# Ein effizienter 2D Warping Algorithmus

Christian Gollan (Betreuer: Daniel Keysers)

28. Februar 2001

## Inhaltsverzeichnis

**Zusammenfassung**

*In dieser Seminararbeit wird ein neuer Algorithmus zur Berechnung der Distanz zwischen zwei Bildern von Seiichi Uchida und Hiroaji Sakoe vorgestellt. Dieses Verfahren basiert auf zweidimensionalen Warping (2DW) unter Beachtung von Monotonie- und Stetigkeitseinschränkungen und kann zur Bewertung von Bildvergleichen angewandt werden. Es wird gezeigt, dass die Laufzeit des vorgestellten Algorithmus selbst unter Verwendung von dynamischer Programmierung (DP) exponentiell ist. Deshalb werden approximierende Versionen des Algorithmus beschrieben, die geringere Laufzeiten haben.*

**Stichwörter:** Bildvergleich, Bildverformung, zweidimensionales Warping, dynamische Programmierung, approximierende Algorithmen

# 1 Einleitung

Bei der Bildklassifikation (z.B. Schrifterkennung oder Gesichtserkennung) sowie beim Image-Retrieval (Bildsuche) benötigt man unter anderem einen Algorithmus, der zwei Bilder nach individuell festlegbaren Kriterien miteinander vergleicht und diesen Vergleich bewertet. Hat man z.B. ein Musterbild und sucht in einer Bilddatenbank nach ähnlichen Bildern, dann kann man einen solchen Algorithmus zum Vergleich des Musterbildes mit jeweils jedem Bild der Bilddatenbank anwenden und anhand der Bewertungen durch den Algorithmus die gesuchten Bilder bestimmen. Bei der Klassifikation eines unbekannten Bildes hat man dagegen mehrere Referenzbilder, jedes Referenzbild gehört zu einer anderen Klasse, und durch die Vergleiche des unbekannten Bildes mit allen Referenzbildern bestimmt man anhand der Ergebnisse die vermutliche Klasse des beobachteten Bildes. Eine Klasse von Algorithmen, mit denen sich ein solcher Bildvergleich bewerten lässt, sind 2DW Algorithmen. Bei diesem Verfahren wird eine Kostenfunktion aufgestellt, die den Vergleich zwischen zwei Bildern bewertet. Bei dem Vergleich zweier Bilder **A** und **B** wird eine eingeschränkte Verformung (Warping) bei Bild **B** durchgeführt, so dass das Ergebnis der Kostenfunktion, angewandt auf das Bild **A** und das verformte Bild **B**, minimiert wird. Die Bildverformung (zweidimensionales Warping) kann man als Pixel auf Pixel abbildende Abbildung (Warpingabbildung) ausdrücken. Unerwünschte Warpingabbildungen, die hier unbrauchbare, verfälschende Verformungen sind, lassen sich durch Festlegung von Einschränkungen der Bildverformung verhindern. Die Aussagekraft der Ergebnisse des Algorithmus bei Bildvergleichen hängt von der Kostenfunktion und den Einschränkungen der Bildverformung ab.

Bisher angewandte Algorithmen, die fast fehlerfreie Ergebnisse bei dieser Methode des Bildvergleiches liefern, haben Laufzeiten, die exponentiell zur Bildgröße sind. Dadurch wird deren Gebrauch für Echtzeit (Realtime)-Bildanwendungen nur für Bilder mit geringer Pixelanzahl praktikabel. Aber auch dann, wenn die Ergebnisse nicht in Echtzeit benötigt werden, ist der Zeitaufwand beispielsweise für Röntgen- oder Satellitenbilder (z.B. $2000 \times 2000$ Pixel) aufgrund ihrer Bildgröße derzeit zu hoch, um diese Verfahren in der Pra-

xis einzusetzen. Deshalb werden effizientere Algorithmen gesucht, die auch bei großen Bildern in Ergebnis und Laufzeit zufriedenstellend sind. Dies war auch das Ziel der hier besprochenen Arbeiten.

Im folgenden werden die 2DW Algorithmen, die auf den Arbeiten [1, 2, 3, 4, 5] von Seiichi Uchida und Hiroaji Sakoe beruhen, vorgestellt. Dabei zeigt sich, dass die Laufzeit des „effizienten“ 2DW Algorithmus exponentiell zur Bildgröße ist. Deshalb wurden im weiterem Verlauf ihrer Aufsätze die approximierenden Versionen ihres Algorithmus mit den so verkürzten Laufzeiten beschrieben. Am Ende der vorliegenden Seminararbeit wird unter Hinzunahme von weiteren Algorithmen [6, 7, 8], die sich mit dem Problem der Bewertung eines Bildvergleiches beschäftigen, ein Vergleich und eine Beurteilung der Algorithmen durchgeführt.

# 2 Bildvergleich durch 2D Warping

Die Definition der Kostenfunktion und die Festlegung der Einschränkungen bei Warpingabbildungen bestimmen, wie zufriedenstellend die Ergebnisse des Algorithmus in der Praxis sein können. In den nächsten zwei Unterkapiteln werden die Kostenfunktion sowie die Monotonie- und Stetigkeitseinschränkungen erläutert.

## 2.1 Kostenfunktion

Gegeben sind zwei miteinander zu vergleichende Bilder $\mathbf{A} = \{\mathbf{a}(i,j)|i = 1,\ldots,I, j = 1,\ldots,J\}$ und $\mathbf{B} = \{\mathbf{b}(x,y)|x = 1,\ldots,X, y = 1,\ldots,Y\}$. Das Bild $\mathbf{B}$ wird durch die Warpingabbildung

$$f(i,j) = \begin{pmatrix} x(i,j) \\ y(i,j) \end{pmatrix} \tag{1}$$

so verformt (Bildverformung), dass folgende Kostenfunktion

$$c(\mathbf{A},\mathbf{B},f) = \sum_{i=1}^{I}\sum_{j=1}^{J} d(\mathbf{a}(i,j),\mathbf{b}(x(i,j),y(i,j))) \tag{2}$$

minimiert wird. Die Bewertungsfunktion $d(\mathbf{a},\mathbf{b})$ legt den Unterschied (Distanz) zwischen den Pixelwerten $\mathbf{a}$ und $\mathbf{b}$ fest. Die Distanz $D(\mathbf{A},\mathbf{B})$ ist definiert als Minimum von (2):

$$D(\mathbf{A},\mathbf{B}) := \min_{f} c(\mathbf{A},\mathbf{B},f) \tag{3}$$

Für die quadrierte Euklidische Distanz $d(\mathbf{a},\mathbf{b}) = ||\mathbf{a}-\mathbf{b}||^2$ ergibt sich beispielsweise die quadrierte Euklidische Distanz zwischen Bild $\mathbf{A}$ und verformtem Bild $\mathbf{B}$. Der Wert $D(\mathbf{A},\mathbf{B})$, die Distanz zwischen Bild $\mathbf{A}$ und Bild $\mathbf{B}$, dient zur Bewertung des Bildvergleiches. Der Name der Kostenfunktion wurde deshalb gewählt, da man sich vorstellen kann, dass die Bildverformung und die Anpassung von Pixelwerten (Pixelwertverformung) unerwünschte Kosten verursachen (Kostenminimierung). Die Kostenfunktion sollte deshalb so definiert sein, dass die

Kosten desto niedriger ausfallen, je ähnlicher sich die beiden verglichenen Bilder sind. Die hier verwendete Kostenfunktion ist ein Beispiel, bei dem keine Kosten durch Bildverformungen entstehen. Sie ist zur Ergebnisoptimierung modifizierbar. Ohne Einschränkungen bei der Bildverformung kann die Warpingabbildung so bestimmt werden, dass die Pixelwerte miteinander verglichen werden, die die geringsten Kosten ergeben. Es ist davon auszugehen, dass das verformte Bild **B**, dass durch eine solche Warpingabbildung entsteht, gar keine Ähnlichkeit mit dem ursprünglichen Bild **B** hat. Das durch diesen Weg erhaltene Ergebnis, die Distanz zwischen Bild **A** und dem verformten Bild **B**, ist dann nicht mehr aussagekräftig.

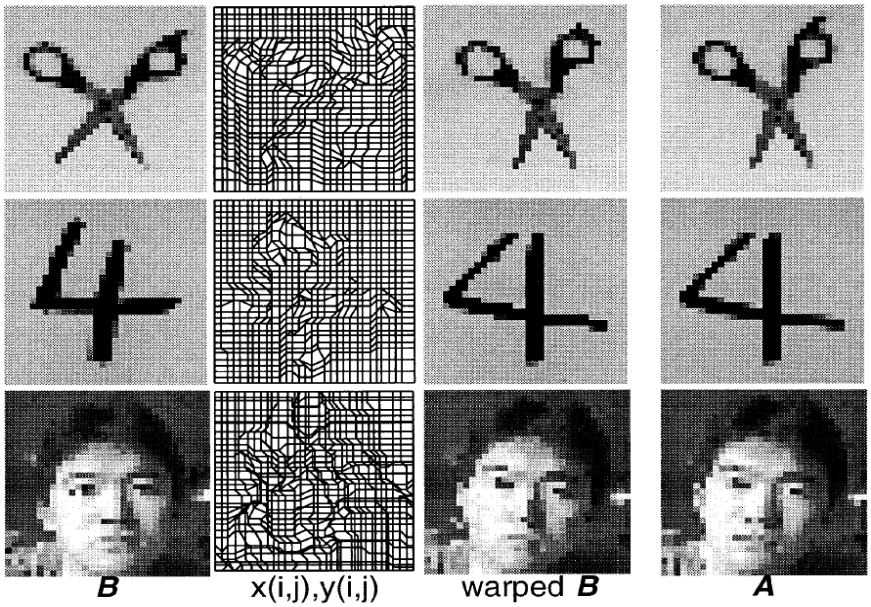


Abbildung 1: 2DW Beispiele (aus [4])

## 2.2 Monotonie- und Stetigkeitseinschränkungen

Der hier vorgestellte 2DW Algorithmus berücksichtigt bei der Berechnung der Distanz mittels der Kostenfunktion nur die verformten Bilder von Bild **B**, die die Monotonie- und Stetigkeitseinschränkungen

$$\begin{aligned} |x(i,j) - x(i-1,j) - 1| &\leq 1 \\ |x(i,j) - x(i,j-1)| &\leq 1 \\ |y(i,j) - y(i,j-1)| &\leq 1 \\ |y(i,j) - y(i-1,j) - 1| &\leq 1 \end{aligned} \tag{4}$$

bei der Warpingabbildung erfüllen, sowie folgende Randbedingungen

$$\begin{aligned} x(1,j) = y(i,1) &= 1 \\ x(I,j) &= X \\ y(i,J) &= Y. \end{aligned} \tag{5}$$

Diese Einschränkungen sollen zu starke Bildverformungen verhindern, damit das verformte Bild **B**, welches durch die Warpingabbildung beschrieben ist, seine Bedeutung durch die Verformung nicht verliert. Die Monotonie- und Stetigkeitseinschränkungen verhindern z.B. eine Spiegelung des Bildes (verhindert Kreuzen). So kann beispielsweise beim Vergleich der Bilder, die ein „b“ und „q“ darstellen, keine große Ähnlichkeit aufgrund einer kleinen Distanz bescheinigt werden. Die oben genannten Einschränkungen werden in diesem 2DW Algorithmus auch benutzt, um die Laufzeit unter Berücksichtigung aller erlaubten Warpingabbildungen zu verringern.

Beispiele für Resultate (die Bildverformung, aus der die geringste Distanz folgt) eines 2DW Algorithmus sind in Abbildung 1 dargestellt.

# 3 Der „effiziente“ 2D Warping Algorithmus

Für die Beschreibung des Algorithmus werden zunächst die Begriffe „Wake“ und „Warped Wake“ erklärt.

## 3.1 Wake und Warped Wake

Der Algorithmus tastet die Bilder, ausgehend vom Bild **A**, vertikal ab, d.h. die Reihenfolge der besuchten Pixel ist: $((1,1),(1,2),\ldots,(1,J),(2,1),\ldots,(I,J))$. Ein *Wake*[1] am Pixel $(i,j)$ (in der Abbildung 2: wake of stage $(i,j)$), sind die letzten $J$ Pixel des vertikalen Abtastungspfad, ein Beispiel ist: $[(i-1,j+1),\ldots,(i-1,J),(i,1),\ldots,(i,j-1),(i,j)]$. Eine mögliche Warpingabbildung von einem Wake am Pixel $(i,j)$ heißt *Warped Wake*, bezeichnet mit $\mathbf{xy}(i,j)$, für dieses Beispiel also $\mathbf{xy}(i,j) = [(x(i-1,j+1),y(i-1,j+1)),\ldots,(x(i-1,J),y(i-1,J)),(x(i,1),y(i,1)),\ldots,(x(i,j-1),y(i,j-1)),(x(i,j),y(i,j))]$. Die Menge aller möglichen Warped Wakes am Pixel $(i,j)$ wird mit $\mathbf{XY}(i,j)$ bezeichnet. Zu jedem Warped Wake $\mathbf{xy}(i,j) \in \mathbf{XY}(i,j)$ kann man eine Menge $\overline{\mathbf{XY}}(\mathbf{xy})$, die maximal neun Warped Wakes als Elemente hat, zuordnen. $\overline{\mathbf{XY}}(\mathbf{xy})$ ist eine echte Teilmenge der Menge $\mathbf{XY}(i,j-1)$ und enthält alle möglichen Vörgänger Warped Wakes. Die Abbildung 2 zeigt ein Warped Wake $\mathbf{xy}$ von einem Wake am Pixel $(i,j)$ und einen der zugehörigen Warped Wakes $\overline{\mathbf{xy}} \in \overline{\mathbf{XY}}(\mathbf{xy})$. Durch die Monotonie- und Stetigkeitseinschränkungen bei der Bildverformung sind die möglichen Abbildungen des Pixels $(i,j)$ des Bildes A auf einen Pixel des Bildes B, von den Koordinaten der abgebildeten Pixel $(i,j-1)$ und $(i-1,j)$ abhängig. Das heißt, unter Beachtung der Koordinaten der letzten $J$ abgebildeten Pixel

[1] engl., etwa „Kielwasser“

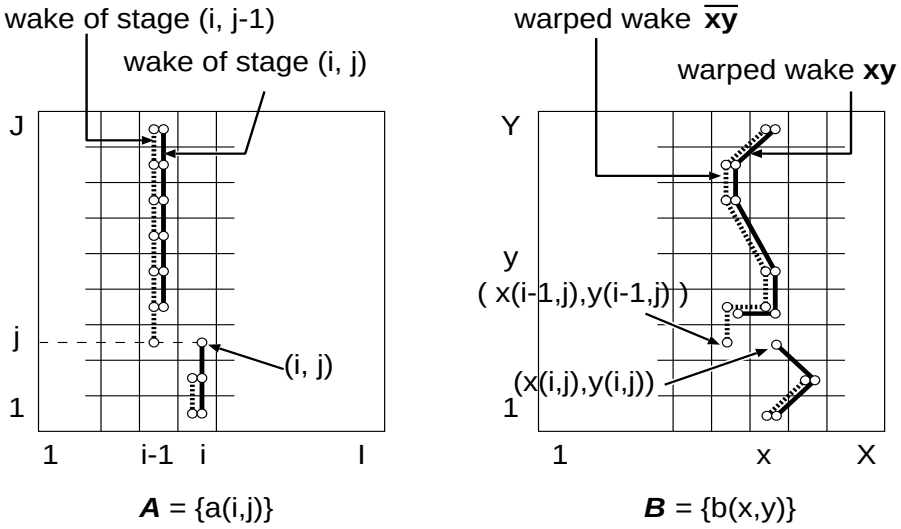


Abbildung 2: Wake und Warped Wake Darstellungen (aus [2])

(Warped Wake), kann man entscheiden, auf welche Koordinaten der nächste Pixel abgebildet werden kann. Ein Warped Wake am Pixel $(i, j-1)$ beinhaltet alle Informationen (Koordinaten der abbgebildeten Pixel, des letzten und des ersten Pixel eines Wakes), um die möglichen Abbildungen des im Abtastungspfad folgenden Pixels $(i, j)$ zu bestimmen. Abbildung 3 zeigt einen Ausschnitt eines Warped Wake am Pixel $(i, j-1)$ und einen zugehörigen Warped Wake. Die möglichen Koordinaten für die Abbildung des Pixel $(i, j)$ sind durch die Schnittmenge der Quadrate (a) und (b) im Bildteil B dargestellt. Umgekehrt betrachtet beinhaltet die Menge $\overline{\mathbf{XY}}(i, j)$ alle möglichen Warped Wakes $\overline{\mathbf{xy}}$, aus denen der Warped Wake $\mathbf{xy}(i, j)$ abstammen könnte.

## 3.2 Beschreibung des DP Algorithmus

Die möglichen Abbildungen eines Pixels $(i, j)$ hängen aufgrund der Monotonie- und Stetigkeitseinschränkungen von den Abbildungskoordinaten seiner Nachbarpixel $(i-1, j)$ und $(i, j-1)$ ab. Die Funktion $d_{xy}(i, j, \mathbf{xy}) = d(a(i, j), b(x(i, j), y(i, j)))$ beschreibt die Distanz zwischen dem Pixel $(i, j)$ aus Bild $\mathbf{A}$ und dem, durch den Warped Wake $\mathbf{xy}$ bestimmten, zugehörigen Pixel aus Bild $\mathbf{B}$. Die Funktion $g(i, j, \mathbf{xy})$ gibt die geringsten Kosten zwischen den Pixeln $(1, 1)$ bis $(i, j)$ und den abgebildeten Pixeln an, wobei die letzten $J$ Pixel nach dem Warped Wake $\mathbf{xy}$ abgebildet sind. Mit Hilfe der Funktion $d(\mathbf{a}, \mathbf{b}))$ werden zu Beginn des Verfahrens die Werte der Funktion $g(1, J, \mathbf{xy})$ für alle möglichen $\mathbf{xy} \in \mathbf{XY}(1, J)$ bestimmt. Man hat also für den Pixel $(1, J)$ eine Liste mit allen möglichen Warped Wakes und den Kosten der dadurch bestimmten Abbildun-

gen bis zum Pixel $(1, J)$. Für jeden Warped Wake der Liste des Pixels $(1, J)$ gibt es maximal neun mögliche Folge-Pixelabbildungen. Die dadurch entstehenden Warped Wakes und aufsummierten Kosten werden in die Liste des Pixels $(2, 1)$ eingetragen ($g(2, 1, \mathbf{xy})$). Dieser Schritt wird Pixel für Pixel entlang des vertikalen Abtastungspfad bis zum letzten Pixel $(I, J)$ ausgeführt. Dabei kommt es vor, dass in einer solchen Liste gleiche Warped Wakes eingetragen sind. Diese gleichen Warped Wakes stehen für unterschiedliche Abbildungen, wenn man ihren Weg durch die Vorgängerlisten verfolgt. Sie sind aber in einem Punkt zusammengekommen (die letzten $J$ Pixelabbildungen stimmen überein), so dass zur Bestimmung des nächsten Pixels nur der Warped Wake herangezogen wird, der die bisher geringsten Kosten verursacht. Abbildung 4 stellt den Wegfall von noch zu verfolgenden Warpingabbildungen grafisch dar. Die geringsten Kosten der Liste des Pixels $(I, J)$ entsprechen dann der Distanz zwischen Bild **A** und Bild **B**.

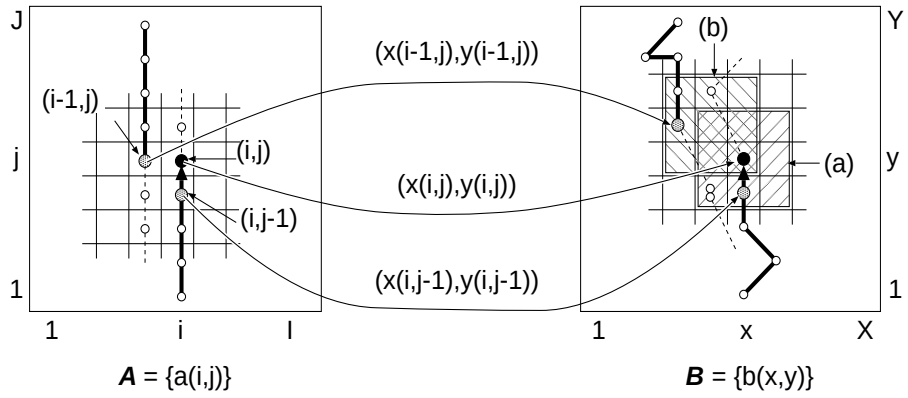


Abbildung 3: Wake und Warped Wake im Detail (aus [2])

### 3.2.1 Rekursionsgleichung

Die entsprechende DP Rekursionsgleichung

$$g(i, j, \mathbf{xy}) = d_{xy}(i, j, \mathbf{xy}) + \min_{\overline{\mathbf{xy}} \in \overline{\mathbf{XY}}(\mathbf{xy})} \begin{cases} g(i-1, J, \overline{\mathbf{xy}}) & \text{falls } j = 1 \\ g(i, j-1, \overline{\mathbf{xy}}) & \text{sonst} \end{cases} \tag{6}$$

ist definiert für jeden Pixel von $(2, 1)$ bis $(I, J)$, wobei zur Initialisierung

$$g(1, J, \mathbf{xy}) = \sum_{j=1}^{J} d(\mathbf{a}(1, j), \mathbf{b}(x(1, j), y(1, j))) \tag{7}$$

ist. Die Distanz $D(\mathbf{A}, \mathbf{B})$ ergibt sich so aus der Gleichung

$$D(\mathbf{A}, \mathbf{B}) = \min_{\mathbf{xy} \in \mathbf{XY}(I,J)} g(I, J, \mathbf{xy}). \tag{8}$$

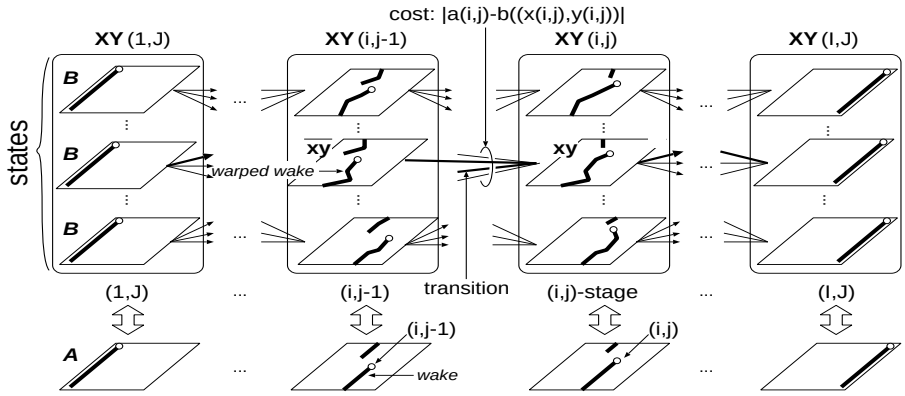


Abbildung 4: Darstellung des Prozesses der dynamischen Programmierung (aus [2])

### 3.2.2 Algorithmus

Im folgenden ist der Algorithmus in Pseudocode-Notation angegeben:

*/*Initialisierung*/*
**for all** $\mathbf{xy} \in \mathbf{XY}(1, J)$ **do**
  $g(1, J, \mathbf{xy}) := \sum_{k=1}^{J} d(\mathbf{a}(1, k), \mathbf{b}(x(1, k), y(1, k)))$

*/*Rekursion*/*
**for** $i := 2$ **to** $I$ **do begin**
  **for all** $\mathbf{xy} \in \mathbf{XY}(i, 1)$ **do**
    $g(i, 1, \mathbf{xy}) := d_{xy}(i, 1, \mathbf{xy}) + \min_{\overline{\mathbf{xy}} \in \overline{\mathbf{XY}}(\mathbf{xy})} g(i-1, J, \overline{\mathbf{xy}})$
  **for** $j := 2$ **to** $J$ **do**
    **for all** $\mathbf{xy} \in \mathbf{XY}(i, j)$ **do**
      $g(i, j, \mathbf{xy}) := d_{xy}(i, j, \mathbf{xy}) + \min_{\overline{\mathbf{xy}} \in \overline{\mathbf{XY}}(\mathbf{xy})} g(i, j-1, \overline{\mathbf{xy}})$
**end**

*/*Terminierung*/*
$D(\mathbf{A}, \mathbf{B}) := \min_{\mathbf{xy} \in \mathbf{XY}(I,J)} g(I, J, \mathbf{xy})$

### 3.2.3 Komplexität

Die Laufzeit des Algorithmus ist abhängig von der Pixelanzahl $O(I \cdot J)$, der Anzahl von möglichen Warped Wakes zu einem Pixel $O(X \cdot 9^J)$ und den 9 möglichen Abbildungen zu jedem Warped Wake in jedem Pixel, $O(I \cdot J) \cdot O(X \cdot 9^J) \cdot O(1) = O(I \cdot J \cdot X \cdot 9^J)$. Die Laufzeit ist also exponentiell zur Bildhöhe. So

muss man für eine große Bildhöhe $J$ einen approximierenden Algorithmus finden, der eine geringere Laufzeit hat. Wenn die Bildbreite für einen exponentiellen Algorithmus klein genug ist, kann man den Algorithmus auf den horizontalen Abtastungspfad umstellen und vertauscht so die laufzeitbestimmenden Größen Bildhöhe und Bildbreite.

Der Speicherbedarf beträgt aufgrund der Liste von Warped Wakes $O(X \cdot 9^J)$. Falls jedoch nicht nur die Distanz sondern auch die zugehörige Warpingabbildung bestimmt werden soll, ist Backtracking auf die Vorgängerlisten der Warped Wakes notwendig und der Speicherbedarf erhöht sich somit auf $O(I \cdot J \cdot X \cdot 9^J)$.

# 4 Approximierende Algorithmusversionen

In diesem Kapitel werden die drei approximierende 2DW Algorithmen, die von dem „effizienten“ 2DW Algorithmus abgeleitet sind, vorgestellt. Diese sind das Ergebnis der Suche von Seiichi Uchida und Hiroaji Sakoe nach effizienteren Verfahren.

## 4.1 Algorithmus mit Beam Search

Um die Laufzeit zu verringern, die abhängig von der Anzahl der Warped Wakes zu jedem Pixel ist, wird die Menge dieser Warped Wakes begrenzt. Die maximale Anzahl, auf die die Warped Wakes begrenzt werden, nennt man *Beam Size* $R$. Es werden bei jedem Pixel $(i, j)$ des Abtastungspfades die bis dahin $R$ besten Warpingabbildungen nach Kostenkriterien ($g(i, j, \mathbf{xy}) \; \forall \; \mathbf{xy} \in \mathbf{XY}(i, j)$) ausgewählt. Alle übrigen Warpingabbildungen werden vom Beam Search Algorithmus zur Distanzberechnung nicht mehr beachtet. Dies kann ein „Risiko“ darstellen, da die so ermittelte Distanz nicht mehr mit dem tatsächlichen Minimum (die durch die Kostenfunktion angestrebte Distanz) übereinstimmen muss. Ein Vorteil aber ist die durch die Beam Size $R$ variable Laufzeit $O(I \cdot J \cdot X \cdot R)$, die durch einen niedrigen Wert für $R$ deutlich verringert werden kann. Die Fehlerrate kann sich bei einer zu niedrig gewählten Beam Size $R$ erhöhen. Deshalb ist die Laufzeit nur scheinbar linear, da $R$ exponentiell wächst, um die Fehlerrate stabil zu halten.

## 4.2 Aufteilen des Bildes in Blöcke

Die Laufzeit des „effizienten“ 2DW Algorithmus ist vor allem von der Bildhöhe $J$ abhängig. Die Laufzeit kann verbessert werden, indem das Bild in mehrere horizontale Bildblöcke mit der Höhe $K$ aufgeteilt wird (Divide and Conquer). Dieser approximierende Algorithmus betrachtet jeden Bildblock nun als einzelnes Bild. Die Summe aller Distanzen dieser Bildblöcke ergibt die Distanz des gesamten Bildes. Alle Randbedingungen der Stoßstellen der zum Gesamtbild zusammengefügten Bildblöcke werden aufgehoben und aneinander angepasst (siehe Abbildung 5). Dies geschieht, um einen fortlaufenden, monotonen und stetigen Übergang zwischen den verformten Bildblöcken zu gewährleisten. Auch

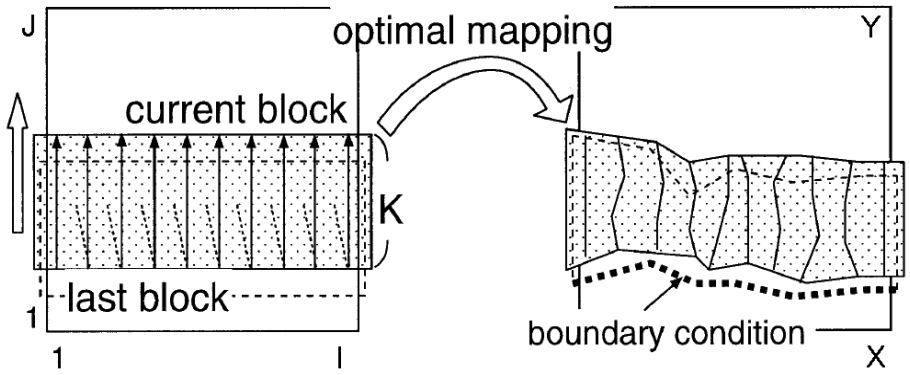


Abbildung 5: Beispiel für Randbedingungen eines Bildblocks (aus [4])

hier kann die so ermittelte Distanz von der Distanz, die durch die Kostenfunktion beabsichtigt war, abweichen, denn es wird lokal optimiert. Die Laufzeit des Algorithmus wird bei diesem Verfahren auf $O(I \cdot K \cdot X \cdot 9^K)$ verkürzt und ist so prinzipiell immer noch exponentiell.

### 4.3 Abschnittweise linearer 2D Warping Algorithmus

Diesen Algorithmus kann man vertikal linear oder horizontal linear ausführen, was im Ergebnis zu unterschiedlichen Distanzen führen kann. Es wird nun die horizontal lineare Version des Algorithmus vorgestellt.

In jeder Zeile $j$, $1 \leq j \leq J$ gibt es $K$, $1 \leq K \leq I$ sogenannte *Pivotpixel*. Wir definieren $i_{j,k}$ ist die i-Koordinate des $k$-ten $1 \leq k \leq K$ Pivotpixels aus der Zeile $j$, dann ist $(x_{j,k}, y_{j,k}) := (x(i_{j,k}, j), y(i_{j,k}, j))$ die Abbildung eines solchen Pivotpixels. Für die Pivotpixel gibt es folgende Einschränkungen

$$\begin{gathered} 1 = i_{j,1} < \ldots < i_{j,k-1} < i_{j,k} < \ldots < i_{j,K} = I \\ |i_{j,k} - i_{j-1,k}| \leq 1. \end{gathered} \tag{9}$$

Bei diesem Verfahren bestimmen nur die zugelassenen Pivotpixelabbildungen die gesamte Bildverformung. Die restlichen Pixelabbildungen werden durch die abgebildeten Pivotpixel definiert. Die Abbildung des Pixels $(i, j)$, $i \in (i_{j,k-1}, i_{j,k})$ wird durch die lineare Interpolation der zwei abgebildeten Pivotpixel $(x_{j,k-1}, y_{j,k-1})$ und $(x_{j,k}, y_{j,k})$ bestimmt

$$\begin{aligned} x(i,j) &= i' * x_{j,k} + (1 - i') * x_{j,k-1} \\ y(i,j) &= i' * y_{j,k} + (1 - i') * y_{j,k-1} \end{aligned} \tag{10}$$

mit $i' := (i - i_{j,k-1})/(i_{j,k} - i_{j,k-1})$. Abbildung 6 zeigt ein Beispiel für eine

abgebildete Bildzeile, die durch lineare Interpolation der abgebildeten Pivotpixel bestimmt ist.

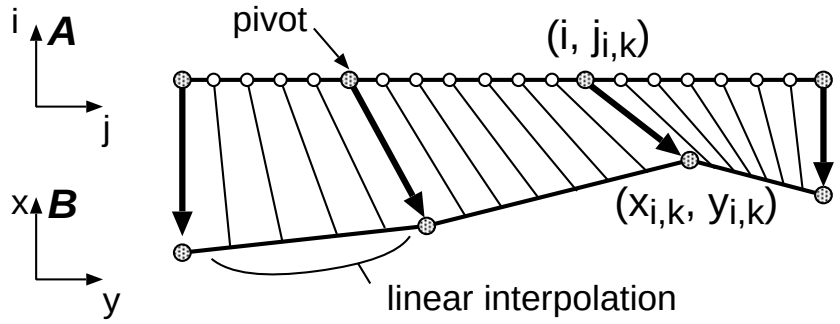


Abbildung 6: Darstellung einer verformten Bildzeile (aus [5])

Wenn die Pivotpixelabbildungen die Monotonie- und Stetigkeitseinschränkungen

$$\begin{aligned} |x_{i,k} - x_{i-1,k} - 1| &\leq 1 \\ |y_{i,k} - y_{i-1,k}| &\leq 1 \\ \left| \frac{y_{i,k} - y_{i,k-1}}{j_{i,k} - j_{i,k-1}} - 1 \right| &\leq 1 \end{aligned} \tag{11}$$

erfüllen, folgt daraus, dass dann auch die linear interpolierten Pixelabbildungen dieselben Einschränkungen erfüllen.

In Abbildung 7 werden Beispielbilder und deren Ergebnisse (Verformung und Distanz) des abschnittweisen linearen 2DW Algorithmus präsentiert. Die Laufzeit des Algorithmus ist hier exponentiell zu $K$, der Anzahl der Pivotpixel in einer Zeile.

# 5 Probleme des 2D Warping Algorithmus

Wie aussagekräftig die Ergebnisse dieser 2DW Algorithmen sind, hängt wie schon erwähnt von der Kostenfunktion und den Einschränkungen der Warpingabbildung ab. Im weiteren werden zwei Probleme, die die Einschränkungen der Warpingabbildung betreffen, beschrieben.

## 5.1 Deformationen

Je größer das zu verformende Bild ist, desto größer ist der maximal mögliche Unterschied der Pixelkoordinaten zwischen den zugehörigen abgebildeten Koordinaten. Um diese unerwünschten Verformungen (Deformationen) abzuwenden werden zwei zusätzliche Methoden vorgestellt.

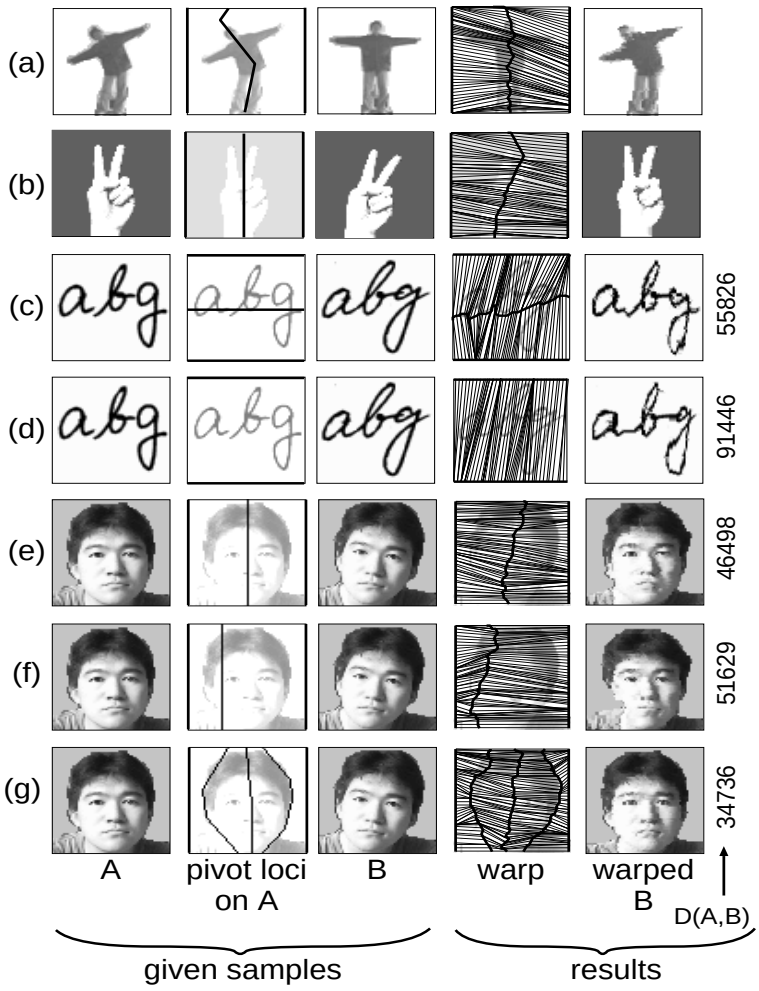


Abbildung 7: Beispielergebnisse des abschnittweise linearen 2DW Algorithmus (aus [5])

### 5.1.1 Warp Range Begrenzung

Die sogenannte *Warp Range*[2] Begrenzung ist ein einfacher Weg, um zu starke Deformationen zu verhindern. Bei Verwendung der Warp Range Begrenzung werden die zusätzlichen Abbildungseinschränkungen

$$|i - x(i,j)| \leq w, \qquad |j - y(j,y)| \leq w \tag{12}$$

eingeführt, wobei $w$ die maximalem Abbildungsweite eines Pixels ist, d.h. die x- und y-Koordinate eines abgebildeten Pixels dürfen sich maximal um den Wert $w$ von den Urspungskoordinaten unterscheiden.

### 5.1.2 Penalty Funktion

Die Penalty Funktion (Bestrafungsfunktion)

$$\begin{aligned} P(i,j) = &|x(i,j) - x(i,j-1)| + |x(i,j) - x(i-1,j) - 1| \\ &+|y(i,j) - y(i,j-1) - 1| + |y(i,j) - y(i-1,j)| \end{aligned} \tag{13}$$

bestraft starke Deformationen mit Kostenerhöhung

$$g(i,j,\mathbf{xy}) := d_{xy}(i,j,\mathbf{xy}) + \min_{\overline{\mathbf{xy}} \in \overline{\mathbf{XY}}(\mathbf{xy})} [g(i,j-1,\overline{\mathbf{xy}}) + \alpha * P(i,j)]. \tag{14}$$

Es sind weiterhin noch alle vorher schon möglichen Warpingabbildungen erlaubt. Die Wahl des Wertes $\alpha$ entscheidet jedoch, ob zu starke Deformationen noch rentabel sind (wie streng starke Verformungen bestraft werden), da zur Kostenfunktion die Penaltykosten aufsummiert werden. Dadurch werden zu niedrige Distanzen wegen zu starker Verformungen ausgeschlossen.

## 5.2 Unterschiedliche Bildgrößen

Die Anzahl der nach Einschränkungen erlaubten Warpingabbildungen wird durch Bildgrößenunterschiede beeinflusst. Wenn das zu verformende Bild doppelt so groß oder größer ist, kann dieser Algorithmus nicht mehr angewandt werden. Die Abbildung 8 zeigt ein Beispiel für zwei derart unterschiedlich große Bilder die der 2DW Algorithmus aufgrund seiner Abbildungseinschränkungen nicht verarbeiten kann. Generell verfälschen unterschiedliche Bildgrößen die Aussagekraft der Distanzwerte. Die Monotonie- und Stetigkeitseinschränkungen sind für gleich große Bilder definiert. So dass bei kleineren zu verformenden Bildern noch stärkere Bildverformungen und bei größeren Bildern nur schwächere Bildverformungen möglich sind. Das Gleichgewicht zwischen Kostenfunktion und Abbildungseinschränkungen wird durch unterschiedliche Bildgrößen gestört. Die von diesem 2DW Algorithmus zu vergleichenden Bilder sollten daher die gleiche Bildgröße haben.

[2] engl., hier etwa „Abbildungsweite“

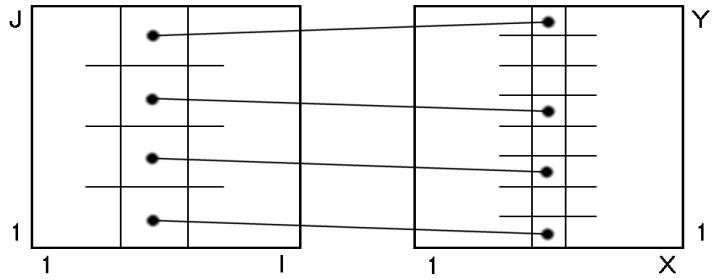


Abbildung 8: Beispiel für Bilder ohne mögliche Warpingabbildungen

# 6 Vergleich mit anderen Algorithmen

Es werden nun vier andere Verfahren zur Bewertung eines Bildvergleiches kurz vorgestellt, die mit den 2DW Algorithmen, die in dieser Seminararbeit erläutert wurden, verglichen werden.

Der Algorithmus von R. K. Moore, den er in der Arbeit [6] beschreibt, erweitert die eindimensionalen Levenshteindistanz für zweidimensionale Fälle. Bei diesem Verfahren gibt es durch die Definition der zweidimensionalen Levenshteindistanz die Einschränkung „kein Kreuzen“. Weitere Einschränkungen können nur mit großem Aufwand, Änderung der Definition der zweidimensionalen Levenshteindistanz, hinzugefügt werden. Die zu vergleichenden Bilder kann man hier auch als Pixel auf Pixel abbildende Abbildung, die durch den zweidimensionalen Austauschsatz bestimmt ist, verstehen. Dabei können Pixel, im Gegensatz zu den „effizienten“ 2DW Verfahren, auf die leere Menge abgebildet werden. Außerdem sind keine lokalen Stetigkeitseinschränkungen vorgesehen. Das Verfahren ist symmetrisch. Der Algorithmus ist mit der Definition der zweidimensionalen Levenshteindistanz schwer intuitiv zu verstehen im Gegensatz zur eindimensionalen Levenshteindistanz.

Der E. Levin und R. Pieraccini Algorithmus aus [7] gehört zu der Klasse der 2DW Algorithmen. Dieses Verfahren hat auch Einschränkungen bei Warpingabbildungen (kein Kreuzen). Es ergibt sich aus der Übertragung von in der Spracherkennung üblichen sogenannten *Hidden Markov Modellen* auf den 2D Fall („planares HMM“). Die Verformung wird aber anders als bei den „effizienten“ 2DW Algorithmen auf das Musterbild angewandt. Der Bildvergleich ist wie bei den „effizienten“ 2DW Verfahren nicht symmetrisch. Es ist möglich die Laufzeit hier erheblich zu verbessern, indem man zu sogenannten „Pseudo 2D HMM“ übergeht.

Die Bewertung eines Bildvergleiches durch die Tangenten Distanz wird in [8] beschrieben. Bei diesem Verfahren werden die Bilder als Elemente, bestimmt durch die Pixelwerte, eines hochdimensionalen Vektorraums betrachtet. Auf das Element, welches das Urspungsbild darstellt, sind Bildtransformationen (Warpingabbildungen) anwendbar. Die dadurch erzeugbaren Bilder kann man

| Algorithmus | Laufzeit |
|---|---|
| effizienter 2DW | $O(J^3 \cdot 9^J)$ |
| R. K. Moore | $O(J^6)$ |
| E. Levin, R. Pieraccini | $O(J^{4J})$ |
| Tangenten Distanz | $O(J^2)$ |
| Image Distortion | $O(J^2)$ |

Tabelle 1: Laufzeiten der vorgestellten Algorithmen ($I = J = X = Y$)

als Funktion im Vektorraum darstellen. Die Tangenten Distanz ist die minimale Distanz zwischen der Tangente dieser Funktion in dem Punkt, der das Ursprungsbild darstellt, und dem Punkt, der das zu vergleichende Bild darstellt. Dieses Verfahren ist nicht symmetrisch, kann jedoch auch symmetrisch (beidseitige Bestimmung der Tangente) angewandt werden.

Das Image Distortion Modell aus [8], dass in der Arbeit auch gemeinsam mit der Tangenten Distanz Anwendung fand, beschreibt die möglichen Bildtransformationen. Die Einschränkungen der Warpingabbildungen sind einzig durch einen Warp Range bestimmt. Dadurch kann eine passende Warpingabbildung lokal unter Betrachtung aller Pixel, die im Warp Range liegen, bestimmt werden.

Die Verfahren können alle durch einfache Änderungen ($D(\mathbf{A}, \mathbf{B}) := \tilde{D}(\mathbf{A}, \mathbf{B}) + \tilde{D}(\mathbf{B}, \mathbf{A})$) zu symmetrischen Verfahren gemacht werden. Die Laufzeiten der Algorithmen kann man einfach vergleichen (siehe Tabelle 1). Schwieriger ist die Bewertung der Ergebnisse der Algorithmen. Die Ergebnisse weichen durch die unterschiedlichen Einschränkungen, also durch die zugelassenen Abbildungen, sowie die gewählte Kostenfunktion (Bewertungsmassstäbe) voneinander ab. So kann man für jeden Algorithmus ein Problem finden, dass dieser am besten löst. Es gibt bisher nur einen Vergleich zwischen den Fehlerraten des 2DW Algorithmus und der *Nearest Neighbor* Methode. Wobei die Verfahren an japanischen Schriftzeichen angewandt wurden. Dabei wurde die Nearest Neighbor Fehlerrate von $6,1\%$ auf $3,2\%$ unter Verwendung von 2DW gesenkt. Letztendlich hängt es von den Bildinhalten und Anwendungszielen ab, welches Verfahren man wählt und gegebenenfalls modifiziert.

# Literatur

[1] S. Uchida und H. Sakoe. A monotonic and continuous two-dimensional warping based on dynamic programming. In *Proceedings of 14th IAPR International Conference on Pattern Recognition (ICPR'98)*, Band 1, Seiten 521–524, August 1998.

[2] S. Uchida und H. Sakoe. An Efficient Two-Dimensional Warping Algorithm. In *IEICE Transactions on Information & Systems*, Band E82-D, Nummer 3, Seiten 693–700, März 1999.

[3] S. Uchida und H. Sakoe. Handwritten Character Recognition Using Monotonic and Continuous Two-Dimensional Warping. In *Proceedings of 5th International Conference on Document Analysis and Recognition (ICDAR'99)*, Seiten 499–502, September 1999.

[4] S. Uchida und H. Sakoe. An Approximation Algorithm for Two-Dimensional Warping. In *IEICE Transactions on Information & Systems*, Band E83-D, Nummer 1, Seiten 109–111, Januar 2000.

[5] S. Uchida und H. Sakoe. Piecewise linear two-dimensional warping. In *Proceedings of 15th IAPR International Conference on Pattern Recognition (ICPR2000)*, Band 3, Barcelona, Spanien, Seiten 538–541, September 2000.

[6] R. K. Moore. A Dynamic Programming Algorithm for the Distance Between Two Finite Areas. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Band 1, Nummer 1, Seiten 86–88, Januar 1979.

[7] E. Levin und R. Pieraccini. Dynamic Planar Warping for Optical Character Recognition. In *ICASSP-92: 1992 IEEE International Conference on Acoustics, Speech and Signal Processing*, Band 3, Seiten 149–152, März 1992.

[8] D. Keysers, J. Dahmen, T. Theiner, und H. Ney. Experiments with an Extended Tangent Distance. In *Proceedings 15th IAPR International Conference on Pattern Recognition (ICPR2000)*, Band 2, Barcelona, Spanien, Seiten 38–42, September 2000.