Shape Based Computer Aided Diagnosis and Automated Navigation in Virtual Colonoscopy

József Koloszár, László Szirmay-Kalos, Zsolt Tarján, Dávid Jocha
Department of Control Engineering and Information Technology
Budapest University of Technology, Hungary
Email: szirmay@iit.bme.hu

Abstract

This paper proposes a robust algorithm to detect colon polyps and cancerous lesions in virtual colonoscopy and present them to the user by automatically guiding the virtual camera. The detection algorithm uses Gaussian filters to construct the Hessian matrix, which represents the second order derivatives of a vector variate scalar valued function. Based on the sign and scale of the eigenvalues of the Hessian matrix, blob like lesions can be selected on a given scale. In the visualization stage the camera is moved along the colon centerline with its speed and viewing direction adopted to the results of detection. The camera path and the viewing direction are described by Kochanek-Bartels splines. The velocity along the path is also governed by a C^2 continuous function. The resulting fly through is smooth and physically plausible, and it is guaranteed that the user can see all regions of interest and spends sufficient time looking at each of them.

1 Introduction

Virtual colonoscopy (VC) [Hong et al. 1997; Wegenkittl et al. 2000] is a non-invasive computerized medical diagnostic method for examining the interior of the human colon from Computer Tomography (CT) data, that is primarily aimed at the detection of polyps on the colon surface.

CT images provide density values of the scanned object at regularly placed locations. The density values are stored in a voxel array. Voxel arrays are visualized by direct or indirect algorithms, allowing the medical doctor to examine the measured data [Drebin et al. 1988; Levoy 1990; Csébfalvi 2001]. Based on the presented images, the physician identifies tumors and decides on the subsequent actions. This process can be speeded up and made easier if the data is preprocessed and the attention of the physician is drawn to those critical points which are primary candidates for being tumors. This helping process is called *computer aided diagnosis (CAD)*.

VC applications are usually based on providing automatic fly-through of the colon, letting the user navigate the colon interactively, or a combination of both. The major disadvantage of automatic fly-through is that regions of the colon wall are not presented in enough detail to support the diagnosis, or even completely missed because of the complex curvatures of the colon and the limited field of view of the camera. Workarounds for this problem include tiling views from multiple viewing directions into a collated super-wide angle view and multiple fly-throughs. The drawbacks of completely interactive examination are the extended time it takes to investigate the entire colon, and the training required to acquaint the physician with the non-medical skills necessary to use the application. Studies suggest that virtual colonography based examinations take 15 to 35 minutes on average [Burling 2004], depending on the tool and technique used and user experience. In order to facilitate screening for colorectal cancer, which is one of the goals of VC research, examination times in the 5 – 10 minutes range are desirable. Examination time can be reduced by incorporating CAD into the system, provided the algorithms are fast enough. Automatic diagnosis may result in two types of false decisions. The decision is false positive if it identifies a shape as a tumor, which is not. On the other hand, the decision is false negative if the process misses a tumor. In medical diagnosis the reduction of false negative cases is essential. Results obtained from automated detection must be reviewed by the physician to rule out false positives, and at least a quick review of the entire colon is still recommended until such systems are proven to be 100 percent sensitive. Thus the need for efficient navigation is not (yet) eliminated by implementing CAD.

In this paper we present a shape based algorithm to detect potential polyps inside a colon, and then use the results to plan assisted camera movement to investigate the regions of interest. If this investigation were made by a medical doctor, he would search for blob like structures on the colon walls, and examining the size, shape and density of these blobs he could decide whether or not the particular blob is a tumor. In order to mimic this procedure, we design a recognition filter that is particularly sensitive to blobs of given size and provides numeric data representing the form of the blob. The recognition filter is based on the second order derivatives of the reconstructed density function. To review the results the camera is guided to mimic manual control using common sense: a fast review of regions of no interest and thorough look at all suspicious regions. Smooth and life-like camera movement throughout the review is ensured by describing the path, viewing direction, and speed by C^2 continuous spline functions.

The structure of this paper is the following. Section 2 discusses previous work on detection algorithms. Section 3 describes our algorithm for detecting polyps in VC datasets. Our camera guidance technique is presented in Section 4. Results are discussed in Section 5 followed by conclusions and notes on future work in Section 6.

2 Previous Work

While shape analysis and shape detection have received significant attention in 2D image processing, they are relatively new in 3D volume analysis. Two dimensional shape detection methods usually measure certain features that can characterize the shapes of interest and make decisions based on the measured values. These features are required to be translation and rotation invariant since translation and rotation do not alter shapes. The practically useful features include, among others, perimeter, area, circularity, spatial moments, main inertia, derivatives, etc.

Having computed the feature values, they are grouped in a vector and this vector is compared with the properties of the shape or shapes to be identified. If the distance from the feature vector of the shape to be identified is small, then we can report that the shape is found. Setting the distance threshold, we can make a compromise between false negative and positive decisions. When such algorithm is designed, the critical problems is the definition of the feature vector, especially when the class of target shapes is not well defined. This is usually the case in medical diagnosis since different polyps are not necessarily similar from geometric point of view.

The compromise between the complexities of feature extraction and decision making greatly impacts performance and practical feasibility. Large feature vectors such as unprocessed local pixel neighborhoods are easily obtained but require more sophisticated classifier automata, such as neural networks trained with rotational variants of numerous candidate shapes to make well-grounded decisions. Gokturk et al. [Gokturk et al. 2001] proposed a technique that uses more elaborate feature extraction (shape signature histogram) and a *Support Vector Machine* [Cristianini and Shawe-Taylor 2000] for learning.

Our technique uses a four dimensional feature vector comprised of the eigenvalues of the Hessian matrix obtained through local measures, and the average density of the candidate shape. Classification is based on simple multi-dimensional thresholding, thus eliminating the need for complex intelligence and training.

2.1 Local Measures

Those features that are based on the Taylor series expansion of the volume data are called local measures since the Taylor series is a good local approximation of a function. For example, the second-degree approximation of a function $f(x)$ around x_0 is:

$$f(x) \approx f(x_0) + \left.\frac{df}{dx}\right|_{x_0} \cdot (x - x_0) + \frac{1}{2} \cdot \left.\frac{d^2f}{dx^2}\right|_{x_0} \cdot (x - x_0)^2.$$

This approximation depends on function value $f(x_0)$, derivative df/dx and second derivative d^2f/dx^2. The first derivative is the slope of the function. The second derivative is the speed describing how fast the slope of the function is changing, thus is a measure of the function curvature. In fact, the absolute value of the second order derivative is inversely proportional to the radius of a tangent circle. The sign of the second derivative also shows whether the high curvature area is a *hill* or a *valley*. Sato et al. have extended this idea to the analysis of 3D density data [Sato et al. 2000; Hladuvka 2002]. Suppose that the density value of the examined object at point $\vec{r} = (x, y, z)$ can be described by function $g(\vec{r})$.

The second-order approximation of the density function around $\vec{r}_0$ is the following:

$$\tilde{g}(\vec{r}) = g(\vec{r}_0) + (\vec{r} - \vec{r}_0)^T \cdot (\nabla g)(\vec{r}_0) +$$
$$\frac{1}{2} \cdot (\vec{r} - \vec{r}_0)^T \cdot (\nabla^2 g)(\vec{r}_0) \cdot (\vec{r} - \vec{r}_0),$$

where

$$\nabla g = (\partial g/\partial x, \partial g/\partial y, \partial g/\partial z)$$

is the *gradient vector* and

$$\nabla^2 g = \begin{bmatrix} \partial^2 g/\partial x^2, & \partial^2 g/\partial x\partial y, & \partial^2 g/\partial x\partial z \\ \partial^2 g/\partial y\partial x, & \partial^2 g/\partial y^2, & \partial^2 g/\partial y\partial z \\ \partial^2 g/\partial z\partial x, & \partial^2 g/\partial z\partial y, & \partial^2 g/\partial z^2 \end{bmatrix} \qquad (1)$$

is the *Hessian matrix*. Unfortunately, the gradient and the Hessian matrix are not invariant under rotations, and they strongly depend on the axes of the coordinate system. Such dependence can be eliminated for the gradient if its absolute value is considered since $|\nabla g|$ becomes independent of the orientation of the axes. In order to make the second derivative measures also rotation independent, we have to select those directions, which correspond to the maximal and minimal second derivatives. As can be proven [Hladuvka 2002], these directions are the eigenvectors of the Hessian matrix, where the corresponding second derivatives are the eigenvalues. Since the Hessian matrix is symmetric, the eigenvalues are real. Let us denote the three eigenvalues by $\lambda_1, \lambda_2, \lambda_3$, and assume that $\lambda_1 \geq \lambda_2 \geq \lambda_3$. These eigenvalues express the minimum and maximum of the second derivatives, or the maximum and the minimum of the curvature radii. If the three eigenvalues have similar sign, then the function is locally approximated by an ellipsoid. If the sign is negative, then we have a hill, otherwise a valley. The minimum and maximum curvatures of an ellipsoid are proportional to the lengths of the main axes. A small $|\lambda|$ value corresponds to a long axis, while a large $|\lambda|$ to a short axis. If all axes are short (i.e. the absolute values of all eigenvalues are large), then the ellipsoid is a small blob (figure 1). If all axes are long, then the ellipsoid cannot be recognized, and is said to be homogeneous on the local level. If one axis of an ellipsoid is much longer than the other two (i.e. the absolute value of one eigenvalue is much smaller than the other two), then the ellipsoid has a tubular shape. If two axes are much longer than the third one, then the ellipsoid looks like a sheet.

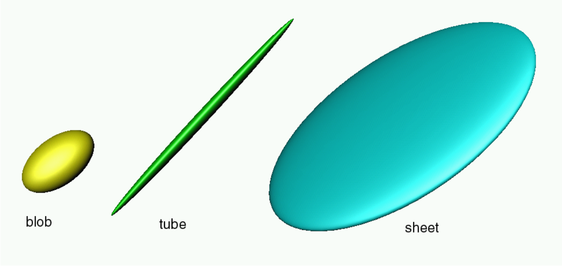

Figure 1: Local quadratic approximation. Based on the relative absolute values of the eigenvalues, blob, tubular and sheet-like structures can be identified.

Table 1 summarizes the possible conclusions that can be drawn by inspecting the three eigenvalues.

	classification
$\lambda_3 \approx \lambda_2 \approx \lambda_1 \ll 0$	blob-like hill
$\lambda_3 \approx \lambda_2 \ll \lambda_1 \approx 0$	tubular-like hill
$\lambda_3 \ll \lambda_2 \approx \lambda_1 \approx 0$	sheet-like hill
$0 \approx \lambda_3 \approx \lambda_2 \approx \lambda_1$	homogeneous area
$0 \approx \lambda_3 \approx \lambda_2 \ll \lambda_1$	sheet-like valley
$0 \approx \lambda_3 \ll \lambda_2 \approx \lambda_1$	tubular-like valley
$0 \ll \lambda_3 \approx \lambda_2 \approx \lambda_1$	blob-like valley

Table 1: Classification of the blob-, tubular-, and sheet-like structures

2.2 Computation of the derivatives of sampled datasets

The measuring process takes discrete samples at regular points $\vec{p} \in P$, thus the function stored in the voxel array is:

$$g_s(\vec{r}) = g(\vec{r}) \cdot \sum_{\vec{p} \in P} \delta(\vec{r} - \vec{p})$$

where δ is the Dirac-delta function. From the sampled signal, the continuous signal can be reconstructed by convolving $g_s(\vec{r})$ with the impulse response of an ideal low-pass filter of limiting frequency equal to the upper band limit of the original signal (the sampling theorem requires the band limit to be lower than the half of the sampling frequency).

The impulse response of the ideal low-pass filter is $sin(\pi x)/\pi x$, which is rather difficult to convolve with, because of its oscillating shape and infinite support. Due to computational issues, the ideal low-pass filter is approximated, for example, by the Gaussian-filter [Mitchell and Netravali 1988; Marschner and Lobb 1994; Möller et al. 1998], thus the reconstructed signal is:

$$g(\vec{r}) \approx (k \otimes g_s)(\vec{r})$$

where k is the impulse response (kernel) of the 3D Gaussian-filter having standard deviation σ:

$$k(\vec{r},\sigma) = \frac{e^{-|\vec{r}|^2/2\sigma^2}}{(\sigma\sqrt{2\pi})^3} = \frac{e^{-x^2/2\sigma^2}}{\sigma\sqrt{2\pi}} \cdot \frac{e^{-y^2/2\sigma^2}}{\sigma\sqrt{2\pi}} \cdot \frac{e^{-z^2/2\sigma^2}}{\sigma\sqrt{2\pi}}.$$

Standard deviation σ should be set to make the filter efficiently cut off frequencies higher than the upper band limit of the measured signal. If the upper band limit is about half of the sampling frequency, which is usually 1 (the samples are at unit distance), then $\sigma \in [0.5, 1]$ is usually satisfactory. If σ is smaller than 0.5, then high-frequency aliasing occurs. On the other hand, if σ is bigger than 1, the higher frequency components of the signal are eliminated. Large σ values represent filters that are combinations of a low-pass filter and the reconstruction filter.

Convolution commutes with differentiation, thus the derivatives of the reconstructed signal can be obtained by convolving with the derivative of the filter kernel. For example, the derivative of $g(\vec{r})$ according to x is (the derivatives according to y and z are similar):

$$\frac{\partial g(\vec{r})}{\partial x} \approx \frac{\partial (k \otimes g_s)(\vec{r})}{\partial x} = \frac{\partial k(\vec{r})}{\partial x} \otimes g_s.$$

Thus to compute all second partial derivatives, the partial derivations of the filter kernel $(\partial^2 k/\partial x^2, \partial^2 k/\partial x \partial y, \partial^2 k/\partial x \partial z, \ldots)$ should be pre-computed, and the sampled data is convolved with the required derivative kernels. The Gaussian-kernel as well as its derivatives are separable, which means that the three-variate function can be expressed as the product of three one-variate functions parameterized by x, y and z, respectively.

3 Polyp Detection Algorithm for VC

Colon polyps are blob-like features in a given size range. The proposed detection algorithm looks for cases corresponding the first row of table 1, but also takes into account the average density as well as the size of the lesion. The following subsection discusses the issue of sensitivity and size.

3.1 Constructing the Detection Filter

The Gaussian-kernel is not a perfect reconstruction kernel, thus the derivatives are just approximations. Standard deviation σ can be used to control the possible range of the derivatives. If we are interested in features where the second derivatives are in a prescribed range, then σ should be set to make the maximum derivatives close to the top of the prescribed range. In this way we can maximize the sensitivity of our filter. This can mean that σ is greater than the value needed for the optimal reconstruction, thus all those features that are smaller than the interesting ones are eliminated.

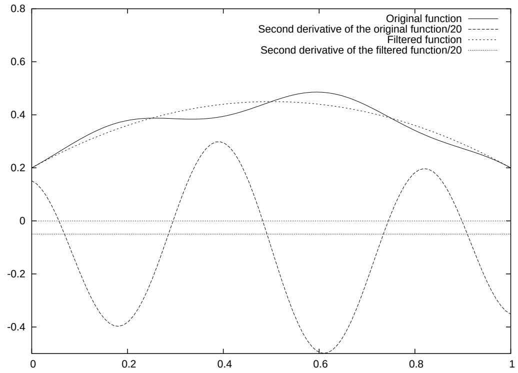

Figure 2: Low pass filtering makes larger features detectable. Note that the second derivatives are scaled in the figure to improve visual comprehension.

Figure 2 demonstrates why the low pass filtering is necessary to allow the detection of larger features. The original function contains a single hill, which is modulated by a high-frequency wave. The curvature values fluctuate according to the high-frequency wave, not permitting to detect the average curvature that corresponds to the hill. However, when low-pass filtering eliminates the high frequency wave, the curvature clearly identifies the original hill. From this observation we can conclude that a crucial part of feature recognition is the definition of the appropriate scale or scales if we examine different sizes simultaneously. Sato et al. approached this problem assuming that the signal is also a Gaussian function. However, in tumor diagnosis, this assumption is not necessarily true, since tumors have a well recognizable boundary.

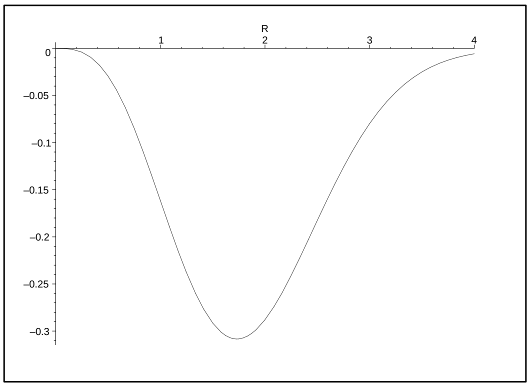

Figure 3: The eigenvalue at a center of a sphere of radius $r = R\sigma$ computed by a Gaussian filter of standard deviation σ

Thus to present a more accurate analysis, let us suppose that the volume data contains a sphere of radius r. The density value is 1 inside the sphere and zero outside. Let us also assume that the sampling frequency is high enough to make the discrete sums close to the continuous integrals.

The first and second derivatives at the center of the sphere are approximated by the convolution of the density function and the filter kernel $\frac{dk(x,\sigma)}{dx} \cdot k(y,\sigma) \cdot k(z,\sigma)$ and $\frac{d^2k(x,\sigma)}{dx^2} \cdot k(y,\sigma) \cdot k(z,\sigma)$, respectively. Consequently, the eigenvalue can be expressed by the following integral:

$$\lambda(r) = \int\limits_{x^2+y^2+z^2 \le r^2} \frac{(x^2-\sigma^2)\cdot e^{-(x^2+y^2+z^2)/2\sigma^2}}{\sigma^7 (2\pi)^{3/2}}\, dxdydz.$$

Applying the $X = x/\sigma, Y = y/\sigma, Z = z/\sigma, R = r/\sigma$ substitutions, we obtain:

$$\lambda(R) = \int\limits_{X^2+Y^2+Z^2 \le R^2} \frac{(X^2-1)\cdot e^{-\frac{X^2+Y^2+Z^2}{2}}}{\sigma^2 (2\pi)^{3/2}}\, dXdYdZ$$

$$= -\frac{1}{\sigma^2}\cdot\sqrt{\frac{2}{9\pi}}\cdot e^{-R^2/2}\cdot R^3$$

Note that product $\lambda\sigma^2$ is a function of $R = r/\sigma$, which is depicted in figure 3. This function has a minimum at $R = r/\sigma = \sqrt{3}$, which corresponds to the largest curvature the filter can appropriately reconstruct. If $r/\sigma > \sqrt{3}$, then the curvature of the sphere decreases, and the reconstructed value follows this change. However, when $r/\sigma < \sqrt{3}$, the curvature of the sphere increases, but the absolute value of the reconstructed eigenvalue decreases. Here, the Gaussian acts as a low pass filter, and instead of computing the second derivative, it gradually eliminates the sphere itself.

Suppose that the smallest feature that should be detected can be approximated by a sphere of radius r. Figure 3 tells us to set the standard deviation of the Gaussian filter to $\sigma \approx r/1.74$. Such setting allows us to detect features of radius $[r, 3r]$ while eliminating features of radius smaller than r. Thus, the size of features is implicitly conveyed to the feature vector by the magnitude of the eigenvectors. Considering the example of testing spherelike structures of different radius with the same kernels of $\sigma \approx r/1.74$ one finds that the absolute values of eigenvectors decrease as the sphere radius diverges from r.

There is one final issue of the Gaussian-kernel, which should be addressed here. The Gaussian-kernel has infinite support, which is impossible to handle during convolutions. To cope with this problem, the Gaussian-kernel is truncated, which means that it is assumed to be zero, where the original kernel is small. The size of the support domain depends on the standard deviation σ. For example, if $\sigma = 0.5$, then the kernel values at the sampling positions $0, 1, 2$ are $0.797, 0.108, 0.00027$, respectively. It means that we can suppose that the kernel is zero outside region $[-2, 2]$. Larger σ values, however, require filters with greater support. If truncation happens at value $t = x/\sigma$, then the integral of the cut off part is $1-\Phi(t)$, where Φ is the Gaussian distribution function. As a rule of thumb we can truncate at $t = 4$, i.e. $x = 4\sigma$ since in this case the integral of the cut off part is less than 0.00003.

The truncation of the kernel might have the undesirable effect that the reconstructed second derivatives of constant functions will not be zero. This is due to the fact that the integral of the second derivative of the truncated kernel is not necessarily zero. Note that this can never happen when the first derivative is computed since the first derivative is anti-symmetric ($dk(-x,\sigma)/dx = -dk(x,\sigma)/dx$). To solve the problem of second derivatives, σ should be fine tuned in a way that the sampled second derivatives of the truncated kernel sums up to zero.

3.2 Implementation

In the preprocessing phase the standard deviation σ is approximated from the expected size of the tumors relative to the sampling resolution of the device. If it turns out that σ is too small (smaller than 0.5), then the criteria of the sampling theorem cannot be met, thus such features cannot be recognized from the given data set. If σ is not too small, then the corresponding window size is determined, and σ is tuned to give zero second derivatives for homogeneous regions.

For purposes of VC, it is desired to detect polyps of size 5 mm and up, as most studies agree that polyps greater than 10 mm constitute significant risk of cancer development [Fuente and Valverde 2004]. We found that choosing the 5 mm as the target lesion radius gives good sensitivity for targeted polyp sizes, the range most interesting polyps fall into. The detector is primed by the largest eigenvalue being smaller than -0.05. Note that a bull's-eye hit would be on a perfect sphere of radius 5 mm resulting in about -0.3 for the eigenvalue (see previous subsection). The threshold proposed accounts for polyps not being perfect spheres (they are all attached to the colonic wall) of the right size, without introducing too many false positives. The threshold has been determined empirically through our testing of the algorithm. A more mathematically rooted definition for this value is to be looked into in the future. To perform a more rigorous automated checking for smaller lesions, one can repeat the filtering with another appropriately sized kernel. The same can be done for larger polyps as well, though we suggest that lesions larger than 15 mm are readily identified even in a fast fly-through review of the dataset.

The diagnosis process starts with identifying the voxels corresponding to the colon boundary. This way the feature detection can be restricted to the neighborhood of the colon walls, since colon tumors can show up only at these regions. In the selected neighborhoods, the shape detection filter is run at voxels on regular grid with $r/3$ mm grid spacing to further reduce unnecessary computations. Implicit binarization of density values (at -750 Hounsfield) is also introduced in the filtering process to maximize air-tissue contrast. If the maximum eigenvalue primes the detection as described above, the average density of tissue in the region is calculated and compared against an appropriate threshold (0 Hounsfield in our case where no contrast enhanced fecal residue was present in the scans). If both the density and the shape are suspicious, the region is marked as a lesion. Our VC engine uses first-hit raycasting based iso-surface rendering of the colon. As a visual cue the neighborhood of lesions is colored red on the 3D endoscopic view (figure 6).

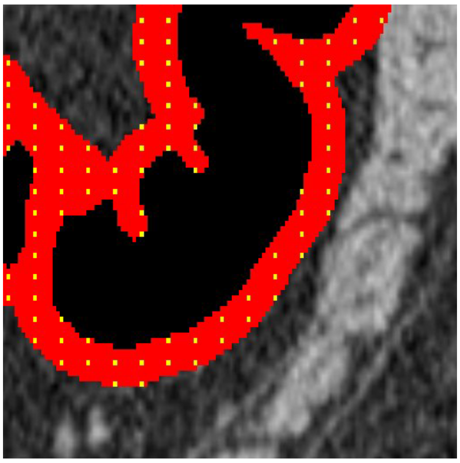

Figure 4: Cross-section of the Human dataset. Regions of interest (the colon wall) is colored red. The yellow pixels indicate voxels that will be tested by the detection filter.

4 Camera Guidance

Once suspicious regions are identified they are to be shown to the user in a fly-through. The first step, which is independent of the detection process, is to extract the centerline of the colon from the data [Bitter et al. 2000; Palágyi et al. 2001]. This is done by an automated fast discrete topological thinning method. Once voxels corresponding the centerline are known, Kochanek-Bartels spline [Kochanek 1984] $S_p(t)$ is constructed by taking the position every nth extracted voxel as the control points for the spline with t values of $1,2,3$ etc.

After detection, the "best view" position along the centerline (t_i, $i = [1,\ldots,N]$, N being the number of lesions) is calculated for every lesion (t is the running parameter for the spline describing the centerline). In our implementation the "best view" is defined as the position from where the camera can have the closest unobscured look at the lesion in question (visibility is tested by casting a few rays using the ray-casting engine).

As the detection algorithm may report hits at more than one sampling points inside the same lesion, groups of hit locations are identified, and the lesion is considered to be centered at the center of mass of the group. Next, an exploration run with $\Delta t = 1$ is performed along the centerline, checking and recording "best view" positions for all lesions. Lesions are sorted according to their t_i values. Rarely there are more lesions sharing a common t value (meaning they are both best viewed from the same point of the centerline in the exploratory run). If this is the case the situation is resolved by spreading the t values regularly in the interval $(t-0.5, t+0.5)$.

The velocity with which the camera is to move along the centerline should be determined by the proximity of lesions. With no lesions to look at, the camera should proceed with constant speed v_0 (i.e. a constant sampling interval along the centerline spline). The objective is to construct velocity function $v(t)$ with the following properties:

1. speed v_0 where no lesions are in proximity,
2. speed v_0/k at "best view" point of a lesion,
3. speed $v_0/(i \cdot k)$ on intervals from which i lesions must be looked at,
4. smooth C^2 properties for physical plausibility.

The third requirement is of special importance, because it implies that the camera always has roughly the same amount of time to show each lesion, if their "best view" positions are stacked closely. Our choice of such a function is:

$$v(t) = \frac{v_0}{1 + \sum_{i=1}^{N} k \cdot e^{-(t-t_i)^2/a}}.$$

We chose $v_0 = 0.1$, $k = 4$, $a = 50$. Parameter a is chosen to define the proximity in which a lesion effects the camera. With the above parameter choices, the influence of a lesion is safely neglected if $|t - t_i| > 20$.

The viewing direction of the camera is controlled by another Kochanek-Bartels spline $S_d(t)$ whose control points (CP) are constructed as described by the following pseudo code (constants are specific to our implementation):

```
if $(t_1 > 20)$ then
        Add CP $[d(0),0]$ and CP $[d(t_1-20),t_1-20]$
else  Add CP $[p_1,0]$
for $i = 1$ to $N-1$ do
        Add CP $[p_i,t_i]$
        $\Delta_i = t_{i+1} - t_i$
        if $(\Delta_i > 40)$ then
           Add CP $[d(t_i+20),t_i+20]$
           Add CP $[d(t_{i+1}-20),t_{i+1}-20]$
        else
           Add CP $[p_i,t_i+\Delta_i \cdot 0.4]$
           Add CP $[p_{i+1},t_i+\Delta_i \cdot 0.6]$
        endif
endfor
if $(t_{\max} - t_N > 20)$ then
        Add CP $[p_N,t_N]$ and CP $[d(t_N+20,t_N+20]$
else  Add CP $[p_N,t_{\max}]$
```

In this program p_i is the position of lesion i, $d(t)$ is point in front of the camera along the centerline spline, obtained as

$$d(t) = S_p(t) + 10 \cdot S_p'(t),$$

i.e. moving in the direction of the derivative (tangent) of the spline. The choice of constant 20 is consistent with our choice for parameter a in function $v(t)$. Values 0.4 and 0.6 control the speed with which the camera switches targets. There is little point at looking between lesions, so these constants should be set to swing the camera between "best view" positions as fast as possible without introducing the sensation of skipping in the animation.

The pseudo code to guide the camera through the entire fly-through is as follows:

```
$t = 0$
while $(t < t_{\max})$
        position = $S_p(t)$
        if $|t - t_i| > 20$ for all $i = 1,\ldots,N$ then
              lookat = $d(t)$
        else lookat = $S_d(t)$
        Set Camera (position, lookat)
        Render Image
        $t$ += $v(t)$
endwhile
```

Note that it is not necessary to render an entire fly through. In our implementation the camera can be interactively moved along the centerline (with a user controlled factor in speed) while being subjected to these rules, and can be "uncoupled" for free interactive investigation any time.

5 Results

The algorithms have been successfully implemented in a PC based virtual colonoscopy tool. Results presented here were obtained by running the application on a 1.7GHz Intel P4-M system with 1GB of RAM. The clinical evaluation of the detection algorithm suggests sensitivity greater than 95% for polyps of size 3 – 10 mm, with a specificity of about 70%.

5.1 Detection

To illustrate performance of the detection process, its stages were timed working on two different datasets (figure 5): a biological phantom colon segment dataset of dimensions $256 \times 256 \times 256$ (sampled at $0.44 \times 0.44 \times 0.62$ mm), and a human full colon dataset of dimensions $512 \times 512 \times 360$ (sampled at $0.71 \times 0.71 \times 0.98$ mm). Both datasets are processed as arrays of Hounsfield values (signed integers with 12 bit precision). The timing results are shown by table 2. Note that filter window size (the bases of the discrete Gaussian kernels in directions x, y, and z) is a function of the sampling resolution of the dataset.

5.2 Camera Guidance

Without camera guidance, constant speed fly-throughs consist of 550 (Phantom) and 2200 (Human) frames. As the presence of polyps introduce slow-downs, which translates to an increase in the number of frames assuming constant frame-rate animations. In case of the Phantom dataset, the camera is always in the vicinity of more than one polyp 85% of its way. With the parameter values set as described in the previous section, a single polyp introduces about 500 additional frames in its proximity (902 instead of 400). Note that it is far more than needed for diagnostic purposes and results in extremely smooth camera motion. The three polyps stacked in the relatively short colon segment of the Phantom increase the frame count to around 1700 for the guided fly-through. Rendering speeds for the ray-casting engine range from around 10 fps for interactive rendering (256×256 resolution, using smart interpolation techniques [Koloszár and Jae-Young 2003]) to 1.5 fps for batch JPEG image export (512×512 resolution and no interpolation). Our results suggest that using the semi-automated navigation, in which the user controls a speed factor in the fly-through, examinations can be performed in ca. 5 minutes on average.

6 Conclusions and Future Work

This paper presented a robust, filter based method to identify polyps in a CT colon data, and a computer guided virtual camera control algorithm to present the results of detection to the user. The algorithms have been successfully implemented in a PC based virtual colonoscopy tool. The filtering technique results in very efficient extraction of feature values. The feature vector is only four dimensional, and candidates are classified using simple two dimensional thresholding. The algorithm is simple and is supported by solid mathematics. Using the proposed algorithm, the examination time can be reduced to less than half of unguided/unaided interactive VC, and smart fly-throughs showing all suspicious regions can also be generated. The camera control algorithm presented can be used in other applications to automate camera direction control in fly throughs, such as architectural and design demonstrations. Future work will focus on improving detection by tuning parameters as well using multiple filters scaled for various lesion sizes to improve specificity (reduce the number of false positives). Also, if deemed necessary, a more complex classifier will be implemented to handle feature vectors of increased size.

Acknowledgements

This work has been supported by OTKA (T042735) and by the National Office for Research and Technology (Hungary).

References

BITTER, I., SATO, M., BENDER, M., MCDONNELL, K., KAUFMAN, A., AND WAN, M. 2000. CEASAR: A smooth, accurate and robust centerline extraction algorithm. In *IEEE Visualization 2000, Conference Proceedings*.

BURLING, D. 2004. CT colonography reporting times: Effect of reader experience, fatigue, and scan findings in a multi-centre setting. In *RSNA*.

CRISTIANINI, N., AND SHAWE-TAYLOR, J. 2000. *An Introduction to Support Vector Machines*. Cambridge University Press.

CSÉBFALVI, B. 2001. *Interactive Volume-Rendering Techniques for Medical Data Visualization*. PhD thesis, Technische Universität Wien, Institut fr Computergraphik und Algorithmen. http://www.cg.tuwien.ac.at/research/theses/.

DREBIN, A., CARPENTER, L., AND HANRAHAN, P. 1988. Volume rendering. *Computer Graphics 22*, 4.

FUENTE, R., AND VALVERDE, F. 2004. Colonic polyps: dysplasia predictive factors. In *Revista Clinica Espanola*.

GOKTURK, S., TOMASI, C., AND ACAR, B. 2001. A statistical 3D pattern processing method for computer-aided detection of polyps in CT colonography. *IEEE Trans. Med. Imaging 20*, 1251–1260.

HLADUVKA, J. 2002. *Derivatives and Eigensystems for Volume-Data Analysis and Visualization*. PhD thesis, Institute of Computer Graphics, Vienna University of Technology, Vienna, Austria.

HONG, L., MURAKI, S., KAUFMAN, A., BARTZ, D., AND HE, T. 1997. Virtual voyage: Interactive navigation in the human colon. In *Proc. ACM SIGGRAPH '97*, 27–34.

KOCHANEK, D. 1984. Interpolating splines with local tension, continuity and bias control. In *Computer Graphics (SIGGRAPH '84 Proceedings)*, 33–41.

KOLOSZÁR, J., AND JAE-YOUNG, Y. 2003. Accelerating virtual endoscopy. *Journal of WSCG 11*, 2.

LEVOY, M. 1990. Efficient ray tracing of volume data. *ATG 9*, 3, 245–261.

MARSCHNER, S., AND LOBB, R. 1994. An evaluation of reconstruction filters for volume rendering. In *Proceedings of IEEE Visualization*, 100–107.

MITCHELL, D., AND NETRAVALI, A. 1988. Reconstruction filters in computer graphics. In *Proceedings of SIGGRAPH*, 221–228.

MÖLLER, T., MUELLER, K., KURZION, Y., MACHIRAJU, R., AND YAGEL, R. 1998. Design of accurate and smooth filters for function and derivative reconstruction. In *Proceedings of IEEE Symposium on Volume Visualization*, 143–151.

PALÁGYI, K., SORANTIN, E., BALOGH, E., KUBA, A., HALMAI, C., ERDŐHELYI, B., AND HAUSEGGER, K. 2001. A sequential 3D thinning algorithm and its medical applications. In *Proc. 17th Int. Conf. Information Processing in Medical Imaging*.

SATO, Y., WESTIN, C., BHALERAO, A., NAKAJIMA, S., SHIRAGA, N., TAMURA, S., AND KIKINIS, R. 2000. Tissue classification based on 3D local intensity structures for volume rendering. *IEEE Transactions on Visualization and Computer Graphics 6*, 2, 160–180.

WEGENKITTL, R., VILANOVA, A., HEGEDUS, B., WAGNER, D., FREUND, M., AND GROELLER, E. 2000. Interactive virtual bronchioscopy on a low-end PC. In *IEEE Visualization 2000*, 461–464.

	Segmentation	Centerline extraction	Region definition	Filtering (number of sample points)	Total	Filter window size
Phantom	2 secs	3 secs	18 secs	30 secs (1696)	53 secs	$78 \times 78 \times 56$
Human	5 secs	8 secs	29 secs	57 secs (14114)	1 min 39 secs	$50 \times 50 \times 35$

Table 2: Processing times

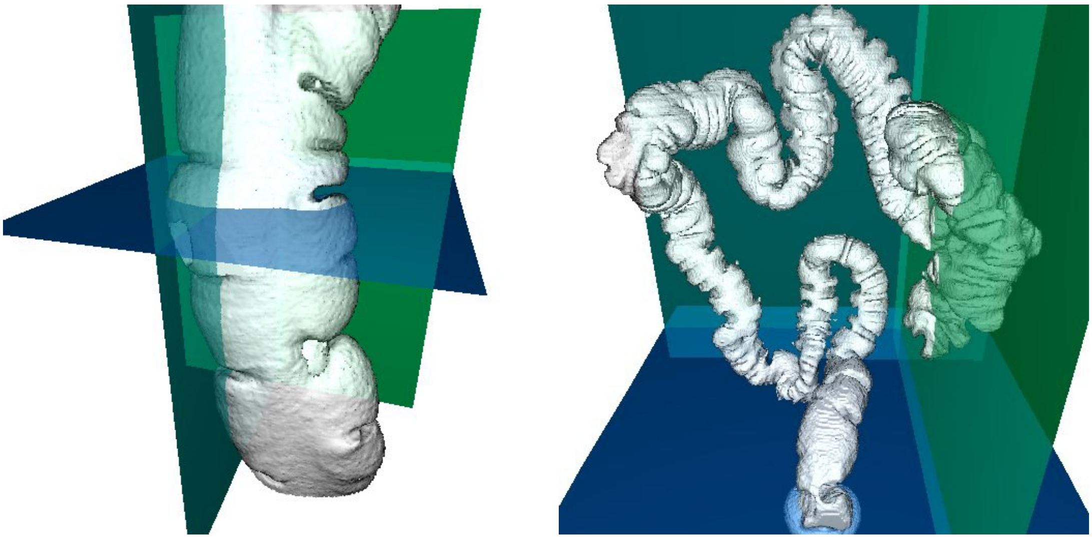

Figure 5: Phantom and human datasets

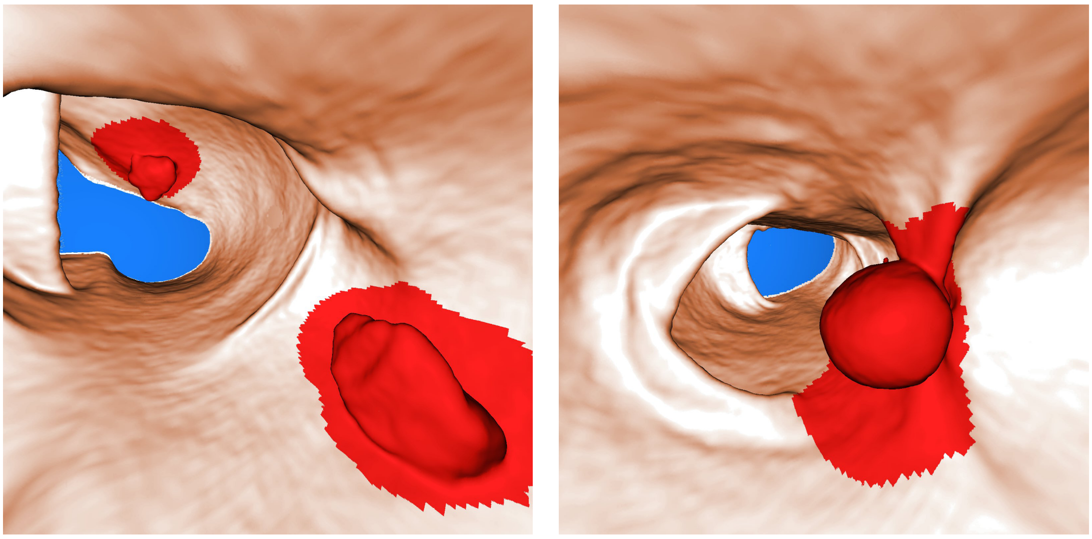

Figure 6: Detected Polyps in the Phantom dataset