Clinical Linguistics & Phonetics, Sept–Nov 2005; 19(6/7): 555–565

Taylor & Francis
Taylor & Francis Group

Detecting the edge of the tongue: A tutorial

KHALIL ISKAROUS

Haskins Laboratories, New Haven, CT

(Received 25 March 2004; accepted 31 January 2005)

Abstract

The goal of this paper is to provide a tutorial introduction to the topic of edge detection of the tongue from ultrasound scans for researchers in speech science and phonetics. The method introduced here is Active Contours (also called snakes), a method for searching for an edge, assuming that it is a smooth curve in the image data. The advantage of this approach is that it is robust to the noisy speckle that clouds edges. This method has been implemented in several software packages currently used for detecting the edge of the tongue in ultrasound images. The tutorial concludes with an overview of the scale-space and Kalman filter approaches, state-of-the-art developments in image processing that will likely influence work on tongue edge detection in the coming years.

Keywords: *Ultrasound, edge detection, tongue, active contours*

Several modalities have been used for imaging the tongue during speech production. Fleshpoint tracking systems like Electromagnetic Articulography (Perkell, Cohen, Svirsky, Matthies, Garabieta, & Jackson, 1992) and X-ray Microbeam (Westbury, 1994) detect a small number of points on the tongue, while imaging modalities like X-ray cineradiography (Perkell, 1969), MRI (Baer, Gore, Gracco, & Nye, 1991) and Ultrasound (Stone, 1990) image continuous sections of the tongue. The advantage of the fleshpoint detection modalities is that their output contains exact information about the location of the points of the tongue tracked. If the shape of the tongue is not very complex, as it is for some vowels for instance, the continuous edge of the tongue can be interpolated from the partial information provided by the few points tracked using a smooth curve (Kaburagi & Honda, 1994). However, the tongue can assume fairly complicated shapes. For instance, when contact with the palate occurs, flattening of portions of the edge can occur. Interpolation between the fleshpoints misses this flattening.

The output of the imaging modalities contains a continuous edge, including flattening and other possible irregularities, but the output is an image, not numerical data. A crucial step for making such images useful is to automatically detect the edge of the tongue from them. Ultrasound has a distinct advantage in this regard, since its principle of operation makes it a physical edge detection device. High frequency sound from the probe travels in the medium until a medium with different sound-transmitting properties is reached, at

Correspondence: Khalil Iskarous, Haskins Laboratories, 270 Crown St., New Haven, CT 06511, USA. Tel: 203 865 6163. Fax: 203 865 8963. E-mail: iskarous@haskins.yale.edu

ISSN 0269-9206 print/ISSN 1464-5076 online © 2005 Taylor & Francis Group Ltd
DOI: 10.1080/02699200500113871

which point some of the sound is reflected back to the receiver, specifying the distance to the mismatch, which is then graphically indicated in the image with brightness proportional to the magnitude of the mismatch. In the application to tongue imaging, the probe is placed under the chin and sound travels through the tongue tissue until it reaches the air layer above. Tissue and air have drastically different sound-transmitting properties so the reflection is very high in magnitude, yielding a distinct edge effect on the image. The actual edge of the tongue is the interface between the bright reflection from the air layer and the darker region that represents the tongue tissue. X-ray and MRI work on different physical principles, which gives them less advantage in edge detection. For X-ray, detecting the edge becomes so difficult that the most usual method is manual tracing.

The raw ultrasound image is close to specifying the edge of the tongue, but it is usually rich in speckle, which results from the scattering of sound from targets smaller than the wavelength of the sound. Reflection from edges are also minimal when the mismatch is parallel to the sound travel direction, so some parts of the tongue will often disappear from the image altogether. Additionally, some structures within the tongue can appear quite brightly. For these reasons, detection of the edge of the tongue from the ultrasound scanner output requires further processing.

There are many approaches to edge detection (Jain, 1989; Pitas, 2000; Gonzalez & Woods, 2002). In this paper, a popular method of edge detection in ultrasound is introduced. Speech scientists and phoneticians are increasingly making use of ultrasound to probe the motion of the tongue during speech production and swallowing. The purpose of this paper is to provide this audience with some background in the edge detection concepts that would allow them to make better use of software developed by engineers and computer scientists. Section II provides a tutorial introduction to edge detection using *snakes* (Kass, Witkin, & Terzopoulos, 1987). Section III provides an overview of state-of-the-art developments in edge detection that have application to tongue edge detection including scale-space techniques and Kalman filter approaches to dynamic edge detection.

Snakes

Ultrasound has been used for about 20 years in the field of tongue imaging (Morrish, Stone, Shawker, & Sonies, 1985). A variety of methods have been attempted to detect the edge of the tongue from ultrasound images, but the most often used is Active Contours or snakes. This approach is based on work in the 1980s (Kass et al., 1987), but it has provided a framework in which many of the current developments in the field are expressed.

There are several requirements that a group of pixels in an image must have to qualify as an edge. The most basic of these criteria is that this group of pixels constitute a discontinuity in some property of the image, usually the brightness. A grayscale image is represented on a computer as a matrix, where the number in each row and column represents the brightness of a pixel in the image. To understand how an edge can be found, we begin with a simpler image that is one dimensional, e.g., [34 37 36 52 49 53 52 53]. It is easy to tell by eye that there is a discontinuity between the third and fourth pixel. How can this discontinuity be automatically detected? The discontinuity is indicated by a large difference between consecutive values. If we calculate the differences we obtain: [3 −1 16 −3 4 −1 1]. The discontinuity can then be detected by picking out the maximum of this signal. Computationally the differencing step is accomplished by multiplying the signal [−1

1] by each consecutive pair of pixel values (componentwise) and summing the result, e.g., [34 37] * [−1 1]=3. This step is called *convolution*, what we have done is to convolve the image with the mask [−1 1].

This is the first step of automatic edge detection in general. In the two dimensional case, we have a matrix of pixel values. To detect an edge, we convolve the matrix with a matrix mask that emphasizes discontinuities in the image. If the mask differences the image, the output of convolution is called the *gradient*. Pixels that are judged by a human to constitute an edge co-occur with maxima of the gradient. However due to noise, there will be far more maxima in the gradient than there are edges in the image. A white pixel of noise in a background of black (e.g., [0 0 256 0 0]) would be registered as part of an edge by a poor edge detection algorithm. Besides the ever-present noise, an additional problem is that, as mentioned earlier, portions of the tongue that are nearly parallel to the direction of sound travel do not reflect sound and therefore show up poorly in the edge. The gradient of the tongue image will therefore be high at some pixels corresponding to the true edge of the tongue, but it may be low at other points of the true edge. In addition the gradient will also be high at non-uniformities inside the tongue as well as at pixel noise and speckle regions.

An additional constraint on edges that is especially true of the tongue edge is that an edge is a smooth entity—two neighboring pixels of an edge are close to each other. Active contours or "snakes" is an iterative approach to edge detection that embodies the smoothness constraint on edges (Kass et al., 1987). A snake is a curve whose shape evolves in time to move closer and closer to the true edge in the object. The initial position of the curve is chosen to be somewhat close to the true edge, and an algorithm iteratively changes the positions of the points on the curve so that they move closer and closer to points with a high gradient, while maintaining a smooth curve. The snake slithers to lock onto the edge, as it were. The snake at time 0 is either provided by the user, who draws it by hand close to the image, or it can be the edge already detected from a previous image, if we know that the true edge in the current frame is not far from the edge already found in the previous image. Two forces act on the curve at each time step (each iteration) to deform it to better approximate the edge. The first type of force is internal to the snake. It is a force that makes the curve smoother and smoother. We can calculate this force, for instance, by comparing the difference of the position of each point in the curve with the position of the two points surrounding it. If the position of the point is far from those that surround it, we say that this point increases the energy of the snake. This is in analogy with an elastic stick one of whose points is far from the points around it. This stick has high elastic energy (tension) that forces the point to go back to its equilibrium position. In the same way, if there is a large difference in the location of a point on the snake from those surrounding it, a large force acts on the snake at that point which forces that point to assume a position nearer to the points surrounding it. At each iteration, therefore, an internal force is calculated for each point of the curve and that force is made to act on the curve, changing its shape to a smoother one. Snake algorithms are called energy minimization algorithms, since at each iteration, the total internal energy of the curve is reduced. Figure 1, shows an initial snake together with twenty five iterations, using an internal force which makes each point assume a location matching the average of the locations of the two points on the curve closest to it. This particular force is one of many that could have been used. As the iterations proceed, the snake becomes smoother, until it becomes a line.

The internal force on the snake by itself is therefore not very useful, unless we always want to recognize lines. Another force coming from the image itself is necessary. This is the

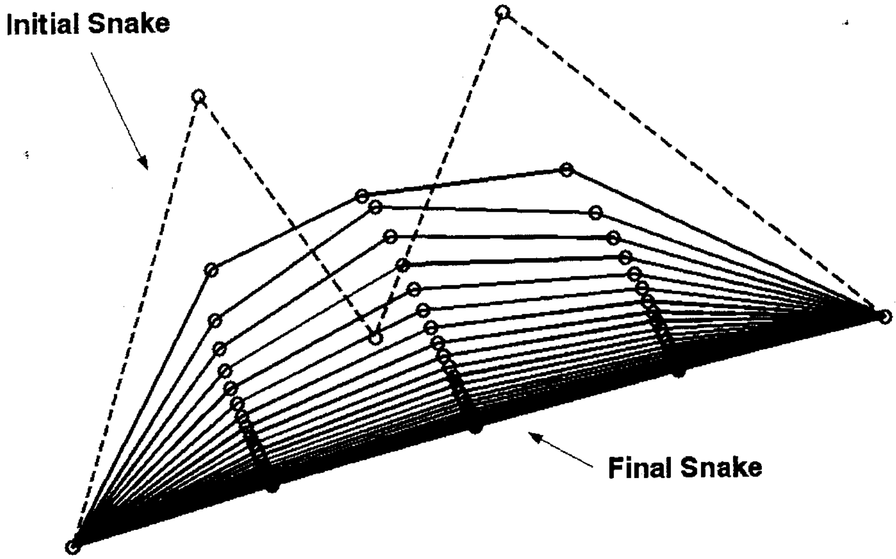

Figure 1. The initial snake converges to a smooth shape, a line, by minimizing its own internal energy.

external force. It arises from features of the image that we aim to latch onto, like high gradient points. At each iteration, an external force acts on each point forcing it to a neighboring point with a higher gradient value. This process can be understood as energy minimization. To see this, imagine a three dimensional representation of the image, where the height of each point represents the negative of the gradient at that point. Edges are valleys in this terrain. The snake starts out on a higher terrain and the external forces attract the snake to the valley. This process can be understood as energy minimization since particles on a high peak have more potential energy than those in a valley. Figure 2 shows an image of a black disk surrounded by a white one. An initial snake is placed in the white region. It deforms due to the external force of the image gradient for 15 iterations. It converges after about the fifth iteration to the gray edge between the white and gray. The implementation used here is called the Greedy Algorithm (Williams and Shah, 1992). At each iteration, a window of 3–5 pixels is defined around every snake point and a search is performed for a pixel with a lower negative gradient (lower energy). That snake point is then replaced with the one with lower energy. This process is repeated several times, and a threshold is set for the distance between consecutive snakes beyond which no more iterations are performed. Iterations are performed until convergence is reached, i.e., the threshold is reached.

The examples presented so far show a snake only under an internal force (Figure 1) or an external force (Figure 2). In practice of course, a snake evolves under both forces. But it often happens that points on the curve move too far apart or too close to each other, altering the size of the initial curve. But the initial snake is often chosen by the researcher to be the basic shape of the edge sought, so it is not desirable for the snake to become much larger or smaller. For this reason, there is an additional internal force that penalizes the change in the distance between snake points.

After this intuitive introduction to snakes, it is possible to appreciate their mathematical specification. The snake is a curve g(s) = [x(s), y(s)], where *s* is a parameter specifying the

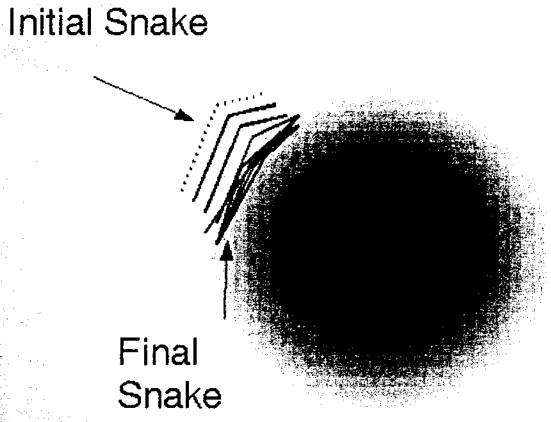

Figure 2. The initial snake converges to the gray edge between the white and black as it minimizes its external energy.

specific point on the snake running usually from 0 to 1, and x and y specify that point's location in the plane. We can specify the energy of the curve as:

$$E = E_{Internal} + E_{External}$$

$$E_{Internal} = \int_0^1 \alpha|g'(s)| + \beta|g''(s)|ds$$

$$E_{External} = -\nabla I(x,y)^2$$

In the internal energy term, the integral in the variable s denotes the fact that we are summing the energy for all the points in the snake. The first term g′(s) is the first derivative of the location of snake points, which specifies the distance between snake points. The internal energy of the curve is therefore greater if the distance between the points is large. The coefficient α is chosen by the user and specifies how significant this term should be in the total energy. The second term g″(s) is the second derivative of the location of the snake points and denotes the difference in location of each point from the average of the surrounding points. The closer each point is to its neighbors, the smoother the curve. β can be set to determine the contribution of this term to the total energy. In general, the coefficients can be set arbitrarily and as a function of s by the user to accelerate the convergence of the snake to the edge. In the external energy term, I(x,y) refers to the intensity or brightness of each point in the image. V is the gradient operator, whose square is taken. As discussed above the external energy is the negative of the gradient. The snake evolves by minimizing its total energy. This is usually done through a search at each iteration for new points whose total energy is less using the Greedy Algorithm. Different snakes implementations differ with respect to the details of the internal and external forces as well as the algorithm used to minimize the energy of the snake.

Akgul, Kambhamettu, & Stone (1999) applied the snakes concepts described above to the detection of the edge of the tongue from ultrasound scans. In their implementation, there are three terms in the internal energy. The first term is the smoothness energy $E_{smo}=1-\cos\theta_i$. θ specifies the angle between the i^{th} and $i+1^{th}$ point on the snake. If the angle at a specific point is 0, the $\cos\theta=1$, and $E_{smo}=0$. This is the maximally smooth case. The opposite case is if the angle is 90°, in which case $E_{smo}=1$, and a strong force would act to reduce this angle at the next iteration. The second term is the similarity energy E_{sim}. This term forces the final snake to be as similar as possible to some model snake S, which encapsulates how the edge of the tongue looks to the user—E_{sim} measures the departure of shape the snake at each iteration from the model snake S. The third term E_{dist} measures the stretching of the snake and penalizes a snake when its points grow farther apart. Its form is: $E_{dist}(v_i)=\left|\frac{|v_i-v_{i-1}|}{\frac{1}{n-1}\sum_{j=2}^{n}|v_j-v_{j-1}|}-1\right|$. v specifies the snake point and the energy contributed by each point is therefore measured by the stretch between itself and the next point as normalized by the total length. The external energy term in their implementation is $E_{Ext}=-|\nabla I(v_i)|$, which is similar to the general term, but is implemented using optical flow, an advanced method for gradient computation. By using these energy terms and a dynamic programming energy minimization algorithm, the authors demonstrated high quality detection of the edge of the tongue. Many examples can be seen in their paper.

Some examples will now be given of the application of the snakes technique to edge detection of the tongue. There are many ways to implement both the external energy and internal energy constraints, and each leads to different performance. One popular method for implementing the external energy is to search for the zero of the derivative of the gradient. This makes sense, since where the gradient is at a maximum, its derivative will be zero. Figure 3 shows the application of this approach to a portion of the tongue edge. The top panel shows a portion of the tongue edge. The middle panel shows the same portion after the application of a Laplacian of Gaussian filter (LoG), which roughly takes the derivative of the gradient of the smoothed image. As can be seen, the edge of the tongue, which is between the white layer and the darker layer below it in the top panel, becomes sandwiched between two layers black and white after the application of the filter, so the edge of the tongue has been emphasized. The bottom layer shows the automatically detected maxima and minima of the output of the LoG filter, and edge of the tongue can be found between them as a zero of the output.

A popular method of implementing the internal energy is the B-spline method (Blake & Isard, 1998; Menet, Saint-Marc, & Medioni, 1990; Blake, Curwen, & Zisserman, 1993), where the edge is specified to contain a small number of cubic curves that join to each other smoothly. B-splines in this context are discussed in (Blake & Isard, 1998; Iskarous, 2001). An application of this technique to four frames from the sequence [ad] can be seen in Figure 4. As can be seen, the method leads to high quality detection for most parts of the tongue. Arrows point to portions where the edge was missed. The main variable that can be changed to improve performance is how many curve segments to include in the edge. If a very small number of segments are chosen (3–5), then nuances in the edge are missed, but if there are too many segments (more than ten), the edge will be over-fitted. An example of such a situation is shown in Figure 5. High quality detection is achieved by manual intervention to change the number of segments, if the fit of the edge is seen to be less than optimal. Unfortunately, most methods still do require human intervention, but as the methods improve, less and less reliance on the operator is needed.

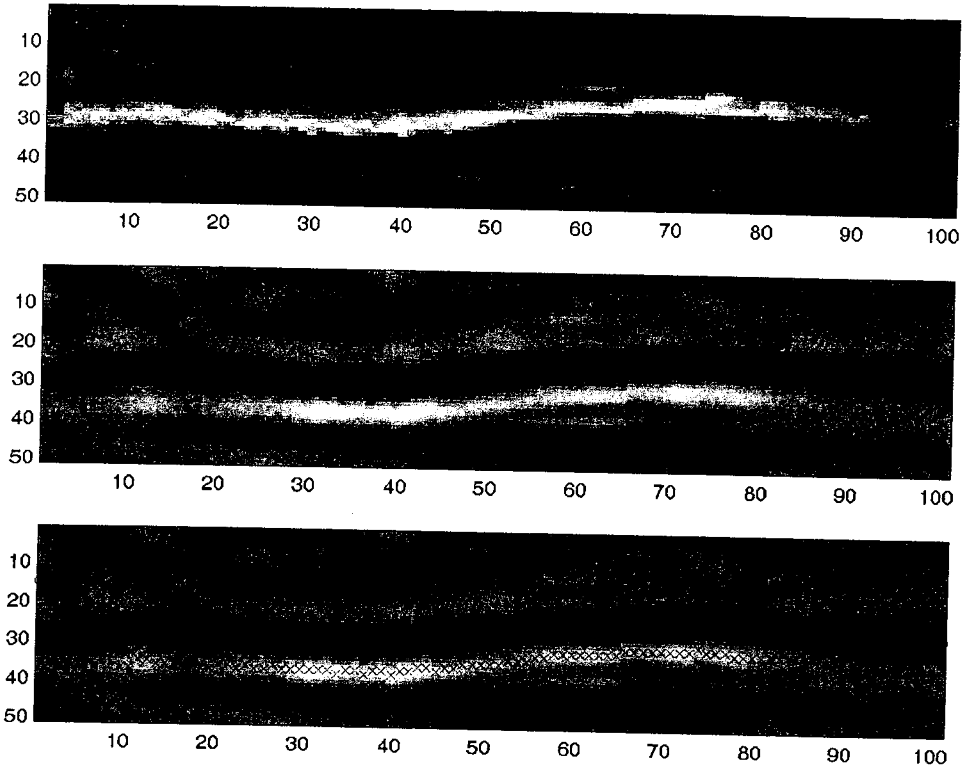

Figure 3. Top: A portion of a tongue edge from an ultrasound image. Middle: The result of convolving the window with LoG. Bottom: Automatically detected minima and maxima of LoG.

Future directions in edge detection

The snakes-based techniques discussed in this paper form the foundation of most techniques used for the application of tongue edge detection. Within the snakes framework,

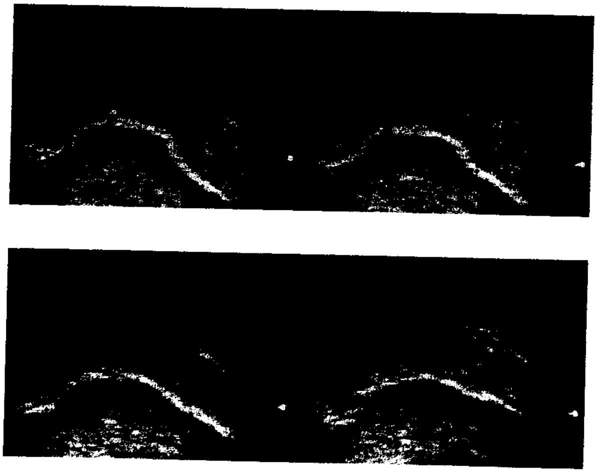

Figure 4. Frames from the sequence [ad] spoken by an American English male speaker. The frames shown are from the lowest position for the [a] to the release of the [d].

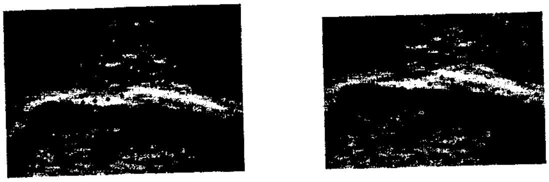

Figure 5. Illustration of the effect of varying number of spline segments in the curve on edge detection quality. The same frame is subjected to detection by a curve of four segments (Left) and ten segments (Right).

several new approaches have emerged in the last ten years that have improved the quality of detection. Two of these will be covered briefly in this section, since they have already been used in tongue tracking and are likely to influence progress in this field: the first is scale-space/wavelets and the second is Kalman filtering.

As discussed in the previous section, edge detection becomes more successful, the more we constrain what is a possible edge, e.g., that it is a smooth entity. Scale-space methods look at edges as features of an image that are independent of the scale at which the image is processed. Any given image can be looked at from a coarse scale, where the details in the image have been removed or from a fine scale, where the details are left. Indeed there is a continuum of scales where layers of details are successively removed. For instance, Figure 6 shows a portion of a tongue edge seen from six different scales from fine to coarse.

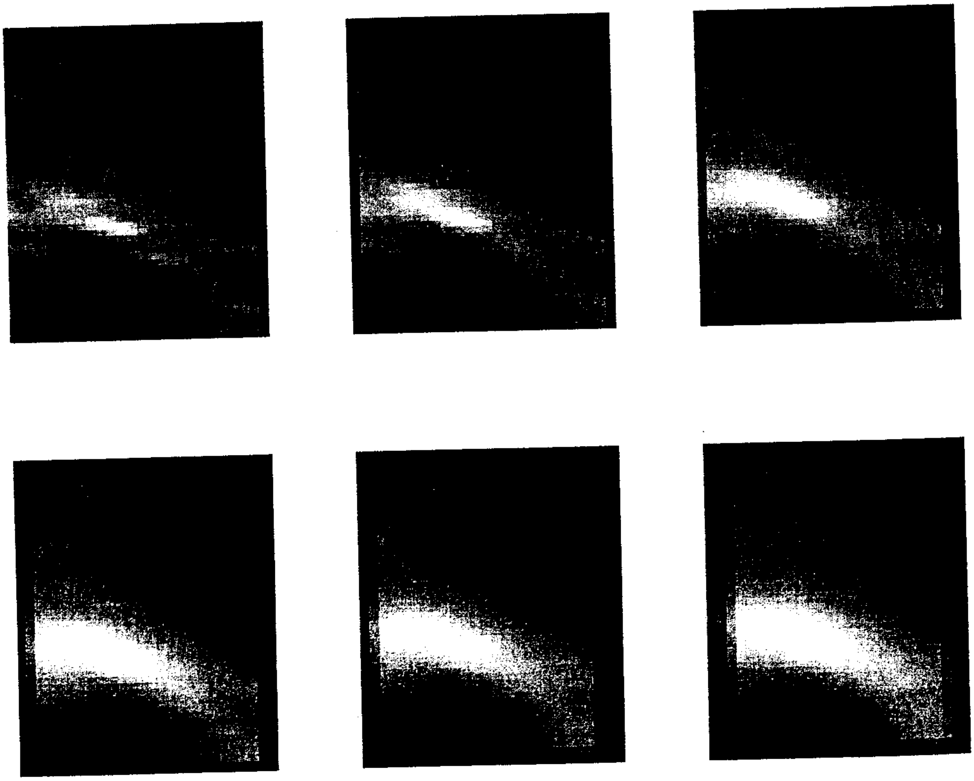

Figure 6. Portion of tongue seen from six different scales formed by Gaussian filtering of the image with sigma levels from 1 to 8.

The different scales were formed by filtering the image with a Gaussian filter with different sigmas. One thing that is preserved at the different scales, however, is the presence of the edge. It was the observation of Marr (1982) that an edge is preserved under the successive blurring of an image, i.e., the edges are scale invariant. Figure 7 shows the gradient of a portion of a tongue edge under a fine scale (Left) and a coarse scale (Right). It can be seen that the gradient is more coherent and more informative of the edge at the lower scale. This approach to edge detection has informed work in snakes based edge detection by performing the search for an edge under different scales and finding the optimal scale at which to search for the edge (Tsap, Goldgof, & Sarkar, 2000; Geiger & Kogler, 1993), and has been applied by Akgul & Kambhamettu (2003) to edge detection in ultrasound images.

All the approaches covered until now are basically static in that each image in a sequence is treated separately. Dynamic approaches to edge detection assume that the edge is a deforming contour that is physically evolving through a sequence. A dynamical system is assumed to underlie the evolution of the edge and information about edge evolution is incorporated into the search for an edge. This is therefore yet another way to improve edge detection by constraining the search to edges that could have physically evolved from an earlier edge. A popular approach to dynamic edge detection that has been incorporated in the snakes approach is the Kalman Filter. In this technique, a dynamical system is assumed to have a number of state variables x describing its configuration variables at any moment in type. The dynamical system then evolves in time according to a difference/differential equation $\dot{x}=Ax+Bu$, where $\dot{x}$ is the change in state, A describes how earlier states influence states, u is a control variable, and B is the relation between u and the change in x. When the state evolves it is then observed as the observable variable $y=Cx+Dn$, where C represents the influence of the observation device, n is noise and D represents the weight of the noise. The set up therefore is that of a dynamical system that evolves physically, but that is observed under noise. In the case of edge detection, the state variables are the parameters of the edge and the observable variable is the noisy edge as seen in an ultrasound image, for example. The goal of the Kalman Filter is to estimate the state variable (edge) from the observed variable (noisy image) by using two types of information: assumptions about A, the dynamic component that allows us to predict how the edge deterministically evolves, and the noisy observations. The Kalman Filter is a way to automatically decide how to weigh the information from previous edges against the information from the noisy

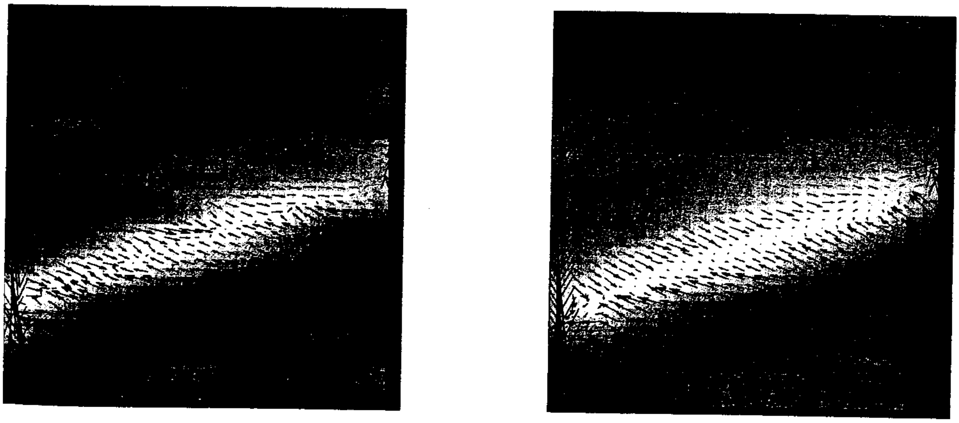

Figure 7. The gradient around the tongue edge at a scale of sigma=2 (Left) and 6 (Right).

observations. For instance, at a certain stage in edge evolution, it may be that the observations become very noisy, in which case the Kalman Filter would find an edge very close to that found at the last stage, weighing the noisy observation very low. At a different point in time, there maybe a lot of doubt as to the assumed dynamics A of the edge evolution process, in which case, the edge would follow the noisy observation more than the edge predicted by the dynamics A. The Kalman filter automatically determines the weights to attach to previous observations vs. noisy current observations by solving a Riccati Equation, a first order differential equation of second degree. This technique to edge detection is explored at great length by Blake and Isard (1998).

Conclusion

In this paper, the author has outlined research in the detection of the tongue edge from ultrasound scans. The progress is from an approach in which only local information about edges is used, to one where the edge is assumed to be smooth. A brief introduction was also given of scale-space and dynamic approaches to edge detection. One of the many avenues of further research in this area is how to combine all the spatiotemporal constraints discussed into a system that uses as much a priori information about the edge to improve the search. One way this will evolve is by learning a great deal more about how the tongue edge actually does deform in different speech tasks, since the more assumptions we can make about A, the dynamics, the more constrained is the search for the edge. This is likely to involve both theoretical work on modeling tongue dynamics by muscle deformation simulations as well as extensive empirical work on how the tongue actually deforms within a syllable across speakers and languages.

Acknowledgments

This research was supported by NIH grant DC-02717 and DC-006705 to Haskins Laboratories. I thank Douglas H. Whalen, Marianne Pouplier, and Heike Lehnert for helpful comments.

References

Akgul, Y. S., & Kambhamettu, C. (2003). A Coarse-to-Fine deformable contour optimization framework. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 25*, 174–186.

Akgul, Y. S., Kambhamettu, C., & Stone, M. (1999). Automatic extraction and tracking of the tongue contours. *IEEE Transactions on Medical Imaging, 18*, 1035–1045.

Baer, T., Gore, J., Gracco, C., & Nye, R. (1991). Analysis of vocal tract shape and dimensions using magnetic resonance imaging: Vowels. *Journal of the Acoustical Society America, 90*, 799–828.

Blake, A., & Isard, M. (1998). *Active Contours.* Berlin: Springer.

Blake, A., Curwen, R., & Zisserman, A. (1993). A framework for spatio-temporal control in the tracking of visual contours. *International Journal of Computer Vision, 11*, 127–145.

Geiger, D., & Kogler, J. E. (1993). Scaling images and image features via the renormalization group. *Computer Vision and Pattern Recognition* 47–53.

Gonzalez, R., & Woods, R. (2002). *Digital Image Processing.* Upper Saddle River, NJ, USA: Prentice-Hall.

Iskarous, K. (2001). Dynamic Acoustic-Articulatory Relations, PhD dissertation. University of Illinois at Urbana-Champaign.

Jain, A. K. (1989). *Fundamentals of Digital Image Processing.* Englewood Cliffs, NJ, USA: Prentice-Hall.

Kaburagi, T., & Honda, K. (1994). Determination of sagittal tongue shape from the positions of points on the tongue surface. *Journal of the Acoustical Society* America, *96*, 1356–1366.

Kass, M., Witkin, A., & Terzopoulos, D. (1987). Snakes: Active contour model. *International Journal of Computer Vision, 1*, 321–331.

Marr, D. (1982). *Vision.* San Francisco: W.H. Freeman: Freeman.

Menet, S., Saint-Marc, P., & Medioni, G. (1990). B-snakes: Implementation and application to stereo. *Image understanding workshop,* 720–726.

Morrish, K., Stone, M., Shawker, T., & Sonies, B. (1985). Distinguishability of tongue shape during vowel production. *Journal of Phonetics, 13,* 189–203.

Perkell, J. S. (1969). Physiology of Speech Production: Results and Implications of a Quantitative Cineradiographic Study, PhD thesis, MIT.

Perkell, J. S., Cohen, M., Svirsky, M., Matthies, M., Garabieta, I., & Jackson, M. (1992). Electromagnetic midsagittal articulometer systems for transducing speech articulatory Movements. *Journal of the Acoustical Society America, 92,* 3078–3096.

Pitas, I. (2000). *Digital Image Processing Algorithms and Applications.* Chichester: Wiley.

Stone, M. (1990). A three-dimensional model of tongue movement based on ultrasound and X-ray microbeam data. *Journal of the Acoustical Society America, 87,* 2207–2217.

Tsap, L., Goldgof, D. B., & Sarkar, S. (2000). Multiscale combination of physically-based registration and deformation modeling. *Proceedings of IEEE conference on Computer Vision and Pattern Recognition,* 2422–429.

Westbury, J. R. (1994). *X-ray Microbeam Speech Production Database User's Handbook.* Madison, WI.

Williams, D. J., & Shah, M. (1992). A fast algorithm for active contours and curvature estimation. *Computer Vision and Image Processing Image Understanding, 55,* 14–26.