

# Kapitel 3

## Merkmalsextraktion

Unabhängig von der spezifischen Aufgabe eines Mustererkennungssystems ist es das Ziel mit Hilfe der Merkmalsextraktion die für eine Erkennung wesentliche Information aus den zur Verfügung stehenden Rohdaten zu gewinnen. Charakteristische Merkmale, welche sich für eine Klassenunterscheidung eignen, sollen hervorgehoben und dem eigentlichen Erkenner zugeführt werden. Zugleich sollten möglichst irrelevante Daten außer Betracht bleiben.

Diese allgemeine und zunächst einfach klingende Spezifikation der Merkmalsextraktion erweist sich bei näherer Betrachtung als eine Aufgabenstellung, deren Lösung bei nahezu jedem neuen Mustererkennungsproblem eine erneute Herausforderung darstellt. Hinsichtlich der speziellen Problemstellungen existieren in nur einigen Fällen optimierte Merkmalsextraktions-Verfahren, die sich als eine Art Standard-Lösung etabliert haben.

Das Fehlen von solchen Standard-Lösungen trifft ebenso für die Online-Handschrifterkennung zu. Das Finden von geeigneten Verfahren erschwert sich auch dadurch, dass einheitliche, öffentlich zugängliche Datenbasen und darauf durchgeführte Evaluierungen anhand derer verschiedene Merkmalsextraktions-Verfahren objektiv verglichen werden können, kaum existieren. Dennoch lassen sich in der Literatur Vorschläge für Merkmale zur Online-Handschrifterkennung finden, die grob in die zwei Gruppen 'Trajektorien-' [Yan95, Sch95] und 'Bitmap-' Merkmale [Man94] eingeteilt werden können.

Trajektorien-Merkmale lassen sich direkt aus einem zeitlich begrenzten Segment der Stiftrajektorie extrahieren. Dieser Merkmalstyp basiert damit auf einer zeitlich lokalen Betrachtungsweise. Bitmap-Merkmale hingegen werten das komplette Schriftbild in globaler Form aus, und werden erst extrahiert, nachdem die zu bearbeitende Schriftprobe vervollständigt wurde. Damit eröffnet sich die Möglichkeit Eigenschaften zu erfassen, die zwar in geometrischer Nachbarschaftsbeziehung zueinander stehen, aber zu völlig unterschiedlichen Zeitpunkten eingegeben wurden (z. B. zeitlich verzögert gesetzte i-Punkte

oder t-Striche).

Die gewählte Struktur dieses Kapitels spiegelt diese Grobunterteilung der Verfahren wieder. Abschnitt 3.1 beschreibt einige vielversprechende Extraktionsverfahren für Trajektorien-Merkmale, während Abschnitt 3.2 eine Übersicht über die untersuchten Bitmap-Merkmale liefert. Im Abschnitt 3.3 werden die behandelten Verfahren einzeln, sowie in möglichen Kombinationen evaluiert. Bei den hier dargestellten Verfahren, handelt es sich lediglich um eine Auswahl der erfolgreichsten Ansätze. Untersuchungen zu weiteren in Betracht gezogenen Verfahren können [Hü99] entnommen werden.

## 3.1 Trajektorienmerkmale

Ausgangspunkt für die Extraktion von Trajektorien-Merkmalen sind die neuabgetasteten Vektorsequenzen der Rohdaten (Glg. (2.1)). Zu jedem Abtastzeitpunkt  $k$  wird durch eine Merkmalsextraktion (ME) aus den vorverarbeiteten Kartesischen Daten, ggf. unter Berücksichtigung des Stiftdrucks, ein Merkmalsvektor  $\vec{m}$  generiert:

$$(x_{Anf}, y_{Anf}, x_{End}, y_{End}, \tilde{p})(k) \xrightarrow{ME} \vec{m}(k). \quad (3.1)$$

Überwiegend wird bei der Berechnung eines Merkmalsvektors zum Abtastzeitpunkt  $k$  zudem ein zeitlicher Kontext  $\{k - k', \dots, k, \dots, k + k'\}$  berücksichtigt

$$\left( \begin{array}{c} (x_{Anf}, y_{Anf}, x_{End}, y_{End}, \tilde{p})(k - k') \\ \dots \\ (x_{Anf}, y_{Anf}, x_{End}, y_{End}, \tilde{p})(k) \\ \dots \\ (x_{Anf}, y_{Anf}, x_{End}, y_{End}, \tilde{p})(k + k') \end{array} \right) \xrightarrow{ME} \vec{m}(k), \quad (3.2)$$

der das Trajektorienstück zwischen den Abtastpunkten  $k - k'$  und  $k + k'$  einbezieht. Das Trajektorienstück wird als 'Fenster'  $\mathcal{F}$  bezeichnet, wobei die Fenstergröße  $F$  mit  $2k' + 1$  angegeben werden kann.

Eine neuabgetastete Schriftprobe, bestehend aus  $K$  Abtastvektoren, führt somit zu genau  $K$  aufeinanderfolgenden Merkmalsvektoren, deren Dimensionalität von der gewählten Extraktionsmethode selbst abhängt. Die aus einer Schriftprobe generierte Folge von Merkmalsvektoren wird im folgenden auch als Merkmalsstrom bezeichnet.

### 3.1.1 Kettenkodierung

Bei der Online-Handschrifterkennung ist eine komplette Schriftprobe durch die zeitliche Funktion der Stiftpositionen in Kartesischen- oder Zylinder-Koordinaten definiert. Nahe-

liegend erscheint es daher, die (planare) Trajektorie in parametrischer Form zu beschreiben [Yan95]. Da lediglich eine Unterscheidung zwischen aufgesetztem oder abgehobenem Stift zu erfolgen hat, gibt der ebenfalls abgetastete Stiftdruck  $\tilde{p}$  hinreichenden Aufschluß über die Stiftposition in der dritten Dimension.  $\tilde{p} > 0$  bedeutet, dass die Stiftspitze das Digitalisiertablett berührt, während  $\tilde{p} \leq 0$  anzeigt, dass der Stift abgehoben ist.

Ist eine planare Kurve mit  $x(t)$  und  $y(t)$  bekannt, stellt die Funktion der Trajektorientangente  $\alpha(t)$  mit

$$\alpha(t) = \arctan\left(\frac{\dot{y}(t)}{\dot{x}(t)}\right) \text{ für } \dot{x}(t) \neq 0, \quad (3.3)$$

eine kompakte und bis auf die absolute Position der Schriftprobe vollständige Beschreibung der Trajektorie dar. Durch Abtastung und Neu-Abtastung wird schließlich die Trajektorientangente durch die Trajektoriensekante  $\alpha(k)$  zwischen den Anfangs- und Endpunkten  $(x_0(k), y_0(k))$  und  $(x_1(k), y_1(k))$  des Abtastvektors mit

$$\alpha(k) = \arctan\left(\frac{y_1(k) - y_0(k)}{x_1(k) - x_0(k)}\right) \text{ für } x_1(k) \neq x_0(k), \quad (3.4)$$

approximiert. Da der Abstand zwischen zwei aufeinander folgenden Punkten durch die Neuabtastung quasi konstant ist, läßt sich der neuabgetastete Schriftzug allein durch  $\alpha(k)$  rekonstruieren. Würde  $\alpha(k)$  nun direkt als Merkmal verwendet, hätte dies den entscheidenden Nachteil, dass sich bei 0 bzw.  $2\pi$  eine Unstetigkeit ergäbe. Dies würde dazu führen, dass  $\alpha(k)$  bei Abtastvektoren in positiver  $x$ -Richtung mit horizontaler Ausrichtung bei kleinen Abweichungen zwischen 0 und  $2\pi$  springen würden. Aus diesem Grund wird die nach Glg. (3.4) ermittelte Sekantensteigung durch Angabe des Sinus- und Cosinuswinkels als Merkmalsvektor verwendet [Rig96a]:

$$\vec{m}_\alpha(k) = \begin{pmatrix} \sin(\alpha(k)) \\ \cos(\alpha(k)) \end{pmatrix}. \quad (3.5)$$

In analoger Weise läßt sich ebenso ein Merkmalsvektor berechnen, um Richtungsänderungen explizit zu erfassen. Mit  $\Delta\alpha(k) = \alpha(k+1) - \alpha(k)$  ist dazu lediglich in Glg. (3.5)  $\alpha(k)$  durch  $\Delta\alpha(k)$  zu ersetzen:

$$\vec{m}_{\Delta\alpha}(k) = \begin{pmatrix} \sin(\Delta\alpha(k)) \\ \cos(\Delta\alpha(k)) \end{pmatrix}. \quad (3.6)$$

Die anschließende Quantisierung der durch Glg. (3.5) gewonnenen Merkmalsvektoren ergibt eine spezielle Form einer Kettenkodierung. Im Gegensatz zu einer Standard-Kettenkodierung [Abm94] wird auf eine statische a priori Einteilung der Richtungsvektoren verzichtet. Vielmehr wird hierbei das Codebuch für die Richtungsquantisierung aus den Trainingsdaten gelernt.

Es ist leicht ersichtlich, dass es sich bei dieser Extraktionsmethode um ein durch Glg. (3.1) typisiertes Verfahren handelt. Ein zeitlicher Kontext läßt sich jedoch

auch hier erfassen. Dies geschieht, indem eine Folge von Merkmalsvektoren  $\vec{m}(k - k'), \dots, \vec{m}(k), \dots, \vec{m}(k + k') = \vec{m}'(k)$  zu einem *Multi-Frame*-Vektor  $\vec{m}'(k)$  zusammengefasst wird, und dieser Multi-Frame-Vektor schließlich dem nachfolgenden Vektorquantisierer präsentiert wird.

### 3.1.2 Diskrete Cosinus Transformation der Trajektorie

Betrachtet man den zeitlichen Verlauf der abgetasteten Stiftposition in Abb. 3.1, separiert nach  $x$  und  $y$ , bietet sich auf Grund der zu beobachtenden Schwingungen eine weitere Repräsentation der Stiftbewegungen durch eine Schwingungsanalyse an. Als vorteilhaft in verschiedenen Bereichen der Mustererkennung oder der Bilddatenkompression (JPEG, MPEG) hat sich die Diskrete Cosinus Transformation (DCT) erwiesen. Die DCT soll hiermit auch für die Online-Handschrifterkennung eingehender untersucht werden. Insbesondere ist bei der DCT die Kovarianz-mindernde Eigenschaft der resultierenden DCT-Koeffizienten hervorzuheben [Pra91].

Der in Abb. 3.1 zu sehende, mit  $k$  tendenziell steigende Verlauf von  $x$ , spiegelt die überwiegend in positive  $x$ -Richtung verlaufende Schreibrichtung wieder. Mit der Verwendung von Sub-Wort-Modellen für die spätere Erkennung kann die absolute  $(x, y)$ -Position eines Buchstabens innerhalb eines Wortes für die Merkmalsextraktion jedoch nicht relevant sein. Dies gilt darüber hinaus auch für einen entstehenden Offset in  $y$ , hervorgerufen

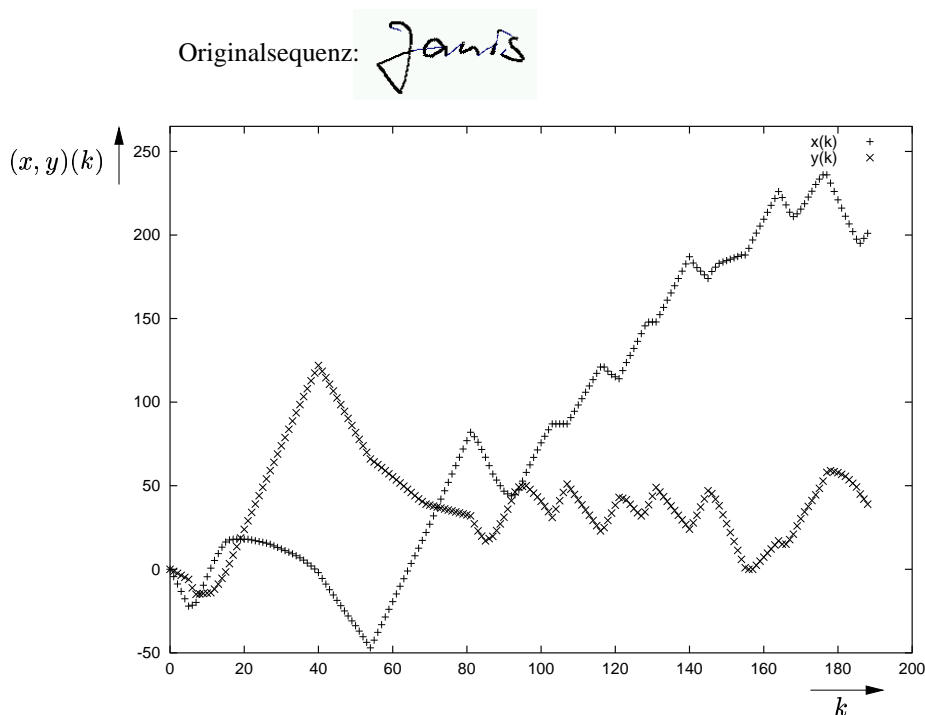


Abbildung 3.1: Verlauf von  $(x, y)(k)$  einer Schriftprobe

durch einen Zeilenwechsel. Aus diesem Grund wird statt des gezeigten absoluten Verlaufes von  $x$  und  $y$  die Analyse innerhalb eines Fensters durchgeführt (entsprechend Glg. (3.2)).

### Fensterung

Bei den Fenster-basierten Extraktionsverfahren sind einige Besonderheiten zu beachten, die stellvertretend für weitere Fenster-Verfahren am Beispiel der DCT erklärt werden. Dazu betrachte man einen Ausschnitt bestehend aus  $F$  Abtastvektoren einer Abtastsequenz  $V = \{\vec{v}(1), \dots, \vec{v}(k - k'), \dots, \vec{v}(k), \dots, \vec{v}(k + k'), \dots, \vec{v}(K)\}$  der Länge  $K$ , mit  $K \gg F = 2k' + 1$ .  $\vec{v}(k)$  bezeichnet dabei den  $k$ -ten Abtastvektor zwischen den Punkten  $\mathbf{v}_0(k) = (x_0, y_0)(k)$  und  $\mathbf{v}_1(k) = (x_1, y_1)(k)$ . Da in einer solchen Sequenz der Endpunkt  $\mathbf{v}_1$  des einen Abtastvektors mit dem Anfangspunkt  $\mathbf{v}_0$  des nachfolgenden Abtastvektors identisch ist, ist es ausreichend in die Berechnungen lediglich die Anfangspunkte einzubeziehen. Das für die Merkmalsberechnungen nötige Fenster um den  $k$ -ten Abtastvektor  $\vec{v}(k)$  wird demnach auf eine Punktmenge  $\mathcal{F}_k$  reduziert.

Unter besonderer Berücksichtigung von Sequenzanfang ( $k = 0, \dots, k'$ ) und -Ende ( $k = K - k', \dots, K$ ) ergibt sich

$$\begin{aligned} \mathcal{F}_k &= \{\mathbf{v}_0(k - k'), \mathbf{v}_0(k - k' + 1), \dots, \mathbf{v}_0(k + k' - 1), \mathbf{v}_0(i + k')\} && \text{für } k \in [k', K - k'] \\ \mathcal{F}_k &= \{\mathbf{v}_0(0), \dots, \mathbf{v}_0(2k')\} && \text{für } k \in [0, k'] \\ \mathcal{F}_k &= \{\mathbf{v}_0(K - 2k'), \dots, \mathbf{v}_0(K)\} && \text{für } k \in [K - k', K]. \end{aligned} \quad (3.7)$$

Wie bereits oben erwähnt, sollte die Merkmalsextraktion unabhängig von der absoluten Position der Trajektorie erfolgen. Werden die Punkte eines Fensters hingegen auf einen gleitenden Referenzpunkt bezogen, lassen sich zudem Effekte ausblenden, die durch die vorherrschende Schreibrichtung (von links nach rechts) oder einen Zeilenwechsel hervorgerufen werden. Als ein solcher Referenzpunkt bietet sich natürlich der mittlere Punkt  $\mathbf{v}_k$  an, zu dem das Fensters  $\mathcal{F}_k$  gebildet wurde. Setzt sich, allgemein betrachtet, das Fenster  $\mathcal{F}_k$  aus den Punkten  $\{\mathbf{f}'_0, \mathbf{f}'_1, \dots, \mathbf{f}'_{F-1}\}$  zusammen, die entsprechend Glg. (3.7) bestimmt wurden, können diese nun um den Vektor  $([0, 0]^T - \mathbf{f}'_{k'})$  in Richtung des Koordinatenursprungs verschoben werden, wodurch sich

$$\mathcal{F}_k = \{\mathbf{f}_0, \dots, \mathbf{f}_{k'}, \dots, \mathbf{f}_{F-1}\} \quad (3.8)$$

mit  $\mathbf{f}_{k'} = [0, 0]^T$  ergibt. Bei allen folgenden Verfahren wird ein solches relatives Fenster zur Berechnung der Trajektorienmerkmale benutzt.

Für die eigentliche Merkmalsextraktion lassen sich nun aus einem relativen Fenster die DCT-Koeffizienten berechnen. Mit der Aufteilung der Punkte  $\mathbf{f}$  nach  $x$  und  $y$  läßt sich mit

$$m_{DCT_x}(i) = \sqrt{\frac{2}{F}} \sum_{k=0}^{F-1} f_{xk} \cos \frac{(2k+1)\pi i}{2F}, \quad (3.9)$$

$$m_{DCT_y}(i) = \sqrt{\frac{2}{F}} \sum_{k=0}^{F-1} f_{yk} \cos \frac{(2k+1)\pi i}{2F}$$

für das  $k$ -te Fenster ein Merkmalsvektor

$$\vec{m}_{DCT}(k) = [m_{DCT_x}(0), \dots, m_{DCT_x}(i), m_{DCT_y}(0), \dots, m_{DCT_y}(i)]^T \quad (3.10)$$

mit  $i + 1$  Merkmalen pro Richtung definieren.

Der Vorteil der Betrachtung eines relativen Fensters (Verschiebung zum Koordinatenursprung) wird bei der Spektralanalyse noch nicht offensichtlich. Ein ähnlicher Effekt hätte hier sicherlich durch eine Unterdrückung der Tiefpaßanteile der Transformierten erzielt werden können. Bei den nachfolgend beschriebenen Verfahren hingegen ist die relative Betrachtungsweise unerlässlich. Die Berücksichtigung der Vergleichbarkeit verschiedener Verfahren, wie auch die Berücksichtigung von Implementierungsaspekten sprechen schließlich für eine konsistente Fensterung, auch bei der DCT.

### 3.1.3 Bézierkurve

Kehrt man aus der 'Frequenzwelt' zurück zu einer geometrischen Betrachtung einer Kurve, ergeben sich potenziell weitere Ansatzpunkte, um ein Kurvenstück einerseits möglichst kompakt, andererseits möglichst genau zu beschreiben. Dazu lassen sich beispielsweise Methoden aus dem Computer Aided Geometric Design (CAGD) heranziehen [Far94, Ram97]. Gegenüber anderen Approximationsverfahren haben Bézierkurven und Splines wichtige CAGD-relevante Eigenschaften, sodass mit ihnen einfach Kurven und Flächen zu gestalten sind.

Bézierkurven oder Splines werden anhand von Kontrollpunkten konstruiert. In CAGD-Systemen, wie in konventionellen Zeichenprogrammen, kann der Benutzer durch einfaches Hinzufügen oder Löschen von Kontrollpunkten und insbesondere mit der Veränderung ihrer Lage schnell und präzise Kurven- bzw. Flächenformen konstruieren und modifizieren. Dabei ist es möglich mit wenigen Kontrollpunkten komplexe Formen zu bilden. Wird dieser Gedanke auf die Handschrifterkennung übertragen, sollten diese Kontrollpunkte aussagekräftige Merkmale über die Form der Stiftrajektorie bieten.

Während man bei CAGD-Anwendungen versucht Kurven mit Kontrollpunkten zu konstruieren, präsentiert sich das Problem für die Merkmalsextraktion in genau umgekehrter Weise. Anhand eines gegebenen Kurvenstückes sind die Kontrollpunkte in der Form zu ermitteln, dass die Approximierende das Kurvenstück möglichst exakt nachbildet.

Bézierkurven und Splines zeichnen sich darüber hinaus durch einige Vorteile aus, die speziell für die Online-Handschrifterkennung relevant sind. Insbesondere sind Bézierkurven

- variationsvermindernd, wodurch sich eine insgesamt glättende Wirkung auf die Trajektorie einstellt.
- Bézierkurven und Splines sind relativ einfach und robust zu konstruieren und

- sind invariant unter affinen Abbildungen. D. h., normalisierende Operationen, wie Skalierungen, Scherungen oder Rotationen können prinzipiell auch nach der Merkmalsextraktion auf die Merkmale angewendet werden.

Bevor die Approximation einer Trajektorie durch eine Bézierkurve beschrieben werden kann, ist es erforderlich deren Konstruktion in Grundzügen zu verstehen. Eine Bézierkurve  $J$ -ten Grades wird allgemein mittels folgender Summe definiert:

$$\vec{B}_J(t) = \sum_{j=0}^J r_{Jj}(t) \vec{a}_j. \quad (3.11)$$

Bei einer Bézierkurve handelt es sich folglich um eine gewichtete Summe der Kontrollpunk-

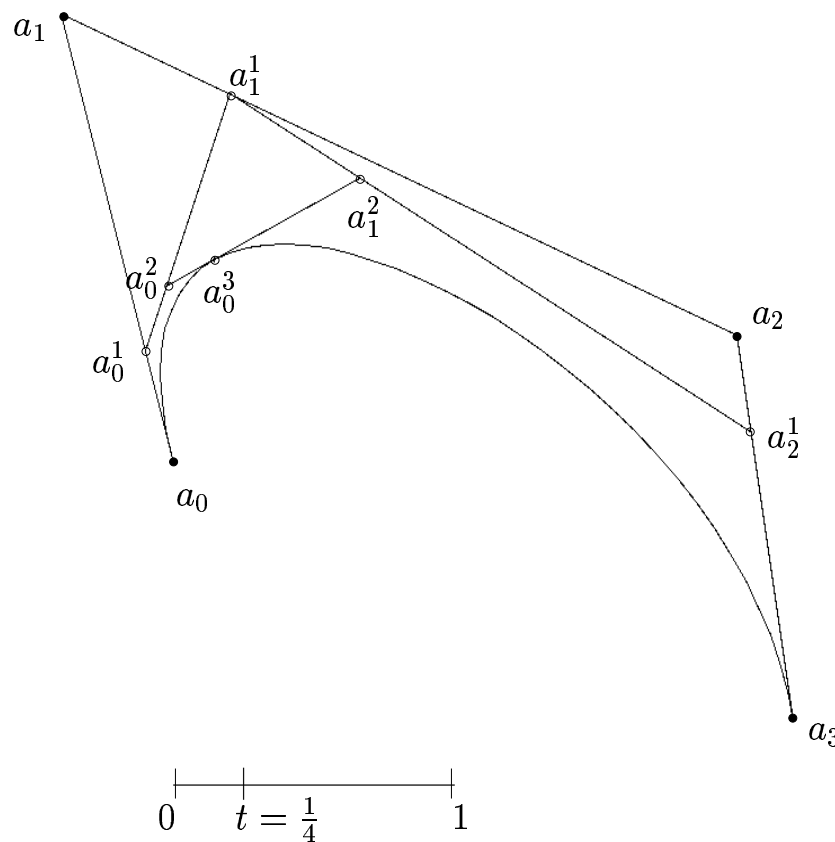


Abbildung 3.2: Konstruktion einer Bézierkurve

te  $\vec{a}_j$ . Bei einem Grad  $J = 1$  ergibt sich die lineare Interpolation zwischen den Kontrollpunkten  $\vec{a}_0$  und  $\vec{a}_1$ . Als Gewichtungsfaktoren werden die Bernstein-Operatoren

$$r_{Jj}(t) = \binom{J}{j} t^j (1-t)^{J-j} \quad (3.12)$$

verwendet. Man beachte, dass die Bernstein-Operatoren ihrerseits wiederum von dem Argument  $t$  abhängen. Für das Argument  $t \in \mathfrak{R}$  gilt hier  $0 \leq t \leq 1$ . Die Skalierung erfolgt über die Position der Kontrollpunkte. Des weiteren genügen die Bernstein-Operatoren der

allgemeinen Normierungsbedingung

$$\sum_{j=0}^J r_{Jj}(t) = 1, \quad (3.13)$$

wodurch die Bézierkurve auch als baryzentrische Kombination gilt. Die konstruierte Kurve liegt bei nicht negativen Gewichtungen  $r_{Jj}(t)$  innerhalb der durch die Kontrollpunkte aufgespannten konvexen Hülle (Abb. 3.2).

Mit dem de-Casteljau-Algorithmus [Far94] lassen sich in rekursiver Form Zwischenpunkte  $\vec{a}_j^l(t)$  berechnen, mit

$$a_j^l(t) = (1-t)a_j^{l-1}(t) + a_{j+1}^{l-1}(t), \quad (3.14)$$

von denen dann  $\vec{a}_0^J(t)$  genau auf der Bézierkurve  $\vec{B}_J$  liegt, sodass  $\vec{a}_0^J(t) = \vec{B}_J(t)$  gilt.

Vorausgesetzt, die Kontrollpunkte  $\vec{a}_0, \vec{a}_1, \dots, \vec{a}_J$  sind gegeben, können die Zwischenpunkte wiederum durch die Bernstein-Operatoren vom Grad  $l$  ausgedrückt werden:

$$\vec{a}_j^l(t) = \sum_{i=0}^l r_{Ji}(t) \vec{a}_{i+j} \quad \text{mit} \quad \begin{cases} l \in \{0, \dots, J\} \\ j \in \{0, \dots, J-l\} \end{cases} \quad (3.15)$$

Abb. 3.2 zeigt eine Bézierkurve vom Grad 3 mit den gegebenen Kontrollpunkten  $\vec{a}_0, \vec{a}_1, \vec{a}_2, \vec{a}_3$  und den konstruierten Zwischenpunkten. Mit der Konvention  $\vec{a}_j^0(t) = \vec{a}_j$  läßt sich aus Abb. 3.2 auch ablesen, dass der Zwischenpunkt  $\vec{a}_j^l$  in dem selben Verhältnis zu  $\vec{a}_j^{l-1}$  und  $\vec{a}_{j+1}^{l-1}$  steht, wie der Kurvenparameter  $t$  zu 0 und 1.

Sehr anschaulich wird die Konstruktion der Bézierkurve in Abb. 3.3, wenn man sich die Konstruktionslinien zwischen Kontroll- und Zwischenpunkten als Stabkonstruktion vorstellt. Die Kontrollpunkte stehen fest, während die Schnittpunkte der Verbindungslinien beidseitig gleitend gelagert sind. Verschiebt man das dehnbare, bewegliche innere Stabwerk unter Berücksichtigung gleicher Schnittverhältnisse und läßt man den Punkt  $\vec{a}_0^3$  im gleichen Schnittverhältnis zu den übrigen Verbindungslinien auf seiner Verbindungslinie mitgleiten, zeichnet der Punkt  $\vec{a}_0^3$  die Bézierkurve.

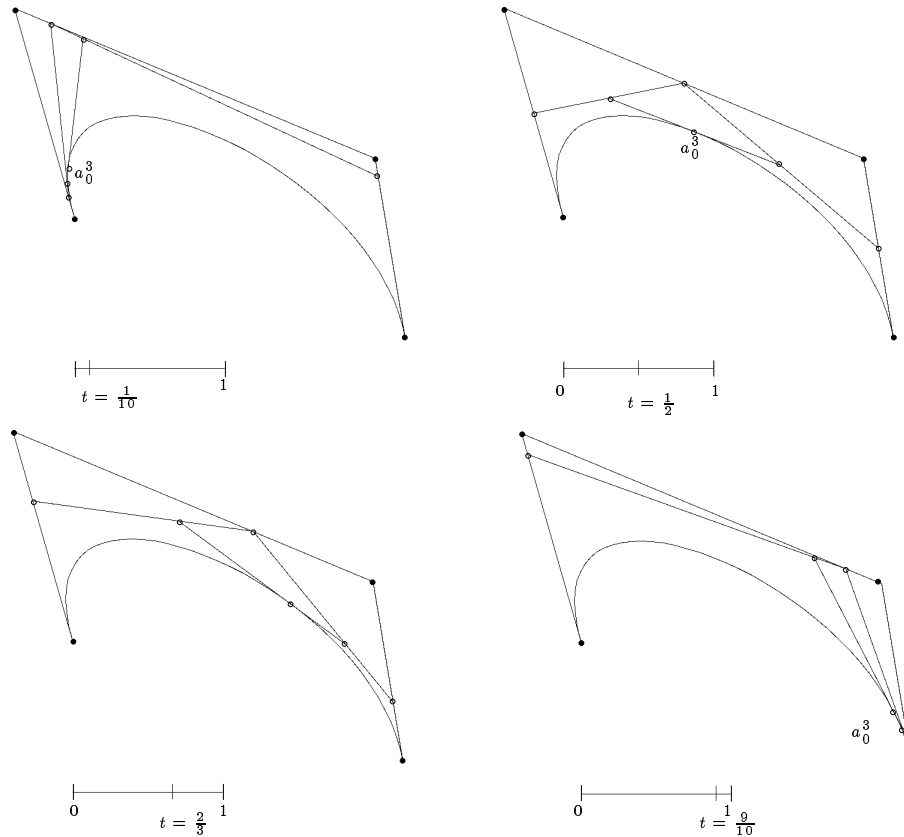
Die entscheidende Frage, wie nun aus einem abgetasteten Kurvenstück die Kontrollpunkte zur Nutzung als Merkmale bestimmt werden, ist allein mit den bisherigen Erläuterungen noch nicht zu beantworten.

Um diese wesentliche Frage schließlich zu klären, wird wieder von einem relativen Fenster gemäß Glg. (3.8) ausgegangen, welches die Abtastpunkte  $\mathbf{f}_k$  des Trajektorienstückes, verschoben zum Koordinatenursprung enthält. Soll nun dieses Trajektorienstück beispielsweise durch vier Kontrollpunkte beschrieben werden, ist eine Approximation durch eine Bézierkurve vom Grade 3 zu wählen. Aufgetrennt nach  $x$  und  $y$  ergäbe dies einen Merkmalsvektor, bestehend aus 8 Komponenten.

Für die Bézierkurve dritten Grades ist folglich

$$\vec{B}_3(t_k) = \begin{bmatrix} x(t_k) \\ y(t_k) \end{bmatrix} \stackrel{!}{=} \begin{bmatrix} f_{xk} \\ f_{yk} \end{bmatrix} = \mathbf{f}_k \quad \text{mit } t_k = \frac{k}{F-1} \quad \text{und } k = 0, \dots, F-1 \quad (3.16)$$



Abbildung 3.3: Bézierkurve -  $\vec{B}_3(t)$ 

zu fordern. Für jeden diskreten Zeitpunkt  $t_k$  ist weiterhin nach Glg. (3.11)

$$\vec{B}_3(t_k) = \sum_{j=0}^3 r_{3,j}(t_k) \vec{a}_j, \quad (3.17)$$

bzw. unter Einbeziehung der Approximationsforderung (Glg. (3.16))

$$\begin{bmatrix} x(t_k) \\ y(t_k) \end{bmatrix} = \sum_{j=0}^3 r_{3,j}(t_k) \begin{bmatrix} a_{xj} \\ a_{yj} \end{bmatrix} \quad (3.18)$$

zu lösen. Eine detailliertere Darstellung dieses Zusammenhangs, separiert nach  $x$ - und  $y$ -Richtung führt zu einem linearen Gleichungssystem:

$$\begin{bmatrix} x(t_0) \\ \vdots \\ x(t_{F-1}) \end{bmatrix} = \begin{bmatrix} r_{3,0}(t_0) & r_{3,1}(t_0) & r_{3,2}(t_0) & r_{3,3}(t_0) \\ \vdots & \vdots & \vdots & \vdots \\ r_{3,0}(t_{F-1}) & r_{3,1}(t_{F-1}) & r_{3,2}(t_{F-1}) & r_{3,3}(t_{F-1}) \end{bmatrix} \begin{bmatrix} a_{x0} \\ a_{x1} \\ a_{x2} \\ a_{x3} \end{bmatrix} \quad (3.19)$$

$$\vec{x} = R \vec{a}_x,$$

bzw. in analoger Weise zu

$$\vec{y} = R \vec{a}_y \quad (3.20)$$

für die  $y$ -Richtung. In der Regel stehen den 4 Kontrollpunkten mehr als 4 Abtastpunkte zur Verfügung. Das Gleichungssystem ist damit überbestimmt. Häufig handelt es sich darüber hinaus bei derartigen Gleichungssystemen um schlecht konditionierte Probleme [Häm89]. D. h. geringe Änderungen in den zugrunde liegenden Daten führen zu erheblichen Änderungen des Ergebnisses. Bei der Wahl des Lösungsverfahrens sollte dies daher entsprechend berücksichtigt werden.

Ein Verfahren, welches bei dieser Art von Problemen gute Ergebnisse liefert, ist die Singulärwertzerlegung [Häm89]. Dabei wird die Matrix  $R$  in der Form

$$R = U\Sigma V^T \quad (3.21)$$

in die orthogonalen Matrizen  $U \in \mathfrak{R}^{(F,F)}$  und  $V \in \mathfrak{R}^{(4,4)}$  zerlegt, sowie in die  $(F \times 4)$ -Diagonalmatrix  $\Sigma$ .

Zur Lösung des Gleichungssystems wird zunächst ein äquivalentes Minimierungsproblem

$$\|R\vec{a}_x^* - \vec{x}\|_2^2 = \min_{\vec{a}_x \in \mathbb{R}^4} \|R\vec{a}_x - \vec{x}\|_2^2 \quad (3.22)$$

aufgestellt, mit der Euklid'schen Norm  $\|\cdot\|_2$ . Es läßt sich weiterhin zeigen, dass ein solches äquivalentes Minimierungsproblem stets lösbar ist [Häm89]. Unter Verwendung der Matrixzerlegung lassen sich mit  $d_i = +\sqrt{\lambda_i}$ , also der Wurzel der Eigenwerte von  $R^T R$ , alle Lösungen durch

$$\vec{a}_x^* = \sum_{i=1}^4 \frac{1}{d_i} [U^T \vec{x}]_i \vec{v}^i + \sum_{i=5}^n z_i \vec{v}^i \quad (3.23)$$

angeben. Eine ausgezeichnete Lösung des äquivalenten Minimierungsproblems ist die Pseudonormallösung

$$\vec{a}_x^* = \sum_{i=1}^4 \frac{1}{d_i} [U^T \vec{x}]_i \vec{v}^i, \quad (3.24)$$

da sie die Lösung mit der minimalen Euklid'schen Norm darstellt. Existenz und Eindeutigkeit der Pseudonormallösung sind ebenfalls garantiert.

Bei konstanter Fenstergröße und Merkmalsanzahl ist die Matrix  $R$  unabhängig von den zu verarbeitenden Abtastpunkten. Für die Merkmalsextraktion ist dies von großem Vorteil, da für jede Kombination Fenstergröße/Merkmalsanzahl die Singulärwertzerlegung vorab durchgeführt werden kann. Die eigentliche Merkmalsextraktion ist dann die Berechnung der Pseudonormallösung nach Glg. (3.24).

Mit Glg. (3.24) wird schließlich die Hälfte des resultierenden Merkmalsvektors, durch die  $x$ -Koordinaten der Kontrollpunkte besetzt. Analog dazu wird die zweite Hälfte durch die  $y$ -Koordinaten der Kontrollpunkte  $\vec{a}_y^*$  besetzt:

$$\vec{m}_{Bezier} = [a_{x,0}^*; \dots; a_{x,3}^*; a_{y,0}^*; \dots; a_{y,3}^*]^T \quad (3.25)$$

Zur visuellen Kontrolle der Merkmalsextraktion können nun die so erzeugten Merkmale wiederum als Kontrollpunkte verwendet werden, aus denen sich eine Bézierkurve rekonstruieren läßt. Somit bietet sich ein Vergleich der originalen Abtastsequenz und der rekonstruierten

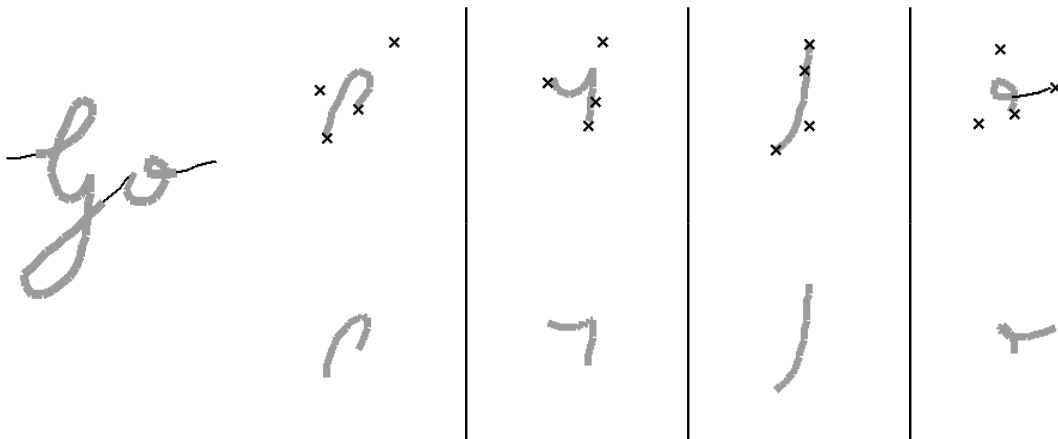


Abbildung 3.4: Abgetastete Trajektorienstücke des Wortes 'Go' mit approximierten Kontrollpunkten und rekonstruierten Bézier-Kurven

Trajektorie an. Dies ist in Abb. 3.4 dargestellt. Die obere Zeile in Abb. 3.4 zeigt die Abtastsequenz mit den daraus extrahierten Kontrollpunkten. Die untere Zeile gibt das dazugehörige Rekonstruktionsergebnis wieder. Erstaunlich ist hierbei, dass es mit einer relativ kleinen Anzahl von Kontrollpunkten möglich ist, komplexe Kurvenverläufe weitgehend fehlerfrei zu modellieren. Ob, und in wie weit sich dies auch auf die Erkennungsgenauigkeit auswirkt, werden später die Ergebnisse zeigen.

Mit der Verwendung von Bézierkurven eröffnet sich eine weitere interessante Option. Bei den abgetasteten Trajektorien handelt es sich um stückweise lineare Funktionen, die im allg. nicht stetig differenzierbar sind. Die Bildung von partiellen Ableitungen, über mehrere Abtastvektoren hinweg, ist damit nicht möglich. Für Bézierkurven, und damit auch für approximierte Trajektorien, existiert hingegen die Möglichkeit Ableitungen zu berechnen. Ausgehend von den berechneten Kontrollpunkten können diese in die Ableitungsvorschriften für Bézierkurven eingesetzt werden:

$$\begin{aligned}
 (\vec{B}_{x3})'(t) &= 3 \sum_{j=0}^2 (\mathbf{a}_{x(j+1)} - \mathbf{a}_{xj}) t^j (1-t)^{2-j} \\
 (\vec{B}_{y3})'(t) &= 3 \sum_{j=0}^2 (y_{(j+1)} - \mathbf{a}_{yj}) t^j (1-t)^{2-j} \\
 (\vec{B}_{x3})''(t) &= 6 [(\mathbf{a}_{x2} - 2\mathbf{a}_{x1} + \mathbf{a}_{x0})(1-t) + (\mathbf{a}_{x3} - 2\mathbf{a}_{x2} + \mathbf{a}_{x1})t] \\
 (\vec{B}_{y3})''(t) &= 6 [(\mathbf{a}_{y2} - 2\mathbf{a}_{y1} + \mathbf{a}_{y0})(1-t) + (\vec{b}_{y3} - 2\mathbf{a}_{y2} + \mathbf{a}_{y1})t].
 \end{aligned} \tag{3.26}$$

Mit  $t = \frac{1}{2}$  können so die Ableitungen zu dem mittigen Abtastpunkt  $\mathbf{f}_{k'}$  des betreffenden Fensters gebildet werden.

### 3.1.4 B-Splines

B-Splines werden - zumindest in CAGD-Anwendungen - gegenüber Bézierkurven bevorzugt. Besonders werden für diese Anwendungen die einfachen Konstruktions- und Modifi-

kationsmöglichkeiten von B-Splines geschätzt. Die Gestalt von B-Splines wird, wie die von Bézierkurven, mit Hilfe von Kontrollpunkten gesteuert. Potenziell können sich B-Splines damit auch für die Modellierung von Trajektorienstücken eignen, wobei sich wiederum die Kontrollpunkte als Merkmale verwenden lassen.

Die Konstruktion eines kubischen B-Splines  $\vec{S}_3$  verläuft anscheinend ähnlich zu der Bildung von Bézierkurven. Dazu werden die Kontrollpunkte  $\mathbf{a}_j$ , gewichtet durch einen Träger, aufsummiert:

$$\vec{S}_3 = \sum_{j=0}^3 \eta_{j,3}(t_k) \mathbf{a}_j \quad (3.27)$$

Bei diesem Träger handelt es sich um normalisierte, kubische B-Splines, die wie folgt berechnet werden:

$$\begin{aligned} \eta_{0,3} &= \frac{1}{6}(1-t)^3, & \eta_{1,3} &= \frac{1}{2}t^3 - t^2 + \frac{3}{2}, \\ \eta_{2,3} &= \frac{1}{2}t^3 - \frac{1}{2}t^2 + \frac{1}{2}t + \frac{1}{6}, & \eta_{3,3} &= \frac{1}{6}t^3, \end{aligned} \quad (3.28)$$

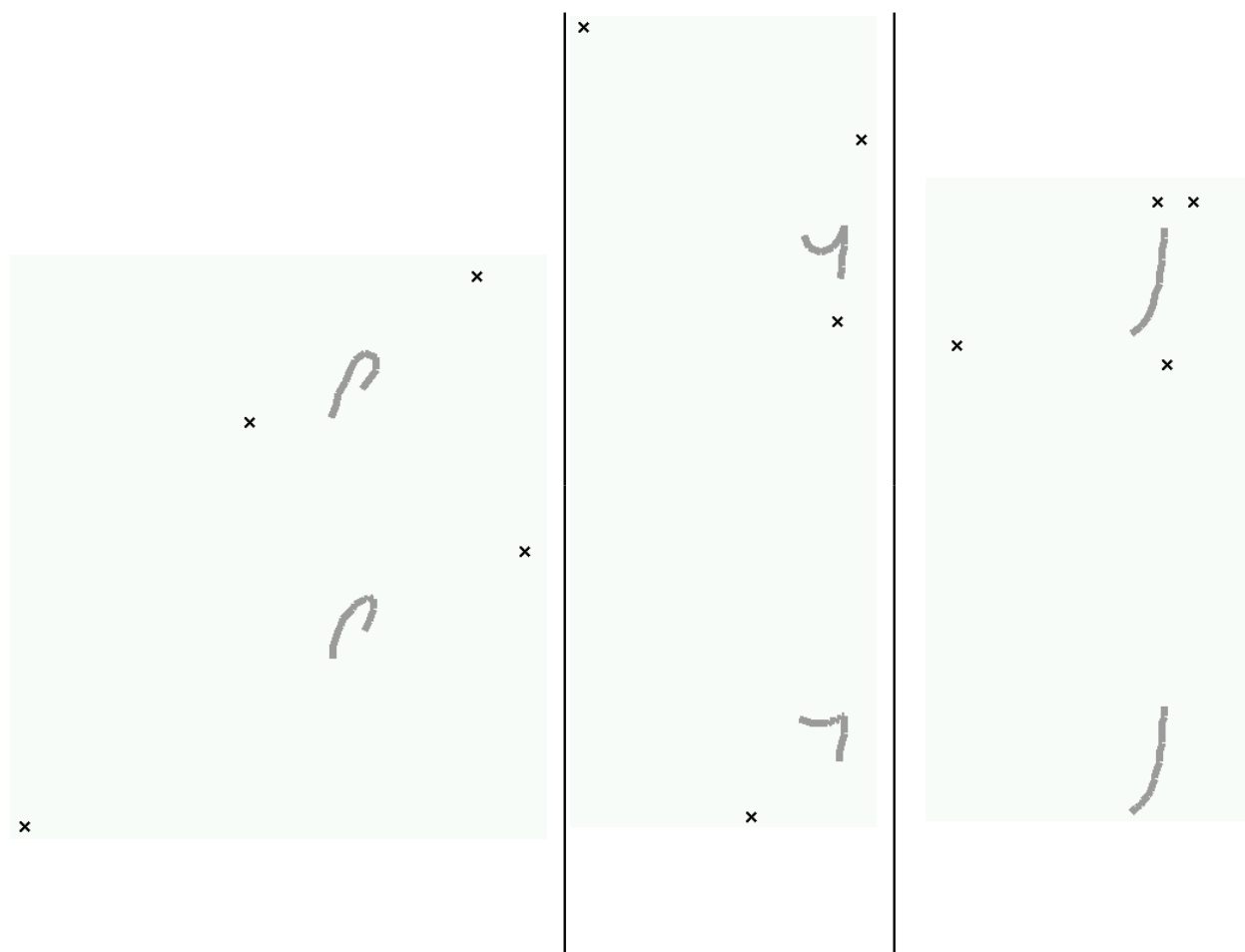


Abbildung 3.5: Abgetastete Trajektorienstücke des Wortes 'Go' aus Abb. 3.4 mit approximierten Kontrollpunkten und rekonstruierten B-Splines

Nach dem gleichen Prinzip wie bei den Bézierkurven kann daraus ein Gleichungssystem aufgestellt werden. Wiederum durch eine Singulärwertzerlegung ergeben sich die Kontrollpunkte aus der Pseudonormallösung.

Neben den gezeigten Verfahren bietet sich das Prinzip der Singulärwertzerlegung darüber hinaus für diverse andere Approximationsansätze, wie z. B. die Approximation durch Polynome an [Hü99].

Abb. 3.5 erlaubt einen ersten Vergleich zwischen Original mit approximierten Kontrollpunkten und rekonstruierter Trajektorie. Bei den gezeigten Trajektorienstücken handelt es sich um die ersten drei Beispiele aus Abb. 3.4 mit identischen Größenverhältnissen. Auch mit der Verwendung von B-Splines scheint die vorgegebene Kurve gut rekonstruierbar zu sein. Verglichen mit den Bézierkurven fällt allerdings eine insgesamt größere Streuung der Kontrollpunkte auf. Zudem ist eine weitere Eigenschaft von B-Splines erkennbar: die Kontrollpunkte liegen nicht zwangsweise (wie bei Bézierkurven) auf den Anfangs- bzw. Endpunkten der Trajektorie.

### 3.1.5 Hauptachsentransformation

In den vorangegangenen Abschnitten wurden die entwickelten Lösungsansätze zur Extraktion von Trajektorienmerkmalen aus verschiedenen Blickwinkeln erarbeitet. Neben parametrischen Beschreibungsformen wurden spektrale (DCT), sowie geometrische Eigenschaften (Bézierkurven und B-Splines) einer Trajektorie einbezogen.

Abb. 3.6 zeigt die Superposition aller relativen Fenster aus einem schreiberabhängigen Trainings-Set. Der Grad der Schwärzung ist proportional zu der beobachteten Häufigkeit der Abtastpunkte. Aus Abb. 3.6 wird unmittelbar ersichtlich, dass bestimmte Vorzugsrichtungen existieren. Neben den parametrischen, spektralen und geometrischen Verfahren ist es daher naheliegend, gewisse statistische Eigenschaften der Trajektorien auszunutzen, um zu einer kompakten Repräsentation einer Trajektorie zu gelangen.

Dazu könnte die Hauptachsentransformation dienlich sein. Für diese Transformation wird zunächst aus den  $K$  relativen Fenstern  $\mathcal{F}_k$  ein Mittelwertvektor berechnet:

$$\vec{f} = \frac{1}{K} \sum_{k=1}^K \vec{f}(k) \quad (3.29)$$

Der Mustervektor  $\vec{f}(k)$  wird mit den  $x$ - und  $y$ -Komponenten der Abtastpunkte aus dem Fenster  $\mathcal{F}_k$  besetzt. Dies geschieht unter Ausschluß des Abtastpunktes  $\mathbf{f}_k$ , da dieser ohnehin mit dem Koordinatenursprung identisch ist. Es werden also  $2(F - 1)$ -dimensionale Mustervektoren gebildet. Aus den Mustervektoren  $\vec{f}(k)$  und dem Mittelwertvektor  $\vec{f}$  wird eine  $M \times M$  Kovarianzmatrix  $C$  berechnet, mit  $M = 2(F - 1)$ . Eine Verschiebung der Mustervektoren um  $\vec{f}$  und eine anschließende Multiplikation mit einer dimensionsreduzierten, orthogonalen

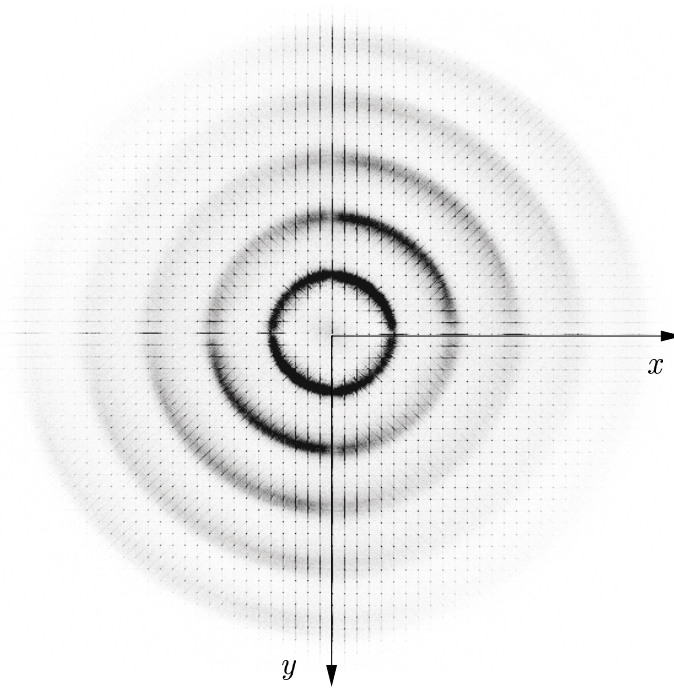


Abbildung 3.6: Superposition relativer Fenster (Trainings-Set)

$(M \times N_m)$ -Matrix  $U$ , mit  $N_m < M$ , bestehend aus den Eigenvektoren von  $C$  liefert schließlich einen Merkmalsvektor der Dimension  $N_m$ .

$$\vec{m}(k) = U(\vec{f}(k) - \vec{f}) \quad (3.30)$$

## 3.2 Bitmap-Merkmale

Die bisher präsentierten Verfahren waren konzeptionell darauf ausgerichtet eine Trajektorie lokal, oder aber innerhalb eines gewissen zeitlichen Abschnittes zu beschreiben. Um geometrische Abhängigkeiten zu erfassen, die nicht unbedingt in einem zeitlichen Zusammenhang stehen, ist es angebracht die zeitliche Folge  $V = \{(\vec{v}, \tilde{p})_0, (\vec{v}, \tilde{p})_1, \dots, (\vec{v}, \tilde{p})_K\}$  einer online abgetasteten Schriftprobe, bestehend aus  $K$  Abtastvektoren zunächst in eine statische Bitmap  $g(x, y) \in [0, 1]$  zu überführen [Man94].

$$\{(\vec{v}, \tilde{p})_0, (\vec{v}, \tilde{p})_1, \dots, (\vec{v}, \tilde{p})_K\} \xrightarrow{BM} g(x, y). \quad (3.31)$$

In dieser binären Bitmap werden die Anfangs- und Endpunkte der Abtastvektoren, abhängig vom Stiftdruck  $\tilde{p}$ , in räumlich diskretisierter Form gesetzt. Durch eine diskretisierte Interpolation werden zudem die Pixel gesetzt, die von den Abtastvektoren mit positivem Stiftdruck überdeckt werden.

Aus der so erzeugten statischen Bitmap wird zur Erzeugung einer Merkmalssequenz die

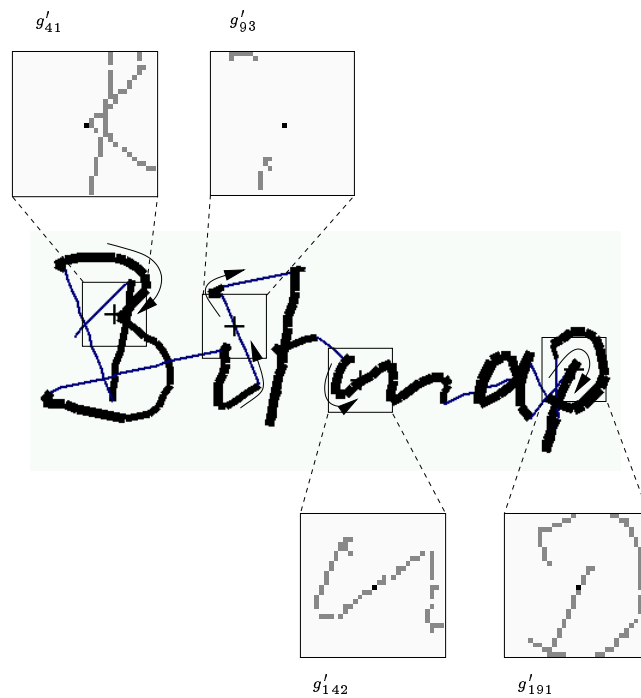


Abbildung 3.7: Gleitende Bitmap

Folge von Abtastvektoren  $V$  durchschritten. Wie in Abb. 3.7 gezeigt, wird auch die Spur der Segmente mit negativem Stiftdruck verfolgt. Zu jedem diskretisierten Punktepaar  $(\bar{v}_{0,x}(k), \bar{v}_{0,y}(k))$  wird ein um  $(\bar{v}_{0,x}(k), \bar{v}_{0,y}(k))$  symmetrischer, quadratischer Ausschnitt  $F_{BM,k}$  gebildet.  $g'_k(x, y)$  mit  $x, y \in F_{BM,k}$  stellt die binären Pixelwerte des betrachteten Fensters dar. Das schwarze Pixel in den gezeigten Ausschnitten dient lediglich dazu den Mittelpunkt  $(\bar{v}_{0,x}(k), \bar{v}_{0,y}(k))$  zu markieren. Die grauen Pixel entsprechen den gesetzten Bits. Als sinnvolle Abmessung für  $F_{BM,k}$  hat sich ein Wert von  $30 \times 30$  Pixel herausgestellt. Abb. 3.7 illustriert diese Vorgehensweise und gibt eine Vorstellung über die Größenverhältnisse der ausgeschnittenen Fenster  $F_{BM,k}$ . Die mit der Abtastfolge  $V = \{(\vec{v}, \vec{p})_0, (\vec{v}, \vec{p})_1, \dots, (\vec{v}, \vec{p})_K\}$  assoziierte Bitmap-Folge  $\{g'_0, g'_1, \dots, g'_K\}$  bildet die Grundlage für die im Folgenden vorgestellten Verfahren zur Extraktion von Bitmap-Merkmalen.

### 3.2.1 Räumliche Unterabtastung der Bitmap

Die direkte Verwendung einer Bitmap  $g'$  als Merkmalsvektor stellt offensichtlich keine vernünftige Lösung dar. Zum einen verhindert dies die Dimension eines solchen Vektors ( $30 \times 30$ ). Zum anderen würden geringe Verschiebungen einer durch ein Fenster verlaufenden Trajektorie um nur ein Pixel zu einem völlig unterschiedlichen Merkmalsvektor führen. Dem kann jedoch damit begegnet werden, indem der Ausschnitt aus  $30 \times 30$  Pixel einer Unterabtastung unterzogen wird. Dazu wird ein Ausschnitt  $F_{BM,k}$  zunächst in  $3 \times 3$  gleich große Blöcke  $\{F_{BM,k,1}, F_{BM,k,2}, \dots, F_{BM,k,9}\}$ , jeweils bestehend aus  $10 \times 10$  Pixel, unter-

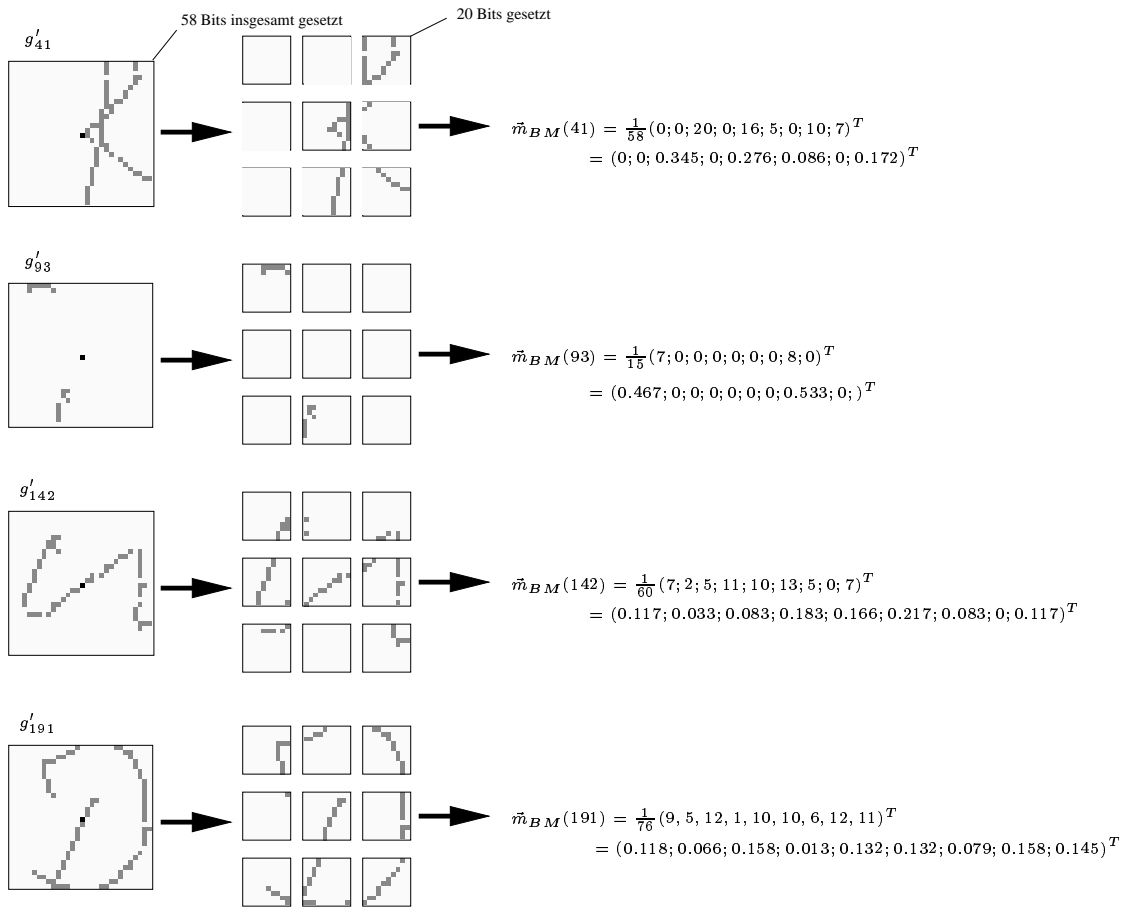


Abbildung 3.8: Unterabtastung der Bitmap

teilt. Die angegebenen Werte sind wiederum aus entsprechenden Experimenten als optimal identifiziert worden. Für jede Bitmap  $g'_{k,i}$  des Blocks  $F_{BM,k,i}$  wird des weiteren ein mittlerer Grauwert  $m_{BM,i}$  bestimmt. Dazu wird die Anzahl der in  $g'_{k,i}$  gesetzten Pixel auf die Gesamtanzahl der in  $g'_k$  enthaltenen und gesetzten Pixel normiert:

$$m_{BM,i}(k) = \frac{\sum_{x \in F_{BM,k,i}} \sum_{y \in F_{BM,k,i}} g'_{k,i}(x, y)}{\sum_{x \in F_{BM,k}} \sum_{y \in F_{BM,k}} g'_k(x, y)} \quad (3.32)$$

$m_{BM,i}(k)$  stellt sodann die  $i$ -te von 9 Komponenten des Merkmalsvektors  $\vec{m}_{BM}(k)$  dar.

### 3.2.2 Diskrete Cosinus Transformation (DCT) der Bitmap

Wie bereits erwähnt, weist die DCT einige positive Eigenschaften auf, die bereits zu erfolgreichen Einsätzen dieser Transformation in verschiedenen Bereichen der Bildverarbeitung führten. Neben der Bildkodierung wurde die Eignung der DCT bereits für verschiedene Problemstellungen im Bereich der Mustererkennung bewiesen. Beispielsweise kann ein auf DCT-Koeffizienten basiertes Erkennungssystem für Bildobjekte direkt auf komprimierten JPEG- oder MPEG-Daten arbeiten [Eic00].



Ausgehend von der Bitmap  $g_k(l, n)$  eines  $30 \times 30$  Pixel großen Ausschnitts  $F_{BM,k}$ , ergeben sich die DCT-Koeffizienten

$$m_{DCT,k}(i, j) = \frac{1}{30^2} \sum_{l=0}^{29} \sum_{n=0}^{29} g_k(l, n) \cos \frac{(2l+1)\pi i}{2(30+1)} \cos \frac{2(n+1)\pi j}{2(30+1)}. \quad (3.33)$$

Der Merkmalsvektor  $\vec{m}_{DCT}(k)$  wird mit den so berechneten DCT-Koeffizienten  $m_{DCT,k}(i, j)$  besetzt. Mit  $i = 0, 1, 2$  und  $j = 0, 1, 2$  ergibt sich folglich ein neundimensionaler Merkmalsvektor  $\vec{m}_{DCT}(k)$ .

### 3.2.3 Walsh Transformation der Bitmap

Neben der DFT und der DCT wird in der Bildverarbeitung häufig die Walsh-Transformation eingesetzt. Die Transformation in den Bildbereich geschieht, ähnlich wie bei der DCT durch eine mit einem Funktionensystem gewichtete Summe der Originalfunktion. Bei dem Funktionensystem handelt es sich um ein binäres (-1 und 1), vollständiges orthogonales Funktionensystem [Gon92]. Anhand der ersten neun Walsh-Funktionen wird wiederum ein neundimensionaler Merkmalsvektor für jeden Ausschnitt  $F_{BM,k}$  berechnet.

## 3.3 Ergebnisse

Um möglichst repräsentative Ergebnisse für eine größere Anzahl von Schreibern zu erzielen, fällt die Wahl bei der Evaluierung der Merkmale wiederum auf das schreiberunabhängige System.

Um die Vergleichbarkeit verschiedener Extraktionsverfahren zu wahren, wird die verwendete Parameteranzahl nach Möglichkeit konstant gehalten. Die Codebuchgröße des Vektorquantisierers von 100 wurde durchgängig für alle Extraktionsverfahren beibehalten.

Der gewählte Ansatz für das Erkennungssystem wird in den nachfolgenden Kapiteln beschrieben. Vorwegnehmend sei hier bereits erwähnt, dass die Erkennungsergebnisse mit diskreten HMM-Systemen ermittelt wurden. Die Erkennungsraten für den Einzelworttest werden anhand der Levenshtein-Distanz

$$\text{Korrektheit} = \frac{\text{korrekt erkannte Worte} - \text{gelöschte Worte}}{\text{Gesamtzahl der Worte}} \quad (3.34)$$

berechnet, mit einer Gesamtanzahl von 4134 Wörtern. Die Ergebnisse wurden mit einem Lexikon bestehend aus 2000 Wörtern erzeugt.

### 3.3.1 Trajektorienmerkmale

Aus vorab durchgeführten Tests kann - weitgehend unabhängig von dem Extraktionsverfahren - als günstige Fenstergröße  $F = 11$  angegeben werden. Die Ergebnisse der

einzelnen Verfahren werden im folgenden in der Reihenfolge dargestellt, wie sie in den vorangegangenen Abschnitten behandelt wurden. Zur übersichtlichen Darstellung der Ergebnisse wird auf die Darstellung aller 20 Einzelergebnisse zu jedem Testschreiber verzichtet. Die Ergebnistabellen enthalten neben der Gesamterkennungsrate (letzte Zeile) die Erkennungsraten der Top-3-Schreiber (1. Zeile), der Flop-3-Schreiber (3. Zeile), sowie der drei Schreiber, deren Ergebnisse am nächsten an der Gesamterkennungsrate liegen (2. Zeile).

Der Blick auf die erzielten Gesamterkennungsraten zeigt zunächst bei den verschiedenen

ank: 86.0 %	bar: 92.0 %	uts: 87.4 %
abr: 70.1 %	dpo: 81.7 %	vdm: 72.7 %
all: 64.1 %	frs: 54.6 %	jmr: 57.6 %
$\phi = 76.7 \%$		

Tabelle 3.1: Erkennungsergebnis - *Sinus- und Cosinuswinkel* mit  $F = 11$  und  $N_m = 22$

ank: 88.7 %	bar: 93.5 %	uts: 89.8 %
abr: 74.1 %	dpo: 80.7 %	vdm: 76.5 %
all: 61.3 %	frs: 60.8 %	jmr: 61.4 %
$\phi = 77.3 \%$		

Tabelle 3.2: Erkennungsergebnis - *DCT* mit  $F = 11$  und  $N_m = 6$

ank: 89.3 %	bar: 91.0 %	uts: 88.8 %
abr: 77.0 %	dpo: 79.7 %	vdm: 75.4 %
all: 59.0 %	frs: 62.5 %	jmr: 63.6 %
$\phi = 77.4 \%$		

Tabelle 3.3: Erkennungsergebnis - *Bézierkurve* mit  $F = 11$  und  $N_m = 8$

Extraktionsverfahren eine relativ geringe Streuung zwischen 77,4 % bei den Bézierkurven (Tab. 3.3) und 74,7 % bei der Hauptachsentransformation (Tab. 3.5). Dies mag zunächst den Schluß nahe legen, dass die Methode mit der die Merkmale aus einer Trajektorie extrahiert werden, nicht sehr ausschlaggebend auf das Ergebnis wirkt. Die hier vorgestellten Extraktionsverfahren stellen allerdings bereits eine Vorauswahl dar. Nicht weiter berücksichtigt wurden hier z. B. die diskrete Fourier Transformation, die mit einer Erkennungsrate von 50,1 % ein eher ernüchterndes Ergebnis lieferte, ähnlich wie die Trajektorienapproximation durch Polynome, mit der maximal 65,6 % Korrektheit erreicht wurden.

Auffällig ist hingegen, dass zum einen die Bézier-Kurven unter den Trajektorienmerkmalen das beste Ergebnis liefert, während die ähnlich konzipierten B-Splines das zweitschlechteste

ank: 84.4 %	bar: 89.1 %	uts: 85.9 %
abr: 74.1 %	dpo: 72.8 %	vdm: 74.9 %
all: 63.1 %	frs: 57.4 %	jmr: 58.2 %
$\phi = 75.5 \%$		

Tabelle 3.4: Erkennungsergebnis - *B-Spline* mit  $F = 11$  und  $N_m = 8$ 

ank: 84.4 %	bar: 90.1 %	uts: 89.3 %
abr: 69.5 %	dpo: 73.3 %	vdm: 74.3 %
all: 57.4 %	frs: 61.4 %	jmr: 59.8 %
$\phi = 74.7 \%$		

Tabelle 3.5: Erkennungsergebnis - *Hauptachsentransformation* mit  $F = 16$  und  $N_m = 8$ 

Ergebnis zeigen. Zu begründen ist diese Abweichung mit der größeren Varianz, die bei den B-Splines zu beobachten ist. Während optisch ähnliche Vektorsequenzen bei den Bézierkurven zu ähnlich gelagerten Kontrollpunkten führen, bedeuten geringe Abweichungen in den Vektordaten bereits größere Lageveränderungen für die Kontrollpunkte der B-Splines.

### 3.3.2 Bitmap-Merkmale

Bei den Bitmap-Merkmalen stellt sich im Vergleich zu den Trajektorien-Merkmalen insgesamt eine etwas höhere Streuung unter den Ergebnissen ein. Während mit der einfachen Unterabtastung der gleitenden Bitmap eine maximale Erkennungsrate von 79,7 % erzielt wird, dicht gefolgt von der Walsh-Transformation der Bitmap mit 79,3 %, zeigt sich das Ergebnis der zweidimensionalen DCT der Bitmap deutlich abgeschlagen mit 75,6 %. Dieses Ergebnis überrascht, da mit der DCT und der Walsh-Transformation vollständig orthogonale Funktionensysteme für die Transformation verwendet werden. Dies wiederum sollte zu einer geringen Korrelation der einzelnen Merkmale untereinander führen, und damit zu einer verbesserten Klassifizierbarkeit.

ank: 89.8 %	bar: 94.5 %	uts: 85.9 %
abr: 75.9 %	dpo: 90.6 %	vdm: 81.3 %
all: 72.8 %	frs: 59.1 %	jmr: 56.5 %
$\phi = 79.7 \%$		

Tabelle 3.6: Erkennungsergebnis - *spatial unterabgetastete Bitmap* mit  $F_{BM} = 30 \times 30$  und  $N_m = 9$

ank: 86.6 %	bar: 90.1 %	uts: 83.0 %
abr: 73.6 %	dpo: 86.6 %	vdm: 75.4 %
all: 63.6 %	frs: 52.3 %	jmr: 54.4 %
$\emptyset = 75.6 \% [N=4134]$		

Tabelle 3.7: Erkennungsergebnis - *DCT der Bitmap* mit  $F_{BM} = 30$  und  $N_m = 9$ 

ank: 88.7 %	bar: 92.5 %	uts: 89.3 %
abr: 82.2 %	dpo: 85.6 %	vdm: 80.8 %
all: 74.9 %	frs: 61.4 %	jmr: 59.2 %
$\emptyset = 79.3 \% [N=4134]$		

Tabelle 3.8: Erkennungsergebnis - *Walsh-Transformation* mit  $F_{BM} = 30$  und  $N_m = 9$ 

### 3.3.3 Merkmalskombination

Nach der separaten Untersuchung der verschiedenen Merkmale scheint eine Merkmalskombination zur Steigerung der Erkennungsgenauigkeit opportun.

Da die Merkmalsextraktion eine der unteren Stufen in der gesamten Verarbeitungskette darstellt, ist es sinnvoll einen Frühindikator zur Bewertung verschiedener Verfahren heranzuziehen. Mit solchen Frühindikatoren ließe sich für bestimmte Merkmale eine Vorauswahl treffen. Von der Simulation, einschließlich Modellierung und Tests, könnten so weniger geeignete Merkmale bereits im Vorfeld ausscheiden.

Der Vorteil eines solchen Vorgehens wird insbesondere bei der Frage nach der günstigsten Merkmalskombination ersichtlich. Während die hier betrachteten acht Einzelmerkmale noch konsequent getestet werden konnten, scheint dies für  $8 \cdot (8 - 1)/2$  Merkmalskombinationen nicht mehr sinnvoll. Allein ein Testdurchlauf für ein Merkmal kann bei vorhandenen Modellen bereits einige Tage an CPU-Zeit in Anspruch nehmen. Unter diesen Randbedingungen scheidet die vollständige Kombination verschiedener Merkmalstypen allein schon aus Rechenzeitgründen aus.

In der Verarbeitungskette Vorverarbeitung, Merkmalsextraktion, Merkmalsquantisierung, Modellierung und Test sind die exakten Erkennungsergebnisse natürlich erst nach dem Test verfügbar. Nach der Modellierung ist bereits die mittlere Auftrittswahrscheinlichkeit der Trainingsdaten unter der Voraussetzung gegebener Modelle bekannt. Nach der Merkmalsquantisierung wiederum, kann die Transinformation der quantisierten Trainingsdaten bestimmt werden. Vereinfacht ausgedrückt gibt die Transinformation das Maß an Information wieder, welches sich effektiv über die Verarbeitungskette an den Erkenner übermitteln läßt. Wie dem nachfolgenden Kapitel weiterhin entnommen werden kann, ist die Berechnung der Transinformation auch für die Multi-Codebuch-Technik möglich. Dabei wird neben der Einzeltransinformation der jeweiligen Merkmalsströme auch deren Korrelation, bzw. die

Trajektorienmerkmale	$I(S, Y)$	$\log P(Y S)$
<i>Sinus- und Cosinuswinkel</i>	0.5264 bit	-2.5677
<i>DCT</i>	1.0142 bit	-2.663
<i>Bézierkurve</i>	1.0851 bit	-2.571
<i>B-Spline-Kurve</i>	0.9364 bit	-2.761
<i>Hauptachsentransformation</i> ( $F = 16$ )	0.9225 bit	-2.702
<i>DFT</i>	0.7240 bit	-3.276
<i>Polynom dritten Grades</i>	0.5527 bit	-3.264
Bitmap-Merkmale		
<i>Unterabtastung der Bitmap</i>	0.7391 bit	-3.241
<i>DCT der Bitmap</i>	0.6901 bit	-3.255
<i>Walsh-Transformation</i>	0.6959 bit	-3.251

Tabelle 3.9: MMI und log-Likelihood von Einzelmerkmalen

Redundanz der Merkmale einbezogen. Die Verwendung der gemeinsamen Transinformation für multiple Merkmale stellt eine frühestmögliche Abschätzung für die Eignung der Merkmalskombinationen dar, da dieses Bewertungsmaß unmittelbar nach der Vektorquantisierung bestimmt werden kann.

Tab. 3.9 zeigt die Transinformation  $I(S, Y)$  und die logarithmierte Auftrittswahrscheinlichkeit der Trainingsdaten. Überwiegend lassen sich innerhalb einer Merkmalsgruppe gute Übereinstimmungen zwischen Transinformation, log-Likelihood und letztendlicher Er-

	<i>DCT auf Bitmap</i>	<i>Walsh- Transformation</i>	<i>Grauwertbestimmung von Bitmapteilen</i>
Verfahren mit $F = 11$	$I(S, Y)=0.6959$	$I(S, Y)=0.6901$	$I(S, Y)=0.7391$
<i>Sinus- und Cosinuswinkel</i> $I(S, Y)=1.0871$	1.3747	1.3698	1.3668
<i>DCT</i> $I(S, Y)=1.0408$	1.3216	1.3135	1.3219
<i>Bézierkurve</i> $I(S, Y)=1.0851$	1.3640	1.3605	1.3549
<i>B-Spline-Kurve</i> $I(S, Y)=0.9364$	1.2690	1.2752	1.1.2727
<i>Hauptachsentransformation</i> ( $F = 16$ ), $I(S, Y)=0.9225$	1.2348	1.2267	1.2290

Tabelle 3.10: Gesamt-Transinformation  $I(S, Y^1, Y^2)$  in bit für Kombinationen aus Bitmap- und Trajektorienmerkmalen

	<i>DCT der Bitmap</i> $I(S, Y) = 0.6959$	<i>unterabgetastete BM</i> $I(S, Y) = 0.7391$
DCT der BM $I(S, Y) = 0.6959$	-	0.8085
Walsh-Transf. $I(S, Y) = 0.7054$	0.7094	0.7783

Tabelle 3.11: Gesamt-Transinformation  $I(S, Y^1, Y^2)$  in bit für zwei Bitmap-Merkmale

	<i>Bézierkurve</i> $I(S, Y) = 1.0851$	<i>DCT</i> $I(S, Y) = 1.0408$
Bézierkurve $I(S, Y) = 1.0851$	-	0.9705
KLT $I(S, Y) = 0.9225$	0.9776 bit	0.9024

Tabelle 3.12: Gesamt-Transinformation  $I(S, Y^1, Y^2)$  in bit für zwei Trajektorienmerkmale ( $F=11$ )

kennungsrate feststellen. Zum Vergleich sind in Tab. 3.9 weiterhin die Werte für die DFT und die Polynomapproximation aufgeführt. Aufgrund der grundsätzlich unterschiedlichen Extraktionsprinzipien der Trajektorien- und Bitmap-Merkmale ist eine maximale Komplementärwirkung bei der Kombination dieser beiden Merkmalstypen zu erwarten. Tab. 3.10 zeigt dazu die gemeinsame Transinformation der Zweier-Kombinationen von Trajektorien- und Bitmap-Merkmalen.

Zum Vergleich ist in Tab. 3.11 die Kombination verschiedener Bitmap-Merkmale zusammengefasst. Der Gewinn liegt hier, wie auch bei der Kombination von Trajektorienmerkmalen in Tab. 3.12 deutlich niedriger. Am Beispiel der Kombination von DCT-Trajektorienmerkmalen ist regelmäßig sogar ein nicht unerheblicher Transinformationsverlust festzustellen.

Die Kombination von Bitmap- und Trajektorienmerkmalen ist demnach die Wahl der Mittel. Es zeigt sich weiterhin, dass eine Dreier-Kombination wegen der offensichtlichen Red-

	<i>DCT</i> <i>der Bitmap</i>	<i>Walsh-</i> <i>Transformation</i>	<i>Grauwertbestimmung</i> <i>von Bitmapteilen (X)</i>
<i>Sinus- und Cosinuswinkel</i>	85.6 %	83.5 %	85.5 %
<i>DCT</i>	85.1 %	85.6 %	85.0 %
<i>Bézierkurve</i>	<b>86.2 %</b>	86.0 %	85.8 %

Tabelle 3.13: Erkennungsraten kombinierter Merkmale

undanz innerhalb einer Merkmalsgruppe keine weiteren Vorteile bringt. Speziell läßt sich aus Tab. 3.10 ableiten, dass sich die Kettenkodierung (Sinus- und Cosinuswinkel), sowie die DCT der Trajektorie und die Bézierkurve gut mit den Bitmap-Merkmalen kombinieren läßt. Die Ergebnisse dieser Kombination sind abschließend in Tab. 3.13 dargestellt.

Hier zeigt sich, dass die Kombination aus DCT der Bitmap mit den Bézier-Merkmalen klar das beste Ergebnis liefert, was wiederum die Korrelation zur gemeinsamen Transinformation (Tab. 3.10) bestätigt.

### 3.4 Kapitelzusammenfassung

In diesem Kapitel wurden verschiedene Verfahren zur Merkmalsextraktion in Online-Handschrifterkennungssystemen präsentiert. Die Verfahren stützen sich auf zwei unterschiedliche Prinzipien der Merkmalsentnahme. Die erste Gruppe von Verfahren basiert auf der Merkmalsextraktion aus einem Fenster der Stiftrajektorie. Die zweite Gruppe nutzt die Information aus einer gleitenden Bitmap, wodurch sich zusätzlich die Möglichkeit eröffnet, die geometrische Umgebung zu erfassen. Die Erfassung der Umgebungsinformation ist dabei nicht auf ein zeitlich begrenztes Fenster beschränkt.

Speziell die Extraktion der Trajektorienmerkmale wurde unter verschiedenen Blickwinkeln betrieben. Neben einer parametrischen Kurvenbeschreibung wurde die Nutzung spektraler, geometrischer und statistischer Eigenschaften für eine kompakte Beschreibungsform untersucht.

Zur Auswahl und Kombination der Merkmale wurden neben der exakten Erkennungsrate weitere Bewertungsmaße - insbesondere die Transinformation - herangezogen, mit der eine sinnvolle Vorauswahl der Merkmale getroffen werden konnte.

Schließlich zeigte sich eine Kombination aus diskreter Cosinus Transformation der Bitmap mit den Bézierkurven als ideale Kombination mit maximaler Erkennungsrate, wobei die Kontrollpunkte der approximierenden Bézierkurve als Merkmale Verwendung fanden. Diese Merkmalskombination sollte demnach fortan eingesetzt werden.

Da im Rahmen dieser Arbeit einige Entwicklungen parallel vorangetrieben wurden, gilt insbesondere für die nachfolgend beschriebenen höheren Systemebenen, dass die Kettenkodierung zusammen mit der unterabgetasteten Bitmap weiterhin als Referenzverfahren eingesetzt wird. Dies stellt die Vergleichbarkeit zu früheren Ergebnissen sicher.