CONTENT ADAPTIVE HETEROGENEOUS SNAKES

András Hajdu, Ioannis Pitas

Department of Informatics, University of Thessaloniki, Box 451, 54124 Thessaloniki, Greece
Email: pitas@aiia.csd.auth.gr

ABSTRACT

Active contour (snake) approaches have been proved to be efficient tools to extract object boundary precisely. One drawback of these methods is that state-of-the-art snake algorithms are usually computationally expensive. To overcome this difficulty, one approach is to reduce spatial resolution to make snake iteration faster. The disadvantage of this method is that the reduction of resolution ignores image content which may lead to losing useful image data. In this paper we propose a snake iteration method on quadtree representations derived from the image. In this way we can reduce resolution in a special and adaptive way. Once the iteration is finished on the sparse quadtree grid, final adjustments can be made by performing pixelwise snake iteration steps.

***Index Terms*—** Image edge analysis, iterative methods, object detection, quadtrees

1. INTRODUCTION

Boundary extraction is an important topic in digital image processing, thus several approaches have been developed in the past to this end. One of them is the snake (active contour) model introduced in [1]. The basic idea here is to evolve a curve iteratively in order to approach the object boundary.

Considering its traditional formulation, the snake is a parametric contour that deforms over a series of iterations influenced by internal and external forces. Internal forces control the snake stretching and bending, while external forces push the snake toward image edges. To assure a prescribed density of the snake points, a snake interpretation step is also applied, to insert or delete snake points, if the snake becomes too sparse or dense, respectively, regarding the Euclidean distance. The problem with the traditional snake model and early algorithms (e.g. [2]) is that it provides poor convergence to object concavities and the initial snake should be close to the desired boundary. Recently, improved snake methods were proposed [3, 4] to overcome these difficulties. However, these approaches are computationally expensive and many iteration steps might be needed to occupy concavities. Thus if the snake contains many points (like in [5], where the final snake is used as an input for object recognition), it is highly recommended to save iteration steps. A simple approach for that is presented in [6]. Throughout this paper, we will consider the Gradient Vector Flow (GVF) snake [4] as a basis for our investigations, which is known about its large capture range and good concavity performance.

A usual approach to save computations within snake iteration processes is to reduce the spatial resolution of the image, see e.g. [7]. Multiresolution techniques are highly popular also in other fields of machine vision [8, 9]. Accordingly, a resolution pyramid is considered, and the search is performed in a coarse to fine way. However, as it can be pointed out (see e.g. [10]), the major drawback of this approach is that it merges groups of pixels in a completely blind way, just according to their spatial position. A more natural approach is to classify pixels to match the feature extraction idea. In this way, we obtain a reduction of the spatial resolution which adapts to the image content. One possible approach to reach this aim is to use e.g. quadtree decomposition of the image, but any other adaptive decompositions can be suitable, where the resulting decomposition grid can be handled easily regarding its spatial topology.

It is natural to perform resolution reduction based on the forces directing the snake. That is, if the snake is directed similarly in a group of pixels, those pixels can be emerged with the average force assigned. Moreover, if we have some tolerance in the emerging process, we can eliminate local small inadequacies (e.g. noise). However, after extracting a more rough representation of the field directing the snake, it is a challenging task to adopt the active contour theory. Now, snake points are represented by larger blocks of varying size, and both the internal and external forces must be defined accordingly. For short correspondence we will refer to our approach as heterogeneous GVF (HGVF) snake.

The paper is organized as follows: section 2 overviews the classic theory of active contour models with focusing on the GVF snake approach. In section 3 we present how the resolution reduction is achieved by considering a quadtree decomposition based on the GVF external force field map. We explain how the pixelwise snake iteration is adopted to quadtree representations in section 4. Section 5 contains experimental results on how the number of iterations can be decreased

Research was partly supported by the project SHARE: Mobile Support for Rescue Forces, Integrating Multiple Modes of Interaction, EU FP6 Information Society Technologies, Contract Number FP6-004218.

on the sparse representation. Finally, some conclusions are drawn in section 6.

2. CLASSIC GVF SNAKE APPROACH

Snakes or active contours became popular after the seminal paper of Kass, Witkin and Terzopoulos [1]. Snakes are formulated as energy-minimizing contours controlled by two forces: internal contour forces which enforce smoothness constraint, and image forces which attracts the contour to the desired features (edges). Representing the position of the snake parametrically by $v(s) = (x(s), y(s))$ $(0 \leq s \leq 1)$, the energy of the snake is defined as a sum of the internal energy of the snake and the image energy:

$$E_{snake} = \int_0^1 E_{int}(v(s)) + E_{image}(v(s))ds.$$

The internal energy is composed of a first-order term controlled by $\alpha(s)$ and a second order term controlled by $\beta(s)$:

$$E_{int} = (\alpha(s)|v_s(s)|^2 + \beta(s)|v_{ss}(s)|^2)/2.$$

The terms v_s and v_{ss} represent the first and second derivative of v, respectively. $\alpha(s)$ describes the tension along the snake and $\beta(s)$ characterizes the bending of the curve. Usually, $\alpha(s)$ and $\beta(s)$ are set as values α and β.

The image energy is usually derived from the edge energy E_{edge} of the image weighted appropriately by a negative weight:

$$E_{image} = -w_{edge}E_{edge}.$$

A snake that minimize the energy functional E_{snake} must satisfy the Euler equations:

$$\alpha x_{ss} + \beta x_{ssss} + \frac{\partial E_{image}}{\partial x} = 0,$$
$$\alpha y_{ss} + \beta y_{ssss} + \frac{\partial E_{image}}{\partial y} = 0,$$

where x_{ss} and x_{ssss} are the second and fourth derivatives of x, and similarly for y_{ss} and y_{ssss}.

In computer implementation, the energy functional is discretized as:

$$E_{snake} = \sum_{i=1}^{n} E_{int}(i) + E_{image}(i),$$

where by $v_i = (x_i, y_i) = (x(ih), y(ih))$, E_{int} can be approximated as:

$$E_{int} = \frac{\alpha_i|v_i - v_{i-1}|^2}{2h^2} + \frac{\beta_i|v_{i-1} - 2v_i + v_{i+1}|^2}{2h^4},$$

for a closed snake with $v_1 = v_n$.

Gradient Vector Flow (GVF) [4] is a type of external force for active contours. The GVF was created to overcome two shortcomings of the original active contour formulation i.e. poor convergence to concave boundaries and sensitivity to initialization. GVF is computed as a diffusion of the gradient vectors of a gray-level edge map derived from the image. The GVF field is defined as the vector field $G(x; y) = (q(x; y); r(x; y))$ that minimizes the energy functional

$$\int\int \mu(q_x^2 + q_y^2 + r_x^2 + r_y^2) + |\nabla E|^2|G - \nabla E|^2 dxdy,$$

where the edge map $E(x; y)$ is derived from the image. Using calculus of variations, the GVF can be found by solving the following Euler equations:

$$\mu\nabla^2 q - (q - E_x^2 + E_y^2) = 0,$$
$$\mu\nabla^2 r - (r - E_x^2 + E_y^2) = 0,$$

where μ is a parameter to be set according to the amount of the noise. q and r can be determined through an iterative process. GVF can be incorporated into the snake algorithm through $q = \frac{\partial E_{image}}{\partial x}$, and $r = \frac{\partial E_{image}}{\partial y}$.

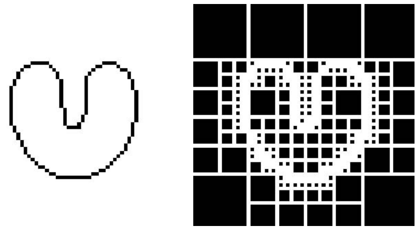
3. QUADTREE DOMAIN FOR SNAKE ITERATION

It is well-known that the parameters of the quadtree procedures provide wide flexibility regarding the desired size and intensity distribution of the cells. We can choose from several approaches to reach such a quadtree representation of the image which is suitable for GVF snake iteration. Naturally, the selected rule must be in accordance to the gradient behavior of the image. Consequently, we can consider simple decomposition based on intensity value (esp. for binary images), and a gradient related approach can be selected in the general grayscale/color image domain. Such a decomposition can be achieved e.g. by calculating the GVF external force field in the way discussed in section 2. Then the rule to stop the consequent decomposition of a cell can be that the direction of the pixelwise force field vectors must be sufficiently homogeneous within the cell. Note that in this way we can influence noise tolerance by adjusting the appropriate threshold. Figure 1 depicts a test input figure (U-shape) together with its quadtree decomposition based on simply the intensity values.

To obtain a quadtree decomposition that is natural for the general (grayscaled/color) case, we might as well consider the divergence of the GVF field to have an input for quadtree decomposition. Let $F(x, y) = P(x, y)\mathbf{i} + Q(x, y)\mathbf{j}$ be the GVF field of the image, where **i** and **j** are the horizontal and vertical unit vectors respectively, and x, y are pixel coordinates. The divergence of F is the scalar field [11]:

$$divF = \frac{\partial P}{\partial x} + \frac{\partial Q}{\partial y}. \quad (1)$$

The physical significance of the divergence is the rate at which flow "density" exits a given region of space. In the



(a) (b)

Fig. 1. Quadtree domain for snake iteration, (a) input image, (b) its quadtree decomposition.

divergence field, low values correspond to the object boundaries, while large values to those areas which are far from the boundaries.

4. FORCES FOR HGVF SNAKE ITERATION

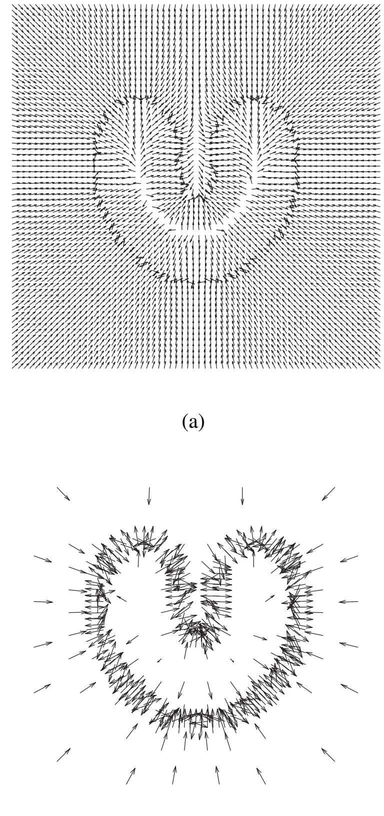
After obtaining the quadtree representation, we consider the average of the GVF external force vectors within each cell to have a mean representation, which vector will be considered in the iterative process to move the snake point represented by the given cell. For the result of the simplification of the GVF external force field, see Figure 2.

The effect of an external force vector is defined in a natural way. Namely, the cell is directed according to it, and we select a corresponding candidate which is 8-connected in the given direction to move to. After finding the candidate cell, the vector connecting the center of the current and candidate cells is considered as a new external force vector. See Figure 3 for the selection of a candidate cell and the corresponding new external force vector.

On the other hand, the allowable snake deformation is ruled by inner forces, as well. The pixelwise procedure can be repeated here, but only for the snake points represented by the cell centers. However, as the Euclidean distance of the consecutive cells can be quite large now, an extended interpolation technique is used to make the consecutive snake cells be 8-connected. Consequently, in every iteration step, the sparse snake points are connected by a sequence of spatially 8-connected cells. To fill in the gap between snake points, all the quadtree cells are inserted into the snake that intersects the digital line (obtained e.g. by Bresenham algorithm [12]) between the selected centers. See also Figure 4 for the procedure to connect HGVF snake points.

5. IMPROVED HGVF ITERATION PERFORMANCE

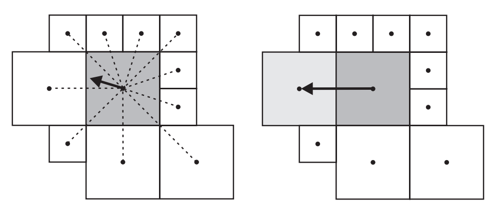
In our example, it can be nicely observed that the HGVF iteration is capable to reach roughly the object boundaries in few



(a)

(b)

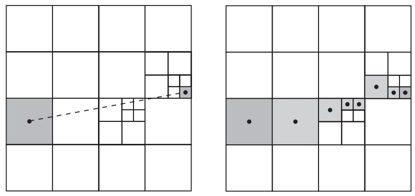
Fig. 2. Simplifying the GVF external force field according the quadtree decomposition, (a) pixelwise GVF force field regarding the original image, (b) simplified force field according the quadtree decomposition.



(a) (b)

Fig. 3. Finding the candidate among neighboring quadtree cells regarding external force field directed movement, (a) comparing the cell mean vector with the ones pointing to the topological neighbors, (b) selecting new external force field vector pointing to the candidate neighbor.

iteration steps. See Figure 5 accordingly.

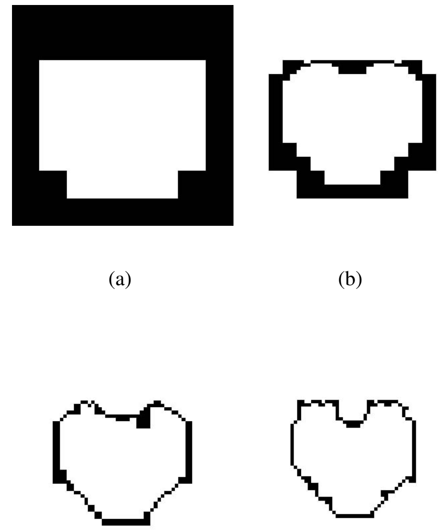


(a) (b)

Fig. 4. Connecting sparse snake points, (a) considering a digital line segments bounded by quadtree cells, (b) adding in-between cells to obtain an 8-connected snake.

6. CONCLUSION AND DISCUSSION

The presented basic model can be improved and extended in many ways. On the one hand, the quadtree (or any other large block) representation offers many advantages. We do not have to mind noise and boundary linkage problems, as these local inadequacies can be covered by larger quadtree blocks. By adjusting the parameters of quadtree decomposition, we can trade between noise (and perhaps useful image data) suppression and higher iteration speed gained by larger quadtree cells. Additional speed-ups can be achieved in the proposed method e.g. with revealing direct relations between the quadtree representation and topological (spatial) placement of the grid elements, like in [13]. The usage of heterogeneous snakes looks promising to gain computational performance. However, to measure up the true efficiency of our method, the above extensions are needed to be implemented. These realizations regarding more realistic scenarios are the main directions of our future research in this topic.



(a) (b) (c) (d)

Fig. 5. Performance of the HGVF snake, (a) initial snake points (quadtree cells), (b) snake after one iteration step, (c) snake after two iterations, (d) snake after three iterations.

7. REFERENCES

[1] M. Kass, A. Witkin and D. Terzopoulos, “Snakes: Active contour models, ” Int. J. Comp. Vis., vol. 1, pp. 321-331, 1987.

[2] L.D. Cohen and I. Cohen, “Finite-element methods for active contour models and balloons for 2-D and 3-D images,” IEEE Trans. PAMI, vol. 15, pp. 1131-1147, 1993.

[3] D. Gil and P. Radeva, “Curvature vector flow to assure convergent deformable models for shape modelling,” LNCS, vol. 2683, pp. 357-372, 2003.

[4] C. Xu and J.L. Prince, “Snakes, shapes, and gradient vector flow,” IEEE Trans. IP, vol. 7, pp. 359-369, 1998.

[5] A. Hajdu, A. Roubies and I. Pitas, “Optimized chamfer matching for snake-based image contour representations,” ICME, 2006.

[6] A. Hajdu, A. Roubies and I. Pitas, “Improving the performance of the GVF snake algorithm,” ISCCSP, 2006.

[7] N. Ray, B. Chanda and J. Das, “A fast and flexible multiresolution snake with a definite termination criterion,” Pattern Recognition, vol. 34(7), pp. 1483-1490, 2001.

[8] G. Borgefors, “Hierarchical Chamfer Matching: A Parametric Edge Matching Algorithm,” IEEE Trans. PAMI, vol. 10(6), pp. 849-865, 1988.

[9] G. Borgefors, G. Ramella and G. Santini di Baja, “Hierarchical Decomposition of Multiscale Skeletons,” IEEE Trans. PAMI, vol. 23(11), pp. 1296-1312, 2001.

[10] D. Huttenlocher, G. Klanderman and W. Rucklidge, “Comparing images using the Hausdorff distance,” IEEE Trans. PAMI, vol. 15(9), pp. 850-863, 1993.

[11] E.C. Young, “Vector and tensor analysis,” *Marcel Dekker*, 1993.

[12] J.E. Bresenham, “Algorithm for computer control of a digital plotter,” IBM Systems Journal, pp. 25-30, 1965.

[13] L. Balmelli, J. Kovacevic and M. Vetterli, “Quadtrees for Embedded Surface Visualization: Constraints and Efficient Data Structures,” ICIP, 1999.