Staff Detection and Rectification of Single Image Musical Documents

Daniel Maier

***Abstract*—Common Western Music Notation (CWMN) is governed by intricate and complex rules, utilizing visual symbols and their relational positions on a musical staff to convey essential elements of a piece such as pitch, rhythm and dynamics. Optical Music Recognition (OMR) is the process of teaching a computer how to interpret these rules and to transform sheet music into a machine-readable format.**

Numerous OMR systems presume pristine, straight images or computer-generated musical documents, yet in practice, input images commonly consist of photographs of book pages or standalone sheets, owing to the widespread use of smartphones and digital cameras. The introduced distortions due to page curl and projective foreshortenings complicate the detection of staff lines and musical symbols, both of which remain ongoing challenges in the OMR process.

In this work, I propose a novel approach that leverages domain knowledge to extract staffs with sub-pixel precision from curved documents. The curve shape of the staffs are then combined with other data to fully rectify the musical document. This enables the application of methods typically used for computer-generated musical documents to be extended to more generally acquired input images, thus enhancing the usability of OMR systems. The proposed method operates on single images without requiring any a-priori knowledge about the camera, page shape, or the content of the musical document.

I. Introduction

The rectification of text-based documents is a well-studied area with numerous existing methods. However, in the context of musical documents, the unique structure of such documents and staff lines in particular can be used to derive a more accurate and specialized rectification process. To the best of my knowledge, there is currently no method that effectively utilizes the specific characteristics of musical documents for this purpose.

Indeed, the importance of relational positions in the notation of musical documents highlights the significance of a rectification pre-processing step. For instance, the pitch of a note is determined primarily by its position on the staff lines, encompassing both on and in-between staff line locations. Additionally, the direction along the staff lines serves as a temporal direction, where the relative distances of notes in this direction encode whether they form chords, appear on the same beat (even across staffs) or follow each other, ultimately influencing the structure of individual musical voices. In this way, an object's position acts as a very important feature that can dictate its attributes and the evolution of voices and time. For algorithms aiming to group notes into chords and separate them into distinct voices, precise positional information is crucial. If the positions are distorted or imprecise, it can lead to erroneous interpretations of the musical content. For that reason, it is essential to rectify the distortions of the input images to a degree where positions of musical objects are aligned as closely as possible to those found in computer-generated documents.

Deep learning models have exhibited a notable degree of resilience to image distortions in the realm of object detection. They also have been used to detect and separate pixels belonging to staff lines from musical notation. While these models usually provide a high detailed segmentation map, encompassing all staff-related pixels, they can not provide a functional description of individual staffs which form the baselines for any rectification process. Conversely, I show that obtaining these pixels can be done in a much simpler, algorithmic manner using local gradients, eliminating the need for complicated models and training data. I further show how these points can be clustered and further processed in a robust way to ultimately derive shape models for individual staffs. These can be used to rectify the image, ensuring that shape and proportions of objects stay consistent over input images. Due to the inherent parallelism and uniform spacing of staff lines, this information can also be leveraged to standardize the scale across all images once the staffs have been recognized. It is worth noting that modern state-of-the-art object detectors, based on deep learning, heavily lean on Convolutional Neural Networks (CNNs), anchor boxes, and the partitioning of images into grids. These elements require fine-tuning to accommodate objects of varying sizes and dictate the density with which objects can be detected within the image. Musical notation poses a unique and substantial challenge in this regard, given the abundance of numerous small objects that are in close proximity or overlap, a scenario in which current state-of-the-art detectors face particular difficulties. By using this method to normalize images and establish a consistent scale as an initial step, it opens the door to designing object detectors that are highly specialized for addressing these specific challenges, possibly leading to elevated performance levels.

This work is divided into two main parts. The first part focuses on the extraction and accurate curve fitting of the staffs present in the image. These curves will serve as baselines for the subsequent rectification process, which is thoroughly explained in the second part.

II. Staff detection

Staff lines play a very important role in CWMN. They act as underlying baselines relative to which all other objects are defined. For that reason, the accurate detection and tracking of

staff lines is usually among the first steps in any OMR system [1]. It is essential for segmenting the music notation into individual systems or tracks and makes subsequent interpretation of musical symbols possible. In addition to that, OMR systems often rely on accurate staff line detection to separate and remove staff lines from the overlaid musical symbols on a binarized image. This makes it possible to isolate symbols and aid in their classification. For that reason, staff detection is usually considered together with their removal, although some authors proposed methods that work without the need to remove staff lines [1, 2].

When given in computer generated documents, staff lines are parallel lines that have uniform thickness and spacing. For printed musical documents captured on camera, however, the staff detection process becomes much more complicated due to the following factors:

- Curvature: Page curl, warping, wrinkles, and perspective camera projection transform staff lines into general curves. Modelling these necessitates a considerably more complex representation, making the detection process susceptibility to noise. This also causes staff lines to only be parallel at a local scale rather than globally.
- Interruption: Due to noise and sampling artifacts, staff lines do not necessarily have to be continuous lines anymore. This is especially true if the image gets binarized first.
- Superimposed objects: Individual staff lines can be occluded and crossed by various symbols, including other line-like objects like beams and ties. These objects can occlude staff lines for an extensive distance or cross them with a similar angle. Separating staff from symbol pixels is often not possible based on local criteria alone, and requires the incorporation of a deeper knowledge about these objects and their relationships.
- Illumination changes: When photographing curled pages, the change of surface normals across the page may cause illumination to change substantially across the image. This can cause the contrast between background and foreground pixels to vary widely across the image.

In the literature, a wide variety of methods exist that aim to overcome these problems. I will give a quick overview of them in the following chapter.

A. Previous work

The detection process usually starts by binarizing the image [1]. This makes it possible to use certain line detection algorithms, morphological operations or dynamic programming. Staff detection is a straight forward problem for computer generated musical documents. In this case staffs are straight and horizontal and can be found simply by finding maxima in the horizontal projection of black pixels [3, 4]. This approach can also be extended to account for a global rotation angle of the image. Because all staff lines are parallel to each other in this case, simple line detection methods like the Hough transform [5] can be used to identify this global rotation angle together with the staff lines [6].

When dealing with curved documents, however, more sophisticated algorithms have to be employed. One important starting point is Run-length coding (RLE), a representation that stores white and black runs of a binarized image together with their counts. This representation can be used to recover the staff line thickness `slt` and interline staff space `iss` from the image. These are parameters that are often used to describe the characteristic scale of staffs, as illustrated in Figure 1. When staff lines are assumed to be more or less horizontal, these parameters can be extracted from a column-major RLE of the image as the most frequent black and white runs respectively [3]. Most staff detection algorithms rely on these two parameters.

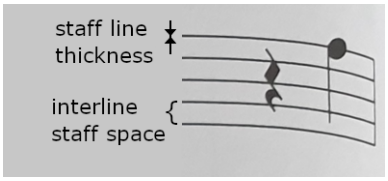


Fig. 1: Staff together with its characteristic staff line thickness and interline staff space.

A common procedure in many approaches is to first identify line segments that probably belong to staffs, and then to merge these staff segments successively into coarser and coarser objects until the shape of the full curve is established. On a binarized image, staff segments can be obtained by discarding black runs whose vertical length is significantly higher than the found staff line thickness [7]. The merging is then usually done using handcrafted heuristics that merge segments based on relative position or angle [8, 9, 1]. Su et al. assume that all staffs share a common staff line shape, which they estimate by analyzing the local pixel orientations of staff segments [7]. At each column of the image, the global staff shape is determined by averaging the orientations of individual staff segments at that column. This makes the approach more stable, but the assumption that all staffs share a common curve is hardly ever true. While they demonstrate the stability of their method on digitally distorted images, it is not applicable to images captured from real printed documents by a camera. In this case, the projection and page shape causes very different staff shapes across the image, as can be seen by looking at Figure 4.

Generally, all of these methods suffer from not incorporating global information about the staffs into the detection process and instead try to track staffs based on local criteria. As a result, none of these methods is applicable to *general* curved musical documents.

Another approach uses the fact that staffs are usually the only horizontal lines that span across the whole document [10]. In this method, staffs are considered as the shortest connected paths between the left and right image margins. They use a scoring system that punishes the travelled distance between active pixels, and minimize this cost by using dynamic programming. Starting from the highest score on the left side of

the image, the shortest path can then be traced back through the active pixels. Although this approach constructs staff lines by means of a global optimization process, it is not very stable in general because it still relies on tracking the staff line locally. In fact, a simple interruption of a staff line is enough to end its path. Given that the image is binarized to make this approach even possible, staff lines are more likely than not to have interruptions. Furthermore, overlapping musical symbols can cause the path to wander between adjacent staff lines. These tracking errors are frequent, especially under the presence of curvature, where the staff line may in fact not be the shortest path. I illustrate this in figure 2. For that reason, this approach requires a post-processing step in which staff lines are uncrossed, organized and trimmed. Experimenting with this approach revealed that it is difficult to guarantee a successful staff line tracking across a wide range of musical documents using this method.

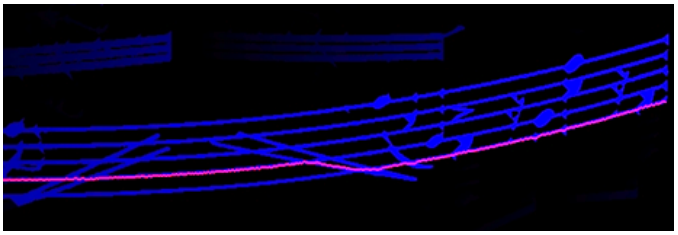


Fig. 2: Visualization of a traceback error using dynamic programming. Blue represents the intensity of the scoring function, and the wrongly traced back path is shown in pink. It can be seen that the shortest path is indeed not the staff line.

B. Proposed method

None of the described methods are versatile and stable enough to work for general curved musical documents, such as those obtained through photographing real printed pages. They also all rely on binarization which degrades the quality of the input image substantially and destroys a lot of information in the process. In contrast to the methods discussed before, I propose a novel staff line detection method that operates on gray valued images. Not only does this preserve the original image for later stages, it also provides the opportunity to make use of gradient information when tracking the staffs. In a first step, I use established methods from the field of computer vision to retrieve points of curvilinear structures based on their local neighborhood. Then, these points get separated into individual staffs using a custom clustering method. Lastly, I perform a custom curve fitting algorithm that heavily incorporates domain knowledge about staffs to obtain an accurate curve model with sub-pixel precision for each staff. By shifting the focus from individual staff lines to broader criteria, this method becomes very stable and applicable to general curved musical documents and removes restrictions on page shapes and camera poses.
The method can be broken down into four major steps that are explained in the following chapers.

1) Parameter acquisition: Musical notation relies on the contrast between black symbols and the white background to convey information. For that reason, input images are usually considered in grayscale format, eliminating the need to consider irrelevant color information.

$$f(x, y) : \mathbb{R}^2 \mapsto \mathbb{R} \tag{1}$$

Due to this bimodal nature, the image is usually segmented into two classes, where only the black foreground pixels are relevant to OMR. While I do not apply binarization to the image, segmenting the image into foreground and background is still necessary to obtain crucial parameters about scale and contrast of the image.
Global illumination and contrast can vary widely from image to image, making the choice of a threshold to separate the image non-trivial. A popular choice would be Otsu's method [11] where a global threshold is calculated by maximizing the inter-class variance of the image intensity histogram. This method works well, but due to the threshold being global, differently lit regions can potentially cause problems. For that reason, I use an adaptive thresholding approach instead, where the threshold is calculated based on the intensity of a local neighborhood. This is done by blurring the image with a Gaussian kernel K_σ. I consider every pixel a background pixel if its intensity is smaller than its smoothed surroundings minus an empirically determined constant C_s

$$f_{ij} \in \begin{cases} \text{foreground} & if\ f_{ij} > (K_\sigma * f)_{ij} - C_s \\ \text{background} & else \end{cases} \tag{2}$$

With that, a globally averaged contrast h can be obtained by subtracting the average intensity of the background from the average intensity of the foreground pixels.
Similar to the method described in [3], I create a column-major RLE that captures information about the scale of staffs in particular. The most frequent run of foreground pixels yields the `slt`. Given the highest peak is situated at x and has value β, its predecessor with value α and successor with value γ, I use quadratic interpolation to get a refined value for `slt`

$$\texttt{slt} = x_f + \frac{\alpha_f - \gamma_f}{2\alpha_f - 4\beta_f + 2\gamma_f} \tag{3}$$

Likewise, the most frequent run of background pixels yields the space *between* staff lines, so `iss` is given as

$$\texttt{iss} = \texttt{slt} + x_b + \frac{\alpha_b - \gamma_b}{2\alpha_b - 4\beta_b + 2\gamma_b} \tag{4}$$

This approach estimates `slt`, `iss` and h for the whole image and therefore assumes them to be constant across it. While `slt` is usually in the orders of a few pixels and the constancy assumption therefore well fulfilled, the same can not be said necessarily about the other two parameters. Assuming `iss` to be constant over the image is reasonable, as long as the image is captured without a significant angle and sufficient distance. While these assumptions have proven their worth in practise, a generalization is also possible and discussed in Chapter IV.

Note that while I use adaptive thresholding to assess global image parameters, it is not used to binarize or alter the image itself in any capacity.

2) Line point detection: The basis for my method is formed by the detection of individual line points, proposed by Steger [12]. Given a grayscale image, he models a bright line on a dark background with width $2w$ and intensity h as having a parabolic profile. He then showed that under these assumptions, the first derivative has a zero crossing exactly at the middle of the line profile. He concludes that potential line points can be found by checking whether the first derivative has a zero crossing within the pixel or not. Similarly, the second derivative takes on its maximum negative value at that point. The profile together with its derivatives are shown in Figure 3. In my case, the line width is given as $2w = \texttt{slt}$.
The direction perpendicular to a potential line profile is the direction in which the second directional derivative becomes maximal in absolute terms. This direction can be determined by convolving the image with the following kernels to calculate its partial derivatives:

$$r_x = (g'_\sigma(x)g_\sigma(y)) * f \tag{5a}$$
$$r_y = (g_\sigma(x)g'_\sigma(y)) * f \tag{5b}$$
$$r_{xx} = (g''_\sigma(x)g_\sigma(y)) * f \tag{5c}$$
$$r_{yy} = (g_\sigma(x)g''_\sigma(y)) * f \tag{5d}$$
$$r_{xy} = (g'_\sigma(x)g'_\sigma(y)) * f \tag{5e}$$

where $g_\sigma(x)$ is the Gaussian smoothing kernel

$$g_\sigma(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp -\frac{x^2}{2\sigma^2} \tag{6}$$

and σ is chosen to be greater than $\frac{w}{\sqrt{3}}$ [12]. The dominant direction can then be found by performing eigenvalue decomposition on the Hessian matrix:

$$H(x,y) = \begin{pmatrix} r_{xx} & r_{xy} \\ r_{xy} & r_{yy} \end{pmatrix} = \begin{pmatrix} e_1 & e_2 \end{pmatrix} \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix} \begin{pmatrix} e_1 \\ e_2 \end{pmatrix} \tag{7}$$

With $|\lambda_1| \geq |\lambda_2| \geq 0$, the eigenvalue λ_1 indicates the variation in the dominant direction and the corresponding normalized eigenvector e_1 points normal to a potential line.

$$e_1 = \begin{pmatrix} n_x \\ n_y \end{pmatrix} \tag{8}$$

The zero crossing criterion from earlier is a necessary condition for being a line point. This criterion can be evaluated by approximating the first derivative in the direction of e_1 by a second order Taylor approximation. The coordinates of the zero crossing of this approximation are given by:

$$\begin{pmatrix} p_x \\ p_y \end{pmatrix} = \begin{pmatrix} tn_x \\ tn_y \end{pmatrix} \mid t = -\frac{r_x n_x + r_y n_y}{r_{xx} n_x^2 + 2r_{xy} n_x n_y + r_{yy} n_y^2} \tag{9}$$

Each line point must then satisfy the necessary criterion:

$$-\frac{1}{2} \leq p_x \leq \frac{1}{2}, -\frac{1}{2} \leq p_y \leq \frac{1}{2} \tag{10}$$

In addition to that, λ_1 is a measure for the curvature of the directional derivative. The stronger the contrast of the line, the higher this value. Therefore, any pixel who fulfills the necessary condition and whose λ_1 is greater than some threshold $T_p(\sigma, w, h)$ is selected as line point. A recommendation for the choice of T_p can be found in the original paper [12]. Each line point is saved with its sub-pixel coordinates p_x, p_y, a direction tangential to the line and the confidence value $\lambda_1 > T_p$. I differentiate points into horizontal and vertical based on the preferred direction of their gradient. The resulting line points are shown in Figure 4.

3) Clustering: Steger suggests merging individual line points to curves by merging adjacent pixels based on the similarity of their angles and their proximity [12]. As already explained in Chapter II-A, such a line tracking approach is insufficient to track individual staff lines reliably. Instead, I suggest using a suitable clustering method to separate the previously obtained line points into individual staffs. This step is not required to find the exact curve model of the staffs, but rather to find the number of staffs in the image, separate the line points accordingly and remove outliers.
While there are many clustering algorithms to choose from, a good clustering algorithm should respect the following criteria:

- As the number of staffs is generally unknown at that point, the clustering algorithm must work with an unknown number of clusters.
- Staffs coincide with the locations of maximal density of line points in the position-angle space. The clustering method therefore should be sensitive to drops in local density in y-direction.
- If we assume the page to be sufficiently smooth, the height and angle of a staff can be modelled as continuous functions of x. As a result, at any given point on the staff, only one specific height and angle exist for the corresponding value of x.

One suitable algorithm would be *Mean-Shift clustering*, which is a density based method that works with sliding-windows [13]. In each iteration, the mean of all data points in a region is calculated and then assigns that mean as the new center of the region. This process is repeated until the number of data points in the region stops increasing. At the end, near-duplicate clusters can be eliminated by checking if they overlap. In the 2D case, clusters can be initialized in a grid pattern which would then be shifted onto the denser staff curves. Because staffs are no centroids, these converged clusters along a curve would then need to be merged into individual staffs in a post-processing step. This would make the algorithm effectively a *line tracking* algorithm that tracks the density of a staff across the page instead of individual lines (making it much more stable). While definitely possible, this approach comes with some drawback:

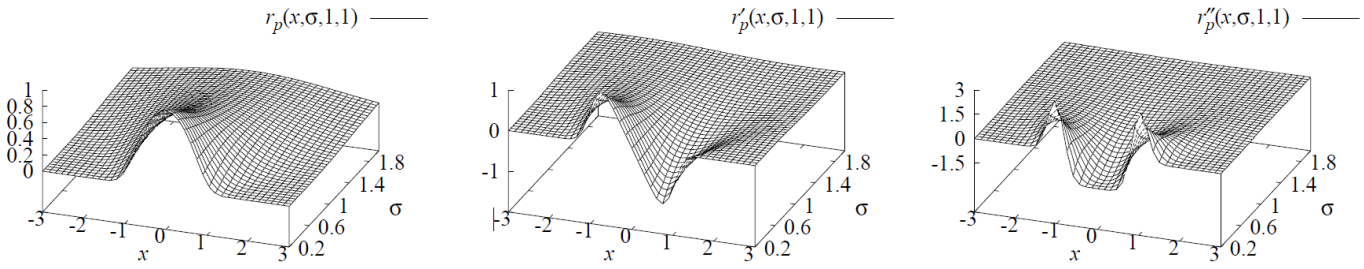


Fig. 3: Scale space behavior of a parabolic profile and its derivatives. The derivatives are obtained by concolving the profile with derivatives of Gaussian kernels, source: [12].

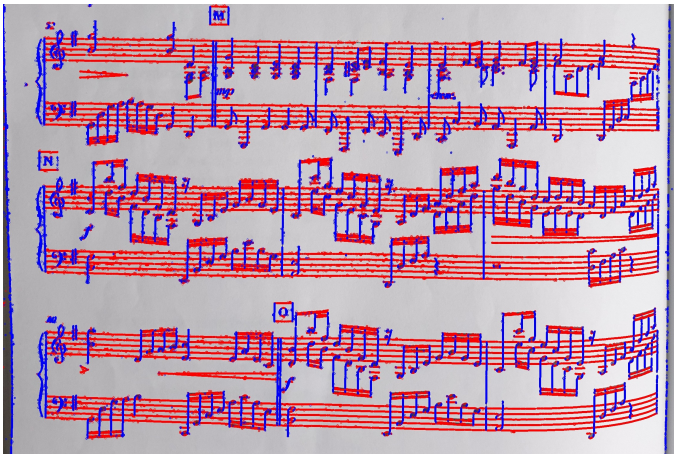


Fig. 4: Detected line points using Stegers method. Horizontal points are shown in red, vertical points in blue.

- Computational cost: In order to reach convergence, the algorithm iterates over the data points several times. This could be combatted by parallelization, since the individual clusters can be updated independently of each other. Still, there exist other clustering methods that consider each data point only once.
- Uneven curve coverage: Because the density of line points is relatively constant along the span of a staff, nothing prevents the clusters from moving along the curve during convergence. This could create cases in which clusters aren't evenly distributed along the x-axis, resulting in an uneven coverage of the staff curve (which is bad for the subsequent curve fitting step). This can be prevented by letting the clusters only inspect data points within a vertical slice around their starting location but this introduces additional complexity and hyperparameters.
- Cluster merging: The merging of individually converged clusters into a staff adds another, critical step to the algorithm.

Another approach would be using Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [14]. Other than mean-shift clustering, this method is especially designed to find clusters with arbitrary shapes, including curves. It also has the advantage of considering every data point only once. The main problem of this clustering method in this context is that its constraints on cluster shapes are too relaxed. Given that there are enough horizontal line points between two staffs, nothing prevents DBSCAN from merging them together. Experimenting with this clustering algorithm showed that it is generally hard to find parameters that exclude this unwanted cases without also excluding points of the staff in places where the curvature of a staff is high due to page curl.

Motivated by those findings, I define a custom clustering algorithm that is designed to work for this particular problem. Based on the assumptions that the y-position $y(x)$ and angle $\phi(x)$ of a staff are continuous in x, this clustering algorithm works by keeping track of the current averaged position in feature space (here consisting of y and ϕ). Data points can then be added based on proximity to that mean position, gradually changing the mean in the process. When feeding data points with increasing x, this results in clusters tracing a smooth path in feature space along the x direction. The mean is averaged over the last n added data points. Since this mean can only change gradually as new points are added to the cluster, this can be interpreted as the cluster keeping its local momentum. This gives the method its name: *Momentum clustering*. For each data point offered to the algorithm, each of the current clusters considers its proximity to that data point. Each cluster whose current distance to the data point is smaller than a user defined *bandwidth* is considered capable of claiming the data point, but only the cluster with the biggest size (highest number of already associated data points) actually claims it. The reason why I use a cluster's size rather than the smallest distance to the data point is to avoid multiple clusters fighting over the same data points in a region. By using the size of a cluster instead, a bigger cluster will gain even more data points over its competitors. This runaway effect ensures that a region is correctly assigned to one cluster only, rather than being split up by various smaller, competing clusters. If the distance exceeds the bandwidth for all clusters, a new cluster is initialized on the data point's location. This ensures that data points are uniquely attributed to clusters and that a full coverage of the data is automatically achieved. This method also examines each data point only once.

The found clusters should also auto-detect the end of curves.

This is done by incorporating a minimum density ρ_{min} into the cluster. Clusters terminate once their own density underpasses this minimum density or until no data are left. A minimum number of points n_{min} safeguards the algorithm from using erroneous density data resulting from an insufficient number of samples.

$$\rho_{min} < \rho = \begin{cases} n/(x_{max} - x_{min}) & if\ n > n_{min}\,, \\ \infty & else. \end{cases} \tag{11}$$

Where n is the current number of data points in the cluster and $x_{min} - x_{max}$ is the current cluster width. The complete algorithm is outlined in the following pseudocode:

```
for dataPoint in dataPoints
  //expects data points in monotonically increasing
    order
  Cluster claiming = null
  int maxSize = 0

  //loop over current clusters
  for cluster in currentClusters
    //remove if density too low
    if (cluster.getDensity() < minDensity)
      safeAndRemoveCluster(cluster)

    float d = distanceFunction(cluster, dataPoint)
    if (d < bandwidth && cluster.size() > maxSize)
      claiming = cluster
      maxSize = cluster.size()

  //let the biggest valid cluster take the data
    point
  if (claiming != null)
    claiming.take(dataPoint)
  //or create a new one if no cluster claims the
    data point
  else
    currentClusters.add(new cluster(dataPoint))

//save all remaining clusters
for cluster in currentClusters
  safeAndRemoveCluster(cluster)
```

Code 1: Momentum clustering routine. The distance function can be defined by the user via lambda function.

This clustering algorithm is very effective in separating the line points into individual staffs. It also serves as a very good *outlier removal* algorithm, in the sense that data points that are too dissimilar to the *local* features are not added to a cluster. To separate line points into staffs, I use momentum clustering with the Euclidean distance and y, ϕ as features. To make angles comparable to differences in height, I multiply the angles differences with a constant gain factor C_ϕ. The distance between a cluster c and line point p is then given as:

$$\sqrt{(c.\bar{y} - p.y)^2 + (C_\phi \cdot (c.\bar{\phi} - p.\phi))^2)} < b \tag{12}$$

I choose $C_\phi = 400$ and a more relaxed bandwidth of $b = 3.8 \cdot \texttt{iss}$ to ensure that staffs are safely captured by one cluster. This way the cluster can safely track the staff even in regions of high curvature. A downside of this is that the more relaxed bandwidth may not filter non-staff lines with similar angle. For that reason, I use momentum clustering once again on the returned subset of points, leveraging its outlier removal capabilities. This time I only consider the angle in the distance function, albeit with a much tighter bandwidth:

$$distance(c, p) = |c.\bar{\phi} - p.\phi| < 0.1 \tag{13}$$

The points that remain in the strongest cluster after this second clustering are considered outlier adjusted and passed to the oncoming curve fitting step. This two-step procedure ensures that even highly curved staffs can be tracked without sacrificing strict outlier removal. It can be seen in figure 5, that the end result in fact mostly consists of staff line pixels. This approach proved to be extremely stable under a wide variety of input images.

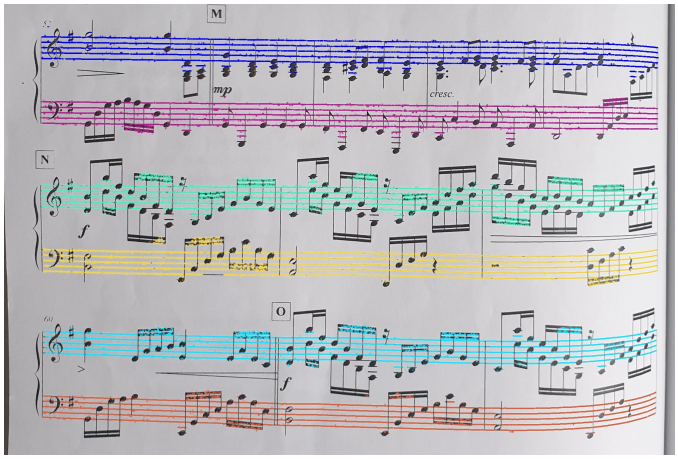


Fig. 5: Resulting clusters after applying momentum clustering. Each color represents a different identified staff.

4) Curve fitting: After the clustering step, the generated sets of staff line points need to be refined into individual curve models. At that point, almost all other near horizontal line segments got removed from the point sets. Despite that, a tracking of individual staff lines is still not feasible as outlined in Chapter II-A. Instead, I propose an approach that uses the information contained in all five staff lines to jointly estimate a common curve model for the staff. Assuming that the depth across a staff varies only marginally compared to the depth of the page from the camera, the projected staff lines can be assumed to be translates of *one* curve, shifted in the y-direction in image space. Exploiting this property, one can fit a curve model to all five staff lines simultaneously, making the process robust to extended gaps in individual lines. As the curve of a staff can be difficult to describe with a single function, I split the staff into N segments in x-direction and describe the staff curve with the help of piecewise defined sub-functions.
Under the made assumption that projected staff lines are translates of the same curve, it follows that the derivatives for a given x are identical for the entire staff. A simple approximation of the staff curve can thus be obtained by averaging the slope of all line points in a segment and using the result as the slope of a linear function. A second order

polynomial can be obtained by fitting a linear function $ax+b$ to the slopes of a segment. Integrating this function yields the piecewise defined functions

$$f_s(x) = \frac{1}{2}ax^2 + bx + c \tag{14}$$

Where c is determined in such a way that there is no discontinuity at segment endpoints. Both of these approaches yield good results, but are susceptible to drift. Usually the drift is in the orders of magnitude of a $\texttt{iss}$ over the course of a whole staff.

So far, the y-position of the line points did not get used to contribute to the estimated model because they belong to different staff lines. I assumed the $\texttt{iss}$ to be constant in Chapter II-B1. Putting that assumption to use, I consider the remainders of y when divided by the $\texttt{iss}$:

$$y \mapsto y \bmod \texttt{iss} \tag{15}$$

This way, y-values are invariant to shifts by integer multiples of a $\texttt{iss}$ or in other words: points of all staff- and ledger lines are identical. This makes it possible to combine the information of all staff lines into one unified metric, this time to use it to correct the drift that occurs over a segment in the given segment function f_s. This can be done by sampling data points at the end of each segment and consider the remainders of their difference to the segment function:

$$\alpha_i = (y_i(x) - f_s(x)) \bmod \texttt{iss} \tag{16}$$

I shift the resulting values to be in range $(-0.25\texttt{iss}, 0.25\texttt{iss}]$ to avoid the discontinuity when crossing $\alpha_i = 0$. The drift that occurs in this segment relative to the segment function can then be obtained by the median of these values

$$\text{drift} = \text{med}(\alpha_i) \tag{17}$$

It is then used to correct the segment function before moving on to the next segment. In the case of the quadratic model functions, either a or b can be chosen to correct the found drift. I choose to use the linear coefficient b for the drift correction to make the approach also applicable to linear segment functions. The corrected linear coefficient b^* can be obtained by solving the equation:

$$\frac{1}{2}ax^2 + bx + c + \text{drift} = \frac{1}{2}ax^2 + b^*x + c \tag{18}$$

yielding the corrected segment function:

$$f_{s,corr}(x) = \frac{1}{2}ax^2 + (b + \frac{\text{drift}}{\text{segment length}})x + c \tag{19}$$

Using quadratic segment functions instead of linear ones results in smoother descriptions of the staff curve. Despite that, it should be noted that this curve description is strictly speaking still not differentiable at segment endpoints, which is irrelevant for my purposes.

The resulting curve description is very accurate, but lacks the real height information of the staff lines. The last step involves finding the height of the five staff lines in the image and translating the curves in y-direction to match them. Using the difference between line points and the corresponding curve height $y_i(x) - f_{corr}(x)$, the position of staff lines become sharp peaks when horizontally projected. I use mean-shift clustering with a bandwidth of $0.18\texttt{iss}$ to differentiate individual staff lines from each other. The five clusters with the highest number of data points are chosen, and their mean height $\bar{y}$ is stored. Using these heights and the curve $f_{corr}(x)$, the height of a staff line i is fully given as

$$y_i(x) = \bar{y_i} + f_{corr}(x) \tag{20}$$

The reason why I use mean shift rather than k-means clustering [15] with $k = 5$ is that beside the staff lines, additional clusters do still exist. These can come from ledger-, volta bracket- or octavation lines, which are all parallel to the staff. By using mean-shift clustering, these clusters won't get merged with the staff line clusters, but rather identified as their own. Because the number of points in these additional clusters is much lower compared to the clusters of staff lines, they get discarded in the cluster selection process.

Not every cluster that was proposed by momentum clustering is actually a staff. I combat this by defining the following requirement that every proposed staff must fulfill:

- length: Staffs usually span across the whole document. For that reason, I discard any proposed staff whose line points don't span more than 60% of the image width.
- equal line length: The amount of line points n_i assigned to each of the five strongest mean-shift clusters must be roughly the same. I consider the normalized standard deviation between the cluster sizes, discarding any instance where it exceeds a threshold of T_n

$$\frac{\sigma}{\bar{n}} = \frac{\sqrt{\frac{(n_i - \bar{n})^2}{5}}}{\bar{n}} < T_n \tag{21}$$

The staff curves that result from this method are extremely accurate. They also reliably find the beginning and end of staffs in the image. A few results of this algorithm are illustrated in Figure 15.

III. Image rectification

Generally, the rectification of an image can be described as a mapping R from the image coordinates to the coordinates of the unrolled plane:

$$R : \begin{pmatrix} x \\ y \end{pmatrix} \mapsto \begin{pmatrix} u \\ v \end{pmatrix}$$

This mapping is uniquely defined by a set of lines that are parallel in u and v, together with their distances Δu and Δv respectively. I will call these lines *latitudes* and *longitudes*. If these lines are known in image coordinates (here they are curves in general), the dense mapping R can be described with the use of a suitable multivariate interpolation function. Put differently, the rectification requires the construction of

a *warping mesh* in image coordinates that has corresponding vertical and horizontal lines in the rectified image. Using the already detected staffs as latitudes, the remaining rectification problem can be broken down into these sub-problems:

- find descriptions of longitudes in image coordinates
- determine Δv for latitudes
- determine Δu for longitudes
- calculate intersections between latitudes and longitudes
- use a suitable interpolation method to define R

A. Background

In this chapter, I will quickly describe the principles of projective geometry and give an overview over mathematical models that are essential in understanding the rectification process.

1) Pinhole camera model: The pinhole camera model is a simple but accurate model to describe how images are formed by our eyes or a camera. In this model, light rays from a scene pass through a small aperture (the pinhole) and project an inverted image onto a flat image plane located behind the aperture. The mapping from 3D coordinates to the 2D image is called *perspective projection*. It is a nonlinear, many-to-one mapping, meaning that an infinite amount of scene points are mapped onto the same point in image space. Some important characteristics of this projection are:

- The distance of an object is inversely proportional to the size of its projection on the image. This foreshortening lends perspective projections their sense of depth
- lines in 3D space stay lines under perspective projection
- Distances and angles are not preserved
- Parallel lines generally don't project to parallel lines on the image. Instead, they converge in a point called *vanishing point*. The only case where parallelity is conserved is when the 3D lines are parallel to the image plane

2) Developable surface: A developable surface is a smooth surface that has zero Gaussian curvature everywhere. As a result, this surface can be flattened onto a plane without stretching or tearing. Putting it the other way around, any surface that can be made with a flat plane without distorting it (like a sheet of paper) is a developable surface. All developable surfaces in 3D space are also ruled surfaces [16], which means that the surface can be created by sweeping a straight line between two curves, as illustrated in Figure 6. These curves are called *directrices* while the straight connecting lines are called *rulings* or *generatrices*. Because the document rectification process tries to find and apply the transformation that flattens a 3D shape onto a plane, the shape being a developable surface is a necessary prerequisite for any meaningful rectification.

3) General Cylindical Surface (GCS): A GCS is a type of developable surface, whose rulings are all parallel to each other. The most prominent example for that kind of surface is the lateral surface of a cylinder in 3D space. In the context of document rectification, GCS are mostly associated with the pages of an opened book, where the spine of the book causes the page to curve in the direction of the text lines. In this case, the rulings are the lines orthogonal to the text lines. Because all

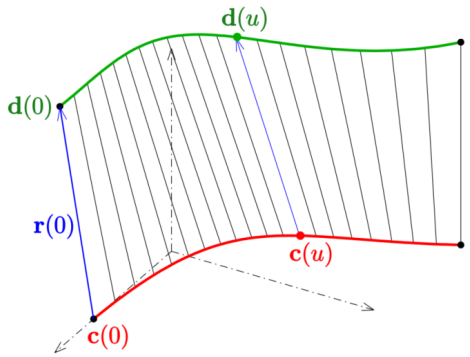


Fig. 6: Ruled surface, created through sweeping rulings (blue) through two directrices (red, green), source: https://en.wikipedia.org/wiki/Ruled_surface .

rulings are parallel in 3D space, their projected lines converge in a single vanishing point. With that orthogonality between rulings and text lines, Meng et al. showed that the tangent lines of corresponding text lines converge in a single vanishing *line* [17]. While imposing some restrictions on the shape of the surface, this makes the whole perspective describable by this vanishing line and the vanishing point of the projected rulings alone, as illustrated in Figure 7. As a result, the 3D curvature of the page can be described by a single directrix in 3D space.

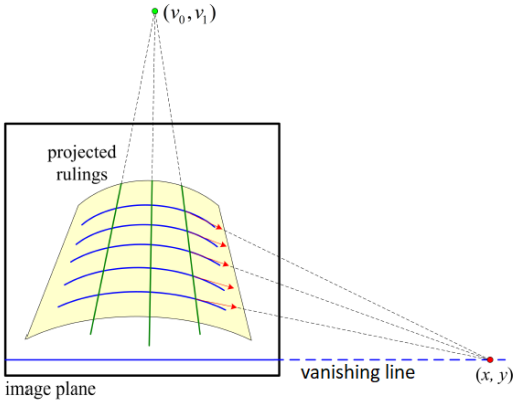


Fig. 7: Geometric characteristics of a GCS under projective projection. The projected rulings (green) converge in a single vanishing point, while the tangent lines of projected directrices (blue) along a ruling intersect along a common vanishing line, source: [18].

B. Related work

To the best of my knowledge, there exists no published approach that focuses on the rectification of musical documents in particular. In this chapter, I will give an overview over the related topic of text-based document rectification, that has been the subject of ongoing research.

When a 3D shape is projected onto an image, the depth information is lost in the process. Especially for curved pages, this causes different foreshortening of lengths across the

image. Rectifying these foreshortenings requires knowledge about the 3D shape of the page that are difficult to obtain from a single image. In fact, 3D reconstruction from a single image is an under-constrained problem that can not be solved without incorporating additional model assumptions or knowledge. Some works obtain 3D data by the use of special scanning contraptions [19, 20] or rely on multiple images to perform the 3D reconstruction [21]. However, the use of specialized hardware makes these approaches inaccessible or impractical to most people. Similarly, the input is mostly given as a single, camera captured image. In the absence of more data to work with, many methods assume strong shape models to construct a suitable warping mesh for rectification.

A popular approach is to assume a GCS as shape model, with rulings perpendicular to text lines [22, 17, 23, 18]. In [22], Cao et al. additionally assume that the photographed page is parallel to the image plane. Under this assumption, the rulings are mapped linearly to their projections. The same can be approximately said about directrices if their distance to the camera varies only marginally. This is a bit contradictory, as the curvature along the directrices makes it impossible to photograph the page parallel to the image plane as a whole. In practice, this imposes heavy restrictions on camera angle and positions, and only very small page curl can be tolerated by the method.

Meng et al. remove these restrictions and use the text lines in the document to jointly estimate the formed vanishing line and the vanishing point of projected rulings, thus fully describing the perspective [17]. They do this by minimizing a single energy functional w.r.t. the vanishing point location and focal length, using the text lines as only features. They then use a paraperspective projection to recover the 3D directrix of the surface. Experimenting with this approach for myself, it seemed that this approach works well for surfaces photographed with a strong angle and page curl, to give clear visual clues about the perspective. When images are photographed head on, however, the curves are much more straight and a big change in the vanishing point location hardly changes the cost, making the objective function incredibly flat around the optimum. As a result, it can be difficult to find the optimal set of parameters that describe the perspective for these camera poses. In practice, it is exactly these camera poses that provide high quality input and equal resolution of details across the whole page. In a later paper by the same author, visual clues of ruling direction from the image are incorporated into the perspective estimation as well. Here they use the Radon transform to generate line segments of assumed rulings and vote for their common vanishing point in spherical coordinates [18]. Using spherical coordinates as voting space prevents the problem from becoming ill-posed as the vanishing point approaches infinity. Once the vanishing point has been computed, they use a vector field of local text lines to construct the vanishing line.

In another work, the surface is described by means of a transformation between the flattened plane and the GCS. They assume that the document surface can be expressed with polynomials and recover the 3D shape by jointly estimating the rotation angles and polynomial coefficients of this transformation in an optimization problem. They need to incorporate additional constraints in their objective function to make it unambiguous. These include an equal spacing of text lines in vertical direction and a horizontal alignment of text in the same text block. These constraints are not applicable to musical documents, where the spacing between staffs varies depending on the content of the staffs and staffs are not aligned frequently.

The assumption of a GCS is reasonably well fulfilled for pages of an opened book. In this case, the connection of the page to the book spine forces the page to curve in the direction of the text lines. Because paper is developable and the Gaussian curvature therefore zero everywhere, it follows directly that the curves perpendicular to the text lines must have zero curvature and in fact be rulings of the surface. This is not the case for freestanding pages, though. In this case, gravity is much more likely to cause the page to sag and cause curvature *perpendicular* to text lines. Slight creases and dog-ears further prevent rulings from being parallel to each other, violating the GCS assumption. For that reason, other works relax the shape model to be developable and smooth only and rely on local image cues instead to construct dense slope fields [24, 25, 26].

The lack of a stricter shape model makes the baseline estimation more difficult and susceptible to outliers and noise. As an example, rulings can not be constructed by passing through their common vanishing point and an arbitrary point on the image anymore. In the absence of these assumptions, local orientation of longitudes comes mostly from vertical character strokes. In [24], the image is divided into small blocks and the most prominent texture flow orientations are extracted for each block. Then they apply a relaxation process to adjust and select flow values that agree with neighboring blocks. They use extra- and interpolation to define dense flow fields for both the vertical and horizontal flow. Tian et al. refine this approach by making the size of the regions in which local orientations are averaged part of an optimization problem [25].

Generally speaking, the main difficulty with approaches like these is to construct *dense* slope fields from *sparsely* distributed orientation information. These approaches are especially sensitive to:

- Patch size: Orientations need to be averaged over their local neighborhood to obtain stable results. If this neighborhood is too small, there may not exist enough data points for a reasonable average or no data points at all. If it is too big, local orientations will get averaged out. The density of data points may vary widely, especially for vertical orientations. This makes dynamic patch sizes almost mandatory.
- Presence of outliers: Averaging is an outlier sensitive process. These outliers need to be detected and removed first in order to not distort the result.

C. Proposed method

With a staff line detection that provides very accurate latitudes already established, the next step in order to do document rectification is the detection of longitudes. So far, no shape model has been imposed on the photographed surface other than it being smooth. Musical documents also contain an abundance of vertical lines including bar lines and stems as can be seen in Figure 4. Motivated by this fact, I propose to use again only cues obtained from the image to construct longitudinal curves without additional shape model assumptions. I avoid using fixed size, boxed shape patches to average orientations and instead use the location and shape of the detected staffs for a more dynamic slope field creation that is tailored to musical documents.

I use the resulting longitudes together with the detected staffs to construct a warping mesh and employ the methods described in [25] to reconstruct the 3D-shape of the surface. Because this method also makes no model assumptions other than the shape being smooth, my approach is applicable to more general page surfaces and makes it possible to rectify book and freestanding pages without loss of quality. The method is described in more detail in the following chapters.

1) Detection of longitudes: Looking at Figure 4, one can see that the curve point detection described in Chapter II-B2 also provides a good basis for the detection of vertical lines. In contrast to staff lines, however, these lines don't span across the whole document and are instead sparsely distributed across the image, which makes the recovery of vertical lines much more challenging than for staffs.

To make the detection more stable, only vertical line points that originate from the musical notation on the detected staffs should be considered. This is important as an image may contain additional vertical lines surrounding the staffs or even fragments of musical notation from a different page whose orientation might differ from the main one. An example of this is shown in Figure 15. Because the staffs are known at that point, the vertical points can be filtered using the staff curves. Only points that are on or closely above or below a staff are kept. This also separates the points to be associated with a specific staff. Similar to the staff detection process, the tangential angle of vertical line points Φ can be described as a continuous function in x with the vast majority of the points contributing to that function. I can therefore use momentum clustering once again to obtain all points that follow the evolution of this dominant direction across each individual staff. I use $|\Phi|$ as distance function with a bandwidth of $b = 0.06\texttt{iss}$. As a result, I obtain the points belonging to the one biggest cluster for each staff. The result is illustrated in Figure 8. Next, I use the resulting points to average local orientation information olong a staff. For that, I add points from left to right to a list and save their aggregate x and Φ. Once the list reaches a size of k elements, I create a sample point on the staff with the coordinates being the mean coordinates of the last k points

$$s = \begin{pmatrix} \bar{x} \\ f_{corr}(\bar{x}) \end{pmatrix} \tag{22}$$

and save it together with the average slope $\bar{m} = \frac{1}{k}\sum_i^k \frac{\Delta x_i}{\Delta y_i}$. The first $\frac{k}{2}$ elements are then removed from the list. This process is repeated until all vertical points associated with the staff have been processed. The resulting control points s_i are saved together with their averaged slope. This way the patches follow the shape of the staff and have no fixed size. Because it is the number of data points k that is constant for a patch, areas of low data point density don't pose a problem.

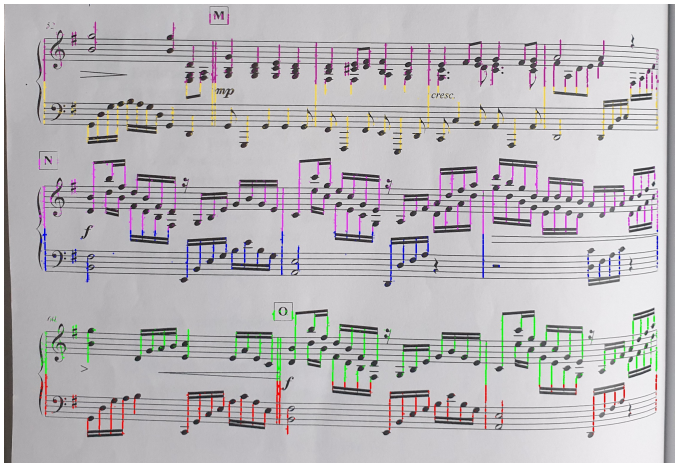


Fig. 8: Remaining vertical points after applying momentum clustering for each staff. The orientations associated with these points represent the local orientation of longitudes very well.

I define a dense slope field by interpolating between the control points s_i:

$$slope(\vec{x}) = \frac{\sum_i K_\epsilon(\vec{x}) \cdot s_i.\bar{m}}{\sum_i K_\epsilon(\vec{x})} \tag{23}$$

where K_ϵ is a Gaussian kernel function defined as:

$$K_\epsilon = \exp - \frac{\|\vec{x} - s_i\|^2}{2\epsilon^2} \tag{24}$$

and ϵ is a smoothing parameter that controls the locality of the interpolation. The kernel is truncated after 3ϵ.

Using this slope field, longitudes can be constructed as flow lines of that field. I assume longitudes to be piecewise linear functions between staffs. Starting from an initial position, I query the slope between the first two staffs and use it to update the current position of the longitude. This process is then repeated for all remaining pairs of staffs from top to bottom until the longitude is fully constructed. Starting from the center of the image, this process is repeated left- and rightwards, until the longitudes cover all staffs. The distance between longitudes, denoted as Δx_{start}, is controlled by a hyperparameter that regulates the density of the created longitudes.

$$\Delta x_{start} = C_l\texttt{iss} \tag{25}$$

The result is illustrated in Figure 9.

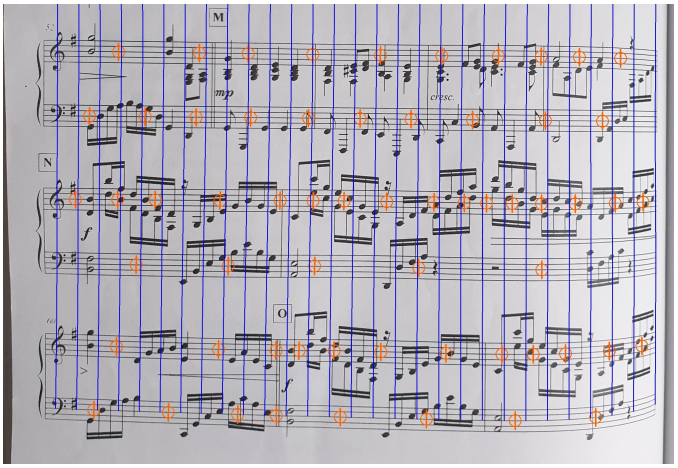


Fig. 9: Longitudes, constructed with $C_l = 4$ from the dense flow field. The control points and their slopes are shown in orange.

2) Mesh creation: Now that latitudes and longitudes are defined in image coordinates, the warping mesh can be fully defined. For latitudes, I choose the center staff line from each staff. The points of the warping mesh are then given by the intersection points of these staff lines with the longitudes. Next, I need to calculate the distance of adjacent latitudes Δu and longitudes Δv in the unrolled plane. These distances also dictate the scale of the rectified image. I use a single parameter, the $\texttt{iss}$ of the rectified image, $\texttt{iss}_r$ to control this scale. This factor can be chosen to be a fixed constant, resulting in every input image to have a same characteristic scale after rectification. This is especially useful for later object detection or deep learning stages, effectively acting as a normalization on the input data. Setting $\texttt{iss}_r$ equal to the image $\texttt{iss}$ results in a rectified image with similar resolution to the original.
Staff lines come as a set of five parallel lines. Measuring the height difference of those lines gives a sense of scale in the direction orthogonal to the staff. This can be exploited to estimate the lengths Δv in the rectified image. Due to the projective projection, these lengths cannot be related with simple ratios because they are distorted by the transformation. There is a metric, however, the *cross-ratio*, that remains invariant under projective transformations. It provides a measure of the relative distances between four collinear points, defined as follows:

$$(A, B; C, D) = \frac{AC \cdot BD}{BC \cdot AD} \tag{26}$$

Regardless of the choice of coordinates or the specific projective projection applied, as long as the four points lie on the same line or projective plane, their cross-ratio will remain constant. This makes it possible to relate lengths in the projected image to lengths in the rectified image. Without loss of generality, I consider a vertical line in image coordinates and its intersections with the first and last staff lines of its surrounding staffs as points A, B, C and D respectively. The location of these four points is shown in Figure 10. All distances that occur in equation (26) are directly given by the known staff curves. The resulting cross-ratio can then be related to the cross-ratio of the rectified image, where the distance between the first and last staff lines of a staff must be $5\texttt{iss}_r$ exactly.

$$\frac{AC \cdot BD}{BC \cdot AD} = \frac{(5\texttt{iss}_r + \Delta v)^2}{\Delta v \cdot (10\texttt{iss}_r + \Delta v)} \tag{27}$$

This equation is then solved for Δv to retrieve the distances of latitudes in the rectified image. This approach also has the advantage that it incorporates scale information of both surrounding staffs into one measure.

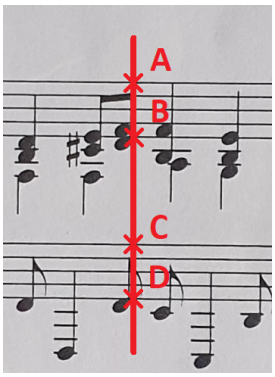


Fig. 10: Construction of four collinear points by intersecting two staffs by a vertical line.

Estimating Δu is much more challenging, as there are no fixed distances in this direction that could be exploited for scale. Additionally, contents along this axis are often subject to intense foreshortening effects when photographing book pages. Rectification of these foreshortenings requires the knowledge about the 3D shape of the page. For this, I employ a method described in [25] to reconstruct the 3D page surface. The idea is to map each cell of the 2D warping mesh to a parallelogram in 3D space. According to the theorem of intersecting lines, the 3D coordinates of a parallelogram's vertices $V_i = (X_i, Y_i, Z_i)$ are equivalent to $(\frac{x_i}{f} Z_i, \frac{y_i}{f} Z_i, Z_i)$, where f is the focal length and x_i, y_i denote coordinates in the image plane (both in pixels) with the principal point being the origin. The focal length f can be obtained from the image dimensions and the camera's respective field of view. By definition, any parallelogram in 3D space must fulfill the equation:

$$-V_{j,1} + V_{j,2} + V_{j,3} - V_{j,4} = \Delta P_j = 0 \tag{28}$$

The enumeration of the vertices is shown in Figure 11. This equation yields three linear constraints per parallelogram j. By adding the constraints of all j cells as rows to a unified vector $\Delta P(Z)$, the vector containing Z-values at each vertex i can be found by minimizing

$$Q(Z) = \|\Delta P\|^2 \tag{29}$$

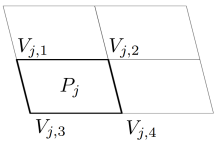


Fig. 11: Parallelograms and the used enumeration of their vertices V.

ΔP can be rewritten as matrix-vector product $\Delta P = AZ$, where A is a coefficient matrix. Using this formula, (29) can be rewritten as

$$Q(Z) = \Delta P^\intercal \Delta P = (AZ)^\intercal AZ = Z^\intercal (A^\intercal A) Z \stackrel{!}{=} 0 \quad (30)$$

for which a solution Z can be obtained up to a scale by finding the null space of $A^\intercal A$. Because this equation is quadratic in Z, only a global minimum exists.
In this form, the optimization problem is susceptible to noise and errors in the warping mesh. Tian et al. therefore propose to relax this equation by additionally estimating X_i and Y_i together with Z_i. These values enter the equation by a regularized term that penalizes re-projection errors, leading to the modified objective function

$$Q^*(X, Y, Z) = \|AV\|^2 + \lambda \|BV\|^2 \quad (31)$$

where λ is a regularization constant and BV is constructed by concatenating rows of two constraints for each vertex V_i

$$BV_i = \begin{pmatrix} X_i - \frac{x_i}{f} Z_i \\ Y_i - \frac{y_i}{f} Z_i \end{pmatrix} \quad (32)$$

Similar to the prior objective function, this can be solved exactly by finding the null space of $A^\intercal A + \lambda B^\intercal B$. The resulting 3D surface estimation is plotted in Figure 12.

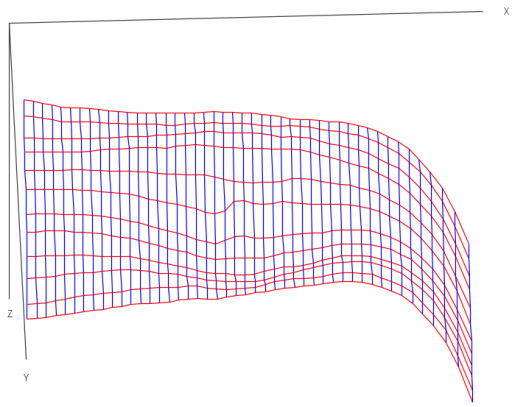


Fig. 12: Estimated 3D surface of the example shown in preceding figures within this chapter. Axis are not shown to scale.

The foreshortenings in u-direction can now be rectified by incorporating 3D lengths of the page surface. Similar to [25], I equate the ratio of side lengths of the 3D parallelogram $\frac{a_j}{b_j}$ to the lengths of its corresponding grid cell j in the rectified image:

$$\frac{a_j}{b_j} = \frac{\Delta u_j}{\Delta v_j} \quad (33)$$

For every column c of the warping mesh, all cells vote for the same Δu. Their estimations usually agree, because the term $\|AV\|^2$ in the optimization problem ensures that opposing side lengths are indeed as equal as possible to form 3D *parallelograms*. Nonetheless, a suitable reduction strategy has to be defined to reduce all estimations of a column to a singular value. I choose the *median* which makes the calculated Δu insensitive to outliers in the estimations.

$$\Delta u_{|c} = \mathrm{med}(\Delta v_j \cdot \frac{a_j}{b_j}) \quad (34)$$

3) Rectification: With a full warping mesh defined in image coordinates as well as in coordinates of the rectified image, the mapping function R can be defined. In practice, one needs to define the inverse mapping function R^{-1} that defines where each pixel in the rectified image is located in the original image. This way, the corresponding pixel values can be interpolated from the location in the original image in a process called backwards registration.

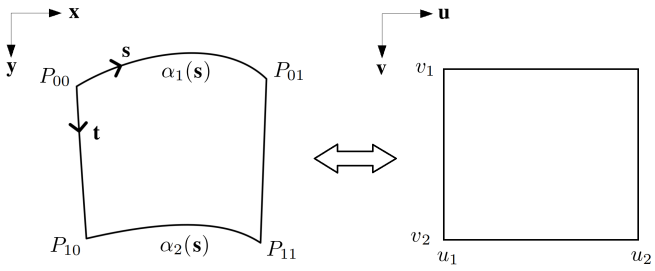


Fig. 13: Geometric relationship between a cell in image coordinates (left) and its rectified cell (right).

For this mapping function, I consider individual cells of the warping mesh in the original image. Each cell is bounded by sections of two staff curves and two straight longitudes, as illustrated in Figure 13. Let the two bounding staff curves be given in parametrized form $\alpha_1(s)$ and $\alpha_2(s)$ with $0 \leq s \leq 1$ as parametrization parameter. R^{-1} can then be defined for that cell by blending linearly between those two curves:

$$R^{-1}(u, v) = t \cdot \alpha_2(s) + (1 - t) \cdot \alpha_1(s) \quad (35)$$

the parametrization parameters s and t can directly be calculated from u and v by means of linear interpolation:

$$s \cdot u_2 + (1 - s) \cdot u_1 = u \quad (36a)$$
$$t \cdot v_2 + (1 - t) \cdot v_1 = v \quad (36b)$$

where u_1, u_2, v_1 and v_2 are the coordinates of the intersection points in the rectified image. At last, it needs to be defined how the staff curves, that are defined in terms of x, can be parametrized in terms of s. With sufficiently small cells, a linear mapping between those two variables can be assumed, yielding:

$$x_i = s \cdot x_{\alpha_i,1} + (1 - s) \cdot x_{\alpha_i,0} \tag{37}$$

where $x_{\alpha_i,0}$ and $x_{\alpha_i,1}$ describe the start and end point's x coordinate of a staff curve in image coordinates. With that, the inverse mapping can be defined for every pixel of the rectified image by following steps:

```
for u in rectified image width
    for v in rectified image height
        //calculate parametrization parameters
        s = s(u,v)
        t = t(u,v)

        //calculate alpha_1(s), alpha_2(s)
        x_alpha1 = x(s, x_10, x_11)
        x_alpha2 = x(s, x_20, x_21)
        y_alpha1 = staffcurve1(x_alpha1)
        y_alpha2 = staffcurve2(x_alpha2)

        //calculate R_inv
        R_invX = t*x_alpha2 + (1-t)*x_alpha1
        R_invY = t*y_alpha2 + (1-t)*y_alpha1
```

Code 2: Construction of the inverse mapping function.

IV. Results and Discussion

A custom dataset was created to test the method on. For that reason, printed sheet music and open music books were photographed with a Samsung Galaxy S9 main camera. The dataset comprises a total of 29 images, exhibiting a range of characteristics, including differences in scale, page curvature, font styles, and varying degrees of degradation. A selection of these images, along with their corresponding rectification results, is showcased in Figure 15.

The algorithm demonstrates a remarkable ability to track the staffs with very high accuracy, including their start- and endpoints, respectively. The detection of longitudes is also very accurate, albeit the relatively sparse data available for this orientation can occasionally result in slight deviations from perfect vertical alignment in the rectified images. Nonetheless, the exceptional accuracy in detected baselines consistently yields very high quality rectified outcomes, where the perspective foreshortenings are entirely eliminated, and a uniform scale is imparted to objects. Furthermore, the process intelligently omits musical notations from different pages that may appear at the image's periphery, as also exemplified in Figure 15. This attribute lends the method significant robustness when handling diverse real-world scenarios.

The staff detector's operational assumption relies on the approximate constancy of the `iss` across the image. Figure 14 highlights the challenges that arise when this assumption is not strictly adhered to.

This image is taken much closer to the camera as it only contains two staffs. It also displays notable page curl. This

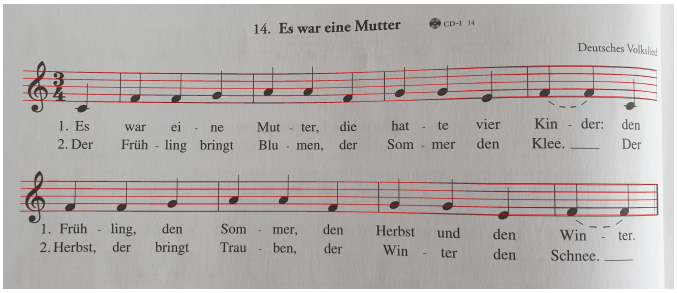


Fig. 14: Staff detection results for a case in which the `iss` constancy assumption is not well fulfilled. Despite this, the tracking accuracy is still very good on average.

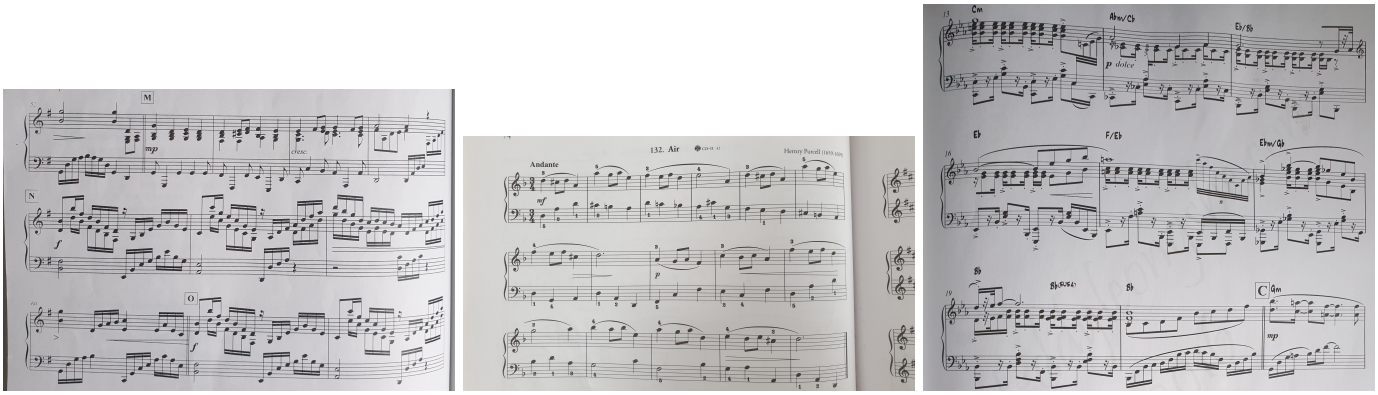
deviation in varying distances between the camera and the different sections of the page leads to variations in `iss` being more pronounced. The constancy assumption, on the other hand, ensures that the vertical spacing between staff lines remains accurate on average, but may appear too tight on one side of the image and too loose on the other. Relaxing this assumption and allowing for the creation of distinct staff models for each staff line would enhance flexibility, but could compromise the method's robustness. It is worth noting that even under such an extreme case, the tracking accuracy remains consistently high, with deviations smaller than $\frac{\texttt{iss}}{4}$. This level of precision is sufficient to avoid misclassification of note positions on the staff. Therefore, the constancy assumption remains a reasonable choice in the overall context.

The rectification process takes $2.07s$ on average, with the most time needed to remap each pixel to the rectified image coordinates (about 50%).

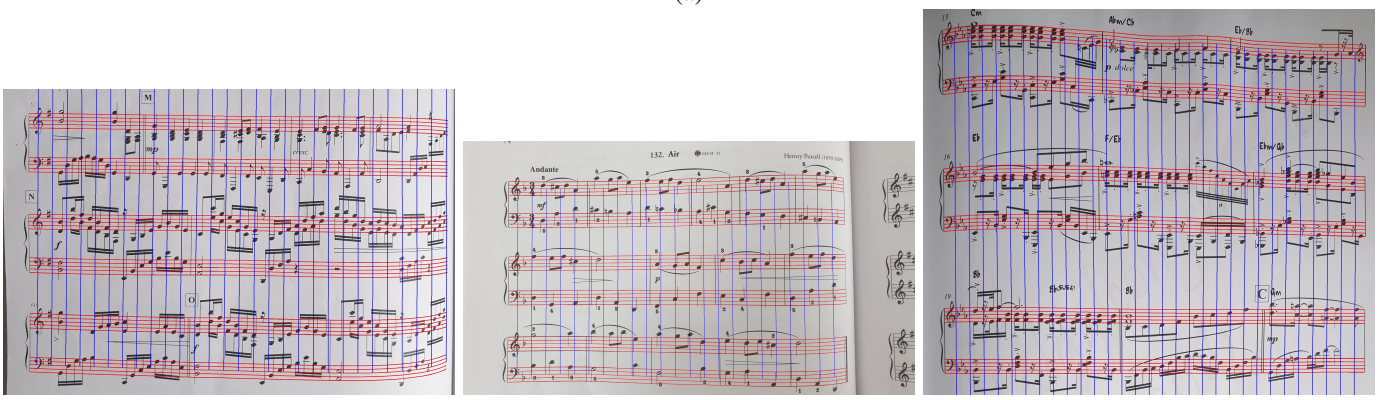
The source code is available on Github at https://github.com/Daniel63656/OpticalMusicRecognition.git. The full dataset is located at \src\main\resources\dataset. To run the rectification process, execute the main method within the *DocumentRectification* class at \src\main\java\com\immersive\omr. Please ensure that the path to the OpenCV library is correctly specified in the run configuration, as described in the documentation. The results will be generated in the \output directory.

V. Conclusions

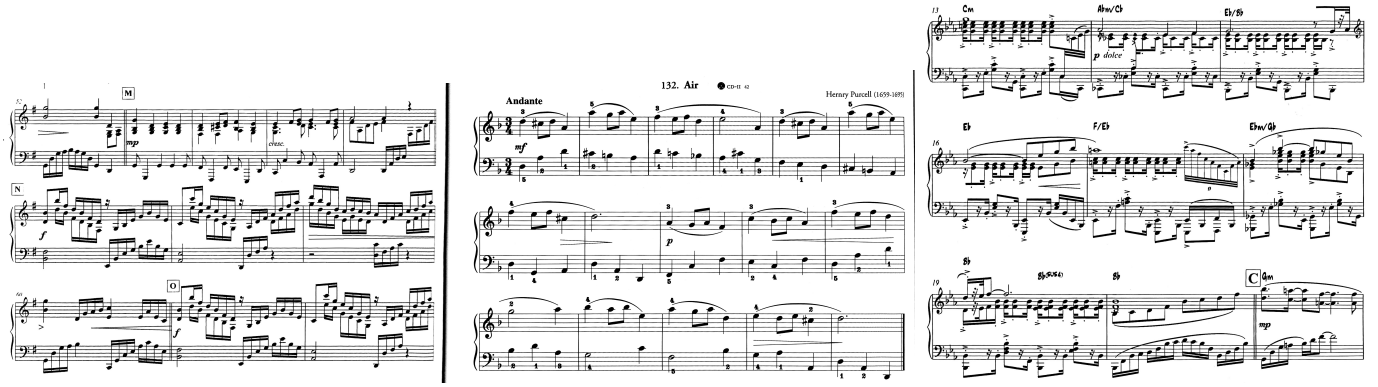
This study presents a novel rectification pipeline tailored specifically for musical documents. By exploiting the distinctive structure and format inherent to these kinds of documents, this method eliminates the necessity to impose rigid shape constraints, such as requiring the surface to be of a general cylindrical shape, as commonly stipulated in the literature. By requiring the page shape to be smooth only, this method becomes stable and applicable to a wide range of real-world applications, such as photographing opened books or freestanding pages. The contributions of this research encompass a robust approach for precisely tracking staffs at a sub-pixel level and generating functional staff descriptions, which subsequently serve as fundamental baselines during the rectification process. Additionally, a method is introduced



(a)



(b)



(c)

Fig. 15: Examples of staff detection and rectification results. Images have been cropped for the sake of space. (a) Curved document images, captured by a smartphone camera. (b) Detected staffs (red) and longitudes (blue). (c) Rectification outcomes. Parallel to the geometric rectification, a custom algorithm was employed to rectify uneven lighting and enhance image contrast.

to estimate perpendicular baselines, leveraging the inherent orientations provided by features such as bar lines and note stems. These baselines are then used to reconstruct the 3D shape of the photographed surface with a shape-from-texture method adopted from the literature [25] and to ultimately flatten the document.

The method operates on single images, requiring neither a calibrated camera nor any prior knowledge about the document's shape or layout. Furthermore, the regular spacing between individual staff lines provide the method a sense of scale, enabling the creation of rectified documents with a user defined scale parameter. holds the potential to enhance the development of more precise and customized object detectors, ultimately improving the accuracy of subsequent stages within the OMR pipeline.

Extensive testing of the method was conducted using images captured under diverse real-world conditions by a smartphone camera. The method exhibits robustness in the face of varying

lighting conditions, scale differences, and shifts in camera positions. It is also ignoring notation of adjacent pages that are visible on the image peripheries, underscoring its practicality and resilience for everyday applications.

REFERENCES

[1] Christoph Dalitz, Michael Droettboom, Bastian Pranzas, and Ichiro Fujinaga. "A Comparative Study of Staff Removal Algorithms". In: *Transactions on Pattern Analysis and Machine Intelligence* 30.5 (2008), pp. 753–766. DOI: 10.1109/TPAMI.2007.70749.

[2] Jorge Calvo-Zaragoza, Isabel Barbancho, Lorenzo J. Tardón, and et al. "Avoiding staff removal stage in optical music recognition: application to scores written in white mensural notation". In: *Pattern Analysis and Applications* 18.4 (2015), pp. 933–943. DOI: 10.1007/s10044-014-0415-5.

[3] Ichiro Fujinaga. "Staff Detection and Removal". In: *Visual Perception of Music Notation: On-line and Off-line Recognition*. Idea Group Inc., 2004, pp. 1–39.

[4] R. Randriamahefa, J.-P. Cocquerez, C. Fluhr, F. Pepin, and S. Philipp. "Printed Music Recognition". In: *Proc. of the Second International Conference on Document Analysis and Recognition*. 1993, pp. 898–901.

[5] Paul V. C. Hough. "Method and means for recognizing complex patterns". In: *US Patent* 3,069,654 (1962).

[6] Genfang Chen, Liyin Zhang, Wenjun Zhang, and Qiuqiu Wang. "Detecting the Staff-Lines of Musical Score with Hough Transform and Mathematical Morphology". In: *Proc. of the International Conference on Multimedia Technology*. 2010, pp. 1–4. DOI: 10.1109/ICMULT.2010.5631269.

[7] Bolan Su, Shijian Lu, Umapada Pal, and Chew Lim Tan. "An Effective Staff Detection and Removal Technique for Musical Documents". In: *Proc. of the International Workshop on Document Analysis Systems*. 2012, pp. 160–164. DOI: 10.1109/DAS.2012.16.

[8] Hiroshi Miyao and Masataka Okamoto. "Stave Extraction for Printed Music Scores Using DP Matching". In: *Journal of Advanced Computational Intelligence and Intelligent Informatics* 8.2 (2004), pp. 208–215.

[9] Mariusz Szwoch. "A Robust Detector for Distorted Music Staves". In: *Proc. of the Computer Analysis of Images and Patterns*. Springer, 2005, pp. 701–708.

[10] Jaime dos Santos Cardoso, Artur Capela, Ana Rebelo, Carlos Guedes, and Joaquim Pinto da Costa. "Staff Detection with Stable Paths". In: *Transactions on Pattern Analysis and Machine Intelligence* 31.6 (2009), pp. 1134–1139. DOI: 10.1109/TPAMI.2009.34.

[11] Nobuyuki Otsu. "A Threshold Selection Method from Gray-Level Histograms". In: *Transactions on Systems, Man, and Cybernetics* 9.1 (1979), pp. 62–66. DOI: 10.1109/TSMC.1979.4310076.

[12] C. Steger. "An unbiased detector of curvilinear structures". In: *Transactions on Pattern Analysis and Machine Intelligence* 20.2 (1998), pp. 113–125. DOI: 10.1109/34.659930.

[13] D. Comaniciu and P. Meer. "Mean shift: a robust approach toward feature space analysis". In: *Transactions on Pattern Analysis and Machine Intelligence* 24.5 (2002), pp. 603–619. DOI: 10.1109/34.1000236.

[14] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. "A density-based algorithm for discovering clusters in large spatial databases with noise". In: *Proc. of the International Conference on Knowledge Discovery and Data Mining*. 2. 1996, pp. 226–231.

[15] Xin Jin, Jiawei" Han, and Geoffrey I. Webb. "K-Means Clustering". In: *Encyclopedia of Machine Learning*. Springer US, 2010, 563–564". DOI: 10.1007/978-0-387-30164-8_425.

[16] Manfredo do Carmo. *Differential Geometry of Curves and Surfaces*. Prentice-Hall, 1976.

[17] Gaofeng Meng, Chunhong Pan, Shiming Xiang, Jiangyong Duan, and Nanning Zheng. "Metric Rectification of Curved Document Images". In: *Transactions on Pattern Analysis and Machine Intelligence* 34.4 (2012), pp. 707–722. DOI: 10.1109/TPAMI.2011.151.

[18] Gaofeng Meng, Yuanqi Su, Ying Wu, Shiming Xiang, and Chunhong Pan. "Exploiting Vector Fields for Geometric Rectification of Distorted Document Images". In: 2018, pp. 180–195. DOI: 10.1007/978-3-030-01270-0_11.

[19] Michael S Brown and W Brent Seales. "Document restoration using 3D shape: a general deskewing algorithm for arbitrarily warped documents". In: *Proc. of the International Conference on Computer Vision*. Vol. 2. 8. IEEE. 2001, pp. 367–374.

[20] Gaofeng Meng, Ying Wang, Shenquan Qu, Shiming Xiang, and Chunhong Pan. "Active Flattening of Curved Document Images via Two Structured Beams". In: *Conference on Computer Vision and Pattern Recognition* (2014), pp. 3890–3897. DOI: 10.1109/CVPR.2014.497.

[21] Shaodi You, Yasuyuki Matsushita, Sudipta Sinha, Yusuke Bou, and Katsushi Ikeuchi. "Multiview rectification of folded documents". In: *Transactions on Pattern Analysis and Machine Intelligence* 40.2 (2018), pp. 505–511.

[22] Huaigu Cao, Xiaoqing Ding, and Changsong Liu. "Rectifying the bound document image captured by the camera: a model based approach". In: *Proc. of the International Conference on Document Analysis and Recognition*. Vol. 1. 7. 2003, 71–75 vol.1. DOI: 10.1109/ICDAR.2003.1227630.

[23] Taeho Kil, Wonkyo Seo, Hyung Il Koo, and Nam Ik Cho. "Robust Document Image Dewarping Method Using Text-Lines and Line Segments". In: *Proc. of the International Conference on Document Analysis and Recognition*. Vol. 1. 2017, pp. 865–870. DOI: 10.1109/ICDAR.2017.146.

[24] Jian Liang, Daniel DeMenthon, and D. Doermann. "Flattening curved documents in images". In: vol. 2. 2005, pp. 338–345. DOI: 10.1109/CVPR.2005.163.

[25] Yuandong Tian and Srinivasa G. Narasimhan. "Rectification and 3D reconstruction of curved document images". In: *Proc. of the Conference on Computer Vision and Pattern Recognition*. 2011, pp. 377–384. DOI: 10.1109/CVPR.2011.5995540.

[26] Gaofeng Meng, Chunhong Pan, Shiming Xiang, and Ying Wu. "Baselines Extraction from Curved Document Images via Slope Fields Recovery". In: *Transactions on Pattern Analysis and Machine Intelligence* 42.4 (2020), pp. 793–808. DOI: 10.1109/TPAMI.2018.2886900.