TEXTURE COORDINATE GENERATION OF COLONIC SURFACE MESHES FOR SURGICAL SIMULATION

Josh Passenger[1], *Oscar Acosta* [1], *Hans de Visser*[1], *Sebastian Bauer*[1], *Christoph Russ*[1]

[1]The Australian e-Health Research Centre, CSIRO ICT Centre, Brisbane, Australia.

Sébastien Ourselin[1,2]

[2]Centre for Medical Image Computing, University College London, Gower St, London, WC1E 6BT, UK.

ABSTRACT

Simulation of surgical procedures requires visualisation of realistic geometric models of organs generated from 3D patient images. Significant improvements in visual realism can be obtained through appropriate surface texturing and shading. In this paper we present a novel technique for continuous surface parameterisation of colonic geometric models utilising surface voxel to centreline correspondences derived from Laplace's equation. Our technique creates continuous texture coordinates which exhibit minimal distortion and which enable us to use advanced texturing and shading techniques. Experimental results from phantom volumes and real cases are detailed.

Keywords: Surface parameterisation, colonoscopy, computed tomography, texture coordinates, surgical simulation

1. INTRODUCTION

Colorectal cancer will be the fourth most commonly diagnosed cancer and the second most common cause of cancer related death in the USA during 2007 [1]. Survival rates from colorectal cancer improve significantly with early detection, but unfortunately there are few early symptoms and risk increases from age 45 onwards. Colonoscopy is currently the best examination technique as it offers the ability to biospy and excise suspicious tissue, but the procedure is difficult to master and most training occurs on real patients. To reduce the risk associated with training on real patients, improvement of colonoscopy simulators for training and eventual certification would be of tremendous value. Existing colonoscopy simulators rate poorly for anatomic and haptic realism as well as case complexity and several studies have found them only useful for novice instruction [2]. For a colonoscopy simulator to be realistic enough to train not only novices, but also intermediate and expert physicians, it needs to incorporate a large number of highly realistic, complex cases.

In our simulation environment each case is generated from patient CT scan data and requires the construction of geometrical models of the colon. This requires several steps, including: segmentation of the colon, extraction of the centreline, creation of surface meshes, surface mesh parameterisation and UV texture coordinate calculation for realistic texturing and shading.

Techniques for surface parameterisation have been published extensively [3]. Some techniques that have been applied to colonic anatomy such as conformal mapping [4][5] are mathematically complex. Other surface parameterisation techniques that have been applied to surgical simulation such as piecewise linear mapping require special handling for seamlines generated during the mesh flattening phase [6]. The primary contributions of this paper are novel techniques for surface parameterisation of colon models and for subsequent texture coordinate interpolation which are not overtly mathematically complex and do not generate seamlines. Whereas previous work is focused on parameterisation of meshes, we first compute a surface parameterisation of our binary segmentation and interpolate that onto our surface mesh. We map each colonic surface voxel to a point on the centreline using non-intersecting correspondence trajectories, derived in an efficient manner using the Laplacian map. With the correspondences computed, we are able to compute UV texture coordinates for all surface voxels and interpolate these onto our geometric model. We provide additional novelty through a technique that successfully eliminates texture mirroring artefacts through combination of two textures where the circumference texture coordinate of one is rotated.

In Section 2 we detail the techniques used for segmentation of CT images, mesh and centreline generation and details of our correspondence and texture coordinate generation techniques. Section 3 provides experimental results on phantoms and real colon images.

2. METHODS

2.1. Segmentation of colon from Computed Tomography images

Segmentation of the colon from an abdominal 3D CT scan is achieved through smoothing of the original image using a curvature anisotropic diffusion filter [7]. A threshold filter is applied to the image to identify air and tagged fluid regions which are subsequently merged using a vertical filter [8]. As several organs can be segmented during this process, the colon must be selected from a labelled image.

2.2. Surface mesh generation

The binary segmentation is meshed using Marching Cubes [9] to generate a high density surface mesh of the colonic lumen. Subsequent Laplacian smoothing [10] and quadric decimation filters [11] are used to obtain a smooth surface mesh (Fig. 1). The smoothing operation is not volume preserving and tends to remove small surface details. Fine surface details are restored during simulation through the use of normal mapping.

2.3. Centreline calculation

A fast marching filter [12] using the Euclidean distance map as the speed term [13] is applied to a binary image generated from the

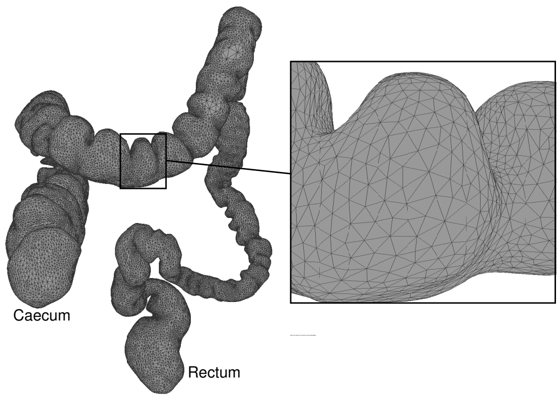


Fig. 1. A high density colonic surface mesh generated from segmented patient CT data.

smoothed and decimated mesh between points at the rectum and caecum. Back propagation [14] involving the selection of the points on each front with the greatest distance in the distance map is used to obtain a connected, voxelised centreline. A moving average filter is then used to smooth the centreline and a Catmull-Rom spline f_c is created using the smoothed centreline voxels.

2.4. Correspondences between the surface and the centreline

Let R be the resulting segmented colon, $\mathbf{S}$ the collection of external boundary voxels and $\mathbf{C}$ the computed centreline, as shown in Fig. 2.

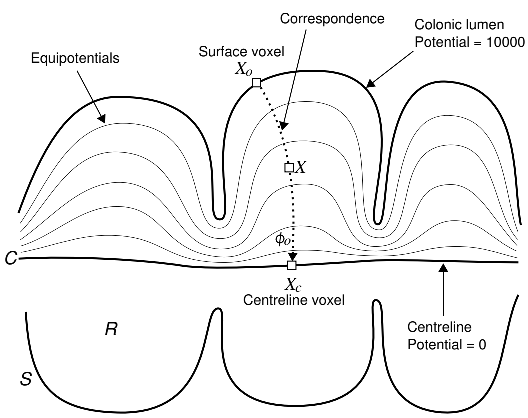


Fig. 2. Solving Laplace's equation between the colon surface and the centreline results in equipotential lines which can be used to calculate surface–centreline correspondences.

In order to obtain a unique association between a point X_o at the outer boundary and a point, $X_c \in \mathbf{C}$ at the centreline , we used the solution of the Laplace's equation ($\Delta u = 0$) over R. This is performed according to Jones's approach [15], with boundary conditions fixed at C and S.

Thus, the centreline C and outer boundary S are set to fixed potential of 0V and 10000V, respectively. The solution is a scalar field u, dividing R, into a set of equipotential layers. The normalised gradient of this scalar field is then computed, giving for each voxel a unit vector $\vec{T}$ perpendicular to the equipotential layer on which it sits as:

$$\vec{T} = \frac{\nabla u}{\mid \nabla u \mid} \tag{1}$$

A family of nonintersecting curves connecting the centreline with the surface can be constructed, following the unit vector field. We refer to these curves as correspondence trajectories. Thus, starting from any point X_o on the outer boundary, there is a unique path to a point on the centreline X_c. To avoid the explicit construction of the trajectories, Rocha et al. [16] proposed an Eulerian approach building upon the method presented by Yezzi and Prince [17]. We improved the implementation of this method, making it general for 3D anisotropic volumes. In this approach, a partial differential equation (PDE) is solved iteratively for finding the correspondences between S and C. Let $\phi_0 : \mathbf{R}^3 \rightarrow \mathbf{R}^3$ be defined as the correspondence function that maps a point $x \in R$ to the centreline. Since ϕ_0 must remain constant along the direction given by the tangent field $\vec{T}$, we have:

$$(\nabla \phi_0).\vec{T} = 0 \tag{2}$$

with the boundary condition:

$$\phi_0(\mathbf{x}) = \mathbf{x}, \forall \mathbf{x} \in \mathbf{C} \tag{3}$$

The solution of equation (2) in $\mathbf{R}^3$ is computed in a similar way to the equations for distances in [17]. In our implementation, these equations have been generalised for anisotropic images. Thus, given a 3D grid with voxel spacing a, b and c, in the x, y and z directions respectively:

$$\begin{aligned}\phi_0(x,y,z) \quad &= \tfrac{1}{bc|T_x|+ac|T_y|+ab|T_z|}\Big[bc\,|T_x|\,\phi_0(x \mp a, y, z) \\ &+ ac\,|T_y|\,\phi_0(x, y \mp b, z) + ab\,|T_z|\,\phi_0(x, y, z \mp c)\Big]\end{aligned} \tag{4}$$

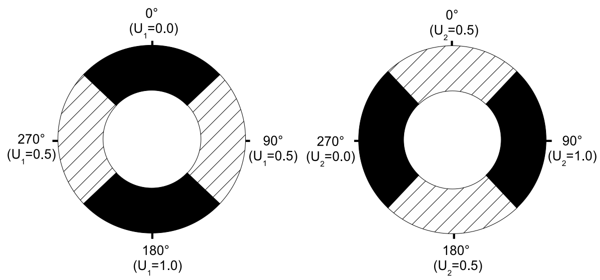
where:

$$x \mp a = \begin{cases} x - a, & T_x > 0 \\ x + a, & T_x < 0 \end{cases}$$

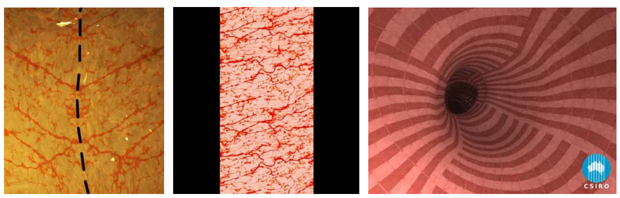
and similarly for y, z, b and c.

2.5. Vertex texture coordinates calculation

Due to limitations in the way that OpenGL interpolates texture coordinates between vertices, continuous texturing of tubular structures is traditionally achieved through mirroring of circumference texture coordinates from 0.0 to 1.0 and back to 0.0. Unfortunately, this leads to texture mirroring artefacts as shown in Fig. 3b. If circumference texture coordinate mirroring is not performed, adjacent vertices at the edge of the circumference texture boundary where 0.0 and 1.0 meet, have large differences in texture coordinates. OpenGL creates texture interpolation artefacts at these discontinuities. Our solution to this problem is to apply two textures on top of each other, each with the traditional mirroring of circumference texture coordinates from 0.0 to 1.0 and back to 0.0, but with the circumference texture coordinates for one of the textures rotated over 90^o (Fig. 3a). We use textures with black edges and detail running in a strip down the middle of the texture (Fig. 3c). The detail strips of the two textures are designed to tile seamlessly with each other. As the circumference texture coordinates for one of the textures are shifted, the black strips of one texture coincide with the detail strip of the other. The mirroring artefacts are now hidden since all mirrored texture boundaries are pure black and in our pixel shader these regions do not contribute to the final pixel colour. The result is a continuously textured circumference that exhibits no mirroring artefacts (Fig. 3d).



a



b c d

Fig. 3. a) Rotated U texture coordinates. b) Artefact due to texture coordinate mirroring. c) Textures used to avoid mirroring. d) Two textures with rotated U texture coordinates.

The surface UV texture coordinates are calculated using the following approach. Given a voxel X_0 on the binary image surface and the correspondence ϕ_0 between the surface and centreline voxels, we calculate the centreline voxel X_c for a particular surface voxel through:

$$X_c = \phi_0(X_0) \tag{5}$$

The V texture coordinate (running along the length of the colon), is then calculated for each surface voxel by solving for the distance along the centreline spline f_c where:

$$f_c(C) \in [0, 1] \tag{6}$$

$$V = f_c(\phi_0(X_0)) \tag{7}$$

$$V = f_c(X_c) \tag{8}$$

To enable circumference texture coordinate rotation we calculate the texture coordinate U in the range [0,2]. First, a local orthonormal basis $O(\hat{X}, \hat{Y}, \hat{Z})$ is constructed using the propagated up-vector $\hat{Y}$ of the centreline spline f_c at the centreline position X_c, the heading vector $\hat{Z}$ of the spline and a vector $\hat{X}$ calculated by $\hat{Y} \times \hat{Z}$ (Fig. 4, Fig. 5a). The texture coordinate U is then calculated by:

$$B = \frac{acos(\hat{Y} \cdot \hat{N})}{\pi} \tag{9}$$

$$U = \begin{cases} B, & (\hat{Y} \times \hat{N}) \cdot \hat{Z} > 0 \\ 2.0 - B, & (\hat{Y} \times \hat{N}) \cdot \hat{Z} < 0 \end{cases} \tag{10}$$

in which $\hat{N}$ is the normalised vector from X_c to X_0. In the vertex shader during rendering, the U texture coordinate is converted from the range [0, 2] back into the mirrored configuration [0, 1, 0] for the two textures as follows (Fig 3a):

$$U_1 = \begin{cases} U, & U < 1 \\ 2.0 - U, & U \geq 1 \end{cases} \tag{11}$$

$$U_2 = \begin{cases} U + 0.5, & U \leq 0.5 \\ 1.5 - U, & 0.5 < U < 1.5 \\ U - 1.5, & U \geq 1.5 \end{cases} \tag{12}$$

Texture coordinates for the vertices of the surface mesh are then derived through inverse distance interpolation of the surface voxel texture coordinates within a cut-off distance. Some local texture artifacts are expected at the rectum and caecum due to our reliance on the centreline for U texture coordinate calculation.

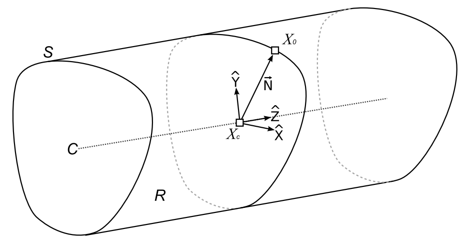
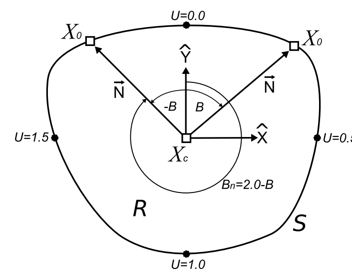
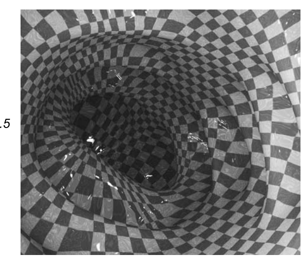


Fig. 4. For each surface voxel, a local orthonormal basis $O(\hat{X}, \hat{Y}, \hat{Z})$ is created at the corresponding centreline voxel.



a



b

Fig. 5. a) Calculation of the U texture coordinate using surface–centreline correspondence. b) Colonoscopy view with a checkerboard texture on a real colon mesh.

3. EXPERIMENTS AND RESULTS

The proposed algorithm was applied to a set of synthetic phantoms (Fig. 6) and to nine patient 3D CT images.

3.1. Synthetic phantoms

The two synthetic phantoms (Fig. 6) were designed to demonstrate the ability of our technique to generate surface parameterisations for curved objects and objects with varying diameters. As can be clearly seen in the Fig. 6b and Fig. 6d the resulting texture coordinates exhibit minimal stretching and no tearing artefacts.

3.2. Real CT data

The algorithm was applied to 9 real CT 3D images which were provided courtesy of Dr. Richard Choi, Virtual Colonoscopy Center, Walter Reed Army Medical Center. The acquisition protocol is detailed by Pickhardt [18]. The CT images used were 512×512 with a varying number of slices (> 400) with either $0.68 \times 0.68 \times 1.0$ mm or $0.74 \times 0.74 \times 1.0$ mm anisotropic voxels. For all of these cases our algorithm successfully computed viable texture coordinates for the generated colonic lumen meshes. A comparison with a real colonoscopy is presented in Fig. 7. An internal colonoscopy

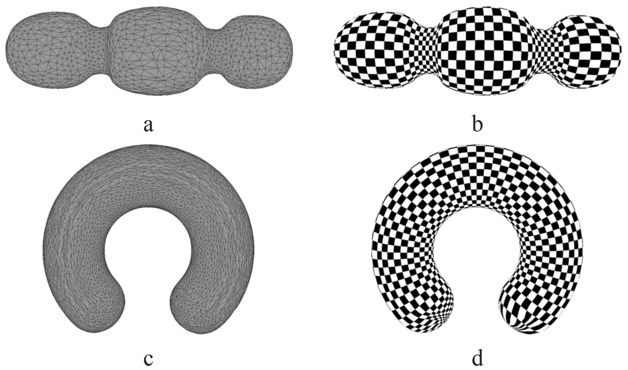


Fig. 6. *a*) Surface mesh of a colon-like phantom. *b*) The colon-like phantom textured. *c*) Surface mesh of the torus phantom. *d*) The torus phantom textured.

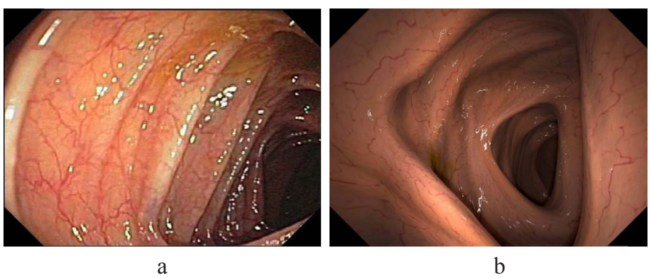


Fig. 7. *a*) Real colonoscopy. *b*) Our texturing technique applied to a colonic mesh with real-time OpenGL shading.

view of a real colon mesh with a checkerboard texture is shown in Fig. 5b. Apart from the previously noted artefacts at the rectum and caecum, the technique is very robust and handles difficult cases such as strictures and sharp bends. The final textured results have been verified for visual realism by an expert endoscopist.

4. CONCLUSION

We have developed a novel technique for producing UV texture coordinates from complex, irregular colonic surface meshes using surface to centreline voxel correspondences. Compared to existing flattening techniques, our technique avoids visible seams and provides a continuous solution suitable for real-time texturing and shading. We have implemented a novel technique that uses U texture coordinate rotation in the vertex shader that successfully avoids texture mirroring artefacts through multi-texturing. We have demonstrated our technique on phantoms and real colon meshes and are using the system to process meshes for our colonoscopy simulator. Future work includes resolving minor artefacts evident at the rectum and caecum.

5. ACKNOWLEDGEMENTS

We would like to thank the CSIRO National Preventative Health Flagship for funding support.

6. REFERENCES

[1] A. Jemal and R. Siegel, "Cancer statistics, 2007," *CA: A Cancer Journal for Clinians*, vol. 57, no. 1, pp. 43–66, 2007.

[2] L.B. Gerson, "Evidence-Based Assessment of Endoscopic Simulators for Training," *Gastro Endosc*, vol. 16, no. 1, pp. 489–509, 2006.

[3] Michael S. Floater and Kai Hormann, "Surface parameterization: a tutorial and survey," in *Advances in multiresolution for geometric modelling*, pp. 157–186. Springer Verlag, 2005.

[4] W. Hong, X. Gu, and *et al.*, "Conformal virtual colon flattening," in *SPM '06: Proceedings of the 2006 ACM symposium on Solid and physical modeling*, 2006, pp. 85–93.

[5] S. Haker and S. Angenent, "Conformal surface parameterization for texture mapping," *IEEE TVCG*, vol. 6, no. 2, pp. 181–189, 2000.

[6] R. Paget, M. Harders, and *et al.*, "A Framework for Coherent Texturing in Surgical Simulators," *Pacific Graphics '05*, pp. 112–114, 2005.

[7] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *IEEE TPAMI*, vol. 12, no. 7, pp. 629–639, July 1990.

[8] M. Sato and S. Lakare, "An automatic colon segmentation for 3D virtual colonoscopy," *IEICE TIS*, vol. e84-d, no. 1, pp. 201–208, 2001.

[9] W. Lorensen and H. Cline, "Marching cubes: A high resolution 3-d surface construction algorithm," *Computer Graphics*, vol. 21, pp. 163–169, 1987.

[10] D. Field, "Laplacian smoothing and Delaunay triangulations," *Communications in Applied Numerical Methods*, vol. 4, pp. 709–712, 1988.

[11] M. Garland and P. Heckbert, "Surface Simplification Using Quadric Error Metrics," *Computer Graphics*, vol. 31, no. Annual Conference Series, pp. 209–216, 1997.

[12] J. Sethian, "A Fast Marching Level Set Method for Monotonically Advancing Fronts," in *Proc. Nat. Acad. Sci.*, 1996, vol. 93, pp. 1591–1595.

[13] R.L. Van Uitert, R.M. Summers, and *et al.*, "Automatic Correction of Level Set Based Subvoxel Precise Centerlines for Virtual Colonoscopy Using the Colon Outer Wall," *MedImg*, vol. 26, no. 8, pp. 1069–1078, August 2007.

[14] T. Deschamps and L.D. Cohen, "Fast extraction of minimal paths in 3D images and applications to virtual endoscopy," Tech. Rep. 0026, Les cahiers du Cérémade, June 2000.

[15] S.E. Jones and B.R. Buckbinder, "Three-dimensional mapping of cortical thickness using Laplace's equation," *HBM*, vol. 11, pp. 12–32, 2000.

[16] K.R. Rocha, A.J. Yezzi, and J.L. Prince, "A hybrid Eulerian-Lagrangian approach for thickness, correspondences, and gridding of annular tissues," *IEEE Transactions on Image Processing*, vol. 16, no. 3, pp. 636–648, 2007.

[17] A.J. Yezzi and J.L. Prince, "An Eulerian PDE approach for computing tissue thickness," *IEEE TMI*, vol. 22, pp. 1332–1339, 2003.

[18] P.J Pickhardt, Choi R.J., and *et al.*, "Computed Tomographic Virtual Colonoscopy to Screen for Colorectal Neoplasia in Asymptomatic Adults," *New Engl J Med*, vol. 349, no. 23, pp. 2191–2200, 2003.