TOWARDS CONTINUOUS IMAGE REPRESENTATIONS

Frédéric Labrosse
Department of Computer Science
University of Wales, Aberystwyth
United Kingdom
ffl@aber.ac.uk

Philip Willis
Department of Mathematical Sciences
University of Bath
United Kingdom
P.J.Willis@bath.ac.uk

ABSTRACT

We propose in this paper a first step towards the creation of continuous, *i.e.* vectorial, representations that are useful for image manipulation. Such pixel-free representations have many advantages and are amenable to operations which are difficult or imprecise with pixels. For example, they can readily be rendered at different resolutions and they are a better choice for ultra-high resolution applications. We explore an approach in which images are decomposed into structural regions that correspond to specified image characteristics. This is done using relaxation labelling. Information taken at different stages during the relaxation is used to extract sub-pixel accurate continuous *structural contours*. This accuracy is obtained by using snakes as well as the blur present in images (because of the acquisition process). We propose solutions, adapted to our context, to often mentioned problems of snakes, namely initialisation, parameter determination, and instability. The interior of structural regions is represented to allow the rendering of images as close as possible to the original ones. We propose here two schemes, one using a single colour for each region, the second sampling the original image to allow smoothly varying colour in each region.

Keywords: Continuous contour, Snake, Sub-pixel accuracy, Image representation

1 INTRODUCTION

When film special effects are created at the post-production stage, they are often performed on digitised versions of the movie frames. By "digitised", we mean "turned into pixels." The required optical quality means that each frame must be digitised at high resolution, when the large number of frames involved combines to ensure that a great deal of processing is required. Moreover, digitisation introduces its own problems. For example, one basic problem is *How do you remove an object when its boundary spreads across several pixels in width because of the blur introduced by the acquisition system?*

We claim that special effects should be performed using vectorial and sub-pixel representations of the images. Indeed, such representations are easy to transform or merge (the problems of pixel alignment or differences in image resolution are not present [Froum00]) and can be rendered efficiently at any resolution [Froum99] or with continuous zoom, without special treatment to preserve image events that must not be altered (*e.g.* boundaries should remain sharp even when increasing the resolution, which is not easy when using traditional interpolation techniques). Moreover, such representations are more robust against object rotations and/or translations between consecutive images[1]. Sub-pixel information is present in images in the form of blur introduced either by the acquisition process or by the anti-aliasing of synthetic images. Using that information, we obtain as accurate a representation as the source data will allow.

We first decompose images into regions that correspond to key parts of the images. This decomposition is based on statistical properties: a region will be a pixel group having a given average and variance of colour. This will allow a *structural* decomposition of the image based on colours and will lead to structural contours. We then represent the interior of the structural re-

[1] The representation of an object should not change when built from two different images where the object is present but rotated or translated, which is often the case in a movie.

gions in order to allow image synthesis as close as possible to the original image. We propose two schemes, one using a single colour for each region, the second sampling the original image to allow smoothly varying colour in each region.

We completely represent the image with a region-based scheme requiring very limited and rough human intervention. Other methods have to specify boundaries precisely, specify interesting contours, group sparse edges, and/or only extract one object from the image [Kass88, Morte95, Elder98]. Moreover, only a few methods explicitly handle sub-pixel accuracy and even fewer methods completely get rid of the pixel nature of the data. We emphasise in our method the sub-pixel accuracy at all stages of the contour extraction. Structural contours are obtained using improved snakes [Kass88] to get better sub-pixel behaviour[2]. Our main contribution is that we combine and improve known techniques to build a fully continuous and sub-pixel representation of the image thus allowing an aliasing-free manipulation and synthesis.

In section 2, we will show how structural contours are extracted while Section 3 will describe how we represent the interior of structural regions. Finally, Section 4 will show some results.

2 STRUCTURAL CONTOURS

The image decomposition into structural regions proceeds in two stages. The first is image segmentation into homogeneous regions. The second is structural contours extraction. The latter stage uses information produced at different steps of the former stage.

2.1 Segmenting an Image into Homogeneous Regions

To segment an image, we use relaxation labelling [Rosen76, Humme83], which builds a mapping from a set of objects to a set of labels by propagating only local rules. In the case of image segmentation, objects are pixels denoted by their coordinates (x, y) and labels correspond to sets of region attributes a_i.

The relaxation labelling produces for each possible attribute set an image whose pixels are white (or very light) if they belong to a region having this attribute set, black (or very dark) if not, and with a grey transition from one value to the other on the region boundary. That transition is narrower than the blur in the original image because of the convergence properties of the relaxation labelling, which will remove small regions and sharpen the transitions between regions.

Following [Garba86, Hanse97], we choose as region attributes the average colour μ and the colour variance σ. The attribute set for the region i is thus $a_i = \{\mu_i, \sigma_i\}$. Typical attribute sets must be specified before the relaxation process begins. This is done interactively by selecting parts in the image to be segmented that are characteristic of each region. The interactive scheme to specify regions is acceptable in our application because we want to analyse image sequences and because the region attributes usually do not change dramatically between two consecutive images. This is the only required human intervention.

The region attributes are computed in the CIE $L^*a^*b^*$ colour space because of its perceptual uniformity [Wysze82, Henri98, Meyer87]. In image segmentation it is common to disregard luminance because changes in luminance do not usually represent object boundaries but rather changes in lighting conditions (clouds, shadows, lights, *etc.*). In our case, we are interested in regions having homogeneous colour properties, not in regions corresponding to physical evidence. We thus consider all three coordinates of the CIE $L^*a^*b^*$ space.

Given these region attributes, we use the Mahalanobis distance to measure the distance between the value I of a pixel at position (x, y) and the region whose attribute set is a_i:

$$d_{(x,y)}(a_i) = \frac{(I(x,y) - \mu_i)^2}{\sigma_i^2}.$$

This distance is used to compute the initial probability of each pixel being associated with each attribute set:

$$p^{(0)}_{(x,y)}(a_i) = \frac{d_{(x,y)}(a_i)^{-1}}{\sum_{j=1}^{m} d_{(x,y)}(a_j)^{-1}},$$

where m is the number of regions. Neighbouring pixels, in an 8-pixels neighbourhood, are compatible if they are associated with the same label. If labels are different, pixels are neither compatible, nor incompatible. Probabilities are updated as in [Rosen76].

We stop the relaxation when the labelling does not change anymore. Note that an early stop will usually produce a great number of small regions. In contrast, a late stop will remove all these small

[2] Snakes are, since their origin, sub-pixel. But this was a side effect of the optimisation method and the convergence stability of the original optimisation method was making sub-pixel accuracy questionable. New methods have been proposed that are not sub-pixel [Geige95, Willi92].

regions but will also remove many details in the images, in particular, it will round the corners. This latter point is not really a problem since the result of the labelling is only a starting point for the sub-pixel extraction (snake potential fields are made from probabilities well before the relaxation ends, as we will now describe). Moreover, some image details could be preserved as in [Richa81].

2.2 Sub-pixel Structural Contours

Snakes allow the extraction of linear events in images. A snake, parametrically represented by $\mathbf{v}(s) = (x(s), y(s))$, has an energy that measures how far from an ideal model the snake is:

$$E = \int E_i(s)ds + \int E_e(\mathbf{v}(s))ds, \qquad (1)$$

where $E_i(s)$ and $E_e(\mathbf{v}(s))$ are respectively the internal and external energy at the curvilinear abscissa s.

The internal energy consists of a second-order (rigidity) term[3]:

$$E_i(s) = \frac{1}{2}\beta(s)\left|\frac{d^2\mathbf{v}(s)}{ds^2}\right|^2, \qquad (2)$$

where $\beta(s)$ controls the importance of the rigidity at the corresponding snake point in the total energy E.

The external energy is the value of a potential field at the corresponding snake point. The field can take into account different kinds of sources: springs, repulsors, and lines, edges, or line terminations detectors [Kass88].

To make the snake fit its ideal model, its energy must be minimised. For this end, the original method, primarily used in [Kass88] uses a variational approach. Basically, (1) can be transformed into:

$$\mathbf{A}\mathbf{v}_t + \left.\frac{\partial E_e}{\partial \mathbf{v}}\right|_{\mathbf{v}=\mathbf{v}_{t-1}} = -\gamma(\mathbf{v}_t - \mathbf{v}_{t-1}) \qquad (3)$$

obtained by introducing a damping over time [Terzo87], by linear abscissa discretisation (which decomposes the snake into control points), and with time derivatives approximated by finite differences. $\mathbf{v}_t$ is a control point at time t, $\mathbf{A}$ is a penta-diagonal banded matrix depending only upon βs, and γ is the damping factor. Because of the properties of $\mathbf{A}$, we can efficiently solve (3).

[3] In [Kass88], a first-order (continuity) term is also used, which tends to shrink the snake, which will make it go inside the true sub-pixel boundary. We have moreover noticed that this term is not useful if the snake is already continuous, which is the case here.

Since we want sub-pixel snakes, we use the minimisation proposed in [Kass88] and not other schemes proposed, *e.g.*, by [Geige95, Willi92] that overcome instabilities and convergence problems but give a solution at the pixel level. One major problem with the method used is that the snake must initially be close to its final position. We thus use the regions produced by the segmentation (Section 2.1) to initialise the snakes. The boundary (at the pixel resolution) of each region will constitute a snake at its initial position. Then, the minimisation of the snake's energy will produce the sub-pixel boundary.

The potential field is made from an edge image that is the gradient modulus of an image I' built from the original image. The image I' is bright in the region whose boundary has made the snake, dark outside of that region, and has a smooth transition between the bright and dark regions that reflects the image blur. It is made of the probabilities at some point during the relaxation. We do not use the final probabilities because they are too sharp and the edges in these do not represent correctly the edges in the original image. On the other hand, the initial probabilities lead to edges that are not well defined (as with the original image). Moreover, it is important to have the same region topology for the initial snakes (taken from the region boundaries at the end of the relaxation) and the edges in the potential fields. Indeed, if they are different, then the snakes will be initially far from their final position and holes can remain between regions (where small regions have disappeared). When the number of labelling changes gets small (with respect to the labelling changes after the first iteration), then changes become very small in terms of regions and we can use the probabilities at that stage to extract the edges to create the potential fields. Moreover, probabilities at that stage are still smooth. Typically, we use the probabilities when the labelling changes cross the threshold of one tenth of the first labelling change. Figure 1 shows an image of a goose and the image I' corresponding to the dark parts. The edge image is then interpolated using bicubic Bézier patches to get a smooth analytic surface which makes the potential field. The term $E_e(\mathbf{v}(s))$ in (1) becomes:

$$\begin{aligned} E_e(\mathbf{v}(s)) &= E_e(x(s), y(s)) \qquad (4)\\ &= -\mathrm{Interp}(|\nabla I'(x(s), y(s))|^2). \end{aligned}$$

In the original implementation of the snakes [Kass88], all parameters are manually set. We propose an automatic way of choosing them. The β parameters in (2) are set to allow angles at some control points in the snake. They are first initialised to 1 for each control point. Then, control

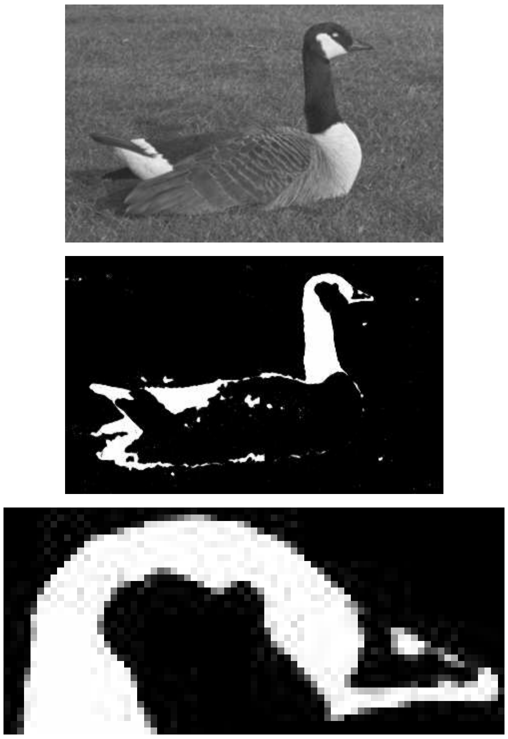

Figure 1: An image of a goose, the image I' for the dark regions, and a close-up of I'

points making angles are sought along the initial snake configuration[4]. At these control points, β is set to the negated, then clamped to 0, cosine of the angle. Finally, all βs are normalised so that the amplitudes of the snake rigidity and external energy are equal. The rigidity is assumed to lie between its initial value and 0 (all control points are roughly aligned at the end, except where a discontinuity has been allowed, in which case the contribution is small). The external energy is assumed to lie between its initial value and the value found by adding the minimum contribution of each control point in its 3×3 neighbourhood (with increments of 0.5 pixels). This latter value is close to the minimum value since the snake control points are initially at less than one pixel from their final position.

The parameter γ in (3) is automatically and dynamically determined. It is initially computed so that a given average displacement (typically a quarter of a pixel) of the control points is allowed. Basically, (3) gives n vectorial equations from which we can extract n values for γ given the required average step-size, where n is the number of control points. The initial value is taken as the average of these n values. Then γ is increased (the step-size reduced) until the snake becomes still. The increase happens when the energy's derivative (averaged over time) falls below a given threshold. Using this method, the energy ends up at a slightly higher value than with a fixed γ, but we have seen no lower quality of the result. The result is insensitive to the amount of increase of the damping factor as well as to the value of the energy derivative threshold.

Though inspired by [Fua90], our approach is different because our context is different. For example, we cannot assume that the initial estimate of the snake is close to the final answer (even if this is true in terms of distance) since this is exactly that difference that makes the contour reaching sub-pixel accuracy.

When the snakes have converged, we use their control points as control points of interpolating curves, *e.g.* NURBS curves, to produce the structural contours.

3 STRUCTURAL REGIONS

Once the structural contours are extracted, we need to represent their interior. Two basic schemes are proposed here. Others (based on texture representation and isochromatic contours) are currently under investigation. More details can be found in [Labro00].

The first scheme (*flat colours*) is particularly appropriate for "simplified" rendering for non-photorealistic applications such as cartoon-like rendering or technical illustration. A unique colour is associated with the region: the average colour of the corresponding image part that was selected by the user.

The second scheme is more suited to realistic rendering where continuous varying colours are needed: *smooth colours* are computed inside the regions from the original image. Each region is triangulated using a quality conforming Delaunay triangulation, where the area and angles of the triangles can be controlled [Shewc96]. A colour is then associated with each vertex of the triangulation, as a function of the colour of the nearest pixel in the original image. This function can be described as follows. The average and variance of the colour specified by the user define an ellipsoid in the colour space. The function is the identity for colours inside that ellipsoid. For colours that fall outside the ellipsoid, the colour is taken on the line from the colour towards its projection onto the ellipsoid (typically, the new colour is at half distance between the pixel's colour and its projection). By using this function, colours still reflect what was initially in the image but

[4] An angle is detected at a control point when the minimum cosine of the angle made from the control point and neighbouring ones is higher than all neighbouring cosines.

are closer to the region properties. This prevents colours which are very different from the ones inside the region spreading into it.

As for the segmentation, colour projections are made in the CIE $L^*a^*b^*$ space. Indeed, these projections are not feasible in the RGB colour space since the results (as well as the Euclidean distance) do not correspond to any psychophysical reality, and would create colours that were visually wrong.

4 RESULTS

We present some results of the extraction of structural contours in simple images to assess the precision of the sub-pixel extraction (Section 4.1). We also show structural contours on a real image as well as the steps leading to these contours and the representation of the structural regions (Section 4.2).

Results of image synthesis as well as some special effects can be seen in [Froum00].

4.1 Precision of the Sub-pixel Extraction in Structural Contours

The close-ups showing the boundaries display the region (in grey) as it is in the original image, the original NURBS curve used to generate the region (dashed line), the snake at its initial position (jagged continuous line), and the snake at its final position (smooth continuous line). They show typical maximum local errors. Errors between the reconstructed shape and the original shape are given as a percentage of the area of the shape for the parts that are outside, inside, and both outside and inside of the original shape. These quantities are measured as follows. We render on the same image at very high resolution and with anti-aliasing (as was the original image) the true shape and the reconstructed shape with two different colours and with appropriate mixing rules such that pixels belonging to one shape *or* the other are of the corresponding colour. Then we count the number of pixels having one or the other colour and the number of pixels in the true shape to obtain the percentages shown. Since we count pixels having *any* amount of one or the other colour as being inside or outside, the percentages we get are over-estimated. In fact, Figure 2 shows the total error depending on the image resolution at which it has been measured for the CARDIOID image. All the values shown are measured at a resolution of 4000 pixels (in fact 4000×4000 pixels, the original images having 200×200 pixels).

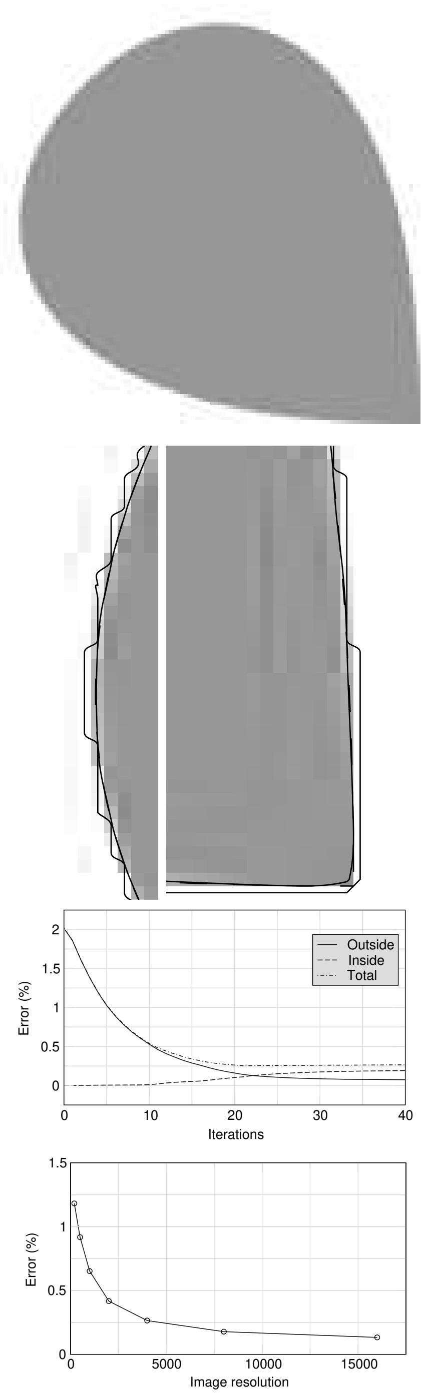

Figure 2: CARDIOID: image. boundary, and errors

Image CARDIOID**.** The first image shows a convex shape having a sharp angle. Figure 2 shows that the extracted contour is very close to the original curve, as the error curve attests. We can see that the sharp angle is detected (an angle is present in the final curve) but inside the original shape. The problem comes from the natural tension in the snake. Note however that the error is only around half a pixel at the resolution of the original image.

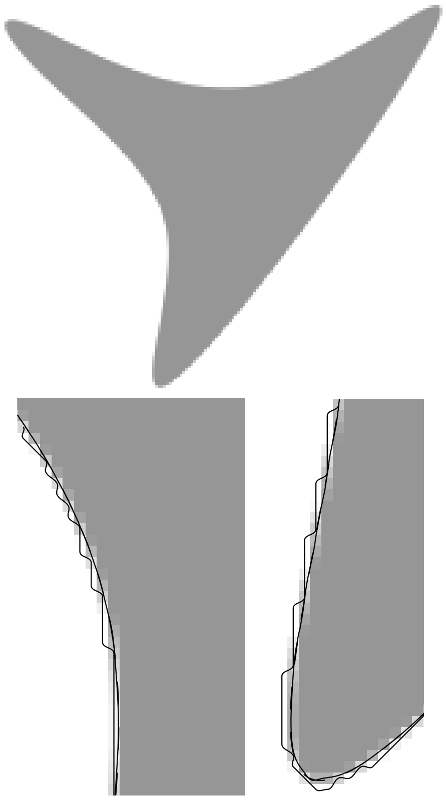

Figure 3: Y: boundary

Table 1: Errors of sub-pixel contours

Image	Out (%)	In (%)	Total (%)
CARDIOID	0.07	0.19	0.26
Y	0.17	0.25	0.42
BLOB	0.07	0.11	0.18

Image Y. In this experiment (Figure 3), we use a shape having no sharp angles but high curvatures and concavities. The final total error is 0.4% of the shape area. Figure 3 shows that the concavities are no problem. We can see that the high curvature parts are still inside the original shape, but much less than in the case of a sharp angle.

Table 1 gives the errors for the two previous images as well as for the BLOB image shown on Figure 4. Note that the error for the Y image seems to be more important compared to the others. This comes from the fact that the curve to surface ratio is greater in the Y image than in the others.

4.2 Image Representation

We show in this section results for the GOOSE image (Figure 1). Figure 5 shows the parts in the image that the user has drawn to specify the region attributes (see Section 2.1). Figure 6 shows

Figure 4: The BLOB image

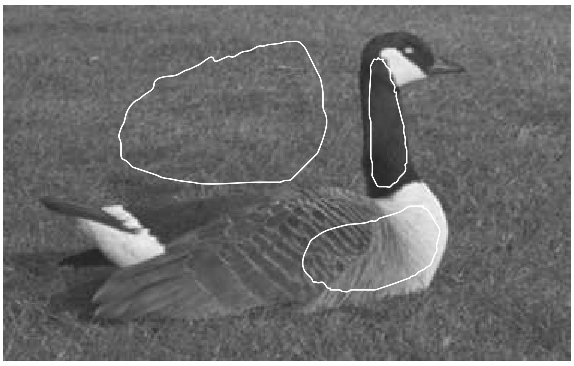

Figure 5: The image parts used to compute the region attributes

the structural contours corresponding to the regions. As can be seen, the contours follow the boundaries in the image. The close-up at the bottom of the neck, however, shows that the boundary of the white part is towards the black of the neck. This is because the white colour is part of a region including the light and dark (brown in the image) parts of the body (see Figure 5) thus having a large value for its variance, thus including colours between the white and the black in the "white-brown" region.

Figure 7, shows the triangulation for one of the regions: the neck of the goose.

Figure 8 shows all the stages as well as the type of information used and created to extract the sub-pixel boundaries of an image and to create its vectorial representation.

5 CONCLUSION

We have proposed in this paper a first step towards the creation of continuous image representations to allow special effects for cinema and many other applications. We have shown that images are decomposed into structural regions, the interior of which being characterised either

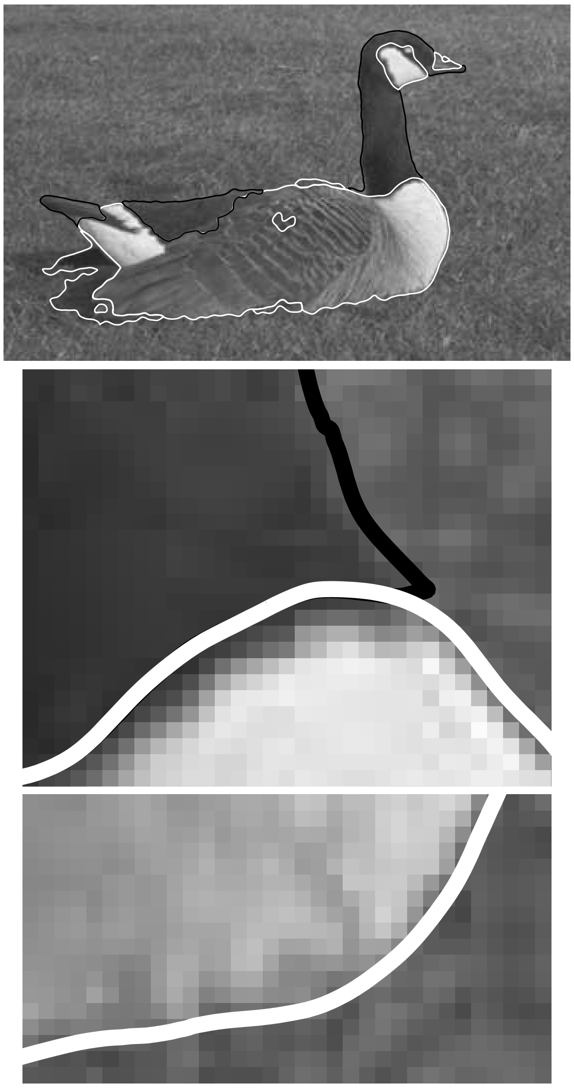

Figure 6: Structural contours extracted with the region attributes selected as in Figure 5

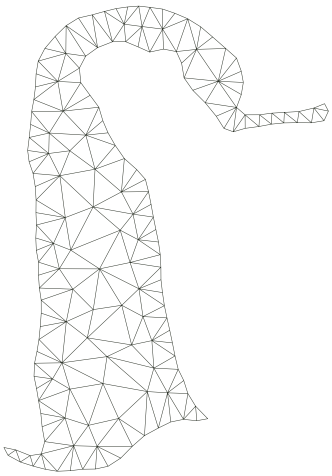

Figure 7: The triangulation of the goose's neck

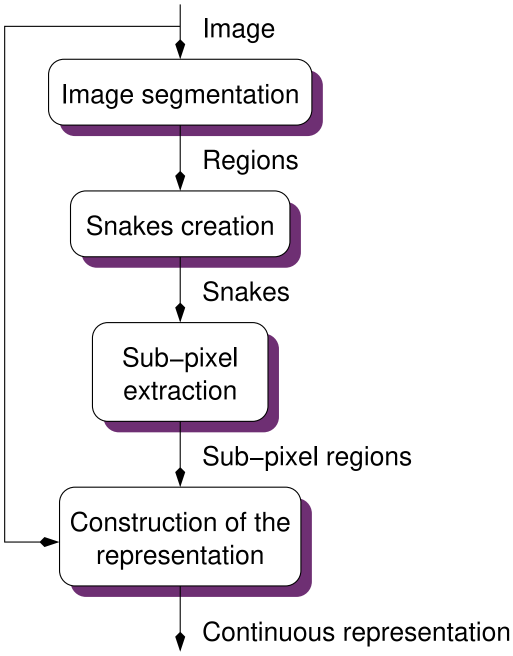

Figure 8: The whole process of the image representation

with a single colour or with a set of colours linked by a triangular mesh. The contours are smooth, continuous, and at sub-pixel accuracy. We have shown that the structural region contours extracted from controlled data are very close to the original data. It has been shown in [Froum99] that such representations can be rendered at very high resolutions in affordable computing times. Moreover, an extensive range of different renderings from such representations has been described in [Froum00].

ACKNOWLEDGMENTS

This work was supported by the European Union *Training and Mobility of Researchers* project and by the UK *Engineering and Physical Sciences Research Council*. We also would like to thank Max Froumentin and Pascal Fua for all the useful discussions we had.

REFERENCES

[Elder98] James H. Elder and Rick M. Goldberg. Image editing in the contour domain. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Santa Barbara, CA, USA, 1998.

[Froum99] Max Froumentin and Philip Willis. An efficient $2\frac{1}{2}$D rendering and compositing system. *Computer Graphics Forum*, 18(3):C385–C394 and C428, 1999.

[Froum00] Max Froumentin, Frédéric Labrosse, and Philip Willis. Vector-based representation for image warping. *Computer Graphics Forum*, 19(3):C419–C425 and C543, 2000.

[Fua90] Pascal Fua and Yvan Leclerc. Model driven edge detection. *Machine Vision and Applications*, 3:45–56, 1990.

[Garba86] Catherine Garbay. Image structure representation and processing: A discussion of some segmentation methods in cytology. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(2):140–146, 1986.

[Geige95] Davi Geiger, Alok Gupta, Luiz A. Costa, and John Vlontzos. Dynamic programming for detecting, tracking, and matching deformable contours. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(3):294–302, 1995.

[Hanse97] Michael W. Hansen and William E. Higgins. Relaxation methods for supervised image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(9):949–962, 1997.

[Henri98] Olof Henricsson. The role of color attributes and similarity grouping in 3-D building reconstruction. *Computer Vision and Image Understanding*, 72(2):163–184, 1998.

[Humme83] Robert A. Hummel and Steven W. Zucker. On the foundations of relaxation labelling processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-5(3):267–287, 1983.

[Kass88] Michael Kass, Andrew Witkin, and Demetri Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331, 1988.

[Labro00] Frédéric Labrosse. Towards continuous image representations. Technical report, Department of Mathematical Sciences, Computing Group, University of Bath, Bath, UK, 2000. `http://www.maths.bath.ac.uk/~masfl`.

[Meyer87] Gary W. Meyer and Donald P. Greenberg. Perceptual color spaces for computer graphics. In H. John Durret, editor, *Color and the Computer*, pages 83–100. Academic Press, Inc., San Diego, CA, USA, 1987.

[Morte95] Eric N. Mortensen and William A. Barrett. Intelligent scissors for image composition. In *Proceedings of SIGGRAPH 95*, *Computer Graphics* Proceedings, Annual Conference Series, pages 191–198, Los Angeles, CA, USA, 1995.

[Richa81] J.A. Richards, D.A. Landgrebe, and P.H. Swain. On the accuracy of pixel relaxation labeling. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-11(4):303–309, 1981.

[Rosen76] Azriel Rosenfeld, Robert A. Hummel, and Steven W. Zucker. Scene labelling by relaxation operations. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-6(6):420–433, 1976.

[Shewc96] Jonathan R. Shewchuk. Triangle: Engineering a 2D quality mesh generator and delaunay triangulator. In Ming C. Lin and Dinesh Manocha, editors, *Applied Computational Geometry: Towards Geometric Engineering*, volume 1148 of *Lecture Notes in Computer Science*, pages 203–222. Springer-Verlag, 1996. From the First ACM Workshop on Applied Computational Geometry.

[Terzo87] Demetri Terzopoulos. Matching deformable models to images: Direct and iterative solutions. In *Topical Meeting on Machine Vision, Technical Digest Series*, volume 12, pages 164–167, Washington, DC, 1987. Optical Society of America.

[Willi92] Donna J. Williams and Mubarak Shash. A fast algorithm for active contours and curvature estimation. *CVGIP: Image Understanding*, 55(1):14–26, 1992.

[Wysze82] Günter Wyszecki and Walter S. Stiles. *Color Science: Concepts and Methods, Quantitative Data and Formulae*. John Wiley & Sons, 2nd edition, 1982.