A Real-Time solution to the image segmentation problem: CNN-Movels

Giancarlo Iannizzotto, Pietro Lanzafame, Francesco La Rosa
University of Messina (VisiLAB)
Italy

1. Introduction

2D Image segmentation has been a main issue in image analysis since the very early years. Traditional literature usually classifies segmentation approaches as *area-based* or *contour-based*. In the second class, among dozens of different approaches, *Active Contours* have recently gained more and more interest. Active contours (also known as *deformable models*) are open or closed curves that can accurately fit to the contours of objects featuring almost any kind of shape. These models are called *active* because they automatically respond to specific characteristics of the points of the image, by changing their shape consequently. For example, an active contour can respond to the *edgeness* values of the image points.

A particular type of active contour is the *snake*: it responds both to the characteristics of the points of the image (through the minimization of a quantity called external energy), and to specific internal laws ruling its shape and way of deformation, tending to minimize a quantity called *internal energy* (Kass et al., 1988; Lai & Chin, 1995).

It usually consist of elastic curves that, located over an image, evolve from their initial shapes and positions in order to adapt themselves to the notable characteristics of the scene. This evolution comes as a result of the combined action of external and internal forces. The external forces lead the snakes towards features of the image, whereas internal forces model the elasticity of the curves. In a parametric representation, a snake appears as a curve u(s)=(x(s),y(s)), s ϵ [0,1], with u(0)=u(1). Its internal energy is often defined as

$$E_i(u(s)) = \alpha|u_s(s)|^2 + \beta|u_{ss}(s)|^2 \tag{1}$$

A snake is made up of two factors: the membrane energy $\alpha|u_s(s)|^2$, which weights its resistance to stretching, and the thin-plate energy $\beta|u_{ss}(s)|^2$, that weights its resistance to bending. The terms $u_s(s)$ and $u_{ss}(s)$ represent the first and second derivatives respectively. The elasticity parameters α and β control the smoothness of the curve. The external energy is generally defined as a potential field P,

$$E_e(u) = \int_0^1 P(u(s))ds \tag{2}$$

Source: Vision Systems: Segmentation and Pattern Recognition, ISBN 987-3-902613-05-9,
Edited by: Goro Obinata and Ashish Dutta, pp.546, I-Tech, Vienna, Austria, June 2007

This external potential is a combination of different terms based on the application and the characteristics of interest.

The total energy of the snake will be the sum of the external and internal energy terms along the curve u(s):

$$E_{snake}(u) = \int [E_i(u(s)) + E_e(u(s))] ds \tag{3}$$

The solution to the problem of detecting the contour is found in the minimization of this energy function.

Some other variations of the snake (*snake spline*) are represented by its parametric formulations much quicker and computationally less expensive (Flickner et al., 1996). Snakes, in high-noise conditions, can *lose contact* with their primary target and can stick to some local maxima of the internal/external energy. On the other hand, very interesting results were obtained for automatic segmentation, even in presence of nested contours, by using *level-set methods* (Malladi et al., 1995).

An interesting unifying approach to segmentation is described in (Malladi & Sethian, 1996), where a class of constrained clustering algorithms for boundary extraction (as a generalization of known algorithms) is introduced. With T-snakes (McInerney, 1997), also the conventional snake approach was extended to provide the ability of splitting and merging. These algorithms generally seem to suffer from an intrinsic high computational complexity and from an effect of *contours smoothing* which can be undesired.

In (Iannizzotto & Vita, 1996) and, later, in (Iannizzotto & Vita, 2000), a new kind of active contour was introduced: this is composed by a chain of autonomous agents (MOVing elements: MOVels), which move independently but in a collaborative fashion over the image, according to some very simple rules and some image features. The idea of exploiting both homogeneity and non-homogeneity as pixel feature for image segmentation appears very attractive to overcome (at least, partly) the problem of noise sensitivity. In (Jones & Metaxas, 1998) an attempt to combine active contours with deformable models is made, but the process is split into two distinct steps: first edge detection, accomplished by means of a similarity-based function; then, a curve fitting process is applied to the resulting binary image, by initializing a balloon-like deformable model (Cohen, 1991) inside each contour and letting it inflate and fit the contour itself. In (Zhu & Yuille, 1996) region-growing and balloon-based approaches are unified in a common framework relying, in order to perform energy minimization, on a competition-based technique. A segmentation algorithm is introduced, based on a basic competitive learning approach according to the classification given in (Theodoridis & Kotroumbas, 1999), integrated with a probabilistic, bayesian decision criterion instead of the common similarity distance, and with a region-merging extension. The described approach assumes that the probability distribution of the point features are gaussian: this is usually not true. In their paper, the authors actually point out this problem, while enforcing the generality of their results for *any* probability distribution. However, no evidence is provided of this generality, and large part of the theoretical results seems to hold only for gaussian distributions. Finally, substantial prior information is exploited and needed, as prior probability distribution for the bayesian decision approach.

Recently, a different approach was introduced, which exploits autonomous agents randomly spread throughout the image (Liu & Tang, 1999). An agent is positioned in an area which is non-homogeneous (in a sense which is defined in the paper), it moves toward a homogeneous area. When it finds it, it breeds, producing new agents which will gradually cover this area. If an agent cannot find an homogeneous area, it doesn't breed and, after a given lifetime, it dies. The overall effect is that after a number of life-cycles, all the pixels in the image will be visited and classified, thus producing a segmentation. Moreover, since the main target of the agents is breeding, and breeding needs space for the offspring, in some sense the agents exhibit a competitive behaviour.

When strong CPU power consumption constraints must be met, and high computation speed is mandatory (real-time processing) advanced computing resources cannot be used and so it is preferable to adopt custom hardware.

An alternative approach to image processing is provided by the *Cellular Neural Network* (CNN) paradigm, introduced by Prof. L.O. Chua in 1988 (Chua & Yang, 1988a; 1988b). A CNN consists of a network of first order nonlinear circuits, locally interconnected by linear (resistive) connections. CNNs have been extensively used in image processing applications (Matsumoto & Yokohama, 1990) such as filtering, edge detection, character recognition (Sziràny & Csicsvàri, 1993) and object recognition (Milanova & Buker, 2000). Thanks to their architecture they can be applied to inherently parallel problems in which traditional methods cannot achieve a high throughput (Manganaro et al., 1999).

Various approaches to implementing real-time segmentation techniques on CNNs have been proposed (Rekeczky, 1999; Kozek & Vilarino, 1999; Vilarino et al., 2003). In (Rekeczky, 1999) “bias controlled trigger-waves” are used to determine the edge of an object in a scene, without, however, solving the problem of searching for nested objects.

(Kozek & Vilarino, 1999) and (Vilarino et al., 2003) proposed an image segmentation strategy based on either a continous or discrete-time CNN architecture, capable of revealing any nested objects in a scene, but the level of accuracy of the edges extracted was not investigated.

In the past, a still image segmentation technique (Iannizzotto et al., 2003) was developed, based on an active contour obtained via single-layer CNNs. The contour initially laid on the frame of the image shrinks, deforms and multiplies until it matches the edges of each of the objects present in the scene. The shape of each object in the image is accurately extracted and nested objects, if any, are correctly detected. Again, this technique suffer from sensitivity to noise as in the most of edge-based methods; noise may create insignificant false edges or determine some “edge fragmentation”.

The aim of this work is to re-formulate the algorithm proposed within (Iannizzotto et al., 2003) in order to step-over the weakness of this, and other similar, works. The technique accurately traces the edges of objects, nested at various levels, even in presence of false (or fragmented) edges.

The input to the system is a gray-scale image obtained by applying to the image a median filter and a gradient operator. Guided by statistical properties of *edgeness* of the image pixels, the chain adapts its shape to that of the objects in the image until it marks out their contours. The output is a set of closed chains of points, each representing the contour of a single object. In the following sections we will describe the techniques developed and present a set of experimental results, laying particular emphasis on the evaluation technique used. In the final section we will draw our conclusions on the work carried out and discuss future lines of research.

2. Cellular Neural Network

As stated in the introduction, a CNN consists of an array of non-linear, locally interconnected, first order circuits. As connections are local, each cell is connected only to the cells belonging to its neighbourhood, as it is shown in Fig.1.

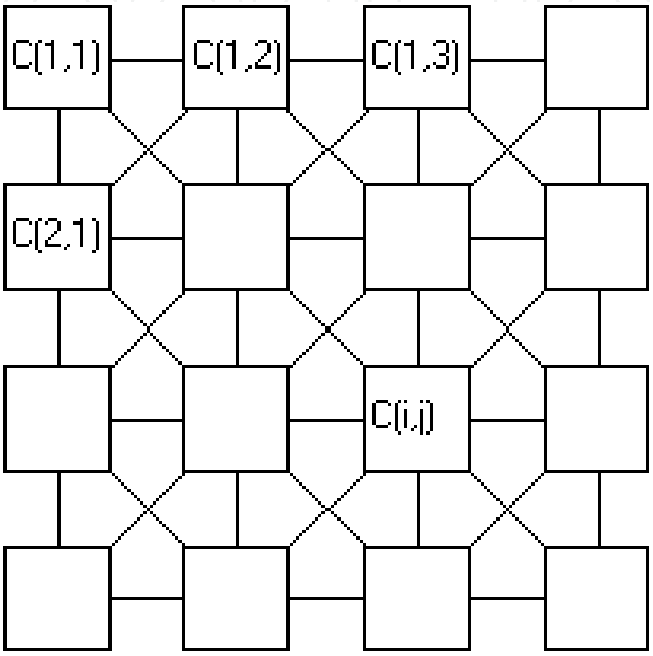

Fig. 1. Architecture of a CNN

If we call the generic cell in the *MxN* array as C_{ij} (the cell on the *i*-th row and the *j*-th column of the array), a formal definition of the neighbourhood of radius *r* of the cell C_{ij}, $N_r(i,j)$, is given by:

$$N_r(i,j) = \{C_{kl} : max\{|k-i|,|l-j|\} \leq r, 1 \leq k \leq M, 1 \leq l \leq N\} \tag{4}$$

An *MxN* CNN, with *MxN* cells arranged in *M* rows and *N* columns, is entirely characterized by a set of *MxN* nonlinear differential equations, associated with each cell. The generic cell x_{ij} is described by the following relations:

$$\begin{aligned} C\frac{dv_{x_{ij}}(t)}{dt} &= -R^{-1}v_{x_{ij}}(t) + \sum_{kl\in N_r} A_{ij,kl}v_{y_{kl}}(t) + \sum_{kl\in N_r} B_{ij,kl}v_{u_{kl}}(t) + I_{ij} \\ &+ \sum_{kl\in N_r} A1_{ij,kl}(\Delta v_{yy}) + \sum_{kl\in N_r} B1_{ij,kl}(\Delta v_{uu}) + \sum_{kl\in N_r} D_{ij,kl}(\Delta v) \\ v_{y_{ij}}(t) &= f\left(v_{x_{ij}}(t)\right) = 0.5\left(\left|v_{x_{ij}}(t)+1\right| - \left|v_{x_{ij}}(t)-1\right|\right) \end{aligned} \tag{5}$$

where:

$$
\begin{gathered}
\mathit{\Delta} v_{yy} = v_{y_{kl}}(t) - v_{y_{ij}}(t) \\
\mathit{\Delta} v_{uu} = v_{u_{kl}} - v_{u_{ij}} \\
\mathit{\Delta} v = v_{u,x,y_{kl}}(t) - v_{u,x,y_{ij}}(t) \\
\left| v_{x_{ij}}(t) \right| \le 1, \left| v_{u_{ij}} \right| \le 1, \left| I_{ij} \right| \le v_{max} \\
1 \le i \le M, 1 \le j \le N
\end{gathered}
\qquad (6)
$$

where $v_{x_{ij}}$, $v_{u_{ij}}$, $v_{y_{ij}}$ are respectively the state, input and output voltage of the CNN cell C_{ij}.

The state and output vary in time, whereas the input is kept constant. The indexes *ij* refer to the position of the cell in the 2D grid, while $kl \in N_r$ is a grid point in the neighborhood within the radius *r* of the cell *ij*. Matrices ***A***, ***B***, ***A1***, ***B1***, ***D***, called *templates*, describe the interaction of the cell with its neighbourhood and regulate the evolution of the CNN state and output vectors. Template connections can be realised by voltage-driven current generators.

$A_{ij,kl}$ is called linear feedback template, $B_{ij,kl}$ the linear control template, I_{ij} is a current bias in the cell. $A1_{ij,kl}$, $B1_{ij,kl}$ and $D_{ij,kl}$ are nonlinear templates respectively applied to $\mathit{\Delta} v_{yy}$, $\mathit{\Delta} v_{uu}$ and $\mathit{\Delta} v$. $A1_{ij,kl}$ is called difference controlled nonlinear feedback template, $B1_{ij,kl}$ is the difference controlled nonlinear control template, $D_{ij,kl}$ is the generalized nonlinear generator. The output characteristic *f* adopted is a sigmoid-type piecewise-linear function.

CNNs are exploited for image processing by associating each pixel of the image to the input or initial state of a single cell. Subsequently, both the state and output of the CNN matrix evolve to reach an equilibrium state. The evolution of the CNN is governed by the choice of the template. A lot of templates have already been defined in order to perform basic image processing operations, like gradient computation, smoothing, hole detection, line deletion, isolated pixel extraction and deletion, and so on. Simple operations can be performed just by using the basic templates ***A***, ***B***, and the bias ***I***, whereas more complicated processing requires the use of the nonlinear templates ***A1***, ***B1***, and the generalized nonlinear generator ***D***. The proposed algorithm can be totally implemented onto a "*CNN Universal Machine*" (CNN-UM), an hardware structure able to implement CNNs (Chua & Roska, 1993).

The main advantage of using CNNs in image processing is related to the increasing of throughput due to the massive parallelism of the structure, joined to the similar way of signal processing, typical of CNNs. In fact they are able to perform a complete image processing analysis in time of order of 10^{-6} s (by using a CNN hardware implementation), this in form of sequences of simple tasks like array target segmentation, background intensity extraction, target detection and target intensity extraction.

Depending on the type of neurons that are basic elements of the network, it is possible to distinguish continuous-time CNN (CTCNN), discrete-time CNN (DTCNN) (oriented especially on binary image processing), CNN based on multi-valued neurons (CNN-MVN) and CNN based on universal binary neurons (CNN-UBN). CNN-MVN makes possible processing, which is defined by some multiple-valued threshold functions, and CNN-UBN allows processing defined not only by threshold, but also by arbitrary Boolean function.

3. Proposed Strategy

In the algorithm presented the input to the system (a *continous-time single layer CNN*) is a gray-scale image processed by applying to the original image median and gradient operators. Guided by statistical properties of *edgeness* of the image pixels, obtained during preprocessing phase, the chain adapts its shape to that of the objects in the image until it marks out their contours. A basic block diagram of this algorithm is shown in fig. 2.

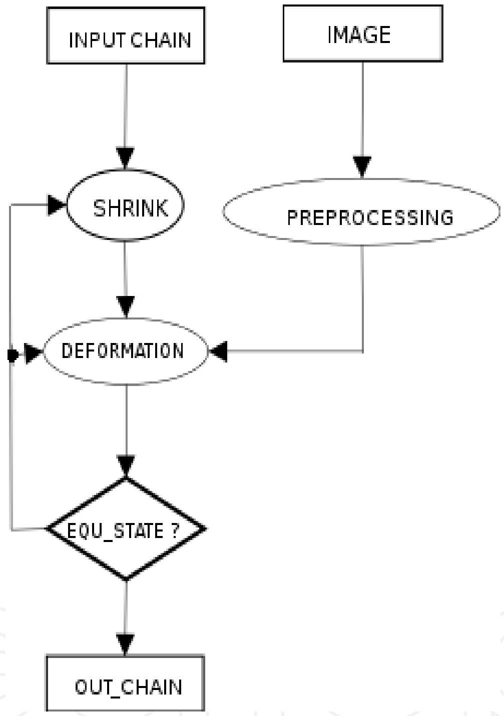

Fig. 2. Basic diagram of the algorithm

The original image is segmented via iterative shrinking and deformation of a chain, initially laid on the frame of the image. The chain shrinks across the whole image so as to reveal all the objects present, and deforms in order to adapt to the objects detected. The chain comprises a set of pixels, arranged over the image in such a way as to form a closed chain.
The sequence of operations adopted to shrink the chains is shown in fig. 3.

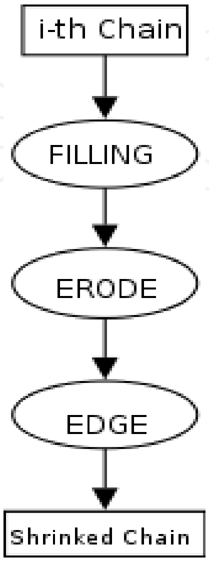

Fig. 3. Basic diagram of the shrinking process

The chain is initialised on the image to ensure that it contains all the objects in the scene. Once laid, the chain undergoes an iterative process of shrinking and deformation in order to adapt itself to the borders of the objects in the image. Shrinking occurs maintaining the shape of the input chain, while deformation is obtained by combining the information from statistical properties of edgeness (mean and standard deviation) of the image pixels and a binary image, result of preprocessing (see fig. 4) the original image. The iteration stops when a steady state is reached, i.e. the chain can't move any further.

The statistical properties (*mean and standard deviation* computed on 5x5 neighbourhood) are obtained applying templates suggested in (Moreira-Tamayos & Gyvez, 1999) on the edgeness image. “Mean” and “standard deviation” images, just obtained, are then combined through a weighted sum, as shown in fig. 5, by means of standard sum and product operation.

When a point of the chain meet a pixel of the image which features a very high value of *edgeness* this means that this point belongs to some object's border, so the chain should stick to this point. The point of the chain will therefore be “disabled”. At each step and for each point in the chain (MOVel) is computed a functional, if its value exceeds a fixed threshold the MOVel is disabled. This functional depend on neighbour edge points number and on pixel statistical properties. The functional is computed and thresholding is applied in the same step by means of templates shown in eq. 7.

This operation return a map of points of the chain that have to be disabled. Each extracted point presents three characteristics:

- high edgeness
- belong to a region with high average edgeness
- edgeness similar to that of its neighbours

In fig. 6 we can see input (Feature-1), bias (Feature-2), mask (i-th chain) and output (disabled points) of the operation just described. The threshold (implicit) depend on adopted parameters (see eq. 7).

$$A = \begin{vmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{vmatrix}$$

(7)

$$B = \begin{vmatrix} d & d & d & d & d \\ d & c & c & c & d \\ d & c & 0 & c & d \\ d & c & c & c & d \\ d & d & d & d & d \end{vmatrix}$$

$$I = Feature1$$

During its evolution, the chain may contain separate, not nested, objects to be detected. If, during iteration, non-adjacent points of the chain overlap, the chain splits into two chains, which continue to evolve independently of each other. To detect the presence of nested objects, if any, a *daughter* chain is generated inside each contour obtained, and the operations mentioned above are repeated on this chain. The *daughter* chains evolve until they reach the contours being sought or, if they do not contain any objects, implode and disappear. The search for nested objects is resumed whenever moving chains reach their steady state. It ends when all the moving chains have imploded.

This means that all the objects in the scene have been detected, making any further search useless.

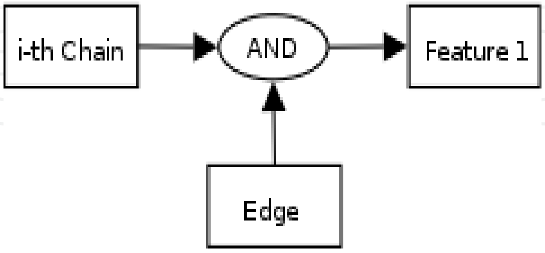

Fig. 4. Feature-1 extraction

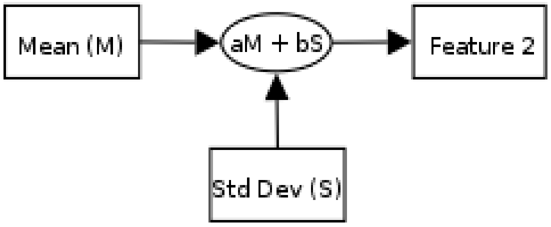

Fig. 5. Feature-2 extraction

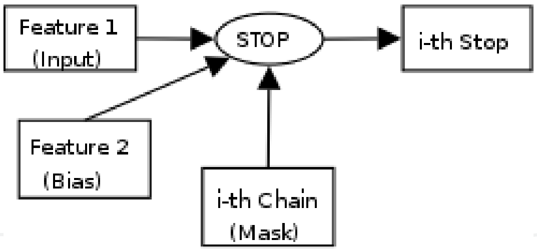

Fig. 6. Stopping phase

4. Accuracy Evaluation

Characterizing the performance of image segmentation approaches has been a persistent challenge. Performance analysis is important since segmentation algorithms often have limited accuracy and precision.

For some applications (e.g. medical images analysis), interactive drawing of the desired segmentation by domain experts has often been the only acceptable approach, and yet suffers from intra-expert and inter-expert variability. Automated algorithms have been sought in order to remove the variability introduced by experts, but no single methodology for the assessment and validation of such algorithms has yet been widely adopted.

An automated algorithm is compared to the segmentations generated by a group of experts, and if the algorithm generates segmentations sufficiently similar to the experts it is regarded as an acceptable substitute for the experts.

The most appropriate way to carry out the comparison of an automated segmentation to a group of experts segmentations is so far unclear. A number of metrics have been proposed to compare segmentations, including volume measures, spatial overlap measures, such as Dice (Dice, 1945) and Jaccard similarities (Jaccard, 1912), and boundary measures, such as the Hausdorff measure (Huttenlocher et al., 1993). Agreement measures between different experts have also been explored for this purpose. Studies of rules to combine segmentations to form an estimate of the underlying true segmentation have as yet not demonstrated any one scheme to be much favourable to another.

We present here a new algorithm for estimating the ground truth segmentation from a group of experts segmentations. Then, we employ the estimated ground truth to assess and validate the results of our segmentation technique. To estimate a ground truth we use a technique known as Active Shape Model (*ASM*) with the aim to synthesize a model representative of a training set (segmentations generated by a group of experts).

For this technique (Cootes & Taylor, 1992) the shape of an object is represented by a set of *n* points, which may be in any dimension. Commonly the points are in two or three dimensions.

The training set typically comes from hand annotation of a set of training images through landmarking. Good choices for landmarks are points which can be consistently located from one image to another: in two dimensions points could be placed at clear corners of object boundaries, "T" junctions between boundaries or easily located biological landmarks. This list would be augmented with points along boundaries which are arranged to be equally spaced between well defined landmark points. By analysing the variations in shape over the training set, a model is built which can mimic this variation.

If a shape is described by *n* points in *d* dimensions we represent the shape by a *nd* element vector formed by concatenating the elements of the individual point position vectors. For instance, in a 2-D image we can represent the *n* landmark points, (x_i; y_i), for a single example as the 2*n* element vector, **x**, where

$$x = \left(x_{1,} \ldots, x_n, y_{1,} \ldots, y_n\right)^T \tag{8}$$

Given *s* training examples, we generate *s* such vectors $\boldsymbol{x}_j$. These vectors form a distribution in the *nd* dimensional space in which they live. If we can model this distribution, we can compare the model obtained in such a way with the segmentation result of our system processing. In particular we seek a parameterized model of the form x = M(b), where b is a vector of parameters of the model.

Through Principal Component Analysis (PCA) we build a model of the object shape to segment obtaining also a dramatic reduction in size of the training set data. To obtain the model we execute the following steps:

1. Compute the mean of the data,

$$\bar{x} = \frac{1}{s}\sum_{i=1}^{s} x_i \tag{9}$$

2. Compute the covariance of the data,

$$S = \frac{1}{s-1}\sum_{i=1}^{s}(x_i - \bar{x})(x_i - \bar{x})^T \tag{10}$$

3. Compute the eigenvectors, φ_i and corresponding eigenvalues λ_i of **S** (sorted so that $\lambda_i \le \lambda_{i+1}$).

If **Φ** contains the *t* eigenvectors corresponding to the largest eigenvalues, then we can approximate any of the training set, **x** using

$$x \approx \bar{x} + \Phi b \tag{11}$$

where **Φ** = ($\varphi_1 | \varphi_2 |$ ... $| \varphi_t$) and **b** is a *t* dimensional vector given by

$$b = \Phi^T (x - \bar{x}) \tag{12}$$

The vector **b** defines a set of parameters of a deformable model. By varying the elements of **b** we can vary the shape, **x** using Equation 11. The variance of the i^{th} parameter, b_i, across the training set is given by λ_i. By applying limits of $\pm 3\sqrt{\lambda_i}$ to the parameter b_i we ensure that the shape generated is similar to those in the original training set.

The number of eigenvectors to retain, *t*, can be chosen so that the model represents some proportion (e.g. 98%) of the total variance of the data, or so that the residual terms can be considered noise.

To estimate the quality of our system applied on a still-image, the results obtained by a group of experts have been collected and then used as training set to build an ASM.

Once built the model, this is compared with the result of our system. The comparisons have been done using a normalized version (with respect to the number of selected landmarks) of the Mahalanobis distance.

Fig. 7 shows an image selected from the test set alongside the relative segmentation images. Fig. 8 and fig. 9 show respectively the segmentation image manually obtained by a human operator and the one produced by our algorithm. In this case the normalised error is equal to 0,2. This is an intermediate value between those obtained but, as direct comparison shows, the resulting segmentation is visually acceptable.

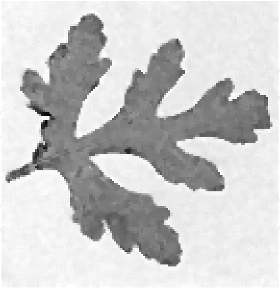

Fig. 7. An image selected from the test set

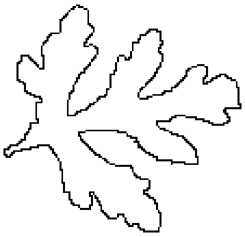

Fig. 8. Manual segmentation

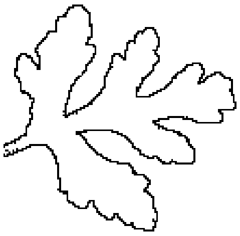

Fig. 9. Automatic segmentation

5. Experimental Results

In order to show the validity of the proposed algorithm, we provide the results obtained on the same set of test images used in (Iannizzotto et al., 2003). Accurate measures were performed on a set of 20 gray-scale images, specially selected for their contents. As mentioned in section 4, the method chosen to evaluate the results obtained by the algorithm is based on a comparison between the segmentation image obtained automatically and the estimated ground truth obtained as previously described (see par. 4). In Fig. 10 a selection of the test images is shown. Figs. 11 and 12 respectively show the segmentation images obtained by a group of experts and those obtained by applying the algorithm being proposed. Fig. 13 is a plotting of the error produced by our algorithm against the processed image. As the graph in Fig. 13 shows, the error on the test images is bounded around an average value of 0.2 in comparison with 0.3 obtained applying the same validation technique to the algorithm described in (Iannizzotto, 2003). These results show a reduction of the average error for the segmented images. It is due both to the used metric, which is less sensitive to impulsive noise, and to an actual improvement of the algorithm performances. In fact, the use of median operator has got rid of some peak in the error trend (Iannizzotto, 2003), caused by noise in the image, and the use of statistical features has made possible a "generalized" reduction of the error.

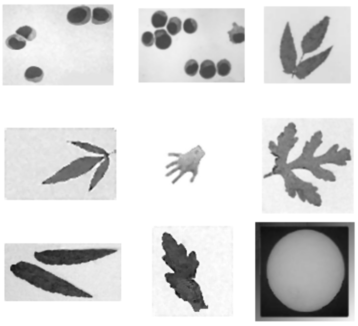

Fig. 10. A selection of the test images

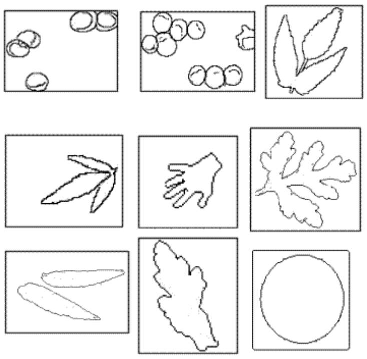

Fig. 11. Manual segmentation of the test images

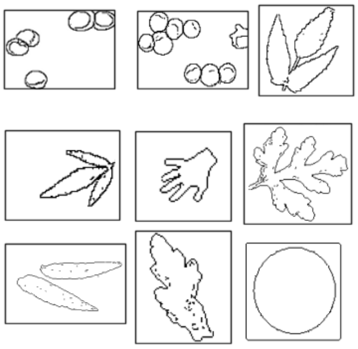

Fig. 12 Automatic segmentation of the test images

6. Competitive Approach

The technique we proposed, although effective in its results, is still affected by some parameter dependences: thresholds and weighting values used during computation. A possible solution, is an approach to image segmentation based on competing chains. Each chain acts as a competitive active contour which reacts to image features. Our work aims at producing a framework in which image segmentation is performed without any user input (namely, *unsupervised segmentation*) and with the minimum amount of prior information.

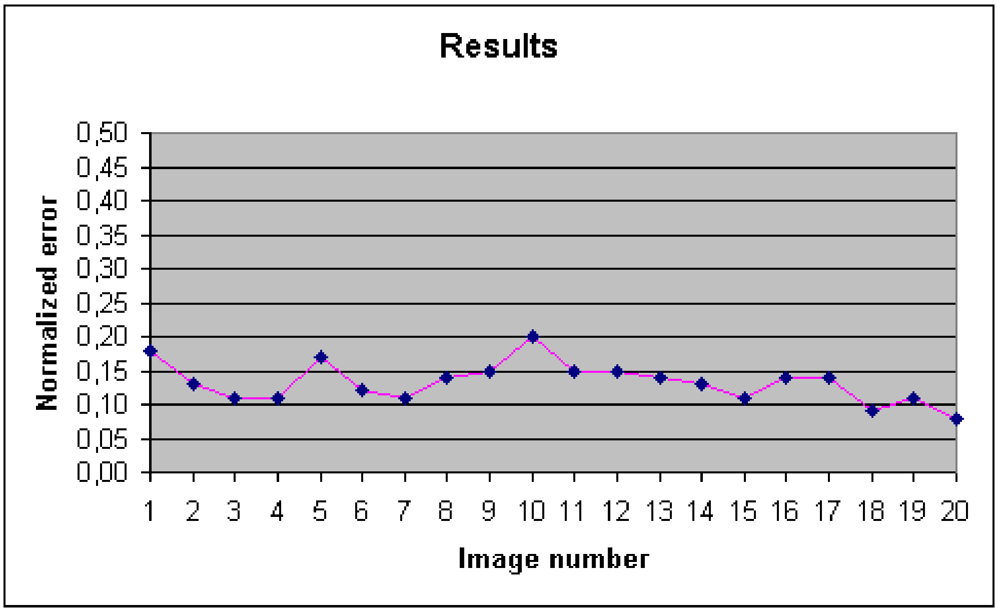

Fig. 13. Graph of results obtained

The competition-based approach will heavily reduce the influence of initialization on the final result (Zhu & Yuille, 1996).

In the following we outline the competition-based approach through a brief description of our algorithm.

At initialization time, *K* chains are generated and uniformly spread over the image. The number *K* depends on the size of the image and will usually be quite large to correctly segment the image independently of the initial position of the chains.

After the initialization, each chain grows in size until it meets another chain or a high edgeness contour. In the latter case, if the contour is closed and surrounds all the chain, the chain sticks to it, stopping its growth process and *breeds*, generating a new chain which will grow beyond the contour. This process allows the algorithm to detect multiple nested objects as *chain hierarchies*.

If the chain meets another chain, they start competing for the territory (i.e. an area in the image), and after a finite time a steady state will be reached. One of the two chains will probably "conquer" some part of the territory, until some line will be found, composed of pixel which are equidistant from both the chains. This line will be the border between the two chains.

At each step the pixel gray-level mean, a feature representative of all pixels surrounded by the chain, is estimated. The chains compete for the territory based on similarity between their “mean” and the gray-level of the point “to conquer”. It cannot happen that one of the chains totally defeats the other, since at least the original area of a chain (i.e. the one surrounded by the chain at initialization time) will always match better its “statistic”. But if two chains lay on the same, uniform, area then they will have the same statistic: in this case, as soon as they meet, they *merge*, thus producing a larger chain with the same statistic. The sequence of operations adopted to let the chains grow is shown in fig. 14.

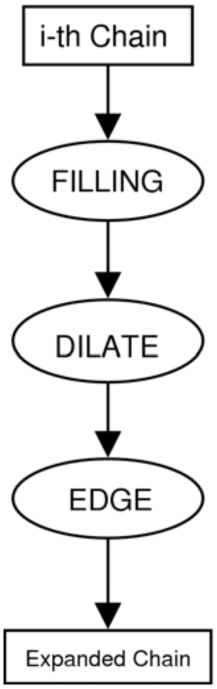

Fig. 14. Expansion phase

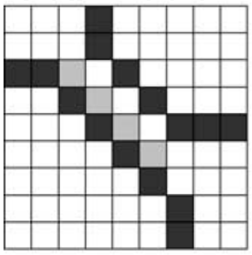

Fig. 15. Collision point detection

The detection of the collision point between chains is obtained using the approach proposed in (Vilarino et al., 2003).
An example of collision detection between two chains and merging chains, automatically handled by the algorithm, is shown in fig. 15.
Fig. 16 is a plotting of the error produced by our algorithm against the processed image. As the graph in Fig. 16 shows, the error on the test images is bounded around an average value of 0.18 in comparison with 0.2 obtained applying the same validation technique to the algorithm described in section 5.

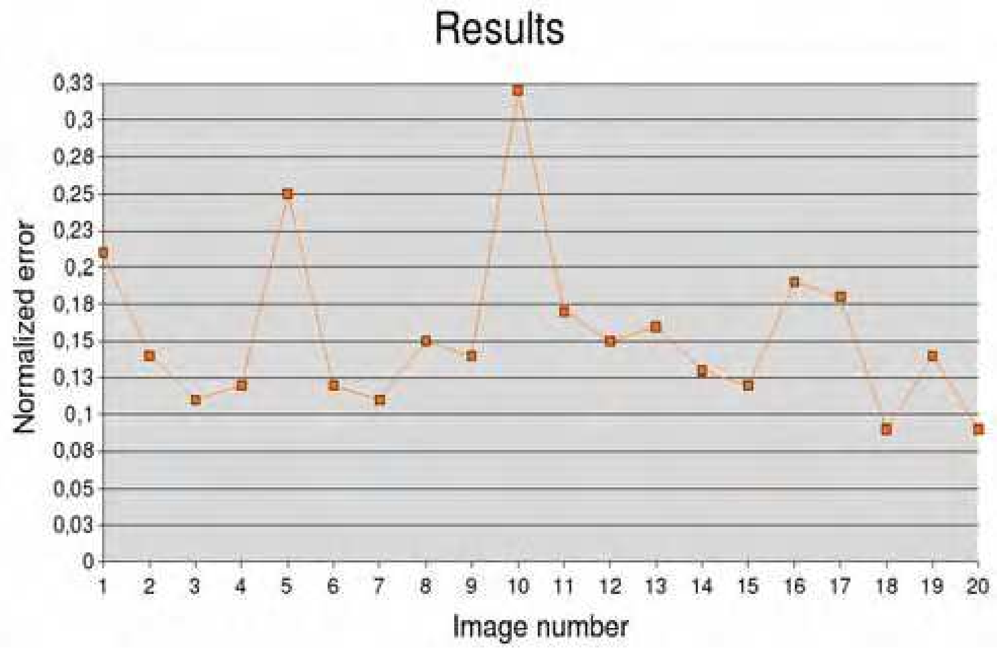

Fig. 16. Graph of results obtained

7. Conclusion

In this work we have described a re-formulation of a 2D still-image segmentation algorithm, implemented on a single-layer CNN, previously proposed (Iannizzotto, 2003). This algorithm is able to step-over limitation inherent to the class of active contours: sensitivity to insignificant false edges or “edge fragmentation”. The approach features an iterative process of uniform shrinking and deformation of the active contour. Guided by statistical properties of *edgeness* of the image pixels, the chain adapts itself to the image contours. Undesirable smoothing of the edges of the objects are prevented by the absence of any particular rigidity constraints on the chain. The technique used for uniform shrinking, which automatically handles any splitting, allows the presence of any nested edges to be detected.

Experimental measures of the accuracy of the segmentation were carried out using a technique based on Active Shape Models. Finally, an alternative competition-based approach, used to reduce some parameter dependences, is outlined in section 6.

8. References

Abe, T. & Matsukawa, Y. (2000). A region extraction method using multiple active contour models,*Proceedings of IEEE Conference Computer Vision and Pattern Recognition (CVPR2000)*

Chua, L. & Yang, L. (1988). Cellular neural networks: Theory, *IEEE Trans. on Circuits and Systems*, Vol. 35, No. 10, (1988), pp. 1257–1272

Chua, L. & Yang, L. (1988). Cellular neural networks: Applications, *IEEE Trans. on Circuits and Systems*, Vol. 35, No. 10, (1988), pp. 1273–1290

Chua, L. & Roska, T. (1993). The cnn universal machine: An analogic array computer, *IEEE Trans. Circuits and Systems II*, Vol. 40, (1993), pp. 163–173

Cohen, L. (1991). On active contour models and balloons, *CVGIP: Image Understanding*, Vol. 53, No. 2, (1991), pp. 211–218

Dice, L. R. (1945). Measures of the amount of ecologic association between species, *Ecology*, Vol. 26, No. 3, (1945), pp. 297–302

Flickner, M.; Hafner, J.; Rodriguez, E. J. & Sanz, J. L. C. (1996). Periodic quasi orthogonal spline bases and applications to least-squares curve fitting in digital images, *IEEE Transactions on Image Processing*, Vol. 5, No. 1, (Jan 1996), pp. 71–88

Frigui, H. & Krishnapuram, R. (1999). A robust competitive clustering al gorithm with applications in computer vision, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 21, No. 5, (May 1999),pp. 450–465

Huttenlocher, D.; Klanderman, G. & Rucklidge, A. (1993). Comparing images using the hausdorff distance, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 15, No. 9, (September 1993),pp. 850–863

Iannizzotto, G. & Vita, L. (1996). A fast accurate method to segment and retrieve object contours in real images, *Proceedings of International Conf on Image Processing*, Lausanne, Switzerland

Iannizzotto G. & Vita L. (2000). Fast and accurate edge-based segmentation with no contour smoothing in 2-d real images, *IEEE Trans. on Image Processing*, July 2000

Iannizzotto, G.; La Rosa, F.; Rizzo, A. & Xibilia, M. (2003). 2d still-image segmentation with cnn-amoeba, *Proceedings of International Work-shop on Computer Architectures for Machine Perception*, New Orleans

Jaccard, P. (1912). The distribution of flora in the alpine zone, *New Phytologist*, Vol. 11, pp. 37–50

Jones, T. & Metaxas, D. (1998). Image segmentation based on the integration of pixel affinity and deformable models, *Proceedings of CVPR98*, pp. 330–337

Kass, M.; Witkin, A. & Terzopoulos, D. (1988). Snakes: Active contour models, *International Journal of Computer Vision*, Vol. 1, pp. 321–333

Kozek, T. & Vilarino, D. L. (1999). An active contour algorithm for continuous-time cellular neural networks, *Journal of VLSI Signal Pro- cessing Systems*, Vol. 23, No. 2-3, (1999), pp. 403–414

Lai, K. & Chin, R. (1995). Deformable contours: Modeling and extraction, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 17, No. 11, pp. 1084–1090, November 1995.

Liu, Y. & Tang, Y. (1999). Adaptive image segmentation with distributed still-image segmentation algorithm, implemented on a single behaviour-based agents, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 21, No. 6, (June 1999), pp. 544–551

Malladi, R.; Sethian, J. A. & Vemuri, B. (1995). Shape modelling with front example of collision between two chains is shown. An example propagation, *IEEE Trans. on PAMI*, Vol. 17, No. 2, (February 1995)

Malladi, R. & Sethian, J. A. (1996). A unified approach to noise removal, image enhancement,and shape recovery, *IEEE Trans. on Image Processing*, Vol. 5, No. 11, (November 1996)

Manganaro, G.; Arena, P. & L. Fortuna (1999). *Cellular Neural Networks: Chaos, Complexity and VLSI processing*, Springer-Verlag

Matsumoto, T.; Yokohama, T.; Suzuki, H. & Furukawa, R. (1990). Several image processing examples by cnn, *Proceedings of IEEE Int. Workshop on Cellular Neural Networks and Their Applications*, Budapest

McInerney, T. J. (1997). Topologically Adaptable Deformable Models. *Ph.D Thesis*, University of Toronto

Milanova, M. & Buker, U. (2000). Object recognition in image sequences with cellular neural networks, *Neurocomputing*, Vol. 31, No. 1-4, pp. 125–141

Moreira-Tamayos, O. & Gyvez, J. P. D. (1999). Subband coding and image compression using cnn, *International Journal of Circuit Theory and Applications*, No. 27, pp. 135–151

C. Rekeczky (1999). Active contour and skeleton models in continous-time cnn, *Proceedings of ECCTD '99*, Stresa, Italy

Szirànyi, T.; & Csicsvàri, J. (1993). High-speed character recognition using a dual cellular neural network architecture (cnnd), Analog and Digital Signal Processing, Vol. 40, No. 3, pp. 223–231

Theodoridis, S. & Kotroumbas, K. (1999). *Pattern Recognition*. Academic Press

Vilarino, D. L.; Cabello, D.; Pardo, X. M. & Brea, V. M. (2003). Cellular neural networks and active contours: a tool for image segmentation, *Image Vision Computing*, Vol. 21, No. 2, pp. 189–204

Zhu, S. & Yuille, A. (1996). Region competition: Unifying snakes, region growing, and bayes/mdl for multiband image segmentation, *IEEE Transactions on Pattern Analysis and Machine Intelligence*,Vol.18, No. 9, (September 1996), pp. 884–900

Vision Systems: Segmentation and Pattern Recognition
Edited by Goro Obinata and Ashish Dutta

ISBN 978-3-902613-05-9
Hard cover, 536 pages
Publisher I-Tech Education and Publishing
Published online 01, June, 2007
Published in print edition June, 2007

Research in computer vision has exponentially increased in the last two decades due to the availability of cheap cameras and fast processors. This increase has also been accompanied by a blurring of the boundaries between the different applications of vision, making it truly interdisciplinary. In this book we have attempted to put together state-of-the-art research and developments in segmentation and pattern recognition. The first nine chapters on segmentation deal with advanced algorithms and models, and various applications of segmentation in robot path planning, human face tracking, etc. The later chapters are devoted to pattern recognition and covers diverse topics ranging from biological image analysis, remote sensing, text recognition, advanced filter design for data analysis, etc.

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821