Medial Skeletons for Haustral Fold Detection in Colonoscopy

Samuel Ehrenstein*[1][0000−0003−2739−5076], Sarah McGill[2][0000−0002−4006−2703], Julian Rosenman[2], and Stephen M. Pizer[1][0000−0002−4250−6531]

University of North Carolina at Chapel Hill, Chapel Hill NC 27514, USA
{ehrensam,pizer}@cs.unc.edu, rosenmju@med.unc.edu, mcgills@email.unc.edu

Abstract. While performing a colonoscopy, a doctor may need to be able to return to a previously viewed location within the colon. In particular, when regions of the colon are identified as unsurveyed (inadvertently not viewed), the colonoscopist must be able to find their way back to them. Returning to previously-seen locations in the colon requires that it work under the types of deformations the colon undergoes and recognizes the fact that the colon does not have significant photometric texture. In this work, we present a novel approach to this task. Salient geometric features known as haustral folds are used as markers to enable precise guidance back to a location, for example, to a given fold next to an unsurveyed region. We present two of the first methods, both medial-shape-based, for detecting individual haustral folds from colonoscopic video: CurvFold, which uses surface curvature derived from monocular depth estimation, and DeepFold, which uses state-of-the-art object detection to directly identify haustral folds. We demonstrate that CurvFold and DeepFold are able to detect haustral folds in video frames with mean chamfer distances to the ground truth of 29 and 9 pixels respectively.

Keywords: Medial representation · Colonoscopy · Virtual tagging · Haustral folds.

Colorectal cancer (CRC) is the third-deadliest cancer in the United States, but chances of survival are greatly increased by regular screening. The "gold standard" screening method for CRC is colonoscopy, in which an endoscope is used to visually inspect the patient's colon for abnormal growths known as polyps. However, this is limited by how much of the colon surface is actually seen. In colonoscopy, a *blind spot* is a region of the inside surface of the colon that is not surveyed by the colonoscopist. In [23], a method was proposed for detecting blind spots during colonoscopy. However, this method does not identify the blind spot at a time shortly after it is passed, so after blind spots are detected, it is necessary to guide the colonoscopist back to re-survey them. This is complicated by the fact that the colon is an elastic organ lacking obvious visual landmarks, or even rich texture features that can be used with standard methods from computer vision. Instead, the most notable features of the colon are the wrinkles

* Corresponding author.

running around the circumference, known as haustral ridges or folds. With a 3D reconstruction, a blind spot can be localized based on nearby folds. Thus, once a blind spot is assigned to a fold, the colonoscopist can be guided to return to it by simply highlighting it on the screen as a visual target. This reduces the problem to one of object detection, in which all objects in an image (in this case, folds) are marked with a defining shape (usually a mask or bounding box, but in this work a skeletal center curve). In this work, we present the first two methods ever of detecting individual folds from colonoscopic video only.

Our problem is to detect objects that are regions on a surface. This is made more complex by the fact that these fold regions are defined by their geometric properties but with wide variations in region of surface coverage and curvature and even moderate deformations over time. This means that standard methods for object detection are not suitable for this purpose. Instead, we propose two novel medial-shape-based methods detecting folds, followed by a novel, detector-agnostic algorithm for producing representative center curves for each fold. The first fold detection method uses a novel geometric definition of haustral folds in order to segment them using geometry (specifically, mean curvature) derived from monocular depth estimation; we have named this method CurvFold. The second employs a modern, Vision Transformer-based detector to directly predict fold center curves from the video frame alone; we have named this method DeepFold. The outputs of either method can then be fed into a novel fold instance separation algorithm that splits and joins center curves based on their medial branching structure and predicted depths. We also introduce a novel algorithm for approximating curvature of a surface reconstructed from a pixel-level depth map that leverages its existing grid structure.

The contributions of this paper are as follows:

- We introduce a novel definition of haustral folds based on mean curvature, as well as a novel method for computing approximate mean curvature from a depth map.
- We introduce the first two methods ever for instance segmentation of folds in optical colonoscopy video, one (CurvFold) based on curvature and one (DeepFold) based on a direct prediction network.
- We demonstrate the suitability of skeletal representations for the task of visual navigation using haustral folds.
- We demonstrate the efficacy of our methods on both synthetic and clinical data.

1 Related Work

Here we review existing research on haustral fold detection and instance segmentation, on the shortcomings in these works as applied to the problem of fold detection, and on how our proposed methods differ.

1.1 Haustral Fold Detection

As mentioned, haustral folds are the most prominent geometric features of many quasi-tubular internal organs, including the colon. Thus, they provide a potential solution to the aforementioned problems involving the alignment of different views of the same colon or segment of colon. This approach has been extensively used in CT colonography, where it is often necessary to align the two 3D reconstructions of a patient's colon taken while they are supine and prone. Specific methods for alignment based on haustral folds include [6, 15, 25], which all define folds via curvature analysis on the 3D colon model or models (as laid out in the previous subsection). However, these are not suitable for use with optical colonoscopy. This is because existing 3D reconstruction methods that can be used during the colonoscopy procedure, such as SLAM and its variants, do not provide a sufficient level of detail or of accuracy in reconstructing the colon surface. Furthermore, they require expensive geometric computations or optimizations, making them impractical for real-time or near-real-time use.

Fold detection methods using monocular video have included XDCycleGAN [11] and its extension in FoldIt [10], both of which trained neural networks to take individual video frames from colonoscopy and output a binary label for each pixel of "fold" or "not fold", i.e., binary semantic segmentation. This was followed by our previous method [3] that directly learned this same task from partial, high-confidence annotation of a large dataset and achieved superior performance to the previous methods. While these neural network-based methods deliver real-time performance, they do not distinguish individual folds. Moreover, their definition has the weakness that multiple folds and the haustra separating them can be labeled as a single fold region when viewed along the tubular axis. As the view will be along that axis when navigating back to a marked fold, this behavior is undesirable. We instead use curvature, which due to its geometric definition, is far more reliable at separating individual fold instances without significantly sacrificing accuracy.

1.2 Skeleton Detection

To account for variations in the boundary of 2D regions, we use skeletal representations to capture shape while discarding unnecessary information. The concept of a medial axis skeleton was first introduced by Blum in 1967 [1]. It is a representation of a shape that captures its topological and geometric properties. The medial axis skeleton of a shape is defined as the set of points that have more than one closest point on the boundary of the shape. Medial axis skeletons can be computed using algorithms such as [22]. They can be used for various applications, such as shape analysis, object recognition, and image segmentation. However, medial axis skeletons suffer from the major drawback of being very sensitive to small perturbations to the shape's boundary, which can lead to many spurious branches and a lack of robustness. However, the concept of skeletal representation, that a 2D shape can be represented by a set of 1D curves, is still attractive for many problems of shape analysis.

In order to be robust to noise, skeletal representations (s-reps) must be quasi-medial; that is, they must allow for some deviation from the true medial axis. This opens the door to deep learning-based methods that treat the generation of s-reps as that of producing a binary pixel mask from an image, which also has the advantage of being able to produce s-reps directly from photographs without having to perform object boundary detection separately [9,16,19]. DeepFlux [20] built on this by training a model to predict the direction vector of the gradient of the distance from the skeleton at each pixel; skeletal points are then those with positive average outward flux of this function. However, all of these methods are ultimately based on classifying image pixels. But skeletons are not defined as collections of pixels; they are defined as collections of curves, and there is no existing work on detecting haustral folds in video that considers them in terms of anything other than pixels.

An alternate approach to finding curvilinear skeletons is found in BlumNet [24], which uses a model based around Deformable DETR [26] to predict line segments that would be part of the skeleton. These are then merged and cleaned using morphological operations. The principle of detecting curve-like features by detecting the line segments that compose their discretization is also found in [12, 13], which instead use a model based on YOLOv2 [18] to trace lane boundaries in self-driving car applications. This principle is attractive for the problem of fold separation, as folds are often thin and elongated and thus can be well-represented by a set of curves.

2 Preliminaries

It is first necessary to recall the definition of curvature. The curvature of a surface is, broadly speaking, a measure of how the surface normal changes with a small step in a certain "walking" direction. More specifically, consider a surface $M \subset \mathbb{R}^3$ and a point $\mathbf{p}$ on M. Let T_p be the tangent space of M at $\mathbf{p}$. For any walking direction designated by a unit vector $\mathbf{v} \in T_p$, the *shape operator* is the operator S such that $S(\mathbf{v}) = \nabla_{\mathbf{v}} \mathbf{N}_p$, where $\mathbf{N}_p$ is the unit outward normal vector at $\mathbf{p}$. Then the *normal curvature* k in the direction $\mathbf{v}$ is

$$k(\mathbf{v}) = S(\mathbf{v}) \cdot \mathbf{v} \tag{1}$$

The mean curvature H is defined as the mean value of k averaged over all possible walking directions. We can now define fold regions as those regions where $H > 0$, as this has been observed to be true in the clinical setting.

2.1 Discrete Curvature

Approximations must be made in order to numerically compute measures of discrete curvature. One such measure is umbrella curvature [4]. Given a surface $M \subset \mathbb{R}^3$ represented by a point cloud $P = \{\mathbf{p}_1, \mathbf{p}_2, \dots \mathbf{p}_N\}$ such that each $\mathbf{p}_i \in M$, the umbrella curvature at point $\mathbf{p}$ is defined as

$$H_{um}(\mathbf{p}) = \sum_{i=1}^{8} \left\| \frac{\mathbf{m_i} - \mathbf{p}}{\|\mathbf{m_i} - \mathbf{p}\|_2} \cdot \mathbf{N}_p \right\|_1 \tag{2}$$

where $\mathbf{m}_i$ is the i-th 8-homogeneous nearest neighbor [4] of $\mathbf{p}$.

3 Methods

This section explains the methods we used to detect individual haustral folds in a video frame, which has not previously been achieved. It will proceed in two parts: first (Section 3.2), instance segmentation of folds; and second (Section 3.3), skeletal filtering of fold regions. A novelty lies in the combination of these methods. In addition, Section 3.2 defines a novel definition of mean curvature for a point cloud that exploits the grid structure of the depth map used to compute it, as well as a novel means of computing a fold region using mean curvature instead of principal curvatures. The use of medial axes in 2D in the production of the final fold segmentation is also novel. The full system pipeline is shown in Fig. 1, depicting both CurvFold and DeepFold for fold center curve detection, which are then processed using the same method for fold region segmentation and clustering.

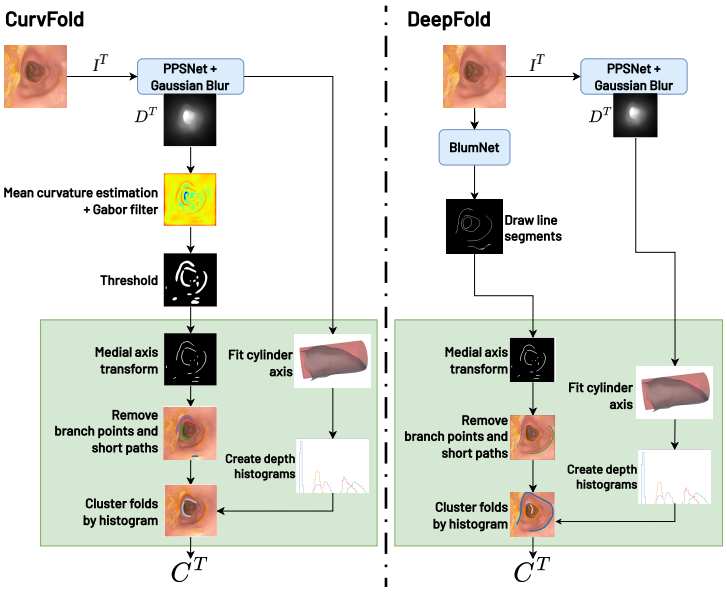

Fig. 1. Algorithms for fold instance segmentation, one using curvature (CurvFold) and one using deep learning (DeepFold). Note that the components inside the green boxes are identical; this is because the method for splitting and joining folds is detector-agnostic. The difference is the method for producing fold center curves. The figure is best viewed in color and with zoom.

3.1 Computation of Curvature

First, recall the definition of the curvature of a surface: broadly speaking, it is a measure of how the surface normal changes with a small step in a certain "walking" direction. The *mean curvature* $H(\mathbf{p})$ at a point $\mathbf{p}$ on a surface $M \in \mathbb{R}^3$ is the normal curvature averaged over all possible walking directions. We can now define folds as regions where $H(\mathbf{p}) > 0$, as this has been observed to be true in the colon.

When working with real data, approximations must be made in order to numerically compute measures of discrete curvature. One such measure is umbrella curvature (Eq. 2). We provide a discrete approximation of mean curvature, the *k-discrete mean curvature* of a point cloud, Q_k:

$$Q_k(\mathbf{p}) = -\frac{1}{k}\sum_{j=1}^{k}\left(\frac{\mathbf{m}_j - \mathbf{p}}{\|\mathbf{m}_j - \mathbf{p}\|_2}\right) \cdot \mathbf{N}_p \tag{3}$$

where $\{\mathbf{m}_1, \mathbf{m}_2 \dots \mathbf{m}_k\}$ are k points sampled in an approximate circle as close to evenly spaced angularly as possible; in this work, we used a circle of 12 points at 2 pixels away. This method is derived from the observation that the walking direction $\mathbf{m}_j - \mathbf{p}$ is approximately tangent to the surface close to $\mathbf{p}$. The

dot product in (3) is the projection of the unit nosedive vector in the walking direction onto the tangent plane; in other words, the swing of the tangent vector in that direction, the normal curvature.

3.2 Instance Segmentation of Folds

We introduce a new definition of haustral folds. A *fold region* R is a connected region *in a 2D image of the colon* such that for all pixels $p \in R$, $Q(p) > \epsilon$. ϵ is a small positive constant, 0.03 in this work, to account for discretization error. This definition is inspired by works such as [25] and [15], which define folds based on negative Gaussian curvature. However, the sign of Gaussian curvature is not robust to noise when both principal curvatures are close to 0. This makes a definition like the one we introduce more attractive, as it is robust even with principal curvatures close to 0. It follows from this definition of a fold region that a physical fold on the colon surface should be a connected 2D subset of the surface that has positive mean curvature. The full algorithm is shown in the green boxes in Fig. 1.

Fig. 2 shows an example of fold region detection using our method. We start from the ground-truth depth map to demonstrate the effect of our method. The depth map is converted to a point cloud by backprojecting each pixel into 3D space using the camera intrinsic matrix. The mean curvature at each pixel is then computed using our method from Section 3.1, resulting in a mean curvature map. Finally, fold regions are detected by thresholding the mean curvature map at ϵ.

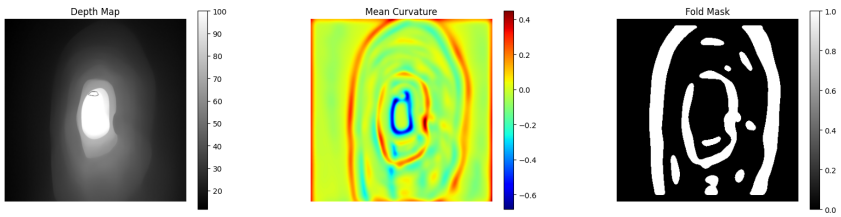

Fig. 2. Left: Ground-truth depth map (in mm) of a frame from C3VDv2 [5]. **Center**: Mean curvature map computed using our method. **Right**: Fold regions detected using our definition, shown in white.

3.3 Fold Separation

We next need to be able to separate regions that are unlikely to correspond to a single fold. We do this by using a skeletal representation, specifically the medial axis transform of each region [22], as shown on the left of Fig. 3. To our knowledge, this is the first use of skeletal representations in the context of haustral folds, or indeed in the endoscopic domain at all.

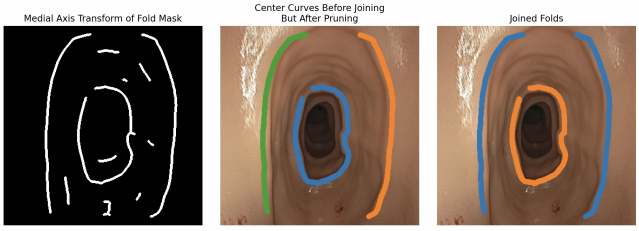

Fig. 3. Left: Initial medial axis transform of fold regions. **Center:** Center curves from the medial axis transform after branch points and short paths are removed, but before final clustering. Each skeleton segment at this step is a different color. **Right:** Final fold center curves after clustering. Each different color is a separate center curve C_j. Note that the green and orange center curves from the center image have been merged into a single center curve (blue), as they have similar depth histograms (see Fig. 4). The blue center curve from the center image has been re-colored orange in the right image; this reflects the re-indexing after clustering.

For a 2D fold region R_i, the set of boundary pixels $\Omega_i = \{p \in R_i | B(p) < 8\}$, where $B(p)$ is the number of nonzero 8-connected neighbors of a pixel p. The skeleton ζ_i of R_i is computed by iteratively removing boundary pixels of R_i until no more pixels can be removed without changing the connectivity of the region. Each ζ_i is then processed further to split the skeleton into a set of paths representing center curves of branches of one or more fold regions. To do this, branch points are detected by computing the degree of each pixel in the skeleton and removing pixels with more than 2 neighbors, along with any other skeletal pixels within a distance of 3 pixel units from them.

The connected components of the pixel connectivity graph are now all topological paths. Components with under 70 pixels are removed, and their regions and skeletons are deleted. Each remaining skeleton ζ_i now consists of a set of one or more connected paths, the union of which can be taken as the new set of center curves and re-indexed accordingly. Every pixel p that is in any of the remaining fold regions is now associated with the path to which it is closest, up to a maximum distance of half the length threshold (70 px in this work). The result of this is depicted in the center of Fig. 3.

3.4 Fold Clustering

The final step is to cluster fold center curves that are part of the same fold together, and this is done by taking advantage of the fact that folds tend to have mostly disjoint z-coordinates when the camera coordinate frame is rotated such that the z-axis is aligned with the longitudinal axis of the cylinder fitted to the depth map. Specifically, we first create a normalized histogram G_i of the z-coordinates of each point in each remaining ζ_i in the cylinder axis-aligned coordinate system. Each histogram in the same frame has the same 50 equally-spaced bins, enabling direct comparison. An example is shown in Fig. 4.

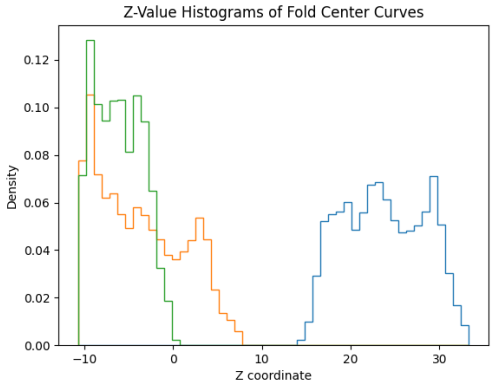

Fig. 4. Histograms of z-coordinates of points in each fold center curve in the axis-aligned coordinate system. Each histogram is normalized to sum to 1, and each is colored according to the corresponding center curve in the center panel of Fig. 3. Histograms are then clustered based on Hellinger distance [7], the idea being that if two center curves are part of the same fold, they should have similar z-coordinate distributions. Due to how the cylinder is fitted, the z-coordinates have an arbitrary offset and scale factor; this does not affect the Hellinger distance between them.

In order to cluster these histograms, we apply a hierarchical clustering method [8] (as implemented in Scipy [14]) to cluster the normalized histograms. We use pairwise Hellinger distances [7] as the distance matrix. An example of the final fold regions after clustering is shown at right in Fig. 3. The clusters of center curves ζ_i are then re-indexed as C_j and each is assigned a numerical label equal to $j + 1$.

3.5 Fold Detection Via Deep Learning

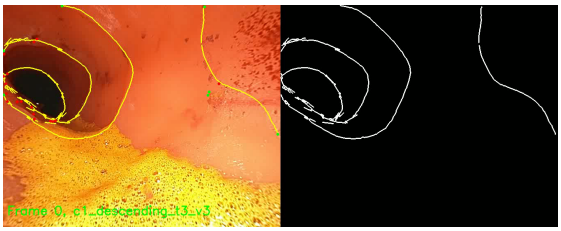

Fig. 5. Left: BlumNet output with predicted line segments shown in yellow and predicted branch points shown in red. **Right:** Binary mask of predicted segments. This is then processed with the described morphological operations to separate different folds and prune small branches.

Since the method described in Sections 3.2 to 3.4 is detector-agnostic, it can be used with methods that produces a set of fold center curves. The method described is somewhat computationally expensive, meaning that this system would benefit from a faster method of producing such center curves. Since the center curves are being produced from the medial axis skeletons of 2D projections of fold regions, it follows that a neural network could be trained to directly produce these skeletons, which would then be processed using the method of Section 3.2 to produce fold regions. We chose to use BlumNet [24], which is based on Deformable DETR [26] and works by predicting line segments that would be part of the skeleton. We trained BlumNet with 128 point features and 3 encoder and decoder layers each for 100 epochs. For training data, we used the raw medial axis skeletons of the 2D ground truth fold regions from C3VDv2 [5]. The predicted skeletons were then processed using the method of Section 3.2 to split, filter, and cluster the generated center curves. An example of the predicted skeletons is shown in Fig. 5. Note the false positive center curve on the right side of the frame.

4 Experiments

4.1 Metrics

In order to evaluate the performance of our fold detection methods, we used the following metrics:

Mean Minimum Distance The mean minimum distance is defined as the mean of the chamfer distance between each predicted fold center curve and the closest ground-truth fold center curve. This metric captures how close the predicted fold center curves are to the ground truth fold center curves, with a lower value indicating better performance.

Pointwise Precision Following Csurka et al. [2], pointwise precision is defined as the fraction of points on the predicted fold center curves that are within a certain distance threshold (30 px in this case) of a ground-truth fold center curve. It is illustrated in Fig. 6. This metric captures how many of the points on the predicted fold center curves are actually close to the ground truth, with a higher value indicating better performance. This differs from fold precision because it considers how close points are to *any* fold, not just the fold it was matched to. This means that errors in fold joining (Sec. 3.4) will not affect pointwise precision, so the discrepancy between pointwise precision and fold precision can be used to evaluate the relative performance of fold joining.

Pointwise Recall Pointwise recall is defined as the fraction of points on the ground-truth fold center curves that are within a certain distance threshold (30 px in this case) of a predicted fold center curve. This metric captures how many

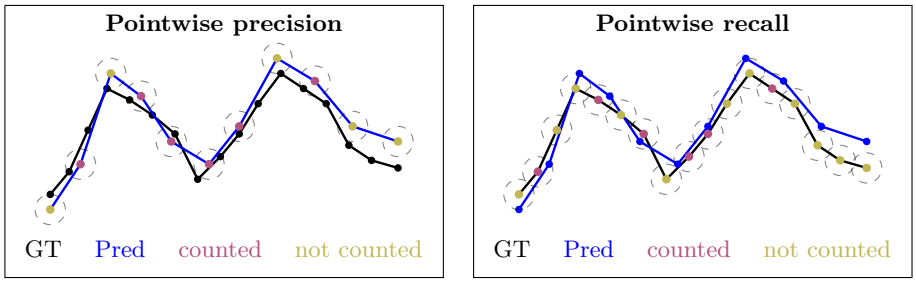

Fig. 6. Polylines illustrating the meaning of pointwise precision (left) and of pointwise recall (right). If a curve such as a skeletal center curve has a ground truth (GT) and a predicted result (Pred), points within the illustrated circles, all of the same radius, would be counted in the precision, resp. recall. Points shown in yellow are not counted. On the left, 6 out of 11 predicted points are within the threshold distance of a ground-truth point, giving a pointwise precision of 0.55. On the right, 6 out of 17 ground-truth points are within the threshold distance of a predicted point, giving a pointwise recall of 0.35. Note that while in this example, some points close to the polyline are not counted because they are not within the threshold distance, in this work, adjacent points will be adjacent pixels and thus the spacing between points is not a concern.

of the points on the ground-truth fold center curves are correctly detected, with a higher value indicating better performance. Like pointwise precision, this metric is not affected by errors in fold joining.

Skeleton F1 Score Inspired by the boundary F1 score used in evaluation of segmentation [2, 21], we introduce a new metric called the *skeleton F1 score*. This is defined as the harmonic mean of pointwise precision and recall. The skeleton F1 score ranges from 0 to 1 (inclusive), where a higher score indicates that the predicted skeleton is closer to the ground truth skeleton. This is again not affected by errors in fold joining, and thus captures the shape accuracy of the predicted fold center curves alone, so a high skeleton F1 score indicates that the predicted fold center curves are close to the ground truth fold center curves, even if they are not correctly joined into folds.

In addition to its use in this work, the skeleton F1 score can also be used to compare more complex 2D skeletal representations. This can be done after performing Procrustes alignment to account for translation, rotation, and scale so that the skeleton F1 score more accurately describes difference in shape alone.

Fold Precision In the same vein as the pointwise precision of [2], fold precision is defined as the fraction of predicted fold center curves that are within a certain distance threshold of a ground-truth fold center curve. We used 30 pixels. This metric captures how many of the predicted folds are actually correct, with a higher value indicating better performance.

Fold Recall Fold recall is defined as the fraction of ground-truth fold center curves that are within a certain distance threshold of a predicted fold center curve, using the same distance threshold as precision. This metric captures how many of the ground-truth folds are correctly detected, with a higher value indicating better performance.

F1 Score The F1 score is the harmonic mean of fold precision and fold recall. It provides a single metric that balances both metrics, with a higher value indicating better performance.

4.2 Detection of Folds on C3VDv2

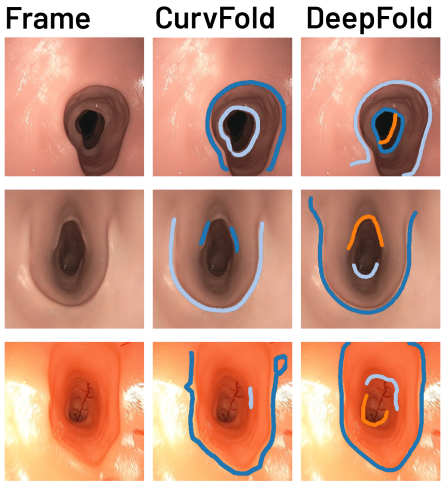

Fig. 7. Example of fold detection results on 3 frames from C3VDv2 [5]. The left column shows the original video frame, the center column shows the fold center curves detected by CurvFold, and the right column shows the fold center curves detected by DeepFold. In all three examples, both are able to detect the fold closest to the camera, but DeepFold performs better at detecting the folds far from the camera.

We first used the dataset C3VDv2 [5] to perform detection experiments against ground truth. This data is produced from robotically controlled colonoscopy on a plastic colon phantom. We randomly selected 27 of the 169 sequences and computed fold center curves on each frame using both methods presented above (CurvFold and DeepFold). We then compared these to ground-truth fold center curves derived from taking the mean curvature of the ground-truth 3D colon models from [5] and thresholding at 0.02. Images are rescaled to 384 by 384 pixels. The results are shown in Table 1, with examples in Figure 7.

Table 1. Fold detection performance comparison on C3VDv2. Results are presented as mean ± standard deviation across all frames considered. Higher values for all metrics except Mean Minimum Distance and Compute Time indicate better performance.

Metric	DeepFold	CurvFold
Mean Minimum Distance (px)	**9.407** ± 7.105	29.27 ± 23.73
Fold Precision	**0.9246** ± 0.0986	0.7179 ± 0.2587
Fold Recall	**0.8488** ± 0.1304	0.6207 ± 0.2658
Fold F1 Score	**0.8677** ± 0.1003	0.6312 ± 0.2225
Pointwise Precision	**0.9972** ± 0.0055	0.9213 ± 0.1379
Pointwise Recall	**0.9673** ± 0.0571	0.8123 ± 0.1616
Skeleton F1 Score	**0.9794** ± 0.0389	0.8529 ± 0.1356
Compute Time (s)	**0.5891** ± 0.0593	0.9737 ± 0.2115

DeepFold outperforms CurvFold on all metrics. However, DeepFold has a much higher fold precision and recall, indicating that it is better at correctly identifying folds and avoiding false positives. It also has a higher pointwise precision and recall, indicating that the predicted fold center curves are closer to the shape and length of the ground truth. Moreover, DeepFold is about 1.7 times as fast as CurvFold. This is due to the fact that curvature computation is much more expensive than the forward pass of a neural network, as well as curvature-based fold detection producing more center curves that need to be pruned. DeepFold does not do this because it is trained on data that has already been processed to remove these.

Detection on clinical data Following the method from [3], we labeled a set of 463 frames from video of colonoscopies (all with the same camera intrinsic matrix) at a university hospital with the closest 3 or 4 folds to the camera, as this is generally the range for which both methods can detect folds. Images were again rescaled to 384 by 384 pixels. We then computed the same metrics as before for CurvFold and DeepFold. Results are shown in Table 2 and examples are shown in Fig. 8. While all metric values except compute time are lower on clinical data compared to the synthetic data in Table 1, DeepFold again outperforms CurvFold on all metrics except pointwise recall, where CurvFold has a slightly higher value. DeepFold and CurvFold have comparable performance in terms of fold recall, meaning that both are similarly capable of avoiding false negative detections. However, DeepFold has a much higher fold precision, indicating that it is better at correctly identifying folds and avoiding false positives. It also has a higher pointwise precision, indicating that the predicted fold center curves are closer to the shape of the ground truth. Finally, DeepFold is about 2.2 times faster than CurvFold on clinical data. While DeepFold's average runtime only increased by approximately 0.02 seconds over synthetic data, CurvFold's average runtime increased by 0.39 seconds (almost 40%). This comes from the increased time for removing branch points and short paths, which is due to the fact that there tend to be more fold regions detected in clinical data. This in turn is due to the fact that the depth estimation used [17] produces noisier depth maps on

clinical data than on C3VDv2, which is expected given that it was trained on synthetic data.

Table 2. Fold detection performance comparison on clinical data. Results are presented as mean ± standard deviation across all frames considered. Higher values for all metrics except Mean Minimum Distance and Compute Time indicate better performance.

Metric	DeepFold	CurvFold
Mean Minimum Distance (px)	**38.24** ± 21.34	53.16 ± 27.55
Fold Precision	**0.5235** ± 0.2588	0.3986 ± 0.1904
Fold Recall	**0.6489** ± 0.3044	0.6417 ± 0.3031
Fold F1 Score	**0.5576** ± 0.2526	0.4737 ± 0.2116
Pointwise Precision	**0.7404** ± 0.1539	0.6887 ± 0.1589
Pointwise Recall	0.8643 ± 0.1426	**0.8993** ± 0.1136
Skeleton F1 Score	**0.7869** ± 0.1241	0.7660 ± 0.1323
Compute Time (s)	**0.6163** ± 0.0667	1.367 ± 0.253

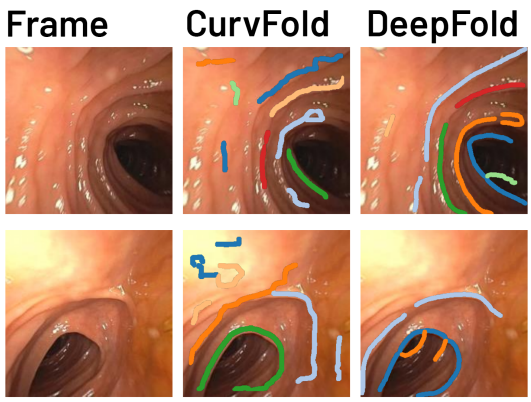

Fig. 8. Example of fold detection results on 2 frames from clinical colonoscopy video. The left column shows the original video frame, the center column shows the fold center curves detected by CurvFold, and the right column shows the fold center curves detected by DeepFold.

5 Discussion

The results demonstrate the efficacy of both CurvFold and DeepFold for detecting haustral folds using only video frames as sensor input. The high recall values on clinical data indicate that most folds are being detected, which is important

for the purpose of virtual tagging. The lower precision value suggests that there are many false positives, which could be addressed in future work. The use of mean curvature as a basis for fold detection is validated by the results.

DeepFold outperforms CurvFold on almost all metrics on both synthetic and clinical data. This is likely due to the fact that the monocular depth maps used in CurvFold are not able to detect folds far from the camera and at the same time create false positives close to the camera. Since DeepFold is trained to predict folds from photometric input directly, it considers all aspects of the photometric information, including non-curvature information. Especially close to the camera, it is easy to visually distinguish false positives from true positives, but CurvFold does not directly have access to this information. This means that the advantage of DeepFold primarily comes from BlumNet itself and not steps later in the pipeline. However, CurvFold is still a useful method for fold detection, as it is more interpretable and does not require training. Moreover, as monocular depth estimation methods improve, the performance of CurvFold is expected to improve as well.

5.1 Limitations and Extensions

One limitation of CurvFold is the reliance on accurate depth estimation. Errors in the depth map can lead to inaccuracies in curvature computation and fold detection. Additionally, both CurvFold and DeepFold currently rely on fitting a cylinder to approximate the colon centerline as a straight line, which is sensitive to the angle of the camera relative to the colon surface. Both methods also sometimes fail to capture most of a fold when it is in several segments, and they can cluster separate folds into a single C_j. Finally, the difficulty of obtaining and accurately labeling ground-truth folds in clinical data makes it challenging to fully evaluate the performance of either method. As monocular depth estimation methods improve, the performances of both methods but especially CurvFold are expected to improve as well.

In addition to guidance back to a blind spot, individual trials have suggested that these methods of haustral fold detection could be used for the following two purposes. First, one could find a polyp to be removed during colonoscope withdrawal that was identified during colonoscope insertion. Second, in a later colonoscopic examination, one could identify a region of previous polyp removal or a decision to monitor a polyp. Finally, our methods could be extended to endoscopic examinations in other parts of the body with ridge-like features.

6 Conclusion

In this work, we have presented CurvFold and DeepFold, the first two methods ever for instance segmentation of haustral folds in colonoscopic video. Both use a novel definition of folds based on mean curvature, as well as a novel method for computing mean curvature from a depth map. They also use medial axis skeletons to separate fold regions, which is also novel in the endoscopic domain.

Experiments on both synthetic and clinical data demonstrate the efficacy of our method.

Ethics Statement. IRB review for use of the clinical colonoscopy video was waived under category 4, "secondary data/specimens", per 45 CFR 46.10. Consent was waived for the retrospective use of de-identified video.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Blum, H.: A Transformation for Extracting New Descriptors of Shape. In: Wathen-Dunn, W. (ed.) Models for the Perception of Speech and Visual Form. MIT Press, Cambridge, Massachusetts (1967)
2. Csurka, G., Larlus, D., Perronnin, F.: What is a good evaluation measure for semantic segmentation? In: Procedings of the British Machine Vision Conference 2013. pp. 32.1–32.11. British Machine Vision Association, Bristol (2013). https://doi.org/10.5244/C.27.32
3. Ehrenstein, S., McGill, S., Pizer, S., Rosenman, J.: Scribble-Supervised Semantic Segmentation for Haustral Fold Detection. In: 2023 Computer Aided Radiology and Surgery (CARS) Congress. Neuhausen-Nymphenburg, Munich, Germany (Jun 2023)
4. Foorginejad, A., Khalili, K.: Using homogeneous neighborhood in point clouds normal vector calculation. Modares Mechanical Engineering **14**(5), 155–163 (2014)
5. Golhar, M.V., Fretes, L.S.G., Ayers, L., Akshintala, V.S., Bobrow, T.L., Durr, N.J.: C3VDv2 – Colonoscopy 3D video dataset with enhanced realism (September 2025). https://doi.org/10.48550/arXiv.2506.24074
6. Hampshire, T., Roth, H.R., Helbren, E., Plumb, A., Boone, D., Slabaugh, G., Halligan, S., Hawkes, D.J.: Endoluminal surface registration for CT colonography using haustral fold matching. Medical Image Analysis **17**(8), 946–958 (December 2013). https://doi.org/10.1016/j.media.2013.04.006
7. Hellinger, E.: Neue Begründung der Theorie quadratischer Formen von unendlichvielen Veränderlichen. Journal für die reine und angewandte Mathematik **1909**(136), 210–271 (1909). https://doi.org/doi:10.1515/crll.1909.136.210
8. Johnson, S.C.: Hierarchical clustering schemes. Psychometrika **32**(3), 241–254 (1967)
9. Liu, C., Ke, W., Qin, F., Ye, Q.: Linear span network for object skeleton detection. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 133–148 (2018)
10. Mathew, S., Nadeem, S., Kaufman, A.: FoldIt: Haustral Folds Detection and Segmentation in Colonoscopy Videos (August 2021). https://doi.org/10.48550/arXiv.2106.12522
11. Mathew, S., Nadeem, S., Kumari, S., Kaufman, A.: Augmenting Colonoscopy using Extended and Directional CycleGAN for Lossy Image Translation. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4695–4704 (June 2020). https://doi.org/10.1109/CVPR42600.2020.00475

12. Meyer, A., Skudlik, P., Pauls, J.H., Stiller, C.: YOLinO: Generic Single Shot Polyline Detection in Real Time. In: 2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW). pp. 2916–2925. IEEE, Montreal, BC, Canada (October 2021). https://doi.org/10.1109/ICCVW54120.2021.00326
13. Meyer, A., Stiller, C.: YOLinO++: Single-Shot Estimation of Generic Polylines for Mapless Automated Diving (February 2024). https://doi.org/10.48550/arXiv.2402.00989
14. Müllner, D.: Modern hierarchical, agglomerative clustering algorithms (September 2011). https://doi.org/10.48550/arXiv.1109.2378
15. Nadeem, S., Marino, J., Gu, X., Kaufman, A.: Corresponding Supine and Prone Colon Visualization Using Eigenfunction Analysis and Fold Modeling. IEEE Transactions on Visualization and Computer Graphics **23**(1), 751–760 (January 2017). https://doi.org/10.1109/TVCG.2016.2598791
16. Nathan, S., Kansal, P.: SkeletonNet: Shape Pixel to Skeleton Pixel. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). pp. 1181–1185. IEEE, Long Beach, CA, USA (June 2019). https://doi.org/10.1109/CVPRW.2019.00156
17. Paruchuri, A., Ehrenstein, S., Wang, S., Fried, I., Pizer, S.M., Niethammer, M., Sengupta, R.: Leveraging Near-Field Lighting for Monocular Depth Estimation from Endoscopy Videos (Aug 2024)
18. Redmon, J., Farhadi, A.: YOLO9000: Better, Faster, Stronger. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 6517–6525. IEEE, Honolulu, HI (July 2017). https://doi.org/10.1109/CVPR.2017.690
19. Tang, X., Zheng, R., Wang, Y.: Distance and Edge Transform for Skeleton Extraction. In: 2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW). pp. 2136–2141. IEEE, Montreal, BC, Canada (October 2021). https://doi.org/10.1109/ICCVW54120.2021.00242
20. Wang, Y., Xu, Y., Tsogkas, S., Bai, X., Dickinson, S., Siddiqi, K.: DeepFlux for Skeletons in the Wild (November 2018)
21. Wu, D., Guo, Z., Li, A., Yu, C., Gao, C., Sang, N.: Conditional boundary loss for semantic segmentation. IEEE Transactions on Image Processing **32**, 3717–3731 (2023)
22. Zhang, T.Y., Suen, C.Y.: A fast parallel algorithm for thinning digital patterns. Commun. ACM **27**(3), 236–239 (March 1984). https://doi.org/10.1145/357994.358023
23. Zhang, Y., Frahm, J.M., Ehrenstein, S., McGill, S.K., Rosenman, J.G., Wang, S., Pizer, S.M.: ColDE: A Depth Estimation Framework for Colonoscopy Reconstruction (November 2021)
24. Zhang, Y., Sang, L., Grzegorzek, M., See, J., Yang, C.: BlumNet: Graph Component Detection for Object Skeleton Extraction. In: Proceedings of the 30th ACM International Conference on Multimedia. pp. 5527–5536. ACM, Lisboa Portugal (October 2022). https://doi.org/10.1145/3503161.3547816
25. Zhu, H., Barish, M.A., Pickhardt, P.J., Liang, Z.: Haustral Fold Segmentation with Curvature-Guided Level Set Evolution. IEEE transactions on bio-medical engineering **60**(2), 321–331 (February 2013). https://doi.org/10.1109/TBME.2012.2226242
26. Zhu, X., Su, W., Lu, L., Li, B., Wang, X., Dai, J.: Deformable DETR: Deformable Transformers for End-to-End Object Detection (March 2021). https://doi.org/10.48550/arXiv.2010.04159