

Interactive Virtual Colonoscopy

Dávid Jocha, József Koloszár
jd202@hszk.bme.hu, kj212@hszk.bme.hu

Computer Graphics Research Group
Department of Control Engineering and Information Technology
Budapest University of Technology and Economics
Budapest / Hungary

Abstract

Virtual colonoscopy is a non-invasive computerized medical examination method for examining the interior of the human colon, aiding the detection of polyps. In our study we have examined the most popular approaches to the problems of high-speed rendering and navigation within the colon, and devised our own algorithms. Our intent was to make them fast and reliable enough to be implementable on a consumer-end PC system, instead of state of the art high-end workstations, and to minimize off-line calculations. In this paper we present our idea of a real-time direct volume rendering based visualization technique combined with an on-line computer aided interactive navigation method for the examination of volume data constructed from CT scans. Parts of this algorithm have been implemented in our colon visualization program called ColVis.

Keywords: 3D Medical Imaging, Virtual Colonoscopy, Endoscopy, Camera Control, Volume Rendering, Interactive Navigation, Cancer Diagnosis.

1. Introduction

According to US figures, colon and rectum cancer (colorectal carcinoma) is the second leading cause of cancer deaths. Polyps and tumors are often detected too late. In order to efficiently diagnose this form of cancer in its early developing stage, a cost-effective patient-comfortable procedure is needed. As will be described, current methods are neither of the above. Computerized visualization, however, may provide a viable alternative: Virtual Colonoscopy.

1.1 Non-computerized methods

Currently, there are two methods for detecting polyps or carcinoma of the colon [2]: colonoscopy, and barium enema. Colonoscopy is an invasive procedure where optics are introduced into the patients rectum carrying a small risk of perforation. Intravenous sedation is required, and the entire procedure may last up to 40-60 minutes. Only part of the colon can be explored in this manner, and often the procedure must be terminated because of patient discomfort. Costs range in \$1000-\$2000.

Barium enema is less expensive. It involves filling the colon with contrast material, and positioning of the patient in front of an x-ray screen. This can be very cumbersome, and a great deal depends on the experience of the doctor interpreting the results. Much debate goes on regarding the efficacy of barium enema. While some claim it can be as sensitive as colonoscopy, others show that sensitivity can be as low as 78% in detection of polyps of size 0.5-2cm [15].

1.2 Virtual Colonoscopy

There are already a number of applications in medical imaging today, which aid diagnostics by rendering images of organs in 3D. These are based on data obtained from Computer Tomography (CT) or Magnetic Resonance Imaging (MRI) [22]. Virtual Colonoscopy is such an application and is meant to deliver images of the interior surface of the colon. It is still a topic of ongoing research.

Under accepted protocols [2], the patient is instructed to consume a contrast material (barium) for the two days prior to the investigation. Before the scan is performed, the colon of the patient is flushed and filled with air (the latter being the most discomforting part of the process). The scan itself takes 30-45 seconds to perform.

The scanner obtains a set of 100-200 2D cross-sectional images (*Figure 1,Left*) of the subject's abdomen, which are then reconstructed into a volumetric data set (a 3D array of voxels), serving as the patient's virtual model for the diagnosis. The obtained dataset may be subjected to various image enhancing algorithms (contrast, normalization), and often segmentation (the extraction of application specific parts from the dataset). Finally the dataset is visualized in a manner that can be easily interpreted by medical personnel, in this case a view (*Figure 1,Right*) similar to the one provided by the optics inserted during non-computerized colonoscopy.

Though research is still in an early phase (to our best knowledge there has not yet been a diagnosis specifically based on this method), we decided to implement algorithms that we determined were sufficiently fast and efficient in our own ColVis software. We found that specific aspects of virtual colonoscopy allowed for simplification of some volume based rendering techniques. We made an effort to see how virtual colonoscopy could be implemented in an operator-friendly manner. By aiming to obtain reasonable response times from a lower-end personal computer, we tried to show that in the foreseeable future hardware would not be the factor limiting the widespread practical application of virtual colonoscopy.

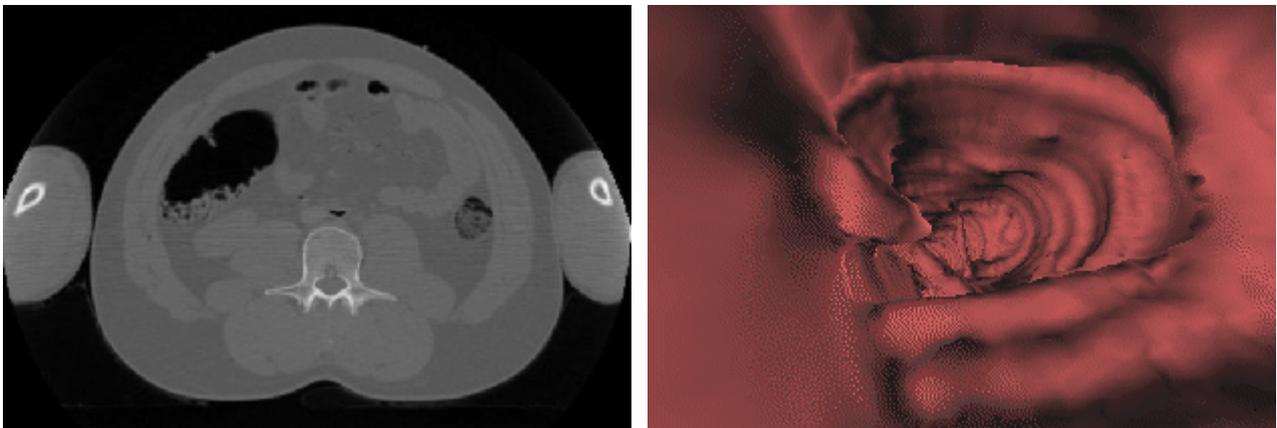


Figure 1: Left: Cross-sectional image obtained from a CT scan.
Right: Virtual view of the human colon.

2. (Near-)Real-time Direct Volume Rendering

Virtual colonoscopy uses algorithms of volume rendering. In the dataset built from CT scans, voxel values represent density measured at points of the body. Bones for example can easily be visualized by only dealing with voxels that have values in a specific range. The following aspects are crucial to optimizing algorithms for virtual colonoscopy. It is obvious that colon interior should be treated as

empty (lowest density), and seeing through the colon wall is not of importance, as it is the inner surface of the colon wall we wish to visualize in detail. As for illuminating the volume, we wish to simulate the “headlight” effect of classic optics employed in colonoscopy, meaning that we only deal with a single source of light with the same location (and orientation) as the camera.

Methods of volume visualization usually fall into two categories: direct or indirect [1].

Indirect methods transform the 3D volume model into something that is “easier” to render. It is typical to obtain surfaces from the volume and representing them as polygon grids, which can be efficiently rendered using hardware acceleration. Marching cubes is a well-known algorithm for surface extraction [11]. While at first this approach may seem perfectly viable for the purposes of virtual colonoscopy, it has a number of drawbacks: a great deal of off-line pre-calculation is required. The number of triangles may raise memory issues. The number of triangles generated cannot be handled in real-time, thus algorithmic data structures must be used to aid visibility checking. Image quality becomes an issue, especially when the camera is close to the colon wall. Finally, in order to generate images consistent with physical reality (meaning that polygons are only rendered to where polygons should be rendered), very careful segmentation is needed [5].

Direct methods work on the voxel array itself to generate images. There are two approaches: volume projection, and volume ray tracing [1]. The former basically projects slices parallel to the viewing window onto the screen. The latter casts rays for key pixels of the screen into the volume, evaluating luminosity and transparency along the way. We found volume ray tracing to be the most viable of the choices above.

2.1 Basics of Colon Visualization

There are two stages of visualizing the colon: off-line pre-calculations, and on-line interactive processing of data.

Off-line preprocessing involves procedures such as algorithmic segmentation, calculating potential fields (to be discussed later), etc. It is possible to process all of the information off-line if interaction is not a requirement. For example an entire fly-through of the colon may generated by computer, yielding video-type output, which can be viewed and reviewed at different speeds. Calculations for such a task, however, may take days to complete on high-end workstations.

All calculations with variables depending on interaction with the operator are performed in the on-line stage. User input typically repositions the camera, and the scene has to be re-rendered. Real-time response is one of the key aspects to keep in mind when evaluating algorithms of on-line processing. Frame rate should be as high as possible in order to immerse the operator and shift his focus to the medical evaluation of visuals.

We felt that off-line calculations should be minimized, in order to cut time and technological resources required by the examination as a whole. Instead, real-time interaction should be emphasized.

2.2 Volume Ray-tracing

The algorithm implemented in ColVis is based on the following variant of volume ray tracing.

Steps in the iteration along the ray are the following [1]. The ray is cast from the pixel ($trace=0$), assuming that we are processed distance s ($trace=s$) along the ray $L(s)$ being the accumulated radiance and $A(s)$ the opacity used to multiply radiance from beyond s . Incrementing s by Δs along the ray, the following formulae apply:

$$\begin{aligned}L(s+\Delta s) &= L(s) + (1-A(s)) * C(s), \\ 1-A(s+\Delta s) &= (1-a(s))(1-A(s)).\end{aligned}$$

$C(s)$ is the color value and $a(s)$ are the color and the opacity of the encountered voxel, respectively. Iteration is carried out until $1-A(s)$ drops below the sensitivity of the camera. At that point, radiance coming from beyond s can no longer significantly influence resulting pixel color. In order to maximize performance, the following conditions should be met:

- The inside of the colon is “empty”, meaning $C=0$ and $a=0$ for all voxels inside the colon. (Proper contrasting is needed.)
- The ray should not traverse deep beyond the colon wall, as it is the inner surface of the colon that should be rendered. (Contrasting and proper tuning of camera sensitivity is needed.)

The above conditions simplify calculations to be made at points sampled along the ray. The camera is always positioned inside the colon. Theoretically we may skip along the ray until we hit the colon wall, as there is nothing to calculate for points inside the colon. We then determine the colon wall normal at the point of intersection. The normal is needed by the shading method to define the color of the pixel.

The shading method of choice for ColVis is a simplified version of Phong shading. As illumination is based on the simple headlight setup, color can be calculated as a linear function of the scalar product of the normalized gradient and direction of the ray. To estimate the gradient (used as the surface normal), we implemented a relatively new method using 4D linear regression [6]. The concept being that the function

$$f(x,y,z) = Ax + By + Cz + D$$

is used to estimate density in the close proximity of the voxel, where x, y, z are distances from the voxel along the corresponding axis. Minimizing the error-squared based on values of the 26 closest neighboring voxels A, B, C , and D can be calculated very efficiently. The (A, B, C) vector is a good estimate of the gradient, and D can be used as the filtered density value for the pixel. For homogenous regions the gradient is always zero. For the inside of the colon the gradient is not relevant. For non-empty homogenous regions, the gradient can be assumed to point at the camera.

We chose to estimate the gradient for the eight closest voxels to the sampling point, and perform tri-linear interpolation to determine the filtered density and the gradient for the sampling point (*Figure 2, Left*). Other interpolation methods suggested by the authors of the 4D linear regression method might also be used at the appropriate cost. In our system tri-linear interpolation is used in high detail mode, while in low detail mode we simply assume the sampling point to be identical in features to the closest voxel. Low detail mode can be used efficiently to skim the colon, while high detail mode can be activated at scenes of interest. A speed relative adaptive detail tuning method is in development for a future version of ColVis.

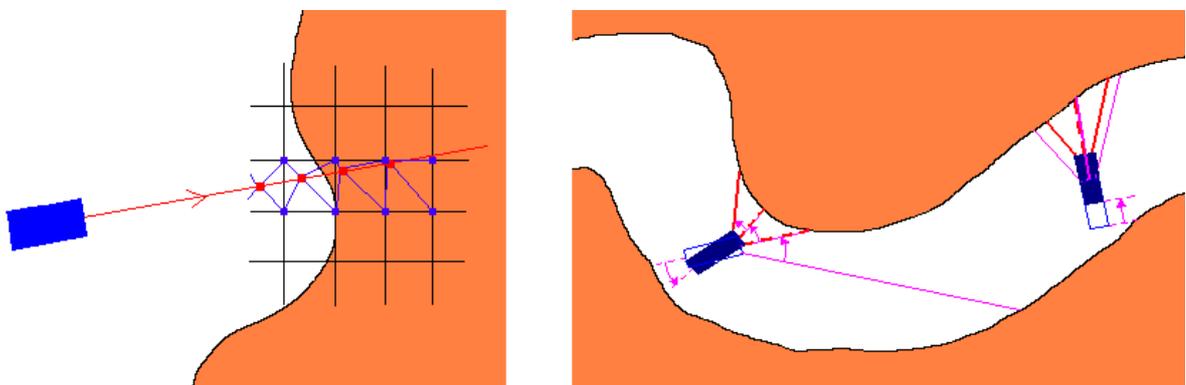


Figure 2: Left: Sampling points (red dots on the ray), and corresponding voxels (blue dots on the grid) where the gradient is evaluated.

Right: How rays in the new frame relate to those in the old in case the camera is rotated (left) or moved forward (right).

There are a number of ways to accelerate the algorithm described above. The first realization is that skipping over the interior of the colon (the first section of the ray) is desirable. A potential-field assisted method [5] has already been suggested for this. In the potential field, for every pixel inside the colon the distance to the colon wall is pre-calculated and stored along with the voxel. When the voxel is encountered during the trace, the distance stored in the potential-field can safely be skipped ahead. (Note that the potential field can also be used to aid camera navigation.) Though this method is fast, since we aimed to minimize pre-calculations (and we had to keep to memory limits), we chose to go with another, slower method. For this a very simple pre-calculation calculation was implemented, and one bit of the eight bits used to store the value of the voxel was sacrificed to function as a homogenous region flag. Due to contrasting, and errors of integer rounding and truncations, this loss of information is of no significance to resulting image quality. It must also be noted that this pre-calculation is by two orders faster than that needed to set up a potential field.

Of course with proper resources (time and memory) the potential field as well as gradients and filtered density values for voxels may be calculated off-line in advance, resulting in significant performance increase. Assuming 8 bits/voxel for storing value (50-100MB for a typical array of voxels) we find the following. For all voxels inside the colon, this amount is at least doubled if the potential field is calculated. For voxels close to the colon wall this value is at least five-folded if (A,B,C) and D are calculated. The resulting memory requirement quickly exceeds the reasonable scope of current PC hardware.

A very fast check of the homogenous region flag quickly reveals that we are sampling in an empty region, and the ray may be incremented without performing further calculations. Once in the proximity of the colon wall an average of 4 sampling steps are needed to determine pixel color.

Because of tri-linear interpolation we calculate the gradient and the filtered voxel value for eight voxels per sampling. As suggested by *Figure 2,Left*, at least a double recalculation of most variables is performed during rendering of an entire frame. This significantly reduces overall performance. Caching gradients at frame level for voxels rendered may significantly reduce this effect. The issue will probably be addressed in a future version of ColVis.

Naturally, hardware specific accelerations may also be implemented. Utilizing SIMD capabilities provided on recent PC processors is a promising aspect. Even switching to fixed-point arithmetic wherever possible is an option. However, the aim of our project was not to optimize code at assembly level.

Assuming that the camera only moves or rotates little at a time, we find that information gained from rendering the previous frame can aid us in rendering the current frame. If the condition is met, frame-to-frame acceleration can easily be implemented. Between frames the camera

- is either rotated by a constant angle around one of its three major axis (roll, pitch, yaw),
- or it moves forward or backward a constant distance.

Though this seems to be a rough limitation at first as real physics based simulation of dynamics works all six degrees of freedom of motion, we suggest that in our application, where the goal is to carefully exploration a 3D environment, these limited controls are sufficient.

In case the camera moves forward, the assumption that distances along the rays to the colon wall (which can be skipped ahead as we trace the ray) in the new frame will roughly equal those in the old frame minus the distance the camera moved can be made (*Figure 2,Right*). Because of perspective projection (implemented with a field of view of 90°), this is trivial for rays at center of the viewing frame, but gradually fails to apply as we near the edges of the frame. With this in mind, a simple “shortening function” may be devised that returns the distance that could be safely skipped ahead depending on information from the previously rendered frame and pixel coordinates. Every Nth frame would be rendered from scratch (key frames), to make sure anomalies in colon topology would not cause significant accumulative errors in the above algorithm.

The case of rotation by a constant angle is even easier to handle. The jump-skip distance for a ray in the new frame can be derived from those obtained in the previous frame for its neighbors corresponding to the angle and axis of rotation (*Figure 2,Right*). It seems possible to extend the algorithm described above to handle 6-DOF interactive motion control. Note that this type of frame-to-frame acceleration is a viable and very fast alternative to potential-field aided ray tracing described earlier in the section.

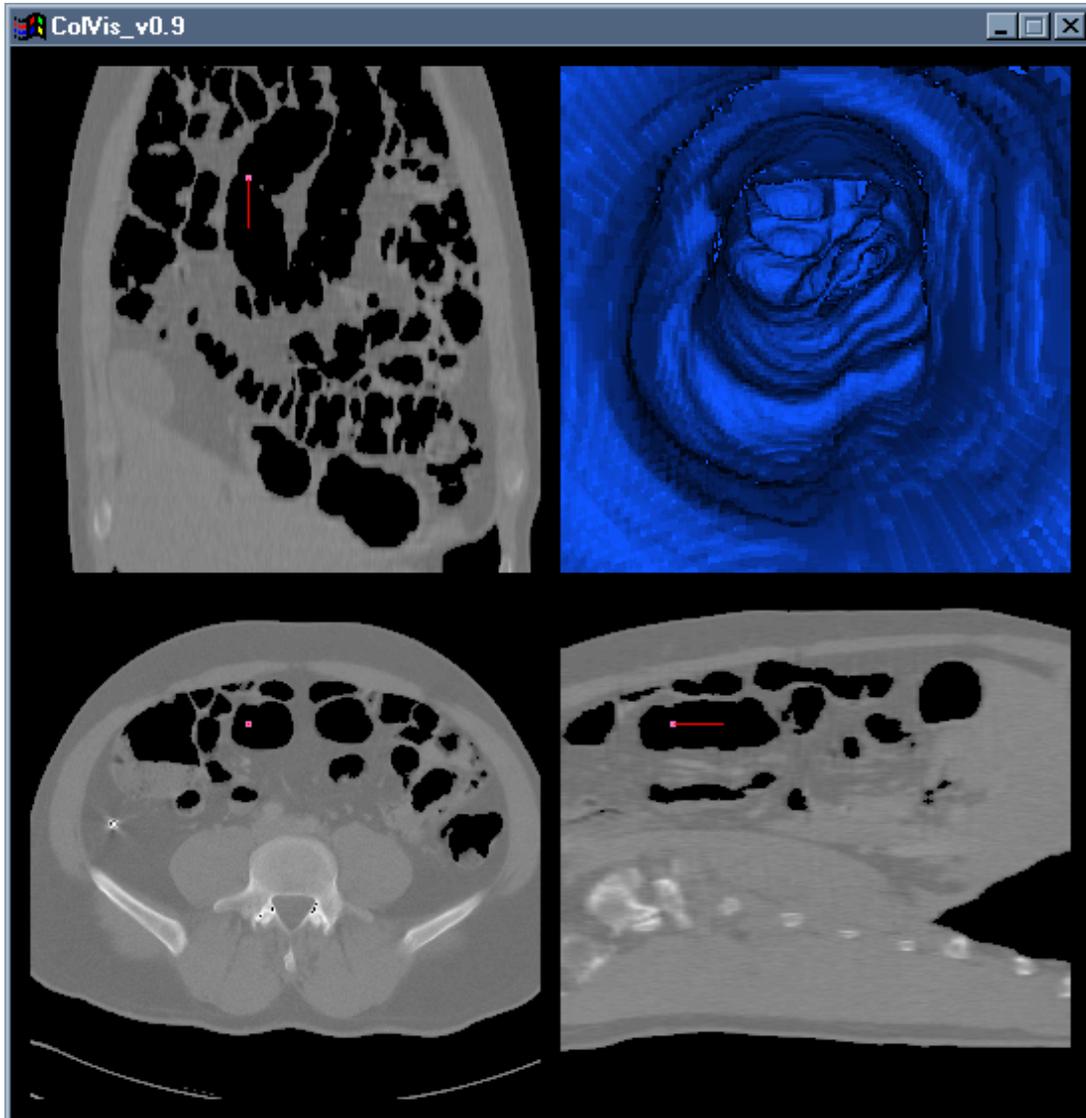


Figure 3: A screenshot of CoVis in action, with tri-linear interpolation turned off (low detail mode).

3. Navigating the Colon

Practical implementation of virtual colonoscopy also raises the issue of navigating the camera in the colon. Manually guiding the camera through the treacherous twists and turns of the virtual colon can prove quite challenging even for flight simulation enthusiasts, not to mention medical personnel. It is thus obvious that camera movement should be automated to some degree [4].

Depending on the actual method used, one or more of the general conditions to automating navigation described below should apply [3]:

- The camera should always be moving from the specified starting point in the direction of the specified endpoint, along a valid path (1).
- The operator should have the power to intervene and influence camera movement and orientation (2).
- The camera should keep as far away from the colonic surface as possible in order to keep the view of the scene optimal in detail (meaning the camera should keep to the centerline of the colon) (3).
- The camera should under no circumstances traverse the colonic surface (the equivalent of colon perforation in non-computerized colonoscopy) (4).

Three different stages of automation are reviewed below based on the points listed above:

- Manual control. The operator must guide the camera manually, like in a 3D computer game. This requires excess concentration from the operator, who might not even be skilled in such practices. Conditions 2 and 4 apply. This method is fully interactive.
- Automated flight planning. After the start- and endpoints are specified, the computer generates the flight path, moves the camera and producing a high-resolution fly-through animation without any kind of user interaction. The animation may later be reviewed and post-processed any number of times. Conditions 1, 3, 4 apply. This is a typical non-interactive, off-line method.
- Aided navigation. Camera motion is automated, but the operator may intervene at special points of interest to alter movement and other camera parameters to perform a thorough investigation for example. Condition 1-4 may all apply. This is the most desired method. It combines all the advantages of the previous two. To successfully implement it however the disadvantages, and drawbacks of both must be overcome. This basically means that very fast and efficient algorithms must be used to guarantee response times fast enough (high frame rates) to provide for comfortable user interaction.

3.1 Methods of Automated Navigation

There a number of methods for path detection, but only of few of them are applicable to virtual colonoscopy. The problem to be solved is that of having to get from point A to point B of an enclosed volume, without leaving the enclosed volume.

To ensure the natural feel of simulated motion the second derivative of the position-to-time function of moving bodies must exist. This is because movement of bodies can only be altered by forces acting on them (rigid bodies are assumed), and force is proportional to acceleration (Newton's Third Law of Dynamics), which is the second derivative mentioned above. Though this condition is already challenged as computer processing involves discretization, it is a good general guideline to use force (or acceleration) based motion control. Examples will be presented later.

The four most common approaches to path building are described below (most of them have migrated to virtual reality from robotics) [10]:

- Roadmap. A polygon based method with at most very limited applicability to virtual colonoscopy.
- Cell Decomposition. The volume to be navigated is subdivided into cells, and information is extracted from cell topology. Not really applicable to our approach of virtual colonoscopy.
- Peel Onion. This method was originally developed for 2D character recognition (OCR) purposes, but has been successfully extended to 3D. The idea is that the outermost layer of the volume enclosing the start- and endpoints are peeled away step by step without compromising volume integrity, resulting in a "thread" at centerline of the volume.

- Potential Field. The camera moves as a result of multiple forces acting on it. The colon wall acts as a source of repulsive forces, while the endpoint is source of an attractive force. This method is highly applicable to virtual colonoscopy, and is detailed in the next subsection.

3.1.1 Potential Field Aided Navigation

As the name suggests, the idea behind the model is to set up a potential field in which the camera should move according to expectations.

The colon wall exerts a repulsive force keeping the camera close to the centerline of the colon (*Figure 4,Left*). The endpoint attracts the camera to ensure the desired general direction of movement along the path forced by the colon wall (*Figure 4,Right*). The colonic wall blocks direct effect of the endpoint.

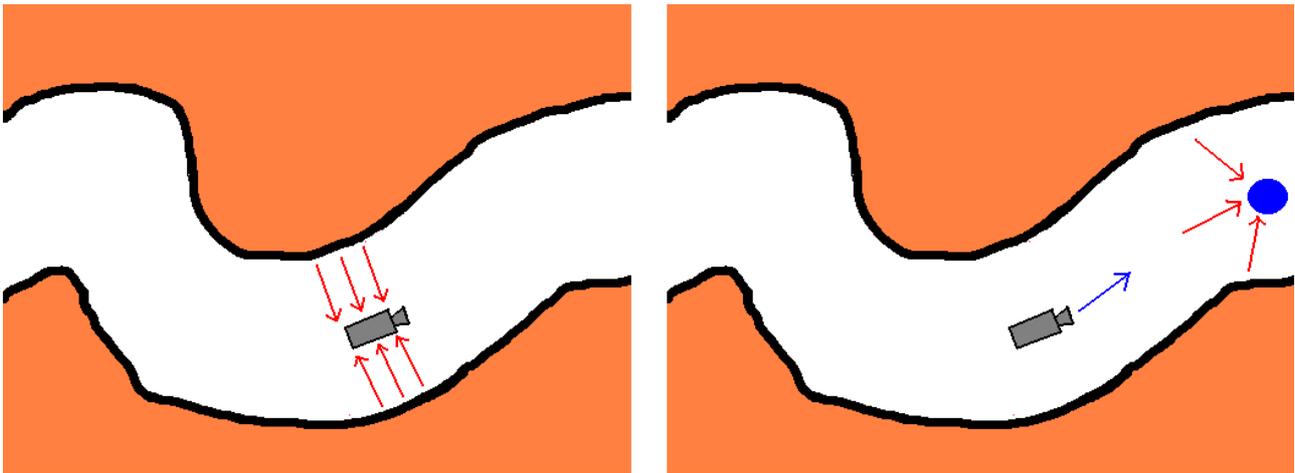


Figure 4: Left: Repulsive forces pushing the camera away from the colonic surface. Right: The endpoint attracting the camera.

The two forces described above can be derived at any point from the distance to colonic surface (D_s) and target (D_t) [14]. Both are usually pre-calculated at a resolution matching that of the voxel array. Note that the D_s field can also be used to accelerate rendering (see section 2.2). With these two variables the potential field at point x can be described with the formulae [3]

$$V(x) = C_t D_t(x) + C_s (p/D_s(x) - 1)^2 \quad \text{if } 0 < D_s < p$$

$$V(x) = C_t D_t(x) \quad \text{otherwise,}$$

where C_t and D_t are parameters influencing force magnitude, and p tunes the distance-of-influence of the repulsive force. The following equation of motion should be satisfied:

$$P'(t) = -\nabla V(X) - k_t P(t) + F_{user}(t),$$

where $P'(t)$ is the time derivative of the linear momentum at time t , $\nabla V(x)$ is $V(x)$'s gradient at point x , $k_t P(t)$ slows the camera in order to prevent it from going too fast, and $F_{user}(t)$ is derived from operator input.

One of the major problems to be dealt with in this model is that the camera may get caught in a local-minimum "trap". The endpoint is desired to be such a trap, but others may exist. Unintended local minimums can be avoided with proper tuning of parameters, methods for which have already been developed in robotics.

The camera can be modeled as a small cylinder with six degrees of freedom. Equations of motion for this model have already been devised and set up in detail [16].

3.2 Tentacles and Mines

To adopt the force field based method to our approach emphasizing on-line calculations as opposed to memory consuming off-line pre-calculations, we developed an algorithm, which we named “Tentacles and Mines”. The idea is colonoscopy specific, and may not as easily be applied to generic virtual endoscopy problems. The name comes from the illustration of the camera being a very military type of beetle crawling around, sensing the surroundings with tentacles and laying mines it does not want to step on.

To keep from the colonic wall we use the tentacles (*Figure 5,Left*), instead of pre-calculating D_s . A tentacle is essentially a ray just like the ones cast for rendering purposes, but only distance is evaluated along the way. Tentacles are directed in a spherically symmetric fashion. The theoretical minimum for tentacles is three, but naturally more should be used. Casting these sensory rays is negligible overhead to the thousands of rays traced for rendering the scene. More tentacles result in finer motion tuning. Theoretically, only extremely uncommon features of the colon and very big polyps can confuse the algorithm. Both cases should be detected and operator intervention should be called for. Information gathered from the tentacles is used to find the equivalent of $D_s(x)$.

Instead of checking distance to target, we lay mines (points acting as of sources of repulsive force), the last few of which are kept track of (*Figure 5,Right*). With properly tuned parameters these prevent the camera from turning back and keep it advancing from the starting point in the general direction of the other end of the colon (the endpoint).

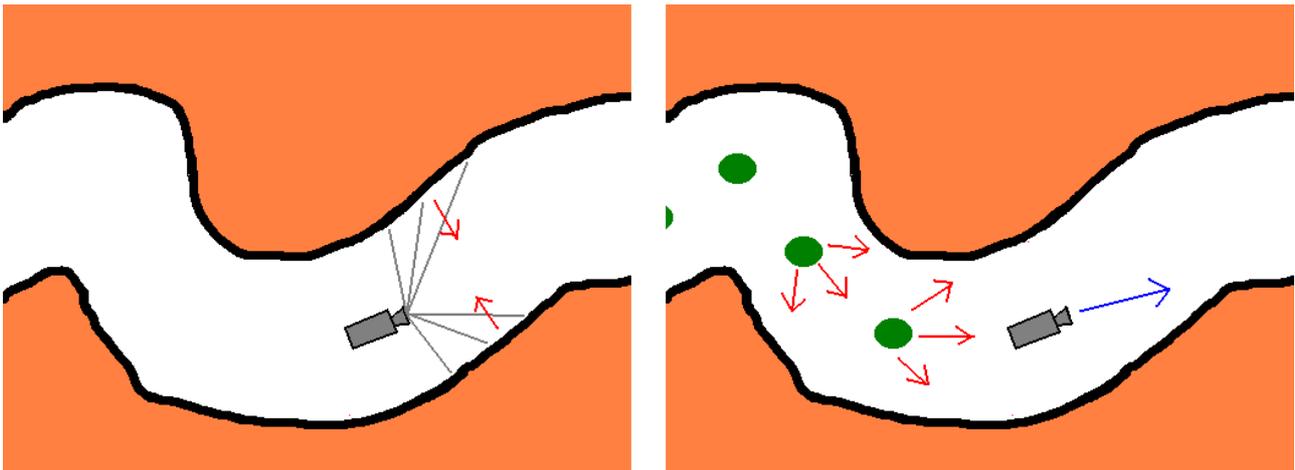


Figure 5: Left: Using tentacles to keep the camera close to the colon centerline.
Right: Mines forcing the camera to advance inside the colon.

Combining the effects of tentacles and mines, a system very similar to that of the force field method is obtained. We suggest that it can be used effectively to aid on-line interactive navigation within the colon. Also, most force field based algorithms can easily be adopted to work with this system. Effects similar to local-minimum traps as well as methods for tuning parameters are under investigation. Though at the time of writing not yet implemented in ColVis, sample programs for testing the algorithms in 2D have already been developed (*Figure 6*), and show very promising results.



Figure 6: A screenshot of testing tentacles and mines in 2D.

4. Conclusions

In this work our primary contributions were a direct volume based rendering algorithm with great potential for practical application, and an efficient on-line computer aided camera navigation method for virtual colonoscopy.

Results from the implementations we have done are promising, though requirements of real-time imaging are not yet met. Rendering time for a scene without any kind of acceleration and no code optimizations is around 2 seconds on 700MHz x86 class PC. Though this is significantly below the desired performance of 0.05 seconds (20 frames/second), we believe that by implementing acceleration methods described in our work and doing platform and processor specific optimizations a frame rate larger than 3 frames/second can be achieved, which could already be used in field-applications. Implementations on high-end graphics workstations would boost performance even higher. Based on our findings we suggest that visualization of virtual colonoscopy will soon be manageable on mainstream hardware.

There is still a lot of research to be done on various aspects of virtual colonoscopy, however. Based on the material we have researched, we found no reliable algorithm for the automatic segmentation (identification of the colon wall, and colon interior) of the volume dataset obtained from CT scans. Though direct volume rendering suggests no need for segmentation, while rendering scenes based on real patient data, we found artifacts (for example little “bridges” arcing across the colon) in the dataset, which we could not remove by contrasting voxels, and thus associated them with colonic residue, or scanning flaws. Artifacts like these can severely confuse path-building algorithms, and because of the lack of color information may also hinder polyp detection.

Camera control is another area that could benefit from improved techniques. Equipping the camera with some limited or extensive AI capabilities could help resolve issues of path planning as well as aid polyp detection.

The relative vulnerability of the colon to cancer mandates the continued exploration of virtual diagnostic techniques as both a supplement to and eventual replacement for traditional non-virtual diagnostic modalities.

5. References

- [1] Dr. Szirmay-Kalos L.: Számítógépes Grafika. Computerbooks, 1999.
- [2] L. Hong, A. Kaufman, Y. Wei, A. Viswambharan, M. Wax, Z. Liang: 3D Virtual Colonoscopy. IEEE Symposium on Biomedical Visualization, October 1995.
- [3] L. Hong, S. Muraki, A. Kaufman, D. Bartz, T. He: „Virtual Voyage: Interactive Navigation in the Human Colon”. SUNY.
- [4] R. Chiou, A. Kaufman, Z. Liang, L. Hong, M. Achniotou: Interactive Path Planning for Virtual Endoscopy
- [5] M. Wan, Q. Tang, A. Kaufman, Z. Liang, M. Wax: Volume Rendering Based Interactive Navigation within the Human Colon
- [6] L. Neumann, B. Csébfalvi, A. König, E. Gröller: Gradient Estimation in Volume Data using 4D Linear Regression
- [7] T. Galyean. Guided Navigation of Virtual Environments. ACM Symposium on Interactive 3D Graphics, ACM, April 1995.
- [8] M. Gleicher and A. Witkin. Through-the-Lens Camera Control. Computer Graphics (SIGGRAPH 92 Conference Proceedings), ACM SIGGRAPH, July 1992.
- [9] N. Greene, M. Kass, and G. Miller. Hierarchical Z-Buffer Visibility. SIGGRAPH 93 Conference Proceedings, Annual Conference Series. ACM SIGGRAPH, August 1993.
- [10] J. Latombe. Robot Motion Planning. Kluwer Academic Publishers, 1991.
- [11] W. Lorensen and H. Cline. Marching Cubes: A High Resolution 3D Surface Construction Algorithm. Computer Graphics (SIGGRAPH 87 Conference Proceedings), ACM SIGGRAPH, July 1987.
- [12] W. Lorensen, F. Jolesz, and R. Kikinis. The Exploration of Cross-Sectional Data with a Virtual Endoscope. In R. Satava and K. Morgan (eds.), Interactive Technology and New Medical Paradigm for Health Care, IOS Press, 1995.
- [13] G. Rubin, C. Beaulieu, V. Argiro, H. Ringl, A. Norbash, J. Feller, M. Dake, R. Jeffrey, and S. Napel. Perspective Volume Rendering of CT and MR Images: Applications for Endoscopic Imaging. Radiology, vol. 199, May 1996
- [14] T. Saito and J. Toriwaki. New Algorithms for Euclidean Distance Transformation of an N-Dimensional Digitized Picture with Applications. Pattern Recognition, vol. 27, no. 11, 1994
- [15] C. Morosi, G. Ballardini, and P. Pisani. Diagnostic Accuracy of the Double-Contrast Enema for Colonic Polyps in Patients with or without Diverticular Disease. Gastrointest Radiology, vol. 16, 1991.
- [16] K. Shoemake. Animation Rotation with Quaternion Curves. Computer Graphics (SIGGRAPH 85 Conference Proceedings), vol. 19, ACM SIGGRAPH, July 1985.
- [17] S. Teller and C. Sequin. Visibility Preprocessing For Interactive Walkthroughs. Computer Graphics (SIGGRAPH 91 Conference Proceedings), vol. 25, ACM SIGGRAPH, July 1991.
- [18] R. Turner, F. Balaguer, E. Gobbetti, and D. Thalmann. Physically-Based Interactive Camera Motion Control Using 3D Input Devices. Computer Graphics International '91. Springer-Verlag, June 1991.

- [19] D. Vining, D. Gelfand, R. Bechtold, E. Scharling, E. Grishaw, and R. Shifrin. Technical Feasibility of Colon Imaging with Helical CT and Virtual Reality. Annual Meeting of American Roentgen Ray Society, 1994.
- [20] C. Ware and S. Osborne. Exploration and Virtual Camera Control in Virtual Three Dimensional Environments. ACM Symposium on Interactive 3D Graphics. ACM, March 1990.
- [21] M. Wan, W. Li, K. Kreeger, I. Bitter, A. Kaufmann, Z. Liang, D. Chen, M. Wax: 3D Virtual Colonoscopy with Real-time Volume Rendering
- [22] Tarján Zs., Makó E., Szirmay-Kalos L., Csébfalvi B., Székely G.: Virtual CT Enteroscopy. 84th Scientific Assembly and Annual Meeting of the Radiological Societies of North America RSNA 98 Conference, Chicago 1998. Abstr: Radiology Vol 209, 1998
- [23] Tarján Zs., Makó E., Szirmay-Kalos L., Csébfalvi B., Székely G.: Early experiences with virtual CT enteroscopy. European Congress of Radiology, Vienna 1999. Abstr: European Radiology Vol 9. 1999