

Self-Training Statistic Snake for Image Segmentation and Tracking

X.M. Pardo
Dept. Electrónica e Computación
Universidade de Santiago de Compostela
15706 Santiago, Spain.
pardo@dec.usc.es

P. Radeva, J.J. Villanueva
Computer Vision Centre
Universitat Autònoma de Barcelona
089193 Bellaterra (Barcelona), Spain.
{petia, villanueva}@cvc.uab.es

Abstract

In this work we propose a new supervised deformable model that generalizes the classical contour-based snake. This model is defined to deform in a feature space generated by a set of Gaussian derivative filter responses. The snake selects and classifies image features by a parametric vector that gives the direction in the feature space minimizing the dissimilarity between the learned and found image features and maximizing the distance between different contour configurations. Each snake curve patch is devoted to search for a special contour configuration. The classes corresponding to different contour configurations are obtained by means of a statistic supervised learning technique using samples of different contour and no contour points. The snake starts with a large set of Gaussian filters that is reduced by means of principal component analysis in a supervised way to optimize it in the feature search.

1. Introduction

Deformable models have received the attention of many researchers in the last decade [10, 1, 3, 8, 6] due to their ability to interpret sparse set of features and link them to obtain object's contours under general assumptions about the contour shape. The classical snake model is a contour-based technique that uses energy terms defined from gradient features to segment image objects [5]. The snake deforms attracted by high gradient image locations meanwhile keeps its smooth shape. When object's contours are manifested by high gradient response, the segmentation task is performed with high rate of success. Still there are many applications where such an assumption is not fulfilled leading to a loss of robustness of the method. Different authors suggest to combine the gradient potential with valley and crest maps [13, 2]. Nevertheless, the selection of potentials and weights is still heuristic and the contribution of different potentials is constant. These characteristics prevent the

snakes from using in automatic systems for real complex applications or lead to a need of an heuristic design to use external energies depending of the application [13].

We propose to generalize in a double sense the classical potential that determines the image interpretation by the snake. First, we do an horizontal generalization by including different Gaussian filters and a vertical generalization by using different scales. The snake learns itself the image features that best represent the object's contour. In particular, the user is freed to determine whether the snake should be attracted by edge, crest and/or valley points. Second, we define a parametric external energy to enhance different features depending of the snake location in the scene. Snake vertices are not going to look for the same features, but each curve patch is guided by a multivalued potential that drives it towards particular features characterizing a certain piece of contour. The potential is constant and multivalued generated by a set of filters applied to the image but, each snake patch interprets it in different way due to the parametric classifier embodied into the snake external energy.

In our approach, a supervised learning in conjunction with a principal component analysis (PCA) is carried out to characterize each contour patch. The statistic snake approach is particularly of interest for segmenting and tracking volumetric or temporal image sequences. There, a user-provided contour is necessary to the purpose of the supervised learning only in the first image. A parametric snake classifier for each image is obtained from the snake converged to the object's contour in the previous image of the sequence. Therefore, the snake is able to learn the changes in contour features inside each image as well as along the sequence in an adaptive way. This property gives the snake robustness and fast convergence.

The paper is organized as follows: in section 2 we give a brief background on snakes. In section 3 we introduce our statistic snake based on PCA and Fisher linear discriminant analysis (FLDA). In section 4 we give the results of applying our snake model to the problem of femur segmentation in CT images and finish the article by conclusions.

2. Snake formulation

A snake is an elastic curve that evolves from its initial shape and position as a result of the combined action of external and internal forces [5]. The external forces push the snake towards features of the image, whereas internal forces model the elasticity of the curve. In a parametric representation, the snake appears as a curve $u(\lambda) = (x(\lambda), y(\lambda))$, λ is the internal parameter of the curve. Its internal energy is defined as follows [5]:

$$E_{int}(u) = \alpha \int |u_\lambda|^2 d\lambda + \beta \int |u_{\lambda\lambda}|^2 d\lambda.$$

The first term weights the resistance to stretching and the second one weights the resistance to bending controlled by the elastic parameters α and β . The external energy is generally defined from a potential field P :

$$E_{ext}(u) = \int P(u(\lambda)) d\lambda.$$

A typical potential is:

$$P(u(\lambda)) = \pm\gamma_1 I(u(\lambda)) - \gamma_2 \nabla(G_\sigma(u(\lambda)) * I(u(\lambda))) \quad (1)$$

where $I(u)$ represents the intensity value and $G_\sigma(u)$ is a Gaussian smoothing function of size σ . This potential pushes the snake curve towards high gradient points of high/low intensity values. The total energy of the snake is the sum of the external and internal energies:

$$E_{snake} = \int [E_{int}(u(\lambda)) + E_{ext}(u(\lambda))] d\lambda$$

The solution to the problem of detecting the contour is found in the minimization of this energy function, which is generally performed using variational principles [5, 1].

3. Supervised feature learning and principal component analysis

To drive a snake towards the contour of interest, each snaxel must be able to distinguish between its corresponding contour patch target and other contours. To this aim, instead of using (1) we construct a feature space applying a set of filters to the image that yields the relevant features to characterize all the contour configurations and we separate different contour patches in this space. We accomplish a supervised learning aimed at the maximization of the between-class scattering while minimizing the hyper-space dimension determined by the feature categories.

Initially, we consider a large set (50) of Gaussian derivative filters [9] that contains derivatives up to degree three and scales to a number sufficient of characterizing all possible configurations (in our case, 5). The application of all

these filters has large requirements of storage and time; it seems natural to perform a reduction of the filter space. We apply a technique commonly used for dimensionality reduction based on PCA [11, 12].

Given the problem of femur segmentation in CT images, a set of N sample image feature vectors of bone and non bone $\{s_1, \dots, s_N\}$ are chosen, taking values in an n -dimensional space. Each component corresponds to the response of a filter applied to the original image. Looking for a certain contour k , each snaxel j can be classified to one of two classes $\{c_k, \bar{c}_k\}$, representing the pixels belonging to the contour and viceversa. Our first goal is to obtain a linear transformation that maps the original n -dimensional space into an 1-dimensional feature space where the classification is applied. Let us consider that the new feature scalar $r_j \in \mathcal{R}$ is defined by the linear transformation $r_j = V_k^T x_j$, $j = 1, \dots, N$, where $V_k \in \mathcal{R}^n$ is a vector that projects the pixel x_j into the feature space to classify pixels belonging to the contour k . In order to find an optimal projection into a feature space where the distance between samples of class c_k and the remainder samples is maximized, we apply the FLDA [4].

As described in [7], the optimal projection V_{k_opt} can be defined as the vector which maximizes the ratio of the determinant of the between-class scatter matrix, S_B , to the within-class scatter matrix, S_W , of the projected samples:

$$V_{k_opt} = \arg \max_{V_k} \frac{|V_k^T S_B V_k|}{|V_k^T S_W V_k|}, \quad (2)$$

$$S_B = N_{c_k} (\mu_{c_k} - \mu)(\mu_{c_k} - \mu)^t + N_{\bar{c}_k} (\mu_{\bar{c}_k} - \mu)(\mu_{\bar{c}_k} - \mu)^t$$

$$S_W = \sum_{x_j \in c_k} (\mu_{c_k} - x_j)^t (\mu_{c_k} - x_j) + \sum_{x_l \in \bar{c}_k} (\mu_{\bar{c}_k} - x_l)^t (\mu_{\bar{c}_k} - x_l)$$

being N_C the number of samples in class C , μ_C its mean feature vector, and μ the mean vector over all samples. V_{k_opt} is the S_W -generalized eigenvector of S_B with the largest corresponding eigenvalue [7].

Using PCA allows to project the initial large n -dimensional hyperspace in a reduced m -hyperspace. We found that this reduction is crucial in order to eliminate spurious potential local minima. As a rule, we only maintain eigenvectors with largest eigenvalue that retain the 90% of the sample variance. Besides reducing the dimensionality, PCA is used to avoid the existence of a singular S_W [7]. As a result of performing PCA, we get:

$$V_{k_opt}^t = V_{k_fld}^t V_{pca}^t,$$

$$V_{pca} = \arg \max_{V_k} |V_k^t S_T V_k|, \quad S_T = \sum_{i=1}^N (x_i - \mu)(x_i - \mu)^t,$$

$$V_{k_fld} = \arg \max_{V_k} \frac{|V_k^t V_{pca}^t S_B V_{pca} V_k|}{|V_k^t V_{pca}^t S_W V_{pca} V_k|},$$

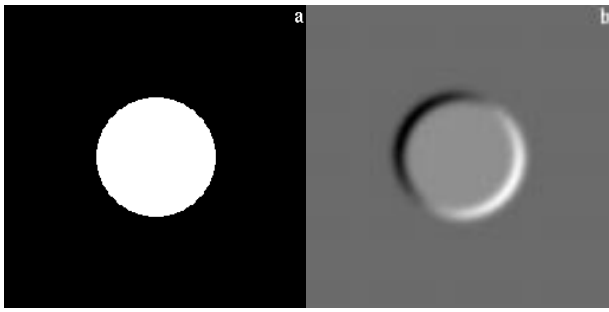


Figure 1. Original image (a); distance map to a contour class in the North-West (N-W) of the circle (b).

V_{pca} is an $n \times m$ matrix formed by the m most important eigenvectors of S_T and V_{k_fld} is a vector of size m formed by the most important S_W -generalized eigenvector of S_B .

Each patch of the curve has its own classifier that defines the bone features it is looking for. The scalar product of the mean feature vector of the patch and its classifier vector gives the projection into the feature space. We define a snake potential by constructing the distance map between the projection of the image feature vector and the class center. Each target contour part generates its own distance map, leading to a parametric potential where each snaxel interprets in different way the image filter responses depending of the goal it carries (the contour type it has learned in its previous stage). Note that the distance map is not necessary to be computed for all image pixels but just around the snake that fastens the snake deformation.

In the synthetic example of Fig. 1(a), we select sample points from the border of the circle and use a single scale ($\sigma = 4$) and zero- and first-order Gaussian derivatives. Each contour part is associated to a patch of a spline curve of the snake and several samples in its neighbourhood are used to train the snake. Samples of the rest of the spline and image background form the complementary class. Fig. 1(b) shows the distance map to a potential minimum determined by a patch in the N-W direction of the contour. Each pixel from the potential is assigned the distance between the projections of the image feature vectors and the class center of the samples representing the N-W object's border. Naturally, maximum distance is in the opposite direction; that is very useful to distinguish contours of close objects.

Fig. 2 illustrates a real example. Fig. 2(a) shows the contour where bone samples are taken; some muscle samples are added to strengthen the counterexample training. Figs. 2(b,c) represent the distance maps of the classes of patches placed on the west and on the east, respectively. Figs. 2(d,e) show the same effect for other two opposite di-

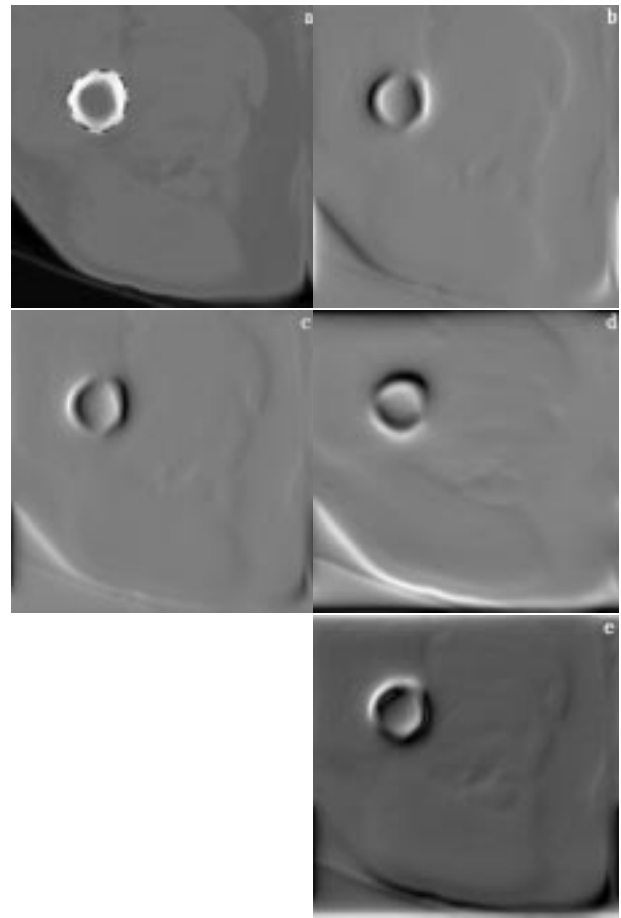


Figure 2. Original image showing the different patches (a); distance map of a contour class of west, east, north and south patches (b)-(e).

rections. As it is desirable, the minimum of the distance map is around each learned contour class and there is a maximum in the opposite direction.

4. Results

As explained above, the goal of this work is the generalization of the classical contour-based snake integrating gradient and texture features obtained by different scales. The snake external energy is parametric in sense that the salient image features looked for, depend on the location of the curve. In our approach each snake patch searches for its specific contour class. This makes the technique more robust and avoids ambiguities in case of similar objects close each other. Fig. 3 illustrates this capability. Fig. 3(a) contains the original image with the object's contour where different types of external cortical bones are learned. Fig. 3(b)

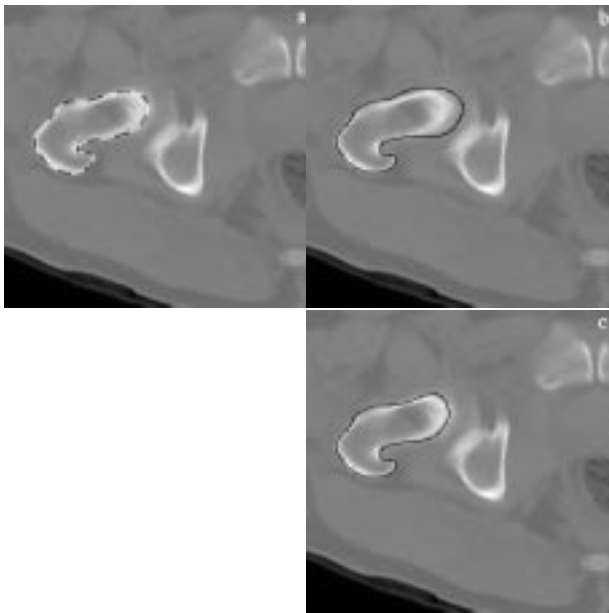


Figure 3. Learning contour (a); initial contour (b) and final delineation of the bone (c).

represents an initial contour placed between two bones and Fig. 3(c) represents the final contour fitted to the bone of interest. Right and left hand of cortical bone have different filter responses, hence they belong to different classes and the snake moves towards the desired contour.

Fig. 4 shows the delineations obtained from the learned contour of Fig. 2(a) using different initializations. In order to make the evolution from distant contours feasible it is necessary to use large scales of the Gaussian filters. For example, the initial contour shown in Fig. 4(d) reaches the cortical bone when using scale with values 1,2,4,8,16 however, using values 1,2,4,8 the snake converges before reaching the contour, because some contour patches have not information about the direction of the evolution.

In the example of Fig. 5, the classifiers are learned in a slice and the final contour is carried to the next slice. Fig. 5(a) contains the final contour obtained from slice 37; here the classifiers are updated. Fig. 5(b) contains the slice 38 with the final contour obtained from slice 37 superposed. Finally, Fig. 5(c) shows the contour obtained after the evolution guided by the classifiers. Afterwards, the classifiers are again updated and transferred together with the final contour to the next slice.

Fig. 6 shows graphical representations of absolute values of the classification vectors V_{k_opt} assigned to each snake patch from Fig. 5. Note that the classification vector determines the weights of different filters i.e. filters non important for a contour part will have weights close to 0 in

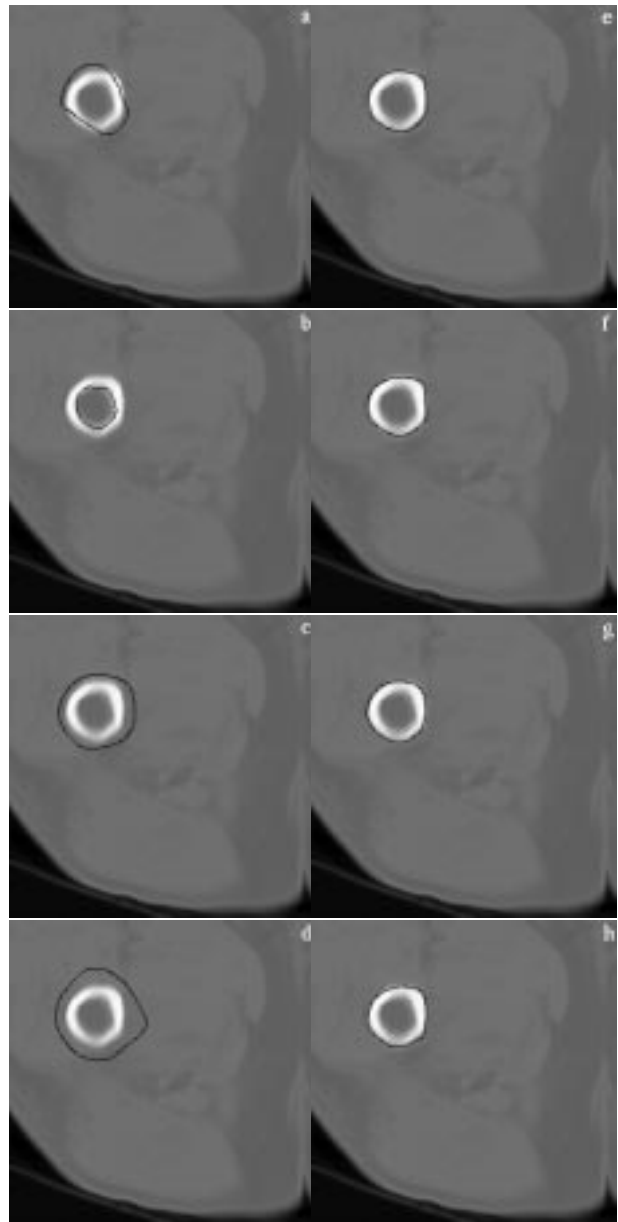


Figure 4. Initial contours (a)-(d); final delineations of the bone (e)-(h).

the classification vector. Fig. 6(a) shows the graphical representation of the classification vectors for the contour in Fig. 5(a). First patch is located on the right (clockwise) of the point more on the left part of the contour, and the remainder patches are ordered clockwise. The first patches have less variance in the contribution of the filters (filter weights), but variance increases with the patch number. The reason is that the contour is better defined in the last part.

Fig. 6(b) contains the classification vector representation

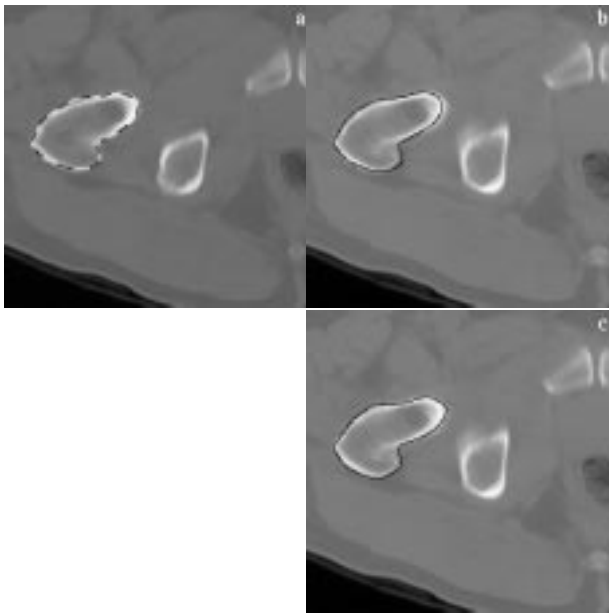


Figure 5. Learning contour in slice 37 (a); initial (b) and final contour in slice 38 (c).

for the adjacent slice, Fig. 5(c). Now the location of the contour is worst defined in average, hence the contribution of the different filters has less variance. Filter weights significantly change in parts where the contour features (bone morphology) change as shown in Fig. 6(c). This occurs in three parts of the bone: first, around patch in the top-middle part of the bone, where a protuberance is emerging; second, around patch in the top right where a strong contour begins to blur; and third, around patch on the bottom-right where a concavity begins to form. The changes in contour features result in changes in classification vectors, Fig. 6(c) shows the absolute values of filter weight differences as a function of patches and filters. As one can see, the snake updates the classifiers of the features it is looking for.

Fig. 7 shows the segmentations of a slice sequence (35-40) where a strong change in contour shape takes place. The supervised learning took placed in the first slice, and final contours and the updating of classifiers were propagated from each slice to the next. In this way, the features of each class are updated in a supervised mode and each patch searches for specific features discriminating between different objects and contour patches. Fig. 8 shows the stacking contours of these sequence.

5. Conclusions

In this work we developed a new deformable model that generalizes the contour-based snake. Given the fact that ob-

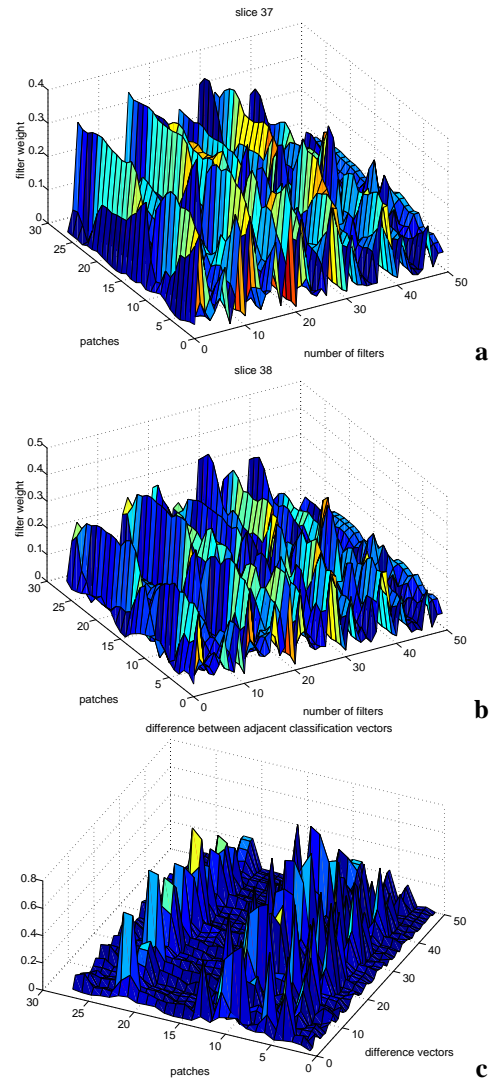


Figure 6. Filter weights along the contour for two adjacent slices: slice 37 (a), slice 38 (b) and their evolution between slices (c).

ject's contour is rarely well defined by high gradient features we integrate gradient and texture features obtained by a set of filters with different scales. In this way, the user is free of designing the external energy, the snake selects and self-trains the image features that are best representing the object's contour. As a result, our model is a natural generalization of contour-, valley- and crest-based snakes.

The new snake applies statistic methods (PCA and FLDA) to learn different contour configurations. A parametric external energy is designed according to the fact that different patches of the snake are looking for different parts of the contour. As a result, we achieved a snake technique

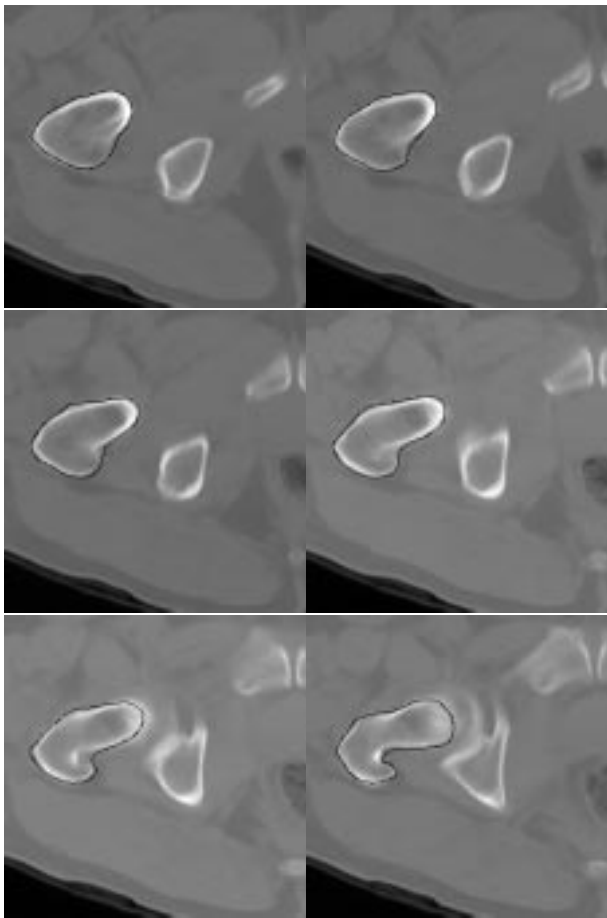


Figure 7. CT images of slices 35-40.

that is more selective and robust and less dependent of near objects. We apply our snake to segment the femur in CT images and obtain that statistic snakes with embodied classifiers to the image features give better results than original snakes even in case of presence of near bones to the femur and changes in the bone morphology.

When the initial contour is distant (in location or features) from the desired contour, neighbour patches could be trapped by the same contour class. As a consequence, some patches are very large and others are very small, so the updating of classifiers are prone to error. This is the main limitation of the method. An *ad hoc* solution was obtained for CT bone segmentation, although, our actual efforts are directed to obtain a context free solution based on a major comprehension of the classifier behaviour.

Acknowledgments

This work is supported by CICYT and EU grants TIC98-1100, 2FD97-0220 and 1FD97-0667. The authors wish

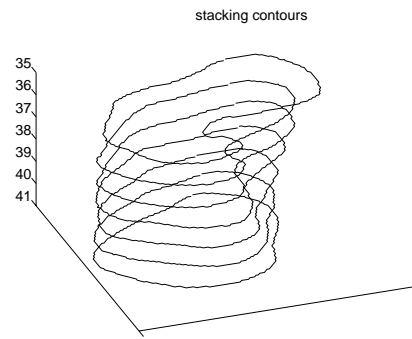


Figure 8. Stacking contours from Fig. 7.

to thank I.J. Pulido and F. Seron from the University of Zaragoza for supplying the femur CT images.

References

- [1] A. A. Amini, T. E. Weymouth, and R. C. Jain. Using dynamic programming for solving variational problems in vision. *IEEE Trans. Patt. Anal. Machine. Intell.*, 12(9):855–867, 1990.
- [2] A. Blake and Isard. *Active contours*. Springer-Verlag, 1998.
- [3] L. D. Cohen and I. Cohen. Finite-element methods for active contour models and balloons for 2-D and 3-D images. *IEEE Trans. Patt. Anal. Machine. Intell.*, 15(11):1131–1147, November 1993.
- [4] R. O. Duda and P. E. Hurt. *Pattern Classification and Scene Analysis*. Wiley-Interscience, New York, 1973.
- [5] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: active contour models. *International Journal of Computer Vision*, 1:321–331, 1988.
- [6] T. McInerney and D. Terzopoulos. Deformable models in medical image analysis: a survey. *Medial Image Analysis*, 1(2):91–108, 1996.
- [7] J. H. P.N. Belhumeur and D. Kriegman. Eigenfaces vs. fisherfaces: recognition using class specific linear projection. *IEEE Trans. Patt. Anal. Machine. Intell.*, 7:711–720, 1997.
- [8] P. Radeva, J. Serrat, and E. Marti. A snake for model-based segmentation. In *Proceedings of International Conference on Computer Vision (ICCV'95)*, MIT, USA, 1995.
- [9] R. Rao and D. Ballard. An active vision architecture based on iconic representations. *Artificial Intelligence*, 78:461–505, 1995.
- [10] D. Terzopoulos. On matching deformable models to images: direct and iterative solutions. In *Topical Meeting on Machine Vision*, volume 12 of *Technical Digest*, pages 160–167. Optical Society of America, 1987.
- [11] M. Turk and A. Pentland. Eigenfaces for recognition. *J. Cognitive Neuroscience*, 3(1), 1991.
- [12] M. Turk and A. Pentland. Face recognition using eigenfaces. In *Proc. IEE Conf. on Computer Vision and Pattern Recognition*, pages 586–591, 1991.
- [13] A. Yuille, P. Hallinan, and D. Cohen. Feature extraction from faces using deformable templates. *IJCV*, 8(2), 1992.