

ALBERT-LUDWIGS-UNIVERSITÄT
FREIBURG
INSTITUT FÜR INFORMATIK

Lehrstuhl für Mustererkennung und Bildverarbeitung
Prof. Dr. Hans Burkhardt



Automatische Erkennung von handgeschriebenen Worten

mithilfe des

Level-building Algorithmus

Studienarbeit

Rudolph Triebel

September 1999 – Dezember 1999

Erklärung

Hiermit erkläre ich, daß die vorliegende Arbeit von mir selbständig und nur unter Verwendung der aufgeführten Hilfsmittel erstellt wurde.

Freiburg, den 20. Dezember 1999

Inhaltsverzeichnis

1. Einleitung	4
2. Modellierung	5
2.1. Der Dynamic Time Warping Algorithmus	5
2.1.1. Zeitliche Angleichung durch Warpingfunktionen	5
2.1.2. Klassifikation anhand des optimalen Warpingpfades	5
2.1.3. Mögliche Einschränkungen der Warpingfunktionen	7
2.1.4. Effiziente Berechnung	9
2.2. Der Level-building Algorithmus	11
2.2.1. Motivation	11
2.2.2. Das Verfahren	11
2.2.3. Komplexität	17
3. Realisierung	19
3.1. Implementation	19
3.1.1. Berechnung der Distanzen	19
3.1.2. Aufbau der Warpingmatrix	20
3.1.3. Auslagerung der Distanzberechnung	20
3.2. Der geeignete Referenzdatensatz	20
3.3. Extraktion der Merkmale	21
3.4. Untersuchung einer Umgebung von \mathcal{T}	23
4. Ergebnisse	24
4.1. Wahl einer geeigneten Evaluationsfunktion	25
4.2. Aufbau der Versuchsreihen	26
4.3. Visualisierung	27
4.4. Auswertung	29
5. Zusammenfassung und Ausblick	29
A. Klassifikationsergebnisse	31
A.1. Referenzdatensatz R_1	31
A.2. Referenzdatensatz R_2	32
A.3. Referenzdatensatz R_3	33
A.4. Referenzdatensatz R_4	34

In der vorliegenden Arbeit soll ein Verfahren zur automatischen Erkennung von handgeschriebenen Wörtern vorgestellt werden. Dabei wird der, nach dem Prinzip des zeitlichen Angleichens (*time warping*) aufgebaute, *Level-building Algorithmus* eingesetzt. Die für die Implementation relevanten Schritte werden erläutert und die dabei zu beachtenden problemspezifischen Anpassungen aufgezeigt. Besonderes Augenmerk wird auf die Wahl des Referenzdatensatzes und die Extraktion der für die Klassifikation geeigneten Merkmale gerichtet. Schließlich werden die Ergebnisse von verschiedenen Versuchsreihen gegenübergestellt und die so gewonnenen Erkenntnisse ausgewertet.

1. Einleitung

Die automatische Erkennung von Handschriften ist ein Teilgebiet der Mustererkennung, welches seit Anfang der 60er Jahre erforscht wird. Die Motivation für die Suche nach einem Verfahren zur Handschriftenerkennung ist vor allem die Möglichkeit, Texte auf bequeme Art einlesen zu können. Dabei soll die Tastatureingabe nicht ersetzt, sondern lediglich ergänzt werden. Denkbare Anwendungsgebiete sind zum Beispiel die Korrektur von Texten oder die Eingabe eines Textes in einer Sprache mit sehr vielen Schriftzeichen, wie beispielsweise Chinesisch, für die eine Tastatur sehr unhandlich wäre.

Man unterteilt die Handschriftenerkennung in die beiden Bereiche *On-line-Erkennung* und *Off-line-Erkennung*, welche sich in ihren Ansätzen grundlegend unterscheiden. Bei der On-line-Erkennung wird der zu erkennende Text meist mithilfe eines elektronischen Schreibbretts aufgenommen. Dabei können, außer den x - und y -Koordinaten des Stiftes, verschiedene weitere Informationen, wie zum Beispiel die Winkel zwischen Stift und Schreiboberfläche, zu jedem Abtastzeitpunkt ermittelt werden (für eine eingehendere Untersuchung des On-Line-Verfahrens siehe auch [4]). Die Off-line-Erkennung dagegen benötigt zur Schriftaufnahme kein spezielles Werkzeug, sondern basiert auf der Extraktion der Schrift aus einem digital eingelesenen Bild. Der Vorteil hierbei ist, daß ein gewöhnliches Schreibgerät benutzt werden kann, jedoch ist die Erkennung im allgemeinen schwieriger zu realisieren.

Desweiteren wird die Handschriftenerkennung hinsichtlich ihrer Anwendungsgebiete aufgeteilt in die Bereiche *Buchstabenerkennung* und *Worterkennung*. Falls die Schrift in vorgegebenen Bereichen buchstabenweise vorliegt, wie zum Beispiel bei einem ausgefüllten Formular, welches einzelne Kästchen vorgibt, so kann die Erkennung auf Buchstabenebene erfolgen. Dies schafft einen erheblichen Vorteil, da das Problem der Segmentierung nicht gelöst werden muß. Im Fall der Worterkennung liegt der Text meist als Aneinanderreihung von nicht segmentierten Worten vor, welche in der Regel aus Groß- und Kleinbuchstaben bestehen können. Dies erschwert die korrekte Erkennung zusätzlich.

Das Verfahren, das mit der vorliegenden Arbeit vorgestellt werden soll, arbeitet nach dem Prinzip der On-line-Erkennung von handgeschriebenen Worten. Dabei sollen zu-

nächst im Abschnitt zur Modellierung die beiden zugrundeliegenden Algorithmen erläutert werden. Danach folgen Ausführungen zur praktischen Realisierung und anschließend die Auswertung der Ergebnisse. Zum Schluß wird die Arbeit zusammengefaßt und ein Ausblick auf zukünftige Untersuchungen gegeben.

2. Modellierung

2.1. Der Dynamic Time Warping Algorithmus

Der Dynamic Time Warping-Algorithmus (DTW) ist eine Methode zur Berechnung eines Distanzmaßes $D(\mathcal{T}, \mathcal{R})$ von zwei, in Dauer und zeitlichem Ablauf voneinander abweichenden, Mustersequenzen \mathcal{T} und \mathcal{R} .

2.1.1. Zeitliche Angleichung durch Warpingfunktionen

Es sei mit $\mathcal{T} = (\mathbf{t}_1, \mathbf{t}_2, \dots, \mathbf{t}_M)$ und $\mathcal{R} = (\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_N)$ ein Test- und ein Referenzmuster gegeben, wobei M bzw. N die Test- bzw. Referenzmusterlänge angibt. Die \mathbf{t}_m bzw. \mathbf{r}_n sind hierbei Merkmalsvektoren beliebiger Dimension. In dem hier zu betrachtenden Fall der Handschriftenerkennung können dies zum Beispiel die x - und y -Koordinaten des Stiftes zu einem Zeitpunkt m , die relativen oder absoluten Winkel des zu einem Zeitpunkt m geschriebenen Geradenteilstücks, oder die Steigungen der Tangenten im Punkt (x_m, y_m) sein.

Zur Klassifikation von \mathcal{T} führt der DTW-Algorithmus einen Vergleich mit \mathcal{R} aus. Dazu müssen zunächst korrespondierende Merkmalsvektoren beider Muster einander zugeordnet werden. Dies geschieht mithilfe zweier *Warpingfunktionen* ϕ_t und ϕ_r , die jedem $k = 1, \dots, T$ auf einer gemeinsamen Zeitachse diejenigen Indizes $m_k \in \{1, 2, \dots, M\}$ bzw. $n_k \in \{1, 2, \dots, N\}$ von \mathcal{T} bzw. \mathcal{R} , zuordnen, deren Merkmale bezüglich einer lokalen Metrik d miteinander zu vergleichen sind. T gibt demnach die Länge der gemeinsamen Zeitachse an. Zunächst sollen keinerlei Anforderungen an ϕ_t und ϕ_r gestellt werden. Später wird gezeigt werden, inwieweit es sinnvoll sein kann, bestimmte Einschränkungen vorzunehmen. Die Abbildungen 1(a) und 1(b) zeigen ein Beispiel für zwei derartige Warpingfunktionen.

2.1.2. Klassifikation anhand des optimalen Warpingpfades

Man kann den Vergleich der beiden Muster \mathcal{T} und \mathcal{R} derart sichtbar machen, daß man in einer zweidimensionalen Matrix die Merkmale des Testmusters gegen die des Referenzmusters aufträgt und zu jedem Zeitpunkt k die Matrixpunkte (m_k, n_k) markiert¹, wobei

¹Man beachte, daß die Indizierung der Matrixelemente hier nicht wie mathematischen Sinne erfolgt, also zuerst Zeilen- dann Spaltenindex. Vielmehr gibt der erste Index die Spalte und der zweite Index die Zeile an, wobei die Zeilen von unten nach oben gezählt werden. Diese Konvention gilt für sämtliche im Folgenden auftretenden Matrizen.

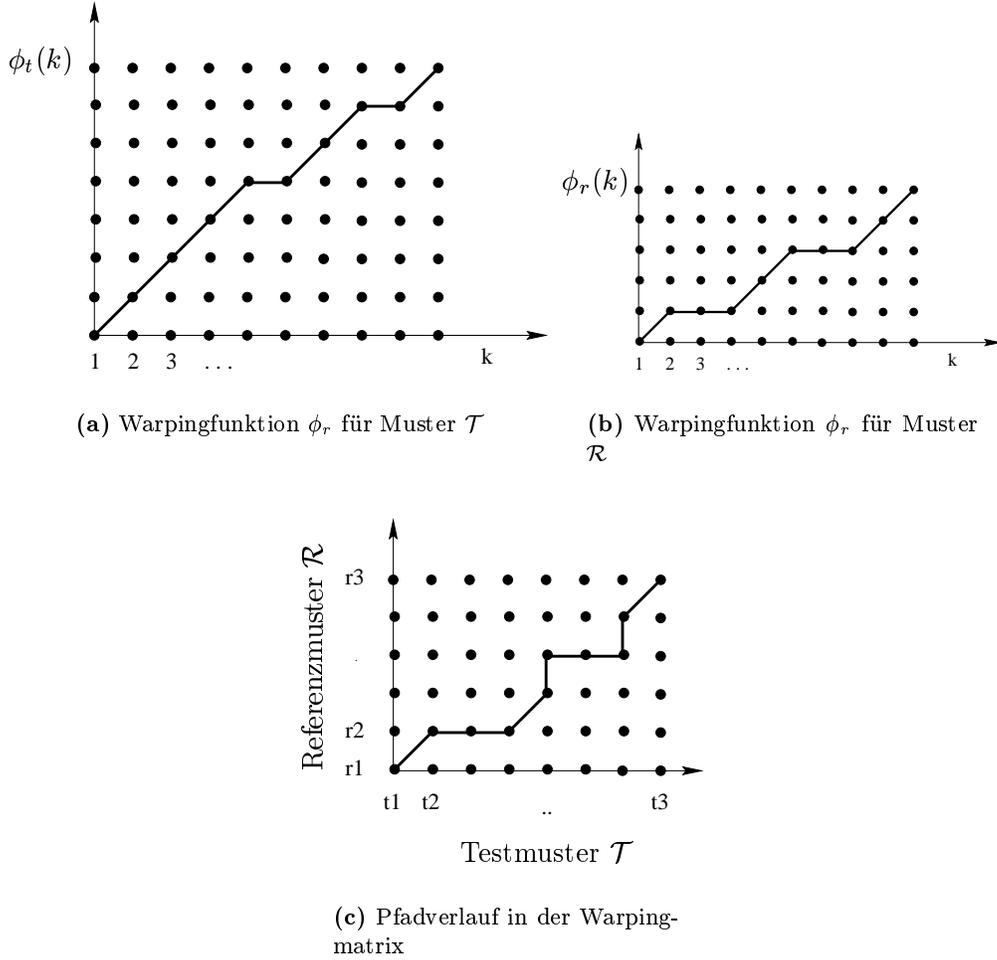


Abbildung 1: Beispiel zweier Warpingfunktionen (Abb.1(a) und 1(b)). Auf der Abszisse ist jeweils die gemeinsame Zeitachse k , auf der Ordinate die musterspezifische Zeitachse m bzw. n aufgetragen. Abbildung 1(c) zeigt die resultierende Warpingmatrix mit dem entsprechenden Warpingpfad.

gilt: $m_k = \phi_t(k)$ und $n_k = \phi_r(k)$. Durch Verbinden aufeinanderfolgender Matrixpunkte (m_k, n_k) und (m_{k+1}, n_{k+1}) entsteht ein *Warpingpfad* durch die Matrix (siehe Abb. 1(c)).

Man definiert nun, aufbauend auf der Metrik d für die Merkmalsvektoren, zu jedem Warpingpfad $\phi = (\phi_t, \phi_r)$ ein Kostenmaß D_ϕ wie folgt:

$$D_\phi(\mathcal{T}, \mathcal{R}) = \sum_{k=1}^T d(\mathbf{t}_{\phi_t(k)}, \mathbf{r}_{\phi_r(k)}) \quad (1)$$

Als lokale Metrik d kann zum Beispiel das Quadrat des Euklidischen Abstandes verwendet werden. Den Warpingpfad $\hat{\phi}$ mit minimalem Kostenmaß $D_{\hat{\phi}}$ bezeichnet man als *optimal*. Der Abstand $D(\mathcal{T}, \mathcal{R})$ zwischen zwei beliebigen Mustern wird nun als die Pfadkosten des optimalen Warpingpfades erklärt:

$$D(\mathcal{T}, \mathcal{R}) = \min_{\phi} \{D_\phi(\mathcal{T}, \mathcal{R})\} \quad (2)$$

Trägt man für jedes $m = 1, \dots, M$ und jedes $n = 1, \dots, N$ die Kosten des optimalen Warpingpfades bis zum Punkt (m, n) in eine Matrix D ein, so heißt D eine *Warpingmatrix*.

Zur Klassifikation bei einer gegebenen Menge $R = \{\mathcal{R}_1, \mathcal{R}_2, \dots, \mathcal{R}_V\}$ von Referenzmustern (Klassen) wird nun ein Testmuster \mathcal{T} derjenigen Klasse v^* zugeteilt, deren Abstand zu \mathcal{T} minimal ist (Minimum-Abstands-Klassifikator), also für die gilt:

$$D(\mathcal{T}, \mathcal{R}_{v^*}) = \min_{v=1 \dots V} \{D(\mathcal{T}, \mathcal{R}_v)\} \quad (3)$$

2.1.3. Mögliche Einschränkungen der Warpingfunktionen

Bei genauerem Betrachten der bisher genannten Eigenschaften von ϕ_t und ϕ_r fällt auf, daß, wenn keinerlei Einschränkungen an ϕ_t und ϕ_r vorgenommen werden, der optimale Warpingpfad eine unerwünschte Angleichung von \mathcal{T} und \mathcal{R} darstellt. Zum Beispiel gilt im Falle, daß \mathcal{T} das zu \mathcal{R} gespiegelte Muster ist, daß der optimale Warpingpfad $\hat{\phi}$ auf der Diagonalen von links oben nach rechts unten durch die Warpingmatrix, verläuft und die einzelnen lokalen Distanzen entlang $\hat{\phi}$ verschwinden. Es gilt dann also $D(\mathcal{T}, \mathcal{R}) = 0$, obwohl \mathcal{T} und \mathcal{R} im allgemeinen nicht der gleichen Klasse zuzuordnen sind.

Um derartige Fehlklassifikationen zu vermeiden, werden an ϕ_t und ϕ_r (und damit auch an jeden Warpingpfad ϕ) unter anderem folgende Forderungen gestellt (hierbei sei mit D die Warpingmatrix bezeichnet):

Endpunktbeschränkung. Die Endpunkte von ϕ sollten mit der linken unteren bzw. rechten oberen Ecke von D übereinstimmen (also: $\phi(1) = (1, 1)$ bzw. $\phi(T) = (N, M)$). Dadurch soll verhindert werden, daß ein Anfangs- oder ein Endstück eines Musters abgeschnitten wird.

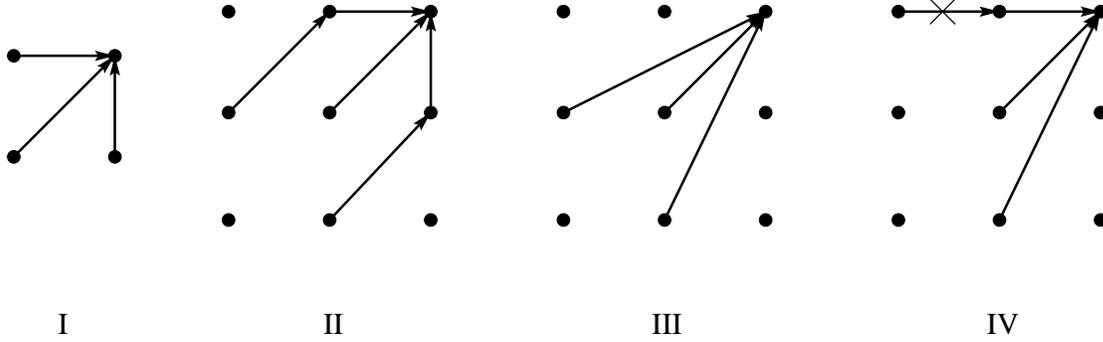


Abbildung 2: Vier Beispiele für Beschränkungen der lokalen Übergänge eines Warpingpfades. Version I läßt nur einzelne Schritte zu, die Versionen II und III betrachten eine größere Nachbarschaft, wobei im einen Fall ein Zwischenschritt eingefügt wird, im anderen nicht. Version IV schließlich läßt weder Schritte nach oben, noch Doppelschritte nach rechts zu.

Monotonie. Die Warpingfunktionen ϕ_t und ϕ_r sollten monoton steigend sein. Dies impliziert auch die Monotonie von ϕ und erzwingt einen Pfadverlauf in chronologischer Abfolge.

Lokale Stetigkeit. Die lokalen Übergänge von ϕ sollten gewissen Stetigkeitsansprüchen genügen. Zu große Sprünge in D sollen vermieden werden (in Abb. 2 sind einige denkbare Möglichkeiten zur Einschränkung von ϕ dargestellt).

Globale Einschränkung. Der Pfadverlauf kann global eingeschränkt werden, zum Beispiel durch Definition eines legalen Bereiches innerhalb von D , welchen ϕ nicht verlassen darf. Dadurch werden Pfade ausgeschlossen, die eine extreme zeitliche Stauchung bzw. Streckung darstellen (siehe Abb. 3).

Gewichtung der Pfadübergänge. Die lokalen Pfadübergänge können mit einem Gewichtungsfaktor versehen werden. Die Idee dabei ist, daß der optimale Warpingpfad bestimmte Richtungen bevorzugen soll.

Für eine detailliertere Untersuchung der einzelnen Pfadbedingungen sei auf [3] verwiesen. Im Folgenden sollen lediglich die drei erstgenannten Bedingungen gelten, wobei speziell die lokalen Übergänge entsprechend Modell I in Abb. 2 gewählt werden. Die Motivation hierfür ist zum einen die Forderung, daß keines der Merkmale beider Muster ausgelassen werden soll, und zum anderen daß die Indizes der zu vergleichenden Merkmale um nicht mehr als einen Schritt auseinanderliegen sollen. Diese Überlegungen sind rein heuristischer Natur und entbehren jeglichen Anspruchs auf Allgemeingültigkeit.

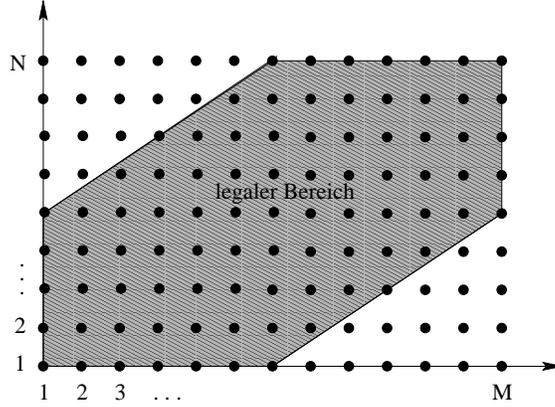


Abbildung 3: Beispiel für eine globale Einschränkung des Warpingpfades. Durch zwei Grenzgeraden wird ein Bereich definiert, den der Pfad nicht verlassen darf.

2.1.4. Effiziente Berechnung

Da die Anzahl der möglichen Warpingpfade exponentiell mit der Referenz- bzw. Testmusterlänge wächst, wäre eine Klassifikation auf der Basis der Minimierung über alle diese Pfade recht ineffizient. Es kann jedoch aufgrund der folgenden Überlegung eine effiziente Berechnungsvorschrift gefunden werden:

Wir betrachten den vereinfachten Fall der monotonen Warpingfunktionen mit den lokalen Übergängen aus Abschnitt 2.1.3 (Modell I). Es bezeichne D die Warpingmatrix, also $D_{MN} = D(\mathcal{T}, \mathcal{R})$. Weiterhin sei $\mathcal{T}^{(m)} = (\mathbf{t}_1, \dots, \mathbf{t}_m)$ das Präfix von \mathcal{T} der Länge m (analog für $\mathcal{R}^{(n)}$). Aus den Bedingungen für die lokalen Pfadübergänge folgt, daß der vorletzte Punkt (\hat{m}, \hat{n}) des optimalen Warpingpfades $\hat{\phi}$ nur einer der Punkte $(M, N-1)$, $(M-1, N)$ oder $(M-1, N-1)$ sein kann. Da $\hat{\phi}$ optimal ist, kann dieser vorletzte Punkt in $\hat{\phi}$ nur derjenige sein, dessen Eintrag in D minimal ist, es muß also gelten:

$$D_{\hat{m}\hat{n}} = \min\{D_{M(N-1)}, D_{(M-1)N}, D_{(M-1)(N-1)}\} \quad (4)$$

Nun gilt aber, daß $D_{\hat{m}\hat{n}}$ den Kosten des optimalen Warpingpfades bis zum Punkt (\hat{m}, \hat{n}) entspricht, denn wenn es einen günstigeren Pfad nach (\hat{m}, \hat{n}) gäbe, so wäre $\hat{\phi}$ nicht optimal (siehe hierzu Abbildung 4). Das bedeutet, daß sich die akkumulierte Distanz entlang $\hat{\phi}$ wie folgt aus den genannten Matrixeinträgen (und damit aus den akkumulierten Teildistanzen) berechnen läßt:

$$D(\mathcal{T}, \mathcal{R}) = \min \left\{ \begin{array}{l} D(\mathcal{T}^{(M-1)}, \mathcal{R}^{(N)}), \\ D(\mathcal{T}^{(M)}, \mathcal{R}^{(N-1)}), \\ D(\mathcal{T}^{(M-1)}, \mathcal{R}^{(N-1)}) \end{array} \right\} + d(\mathbf{t}_M, \mathbf{r}_N) \quad (5)$$

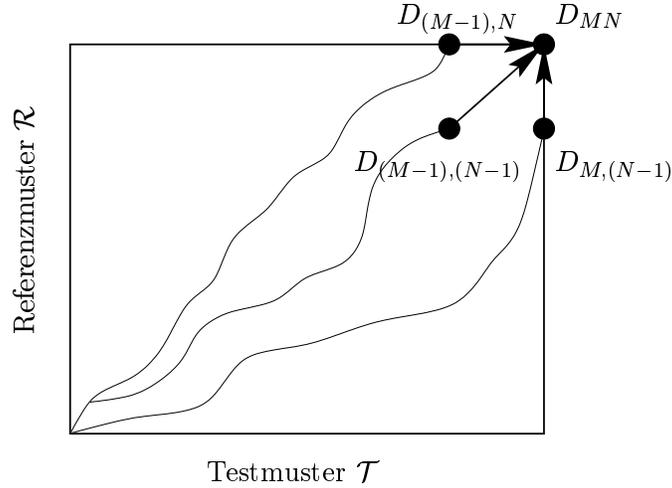


Abbildung 4: Mögliche Vorgänger des letzten Punktes im optimalen Warpingpfad $\hat{\phi}$. Die akkumulierte Distanz entlang $\hat{\phi}$ läßt sich berechnen aus der Summe der akkumulierten Distanz bis zu diesem vorletzten Punkt und der lokalen Distanz in (M, N) .

Diese Gleichung kann nun rekursiv auf die akkumulierten Teildistanzen angewendet werden, und zwar so lange, bis das erste Element eines der Muster erreicht ist (dies ist dann die erste Zeile oder die erste Spalte der Warpingmatrix). Von dort gibt es nur noch eine Möglichkeit, zum Anfangspunkt zu gelangen, sodaß folgende Rekursionsgleichung aufgestellt werden kann:

$$D(\mathcal{T}^{(1)}, \mathcal{R}^{(1)}) = d(\mathbf{t}_1, \mathbf{r}_1) \quad (6)$$

$$D(\mathcal{T}^{(1)}, \mathcal{R}^{(m)}) = D(\mathcal{T}^{(1)}, \mathcal{R}^{(m-1)}) + d(\mathbf{t}_1, \mathbf{r}_m) \quad (7)$$

$$D(\mathcal{T}^{(n)}, \mathcal{R}^{(1)}) = D(\mathcal{T}^{(n-1)}, \mathcal{R}^{(1)}) + d(\mathbf{t}_n, \mathbf{r}_1) \quad (8)$$

$$D(\mathcal{T}^{(m)}, \mathcal{R}^{(n)}) = \min \left\{ \begin{array}{l} D(\mathcal{T}^{(m-1)}, \mathcal{R}^{(n)}), \\ D(\mathcal{T}^{(m)}, \mathcal{R}^{(n-1)}), \\ D(\mathcal{T}^{(m-1)}, \mathcal{R}^{(n-1)}) \end{array} \right\} + d(\mathbf{t}_m, \mathbf{r}_n) \quad (9)$$

Gleichungen (6)-(9) können nun benutzt werden, um einen Algorithmus nach dem Prinzip der Dynamischen Programmierung zu formulieren (siehe Algorithmus 1). Eine eingehendere Untersuchung von Algorithmen, die durch Dynamisches Programmieren realisiert werden können, ist in [1] zu finden.

Algorithmus 1 DTW-Algorithmus**Eingabe:** Testmuster \mathcal{T} , Referenzmuster \mathcal{R} **Ausgabe:** DTW-Matrix D

```

 $D_{11} \leftarrow d(\mathbf{t}_1, \mathbf{r}_1);$ 
 $D_{m1} \leftarrow D_{(m-1)1} + d(\mathbf{t}_m, \mathbf{r}_1) \quad \forall m = 2, \dots, M;$ 
 $D_{1n} \leftarrow D_{1(n-1)} + d(\mathbf{t}_1, \mathbf{r}_n) \quad \forall n = 2, \dots, N;$ 
for  $m = 2 \dots M$ ,  $n = 2 \dots N$  do
   $D_{mn} \leftarrow \min\{D_{m(n-1)}, D_{(m-1)n}, D_{(m-1)(n-1)}\} + d(\mathbf{t}_m, \mathbf{r}_n);$ 
end for
return  $D$ 

```

2.2. Der Level-building Algorithmus**2.2.1. Motivation**

Bei der Beschreibung des DTW-Algorithmus wurde bisher noch nichts über die genaue Repräsentationsform der zu erkennenden Testmuster ausgesagt. Diese hängt natürlich vom Anwendungsgebiet des Verfahrens ab. Zum Beispiel bestünde für den Fall der Erkennung von segmentierten Buchstaben ein Testmuster aus einer Sequenz von Merkmalen eines Buchstabens. Für die Klassifikation würde man den zu erkennenden Buchstaben mit jedem Element einer geeignet zu wählenden Menge von Referenzbuchstaben nach dem DTW-Algorithmus vergleichen und entsprechend Gleichung (3) der Klasse mit minimaler Distanz zuordnen.

Besteht das Testmuster jedoch aus einer Kette von nicht segmentierten Buchstaben, also einem Wort, so müsste der Referenzdatensatz entweder ebenfalls aus ganzen Wörtern bestehen, oder das Testmuster müsste mithilfe eines geeigneten Algorithmus vorsegmentiert und anschließend auf Buchstabenebene klassifiziert werden. Da im allgemeinen die Anzahl möglicher Referenzwörter sehr groß ist und eine Vorsegmentierung algorithmisch schwierig zu realisieren ist, sollte für den Fall der Worterkennung ein Verfahren gefunden werden, welches Buchstaben als Referenzmuster verwendet und keine Vorsegmentierung benötigt. Im Folgenden wird mit dem Level-building Algorithmus (in der Literatur oft auch mit LB bezeichnet) ein derartiges Verfahren beschrieben werden, wobei detailliertere Betrachtungen hierzu in [2] zu finden sind.

2.2.2. Das Verfahren

Als Testmuster \mathcal{T} sei ein handgeschriebenes Wort gegeben. Zu diesem soll nun ein *Super-Referenzmuster* \mathcal{R}^s gefunden werden, welches aus einer bestimmten Anzahl von Referenzmustern zusammengesetzt ist und bezüglich des in Abschnitt 2.1.2 definierten Distanzmaßes D minimalen Abstand zu \mathcal{T} hat. Bei einer gegebenen Menge $R = \{\mathcal{R}_1, \mathcal{R}_2, \dots, \mathcal{R}_V\}$ von Referenzmustern sind also eine Zahl L und Indizes v_1, v_2, \dots, v_L gesucht, sodaß gilt:

$$\mathcal{R}^s = \mathcal{R}_{v_1} \oplus \mathcal{R}_{v_2} \oplus \dots \oplus \mathcal{R}_{v_L} =: \mathcal{R}_{v_1, \dots, v_L} \quad (10)$$

und

$$D(\mathcal{T}, \mathcal{R}^s) = D(\mathcal{T}, \mathcal{R}_{v_1, \dots, v_L}) = \min_{u_1, \dots, u_L \in \{1, \dots, V\}} D(\mathcal{T}, \mathcal{R}_{u_1, \dots, u_L}) \quad (11)$$

Die naive Vorgehensweise, alle möglichen Superreferenzmuster \mathcal{R}^s zu erstellen und mit \mathcal{T} zu vergleichen, wäre jedoch aufgrund der exponentiell ansteigenden Anzahl möglicher Superreferenzmuster nicht effizient. Im folgenden wird gezeigt, wie eine effiziente Berechnung durch stufenweisen Aufbau der Warpingmatrix erfolgen kann.

Stufenweise Berechnung der Warpingmatrix Es folgen zunächst einige Notationen, die für die algorithmische Beschreibung des Verfahrens benötigt werden:

- N_v ist die Länge des v -ten Referenzmusters.
- $\psi(l) = \sum_{i=1}^l N_{v_i}$ ist die Gesamtlänge der ersten l Referenzmuster in $\mathcal{R}_{v_1}^s$ (demnach ist $\psi(L)$ die Gesamtlänge von \mathcal{R}^s).
- $D_l(m, n) = D(\mathcal{T}^{(m)}, \mathcal{R}_{v_1 \dots v_l}^{(\psi(l-1)+n)})$ für $m = 1, \dots, M$, $n = 1, \dots, N_{v_l}$ und $l = 1, \dots, L$ ist die akkumulierte Distanz entlang des optimalen Warpingpfades zwischen dem Präfix von \mathcal{T} der Länge m und dem Präfix von $\mathcal{R}_{v_1 \dots v_l}$ der Länge $\psi(l-1) + n$
- $\tilde{D}_l(m) = D_l(m, N_{v_l}) = D(\mathcal{T}^{(m)}, \mathcal{R}_{v_1 \dots v_l})$ ist die akkumulierte Distanz zwischen dem Muster $\mathcal{R}_{v_1 \dots v_l}$ und dem Präfix von \mathcal{T} der Länge m .

Falls ein Superreferenzmuster \mathcal{R}^s mit minimalem Abstand zu \mathcal{T} bekannt ist, so kann die Berechnung des optimalen Warpingpfades zwischen \mathcal{T} und \mathcal{R}^s unter Verwendung des DTW-Algorithmus stufenweise erfolgen (siehe Algorithmus 2).

Man beachte die Erweiterung der Warpingmatrix um eine nullte Zeile und eine nullte Spalte. In die nullte Zeile einer jeden Stufe werden jeweils die akkumulierten Distanzen der vorhergehenden Stufe kopiert. Auf der ersten Stufe wird diese Zeile mit ∞ vorinitialisiert, ebenso wie die nullte Spalte der gesamten Matrix. Lediglich der Punkt $(0, 0)$ wird auf Null gesetzt, um dadurch den Punkt $(1, 1)$ als erstes Element des Warpingpfades zu erzwingen. Die Vorgehensweise des Verfahrens wird durch Abbildung 5 veranschaulicht. Beispielhaft sind für jede Stufe einige der möglichen Pfade eingezeichnet.

Algorithmus 2 Stufenweise Berechnung von $D(\mathcal{T}, \mathcal{R}^s)$

Eingabe: Testmuster \mathcal{T} , Superreferenzmuster \mathcal{R}^s

Ausgabe: Akkumulierte Distanz entlang des optimalen Warpingpfades

Initialisiere $\tilde{D}_0(m) \leftarrow \begin{cases} 0 & \text{für } m = 0 \\ \infty & \text{für } m \neq 0 \end{cases}$

for $l = 1, \dots, L$ **do**

$D_l(m, 0) \leftarrow \tilde{D}_{l-1}(m) \quad \forall m = 0, \dots, M$

$D_l(0, n) \leftarrow \infty \quad \forall n = 1, \dots, N_{v_l}$

for $m = 1, \dots, M, n = 1, \dots, N_{v_l}$ **do**

$D_l(m, n) \leftarrow \min\{D_l(m-1, n), D_l(m, n-1), D_l(m-1, n-1)\} + d(\mathbf{t}_m, \mathbf{r}_n)$

end for

$\tilde{D}_l(m) \leftarrow D_l(m, N_{v_l}) \quad \forall m = 1, \dots, M$

end for

return $\tilde{D}_L(M)$

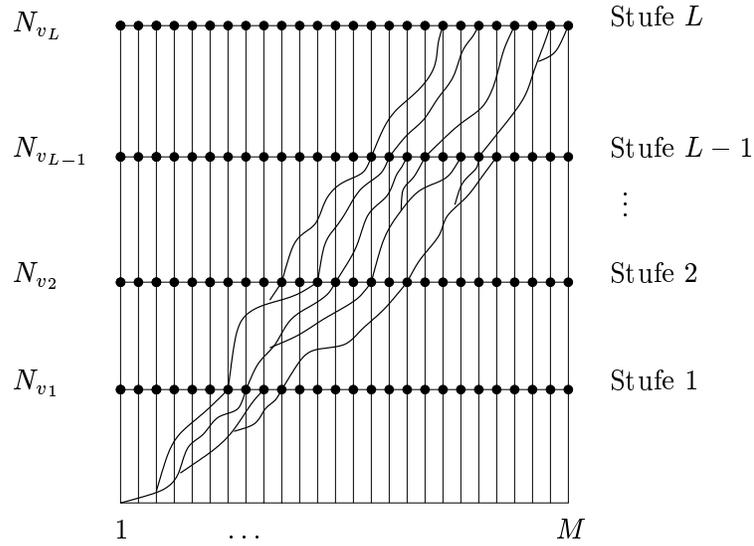


Abbildung 5: Schema der stufenweisen Berechnung von $D(\mathcal{T}, \mathcal{R}^s)$

Bestimmung der Superreferenzmusters \mathcal{R}^s mit kleinster Warpingdistanz Das Problem der Suche nach einem Superreferenzmuster mit minimaler Distanz zu \mathcal{T} ist natürlich durch Algorithmus 2 noch nicht gelöst. Insbesondere sind die genaue Länge L und die Indizes v_1, \dots, v_L der Referenzmuster in \mathcal{R}^s zu bestimmen. Nehmen wir zunächst an, L sei bekannt und betrachten hierbei zunächst den Fall $L = 1$, so kann die Ermittlung der Distanz durch Minimierung über die Distanzen aller Referenzmuster \mathcal{R}_v mit $v \in \{1, \dots, V\}$ nach Gleichung (3) erfolgen. Definiert man nun mit $\tilde{D}_1^v(m) = D(\mathcal{T}^{(m)}, \mathcal{R}_v)$ für $m = 1, \dots, M$ den Zeilenvektor der akkumulierten Distanzen zwischen $\mathcal{T}^{(m)}$ und \mathcal{R}_v , so kann Gleichung (3) umgeschrieben werden zu:

$$D(\mathcal{T}, \mathcal{R}_{v^*}) = \min_v \tilde{D}_1^v(M) = \tilde{D}_1^*(M) \quad (12)$$

mit $\tilde{D}_1^*(m) := \min_v \tilde{D}_1^v(m)$ für $m = 1, \dots, M$.

Sei nun $L > 1$ und es bezeichne $\mathcal{T}^{(m \dots M)} := (\mathbf{t}_m, \dots, \mathbf{t}_M)$ das Suffix von \mathcal{T} der Länge $M - m + 1$. Dann gilt für den optimalen Warpingpfad $\hat{\phi}$, daß jeder Teilpfad von $\hat{\phi}$ ebenfalls optimal sein muß, denn sonst wäre die Optimalität von $\hat{\phi}$ verletzt. Daraus folgt, daß die akkumulierte Distanz entlang $\hat{\phi}$ gleich der Summe der akkumulierten Distanzen entlang der Teilpfade ist.

Bezeichnet man mit e_l die Spalte der Warpingmatrix, an welcher $\hat{\phi}$ die oberste Zeile der Stufe l schneidet und teilt man nun $\hat{\phi}$ derart in zwei Teilpfade ϕ_1 und ϕ_2 der Länge \bar{t} bzw. $T - \bar{t} + 1$ auf, sodaß gilt:

$$\begin{aligned} \phi_1(t) &= \phi(t) \quad \forall t = 1, \dots, \bar{t} \\ \phi_2(t) &= \phi(t + \bar{t} - 1) \quad \forall t = 1, \dots, T - \bar{t} + 1 \end{aligned} \quad (13)$$

und

$$\phi_1(\bar{t}) = (e_{L-1}, \psi(L-1)) = \phi_2(1) \quad (14)$$

so folgt wegen der Optimalität von ϕ_1 und ϕ_2 :

$$D(\mathcal{T}, \mathcal{R}_{v_1 \dots v_L}) = D(\mathcal{T}^{(e_{L-1})}, \mathcal{R}_{v_1 \dots v_{(L-1)}}) + D(\mathcal{T}^{(e_{L-1} \dots M)}, \mathcal{R}_{v_L}) \quad (15)$$

Verdeutlicht wird dies durch Abbildung 6.

Unter Verwendung von Gleichung (11) folgt nun

$$\begin{aligned} D(\mathcal{T}, \mathcal{R}^s) &= \min_{u_1, \dots, u_L} \{D(\mathcal{T}^{(e_{L-1})}, \mathcal{R}_{u_1 \dots u_{L-1}}) + D(\mathcal{T}^{(e_{L-1} \dots M)}, \mathcal{R}_{u_L})\} \\ &= \min_{u_1, \dots, u_{L-1}} D(\mathcal{T}^{(e_{L-1})}, \mathcal{R}_{u_1 \dots u_{L-1}}) + \min_{u_L} D(\mathcal{T}^{(e_{L-1} \dots M)}, \mathcal{R}_{u_L}) \end{aligned} \quad (16)$$

Angenommen, die e_l seien für alle Stufen $l = 1, \dots, L$ bekannt, so kann die Bestimmung der Gesamtdistanz auf Stufe L nach Gleichung (16) rekursiv berechnet werden, wobei mit den Notationen aus Gleichung (12) für die erste Stufe gilt

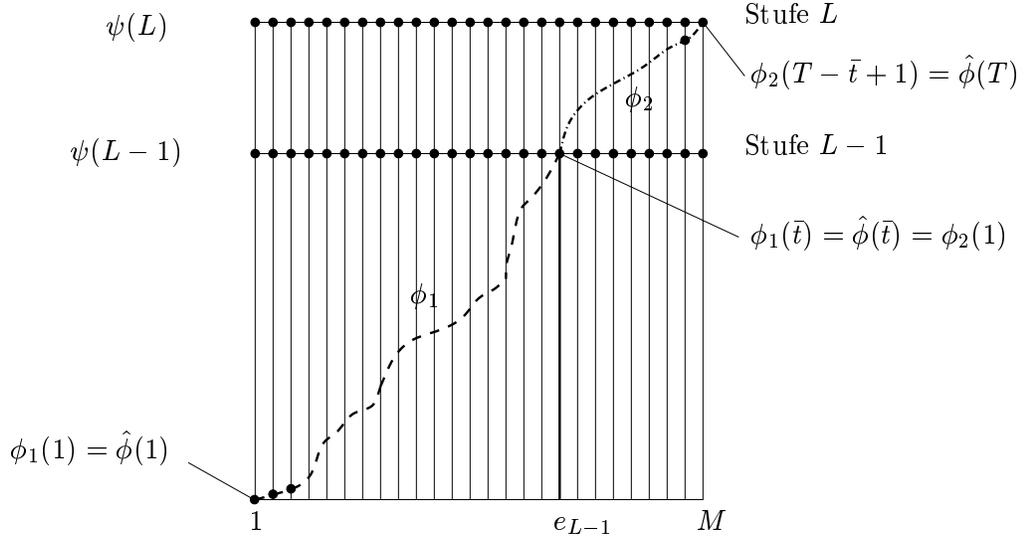


Abbildung 6: Veranschaulichung des LB Verfahrens auf der L -ten Stufe. Die Gesamtdistanz entlang des optimalen Warpingpfades $\hat{\phi}$ entspricht der Summe aus der Gesamtdistanz bis zur Stufe $L - 1$ an der Stelle e und der akkumulierten Distanz entlang $\hat{\phi}$ von e bis M .

$$\tilde{D}_1^*(e_1) = \min_{u_1} D(\mathcal{T}^{(e_1)}, \mathcal{R}_{u_1}) = \min_{u_1} \tilde{D}_1^{u_1}(e_1) \quad (17)$$

und für alle weiteren Stufen

$$\begin{aligned} \tilde{D}_l^*(e_l) &:= \min_{u_1 \dots u_l} D(\mathcal{T}^{(e_l)}, \mathcal{R}_{u_1 \dots u_l}) \\ &= \tilde{D}_{l-1}^*(e_{l-1}) + \min_{u_l} D(\mathcal{T}^{(e_{l-1} \dots e_l)}, \mathcal{R}_{u_l}) \end{aligned} \quad (18)$$

Dies bedeutet, daß die Minimierung über alle möglichen Kombinationen von Referenzmustern zurückgeführt werden kann auf eine Minimierung über alle einzelnen Referenzmuster auf jeder Stufe. Das Problem hierbei ist jedoch, daß die Endpunkte e_l erst ermittelt werden können, wenn der gesamte optimale Warpingpfad $\hat{\phi}$ bekannt ist, also erst auf Stufe L . Es muß also jedes Spaltenelement am Ende einer Stufe als potentielles Element von $\hat{\phi}$ betrachtet werden, sodaß die Minimierung für alle Spalten erfolgen muß. Diese Minima werden dann an die nächste Stufe weitergegeben, bis die Stufe L erreicht ist. Wegen $e_L = M$ folgt dann

$$D(\mathcal{T}, \mathcal{R}^s) = \tilde{D}_L^*(M) \quad (19)$$

Zwar ist die Gesamtdistanz damit berechnet, jedoch sind die Indizes des Superreferenzmusters mit minimaler Distanz noch nicht ermittelt. Dies geschieht dadurch, daß auf

jeder Stufe l außer der minimalen Distanz aller Referenzmuster auch der Index dieses Musters für jede Spalte gespeichert wird, also ein Zeilenvektor $I_l(m)$ mit

$$I_l(m) = \arg \min_v \tilde{D}_l^v(m) \quad \forall m = 1, \dots, M \quad (20)$$

Für die Indizes v_1, \dots, v_L mit $\mathcal{R}^s = \mathcal{R}_{v_1, \dots, v_L}$ gilt dann:

$$v_l = I_l(e_l) \quad \forall l = 1, \dots, L \quad (21)$$

Damit besteht wieder das Problem der expliziten Berechnung der e_l . Hierfür wird folgendermaßen vorgegangen:

Beim Durchlaufen jeder Stufe wird am Ende für den in Spalte m mit $m = 1, \dots, M$ endenden Pfad der Spaltenindex seines Anfangspunktes gespeichert. Dies geschieht dadurch, daß in einer Matrix F_l in jedem Punkt (m, n) der Spaltenindex des Anfangspunktes desjenigen Pfades gespeichert wird, welcher in (m, n) endet. Die nullte Zeile von F wird dabei mit den Spaltenindizes vorinitialisiert, also

$$\begin{aligned} F_l(m, 0) &= m \quad \forall m = 0, \dots, M \\ F_l(m, n) &= F_l(\hat{m}, \hat{n}) \quad \forall m = 1, \dots, M \quad n = 1, \dots, N_v \end{aligned} \quad (22)$$

wobei \hat{m} und \hat{n} die Indizes des Vorgängers von (m, n) bezeichnen.

Auf diese Weise entsteht für jedes Referenzmuster v eine derartige Matrix F_l , und die letzte Zeile $\tilde{F}_l^v(m)$ dieser Matrizen enthält dann jeweils die Spaltenindizes der Pfadstartpunkte beim Vergleich mit Referenzmuster v auf Stufe l , also

$$\tilde{F}_l^v(m) = F_l(m, N_v) \quad \forall m = 1, \dots, M \quad (23)$$

Schließlich wird für eine feste Spalte m der Eintrag desjenigen Vektors $\tilde{F}_l^v(m)$ kopiert, dessen zugehöriges Referenzmuster \mathcal{R}_v die Optimalität des in $(m, \psi(l))$ endenden Pfades herbeigeführt hat:

$$\tilde{F}_l^*(m) = \tilde{F}_l^{I_l(m)}(m) \quad \forall m = 1, \dots, M \quad (24)$$

Zur Bestimmung der gesuchten Endpunkte e_l genügt es nun, diese Zeilenvektoren von Stufe L bis zur Stufe 1 zu Durchlaufen gemäß der Vorschrift:

$$\begin{aligned} e_L &= M \\ e_l &= \tilde{F}_{l+1}^*(e_{l+1}) \quad \forall l = L-1, \dots, 1 \end{aligned} \quad (25)$$

Abbildung 7 veranschaulicht dieses Verfahren.

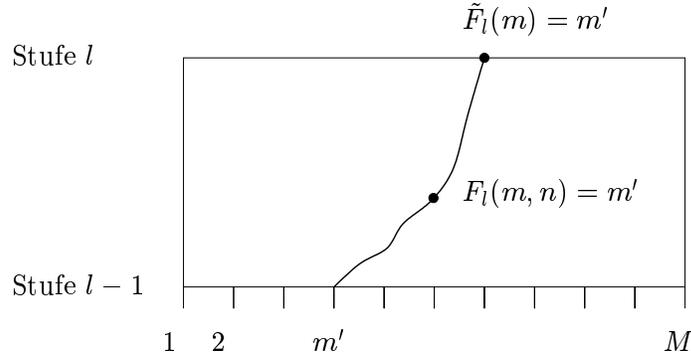


Abbildung 7: Propagierung des Spaltenindex m' des Startpunktes eines jeden Warpingpfades beim Durchlaufen der Stufe l

Bestimmung der Länge L des optimalen Superreferenzmusters Die einzige Einschränkung für alle bisherigen Überlegungen ist, daß die Anzahl der zu durchlaufenden Stufen L bekannt sein muß. Dies ist im allgemeinen jedoch nicht so. Vielmehr wird eine obere Schranke L_{max} für L gegeben. Um nun die Berechnungen wie beschrieben durchführen zu können, wird die Warpingmatrix zunächst bis zur Stufe L_{max} berechnet und anschließend der Index der Stufe ermittelt, auf welcher die akkumulierte Distanz entlang des optimalen Warpingfadens minimal ist, also gemäß der Gleichung

$$L = \arg \min_l \tilde{D}_l^*(M). \quad (26)$$

Die Zusammenfassung der beschriebenen Einzelschritte des LB-Verfahrens ist in Algorithmus 3 zu sehen.

2.2.3. Komplexität

Die für die Berechnung der Einträge in der Warpingmatrix benötigte Zeit hat eine Komplexität von $O(L_{max} \cdot V \cdot M \cdot \bar{N})$, wobei \bar{N} die mittlere Länge der Referenzmuster angibt. Kritisch hierbei ist hauptsächlich der Faktor V für die Anzahl der Referenzmuster, da die Qualität der Klassifikation stark von V abhängt. Um jedoch die Berechnungszeit auch im Falle von vielen Referenzmustern möglichst klein zu halten, ist es nötig, die Effizienz des Verfahrens zu steigern.

Dies kann zum Beispiel dadurch erreicht werden, daß die Berechnung der lokalen Distanzen $d(m, n)$ bezüglich eines Referenzmusters \mathcal{R}_v nicht für jede Stufe erneut erfolgt, sondern nur auf der ersten Stufe, da diese lokalen Distanzen auf allen Stufen für die jeweiligen Referenzmuster gleich bleiben. Die Berechnung der lokalen Distanzen auf den Stufen 2 - L_{max} geht dann über in einen einfachen Speicherzugriff auf bereits berechnete Werte $d(m, n)$. Tests ergaben, daß sich dadurch die Berechnungszeit für $L_{max} = 6$

Algorithmus 3 Level-building Algorithmus

Eingabe: Testmuster \mathcal{T} , Menge $R = \{\mathcal{R}_1, \mathcal{R}_2, \dots, \mathcal{R}_V\}$ von Referenzmustern

Ausgabe: Länge L von \mathcal{R}^s , $D(\mathcal{T}, \mathcal{R}^s)$ und v_1, \dots, v_L , wobei $\mathcal{R}^s = \mathcal{R}_{v_1, \dots, v_L}$ minimalen Abstand zu \mathcal{T} hat

$$\tilde{D}_0^*(0) \leftarrow 0$$

$$\tilde{D}_0^*(m) \leftarrow \infty \quad \forall m = 1, \dots, M$$

$$\tilde{D}_l^*(0) \leftarrow \infty \quad \forall l = 1, \dots, L_{max}$$

for $l = 1, \dots, L_{max}$ **do** {für alle Stufen}

for $v = 1, \dots, V$ **do** {für alle Referenzmuster}

$$D_l(m, 0) \leftarrow \tilde{D}_{l-1}^*(m)$$

$$F_l(m, 0) \leftarrow m \quad \forall m = 0, \dots, M$$

for $m = 1, \dots, M, n = 1, \dots, N_v$ **do**

 Bestimme Indizes \hat{m} und \hat{n} sodaß gilt

$$D_l(\hat{m}, \hat{n}) = \min \left\{ \begin{array}{l} D_l(m-1, n), \\ D_l(m, n-1), \\ D_l(m-1, n-1) \end{array} \right\}$$

$$D_l(m, n) \leftarrow D_l(\hat{m}, \hat{n}) + d(\mathbf{t}_m, \mathbf{r}_{\psi(l-1)+n})$$

$$F_l(m, n) \leftarrow F_l(\hat{m}, \hat{n})$$

end for

$$\tilde{D}_l^v(m) \leftarrow D_l(m, N_v) \quad \forall m = 1, \dots, M$$

$$\tilde{F}_l^v(m) \leftarrow F_l(m, N_v) \quad \forall m = 1, \dots, M$$

end for

$$\tilde{D}_l^*(m) \leftarrow \min_{1 \leq v \leq V} \tilde{D}_l^v(m) \quad \forall m = 0, \dots, M$$

$$I_l(m) \leftarrow \arg \min_{1 \leq v \leq V} \tilde{D}_l^v(m) \quad \forall m = 0, \dots, M$$

$$\tilde{F}_l^*(m) \leftarrow \tilde{F}_l^{I_l(m)}(m)$$

end for

$$L \leftarrow \arg \min_{1 \leq l \leq L_{max}} \tilde{D}_l^*(M)$$

$$e_L \leftarrow M$$

$$e_l \leftarrow \tilde{F}_{l+1}^*(e_{l+1}) \quad \forall l = L-1, \dots, 1$$

$$v_l \leftarrow I_l(e_l) \quad \forall l = 1, \dots, L$$

return $L, \tilde{D}_L^*(M), (v_1, \dots, v_L)$

ungefähr halbiert.

3. Realisierung

In diesem Abschnitt soll auf die Einzelheiten eingegangen werden, die bei der Umsetzung des Verfahrens von besonderer Bedeutung waren. Insbesondere sind dies, neben einer Beschreibung der konkreten Implementation, einige Erläuterungen zum gewählten Referenzdatensatz und zur Extraktion der am geeignetsten erscheinenden Merkmale. Schließlich wird auf eine Möglichkeit eingegangen, weitere Superreferenzmuster in einer nahen Umgebung von \mathcal{T} zu finden.

3.1. Implementation

Im Wesentlichen besteht das Verfahren aus einer Teilfunktion `mdist` zur Berechnung der Warpingdistanzen auf jeder Stufe und einer weiteren Funktion `dtw_levelbuilding`, welche den stufenweisen Aufbau der Warpingmatrix durchführt. Diese beiden Funktionen sollen im folgenden kurz beschrieben werden.

3.1.1. Berechnung der Distanzen

Die Funktion `mdist` berechnet zu einem gegebenen Testmuster \mathcal{T} und einem Referenzmuster \mathcal{R} den Abstand $D(\mathcal{T}, \mathcal{R})$, die Warpingmatrix D und eine *Pfadmatrix* P nach dem in Abschnitt 2.1 beschriebenen DTW-Algorithmus. Dabei stehen in $P_{m,n}$ die Zeilen- und Spaltenindizes des vorletzten Punktes im optimalen Warpingpfad, welcher in (m, n) endet. Mithilfe von P kann dann der optimale Warpingpfad $\hat{\phi}$ durch Rückverfolgen der Einträge in P ermittelt werden. Dies geschieht, indem, beginnend beim Punkt (M, N) , in einer Schleife die jeweiligen Vorgänger von $\hat{\phi}$ ausgelesen und in einer Indexsequenz $\bar{\phi}$ gespeichert werden, also wie folgt:

$$\begin{aligned}\bar{\phi}(1) &= (M, N) \\ \bar{\phi}(k) &= P_{\bar{\phi}(k-1)}\end{aligned}$$

Die Abbruchbedingung für die Schleife ist $\bar{\phi}(k) = (1, 1)$. Der optimale Warpingpfad $\hat{\phi}$ ergibt sich dann als die Spiegelung von $\bar{\phi}$.

Es kann an die Funktion ein weiteres, optionales Argument `bot` übergeben werden, welches die nullte Zeile angibt, mit der die Warpingmatrix initialisiert werden soll (*bottom line*). Voreinstellung ist hierbei der Vektor $(0, \infty, \dots, \infty)$. In Algorithmus 3 entspricht `bot` dem Zeilenvektor \tilde{D}_0^* .

3.1.2. Aufbau der Warpingmatrix

Bei einer gegebenen Menge $R = \{\mathcal{R}_1, \dots, \mathcal{R}_V\}$ von Referenzbuchstaben, einer Maximalzahl L_{max} von Stufen und einem Wort \mathcal{T} als Testmuster berechnet die Funktion `dtw_levelbuilding` dasjenige Superreferenzmuster \mathcal{R}^s , welches minimalen Abstand zu \mathcal{T} hat. Zu diesem Zweck wird auf jeder Stufe für alle Referenzen die Funktion `mdist` aufgerufen. Dabei tritt in der ersten Stufe der Vektor `bot` nicht in der Argumentliste von `mdist` auf, in jeder weiteren Stufe l wird gemäß Algorithmus 3 der Vektor \tilde{D}_{l-1}^* an `mdist` übergeben.

3.1.3. Auslagerung der Distanzberechnung

Bei genauerer Betrachtung von Algorithmus 3 fällt auf, daß in den beiden inneren Schleifen mit den Laufvariablen m bzw n die Berechnung eines einzelnen Matrixelementes von den zuvor berechneten abhängt. Das bedeutet, daß diese Berechnungen nicht als Matrixoperationen formuliert werden können, sondern für jedes Element einzeln erfolgen müssen. Bekanntlich sind jedoch Schleifenoperationen sehr zeitintensiv, wenn sie in einer interpretativen Sprache implementiert werden. Aus diesem Grund wurde die Funktion `mdist`, welche im wesentlichen diese Schleifenoperationen durchführt, in einer maschinen-näheren Compilersprache verfaßt und in die Funktion `dtw_levelbuilding` eingebunden. Die Berechnungszeit konnte dadurch auf bis zu einem Sechstel verringert werden.

3.2. Der geeignete Referenzdatensatz

Die Wahl der geeigneten Referenzdaten ist für die Güte des Klassifikationsergebnisses von entscheidender Bedeutung. Zwar ist es für die zeitliche Optimierung des Verfahrens von Interesse, die Anzahl der verwendeten Referenzmuster möglichst klein zu halten. Es hat sich jedoch gezeigt, daß die verschiedenen Ausprägungen eines einzelnen Buchstabens sowohl von Schreiber zu Schreiber als auch für einen einzelnen Schreiber derart variieren, daß in der Regel mehrere Referenzmuster pro Buchstabe zur Klassifikation zugrundegelegt werden müssen. Ein Hauptziel bei der Erzeugung derartiger Referenzmuster ist ein gute Generalisierungseigenschaft der Referenzen, das heißt, es sollen möglichst viele verschiedene Erscheinungsformen eines jeden Buchstaben in der Referenzmenge repräsentiert werden.

Hierzu sind unter anderem die folgenden Vorgehensweisen denkbar:

Manuelle Methode: Es wird von Hand ein Referenzdatensatz erzeugt, der möglichst jede denkbare Variation eines Buchstaben beinhaltet. Diese Methode ist jedoch, wie sich gezeigt hat, aufgrund ihrer geringen Generalisierung nicht anwendbar.

Erzeugen von Buchstabenstichproben: Es werden von verschiedenen Schreibern einzelne Buchstaben aufgenommen und deren genaue Lage im Merkmalraum untersucht.

Diejenigen Stichproben, die bezüglich eines Distanzmaßes D einen Abstand voneinander haben, der unter einem gewissen Schwellwert liegt, werden einer Klasse zugeteilt (*Clusterbildung*). Anschließend wird ein geeigneter Repräsentant der Klasse als Referenzmuster ausgewählt. Diese Methode generalisiert besser, da die Informationen über alle aufgenommenen Stichproben direkt in die Berechnung eingehen.

Segmentierung von Wortstichproben: Hierbei werden die einzelnen Stichproben nicht gesondert erzeugt, sondern durch Segmentierung von handgeschriebenen Wörtern extrahiert. Die Generierung der Referenzklassen erfolgt dann ebenso durch Clusterbildung. Der Vorteil dieser Methode ist, daß die einzelnen Referenzmuster für die Worterkennung geeigneter sind als bei der Erzeugung von Buchstabenstichproben, da die Gestalt eines Buchstaben, der im Kontext eines Wortes geschrieben wurde, stark von der eines einzeln geschriebenen Buchstaben abweichen kann. Ein Nachteil des Verfahrens besteht jedoch darin, daß der geschriebene Text so lang sein muß, daß von allen Buchstaben eine ausreichende Anzahl von Stichproben vorliegt, um die einzelnen Klassen definieren zu können.

Bei der Erzeugung der Referenzmuster durch Clusterbildung ist es von besonderer Bedeutung, auf welche Art der Repräsentant einer Klasse bestimmt wird. Beispielsweise kann dies erfolgen durch Ermittlung des Musters mit minimalem Gesamtabstand zu allen anderen Mustern. Liegt jedoch, wie hier, keine Metrik im mathematisch strengen Sinne zur Abstandsbestimmung vor, so wird üblicherweise der *Median* der Klasse bestimmt. Dieser ist definiert als das Muster mit minimalem mittleren Abstand zu allen anderen Mustern. Dabei wird der mittlere Abstand eines Musters \mathcal{T} berechnet durch aufsteigende Sortierung der Abstände aller Muster zu \mathcal{T} und anschließender Bestimmung des mittleren dieser Werte (siehe hierzu [6]).

3.3. Extraktion der Merkmale

Wie bereits in Abschnitt 2.1.1 erwähnt, sind viele verschiedene Merkmale für die Klassifikation denkbar, jedoch stellt sich heraus, daß nur wenige im Fall der Worterkennung anwendbar sind. Um zum Beispiel die Ortskoordinaten eines geschriebenen Buchstaben als Merkmale verwenden zu können, müssen diese Koordinaten zunächst geeignet normiert werden, um dadurch die Lageinvarianz der Merkmale zu gewährleisten.

Da jedoch als Testmuster ganze Worte erkannt werden sollen, ist eine Normierung nur für y -Koordinaten möglich, denn nur für diese können gemeinsame *Referenzlinien* in Test- und Referenzmuster gefunden werden. Eine Normierung der x -Koordinaten müßte auch für die Testmuster auf Buchstabenebene erfolgen, jedoch sind die Übergangsstellen von einem Buchstaben zum nächsten innerhalb eines Wortes in der Regel nicht bekannt.

Die Normierung der y -Koordinaten erfolgt bezüglich der vier Referenzlinien eines Wortes. Diese heißen *Untertlängenlinie*, *Grundlinie*, *Kernlinie* und *Obertlängenlinie* und

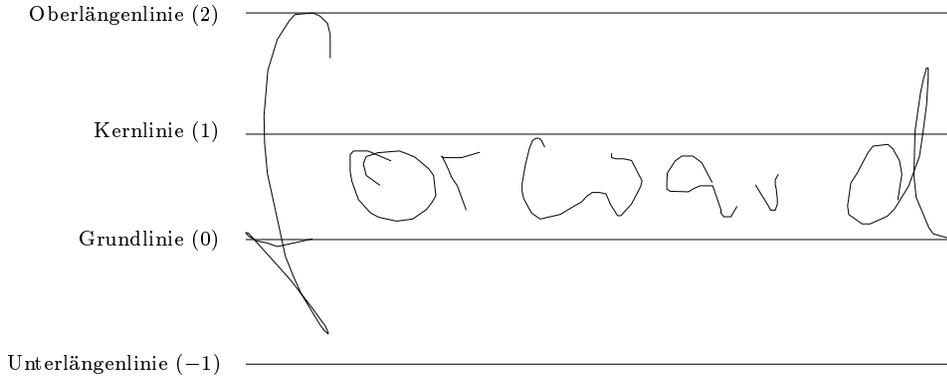


Abbildung 8: Normierung der vertikalen Koordinaten mithilfe der vier Referenzlinien eines Buchstabens

erhalten die Werte $-1, 0, 1$ und 2 (siehe Abbildung 8). Der Wertebereich der normierten y -Koordinaten ist also auf das Intervall $[-1; 2]$ beschränkt.

Einen lage- und skalierungsinvarianten Satz von Merkmalen stellen die Winkel α_r für $r = 2, \dots, N_v - 1$ zwischen der Tangenten im Punkt (x_r, y_r) und der x -Achse dar. Die Richtung der Tangenten wird dabei durch den Vektor $\begin{pmatrix} x_{r+1} - x_{r-1} \\ y_{r+1} - y_{r-1} \end{pmatrix}$ angenähert. Für den Anfangs- und den Endpunkt werden ebenfalls Merkmale berechnet. Da jedoch an diesen Stellen keine Tangentenbestimmung möglich ist, werden hierfür die Vektoren $\begin{pmatrix} x_2 - x_1 \\ y_2 - y_1 \end{pmatrix}$ bzw. $\begin{pmatrix} x_{N_v} - x_{N_v-1} \\ y_{N_v} - y_{N_v-1} \end{pmatrix}$ eingesetzt. Der Grund für dieses Vorgehen ist, daß von jedem Merkmal eine gleiche Anzahl von Meßdaten zur Verfügung stehen muß, um jeweils alle Merkmalsvektoren zweier Muster miteinander vergleichen zu können.

Ein graphische Darstellung der α_r ist in Abbildung 9 zu sehen. Für die Repräsentation dieser Winkel wurden aus implementationstechnischen Gründen die Sinus- bzw. Cosinuswerte der α_r gewählt.

Ein Nachteil dieser Merkmale ist jedoch, daß sie keine Information über die relative Größe eines Buchstaben innerhalb eines Wortes beinhalten (zum Beispiel können die α_r für ein 'e' und ein 'l' nahezu gleich sein). Aus diesem Grund wurden für die Implementation außer den Winkeln α_r auch die normierten y -Koordinaten \bar{y}_r als Merkmale extrahiert.

Die für die Implementation verwendeten Merkmalsvektoren für ein Referenzmuster $\mathcal{R}_v = (\mathbf{r}_1, \dots, \mathbf{r}_{N_v})$ und ein Testmuster $\mathcal{T} = (\mathbf{t}_1, \dots, \mathbf{t}_M)$ ergeben sich somit zu:

$$\mathbf{r}_n = \begin{pmatrix} \bar{y}_n \\ \sin(\alpha_n) \\ \cos(\alpha_n) \end{pmatrix} \quad \mathbf{t}_m = \begin{pmatrix} \bar{y}_m \\ \sin(\alpha_m) \\ \cos(\alpha_m) \end{pmatrix} \quad (27)$$

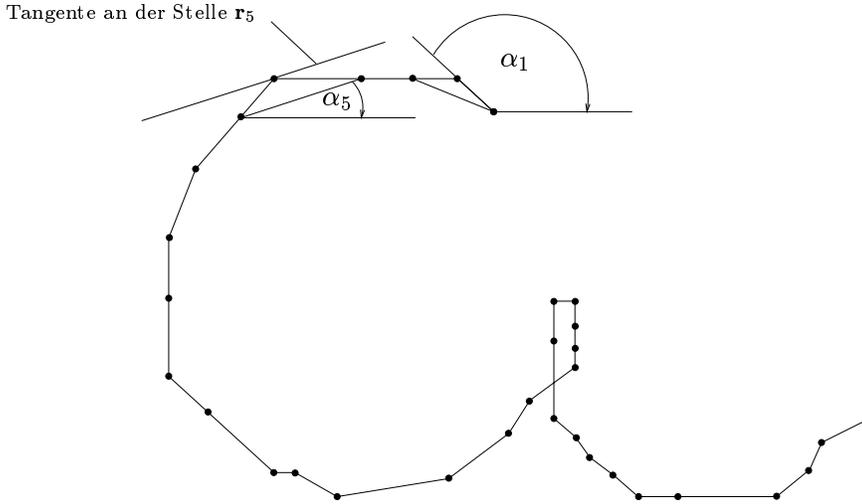


Abbildung 9: Extraktion der Winkel zwischen den Tangenten und der x -Achse als lage- und skalierungs-invarianter Satz von Merkmalen am Beispiel des Buchstaben 'a'. Für den ersten und letzten Abtastwert werden anstelle dieser Winkel die Neigungswinkel des entsprechenden Streckenabschnitts verwendet.

3.4. Untersuchung einer Umgebung von \mathcal{T}

Um eine Möglichkeit zu schaffen, ein passendes Muster aus einer Menge von möglichen Kandidaten für die Klassifikation auszuwählen, ist es sinnvoll, statt eines einzigen Superreferenzmusters \mathcal{R}^s mit minimalem Abstand zu \mathcal{T} eine Menge von Superreferenzmustern $\mathcal{R}_1^s, \dots, \mathcal{R}_p^s$ in einer bestimmten Umgebung $U(\mathcal{T})$ bezüglich der Metrik D zu betrachten. Innerhalb dieser Menge können dann diejenigen Muster ausgeschlossen werden, die aufgrund von Vorwissen über die Beschaffenheit der zu erkennenden Testmuster nicht in Frage kommen. Um eine derartige Menge zu erhalten, kann man innerhalb des LB-Algorithmus folgendermaßen vorgehen (siehe [3] für weitere Details hierzu):

Während des Aufbaus der Warpingmatrix werden am Ende jeder Stufe l außer den akkumulierten Teildistanzen des besten Warpingpfades zu einem Punkt $(m, \psi(l))$ der Matrix auch diejenigen entlang des zweitbesten Pfades zu $(m, \psi(l))$ gespeichert. Für jede Stufe wird also ein weiterer Zeilenvektor $\tilde{D}_l^{**}(m)$ für $m = 1, \dots, M$ berechnet, der jeweils die zweitbeste Distanz zu $(m, \psi(l))$ angibt. Ebenso werden Zeilenvektoren $\tilde{F}_l^{**}(m)$ und $I_l^{(2)}(m)$ berechnet, welche, analog zu $\tilde{D}_l^{**}(m)$ die Endpunkte bzw. Referenzmusterindizes entlang des zweitbesten Pfades durch Stufe l speichern. Veranschaulicht wird diese Vorgehensweise durch Abb. 10. Nachdem der Stufenindex L des besten Warpingpfades wie in Algorithmus 3 bestimmt wurde, gibt es nun insgesamt 2^L mögliche Warpingpfade. Aus dieser Menge von Pfaden werden nun diejenigen ausgewählt, welche beim Durchlaufen von Stufe l mit $l = 1, \dots, L$ den zweitbesten Teilpfad enthalten (im Beispiel von Abb.

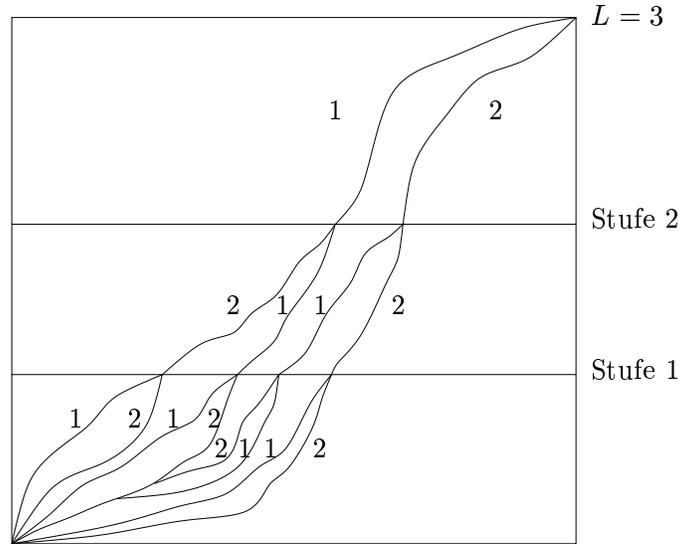


Abbildung 10: Bestimmung des zweitbesten aller optimalen Warpingpfade auf jeder Stufe (hier mit 2 markiert, 1 bezeichnet den jeweils besten Pfad). Es entsteht ein binärer Baum von möglichen Warpingpfaden, entlang derer die akkumulierten Distanzen berechnet werden. Anschließend werden diese Warpingpfade bezüglich ihrer Distanzen aufsteigend sortiert und die zugehörigen Superreferenzmuster ausgegeben.

10 sind dies die Pfade $(2, 1, 1)$, $(1, 2, 1)$ und $(1, 1, 2)$). Anschließend werden die Distanzen entlang dieser Pfade berechnet und aufsteigend sortiert. Die zugehörigen Superreferenzmuster können dann ebenfalls als mögliche Kandidaten für die Klassifikation in Betracht gezogen werden.

Bemerkung: Die Menge der so erhaltenen Superreferenzmuster entspricht nicht der Menge der bezüglich D nächstliegenden Muster, da nur auf Stufe L nach weiteren Kandidaten gesucht wird. Es können jedoch durchaus Superreferenzmuster kleinerer oder größerer Länge existieren, deren Warpingdistanz noch kleiner ist als die der $\mathcal{R}_1^s, \dots, \mathcal{R}_\nu^s$.

4. Ergebnisse

Im folgenden Abschnitt sollen die Klassifikationsergebnisse des implementierten Verfahrens dokumentiert werden. Zunächst wird eine Evaluationsfunktion definiert, mithilfe derer die verschiedenen Ergebnisse verglichen werden sollen. Anschließend folgen eine kurze Beschreibung der durchgeführten Versuchsreihen und Erläuterungen zur Visualisierung. Schließlich wird die Auswertung der Ergebnisse vorgenommen und interpretiert.

4.1. Wahl einer geeigneten Evaluationsfunktion

Um ein Klassifikationsergebnis bewerten zu können, muß eine Funktion f gefunden werden, welche jedem Paar $(\mathcal{T}, \mathcal{R}^s)$ von Testmuster und erkanntem Superreferenzmuster eine reelle Zahl als Bewertung der Klassifikation zuordnet. Dabei soll die intuitive Beurteilung des Klassifikationsergebnisses möglichst gut durch eine mathematische Anordnung modelliert werden. Zu diesem Zweck sollten alle Parameter der Klassifikation in die Modellierung mit eingehen. Im Fall der Worterkennung wäre beispielsweise die Anzahl der richtig erkannten Buchstaben im Wort für eine Bewertung ungeeignet, da die Länge des erkannten Wortes nicht berücksichtigt wird.

Die einfachste denkbare Funktion wäre die, die genau dann 1 liefert, wenn das Wort richtig erkannt wurde, und ansonsten 0, jedoch könnten anhand dieser Funktion zu wenige Aussagen über das Ergebnis getroffen werden. Beispielsweise würde ein Ergebnis, bei dem das erkannte Wort nur einen Buchstaben vom Testmuster abweicht, mit 0 bewertet, obwohl das Ergebnis intuitiv sehr gut ist.

Um die Ähnlichkeit der beiden zu vergleichenden Zeichenketten mit in die Modellierung aufzunehmen, kann die Berechnung der *Editierdistanz* zwischen \mathcal{T} und \mathcal{R}^s benutzt werden. Hierbei ist die Editierdistanz $\text{ed}(\mathbf{a}, \mathbf{b})$ zwischen zwei Zeichenketten $\mathbf{a} = (a_1, a_2, \dots, a_\mu)$ und $\mathbf{b} = (b_1, b_2, \dots, b_\nu)$ definiert als die minimalen Kosten, um \mathbf{a} in \mathbf{b} mithilfe der Editieroperationen *löschen*, *einfügen* und *ersetzen* zu überführen. Die Kosten, um ein einzelnes Zeichen zu löschen oder einzufügen, werden mit 1 bewertet, die Kosten $c(a, b)$ für das Ersetzen eines Zeichens a durch ein Zeichen b werden definiert als

$$c(a, b) := \begin{cases} 1 & \text{falls } a \neq b \\ 0 & \text{sonst} \end{cases} \quad (28)$$

Die Editierdistanz zwischen \mathbf{a} und \mathbf{b} kann wie folgt rekursiv berechnet werden (dabei wird, wie beim DTW-Algorithmus, das Prinzip der Dynamischen Programmierung zugrundegelegt; näheres hierzu ist u.a. in [5] zu finden):

$$\text{ed}(\mathbf{a}, \mathbf{b}) = \min \left\{ \begin{array}{l} \text{ed}(\mathbf{a}', \mathbf{b}') + c(a_\mu, b_\nu), \\ \text{ed}(\mathbf{a}', \mathbf{b}) + 1, \\ \text{ed}(\mathbf{a}, \mathbf{b}') + 1 \end{array} \right\} \quad (29)$$

mit $\mathbf{a}' = (a_1, a_2, \dots, a_{\mu-1})$ und $\mathbf{b}' = (b_1, b_2, \dots, b_{\nu-1})$. Das Abbruchkriterium für die Rekursion lautet dann

$$\begin{aligned} \text{ed}(\mathbf{a}, \epsilon) &= \text{len}(\mathbf{a}) \\ \text{ed}(\epsilon, \mathbf{a}) &= \text{len}(\mathbf{a}) \end{aligned} \quad (30)$$

wobei ϵ die leere Zeichenkette bezeichnet und $\text{len}(\mathbf{a})$ die Länge einer Zeichenkette \mathbf{a} angibt.

	einzelne Buchstaben	segmentierte Wörter
alle Schreiber	R_1	R_2
ein Schreiber	R_3	R_4

Tabelle 1: Erzeugung von vier verschiedenen Referenzdatensätzen $R_1 - R_4$ zur Auswertung des implementierten Verfahrens

Mithilfe der Editierdistanz kann zwar die Ähnlichkeit zwischen zwei Zeichenketten gemessen werden, jedoch geht die Gesamtlänge der Zeichenketten nicht in die Bewertung ein. Darum wurde als Evaluationsfunktion f der Quotient aus Editierdistanz zwischen \mathcal{T} und \mathcal{R}^s und der Anzahl der Buchstaben $\text{len}(\mathcal{T})$ in \mathcal{T} gewählt, also

$$f(\mathcal{T}, \mathcal{R}^s) := \frac{\text{ed}(\mathcal{T}, \mathcal{R}^s)}{\text{len}(\mathcal{T})} \quad (31)$$

Die Evaluationsfunktion kann nun so interpretiert werden, daß der Wert $f(\mathcal{T}, \mathcal{R}^s) = 0$ einer fehlerfreien Erkennung des Testmusters entspricht und für $f(\mathcal{T}, \mathcal{R}^s) \neq 0$ die Güte des Klassifikationsergebnisses mit steigenden Werten von f fällt ².

4.2. Aufbau der Versuchsreihen

Zur Beurteilung des Verfahrens waren zwei Fragestellungen maßgebend. Einerseits sollte untersucht werden, welchen Einfluß die Generierung der Referenzmuster auf die Erkennungsrate der Testmuster hat, also ob die Erzeugung von Buchstabenstichproben der Segmentierung von Wortstichproben vorzuziehen ist oder nicht (siehe hierzu Abschnitt 3.2). Andererseits war die Generalisierungseigenschaft des Verfahrens von Interesse, also inwieweit Vorwissen über den Schreiber nötig ist, um eine ausreichende Anzahl von Worten zu erkennen.

Zur Untersuchung dieser Fragen wurden zunächst von mehreren Schreibern sowohl einzeln geschriebene Buchstaben, als auch fortlaufende Texte aufgenommen. Aus diesen Daten wurden nach den in Abschnitt 3.2 genannten Methoden jeweils zwei verschiedene Referenzdatensätze R_1 und R_2 bzw. R_3 und R_4 erzeugt. Tabelle 1 veranschaulicht die Benennung der einzelnen Datensätze.

Für alle Versuchsreihen bestand die Menge der Testmuster aus einem handgeschriebenen Text in englischer Sprache, welcher 198 Worte umfaßte. Dabei wurde diese Testmustermenge disjunkt zu den Referenzdatensätzen gewählt. Für jedes Testmuster wurden,

²Strenggenommen müßten die Argumente der Funktion f in Gleichung (31) die Worte sein, die die semantische Bedeutung der Muster \mathcal{T} und \mathcal{R}^s wiedergeben und nicht die Muster selbst, da die Editierdistanz ed und die Länge len nur für Zeichenketten definiert sind. \mathcal{T} und \mathcal{R}^s sind jedoch als Sequenzen von Merkmalvektoren definiert und demnach keine Zeichenketten.

neben dem Superreferenzmuster $\mathcal{R}^s = \mathcal{R}_{v_1 \dots v_L}$ mit minimaler Waringdistanz, auch L weitere Muster $\mathcal{R}^{s^1} \dots \mathcal{R}^{s^L}$ berechnet. Für diese weiteren Muster wurde nach dem in Abschnitt 3.4 beschriebenen Verfahren auf jeder Stufe das Referenzmuster mit zweitbestem optimalen Waringpfad ermittelt und die Muster anschließend bezüglich der Waringdistanz sortiert.

Für die spätere Auswertung wurde eine Tabelle erstellt, in die die erkannten Wörter und ihre Editierdistanz zum Testwort spaltenweise eingetragen wurden. Tabelle 2 zeigt beispielhaft einen Auszug aus einem der Klassifikationsergebnisse. Am Ende jeder Tabelle wurde eine Gesamtbewertung \bar{f} angefügt, welche die über alle Testworte gemittelte Evaluationsfunktion f wiedergibt. Für einen Text $\Theta = (\mathcal{T}_1, \dots, \mathcal{T}_W)$, welcher W Worte umfaßt, wird also \bar{f} wie folgt berechnet:

$$\bar{f} = \frac{1}{W} \sum_{i=1}^W f(\mathcal{T}_i, \mathcal{R}_i^s) \quad (32)$$

wobei \mathcal{R}_i^s das Superreferenzmuster mit minimaler Waringdistanz zu \mathcal{T}_i angibt.

Testwort	\mathcal{R}^s	ed	\mathcal{R}^{s^1}	ed	\mathcal{R}^{s^2}	ed	\mathcal{R}^{s^3}	ed	\mathcal{R}^{s^4}	ed
⋮	⋮		⋮		⋮		⋮		⋮	
The	Tne	1	The	0	TwI	2	Yne	2		
first	IiyV	4	IiyZ	4	IigV	4	IoyV	5	IiyV	4
part	parb	1	parV	1	pavb	2	purb	2	barb	2
is	Is	1	xs	1	Ix	2				
easy	easy	0	eusy	1	easg	1	casy	1	eaxy	1
⋮	⋮		⋮		⋮		⋮		⋮	

Tabelle 2: Beispiel für eine Ausgabetable des Klassifikationsprogramms. Für jedes zu erkennende Wort werden mehrere mögliche Superreferenzmuster und ihre Editierdistanz ed zum Testwort ausgegeben.

4.3. Visualisierung

Um das Klassifikationsergebnis sichtbar zu machen, wurden für Test- und Superreferenzmuster jeweils die Ortskoordinaten gegeneinander aufgetragen und zusammen mit den zugehörigen Waringmatrizen angezeigt. Die Einträge der Waringmatrizen wurden dabei als Farbwerte kodiert und der optimale Waringpfad als schwarze Linie in die Waringmatrix eingetragen. Im Testmuster wurden zusätzlich senkrechte Trennlinien an den Stellen eingetragen, an denen der Algorithmus die Segmentierung des Wortes vorgenommen hat (diese entsprechen den Endpunkten e_l in Algorithmus 3). Außerdem wurden korrespondierende Teilstücke in Test- und Superreferenzmuster farblich hervorgehoben. Ein Beispiel für eine derartige Darstellung ist in Abbildung 11 zu sehen.

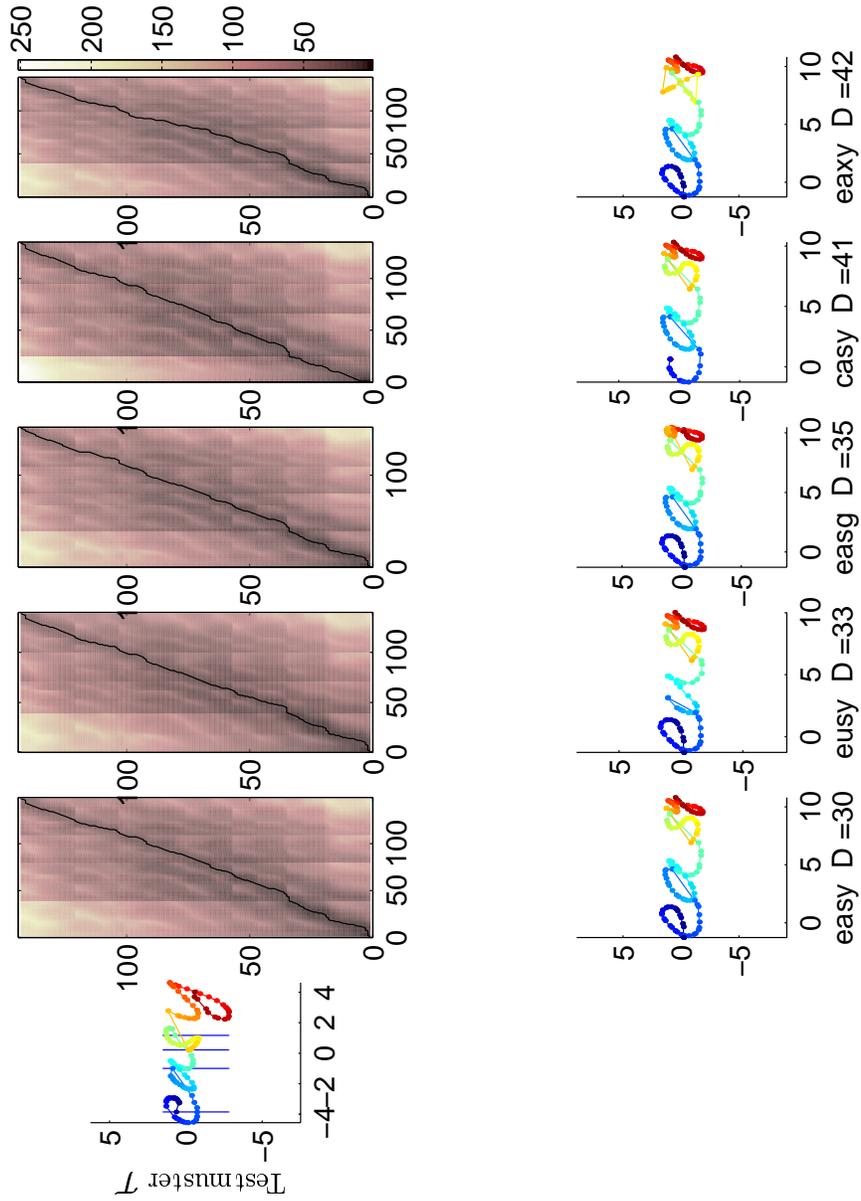


Abbildung 11: Beispiel zur Visualisierung des Klassifikationsergebnisses.

4.4. Auswertung

Auszüge der einzelnen Klassifikationsergebnisse sind in Anhang A zu sehen. Schon der Vergleich dieser vier Tabellen zeigt starke Unterschiede in den Ergebnissen. Vor allem ist die Erkennungsrate mit segmentierten Wortstichproben aller Schreiber als Referenzdatensatz (also R_2) als besonders schlecht zu beurteilen. Die Gesamtbewertungen \bar{f} für die einzelnen Versuchsreihen sind in Tabelle 3 aufgelistet. Aus diesen Bewertungen kann gefolgert werden, daß die segmentierten Wortstichproben (Datensätze R_2 und R_4) nicht zur Erzeugung eines Referenzdatensatzes geeignet sind. Die Anzahl der in einem durchschnittlichen Text vorkommenden Einzelbuchstaben variiert so stark, daß eine viel größere Datenmenge aufgenommen werden muß, um die einzelnen Buchstaben geeignet repräsentieren zu können.

Eine weitere Erkenntnis, die aus Tabelle 3 gezogen werden kann, ist, daß das Verfahren in der vorgestellten Version nicht schreiberunabhängig eingesetzt werden kann. Die Ursache dafür ist, daß zu wenig Information über die Verteilung der Stichproben in einer Klasse in die Berechnung eingeht.

Im Fall des Referenzdatensatzes R_3 kann die Erkennungsrate positiv beurteilt werden, wenn sie auch kein überzeugendes Ergebnis darstellt. Dennoch lag der Anteil der erkannten Worte, welche eine Editierdistanz von 0 oder 1 hatten, bei 44,4%. Die höchste Editierdistanz lag bei 6 und wurde nur in 1,1% der Fälle erreicht. Dies bedeutet, daß dieses Ergebnis als Vorverarbeitungsschritt eines schreiberabhängigen Verfahrens dienen könnte, wobei mithilfe geeigneter Methoden das naheliegendste der erkannten Worte ausgegeben würde.

Referenzdatensatz	Gesamtbewertung
R_1	$\bar{f} = 0,528$
R_2	$\bar{f} = 1,006$
R_3	$\bar{f} = 0,481$
R_4	$\bar{f} = 0,651$

Tabelle 3: Gesamtbewertung der einzelnen Versuchsreihen

5. Zusammenfassung und Ausblick

In der vorliegenden Arbeit wurde versucht, ein aus der Spracherkennung bekanntes Verfahren auf die Anwendung der Handschriftenerkennung zu übertragen. Die Idee dabei war, daß die Problemstellung von prinzipiell gleicher Natur ist, nämlich die Klassifikation von Mustern, die einen zeitlichen Verlauf darstellen. Zu diesem Zweck wurde eine Funktion zur zeitlichen Angleichung zweier Muster bereitgestellt. Auf der Basis dieser

Angleichung wurden dann weitere Überlegungen zur Worterkennung durch den Level-building Algorithmus angestellt.

Die Hauptproblematik im Fall der Erkennung von handgeschriebenen Wörtern lag jedoch in der Wahl der geeigneten Merkmale und der Art des Aufnahmeverfahrens für den Referenzdatensatz. Die vorgelegten Ergebnisse haben gezeigt, daß vor allem Änderungen am Referenzdatensatz einen entscheidenden Einfluß auf die Güte des Klassifikationsergebnisses haben. Hinsichtlich der Merkmalextraktion ist zu sagen, daß aufgrund der Verschiedenartigkeit von Test- und Referenzmuster und wegen weiterer Einschränkungen, wie zum Beispiel der Lageinvarianz, die Wahl der geeigneten Merkmale äußerst schwierig ist.

Um das Verfahren weiter zu verbessern, erscheint es unter anderem sinnvoll, weitere Informationen über die stochastischen Verteilungen der einzelnen Buchstabenstichproben mit in die Modellierung aufzunehmen. Durch die einfache Wahl des Medians als Repräsentanten einer Referenzklasse kann zum Beispiel keine Aussage über die Varianz der einzelnen Merkmale innerhalb der Klasse getroffen werden. Ein zusätzliches Wissen über stochastische Parameter der einzelnen Klassen ist aber unerlässlich, um eine möglichst schreiberunabhängige Klassifikation vornehmen zu können. Weitere Untersuchungen dahingehend werden bereits unternommen.

Abschließend kann bemerkt werden, daß ein schreiberunabhängiges Verfahren zur Worterkennung wohl nur dann stabil arbeiten kann, wenn Kontextwissen über die zu erkennenden Wörter mit in die Modellierung einfließt. Für das vorgestellte Verfahren würde dies zum Beispiel eine Suche der erkannten Wörter in einem geeigneten Wörterbuch bedeuten, wobei das in einem gewissen Sinne ähnlichste Wort des Wörterbuchs als Klassifikationsergebnis ausgegeben würde. Auch hierbei erscheinen weitere Forschungen sinnvoll.

A. Klassifikationsergebnisse

A.1. Referenzdatensatz R_1

Test wort	\mathcal{R}^s	ed	\mathcal{R}^{s^1}	ed	\mathcal{R}^{s^2}	ed	\mathcal{R}^{s^3}	ed	\mathcal{R}^{s^4}	ed
The	The	0	Tne	1	Thc	1	Zhe	1		
Hitch	VIIWh	4	VIIWl	5	IleWh	4	VINXh	4	IIIWh	4
Hiker's	IWOlzevs	7	IWOlzers	6	Vliilzevs	6	VWOlzevs	7	IWOlzevs	7
Guide	Gzwle	3	Guele	2	Gzwhl	4	tzwle	4	Gzwle	3
to	lv	2	Gv	2	lz	2				
the	lne	2	Vhe	1	Lne	2	Inc	3		
Galaxy	lelaey	3	Ielaey	3	leluey	4	lelaay	3	lelaeg	4
has	has	0	hus	1	nas	1	heq	2		
this	VhTs	2	ThTs	2	VkTs	3	VVIS	4	VhVs	2
to	IZ	2	lZ	2	It	2				
say	say	0	suy	1	sag	1	Say	1		
to	Iv	2	Lv	2	Iu	2				
the	Ihe	1	Iha	2	lhe	1	Ike	2		
subject	subpaV	4	subpaX	4	subyaV	4	sabpaV	5	subywV	4
of	of	0	xf	1	ol	1				
flying	flmg	3	fbymg	3	flgmy	4	llmg	4	flmg	4
There	Thqe	2	Tnqe	3	lhqe	3	Thae	2	Thql	3
is	Is	1	IS	2	ls	1				
an	an	0	om	2	un	1				
art	mN	3	mb	3	wN	3				
it	IV	2	IX	2	LV	2				
says	suys	1	sugs	2	says	0	suyx	2	Suys	2
or	ov	1	or	0	xv	2				
rather	IaVhor	3	IaVhlv	4	IaVhlr	3	IaThor	3	IuVhor	4
a	a	0	u	1						
knack	Iznwh	5	Izmah	5	Iznwk	4	Izhwh	5	lznwh	5
to	lTl	3	lTI	3	lXl	3	OTl	3		
flying	flymg	2	flymy	3	flmg	3	fbymg	3	llymg	3
The	TIhe	1	TIhl	2	TIke	2	TLhe	1	IIhe	2
knack	IznwIz	5	IzhwIz	6	Iznwlz	5	hnaXIz	4	lznwIz	5
lies	les	1	Ies	2	Ias	3	leS	2		
in	m	2	n	1						
learning	Iammg	6	lemmg	5	lammg	5	lanmg	4	lamwy	6
how	hou	1	how	0	hxu	2	kou	2		
to	Vo	1	To	1	It	2				
throw	Ihmv	4	Ihhw	3	Lhmv	4	Ikmv	5	Ibnv	5
yourself	yonfaLf	5	yonfalf	5	yomfaLf	5	yOnfaLf	6	yovvfaf	5
at	aV	1	aT	1	uV	2				
the	Imo	3	Lmo	3	Imf	3	Wlo	3		
ground	gmeml	5	geuml	4	gvamd	4	gmml	5	ymeml	6
and	and	0	und	1	aml	2	aud	1		
miss	mss	1	mys	2	msx	2	nss	2		
⋮	⋮		⋮		⋮		⋮		⋮	

A.2. Referenzdatensatz R_2

Testwort	\mathcal{R}^s	ed	\mathcal{R}^{s^1}	ed	\mathcal{R}^{s^2}	ed	\mathcal{R}^{s^3}	ed	\mathcal{R}^{s^4}	ed
The	the	1	wre	2	thc	2	rhe	1		
Hitch	litilirih	7	litilirih	7	iitilirih	7	litilirivh	7	litilirih	7
Hikers	ifumiaeiis	8	hiuhriaeiis	8	hiumiaeiis	8	iriumiaeiis	8	hiiumiaeiis	8
Guide	hmde	3	bmde	3	hmle	4	hmdl	4	kdde	3
to	tr	1	tt	1	lr	2				
the	lhe	1	ihe	1	lne	2	lwi	3		
Galaxy	tieleieiy	7	tieleieiy	7	tieleieiy	7	tieleeiey	7	tieleiuiy	7
has	hiis	2	heis	2	hies	2	niis	3	hiis	2
this	lhius	2	lhivs	2	lkhus	3	ihius	2	lhius	2
to	iirui	6	lirui	6	iirui	6	lriui	6	iurui	6
say	seey	2	seey	2	seiy	2	saey	1	beey	3
on	as	2	ar	2	ds	2				
the	lrre	3	irre	3	Trre	3	Thie	2	Thei	2
subject	sviuiisiedu	9	sviuiisieeli	9	sviuiisieefu	9	sviuiisiedu	10	sviuiisiedu	9
of	ef	1	if	1	di	2				
flying	flyisiiiy	6	flyisiiiy	6	flysuriiy	6	ffisiiisyy	8	flyiuiiyy	6
There	tireie	5	tireie	5	tirene	5	tireie	5	thieie	4
is	is	0	is	0	ib	1				
an	ais	2	air	2	ais	2	dis	3		
art	auit	3	auit	3	auir	4	aiit	3	duit	4
it	iiT	1	iiT	2	iut	1	iiT	1		
says	siiys	2	sieys	2	siiys	2	siiys	2	siiys	2
or	iurii	4	iurii	4	iiuri	4	iiur	4	vrii	4
rather	siatmeii	5	siatmeii	5	riatmeii	4	siarmeii	6	seetmeii	6
a	a	0	d	1						
knack	isuiiidii	9	isuiiidii	9	isuiiidii	9	isiiidi	9	biuiiidii	9
to	iir	3	iur	3	iir	3	lri	3		
flying	flyiuiy	3	flyiuiy	3	flyiuiy	4	flyiivy	3	flyiuiy	3
The	the	1	tne	2	thi	2	vhe	1		
knack	kireedsi	7	kireedsi	7	kiieedsi	7	kiieedi	7	kireedsi	7
lies	luiies	2	luiies	2	luiies	2	uiies	3	lvies	2
in	en	1	eu	2	in	0				
larning	teeimiihyi	9	teeimiihyi	9	teeimiihyi	9	teeimiihyi	9	teeimiihyi	9
how	hoau	2	hoai	2	hou	2	hdau	3	noau	3
to	livo	3	iivo	3	luvo	3	liio	3	livu	4
throw	ifsiiuiiu	9	ifsiiuiiu	9	ifsiiuiiu	9	ifsiiioviu	9	irsiiuiiu	9
yourself	yeureiiself	4	yeureiiself	4	yeueiiself	5	yeireiiself	5	yeuriiiself	4
at	afi	2	afu	2	edi	3	dfi	3		
the	iirse	4	Tisie	4	lirse	4	Tiise	4	iurse	4
ground	ysiuriiirud	9	ysiuriiirud	9	ysiuriiisud	9	ysiuiiirud	10	ysiuriiirel	10
and	auiid	3	aiiid	3	aiiel	4	auiid	3	duiid	4
miss	miiiss	2	miiivss	2	imiiss	2	miiis	3	mieiss	2
:	:		:		:		:		:	

A.3. Referenzdatensatz R_3

Testwort	\mathcal{R}^s	ed	\mathcal{R}^{s^1}	ed	\mathcal{R}^{s^2}	ed	\mathcal{R}^{s^3}	ed	\mathcal{R}^{s^4}	ed
The	The	0	Tne	1	Ihe	1	Tkc	2		
Hitch	Hitch	0	Hvtch	1	HiIh	2	VItch	2	Hitcn	1
Hikers	Himoevs	3	Himoeis	3	Himxavs	4	Himvevs	3	VWmoevs	5
Guide	Gual	3	Guul	3	Gaal	4	Gmde	2	tual	4
to	t	1	W	2						
the	me	2	mc	3	ne	2				
Galaxy	Galag	2	talag	3	Godag	4	tulag	4	Galxy	1
has	has	0	hus	1	nas	1	hac	1		
this	Hws	3	Hus	3	Kws	3	Hve	4		
to	tv	1	tV	1	Vv	2				
say	say	0	sag	1	jay	1	suy	1		
on	on	0	oh	1	vn	1				
the	Vhe	1	Vhf	2	VKe	2	Vhe	1		
subject	salyal	6	sabjod	4	sabjal	4	sulyal	5	salyal	6
of	d	2	f	1						
flyng	lynng	2	lyvmg	3	lynny	3	lyng	1	fgng	3
There	Tnea	3	Thea	2	Twwe	3	Tnla	4	Inea	4
is	os	1	vs	1	vS	2				
an	an	0	om	2	un	1				
art	art	0	arV	1	avt	1	urt	1		
it	t	1	w	2						
says	says	0	suys	1	sags	1	svap	3	Says	1
or	cr	1	vr	1	ov	1				
rather	raIwa	4	raIww	4	raIua	4	ralwa	4	ruIwa	5
a	a	0	o	1						
knack	bwam	4	bwum	5	buam	4	bwau	4	Pwam	4
to	w	2	t	1						
flyng	lywy	4	lgwy	5	lyuy	4	lymj	4	fywy	4
The	Ihe	1	The	0	IHe	2	Ilu	3		
knack	kwoak	3	kwack	1	kwaac	3	kuoak	3	kwvak	3
lies	ues	2	Ues	2	Vas	3	Lia	3		
in	m	2	n	1						
lerning	Ieammg	5	Ieammy	6	Voummg	6	Ilammg	6	Teammg	5
how	hoa	1	how	0	koa	2	hia	2		
to	to	0	Vo	1	tv	1				
throw	mrow	2	Hwow	3	mrrou	3	Hwvw	4	nrow	2
yourself	yawself	3	yuwself	2	yaaself	3	yawself	4	vjawsef	5
at	al	1	ai	1	ul	2				
the	Vhe	1	the	0	Vne	2	Vhc	2		
ground	grund	2	gramd	3	yrumd	3	gmivl	5	gvumd	3
and	awd	1	aad	1	uwd	2	aml	2		
miss	mis	1	mws	2	mip	2	xis	2		
⋮	⋮		⋮		⋮		⋮		⋮	

A.4. Referenzdatensatz R_4

Testwort	\mathcal{R}^s	ed	\mathcal{R}^{s^1}	ed	\mathcal{R}^{s^2}	ed	\mathcal{R}^{s^3}	ed	\mathcal{R}^{s^4}	ed
The	The	1	Tnc	2	The	0	lhc	2		
Hitch	Hitoh	1	Hitoh	1	Hitoh	1	Hiton	2	Hitih	1
Hikers	tiimoocri	7	riimoocri	7	tiimooers	6	tiimoocri	7	tiimooiri	7
Guide	Giiolc	4	Gicolc	5	Guiolc	3	Giiole	3	Giiotc	4
to	tu	1	tr	1	tu	1				
the	thc	1	tnc	2	rhc	2	twi	2		
Galaxy	Goolccroy	6	Goolccioy	6	Goolciroy	6	Goolccroy	6	toolccroy	7
has	hiis	2	hiii	3	hics	2	hcis	2	niis	3
this	tliis	2	tliis	2	tliis	2	tliis	2	tliii	3
to	tu	1	tr	1	uu	2				
say	rooiy	4	rooiy	4	roooy	4	roocg	5	rcoiy	4
on	on	1	on	0	eh	2				
the	thc	1	rhc	2	the	0	tnc	2		
subject	sicbifcct	5	siibifcct	5	sccbifcct	5	sicbifcct	5	sicbiicct	5
of	cf	1	if	1	cl	2				
flying	flijinig	3	fflyinig	3	fflyniig	4	flijcnig	4	fflyning	3
There	Tiierc	4	Tiicuc	5	Tiierc	4	Tiierc	3	Tiicri	4
is	is	0	is	0	ii	1				
an	ccn	2	icn	2	cin	2	ccn	2		
art	oirt	2	oirt	2	ourt	2	iirt	2	oiut	3
it	it	0	ot	0	ot	1				
says	rooys	3	sioys	2	rooyi	4	sicgs	3	sivys	2
or	ur	1	rr	1	uu	2				
rather	rocther	3	roithcr	3	roctncr	4	rocther	3	rocthcu	4
a	ci	2	ci	2	ii	2				
knack	biicicic	7	biioicic	7	biiiiicic	7	biicicii	7	biicicic	7
to	tu	1	uu	2	tr	1				
flying	flgiry	3	flyiry	2	flgiuy	3	llgiry	4	flgiry	3
The	thc	2	lhc	2	the	1	tnc	3		
knack	kiiccek	4	kiiccek	4	kiiccek	4	kiicick	4	kiiccek	4
lies	tics	2	lics	1	tici	3	ties	1	tucs	3
in	cn	1	in	0	ch	2				
larning	tccirniiy	7	tcciuuiiy	8	lccirniiy	6	tccirnify	7	tcccrniiy	7
how	hiii	3	hiui	3	hiio	3	niii	4	hiii	3
to	to	0	te	1	uo	1				
throw	rhrooi	3	rnrooi	4	rhroii	3	rtirow	2	rhuooi	4
yourself	ycicrsclf	4	yciirsclf	4	yiicrsclf	4	ycicrsclf	3	yiicrsclf	4
at	cct	2	cit	2	ict	2	cit	2		
the	tnc	2	thc	1	tni	2	unc	3		
ground	grouiicl	4	grouiicl	4	grouiil	4	guouiicl	5	gruuiicl	5
and	oiiccl	6	ociiccl	6	oiiiiil	6	iiiiccl	6	oiiccl	6
miss	miss	0	misi	1	miis	1	miss	0	niss	1
⋮	⋮		⋮		⋮		⋮		⋮	

Literatur

- [1] BELLMANN, R.: *Dynamische Programmierung und selbstanpassende Regelprozesse*. R. Oldenbourg Verlag, München, Wien, 1967.
- [2] MYERS, C.S. und L.R. RABINER: *A level building dynamic time warping algorithm for connected word recognition*. IEEE Trans. Acoustics, Speech, Signal Proc., ASSP-29, Seiten 284–297, April 1981.
- [3] RABINER, L. und B. JUANG: *Fundamentals of Speech Recognition*. Prentice Hall, 1993.
- [4] TAPPERT, C. C., C.Y. SUEN und T. WAKAHARA: *The State of the Art in On-Line Handwriting Recognition*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 12(8):787–808, August 1990.
- [5] T. CORMEN, C. LEISERSON, R. RIVEST: *Introduction to Algorithms*. The MIT Press, 1994.
- [6] THEODORIDIS, S. und K. KOUTROUMBAS: *Pattern Recognition*. Academix Press, 1999.