Virtual unrolling and information recovery from scanned scrolled historical documents

Oksana Samko, Yu-Kun Lai, David Marshall and Paul L. Rosin

School of Computer Science & Informatics, Cardiff University, UK

Abstract

The objective of our work is to enable the reading of fragile scrolled historical parchments without the need to physically unravel them, thus providing valuable information to a wide range of scholarly disciplines. This problem has not been investigated by the computer vision community properly yet due to the need for parchment scanning technology: standard x-ray machinery is not sufficient as there is a requirement to extract out parchment ink in addition to the parchment's underlying structure. Effective data recovery is also compromised as content from historical scrolled documents is inaccessible due to the deterioration of the parchment. We create a 3D volumetric model of a scrolled parchment's underlying geometry and perform digital unwrapping of the parchment, producing a readable image of the text as an output. The proposed recovery framework consists of structure preserving anisotropic filtering in combination with robust segmentation, surface modelling and ink projection. We demonstrate with real examples how our algorithm is able to recover the underlying text and to solve the major challenge for scrolled parchment analysis, namely segmentation of connected layers and processing the data without user interaction.

Keywords: Parchment restoration, digital unwrapping, document processing, text retrieval, volumetric scanning

1. Introduction

Much of the history of the western world is written on parchment, a dry, treated, skin-derived writing medium [25]. The material was primarily designed as a writing medium that was smooth and flat; durability over millennia was probably not a prime consideration. Now, the information content of this complex medium is sometimes impossible to access without causing considerable damage or permanently altering the object to an unacceptable level. In some cases, their physical deterioration is at such an advanced state that any attempt to unravel the document manually would cause catastrophic fragmentation, destroying the internal information. Use of X-ray microtomography, a new direction in digital document analysis [18], provides a digital copy of a scrolled parchment as a 3D volumetric object, see Fig. 1. We utilise this 3D representation to recreate a virtual parchment model as input for a subsequent information recovery framework.

Digital document restoration has been an extremely active area of research in recent years [6, 8, 12, 15, 23, 28, 32, 35]. Current efforts have provided a new level of accessibility to many valuable literary works. However, not much attention has been paid to the analysis of scrolled

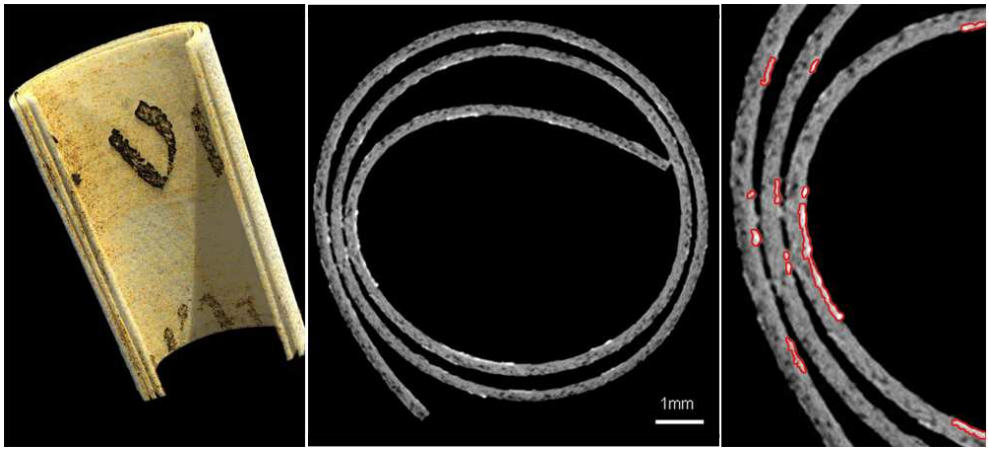

Figure 1: A small cut sample from a historical parchment scanned with the high definition XMT scanner. Left: volume rendered cutaway view with pseudo-colouring. Middle: tomographic slice with ink on the surface of the parchment (bright pixels). Right: close up of the slice with possible ink locations highlighted (red regions).

parchments. Traditionally document restoration approaches concentrate on regular photographic images and non-scrolled surfaces [20], which are easier to process.

Brown and Seales in [6] proposed a general de-skewing algorithm for arbitrary warped documents based on 3D shape. Doncescu et al. in [12] reported a similar method, where a laser projector is used to project a 2D light network on the document surface to capture 3D shape, and then 2D distortions of the surface are corrected with a two-pass mesh de-warping algorithm. Cao et al. in [8] presented an algorithm to rectify the warping of a bound document image: they built a general cylindrical model, and then used the skeleton of horizontal text lines in the image to estimate the model parameters. Pilu in [23] introduced a method for distorted document restoration which is based on physical modelling of paper deformation by an applicable surface. Yamashita et al. in [32] introduced a shape reconstruction method using a two-camera stereo vision system. Except for Cao's work [8] and a few others [34, 35], most of the current approaches require special setup (equipment, illumination) to assist in 3D shape discovery. Moreover, they can only handle smooth distortion of the image surfaces.

The most related work was undertaken by the EDUCE project [17], which attempted to read a scrolled document from a 3D scan. However, very few results on document unrolling have been reported [21, 26]. The results were only shown on small contrived samples and would not scale up to real parchment with many layers which are frequently compacted together. The segmentation stage of that work was performed semi-manually [21]. Apart from that, no other results on virtual parchment unrolling have been reported. X-ray scanning technology that is typically deployed for medical data analysis [7, 14, 33] does not meet a key requirement of our application: precise recovery of the ink from a parchment's weak boundaries.

The parchment shape – a tightly scrolled 3D object – makes its processing more challenging than the traditional document information recovery models. The separation of parchment layers is a major problem for parchment analysis. Parchment is essentially animal skin and has an irregular sponge-like structure; also its thickness may vary across a document surface. As a result of degradation over time, parchment may convert to its entropic form, gelatin, making the boundary between its layers difficult to observe even with the human eye. Image noise, low contrast and

scanning artifacts may lead to even more indistinct parchment structure boundaries. As can be seen in Fig. 2, it is difficult to handle the parchment segmentation task satisfactorily. A general algorithm can destroy damaged areas because of parchment's latent texture (oversegment), and fail to split tightly connected layers with zero gradient (undersegment) at the same time.

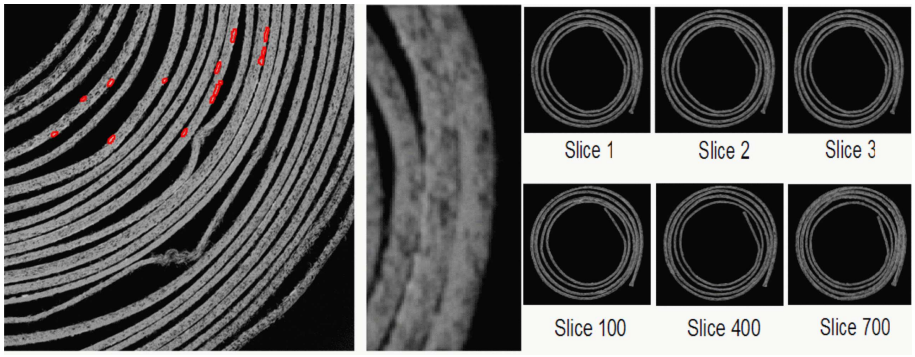

Figure 2: An example of parchment data. On the left is a section which contains ink on both (inner and outer) sides of the parchment; ink appearances are partially indicated by the red regions. A close up is shown in the middle demonstrating the weak boundaries between layers. On the right are shown several slices of the same scroll.

The shape of the parchment smoothly changes from slice to slice, but can differ significantly across the whole scroll. The parchment ink thickness is only a few voxels deep (represented by the light pixels close to the parchment boundary), thus it is very important to carefully process the boundary to avoid losing important information due to incorrect segmentation. Poor contrast between ink and the parchment itself makes this task even more difficult. Also often the ink remains inside the parchment layer but is lost from its surface due to natural decay of the parchment ink elements. Because of these parchment ink properties, traditional mapping techniques are inapplicable for parchment visualisation. Other challenging parchment characteristics are arbitrary wrapping shape, multiple page parchments, and parchments with writing on both sides. We handle the above challenges by building a flexible framework that can be controlled by several parameters. A scanned parchment contains hundreds or thousands of (volumetric scanner) slices, and a large proportion of the scrolled parchment contains writing, so ink is present in a significant proportion of slices. Fig. 3 demonstrates a photographic image of unrolled parchment and the result of our framework: recovered from the scanned scrolled parchment. We produce a virtually unrolled view of the text which is readable and comparable with its physically unrolled version. Such an image gives a clear representation of the parchment content and also can be used by palaeographers, for example, for the further analysis. Note that parchment scan quality is an important issue — developing effective parchment scanners remains an under-active research area.

The three horizontal bands visible in the right part of Fig. 3 are artifacts caused by variations in the SNR at different steps of the X-ray scanning process. As can be seen, the ink is still captured well, while the reconstructed parchment background is more affected. This effect does not worsen the text appearance as it can be seen in Fig. 3, only causing differences in the reconstructed background. Fig. 4 shows the above mentioned blocks on the small lengthwise cut of the original scan. Scanning artifacts may also include substantial image speckle and blurring, making the information recovery process more complicated.

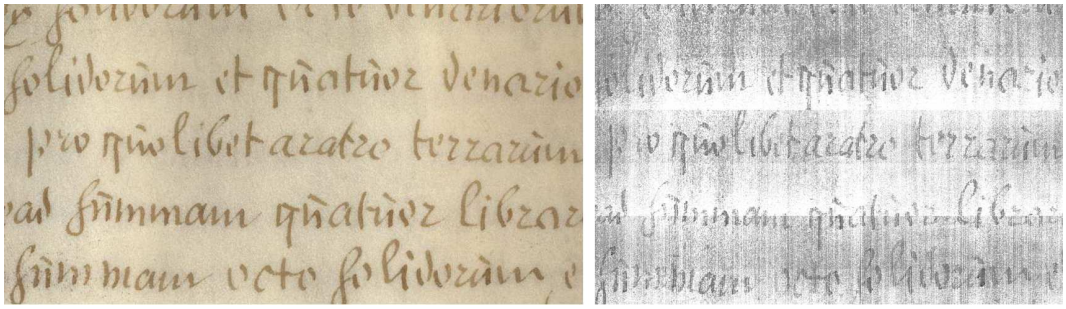

Figure 3: Left: example of unrolled photographed parchment. Right: reconstruction of its scanned scrolled version by the proposed framework.

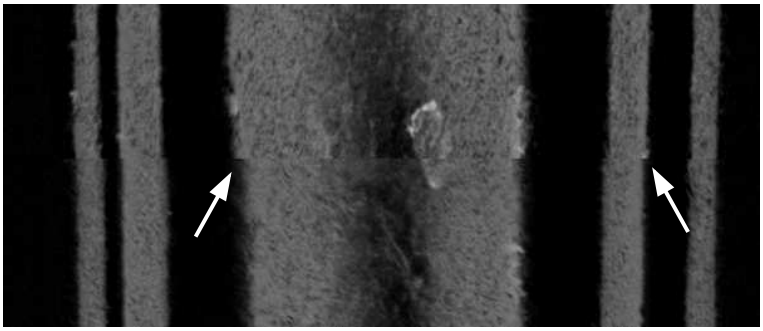

Figure 4: Block effect encountered during the X-ray scan, causing a horizontal discontinuity (highlighted by the arrows).

This paper extends our previous work on scrolled parchment segmentation [22] and describes a novel virtual parchment information recovery framework summarised in Fig. 5. We tested our framework on real historical parchment data.

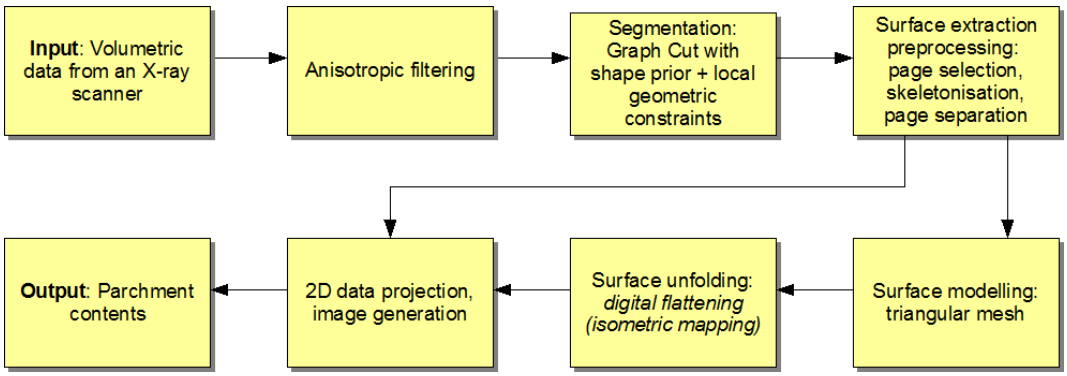

Figure 5: An overview of the virtual parchment unrolling framework

The first step, anisotropic filtering, makes the parchment structure more homogeneous, simultaneously preserving the parchment's layer boundaries. At the next step we introduce the

main segmentation routine, based on Graph Cut [3], in which a novel *shape prior optimisation* is included that incorporates parchment layer thickness information as a shape prior together with the traditional pixel intensity. This makes the segmentation more robust; however a few fused connections between layers may still remain. The reason for such incorrect connections is that the local boundary features may not exist, or it may be difficult to detect them reliably using the global optimisation. Therefore instead of involving time-consuming user interaction which also requires great accuracy from the user, we employ local geometric constraints to automatically separate such connections.

After the segmentation is complete and the parchment boundaries are identified, we can model parchment as a 3D object. If the parchment contains several pages, we consider the pages in turn. To analyse the ink appearance on a parchment's outer layers separately, for every page we find the parchment inner and outer regions by skeletonisation, and model them as individual objects. We then use the outer layers as an input for the further processing: first for the flattening modelling and then as the data source to map ink from the scrolled volumetric object to the plain surface. We define the shape of the parchment by representing the object's surface as a tetrahedral mesh, which we then unroll with minimum distortion isometric embedding. Finally we detect and project ink from the scrolled parchment to the unrolled surface to obtain the resulting image.

We consider related problems to parchment unrolling in more detail and evaluate our algorithm by applying it to five different parchment data sets, which vary in the parchment's properties, condition, size and number of layers. Our experiments demonstrate that our framework is able to successfully unroll the parchments and recover the reliable underlying text/ink information.

The remainder of this paper is organised as follows: Section 2 details the parchment filtering and segmentation stages. Section 3 addresses its surface modelling, followed by the ink projection stage in Section 4. Section 5 presents results, and Section 6 provides a summary and conclusion.

2. Parchment filtering and segmentation

We treat the data, volumetric images from a bespoke X-ray scanner, as a set of volumetric slices, as in Fig. 1. We initially filter this data in an attempt to complete missing portions of data. We then segment the scrolled parchment boundaries and ultimately unroll the parchment. Using the fact that the parchment structure is only changing slightly from slice to slice, we use the segmentation of a slice as the initialisation for the next slice in the set.

2.1. Data filtering

We use Coherence-Enhancing Diffusion filtering (CED) as a segmentation preprocessing step due to its property of completion of interrupted lines [29]. CED uses a nonlinear diffusion process whose diffusion tensor allows anisotropic smoothing by acting mainly along the preferred structure direction. This so-called coherence orientation is determined by the eigenvector of the structure tensor with the smallest eigenvalue [29]. Using the CED filter enables us to preserve the topology of the parchment layers, while the internal variation caused by the parchment's sponge-like structure is diminished.

A gray-scale image $p(x, y)$ can be treated as a surface corresponding to the mass concentration (the grey level). The equation describing the diffusion of the mass concentration is

$$d_t p = div(D \cdot \nabla p), \tag{1}$$

where ∇p is the concentration gradient, t is the diffusion time and D is the diffusion tensor – a function of local image structure. As in [29], we define D using the regularised structure tensor matrix $J_\rho(\nabla p_\tau)$:

$$J_\rho(\nabla p_\tau) = N_\rho * (\nabla p_\tau \otimes \nabla p_\tau), \tag{2}$$

where ρ is the integration scale, and p_τ is the regularised image of p obtained by convolution with a Gaussian $N_\tau(x, y) = (2\pi\tau^2)^{-1} \exp(-\frac{x^2+y^2}{2\tau^2})$. The eigenvectors of J_ρ give the preferred local orientations, and the corresponding eigenvalues denote the local contrast along these directions.

For a given parchment image $p(x, y)$, we have three parameters to define: diffusion time t, local scale τ and integration scale ρ. We set $\tau = 1.5$ for our framework; this small value gives us a uniform blurring over the whole object. The integration scale ρ reflects the characteristic size of the texture and is defined individually for each parchment. Correctly adjusted, it plays an important role in "reconstructing" a parchment's damaged areas, whilst preserving boundaries. Small ρ (1,2) does not perform sufficient filtering as it does not produce the dominant coherence orientations and does not remove parchment boundary irregularities. For parchments with few layers there is proportionally more background (air) in the data volume, and we get good results with $\rho = 4$. Generally larger parchment scrolls with many layers require a bigger ρ value; but excessive ρ values may deform the parchment boundary into a shapeless mass. Finally, the diffusion time t is also defined for each parchment individually. Larger t values produce increased blurring. This parameter depends on the image resolution and parchment condition. If the image resolution is small, a large time scale may dissolve the parchment boundary. If the parchment condition is poor (damaged, many pores), large t is preferable. In our experiments we set $t = 4$ for the small parchment, and $t = 12$ for the damaged parchment. Fig. 6 shows an example of the original parchment image, and the results of applying CED with different parameters to it. Note that the effect of a large ρ is most significant in very damaged areas (e.g. the circled area), which contain boundaries that we also need to preserve. The optimal parameter setting leads to segmented results with minimal fragmentation and the structures better preserved.

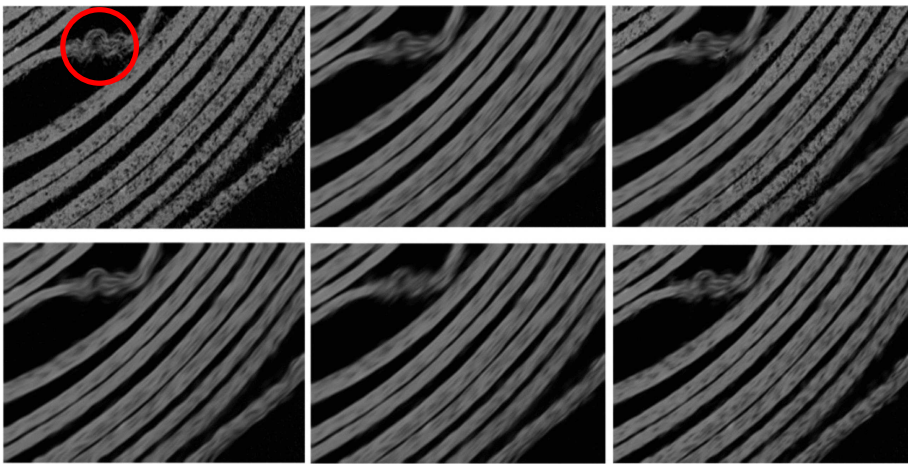

Figure 6: ¿From left to right: original parchment fragment; the CED filtered images with parameters (τ, ρ, t): (1.5, 10, 12) - optimal, (10, 10, 12) - nonuniform blurring, (1.5, 4, 12) - small ρ, (1.5, 20, 12) - large ρ, (1.5, 10, 4) - small t. The red circle highlights parchment damage.

2.2. Graph Cut with shape prior

We use a Graph Cut based optimisation for our main segmentation step. Boykov and Jolly [3] formulated Graph Cut segmentation as a binary labelling problem, i.e. each pixel in the image has to be assigned a label from the label set $\{0, 1\}$, where 0 and 1 stand for the background and the object, respectively. The labelling corresponding to the minimum energy is chosen as the solution.

The Graph Cut energy is formulated as a function of the pixel assignment:

$$E(f) = \sum_{p \in P} D_p(f_p) + \lambda \sum_{(p,q) \in N} V_{pq}(f_p, f_q) \tag{3}$$

Here P is the set of image pixels, f_p is the binary label assigned to pixel p, N is the set of neighbourhood pixel pairs, D_p is the data term (negative log likelihoods of the constructed background/foreground models [3]), V_{pq} is the smoothness (boundary) term for two neighbouring pixels, parameter $\lambda \geq 0$ specifies the relative importance between these two terms. To avoid user interaction, we initialise these models through GMM learning. We take $D_p(f_p)$ as the negative log likelihoods of the constructed background/foreground models. The smoothness term counts the weighted sum of discontinuities in f:

$$V_{pq}(f_p, f_q) = w_{pq} \varsigma(f_p, f_q) = \exp\left[-\frac{(I_p - I_q)^2}{2\sigma^2}\right] \frac{1}{dist(p,q)} \varsigma(f_p, f_q) \tag{4}$$

Here $\varsigma(f_p, f_q)$ is 0 if $f_p = f_q$ and 1 otherwise, w_{pq} is the weight, I_p is the intensity of pixel p, σ is the intensity variance, and $dist(p,q)$ is the Euclidean distance between two pixels. To minimise the energy from Eq. 3, Boykov and Jolly in [3] used the minimum cut on the constructed graph. Later in [4], Boykov and Kolmorogov presented an efficient algorithm for computing the minimum cut, which we apply in our framework.

Fig. 7 (left) shows the result of applying the standard Graph Cut to the parchment data from Fig. 1. It can be seen that after segmentation we still have many interlayer connections. Incorporating shape prior information, based on the parchment thickness, should make the segmentation more robust. This is performed as follows: We first estimate the average parchment thickness m. At the initialisation stage, we separate the image into background and foreground and apply a morphological dilation to the foreground to obtain the region U. This helps to fill in small holes. Neighbourhood pixel pairs in U are defined as N_u. The parchment thickness parameter can then be estimated as the mean distance between opposite boundaries of parchment layers (boundaries of U). For each pixel x from U, we further define the distance d_x from it to its closest boundary. Using these settings, we define our shape prior energy, and rewrite the energy function from Eq. 3 as:

$$E(f) = \sum_{p \in P} D_p(f_p) + \lambda \sum_{(p,q) \in N} w_{pq} \varsigma(f_p, f_q) + \mu \sum_{(p,q) \in N_u} s_{pq} \rho(f_p, f_q) \tag{5}$$

Here $\rho(f_p, f_q)$ is 0 if $f_p = f_q$ and 1 otherwise, μ is the shape parameter to control the relative importance of the shape term, s_{pq} is the shape weight. In our parameter settings we always bias intensity over shape due to the parchment structure and irregularity in thickness. The shape weight controls the layer thickness, we use the following form:

$$s_{pq} = \exp\left[-\frac{\left(-\frac{d_p + d_q}{2} - km\right)^2}{2\sigma_u^2}\right] \frac{1}{dist(p,q)} \tag{6}$$

where k is an estimate of the layer counter, which is the distance from the current pixel pair to the boundary in terms of the number of mean layer thicknesses, $k = 1 \dots M$, $M = max_{d_x} \lceil \frac{d_x}{m} \rceil$, σ_u is a parameter which we estimate as $\sigma_u \approx m + 1$. To define k, we calculate $\frac{d_p+d_q}{2}$. If $0 < \frac{d_p+d_q}{2} < \frac{3m}{2}$, then $k = 1$, and generally $\frac{(2k-1)m}{2} < \frac{d_p+d_q}{2} < \frac{(2k+1)m}{2}$ for $k = 2 \dots M$.

The energy based on our shape prior will be low for neighbouring pixels p, q which are close to the estimated parchment boundary and have different labels. Assuming that the shape prior energy is 0 outside U, and considering all the above, we can conclude that our shape term $S_{pq}(f_p, f_q) = s_{pq}\rho(f_p, f_q)$ satisfies the following property

$$S_{pq}(0, 0) + S_{pq}(1, 1) \leq S_{pq}(1, 0) + S_{pq}(0, 1) \tag{7}$$

and therefore according to [19] can be minimised using Graph Cut. Fig. 7 (right) illustrates how Graph Cut with our shape prior works. We get much less interlayer connections in comparison with the original Graph Cut, did not get any extra holes inside the parchment layers, and retain the ink at the surface, i.e. avoiding eroding the parchment surface.

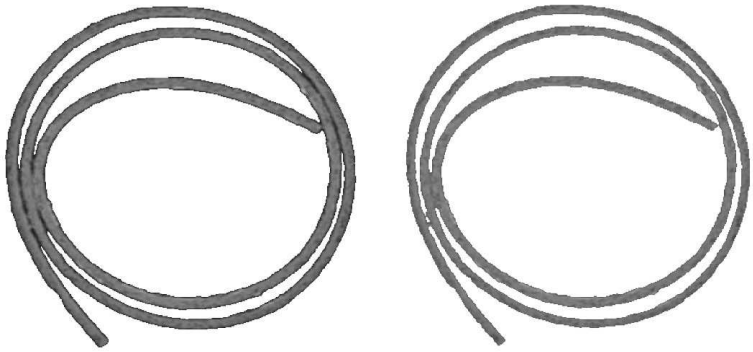

Figure 7: Segmentation using Graph Cut (left), and Graph Cut with shape prior (right). Incorporating the shape prior has reduced false connections between layers

2.3. Postprocessing

Our segmentation result in the right part of Fig. 7 indicates that there are still some false connections between the parchment layers. The problem appears in the areas which are fused together such that it is impossible to even see the boundary between them, as also demonstrated in Fig. 8. Many popular segmentation methods rely on user corrections during the segmentation process [16, 24], but this is time consuming, and they still require the presence of boundary features. We use a purely geometrical approach, based on the parchment's local features, to separate such areas.

The idea is to recreate a missing boundary from the preserved boundary of the opposite side of the same layer, otherwise from the closest preserved boundary. The basic stages of our postprocessing algorithm, illustrated in Fig. 9, are:

1. Detect a layer's false connections using existing boundary information. We extract the parchment boundary, and apply simple rules based on the patterns in a moving 7×7 window to detect and disconnect these connections.

Figure 8: Examples of segmentation containing false connections between parchment layers

2. Endpoint linking. For each detected endpoint, we find its pair and the opposite side of their layer. Between these endpoints we construct links parallel to the opposite parchment side, adjusting it if necessary to the local parchment thickness. Next we similarly reconstruct the boundary of the joined layer using the closest (reconstructed) boundary, maintaining the distance. We perform analysis on a layer by layer basis, starting from the outermost layer.
3. If necessary, adjust the obtained boundary to avoid connections with the previous slice.

It is not necessary to perform all these steps for all slices; since the parchment topology changes smoothly, we postprocess the first slice using the described algorithm, adjust the result for the second slice, and so on. Also note that depending on the parchment condition, postprocessing may not be required.

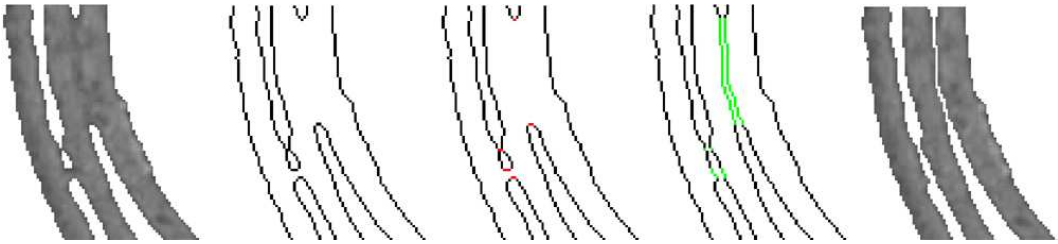

Figure 9: From left to right: segmented parchment fragment; its contour; detection of endpoints (red); construction of links (green); postprocessed fragment with false connections removed

3. Surface modelling

3.1. Surface analysis

After the parchment shape is defined and its layers are separated after the segmentation, we can analyse it further, taking into account the parchment content. A scrolled parchment may consist of several pages and may have ink content embeded within the scroll surface as well as on the surface.

If the parchment consists of several pages, it is straightforward to separate them using connected component labelling [11] since our segmentation was binary. An example is shown in Fig. 10.

A parchment may be written on both its sides, making it necessary to process each side individually. Also the parchment's ink may lie several voxels deep in the surface, and not necessarily

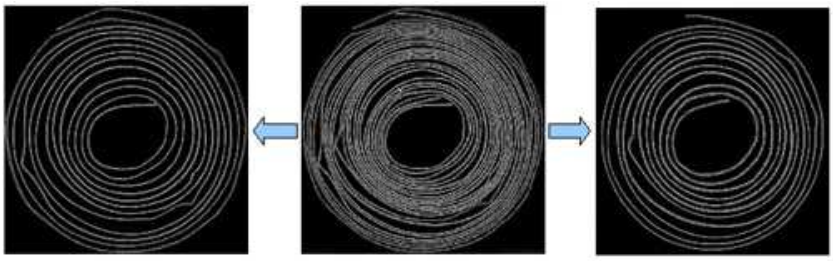

Figure 10: A scroll consisting of two pages separated into individual data sets

on the surface, because of the parchment structure or dust captured in the scan. Taking into account all the above, as well as geometrical and topological properties of the parchment, we aim to keep as much information as possible for each parchment side, not only the covering surface.

We divide every parchment page into two halves with skeletonisation by morphological thinning. This provides a simple and compact representation of the parchment shape. The morphological thinning is implemented via the hit-or-miss transform as a limited form of erosion. The parchment's structure may vary, and the obtained skeleton has many spurious components, or spurs, due to the parchment irregularity – the more irregularity, the more spurs. We fix the parchment's skeleton extreme end points, and remove unnecessary spurs by an iterative pruning operation until we get one connected component with exactly two end points. The pruned skeleton is used to separate the parchment into inner and outer parts, thereby assigning the ink to one of the parchment sides. Fig. 11 demonstrates our page division stages on a small parchment fragment.

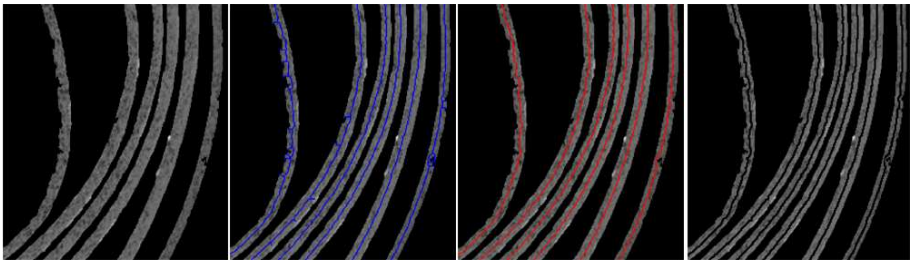

Figure 11: From left to right: parchment fragment; fragment overlaid with its skeleton (blue); fragment overlaid with the pruned skeleton with spurs removed (red); separated page

After the division, we separate the data again using connected component labelling similarly to page separation. Thus as a result of the surface analysis stage we have several voxels deep volumetric representations of each parchment side, which contain only the related ink and the background.

3.2. Tetrahedral surface meshing

Having the final data sets, *i.e.* volumetric parchment sides, we investigate its shape by approximating its surface using tetrahedral meshing. The process of mesh generation is divided into four stages. At the first stage, tetrahedral meshes with specified densities are extracted from

the input 3D image using a constrained Delaunay tetrahedralization (CDT) approach [27]. Next, a mesh repairing process is applied to the resulting surface to remove isolated vertices, duplicated elements and non-manifold vertices. At the third stage, we make a water tight surface by applying the regional exclusion algorithm [13]. This operation removes open edges and holes. Finally, low-pass filter methods [1] are applied for the surface mesh smoothing. These filters provide significantly improved volume conservation in comparison to the simple Laplacian operator [1].

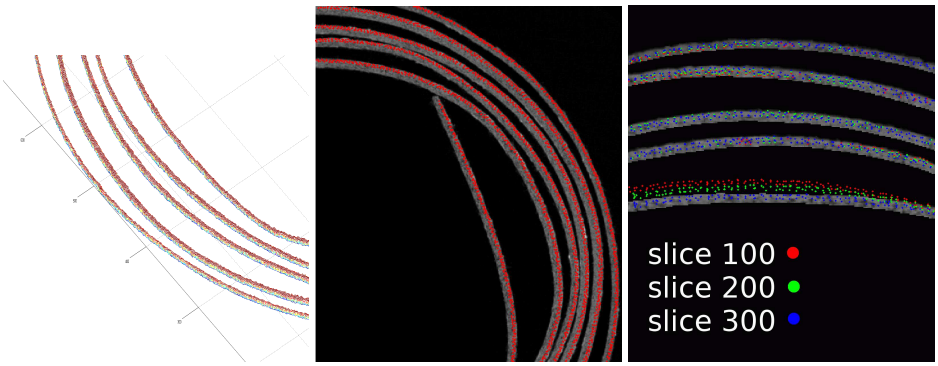

Figure 12: From left to right: fragment of a parchment slice mesh; fragment of a parchment slice with mesh nodes of its outside side marked in red; fragment of outside side of a parchment slice 300 with marked mesh nodes for slices 100 (red), 200 (green) and 300 (blue)

An example of a generated mesh is shown in right part of Fig. 12. Because the mesh computation procedure is quite demanding, we generate the mesh for one end (10 slices) only, and track it through the whole volume. This allows us to speed up the process of mesh generation and to have a mesh with a denser element distribution (Fig. 12, middle). After mesh generation, we track the mesh points over the volume relative to the parchment skeleton using correspondence estimation algorithm based on skeleton matching between the parchment slices. We can assume small inter-slice change and therefore it is sufficient to find correspondences between skeletons by finding the closest match. Given the initial mesh and skeletons for the parchment slices, we calculate the skeleton deformation from slice to slice, and interpolate mesh points to a new slice using the deformation values, as illustrated in the left part of Fig. 12. With our mesh moving strategy based on skeleton correspondences our parchment side from slice 300 is well represented by the deformed mesh (blue dots), initially constructed for the first 10 slices. The lower layer here clearly moves down, from slice 100 (red) to slice 300 (blue). There are small opposite movements, down-top, in the upper two layers; but mainly the parchment remains static with insignificant local variations.

3.3. Volumetric surface flattening

The parchment surface flattening can be interpreted as a mapping between 3D and 2D images, and the resulting 2D image matches what the parchment surface would look like if it was physically unrolled. The flattening mapping must be isometric, i.e. preserve distances, which minimises text distortion in the parchment. Taking into account that the parchment thickness is significantly small in comparison to the parchment length, we can say that our mapping is close to $(x\cos(x), y, x\sin(x)) \rightarrow (x, y)$ mapping, i.e. the swissroll mapping, demonstrated in the left part of Fig. 13.

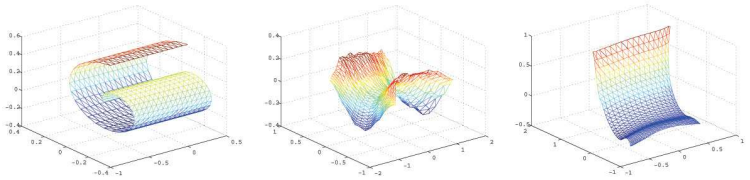

Figure 13: From left to right: the Swiss roll data set and its unfolding using classical MDS and MG-MDS

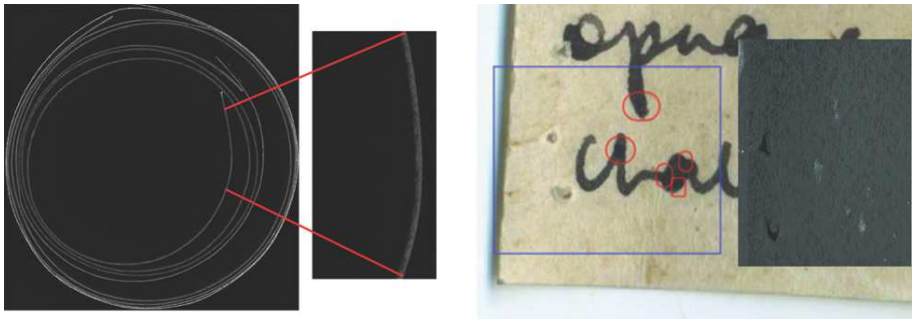

Figure 14: An example in which the ink is unreadable: slice of the scanned parchment with the experimental area; photo of the unrolled parchment, and reconstructed from the scan parchment surface with the remnants of the ink

The multigrid multidimensional scaling (MG-MDS) algorithm was proposed as the method for isometry-invariant matching of surfaces [5]. The key idea of this algorithm is computing the minimum-distortion mapping between two surfaces. The advantages of MG-MDS algorithm over classical MDS [2] are the use of SMACOF (Scaling by Majorizing a Convex Function) iterations as the relaxation procedure, and better recovery of the underlying geometry [5]. As a result, the MG-MDS outperforms the original MDS in the swissroll unrolling task, as shown in Fig. 13.

In our flattening stage we project a parchment's 3D coordinates into its 2D surface plane representation using MG-MDS. To calculate distances between the mesh points we use Dijkstra's algorithm over the mesh vertices. To correct for distortion (stretching), we scale the vertical coordinates, taking into account the parchment width.

4. Text revealing

It is only the iron in the ink that provides contrast in the X-ray images. The distribution of iron in the parchment ink is not uniform, therefore the X-ray scanned text looks faded and unclean, see Fig. 1, right. Depending on the ink content, it may be even impossible to see ink at all – if the iron content in there is low, see Fig. 14 for an example. Plus, there is a problem known as "iron gall ink corrosion": over time chemical processes, catalysed by ferrous ions, cause the slow deterioration of the ink; this however is dependent not only on the particular ink composition but also the storage conditions of the parchment [30].

Having flattened the parchment surface we now model the readable text on it, which we take from the scrolled parchment side obtained after our surface analysis stage. We are not interested

Example	Size (pixels)	Unrolled size	t	ρ	m	λ	μ	Density	Ink
Small	$430 \times 430 \times 708$	1450×708	4	4	12	0.45	0.3	0.09	0.52
Medium	$530 \times 530 \times 708$	4257×708	6	5	11	0.5	0.2	0.04	0.53
Large	$1702 \times 1732 \times 423$	$34560/38457 \times 423$	12	10	12	0.4	0.35	0.10	0.53
Norfolk	$774 \times 894 \times 4450$	9919×4450	6	9	11	0.48	0.3	0.07	0.59
1824	$934 \times 1124 \times 1900$	9771×1900	6	7	12	0.43	0.28	0.08	0.45

Table 1: Parchment dimensions (width $\times$ height $\times$ slices) and parameters used during processing for all examples in Section 5

in recovering the parchment background, and only project the brightest pixels, which correspond to the ink (i.e. the written content). This speeds up the process, but does not limit it: one can use the same procedure to recover the full data.

We project the ink into 2D space using barycentric coordinates [9]. These barycentric coordinates are used in the matching triangle in the flattened parchment surface to find the corresponding pixel coordinates. We keep all ink from the parchment surface, and also we recover ink from inside of the parchment. We analyse the parchment layer, and project the brightest pixels, giving priority to the pixels which lie close to the layer surface.

5. Experiments

We demonstrate the application of our framework to five real parchment examples, which range in size and complexity. Some parchments were created by ourselves so that we have absolute control and some baseline in the data. One example is an actual historical scroll from the Archive Centre at the Norfolk Record Office. It is difficult to obtain such scrolls for experimentation. The scrolls were scanned via high definition X-ray microtomography development developed by one of our project partners [10]. The sizes of parchments and the parameters used for their processing are given in Table 1.

The first parchment is shown in Fig. 1 and 7. As was revealed by the initial parchment examination, this data is a single page written on one side. It has fairly good condition, excluding a few areas where layers are stuck together. Fig. 15 illustrates the stages of applying our segmentation algorithm to this data. The segmentation results are quite similar for all slices of the same scroll, so our pictures represent the segmentation performance for the whole data set.

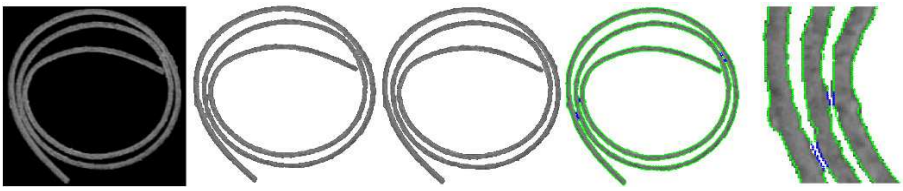

Figure 15: From left to right: original slice; segmentation by Graph Cut with the shape prior; final result after postprocessing to separate layers; postprocessed slice and contour of its neighbour slice (shown in green, with blue marks indicating postprocessing areas); close-up

Fig. 16 demonstrates comparison of the segmentation results with snakes [31], the original Graph Cut, and our algorithm. Since the original slice does not have tight connections our Graph

Cut with the shape prior was able to separate its layers without any postprocessing, while the original Graph Cut and snakes both failed. Although we give more penalty to the parchment boundaries, the result is not oversegmented because the ink is always considered as foreground. More numerical evaluation for the segmentation stage can be found in [22].

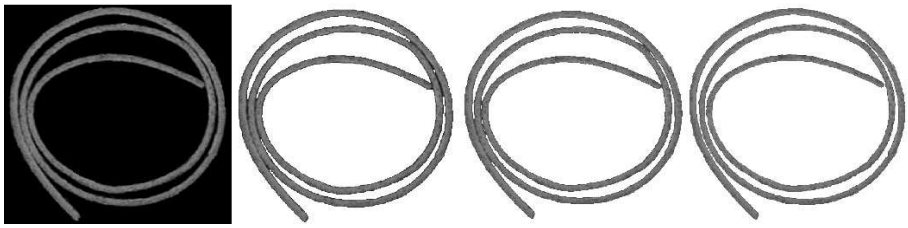

Figure 16: From left to right: original slice; segmented by snakes; segmented by Graph Cut; segmented by our method (Graph Cut with shape prior0

To model the parchment surface we construct the volumetric mesh with density (intensity) 0.09. This density allows us to keep all the necessary information about the parchment shape while losing insignificant small variations. Fig. 17 demonstrates the resulting image of recovered text.

The next parchment example is more complicated, more damaged and has more geometric variation. Again, this is a single page containing one side (inside) written text. Due to the variation in layer thickness we reduce our shape penalty to have less effect on very thick layers. Fig. 18 demonstrates our segmentation, and the result obtained with the original Graph Cut. With Graph Cut, we got undersegmentation (joined layers) simultaneously with oversegmentation (a large hole at the bottom left) even after the filtering. Our Graph Cut with shape prior was able to extract the parchment without holes, by giving them less penalty in comparison to the parchment boundary. Also we were able to separate most of the layers without postprocessing. In comparison to the previous example, this parchment needed more postprocessing to completely separate its layers.

Fig. 19 illustrates the unrolled parchment: a vertically cut strip of a document. There are visible letters on the parchment, however it is difficult to read them because the contrast is poor. Thus here we have a rough representation of the parchment's contains, which can be used by historians during palaeographic analysis, for example.

Now we consider a large scroll containing two interleaved parchments, shown in Fig. 20. Both parchments have writing on *both* sides, so we need to obtain four images in total for the whole data. Also with this example we demonstrate how our method can help process parchment with highly damaged layers. For the previous (less damaged) examples, we detected the boundaries of our initialised models and calculated the distances d_x. In this case we do not have a solid boundary in some areas, see the top middle of Fig. 20. The third layer there is very weak even after the filtering to detect its boundary with the initialisation. Therefore we "reconstruct" a boundary in the following manner: we dilate the image to fully include the weak layer, and define the obtained region as U. After that we assign the parchment boundaries as one pixel away from the obtained mask boundary, and calculate distances d_x to these "reconstructed" boundaries. As a result we get more joined layers which we separate using our shape prior, but also we save the weak boundaries. The right part of Fig. 20 shows a fully segmented slice of the scroll. Our Graph Cut with the shape prior was able to separate most of the data, we needed only minimal

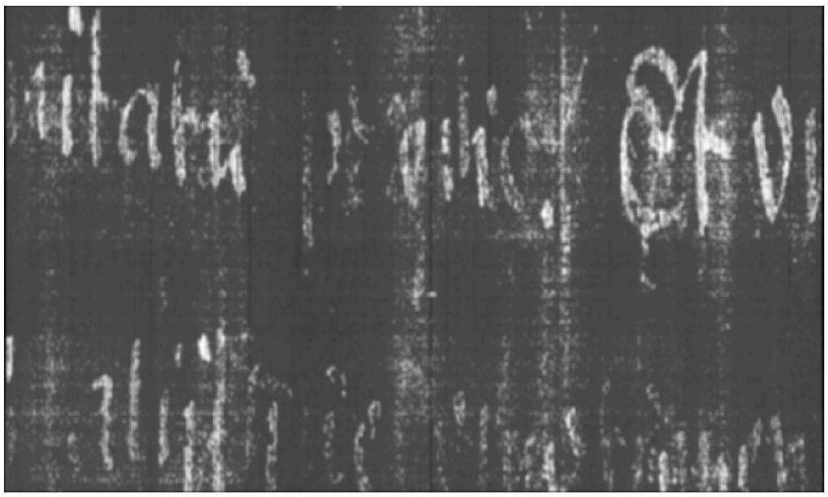

Figure 17: Small parchment: revealed text

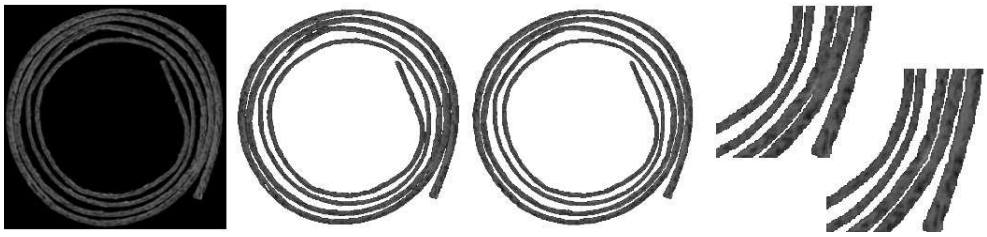

Figure 18: From left to right: original slice; segmentation by Graph Cut; segmentation by our method; a fragment before and after postprocessing

Figure 19: Medium scroll example: revealed text

(automatic) postprocessing for the areas with very tight and thin layers.

Fig. 21 demonstrates the result of our framework, and photographs of the manually unrolled scroll. The scroll consists of two pages, which are identical in their properties and were processed with the same parameters. The example of separated pages for this data is shown in Fig. 10. We divide the pages further into inside-outside parts, construct a non-dense mesh for a small cut of the each set and track it for the whole data. As stated in Table 1, length of the unrolled pages

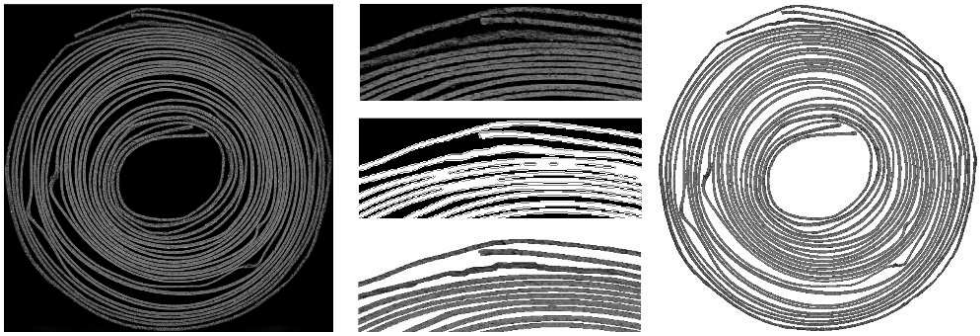

Figure 20: From left to right: original slice; fragments: original, thresholded with marked assigned boundary (gray), result; segmentation by our method

are 34560 and 38457 pixels, the second page is slightly shorter. From Fig. 21, the recovered text is recognisable on both pages, inside and outside, and closely matches the photographs of the original text. Note that the background intensity varies depending on its position in the scroll: the outer end of the scroll receives more radiation and is consequently lighter than the inner end.

The next example is an off-cut of parchment with hand written script using iron gall ink, provided by the Archive Centre at the Norfolk Record Office. This is a tightly wound roll of parchment, not openable due to its extreme dryness, and has a metallic foil strip glued to the inner edge of the parchment. The scanned scroll has 4450 slices, and is our longest scroll so far.

This parchment is in a relatively good condition, one (outer) side contains writing, with lots of confusing bright spots of metal dust on both sides of the parchment as demonstrated in Fig. 22. Another challenge is the metal strip twisted along the parchment outer side, which has the same intensity as the ink, and is very tightly connected to the parchment. Although we track the strip and remove it as a separate object in a similar way as we did for the two page scroll, there were areas impossible to separate without leaving parts of the strip on the parchment, see Fig. 23 for a detailed view. Fig. 24 shows the unrolled scroll.

Finally we demonstrate the parchment unrolling framework applied to a two hundred year old scroll dating back to 1824 (Fig. 25, left). The parchment was relatively well preserved without tight connections. The segmentation stage required little postprocessing. Fig. 25, right shows the parchment shape changes from slice to slice, and Fig. 26 shows the final result. This example was demonstrated in Fig. 4 and has three joint blocks, which appear through the parchment background.

6. Conclusion

We have presented a novel algorithm to virtually unroll and present the information from scrolled historical parchments that are completely inaccessible for manual reading. Technologies for the effective scanning of parchment are under active investigation, creating a demand for methods of analysing scanned scrolls. Our algorithm is the first attempt to automate virtual reading of a real historical parchment. The parchment information recovery framework consists of anisotropic filtering, segmentation with a shape prior and local geometric constraints, surface modelling and flattening, and text generation. The algorithm is able to reveal a scrolled parchment's content as long as its ink contains enough iron to be visible after the scanning procedure.

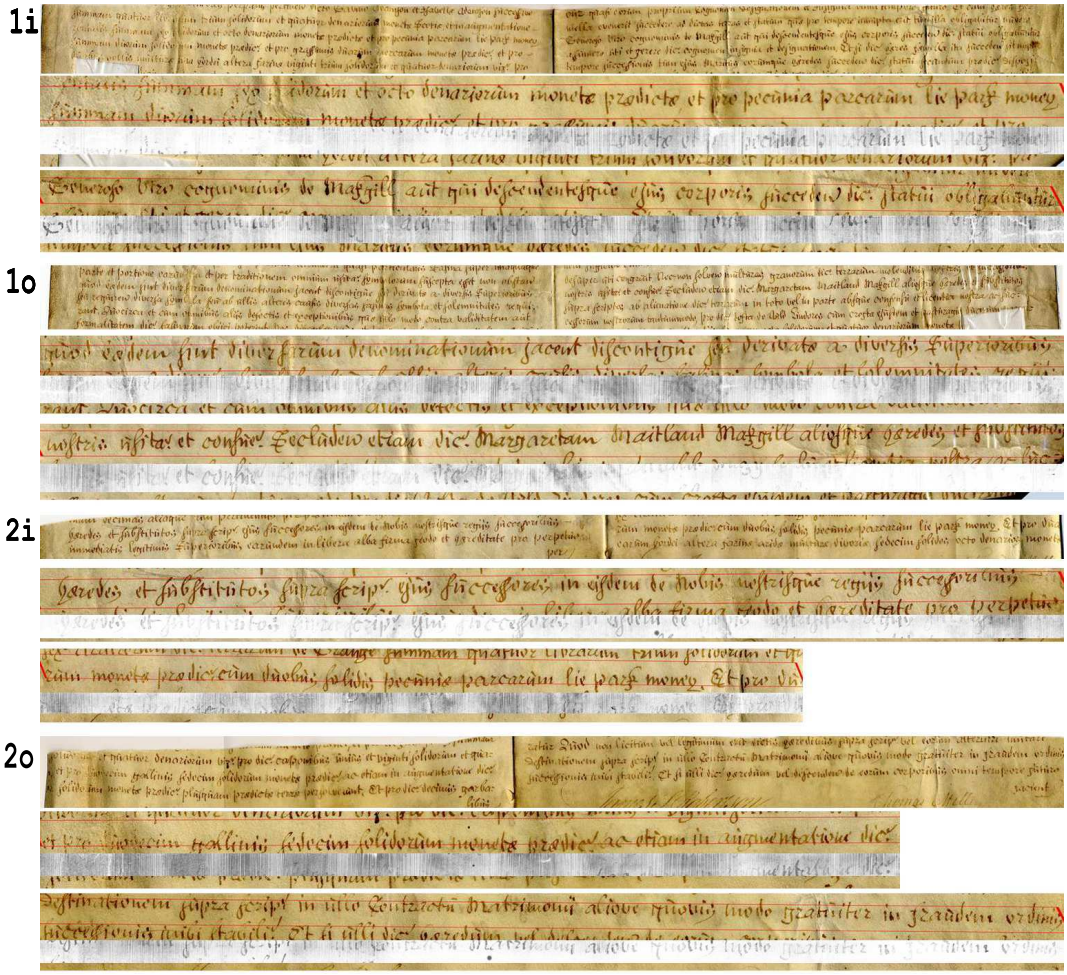

Figure 21: Unrolled large scroll: photograph of original and our result. (1i) - page 1, inside: general view and scaled processed areas (approximate boundary marked red) to compare with the result of our framework (grayscale strips; (1o) – same page outside; (2i) – page 2 inside; (2o) - page 2 outside.)

The presented method does not require user interaction and incorporates global parchment characteristics: thickness to separate the parchment layers and ink intensity to recover the hidden text. We illustrated the performance of our algorithm with five different real scrolls, and were able to recover information from parchments with very damaged areas and layers that were tightly stuck together.

References

[1] R. Bade, J. Haase, and B. Preim. Comparison of fundamental mesh smoothing algorithms for medical surface models. In *Proc. of Simulation and Visualization*, pages 289–304, 2006.

[2] I. Borg and P. Groenen. Modern multidimensional scaling theory and applications. *Springer*, 1997.

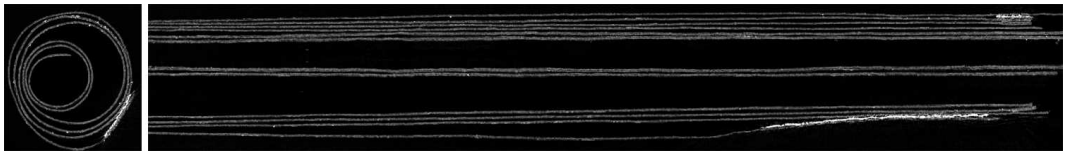

Figure 22: Norfolk scroll: vertical and horizontal cuts

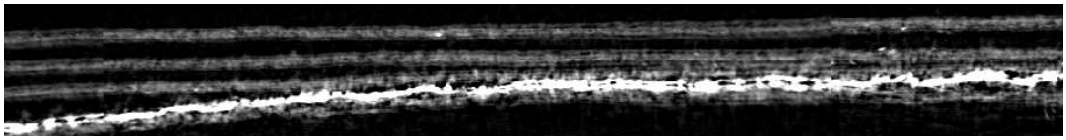

Figure 23: Norfolk scroll: horizontal cut fragment with the bright metal grip

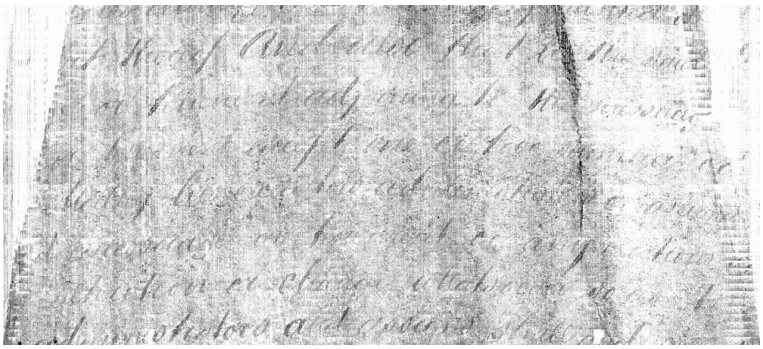

Figure 24: Norfolk scroll unrolled and text revealed

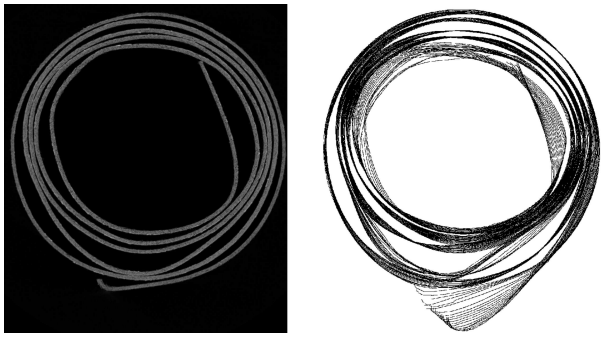

Figure 25: 1824 scroll: example slice and skeletons across slices

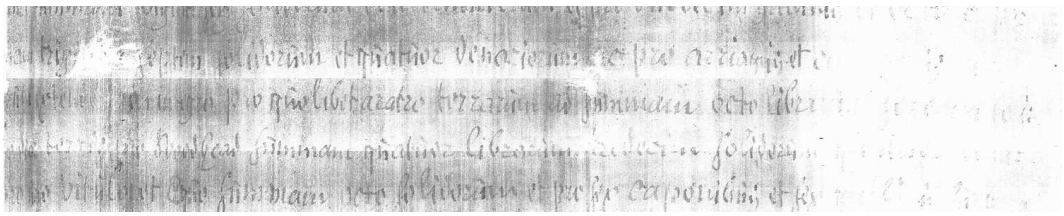

Figure 26: 1824 scroll: unrolled and text revealed

[3] Y. Boykov and M.P. Jolly. Interactive graph cuts for optimal boundary and region segmentation. *Proceedings of ICCV*, 1:105–112, 2001.

[4] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimisation in vision. 26(9):1124–1137, 2004.

[5] M. M. Bronstein, A. M. Bronstein, R. Kimmel, and I. Yavneh. Multigrid multidimensional scaling. *Numerical Linear Algebra with Applications (NLAA), Special issue on multigrid methods*, 13/2-3:149–171, 2006.

[6] M.S. Brown and W.B. Seales. Document restoration using 3D shape: a general deskewing algorithm for arbitrarily warped documents. *Proceedings of ICCV*, 2:367–374, 2001.

[7] P. Campadelli, E. Casiraghi, and G. Lombardi. Automatic liver segmentation from abdominal CT scans. In *Proceedings of the 14th International Conference on Image Analysis and Processing*, ICIAP '07, pages 731–736, 2007.

[8] H. Cao, X. Ding, and C. Liu. A cylindrical model to rectify the bound document image. *Proceedings of ICCV*, 2:228–233, 2003.

[9] H.S.M. Coxeter. Introduction to Geometry. *New York: Wiley*, 2:216–221, 1969.

[10] G.R. Davis and J.C. Elliott. High definition X-ray microtomography using a conventional impact X-ray source. *J. Phys. IV*, 104:131–134, 2003.

[11] M.B. Dillencourt, H. Samet, and M. Tamminen. A general approach to connected-component labeling for arbitrary image representations. *J. ACM*, 39(2):253–280, 1992.

[12] A. Doncescu, A. Bouju, and V. Quillet. Former books digital processing: image wrapping. *Proc. International Workshop Document Image Analysis*, pages 5–9, 1997.

[13] Q. Fang and D. Boas. Tetrahedral mesh generation from volumetric binary and gray-scale images. In *Proc. of IEEE International Symposium on Biomedical Imaging (ISBI 2009)*, pages 1142–1145, 2009.

[14] B. Fischl, D.H. Salat, E. Busa, M. Albert, M. Dieterich, C. Haselgrove, A. van der Kouwe, R. Killiany, D. Kennedy, S. Klaveness, A. Montillo, N. Markis, B. Rosen, and A.M. Dale. Whole brain segmentation: automated labeling of neuroanatomical structures in the human brain. *Neuron*, 33(3):341–355, 2002.

[15] B. Gatos, I. Pratikakis, and K. Ntirogiannis. Segmentation based recovery of arbitrarily warped document image. *Proceedings of ICDAR*, pages 989–993, 2007.

[16] L. Grady. Random walks for image segmentation. *PAMI*, 28(11):1768–1783, 2006.

[17] http://viscenter.com/educe. Enhanced digital unwrapping conservation exploration (EDUCE) project.

[18] C.J. Kennedy and T.J. Wess. The use of X-ray scattering to analyse parchment structure and degradation. *Physical Techniques in the Study of Art, Archaeology and Cultural Heritage*, (1):151–172, 2006.

[19] V. Kolmogorov and R. Zabih. What energy function can be minimised via graph cuts? *PAMI*, 26(2):147–159, 2004.

[20] J. Liang, D. Doermann, and H. Li. Camera-based analysis of text and documents: A survey. *Document Analysis and Recognition*, 7(2-3):84–104, 2005.

[21] Y. Lin and W.B. Seales. Opaque document imaging: building images of inaccessible texts. *Proc. ICCV*, 1:662–669, 2005.

[22] O.Samko, Y.Lau, A.D.Marshall, and P.L.Rosin. Segmentation of parchment scrolls for virtual unrolling. In *Proceedings of BMVC*, pages 37.1–37.11, 2011.

[23] M. Pilu. Undoing page curl distortion using applicable surfaces. *Proc. of CVPR*, 1:67–72, 2001.

[24] C. Rother, V. Kolmogorov, and A. Blake. Grabcut-interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics*, 23(3):309–314, 2004.

[25] R. Reed. The nature and making of parchment. *Leeds: Elmete Press*, 1975.

[26] W.B. Seales and Y. Lin. Digital restoration using volumetric scanning. *Proc. joint ACM/IEEE Conference on Digital Libraries*, pages 117–124, 2004.

[27] J.R. Shewchuk. Constrained delaunay tetrahedralizations and provably good boundary recovery. In *Proc. of*

Eleventh International Meshing Roundtable, pages 193–204, 2002.

[28] A. Ulges, C.H. Lampert, and T.M. Breuel. Document capture using stereo vision. *Proceedings of ACM Symposium on Document Engineering*, pages 198–200, 2004.

[29] J. Weickert. Coherence-enhancing diffusion filtering. *International Journal of Computer Vision*, 31:111–127, 1999.

[30] J. Wouters and G. Banik. Inks from the middle ages: Old recipes, modern analysis and future decay. *Les Chroniques De Hainaut, Ou Les Ambitions D'Un Prince Bouguigon*, pages 141–148, 2000.

[31] C. Xu and J.L. Prince. Snakes, shapes, and gradient vector flow. *IEEE Transactions on Image Processing*, 7(3):359–369, 1998.

[32] A. Yamashita, A. Kawarago, T. Kaneko, and K.T. Miura. Shape reconstruction and image restoration for non-flat surfaces of documents with a stereo vision system. *Proceedings of ICPR*, pages 482–485, 2004.

[33] P.A. Yushkevich, J. Piven, C.H. Cody Hazlett, C. R. Gimpel Smith, C.S. Ho, J.C. Gee, and G. Gerig B. User-guided 3D active contour segmentation of anatomical structures: Significantly improved efficiency and reliability. *NeuroImage*, 31:1116–1128, 2006.

[34] Z. Zhang and C.L. Tan. Correcting document image warping based on regression of curved text lines. *Proceedings of ICDAR*, pages 589–593, 2003.

[35] Z. Zhang, C.L. Tan, and L.Y. Fan. Restoration of curved document images through 3d shape modeling. *Proceedings of CVPR*, 1:10–16, 2003.