Focal-Plane Moving Object Segmentation for Real-Time Video Surveillance

David Lopez Vilariño
Dep. Electronics and Comp. Science
University of Santiago de Compostela
Santiago de Compostela, SPAIN

Piotr Dudek
Dep. Electrical and Electronics Eng.
University of Manchester
Manchester, UK

Diego Cabello Ferrer
Dep. Electronics and Comp. Science
University of Santiago de Compostela
Santiago de Compostela, SPAIN

Abstract— **In this paper a new technique for segmenting and tracking moving objects in a user-defined control area is presented. It is based on an active contours technique called Pixel-Level Snakes (PLS) whose capabilities to manage changes of contour topology and to introduce additional constraints in the contour evolution are used to define a control area as well as to segment and track moving objects. Furthermore, PLS can reach a very high speed of response when they are implemented on a pixel-parallel hardware platform. To illustrate the validity of the proposal some examples and results regarding the computation time achieved in the implementation of the proposed algorithm on a cellular processor array (SCAMP-3 vision chip) have been included.**

I. INTRODUCTION

In recent years, applications dealing with automatic surveillance and monitoring have become increasingly important. For example, traffic monitoring tasks such as detection of anomalies (slow or heavy traffic, vehicles stopped in a crossroad, etc.), classification or counting of automobiles for statistics and forecasts on the traffic fluidity, and detection and alert of accidents, benefit from automatic systems [1].

A complete video monitoring system should contain a low-level processing stage in order to detect and segment the moving objects from video sequences; a middle-level processing stage where primitives are extracted and finally a high-level processing stage where the results are interpreted and suitable actions are carried out (alarm activation, semaphores managing, etc.).

In this work, we focus our attention on the low-level tasks: the moving object detection and segmentation. These probably represent the more critical tasks in the complete system for two reasons: on one hand, their efficiency is fundamental to successfully approach the subsequent classification and interpretation steps; on the other hand, this stage acts on a greater volume of data and so it is essential to guarantee the high speed response needed in real-time systems. A good survey of moving object segmentation techniques can be found in [2].

Active contours [3,4] are widely acknowledged as a powerful technique for segmentation and tracking because of their flexibility and ability to integrate information from different sources. Furthermore, the assumption that the moving objects appear only slightly shifted and/or deformed in two consecutive frames of a video sequence makes their segmentation particularly suitable to the application of the active contour techniques. Once the active contours delineate the objects of interest in one frame the segmentation proceeds on to the next frame taking as the initial contours the ones resulting from the previous frame. However the problem of initialising the contours in the first frame of the sequence still needs to be solved. In some applications, particularly in medical image processing, estimated models can help to define the initial contours. Unfortunately this approach is not suitable for video surveillance applications where the models of the objects to be tracked are usually unavailable. Therefore it is necessary to provide the first set of initial contours automatically.

The approach to setting the initial contours adopted in this paper is to fix one (or several) initial contours defining the perimeter of the region to be controlled. Thus, in contrast to the usual approach in which the contour is initialised away from the object of interest and then evolves towards it, in our case it is the moving object which comes towards the initial contour. Figure 1 illustrates this approach. A rectangular control area is defined, in such a way that only those objects which are going through this control area should be detected and tracked.

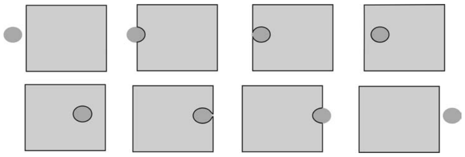

Figure 1: Processing of frames of an artificial sequence. A moving object (the circle) is tracked only when it stays inside the control area (the rectangle).

The moving object coming close to the control area will attract the active contour. When the object is completely inside of the control area, a topological transformation splits the initial contour in two new ones. One of this will delimit the object on its way inside the control area and the other one will restore the perimeter of the control area. Finally another topological transformation occurs when the object goes out of the control area merging again the two active contours into one. Therefore, the first step is to define a control area by an initial active contour surrounding the region to be controlled. In order to keep this initial active contour fixed:

1. The contour evolution outside of the control area has to be prevented.
2. Inflating forces are imposed to push the active contour outwards. Since we impose that the contour cannot flow out of the control region, it will keep the shape of the control perimeter when no other perturbation is present.

As a conclusion from the aforementioned methodology, the deformable contours will have the following features:

1. Capability for contour topology transformations, to segment moving objects.
2. Implementation of balloon forces, to restore and fix the control area after a topologic transformation.
3. Capability to introduce restrictions in the contour evolution, to prevent the expansion of the control area out of the defined perimeter.
4. Fast computation speed, to achieve real time video-flow processing.

An active contour technique which meets all the aforementioned requirements is that of so-called pixel-level snakes (PLS). PLS are defined as pixel-parallel deformable contours which allow to reach a high speed of computation when they are implemented in a massively parallel hardware system. In this paper we have implemented the above outlined strategy for tracking and segmenting moving objects based on the PLS technique. Furthermore, in order to provide a real-time and stand-alone system we have implemented the algorithms on the focal plane processor array SCAMP-3 vision chip [5].

The remainder of the paper is structured as follows: In Section 2, PLS are briefly described. In Section 3, the PLS-based algorithm for moving object segmentation is introduced. In Section 4, some experimental results from the implementation of the application on the SCAMP-3 vision chip are shown. Finally the conclusions are discussed in Section 5.

II. Pixel level snakes

Pixel-level snakes (PLS) are a topographic and iterative active contour technique based on a pixel-level discretization of the contours and a massively parallel processing of every contour cell. In the context of PLS, the active contours are represented as sets of 8-connected activated pixels in a binary image called contour image. This binary image has the same dimensions as the original image containing the objects or regions to be defined. The contour evolution consists of an iterative process of activation and deactivation of contour pixels based on binary local morphological operations extended along the four cardinal directions (dilation and thinning operations). The goal after each cycle is to obtain new well-defined contours slightly shifted and/or deformed so that they come closer and fit themselves to the boundaries of interest. Like conventional active contour techniques, the contour evolution is controlled by forces derived from external and internal potential fields. The external potential should be defined in such a way that the boundaries of interest coincide with the valleys of the potential field. The internal potential is estimated directly from the contours by means of a recursive low-pass filtering from which internal forces depending on the local curvature are derived. These internal forces push the contour to keep a smooth shape. Furthermore PLS incorporate the estimation of the so-called balloon potential from which inflating/deflating forces are derived to the contour expansion/compression.

The iterative pixel to pixel shift and the explicit definition of the pixel-level active contours allow a high degree of control on the contour topology making possible to preserve it or to manage topologic transformations in a simple way. The topology of the contours can be preserved by a pre-estimation of the contour location and shape in the next iteration from which the possible collision between contours can be foreseen and consequently prevented (collision point detection). The changes in the contour topology can be managed by simple inspections of the contour map. There are different algorithmic proposals to approach the topologic transformations with PLS. One of the more efficient consists of a redefinition of the contours as the sets of boundary pixels of the associated regions (area enclosed by the active contours). As a consequence, the topologic transformations are implicitly performed because these contour pixels that are representing a collision no longer belong to the sets of frontier pixels of the associated regions and therefore they will not be considered to be contour pixels after the contour map redefinition. Figure 2 shows a block diagram with the main modules to be implemented in a PLS algorithm. A comprehensive review of PLS can be found in [6].

PLS have been implemented on hardware architectures with pixel-parallel processing capabilities, such as ACE4K [6,7] (based on cellular neural network paradigm) and SCAMP-3 [8,9] (based on SIMD processor array). As a consequence a high speed of processing is achieved.

III. Moving object segmentation approach

As it was mentioned earlier, the active contours technique PLS is suitable to perform the moving object segmentation discussed in this paper. However in order to exploit the capabilities of PLS, an appropriate external potential field has to be derived from the images under processing. This should contain enough information to guide the active contours toward the boundaries of the moving objects inside the control area. In addition, the operations involved in the external potential estimation have to be simple enough to be directly implemented in a cellular processor array and then to achieve high speed of processing.

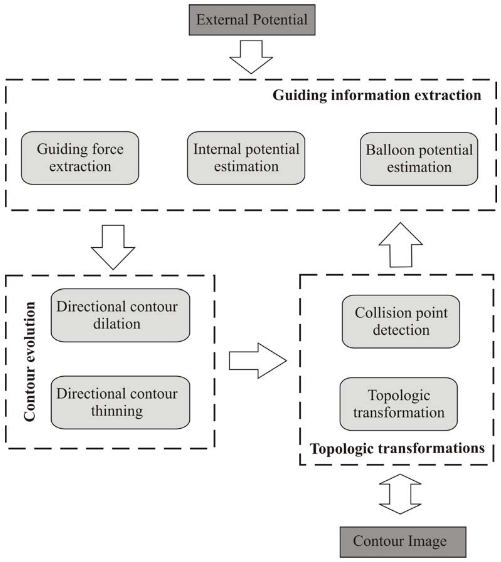
External Potential
Guiding information extraction
Guiding force extraction
Internal potential estimation
Balloon potential estimation
Contour evolution
Directional contour dilation
Directional contour thinning
Collision point detection
Topologic transformation
Topologic transformations
Contour Image

Figure2: PLS conceptual block diagram.

Our proposal consists of the estimation of the external potential field from the difference between the current frame and a reference frame (usually the first frame of the sequence). The resulting difference image is binarized by thresholding (adaptive or simple) and morphological operators (opening and closing) are applied in order to eliminate noise. Then, a binary edge detection is computed followed by a distance to border estimation. This last operation consists of the assignment of a value to each pixel of the image proportional to the distance to the closest boundary (based on City-Block distance). Finally, the distance map is diffused to alleviate the complexity of the directional gradient estimator needed to derive the external forces from this potential field. In Figure 3 the flow diagram of the operations involved in the external potential estimation is shown. To be able to implement the external potential estimator in current SIMD with local connectivity the distance to border map is created by a weighed combination of the outputs from recurrent dilations on the edge map.

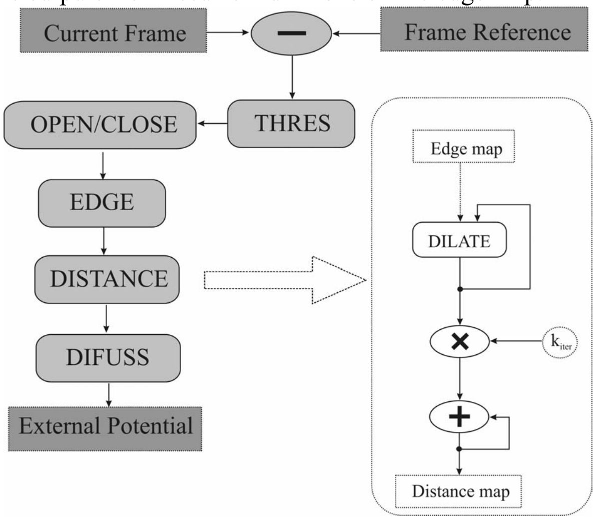

Figure 3: Estimation of the external potential field from the image under processing

The weight k_{iter} is included to prevent a uniform potential field (i.e. with a constant gradient). Thus, the guiding forces originating from the external potential field will be stronger close to the edges. Therefore, when a moving object is completely inside the control area, after the topological transformation the inflating forces will overcome the external forces and this will allow to restore the control area. Note that the number of iterations in the distance map estimation (*i.e.*, the number of dilation operations) is related to the range of the guiding forces. The more iterations the longer the distance between the initial contour (from previous frame) and the objective (current frame) can be managed. On the other hand, this number of iterations is related to the number of bits required to define the external potential field. This number cannot be determined in a general form. It will depend on the camera position and the speed of the moving objects. Nevertheless, it is obvious that a high speed of computation (high frame rate) will reduce the distance between contours in consecutive frames and consequently reduce the requirements for long distance computation.

The described segmentation approach has been implemented based on the PLS algorithms introduced in [8,9] and tested on traffic sequences[1]. The frame size of the image sequences is of 768x577 pixels. Such image size can be managed with PLS implemented on a PC. Nevertheless, PLS are really advantageous when they are implemented on focal-plane massively parallel processor arrays like ACEx or SCAMP visual processors. Unfortunately, the current generation of these vision chips includes arrays of only 128x128 processing elements. With the aim of getting more reliable results from simulation, the images to be processed are fitted to that chip size by sampling the region of interest to suitable dimensions for the on-chip computation as it is shown in Figure 4

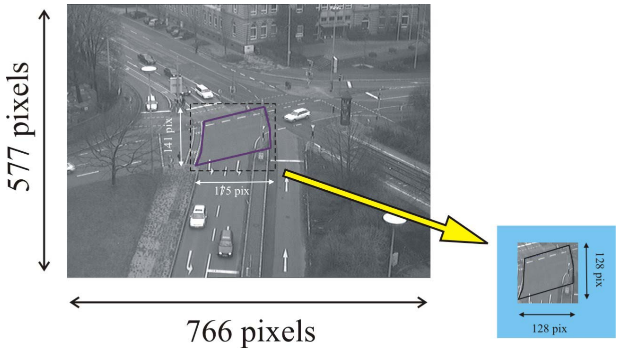

Figure 4: Adaptation of the video surveillance application to the current cellular SIMD chip-size.

In Figure 5 some results of the processing of an image sequence are shown. Note that the shape of the control area can be irregular and not necessarily rectangular.

[1] Copyright ©1998 by H.H. Nagel. Institut Für Algorithmen und Kognitive Systeme. Fakultät für Informatik, Universität Karlsruche (TH). Postfach 6980, D-76128 Karlsruhe, Germany. http://i21www.ira.uka.de/.

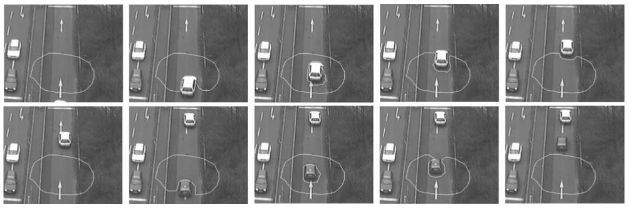

Figure 5: Processing of a traffic sequence based on the proposed method.

IV. ON-CHIP EXPERIMENTS

The proposed technique for video surveillance has been implemented on the SCAMP-3 vision system. The SCAMP-3 is a general-purpose programmable vision chip with a 128x128 SIMD processor-per-pixel array (with embedded photosensors) fabricated in a 0.35μm CMOS technology. The chip executes a sequence of simple array instructions (addition, inversion, one-pixel shift, etc.), which operate at a rate of 1.25 MOPS per pixel with a very low power consumption (below 250mW for the entire chip). These features make SCAMP-3 a good candidate to host applications with fast computation speed and/or low power consumption requirements, and several algorithms implementing pixel-level snakes have been executed on the SCAMP-3 system. The testing results demonstrated that the system can execute PLS in about 0.5 ms/iteration (about 2000 iterations per second). Taking into account that usually each frame processing requires about 15 PLS iterations, the combination PLS-SCAMP will be able to compute at higher frame rates than conventional active contour-PC approaches.

In Figure 6 some experimental results from the image capturing and processing with the SCAMP system are shown. In these experiments, the external potential was estimated using edge detection. Note that a merging of the active contours following two different objects occurs. This is so, because the camera in its position actually captures an overlapping between objects. This occlusion problem should be taken into account (and eventually prevented) in subsequent high level processing.

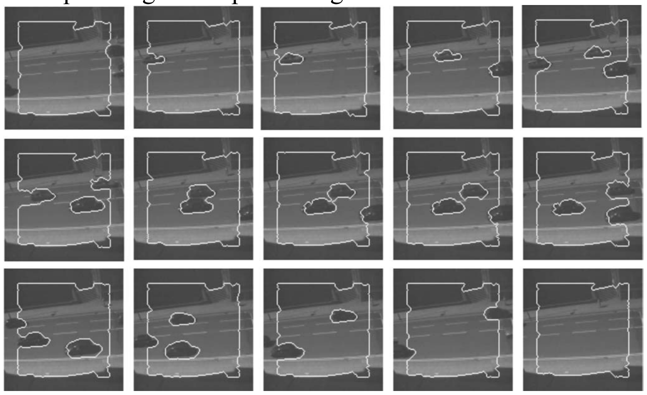

Figure 6: Sequence captured and processed with SCAMP-3 vision chip.

In Figure 7 another sequence of the processing with the SCAMP system is shown. In this case a pedestrian is detected and tracked when walks inside the predefined control area.

V. CONCLUSIONS

In this paper, it is proposed a technique for segmenting moving objects intended for video surveillance applications. It is based on the definition of control areas with active contours which will be only perturbed when any object tries to pass through them. If that happens the corresponding active contour will deform to involve the moving object and eventually will split in two contours. One of these will restore the control area whereas the other one will fit the boundaries of the moving object. When the object reaches the control area limits a new topologic transformation will result in the merging of the active contour which defines the control area and that which followed the object. Thus, the object is no longer tracked when it goes out of the control area.

These active contours are implemented as pixel level snakes provided with all the required functionalities to perform the proposed methodology. The pixel-wise processing allows the implementation of the algorithm on fine-grain SIMD processor arrays to achieve high speed of response. In this paper we have developed a suitable control for PLS in the described framework of moving object segmentation. We have implemented and tested the complete algorithm performing real-time segmentation on a cellular processor array chip. The PC simulations and the experimental results from the on-chip implementation of the algorithm, demonstrate the validity of this approach in terms of efficiency and speed of the image computation.

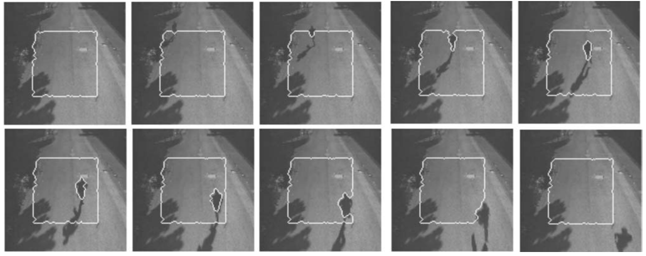

Figure 7: Sequence captured and processed with SCAMP-3 vision chip.

ACKNOWLEDGMENT

The authors would like to thank to Xunta de Galicia, which has partly supported this work (PGIDIT06TIC10502PR).

REFERENCES

[1] J.B. Kim, H.J. Kim. Efficient Region-Based Motion Segmentation for a Video Monitoring System. Patt. Recognition Letters, 24:113-128, 2003.

[2] D. Zhang and G. Lu. Segmentation of Moving Objects in Image Sequence: A Review. Circuits Syst. Signal Proc. 20(2):143-183, 2001.

[3] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active Contours Models. International Journal on Computer Vision, 1:321–331, 1988.

[4] A. Blake and M. Isard. Active Contours. Springer-Berlag, 1998.

[5] P. Dudek and S.J. Carey. A General Purpose 128x128 SIMD Processor Array with Integrated Image Sensor. Elect. Lett. 42(12)678-679, 2006.

[6] D.L. Vilariño and Cs. Rekeczky. Pixel-Level Snakes on the CNNUM: Algorithm Design, On-Chip Implementation and Applications. Int. J. Circuit Theory Applicat. (21)2:189-204, 2005.

[7] G. Liñan, S. Espejo, R. Dominguez-Castro and A. Rodriguez-Vazquez. ACE4k: An analog I/O 64_64 Visual Microprocessor Chip with 7-bit Analog Accuracy. Int. J. Circuit Theory Applicat. 30:89–116, 2002.

[8] P. Dudek and D.L. Vilariño. A Cellular Active Contours Algorithm Based on Region Evolution. CNNA 2006. 269-274, 2006.

[9] D.L.Vilariño and P. Dudek,. Evolution of Pixel Level Snakes Towards an Efficient Hardware Implementation., ISCAS2007, 2678-2681, 2007.