Chapter 5

Historical Manuscript Dating Using Contour and Stroke Fragments.

Abstract

Historical manuscript dating has always been an important challenge for historians but since countless manuscripts have become digitally available recently, the pattern recognition community has started addressing the dating problem as well. In this chapter, we present a family of local contour fragments (kCF) and stroke fragments (kSF) features and study their application to historical document dating. kCF are formed by a number of k primary contour fragments segmented from the connected component contours of handwritten texts and kSF are formed by a segment of length k of a stroke fragment graph. The kCF and kSF are described by scale and rotation invariant descriptors and encoded into trained codebooks inspired by classical bag of words model. We evaluate our methods on the Medieval Paleographical Scale (MPS) data set and perform dating by writer identification and classification. As far as dating by writer identification is concerned, we arrive at the conclusion that features which perform well for writer identification, are not necessary suitable for historical document dating. Experimental results of dating by classification demonstrate that a combination of kCF and kSF achieves optimal results.

5.1 Introduction

Handwritten historical documents are the most important sources of information about the past, especially where the more distant past is concerned, before the wide spread dissemination of printing and semi-mechanical text production. Increasing numbers of such documents are currently being digitized and stored in the computer, as in the Monk system (Van der Zant et al., 2008), which contains more than 100K scanned page images. Thanks to this development, pattern recognition techniques can now be applied to solve historical document problems, which has already been attempted at length in the case of writer identification (Brink et al., 2012) Arabadjis et al., 2013 and word spotting (Van Oosten and Schomaker, 2014).
Rusinol et al., 2015 These methods aim to provide efficient tools for scholars in the humanities to discover informative patterns in large digital collections. The Monk system (Van der Zant et al., 2008), providing a web-based search engine for characters and words annotation, recognition and retrieval, can serve as an example.

We have proposed a number of features (Brink et al., 2012; Schomaker and Bulacu, 2004; Bulacu and Schomaker, 2007) to capture handwriting styles. However, there is one aspect of the visual appearance of handwritten samples that has not been addressed yet. In Fig. 1.8 a sample is shown. As we can see, the visual appearance is dominated by long curved stroke elements crossing other ink stroke traces in an irregular manner. Such a complicated thread structure was not covered by the proposed junction feature in Chapter 4 nor by other methods (Brink et al., 2012; Schomaker and Bulacu, 2004; Bulacu and Schomaker, 2007). In addition, the existing methods concern low-level features, which cannot capture the properties of mid-level graphemes or stroke information. The research questions then are as follows: (1) How to define a feature that addresses the aspect of style at intermediate scale? (2) Which type of properties of handwritten strokes in historical documents contain the temporal information that can be used for dating? (3) What degree of feature complexity is required to obtain the optimal year estimation performance?

In this chapter, we propose a family of local contour and stroke features and their application to historical document image dating. These features are small fragments of contours and strokes, called \( k \) Contour Fragments (\( k \)CF) and \( k \) Stroke Fragments (\( k \)SF), respectively. The fragments in \( k \)CF are the contour fragments resulting from a combination of a number of \( k \) consecutive primary fragments generated by the discrete contour evolution (DCE) (Latecki and Lakämper, 1999) and the fragments in \( k \)SF form a segment of length \( k \) of a stroke fragment graph (SFG). The larger the number \( k \) of contour and stroke fragments in \( k \)CF and \( k \)SF, the more complex the contour and stroke fragment structures it can capture. We use the relative coordinates of the fragment points of \( k \)CF as the feature vector and use the junction feature to describe the \( k \)SF.

The proposed \( k \)CF and \( k \)SF can be considered as grapheme-based representations and have several attractive properties: (1) \( k \)CF and \( k \)SF cover short contour and stroke fragments of the connected components in handwritten documents, which are probably shared between different characters and allographes. The statistical distribution of these small fragments can capture the handwriting style of historical documents; (2) for a certain range of \( k \), both \( k \)CF and \( k \)SF can discover the meaningful and intermediate complexity patterns in a large connected component which may span several lines due to touching ascenders and descenders in cursive handwriting; (3) the descriptors of the \( k \)CF and \( k \)SF are insensitive to the scale and rotation of document images, which are very important properties in historical document analysis because historical documents are often digitized with different resolutions and font sizes in different documents are also different, making them sensitive to scale and rotation.

Inspired by the bag-of-words model (Csurka et al., 2004), we construct codebooks of \( k \)CF and \( k \)SF with different complexity degrees \( k \), each of which capture statistical information with different degrees of complexity of local fragments. All the \( k \)CF and \( k \)SF detected
5.2. \textit{k} Contour Fragments (\textit{kCF})

The contours of handwritten texts encapsulate the handwriting style and a wide variety of approaches have been proposed to extract features on writing contours, such as the \textit{CO}$^3$ (Schomaker and Bulacu, 2004), chain codes (Siddiqi and Vincent, 2010) and contour fragments (Ghiasi and Safabakhsh, 2013). In this section, we propose a novel framework to extract contour fragments, called \textit{k} Contour Fragments (\textit{kCF} for short), on contours of handwritten texts in historical document images. Our method is more flexible and insensitive to scale and rotation transform. The computational procedure will be presented in the following sections.

5.2.1 Detecting \textit{kCF}

Contours are first extracted by the contour tracing method proposed in (Brink et al. 2012), which extracts 8-connected circular trajectories of black pixels that are adjacent to white pixels on the binary image. Key points which have a higher curvature on a contour are detected by the discrete contour evolution (DCE) approach (Latecki and Lakämper 1999) and the contour can be approximately represented by a polygon with these key points as
Historical Manuscript Dating Using Contour and Stroke Fragments.

Figure 5.2: Examples of contour fragments with different contour complexity degrees $k$ extracted from the contour in Fig. 5.1. The bold parts are the new added contour fragments when $k$ grows.

vertices. We denote the detected key points as:

$$\vec{p} = \{p_1, p_2, \ldots, p_T\}$$

(5.1)

where $T$ is the number of vertices and can be controlled by a threshold in the DCE method. Fig. 5.1 shows an example of detected key points (the points within the circles) on the contour of a connected component.

The method proposed in \cite{Wang et al., 2014} collects contour fragments between every pair of key points on the shape contour. However, we think that the context around key points (which are high curvature points) contains useful information about the handwriting style. In order to maintain the informative context around key points, we define break points $\vec{b} = \{b_1, b_2, \ldots, b_T\}$ as the midpoints along the contour between two consecutive key points: the point $b_i$ is the middle point on the contour fragment beginning at point $p_i$ and end at point $p_{i+1}$. Fig. 5.1 shows an example of break points (the points within the rectangles).

Given the contour and break points $\vec{b} = \{b_1, b_2, \ldots, b_T\}$, primitive contour fragments can be obtained by segmenting the contour between pairs of consecutive break points $(b_i, b_j)$, which are the short-range contour fragments. The long-range contour fragments can be obtained by concatenating $k$ consecutive primitive contour fragments, which refers to $k$ Contour Fragments ($k$CF). Fig. 5.2 shows $k$CF extracted from the contour in Fig. 5.1. From the figure
5.2. k Contour Fragments (kCF)

Figure 5.3: An example of end point selection in a kCF. The points $p_1$ and $p_2$ are two end points and the point $m$ is the midpoint. We select the starting endpoint $p_2$ if $e_{p_2} < e_{p_1}$.

we can see that as $k$ grows, more and more complex and informative contour fragments can be obtained.

5.2.2 Describing kCF

It is important to develop a proper way to describe the detected informative kCF to facilitate comparing. The shape context (Belongie et al., 2002) is used in (Wang et al., 2014) to describe contour fragments based on 5 reference points sampled equidistantly on the normalized contour fragments. However, determining the size of the shape context is arbitrary. In order to achieve the scale-invariant property, we use the relative coordinates of the fragment points as the feature vector, following the methods in (Schomaker and Bulacu, 2004; Ghiasi and Safabakhsh, 2013). Each contour fragment in a kCF is resampled such that it contains $N_c$ coordinate points and then they are normalized to an origin of $(0, 0)$ and a standard deviation of radius 1 by:

\[
\begin{align*}
\vec{x} &\leftarrow (\vec{x} - \mu_x)/\sigma_x \\
\vec{y} &\leftarrow (\vec{y} - \mu_y)/\sigma_y
\end{align*}
\]  

(5.2)

where $\vec{x}$ and $\vec{y}$ are the collections of $x$ and $y$ coordinates of a contour fragment, $\mu_x$ and $\mu_y$ are averages of the $\vec{x}$ and $\vec{y}$ coordinates of the contour fragments and the $\sigma_x$ and $\sigma_y$ are the corresponding standard deviations. The final feature vector contains the normalized $N_c$ $\vec{x}$ and $\vec{y}$ values and the dimension of the feature vector is $2N_c$.

There are two endpoints in each contour fragment ($p_1$ and $p_2$ in Fig. 5.3) and two feature vectors can be produced by starting at different endpoints. In order to make the final feature vector insensitive to the starting point, we carefully select the starting endpoint as follows. First, we find the midpoint $M = (x_m, y_m)$ of the contour fragment and the normalized distance
of the pixels in each branch to the midpoint is given by:

\[
e_{p1} = \sum_{i=1}^{m} (|x_i| + |y_i|) \\
e_{p2} = \sum_{i=m+1}^{N} (|x_i| + |y_i|)
\]

(5.3)

where \(N\) is the number of points on the contour fragment. We select the starting endpoint \(p\) of the branch with the minimal value \(e_p\).

Given a document from the MPS data set, we extract the contour fragments and use the proposed description method to represent the contour fragments. Fig. 5.4 shows four randomly selected contour fragments with 4CF and contour fragments on each row are found by the \(K\) nearest neighbor method with the Euclidean distance function, from which we can conclude that similar contour fragments may be from the same character or may be shared between different characters. Therefore the detected contour fragments can capture local contour structures and are informative and repeatable as well.

Our proposed method is different from the method proposed in (Ghiasi and Safabakhsh, 2013), in which contour fragments with a specific length or number of points are extracted from contours, making the extracted contour fragments sensitive to image scaling. The proposed \(k\)CF is scale-invariant because key points detected by DCE are insensitive to scale changes. A connected component in historical documents may span several words or even several lines due to the touching strokes. Therefore, the \(CO^3\) (Schomaker and Bulacu, 2004) extracted on these large connected components are sensitive to the touching strokes, making them non-repeatable. Our proposed \(k\)CF can solve such problem and is robust and more flexible than the \(CO^3\).
5.2.3 Encoding $k$CF

The detected $k$CF can be considered as basic handwriting contours and the probability distribution of $k$CF can characterize the handwriting style. We construct codebooks for $k$CF with different $k$ using clustering methods. It has been shown in [Bulacu and Schomaker, 2005] that the same performance was obtained for k-means, 1D Kohonen Self-Organizing Map (SOM) [Kohonen, 1988] and 2D SOM clustering methods. In this chapter, we use the standard 2D SOM clustering method to train codebooks for $k$CF with Euclidean distance. Finally, one feature vector can be obtained for one document image and the dimension of the feature vector is determined by the size of the codebook.

5.3 $k$ Stroke Fragments ($k$SF)

In general, handwritten characters are written by one or several strokes and the writing style can be represented by structures or shapes of strokes. In this section, we present three crucial steps to extract, describe and encode handwritten stroke fragments in document images.

5.3.1 Detecting $k$SF

In the literature, the term “stroke” in handwritten documents is used in slightly different ways. In on-line handwriting, strokes are determined by the velocity of the movement of the pen, or the writing speed [Schomaker and Teulings, 1990]. In this case, strokes are “the pieces of handwriting movement bounded by minima in the tangential pen-tip velocity [Schomaker, 1993]”. That also means “a stroke is a trace of pen-tip movement which starts at pen-down and ends at pen-up [Kato and Yasuhara, 2000]”. In order to provide clarity about the way the term “stroke” is used in this chapter, we define the stroke in off-line handwritten documents as:

**Definition 1:** A stroke is a connected component of an ink trace which has two end points (one corresponds to the pen-down point and another to the pen-up point) on the stroke skeleton line.

One exception of this definition is the circle stroke, in which there are no end points (the skeleton line is also a circle). In order to integrate such circle strokes into our definition, we regard the left-most point in the skeleton line as the shared end points [Schomaker and Bulacu, 2004].

In a cursive handwritten document touching characters often form a large connected and complex structure and there is no obvious way to dissect it into stroke fragments. Fig. 5.5 gives an example of one connected component of the ink trace. The skeleton line of the connected component can be computed by thinning methods and there are two types of feature points on the skeleton line: end points and fork points. An end point refers to the beginning or end of a stroke, and a fork point (see an example in Fig. 5.5) is the location
5. Historical Manuscript Dating Using Contour and Stroke Fragments.

![Image of a connected component and its corresponding stroke fragment graph (SFG).]

Figure 5.5: The left figure shows an example of a connected component in a historical document. The white line is the skeleton line of the ink, black points are the fork points and end points. The connected component can be decomposed into seven parts segmenting at the fork points. The right figure shows the corresponding stroke fragment graph (SFG).

where at least two strokes meet \cite{Liu1999}. Similar graph structures have been used for the temporal reconstruction of strokes from a static image \cite{Kato2000}.

In this chapter, we consider fork points as the shared end points between touching strokes. Thus, the connected component can be decomposed into “strokes” segmenting at fork points, yielding stroke fragments between end points and fork points according to definition 1 and these are called primary stroke fragments. For example, Fig. 5.5 shows a connected component with five end points and three fork points, and seven primary stroke fragments can be obtained, which are denoted by numbers 1 to 7. We refer to these stroke fragments as primitive stroke fragments because they are the minimal fragments which can be segmented from the connected component according to definition 1.

This segmentation method is simple, intuitive and independent from any line detection or segmentation methods. However, it also yields fragments which are so small (especially the fragments between two fork points) that they become meaningless and can in some cases be regarded as noise (for example the 4th and 5th stroke fragments in Fig. 5.5). In order to detect longer and more complex stroke fragments which are more informative, we build a stroke fragment graph (SFG) inspired by \cite{Ferrari2006,Ferrari2008} as follows. Each node in the SFG corresponds to a primary stroke fragment and two nodes are linked if the two primary stroke fragments connect to each other, which means they share at least one fork point. Fig. 5.5 shows the SFG built from the primary stroke fragments in Fig. 5.5. The SFG reflects the relationship of connections between primitive stroke fragments of one connected component.

One important observation is that any connected sub-graph in the SFG without loops corresponds to a stroke according to our stroke definition 1. For example, the sub-graph
5.3. $k$ Stroke Fragments (kSF)

Figure 5.6: Stroke fragments of 3SF generated in the SFG in the right figure of Fig. 5.5. The corresponding nodes from left to right are: \{1,2,4\}, \{1,3,4\}, \{1,5,6\}, \{1,5,6\}, \{(2,4,5)\}, \{(3,4,5)\}, \{(4,5,6)\}, \{(4,5,7)\}.

containing nodes \{1, 4, 2\} in the SFG in Fig. 5.5 can form a stroke which has two end points. In contrast, the sub-graph containing nodes \{2, 3, 4\}, which contains a loop, does not correspond to an effective stroke, because it has three end points and cannot be drawn in one time. We refer to strokes which contain a number of $k$ primary stroke fragments (the length of the path between two vertexes in the SFG) as $k$ stroke fragments or $k$SF. When $k = 1$, 1SF are primitive stroke fragments. As $k$ grows, more and more complex and informative strokes can be obtained. Fig. 5.6 gives an example of stroke fragments detected in the SFG in Fig. 5.5 when $k = 3$ (3SF). In practice, given the value of $k$, all the connected paths without loops can be efficiently computed using the depth-first search method on the SFG.

5.3.2 Describing $k$SF

We use the junction feature proposed in Chapter 4 to describe $k$SF. The computation of the junction feature is as follows: given a reference point $p_i = (x, y)$ and a direction $\phi$, the distance from $p_i$ to the ink boundary, called partial length $d_p(\phi)$, can be easily computed by searching the ink pixels following a ray in the direction $\phi$ (Epstein et al., 2010). A simple and efficient algorithm based on Bresenham’s algorithm (Hearn and Baker, 1997) is used to compute the distance from $p_i$ to the ink boundary inspired by (Brink et al., 2012). The end point $p_e = (x_e, y_e)$ is computed by

$$
x_e = x + m \cdot \cos(\phi) \\
y_e = y + m \cdot \sin(\phi)
$$

(5.4)

where the parameter $m$ determines the maximum partial length or the maximum search space from $p_i$ to $p_e$. An approximated linear path from $p_i$ to $p_e$ is constructed and the background point $p_b = (x_b, y_b)$ is found by tracing points starting from $p_i$ towards to the end point $p_e$. The partial length is measured using a simple Euclidean distance:

$$
d_p(\phi) = \sqrt{(x - x_b)^2 + (y - y_b)^2}
$$

(5.5)

(More details of the computation of $d_p(\phi)$ can be found in (Brink et al., 2012) and in Chapter 4).
5. Historical Manuscript Dating Using Contour and Stroke Fragments.

A partial length distribution is built on the reference point \( p_i \) by computing the partial length in every direction \( \phi \) in a discrete set \( D = \left\{ \frac{2\pi k}{N}; k = 0, \cdots, N - 1 \right\} \), where \( N \) is the number of directions we consider. This distribution is considered as the junction distribution of the point \( p_i \), which is a local descriptor. Fig. 5.7 shows two examples of the junction descriptors on the reference points in stroke fragments. Finally, the descriptor is normalized in order to make it scale-invariant. The junction descriptor is a rich descriptor, especially when the reference points lie on the fork points. In this case, it reflects the junction structure information in handwritten strokes, such as the radius and the number of branches of the junction region (Parida et al., 1998) (see example of Fig. 5.7).

The features of each \( kSF \) are computed as follows: \( N_r \) reference points on the skeleton line of \( kSF \) are sampled equidistantly and described by the junction descriptor. Finally, these \( N_r \) junction descriptors are concatenated into one feature vector to describe the corresponding \( kSF \). In principle, the large number of \( N_r \) leads to a rich descriptor. However, when the \( N_r \) is too larger, the descriptor contains too much redundant information and the dimension of the descriptor is also high which needs a lot of computational time. In practice, we suggest the \( N_r \in [5, 10] \). Fig. 5.8 gives an example of this method with 5 sample points.

In order to make \( kSF \) invariant to rotation, a relative horizontal direction should be used instead of the absolute horizontal direction in order to construct the junction feature on each
5.3. k Stroke Fragments (kSF)

Figure 5.9: A number of similar stroke fragments with $k = 1$ (1SF) detected in documents in the MPS data set. The red lines are the skeleton lines and white points are the sampled reference points of junction descriptors.

The relative horizontal direction can be estimated by averaging the tangent angles of sampled points. Fig. 5.8 shows an example of the estimated relative direction.

Fig. 5.9 shows a number of stroke fragments with $k = 1$ (1SF), which is also known as Strokelets (He and Schomaker, 2015). Similar to $k$CF, $k$SF are also informative and repeatable and can be considered as mid-level representations.

As a grapheme-based method, our proposed $k$SF has several advantages: (1) Compared to the Junclets (proposed in Chapter 4), the $k$SF captures the stroke properties in a large area and can be considered as a macro mid-level feature. (2) Compared to the Fraglets (Bulacu and Schomaker, 2007), our proposed $k$SF is easy to compute. Most importantly, the $k$SF is a script-independent grapheme-based method which can be used in any script. The descriptor of the $k$SF reflects the stroke properties, such as stroke width and stroke structures, which are lost in other methods (Schomaker and Bulacu, 2004; Bulacu and Schomaker, 2007; Siddiqi and Vincent, 2010).

5.3.3 Encoding $k$FS

In order to build a global feature representation for a historical document image, all $k$SF extracted from the image are mapped into a common space (named codebook) using the bag-of-words model (Csurka et al., 2004). As discussed in (Bulacu and Schomaker, 2007), there is no difference existed between the performance of the codebooks trained by K-means, Kohonen SOM 1D and Kohonen SOM 2D. Similar to $k$CF, we use the Kohonen SOM 2D method (Kohonen, 1988) to train the codebook.
5.4 Experiments

5.4.1 Experimental settings

In the computation of the $k_{CF}$ and $k_{FS}$, a binary method is needed to obtain the binary document image and compute contours and skeleton lines of the ink traces. Although several binarization methods have been proposed in the literature, such as (Moghaddam and Cheriet, 2012), we apply the simple and efficient Otsu threshold algorithm (Otsu, 1975) in our experiments, followed by the guided filter (He et al., 2013) to remove noise and make contours smooth. Each contour fragment of $k_{CF}$ is resampled to contain 100 points and the feature dimension is $100 \times 2 = 200$. The number of directions of the junction descriptor $N$ is set to 120, which is the dimension of the junction descriptor. In this chapter, 10 points are sampled on each stroke fragment and each point is described by a junction descriptor. Therefore, the dimension of $k_{FS}$ is $120 \times 10 = 1200$.

We employed two widely used measures for performance evaluation: the Mean Absolute Error (MAE) and Cumulative Score (CS) (Geng et al., 2007). The MAE is a Manhattan-type distance, which is typically defined as:

$$MAE = \frac{\sum_{i=1}^{N} |K(y_i) - \overline{K(y_i)}|}{N}$$  \hspace{1cm} (5.6)

where $K(y_i)$ is the ground-truth of the input document $y_i$ and $\overline{K(y_i)}$ is the estimated key year, while $N$ is the number of test documents. The Cumulative Score (CS) is typically defined as (Geng et al., 2007):

$$CS(\alpha) = \frac{N_{e \leq \alpha}}{N} \times 100\%$$  \hspace{1cm} (5.7)

where $N_{e \leq \alpha}$ is the number of test images on which the key year estimation makes an absolute error $e$ no higher than the acceptable error level: $\alpha$ years. For historians, an error of $\pm 25$ is, more often than not, acceptable when dating historical documents. Therefore, we report the Cumulative Score with error level $\alpha = 25$ years in the experiments.

5.4.2 Historical document dating by general handwriting style identification

As we mentioned before, writing charters in the Middle Ages was a profession and the number of scribes simultaneously active in each city was limited. Therefore, an undated document can be dated by identifying the writer. This is reasonable because if we know the writer and his active period, the date of the document can be directly obtained (Panagopoulos et al., 2009; Arabadjis et al., 2013). We conduct experiments on writer identification on the MPS data set as well as historical document dating by handwriting style identification.

The writers of some charters are known in MPS and others are not. We term the subset of documents with writers who produced as least two samples as MPS-writer known with
Table 5.1: The performance of writer identification and dating by handwriting style identification in terms of MAEs and CS(α = 25) of the \( k \)CF, \( k \)SF and other features.

<table>
<thead>
<tr>
<th>Method</th>
<th>Writer identification</th>
<th>Dating by writer identification (KNN)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( \chi^2 ) difference using the K nearest neighbors (KNN)</td>
<td>( K=5 )</td>
</tr>
<tr>
<td></td>
<td>( \text{MAEs} )</td>
<td>( \text{CS}(\alpha = 25) )</td>
</tr>
<tr>
<td>Quill</td>
<td>61.7</td>
<td>82.2</td>
</tr>
<tr>
<td>Hinge</td>
<td>71.8</td>
<td>85.9</td>
</tr>
<tr>
<td>Junclets</td>
<td>59.9</td>
<td>79.3</td>
</tr>
<tr>
<td>2CF</td>
<td>37.6</td>
<td>73.6</td>
</tr>
<tr>
<td>3CF</td>
<td>42.9</td>
<td>77.9</td>
</tr>
<tr>
<td>4CF</td>
<td>45.3</td>
<td>77.9</td>
</tr>
<tr>
<td>5CF</td>
<td>48.6</td>
<td>78.2</td>
</tr>
<tr>
<td>1SF</td>
<td>64.3</td>
<td>84.6</td>
</tr>
<tr>
<td>2SF</td>
<td>56.6</td>
<td>78.8</td>
</tr>
<tr>
<td>3SF</td>
<td>47.6</td>
<td>71.3</td>
</tr>
</tbody>
</table>

We perform writer identification on the MPS-WKM data set with \( \chi^2 \) difference using the K nearest neighbors (KNN) method, following [Bulacu and Schomaker, 2007, Siddiqi and Vincent, 2010]. We utilize the “leave-one-out” strategy which is widely used for writer identification: taking the query document out and sorting the rest of the documents according to the distance function to output a hit list. The query document is recognized as the writer of the document on the top-\( x \) of the hit list, corresponding to the top-\( x \) performance. Usually, the Top-1 and Top-10 performances are reported.

We also carry out historical document dating by general handwriting style identification. The combined MPS-WKM and MPS-WKS data sets with writer labels are considered as the reference data set. For each undated document in the MPS-WU data set, we find the K nearest neighbors using KNN in the reference data set and we assign the year to the undated document as the most represented years within the K nearest neighbors.

**Performance of writer identification and dating**

In this section, we present the performance of our proposed methods for writer identification and dating. We explore the degrees of complexity \( k \in \{2, 3, 4, 5\} \) for \( k \)CF and \( k \in \{1, 2, 3\} \) for \( k \)SF. We do not consider 1CF because they contain less discriminative information as their lengths are too small. The feature dimensions of \( k \)CF and \( k \)SF are discussed in Section 5.4.3. Table 5.1 shows the performance of \( k \)CF and \( k \)SF for writer identification and dating, as well as Hinge [Bulacu and Schomaker, 2007], Quill [Brink et al., 2012] and Junclets (proposed in Chapter 4), from which we can conclude that the writer identification rates increase for \( k \)CF while they decrease for \( k \)SF when \( k \) grows. A similar trend can be found for the dating performance. The writer identification performances of \( k \)SF are better than \( k \)CF, except 3SF and 5C, while the dating performances of \( k \)SF are worse than \( k \)CF, for all \( k \). We can also find
that Hinge achieves the best performance for writer identification and 3CF achieves the best performance for dating.

One interesting observation is that writer identification results of $k$CF are worse than with all other features (except 3SF), while its dating results are better than all other ones. The Hinge feature achieves the best performance for writer identification, while the dating performance is worse than Junclets, $k$CF($k = 2, 3, 4, 5$) and $k$SF($k = 1, 2$). We can obtain the conclusion that: Features which achieve a good performance on writer identification are not necessarily suitable for historical document dating via writer identification when there exists no sample for a target writer in the training set. The main reason is that dating requires features to capture the general writing style in a certain period whereas writer identification needs features to capture the writing style characteristic for individuals precisely.

From Table 5.1 we can also find that for features which are good in writer identification, the dating performance increases when $K$ of KNN decreases, such as in the Hinge, Quill, Junclets, 1SF and 2SF features. However, for $k$CF, the best dating performances are mostly achieved when $K = 20$.

In practice, we have found that combining the $k$CF and $k$SF do not improve the performance for both writer identification and dating. Therefore, their results are not reported in this chapter.

5.4.3 Historical document dating by classification

The dating problem can be considered as either a classification or a regression problem. In this chapter, we regard it as a classification problem because the document distribution in our data set over the period of 1300-1550 CE has an obvious border between nearby key years. All the documents from each key year form a class and there are 11 classes which correspond to the 11 key years in the MPS data set. We train 11 corresponding classifiers using a linear SVM (LIBSVM in this thesis) with a one-versus-all strategy and the undated document is assigned to the key year which has the maximum value of the 11 softmax output scores. The parameter $C$ of the linear SVM is estimated by a grid search method. We split the data set into training (70%) and testing (30%) sets. The experiment is repeated 20 times and the average results are reported together with the standard deviation in the following experiments.

We consider two different evaluation scenarios for historical document dating. In the first one, we carefully split the data set into training and testing subsets to make sure that the same writer never appears in both training and test sets, which means that all documents from the same hand should be only in the training set or only in the test set. For documents without writer labels, we randomly split them into the training and test set. We term this scenario as excluding writer duplicates or wr.excl. for short. In the second scenario, we randomly split the data set into training and test sets without considering writer labels. We term this scenario as including writer duplicates or wr.incl. for short. In the wr.excl. scenario, the system performs the dating based on the general writing style built by other writers. However, in the
5.4. Experiments

Table 5.2: MAEs and CS($\alpha = 25$) of the $k_{CF}$ and $k_{SF}$.

<table>
<thead>
<tr>
<th>Method</th>
<th>wr.excl. scenario</th>
<th>wr.incl. scenario</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MAEs</td>
<td>CS($\alpha = 25$)</td>
</tr>
<tr>
<td>2CF</td>
<td>26.7±3.9</td>
<td>76.0±4.3%</td>
</tr>
<tr>
<td>3CF</td>
<td>23.8±2.1</td>
<td>80.9±2.4%</td>
</tr>
<tr>
<td>4CF</td>
<td>22.8±2.7</td>
<td>80.7±3.8%</td>
</tr>
<tr>
<td>5CF</td>
<td><strong>21.7±2.8</strong></td>
<td><strong>82.0±3.6%</strong></td>
</tr>
<tr>
<td>1SF</td>
<td>22.1±2.9</td>
<td>79.8±3.1%</td>
</tr>
<tr>
<td>2SF</td>
<td><strong>18.9±2.0</strong></td>
<td><strong>84.3±3.0%</strong></td>
</tr>
<tr>
<td>3SF</td>
<td>23.8±3.0</td>
<td>78.9±3.0%</td>
</tr>
</tbody>
</table>

wr.incl. scenario, the processing of writer identification is probably involved in the dating.

**Performance of $k_{CF}$ and $k_{SF}$**

Table 5.2 shows the performance of historical document dating in terms of MAEs and CS($\alpha = 25$) of the $k_{CF}$ and $k_{SF}$ in the wr.excl. and wr.incl. scenarios. The codebook sizes of $k_{CF}$ and $k_{SF}$ are set to $50 \times 50$ and $30 \times 30$, respectively. The selection of sizes is discussed in the next section. From the table we can find that for $k_{CF}$, the MAEs decreases when $k$ increases and the 5CF performs best. The MAE of 5CF is lower than 2CF by 5 and 4.4 years in the wr.excl. and wr.incl. scenarios, respectively. The same trend is also found in terms of CS($\alpha = 25$) and 82.8±3.6% documents are correctly estimated with error level no higher than 25 years in the wr.excl. scenario and the corresponding percentage in the wr.incl. scenario is 88.4±1.6%. The results demonstrate that $k_{CF}$ with a higher $k$ in a certain range offer informative, repeatable and discriminative contour fragments which capture the handwriting style in historical documents.

From the results of the three degrees of $k_{SF}$ complexity in Table 5.2 we find that 2SF performs best overall. The average MAEs of the 2SF are 18.9/11.1 (for the wr.excl./wr.incl. scenarios) versus 22.1/12.6 and 23.8/15.1 of 1SF and 3SF, respectively. The CS($\alpha = 25$) scores of 2SF in the two scenarios are also higher than the ones of 1SF and 3SF. The following order can be obtained: 2SF > 1SF > 3SF, by ranking $k_{SF}$ according to the average MAEs and CS($\alpha = 25$) scores. The performance of 3SF is even worse than 1SF and the reason may be that 3SF contains too much artificial stroke fragments (see Fig. 5.6).

From Table 5.2 we also find that the performance of 2SF is better than 5CF by 2.8 and 1.8 years in terms of MAEs in the wr.excl. and wr.incl. scenarios, respectively. The descriptors of $k_{SF}$ do not only contain the curvature information of strokes, but also the stroke length distribution which reflects the stroke width and stroke distribution around sample points and the informative and discriminative information contained in the stroke fragments can be found by SVM.
The effect of codebook size

In this section, we conduct experiments to evaluate the performance of historical document dating by classification with different sizes of codebooks of the $k_{CF}$ and $k_{SF}$. Fig. 5.10 and Fig. 5.11 show the results of the $k_{CF}$ and $k_{SF}$, respectively. The two figures show that the MAEs of both $k_{CF}$ and $k_{SF}$ decrease as the size of the codebook increases.

The left figure in Fig. 5.10 shows the performance of $k_{CF}$ with $k = 2, 3, 4, 5$ in the $\text{wr.excl.}$ scenario. The best performances are achieved for $k_{CF}$ with a codebook size of $50 \times 50$, except the $2CF$ with $40 \times 40$. The right figure in Fig. 5.10 shows the MAEs of $k_{CF}$ with $k = 2, 3, 4, 5$ in the $\text{wr.incl.}$ scenario and the lowest MAEs are obtained when the codebook size is $50 \times 50$. Therefore, the size of the codebook of $k_{CF}$ is set to $50 \times 50$ for $k = 2, 3, 4, 5$ in both the $\text{wr.excl.}$ and the $\text{wr.incl.}$ scenarios in the following experiments.

Similarly, the left and right figures in Fig. 5.11 show the MAEs of $k_{SF}$ ($k = 1, 2, 3$) in
5.4. Experiments

the wr.excl. and wr.incl. scenarios, respectively. From the two figures we can find that the best performance is achieved with a codebook size of $30 \times 30$.

Performance of combined $k$CF and $k$SF

In this section, we evaluate performances when using several degrees of $k$CF and $k$SF simultaneously in the feature space. Table 5.3 gives the results of combined $k$CF and $k$SF in both the wr.excl. and wr.incl. scenarios. Generally, the $k$CF and $k$SF combined achieve better results than each $k$ of the $k$CF and $k$SF separately. In the wr.excl. scenario, the $\{2345\}$CF achieves the lowest MAE (19.2 years), which is better than other combinations. Although the best performance in term of MAE is obtained by $\{345\}$CF in the wr.incl. scenario, there is no obvious difference between the performance of $\{345\}$CF and $\{2345\}$CF and the CS($\alpha = 25$) score of $\{2345\}$CF is higher than the one of $\{345\}$CF. Comparing the results of Table 5.3 with the ones of Table 5.2, we find that the combination of $k$CF improves the best performance of single $k$CF from 21.7 to 19.2 (MAE) and from 82.0% to 85.8% (CS($\alpha = 25$)) in the wr.excl. scenario. Correspondingly, in the wr.incl. scenario, the best performance is improved from 12.9 to 10.7 (MAE) and from 88.4% to 90.8% (CS($\alpha = 25$)).

Although the performance of 3SF is worse than 1SF and 2SF, combining it with $\{12\}$SF achieves the best results, which demonstrates that 3SF can provide some useful information discovered by SVM. Comparing Table 5.3 with Table 5.2, the MAEs and CS($\alpha = 25$) in the wr.excl. and wr.incl. scenarios are improved by 1.5/2.5%, 1.2/1.7%, respectively.

We also combine $\{2345\}$CF and $\{123\}$SF together and the results are shown in the bottom row of Table 5.3. The combined performance outperforms all individual features ($\{2345\}$CF and $\{123\}$SF) involved in the combination. The MAEs of the combined $\{2345\}$CF and $\{123\}$SF are 14.9 and 7.9 in the wr.excl. and wr.incl. scenarios, respectively, which are the best ones among all the combinations. The results demonstrate that the $k$CF and $k$SF capture different types of information about handwriting styles and combining them can improve performance.

Comparison with other features

In Table 5.4, we present the performances of other existing features, such as the Quill (Brink et al., 2012), Hinge (Bulacu and Schomaker, 2007) and Junclets (proposed in Chapter 4). From Table 5.4 we can see that the performances of $\{2345\}$CF, $\{123\}$SF and the combined $\{2345\}$CF and $\{123\}$SF are better than performance of Quill, Hinge and Junclets.

In practice, we have found that there is no significant difference between the combination of $\{2345\}$CF and $\{123\}$SF and the combination of $\{2345\}$CF and $\{123\}$SF with Quill, Hinge and Junclets. The main reason is that $k$CF captures curvature information of contours with Quill and Hinge that is similar to the stroke structures captured by $k$SF with Junclets. In fact, $k$SF contains junction information because we consider fork points as the shared end points and descriptors of these end points are included in $k$SF. Furthermore, the proposed $k$CF and $k$SF are more flexible and insensitive to the scale and rotation transform. Fig. 5.12
### 5. Historical Manuscript Dating Using Contour and Stroke Fragments

Table 5.3: MAEs and CS(\(\alpha = 25\)) scores of \(k\)CF and \(k\)SF combined.

<table>
<thead>
<tr>
<th>Method</th>
<th>wr.excl. scenario MAEs</th>
<th>wr.excl. scenario CS((\alpha = 25))</th>
<th>wr.incl. scenario MAEs</th>
<th>wr.incl. scenario CS((\alpha = 25))</th>
</tr>
</thead>
<tbody>
<tr>
<td>(2+3)CF</td>
<td>22.9±3.2</td>
<td>80.8±3.2%</td>
<td>14.2±0.9</td>
<td>87.4±1.8%</td>
</tr>
<tr>
<td>(3+4)CF</td>
<td>22.4±3.3</td>
<td>81.7±3.5%</td>
<td>12.1±0.9</td>
<td>89.4±1.5%</td>
</tr>
<tr>
<td>(4+5)CF</td>
<td>20.3±2.9</td>
<td>83.4±3.3%</td>
<td>11.8±0.8</td>
<td>89.9±1.5%</td>
</tr>
<tr>
<td>(2+3+4)CF</td>
<td>21.5±3.1</td>
<td>82.4±4.5%</td>
<td>12.0±0.8</td>
<td>89.2±1.3%</td>
</tr>
<tr>
<td>(3+4+5)CF</td>
<td>20.0±2.9</td>
<td>83.6±3.2%</td>
<td><strong>10.7±1.1</strong></td>
<td>90.5±1.9%</td>
</tr>
<tr>
<td>(2+3+4+5)CF</td>
<td><strong>19.2±3.5</strong></td>
<td><strong>85.8±2.8%</strong></td>
<td>10.8±0.9</td>
<td><strong>90.8±1.1%</strong></td>
</tr>
<tr>
<td>(1+2)SF</td>
<td>18.6±2.3</td>
<td>84.5±3.6%</td>
<td>10.1±0.7</td>
<td>91.2±1.3%</td>
</tr>
<tr>
<td>(1+2+3)SF</td>
<td><strong>17.4±1.9</strong></td>
<td><strong>86.8±2.0%</strong></td>
<td><strong>9.9±0.6</strong></td>
<td><strong>91.8±1.5%</strong></td>
</tr>
<tr>
<td>(1+2+3)SF+(2+3+4+5)CF</td>
<td><strong>14.9±1.7</strong></td>
<td><strong>89.2±2.4%</strong></td>
<td><strong>7.9±1.0</strong></td>
<td><strong>93.2±1.3%</strong></td>
</tr>
</tbody>
</table>

Table 5.4: MAEs and CSs of the combination of other features with the proposed \(k\)CF and \(k\)SF.

<table>
<thead>
<tr>
<th>Method</th>
<th>wr.excl. scenario MAEs</th>
<th>wr.excl. scenario CS((\alpha = 25))</th>
<th>wr.incl. scenario MAEs</th>
<th>wr.incl. scenario CS((\alpha = 25))</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quill (Brink et al., 2012)</td>
<td>23.7±2.9</td>
<td>80.6±3.0%</td>
<td>12.1±0.9</td>
<td>89.5±1.3%</td>
</tr>
<tr>
<td>Hinge (Bulacu and Schomaker, 2007)</td>
<td>22.1±2.9</td>
<td>80.6±3.1%</td>
<td>12.2±0.9</td>
<td>89.6±1.3%</td>
</tr>
<tr>
<td>Junclets</td>
<td>21.5±3.3</td>
<td>81.9±3.9%</td>
<td>12.0±0.7</td>
<td>89.2±1.4%</td>
</tr>
<tr>
<td>(2+3+4+5)CF</td>
<td>19.2±3.5</td>
<td>85.8±2.8%</td>
<td>10.8±0.9</td>
<td>90.8±1.1%</td>
</tr>
<tr>
<td>(1+2+3)SF</td>
<td>17.4±1.9</td>
<td>86.8±2.0%</td>
<td>9.9±0.6</td>
<td>91.8±1.5%</td>
</tr>
<tr>
<td>(1+2+3)SF+(2+3+4+5)CF</td>
<td><strong>14.9±1.7</strong></td>
<td><strong>89.2±2.4%</strong></td>
<td><strong>7.9±1.0</strong></td>
<td><strong>93.2±1.3%</strong></td>
</tr>
</tbody>
</table>

Figure 5.12: CS curves of the error level from 0 to 100 years of different methods applied to the MPS data set in the wr.excl. (the left figure) and wr.incl. (the right figure) scenarios. Note that the ranges of CS axes are different between two figures in order to make curves clear.
shows the CS curves of Quill, Hinge and Junclets and the proposed \{2345\}CF and \{123\}SF combined. From the figure we can find that the CS curve of our proposed method is above that of Quill, Hinge and Junclets and our proposed method improves performance, especially when the error level is small (\( \alpha \leq 50 \)).

5.5 Discussion and conclusion

We have introduced the \( k \)CF and \( k \)SF family of contour and stroke fragment features and applied them to historical document dating based on the MPS data set. The \( k \)CF and \( k \)SF are scale and rotation invariant grapheme-based features which can capture the handwriting style of handwritten documents. We approached dating in two ways: by handwriting style identification and by classification. Concerning dating by handwriting style identification, we found that features which achieve good performance for writer identification, are not suitable for historical document dating by handwriting style identification by means of writer identification when there is no duplicated document existed in the training set. For example, \( k \)CF performed worse for writer identification than other methods but better than others for dating.

As for dating by classification is concerned, we evaluated the performance of the proposed \( k \)CF and \( k \)SF in two scenarios: excluding writer duplicates (\( \text{wr.excl.} \)) and including writer duplicates (\( \text{wr.incl.} \)) and experimental results demonstrated that a combination of \( k \)CF and \( k \)SF achieves state-of-the-art results on the MPS data set. Several interesting conclusions can be drawn from our experimental results. First, the performance of \( k \)CF increases with an increasing complexity \( k \). However, with a large \( k \), the \( k \)CF may contain long contour fragments which are not informative or repeatable in the document images. This is also true for \( k \)SF and 2SF performs better than either 1SF or 3SF. Secondly, \( k \)CF and \( k \)SF contain different information. \( k \)CF captures the curvature information under different scales which contains both local (small \( k \)) and intermediate (large \( k \)) contour information of the handwriting style, while \( k \)SF captures the stroke structure caused by both the writing instrument and handwriting style. Therefore, only by combining them we achieved an optimal performance.

The proposed features are extracted based on binarized images. However, obtaining a very good binarization is a challenging problem for historical manuscripts with a high degradation. Therefore, our proposed \( k \)CF and \( k \)SF might be very sensitive to the quality of historical manuscripts. In the next chapter, we will present a novel feature vector, which is robust to the quality of historical manuscripts. In addition, we will investigate the codebook trained in a supervised way, which can discover the correlations between the low-level visual elements and their labels.