New Techniques for Fast Sliding Thin-Slab Volume Visualization

Janice Z Turlington

William E. Higgins

(JZT) Penn State University, The BioEngineering Program, University Park, PA 16802.

(WEH) Penn State University, Departments of Electrical Engineering, Computer Science and Engineering, and the BioEngineering Program.

(JZT) turling@aaron.ee.psu.edu, (WEH) weh2@psu.edu

The 3D CT data was provided by Drs. Eric Hoffman and Geoffrey McLennan of the University of Iowa, Iowa City. This publication was supported in part by grant # CA-74325 from the National Cancer Institute of the National Institutes of Health and by grants from The Whitaker Foundation. A preliminary version of this work appeared at the unrefereed conference [1].

Abstract

High-resolution three-dimensional (3D) volumetric images obtained by today's radiologic imaging scanners are rich in detailed diagnostic information. Despite the many visualization techniques available to assess such images, there remains information that is challenging to uncover, such as the location of small structures (e.g., mediastinal lymph nodes, narrowed-airway regions). Recently, sliding thin-slab (STS) visualization was proposed to improve the visualization of interior structures. These STS techniques require considerable computation on a general-purpose computer and sometimes depend on user specifications or extra preprocessing. Further, other rendering approaches that use the general STS mechanism are conceivable. We introduce two fast direct techniques for STS volume visualization. The first, a depth (perspective) rendering process, produces an unobstructed, high-contrast 3D view of the information within a thin volume of image data. Results are a function of relative planar locations. Thus, rendered views accurately depict the internal properties that were initially captured as position and intensity. The second method produces a gradient-like view of the intensity changes in a thin volume. Results can effectively detect the occurrence and location of dramatic tissue variations, often not visually recognized otherwise. Both STS techniques exploit the concept of temporal coherence to form sequences of consecutive slabs, using information from previously computed slabs. This permits rapid real-time computation on a general-purpose computer. Further, these techniques require no preprocessing, and results are not dependent on user knowledge. Results using 3D CT chest images show the computational efficiency and visual efficacy of these new STS techniques.

Keywords

sliding thin-slab visualization, volume visualization, projection imaging, virtual endoscopy, volume rendering, 3D imaging, pulmonary imaging, virtual bronchoscopy, sliding thin-slab projection, image display, lung imaging

I. INTRODUCTION

State-of-the-art radiologic imaging scanners offer high-resolution three-dimensional (3D) volumetric images of the human anatomy. Such 3D images, or volumes, depict considerable information for precise evaluation and subsequent treatment delivery. Traditionally, a physician evaluates a 3D image by mentally reconstructing structures visible in the individual two-dimensional (2D) slice images [2]. This approach is difficult, especially for complex regions such as the chest. Further, this approach does not fully exploit the inherent 3D structure of the data. Computer graphics techniques have shown much promise for assisting the physician in 3D image evaluation. Among the techniques available are:

1. projection imaging: integrates data along rays cast through an entire volume to produce a 2D

image resembling a chest X-ray [3, 4].

2. oblique slice viewing (or multi-planar reconstruction): gives 2D cross-sectional data at arbitrary orientations [5–7].
3. curved-section reformatting (or "tube" view): provides mathematically straightened 2D views of data along an arbitrary path [5, 8, 9].
4. volume or surface rendering: permits global 3D viewing of structures in a 3D image [2, 10, 11].
5. virtual endoscopic rendering: provides 3D renderings of interior endoluminal structures [9, 12].

Complete systems exist that integrate many of the visual tools above [13, 14]. Also, systems have been proposed for virtual-endoscopic analysis [8, 9, 12, 15–18].

Unfortunately, the techniques noted above tend to obscure small, thin, interior structures, such as airways, blood vessels, and lymph nodes. Conventional projection images consist of contributions from all elements in a volume. Elements overlap and obscure information from individual structures. Thus, small interior structures are typically not visible. 2D oblique views and curved-section views, regardless of their orientation, often cannot provide enough contextual information to depict small structures. Both 3D rendering and virtual-endoscopic rendering often require prior removal of obscuring structures via image segmentation [2]. Thus, the clarity of the remaining rendered data is not assured, since beneficial data along with undesirable data may be removed.

Recently, Napel *et al.* proposed sliding thin-slab (STS) visualization as a technique to better visualize interior structures [19–21]. In this approach a small window, bounded by clipping planes, encloses a chosen depth of image data. A projection-like image is computed for data within the window to give a 2D STS view. The window can be progressively moved through a volume, forming a sequence of STS views. The STS technique gives a mechanism for seeing smaller structures, without occlusion from large, opaque structures, as evidenced by early applications to the chest and colon [19–22].

Work has been done to adapt the STS mechanism for arbitrary movements within a volume [20, 21, 23]. Thus far, however, only standard image operations have been used to produce STS images, such as the maximum intensity projection (MIP), maximum opacity projection, and volume rendering using an over operator [19, 21]. Aside from the MIP, applying these operations requires expertise in assigning opacity mappings and rendering parameters [7]. Further, running these techniques require significant memory and considerable computation. Finally, efficient algorithms for STS calculation on a general-purpose computer do not exist; this is important for dynamic viewing of STS sequences.

We propose two new techniques for fast dynamic STS visualization on a general purpose computer: the Depth-Weighted Maximum (DWmax) technique and Extreme-Gradient (EG) technique. The DWmax technique produces striking locally-focused renderings with added depth information. The EG technique provides a contour-like map (borders) of local structure changes. A significant aspect of our

work is that the algorithms for both techniques exploit the redundant computations that occur when computing a consecutive sequence of sliding thin slabs. This makes the algorithms computationally efficient and particularly well suited for interactive viewing, for example, in a virtual-endoscopy system [24]. Section II describes our proposed STS methods. Section III presents examples for 3D CT pulmonary images. Section IV discusses the computational efficiency of the proposed techniques and Section V offers final comments.

II. METHODS

The beginning of this section overviews notation and terminology. Section II-B gives a hierarchical series of generic processing algorithms common to both the DWMax and EG STS techniques. Finally, Section II-C presents low-level algorithms specific to each STS rendering technique.

A. Notation

STS visualization is a form of volume rendering that utilizes a sliding window to step through a volume of 3D data [2,20]. A window of constant depth defines the set of image slices that are combined to form a single 2D image slab. The window of data is processed according to the method of choice. These methods include the new DWmax and the EG, plus the previously proposed maximum-intensity and minimum-intensity projections, as listed below:

1. *Depth-Weighted Maximum (DWMax)* – Combines depth and intensity information. This view gives the perception of 3D depth and transparency without the need for gradient or opacity processing.
2. *Extreme Gradient (EG)* – Merges the maximum and minimum intensities to produce a view that depicts local intensity changes, such as boundaries of thin structures (i.e., airway or vessel walls) within a thin volume, without the need for prior segmentation.
3. *Maximum-Intensity Projection (MIP)* – The concept first utilized as an STS operation by Napel *et al.* [19].
4. *Minimum-Intensity Projection* – A 'minimum' version of the MIP [19].

For STS visualization, a parallel-rendering geometry is assumed when computing a slab from a window (view volume) of image data. This amounts to a straightforward integration of voxels along a ray, or column, that exist within the window. As we move through the 3D image, the window incrementally changes and new slabs are computed. As new slabs are computed, a consecutive sequence results. Viewing such a sequence dynamically can reveal more information on small structures than mere static viewing of single STS views.

Throughout the paper, I refers to the given 3D volume of image data. The intensity value of a particular voxel (x, y, z) in this volume is given by $I(x, y, z)$. For simplicity, we assume movements are made in the z direction. Thus, all algorithms produce transverse-plane (x-y plane) STS views.

Computing STS sequences in either the x (sagittal view) or y (coronal view) direction merely involves interpreting the data axes differently in the algorithms. Also, the end border effect (insufficient number of slices to fill the window) is disregarded for simplicity.

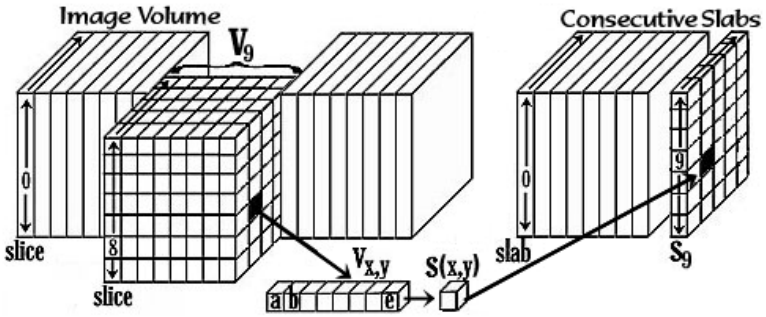


Fig. 1. Schematic illustration of computing a sequence of sliding thin-slabs, $S = \{s_0,\ s_1,\ s_2....\}$, from an input 3D image I. The left side of the figure shows the input image data used for composing the slabs. The right side of the figure depicts the sequence of computed slabs S. The figure depicts the situation for the particular case where the slab depth d_s=7. V_9 is the subvolume of image data used to compute slab s_9. One 8-voxel column of image data, labeled $v_{x,y}$, is singled out of V_9. This is the data used to compute point $s(x,y)$ in slab s_9. Key column locations a, b, and e are discussed in Subsection II-C.

V_i denotes the subvolume of image data used to compute s_i, the i^{th} slab. This subvolume contains $(d_s + 1)$ slices and is given by:

$$V_i = \{I(\cdot,\cdot,z),\ \text{ where } (i-1) \le z \le (i + d_s - 1)\} \tag{1}$$

The parameter d_s, the depth of the slab, is selected relative to the sampling intervals $\{\Delta x, \Delta y, \Delta z\}$ of the image data. For the case of computing transverse-oriented slabs moving in the z direction,

$$d_s = \text{int}\left\{\frac{\text{desired thickness of slabs}}{\Delta z}\right\} \tag{2}$$

where int$(\cdot)$ is the integer function and Δz is the image-slice thickness. The slices of V_i constituting the view volume, or field of view, for slab s_i are slices $I(\cdot,\cdot,i)$ through $I(\cdot,\cdot,i + d_s - 1)$. $I(\cdot,\cdot,i-1)$, which served as the initial slice of slab s_{i-1}'s view volume, acts as the initial slice of V_i for fast computation. As we'll see, slab s_0 is computed brute force. Thus, the lack of a "slice" $I(\cdot,\cdot,-1)$ for V_0 is not an issue when computing the initial slab s_0. The complete sequence of slabs is given by $S = \{s_0, s_1, s_2, \ldots\}$. The intensity value of pixel (x,y) in slab s_i is given by $s_i(x,y)$. The column of image voxels used to compute $s_i(x,y)$ is represented by $V_i(x,y)$. Thus, per (1),

$$V_i(x,y) = \{I(x,y,i-1), I(x,y,i), \ldots, I(x,y,i + d_s - 1)\} \tag{3}$$

Figure 1 depicts the defined variables. Most of the STS computations do not depend on the slab index i or particular slab pixel (x,y) being considered. Thus, in the algorithms below, we exclude the subscript i or notation (x,y) when it unnecessarily complicates the discussion. So, $s(x,y)$ will refer to a single slab pixel, $v_{x,y}$ will refer to the column of data used for computing $s(x,y)$, and $v(z)$ will refer to the z^{th} voxel in a column of image data $v_{x,y}$.

B. Top-Level Algorithms

This subsection presents the top-level algorithms used for computing an STS sequence. Since the slab sequence S evolves from incremental movements, the data volume V_i differs only by two slices from data volume V_{i-1}, per (1). Thus, many of the computations performed for s_{i-1} can be saved and reused for building the next slab s_i. This concept is related to the idea of temporal coherence. This imaging concept states that if one's point of view varies slowly, then consecutive views change little [9,25]. Our proposed algorithms draw upon this concept. As Section IV shows, this results in a large computational reduction.

All STS techniques presented in this paper use the same top level routines for window movement through the image and placement of a computed slab into the final slab sequence; see Figures 2 and 3. To start the computation of a slab sequence, the initial slab s_0 is always computed using a straightforward brute-force computation (routine **BruteSlab**). Also at this stage, we retain certain intermediate processing data in a 2D slab-sized storage buffer H. To compute slab s_1 and all subsequent slabs, we enter a fast stage. The processing window is slid forward to define the next needed subvolume V_1. V_1 differs from V_0 by a newly added end slice and a dropped beginning slice. Routine **FastSlab** considers these two slices in regards to the previous slab to quickly produce s_1 and update H. Processing continues using the fast approach for slabs $s_2,\ s_3, \ldots$, etc.

Routines **BruteSlab** and **FastSlab**, shown in Figure 3, are individual slab-computation functions. Each routine cycles through the column calculations needed for computing each point (x, y) of a requested slab and filling the respective buffer values. The functions, **BrutePoint** and **FastPoint**, are "place holders" for technique-specific STS functions, given in Section II-C.

```
INITIAL SLAB COMPUTATION  (i = 0)
    (s_0, H) = BruteSlab(V_0, d_s, d_v)

FAST SEQUENCE COMPUTATION  (i = 1, 2, 3, ..., n)
    for each slab s_i ϵ S
        (s_i, H) = FastSlab(V_i, d_s, d_v, s_{i-1}, H)
```

Fig. 2. Top-level algorithm for computing a slab sequence. **BruteSlab** computes slab s_0 brute-force. The subvolume V_0 and slab parameters d_s and d_v are input arguments. The 2D array, H, stores data required for fast processing of the next slab. The array H has one or two elements per (x, y) location, depending on the specific STS technique. The output at $i = 0$ is s_0 and H. **FastSlab** quickly computes subsequent slabs. For each s_i, the input arguments to the slab-computation routine are V_i, d_s, d_v, s_{i-1}, and H. The outputs are the computed slab s_i and the updated H. Functions **BruteSlab** and **FastSlab** are presented in Figure 3.

C. Specific Point Functions

Each STS technique has its own specific point-level functions. A slab point $s_i(x, y)$ is computed from a column of image values that come directly from the associated subvolume V_i. This column contains the original image values $I(x, y, \cdot)$ for all slices in V_i. All columns follow a standard arrangement,

```
Function BruteSlab(V, d_s, d_v)
    for each point (x, y) ϵ s
        (s(x, y), h_{x,y}) = BrutePoint(v_{x,y}, d_s, d_v)
    return(s, H)

Function FastSlab(V, d_s, d_v, s_pre, H)
    for each point (x, y) ϵ s
        (s(x, y), h_{x,y}) = FastPoint(v_{x,y}, d_s, d_v, s_pre(x, y), h_{x,y})
    return(s, H)
```

Fig. 3. Generic algorithms for computing individual slabs. The brute-force routine **BruteSlab** requires a data volume V and slab parameters, d_s and d_v. **BrutePoint** computes an individual slab point $s(x, y)$ and voxel-level buffer results $h_{x,y}$, where $h_{x,y}$=$H(x, y, \cdot)$. For fast calculations, **FastSlab** requires V, d_s, d_v, and H. The previously computed slab s_{pre} may also be required. For each (x, y), routine **FastPoint** uses the corresponding column of image data associated with (x, y) to return a slab point value $s_{x,y}$ and an updated $h_{x,y}$. Point routines **BrutePoint** and **FastPoint**, specific to each particular STS technique, are discussed in Section II-C.

regardless of slab size or slab technique. Three column elements have special significance. They are: $v_{x,y}(0)$ and $v_{x,y}(1)$, the base values of the previous and present view volumes; and $v_{x,y}(d_s)$, the final column element. These key elements will be denoted $v(a)$, $v(b)$, and $v(e)$ in the point functions; recall Figure 1.

C.1 Depth-Weighted Maximum (DWmax) Point Functions

Figure 4 gives a model of the DWMax technique. DWmax interprets sight as a process of peering through a group of progressive planes. These planes are layered one after another, along a plane's normal. (or along the path of an ideal X-ray). DWMax gives the front visual plane the greatest intensity. Successive planes get diminishing intensity, based on distance. But, to retain the selectivity of maximum intensity projection, the overall brightest value along this fading line of sight passes as the perceived value. With this viewing mechanism, both depth and intensity are integrated.

Depth weighting of column values is central to DWMax. These weights are based on two parameters: the field of view d_s and depth of vision d_v. Consider Figure 1 again. For a given column of data (line of sight!) $v_{x,y}$, the base slice, or slab base, is the first plane in the line of sight for the slab. The depth of vision d_v is the distance into an image that is visible from the base slice. The field of view d_s, defined by (2), is the slice depth of the incoming image data that will be processed from the base slice. A column element's depth weight w_i is a function of the element's slice location within the window (i.e., column index) and viewing parameters:

$$w(i) = d_v - (i - 1) \tag{4}$$

where $i = 1, 2, \cdots, d_s$. Weighting is largest at the base slice, where it is equal to the depth of vision d_v. The depth weight decreases with each slice movement from the base. Very importantly, these weights are based strictly on the viewing geometry: A windowed voxel's depth weight is not a function of its individual intensity nor is it a function of the intensities of its neighboring voxels. Thus, a DWmax

slab point $s(x,y)$ is the normalized (by d_v) maximum value from the depth-weighted values of the column $v_{x,y}$:

$$s(x,y) = \max\left(\frac{\left[v(b)*w(b)\right]}{d_v}, \frac{\left[v(2)*w(2)\right]}{d_v}, \cdots, \frac{\left[v(d_s-1)*w(d_s-1)\right]}{d_v}, \frac{\left[v(e)*w(e)\right]}{d_v}\right) \quad (5)$$

The maximum operation of DWmax permits the observer to see distant, but bright, occluded objects. The depth weighting, however, keeps such objects in proper perspective: as the object is approached in suubsequent slabs, it appears brighter and closer. As another point, the field of view d_s models the presence of a view plane that stops all viewing short of the farthest visible plane d_v, such as a billboard or wall of a house: this is in keeping with the spirit of thin-slab viewing.

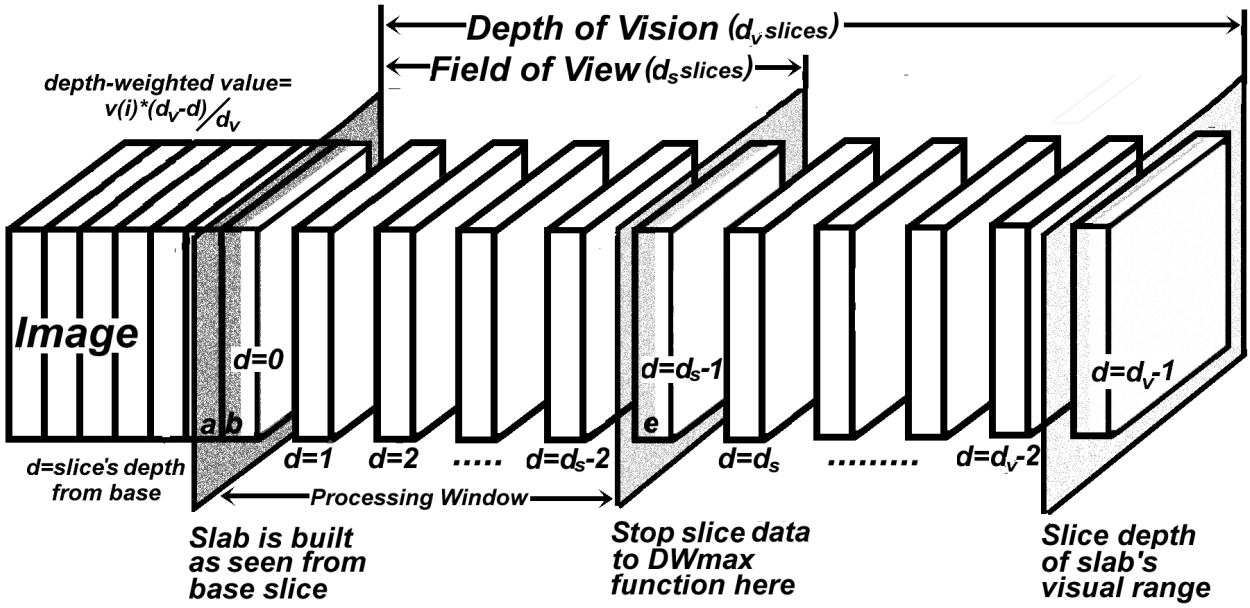


Fig. 4. The DWmax model. All image data values are weighted by the normalized distance between the base slice and the data's originating slice. Only the slices within the field of view d_s are processed. Their distances from the base b (slice where the slab is viewed from) range from $d = 0$ to $d = d_{s-1}$. Theses slices are placed in a relative depth perspective, as they are normalized by the number of slices defining the depth of vision. This depth d extends from 0 to d_{v-1}.

Figure 5 gives the point-level algorithms for DWMax. Fast point processing determines the location of the present depth-weighted maximum column value based on the previous point $s_{i-1}(x,y)$. The intensity and weight of $s_{i-1}(x,y)$ are given in $h_{x,y}$. Two tests determine the location:

1. Is the present depth-weighted end value $v(e) * w(e)$ (now entering the processing window) the present maximum?
2. Is the previous depth-weighted base value $v(a) * w(a)$ (now being discarded from the processing window) the previous maximum?

A positive return from the first test identifies the depth-weighted present end as the present maximum (i.e., the slab point to be returned). Otherwise, we proceed to the second test where a positive return

```
Function BruteDWmax(v, d_s, d_v)
    h(0) = v(b)
    h(1) = w(b)
    for z = 2 to d_s
        if (v(z) * w(z)) ≥ (h(0) * h(1))
            h(0) = v(z)
            h(1) = w(z)
        end
    end
    return ((h(0)*h(1))/d_v, h)
```

```
Function FastDWmax(v, d_s, d_v, h)
    if (v(e) * (w(e) − 1)) ≥ (h(0) * h(1))
        h(0) = v(e)
        h(1) = w(e)
        return ((h(0)*h(1))/d_v, h)
    else if h(1) = w(b)
        return (BruteDWmax(v, d_s, d_v))
    else
        h(1) = h(1) + 1
        return ((h(0)*h(1))/d_v, h)
```

Fig. 5. DWMax point functions. These functions act as **BrutePoint** and **FastPoint** generic slab algorithms of Figure 3. Function **BruteDWmax** brute forces the computation of point $s(x, y)$. This function computes all points in the first slab s_0. **BruteDWmax** also comes into play for certain points on subsequent slabs. The field of view d_s and the depth of vision d_v are input parameters. The column base value $v(b)$ and its weight $w(b)$ initiate the search for the maximum depth-weighted column value. All values are weighted and compared. The maximum is normalized and returned. The image point $v(z)$ and corresponding weight $w(z)$ that produce the DWMax value are saved in the buffer array $h_{x,y}$ as $h(0)$ and $h(1)$. This array assists fast processing of point (x, y) in the next slab. Function **FastDWmax** is for fast processing of DWmax point values. In most cases new slab points are computed immediately. But if the previous slab maximum's weight, as stored in $h(1)$, is a base weight $w(b)$, then previous point information doesn't help in computing the present point value. In these cases, brute-force computation is required.

implies that brute-force processing of the total column is necessary. If both tests fail, the previous point is deemed the present maximum after its weight has been adjusted.

The farthest visible plane, modeled by the depth of vision d_v, proves to be the key to creating a realistic depth perspective. The choice of d_v can influence the visual appearance of a slab point in a DWmax slab. Four details come to light when selecting d_v. First, when $d_v = d_s$, end slice intensities are overly decreased. The original end slice values do not retain a level of tissue information that has meaning in the processing of the slab; thus, it is best to have $d_s < d_v$. Second, in our experience, the range of values $(1.5 * ds) \leq d_v \leq (2 * d_s)$, produce the most effective visual differentiations of like tissues located on diverging slices. Third, when $d_v \geq (2 * d_s)$, depth perception suffers due to a diminished differentiation on diverging slices. Fourth, a large $d_v \gg d_s$ reduces slice differentiation to the degree that depth becomes visually undetectable.

End-slice contributions resulting from DWmax processing without an extended range of vision (i.e., $d_v = d_s$) and contributions processed with a better d_v are shown in Figure 6. A depth of vision that does not extend beyond the field of view diminishes bone situated in the end slice to air. Thus, the end slice does not contribute tissue information to the slab. However, processing with $d_v = (2 * d_s)$ sets bone to a value (soft tissue range) that will add to a slab's depth perception. Therefore, the degree a slice intensity value contributes to a slab is dependent on the choice of d_v. An effective d_v provides a sufficient distance differentiation of like tissues to portray depth.

Slab Size	Vision vs. View	Calculation of DWmax value	end slice bone value↦slab tissue value
7	$d_v = d_s$	$\left(\frac{(1047+1000)(7-6)}{7}\right) - 1000 = -708$	bone (1047)↦air (-708)
7	$d_v = (2 * d_s)$	$\left(\frac{(1047+1000)(14-6)}{14}\right) - 1000 = 170$	bone (1047)↦soft tissue (170)
4	$d_v = d_s$	$\left(\frac{(1047+1000)(4-3)}{4}\right) - 1000 = -488$	bone (1047)↦just above air (-488)
4	$d_v = (2 * d_s)$	$\left(\frac{(1047+1000)(8-3)}{8}\right) - 1000 = 279$	bone (1047)↦above soft tissue (279)

Fig. 6. Depth-of-vision (d_v) effects on DWMax end-slice contribution. The calculations consider standard 16-bit CT HU intensity values, where air is -1000 HU and solid bone is 1047 HU [26]. Solid bone, which appears very bright on a standard CT slice, is assumed to be in the end slice of a slab's image data. The slabs are sized at $d_s = 7$ (rows 1 and 2) and $d_s = 4$ (rows 3 and 4). The calculation of the end-slice contribution, combining depth weighting (4) and normalization (5), is shown in the third column. When the depth of vision does not add distance to the field of view ($d_v = d_s$), bone appears as air, as shown in the fourth column, and does not contribute any tissue definition to the slab. In addition, all end-slice intensities appear as air when using $d_v = d_s$. Thus, the end slice becomes essentially invisible. However, with $d_v = (2 * d_s)$ the bone retains soft tissue values that add legitimate definition to slab results when unobstructed.

C.2 Maximum and Minimum Point Functions

The well-known STS-MIP operation causes the suppression of low densities [19] and focuses on bright structures. This section provides fast general-purpose algorithms for computing the STS-MIP and the corresponding minimum operation. In mathematical form, the STS-MIP function determines the maximum value within a column of image values $v_{x,y}$:

$$s(x,y) = \max\left(v(b),\ v(2),\ \cdots,\ v(d_s - 1),\ v(e)\right) \tag{6}$$

Figure 7 gives the straightforward STS-MIP point-level algorithms, **BruteIMax** and **FastImax**. Algorithms for the minimum function are identical, except that a minimum value is returned.

```
Function BruteIMax(v, d_s)
    v(0) = v(b)
    for z = 2 to d_s
        if v(z) ≥ v(0)
            v(0) = v(z)
        end
    end
    return(v(0))
```

```
Function FastIMax(v, d_s, s_pre)
    if v(e) ≥ s_pre
        return(v(e))
    else if v(a) = s_pre
        return(BruteIMax(v, d_s))
    else return(s_pre)
```

Fig. 7. Point functions for STS-MIP. The brute-force maximum point function **BruteIMax** initializes its search for slab point $s_i(x,y)$ at the base $v_{x,y}(b)$ of the incoming data column $v_{x,y}$. The column size d_s is received as an incoming parameter. Upon searching the entire column, the maximum value is returned. The fast maximum point function **FastIMax** brings in the previous maximum $s_{i-1}(x,y)$. The incoming column's end $v(e)$, the new addition to the column, is returned if found to be the maximum. If the previous base $v(a)$ was the maximum in the previous slab $s_{i-1}(x,y)$, then a complete column search for the maximum is required. Otherwise, the existing maximum becomes the present maximum.

C.3 Extreme Gradient (EG) point functions

STS maximum-intensity and minimum-intensity slabs only contain information concentrated toward one extreme end of the intensity range. Therefore, the resulting slabs have limited usefulness [7]. We

propose the *Extreme Gradient* (EG) STS technique. The EG technique combines the maximum and minimum values:

$$s(x,y) = s_{max}(x,y) - s_{min}(x,y)$$

where $s_{max}(x,y)$ and $s_{min}(x,y)$ are point-level outputs from the maximum and minimum STS functions, per (6). The EG, reminiscent of the morphological gradient found in mathematical morphology [27], provides information not shown by DWmax or other slab techniques. Figure 8, which draws upon the basic max and min STS functions of Section II-C.2, shows the point-level functions for EG.

Function **BruteEG**$(v,\ d_s)$
$h(0) =$ **BruteIMax**$(v,\ d_s)$
$h(1) =$ **BruteIMin**$(v,\ d_s)$
return$\Big(\{h(0) - h(1)\},\ h\Big)$

Function **FastEG**$(v,\ d_s,\ h)$
$h(0) =$ **FastIMax**$\Big(v,\ d_s,\ h(0)\Big)$
$h(1) =$ **FastIMin**$\Big(v,\ d_s,\ h(1)\Big)$
return$\Big(\{h(0) - h(1)\},\ h\Big)$

Fig. 8. Point functions for the Extreme Gradient. Function **BruteEG** brings in a column of data $v_{x,y}$ and its size d_s. Upon calculation of the difference between the brute maximum and minimum intensities the EG point $s_i(x,y)$ is returned; see Figure 7. The extremes are placed in $h_{x,y}$ for fast processing of $s_{i+1}(x,y)$. Function **FastEG** subtracts the maximum and minimum intensities as returned from the fast extreme intensity point functions. The stored values $h_{x,y}$ are updated with the present point extremes and returned.

At a simple level, EG is a map of the maximum change of tissue density relative to depth. As the results show, the EG STS function can detect and preserve significant small tissue changes, especially if suitable intensity windowing is used. Changes within a slab are amplified by their repeated detection and subsequent recording into contiguous EG slabs. This extended duration greatly aids with the recognition of sudden or subtle changes. Further, EG does not employ a thresholding process during processing. The EG method can highlight local borders between different tissue types. The changes detected by EG can be particularly dramatic when viewing a sequence of slabs.

III. Results

We use three high-resolution 3D CT scans of the chest (no contrast agents introduced) to illustrate the proposed DWmax and EG STS methods. The results, which appear in Figures 9 through 15, show the visual efficacy of the methods. Additional results with other images appear in [1,28].

Figure 9 shows how the previously proposed STS-MIP view gives no impression of depth or 3D information. Also, much available local structural information is not visible. The corresponding DWmax view, on the other hand, provides detailed 3D local structure, a strong impression of depth, and endoluminal information. Note how the shaded carina near the center of the DWMax views indicates the dark airway branchings into the left and right main bronchi. Further, the sequence of DWMax views give considerable 3D information as the chest structures evolve. These views provide depth perception, effectively show the physical structure of interior tissues, provide a clear visual perspective into the cavity, and give insight into the external and internal structural relationships.

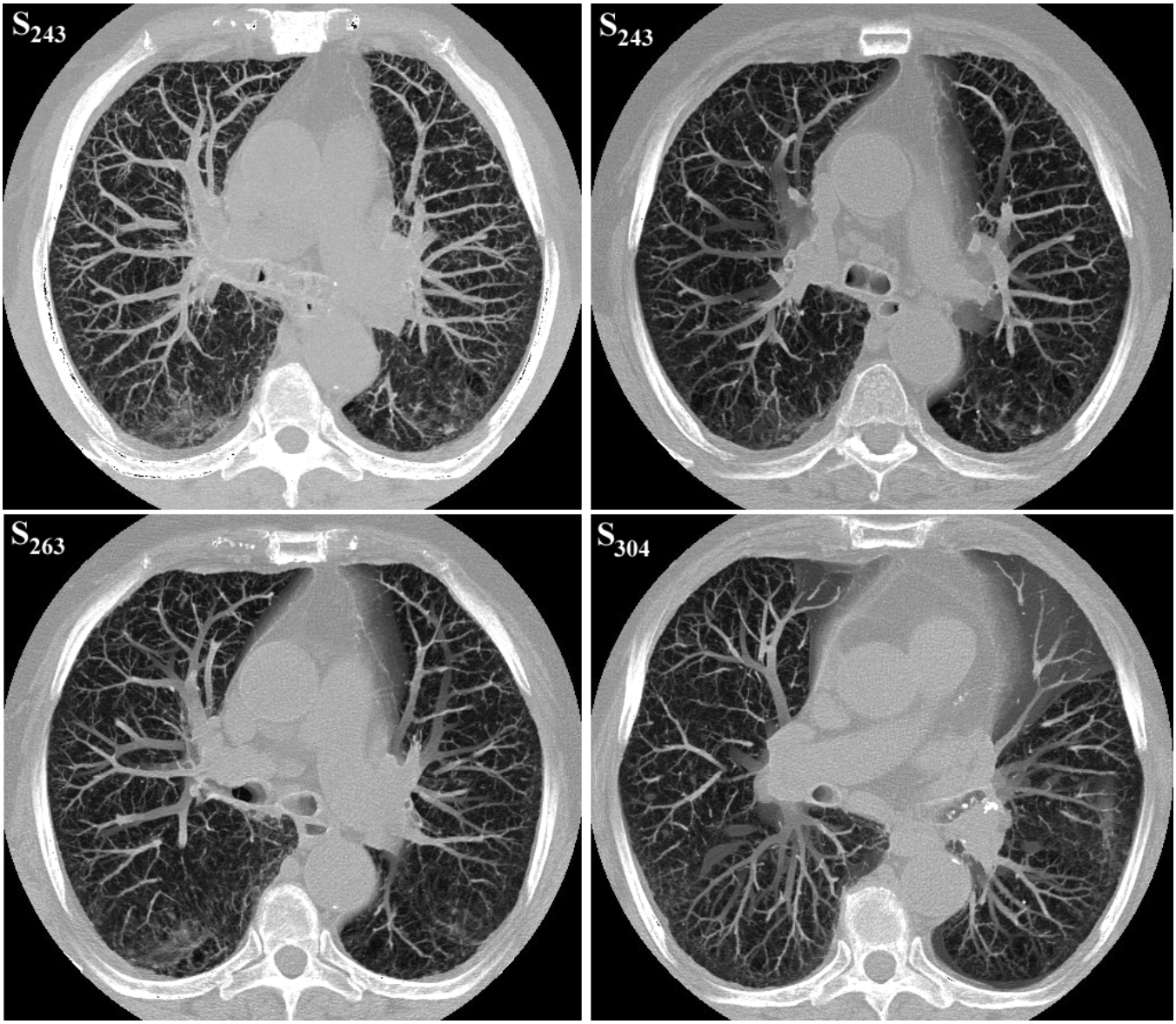


Fig. 9. Comparison of standard STS-MIP and DWmax slabs. This case was generated with a Toshiba Aquillon helical CT scanner. The image consists of 406 contiguous 512×512 slices with thickness $\Delta z = 1.0mm$, $0.5mm$ slice overlap, and axial-plane resolution $\Delta x = \Delta y = 0.5mm$. Top left: standard STS-MIP view s_{243}; $d_s = 40$. Top right: corresponding DWMax view s_{243}. Bottom views: further DWMax slabs, s_{263} and s_{304}. All DWMax views use $d_s = 40$ and $d_v = 60$.

Figure 10 demonstrates how standard global projection techniques (all slices compiled to form one coronal-oriented image) either obscure most structures (MIP) or show little detail (weighted sum). Further, the coronal multiplanar reformatted slice gives no actual 3D information. The coronal sequence of STS-DWMax views, however, offer a dramatic dynamic view of the chest. Vessels, airways, and soft tissues of the heart are apparent. Especially striking is how this sequence allows the observer to perceive the airway tree's endoluminal structure.

Figure 11 illustrates how the STS-DWMax technique gives clear views of the "budding trees" for the airways and vessels. No other viewing technique would be able to provide such views of both the

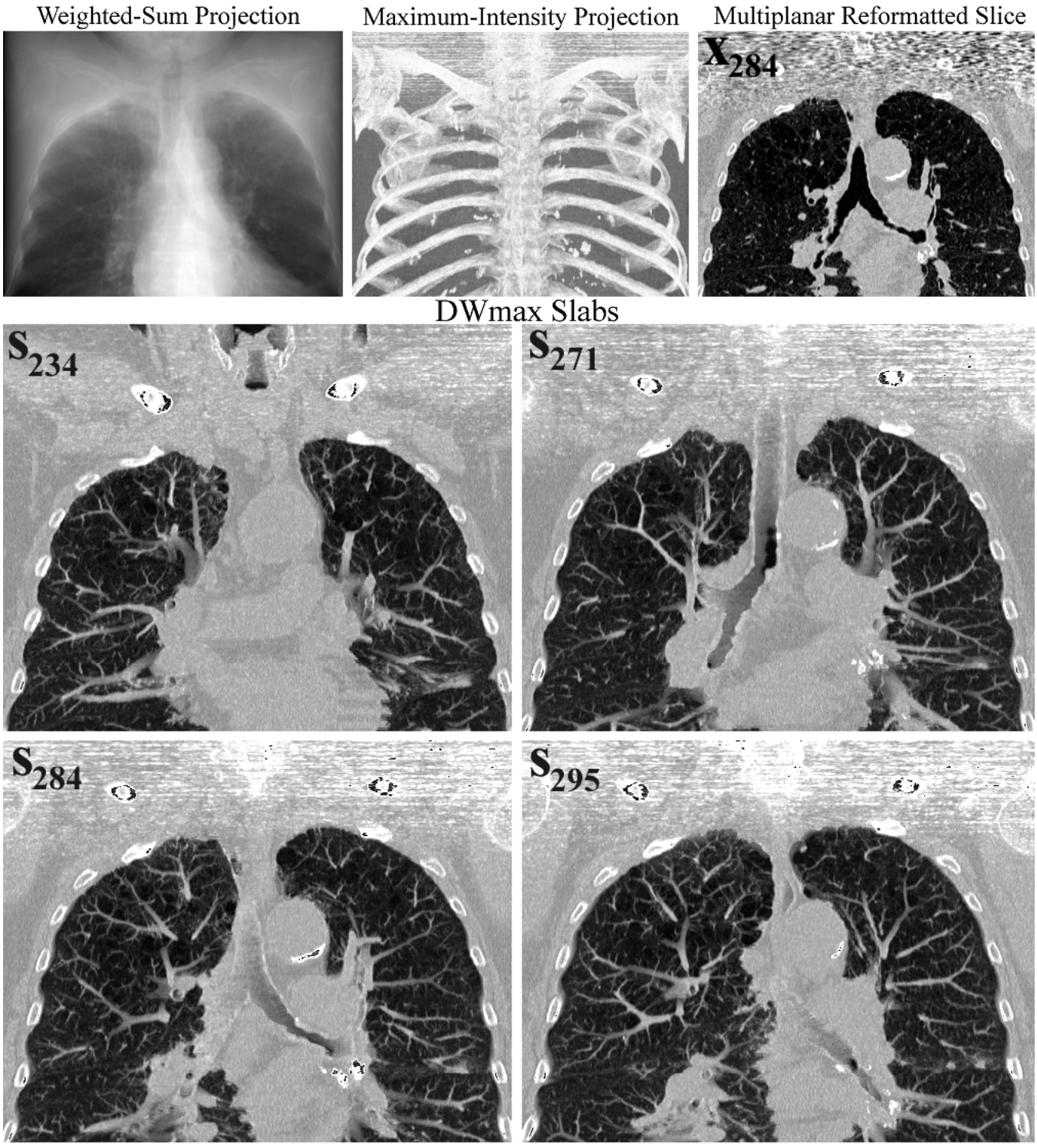


Fig. 10. Standard projection viewing versus STS viewing in the coronal direction. Data of Figure 9 used. Top row: (left) weighted-sum coronal projection image; (center) maximum-intensity projection image; (right) coronal multiplanar reformatted view $y = 284$. Bottom two rows: four sequential DWmax views travelling through the chest from front to back, s_{234} ($d_s = 25,\ d_v = 40$), s_{271} ($d_s = 20,\ d_v = 35$), s_{284} ($d_s = 25,\ d_v = 35$), and s_{295} ($d_s = 25,\ d_v = 30$).

global and local structure of such an interior. Figure 12 dramatically illustrates the greater detail afforded by DWmax slabs over raw slice data. Notice the evolution of the airway and vessel trees in

the slabs; also the shape and depth of some of the major airways are very apparent.

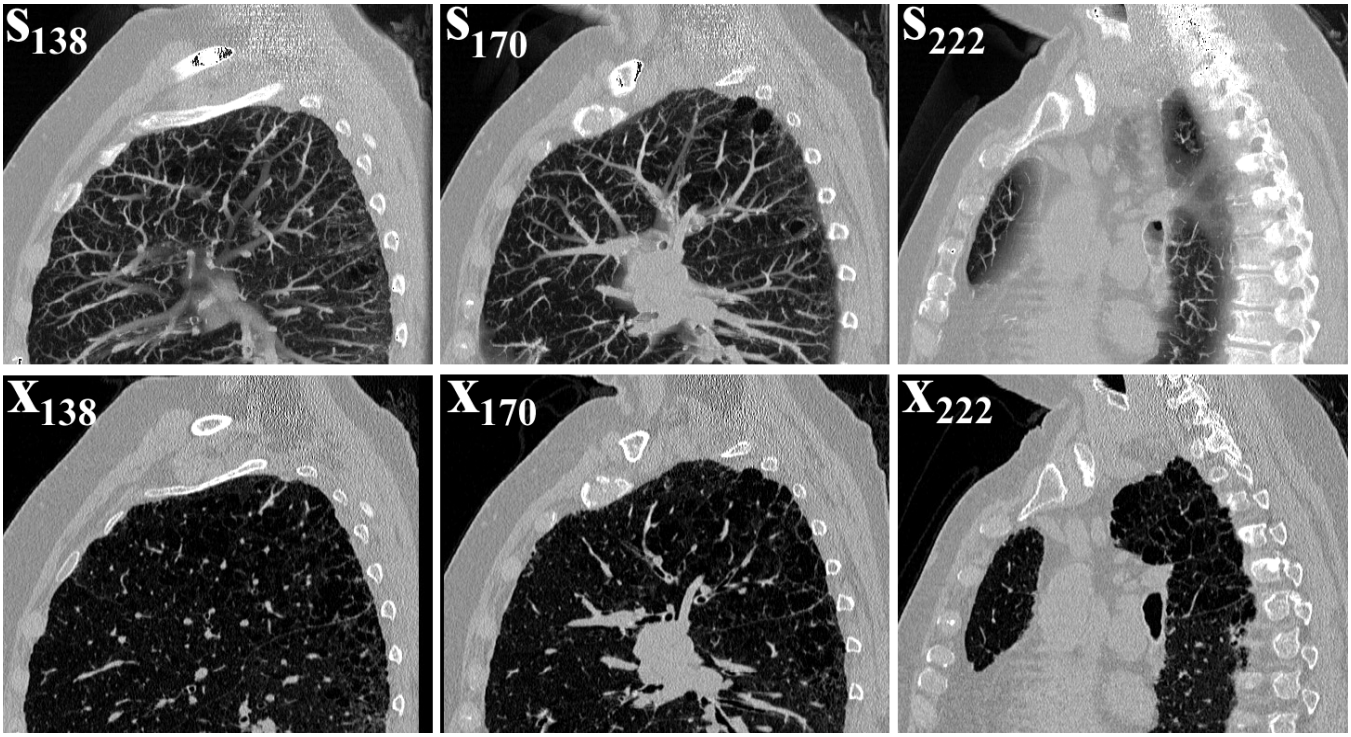


Fig. 11. Multiplanar reformatted (MPR) viewing versus slab viewing for the sagittal direction. Marconi Mx8000 multidetector helical CT scan made of a healthy subject. The image consists of 165 contiguous 512×512 slices with thickness $\Delta z = 1.0mm$, and axial-plane resolution $\Delta x = \Delta y = 0.35mm$. Varied sized slabs are used to increase visible structural information. Top row: three sequential DWmax views traveling across the chest from left to right, s_{138} ($d_s = 40,\ d_v = 50$), s_{170} ($d_s = 30,\ d_v = 45$), s_{222} ($d_s = 20,\ d_v = 30$). Bottom row: corresponding raw sagittal MPR slice data, which act as the base slices for the slabs above, x_{138}, x_{170}, and x_{222}.

Figures 14-15 illustrate the usefulness of the EG technique and how it can be used in conjunction with DWMax. (Figure 13 shows samples of the original raw transverse-plane slice data used for both Figures 14 and 15.) The raw EG slabs are useful for seeing any and all large local changes. Again, viewing a slab sequence is particularly enlightening for seeing the evolution of the changes. By intensity windowing the EG slabs, specific changes, corresponding to particular tissue transitions, can be isolated. Consider the case of 3D CT chest imaging. Since the HU (intensity) ranges of various tissue types is known, these can easily be focused on with windowed EG views. The middle row of Figure 14 illustrates how structural detail can be isolated. Note the concentric ring-like borders extracted for the trachea and cancer. The encroaching cancer within the trachea becomes very apparent in the EG views. By varying the slab size, as done in this figure, we can observe the extent of the cancer under several different viewing conditions. Figure 14 also shows the impact of slab size variation on DWMax.

Figure 15 shows the intriguing possibility of merging STS-EG and STS-DWMax views. The DWMax views show the 3D structure, while the EG views focus on the "interesting changes." In the case of Figure 15, the primary tracheal borders, concentric impinging-cancer borders, and esophagus are

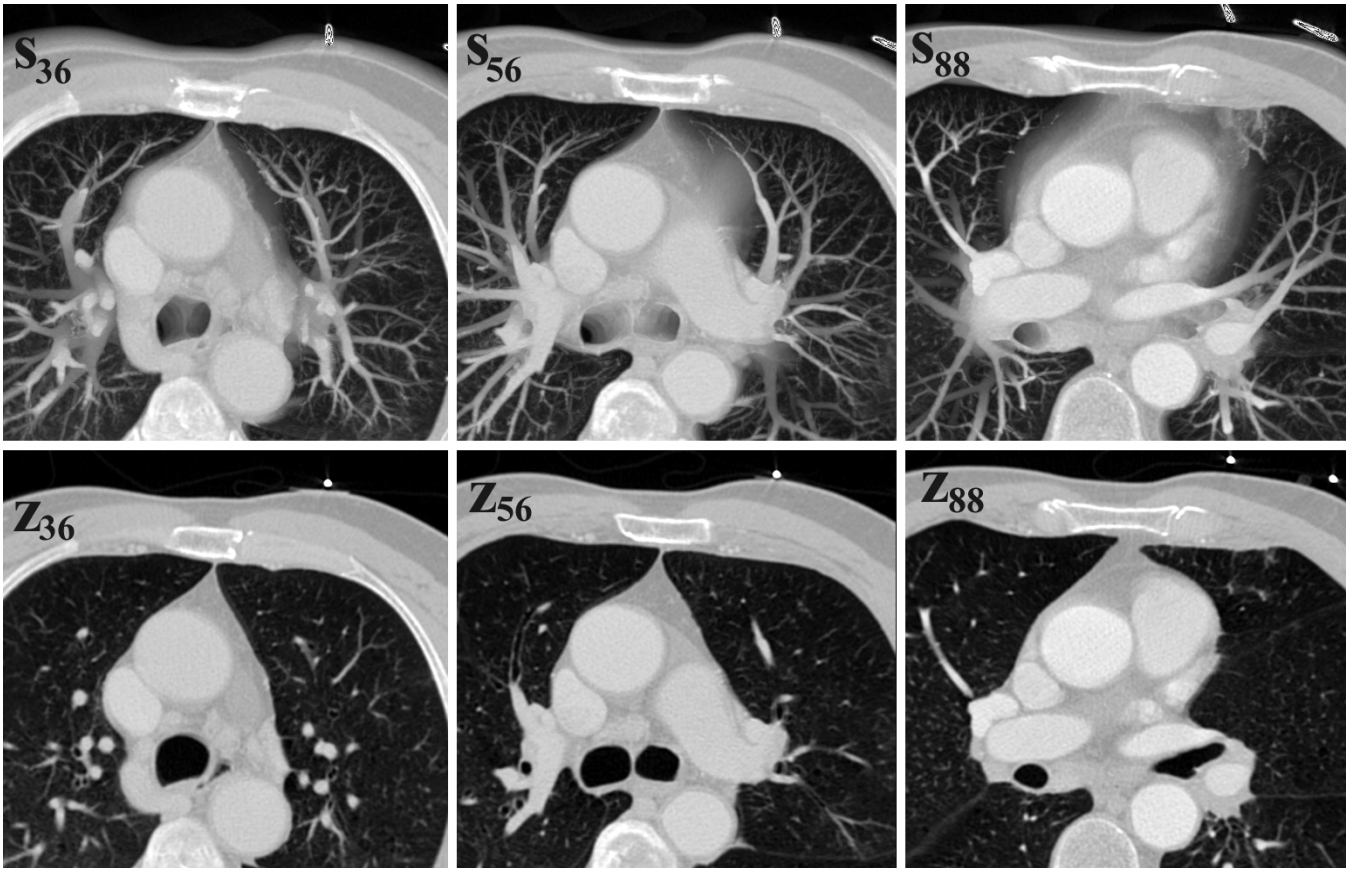
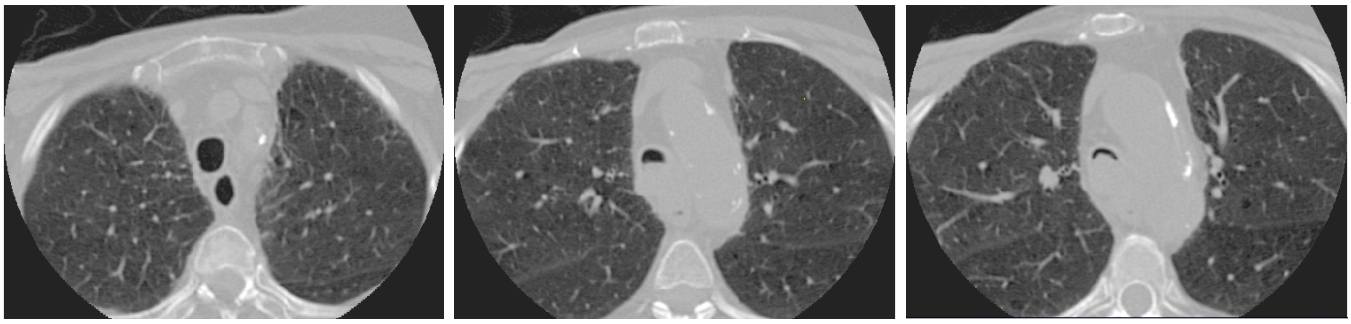


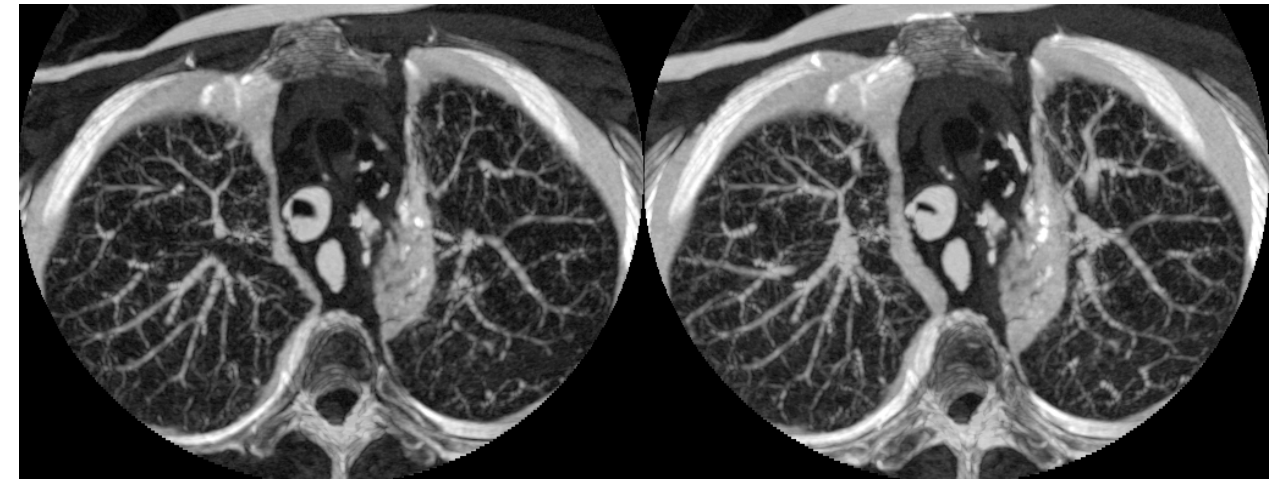
Fig. 12. Transverse slab viewing. Same data as Figure 11 used. Top row: three sequential DWmax views traveling through the chest from top to bottom, s_{36} ($d_s = 35,\ d_v = 50$), s_{56} ($d_s = 35,\ d_v = 50$), and s_{88} ($d_s = 30,\ d_v = 40$). Bottom row: corresponding raw slice data, which act as the base slices for the slabs above, z_{36}, x_{56}, and x_{88}.



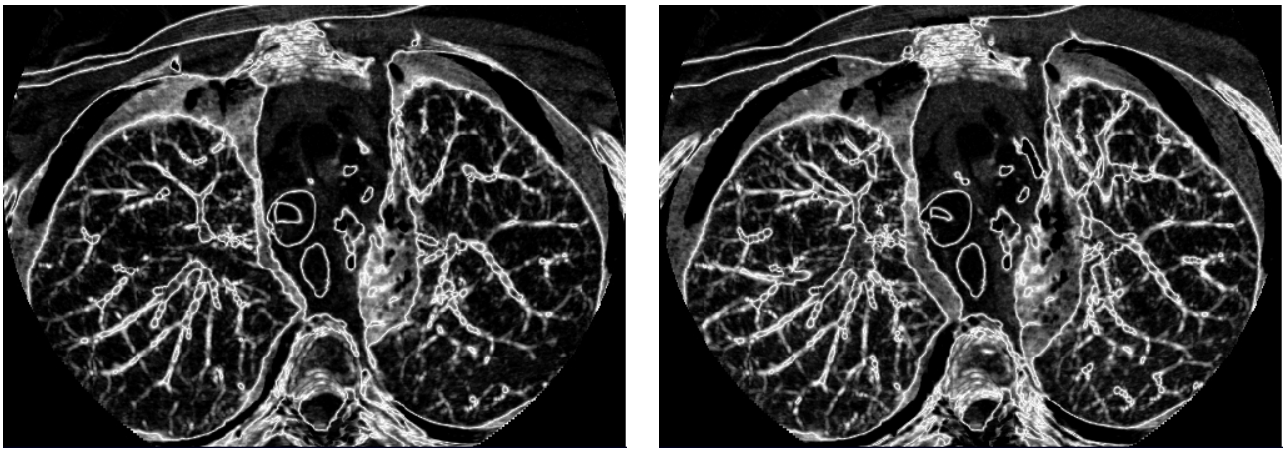
CT slice 5 CT slice 11 CT slice 14

Fig. 13. Original CT slices for a human lung-cancer case; these data are used for Figures 14 and 15. (For further reference, this case was used in [18].) The original 3D image consist of 25 contiguous 512×512 slices with thickness $\Delta z = 3.0mm$ and axial-plane resolution $\Delta x = \Delta y = 0.41mm$.

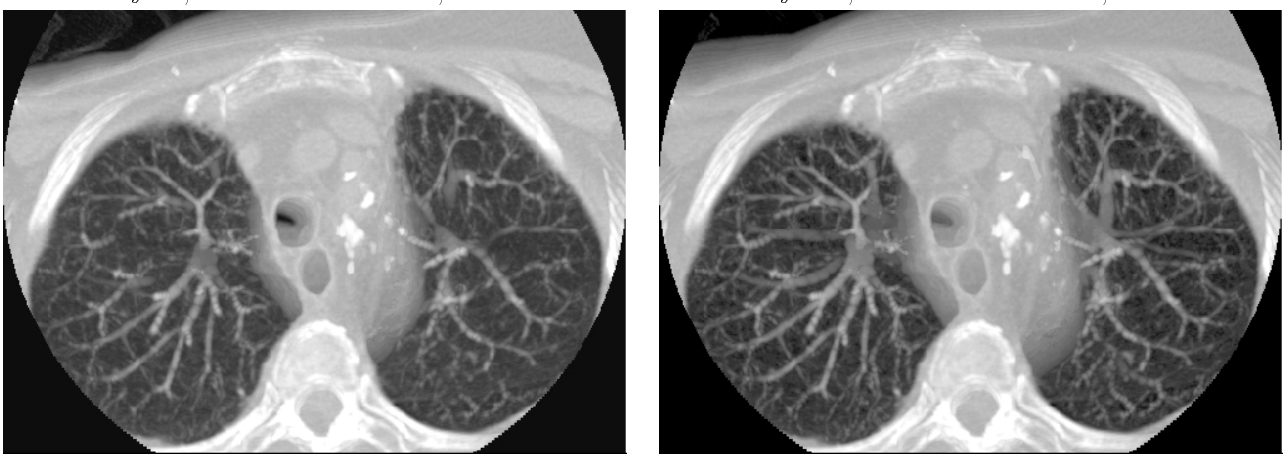
displayed. In our experience, the EG technique can provide raw change information, tissue borders, and evolving "change patterns." Because the EG technique is a function of slice-intensity extremes, changes remain consistent over a range of slabs and over a multitude of slab sizes. Viewing consecutive slabs or various-sized slabs can help identify tissues, locate steady areas of growth (i.e., wall thickness), and locate sudden or suspicious growth (i.e., airway obstructions).



EG: d_s=7 **EG:** d_s=10



EG: d_s=7, window: mid=85, width=70 **EG:** d_s=10, window: mid=85, width=70



Dwmax: d_s=7, d_v=14, base=slice 5 **Dwmax:** d_s=10, d_v=20, base=slice 5

Fig. 14. Demonstration of EG and DWMax for the lung-cancer case of Figure 13. The slices of Figure 13 show the unprocessed base ($v(b) = I(\cdot,\cdot,5)$) and ends ($v(e) = I(\cdot,\cdot,11)$, $v(e) = I(\cdot,\cdot,14)$) of the slabs shown in this Figure. Top row shows the raw EG slabs. Middle row shows the results of intensity windowing the raw EG slabs (8-bit gray-scale assumed; width = width of intensity window; mid = middle gray-level value of window). Bottom row shows corresponding DWMax STS views. By varying the slab size, the windowed STS-EG views show the evolving shape and size of the cancer impinging on the trachea (note center contours in the windowed EG views, which show the tracheal border and cancer border). Also, the thicker-slab DWMax view (bottom right) shows extra depth and visual extension of bronchi and vessels.

IV. Computational Efficiency

We performed complete operation counts for both the brute-force and corresponding fast versions of each STS technique. To arrive at these results, we counted all operations, including additions,

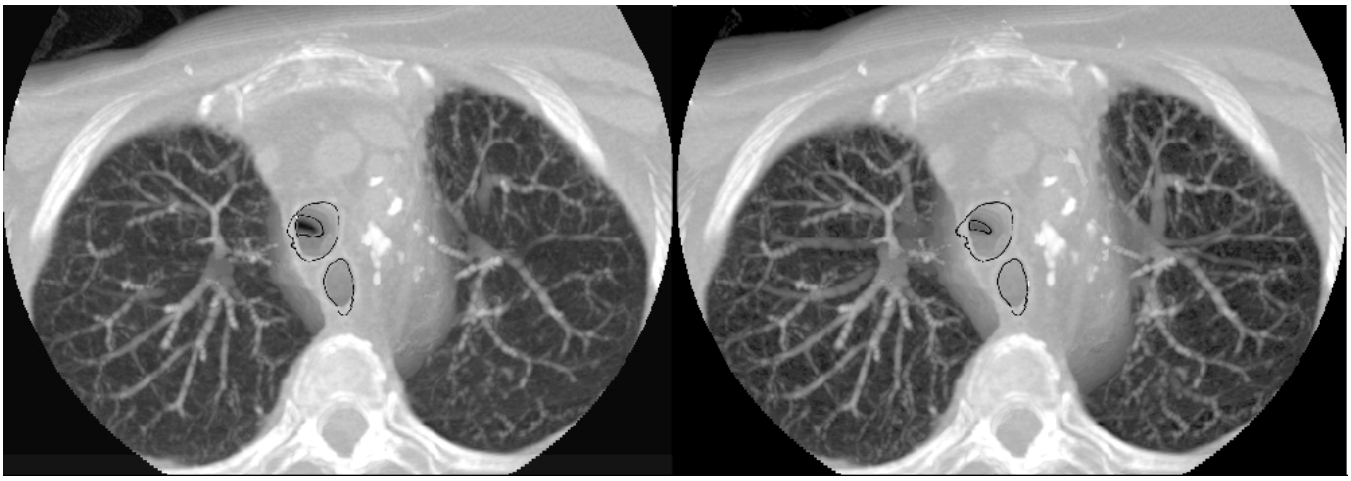


Fig. 15. Combination of STS-EG and STS-DWMax views. This figure combines the bottom two rows of Figure 14. For emphasis, we manually edited the windowed EG views, before superimposing them on the DWMax views, to focus on the EG borders for the trachea, impinging cancer, and esophagus.

subtractions, multiplications, divisions, logical operations, assignments, and memory retrievals, per the algorithms of Figures 2, 3, 5, and 7-8. All operations were assumed to have the same cost. Tables I and II summarize these results. More discussion appears below and complete details appear in [28].

The calculations assume $x \times y$ slices and a slab sequence S containing t slabs. The fast STS procedure requires a combination of brute-force processing and fast processing, as depicted in Figures 2-3. To compute a particular slab s_i in the sequence S, two cases can occur:

1. WORST case – the slab s_i requires brute-force processing. This happens when the sliding field of view V_i drops off a slice of data that forces a complete redefinition of the slab's values from the previous slab s_{i-1}.
2. TYPICAL case – the slab s_i can be computed with fast processing. This happens when the present slab's values are based on the same data existent in the previous slab's field of view V_{i-1} or if the new slice appended to V_i determines the slab's values.

With a slab thickness d_s, a reasonable assumption is that every d_s^{th} slab on the average requires WORST case, or brute-force, computation in the fast procedure. Thus, the operation count for a particular method to compute a sequence of t slabs using slab thickness d_s is given by

$$\text{operation count} \;=\; \frac{t(d_s-1)}{d_s}(\text{TYPICAL}) + \frac{t}{d_s}(\text{WORST}) \tag{7}$$

where TYPICAL and WORST are replaced by specific counts depending on the technique considered. Table I gives the raw operation counts, based on this set-up. Column 2 assumes a complete brute-force calculation. Column 3 gives the results for the fast procedure, based on the algorithms of Figures 2 and 3 and the assumption (7). The last column of Table I gives the limit of the ratio of pure brute-force computation and fast processing, as the slab thickness gets large; as is apparent, the fast approach

typically is much faster than pure brute-force calculation. Table II bears this conclusion out with several specific values of d_s. This table works in the necessity of always having to do some brute-force processing. Clearly, as slab size increases, the amount of brute-force processing required decreases. In all cases the fast methods decrease the computational burden by a factor greater than 2.3. For all methods, the improvement over brute force computation increases almost linearly as the slab size increases.

Technique	Brute-Force	Fast Processing (uses eq. (7))	$\frac{\text{brute force}}{\text{fast}}$	$\lim_{d_s\to\infty}\frac{\text{brute force}}{\text{fast}}$
DWmax	$txy(1+10d_s)$	$\frac{t(d_s-1)}{d_s}(3+14xy)+\frac{t}{d_s}(3+2xy(7+3d_s))$	$\frac{xy(1+10d_s)}{3+20xy}$	$\frac{d_s}{2}$
EG	$txy(9+8d_s)$	$\frac{t(d_s-1)}{d_s}(4+13xy)+\frac{t}{d_s}(4+xy(13+6d_s))$	$\frac{xy(9+8d_s)}{4+19xy}$	$\frac{8d_s}{19}$
STS-MIP	$txy(1+4d_s)$	$\frac{2t(d_s-1)}{d_s}(2+3xy)+\frac{t}{d_s}(4+3xy(2+d_s))$	$\frac{xy(1+4d_s)}{4+9xy}$	$\frac{4d_s}{9}$

TABLE I

Operation Count Analysis for STS algorithms. A slab volume made up of t slabs and $x \times y$ values per slab is assumed. The analysis of (7) is used to arrive at the Fast Processing calculations.

Slab Technique	DWmax		EG		STS-MIP	
Slab Size	speed-up factor	% Brute time	factor	% Brute	factor	% Brute
$d_s = 5$	2.55	39.22	2.58	38.76	2.33	42.92
$d_s = 7$	3.55	28.17	3.42	29.24	3.22	31.06
$d_s = 19$	9.55	10.47	8.47	11.81	8.56	11.68
$d_s = 27$	13.55	7.38	11.84	8.45	12.11	8.26

TABLE II

Computation Efficiency of Fast STS algorithms for 512×512 slices and various d_s. Along with each speed-up factor (first column for each technique), a percentage of brute-force processing time required to process the fast algorithm has been given (second column for each technique).

V. Discussion

A. Dynamic Viewing of Slabs

Remy-Jardin *et al.* evaluated the clinical value of STS-MIP for CT scans and found "...sliding-thin-slab MIPs enabled a more precise characterization of the distribution of lung changes than did 1-mm-thick conventional CT scans" [6]. We believe this to be true to a great extent for both the DWmax and EG. The information provided by DWmax and EG can be comprehended most effectively when a consecutive sequence of slabs are viewed dynamically. Utilizing the mind's quick response to visual

input, a viewer's immediate attention is drawn to changes in slab content that occur from slab to slab. The more striking and abnormal, the more a change will "jump out" like a beacon. Thus, dynamic viewing leads to a quick and effective recognition of the specific locations that need closer scrutiny.

B. Slab Size

Slab size d_s is an important parameter for the observation of structures within a slab volume. A clinical study on the ability of STS-MIP to detect mild micronodular patterns led to the conclusion that slabs thinner than micronodular patterns are insufficient [6]. Our experience with the EG and DWMax methods leads us to a similar belief. We found a small structure is best detected if it provides one of the intensity extremes that define the EG value. This is most likely to occur when the slab is no bigger than what is necessary to enclose the structure, because the structure then can not be obstructed by the presence of additional structures in the data. For a larger structure, where the slab encloses only a part of the structure, the tissue is shown as an outlined area of constant value (see Figure 14). EG slab-size variations affect the width of the evolution patterns, not its EG value.

The question of slab size d_s is less vital for the DWmax method. DWmax allows a large range of structures (values) to impact the slab, because DWmax's depth weighting removes the effect of blind maximum decisions (the basis for STS-MIP slabs). A small structure's impact is partly decided by its location in the slab, since its distance from the base slice determines the degree its intensity will be scaled. Most importantly, a small structure has its strongest impact when it is near or passes through the base slice of a slab. Thus, small structures can "come into view" as one dynamically observes a slab sequence. The additional information shown by DWmax is striking when compared to STS-MIP results (see Figure 9). For larger structures such as airways, DWmax provides the best view of interior details when the structure is only partially contained in the slab, as in Figures 9-11 and Figure 14.

C. Meeting the Shortcomings of Present Volumetric CT Techniques

Naidich *et al.* [7] pointed out the limitations of current volumetric CT processing techniques. The DWmax and EG methods do not use these techniques. Specifically, intensity thresholding (i.e., voxel-value dependent processing) is not needed for 1) thin-slice parallel processing along standard axes, 2) slice-level weighting that creates DWmax slabs, or 3) the determination of EG extremes. Both DWmax and EG incorporate the entire intensity range of the incoming data. Neither intensity windowing nor thresholding is used. DWmax and EG do not use partial-volume processing, which bring in noise and certain artifacts into standard 3D reconstruction. Furthermore, the DWmax and EG techniques produce true image interfaces versus the production of voxel boundaries and blending artifacts that are characteristic of 3D reconstruction [7]. In addition, the DWmax and EG methods require very few parameters. Therefore, our proposed methods are far less a function of user decisions than other

techniques.

D. Final Comments

DWmax weighting decisions are determined by window slice locations only. Therefore, a single set of column weights apply to all columns, without regard to column location in the processing window or window location in the image. The EG method extends slab information two ways. First, structural details and changes, not found in DWmax results, are captured by the EG "evolution map" (in one slab or in a set of consecutive slabs). Second, the EG method produces structural edges.

To date, our testing has focused on 3D CT pulmonary images. Further work is needed in developing protocols for specific medical diagnostic procedures. Our present experience shows that the DWmax and EG methods have great promise in filling the emerging need of assessing very large 3D volumes, as produced by new multi-detector helical CT scanners. DWmax slab views are strongly reminiscent of standard 2D CT slices. Hence, the interpretation of DWmax slabs requires a minimal learning curve and allow past experience to be of direct benefit. The fast algorithms proposed for DWmax, EG, and the standard STS-MIP rapidly produce slab sequences on a general-purpose computer. This makes the real-time analysis of a given 3D volume image feasible, as we are doing in a system for virtual bronchoscopy [24].

References

[1] J Z Turlington and W E Higgins, "Improved techniques for fast sliding thin-slab volume visualization," *SPIE Medical Imaging 2000: Image Display and Visualization*, vol. 3976, pp. 110–122, S.K. Mun, ed., 2000.

[2] D R Ney, E K Fishman, and D Magid, "Volumetric rendering of computed tomography data: Principles and techniques," *IEEE Computer Graphics and Applications*, pp. 24–32, March 1990.

[3] K H Hoehne(sic), R L Delapaz, R Bernstein, and R C Taylor, "Combined surface display and reformatting for the three-dimensional analysis of tomographic data," *Invest. Radiol.*, vol. 22, pp. 658–664, Aug. 1987.

[4] S Napel, M P Marks, and G D Rubin, et al., "CT angiography using spiral CT and maximum intensity projections," *Radiology*, vol. 185, pp. 607–610, 1992.

[5] R A Robb and C Barillot, "Interactive 3-D image display and analysis," *SPIE Symp. Hybrid Image Signal Proc.*, vol. 939, pp. 173–202, 1988.

[6] M Remy-Jardin, J Remy, D Artaud, F Deschildre, and A Duhamel, "Diffuse inflitrative lung disease: clinical value of sliding-thin-slab maximum intensity projection CT scans in the detection of mild micronodular patterns," *Radiology*, vol. 200, pp. 333–339, August 1996.

[7] D P Naidich, J F Gruden, G McGuinness, D I McCauley, and M B Bhalla, "Volumetric CT (VCT) of the airways," *J. Thoracic Imag.*, vol. 12, no. 1, pp. 11–28, March 1997.

[8] A K Hara, C D Johnson, J E Reed, D A Ahlquist, H Nelson, R L Ehman, C M McCollough, and D M Ilstrup, "Detection of colorectal polyps by computed tomographic colography: feasibility of a novel technique," *Gastroenter.*, vol. 110, no. 1, pp. 284–290, Jan. 1996.

[9] K Ramaswamy and W E Higgins, "Interactive dynamic navigation for virtual endoscopy," *Computers in Biology and Medicine*, vol. 29, no. 5, pp. 303–331, Sept.-Oct. 1999.

[10] R A Drebin, L Carpenter, and P Hanrahan, "Volume rendering," *Comp. Graph.*, vol. 22, no. 4, pp. 65–74, August 1988.

[11] U Tiede, K H Hohne, M Bomans, A Pommert, M Riemer, and G Wiebecke, "Investigation of medical 3D-rendering algorithms," *IEEE Comput. Graph. Applic.*, vol. 9, no. 2, pp. 41–53, Mar. 1990.

[12] D J Vining, R Y Shifrin, E K Grishaw, K Liu, and D W Gelfand, "Virtual colonoscopy," *Radiol. Scient. Prog.*, vol. 193, pp. 446, 1994.

[13] R A Robb, *Three-Dimensional Biomedical Imaging: Principles and Practice*, VCH Publishers, New York, 1994.

[14] E A Hoffman, D Gnanaprakasam, K B Gupta, J D Hoford, S D Kugelmass, and R S Kulawiec, "VIDA: An environment for multi-dimensional image display and analysis," *SPIE Proc. Biomed. Image Proc. and 3D Microscopy*, vol. 1660, pp. 694–711, 1992.

[15] W E Lorensen, F A Jolesz, and R Kikinis, "The exploration of cross-sectional data with a virtual endoscope," *Interactive Technology and the New Health Paradigm*, pp. 221–230, Jan. 1995.

[16] D J Vining, R J Zagoria, K Liu, and D Stelts, "CT cystoscopy: an innovation in bladder imaging," *Am. J. Roentgen.*, vol. 166, no. 2, pp. 409–410, Feb. 1996.

[17] R M Summers, "Navigational aids for real-time virtual bronchoscopy," *Am. J. Roentgenology*, vol. 168, no. 5, pp. 1165–1170, May 1997.
[18] W E Higgins, K Ramaswamy, R Swift, G McLennan, and E A Hoffman, "Virtual bronchoscopy for 3D pulmonary image assessment: State of the art and future needs," *Radiographics*, vol. 18, no. 3, pp. 761–778, May-June 1998.
[19] S Napel, G D Rubin, and R B Jeffrey, "STS-MIP, a new reconstruction technique for CT of the chest," *Journal of Computer Assisted Tomogr*, vol. 17, no. 5, pp. 832–838, 1993.
[20] S Y Yen, G D Rubin, and S Napel, "Fast sliding thin slab volume visualization," *Proceedings of Symposium on Volume Visualization*, pp. 79–86, 1996.
[21] S Y Yen, G D Rubin, and S Napel, "Fast sliding thin slab visualization of CT and MR angiograms," *RSNA EJ*, vol. 1, pp. 832–838, March 1997.
[22] E G McFarland, J A Brink, and J P Heiken, et al., "Spiral CT colonography (virtual colonoscopy): Multiobserver study of different image display techniques compared to colonoscopy," in *SPIE Medical Imaging 1999: Physiology and Function from Mulitidimensional Images*, C.T. Chen and A.V. Clough, Eds., pp. 106–108. vol. 3660, Feb, 1999.
[23] S Huang, M Horng, and Y Sun, "Index based interpolation for maximum intensity projection display," *SPIE Proceedings, Medical Imaging 1997: Image Display*, vol. 3031, pp. 563–570, Feb. 1997.
[24] A J Sherbondy, A P Kiraly, A L Austin, J P Helferty, S Wan, J Z Turlington, E A Hoffman, G McLennan, and W E Higgins, "Virtual bronchoscopic system combining 3D CT and endoscopic video," *SPIE Medical Imaging 2000: Physiology and Function from Multidimensional Images*, vol. 3978, pp. 104–116, A. Clough and C.T. Chen, eds., 2000.
[25] B Gudmundsson and M Randen, "Incremental generation of projections of CT volumes," in *Proc. Visual. in Biomed. Computing*, Atlanta, GA, May 1990, pp. 27–34.
[26] D N Laredo, "Relaxation labeling for 3D anisotropic image segmentation," *M.S. thesis, Penn State University*, August 1997.
[27] R C Gonzalez and R E Woods, *Digital Image Processing*, Addison Wesley, Reading, MA, 1992.
[28] J Z Turlington, "Fast techniques for 3D thin-volume visualization," *M.S. thesis, Penn State University*, May 2000.