# Model-based Solid Texture Synthesis
# for Anatomic Volume Illustration

Ilknur Kabul[1], Derek Merck[1], Julian Rosenman [2] and Stephen M. Pizer[1]

[1] Department of Computer Science, UNC Chapel Hill, USA     [2] Department of Radiation Oncology, UNC Chapel Hill, USA

**Abstract**
*Medical illustrations can make powerful use of texture synthesis to convey information about anatomic structures in an attractive, effective and understandable way. Current visualization methods are not capable of conveying detailed information about the orientation, internal structure, and other local properties of the anatomical objects for a particular patient because imaging modalities such as CT or MRI do not capture this information. In this paper, a new anatomical rendering method that utilizes model-based synthesis of 3D textures is proposed in order to distinguish and illustrate different structures inside the model. The goal of our volume illustration approach is to visualize structural information by considering directions and layers in synthesizing high-quality, high-resolution solid textures. Our method uses medial coordinates of 3D models and 2D exemplar textures to generate solid textures that change progressively in orientation and material according to the local orientation and transition information implicit in the anatomic region.Discrete medial 3D anatomical models ("m-reps") provide the orientation field and texture variation maps inside image regions. In our paper, we demonstrate the robustness of our method with a variety of textures applied to different anatomical structures, such as muscles, and mandible.*

## 1. Introduction

The main goal of medical visualization is to create understanding of the information present in the given data. Such illustrations play a major role in many different areas such as medical education, surgical training, and procedure planning. Textbook illustrations of different anatomical structures are commonly used in medical education. These illustrations are effective in explaining the structural properties (e.g., shape, size, appearance, etc.) of the organs and the spatial relations between them. Many of these illustrations use textures both as a means to distinguish structures from one another by appearance, as well as to convey local structural detail such as orientation and scale. For example, Fig. 1 shows an illustration of a slice through the upper abdomen and a corresponding magnetic resonance image (MRI) [LE05]. In this figure, the textures help to visualize details of the organs.

Many anatomical objects have slowly changing variations in local material properties, such as material type, scale and orientation, over the surface of the model and inside the model. In hand-drawn medical illustrations commonly used for anatomy reference, many anatomical struc-
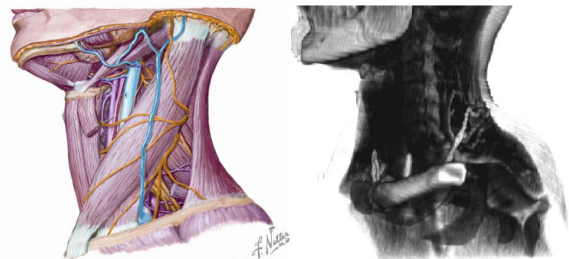


**Figure 1:** *A scientific illustration [Net09] shows more general information than the corresponding*

tures are shown with appearances that suggest features arranged along, across and around the object, and inside the object. For example, the layered muscle linings of the duodenum and the stomach have respective directional components along and around the object. In addition, the representation of the layers inside the object should be considered in the visualization of anatomical organs. For example, bone has hard exterior, soft interior, and soft marrow interior regions.

In order to provide this behavior (such as the ones in medical textbooks) for the illustrative visualization of patient

data sets in a 3D environment, we present a method for synthesizing patient specific progressively variant solid textures for segmented patient data. Our method produces solid textures that are constrained by a 3D vector field, as well as by material information specific to the patient's anatomical structures. To capture model-based directions, material transitions and depth of layer information consistently inside the model for the segmented patient data, our texture synthesis approach utilizes a model-based coordinate system.

There are two main contributions of our approach:

1. We propose a consistent 3D volume vector field computation approach using model-based coordinate systems. Using these vector fields, we can synthesize orientationally appropriate model-based surface and solid textures in order to visualize the appearance of models such as muscle. In this paper, the focus is on solid textures. This includes creating depth-varying regionally appropriate solid textures inside the object for the anatomical structures such as bone, which have distinct exterior and interior appearances.
2. We present a method for synthesizing model-based progressively variant solid textures. Our method considers orientation, and material information for guiding the texture synthesis.

We demonstrate the results of our approach on several examples including thyroid, sternocleidomastoid muscle (scm), and mandible. The method has been shown to work for the anatomical organs in head and neck region. However, our method is extensible to different instances of many organs, such as muscles. The goal of our approach is eventually be able to fill a space with synthesized solid textures to create an illustrated look for a particular patient. In our method, adding texture on top of standard volume or surface rendering helps to restore the information lost by the imaging device.

The rest of the paper is organized as follows: In Section 2 we discuss related work in solid texture synthesis, guided solid texture synthesis and texture synthesis for anatomic illustration. In Section 3 we present the overview of our approach. In Section 4 we give a brief overview of the particular model-based coordinate system we use to compute the 3D vector field and regions at certain depths. We describe our solid texture synthesis approach in detail in Section 5. We present our results and possible applications of our approach in Section 6.

## 2. Background

In this section we give a brief overview of prior work in exemplar-based solid texture synthesis, guided solid texture synthesis and texture synthesis for medical illustrations.

### 2.1. Solid Texture Synthesis

Exemplar-based solid texture synthesis has been the subject of considerable interest in the recent years. It denotes the process of constructing a large 3D volume texture from a small 2D or 3D sample texture by considering the content inside the sample. The first work in this direction was presented by [Wei03]. The key idea in this method is to consider a 2D exemplar texture for each orthogonal direction (basically x, y and z directions) in neighborhood search to find the best match and using local optimization to converge to the best color. The main problems in this approach are blurring and artifacts in synthesizing solid textures that have high structural information. Later [JDR04] presented a stereology-based technique for the textures that are composed by models for different particle shapes. This approach produces good results, but it is restricted to the solid textures that have particle shapes. In the method presented in [QY07] a volumetric texture is synthesized using Basic Gray Level Aura Matrices (BGLAMs), which capture the input texture's co-occurrence probability distributions of gray levels. The main drawback of this approach is that it only works for greyscale images.

[KFCO*07] extended the method presented in [Wei03] for synthesizing homogeneous solid textures from 2D exemplars by texture optimization and histogram matching techniques. Although this approach produces good results for isotropic cubic solid textures, it does not handle anisotropy and spatial variation of textures inside the model.

### 2.2. Guided Solid Texture Synthesis

The purpose of progressively variant solid textures is to illustrate models in 3D using their material and shape properties. In order to synthesize these textures, the approach needs to be guided by the characteristics that are specific to a model. The approach presented by [ONOI04] provides a way to create quasi-solid textures for illustrating object interiors. This approach utilizes user input to synthesize 2D textures on the cross section every time the model is cut, leading to inconsistency among different cross-sections. In [TOII08] a method that builds progressively changing oriented solid textures by repeatedly pasting 3D texture exemplars within a tetrahedral mesh is presented.In both of these approaches the user interface that they provide allows a user to manually input quasi-medial coordinates to identify the regions inside the model. In our approach, we get this information from model based coordinate systems.

There are also some methods proposed for synthesizing oriented solid textures. [CYZ*09] proposed a method for visualization of muscle based on diffusion tensor images (DTI) using oriented solid texture. They utilized a vector field computed from the DTI data as a guide to synthesize solid textures.

In order to synthesize oriented solid textures, which depicts the essential shape characteristics of the model, we need to have a 3D volumetric vector field. [TAI*08] proposes sketch-based approaches for generating volumetric vector

field. In this approach, the user specifies the strokes and the layers by utilizing a user interface, which makes this method unsuitable for our target problem of patient-specific synthesis. Therefore, our paper includes a method for computing such 3D vector fields and layers automatically.

### 2.3. Solid Texture Synthesis for Medical Visualization

Volume illustrations became popular over the past decade for its effectiveness in illustrating the material and structural features of a data set while emphasizing important details. One approach is to generate volumetric illustrations by synthesizing volume data with model-specific textures and different rendering techniques. [DC05] proposed a method for synthesizing oriented volumetric solid textures via a patch-based solid texture synthesis algorithm that uses the orientation field extracted from the CT and user-provided 3D exemplar textures. Later, [LE05] proposed a method to generate illustrative 3D textures using Wang Cubes. The main disadvantages of this method are that the textures are not region-appropriate but rather are homogeneous throughout the volume, nor do they reflect scale and orientation information. In addition, obtaining 3D texture samples is not easy to capture, and tiling of the 3D textures leads to repetitive patterns.

### 3. Overview of Our Method

Our method synthesizes progressively changing solid textures specific to a model of an anatomical object by considering the orientation and variation of the textures inside the model. The purpose of our method is to obtain illustrative renderings of patient-specific anatomical structures. The input to our system consists of a 3D triangular mesh model, a model-based coordinate system for the model (in our method obtained from a medial model of the object), and a predetermined set of 2D exemplar textures. The output is a solid textured model. Thus, our model-guided texture synthesis is a post-segmentation process.

Our method consists of two main steps. In the first step we construct a 3D vector field using the model-based coordinate system to guide the texture and its variation(including depth of layer). In the second step solid textures are synthesized according to this guidance. Fig. 2 shows the pipeline of our approach.

In our method the 3D guidance vector field is obtained from a medial representation called 'm-reps'. M-reps nicely provide a 3D along-object, across-object, and through-object parameterization on and within the model. Details of this approach are explained in Section 4.

We consider two important features in the texture synthesis step: the guidance vector field, and the variation of the textures inside the model. Depending on the anatomical structure, the texture may vary along the depth of the model or along the surface and inside of the model(Fig. 3). For example, the mandible has many tissue types: osseous tissue,
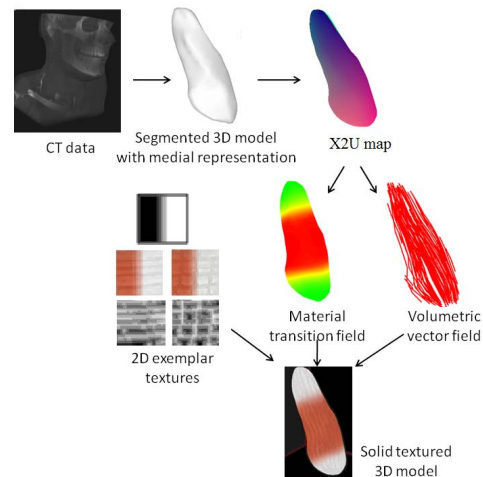


**Figure 2:** *Pipeline of our approach*

marrow, endosteum, and periosteum. Osseous tissue makes up bone, also called bone tissue, and endosteum and periosteum line the outside and inside surfaces, respectively, of the bony tissue. This variation information is obtained from the model-based coordinate system from the first stage and is used to guide the solid texture synthesis.

For some of the anatomical models, the textures look different depending on the orientation of the clipping plane. For example, the texture on the outside of the sternocleidomastoid muscles (scm) looks like stream of lines, whereas when you cut the scm, you see circular blobs textures. In order to obtain this effect in our visualization, we again use the 3D model based guidance information computed in stage 1.
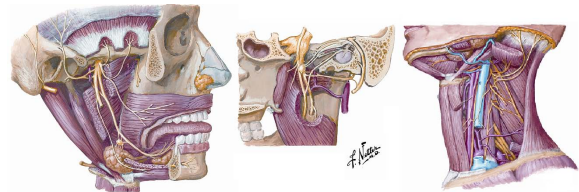


**Figure 3:** *Sample illustrations from [Net09]*

**Notation:** In the following sections $e$ denotes the input exemplar texture, $ef$ is the feature image (e.g. signed distance field) of $e$ and $et$ stores transition of the material information (see Section 5 for explanations of $ef$ and $et$). Also, $s$ refers to the synthesized solid texture. Three model-based coordinates at each position in the object are defined as: along-object ($u$), around-object ($v$) and through-object ($\tau$) coordinates. Three special orientations at these positions are: $du = \nabla u$, $dv = \nabla v$, $d\tau = \nabla \tau$; these three are not necessarily orthogonal in x,y,z space. The neighborhood of the voxel $w$ in the slice perpendicular to the i-th spatial axis (i=x,y,z) is represented by $s_{w,i}$. The neighborhoods for the exemplar texture and for the synthesized solid texture at voxel $w$ are denoted by $e_w$ and $s_w$, respectively. For region based textures,

*st* denotes the 3D transition field inside the model, which is computed using model parameterization.

## 4. Model-based Coordinate System

The problem of guided solid texture synthesis for anatomical structures is where to get the guidance. As discussed earlier, many textures on or within anatomical organs are oriented along or across the model, and they vary progressively depending on their position with respect to model. Thus, we propose a method for computing the along-object ($u$), around-object ($v$) and through-object ($\tau$) coordinates for any point in world space within each region inside the model in the scene (Fig. 4). These coordinates should provide correspondence between different instances of the anatomical models. That is, corresponding points in different instances of an anatomic object, such as an anatomical structure in different patients, should have the same $u,v,\tau$ values.

M-reps provide the three desired model-based coordinates ($u,v,\tau$). They can also be used to obtain consistent orientation fields, and region classifications for different instances of the anatomic objects. So called "'single figure slabular"' m-reps [SP08] represent 3D objects by a single continuous medial sheet and a set of spokes (Fig. 1). The medial sheet is the skeletal surface of the object, lying midway between opposing surfaces of the object. The medial sheet, M, of a 3D object is parameterized by ($u,v$), where $u$ and $v$ take on the index numbers for sample positions along and across M. At each ($u,v$) a spoke extends from each side of the medial sheet to the object boundary. Position through the object is given by the parameter $\tau$, which is the fraction of the spoke from the medial sheet to the boundary. Comparing $\tau$ to 1 distinguishes the inside of an object from the outside. The medial sheet can transfer its coordinates to the object boundary using the fact that the m-rep implies the boundary. The coordinates $u$ and $v$ parameterize this boundary.
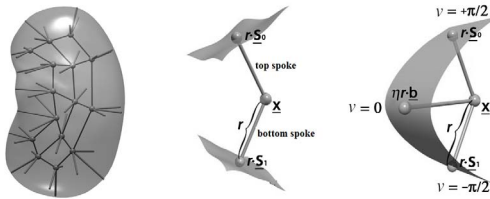


**Figure 4:** *Left: a single figure m-rep for a kidney and the object boundary implied by it. Middle: an internal medial atom. Right: a crest atom.*

Image segmentation methods that produce m-reps are described in [SP08]. [Mer09] describes the power of m-reps in volume visualization.

### 4.1. Volumetric Model Coordinate Generation

Each boundary point carries a normal in the direction of the spoke at that position and two linearly independent directions in the tangent plane to the boundary corresponding to moving in the medial $u$ and $v$ coordinates respectively. Providing the $uv$ coordinates of each vertex on the boundary allows our algorithm to compute model-based coordinates inside the model.

The required coordinates from world to object coordinate system (X2U map) stores an object coordinate ($uv\tau$) for each world-space (xyz) triplet inside the model. This coordinate system is precomputed and stored as a lookup table for each model.

In order to compute X2U map for a model, a fast coordinate scan-conversion algorithm is used. This method repeatedly rasterizes a large number of onion skins color-coded with ($uv\tau$) coordinates and then clips it to individual planes. By combining these planes into a volume, ($uv\tau$) coordinates at every voxel inside the model are obtained and stored as a look up table. The details of this approach can be found in [Mer09].

### 4.2. Model-based Guidance Field Computation

Beginning with a segmented model and the model-based coordinates, storing the X2U map has great advantages over other methods which encodes only 3D distance information from boundary, since this approach allows automatic computation of 3D model-based orientation information ($du,dv,d\tau$), and material transition information ($st$) for illustrative purposes.
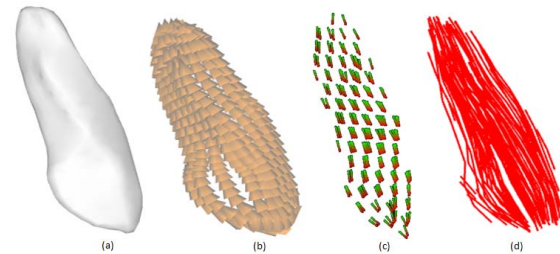


**Figure 5:** *Along object volumetric vector fields computed for scm*

1. **3D volumetric vector field:** In our method, vector fields along, across, and through the object is derived automatically from the gradient of the $uv\tau$ coordinates. Later, a Laplacian smoothing operation is applied to the vector fields in order to obtain smooth streamlines for texturing purposes.
2. **3D material transition field:** Depending on the model being considered, a thresholding scheme is applied to the values in X2U map. For instance, for the models in which the material is changing according to the depth of field (i.e., bone), a thresholding scheme is applied on the depth information that is provided by the parameter $\tau$. By using the fact that $\tau$ is zero at the medial sheet and 1 on

the boundary, the regions are classified by setting threshold values for $\tau$ (e.g., $0 < \tau < 0.5$ is the marrow region, $0.5 < \tau < 0.7$ is the interior bone region, and $0.7 < \tau < 1$ is the exterior bone region). After regions are classified, this information is used to compute a smooth transition field, $st$, for each voxel inside the model.

## 5. Solid Texture Synthesis

Given a surface model with its medial representation and 2D exemplar textures, our method first computes the model based vector fields and region masks, and then it uses them to synthesize model-based progressively changing solid textures inside the models. The 2D input exemplar textures suitable for synthesizing anatomic structures are taken from illustrated sources such as [Net09] or from specifically designed anatomic texture catalogs.

Our solid texture synthesis method is based on the basic non-parametric exemplar-based techniques used in [KFCO*07], in which uniform cubic solid textures are synthesized from 2D exemplar textures. The major contribution of our method is providing an extension to this method to efficiently support object-specific progressively changing textures. In addition, we add model-based texture synthesis control to restrict the appearance of the textures for clipping planes that are rotated based on the orientation of medial axis. As opposed to the approach presented in [TOII08], control of texture variation and texture orientation is accomplished by generating the textures "in-place" for a model in world space instead of in model space. This approach prevents the possible seams that might be caused by bending and twisting a standard texture cube according to a particular model.

### 5.1. Iterative Solid Texture Synthesis

The approach presented in [KFCO*07] is a two phase optimization method that tries to minimize the sum of differences under the $L_p$ norm between each local neighborhood $s_{w,i}$ (i=x,y,z) of a voxel $w$ in the output texture and a corresponding neighborhood $e_{w,i}$ (i=x,y,z) in the 2D example texture $e$. Initially the colors are randomly generated for each voxel inside the cube from the 2D examplar textures. Then in the first stage the matching neighborhood of each voxel is determined by optimizing the energy function $E_t$:

$$E_t(\mathbf{s}; \{\mathbf{e}\}) = \sum_w \sum_{i \in \mathbf{x,y,z}} ||\mathbf{s}_{w,i} - \mathbf{e}_{w,i}||^r \quad (1)$$

In the second stage the best matching neighborhood from the exemplar texture is searched for each voxel. The details of this approach can be found in [KFCO*07].

### 5.2. Vector Guided Solid Texture Synthesis

In order to synthesize solid textures that are oriented along the model based vector field, we extend the neighborhood

search phase of the synthesis algorithm. In this phase of the approach, the orthogonal neighborhoods $s_{w,i}$ (i=x,y,z) are rotated in 3D space around their center by a rotation matrix $R_w$, which is computed by considering the vectors $du_w$, $dv_w$, and $d\tau_w$ at that voxel $w$.

$$R_w = \begin{bmatrix} du_w^x & dv_w^x & d\tau_w^x \\ du_w^y & dv_w^y & d\tau_w^y \\ du_w^z & dv_w^z & d\tau_w^z \end{bmatrix}$$

$$sr_{w,i} = R_w * s_{w,i}$$

Here, the rotated neighborhoods are denoted by $sr_{w,i}$ (i=x,y,z). In our case, the pixels in $sr_{w,i}$ (i=x,y,z) are resampled from that of synthesized texture s.

---

**Algorithm 1** Guided Solid Texture Synthesis

---

$e_w^0 \leftarrow$ **random neighborhood in e**
$et_w^0 \leftarrow$ **random neighborhood in et**
$R_w \leftarrow$ **Compute rotation matrix using** $du_w, dv_w$ **and** $d\tau_w$

**for** $n = 0$ to N **do**
  $s_w^{n+1} \leftarrow$ **arg min** $E_t(s_w; e_w^n)$
  **for** $i = (x,y,z)$ **do**
    $sr_{w,i}^{n+1} \leftarrow$ **Rotate** $s_{w,i}$ **using** $R_w$
    $str_{w,i}^{n+1} \leftarrow$ **Rotate** $st_{w,i}$ **using** $R_w$
    $e_{w,i}^{n+1} \leftarrow$ **Find nearest neighbor of** $sr_{w,i}^{n+1}$ **in** $e_i$
    **which satisfies** $||et_{w,i}^{n+1} - str_{w,i}^{n+1}|| < threshold$
  **end for**
  **if** $e_w^{n+1} = e_w^n$ **then**
    $s_w \leftarrow s_w^{n+1}$ **break**
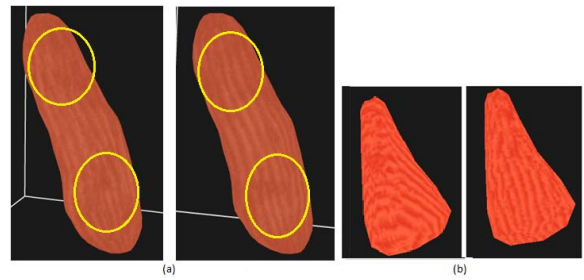  **end if**
**end for**

---



**Figure 6:** *(a) Solid textures synthesized for scm (Neighborhoods are rotated by considering (left)Only du (right) Both du and dv); (b) Comparison of [CYZ*09]'s approach with our method: (left) [CYZ*09]'s approach (right) our approach*

Our approach for synthesizing oriented solid texture is an improvement over the method presented in [CYZ*09], since we avoid the misalignment of the structural feature of the texture along the vector field by rotating the neighborhoods in 3D space using the rotation matrix that is computed by considering both *du* and *dv*. Fig. 6.a illustrates that using

both $du$ and $dv$ in the computation of the rotation matrix improves the result. In [CYZ*09], the neighborhood for each plane is rotated in 2D space by considering the projection of the orientation vector on that plane (Fig. 7). Using this method, the continuity of the structures cannot be preserved and the structural features of texture may not follow the guidance vector field. However, in our approach we can preserve the continuities since the neighborhoods are rotated by considering model based smooth vector fields. Fig. 6.b illustrates the effects of our algorithm compared to the approach presented in [CYZ*09]. For both illustrations, same 2D example textures are used in texture synthesis. For the first illustration, only the along object vector $du_w$ is considered in texture synthesis for rotating the neighborhoods using the approach presented in [CYZ*09]. In the second illustration, the results are obtained by using our method.
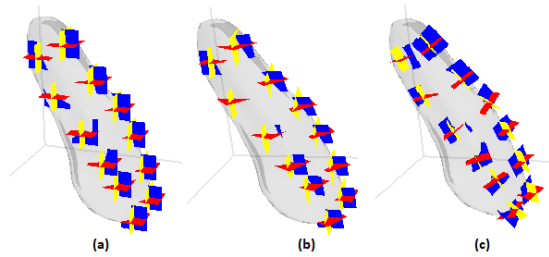


**Figure 7:** *Comparison of neighboorhoods in solid texture synthesis (a) Orthogonal neighboorhoods (b) Neighborhoods in [CYZ*09] (c) Neighboorhoods computed using our approach*
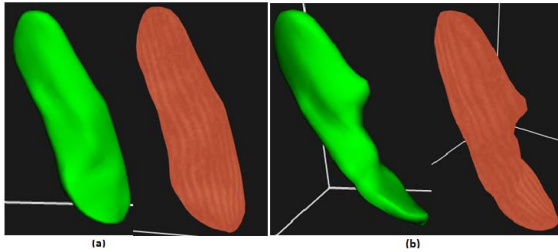


**Figure 8:** *Solid textures synthesized for two different scm models.*

The synthesis order of the voxels has a significant effect on the output texture's quality. Most algorithms such as [KFCO*07] grow a synthesized patch by selecting the voxels randomly or sequantially in the search phase. This approach is not appropriate in oriented solid texture synthesis since the structural features of the textures may not be preserved along the orientation field. In order to solve this problem, we adapt the method presented in [ZZV*03] for solid textures. We first compute streamlines from the 3D vector files. Then, the texture is synthesized by considering the individual streamlines in deciding the order of voxels to be considered in the neighborhood search. Our algorithm is also consistent for different instances of the same model. By

using our method, two different scm models with the same medial representation are textured automically(Fig. 8).

### 5.3. Material Guided Solid Texture Synthesis

Progressively variant solid textures need to be synthesized for anatomical organs, such as the mandible and the scm (Fig. 3). For example, the mandible has different tissue types inside which changes along the depth of layer. On the other hand, the scm has homogeneous tissue strands inside it but attaches to the bone at each end by tendons, which look like white muscle fibers. In order to imitate these features in our illustrations, the oriented solid texture synthesis method outlined above is further extended by considering model specific material constraints in choosing region of the exemplar texture that is going to be used during texture synthesis. As in [ZZV*03], we use a 2D exemplar texture that varies in color or material. Each exemplar texture has a feature image $ef$ and a transition image $et$. The feature image stores the structural properties of the textures, such as signed distance field obtained from binary mask of the image, and the transition image stores the transition of color or material information in the texture. These textures can be either designed by using imaging tools or other techniques [ZZV*03].

In the neighborhood search and optimization phase of the solid texture synthesis, only the exemplar pixels that satisfy the condition given in 2 are considered for computing the color value in voxel $w$.

$$||\mathbf{st}_{w,i} - \mathbf{et}_{w,i}|| < threshold \qquad (2)$$

Synthesizing progressively variant textures can be applied to many organ illustrations. For the illustration of the scm, there are two different regions along the model. In our illustration of the mandible, our objective is to obtain an illustration similar the one in [Net09] (Fig. 3). In the anatomic illustration of the mandible, the texture elements are different for the inner layers (bone tissue) and outer layers (endosteum tissue). Both of these features are illustrated by synthesizing solid textures for a specific patient's mandible using our approach (Fig. 9 and Fig. 10).
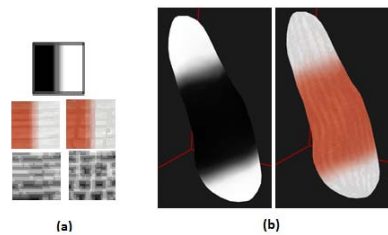


**Figure 9:** *Different textures are synthesized along the model for different regions of scm*

### 5.4. Model-based Texture Synthesis Control

In anatomical illustrations such as Fig. 3, the muscle textures look different for different clipping planes. The important thing here is that the texture looks different based on the
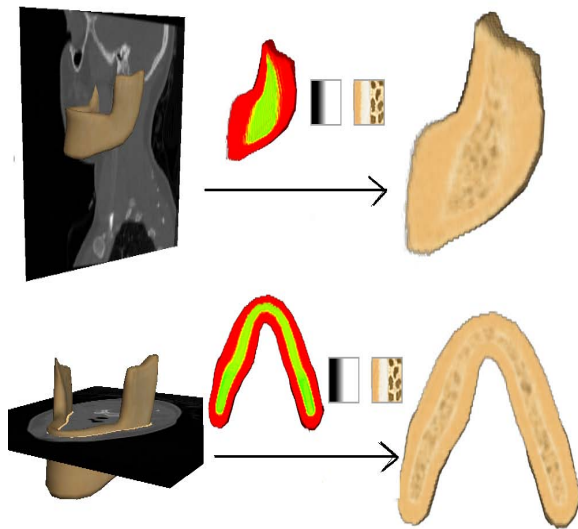
**Figure 10:** *Different textures are synthesized for different regions of mandible (left) clipping planes along and through the mandible (middle) inputs to solid texture synthesis: transition field for the mandible, exemplar texture with its transition image (right) synthesized solid texture visualized by clipping planes*

orientation of the clipping plane. It is very hard to constrain the texture on the clipping plane for every possible orientation in texture synthesis, since we have to do neighborhood search for each constrained direction. In anatomical illustrations the models are mostly clipped either perpendicular to the medial axis or along the medial axis. Thus, in texture synthesis we constrain our texture synthesis in these directions. To do so, different 2D exemplar textures are picked as input for different longituadinal vs axial directions, and neighborhoods are rotated in the search phase of the approach as it is done in oriented solid texture synthesis. The selection of 2D exemplar textures in this method is very important, since they should contain similar colors. Otherwise, the synthesized texture may contain the colors that are not in any of the exemplar textures due to the optimization phase in the method. To solve this problem, we use the approach presented in [WAM02] to match the colors of the exemplar textures in texture synthesis. We transfered color from one exemplar texture to the other one. (Fig. 11) illustrates our result for scm.

## 6. Applying Our Method

Given models, which can be generated automatically using methods from image analysis, and given exemplar textures from medical textbooks and other medical illustrations, our system is fully automatic and does not require any user interaction during either model based part or texture synthesis part. There are several parameters in our system that can be set by the user, such as the thresholds for transition regions and the size of the neighborhoods.
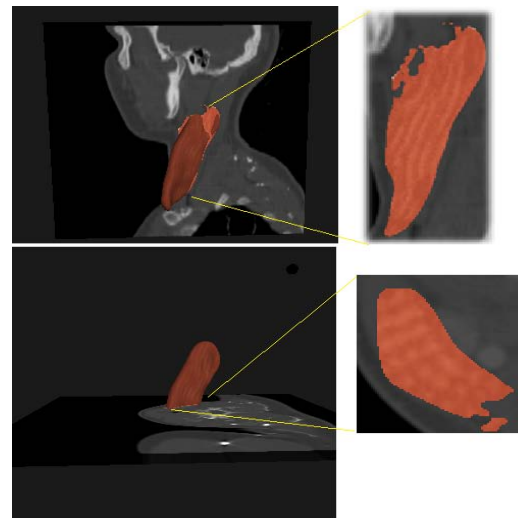


**Figure 11:** *Texture appearance changes based on the orientation of the clipping plane ((top) along object clipping plane (bottom) through object clipping plane)*

For improving the understanding of the patient data, we combine our results for scm and thyroid with the CT data using the Model Guided Rendering (MGR) framework (Fig. 12) [Mer09].

**Further Applications:** Many medical anatomy textbooks and atlases use illustration techniques in order to emphasize the important features of the model or models for a specific purpose. Artists and anatomists have worked for centuries on improving 2D anatomical renderings in order to obtain attractive and effective illustrations for making the data more understandable. They sometimes utilized contours, lines, textures, and a variety of lighting and shading techniques in order to achieve that goal. Moreover, they combine the medical images (such as CT or MRI) with the illustrations in order to highlight the model's material features in the real medical data. These illustrations are very important for many applications, such as education and procedure planning. In addition, these illustrations are taken in the operating room for reference during the surgery. However there are two huge drawbacks of these illustrations: first, they are not of this particular surgical patient's anatomy and second, they are not interactive.

The main importance of our approach is the generation of patient-specific volumetric illustrations by utilizing textures. The methods we are developing are intended to make anatomical illustrations useful not only as references to "normal" anatomy but for understanding the anatomy of the particular clinical patient being examined, planned, or operated on. For teaching, these methods could be applied to convert database of teaching CT's showing various different types of conditions into Netterly renderings.

Our illustrations can be used to improve the communication between patients and doctors. For example, surgical procedures or the change in the shape of the anatomical or-
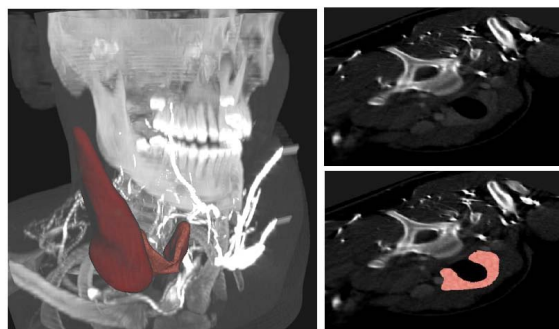
**Figure 12:** *Solid textures for scm and thyroid in MGR framework*

gans due to a disease can be explained to the patient using our illustrations. Patients can easily see and recognize the organs inside the CT data using our illustrations. This may have a profound positive impact on the patient recovery period and communication between patients and doctors.

These illustrations can be used in image guided clinical treatment planning in order to understand the relationship between anatomic features in space. Different solid textures can be produced for different models using our approach, and they can easily be integrated with the CT image data. In addition, our illustrations can be used in order to highlight the models inside the CT data for surgical planning.

## 7. Conclusion and Future Work

Our approach has advantages over the other approaches [CYZ*09] because it can generate consistent and detailed progressively changing solid textures from 2D exemplar textures. The drawback, however, is the cost in both computation and memory, as it explicitly computes and stores a dense 3D array of voxels covering the entire target model. In the future, we would like to work on these problems. Moreover, in this paper we used atlas textures for showing a "most likely" appearance of the segmented 3D anatomical models. In the future, we would like to extend our approach to using more information from the CT data. We would like to illustrate not only normal but also abnormal and non-atlas structures with textures appropriate to a target patient's condition as determined from prior clinical knowledge, statistical estimates or analysis of the particular patient image. For example, lesions obvious in the gray image might be rendered to look pathological, and regions with clinically identified cancers might look cancerous [Net09].

## References

[CYZ*09] CHEN W., YAN Z., ZHANG S., CROW J. A., EBERT D. S., MCLAUGHLIN R. M., MULLINS K. B., COOPER R., DING Z., LIAO J.: Volume illustration of muscle from diffusion tensor images. *IEEE Transactions on Visualization and Computer Graphics 15*, 6 (2009), 1425–1432. 2, 5, 6, 8

[DC05] DONG F., CLAPWORTHY G. J.: Volumetric texture synthesis for non-photorealistic volume rendering of medical data. *The Visual Computer 21*, 7 (2005), 463–473. 3

[JDR04] JAGNOW R., DORSEY J., RUSHMEIER H.: Stereological techniques for solid textures. *ACM Trans. Graph. 23*, 3 (2004), 329–335. 2

[KFCO*07] KOPF J., FU C.-W., COHEN-OR D., DEUSSEN O., LISCHINSKI D., WONG T.-T.: Solid texture synthesis from 2d exemplars. *ACM Transactions on Graphics 26*, 3 (July 2007), 2:1–2:9. 2, 5, 6

[LE05] LU A., EBERT D. S.: Example-based volume illustrations. In *Proceedings of IEEE Visualization* (2005), pp. 655–662. 1, 3

[Mer09] MERCK D.: *Model-Guided Rendering for Medical Images*. Ph.D. Dissertation, UNC Chapel Hill, 2009. 4, 7

[Net09] NETTER F. H.: *Atlas of Human Anatomy*. Rittenhouse Book Distributors Inc., 2009. 1, 3, 5, 6, 8

[ONOI04] OWADA S., NIELSEN F., OKABE M., IGARASHI T.: Volumetric illustration: designing 3d models with internal textures. In *SIGGRAPH '04: ACM SIGGRAPH 2004 Papers* (New York, NY, USA, 2004), ACM, pp. 322–328. 2

[QY07] QIN X., YANG Y.-H.: Aura 3d textures. *IEEE Transactions on Visualization and Computer Graphics 13*, 2 (2007), 379–389. 2

[SP08] SIDDIQI K., PIZER S.: *Medial Representations: Mathematics, Algorithms and Applications*. Springer, 2008, pp. 1–450. 4

[TAI*08] TAKAYAMA K., ASHIHARA T., IJIRI T., IGARASHI T., HARAGUCHI R., NAKAZAWA K.: A sketch-based interface for modeling myocardial fiber orientation that considers the layered structure of the ventricles. *The Journal of Physiological Sciences 58*, 7 (2008), 487–492. 2

[TOII08] TAKAYAMA K., OKABE M., IJIRI T., IGARASHI T.: Lapped solid textures: filling a model with anisotropic textures. In *SIGGRAPH '08: ACM SIGGRAPH 2008 papers* (New York, NY, USA, 2008), ACM, pp. 1–9. 2, 5

[WAM02] WELSH T., ASHIKHMIN M., MUELLER K.: Transferring color to greyscale images. *ACM Trans. Graph. 21*, 3 (2002), 277–280. 7

[Wei03] WEI L.-Y.: Texture synthesis from multiple sources. In *SIGGRAPH '03: ACM SIGGRAPH 2003 Sketches & Applications* (New York, NY, USA, 2003), ACM, pp. 1–1. 2

[ZZV*03] ZHANG J., ZHOU K., VELHO L., GUO B., SHUM H.-Y.: Synthesis of progressively-variant textures on arbitrary surfaces. In *SIGGRAPH '03: ACM SIGGRAPH 2003 Papers* (New York, NY, USA, 2003), ACM, pp. 295–302. 6