# Subpixel edge refinement using deformable models

**Frédéric Bouchara[1,*] and Sofiane Ramdani[2]**

[1]*UMR CNRS 6168, Laboratoire des Sciences de l'Information et des Systèmes, Université du Sud Toulon-Var, 83957 La Garde Cedex, France*

[2]*Equipe d'Accueil 2991 Efficience et Déficience Motrices, Université de Montpellier I, 34090 Montpellier, France*

**Corresponding author: bouchara@univ-tln.fr*

Received August 14, 2008; revised December 9, 2008; accepted January 21, 2009; posted February 2, 2009 (Doc. ID 100163); published March 18, 2009

We propose an approach for subpixel edge refinement based on deformable models. In our approach, the result is constrained by two kinds of information: orientation information derived from the gradient and position information. The problem is formulated in a statistical framework: the likelihood function of the observations is computed and used in a classical maximum *a posteriori* estimator. Two different models are proposed: a parametric model based on B-spline functions and a sampled model. Experiments on both synthetic and natural images are described that show the adequacy and effectiveness of these algorithms. © 2009 Optical Society of America

*OCIS codes:* 100.2000, 100.6640.

## 1. INTRODUCTION

Image segmentation, or the extraction of the boundaries of objects, is one of the most important problems in image and video processing. Indeed, numerous machine vision tasks are based on this low-level information, and the precise localization of image edges can be of critical importance. Thus, in a 3D vision system the camera calibration step and the stereo disparity maximum *a posteriori* (MAP) estimation (in the feature-based approach) are usually based on subpixel edge extraction [1].

Several approaches used to estimate the subpixel position can be found in the literature. In moment-based algorithms, the edge is usually modeled as an ideal step of which parameters are linked to the moments of the image. Different kinds of moments have been proposed. In [2], Tabatabai and Mitchell used the first three gray level moments to estimate the subpixel edge position. More recently, different techniques based on spatial moments have been proposed in the case of monochromatic [3–5] or color images [6]. The Zernike moments have also been successfully employed [7,8].

Another well-known approach is based on the interpolation of the gray level or gradient image. In [9], Nomura *et al.* used the normal function to approximate the first derivative perpendicular to the edge. Other studies are based on linear [10], quadratic [11], B-spline [12], or other kinds of nonlinear approaches [13,14].

In another study [15], Kisworo *et al.* used a local energy function to match the edge among eight parametric models. A least-squared error method was then used to refine the fit. Others extended the subpixel localization technique to corners and vertex [16,17] or circular edges [18,19]. However, in all of these approaches, the estimation is based on a local computation that leads to a sensitivity of the algorithm to perturbations such as noise or texture.

Deformable models, such as snakes, are another approach that can be used to locate the edge with subpixel accuracy. The snake model, as originally proposed by Kass *et al.* [20], is described as a virtual object whose deformation is guided by external and internal forces. The location of the edge, which can be noninteger, is obtained as the equilibrium state of these forces. In contrast with the previous local algorithms, the snake model includes by nature an *a priori* information (its shape). In more recent works, the deformable contour is modeled using piecewise polynomial functions (B-spline snakes) [21,22]. This formulation allows local control, compact representation, and it is mainly characterized by few parameters. In [23], Venkatesh *et al.* proposed such a parametric model for the interpolation of subpixel edge coordinates. In this algorithm, the subpixel measurements are interpolated by using cubic Hermites functions. In the proposed approach, however, the different measurements have equal weight.

However, when fitting the parameters of a model to noisy data, it is well known that the fit is improved if an estimate of the uncertainty is incorporated into the process. This is usually accomplished with MAP estimation [or a maximum likelihood (ML) method when an *a priori* model is not available]. This approach usually leads to a weighted least-squares fit that minimizes the Mahalanobis distance between the model and the data [24].

In this paper, we propose a MAP-based algorithm for subpixel edge refinement by using deformable models. In our approach, the result is constrained by two kinds of information: an orientation information (computed from the gradient of the image) and a position information. The statistical properties of the observations are computed and used in the MAP estimation, which insures an efficient filtering of the noisy data. This paper is the extended version of a preliminary work where only the B-snake model was considered [25]. In this study, we extend this approach to the case of the sampled model, i.e., the original snake model proposed in [20].

The remainder of this paper is organized as follows. In

 © 2009 Optical Society of America

Section 2 we briefly recall the principles of the active contour models. Section 3 is devoted to the derivation of the proposed approach. Experiments on a variety of images are presented in Section 4, after which we conclude in Section 5.

## 2. ACTIVE CONTOUR AND RELATED APPROACHES

### A. Sampled Model

As originally proposed, a snake is a curve $v(t)=[x(t),y(t)]$ subject to two different forces that represent its elasticity and a potential field. Here, $t$ represents the curvilinear abscissa defined in the interval [0, 1]. The equilibrium configuration is a compromise between these two forces and corresponds to the minimum of a global energy function:

$$v = \arg\min_{v} E_{int}(v) + E_{ext}(v,I), \tag{1}$$

where $E_{int}$, the internal energy, is defined by

$$E_{int} = \int \frac{1}{2}\left[\alpha\left|\frac{\partial v}{\partial t}\right|^2 + \beta\left|\frac{\partial^2 v}{\partial t^2}\right|^2\right]\mathrm{d}t. \tag{2}$$

The external energy $E_{ext}$ is usually a function of the image gradient.

In a Bayesian viewpoint, snakes are interpretable as MAP contour estimators where the internal and external energies are associated with the *a priori* probability function and the likelihood function, respectively:

$$P(v) = \frac{1}{Z_{int}}\exp[-E_{int}(v)], \tag{3}$$

$$P(I/v) = \frac{1}{Z_{ext}}\exp(-E_{ext}(v,I)), \tag{4}$$

and where $Z_{ext}$ and $Z_{int}$ are normalizing constants. The MAP estimation is then classically given by

$$v = \arg\max_{v} P(I/v)P(v). \tag{5}$$

With the previous definitions of the prior and the likelihood, this estimation leads to Eq. (1) if the $Z$ are constants.

In its sampled version, the snake is defined by the set of points $v_i=(x_i,y_i)=[x(i\cdot\Delta),y(i\cdot\Delta)]$. Approximating the derivatives with finite differences, the internal energy has the expression

$$E_{int} = \sum_i \frac{\alpha}{2h^2}|v_i - v_{i-1}|^2 + \frac{\beta}{2h^4}|v_{i-1} - 2v_i + v_{i+1}|^2. \tag{6}$$

From this expression, problem (1) becomes equivalent to the following Euler equations in matrix form:

$$\mathbf{Ax} + \mathbf{f}_x(\mathbf{x},\mathbf{y}) = 0, \tag{7}$$

$$\mathbf{Ay} + \mathbf{f}_y(\mathbf{x},\mathbf{y}) = 0, \tag{8}$$

where $\mathbf{f}_{xi}=\partial E_{ext}/\partial x_i$, $\mathbf{f}_{y_i}=\partial\partial E_{ext}/\partial y_i$, $\mathbf{A}$ is a pentadiagonal banded matrix, $\mathbf{x}=(x_1,\ldots,x_M)$ and $\mathbf{y}=(y_1,\ldots,y_M)$.

In the original problem, these equations are solved by using a gradient-descent method that leads to an iterative updating rule of the snake.

### B. Parametric Approaches

The parametric models constitute an extension of the original snake model in which the contour itself is parametrically described. Such an approach presents several advantages: the contour is represented by few parameters, and by using a low-order parameterization the internal energy term can be dropped. Among the different models proposed in the literature we are concerned with the B-spline approach, which describes the contour with piecewise polynomial functions. To present a self-sufficient paper we give here a brief theoretical review of B-spline contour formulation (for more details see [26]).

Let $\{t_0,t_1,\ldots,t_{k-1}\}$, $t_i\in[0,1]$ be the set of so-called knots. By definition, spline functions are polynomials inside each interval $[t_{i-1},t_i]$ and exhibit a certain degree of continuity at the knots. The set $\{B_{i,n(t)}, i=0,\ldots,k-n-1\}$ of the so-called B-splines constitutes a basis for the linear space of all the splines on $[t_n,t_{k-n}]$. Thus, a spline curve $f(t)$ of degree $n$ is given by

$$f(t) = \sum_{i=0}^{N-1} p_i B_{i,n}(t), \tag{9}$$

where $p_i$ are the weights applied to the $N=k-n$ basis functions $B_{i,n}$. The B-spline functions are defined by the Cox–deBoor recursion formulas

$$B_{i,0}(t) = \begin{cases} 1, & \text{if } t_i \leqslant t \leqslant t_{i+1} \\ 0, & \text{otherwise} \end{cases}, \tag{10}$$

and

$$B_{i,n}(t) = \frac{(t-t_i)B_{i,n-1}(t)}{t_{i+n}-t_i} + \frac{(t_{i+n+1}-t)B_{i+1,n-1}(t)}{t_{i+n+1}-t_{i+1}}, \tag{11}$$

The $B_{i,n}(t)$ are nonnegative, and $B_{i,n}(t)\geqslant 0$ and verify the partition of unity property: $\Sigma_i B_{i,n}(t)=1$ for all $t$.

In the remainder of the paper, cubic B-spline will be used, and without loss of generality we will drop the index $n$. Relation (9) can be expressed compactly in matrix notation as

$$f(t) = B^T(t)\cdot\theta. \tag{12}$$

$B(t)$ is the vector of B-spline functions: $B(t)=(B_0(t),B_2(t),\ldots B_{N-1}(t))$, and $\theta$ is the vector of weights: $\theta=(p_0,\ldots,p_{N-1})^T$.

The $\mathbb{R}^2$ version of relation (9) describes an open parametric snake on the plane: $v(t)=(x(t),y(t))$. The $p_i=(p_{xi},p_{yi})$ are now 2D vectors and are called the control points. To describe a closed curve, which is often the case of interest for boundary representation, a periodic extension of the basis functions and of the knot sequence is generally used.

## 3. SUBPIXEL SNAKE MODEL

### A. Observational Model

In this subsection we shall derive the expression of the likelihood of the observation. In order to get tractable

computations, we have developed our model in the case of a white additive Gaussian noise $b$. We denote by $\sigma_b$ its standard deviation.

Let $v_e=\{v_{ei}=(x_{ei},y_{ei}), i\in\{1,\ldots,L\}\}$ be a set of edge pixels with integer coordinates computed by a classical algorithm (such as the classical active contour algorithm). For each pixel of this set we suppose available the observation vector $O_i=(X_i,H_i)$. The vector $H_i=(H_{xi},H_{yi})$ is the local gradient of the image and $X_i$ corresponds to an estimation of the subpixel position of the edge along $H_i$. The value $X_i$ is estimated in the local coordinate system of the pixel thanks to a quadratic interpolation (see [27] and Appendix A for further explanation). These variables are assumed to be the observation version of the "true" variables $(Y_i,G_i)$. The Cartesian coordinates of the observation $O_i$ will be denoted $(x_{oi},y_{oi})$ in what follows.

Now let $v_s$ be the subpixel contour to be estimated. In the case of the B-snake model, this contour is described by a couple of spline functions $v_s(t)=(x_s(t),y_s(t))=(B(t)\cdot\theta_x,B(t)\cdot\theta_y)$, where $\theta=(\theta_x,\theta_y)$ is the vector of $k$ control points (with $L\gg k$).

In the case of the sampled model, the subpixel contour is described by a set of edge points (the snaxels): $v_s=\{v_{sj}\}=\{(x_{sj}=x_s(j\cdot\Delta_s),y_{sj}=y_s(j\cdot\Delta_s)), j\in\{1,\ldots,M\}\}$ (with $M=m\cdot L$, $m\in\mathbb{N}^*$).

The likelihood $P(O_i/v_s)$ can be expressed as the product of two elementary probability-density functions (PDFs):

$$P(O_i/v_s)=P(X_i/H,v_s)P(H_i/v_s). \tag{13}$$

In order to derive the first factor of this expression, we make the assumption that $X_i$ depends only on $v_s(t_i)=[x_s(t_i),y_s(t_i)]$, the corresponding edge point in the direction $H_i$ of which the polar coordinates are $(Y_i,H_i)$. For simplicity, the $t_i$ coordinates will be distributed using a uniform strategy. Thus, in the parametric case $t_i=i/L$, and in the sampled case $t_i=i\cdot m\cdot\Delta_s$. The $j$th snaxel, with $j=i\cdot m$ corresponding to the $t_i$ coordinate, will be denoted $v_{sj(i)}$.

The PDF $P(X_i/H_i,Y_i)$ will be modeled by a Gaussian law. According to our tests, this approximation remains accurate even for significant noise levels. Then we get

$$P(X_i/H_i,v_s)=P(X_i/H_i,Y_i)=\frac{1}{\sqrt{2\pi}\sigma_{Xi}}\exp\left[-\frac{(X_i-Y_i)^2}{2\cdot\sigma_{Xi}^2}\right]=\frac{1}{\sqrt{2\pi}\sigma_{Xi}}\exp\left(-\frac{[x_{oi}-x_s(t_i)]^2+[y_{oi}-y_s(t_i)]^2}{2\cdot\sigma_{Xi}^2}\right). \tag{14}$$

The computation of $\sigma_{X_i}$ based on a modified version of the method proposed in [28] is described in Appendix A.

Let us now compute the expression of $P(H_i/v_s)$. The components of $H_i$ are classically computed by convolving the original image with derivative kernels noted $K_x$ and $K_y$. Using the assumption of a Gaussian additive noise, it is straightforward to show that $P(H_i/G_i)$ is also Gaussian. The crosscorrelation and autocorrelation functions of the vertical and horizontal components of $H$ are given by the well-known relation

$$C_{xy}(x_i-x_j,y_i-y_j)=E[H_{xi}\cdot H_{yj}]-E[H_{xi}]E[H_{yj}]=\sigma_b^2\delta(x_i-x_j,y_i-y_j)\otimes K_x(x,y)\otimes K_y^*(-x,-y), \tag{15}$$

where $K_x^*$, $K_y^*$ are the complex conjugates of the matrices $K_x$, $K_y$ and $\otimes$ represents the convolution operator.

Using this relation it is easy to show that

$$E[H_{xi}\cdot H_{yi}/G_{xi}G_{yi}]-E[H_{xi}/G_{xi}G_{yi}]E[H_{yi}/G_{xi}G_{yi}]=0, \tag{16}$$

$$E[H_{xi}\cdot H_{xi}/G_{xi}]-E[H_{xi}/G_{xi}]E[H_{xi}/G_{xi}]=\sigma_b^2\cdot\sum_{i,j}K_x(i,j))^2=\sigma_b^2\cdot\sum_{i,j}K_y(i,j)^2=\sigma_H^2, \tag{17}$$

where $E[w/z]$ represents the conditional expected value of $w$ assuming $z$.

Let $J_i$ be the vector defined by $J_i=(G_{yi},-G_{xi})$, and let $R_i$ be the unit vector collinear to $v_s$ in $v_{si}$. $H_i$ and $J_i$ are clearly related by a normal distribution.

To express the law $P(H_i/v_s)$ we use the well-known property of the edge to be orthogonal to the local gradient. Then, we get

$$P(H_i/v_s)=P[H_i/R_i(v_s)]=\int\frac{P(H_i/J_i)P(J_i)}{P[R_i(v_s)]}, \tag{18}$$

where the integration is achieved for all the values of $J_i$ collinear to $R_i$, that is, for $J_i=a\cdot R_i$. Equation (18) can be expressed by

$$P(H_i/v_s)=\frac{(2\pi\sigma_H^2)^{-1}}{P(R_i)}\int_{-\infty}^{+\infty}\exp\left[-\frac{(H_{yi}-a\cdot R_{xi})^2}{2\cdot\sigma_H^2}\right]\times\exp\left[-\frac{(H_{xi}+a\cdot R_{yi})^2}{2\cdot\sigma_H^2}\right]P(J_i)\mathrm{d}a. \tag{19}$$

In Eq. (19), $P(J_i)$ is the *a priori* probability of $J_i$. The gradient has no privileged direction and the uniform regions are supposed to be more frequent than the regions with a high gradient magnitude. Thus, $P(J_i)$ is modeled with the following centered normal law:

$$P(J_i)=\frac{1}{\sqrt{\pi}\Sigma}\exp\left(-\frac{\|J_i\|^2}{\Sigma^2}\right)=\frac{1}{\sqrt{\pi}\Sigma}\exp\left(-\frac{a^2}{\Sigma^2}\right). \tag{20}$$

Using this assumption we can compute Eq. (19) by

$$P(H_i/v_s)=K\cdot\exp\left(-\frac{H_{yi}^2+H_{xi}^2}{2\sigma_H^2+\Sigma^2}\right)\times\exp\left[-\frac{(H_{xi}R_{xi}+H_{yi}R_{yi})^2\Sigma^2}{2\sigma_H^2(2\sigma_H^2+\Sigma^2)}\right] \tag{21}$$

with $K$ a normalization constant involving $P(R_i)$, $\Sigma$, and $\sigma_H$.

In the above, vector $R_i$ is given by

$$R_i=\frac{T_i}{\|T_i\|},\quad\text{where } T_i=\left[\frac{\partial x_s(t_i)}{\partial t},\frac{\partial y_s(t_i)}{\partial t}\right]^T,$$

and $T=\|T_i\|$ is assumed to be constant along the edge.

The computation of vector $T_i$ depends on the chosen model. In the case of the B-snake model, the components of $T_i$ are also spline functions of which the B-spline basis functions have the following expression:

$$\frac{\partial B_{n,i}(t)}{\partial t} = \frac{n \cdot B_{i,n-1}(t)}{t_{i+n} - t_i} - \frac{n \cdot B_{i+1,n-1}(t)}{t_{i+n+1} - t_{i+1}}. \tag{22}$$

In the case of the sampled model, $T_i$ is simply estimated with a finite difference approximation.

Finally, the global likelihood is estimated by assuming that the $O_i$ are conditionally independent: $P(O/v_s) = \Pi_i P(O_i/v_s)$.

### B. Numerical Implementation

*1. Parametric Model*

In the parametric approach, the contour is described by the parameter vector $\theta$. As stated above, such a model allows dropping the internal energy term which leads to a ML formulation of the estimation problem, that is,

$$\hat{\theta} = \arg \cdot \max_{\theta} \prod_i P(X_i/H_i, v_s) P(H_i/v_s). \tag{23}$$

This relation classically yields the minimization of an energy function that, in our case, is the quadratic function

$$E(\theta) = (\mathbf{d} - \boldsymbol{\mathcal{B}}\theta)^T W (\mathbf{d} - \boldsymbol{\mathcal{B}}\theta). \tag{24}$$

In the above, $\mathbf{d}$ is a $(3L \times 1)$ vector defined by

$$\mathbf{d} = (x_{o1}, \ldots, x_{oL}, y_{o1}, \ldots, y_{oL}, 0, \ldots, 0)^T,$$

$W$ is a $(3L \times 3L)$ diagonal matrix with $W_{i,i} = W_{i+L,i+L} = 1/\sigma_{X_i}^2$,

$$W_{i+2L,i+2L} = \frac{\Sigma^2}{T^2 \sigma_H^2 (2\sigma_H^2 + \Sigma^2)},$$

and $\boldsymbol{\mathcal{B}}$ is a $(3L \times 2N)$ matrix: $\boldsymbol{\mathcal{B}} = \begin{bmatrix} \mathbf{B} & 0 & \mathbf{B}_x \\ 0 & \mathbf{B} & \mathbf{B}_y \end{bmatrix}^T$.

The components of matrix $\boldsymbol{\mathcal{B}}$ are obtained from the $B_i(t)$ functions $\boldsymbol{\mathcal{B}}_{ij} = B_j(t_i)$ with $t_i = i/L$. The components of $\boldsymbol{\mathcal{B}}_x$ and $\boldsymbol{\mathcal{B}}_y$ are computed from Eqs. (21) and (22): $\boldsymbol{\mathcal{B}}_{x_{ij}} = L \cdot H_{xi}[B_{i,2}(i/L) - B_{i+1,2}(i/L)]$ and $\boldsymbol{\mathcal{B}}_{y_{ij}} = L \cdot H_{yi}[B_{j,2}(i/L) - B_{j+1,2}(i/L)]$.

As classically, the solution $\hat{\theta} = \arg\min_\theta E(\theta)$ is given by the weighted least-squares relation

$$\hat{\theta} = (\boldsymbol{\mathcal{B}}^T W \boldsymbol{\mathcal{B}})^{-1} \mathcal{B}^T W \mathbf{d}.$$

*2. Sampled Model*

In the sampled model, the unknown to be estimated is constituted by the set of snaxels. The external energy is obtained similarly from relation (4) by using a finite-difference approximation of $T_i$:

$$E_{ext} = C + \sum_i \frac{[x_{oi} - x_{sj(i)}]^2 + [y_{oi} - y_{sj(i)}]^2}{2 \cdot \sigma_{Xi}^2} + \frac{\{H_{xi}[x_{sj(i)+1} - x_{sj(i)}] + H_{yi}[y_{sj(i)+1} - y_{sj(i)}]\}^2 \Sigma^2}{2\sigma_H^2 T^2 (2\sigma_H^2 + \Sigma^2)}, \tag{25}$$

with $C$ a constant.

The previous expression of the external energy leads to the following Euler equations for the $x_s$ coordinates:

$$\boldsymbol{\mathcal{A}}_{j(i)} \mathbf{v}_s + \left[ \frac{1}{\sigma_{X_i}^2} + \frac{H_{xi}^2 \Sigma^2}{\sigma_H^2 T^2 (2\sigma_H^2 + \Sigma^2)} \right] x_{sj(i)} - \frac{H_{xi}^2 \Sigma^2}{\sigma_H^2 T^2 (2\sigma_H^2 + \Sigma^2)} x_{sj(i)+1} + \frac{H_{xi} H_{yi} \Sigma^2}{\sigma_H^2 T^2 (2\sigma_H^2 + \Sigma^2)} [y_{sj(i)} - y_{sj(i)+1}] = \frac{x_{oi}}{\sigma_{X_i}^2}, \tag{26}$$

$$\boldsymbol{\mathcal{A}}_{j(i)+1} \mathbf{v}_s + \frac{H_{xi}^2 \Sigma^2}{\sigma_H^2 T^2 (2\sigma_H^2 + \Sigma^2)} [x_{sj(i)+1} - x_{sj(i)}] + \frac{H_{xi} H_{yi} \Sigma^2}{\sigma_H^2 T^2 (2\sigma_H^2 + \Sigma^2)} [y_{sj(i)+1} - y_{sj(i)}] = 0, \tag{27}$$

$$\boldsymbol{\mathcal{A}}_j \mathbf{v}_s = 0, \qquad \text{if } j \neq m \cdot i \text{ with } m \in \mathbb{N}. \tag{28}$$

The corresponding equations relative to the $y_s$ coordinates are derived similarly.

In Eqs. (26)–(28), $\boldsymbol{\mathcal{A}}_j$ represents the $j$th row of the matrix $\boldsymbol{\mathcal{A}}$ defined by $\boldsymbol{\mathcal{A}} = \begin{bmatrix} \mathcal{A} & 0 \\ 0 & \mathcal{A} \end{bmatrix}$ and $\mathbf{v}_s = (\ldots x_{sj}, x_{sj+1}, \ldots, y_{sj}, y_{sj+1}, \ldots)^T$ is the vector of the estimated subpixel coordinates. In our model we have kept the initial definition of the internal energy. The expression of the matrix $\boldsymbol{\mathcal{A}}$ is hence the pentadiagonal matrix proposed in [20].

The above equations can be written as the linear system

$$(\boldsymbol{\mathcal{A}} + \boldsymbol{\mathcal{D}})\mathbf{v}_s = \mathbf{c}, \tag{29}$$

where $\mathbf{c}$ is a $(2M \times 1)$ vector with

$$\mathbf{c}_{j(i)} = x_{oi}/\sigma_{X_i}^2, \quad \mathbf{c}_{j(i)+M} = y_{oi}/\sigma_{X_i}^2,$$

$$\mathbf{c}_j = 0, \qquad \text{if } j \neq m \cdot i \text{ with } m \in \mathbb{N},$$

and $\boldsymbol{\mathcal{D}} = \begin{bmatrix} \mathcal{D}_x & \mathcal{D}_{xy} \\ \mathcal{D}_{xy} & \mathcal{D}_y \end{bmatrix}$ is a $(2M \times 2M)$ matrix where $\boldsymbol{\mathcal{D}}_x$, $\boldsymbol{\mathcal{D}}_y$, and $\boldsymbol{\mathcal{D}}_{xy}$ are tridiagonal.

*3. Computational Issues*

From a computational cost point of view, both algorithms are based on a matrix inversion and hence have the same complexity [i.e., $O(n^3)$ with a Gaussian elimination algorithm]. However, the dimension of the data is clearly much higher in the case of the sampled model, which leads to a more expensive algorithm. With regard to the nonweighted approaches (such as the method proposed in [23]), the proposed algorithms require the estimation of the statistical parameter $\sigma_X$. However, this computation

is achieved only on edge pixels (for instance, the time needed for the computation of a 1000 pixel length edge is ≈0.3 s on a 3 Mhz P4 under Linux). Moreover, since it can be achieved independently for each pixel, it is linear in time.

## 4. EXPERIMENTAL RESULTS

To validate the proposed algorithms, we have carried out several tests on synthetic and real images. For all the tests we have computed the gradient by using the derivative of a Gaussian kernel with a standard deviation equal to one. The number $M$ of interpolated snaxels of the sampled model has been set to $3\times L$. The number of knots $N$ of the parametric model has been set to $L/4$. For each experiment, we have compared the result of the proposed algorithms with two references on separated figures: the "true" edge obtained from a high resolution image and the result of an algorithm of the literature. These two algorithms are, respectively, the method proposed by Venkatesh *et al.* in [23] in the case of the sampled model and a classical B-spline interpolation in the case of the parametric model.

### A. Synthetic Images

In this subsection we investigat the behavior of the proposed algorithms for noise-free and noisy images. The first test image is the synthetic $128\times128$ image, normalized between 0 and 1, of Fig. 1. We have numerically simulated the optical integration of the intensity over the pixels in low-pass filtering and subsampling a $4096\times4096$ version of this image. We present the results for several different areas corresponding to flat or curved contours designated by small squares in Fig. 1.

In order to validate the definition of the subpixel coordinates from a quadratic interpolation approach, we first present in Figs. 2 and 3 the result obtained with the noise-free image. In the flat region A, the accuracy is ≈1/30th pixel, which is close to the resolution of the native image (1/32th pixel). In the curved region B, the accuracy is ≈1/10th pixel. This gap is in part due to the smoothing effect of the algorithm. In the two cases, we also present the classical snake result, which presents a bias.

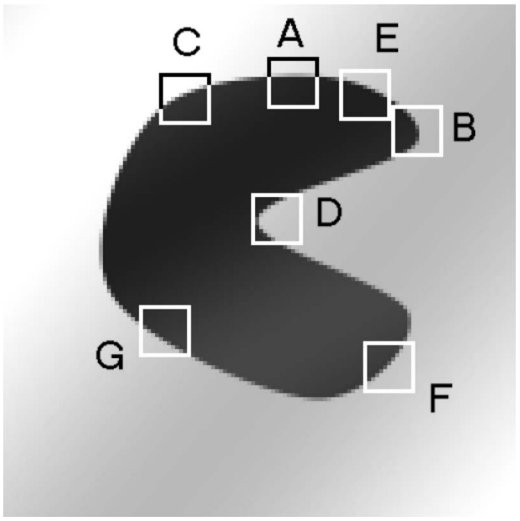


Fig. 1. First synthetic test image.

#### 1. Effect of White Gaussian Noise

We have first tested this algorithm for different white additive Gaussian noises, which correspond to images with SNR, respectively, equal to 400, 100, 25, and 10. Figures 4 and 5 give the results obtained with the two algorithms for the flat area C. The white crosses represent the measured subpixel coordinates. For high levels of noise [Figs. 4(c), 4(d), 5(c), and 5(d)], the filtering effects of the proposed approach are evidenced by comparison with the other two algorithms. In the case of the classical B-spline fitting, the gap between the estimated and the "true" edge is of the order of several pixels, whereas in the case of the proposed models it remains less than one pixel. Although it is less sensitive to noise, the Venkatesh algorithm also shows some important inaccuracy for low SNR. We draw the same conclusion when we consider Figs. 6 and 7, which correspond to the results obtained with the curved area D.

#### 2. Effect of White Salt and Pepper Noise

The proposed model has been developed in the case of a Gaussian noise. However, it remains efficient for other kinds of noise. In Figs. 8 and 9 we present the results obtained with a salt and pepper noise of which the PDF is defined by

$$P(x) = p_0\delta(x-\gamma) + (1-2\cdot p_0)\delta(x) + p_0\delta(x+\gamma),$$

where $\delta(.)$ is the Dirac function. The standard deviation of such a noise is given by $\sigma_n = \gamma\sqrt{2p_0}$. The filtering effect of the algorithms is similar to that of the Gaussian noise case.

#### 3. Quantitative Assessment

The second test is achieved on a simple $128\times128$ circle image. The aim of this test is to quantify the influence of the noise on the estimation of the radius of the circle. As previously, this image is obtained from a $4096\times4096$ high-resolution one. Thus, in the low-resolution version, the radius of the circle is noninterger (equal to 31.27).

Table 1 gives the result obtained with three different SNR (100, 25, and 4) for the four algorithms (classical spline interpolation, Venkatesh *et al.*, parametric model, and sampled model). We can observe that for all the noise levels, the inaccuracy in the computation of the radius (estimated by its standard deviation) remains lower for the proposed methods than for the two other algorithms. When the noise level is increased the standard deviation of the estimation of our algorithms is increased moderately when compared with the two others. For instance, the ratio of the standard deviation obtained with SNR $=4$ and SNR$=100$ is of the order of 27 for the classical spline, 5.6 for Venkatesh, and 2.7 and 2.5 for the two proposed algorithms.

### B. Real Images

The next tests were achieved on the two normalized images of Fig. 10. The sizes of these images are, respectively,

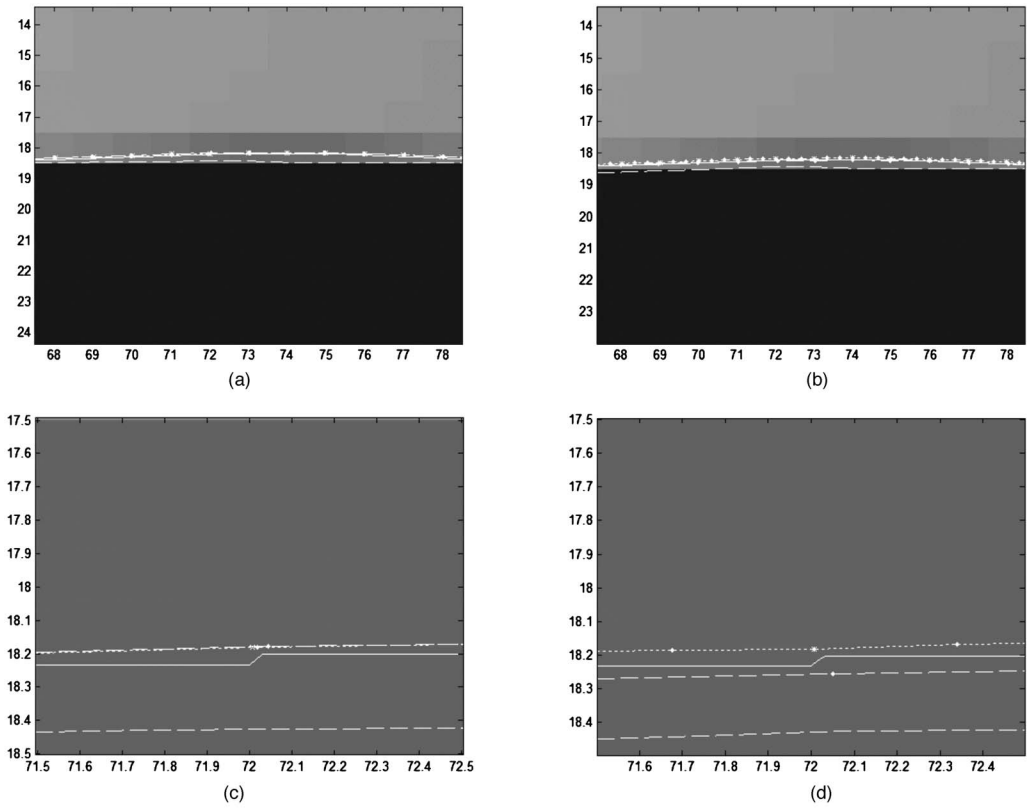


Fig. 2. Results in the noise-free case for the flat region A. (a) (c) Parametric model (dotted curves) compared with the classical spline interpolation (dashed curves with dots) and the "true" edge (solid curves). (b)(d) Sampled model (dotted curves) compared with the Venkatesh *et al.* method (dashed curves with dots) and the "true" edge (solid curves). In this figure and Fig. 2, we have also represented the classical snake result (dashed curves).

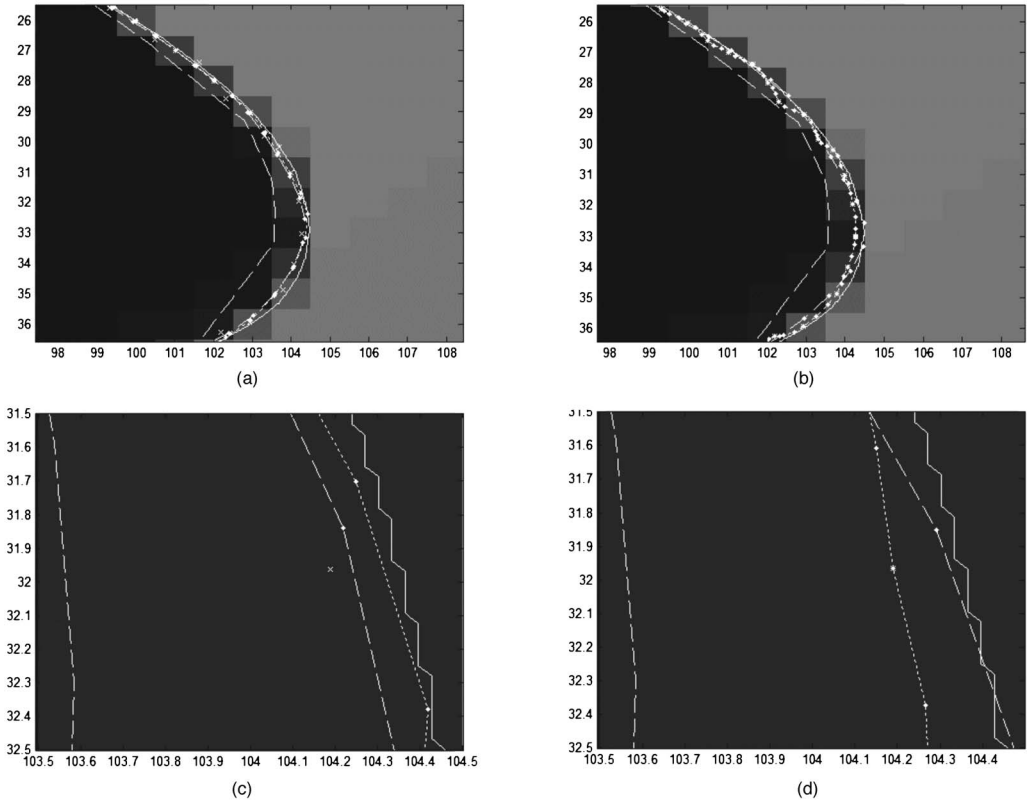


Fig. 3. Same results as Fig. 2 for the curved region B. See caption for Fig. 2.

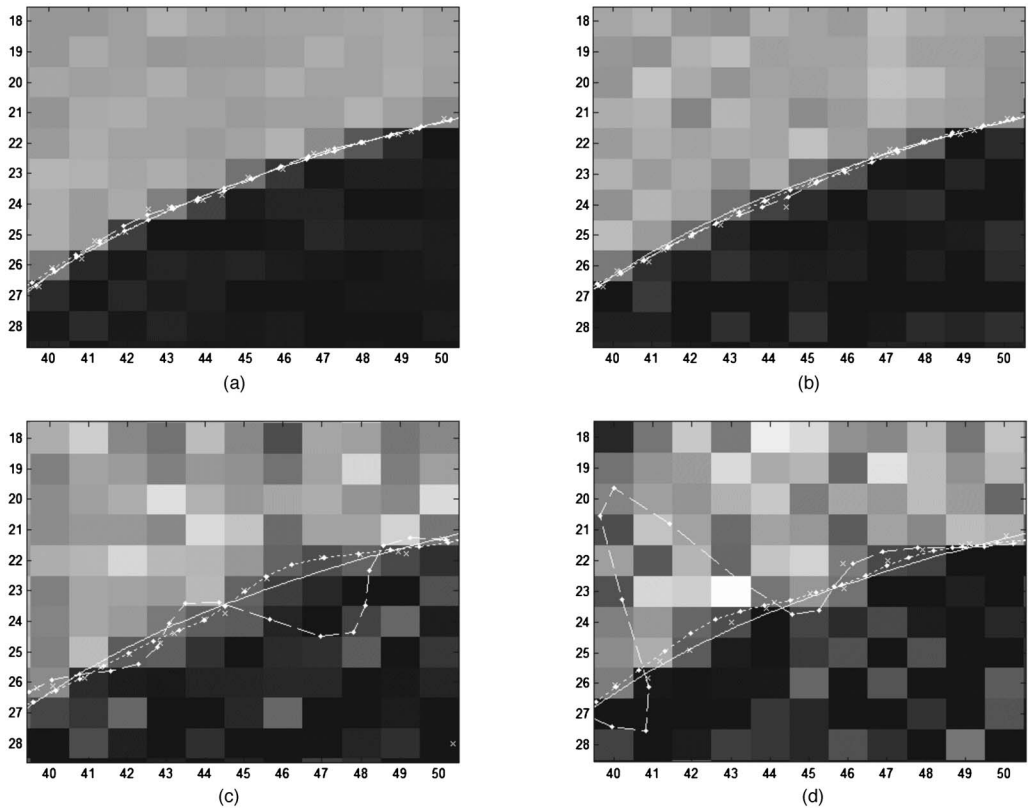


Fig. 4. Comparison (region C) of the parametric model (dotted curves) with the classical spline interpolation (dashed curves) and the true edge (solid curves) in the case of an additive Gaussian noise for SNR respectively equal to (a) 400, (b) 100, (c) 24, (d) 10.

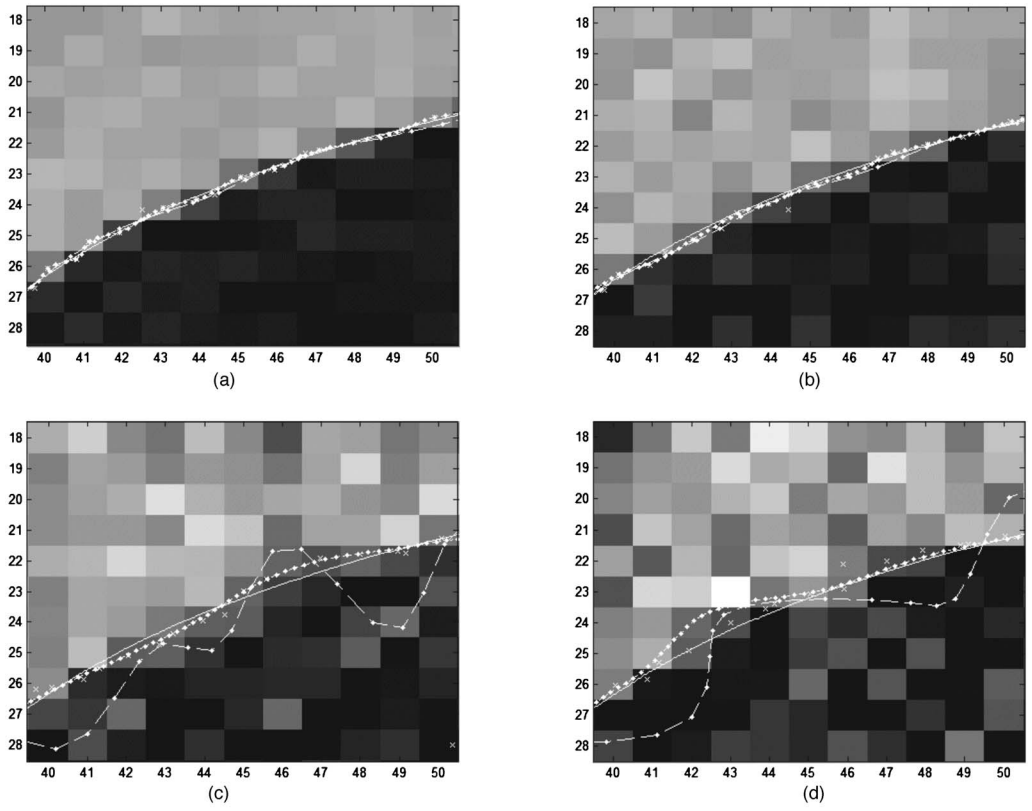


Fig. 5. Comparison of the sampled model (dotted curves) with the Venkatesh method (dashed curves) and the true edge (solid curves) with the same SNRs as for Fig. 4.

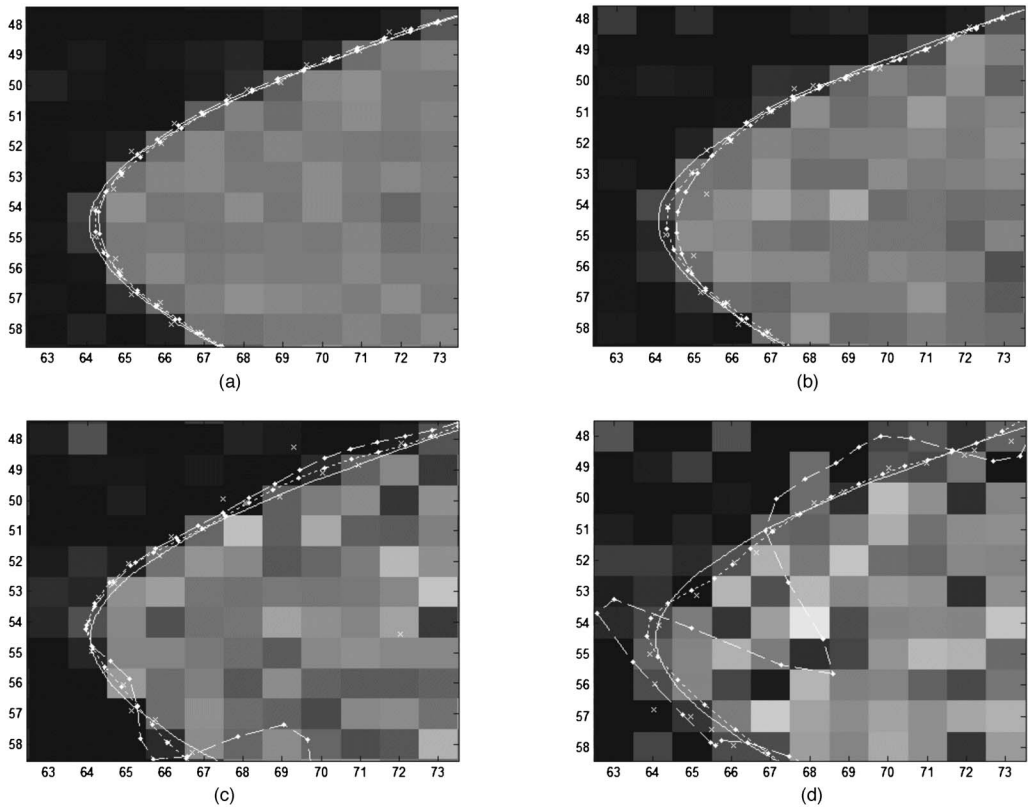


Fig. 6. Same as Fig. 4 for the curved region D.

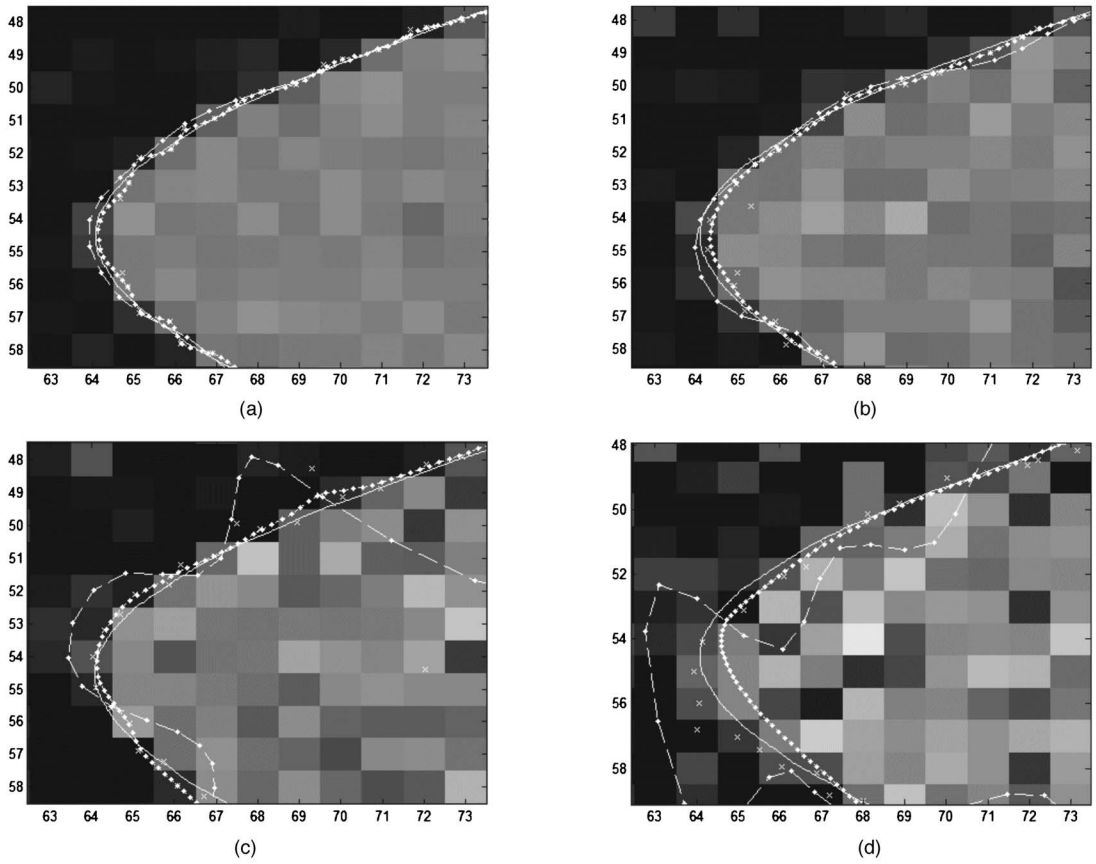


Fig. 7. Same as Fig. 5 for the curved region D.

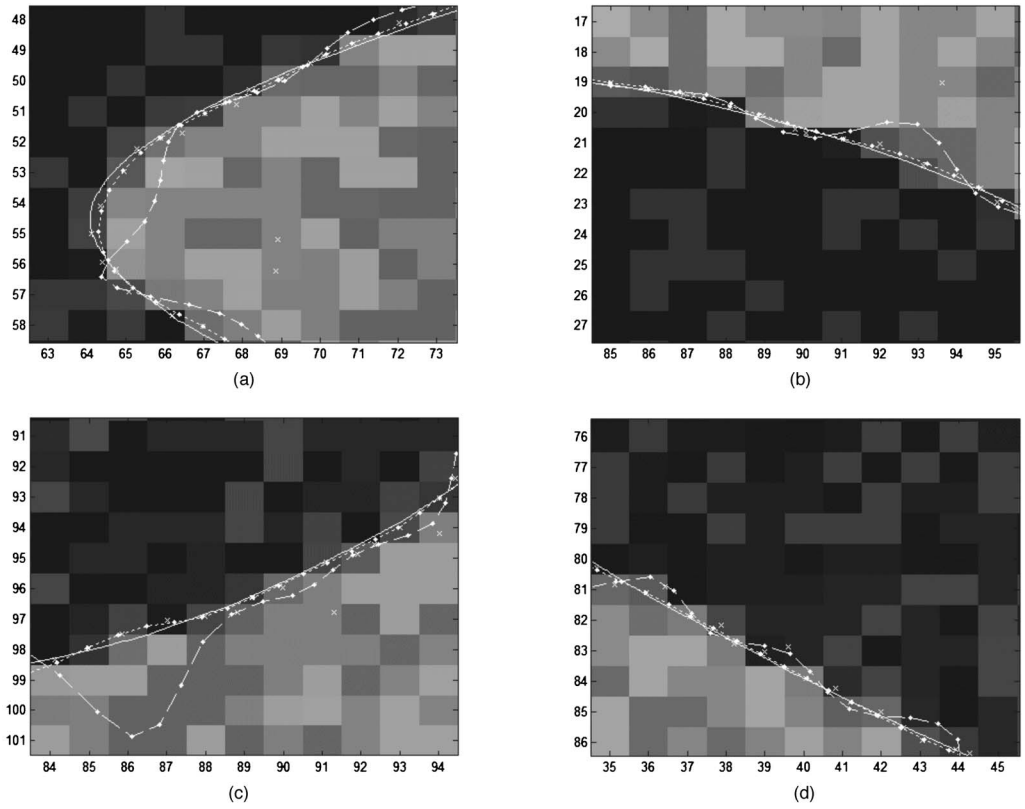


Fig. 8. Comparison of the parametric model with the classical spline interpolation and the true edge in the case of an impulse noise. (a) Region D, (b) region E, (c) region F, (d) region G.

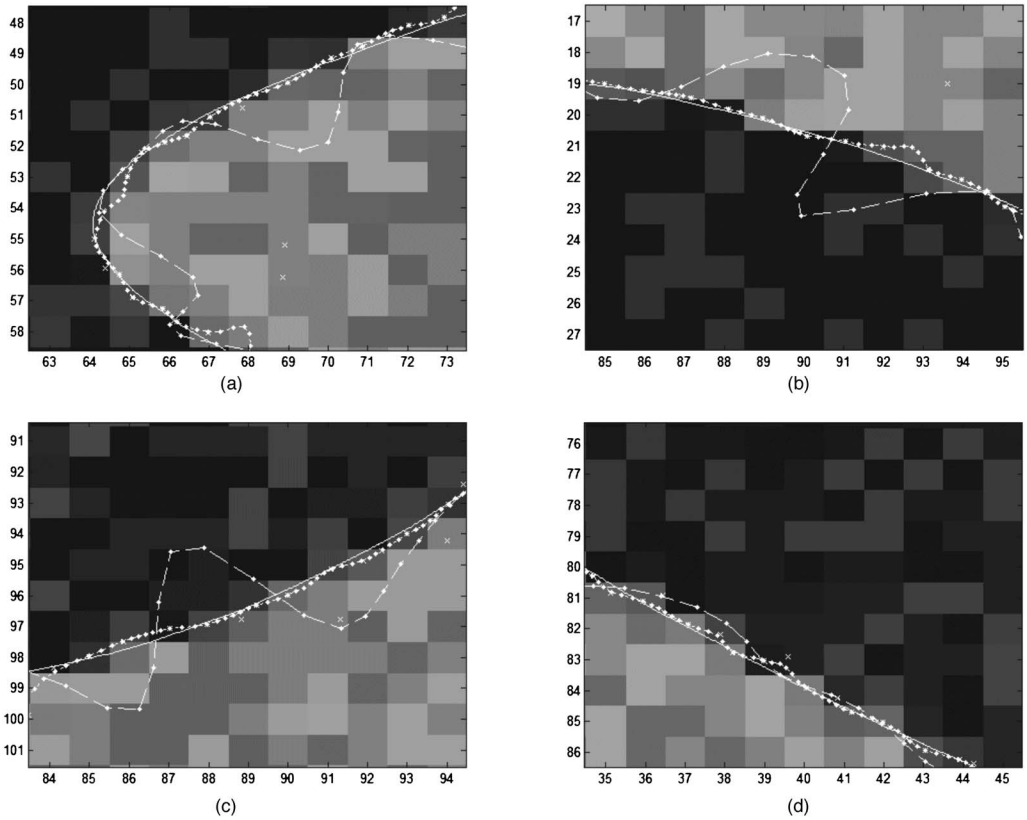


Fig. 9. Comparison of the sampled model with the Venkatesh method and the true edge with the same impulse noise and on the same regions as for Fig. 8.

**Table 1. Comparative Expectation and Standard Deviation of the Estimation of the Radius of a Circle ("True" Value 31.27) with SNR of 100, 25, and 4**

| | Algorithm | | | |
|---|---|---|---|---|
| SNR | Classical Spline | Venkatesh *et al.* | Parametric | Sampled |
| Radius | | | | |
| SNR=100 | 31.26 | 31.26 | 31.27 | 31.26 |
| SNR=25 | 31.24 | 31.26 | 31.27 | 31.28 |
| SNR=4 | 30.9 | 31.22 | 31.25 | 31.29 |
| Std. Dev. | | | | |
| SNR=100 | 0.157 | 0.154 | 0.130 | 0.135 |
| SNR=25 | 0.730 | 0.309 | 0.180 | 0.191 |
| SNR=4 | 4.22 | 0.860 | 0.349 | 0.334 |

$384\times256$ and $324\times243$ and are also obtained from two high-resolution versions from which are computed the reference true edges.

Figures 11 and 12 show the results obtained on the two regions A and B of Fig. 10(a). For these two examples, the initial contour is computed with the classical Canny algorithm, and we have used a nonperiodic version of the models. On this weakly noisy image we obtained similar results as for the synthetic cases. The Venkatesh algorithm and the classical spline interpolation present a gap with the true edge of about half a pixel, whereas the proposed method remains accurate.

Figure 13 shows two examples of results obtained with the sampled model applied on the regions A and B of the marble cup image [Fig. 10(b)]. The dashed curve represents the classical snake result that constitutes the starting point of the proposed algorithm. The small circles are the centers of the pixels used for the computation of the observation (i.e., the integer coordinates $v_{ei}$). Some of these pixels do not contain the true edge represented by a solid line—pixels (207,113) of Fig. 13(a), (184,131) or (186,130) of Fig. 13(b), for instance. We can observe that the subpixel coordinates $(x_{oi},y_{oi})$—represented by oblique crosses—computed at these locations are inaccurate. However, the estimated standard deviation $\sigma_{X_i}$ of these measures are much higher than those of the neighboring pixels [for instance, we have $\sigma_{X(207,113)}=2.37$, $\sigma_{X(184,131)}=3.7$, and $\sigma_{X(186,130)}=20.6$, whereas the values of $\sigma_X$ for accurate measures in the neighborhood are close to 0.1). These measures are then filtered.

Figure 14 is another example of result obtained with the parametric model applied on regions C and D of the marble cup image. In this example, the noisy subpixel measures generate instability and oscillations on the classical spline result.

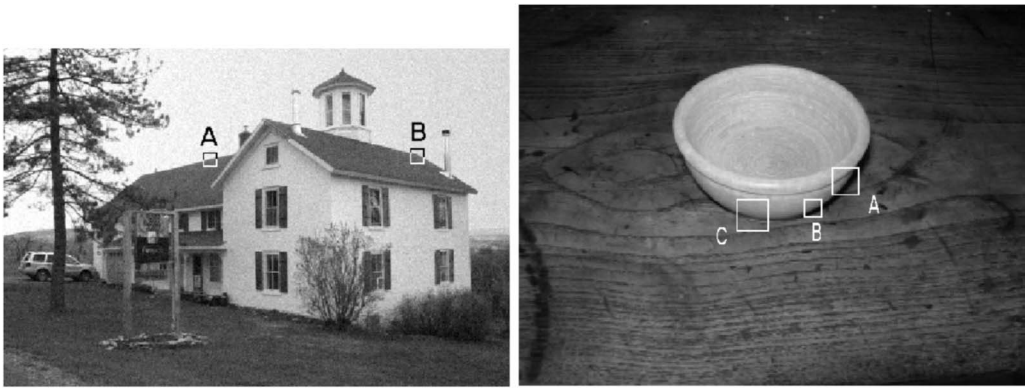


Fig. 10. Real images: (a) house, (b) marble cup.

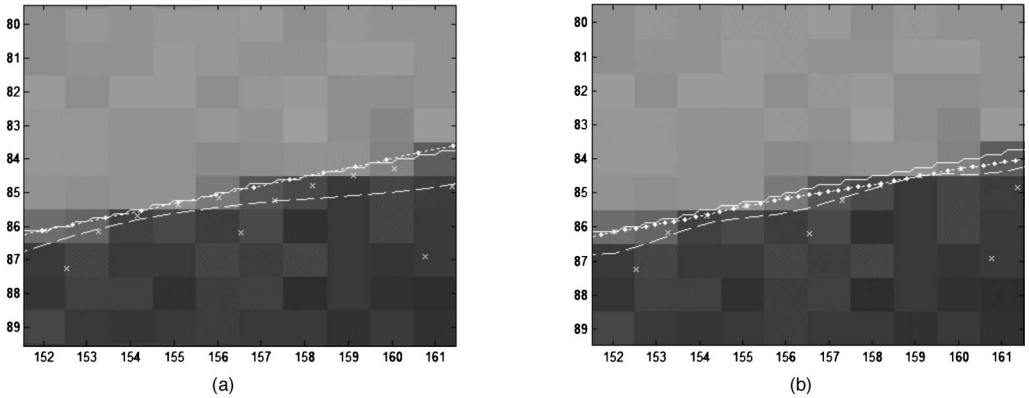


Fig. 11. Comparison of the parametric model with the classical spline interpolation and the true edge for the house image [Fig. 10(a)]. (a) Region A, (b) region B.

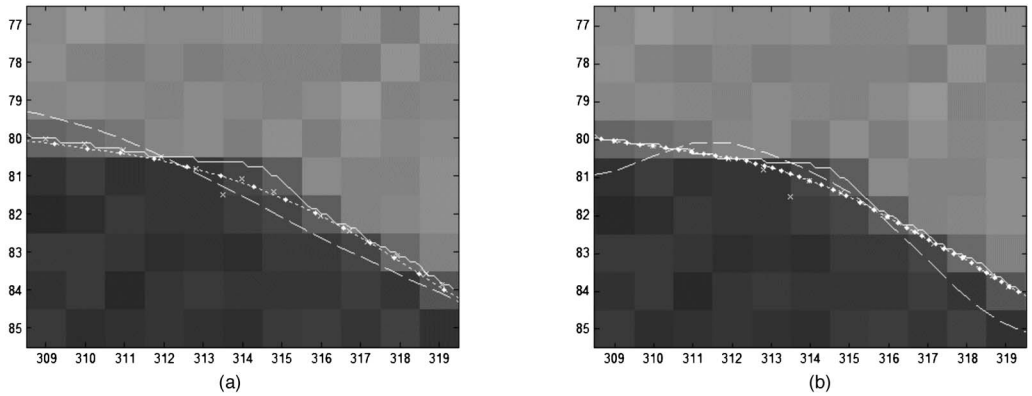


Fig. 12. Comparison of the sampled model with the Venkatesh method and the true edge for the house with the same region as for Fig. 11.

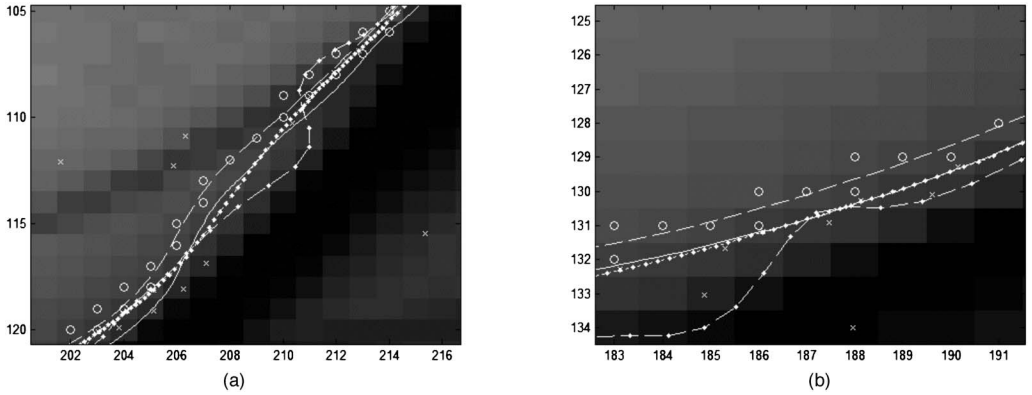


Fig. 13. Comparison of the parametric model with the classical spline interpolation and the true edge for the marble cup image [Fig. 10(b)]. (a) Region A, (b) region B. The dashed curves represent the classical snake result.

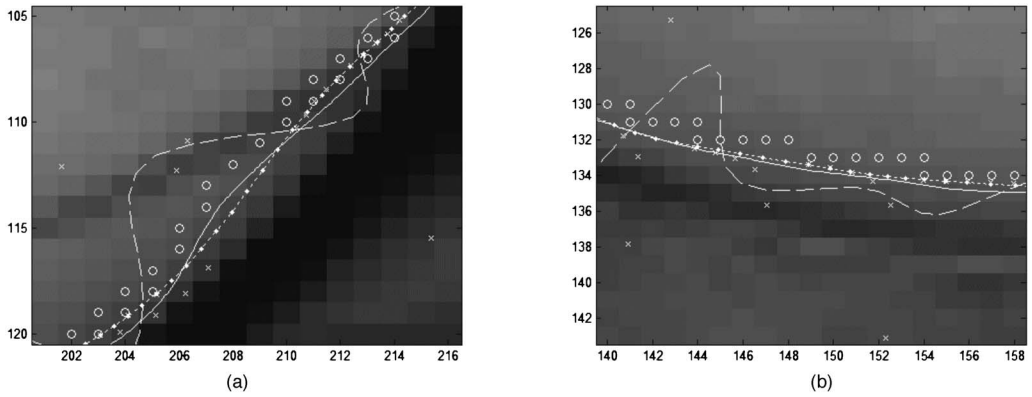


Fig. 14. Comparison of the sampled model with the classical snake model and the true edge for the marble cup image [Fig. 10(b)] (a) Region A, (b) region C.

## 5. CONCLUSION

In this paper we have presented an algorithm based on a deformable model for the estimation of an edge in an image with subpixel accuracy. In the proposed approach, the likelihood of the data is computed from the observation model that includes both orientation and position information. Two different implementations are proposed: a continuous model based on a B-spline description and a discrete model.

Comparative experiments of this algorithm with other algorithms of the literature have been carried out on synthetic images for two different kinds of noise: a Gaussian noise and a salt and pepper noise. The results show that the method remains accurate even when the noise is not Gaussian and for a high noise level. We also illustrate the relevance of the approach for detecting accurate subpixel edges in natural images. In particular, we show that the algorithm effectively filters inaccurate measures due to local perturbations or texture.

## APPENDIX A: COMPUTATION OF $\sigma_X$

The method we use to interpolate the subpixel coordinate is basically an improvement of a well-known edge detec-

tion method sometimes called NMS (nonmaxima suppression). The NMS method consists of the suppression of the local nonmaxima of the magnitude of the gradient of image intensity in the direction of this gradient [29]. The subpixel approximation proposed in [27] adds a new step to the NMS process: when the point $(x,y)$ is a local maximum then the position of the edge point in the direction of the gradient is given by the maximum of a quadratic interpolation of the values of the squared gradient norms at $(x,y)$ and the neighboring points.

Let $s$ be an edge pixel and let $a$, $b$, $c$, $d$, $e$, $f$, $u$, and $v$ be its vicinity. Let $N_a$, $N_b$…, and $N_s$ be the corresponding squared gradient norms (Fig. 15). Let $(H_{xs},H_{ys})$ be the components of the gradient vector in $s$ and let

$$\alpha = \frac{H_{ys}}{H_{xs}}. \tag{A1}$$

We consider here the case $\alpha<1$. In the direction defined by $\alpha$ the squared gradient norms $N_k$ and $N_l$ in $k$ and $l$ are interpolated using two quadratic functions involving, respectively, $N_a$, $N_b$, $N_c$ and $N_d$, $N_e$, $N_f$. The subpixel coordinate $X_s$ of the edge is obtained by computing the maximum of a third quadratic curve defined by the points $k$, $l$ and $s$. After some computations we obtain

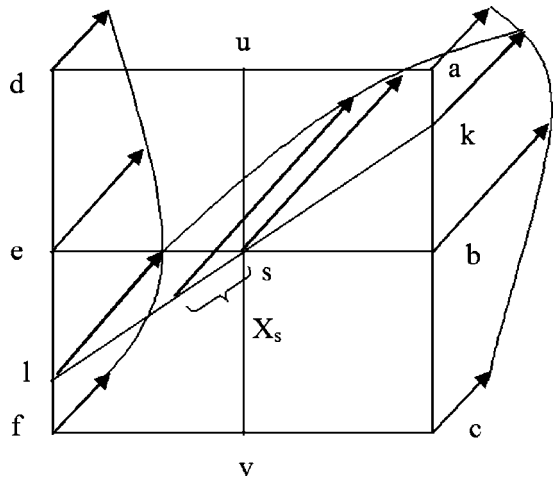


Fig. 15. Estimation of the subpixel location $X_s$ by using a quadratic interpolation of the square gradient norm.

$$X_s = \frac{(N_e - N_b) + \frac{1}{2}(N_a - N_c + N_d - N_f)\alpha + \cdots \quad \cdots \frac{1}{2}(2N_b - N_a - N_c + N_d - 2N_e + N_f)\alpha^2}{2(N_b + N_e - 2N_s) + (N_c - N_a + N_d - N_f)\alpha + \cdots \quad \cdots (N_a - 2N_b + N_c + N_d - 2N_e + N_f)\alpha^2}, \tag{A2}$$

which can be written

$$X_s = \frac{(M_1 \otimes N)_{(0,0)}}{(M_2 \otimes N)_{(0,0)}} = \frac{A}{B}, \tag{A3}$$

where $\otimes$ represents the convolution operator. Matrices $M_1$ and $M_2$ have the following expressions:

$$M_1 = \begin{bmatrix} -\alpha - \alpha^2 & 0 & -\alpha + \alpha^2 \\ -2 + 2\alpha^2 & 0 & 2 - 2\alpha^2 \\ \alpha - \alpha^2 & 0 & \alpha + \alpha^2 \end{bmatrix},$$

$$M_2 = \begin{pmatrix} 2\alpha + \alpha^2 & 0 & -2\alpha + \alpha^2 \\ 4 - 4\alpha^2 & -4 & 4 - 4\alpha^2 \\ -2\alpha + 2\alpha^2 & 0 & 2\alpha + 2\alpha^2 \end{pmatrix}.$$

To compute the second-order moments of $X_s=A/B$ we first define this ratio of two random variables as

$$\frac{A}{B} = \frac{A_c + E[A]}{B_c + E[B]}. \tag{A4}$$

In the above, $A_c$ and $B_c$ are the centered parts of $A$ and $B$. We then apply a second-order power series approximation on the denominator variable around $E[B]$. After some computations we get

$$E\left[\frac{A}{B}\right] \approx \frac{E[A]}{E[B]} - \frac{C_{AB}}{E[B]^2} + \frac{E[A]\sigma_B^2}{E[B]^3}, \tag{A5}$$

$$E\left[\left(\frac{A}{B}\right)^2\right] \approx \frac{E[A]^2}{E[B]^2} + \frac{\sigma_A^2}{E[B]^2} + 3\frac{E[A]^2\sigma_B^2}{E[B]^4} - 4\frac{E[A]C_{AB}}{E[B]^3} + 3\frac{\sigma_A^2\sigma_B^2 + 2C_{AB}^2}{E[B]^4}, \tag{A6}$$

where $\sigma_A^2=E[A_c^2]$, $\sigma_B^2=E[B_c^2]$, and $C_{AB}=E[A_cB_c]$.

To compute Eqs. (A5) and (A6) we need the expressions of $E[A]$, $E[B]$, $E[A^2]$, $E[B^2]$ and $E[AB]$. From Eq. (A3) and using a classical result we can write

$$E[A] = (M_1 \otimes E[N])_{(0,0)},$$

$$E[B] = (M_2 \otimes E[N])_{(0,0)}. \tag{A7}$$

Note that in the above, the expectations are conditional assuming $H_s$ and hence $\alpha$ and deterministic. The expected value $E[N_s]$ on a given pixel $s$ is simply obtained from the expression of $N_s$ by using Eq. (15):

$$E[N_s] = E[H_{xs}^2 + H_{ys}^2] = E[H_{xs}^2] + E[H_{ys}^2] = C_{xx}(0,0) + C_{yy}(0,0) + E[H_{xs}]^2 + E[H_{ys}]^2. \tag{A8}$$

To compute the joint second moments of $A$ and $B$ we first express the product $AB$ as follows (the demonstration is given for $E[AB]$ but the generalization to $E[A^2]$ and $E[B^2]$ is straightforward):

$$AB = \sum_s \sum_t M_1(-s)M_2(-t)N(s)N(t), \quad \text{(A9)}$$

which yields

$$E[AB] = \sum_s \sum_t M_1(-s)M_2(-t)E[N(s)N(t)]. \quad \text{(A10)}$$

Using the same method as for Eq. (A8) we can compute the joint second moment $E[N_sN_t]$. After some computations, one can express $E[N_sN_t]$ with the compact relation

$$E[N_sN_t] = 4V_s^t \cdot M_{st} \cdot V_t + 2\ \text{Trace}[M_{st} \cdot M_{st}] + E[N(s)]E[N(t)], \quad \text{(A11)}$$

where $M_{st} = \begin{bmatrix} C_{xx}(s,t) & C_{xy}(s,t) \\ C_{xy}(s,t) & C_{yy}(s,t) \end{bmatrix}$ and $V_i = (H_{xi}, H_{yi})^T$ with $i \in \{s,t\}$.

## REFERENCES


1. P. Eisert, E. Steinbach, and B. Girod, "Automatic reconstruction of stationary 3-D objects from multiple uncalibrated camera views," IEEE Trans. Circuits Syst. Video Technol. **10**, 261–277 (2000).
2. A. J. Tabatabai and R. Mitchell, "Edge localisation to sub-pixel values in digital imagery," IEEE Trans. Pattern Anal. Mach. Intell. **6**, 188–201 (1984).
3. E. P. Lyvers and O. R. Mitchell, "Precision edge contrast and orientation estimation," IEEE Trans. Pattern Anal. Mach. Intell. **10**, 927–937 (1988).
4. E. P. Lyvers, O. R. Mitchell, M. L. Akey, and A. P. Reeves, "Sub-pixel measurements using a moment based edge operator," IEEE Trans. Pattern Anal. Mach. Intell. **11**, 1293–1309 (1989).
5. Y. Shan and G. W. Boon, "Sub-pixel localisation of edges with non-uniform blurring: a finite closed-form approach," Image Vis. Comput. **18**, 1015–1023 (2000).
6. S.-C. Cheng and T.-L. Wu, "Subpixel edge detection of color images by principal axis analysis and moment-preserving principle," Pattern Recogn. **38**, 527–537 (2005).
7. S. Ghosal and R. Mehrotra, "Orthogonal moment operators for sub-pixel edge detection," Pattern Recogn. **26**, 295–306 (1993).
8. Q. Ying-Donga, C. Cheng-Songa, C. San-Benb, and L. Jin-Quan, "A fast subpixel edge detection method using Sobel–Zernike moments operator," Image Vis. Comput. **23**, 11–17 (2005).
9. Y. Nomura, M. Sagara, H. Naruse, and A. Ide, "Edge location to sub-pixel precision and analysis," Syst. Comput. Japan **22**, 70–80 (1991).
10. S. Hussmann and T. H. Ho, "A high-speed subpixel edge detector implementation inside a FPGA," Real-Time Imag. **9**, 361–368 (2003).
11. M. Baba and K. Ohtani, "A novel subpixel edge detection system for dimension measurement and object localization using an analogue-based approach," J. Opt. A, Pure Appl. Opt. **3**, 276–283 (2001).
12. F. Truchetet, F. Nicolier, and O. Laligant, "Subpixel edge detection for dimensional control by artificial vision," J. Electron. Imaging **10**, 234–239 (2001).
13. K. Jensen and D. Anastassiou, "Subpixel edge localization and the interpolation of still images," IEEE Trans. Image Process. **4**, 285–295 (1995).
14. T. Hermosilla, E. Bermejo, A. Balaguer, and L. A. Ruiz, "Non-linear fourth-order image interpolation for subpixel edge detection and localization," Image Vis. Comput. **26**, 1240–1248 (2008).
15. M. Kisworo, S. Venkatesh, and G. West, "Modeling edges at sub-pixel accuracy using the local energy approach," IEEE Trans. Pattern Anal. Mach. Intell. **16**, 405–410 (1994).
16. R. Deriche and G. Giruadon, "A computational approach for corner and vertex detection," Int. J. Comput. Vis. **10**, 101–124 (1993).
17. H. Wang and M. Brady, "Real-time corner detection algorithm for motion estimation," Image Vis. Comput. **13**, 695–703 (1995).
18. F.-L. Chen and S.-W. Lin, "Subpixel estimation of circle parameters using orthogonal circular detector," Comput. Vis. Image Underst. **78**, 206–221 (2000).
19. V. Ayala-Ramirez, C. H. Garcia-Capulin, A. Perez-Garcia, and R. E. Sanchez-Yanez, "Circle detection on images using genetic algorithms," Pattern Recogn. Lett. **27**, 652–657 (2006).
20. M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," Int. J. Comput. Vis. **1**, 321–331 (1988).
21. P. Brigger, J. Hoeg, and M. Unser, "B-spline snakes: A flexible tool for parametric contour detection," IEEE Trans. Image Process. **9**, 1484–1087 (2000).
22. Ravinda G. N. Meegama and Jagath C. Rajapakse, "NURBS," Image Vis. Comput. **21**, 551–562 (2003).
23. S. Venkatesh, M. Kisworo, and G. A. West, "Detection of curved edges at sub-pixel accuracy using deformable models," in *Proceedings of IEEE Conference on Vision, Imaging and Signal Processing* (IEEE, 1995), 304–312.
24. A. Bjōrck, *Numerical Methods for Least Squares Problems* (SIAM, 1996).
25. F. Bouchara, M. Bertrand, S. Ramdani, and Mahmoud Haydar, "Sub-pixel edge fitting using B-spline," in *Proceedings of MIRAGE 07*, Lecture Notes in Computer Science, Vol. 4418 (Springer, 2007), pp. 353–364.
26. C. deBoor, *A Practical Guide to Splines* (Springer-Verlag, 1978).
27. F. Devernay, "A Non-Maxima Suppression Method for Edge Detection with Sub-Pixel Accuracy," Institut National de Recherche en Informatique et en Automatique, Sophia Antipolis Research Report No. 2724 (1995).
28. F. Bouchara, "Efficient algorithm for computation of the second-order moment of the subpixel-edge position," Appl. Opt. **43**, 4550–4558 (2004).
29. R. Deriche, "Using Canny's criteria to derive a recursively implemented optimal edge detector," Int. J. Comput. Vis. **1**, 167–187 (1987).


146.112.31.89 (6/2019-12-17 23:24:53)