Virtual Colon Unfolding

Anna Vilanova Bartrolí[*] Rainer Wegenkittl[†] Andreas König [*†] Eduard Gröller[*]

[*] Institute of Computer Graphics and Algorithms
Vienna University of Technology
[†] Tiani Medgraph

For reviewing further results: http://www.cg.tuwien.ac.at/research/vis/vismed/ColonUnfolding/

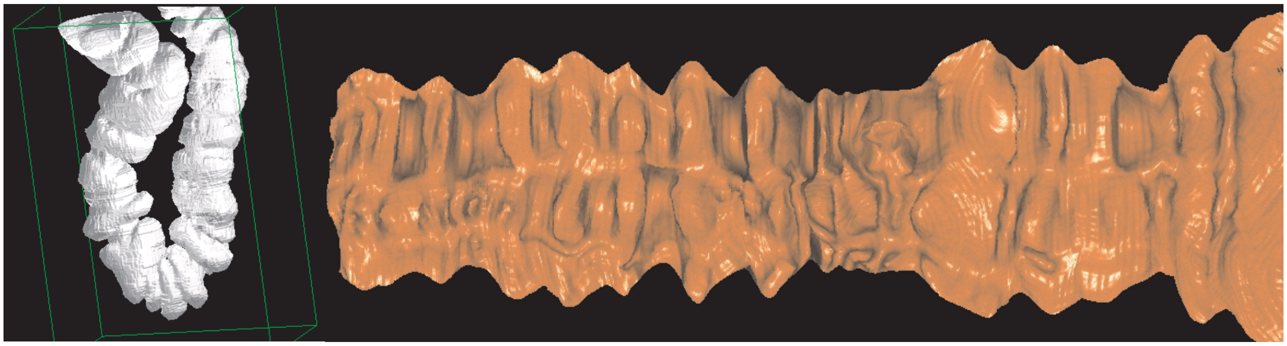


Figure 1: Outside view and virtual unfolding of a segment of a CT data set of a healthy colon with a resolution of 198x115x300.

Abstract

The majority of virtual endoscopy techniques tries to simulate a real endoscopy. A real endoscopy does not always give the optimal information due to the physical limitations it is subject to. In this paper, we deal with the unfolding of the surface of the colon as a possible visualization technique for diagnosis and polyp detection. A new two-step technique is presented which deals with the problems of double appearance of polyps and nonuniform sampling that other colon unfolding techniques suffer from. In the first step, a distance map from a central path induces nonlinear rays for unambiguous parameterization of the surface. The second step compensates for locally varying distortions of the unfolded surface. A technique similar to magnification fields in information visualization is hereby applied. The technique produces a single view of a complete virtually dissected colon.

Keywords: Volume Rendering, Virtual Endoscopy

1 Introduction

The inspection of anatomical cavities using medical imaging (e.g., CT and MRI) and computer visualization techniques is called virtual endoscopy. A real endoscopy is an invasive procedure which involves a certain degree of risk for the patient. In some diagnostic procedures, virtual endoscopy has the potential to be used in clinical routine to avoid the inconvenience of a real endoscopy.

Most of the developed techniques concentrate on simulating the results that would be obtained with a real endoscope [6, 11]. Although this can be useful in some cases, like in an intraoperative scenario, virtual endoscopy should not constrain itself to simulate the results of a real endoscope. In a virtual environment, without physical limitations, more suitable visualization techniques can be offered. In the case of diagnosis, the physicians are mainly interested in exploring the inner surface of the organ where polyps might be detected. Large polyps are more likely to develop into malignities than small ones. Usual endoscopic views visualize just a small part of the surface of the organ. Furthermore, it is difficult to detect polyps that are located behind folds. In this paper, we concentrate on virtual colonoscopy, which focuses on the examination of the colon.

Recently, new methods to visualize anatomical cavities (e.g., the colon) have been proposed. These methods are based on the fact that an efficient way to inspect the inner surface of the colon would be to open and unfold it, in order to examine its internal surface. In the real world this is not possible, but there is no patient damage involved if this dissection of the organ can be achieved virtually with the medical data obtained by CT or MRI (i.e., the virtual organ).

In our previous work [3], a method for unfolding the colon was presented. This method results in a video with correct unfolded portions of the colon in each frame. The method solves problems that previous techniques have, like double polyp appearance (i.e., the same polyp can show up more than once). On the other hand, it has the drawback that the physician has to review a video and cannot visualize the complete surface at once. In this paper, we describe a new method to obtain the whole colon unfolded in one view. The physician can easily detect areas where polyps appear, and furthermore get an idea of their shape and size. Afterwards, the physician can concentrate in doing an exhaustive inspection of the problematic areas. This method proposes solutions to several of the problems that appear in previous similar techniques. Although in this paper we concentrate on the colon, the technique could be applied to any tubular organ.

In the next section, we review related work and discuss the differences to our method. Section 3 gives an overview of the new method. Sections 4 and 5 present in detail the main steps of the

[*] email:{anna,koenig,meister}@cg.tuwien.ac.at, http://www.cg.tuwien.ac.at

[†] email:{rainer.wegenkittl}@tiani.com, http://www.tiani.com

method. Results are presented in section 6 and conclusions and future work are described in section 7.

2 Related Work

The straightforward techniques to unfold an organ [10, 14] start with defining a path which is placed as close to the center of the cavity as possible. After that, a sequence of frames is calculated. For each frame, a cross-section orthogonal to the path is computed. Then the central path is straightened and the cross-sections are piled up to form a stack. As a last step, the straightened colon is unfolded obtaining a volume model of the unfolded colon. The model is displayed afterwards using standard volume rendering techniques.

These methods allow to visualize the complete surface at once. One of the main problems of these techniques is present in high curvature areas of the central path, i.e., at path locations where the radius of curvature is bigger than the organ diameter. In such cases, orthogonal cross-sections intersect each other or are far apart in other regions (see figure 2). As a consequence, a polyp can appear twice in the flattened model or it can be missed completely. Wang et al. in later work [13, 12] try to overcome this problem. The authors use electrical field lines generated by a locally charged path to govern curved cross-sections instead of the planar sections. The cross-sections tend to diverge avoiding conflicts. If the complete path is charged then the curved planes will not intersect. However, for each point of the field lines the contribution of each charge in the path must be calculated. This operation is so computationally expensive that the authors propose to just locally charge the path. A small segment of the path contains the charges for each cross-section. In this way, the method is feasible in practice, but it cannot ensure that the curved cross-sections will not cross anymore.

Haker et al. [5] use conformal mapping, which is angle preserving, to project the polygonal colon surface, colored with its Gaussian curvature, to a plane. One of the main problems of this method is that a highly accurate segmentation is necessary to ensure good results for diagnosis. The whole surface is being flattened, so the information by which the physician has to recognize a polyp is the shape of the color-coded Gaussian curvature of the extracted surface.

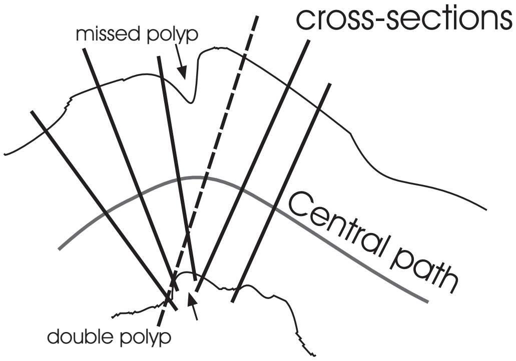


Figure 2: Illustration of the possible undersampling and double appearance of polyps due to intersections of the cross-sections in high curvature areas. The dashed cross-section line produces a double appearance of a polyp.

Paik et al. [9] propose various kinds of camera projections for virtual endoscopy. With a normal endoscopic view just 8% of the solid angle of the camera is visible in each frame. Paik et al. project the whole solid angle of the camera by techniques known from cartographic projections. They suggest the usage of the Mercator projection to map the solid angle to the final image. This technique samples the solid angle of the camera, after which the solid angle is mapped onto a cylinder which is mapped finally to the image.

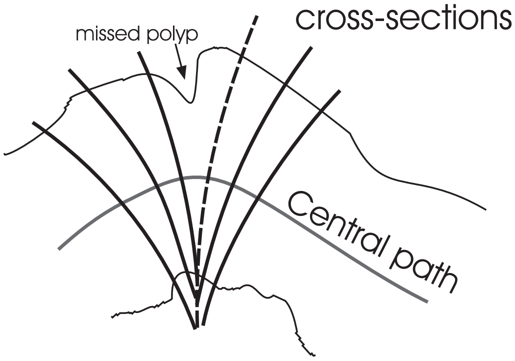


Figure 3: Illustration of the elimination of double polyp appearance by nonlinear ray casting.

In our previous work [3], a video is generated by projecting the organ surface to a cylinder which is moved along the path. This avoids the appearance of double polyps since intersections can just appear between different frames.

On the other hand, it has the drawback, as the method of Paik et al., that the physician does not have a complete view of the colon and has to inspect a complete video.

All of the unfolding methods introduce some kind of deformation since it is mathematically impossible to perform a mapping between two surfaces which preserves angles and area at the same time if the two surfaces do not have the same Gaussian curvature.

In this paper, a method is proposed which results in a complete view of the unfolded colon avoiding the problem of double appearance of polyps and the possibility of missing polyps. In our approach the unraveling does not just produce a surface but also a height field (distance of the colon surface to central path), which avoids that the polyps are flattened. In the next sections, this method will be discussed in details.

3 Method Overview

Our method for unfolding the colon can be divided into two main steps: nonlinear ray casting, which solves the problem of double appearance of polyps, and nonlinear 2D scaling, which improves the nonuniform sampling, and therefore avoids that polyps are missed.

The input data of the technique is a segmented volume data set of a colon. The segmentation is needed to find the central path. It is important that the segmentation is conservative, meaning that it does not contain any point outside of the colon. In our approach the central path is computed by thinning the segmented volume. The path is smoothed, but ensuring that it will not cross the colon surface. We use the algorithm presented in our previous work [2].

A distance map is generated from the calculated central path. A distance map is a volume data set whose voxels contain the distance to the nearest object voxel [8]. In our case, the object is the central path.

A coordinate frame is moved through the path. For each position along the path, rays are traced starting in the plane orthogonal to the path, and following radial directions (constant angle sampling). To avoid double polyp appearance in high curvature areas of the path, the rays follow the negative gradient direction of the precalculated distance map. The rays are not straight lines any more, but they do not intersect (see figure 3). Nonlinear ray casting has already been investigated before [4]. Section 4 explains how these rays are traced.

Along each curved ray the volume is sampled in a uniform way and direct volume rendering is performed. The ray is terminated when it hits the surface of the colon. The result of the nonlinear ray casting can be interpreted as a 2D parameterization of the inner colon surface projected to the central path. One parameter corresponds to the position along the path. The second parameter specifies the ray within the plane orthogonal to the current path position. The distances between ray origins (i.e., in the central path) and hit surface points (i.e., on the colon surface) determine a height field. Afterwards, the height field is unfolded. Actually, the result corresponds to a parallel projection of the unfolded height field.

Nonlinear ray casting samples the height field nonuniformly. A straightforward unfolding to a regular grid contains severe area distortions and is therefore not optimal. In the second step an iterative method transforms the previously generated 2D parameter grid to compensate for these distortions. Afterwards, the ratios between the area that the samples represent and their area in the 2D grid are approximately equal. The second step is based on nonlinear scaling that is used in a similar way for magnification fields in information visualization [7]. In section 5, the algorithm is described in detail. Next, the surface is resampled with a quasi-uniform sample rate using the 2D grid after the nonlinear 2D scaling.

The main steps of the outline presented in this section will be discussed in more detail in the next sections.

4 Nonlinear Ray Casting

The curve $C(v)$, which represents the central path of the colon, is calculated using thinning and the smoothing algorithm presented in our previous work [2]. In order to trace the nonlinear rays, a distance map $D(p)$ of a point p to the central path $C(v)$ is computed. $D(p)$ is calculated in discrete space. $D(p)$ is a volume data set that contains the minimum distance to the path for each voxel. A reconstruction filter is used to approximate the distance map $d(p)$ in continuous space.

For the calculation of $D(p)$, a minor modification of the Euclidean distance transformation presented by Lohmann [8] is used. The modified distance transformation saves for each voxel the vector $\vec{D}(p)$ from the voxel to the closest object point, $D(p) = ||\vec{D}(p)||$. Using this information, the distance map $d(p)$ in any point $p \in \mathbb{R}^3$ is defined using the following reconstruction function.

$$d(p) = \{ \; min(||p_i + \vec{D}(p_i) - p||) \mid \forall i \; 0 \leq i \leq 7, \; p_i \in \mathbb{N}^3 \; \}$$

where p_i represent the eight voxels of the cell that contains p. This reconstruction gives the exact value when one of the eight neighbors p_i has the same closest object point as p. Trilinear interpolation supposes a linear behavior of the distance map function which is just true in some special cases.

The nonlinear rays are traced from the central path uphill, i.e., along the negative gradient direction, $-\nabla d(p)$, of the distance map $d(p)$. $d(p)$ is continuous in the first derivative nearly everywhere, except for ridge and valley lines. The distance map induces a vectorfield which is defined by the gradient of the distance map (i.e., $\nabla d(p)$). Trajectories traced in this vectorfield are unambiguous. These trajectories correspond to our nonlinear rays. It is known that trajectories of such vectorfields will not intersect [1]. Furthermore, they will not produce cycles, since it is impossible to return to the same point if you always move uphill. In the worst case, the nonlinear rays will merge in ridge and valley lines, but they will not intersect. With these curved rays the appearance of double polyps is avoided and an unambiguous parameterization of the colon inner surface is obtained.

4.1 Casting of Nonlinear Rays

The nonlinear rays are traced by moving a coordinate frame along the curve $C(v)$. The frame is moved using a minimal rotation frame (see [3] for more details). For each position in the path, a constant number of rays is traced. The initial point of each ray is placed in the plane orthogonal to the path. Note that the gradient is not defined within the path since it is a valley line of the distance map $d(p)$. Therefore, the initial points are located in a circular arrangement at a small distance from the path position. Once the initial points have been determined (u parameterization), the rays are traced incrementally following the negative gradient of the distance map (see figure 4).

The rays have the tendency to be perpendicular to the path since it is the direction of maximal change of $d(p)$ in linear segments of the path. The rays become curved in areas where the curvature of the path increases (see figure 3 and 4).

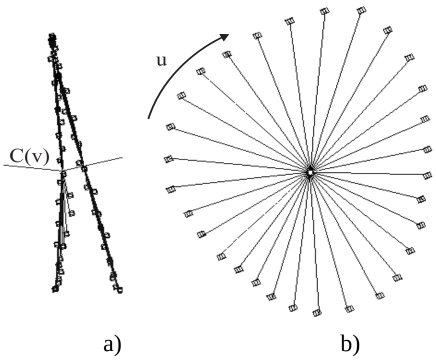


Figure 4: The nonlinear rays traced from a specific path position: a) the curved rays in areas of high curvature for two consecutive points along $C(v)$, b) nonlinear rays traced in u direction.

4.2 Colon Surface Parameterization

The previous section presented how to trace curved rays from the central path which will not intersect each other. While the rays are traced, direct volume rendering is performed. The ray terminates when the surface is hit. The result of the nonlinear ray casting is then a sampling of the inner surface of the organ.

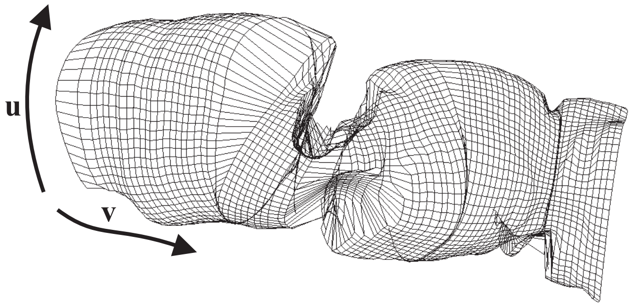


Figure 5: Surface obtained after nonlinear ray casting. It can be observed that the sampling of the surface is nonuniform.

The tracing of the nonlinear rays defines an unambiguous parameterization of the inner colon surface $s(u,v)$. Here, v is the parameter of the curve which describes the central path $C(v)$, and u is the radial angle along which the nonlinear rays are started ($u \in [0,2\pi]$).

Figure 5 shows $s(u,v)$ which results from applying nonlinear ray casting to a piece of the colon. The lines correspond to the isolines of the parametric surface. The parametric space is sampled uniformly in the u and v direction, but this does not correspond to a uniform sampling of $s(u,v)$.

A straightforward mapping of a parametric surface onto a plane can be easily achieved by mapping the parametric space. Figure 6b shows the mapping of the surface $s(u,v)$ shown in figure 5 to a 2D regular grid (in parametric space).

Nonlinear ray casting avoids that features appear more than once, but on the other hand the sampling of the surface is far from being uniform. There are oversampled areas, which lead to geometric deformations, and also undersampled areas. In the last case, not just deformations appear but features of the surface can be missed.

Figure 6a is the result of unfolding to a regular grid using straight rays (see figure 2). The solid circles indicate areas where double features appear. It can be observed that in the same areas using nonlinear ray casting, the double polyps disappear, and instead an enlargement of the feature appears. The areas encircled with dashed circles indicate undersampled areas and therefore areas where features are possibly missed. Note that the same undersampled areas are present in both figures.

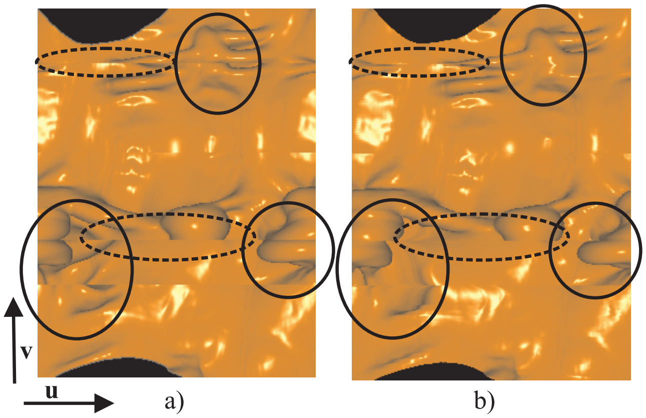


Figure 6: a) Unfolding of the colon surface of the data set presented in figure 5 using straight rays. Solid circles indicates double polyp appearance areas. Dashed circles indicate undersampled areas. b) unfolding of the parametric surface shown in figure 5 by mapping the parametric space to a regular grid.

In the next section, an algorithm is presented to obtain an unfolding of the parametric surface $s(u,v)$ avoiding geometric deformations and missing features.

5 Nonlinear 2D Scaling

5.1 Height field unfolding

In the previous section, an unambiguous parameterization of the inner surface projected to the central path has been introduced. The distance between the surface point $s(u,v)$ and the corresponding path position $C(v)$ defines a height field $r(u,v)$.

Figure 7a is an illustration of a cross-section of the height field where v is constant. The unfolding to a 2D regular grid of $r(u,v)$ in figure 7a corresponds with figure 7b. The parallel projection of the unfolded height field corresponds to the mapping of the surface $s(u,v)$ presented in section 4.2. It can be observed that the object is locally scaled, and geometrically deformed, depending on the value of the height field $r(u,v)$.

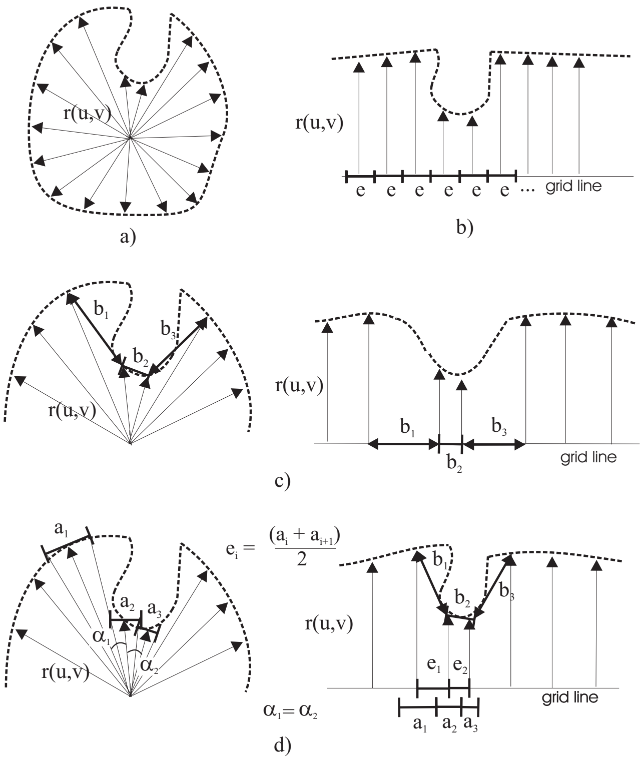


Figure 7: Illustration of height field unfolding: a) cross-section of $r(u,v)$ for a fixed value of v, b) unfolding to a regular grid, c) unfolding preserving the edge lengths of the 3D quadrilateral mesh in the 2D grid d) unfolding of the height field preserving the edge lengths of the 3D quadrilateral mesh.

The sampling of the surface $s(u,v)$ defines a quadrilateral mesh on the surface of the colon (see figure 5). The quadrilaterals vary considerably in size. From the 3D quadrilateral mesh, we know the distances between adjacent points (i.e., the length of the edges of the quadrilateral). If these distances are preserved in the 2D grid, the sizes of the quadrilaterals will be approximately preserved.

Figure 7c shows the result of preserving the 3D edges of the quadrilateral mesh that corresponds to figure 7a. Geometric deformations still appear. The geometric deformations depend on the values of the height field $r(u,v)$.

Using the previous method, we do not preserve the edges of the 3D quadrilateral mesh in the unfolded height field, but in its parallel projection to the 2D grid. This is not what is desirable. The edges of the unfolded height field have to have the same length as the edges of the 3D quadrilateral mesh. This implies that the distance between edges in the 2D grid has to correspond with the projection of the edges of the 3D quadrilateral edges to the central path.

To calculate the projected edge values we use different approximations for the u and v direction. For the edges in the u direction (see figure 7d), each sample represents a length in the u direction approximated by

$$a(p_{i,j}) = 2 * \tan\left(\frac{\alpha}{2}\right) * r(p_{i,j})$$

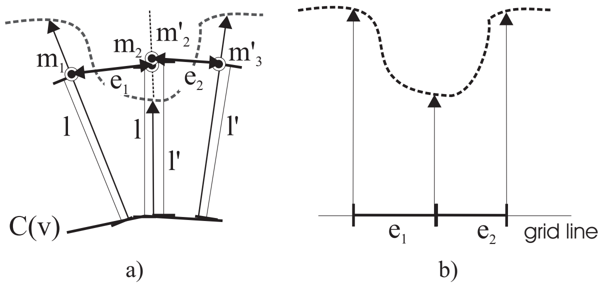
where $p_{i,j}$ corresponds to (u_i, v_j), u_i is the ith sampled parameter value in the u direction and v_j the jth in the v direction. α is the angle between the straight rays from the path to $p_{i,j}$ and $p_{i+1,j}$ (see figure 7d). For simplicity we illustrate in figure 7d the case where the angle α is the same between consecutive rays. α is not constant due to the curvature of the nonlinear rays. The projected edge distance between two consecutive samples in the u direction is approximated by

$$e(p_{i,j}, p_{i+1,j}) = \frac{a(p_i) + a(p_{i+1,j})}{2} \tag{1}$$

This equation cannot be used for the v direction since angle α is not defined. Therefore, in the v direction the following expression is used

$$\begin{aligned} l &= \frac{r(p_{i,j}) + r(p_{i,j+1})}{2} \\ \vec{m}_j &= C(v_j) + \frac{s(u_i, v_j) - C(v_j)}{r(p_{i,j})} * l \\ e(p_{i,j}, p_{i,j+1}) &= \|\vec{m}_j - \vec{m}_{j+1}\| \end{aligned} \tag{2}$$

Equation 2 gives an approximation of the projected distance between samples (see figure 8).



a) b)

Figure 8: Illustration of height field unfolding in the v direction.

Using equations 1 and 2, we have defined the length of the edges that the 2D grid has to have in order that the 3D edges of the quadrilateral mesh are preserved in the unfolded height field. In the next section, an algorithm to construct such a 2D grid is presented.

5.2 Nonlinear 2D scaling

The objective of the nonlinear 2D scaling algorithm is to generate a 2D grid whose edges preserve the size of the corresponding projected edges calculated using $e(p_{i,j}, p_{k,l})$. An analytical solution to the problem is too complex, so a numerical solution is adopted. To generate such a grid we will use an approach similar to the one presented by Keahey et al [7]. The main difference is that the new algorithm achieves not just area preservation, but also the edge lengths are preserved.

The idea is that we want to find a transformation $T(i,j) : \mathbb{N}^2 \rightarrow \mathbb{R}^2$ of a 2D regular grid such that for all values of i and j the equation $e(p_{i,j}, p_{k,l}) = \|T(i,j) - T(k,l)\|$ holds, where $p_{k,l}$ is a 4-connected neighbor of $p_{i,j}$.

$T(i,j)$ has to be C^0-continuous and it should preserve the order (no edge or grid node flipping). The condition to preserve the order is defined as

$$\begin{aligned} (x,y) = T(i,j) \quad & (x',y') = T(k,l) \\ i < k \Longleftrightarrow x \leq x' \quad & j < l \Longleftrightarrow y \leq y' \end{aligned} \tag{3}$$

We define a 2D scaling field S as a field of scalar values. Each edge scalar value indicates the scaling factor that a transformation has applied to the edge. The scaling field for an edge defined between $T(i,j)$ and $T(k,l)$ is $S(i,j,k,l) = \|T(i,j) - T(k,l)\|$. A 2D scaling field S is defined for any transformation T.

We define $S_g(i,j,k,l) = e(p_{i,j}, p_{k,l})$. Using the 2D scaling field S_g we want to construct its corresponding C^0-continuous and order preserving transformation T_g. T_g will correspond to the desired nonlinear scaled 2D grid. The major problem is to find the right coordinates (x,y) of the transformation T_g given a scalar value of the 2D scaling field S_g. It is clear that for the same 2D scaling field several transformations are possible.

To approach this problem, we have used an iterative method which will provide a numerical solution. The goal of the algorithm is to find a transformation T_c that provides a good approximation of T_g.

Given a transformation T_c, the corresponding scaling field can easily be calculated by $S_c(i,j,k,l) = \|T_c(i,j) - T_c(k,l)\|$. A scaling field error can then be computed by $S_e = S_g - S_c$. S_e gives the difference between the computed scaling field S_c and the desired scaling field S_g.

The iterative algorithm starts with a regular grid T_c. Then S_c and S_e are calculated. The algorithm iterates over each node of the grid. For each node, the value of S_e for each of the 4-connected neighbors is consulted. If $S_e > 0$ (i.e., the edge in T_c is not long enough) then the neighbor is moved away from the node . If $S_e < 0$ (i.e., the edge T_c is too long) then the neighbor is pulled towards the node. The edge is modified by a length of $\frac{S_e(i,j,k,l)*C_r}{2}$ where $C_r \in [0,1]$ is a parameter of the algorithm. The division by 2 is necessary because each edge is treated twice, once for each end point of the edge. Changing an edge is thus done by modifying each of its end nodes. An important requirement of the algorithm is to preserve the order. So the neighbors are moved as far as S_e and C_r allow without violating equation 3.

The neighboring nodes are changed with coordinate-aligned movements. The movements correspond to the original orientation of the edges in the regular grid (i.e., horizontal and vertical). This means that if the current node is (i,j) and the node to be moved is $(i+1,j)$, the translation of the neighbor will be in the horizontal direction. The movement is computed such that the resulting edge has the expected length determined by S_e and C_r. In the same way, if the neighbor to be moved is $(i,j+1)$, the translation will be in the vertical direction. These movements have the tendency to preserve the rectangular appearance of the quadrilateral defined by $\{T_c(i,j), T_c(i+1,j), T_c(i+1,j+1), T_c(i,j+1)\}$, which for example does not degenerate to a triangle.

Once the iteration has run for all the nodes, the new T_c is generated. Then, a new S_c and a new S_e are calculated from the resulting T_c. S_e is calculated just once per iteration.

The algorithm's convergence depends on the complexity of the scaling field S_g and the value of C_r. If $C_r = 0$, no movement will occur. If $C_r = 1$, the neighbors will be moved the maximum possible displacement. This can make the convergence faster, but can also lead the approximation to a damaged state and then to even not converge at all. A trade-off between speed and quality has to be made in defining the value of C_r.

The convergence factor is measured using the distance between the approximated scaling field S_c and the desired 2D scaling field S_g. This is expressed as the root mean squared error σ. Given a grid of size $n \times m$ the convergence factor is defined as follows.

$$\sigma = \sqrt{\frac{\sum_{j=0}^{m-1}\sum_{i=0}^{n-2} S_e(i,j,i+1,j)^2 + \sum_{j=0}^{m-2}\sum_{i=0}^{n-1} S_e(i,j,i,j+1)^2}{2nm - n - m}}$$

The algorithm terminates when the value of σ is smaller than a defined constant or when after several iterations no considerable

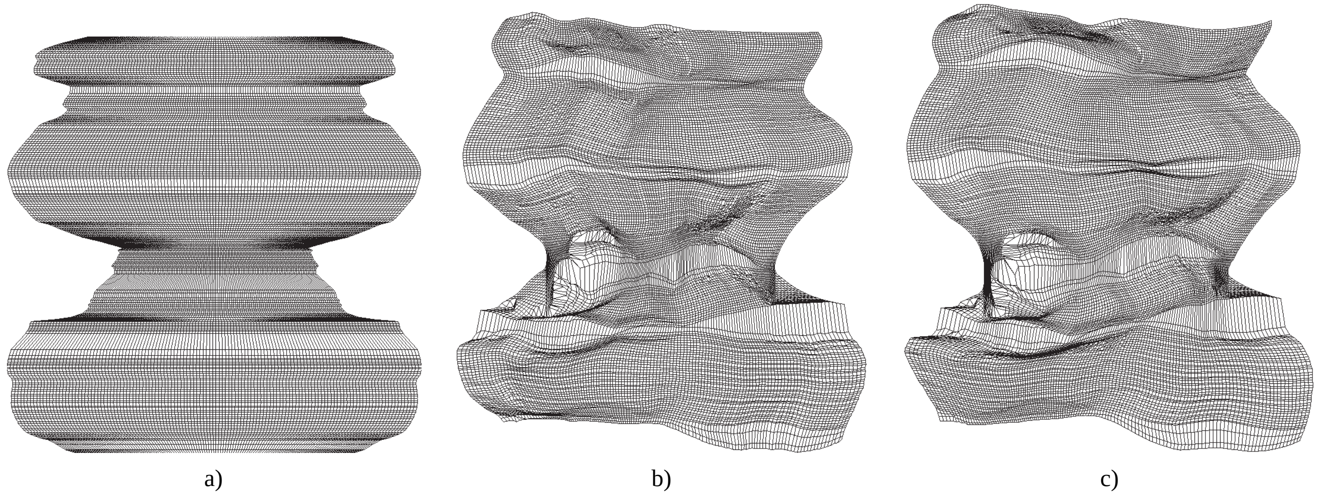


Figure 9: Illustration of the nonlinear 2D scaling algorithm using the same data set than in figure 5. a) Initial T_c corresponding to a 128x171 grid. b) T_c after 960 iterations of the algorithm. c) T_c after 1687 iterations.

improvement occurs anymore. The value of σ does not vary much in successive iterations.

The convergence of the algorithm can be improved by starting with a T_c which is a closer approximation of the desired result than a regular grid. The length of the edges within a horizontal line (i.e., the horizontal edges between nodes with the same j value) are set such that the line approximates the perimeter of the colon in the corresponding cross-section (i.e., $\frac{\sum_{i=0}^{n-2} S_g(i,j,i+1,j)}{n-1}$). The distance between two consecutive horizontal lines is set to an average of the vertical edge lengths in S_g which join the nodes between the two lines (i.e., $\frac{\sum_{i=0}^{n-1} S_g(i,j,i,j+1)}{n}$). This method has shown to accelerate the convergence without altering the results obtained with a regular grid.

The pseudocode that illustrates the presented algorithm is given below.

```
proc NonLinear2DScaling(
           in:    2DScalingField $S_g$, float $C_r$,
           out:   2DTransformation $T_c$)
{
  2DScalingField $S_c$;
  2DScalingField $S_e$;
  boolean $bNoEnd$;

  $T_c$ = IntialGrid($S_g$);
  $S_c$ = Calculate2DScalingField($T_c$);
  $S_e$ = $S_c - S_g$;
  $\sigma$ = CalculateRootMeanSquare($S_e$);
  $bNoEnd$ = FinishApproximation($\sigma$);

  while ($bNoEnd$) {
    forall $T_c(i,j)${ MoveNeighbors($T_c$,$i$,$j$,$S_e$,$C_r$);}
    $S_c$ = Calculate2DScalingField($T_c$);
    $S_e$ = $S_c - S_g$;
    $\sigma$ = CalculateRootMeanSquare($S_e$);
    $bNoEnd$ = FinishApproximation($\sigma$);
  }
}
```

Figure 9a shows the initial grid T_c for the segment of the colon presented in figure 5. The resolution of the grid is 128x171 and the initial value of σ is 0.8008. After 960 iterations T_c has been developed into what can be seen in figure 9b. In this case the value of σ is 0.2808. Figure 9c is the result of the algorithm after 1687 iterations. It can be observed that the grids are similar. In this case the value of σ is 0.2650.

Once the nonlinear 2D scaling has been done, we obtain the parallel projection of the unfolded height field to a 2D grid preserving the edge lengths. This avoids geometry deformation due to the nonuniform sampling. In the next section, a method to resample the undersampled areas and to detect possible missing features is presented.

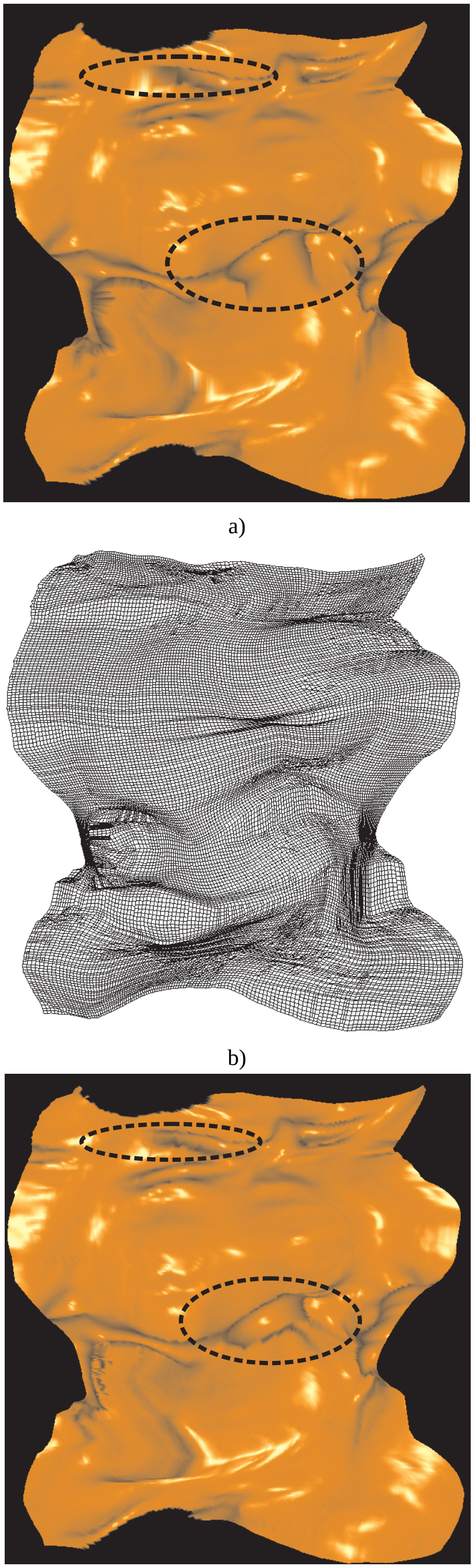
5.3 Resampling

The nonlinear 2D scaling algorithm provides a mapping between the 3D quadrilateral mesh and a 2D grid avoiding geometric deformations. For each node of the 2D grid, the corresponding color obtained in the nonlinear ray casting is assigned. Bilinear interpolation is used to fill the quadrilaterals of the grid. An example can be seen in figure 10a. The areas encircled by dashed ellipses are the same as in figure 6. It can be observed that features are missing due to undersampling.

The undersampled areas are easily identifiable from the 2D grid. A minimum sample step h for the 2D grid is defined. The sample step corresponds directly with a sample step in the 3D space. Each of the 2 quadrilaterals with one or more edges with a length bigger than h are subdivided. For a quadrilateral, the subdivision consists in generating a subgrid whose edge lengths are smaller or equal to h. The result of such a subdivision operation can be seen in figure 10b.

The resulting subdivided grid has assigned color values just in the nodes of the original quadrilateral. Therefore a resampling of the colon surface has to be done for each of the newly generated nodes. Each point in the grid can easily be identified with its corresponding point in 3D using linear interpolation. The 3D points do not correspond to surface points, but they are close to the colon surface. Therefore, curved rays are traced several steps backwards following the nonlinear ray casting algorithm. Then the rays are traced forward again to find the correct surface point. The color of the sampled point is determined by the same procedure as described in section 4.

The resulting color values are mapped directly to the corresponding point in the 2D grid. The results of the resampling can be seen in figure 10c. The encircled areas show areas where features were not present in figure 10a but now are present in figure 10c.



a)

b)

c)

Figure 10: Resampling after the nonlinear 2D scaling. a) 128x171 shaded grid using bilinear interpolation. b) the resulting grid of the nonlinear 2D scaling after resampling c) Shading of the resampled grid.

6 Results

In this section, we describe the results of the virtual colon unfolding algorithm with several data sets.

One data set is a CT volume data of an extracted colon with a resolution of 381x120x632 (see figure 11). The colon is 50 cm long and contains 13 polyps. These polyps have a size between 3.5x2.5 mm and 11.8x9.0 mm. The results of the unfolding procedure for this colon can be seen in figure 12. All the polyps could be detected by inspection. The extracted colon was physically dissected and several pictures of the dissected colon were also taken. These pictures enable a qualitative comparison between the real data and the results of the presented algorithm (see figure 12 right).

A second data set is a segment of the extracted colon with a resolution of 190x120x150. This data set has been used throughout the paper to illustrate the algorithm. This data corresponds with the right-bottom picture of figure 12 and contains three polyps.

The third data set is from a CT data set of a healthy colon segment with a resolution of 198x115x300. The unfolding of the data set and an outside view can be seen in figure 1. This colon is healthy, so no polyps could be found on the unfolding.

For videos and further images please refer to:
`www.cg.tuwien.ac.at/research/vis/vismed/ColonUnfolding/`.

The table 1 shows the calculation times of the presented method on a Pentium II (400MHz). The times correspond to preprocessing since once the method has been executed the physician can inspect the results interactively. The table is divided in three parts. The first part presents the times obtained from generating the nonlinear rays and doing the direct volume rendering. The times depend on the resolution of the sampling, i.e., the number of samples taken in the u and v direction, and the data set resolution.

The middle part of the table shows the times for the nonlinear 2D scaling algorithm. The speed of the convergence of the algorithm depends basically on the grid resolution and its complexity, the value of C_r and σ (i.e., the precision achieved by the unfolding). It can be seen that in the first two data sets the increase of C_r produces a reduction of the convergence rate while for the healthy colon it improves. The value of C_r has to be tuned for each data set.

Finally, a column for the resampling process is presented. The times depend basically on the sampling step of the grid and the difference between the original grid and the resampled grid (i.e., number of resampling points). We present the times with a resampling step h of 0.5 and 1 times the size of a voxel in 3D space.

The time of the whole process depends on the data set and the precision of the results. In total, the time of the whole process, including the distance map calculation, is in the range of hours. This time is a preprocessing time. Once the computation has been done, the resulting 2D grid can be inspected interactively by the physician. The physician uses the result like an overview map. If suspicious areas are identified he inspects them more carefully using other tools like cross-sections of the original data set or conventional rendering of the area.

Experiments have shown that the algorithm is quite sensitive to the smoothness of the path. For a good result, it is important that the path is smooth and it has as many linear segments as possible.

The nonlinear ray casting features a problem that occurs when a ridge line is inside the colon. Then, there will be rays that would never reach the surface of the colon. Although this case never appear in the tested data sets, a solution to this situation has to be studied. A possible solution would be to combine the path distance map with the surface distance map.

In the presented algorithm just the areas that are directly projected to the central path are taken into account. Areas which are behind mushroom-like folds are not visualized. Using more than one surface point per ray, could solve the problem of occluded areas due to mushroom-like folds.

Data Set	Nonlinear Ray Casting		Nonlinear 2D Scaling				Resampling	
	Grid	time(min)	C_r	iterations	σ	sec./iter.	h	time(min)
Extracted Colon	128x890	21.65	0.3	352	0.2797	0.797	0.5	83.80
			0.5	601	0.3165		1.0	26.50
Segment of the Extracted Colon	126x171	3.4	0.3	960	0.2808	0.141	0.5	15.46
			0.5	1687	0.3511		1.0	5.32
Healthy Colon	200x1178	13.2	0.3	70	0.2799	1.656	0.5	59.18
			0.5	58	0.2797		1.0	6.75

Table 1: This table illustrates the calculation times depending on the different parameters of the algorithm. The times are calculated using a Pentium II CPU (400MHz).

7 Conclusions and Future Work

In this paper, a new unfolding method to inspect cavity organs has been presented. The goal is to enable the physician to inspect and get as much information as possible of the inner organ surface at a first glance. The problematic areas can be identified quickly and inspected later in more detail. The presented approach solves the problem of double appearance of polyps using nonlinear ray casting. Compensation of the distortions due to the unfolding of the colon is achieved using an iterative method called nonlinear 2D scaling which is similar to the nonlinear magnification fields used in information visualization. Finally, a method is presented to resample in areas where features have been missed due to undersampling. The methods have been tested with several data sets. One of them enabled a qualitative comparison of the resulting images with images of the corresponding extracted colon.

Improvements of the method for the cases of ridge lines inside the colon and occluded areas due to mushroom-like folds are subject of future work. Acceleration of the different preprocessing steps is also a subject of future study.

The presented method has shown to be promising, and it should be tested with more data sets of real pathological cases. The algorithm is not restricted to the colon. It has the potential of being used for any tubular organ.

Acknowledgements

The work presented in this publication has been funded by the V^{is}M^{ed} project. V^{is}M^{ed} is supported by *Tiani Medgraph*, Vienna (`http://www.tiani.com`), and the *Forschungsförderungs fonds für die gewerbliche Wirtschaft*, Austria.

See `http://www.vismed.at` for further information on this project.

We thank Dr. Erich Sorantin from the Department of Radiology in Graz for his collaboration and for providing the data sets used in this paper and the images of the dissected colon.

References

[1] R. H. Abraham and C. D. Shaw. *Dynamics: The Geometry of Behavior*. Addison-Wesley, 1992.

[2] A. Vilanova Bartrolí, A. König, and E. Gröller. Cylindrical approximation of tubular organs for virtual endoscopy. In M.H. Hamza, editor, *Proceedings of Computer Graphics and Imaging 2000*, pages 283–289. IASTED/ACTA Press, 2000.

[3] A. Vilanova Bartrolí, R. Wegenkittl, A. König, E. Gröller, and E. Sorantin. Virtual colon flattening. In *VisSym '01 Joint Eurographics - IEEE TCVG Symposium on Visualization*, 2001. http://www.cg.tuwien.ac.at/research/vis/vismed/ColonFlattening.

[4] E. Gröller. Nonlinear ray tracing: Visualizing strange worlds. *The Visual Computer*, 11:263–274, 1995.

[5] S. Haker, S. Angenent, Allen Tannenbaum, and R. Kikinis. Nondistorting flattening maps and the 3D visualization of colon CT images. *IEEE Transactions on Biomedical Engineering*, 19(7):665–671, July 2000.

[6] L. Hong, S. Muraki, A. Kaufman, D. Bartz, and T. He. Virtual voyage: Interactive navigation in the human colon. In *SIGGRAPH 97 Conference Proceedings*, Annual Conference series, pages 27–34. ACM SIGGRAPH, Addison Wesley, August 1997.

[7] T.A. Keahey and E.L. Robertson. Nonlinear magnification fields. In *IEEE Information Visualization*, pages 51–58, 1997.

[8] G. Lohmann. *Volumetric Image Analysis*. Chichester Wiley, 1998.

[9] D.S. Paik, C.F. Beaulieu, R. B. Jeffrey, Jr. C.A. Karadi, and S. Napel. Visualization modes for CT colonography using cylindrical and planar map projections. *Journal of Computer Tomography*, 24(2):179–188, 2000.

[10] E. Sorantin, E. Balogh, K. Palagy, G. Werkgartner, E. Spuller, and S. Loncaric. *MEDICAL RADIOLOGY - Diagnostic Imaging*, chapter "Technique of Virtual Dissection of the Colon based on Spiral CT data". Springer Verlag Press, 2001.

[11] M. Wan, Q. Tang, A. Kaufman, Z. Liang, and M. Wax. Volume rendering based interactive navigation within the human colon. In *IEEE Visualization '99*, pages 397–400. IEEE, nov 1999.

[12] G. Wang, S.B. Dave, B.P. Brown, Z. Zhang, E.G. McFarland, J.W. Haller, and M.W. Vannier. Colon unraveling based on electrical field: Recent progress and further work. In *Proceedings SPIE*, volume 3660, pages 125–132, May 1999.

[13] G. Wang, E. G. McFarland, B. P. Brown, and M. W. Vannier. GI tract unraveling with curved cross sections. *IEEE Transactions on Medical Imaging*, 17:318–322, 1998.

[14] G. Wang and M.W. Vannier. GI tract unraveling by spiral CT. In *Proceedings SPIE*, volume 2434, pages 307–315, 1995.

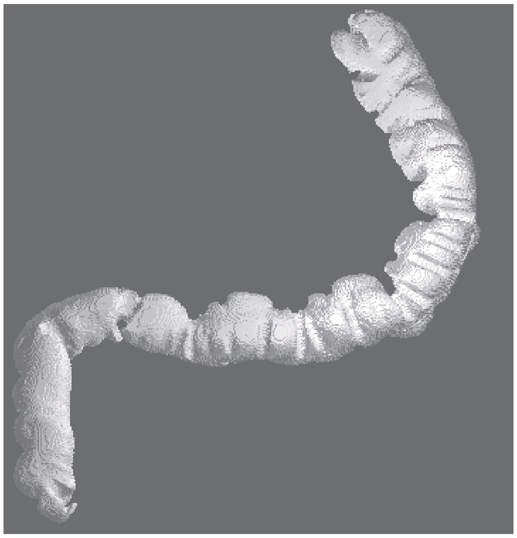


Figure 11: Outside view of the segmented surface of the extracted colon CT data set with resolution 381x120x632.

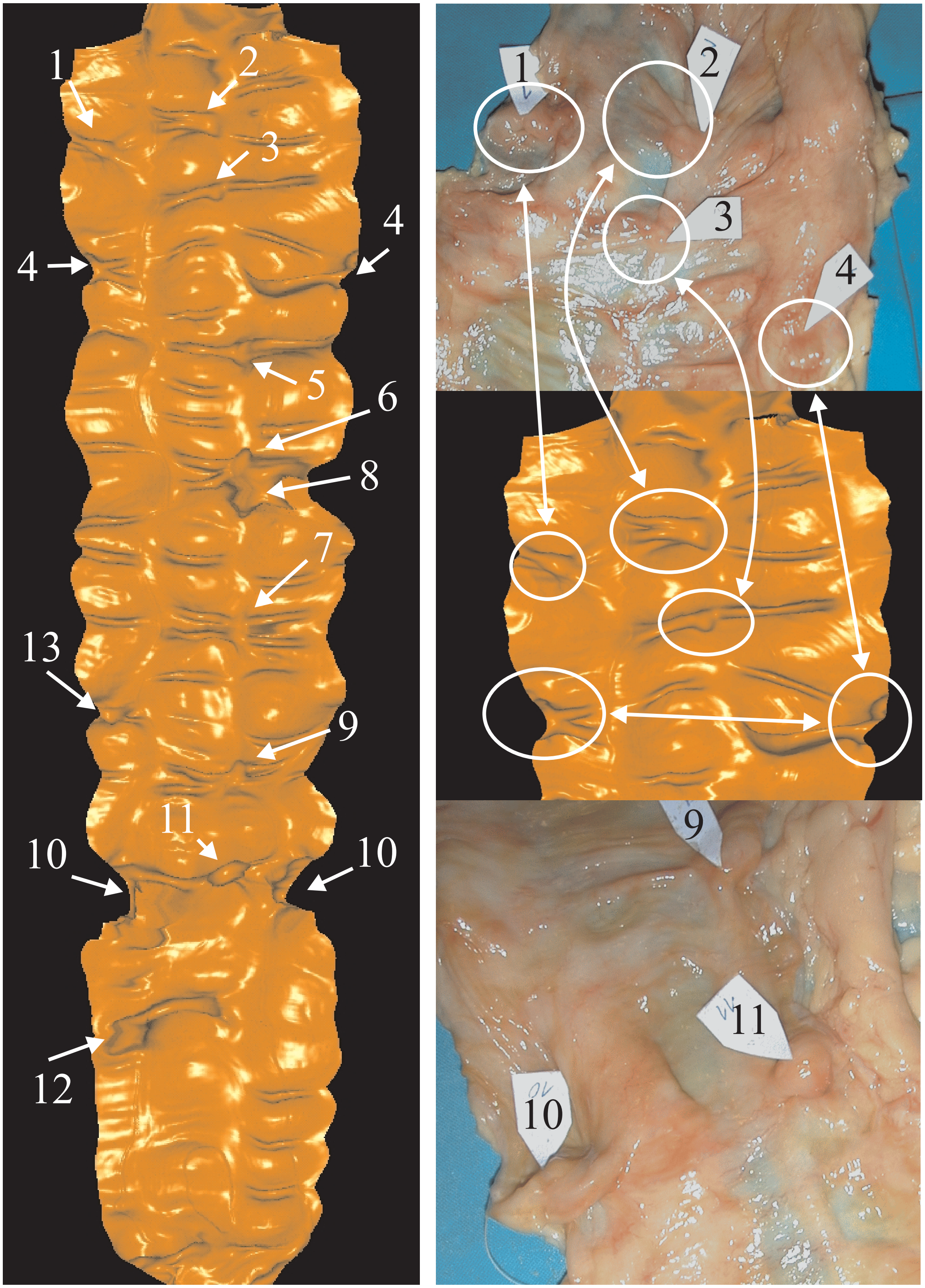


Figure 12: **Left** Virtually unfolded extracted colon with the polyps numbered according to the real dissection. **Right** qualitative comparison of the virtually unfolded colon with pictures taken from the real dissection. **Right bottom** image corresponding to the segment of the extracted colon data set used throughout this paper as example. The orientation in which the pictures were taken does not correspond with the orientation of the virtually unfolded colon.