Line moments and invariants for real time processing of vectorized contour data

Georg Lambert[1], Hua Gao[2]

[1] Technical University of Darmstadt, Control System Theory and Robotics Dept., Landgraf-Georg-Str.4, 64283 Darmstadt, Germany
[2] Technical University of Darmstadt, Institute of Production Engineering and Machine Tools, Petersenstr. 30, 64287 Darmstadt, Germany

Abstract. In this article a new concept for real time analysis of images based on the vectorized contours is presented. Given the polygone approximation of contours the geometric moments can be formulated for line contours. The resulting line moments (LM's) can be calculated for an arbitrary collection of contour fragments. Neither the existence of an area nor a closed contour is prerequisite to characterize an object by line moments. Due to these properties of the LM's time consuming preprocessing of noisy object contours is not required. Even objects with structured surfaces can be characterized by LM's. The formalism to apply the geometric moments to contour patterns is derived. A direct and a recursive algorithm to efficiently compute the LM's is given. As an application of the LM's an analysis system for structured textures is presented and results are discussed.

1 Indroduction

The application of image processing systems in industrial environments gains more and more in significance. One reason why image processing has been unattractive for many applications in the past was the lack of computation power to meet real time requirements.
In this paper we present an approach to real time analysis of images based on the contour data of the scene. In this context we consider the extraction of contours as a standard operation, which is universally applicable to many image processing and analysis tasks. Special hardware processors or implementations on signal–processor basis can realize contour extraction in video real time.
For this work we used a hardware contour vectorizer called *VectEx*[3] which delivers a polygone approximation of the gray level edges of images in video real time. (For a description of the processor see [HERR90].) As an example figure 1a shows the vectorized contours of a structured texture.
Based on the contour data we developed a fast recursive algorithm to compute the line moments (LM's) of patterns. Therefore the extension of the conventional moment definition (e.g. [HU62]) to LM's is one main aspect of this work. Due to this interpretation the computation of moments is no longer restricted

[3] Distributed by ELTEC Elektronik GmbH, D–55071 Mainz, Germany

to elements with a definite area and a closed contour. An element is considered to be composed of an arbitrary collection of line fragments.
Based on LM's the entire theory of moments and moment invariants can be applied for element recognition and analysis.

2 Line moments of contour patterns

Applying a contour vectorization on objects of a real scene yields in object contours broken down into lots of fragments. To extract objects from the contour data logically related fragments have to be recognized. This might be done by some rule based algorithms or by statistical analysis and segmentation. In [AMEL94] an algorithm is described, which assigns all contour fragments situated in local neighbourhoods of edge points recursively to the same object. Non–overlapping and not directly neighbouring objects like in figures 1a and 5a can be segmented efficiently through this algorithm.

Given the contour fragments C_j^* of an object B (figure 1b), the contour pattern C of B can be expressed as

$$C = \bigcup_j C_j^* . \tag{1}$$

If the contour fragments are approximated by polygones, the contour can be expressed in terms of line fragments c_{ji}

$$C_j^* = \bigcup_i c_{ji} \quad , \qquad C = \bigcup_j \bigcup_i c_{ji} . \tag{2}$$

A contour pattern C of an object B is consequently defined as an arbitrary collection of line fragments. There is no need to define either the area or the perimeter of an object.

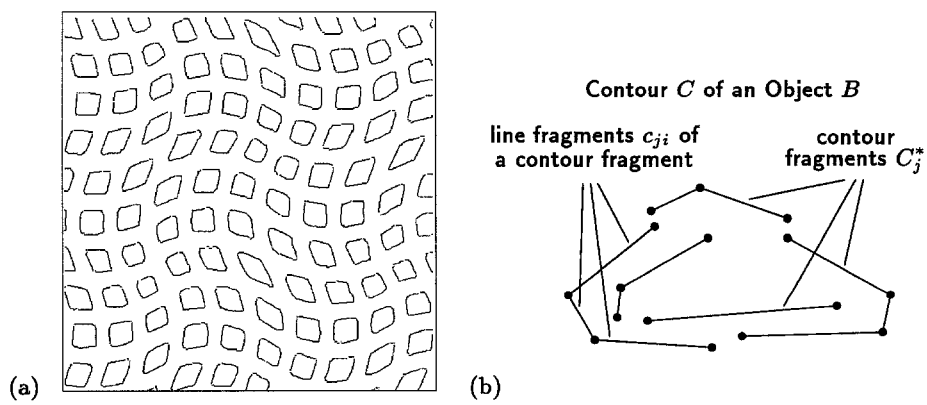

Fig. 1. **(a)** Example of a vectorized texture scene. **(b)** Definition of a contour pattern

After the definition of contour patterns we will focus on the derivation of line moments. Let us first consider the well known geometric moments of the form $m_{pq}^{(2)} = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} x^p y^q f(x,y)\, dx\, dy$. Given a gray level image we can understand $f(x,y)$ as the gray level values. If the domain of definition $D(B)$ of an object

B is known, e.g. through segmentation, the object is uniquely determined by its moments

$$m_{pq}^{(2)} = \int\int_{(x,y)\in D(B)} x^p y^q f(x,y)\, dA \qquad p,q = [0,1,\ldots,\infty] \tag{3}$$

according to the Uniqueness Theorem ([HU62]). If we consider line contours, the line moments along a contour C are given by

$$m_{pq}^{(1)} = \int_C x(s)^p y(s)^q f(s)\, ds \tag{4}$$

where $f(s)$ is called the linear density and s stands for the arc length. In the case of describing the contour shape, $f(s)$ can be set to $f(s) = 1$. Equation (4) can also be expressed in terms of the cartesian coordinates x or y

$$m_{pq}^{(1)} = \int_C x^p y(x)^q \sqrt{1+y'(x)^2}\, dx = \int_C x(y)^p y^q \sqrt{1+x'(y)^2}\, dy. \tag{5}$$

If the contour pattern is approximated by a polygone with edge–points (x_i, y_i), $i = 1,2,\ldots,n+1$, each line segment can be parameterized to

$$c_i: \quad y = a_i x + y_i - a_i x_i\ , \quad x_i \le x \le x_{i+1}\ , \quad i = 1,2,\ldots,n \tag{6}$$

whith $a_i = (y_{i+1} - y_i)/(x_{i+1} - x_i)$ as the slope of segment i. (The index j is omitted in the following.) Let D_i be the contribution of c_i to the line integral of eq.(5) the moments are given by

$$m_{pq}^{(1)} = \textstyle\sum_{i=1}^n D_i\ , \qquad D_i = \int_{c_i} x^p y^q \sqrt{1+a_i^2}\, dx\ . \tag{7}$$

Inserting eq.(6) in (7) we get

$$\begin{aligned} D_i &= \int_{x_i}^{x_{i+1}} x^p (a_i x + y_i - a_i x_i)^q \sqrt{1+a_i^2}\, dx \\ &= \sqrt{1+a_i^2} \cdot \textstyle\sum_{k=0}^q \left\{ \binom{q}{k} a_i^k (y_i - a_i x_i)^{q-k} \cdot \frac{x_{i+1}^{p+k+1} - x_i^{p+k+1}}{p+k+1} \right\}\ . \end{aligned} \tag{8}$$

If c_i is vertical, we take the alternative parameterization

$$c_i: \quad x = x_i\ , \quad y_i \le y \le y_{i+1}\ , \quad i = 1,2,\ldots,n \tag{9}$$

which leads to

$$D_i = \int_{y_i}^{y_{i+1}} x^p y^q \sqrt{1+x'(y)^2}\, dy = \int_{y_i}^{y_{i+1}} x_i^p y^q\, dy = x_i^p \cdot \frac{y_{i+1}^{q+1} - y_i^{q+1}}{q+1}\ . \tag{10}$$

To obtain invariance under translation the central moments

$$\mu_{pq}^{(1)} = \int_C (x(s) - \overline{x})^p (y(s) - \overline{y})^q f(s)\, ds \tag{11}$$

have to be calculated where $\overline{x} = m_{10}^{(1)}/m_{00}^{(1)}$, $\overline{y} = m_{01}^{(1)}/m_{00}^{(1)}$. To obtain scaling invariance the following normalisation by the contour length l_C leads to the result

$$\eta_{pq}^{(1)} = \int_C \left(\frac{x(s)-\overline{x}}{l_C}\right)^p \left(\frac{y(s)-\overline{y}}{l_C}\right)^q \frac{ds}{l_C} = \frac{\int_C (x(s)-\overline{x})^p (y(s)-\overline{y})^q\, ds}{l_C^{p+q+1}} = \frac{\mu_{pq}^{(1)}}{(\mu_{00}^{(1)})^\gamma} \tag{12}$$

where $\gamma = p+q+1$. It has to be stated, that in the case of LM's γ differs from the exponent, Hu derived for area based moments in [HU62]. The normalisation of eq.(12) is also given in [SARD94].

3 Algorithms

In this section we will give a direct and a fast, recursive algorithm to compute the LM's. The algorithm is similar to those given in [JIAN91]. Nevertheless it has to be emphasized that Jiang and Bunke computed area based moments out of the closed contours of objects through the Green's formula which is totally different from the approach to LM's presented here.
Figure 2 gives the direct algorithm to compute the LM's of polygonal contour patterns.

forall (pq) **do** $m_{pq}^{(1)} = 0$;
for $i = 1$ **to** n **do begin**
  if vertical(c_i) **then**
    forall (pq) **do** $m_{pq}^{(1)} = m_{pq}^{(1)} + D_i$; (* D_i by eq.(10) *)
  else
    forall (pq) **do** $m_{pq}^{(1)} = m_{pq}^{(1)} + D_i$; (* D_i by eq.(8) *)
end

Fig. 2. Direct algorithm to compute the line moments

The direct algorithm does not take into account relationships between the moments of higher order to those of lower order. To derive a recursive algorithm we define

$$A_i(p,q) = \int_{x_i}^{x_{i+1}} x^p \, (a_i x + y_i - a_i x_i)^q \, dx \tag{13}$$

(see eq.(8)). For $q \geq 1$ we get

$$\begin{aligned} A_i(p,q) &= \int_{x_i}^{x_{i+1}} x^p \, (a_i x + y_i - a_i x_i)^{q-1} (a_i x + y_i - a_i x_i) \, dx \\ &= a_i \int_{x_i}^{x_{i+1}} x^{p+1} \, (a_i x + y_i - a_i x_i)^{q-1} \, dx \; + \\ &\quad (y_i - a_i x_i) \int_{x_i}^{x_{i+1}} x^p \, (a_i x + y_i - a_i x_i)^{q-1} \, dx \\ &= a_i A_i(p+1, q-1) + (y_i - a_i x_i) A_i(p, q-1) \; . \end{aligned} \tag{14}$$

To start the recursion we have

$$A_i(p,0) = \int_{x_i}^{x_{i+1}} x^p \, dx = \frac{x_{i+1}^{p+1} - x_i^{p+1}}{p+1} \; . \tag{15}$$

The recursive dependencies of the terms $A_i(p,q)$ are illustrated in figure 3a for an example with $(p,q) \leq 4$.
The notation $(p_1, q_1) \rightarrow (p_2, q_2)$ expresses that for the calculation of $A_i(p_2, q_2)$ the value of $A_i(p_1, q_1)$ has to be known. Based on these dependencies the order of calculation for all $A_i(p,q)$ with $(p,q) \leq 4$ can be derived as shown in figure 3b. The variables $B[0] \ldots B[4]$ (see also fig. 4) store temporary results during

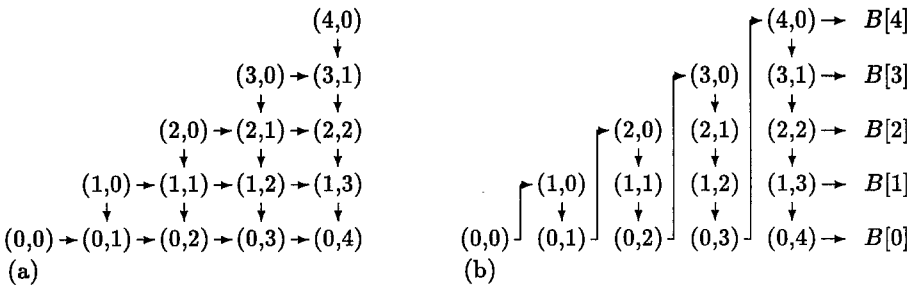

Fig. 3. **(a)** recursive dependencies of $A_i(p,q)$ **(b)** order of computation

for $order = 0$ **to** max_order **do begin**
  $B[order] = A_i(order, 0);$ (* by eq.(15) *)
  $m^{(1)}_{order,0} = m^{(1)}_{order,0} + \sqrt{1+a_i^2} \cdot B[order];$
  for $p = order - 1$ **downto** 0 **do begin**
    $q = order - p;$
    $B[p] = a_i B[p+1] + (y_i - a_i x_i) B[p];$ (* eq.(14) *)
    $m^{(1)}_{p,q} = m^{(1)}_{p,q} + \sqrt{1+a_i^2} \cdot B[p];$
  end;
end;

Fig. 4. Recursive algorithm to compute the line moments

computation. To evaluate the LM's according to eq.(7) we get (compare (8) and (13))

$$D_i = \sqrt{1+a_i^2} \cdot A_i(p,q) \ . \tag{16}$$

Hence the recursive algorithm shown in figure 4 is appropriate to calculate the LM's in a very efficient way.

4 Experimental results

As an application of the LM's we present an analysis system for quality control of structured textures in this section.
It has to be emphasized, that the whole theory of moment invariants is applicable to LM's. Since the various types of moments, e.g. Legendre Moments, Zernike Moments, Complex Moments and others can be expressed through the geometric moments ([TEH88]) their formulation as line moments is possible.
In our experiments we calculated line moment invariants (LMI's) to characterize the extracted texture primitives. We implemented the 7 well known Hu moment invariants ([HU62]), as they are still a benchmark for other invariants. To extract texture primitives from the vectorized contour data a rule based algorithm described in [AMEL94] was applied. In a learning phase the LMI's of the different texels are calculated from texture samples and stored as feature vectors. During the defect detection phase the following tasks have to be applied to each texture scene: texel segmentation, calculation of LMI's and classification with a Mahalanobis distance classifier. The processing is based on the output of the hardware vectorizer *VectEx*.

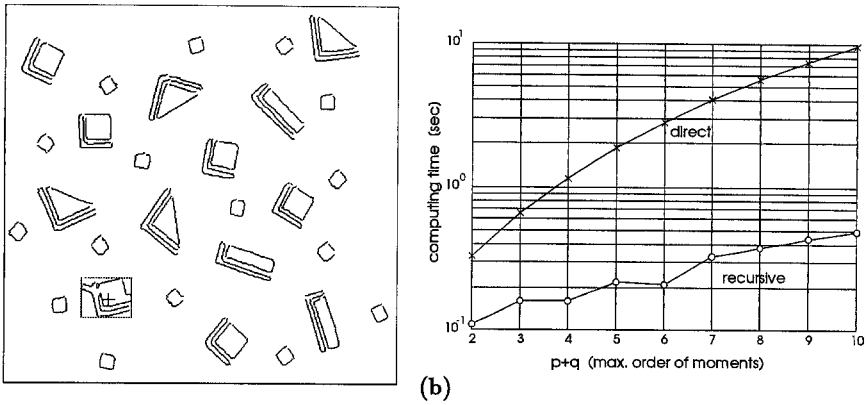

Fig. 5. **(a)** Detection of the corrupted texture element marked by a frame. **(b)** Computing time for the direct and the recursive algorithm (80486DX2-66MHz processor).

Figure 5a shows the result of a texture analysis with 4 different texel classes. The corrupted texture primitive is detected by the system. The computation has been done on a PC with a 80486 processor, 66 Mhz clock rate and VL bus. To calculate the HU invariants all moments up to the maximum order of $(p+q) \leq 3$ have to be determined. Using the iterative algorithm, the moment computation takes 0.16 sec for the above scene. Computation of all other tasks as texture primitive generation, invariants calculation and classification takes 0.15 sec. Hence the whole analysis requires 0.31 sec. on a scene with 1505 polygone points in 213 contours and 29 texture elements.
To point out the effectiveness of the iterative algorithm compared with the direct one, figure 5b depicts computation times for all the line moments up to an order $(p+q) \leq N$. For $N = 10$ the iterative algorithm is 20 times faster than the direct one.

References

[AMEL94] Jörg Amelung, Georg Lambert und Jörg Pfister: Ein vektorbasiertes Verfahren zur schnellen Fehlererkennung in strukturierten Texturen; Proc.16. DAGM Symp. Mustererkennung, Wien, 1994, pp. 666–675.

[HERR90] E.Herre und R.Massen: Symbolische konturorientierte Bildverarbeitung durch Echtzeit-Polygon-Approximation; Technisches Messen TM Vol.57 Nr.10, 1990, pp.384-388.

[HU62] Ming–Kuei Hu: Visual Pattern Recognition by Moment Invariants; IRE Trans. on Information Theorie, Vol. 8, 1962, pp.179-187.

[JIAN91] X. Y. Jiang, H. Bunke: Ein konturbasierter Ansatz zur Berechnung von Momenten; Proc.13. DAGM Symp. Mustererkennung, Munich, 1991, pp. 143-150.

[SARD94] H. K. Sardana, M. F. Daemi and M.K. Ibrahim: Global description of edge patterns using moments; Pattern Recognition, Vol. 27, No. 1, 1994, pp. 109-118.

[TEH88] Cho-Huak Teh, Roland T. Chin: On Image Analysis by the Methods of Moments; Trans. on Pattern Analysis and Machine Intelligence, Vol. 10, No. 4, 1988, pp.496-513.