost of evaluating $\mathbf{R1}_{mm}$ or $\mathbf{R3}_{mm}$ entails $m(w-1)$ additions.
- The cost of evaluating $\mathbf{R12}_{mm}(\mathbf{U}_m)^{-1}$ entails $m(m-1)$ additions.
- $\mathbf{R1}_{mm}[:,1]\mathbf{u}_m$ does not entail any product or addition.
- $\mathbf{R3}_{mm}[:,1]\mathbf{z}^w_m$ requires m^2 products.
- Adding up the three terms requires $2m^2$ additions.

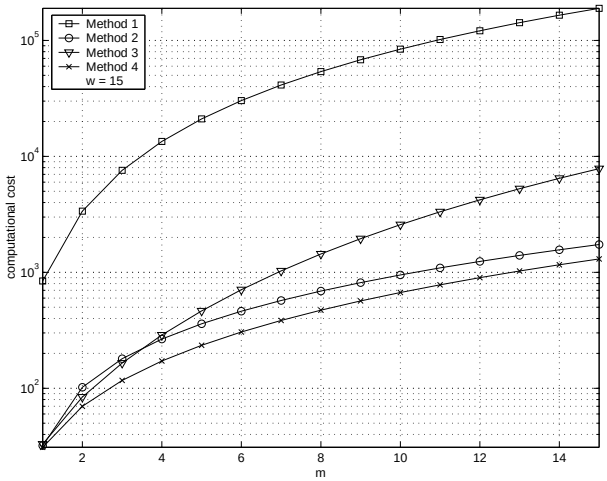
As a consequence, the total cost is $2m(w-1)+3m^2-m$ additions and m^2 products.

If the computation of accumulation moments is used as an intermediate step to obtain the corresponding geometric ones, the cost of computing (23) —$m(m-1)^2$ additions and $m^2(m+1)$ multiplications— must be added, no matter the window size.

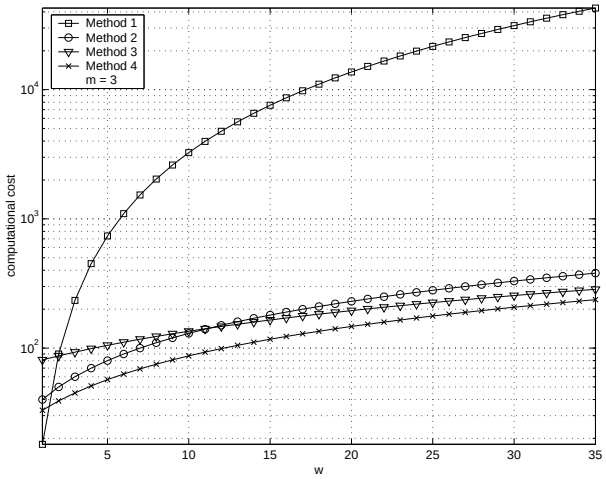
The traditional convolution approach would require the convolution with m^2 masks. Then, the total number of additions and multiplications would be $m^2w(w-1)$ and $m^2w(w+1)$, respectively. Note that the accumulation moments can also be obtained by convolving with masks using (30) or (31) in Appendix A.

Table II compiles all these costs and Figure 5 plots them for different values of m and w assuming the same cost for multiplications and additions. Two main consequences can be drawn from these costs:

- While the computation of geometric moments using convolution (method 1) is quadratic with the size of the window and with the maximum order moment, the presented



(a)



(b)

Fig. 5. *Relative computational costs of the four methods considered in Table III for increasing moment orders* **(a)**, *and increasing window sizes* **(b)**.

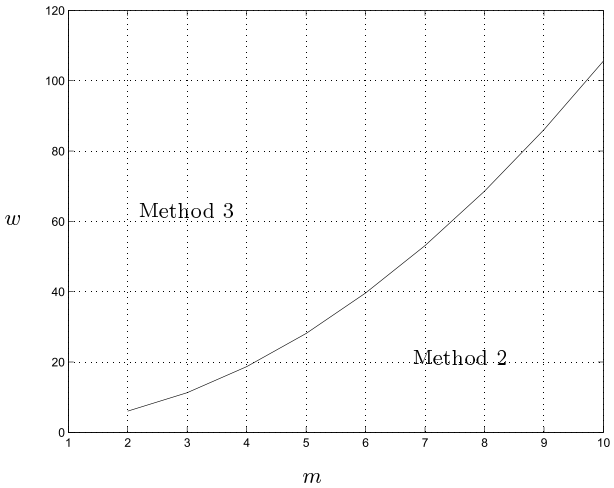


Fig. 6. Regions of the (m, w) plane where it is better to use either method 3 or method 2.

Moments	*Method*	*Additions*	*Multiplications*
Accumulation or geometric	1- Convolution	$m^2w(w-1)$	$m^2w(w+1)$
Geometric	2- Evaluation of (26)	$2m(w-1)+\frac{5}{2}m^2-\frac{m}{2}$	$2w(m-1)+\frac{3}{2}m^2+\frac{m}{2}$
Geometric	3- Evaluation of (29) and (23)	$2m(w-1)+m^3+m^2$	m^3+2m^2
Accumulation	4- Evaluation of (29)	$2m(w-1)+3m^2-m$	m^2

TABLE II
NUMBER OF OPERATIONS REQUIRED TO UPDATE GEOMETRIC AND/OR ACCUMULATION MOMENTS IN A SLIDING WINDOW USING FOUR DIFFERENT METHODS.

alternatives (methods 2 and 3) are linear with the maximum moment order. Then, although both alternatives are greatly advantageous with respect to convolution, it can be easily shown that method 3 —the one that uses accumulation moments as an intermediate step to compute geometric moments— is the best choice if

$$w > \frac{2m^3 - m^2}{2(m-1)}.$$

Figure 6 plots this region. Then, for example, if $m = 3$ and $w > 11$, it is better to use method 3.

• Since the computation of accumulation moments (method 4) is quadratic with the maximum order moment but linear with the size of the window, important computational saves can be obtained when directly working in terms of accumulation moments.

The actualization of local moments using the above methods have been implemented in C on a Pentium II/Windows NT running at 400 Mhz. For the 110×503 image of a fabric texture shown in figure 7a, the computation of their accumulation moments up to order $(4, 4)$ using a 32×32 sliding window takes 400 milliseconds and the computation of moments up to order $(8, 8)$ takes 855 milliseconds. When the size of the window is reduced to 16×16, these times drop to 273 and 638 milliseconds, respectively. If the same moments are obtained using convolution, the cost is about two orders of magnitude higher.

Figures 7b and 7c represent $\mathbf{R}[3, 0]$ and $\mathbf{R}[0, 3]$ for a 32×32 sliding window.

IV. Conclusions

If the values of a matrix are accumulated along its rows or columns, we obtain a vector of accumulated values. This vector can also be accumulated to yield a single number. If these accumulations are repeated several times on the already accumulated matrix or vector, the result is a set of numbers that we call here accumulation moments. These numbers can be properly called moments because they can also be obtained by projecting the image onto a polynomial basis.

Since the definition of the different kinds of moments only differ in the specific polynomial basis set and the chosen coordinate system, either polar or cartesian, it is always possible to find the accumulation moments of an image first and then to use them to compute other kinds of moments by changing the basis.

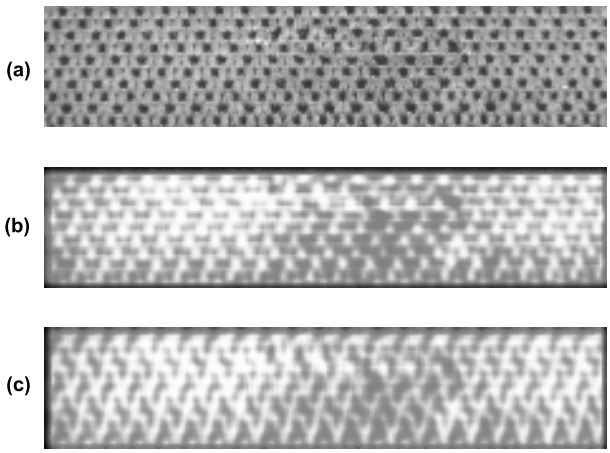


Fig. 7. **(a)** 110×503 *fabric texture;* **(b)** $\mathbf{R}[3, 0]$, *and* **(c)** $\mathbf{R}[0, 3]$ *for a* 32×32 *window.*

These ideas have been formalized using a simple matrix formalism which has allowed us to devise an efficient algorithm for the computation of local moments in a window sliding all over a region of interest in an image. The obtained computational complexities for the different presented alternatives are far lower than that of convolving the image with a set of masks.

The main advantage of accumulation moments over other well-known moments is that they can be efficiently computed using accumulations filters. Hopefully, future hardware designs will possibly benefit from the approach presented here.

Finally, it is worth to mention that the concepts of accumulation moment and sliding windows can be extended to higher dimensions, in particular to volumes described in terms of voxels. In this sense, 3D accumulation moments would be useful to accelerate the computation of mechanical parameters, such as center of mass, inertia axes, etc., and the segmentation of volumes, in applications dealing with non-homogeneous objects. This is a point that deserves further attention.

Appendix A

Image moments are obtained by projecting the image onto a particular polynomial basis set. In this appendix the polynomials associated with the accumulation moments are derived.

Direct and reverse accumulation moments can be expressed as:

$$\mathbf{L}_{mn}[k,l] = \sum_{x=1}^{a}\sum_{y=1}^{b}\binom{a-x+k-1}{a-x}\binom{b-y+l-1}{b-y}\mathbf{I}_{ab}[x,y] \tag{30}$$

and

$$\mathbf{R}_{mn}[k,l] = \sum_{x=1}^{a}\sum_{y=1}^{b}\binom{x+k-2}{x-1}\binom{y+l-2}{y-1}\mathbf{I}_{ab}[x,y], \tag{31}$$

respectively.

Since combinatorial numbers can be expressed in a polynomial form as:

$$r_q^p(t) = \binom{p-t+q-1}{p-t} = \frac{(-1)^{q-1}}{(q-1)!}\prod_{i=1}^{q-1}(t-p-i), \tag{32}$$

where the order of the polynomial is $q-1$. Then, accumulation moments can be seen as the projection coefficients of an image onto the basis set defined by these polynomials.

Hence, for direct accumulation moments, the underlying polynomial basis is $r_k^a(x)\, r_l^b(y)$, for $k = 0,\ldots,m-1$, and $l = 0,\ldots,n-1$. And, for reverse accumulation moments, the polynomial basis is $r_k^{-1}(-x)\, r_l^{-1}(-y)$, for $k = 0,\ldots,m-1$, and $l = 0,\ldots,n-1$.

Notice that the coefficients of $r_q^p(t)$ correspond to the elements of $\mathbf{G}_q^p[:,q]$. Likewise, the coefficients of $r_q^{-1}(t)$ correspond to the elements of $\mathbf{H}_q[:,q]$.

Appendix B

It is obvious that the direct accumulation moments of an image $\mathbf{I}_{ab}[k,l]$ and the reverse accumulation moments of the reflected and displaced image $\mathbf{I}_{ab}[(a+1)-k,(b+1)-l]$ are equal. Let us distinguish the moment matrices associated with each image by numbering them 1 and 2, respectively. Then, the relationship between the geometric moments of both images is:

$$\mathbf{M2}_{mn} = \left(\mathbf{T}_m^{-(a+1)}\right)^t\, \mathbf{D}_m\, \mathbf{M1}_{mn}\, \mathbf{D}_n\, \mathbf{T}_n^{-(b+1)},$$

where $\mathbf{D}_p$ are diagonal matrices of the form $\mathbf{D}_p = \mathrm{diag}\{1,-1,\ldots,(-1)^{p-1}\}$ that account for the reflection of the image, and $\mathbf{T}_p^q$ are obtained according to Corollary 2.

Using corollary 4,

$$\mathbf{R2}_{mn} = (\mathbf{H}_m)^t\, \left(\mathbf{T}_m^{-(a+1)}\right)^t\, \mathbf{D}_m\, \mathbf{M1}_{mn}\, \mathbf{D}_n\, \mathbf{T}_n^{-(b+1)}\, \mathbf{H}_n.$$

Since $\mathbf{R2}_{mn} = \mathbf{L1}_{mn}$, and from Corollary 3, one obtains:

$$\mathbf{G}_p^q = \mathbf{D}_p\, \mathbf{T}_p^{-(q+1)}\, \mathbf{H}_p, \tag{33}$$

Hence,

$$\left(\mathbf{G}_p^q\right)^{-1} = (\mathbf{H}_p)^{-1}\, \mathbf{T}_p^{(q+1)}\, \mathbf{D}_p. \tag{34}$$

Now, using corollaries 3 and 4, the direct ($\mathbf{L}_{mn}$) and reverse ($\mathbf{R}_{mn}$) accumulation moment matrices of an image $\mathbf{I}_{ab}$ can be related as follows:

$$\mathbf{R}_{mn} = (\mathbf{W}_m^a)^t\, \mathbf{L}_{mn}\, \mathbf{W}_n^b, \tag{35}$$

and

$$\mathbf{L}_{mn} = (\mathbf{W}_m^a)^t\, \mathbf{R}_{mn}\, \mathbf{W}_n^b, \tag{36}$$

where

$$\mathbf{W}_p^q[k,l] = \begin{cases} \frac{(-1)^{k+1}}{(l-1)!}\, B_{l-1,l-1}^{q+k-1}, & \text{if } \ l \geq k, \\ 0, & \text{otherwise.} \end{cases} \tag{37}$$

Note that $\left(\mathbf{W}_p^q\right)^{-1} = \mathbf{W}_p^q$.

References

[1] P. Baglietto, M. Maresca, M. Migliardi, and N. Zingirian, "Image processing on high performance RISC systems," *Proceedings of the IEEE*, Vol. 84, No. 7, pp. 917-930, 1996.

[2] L. Bronshtein, *Handbook of mathematics for engineers*, 4th. edition. Moscow: Mir Publishing Co., 1982, chapter II, pp. 158-159.

[3] Y-Q. Chen, M.S. Nixon and D.W. Thomas, "Statistical geometrical features for texture classification," *Pattern Recognition*, Vol. 28, pp. 537-552, 1995.

[4] S. Ghosal and R. Mehrotra, "Robust optical flow estimation," *Proc. of the 1st IEEE Int. Conf. on Image Processing*, Part. 2, pp. 780-784, 1994.

[5] S. Ghosal and R. Mehrottra, "A moment-based unified approach to image feature detection," *IEEE Trans. on Image Processing*, Vol. 6, No. 6, pp. 781-793, 1997.

[6] G.H. Golub and C.F. van Loan, *Matrix computations*, The Johns Hopkins University Press, Baltimore, Maryland, 1996.

[7] M. Hatamian, "A real-time two-dimensional moments generating algorithm and its single chip implementation," *IEEE Trans. on Acoustics, Speech and Signal Processing*, Vol. 34, No. 3, pp. 546-553, 1986.

[8] H.S. Hsu, "Moment preserving edge detection and its application to image data compression," *Optical Engineering*, Vol. 32, No. 7, pp. 1596-1608, 1993.

[9] M-K. Hu, "Visual pattern recognition by moment invariants," *IRE Trans. on Information Theory*, pp. 179-187, 1962.

[10] X.Y. Jiang and H. Bunke, "Simple and fast computation of moments," *Pattern Recognition*, Vol. 24, No. 8, pp. 801-806, 1991.

[11] D.C. Kar and V.V.B. Rao, "A CORDIC-based unified systolic architecture for sliding window applications of discrete transforms," *IEEE Trans. on Signal Processing*, Vol. 44, No. 2, pp. 441-444, 1996.

[12] C.-Y. Lee and D.B. Cooper, "Structure from motion: A region based approach using affine transformations and moment invariants," *IEEE Int. Conf. on Robotics and Automation*, pp. 120-127, 1993.

[13] J.-G. Leu, "Computing a shape's moments from its boundary," *Pattern Recognition*, Vol. 24, No. 10, pp. 949-957, 1991.

[14] B. Li, "High-order moment computation of grey-level images," *IEEE Trans. on Image Processing*, Vol. 4, No. 4, pp. 502-505, 1995.

[15] Y. Li, "Reforming the theory of invariant moments for pattern recognition," *Pattern Recognition*, Vol. 25, No. 7, pp. 723-730, 1992.

[16] E.P. Lyvers, O.R. Mitchell, M.L. Akey and A.P. Reeves, "Subpixel measurements using a moment-based edge operator," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 11, No. 12, pp. 1293-1309, 1989.

[17] V. Markandey and R.J.P. deFigueiredo, "Robot sensing techniques based on high-dimensional moment invariants and tensors," *IEEE Trans. on Robotics and Automation*, Vol. 8, No. 2, pp. 186-194, 1992.

[18] P.M. Narendra, "A separable median filter for image noise smoothing," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 3, No. 1, pp. 20-29, 1981.

[19] J. Martinez, *Accumulation moments. Theory and applications*, Ph.D. thesis, Polytechnical University of Catalonia, 1998.

[20] J. Martinez and F. Thomas, "A reformulation of grey-level image geometric moment computation for real-time applications," *1996 IEEE Conf. on Robotics and Automation*, Vol. III, pp. 2315-2320, 1996.

[21] J. Martinez and F. Thomas, "Fast actualization of moments in sliding window applications", *EUSIPCO-98. Ninth European Signal and Image Processing Conference*, Vol. I, pp. 145-149, 1998.

[22] R. Mukundan and K.R. Ramakrisnan, *Moment functions in image analysis. Theory and applications*, World Scientific Publishing Co., 1998.

[23] T.H. Reiss, "The revised fundamental theorem of moment invariants," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 13, No. 8, pp. 830-834, 1991.

[24] A.P. Reeves, "A parallel mesh moment computer," *Proc. of the 6th Int. Conf. on Pattern Recognition*, pp. 465-467, 1982.

[25] S. Seely, "Laplace transforms," chapter 5 in *The Transforms and Applications Handbook*, Ed. A. Poularikas, Florida: CRC Press and IEEE Press, pp. 334-346, 1996.

[26] T-W. Shen, D.P.K. Lun, and W.C. Siu, "Fast algorithm for 2-D image moments via the Radon transform," *1996 Int. Conf. on Acustics, Speech and Signal Processing*, Vol. 3, pp. 1327-1330, 1996.

[27] H.M. Stellakis, "Adaptive computation of higher order moments and its systolic realization," *Int. Journal of Adaptive Control and Signal Processing*, Vol. 10, No. 2-3, pp. 283-302, 1996.

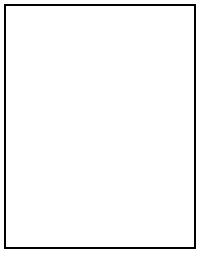
[28] C-H. Teh and R.T. Chin, "On image analysis by the method of moments," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 10, No. 4, pp. 496-513, 1988.

[29] M. Tuceryan, "Moment based texture segmentation," *Pattern Recognition Letters*, Vol. 15, No. 7, pp. 659-667, 1994.

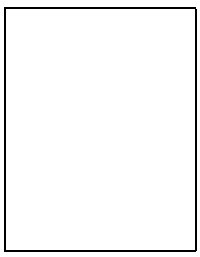
[30] W-H. Wong, W-C. Siu and K-M. Lam, "Generation of moment invariants and their use for character recognition," *Pattern Recognition Letters*, Vol. 16, pp. 115-123, 1995.

[31] L. Yang, F. Albregtsen and T. Taxt, "Fast computation of 3-D geometric moments using a discrete Gauss' theorem, *Proc. CAIP'95*, Lecture Notes in Computer Science, Vol. 970, pp. 649-654, 1995.

[32] L.P. Yaroslavsky, "Local adaptive image restoration and enhancement with the use of DFT and DCT in a running window," *Wavelet Applications in Signal Processing IV*, Denver, Colorado, SPIE Proc. Series, Vol. 2825, pp. 1-13, 1996.



Judit Martínez received the B.Sc. degree in 1993 and the PhD degree (with honors) in 1998, both in telecommunications engineering from the Polytechnic University of Catalonia. She developed her research at the Industrial Robotics Institute of the Spanish High Council for Scientific Research. Since 1999 she is member of the research staff of the Computer Vision Center, a R&D center founded by the Autonomous University of Barcelona and the Autonomous Government of Catalonia. She has been principal researcher of several industrial and research projects related to computer vision technologies. Her research interests include industrial applications of machine vision, efficient algorithms for low-level image processing, multiresolution mathematical models, statistical clustering, pattern classification and inverse problems. Her web page is `http://www.cvc.uab.es/shared/staff/all_staff/judit.htm`.



Federico Thomas received the B.Sc. degree in telecommunications engineering in 1984 and the Ph.D. degree (with honors) in computer science in 1988, both from the Polytechnic University of Catalonia. Since March 1990, he has been a Research Scientist in the Industrial Robotics Institute of the Spanish High Council for Scientific Research. His research interests are in geometry and kinematics, with applications to robotics, computer graphics and machine vision. He has published more than fifty research papers in international journals and conferences. His web page is `http://www-iri.upc.es/people/thomas`.