ANGIOGRAM SIMULATION SOFTWARE DOCUMENTATION

Jannick P. Rolland and Derek T. Puff

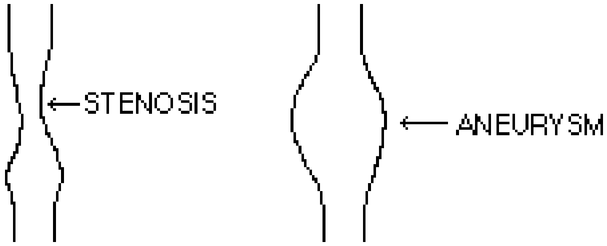
Department of Computer Science
University of North Carolina-Chapel Hill

ABSTRACT. We have developed software for the realistic simulation of a single vessel from an angiogram. Initially a parameterized space curve is generated which specifies the path of the vessel in a three-dimensional space. The position of an aneurysm or stenosis along the path of the vessel may then be specified. Finally, the volume of the vessel is constructed along the space curve, and the resultant image is acquired from projection through the vessel volume. This software enables the generation of remarkably realistic angiogram simulations for application in various medical image interpretation studies.

INTRODUCTION

Angiography is a radiological imaging modality that allows visualization of the normally radioluscent vascular anatomy. High radiographic attenuation is achieved in the vessels of interest by the injection, prior to exposure with diagnostic radiation, of a radio dense contrast agent (see example angiogram below).

One of the many tasks performed with this modality is to detect or measure the extent of an aneurysm or stenosis in the vessel. The presence of such a vessel abnormality may be indicative of a vascular disease; accurate diagnosis is essential in prevention of catastrophic vascular malfunction resulting from the rupture of an aneurysm or constricted blood flow in a major vessel. The vessel below possesses a moderate stenosis of the internal carotid artery (the left branch of the artery, distal to the bifurcation).



We are conducting experiments to study the perception and performance of either the human observer or a mathematical observer in several aneurysm/stenosis detection and estimation tasks, and we have utilized the angiogram simulation software described in this report for generating the realistic stimuli in these studies. Many aspects of the vessel generation are parameterized in order to enable acquisition of tightly-specified stimuli.

The simulation of an angiogram may be achieved by simulating individual blood vessels followed by the "artistic" summation of several of them. Clearly the real angiogram possesses many, branching blood vessels. We did not however find such a complicated simulation necessary for our studies, and the simplified approach that consists of adding individual blood vessels together leads to images that were actually mistaken to be real angiograms by several radiologists. Our general approach was to view three-dimensional (3D) blood vessels as general cylinders defined by a medial axis and a corresponding width function associated with each point along the medial axis. The simulation of an individual blood vessel consists then of the following steps 1) generating a 3D space curve that mathematically represents the medial axis of the blood vessel, 2) defining a region of interest on the 2D projection of the 3D space curve to locate potential stenoses or aneurysms, 3) generating the 3D blood vessel from the 3D space curve and the parameters that characterize the blood vessel; e.g. width, type of defect if any, defect parameters such as the strength and length of the defect, 4) generating one or several 2D projections of the 3D blood vessel. Each step of the simulation shall now be described.

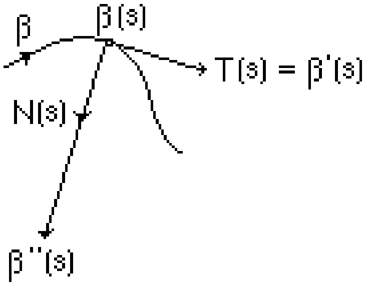
SPACE CURVE GENERATION

Introduction. Initially, a curve is generated which specifies the path of the vessel in three-dimensional space.

Frenet equations. The curve is specified by the Frenet formulas, mathematical specifications (or measurements) for the curvature and torsion of a 3D curve. The Frenet description given by O'Neill[1] is as follows:

The velocity vector of a curve β at position s is $\beta'(s)$, and the speed of the curve at s is the length of the velocity vector, or $||\beta'(s)||$. Let β be a unit speed curve; that is, for each s along β, $||\beta'(s)|| = 1$. Then $T=\beta'$ is called the unit *tangent vector* field on β. The derivative of $T' = \beta''$ measures the way that the curve is turning in three-space, and is called the *curvature vector* field of β. The length of T' gives a numerical measurement of the turning of β. The function $\kappa(s) = ||T'(s)||$ for all s along the curve is called the *curvature function* of β. $\kappa >= 0$, and the larger κ is the sharper is the turning of β.

$T'=\beta''$ is always normal to β. If we impose the restriction that $\kappa > 0$, then the unit vector field $N = T'/\kappa$ on β indicates the *direction* in which β is turning at each point, and is called the *principal normal vector* field of β. Then the vector field $\beta = T \times N$ on β is called the *binormal vector* field of β



from O'Neill, Fig 2.7, p. 57

The three vector fields T,N, and B are unit vectors which are mutually orthogonal at each point along the curve. The set of these three vectors (T,N,B) is called the Frenet frame field of β. The key to the successful study of the geometry of a curve β is to use its Frenet frame field whenever possible, for the Frenet frame field of β is full of information about β.

The Frenet formulas express the derivatives (T',N',B') of the Frenet frame field (T,N,B). If β is a unit-speed curve with curvature $\kappa > 0$ and torsion τ , then

$$T' = \kappa N$$

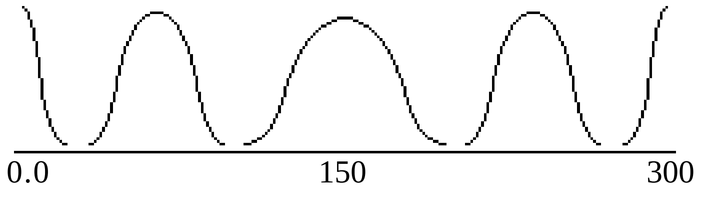
$$N' = -\kappa T + \tau B$$

$$B' = -\tau N$$

Given the curvature and torsion functions or values at each point along the arc length of the space curve, the Frenet formulas can be used to compute the new Frenet frame field at a point $s+\delta s$ as a function of κ, τ and (T,N,B) at s. In the angiogram software, a curvature κ and torsion τ are chosen at each point along the space curve according to some specified statistical distribution about the previous values of κ and τ as described in the next two sections. The next location in space (x,y,z) of the space curve is then computed given a small increment in arclength δs and the coordinates of the tangent vector T_x, T_y, and T_z at s.

<u>Curvature and Torsion Specification with No User Constraints.</u> If the user does not specify any constraints for the curvature and torsion command-line parameters, referred to as kappa and tau in the software, respectively, the curvature and torsion at each point along the arc length are chosen in the following manner.

Curvature. The curvature may take on values greater than 0 and less than or equal to 0.3. Those two values correspond to radii of curvature, infinity (or straight curve) and 3.33 (strongly bended curve), respectively. Due to the requirement that kappa be positive, a Poisson distribution or an approximation to a Poisson distribution is chosen for kappa. The value of kappa is first scaled by a factor of 1000 for all the values to be integers ranging from 0 to 300. If the scaled previous curvature value is less than 10 or greater than 290, the next value is chosen according to a Poisson distribution about the scaled previous value. If the previous curvature value is within those two values, the next value is chosen from a Gaussian distribution whose standard deviation equal to the mean and of mean value the scaled previous curvature value. The distributions are reflected about 150, so that previous curvature values greater than 150 are subtracted from 300 before selecting a new value via the described process.



Torsion. Torsion may take on values greater than or equal to -0.3 and less than or equal to 0.3. New values of the torsion are sampled from a Gaussian distribution of standard deviation 0.02 (value chosen empirically) and mean value the previous torsion value.

Additional constraints. Finally, if the newly selected curvature or torsion values are outside their range of variation (0.0 to 0.3 for kappa, and -0.3 to 0.3 for tau), or the percent change in the curvature or torsion is greater than 30%, the current curvature/torsion is assigned to the previous curvature/torsion. There exists also a mechanism for preventing extensive or continuous high curvature or torsion "loops": a count of consecutive curvatures/torsions that are greater than 0.29 is kept, and when the count exceeds 30 (empirical value that is a function of δs, the step size along the space curve), a new curvature/torsion is assigned which is the previous curvature or torsion divided by 3.5.

Curvature and Torsion Specification with User constraints. Curvature and torsion constraint parameters may be set by the user with command line specifications to the program. For both kappa (curvature) and tau (torsion), the current value is chosen with Gaussian probability (of a specified standard deviation) about the previous value. If the newly selected value is not within the specified upper and lower bounds, the previous value is returned.

Usage. At the time of this publication, the software resides in the directory /unc/rolland/vision/angio/src, and may be incorporated into the /usr/image library. Typing just "space_curve" will provide the following usage message:

```
Usage: space_curve [options] outfile
        [-D]        Print debugging information
        [-p <> <> <>]     Vessel starting position [def size/2]□
        [-K <> <> <>]     Kappa  sigma, lower bound, upper bound [def 0.01,0.0001,0.3]
        [-T <> <> <>]     Tau  sigma, lower bound, upper bound [def 0.01,-0.2,0.2]
        [-V Tx Ty Tz Nx Ny Nz Bx By Bz]  Init.Vec.(Tan,Norm,BiTan) [d random]
        [-d <> <> <>]     Output image dimensions [def 256]
        [-s <>]   Delta arc length [def 0.05]
        [-l <>]   Arc length (#increments of ds) [def ∞]
        [-m <>]  Multiple output files [def 1]
```

Parameters. The usage message indicates the parameters that may be specified on the command line for the "space_curve" program.

-D instructs the program to print to the screen a host of debugging messages.

-p designates the starting position in the three-dimensional space for the space curve. The default is for the starting positions to be the x,y, and z dimensions of the 3-D space divided by 2 (thus the center of the 3-D volume), but the x,y, and z values may be set from 0 to anything less than the x,y, and z dimensions. High values for starting positions in the x dimension correspond to starting positions to the "right" in the image; high values for starting positions in the y dimension correspond to starting positions at the "bottom" of the image; high values for starting positions in the z dimension correspond to starting positions towards the "back" of the 3-D volume.

-K allows user constraint of the curvature (kappa). The new curvature chosen at each position along the curve will be chosen according to a Gaussian distribution (of the specified standard deviation) centered about the previous curvature value. If the newly chosen curvature value is not within the specified bounds, the resulting curvature is assigned the previous curvature value. If neither kappa nor tau are specified by the user, the curvature and torsion are chosen as described in the previous section. If only one of the parameters is specified by the user, the other is defaulted according to the usage message.

-T allows user constraint of the torsion (tau). The new torsion chosen at each position along the curve will be chosen according to a Gaussian distribution (of the specified standard deviation) centered about the previous torsion value. If the newly chosen torsion value is not within the specified bounds, the resulting torsion is assigned the previous torsion value. If neither kappa nor tau are specified by the user, the curvature and torsion are chosen as described in the previous section. If only one of the parameters is specified by the user, the other is defaulted according to the usage message.

-V allows specification of the direction of the three initial vectors (T,N, and B). These three vectors must be triorthonormal. Thus the dot product of any two of the vectors must be zero, and the norm of each vector must be one. The default settings are random.

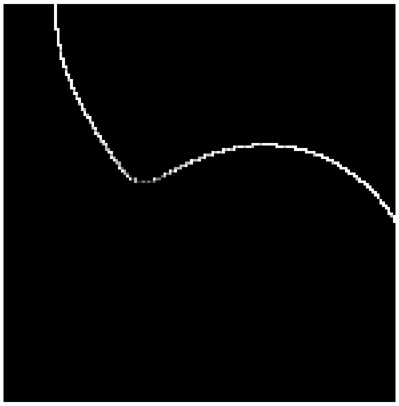
-d specifies the dimensions of the three-dimensional space and the dimensions of the resulting output image. The default value for each of the dimensions is 256.

-s allows adjustment of the size of the increment in arc length as the curve is constructed. The default is 0.05.

-l specifies the number of increments of the arc length to be computed before termination. The program normally terminates when the space curve encounters one of the boundaries of the 3-D space. This parameter instructs the program to also terminate after the space curve has attained the specified length. This parameter is dependent on the size of the increment of the arc length. The default is infinity (termination when collision with space boundary).

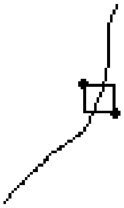
-m allows generation of multiple space curves, each in a different output file. The files are named according to the specified output name and a numerical suffix. The default is for a single output file.

Outputs. The program creates an output image (a /usr/image format BYTE image) containing a 2D projection of the single-pixel-wide space curve (see example below). The program also creates a parameter file (named with the output name and a "_parm" suffix) which contains the s, $\kappa(s)$, $\tau(s)$ values for each position along the curve as well as $\kappa_{xy}(s)$ the 2D curvature values of the projected space_curve (O'Neill, p75 ex14); it also creates a file (named with the output name and a "_st" suffix) which contains the 3D space curve location, x, y, z, as well as s and $\kappa_{xy}(s)$. Those files are required for use with the subsequent software.



REGION OF INTEREST (ROI) SPECIFICATION

Introduction. This software allows for the specification of a position along the space curve at which an aneurysm or stenosis can be formed in the vessel. When the program is executed, the single-pixel-wide space curve is displayed in a window. The user clicks with the left mouse button at a point at a corner of the region of interest for the position of the abnormality, and then clicks a second time to complete a box that forms the region of interest. Subsequent clicks with the left button allows the corners to be selected again. Clicking on the middle button terminates the program without saving an output file. Clicking with the right button terminates the program normally, writing the ROI dimensions to a file (see description in the "Outputs" section below) required by the subsequent angio program.



The way that the ROI information is used by the "angio" software is described in the following section.

Usage. At the time of this publication, the software resides in the directory /unc/rolland/vision/angio/src. Typing just "ROI" will provide the following usage message:

```
Usage: ROI input_image
```

The name specified for the input_image with "ROI" should be the same name as the output image from "space_curve."

Outputs. ROI simply produces a parameter file (named with the input name and a "_roi" suffix) for use with the "angio" software that contains the coordinates of the specified corners (x_1,y_1,x_2,y_2) of the region of interest.

ANGIO PROGRAM

Introduction. The "angio" software produces the simulated vessel by constructing the 3D vessel volume (whose position was determined by the "space_curve" software), and then projecting through the volume to form the 2D projection image. At each position along the space curve, a solid sphere of desired diameter is constructed. The simulation of the 3D blood vessel is equivalent to rolling the sphere along the space curve. The "angio" software also places a specified aneurysm or stenosis at a position in the vessel within the region of interest specified with the "ROI" software. More specifically, the center of the stenosis/aneurysm is placed at the point of highest 2D curvature value κ_{xy} within the ROI. Both the 2D location x and y as well as the value of the 2D curvature found are saved in the file "_parm" described in the space_curve generation section. The shape of the aneurysm or stenosis is characterized by a Gaussian, and as such has a height and a width. The height of the Gaussian is specified in terms of the percent fraction of the blood vessel width, while the width of the Gaussian is specified in terms of the length of the stenosis/aneurysm expressed in pixels unit. The diameter of the rolling sphere can be described by a constant plus a Gaussian function centered at the location of the stenosis/aneurysm.

Usage. At the time of this publication, the software resides in the directory /unc/rolland/vision/angio/src and is compiled for the DEC architecture. Typing just "angio" will provide the following usage message:

```
Usage: angio inputfile
        [-D]      Print debugging information
        [-t "title"]        Output image title [def "<>_blo"]
        [-a diameter (+/-)aneurysm length]
        [-f (+/-)rad.factor] fraction taper/exp (def 0)□
```

The name specified for the input_image with "angio" should be the same name as the output image from "space_curve."

Parameters. The usage message indicates the parameters that may be specified on the command line for the "angio" program.

-D instructs the program to print to the screen a host of debugging messages.

-t specifies the name of the output image. The default output image is the input image with the "_blo" suffix.

-a specifies three parameters of the vessel. The first parameter following the -a option specifies the diameter (in pixels) of the generated vessel. The second parameter determines the extent of aneurysm or stenosis (specifying 0 places no abnormality in the vessel, while aneurysms are positive percentages of the regular vessel diameter, and stenoses are negative percentages). The third parameter specifies the length of the aneurysm/stenosis in pixels. If the -a parameters are not specified, the user will be prompted upon execution of the program to enter them interactively.

-f allows the vessel to be tapered (or expanded) by some constant factor. At each position along the vessel, the diameter of the vessel is determined from the previous diameter times the quantity 1 plus this radius factor. A radius factor of -0.001 causes the vessel to be slightly tapered over its length.

Outputs. The program produces a /usr/image BYTE format image (named with the output name of the space_curve and a "_blo" suffix) which contains the simulated vessel obtained from projection through the three-dimensional volume. Finally, the parameters necessary to describe the simulated blood vessel as well as their English description are appended to the file having the suffix "_parm".

EDGE NOISE

Noise, in the form of "wiggle," may be added to the vessels to give them an even more realistic appearance. In a healthy patient, vessel walls are indeed smooth. However, in older patients or patients with vascular disease such as atherosclerosis, small plaques form on the interior wall of the vessel, causing the vessel, when imaged with contrast agent, to have a lumpy or non-smooth boundary. Software developed by Ross Whitaker of the Department of Computer Science has been applied very successfully to the images resulting from the angiogram simulation, in order to achieve this appearance. The program employs greyscale morphology methods to randomly erode or dilate the level contours in the image.

The program, "noise_image" (located at the time of this writing in /unc/puff/research/wiggle), runs on the MasPar, and requires three command-line parameters, the input image (which must be a REAL, three-dimensional image containing one slice), a parameter file, and the name of the desired output image. The parameter file should, for the generation of noise relevant to this application, contain two lines: 1) one line containing the "edge" keyword followed by two numbers, the degree of spatial correlation of the noise, and then the time step, 2) and a second line containing the "num_iterations" keyword followed by a number. Reasonable values for spatial correlations range from 0.1 to 5 (higher values are more highly correlated). The time step, the "dt" from the differential equation involved in the computation of the noise, should be set to 0.1 or 0.2. The number of iterations dictates, along with the correlation of the noise, the extent of the effect; higher iterations generates more "wiggle." An example parameter file consists of the following:

```
edge 1.0 0.2
```

```
num_iterations  100
```

APPLICATIONS

Psychophysical Analysis of Medical Image Interpretation. A primary basis for our current research has been the multiscale visual object representation (MVOR). This representation has two interrelated components, the multiscale medial axis (MMA) and the multiscale boundary properties. According to the MMA model, objects are represented by a set of loci in scale space that simultaneously define middles and widths of objects, object details, and subobjects directly from image intensities. These loci capture the way an object bends, branches and truncates, and the way it bulges and compresses as we move along its middle. Together with constants to represent the overall size, orientation, and aspect ratio, the MMA represents any object or image made of a collection of objects. In this application, we are testing whether some aspects of a medial representation for object are especially relevant to shape judgment. We measure the ability of human observers to detect a change in shape of an object embedded in a background of objects of similar shapes. More specifically, we are looking at the detection of aneurysms in computer-simulated angiograms as a function of the curvature of the medial axis at the location of the aneurysms (see Figure below). We predict that the ability to detect an aneurysm will be a function of the curvature of the medial axis of the blood vessel where the aneurysm is located and independent of the width of the blood vessel. The software described in this report was especially developed to be able to simulate realistic angiogram images with absolute truth about any potentially present abnormality. This application, originally created to test some hypotheses of the MMA, aims, on a broader level, at studying the importance of features geometry and background complexities in different medical imaging tasks.

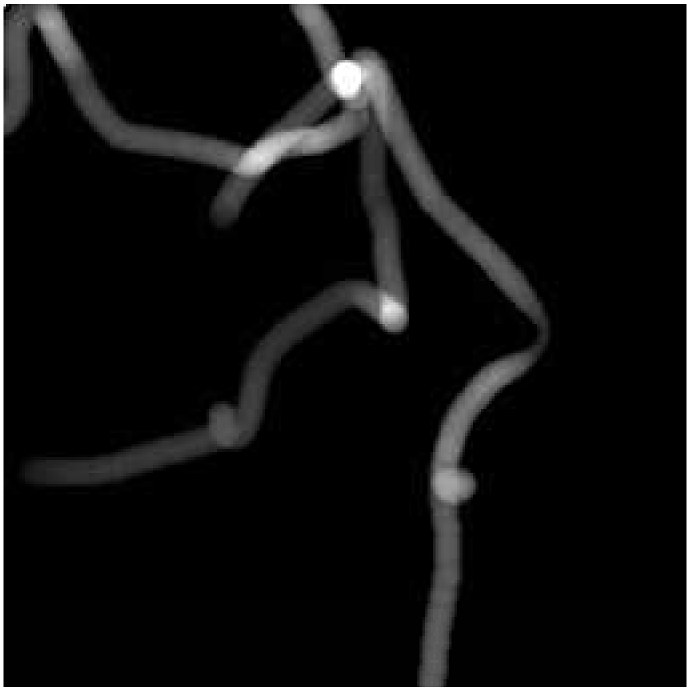


Image Simulations for Testing a Model of Image Quality. The angiogram simulation software has been utilized in generating a set of images to test the feasibility of a measure of image quality [Puff dissertation proposal]. The image quality measure incorporates the multiscale medial model of human vision in a simulation of the performance of the human observer for the task of stenosis estimation. The accuracy of the performance of the model for the task is used as the measure of the quality of the image.

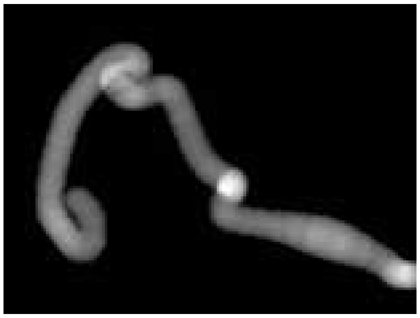
Using the angiogram simulation software, vessels containing stenoses have been generated and embedded in diagnostic x-ray images of the head and neck region to simulate an angiogram of the carotid artery (see below).

Parameters dictating the quality of the image, such as image blur and contrast in the vessel, will be varied to generate a set of experimental images. Model calculations of the extent of stenosis of the vessel in each of the images will be performed. Those images for which the model estimates the extent of the stenosis most accurately will be deemed of optimal image quality. Ultimately, the assessment provided with the model will be compared with that of human observers for the same test images.

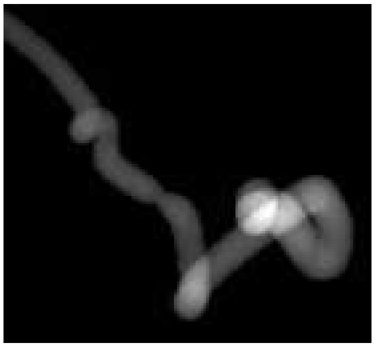
EXAMPLES

We have used this software very successfully to generate realistic angiogram simulations. The projection through the volume of the vessel, which simulates the projection of the x-ray beam through the anatomy, allows for higher intensities in the middle of the vessels, and at points of vessel cross-over or for vessel alignment perpendicular to the viewing plane.

Included below are several example results along with their corresponding parameter settings.

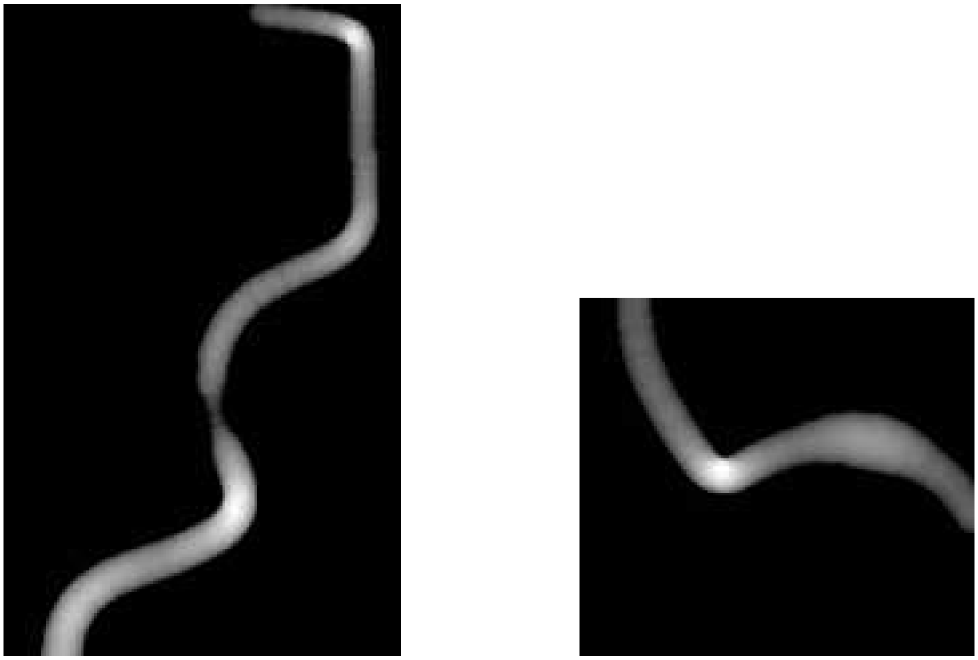


(1)



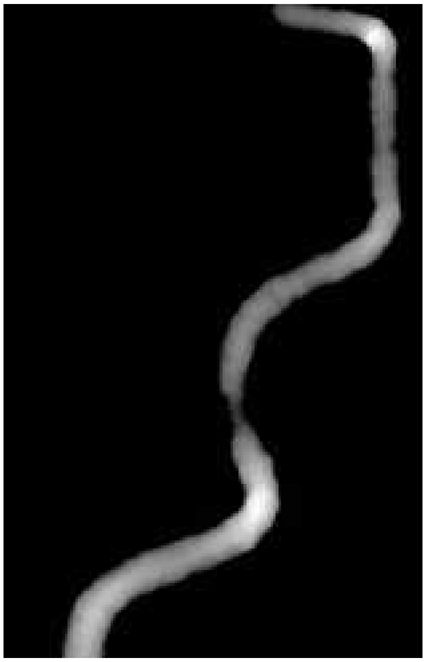
(2)

The images above were generated with an unconstrained space curve. (1) The vessel on the left has a diameter of 10 pixels, and a 50% aneurysm of length 20 pixels located in the lower right corner. (2) The vessel on the right has an initial diameter of 15 pixels which tapers by a factor of -0.001 from its starting position near the center of the image, and a 50% stenosis of length 10 pixels positioned roughly half way along the path of the vessel, near the center of the image. The images shown here were cropped from a 256^2 image.



(3) (4)

The images above were generated with a constrained space curve. (3) The vessel on the left has a diameter of 15 pixels tapering by a factor of -0.001 as the vessel moves upward, and a 60% stenosis of length 20 pixels located approximately half way up the vessel. The image is cropped from a 256^2 image with a starting position of 128, 255, 128 (in the middle of the image from left to right and in depth, and at the bottom of the image). The initial tangent vector was specified at 0,-1,0, or directly upward. The standard deviation for kappa was 0.05, with bounds of 0.001 and 0.003. The standard deviation for tau was 0.01, with bounds of -0.05 and 0.05. The length was terminated after 425 increments of the arc length. (4)The vessel on the right has a diameter of 15 pixels. Its starting position was (in the original 256^2 image) at 128,0,128. The initial tangent vector was directly downward (0,1,0). The standard deviation for kappa was 0.05, with bounds of 0.001 and 0.0035. The standard deviation for tau was 0.05, with bounds of -0.05 and 0.05. There is a 50% aneurysm of length 30 pixels at the right side of the image.



The image above was generated by adding edge noise to the image 3. The variance of the noise was 2, and 50 iterations were specified.

CONCLUSION

A description of the software used to simulate angiograms with different levels of realism has been provided. The procedure consists of specifying a 3D space curve, a region of interest for potential defects, and applying the angio program that provides the user with a 2D projection of the 3D blood vessel. Noise along the edges of the simulated 2D projection of the 3D blood vessel can also be added if necessary. Realistic angiograms can be simulated by selectively adding several blood vessels together.

ACKNOWLEGMENTS

The research reported here was supported by NIH grant #PO1 CA47982. We are indebted to John Rhoades for suggesting the rolling sphere algorithm to generate the 3D blood vessels. We are grateful to Stephen Pizer and Christina Burbeck for their encouragements during the early development of this tool. We thank Graham Gash for computer systems support.

REFERENCES

O'Neill, Barrett. Elementary Differential Geometry. Orlando: Academic Press, 1966.

Pizer, S.M., Burbeck, C.A., Coggins, J.M., Fritsch D.S., Morse, B.S. "Object Shape before Boundary Shape: Scale-space Medial Axes" UNC Department of Computer Science TR92-025, September, 1992.

Puff, Derek T. UNC Department of Biomedical Engineering Dissertation Proposal. May, 1993.

Whitaker, Ross T., and Puff, Derek T. "Boundary noise: Perturbing objects in images through grey-scale morphology." to appear.