

Adaptive Waveletmethoden zur Approximation von Bildern

DISSERTATION

zur Erlangung des mathematisch-naturwissenschaftlichen Doktorgrades
“Doctor rerum naturalium”
der Georg-August-Universität Göttingen

vorgelegt von Stefanie Tenorth
aus Wesel

Göttingen 2011

Referentin/Referent: Prof. Dr. Gerlind Plonka-Hoch

Korreferentin/Korreferent: Prof. Dr. Armin Iske

Ggf. weitere Referentin/weiterer Referent: -

Tag der mündlichen Prüfung: 08.07.2011

Inhaltsverzeichnis

1	Einleitung	5
1.1	Nichtadaptive Algorithmen	5
1.2	Adaptive Algorithmen	6
1.3	Aufbau der Arbeit	8
2	Grundlagen	11
2.1	Das Haar-Wavelet	11
2.2	Multiresolutionsanalyse und orthonormale Wavelets	15
2.3	Die schnelle orthogonale Wavelet-Transformation	17
2.4	Biorthogonale Wavelets	21
2.4.1	Schnelle biorthogonale Wavelet-Transformation	22
2.4.2	Beispiel für biorthogonale Wavelets	24
2.5	Tensorprodukt-Wavelet-Transformation	24
2.6	Weitere Begriffe	24
3	Die Easy-Path-Wavelet-Transformation	27
3.1	Notation	27
3.2	Die EPWT	29
3.2.1	Erstes Level der EPWT	29
3.2.2	Bestimmung des Pfades p^{2J}	29
3.2.3	Anwendung der Wavelettransformation	30
3.2.4	Zweites Level der EPWT	31
3.2.5	Bestimmung des Pfades p^{2J-1}	31
3.2.6	Anwendung der Wavelet-Transformation	32
3.2.7	Weitere Level der EPWT	32
3.3	Eine Relaxation der EPWT	34
3.4	Reduzierung der Pfadkosten in weiteren Leveln	39

3.4.1	Erste Variante	39
3.4.2	Direkte-Nachbarn-EPWT	41
3.4.3	Mittelpunkt-EPWT	43
3.4.4	Einfach-EPWT und Halbe EPWT	44
3.5	Numerische Ergebnisse	45
3.6	Ausblick	52
4	Hybrid-Algorithmen	55
4.1	1. Schritt: Zerlegung des digitalen Bildes	55
4.2	2. Schritt: Approximation des geglätteten Bildes	56
4.3	3. Schritt: Approximation der Kanten und Texturen	57
4.3.1	Beispiel	60
4.4	4. Schritt: Zusammenfügen	62
4.5	Ein Beispiel zur Erläuterung	64
4.6	Numerische Ergebnisse	66
5	Approximationsresultate	77
5.1	Approximationseigenschaften der EPWT - der Fall $0 < \alpha \leq 1$	78
5.1.1	Adaptive Haarwaveletbasis	82
5.1.2	Anforderungen an die Pfade	84
5.1.3	Beispiel	85
5.1.4	Algorithmus zur Bestimmung des Pfades	87
5.1.5	Approximationseigenschaften der EPWT	89
5.2	Approximationseigenschaften der EPWT - der Fall $\alpha > 0$	99
5.2.1	Eine leichte Abänderung der EPWT	101
	Beispiel	104
5.2.2	Einige Abschätzungen für das erste Level	106
5.2.3	Anforderungen an die Pfadvektoren	110
5.2.4	Abschätzungen für höhere Level der EPWT	111
5.2.5	Beweis zur N-Term-Approximation für $\alpha > 0$	114
	Literaturverzeichnis	119

1 Einleitung

Wavelets sorgen im eindimensionalen Fall für eine effektive Darstellung eines stückweise glatten Signals. Bei zweidimensionalen Daten wie z.B. Bildern liefern Tensorprodukt-Wavelet-Basen allerdings nur für global glatte Bilder optimale dünn besetzte Darstellungen. Für Bilder mit Ecken und Kanten lassen sich mit Tensor-Produkt-Wavelets keine so effizienten Approximationen finden. Deshalb wurden in den letzten Jahren einige Verfahren entwickelt, die bessere Resultate bei der Approximation von Bildern erzielen.

1.1 Nichtadaptive Algorithmen

Zum Einen sind dies nicht-adaptive Methoden wie zum Beispiel die von Candès und Donoho entwickelte *Ridgelet*-Transformation [8], die grob gesagt darin besteht, zuerst eine Radon-Transformation auf das gegebene Bild anzuwenden und dann auf die resultierenden Schnitte eine eindimensionale Wavelet-Transformation anzuwenden. Die anisotropen Ridgelets können gerade Kanten besser approximieren als Wavelets, sind bei gekrümmten Kanten allerdings wenig erfolgreich. Mit dem Ziel, auch die gekrümmten Kanten effizient approximieren zu können, wurden deshalb aus den Ridgelets die *Curvelets* [9, 7] entwickelt. Die Curvelets der „ersten Generation“ beruhen auf der Idee, vgl. auch [25], dass gekrümmte Kanten lokal in kleinen Ausschnitten fast gerade sind. Das Bild wird deshalb in kleine Kästchen zerlegt und in jedem Kästchen die Ridgelet-Transformation angewendet. Die Curvelet-Funktionen der ersten Generation besitzen sieben Indizes und sind in der Praxis nicht gut handhabbar. Außerdem erhält man aufgrund der Überlappungen bei der Zerlegung eine hohe Redundanz, die durch die Curvelets der „zweiten Generation“, bei denen es „nur“ die drei Parameter (bzw. Indizes) Skalierung, Winkel und Ort gibt, verringert wird. Allerdings ist die Redundanz auch hier immer noch sehr hoch, so dass sich Curvelets eher für die Entstörung als für die Kompression von Bildern eignen. Für Curvelets erhält man das theoretische Resultat (siehe [10]), dass die Approximation f_M einer Funktion f , aus einer Curvelet-Entwicklung von f mit nur M günstig gewählten Termen entsteht, die

Abschätzung

$$\|f - f_M\|^2 \leq CM^{-2} (\log_2 M)^3$$

erfüllt, sofern f stückweise Hölder-glatt der Ordnung 2 ist und Unstetigkeiten lediglich entlang C^2 -Kurven besitzt. Zum Vergleich: eine Tensor-Produkt-Wavelet-Transformation hat den Approximationsfehler $\mathcal{O}(M^{-1})$, siehe [38]. Ein weiteres Beispiel für nicht-adaptive Methoden sind die von Guo und Labate entwickelten *Shearlets* [27, 28]. Das Shearlet-System wird von einer Mutter-Shearlet-Funktion erzeugt, die skaliert, geschert („sheared“) und verschoben wird. Der Scherungsparameter sorgt dafür, dass die Richtungen der Kanten des Bildes mitberücksichtigt werden. Shearlets sind ebenfalls stark redundant. Für Shearlets gilt ein ähnliches Resultat bezüglich der M -Term-Approximation, wie für Curvelets, vgl. [27]. Wenn das Bild allerdings Kanten besitzt, die nicht mehr stückweise C^2 -glatt sind, verlieren sowohl Shearlets als auch Curvelets diese optimale Approximationseigenschaft. Dies ist der wesentliche Grund dafür, dass auf Curvelets bzw. Shearlets beruhende Kompressionsverfahren keine genügend guten Resultate liefern. Weitere Beispiele für nicht-adaptive Verfahren sind z.B. die *Directionlets* [55], und die *Contourlets* [23], die im Vergleich zu Curvelets eine nicht ganz so hohe Redundanz aufweisen.

1.2 Adaptive Algorithmen

Viele Approximationsmethoden [1, 4, 5, 12, 17, 19, 21, 23, 24, 26, 30, 31, 35, 42, 39, 49, 50, 51, 56, 37] folgen dem Ansatz, nicht zu Beginn eine Basis oder einen Frame festzulegen, sondern das Approximationsverfahren an die Geometrie des Bildes anzupassen. So besteht die Idee der von Donoho entwickelten *Wedgelets* [24] zum Beispiel darin, das Bild in Keile („wedges“) zu unterteilen. Diese keilförmigen Gebiete entstehen dadurch, dass man dyadische Quadrate mit einem geraden „Schnitt“ in zwei Teile teilt. Die Wedgelets sind nun charakteristische Funktionen auf diesen Keilen. Ein Kompressionsverfahren, das Wedgelets und Wavelets kombiniert um bessere Resultate zu erzielen ist von Wakin, Romberg, Choi und Baraniuk vorgestellt worden [56]. Die von LePennecc und Mallat konstruierten *Bandelets* [42] sind anisotrope Wavelets, die entlang des geometrischen Flusses des Bildes gekrümmt werden. Für Bandelets lassen sich ebenfalls optimale Approximationsresultate zeigen, [42]. Von Krommweh [37] stammen die *Tetrolets*, die verallgemeinerte Haarwavelets auf Tetrominoes (bekannt auch aus dem Computerspiel Tetris) sind. Die Tetromino-Zerlegung des Bildes wird an die Geometrie des Bildes angepasst, so dass sich

ein Tetromino möglichst nicht über eine Kante erstreckt. Da es sehr viele Möglichkeiten für eine Tetromino-Zerlegung des ganzen Bildes gibt, beschränkt man sich hier auf die Bestimmung von Tetromino-Zerlegungen von 4×4 -Quadraten des Bildes. Die *Grouplets* von Mallat [39] werden mit Hilfe von Assoziationsfeldern erzeugt, die Punkte sinnvoll unter Berücksichtigung der Geometrie des Bildes gruppieren. Auf die gruppierten Punkte wird eine gewichtete Haar-Wavelet-Transformation angewendet. Demaret, Dyn und Iske [21] entwickelten ein Kompressionsverfahren, bei dem das Bild durch einen linearen Spline auf einer adaptiven Delaunay-Triangulierung des Bildes approximiert wird. Die Punkte des Bildes, die für die Delaunay-Triangulierung verwendet werden, erhält man durch eine sukzessive Ausdünnung der Bildpunkte, so dass nur die wichtigsten Punkte übrig gelassen werden. Von Claypoole, Davis, Sweldens und Baraniuk wurden mittels *Lifting-Schema* nichtlineare Wavelet-Transformationen entwickelt, [12]. Heijmans, Pesquet-Popescu und Piella entwarfen ein auf dem Lifting-Schema beruhendes Verfahren, das den Prediction-Schritt fest lässt, aber für den Update-Schritt aus zwei verschiedenen Update-Operatoren bei einem kleinen lokalen Gradienten den einen Update-Operator und bei einem großen lokalen Gradienten den anderen Update-Operator auswählt, [32]. Plonka und Tenorth [45] stellten im Rahmen der SampTA 2009 in Marseille ebenfalls ein Verfahren vor, das das Liftingschema verwendet. Während für den Update-Operator ein linearer Operator gewählt wird, beruht die Konstruktion des adaptiven Prediction-Operators auf dem Konzept der nichtlinearen Diffusion. Nichtlineare Multiresolutionszerlegungen, die auf *ENO-* (*essentially-non-oscillatory*) und *ENO-SR-Methoden* (*subcell resolution*) basieren, wurden als Erstes von Harten für den eindimensionalen Fall entworfen [30, 31]. Zur ENO- (und ENO-SR-) Methode gibt es zum Beispiel von Arandiga, Cohen, Matei et. al. einige Arbeiten [3, 5, 1, 4, 17], inklusive einer zweidimensionalen kantenadaptierten Erweiterung (*ENO-EA-Methode*, *edge-adapted*) des von Harten entwickelten Verfahrens. Die ENO-EA-Methode liefert laut [3] für stückweise glatte Funktionen f mit Unstetigkeiten entlang C^2 -Kurven eine optimale N -Term-Approximation

$$\|f - f_M\|^2 \leq CM^{-2}.$$

Elad, Ram und Cohen verallgemeinerten in [50] ein von Gavish, Nadler und Coifman in [26] vorgestelltes Verfahren einer Haar-Wavelet-Transformation, die auf einer Baumstruktur beruht. Dabei werden die Pixel als Blattebene eines binären Baumes betrachtet und davon ausgehend Ebene für Ebene die Vaterknoten der Knoten der zuvor bestimmten Ebene berechnet. Hierbei bekommen zwei Knoten den gleichen Vater, die besonders „ähnlich“

sind. In jedem Level des Baumes kann nun auf die Funktionswerte der zugehörigen Knoten in der durch den Baum festgelegte Sortierung ein Level der Haar-Wavelet-Transformation angewendet werden.

1.3 Aufbau der Arbeit

Diese beiden zuletzt genannten Verfahren sind der von Plonka in [43] vorgestellten *Easy-Path-Wavelet-Transformation* (EPWT) sehr ähnlich. Dort wird zunächst ein Pfad durch die Pixel eines gegebenen Bildes gesucht, so dass auf diesem Pfad benachbarte Pixel einen ähnlichen Funktionswert besitzen. Der Pfad soll durch jedes Pixel genau einmal gehen. Auf die so entstehende Permutation der Funktionswerte der Pixel wird nun eine eindimensionale Wavelet-Transformation angewendet. Durch die so entstehenden Tiefpass-Koeffizienten wird nun wieder ein Pfad gesucht und entlang dessen ein weiteres Level der EPWT angewendet. Wie bei den meisten anderen adaptiven Verfahren (z.B. bei Elad et.al.) entstehen auch hier zusätzliche Speicherkosten (bei Elad durch die Speicherung der Baumstruktur), da der Pfad gespeichert werden muss, um eine Rekonstruktion des Originalbildes zu ermöglichen. Um diese Speicherkosten zu verringern, wurde eine relaxierte Version der EPWT entwickelt. Nach einer kurzen Einführung der benötigten Grundlagen in Kapitel 2 stellen wir die EPWT und die relaxierte EPWT in Kapitel 3 ausführlich vor. Außerdem werden weitere neue Möglichkeiten zur Reduzierung der Adaptivitätskosten, die über [43] hinausgehen und teilweise in unserer Arbeit [48] für die auf der EPWT basierenden Hybrid-Methode vorgestellt wurden, angeführt und numerische Ergebnisse präsentiert.

Im vierten Kapitel beschäftigen wir uns mit der zuvor erwähnten von uns neu entwickelten Hybrid-Methode [48]. Hier wird die EPWT nicht auf das ganze Bild angewendet, sondern nur auf den Teil des Bildes, der Unstetigkeiten (Kanten) und Texturen enthält. Im ersten Schritt wird das Originalbild in einen glatten Teil und einen Teil, der die Kanten und Texturen enthält, zerlegt. Auf den glatten Teil wird eine Tensor-Produkt-Wavelet-Transformation angewendet. Die Kanten und Texturen des Bildes werden mit der EPWT approximiert. Auch hier werden wir zum Abschluss des Kapitels die Güte des Verfahrens durch numerische Ergebnisse demonstrieren.

Das fünfte Kapitel behandelt die Approximationseigenschaften der EPWT. Für stückweise Hölder-stetige Funktionen der Ordnung α , die nur entlang von endlichen Kurven Singularitäten besitzen, können wir, wenn die Pfade sinnvoll gewählt werden, zeigen, dass

die optimale Abschätzung

$$\|f - f_N\|_2^2 \leq CN^{-\alpha}$$

gilt, wobei f_N die Approximation von f ist, die entsteht, wenn man für die Rekonstruktion von f nur die wichtigsten N EPWT-Koeffizienten behält. Dieses Approximationsresultat wird zunächst für $0 < \alpha \leq 1$ gezeigt, siehe hierzu auch unser Paper [46]. Der allgemeine Fall $\alpha > 0$ (siehe auch [47]) wird dann im zweiten Teil des letzten Kapitels bewiesen.

Danksagung

Ich danke meiner Betreuerin Prof. Gerlind Plonka-Hoch für ihre Unterstützung und Hilfsbereitschaft. Außerdem möchte ich ihr dafür danken, mich dazu ermutigt zu haben, diese Dissertation zu schreiben. Ich bedanke mich ebenfalls bei der gesamten Arbeitsgruppe für das nette Arbeitsklima, das sicher nicht überall selbstverständlich ist. Prof. Armin Iske danke ich dafür, dass er sich dazu bereit erklärt hat, das Zweitgutachten zu schreiben.

Diese Dissertation ist im Rahmen des DFG-Schwerpunktprogramms 1324 „Extraktion quantifizierbarer Information aus komplexen Systemen“ entstanden, dem ich an dieser Stelle für seine finanzielle Unterstützung danken möchte.

Desweiteren möchte ich mich bei meiner Familie und bei meinem Freund dafür bedanken, dass sie stets für mich da sind.

2 Grundlagen

In diesem Kapitel widmen wir uns einigen orthogonalen und biorthogonalen Waveletfiltern, die von Haar, Daubechies und unter anderem auch von Cohen und Feauveau entwickelt wurden, siehe auch [18]. Unser Ziel ist es, ein gegebenes eindimensionales oder zweidimensionales Signal so zu transformieren, dass das neue Signal sehr gut dekorreliert ist, d.h. viele betragsmäßig kleine Koeffizienten enthält, während die Hauptinformation des Signals in wenigen betragsmäßig großen Koeffizienten gespeichert ist. Zusätzlich soll gegeben sein, dass man aus dem neuen Signal wieder das ursprüngliche Signal (oder zumindest eine gute Näherung davon) rekonstruieren kann. Diese dünnbesetzte Darstellung lässt sich natürlich effizienter kodieren. Als erstes Beispiel widmen wir uns der Haarwavelet-Transformation, die von Alfréd Haar 1910 in [29] vorgestellt wurde.

2.1 Das Haar-Wavelet

Zu einer Funktion $f : [0, N[\rightarrow \mathbb{R}$ mit $N = 2^n$, seien $n \in \mathbb{N}$ gleichmäßig verteilte Werte $\{\mathbf{f}_0, \dots, \mathbf{f}_{N-1}\}$ mit $f_j = f(j)$ bekannt. Die Originalfunktion f ist somit näherungsweise durch

$$\tilde{f}(x) := \sum_{k=0}^{N-1} f_k \chi_{[k, k+1[}(x) \quad \text{mit} \quad \chi_{[k, k+1[}(x) := \begin{cases} 1 & \text{wenn } x \in [k, k+1[\\ 0 & \text{sonst} \end{cases}$$

gegeben. Unser Ziel ist es nun, eine effizienter zu speichernde Approximation von \tilde{f} zu bestimmen. Die Idee der Haarwavelet-Transformation besteht darin, zwei Treppenstufen der Treppenfunktion \tilde{f} durch eine doppelt so breite Treppenstufe, deren Wert die Werte der beiden schmalen Treppenstufen mittelt, auszudrücken. Um eine perfekte Rekonstruktion gewährleisten zu können, muss dann selbstverständlich noch die Differenz von der breiten

Treppenstufe zu den schmaleren gespeichert werden. Dazu betrachten wir

$$\varphi(x) := \chi_{[0, \frac{1}{2}[}(x) + \chi_{[\frac{1}{2}, 1[}(x) = \begin{cases} 1 & \text{wenn } x \in [0, 1[\\ 0 & \text{sonst} \end{cases},$$

die wir auch als *Skalierungsfunktion* bezeichnen, und

$$\psi(x) := \chi_{[0, \frac{1}{2}[}(x) - \chi_{[\frac{1}{2}, 1[}(x) = \begin{cases} 1 & \text{wenn } 0 \leq x < \frac{1}{2} \\ -1 & \text{wenn } \frac{1}{2} \leq x < 1 \\ 0 & \text{sonst} \end{cases},$$

unser sogenanntes *Mutterwavelet* oder *Waveletfunktion*.

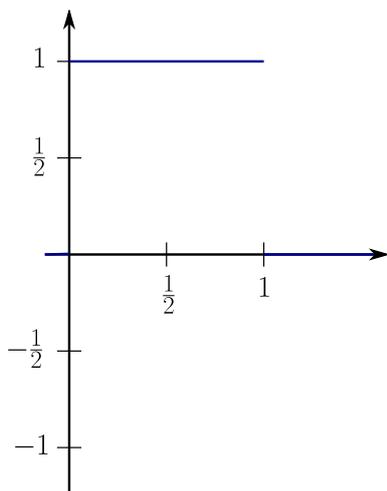


Abbildung 2.1: Skalierungsfunktion

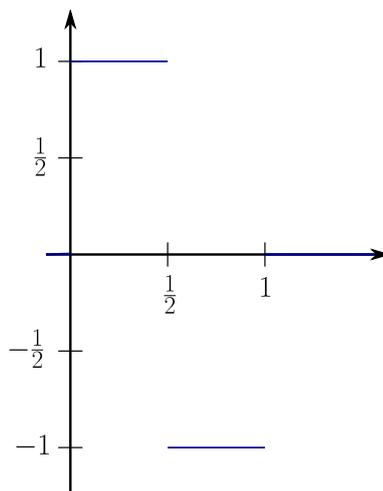


Abbildung 2.2: Mutterwavelet

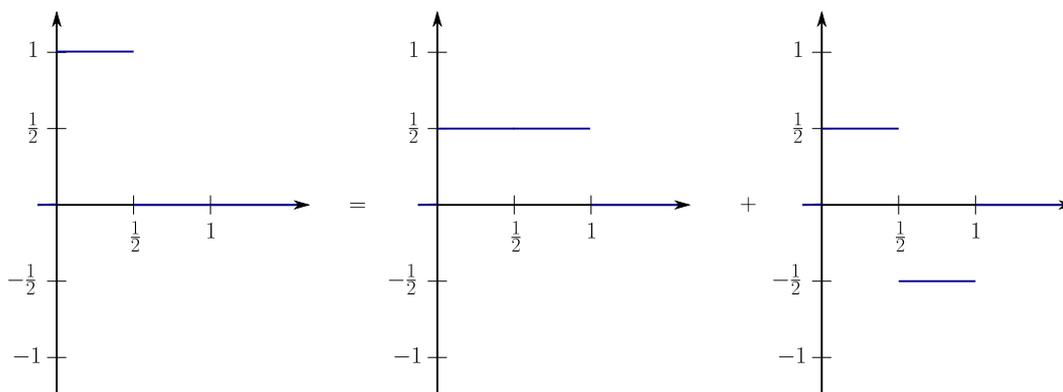
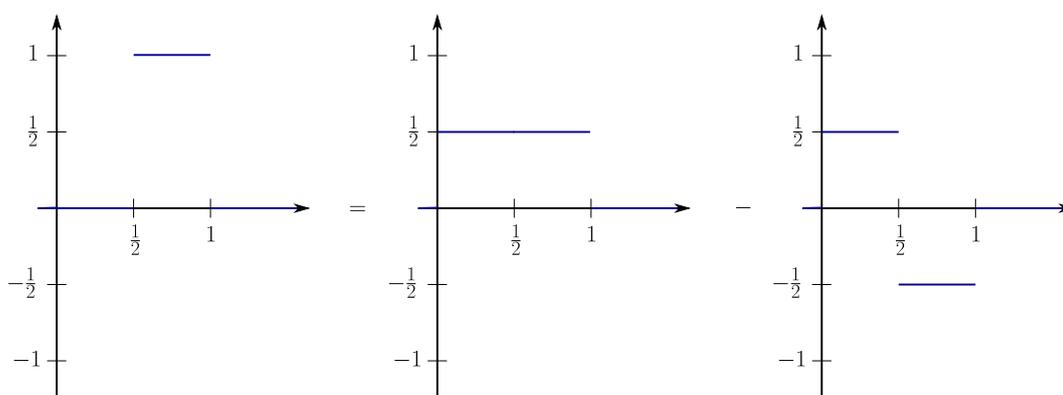
Die einzelnen Stufen $\chi_{[0, \frac{1}{2}[}$ und $\chi_{[\frac{1}{2}, 1[}$ lassen sich nun mittels einer „großen Stufe“ $\chi_{[0, 1[} = \varphi$ und der jeweiligen Differenz darstellen, das heißt, es gilt

$$\chi_{[0, \frac{1}{2}[}(x) = \frac{1}{2} (\varphi(x) + \psi(x))$$

und

$$\chi_{[\frac{1}{2}, 1[} = \frac{1}{2} (\varphi(x) - \psi(x)),$$

siehe auch Abbildung 2.3 und 2.4.

Abbildung 2.3: $\chi_{[0, \frac{1}{2}[}(x) = \frac{1}{2}(\varphi(x) + \psi(x))$ Abbildung 2.4: $\chi_{[0, \frac{1}{2}[}(x) = \frac{1}{2}(\varphi(x) - \psi(x))$

Mit ψ_k^j und φ_k^j seien die Funktionen

$$\psi_k^j(x) := 2^{\frac{j}{2}} \psi(2^j x - k) = \begin{cases} 2^{\frac{j}{2}} & \text{wenn } x \in [\frac{k}{2^j}, \frac{2k+1}{2^{j+1}}[\\ -2^{\frac{j}{2}} & \text{wenn } x \in [\frac{2k+1}{2^{j+1}}, \frac{k+1}{2^j}[\\ 0 & \text{sonst} \end{cases} \quad j, k \in \mathbb{Z} \quad (2.1)$$

und

$$\varphi_k^j(x) := 2^{\frac{j}{2}} \varphi(2^j x - k) = \begin{cases} 2^{\frac{j}{2}} & \text{wenn } x \in [\frac{k}{2^j}, \frac{k+1}{2^j}[\\ 0 & \text{sonst} \end{cases} \quad j, k \in \mathbb{Z} \quad (2.2)$$

bezeichnet, die durch Skalierung, Verschiebung und Streckung/Stauchung von ψ und φ entstehen. Für festes j sind die ψ_k^j und φ_k^j Verschiebungen der Funktionen $2^{\frac{j}{2}}\psi(2^j x)$ bzw.

$2^{\frac{j}{2}}\varphi(2^j x)$ um $\frac{k}{2^j}$, $k \in \mathbb{Z}$. Es gilt offensichtlich, dass sich die Funktion φ_k^{j-1} durch die zwei Funktionen φ_{2k}^j und φ_{2k+1}^j auf folgende Weise

$$\varphi_k^{j-1} = 2^{\frac{j-1}{2}} \chi_{[\frac{2k}{2^j}, \frac{2(k+1)}{2^j}[} = \frac{2^{\frac{j}{2}}}{\sqrt{2}} \left(\chi_{[\frac{2k}{2^j}, \frac{2k+1}{2^j}[} + \chi_{[\frac{2k+1}{2^j}, \frac{2(k+1)}{2^j}[} \right) = \frac{\varphi_{2k}^j + \varphi_{2k+1}^j}{\sqrt{2}}$$

ausdrücken lässt. Deswegen gilt für die Räume

$$V_j := \overline{\text{span} \{ \varphi_k^j \mid k \in \mathbb{Z} \}}, \quad j \in \mathbb{Z},$$

wobei der Abschluss bezüglich L^2 gemeint ist, folgende Schachtelung

$$\dots \subset V_{j-1} \subset V_j \subset \dots \subset V_{-1} \subset V_0 \subset V_1 \subset V_2 \subset \dots$$

Die Räume V^j bestehen aus stückweise konstanten Funktionen, die höchstens an den Stellen $\frac{k}{2^j}$, $k \in \mathbb{Z}$ Sprünge besitzen. Mit W_j , $j \in \mathbb{Z}$ seien die Räume

$$W_j := \overline{\text{span} \{ \psi_k^j \mid k \in \mathbb{Z} \}}, \quad j \in \mathbb{Z},$$

bezeichnet, die aus Funktionen bestehen, die höchstens an den Stellen $\frac{k}{2^{j+1}}$, $k \in \mathbb{Z}$, Sprünge besitzen und in den Intervallen $[\frac{k}{2^j}, \frac{k+1}{2^j}[$ den Mittelwert 0 besitzen. Es gilt

$$\langle \psi_k^j, \phi_k^j \rangle = 2^{\frac{j}{2}} \int_{\frac{k}{2^j}}^{\frac{k+1}{2^j}} \psi_k^j = 0$$

und

$$\begin{aligned} \varphi_{2l}^j &= 2^{\frac{j}{2}} \chi_{[\frac{2l}{2^j}, \frac{2l+1}{2^j}[} = \frac{1}{\sqrt{2}} 2^{\frac{j-1}{2}} \left(\chi_{[\frac{l}{2^{j-1}}, \frac{l+1}{2^{j-1}[} + \chi_{[\frac{l}{2^{j-1}}, \frac{l+\frac{1}{2}}{2^{j-1}[} - \chi_{[\frac{l+\frac{1}{2}}{2^{j-1}}, \frac{l+1}{2^{j-1}[} \right) \\ &= \frac{1}{\sqrt{2}} (\varphi_l^{j-1} + \psi_l^{j-1}) \end{aligned}$$

sowie

$$\begin{aligned} \varphi_{2l+1}^j &= 2^{\frac{j}{2}} \chi_{[\frac{2l+1}{2^j}, \frac{2l+2}{2^j}[} = \frac{1}{\sqrt{2}} 2^{\frac{j-1}{2}} \left(\chi_{[\frac{l}{2^{j-1}}, \frac{l+1}{2^{j-1}[} - \left(\chi_{[\frac{l}{2^{j-1}}, \frac{l+1}{2^{j-1}[} - \chi_{[\frac{l+\frac{1}{2}}{2^{j-1}}, \frac{l+\frac{1}{2}}{2^{j-1}[} \right) \right) \\ &= \frac{1}{\sqrt{2}} (\varphi_l^{j-1} - \psi_l^{j-1}). \end{aligned}$$

Dann gilt $V_j = V_{j-1} \oplus W_{j-1} = V_{j-2} \oplus W_{j-2} \oplus W_{j-1} = \dots$. Mit $P_j : L^2 \rightarrow V_j$ sei der orthogonale Projektor von L^2 auf V_j bezeichnet. Somit erhalten wir für $f \in L^2$

$$P_j f = \sum_{k \in \mathbb{Z}} \langle P_j f, \varphi_k^j \rangle \varphi_k^j = \sum_{k \in \mathbb{Z}} \langle f, \varphi_k^j \rangle \varphi_k^j = \sum_{k \in \mathbb{Z}} 2^j \left(\int_{\frac{k}{2^j}}^{\frac{k+1}{2^j}} f(x) dx \right) \chi_{[\frac{k}{2^j}, \frac{k+1}{2^j}[}$$

Für $j \rightarrow \infty$ konvergiert $P_j f$ gegen f , das heißt, dass für den Abschluss der Vereinigung der V_j bezüglich L^2

$$\overline{\bigcup_{j \in \mathbb{Z}} V_j} = L^2$$

gilt. Die Approximation einer Funktion $f \in L^2$ durch eine auf Intervallen der Länge $2^{-(j+1)}$ konstante Funktion $P_{j+1} f$ lässt sich als Summe ihrer „nächstgrößeren“ Approximation im Raum der auf Intervallen der Länge 2^{-j} konstanten Funktionen, V_j , und den Details von f , die bei der Verringerung der Auflösung verloren gehen, nämlich

$$Q_j f := \sum_{k \in \mathbb{Z}} \langle f, \psi_{j,k} \rangle \psi_{j,k} \in W_j,$$

darstellen, d.h. $P_{j+1} f := P_j f + Q_j f$. Die einzige Funktion, die in $\bigcap_{j \in \mathbb{Z}} V_j$ liegt, ist die Nullfunktion, da keine andere Funktion der Form

$$f(x) = \begin{cases} a & x > 0 \\ b & x \leq 0 \end{cases}$$

mit $a, b \in \mathbb{R}$ in L^2 ist. Die Funktionen aus $\{\psi_{j,k} \mid j, k \in \mathbb{Z}\}$ bilden eine Orthonormalbasis von L^2 . Außerdem sind die $\{\varphi_k^j\}_{k \in \mathbb{Z}}$ eine Orthonormalbasis von V_j , was direkt an der Definition von V_j und φ zu sehen ist. Aus der Definition der φ_k^j , $k \in \mathbb{Z}$ folgt, dass $f(t) \in V_j \Leftrightarrow f(2t) \in V_{j+1}$ und $f(t) \in V_0 \Leftrightarrow f(t-k) \in V_0$ für alle $k \in \mathbb{Z}$.

2.2 Multiresolutionsanalyse und orthonormale Wavelets

2.2.i Definition. Eine Zerlegung von $L^2(\mathbb{R}^d)$ in geschachtelte abgeschlossene Unterräume

$$\dots \subset V_2 \subset V_1 \subset V_0 \subset V_1 \subset V_2 \subset \dots \subset L^2(\mathbb{R}),$$

die die Eigenschaften

- (i) $\lim_{j \rightarrow -\infty} V_j = \bigcap_{j \in \mathbb{Z}} V_j = \{0\}$, $\lim_{j \rightarrow \infty} V_j = \bigcup_{j \in \mathbb{Z}} V_j$ dicht in $L^2(\mathbb{R})$
- (ii) $f(t) \in V_j \Leftrightarrow f(2t) \in V_{j+1}$
- (iii) $f(t) \in V_0 \Leftrightarrow f(t - k) \in V_0$ für alle $k \in \mathbb{Z}$
- (iv) $\exists \varphi \in V_0$ so dass $\{\varphi(t - k)\}_{k \in \mathbb{Z}}$ eine orthonormale Basis von V_0 ist

hat, wird *Multiresolutionsanalyse* (MRA) oder *Multiskalenanalyse* genannt [38, 40].

Bei Punkt (iv) kann die Forderung der orthonormalen Basis durch die Forderung nach einer Rieszbasis ersetzt werden.

2.2.ii Definition. Mit *Rieszbasis* eines Hilbertraums H wird eine Menge $B := \{b_n \mid n \in \mathbb{Z}\}$ bezeichnet, für die

- $\overline{\{b_n \mid n \in \mathbb{Z}\}} = H$
- $\exists C_1, C_2, 0 < C_1 \leq C_2 < \infty$, so dass $C_1 \|c\|_{l^2}^2 \leq \|\sum_{n \in \mathbb{Z}} c_n b_n\|_H^2 \leq C_2 \|c\|_{l^2}^2$ für alle $c := \{c_n\}_{n \in \mathbb{Z}} \in l^2$

gilt.

Die Funktionen

$$\varphi_{j,k}(t) := 2^{\frac{j}{2}} \varphi(2^j t - k) \in V_j$$

bilden nach Definition 2.2.i eine Orthonormalbasis von V_j . Die orthogonale Projektion einer Funktion $f \in L^2(\mathbb{R})$ auf V_j ist, wie auch schon im Beispiel zum Haarwavelet, durch

$$P_j f(t) := \sum_{k \in \mathbb{Z}} \langle f, \varphi_{j,k} \rangle \varphi_{j,k}(t)$$

festgelegt. Die Differenz zwischen einer Projektion auf V_j und einer Projektion auf einen feineren Raum V_{j+1} sei mit $Q_j f := P_{j+1} f - P_j f$ bezeichnet. Der abgeschlossene Teilraum, auf den die orthogonale Projektion Q_j abbildet, sei mit W_j bezeichnet. Es gilt $V_{j+1} = V_j \oplus W_j$, und da $\bigcup_{j \in \mathbb{Z}} V_j$ dicht in $L^2(\mathbb{R})$ ist, folgt

$$L^2(\mathbb{R}) = \overline{\bigoplus_{j \in \mathbb{Z}} W_j}.$$

Mit Hilfe von φ lässt sich dann (vgl. auch [38]) ein *Wavelet* ψ bestimmen, so dass

$$\{\psi(x - k)\}_{k \in \mathbb{Z}}$$

eine Orthonormalbasis von W_0 ist. Wir setzen

$$\psi_{j,k}(t) := 2^{\frac{j}{2}} \psi(2^j t - k),$$

dann gilt

$$W_j = \overline{\text{span}\{\psi_{j,k}\}_{k \in \mathbb{Z}}}.$$

Aufgrund der Orthonormalität der $\{\psi_{j,k}\}_{j,k \in \mathbb{Z}}$ gilt für die orthogonale Projektion von f auf W_j

$$Q_j f = \sum_{k \in \mathbb{Z}} \langle f, \psi_{j,k} \rangle \psi_{j,k}.$$

Demnach lässt sich eine Funktion $f \in L^2(\mathbb{R})$ in der Form

$$f(x) = \sum_{j \in \mathbb{Z}} \sum_{k \in \mathbb{Z}} c_{j,k} \psi_{j,k} \quad \text{mit } c_{j,k} := \langle f, \psi_{j,k} \rangle$$

darstellen. Die $c_{j,k}$ werden auch *Waveletkoeffizienten* genannt.

2.3 Die schnelle orthogonale Wavelet-Transformation

Sei eine Funktion $f \in V_{j_0}$, $j_0 \in \mathbb{Z}$ fest, durch

$$f(x) = \sum_{k \in \mathbb{Z}} a_{j_0,k} \varphi_{j_0,k} \quad \text{mit } a_{j_0,k} := \langle f, \varphi_{j_0,k} \rangle$$

gegeben. Wir können o.B.d.A. annehmen, dass $j_0 = 0$ ist. Gesucht ist nun eine Darstellung von f mittels der Waveletkoeffizienten, d.h. in der Form

$$f(x) = \sum_{j=-\infty}^{-1} \sum_{k \in \mathbb{Z}} c_{j,k} \psi_{j,k}.$$

Wegen $\varphi \in V_0 \subset V_1$ existieren Koeffizienten $\{h_k\}_{k \in \mathbb{Z}}$, so dass

$$\varphi(t) = \sum_{k \in \mathbb{Z}} h_k \varphi_{1,k}(t) = \sqrt{2} \sum_{k \in \mathbb{Z}} h_k \varphi(2t - k)$$

gilt, womit

$$\varphi_{j,k}(x) = 2^{\frac{j}{2}} \varphi(2^j x - k) = \sum_{n \in \mathbb{Z}} h_n \varphi_{j+1,2k+n}(x) = \sum_{m \in \mathbb{Z}} h_{m-2k} \varphi_{j+1,m}(x) \quad (2.3)$$

und schließlich

$$a_{j,k} = \langle f, \varphi_{j,k} \rangle = \sum_{m \in \mathbb{Z}} \overline{h_{m-2k}} \langle f, \varphi_{j+1,m} \rangle$$

folgt. Analog gilt für $\psi \in W_0 \subset V_1$

$$\psi(t) = \sum_{k \in \mathbb{Z}} g_k \varphi_{1,k}(t) = \sqrt{2} \sum_{k \in \mathbb{Z}} g_k \varphi(2t - k).$$

Somit erhalten wir

$$\psi_{j,k}(x) = 2^{\frac{j}{2}} \psi(2^j x - k) = 2^{\frac{j}{2}} \sum_{n \in \mathbb{Z}} g_n \sqrt{2} \varphi(2^{j+1} x - 2k - n) = \sum_{n \in \mathbb{Z}} g_n \varphi_{j+1,2k+n}(x), \quad (2.4)$$

woraus wiederum

$$c_{j,k} = \langle f, \psi_{j,k} \rangle = \langle f, \sum_{n \in \mathbb{Z}} g_n \varphi_{j+1,2k+n} \rangle = \sum_{n \in \mathbb{Z}} \bar{g}_n \langle f, \varphi_{j+1,2k+n} \rangle = \sum_{m \in \mathbb{Z}} \bar{g}_{m-2k} \langle f, \varphi_{j+1,m} \rangle$$

folgt. Aufgrund von $f(x) = P_0 f(x) = P_{-1} f(x) + Q_{-1} f(x)$ mit

$$P_{-1} f(x) = \sum_{n \in \mathbb{Z}} \langle f, \varphi_{-1,n} \rangle \varphi_{-1,n}(x), \quad Q_{-1} f(x) = \sum_{n \in \mathbb{Z}} \langle f, \psi_{-1,n} \rangle \psi_{-1,n}(x)$$

gilt deshalb

$$a_{-1,k} := \langle f, \varphi_{-1,k} \rangle = \sum_{n \in \mathbb{Z}} \overline{h_{n-2k}} \langle f, \varphi_{0,n} \rangle = \sum_{n \in \mathbb{Z}} \overline{h_{n-2k}} a_{0,n}$$

und

$$c_{-1,k} := \langle f, \psi_{-1,k} \rangle = \sum_{n \in \mathbb{Z}} \overline{g_{n-2k}} \langle f, \varphi_{0,n} \rangle = \sum_{n \in \mathbb{Z}} \overline{g_{n-2k}} a_{0,n}.$$

Dies kann nun so weitergeführt werden: aus $P_{-1}f = P_{-2}f + Q_{-2}f$ mit

$$P_{-2}f = \sum_{n \in \mathbb{Z}} \langle f, \varphi_{-2,n} \rangle \varphi_{-2,n}, \quad Q_{-2}f(x) = \sum_{n \in \mathbb{Z}} \langle f, \psi_{-2,n} \rangle \psi_{-2,n}$$

folgt

$$a_{-2,k} := \langle f, \varphi_{-2,n} \rangle = \sum_{n \in \mathbb{Z}} \overline{h_{n-2k}} a_{-1,n}$$

und

$$c_{-2,k} := \langle f, \psi_{-2,n} \rangle = \sum_{n \in \mathbb{Z}} \overline{g_{n-2k}} a_{-1,n},$$

usw.

Algorithmus (Wavelet-Transformation)

Gegeben: Folge $\mathbf{a}^0 := (a_{0,n})_{n \in \mathbb{Z}}$, $\{h_n\}_{n \in \mathbb{Z}}$, $\{g_n\}_{n \in \mathbb{Z}}$

For $j = 1, \dots, J$

berechne $\mathbf{a}^{-j} := \{a_{-j,n}\}_{n \in \mathbb{Z}}$, $\mathbf{c}^{-j} := \{c_{-j,n}\}_{n \in \mathbb{Z}}$ aus

$$a_{-j,k} = \sum_{n \in \mathbb{Z}} \overline{h_{n-2k}} a_{-j+1,n}, \quad c_{-j,k} = \sum_{n \in \mathbb{Z}} \overline{g_{n-2k}} a_{-j+1,n}$$

end

Ausgabe:

$$f(x) = \sum_{n \in \mathbb{Z}} a_{-J,n} \varphi_{-J,n} + \sum_{l=-J}^{-1} \sum_{n \in \mathbb{Z}} c_{-l,n} \psi_{-l,n}$$

Wenn $\mathbf{a}^0 \in \mathbb{R}^N$ mit $N = 2^J$ ist und die Folgen h und g nur M_1 bzw. M_2 Nichtnullelemente besitzen ($M_1, M_2 \ll N$), so beträgt die arithmetische Komplexität dieses Algorithmus $\mathcal{O}(N)$, da es im j -ten Level nur $(M_1 + M_2) \frac{N}{2^j}$ Multiplikationen und ca. $(M_1 + M_2 - 2) \frac{N}{2^j}$ Additionen sowie höchstens J Level gibt.

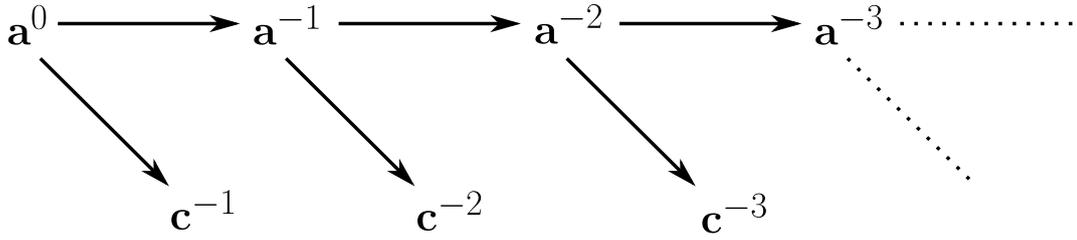


Abbildung 2.5: Skizze zur Funktionsweise des Zerlegungsalgorithmus. Hier sind $\mathbf{a}^j := \{a_{j,n}\}_{n \in \mathbb{Z}}$, $\mathbf{c}^j := \{c_{j,n}\}_{n \in \mathbb{Z}}$

Wenn man die Koeffizienten $\{a_{-J,n}\}_{n \in \mathbb{Z}}$, $\{c_{-l,n}\}_{n \in \mathbb{Z}}$, $l = J, \dots, -1$ kennt und daraus wieder die ursprünglich gegebenen Koeffizienten $\{a_{0,n}\}_{n \in \mathbb{Z}}$ berechnen will (inverse Wavelet-Transformation), so geschieht dies folgendermaßen. Wegen $P_{-j}f = P_{-j-1}f + Q_{-j-1}f$, $j = 0, \dots, J-1$, gilt

$$P_{-j}f = \sum_{k \in \mathbb{Z}} a_{-j,k} \varphi_{-j,k} = \sum_{k \in \mathbb{Z}} a_{-j-1,k} \varphi_{-j-1,k} + \sum_{k \in \mathbb{Z}} c_{-j-1,k} \psi_{-j-1,k}$$

und somit

$$a_{-j,n} = \langle P_{-j}f, \varphi_{-j,n} \rangle = \sum_{k \in \mathbb{Z}} a_{-j-1,k} \langle \varphi_{-j-1,k}, \varphi_{-j,n} \rangle + \sum_{k \in \mathbb{Z}} c_{-j-1,k} \langle \psi_{-j-1,k}, \varphi_{-j,n} \rangle.$$

Mit (2.3) folgt

$$\langle \varphi_{-j-1,k}, \varphi_{-j,n} \rangle = \left\langle \sum_{l \in \mathbb{Z}} h_{l-2k} \varphi_{-j,l}, \varphi_{-j,n} \right\rangle = h_{n-2k}$$

und mit (2.4)

$$\langle \psi_{-j-1,k}, \varphi_{-j,n} \rangle = \left\langle \sum_{l \in \mathbb{Z}} g_{l-2k} \varphi_{-j,l}, \varphi_{-j,n} \right\rangle = g_{n-2k}.$$

Also gilt

$$a_{-j,n} = \sum_{k \in \mathbb{Z}} a_{-j-1,k} h_{n-2k} + \sum_{k \in \mathbb{Z}} c_{-j-1,k} g_{n-2k}.$$

Algorithmus (inverse Wavelet-Transformation)

Gegeben: Folgen $\mathbf{a}^{-J}, \mathbf{c}^{-J}, \mathbf{c}^{-J+1}, \dots, \mathbf{c}^{-1}$

For $j = J - 1, \dots, 0$

$$\text{berechne } a_{-j,n} = \sum_{k \in \mathbb{Z}} a_{-j-1,k} h_{n-2k} + \sum_{k \in \mathbb{Z}} c_{-j-1,k} g_{n-2k}$$

Ausgabe:

$$f(x) = \sum_{n \in \mathbb{Z}} a_{0,n} \varphi_{0,n}(x)$$

Ein Beispiel für ein orthogonales Wavelet, nämlich das Haarwavelet, haben wir schon oben gesehen. Hier sind die von Null verschiedenen Koeffizienten mit $h_0 = \frac{1}{\sqrt{2}}, h_1 = \frac{1}{\sqrt{2}}, g_0 = -\frac{1}{\sqrt{2}}$ und $g_1 = \frac{1}{\sqrt{2}}$ gegeben. Ein weiteres Beispiel sind die $D4$ -Wavelets, für die $h_0 = \frac{1+\sqrt{3}}{4}, h_1 = \frac{3+\sqrt{3}}{4}, h_2 = \frac{3-\sqrt{3}}{4}, h_4 = \frac{1-\sqrt{3}}{4}$ und $h_i = 0$ für $i \notin \{1, \dots, 4\}$, bzw. $g_i = (-1)^i h_{1-i}, i \in \mathbb{Z}$ gilt.

2.4 Biorthogonale Wavelets

Es ist nicht möglich, orthogonale Wavelets zu finden, die gleichzeitig symmetrisch und glatt sind sowie einen kompakten Träger besitzen (siehe [40], Kapitel 3.4.2). Da symmetrische Fehler vom menschlichen Auge allerdings eher toleriert werden als asymmetrische (siehe [18]), wäre es wünschenswert, Wavelets zu finden, die ebenfalls symmetrisch sind. Biorthogonale Wavelets besitzen diese zusätzliche Eigenschaft.

2.4.i Definition. *Biorthogonale Wavelets* sind zwei Funktionen $\psi, \psi^* \in L^2(\mathbb{R})$, für die folgendes gilt:

- Die Menge $\{\psi_{j,k} \mid j, k \in \mathbb{Z}\}$ mit $\psi_{j,k} := \psi(2^j x - k)$ bildet eine Rieszbasis von $L^2(\mathbb{R})$;
- Die Menge $\{\psi_{j,k}^* \mid j, k \in \mathbb{Z}\}$ mit $\psi_{j,k}^* := \psi^*(2^j x - k)$ bildet eine Rieszbasis von $L^2(\mathbb{R})$;
- $\langle \psi_{j,k}, \psi_{m,n}^* \rangle = \delta_{j,m} \delta_{k,n}$ für alle $j, k, m, n \in \mathbb{Z}$.

Die biorthogonale MRA, die von Cohen, Daubechies und Feauveau in [15] vorgestellt wurde, besteht aus zwei Multiresolutionsanalysen mit Skalierungsfunktionen φ und φ^* .

Das heißt, sowohl die Zerlegung

$$\dots \subset V_2 \subset V_{-1} \subset V_0 \subset V_1 \subset V_2 \subset \dots \subset L^2$$

als auch die Zerlegung

$$\dots \subset V_2^* \subset V_{-1}^* \subset V_0^* \subset V_1^* \subset V_2^* \subset \dots \subset L^2$$

erfüllen die Voraussetzungen 2.2.i(i)-(iii), und es existieren $\varphi \in V_0$ (bzw. $\varphi \in V_0^*$), so dass $\{\varphi(t-k)\}_{k \in \mathbb{Z}}$ (bzw. $\varphi^* \in V_0^*$) eine Rieszbasis von V_0 (bzw. eine Rieszbasis von V_0^*) bilden. Die Basen sind dual, d.h. es gilt $\langle \varphi_{0,m}, \varphi_{0,k}^* \rangle = \delta_{k,m}$ und für $f \in V_0$ gilt, dass $f = \sum_{k \in \mathbb{Z}} \langle f, \varphi_{0,k}^* \rangle \varphi_{0,k}$ bzw. für $f^* \in V_0^*$ gilt $f^* = \sum_{k \in \mathbb{Z}} \langle f^*, \varphi_{0,k} \rangle \varphi_{0,k}^*$.

Aufgrund der Biorthogonalität ist V_{j+1} die direkte Summe von V_j und W_j bzw. V_{j+1}^* die direkte Summe von V_j^* und W_j^* und es gilt

$$V_j \perp W_j^* \quad \text{bzw.} \quad V_j^* \perp W_j$$

sowie $L^2(\mathbb{R}) = \overline{\bigoplus_{j \in \mathbb{Z}} W_j} = \overline{\bigoplus_{j \in \mathbb{Z}} W_j^*}$, wobei \bigoplus die direkte Summe bezeichne.

2.4.1 Schnelle biorthogonale Wavelet-Transformation

Sei $f \in V_0$ gegeben, d.h. $f = \sum_{k \in \mathbb{Z}} \langle f, \varphi_{0,k}^* \rangle \varphi_{0,k}$. Wegen $\varphi^* \in V_0^* \subset V_1^*$ bzw. $\psi^* \in W_0^* \subset V_1^*$ gilt

$$\varphi^* = \sqrt{2} \sum_{k \in \mathbb{Z}} h_k^* \varphi^*(2x - k) \quad \text{bzw.} \quad \psi^* = \sqrt{2} \sum_{k \in \mathbb{Z}} g_k^* \varphi^*(2x - k).$$

Hieraus folgt analog zu (2.3) und (2.4)

$$\psi_{j,k}^*(x) = \sum_{n \in \mathbb{Z}} g_n^* \varphi_{j+1,2k+n}^*(x) \quad \text{und} \quad \varphi_{j,k}^*(x) = \sum_{n \in \mathbb{Z}} h_n^* \varphi_{j+1,2k+n}^*(x).$$

Wir setzen $a_{0,n} := \langle f, \varphi_{0,n}^* \rangle$. Es gilt also für $j = -1, \dots, J$, wobei J eine fest vorgegebene Anzahl von Leveln ist, dass

$$a_{j,k} := \langle f, \varphi_{j,k}^* \rangle = \langle f, \sum_{n \in \mathbb{Z}} h_n^* \varphi_{j+1,2k+n}^* \rangle = \sum_{n \in \mathbb{Z}} \overline{h_{n-2k}^*} \langle f, \varphi_{j+1,n}^* \rangle = \sum_{n \in \mathbb{Z}} \overline{h_{n-2k}^*} a_{j+1,n}$$

und

$$c_{j,k} := \langle f, \psi_{j,k}^* \rangle = \langle f, \sum_{n \in \mathbb{Z}} g_n^* \varphi_{j+1,2k+n} \rangle = \sum_{n \in \mathbb{Z}} \overline{g_{n-2k}^*} \langle f, \varphi_{j+1,n} \rangle = \sum_{n \in \mathbb{Z}} \overline{g_{n-2k}^*} a_{j+1,n}.$$

Man erhält schließlich nach J Leveln die Darstellung

$$f(x) = \sum_{n \in \mathbb{Z}} a_{-J,n} \varphi_{-J,n} + \sum_{l=-J}^{-1} \sum_{n \in \mathbb{Z}} c_{-l,n} \psi_{-l,n}. \quad (2.5)$$

Die ursprüngliche Darstellung

$$f(x) = \sum_{n \in \mathbb{Z}} a_{0,n} \varphi_{0,n}$$

erhält man folgendermaßen aus der neuen Darstellung (2.5). Wir nutzen hier, dass für $j = 0, \dots, J-1$

$$P_{-j}f = P_{-j-1}f + Q_{-j-1}f = \sum_{k \in \mathbb{Z}} a_{-j-1,k} \varphi_{-j-1,k} + \sum_{k \in \mathbb{Z}} c_{-j-1,k} \psi_{-j-1,k}$$

gilt und wir somit

$$a_{-j,n} = \langle P_{-j}f, \varphi_{-j,n}^* \rangle = \sum_{k \in \mathbb{Z}} a_{-j-1,k} \langle \varphi_{-j-1,k}, \varphi_{-j,n}^* \rangle + \sum_{k \in \mathbb{Z}} c_{-j-1,k} \langle \psi_{-j-1,k}, \varphi_{-j,n}^* \rangle,$$

erhalten. Wegen

$$\langle \psi_{-j-1,k}, \varphi_{-j,n}^* \rangle = \sum_{m \in \mathbb{Z}} g_{m-2k} \langle \varphi_{-j,m}, \varphi_{-j,n}^* \rangle = g_{n-2k}$$

und

$$\langle \varphi_{-j-1,k}, \varphi_{-j,n}^* \rangle = \sum_{m \in \mathbb{Z}} h_{m-2k} \langle \varphi_{-j,m}, \varphi_{-j,n}^* \rangle = h_{n-2k}$$

folgt

$$a_{-j,n} = \sum_{k \in \mathbb{Z}} a_{-j-1,k} h_{n-2k} + \sum_{k \in \mathbb{Z}} c_{-j-1,k} g_{n-2k}.$$

2.4.2 Beispiel für biorthogonale Wavelets

Ein Beispiel für biorthogonale Wavelets sind die CDF 9/7-Wavelets von Cohen, Daubechies und Feauveau ([15]), die wir in unseren weiteren Betrachtungen verwenden werden. Die Koeffizienten sind in der Tabelle (2.1) dargestellt.

n	h_n^*	h_n
0	0,85269867900889	0,78848561640637
-1, 1	0,37740285561283	0,4180922732204
-2, 2	-0,11062440441844	-0,04068941760920
-3, 3	-0,02384946501956	-0,06453888262876
-4, 4	0,03782845554969	0

Tabelle 2.1: Die Koeffizienten h_n^* und h_n für $n = -4, \dots, 4$. Für $n \in \mathbb{Z} \setminus \{-4, \dots, 4\}$ gilt $h_n^* = h_n = 0$. Es gilt $g_n = (-1)^{1-n}h_{1-n}$ sowie $g_n^* = (-1)^{1-n}h_{1-n}^*$ für $n \in \mathbb{Z}$.

2.5 Tensorprodukt-Wavelet-Transformation

Wenn man zweidimensionale Daten, z.B. ein Bild, gegeben hat, kann man eine Tensorprodukt-Wavelet-Transformation anwenden. Diese lässt sich leicht auf die eindimensionale Wavelet-Transformation zurückführen. Man wendet zunächst eine eindimensionale Wavelet-Transformation auf die einzelnen Zeilen des Bildes an. Auf die so erhaltenen neuen Daten wird nun eine eindimensionale Wavelet-Transformation entlang der Spalten angewendet.

2.6 Weitere Begriffe

Im Folgenden stellen wir zwei Begriffe vor, die in den Unterkapiteln zu den numerischen Ergebnissen verwendet werden.

Der *PSNR* (peak-signal-to-noise-ratio), ist folgendermaßen definiert

$$\text{PSNR} := 20 \log_2 \frac{m}{\|f - \tilde{f}\|_2},$$

wobei m der größtmögliche Pixelwert des Bildes und $\|\cdot\|_2$ die euklidische Norm von $f - \tilde{f}$ im Falle von Vektoren und die Frobeniusnorm von $f - \tilde{f}$ im Falle von Matrizen ist. Dabei

bezeichnet f das Originalbild sowie \tilde{f} die Approximation von f . Der PSNR ist ein Maß um die Bildqualität der Approximation \tilde{f} mathematisch zu erfassen. Ein hoher Wert steht hier für eine gute Approximation des Originalbildes.

Die *Entropie* eines Vektors p wird mit

$$\text{Entropie} := - \sum_{j=0}^n \frac{h_j}{N_1 N_2} \log_2 \left(\frac{h_j}{l} \right)$$

berechnet, wobei l die Länge des Vektors ist, n die Anzahl der verschiedenen Werte, die in p vorkommen, und h_1, \dots, h_n bezeichnen, wie oft jeder einzelne der n Werte vorkommt. Die Entropie ist ein Maß für den mittleren Informationsgehalt des Vektors p . Man beachte, dass die Entropie die notwendigen Speicherkosten für den Vektor p weit überschätzt, da bei der Berechnung außer acht gelassen wird, dass benachbarte Komponenten des Vektors p miteinander korrelieren, d.h. dass sie viele Aneinanderreihungen von gleichen oder ähnlichen notwendigen Werten (zum Beispiel vielen aufeinanderfolgenden Nullen) besitzen.

3 Die Easy-Path-Wavelet-Transformation

In diesem Kapitel stützen wir uns auf die Ergebnisse aus [43] und beschreiben die grundlegende Konstruktionsidee der Easy-Path-Wavelet-Transformation. Dabei konzentrieren wir uns besonders auf die genaue Beschreibung der Konstruktion von Pfadvektoren, die eine effiziente Speicherung zulassen. Hier werden auch einige neue Methoden, wie Direkt-Nachbarn-EPWT, Mittelpunkt-EPWT, Halbe-EPWT und Einfach-EPWT, vorgestellt, die über die Arbeit [43] hinausgehen. Sie wurden teilweise schon in unserer Arbeit [48] erwähnt - dort wurden sie für die Approximation eines Teilbildes benutzt. Näheres zur Arbeit [48] ist im nächsten Kapitel zu lesen.

Die Idee der Easy-Path-Wavelet-Transformation (EPWT) ist es, einen zusammenhängenden Pfad durch die Pixel eines gegebenen Bildes zu finden, so dass die Pixel, die auf diesem Pfad benachbart sind, möglichst ähnliche Werte haben. Dieser Pfad soll eine Permutation der Pixel sein. Entlang dieses Pfades wird dann eine eindimensionale Wavelet-Transformation angewendet. Es lassen sich nun noch weitere Level der EPWT anwenden, die darin bestehen, die im vorigen Level erhaltenen Tiefpasswerte sinnvoll zu sortieren und auf die Tiefpasswerte in der neuen Reihenfolge eine eindimensionale Wavelet-Transformation anzuwenden.

3.1 Notation

Um das Konzept der Easy-Path-Wavelet-Transformation im Folgenden besser erklären zu können, legen wir nun folgende Notation fest. Das gegebene Bild bestehe aus $N_1 \times N_2$ Pixeln, wobei N_1 und N_2 natürliche Zahlen sind, deren Produkt ein Vielfaches einer Viererpotenz ist ($N_1 N_2 = 2^{2J} s$ mit $J, s \in \mathbb{N}$). Die digitale Funktion, die das Bild beschreibt, sei mit $f = \{f(i, j)\}_{i,j=0}^{N_1, N_2}$ bezeichnet. Die Menge $I_{2J} = \{(i, j) \mid i = 0, \dots, N_1 - 1, j = 0, \dots, N_2 - 1\}$ wird *Bildindexmenge* genannt. Im weiteren Verlauf werden wir auch oft die eindimensionale

Darstellung der Bildindexmenge I_{2J} benutzen, die durch die bijektive Funktion

$$\tilde{J} : I_{2J} \rightarrow \{0, \dots, N_1 N_2 - 1\} \quad \text{mit} \quad \tilde{J}((i, j)) := i + j N_1$$

beschrieben wird. Die inverse Abbildung dazu ist durch

$$j = \lfloor \frac{\tilde{J}((i, j))}{N_1} \rfloor, \quad i = \tilde{J}((i, j)) - j N_1$$

gegeben. Sei $\mathbf{f}^{2J} \in \mathbb{R}^{N_1 N_2}$ der Vektor, der entsteht, wenn man die Funktionswerte von f spaltenweise aneinanderhängt:

$$\mathbf{f}^{2J} = (f(0, 0), f(1, 0), \dots, f(N_1 - 1, 0), f(0, 1), f(1, 1), \dots, f(N_1 - 1, N_2 - 1))^T.$$

Eine *Nachbarschaft* eines Indizes $(i, j) \in I_{2J}$ wird durch

$$N(i, j) = \{(i_1, j_1) \in I_{2J} \setminus \{(i, j)\} \mid |i - i_1| \leq 1, |j - j_1| \leq 1\} \quad (3.1)$$

festgelegt. Somit hat jeder Index, der im Inneren liegt (d.h., $i \notin \{0, N_1 - 1\}$ und $j \notin \{0, N_2 - 1\}$) 8 Nachbarn. Jeder Index, der am Rand liegt, aber keine Ecke ist, d.h. $i \notin \{0, N_1 - 1\}$, $j \in \{0, N_2 - 1\}$ oder $i \in \{0, N_1 - 1\}$, $j \notin \{0, N_2 - 1\}$, hat 5 Nachbarn, und jede Ecke hat 3 Nachbarn. Wenn wir die eindimensionale Darstellung der Indexmenge benutzen, so ist die Nachbarschaft eines Indizes $l \in \tilde{J}(I_{2J}) = \{0, 1, \dots, N_1 N_2 - 1\}$ mit $l = \tilde{J}((i, j))$ durch $N(l) = \tilde{J}(N(i, j))$ bestimmt. Wenn der Index l also im Inneren liegt, besteht seine Nachbarschaft aus

$$N(l) = \{l + 1, l - 1, l + N_1, l - N_1, l + N_1 + 1, l + N_1 - 1, l - N_1 + 1, l - N_1 - 1\}.$$

Da wir im weiteren Verlauf auch von Nachbarschaften von disjunkten Teilmengen von $\tilde{J}(I_{2J})$ reden werden, folgt nun an dieser Stelle die Definition des Begriffs *Nachbarschaft* für disjunkte Teilmengen. Seien $I_\mu \subset \tilde{J}(I_{2J})$ und $I_\nu \subset \tilde{J}(I_{2J})$ mit $I_\mu \cap I_\nu = \emptyset$ für $\mu \neq \nu$. Wenn ein Index $l \in I_\nu$ und ein Index $l_1 \in I_\mu$ mit $l \in N(l_1)$, $\mu \neq \nu$ existieren, sagen wir, dass die Teilmengen I_μ und I_ν benachbart sind und schreiben $I_\nu \in N(I_\mu)$.

Als *zusammenhängend* bezeichnen wir einen Indexvektor (l_k, \dots, l_{k+n}) , für den $l_{\nu+1} \in N(l_\nu)$ für $\nu = k, \dots, k + n - 1$, $0 \leq k < k + n \leq N_1 N_2 - 1$ gilt. Dieser Indexvektor wird *Pfadstück* genannt. Unser Ziel ist es, einen *vollständigen Pfad* durch $\tilde{J}(I_{2J})$ zu finden, d.h., einen Vektor der Länge $N_1 N_2$ der alle Indizes von $\tilde{J}(I_{2J})$ in einer sinnvoll-

len Reihenfolge geordnet, enthält. Dieser vollständige Pfad ist also eine Permutation von $(0, 1, 2, \dots, N_1 N_2 - 1)^T$. Er kann aus mehreren Pfadstücken bestehen, d.h. $(\mathbf{p}_1^T, \mathbf{p}_2^T, \dots, \mathbf{p}_r^T)^T$, wobei \mathbf{p}_ν , $\nu = 1, \dots, r$, zusammenhängende (Spalten-)Indexvektoren sind. Der Übergang von einem Pfadstück zum Nächsten heißt *Sprung*.

3.2 Die EPWT

Nun widmen wir uns der Easy-Path-Wavelet-Transformation im Detail.

3.2.1 Erstes Level der EPWT

Es seien die Funktionswerte unseres gegebenen Bildes wie oben beschrieben in einem Vektor \mathbf{f}^{2J} der Länge 2^{2J} gespeichert. Das erste Level der EPWT besteht nun darin, einen vollständigen Pfad \mathbf{p}^{2J} durch $\tilde{J}(I_{2J})$ zu finden und auf die Funktionswerte auf diesem vollständigen Pfad eine eindimensionale Wavelet-Transformation anzuwenden. Dieser Pfad soll so beschaffen sein, dass Funktionswerte, die auf diesem Pfad benachbart sind, eine möglichst geringe Differenz aufweisen und sich die Anordnung des Pfades, d.h. die Permutation der Indizes $0, \dots, N_1 N_2 - 1$, trotzdem noch effizient speichern lässt.

3.2.2 Bestimmung des Pfades p^{2J}

Gestartet wird dieser Pfad mit dem ersten Pixel, dem Pixel mit dem Index $(0, 0)$, bzw. in eindimensionaler Darstellung mit 0 , d.h. $\mathbf{p}^{2J}(0) := 0$. Selbstverständlich könnte man auch mit jedem beliebigen anderen Pixel anfangen. Um den nächsten Index zu bestimmen, durch den unser Pfad führt, betrachten wir die absolute Differenz des Funktionswertes zum Index 0 zu den Funktionswerten der Nachbarn unseres Indizes 0 , d.h., zu den Funktionswerten der Indizes $1, N_1$ und $N_1 + 1$. Der benachbarte Index mit der kleinsten absoluten Differenz ist dann unser $\mathbf{p}^{2J}(1)$, d.h. also

$$\mathbf{p}^{2J}(1) := \operatorname{argmin}_{k \in \{1, N_1, N_1 + 1\}} |\mathbf{f}^{2J}(0) - \mathbf{f}^{2J}(k)|.$$

Der nächste Index auf unserem Pfad wird nun auf die gleiche Art und Weise bestimmt. Auch hier betrachtet man die absoluten Differenzen des zu $\mathbf{p}^{2J}(1)$ gehörigen Funktionswertes zu den Funktionswerten der Nachbarn, mit Ausnahme des Nachbarindizes 0 , der ja schon in unserem Pfadvektor vorkommt. Im Allgemeinen funktioniert die Bestimmung des n -ten

Eintrages, wenn man schon die ersten $n-1$, $n \in \{2, \dots, N_1 N_2 - 1\}$ Einträge des Pfadvektors bestimmt hat, wie folgt

$$\mathbf{p}^{2J}(n) := \underset{\substack{k \neq \mathbf{p}^{2J}(\nu), \nu=0, \dots, n-1 \\ k \in N(\mathbf{p}^{2J}(n-1))}}{\operatorname{argmin}} \left| \mathbf{f}^{2J}(\mathbf{p}^{2J}(n-1)) - \mathbf{f}^{2J}(k) \right|. \quad (3.2)$$

Da es vorkommen kann, dass das Minimum an mehreren Indizes angenommen wird, muss man im Vorhinein festlegen, welcher der in Frage kommenden Indizes als Eintrag für den Pfadvektor genommen wird. Eine Möglichkeit wäre zum Beispiel, alle Nachbarn, durch die der Pfad noch nicht gegangen ist, im Uhrzeigersinn beginnend mit dem rechten Nachbarn zu ordnen. In dem Fall, dass der Index im Inneren liegt und der Pfad noch durch keinen benachbarten Index außer $\mathbf{p}^{2J}(n-1) + 1$ geht, wären sie also in der Folge $\mathbf{p}^{2J}(n-1) + N_1, \mathbf{p}^{2J}(n-1) + N_1 + 1, \mathbf{p}^{2J}(n-1) - N_1 + 1, \mathbf{p}^{2J}(n-1) - N_1, \mathbf{p}^{2J}(n-1) - N_1 - 1, \mathbf{p}^{2J}(n-1) - 1$ geordnet. Nun nimmt man den in dieser Ordnung ersten Index, für den das Minimum angenommen wird. Würde das Minimum sowohl bei $\mathbf{p}^{2J}(n-1) + N_1 + 1$ als auch bei $\mathbf{p}^{2J}(n-1) - N_1$ angenommen, würde man $\mathbf{p}^{2J}(n) := \mathbf{p}^{2J}(n-1) + N_1 + 1$ setzen.

Falls es vorkommt, dass der bisher bestimmte Pfad schon durch alle Nachbarn von $\mathbf{p}^{2J}(n-1)$ gegangen ist und somit kein Minimierer von (3.2) existiert, so beginnt man ein neues Pfadstück. Eine einfache Möglichkeit wäre zum Beispiel, den kleinsten Index ν zu nehmen, für den $\nu \neq \mathbf{p}^{2J}(\mu)$ für $\mu = 0, \dots, n-1$ gilt. Man könnte ebenso den Index ν mit $\nu \neq \mathbf{p}^{2J}(\mu)$ für $\mu = 0, \dots, n-1$ nehmen, dessen Funktionswert die geringste absolute Differenz zu dem Funktionswert von $\mathbf{p}^{2J}(n-1)$ aufweist, oder der vom euklidischen Abstand her am Nächsten ist.

Das obige Verfahren führt man solange durch, bis man einen Pfad durch alle Indizes, d.h. einen Pfadvektor, dessen Einträge eine Permutation der Indizes $0, \dots, N_1 N_2 - 1$ sind, gefunden hat.

3.2.3 Anwendung der Wavelettransformation

Nun wendet man eine eindimensionale periodische Wavelet-Transformation auf die Funktionswerte entlang des Pfadvektors, d.h. auf $(\mathbf{f}^{2J}(\mathbf{p}^{2J}(k)))_{k=0}^{N_1 N_2 - 1}$, an. Wir erhalten nun einen Vektor $\mathbf{f}^{2J-1} \in \mathbb{R}^{N_1 N_2 / 2}$, der den Tiefpassanteil enthält, sowie die Waveletkoeffizienten $\mathbf{g}^{2J-1} \in \mathbb{R}^{N_1 N_2 / 2}$. Auf den Tiefpassanteil wird nun das zweite Level der EPWT angewandt.

Da unser Vektor \mathbf{f} von der Länge 2^{2J} ist, können wir $2J$ Level der EPWT anwenden.

3.2.4 Zweites Level der EPWT

Das zweite Level funktioniert ähnlich wie das erste Level. Hier wird ein Pfad durch die $\frac{N_1 N_2}{2}$ Tiefpasswerte in f^{2J-1} bestimmt, so dass die Tiefpasswerte so angeordnet sind, dass Tiefpasswerte mit einer geringen absoluten Differenz benachbart sind. Zugleich soll auch hier der Pfadvektor, der die Angaben über die Reihenfolge der Tiefpasswerte enthält, möglichst einfach bzw. effizient zu speichern sein. Auf die neu angeordneten Tiefpasswerte wird nun wiederum eine eindimensionale Wavelet-Transformation angewandt, so dass man wieder einen Tiefpassanteil, im Folgenden mit $\mathbf{f}^{2J-2} \in \mathbb{R}^{N_1 N_2 / 4}$ bezeichnet, und einen Hochpassanteil, bezeichnet mit $\mathbf{g}^{2J-2} \in \mathbb{R}^{N_1 N_2 / 4}$, erhält. Die Bestimmung des Pfadvektors im zweiten Level ähnelt der Bestimmung im ersten Level. Hier beginnt man den Pfad mit einer Indexmenge (nähere Einzelheiten folgen) und sucht sich als nächsten Eintrag des Pfadvektors den Index der Indexmenge unter den benachbarten Indexmengen aus, der der ähnlichste Wert zugeordnet ist. Die Indexmengen definieren wir als

$$I_l^{2J-1} := \{\mathbf{p}^{2J}(2l), \mathbf{p}^{2J}(2l+1)\}, \quad l = 0, \dots, \frac{N_1 N_2}{2} - 1.$$

Jeder dieser Indexmengen I_l^{2J-1} ordnen wir nun einen Funktionswert zu, und zwar den Tiefpasswert $\mathbf{f}^{2J-1}(l)$. Zwei Indexmengen I_m^{2J-1} und I_n^{2J-1} (bzw. deren Funktionswerten) bezeichnet man nun als benachbart, wenn

$$\exists k \in I_m^{2J-1}, q \in I_n^{2J-1} \text{ mit } k \in N(q). \quad (3.3)$$

Die Menge aller zu I_m^{2J-1} benachbarten Indexmengen wird im weiteren Verlauf mit $N(I_m^{2J-1})$ bezeichnet.

3.2.5 Bestimmung des Pfades p^{2J-1}

Gestartet wird unser Pfad \mathbf{p}^{2J-1} mit der Indexmenge I_0^{2J-1} und dem zugehörigen Funktionswert $\mathbf{f}^{2J-1}(0)$, d.h., $\mathbf{p}^{2J-1}(0) = 0$. Der nächste Eintrag unseres Pfadvektors, $\mathbf{p}^{2J-1}(1)$, wird nun wie folgt bestimmt. Wir setzen

$$\mathbf{p}^{2J-1}(1) := \operatorname{argmin}_{k \neq 0} \{|\mathbf{f}^{2J-1}(0) - \mathbf{f}^{2J-1}(k)| \mid I_k^{2J-1} \in N(I_0^{2J-1})\}.$$

Die restlichen Einträge $\mathbf{p}^{2J-1}(m)$ mit $m = 3, \dots, \frac{N_1 N_2}{2} - 1$ werden durch

$$\mathbf{p}^{2J-1}(m) := \operatorname{argmin}_{k \in U} \{ |\mathbf{f}^{2J-1}(m-1) - \mathbf{f}^{2J-1}(k)| \mid I_k^{2J-1} \in N(I_{m-1}^{2J-1}) \}$$

mit

$$U := \left\{ k \mid k \in \left\{ 0, \dots, \frac{N_1 N_2}{2} - 1 \right\}, k \neq p^{2J-1}(l), l = 0, \dots, m-1 \right\}$$

definiert.

3.2.6 Anwendung der Wavelet-Transformation

Jetzt wenden wir auf die Tiefpasswerte \mathbf{f}^{2J-1} des ersten Levels, angeordnet gemäß \mathbf{p}^{2J-1} , d.h. also, auf $\{ \mathbf{f}^{2J-1}(\mathbf{p}^{2J-1}(k)) \}_{k=0}^{\frac{N_1 N_2}{2}-1}$ wiederum eine eindimensionale periodische Wavelet-Transformation an und erhalten den Tiefpassanteil $\mathbf{f}^{2J-2} \in \mathbb{R}^{\frac{N_1 N_2}{4}}$ und den Hochpassanteil $\mathbf{g}^{2J-2} \in \mathbb{R}^{\frac{N_1 N_2}{4}}$.

3.2.7 Weitere Level der EPWT

Für $N_1 N_2 > 4$ lassen sich anschließend noch weitere Level anwenden. Die Indexmengen der weiteren Level setzen sich stets aus zwei Indexmengen des vorigen Levels zusammen. Im lev -ten Level, $lev \in \mathbb{N}$, $2J - lev \geq 0$ und $lev \geq 3$, sind die Indexmengen dementsprechend folgendermaßen

$$I_l^{2J-lev+1} := I_{\mathbf{p}^{2J-lev+2}(2l)}^{2J-lev+2} \cup I_{\mathbf{p}^{2J-lev+2}(2l+1)}^{2J-lev+2}, \quad \text{mit } l = 0, \dots, \frac{N_1 N_2}{2^{lev-1}} - 1$$

bestimmt. Jeder dieser Indexmengen $I_l^{2J-lev+1}$ wird jeweils der Wert $\mathbf{f}^{2J-lev+1}(l)$, $l = 0, \dots, \frac{N_1 N_2}{2^{lev-1}} - 1$ zugeordnet. Der Begriff der Nachbarschaft wird analog zu (3.3) definiert, d.h., zwei Indexmengen $I_m^{2J-lev+1}$ und $I_n^{2J-lev+1}$ heißen benachbart, wenn

$$\exists k \in I_m^{2J-lev+1}, q \in I_n^{2J-lev+1} \quad \text{mit } k \in N(q).$$

Der erste Eintrag des Pfadvektors im lev 'ten Level, $\mathbf{p}^{2J-lev+1}(0)$, ist der Index der Indexmenge, mit der der Pfad gestartet wird. Wenn der Pfad mit der Indexmenge $I_0^{2J-lev+1}$ begonnen wird, so ist demnach $\mathbf{p}^{2J-lev+1}(0) = 0$. Hat man die ersten m , $0 < m < \frac{N_1 N_2}{2^{lev-1}} - 1$,

Indizes bestimmt, so lässt sich der $(m + 1)$ -te Eintrag durch

$$\mathbf{p}^{2J-lev+1}(m) := \operatorname{argmin}_{k \in U} \left\{ \left| \mathbf{f}^{2J-lev+1}(m-1) - \mathbf{f}^{2J-lev+1}(k) \right| \mid I_k^{2J-lev+1} \in N(I_{m-1}^{2J-1}) \right\}$$

mit

$$U := \left\{ k \mid k \in \left\{ 0, \dots, \frac{N_1 N_2}{2^{lev-1}} - 1 \right\} \text{ und } k \neq p^{2J-lev+1}(l), l = 0, \dots, m-1 \right\}$$

berechnen. Auf die Tiefpasswerte des vorigen Levels, $\mathbf{f}^{2J-lev+1}$, angeordnet gemäß des Pfades, d.h., auf $\left\{ \mathbf{f}^{2J-lev+1}(\mathbf{p}^{2J-lev+1}(k)) \right\}_{k=0}^{\frac{N_1 N_2}{2^{lev-1}} - 1}$, wird nun eine eindimensionale periodische Wavelet-Transformation angewendet, so dass man Tiefpasswerte $\mathbf{f}^{2J-lev} \in \mathbb{R}^{\frac{N_1 N_2}{2^{lev}}}$ und Hochpasswerte $\mathbf{g}^{2J-lev} \in \mathbb{R}^{\frac{N_1 N_2}{2^{lev}}}$ erhält. Wenn $2J - lev > 0$ ist, kann man jetzt noch mindestens ein weiteres Level der Easy-Path-Wavelet-Transformation durchführen.

Man erhält also nach $2J$ Leveln der EPWT den Koeffizientenvektor

$$\mathbf{g} = \left((\mathbf{f}^0)^T, (\mathbf{g}^0)^T, (\mathbf{g}^1)^T, \dots, (\mathbf{g}^{2J-1})^T \right)^T \in \mathbf{R}^{N_1 N_2},$$

und den Pfadvektor

$$\mathbf{p} = \left((\mathbf{p}^1)^T, (\mathbf{p}^2)^T, \dots, (\mathbf{p}^{2J})^T \right)^T \in \mathbb{R}^{2N_1 N_2 - 2}$$

der alle Pfadvektoren in den verschiedenen Leveln der EPWT enthält. Anhand dieser beiden Vektoren \mathbf{g} und \mathbf{p} lässt sich das ursprüngliche Signal $f \in \mathbb{R}^{N_1 \times N_2}$ (bzw. $\mathbf{f}^{2J} \in \mathbb{R}^{N_1 N_2}$) exakt rekonstruieren.

Um von \mathbf{f}^{2J} eine dünner besetztere Darstellung als \mathbf{g} zu erhalten, setzt man die vom Betrag her kleinsten Einträge von \mathbf{g}^i , $i = 0, \dots, 2J - 1$, Null.

Die Rekonstruktion von f (bzw. \mathbf{f}^{2J}) aus \mathbf{g} und \mathbf{p} funktioniert wie folgt.

Algorithmus

For $j = 0$ **to** $2J - 1$ **do**

- Wende die zu der verwendeten eindimensionalen Wavelet-Transformation inverse Wavelet-Transformation auf $\begin{pmatrix} \mathbf{f}^j \\ \mathbf{g}^j \end{pmatrix} \in \mathbb{R}^{2^{j+1}}$ an und erhalte $\mathbf{f}_p^{j+1} \in \mathbb{R}^{2^{j+1}}$.
- Ordne die Funktionswerte gemäß der Anordnung der Indexmengen im $j + 1$ 'ten Level, d.h.

$$\mathbf{f}^{j+1}(\mathbf{p}^{j+1}(k)) := \mathbf{f}_p^{j+1}(k), \quad k = 0, \dots, 2^{j+1} - 1.$$

end do.

3.3 Eine Relaxation der EPWT

Ein Nachteil der EPWT ist, ähnlich wie bei den meisten adaptiven Approximationsalgorithmen, dass man zusätzlich zum Koeffizientenvektor noch die Anordnung der Funktionswerte in den einzelnen Levels speichern muss. Deshalb ist es wichtig, auch für eine effiziente Speicherung des Pfadvektors zu sorgen. Der Pfadvektor des ersten Levels lässt sich beispielsweise mit nur höchstens drei Bit pro Eintrag speichern, wenn man statt dem nächsten Index lediglich die Richtung speichert, die der Pfad einschlägt. Das heißt, man weist jedem der höchstens 8 noch nicht für den Pfad benutzten Nachbarpixel einen Wert von 0 bis höchstens 7 zu. Die Werte werden dem Uhrzeigersinn nach verteilt, wobei das Nachbarpixel den Wert 0 zugewiesen bekommt, das in der Richtung des bereits eingeschlagenen Pfades liegt. Die Abbildung (3.1) veranschaulicht diese Prozedur. Wir betrachten alle Nachbarn des Pixels in der Mitte, dessen Index im Pfad an ℓ -ter Stelle vorkommt. Hier wurde nur der Nachbar in der linken unteren Ecke schon für den Pfad verwendet, und zwar steht er an $(\ell - 1)$ -ter Stelle des Pfades. Der rote Pfeil symbolisiert den Pfad, der bis jetzt eingeschlagen wurde, während der blaue Pfeil in die Richtung zeigt, in die man gehen müsste, damit sich die eingeschlagene Richtung nicht ändert. Somit wird dem Pixel, auf das der blaue Pfeil zeigt, die Zahl 0 zugewiesen, dem nächsten Pixel im Uhrzeigersinn die Zahl 1. Die beiden darauffolgenden Pixel bekommen die Nummern 2 bzw. 3. Das nächste

Pixel, nämlich das in der linken unteren Ecke, wurde schon für den Pfad genommen und wird deshalb bei der Nummerierung ausgelassen, so dass das darauffolgende Nachbarpixel (das linke) die Zahl 4 zugewiesen bekommt. Die im Uhrzeigersinn folgenden Pixel werden analog aufsteigend durchnummeriert.

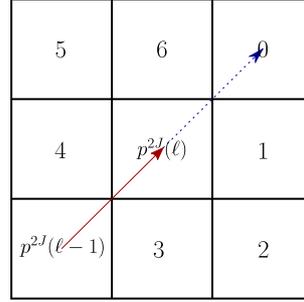


Abbildung 3.1: Illustration zur Nummerierung der Nachbarn

Wir wollen nun einen Pfad $\tilde{\mathbf{p}}^{2J}$ konstruieren, der in jeder Komponente l nur die Richtung bei Übergang von $\mathbf{p}^{2J}(l-1)$ zu $\mathbf{p}^{2J}(l)$ speichert. Dieser Pfadvektor enthält noch die vollständige Information über die Permutation der Indexmenge in \mathbf{p}^{2J} , lässt sich jedoch mit wesentlich geringeren Kosten speichern. Der erste Pfad lässt sich folgendermaßen in $\tilde{\mathbf{p}}^{2J}$ speichern. Wenn wir den Pfad \mathbf{p}^{2J} bestimmt haben, so wird der erste Eintrag des neuen Pfades gleich dem ersten Eintrag des alten Pfades gesetzt ($\tilde{\mathbf{p}}^{2J}(0) := \mathbf{p}^{2J}(0) = 0$). Um den nächste Eintrag von $\tilde{\mathbf{p}}^{2J}$ zu bestimmen, betrachtet man im Uhrzeigersinn alle Nachbarn von $\mathbf{p}^{2J}(0)$ und setzt

$$\tilde{\mathbf{p}}^{2J}(1) := \begin{cases} 0 & \text{falls } \mathbf{p}^{2J}(1) = N_1, \\ 1 & \text{falls } \mathbf{p}^{2J}(1) = N_1 + 1, \\ 2 & \text{falls } \mathbf{p}^{2J}(1) = 1. \end{cases}$$

Für die weiteren Einträge verfährt man folgendermaßen. Wenn die Richtung des Pfades, die einmal eingeschlagen wurde, beibehalten wird, das heißt, wenn

$$\mathbf{p}^{2J}(l+1) = \mathbf{p}^{2J}(l) + (\mathbf{p}^{2J}(l) - \mathbf{p}^{2J}(l-1)) = 2\mathbf{p}^{2J}(l) - \mathbf{p}^{2J}(l-1)$$

gilt, so wird $\tilde{\mathbf{p}}^{2J}(l+1) := 0$ gesetzt. Wenn sich die Richtung gegenüber der eingeschlagenen Richtung ändert, so betrachten wir den Vektor $\mathbf{q} = (\mathbf{q}(\mu))_{\mu=0}^7 \in \mathbb{Z}^8$,

$$\mathbf{q} := (N_1, N_1 + 1, 1, -N_1 + 1, -N_1, -N_1 - 1, -1, N_1 - 1)^T,$$

so dass $\mathbf{p}^{2J}(l+1) \in \{\mathbf{p}^{2J}(l) + \mathbf{q}(\mu) \mid \mu = 0, \dots, 7\}$ (vorausgesetzt, dass unter den Nachbarn von $\mathbf{p}^{2J}(l)$ mindestens einer ist, der für den Pfad noch nicht verwendet wurde). Wir suchen nun den Index $\tilde{\mu} \in \{0, \dots, 7\}$ mit

$$\mathbf{q}(\tilde{\mu}) = \mathbf{p}^{2J}(l) - \mathbf{p}^{2J}(l-1)$$

und bilden nun durch periodische Verschiebung einen neuen Vektor $\tilde{\mathbf{q}}$, der durch Aneinanderreihung der beiden Teile $(\mathbf{q}(\tilde{\mu}), \dots, \mathbf{q}(7))$ und $(\mathbf{q}(0), \dots, \mathbf{q}(\tilde{\mu}-1))$ entsteht. Man beachte, dass $\mathbf{p}^{2J}(l-1)$ ein Nachbar von $\mathbf{p}^{2J}(l)$ ist, der schon für den Pfad verwendet wurde. Also kann $\mathbf{p}^{2J}(l+1)$ nicht gleich $\mathbf{p}^{2J}(l) + \tilde{\mathbf{q}}(4)$ sein, da $\tilde{\mathbf{q}}(4) = \mathbf{p}^{2J}(l-1) - \mathbf{p}^{2J}(l)$ ist. Wenn alle übrigen sieben Nachbarn von $\mathbf{p}^{2J}(l)$ noch nicht für den Pfad verwendet wurden, so setzt man

$$\tilde{\mathbf{p}}^{2J}(l+1) := \begin{cases} 0 & \text{wenn } \mathbf{p}^{2J}(l+1) = \mathbf{p}(l) + \tilde{\mathbf{q}}(0), \\ 1 & \text{wenn } \mathbf{p}^{2J}(l+1) = \mathbf{p}(l) + \tilde{\mathbf{q}}(1), \\ 2 & \text{wenn } \mathbf{p}^{2J}(l+1) = \mathbf{p}(l) + \tilde{\mathbf{q}}(2), \\ 3 & \text{wenn } \mathbf{p}^{2J}(l+1) = \mathbf{p}(l) + \tilde{\mathbf{q}}(3), \\ 4 & \text{wenn } \mathbf{p}^{2J}(l+1) = \mathbf{p}(l) + \tilde{\mathbf{q}}(5), \\ 5 & \text{wenn } \mathbf{p}^{2J}(l+1) = \mathbf{p}(l) + \tilde{\mathbf{q}}(6), \\ 6 & \text{wenn } \mathbf{p}^{2J}(l+1) = \mathbf{p}(l) + \tilde{\mathbf{q}}(7). \end{cases}$$

Die Nachbarindizes von $\mathbf{p}^{2J}(l)$, die in $\tilde{J}(I_{2J})$ liegen und noch nicht für den Pfad verwendet wurden, werden „zulässig“ genannt. Wenn nicht alle 7 Nachbarindizes zulässig sind, so wird $\tilde{\mathbf{q}}$ geändert in dem man alle Komponenten aus $\tilde{\mathbf{q}}$ herausstreicht, die nicht zu einem zulässigen Index $\mathbf{p}^{2J}(l+1) = \mathbf{p}^{2J}(l) + \tilde{\mathbf{q}}(\mu)$ führen. Sei $\tilde{\mathbf{q}}^{sh}$ der Vektor, der nur die zulässigen Komponenten von $\tilde{\mathbf{q}}$ enthält (in der gleichen Reihenfolge). Dann setzt man

$$\tilde{\mathbf{p}}^{2J}(l+1) := \mu \text{ wenn } \mathbf{p}^{2J}(l+1) = \mathbf{p}^{2J}(l) + \tilde{\mathbf{q}}^{sh}(\mu).$$

Wenn der Pfad \mathbf{p}^{2J} einen Sprung besitzt, so hängt die Bestimmung von $\tilde{\mathbf{p}}(l+1)$ davon ab, wie der erste Index des neuen Pfadstücks bestimmt wurde. Wenn man einfach den kleinsten Index, der noch nicht für den Pfad verwendet wurde, genommen hat, so reicht es, $\tilde{\mathbf{p}}(l+1) := 0$ zu setzen.

Wenn der Pfad also lange derselben Richtung folgt, erreicht man somit eine Aneinan-

derreihung von Nullen und dementsprechend einen dünner besetzten Pfadvektor.

Eine leichte Abwandlung der EPWT nutzt diese Tatsache aus.

In dieser Version der EPWT, die wir im Folgenden auch „relaxierte EPWT“ nennen werden, wird die Richtung des Pfades möglichst lange beibehalten und nur gewechselt, wenn die Differenz der Funktionswerte größer als eine gewisse Schranke wird. Eine hohe Schranke hält die Adaptivitätskosten niedrig, da sie dafür sorgt, dass die Richtung des Pfades lange gleich bleibt und man somit nur wenige Richtungswechsel zur Rekonstruktion des Pfades speichern muss. Gleichzeitig sorgt eine hohe Schranke und somit eine seltene Änderung der Richtung des Pfades aber auch dafür, dass die Differenzen der auf dem Pfad benachbarten Werte größer sind, als wenn man eine kleinere Schranke gewählt hätte, so dass man größere Waveletkoeffizienten erhält.

Genauer gesagt, funktioniert diese relaxierte Form der EPWT wie folgt. Wir setzen als Erstes eine Schranke S fest und starten dann wieder mit einem beliebigen Pixel, zum Beispiel mit dem Pixel ganz oben links. Weil dieses Pixel auf dem Pfad noch kein Vorgängerpixel hat, da der Pfad eben erst gestartet wurde, wird eine beliebige Richtung im Vorhinein als bevorzugte Richtung festgelegt, zum Beispiel, den Pfad in die rechte Richtung einzuschlagen. Dementsprechend wird also $\tilde{\mathbf{q}}^{sh}(0) := N_1$, $\tilde{\mathbf{q}}^{sh}(1) = N_1 + 1$ und $\tilde{\mathbf{q}}^{sh}(2) = 1$ gesetzt. Nun wird gemäß dieser Reihenfolge geprüft, ob die Differenz der Funktionswerte von dem linken oberen Pixel und dem Nachbarpixel betragsmäßig kleiner als die zu Anfang festgesetzte Schranke S ist, d.h., wir prüfen

$$|f(\mathbf{p}^{2J}(0)) - f(\mathbf{p}^{2J}(0) + \tilde{\mathbf{q}}^{sh}(\mu))| \leq S. \quad (3.4)$$

Sobald dies für ein μ der Fall ist, wird das Nachbarpixel $\mathbf{p}^{2J}(0) + \tilde{\mathbf{q}}^{sh}(\mu)$ als nächstes Pixel auf dem Pfad gesetzt, d.h. $\mathbf{p}^{2J}(1) = \mathbf{p}^{2J}(0) + \tilde{\mathbf{q}}^{sh}(\mu)$, bzw. $\tilde{\mathbf{p}}^{2J}(1) = \mu$. Wenn es kein μ gibt, für das (3.4) erfüllt ist, so wird

$$\tilde{\mathbf{p}}^{2J}(1) = \operatorname{argmin}_{\mu \in \{0,1,2\}} |f(\mathbf{p}^{2J}(0)) - f(\mathbf{p}^{2J}(0) + \tilde{\mathbf{q}}^{sh}(\mu))|$$

gesetzt (bzw. $\mathbf{p}^{2J}(1) = \mathbf{p}^{2J}(0) + \tilde{\mathbf{q}}^{sh}(\mu)$). Wenn man die ersten n Einträge des Pfades \mathbf{p}^{2J} (bzw. von $\tilde{\mathbf{p}}^{2J}$) bestimmt hat, so wird der Eintrag $\tilde{\mathbf{p}}^{2J}(n)$ (bzw. $\mathbf{p}^{2J}(n)$) wie folgt bestimmt. Wir setzen $\tilde{\mathbf{p}}^{2J}(n)$ gleich dem kleinsten μ , das

$$|f(\mathbf{p}^{2J}(n-1)) - f(\mathbf{p}^{2J}(n-1) + \tilde{\mathbf{q}}^{sh}(\mu))| \leq S$$

erfüllt. Wenn es keinen Nachbarindex $\mathbf{p}^{2J}(n-1) + \tilde{\mathbf{q}}^{sh}(\mu)$ gibt, für den obige Ungleichung erfüllt ist, so bestimmt man das μ , das

$$|f(\mathbf{p}^{2J}(n-1)) - f(\mathbf{p}^{2J}(n-1) + \tilde{\mathbf{q}}^{sh}(\mu))|$$

minimiert und setzt $\tilde{\mathbf{p}}^{2J}(n) := \mu$ bzw. $\mathbf{p}^{2J}(n) := \mathbf{p}^{2J}(n-1) + \tilde{\mathbf{q}}^{sh}(\mu)$. Falls $\mathbf{p}^{2J}(n-1)$ keine Nachbarn besitzt, die bis jetzt noch nicht für den Pfad verwendet wurden, so wird $\mathbf{p}^{2J}(n)$ gleich dem kleinsten Index gesetzt, der bis jetzt noch nicht für die Pfadeinträge $\mathbf{p}^{2J}(0), \dots, \mathbf{p}^{2J}(n-1)$ verwendet wurde. In diesem Fall muss man keinen Eintrag in $\tilde{\mathbf{p}}^{2J}$ speichern.

Offenbar entspricht diese relaxierte Form der EPWT für $S = 0$ der am Anfang vorgestellten „strikten“ EPWT. Die folgende Abbildung zeigt zum Einen den ersten Pfad für $S = 0$ (links) und zum Anderen den ersten Pfad für $S = 36$ (rechts). Die Kreise markieren jeweils den Anfang eines neuen Pfadstücks. Man sieht, dass eine Schranke von $S = 35$ durchaus noch dafür sorgt, dass große Unterschiede in den Grauwerten entlang des Pfades vermieden werden. Außerdem enthält der Pfad des zweiten Bildes weniger Sprünge. Für $S = 0$ enthält der erste Pfad $\tilde{\mathbf{p}}^{2J}$ 120 Nullen, 57 Einsen, 42 Zweien, 18 Dreien, 10 Vieren, 4 Fünfen und 5 Sechsen - die Entropie liegt bei 2,08 Bits pro Pixel. Der erste Pfad $\tilde{\mathbf{p}}^{2J}$ des zweiten Bildes (also für $S = 35$) enthält 241 Nullen, 10 Einsen, 2 Zweien und 3 Dreien - die Entropie liegt hier bei 0,39 Bits pro Pixel.

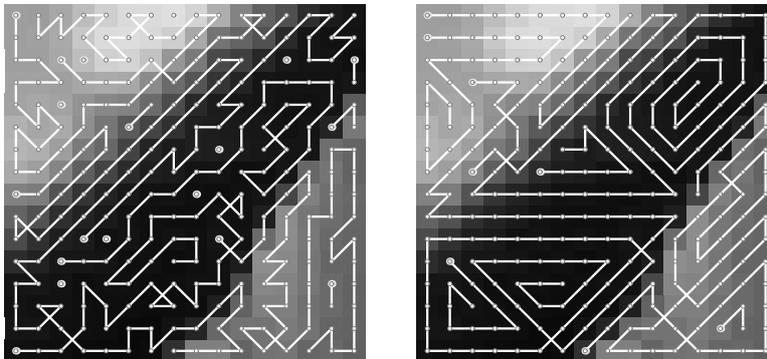


Abbildung 3.2: (a) Strikte EPWT ($S = 0$), (b) relaxierte EPWT ($S = 36$) .

3.4 Reduzierung der Pfadkosten in weiteren Leveln

Bis jetzt haben wir nur beschrieben, wie sich die Pfadkosten des ersten Pfades reduzieren lassen. Wenn man für die EPWT die maximale Anzahl von Leveln, die möglich sind, durchführt, und in jedem Level einen Pfad bestimmt, so ergeben diese Pfade der weiteren Level zusammen einen Vektor, der so lang ist, wie der Vektor des ersten Pfades. Trotz der Tatsache, dass wir somit in der Lage sind, die Hälfte der Pfadkosten durch eine leicht geänderte Bestimmung des ersten Pfades zu verringern, ist es also wichtig, sich auch für die andere Hälfte der Pfadkosten, d.h. für die Pfade der weiteren Level, zu überlegen, wie die Bestimmung des Pfades leicht abgewandelt werden um eine effizientere Speicherung zu ermöglichen.

3.4.1 Erste Variante

Eine Möglichkeit wäre, im $j + 1$ -ten Level folgendermaßen zu verfahren.

Man starte den Pfad mit einer beliebigen Indexmenge, zum Beispiel mit der ersten, d.h. $\mathbf{p}^{2^{J-j}}(0) = \tilde{\mathbf{p}}^{2^{J-j}}(0) = 0$. Wenn man den Pfad bis zu einem Index l bestimmt hat, so verfährt man für die Bestimmung von $\mathbf{p}^{2^{J-j}}(l + 1)$ folgendermaßen. Man schreibt die Indizes aller noch nicht für den Pfad verwendeten Nachbarindexmengen in einen Vektor \mathbf{n} , wobei man die Indizes $\mathbf{p}^{2^{J-j}}(l) + 1$ und $\mathbf{p}^{2^{J-j}}(l) - 1$, die zu Nachbarindexmengen gehören, von denen man vermutet, dass ihr zugehöriger Funktionswert dem von der Indexmenge mit dem Index $\mathbf{p}^{2^{J-j}}(l)$ sehr ähnlich ist, an die beiden ersten Stellen des Vektors \mathbf{n} schreibt und die Indizes der restlichen Nachbarindexmengen aufsteigend der Größe nach geordnet an die folgenden Stellen von \mathbf{n} (selbstverständlich nur dann, wenn die jeweilige Nachbarindexmenge noch nicht für den Pfad verwendet wurde). Nun wird der Reihe nach für jeden Eintrag von \mathbf{n} geprüft, ob die zugehörige Indexmenge einen Funktionswert besitzt, der sich von dem Funktionswert der Indexmenge mit dem Index $\mathbf{p}^{2^{J-j}}(l)$ nur um höchstens S unterscheidet. Der erste Eintrag $\mathbf{n}(i)$ aus \mathbf{n} , der dies erfüllt, wird als nächster Eintrag des Pfadvektors genommen, d.h. $\mathbf{p}^{2^{J-j}}(l + 1) := \mathbf{n}(i)$ - falls kein Eintrag aus \mathbf{n} dies erfüllt, so nimmt man den Eintrag, so dass die Differenz am kleinsten ist. In $\tilde{\mathbf{p}}$ speichert man nun nicht den Index der Indexmenge, sondern seine Position in \mathbf{n} , d.h. $\tilde{\mathbf{p}}^{2^{J-j}}(l + 1) = i$, was dazu führt, dass die Zahlen, die man in $\tilde{\mathbf{p}}^{2^{J-j}}$ speichert, kleiner und im Gegensatz zu $\mathbf{p}^{2^{J-j}}$ öfters die gleichen sind, wodurch dieser Pfadvektor effizienter zu speichern ist. Der folgende Algorithmus beschreibt die Funktionsweise dieser Methode noch ein wenig genauer.

Algorithmus

(i) Setze $\mathbf{p}^{2^{J-j}}(0) := 0$ und $\tilde{\mathbf{p}}^{2^{J-j}}(0) := 0$

(ii) For $l = 0, \dots, \frac{N_1 N_2}{2^j} - 2$ do

- Falls die Menge M aller Indizes aus $\{0, \dots, \frac{N_1 N_2}{2^j} - 1\}$, die zu Nachbarindexmengen von der Indexmenge mit dem Index $\mathbf{p}^{2^{J-j}}(l)$ gehören und noch nicht für den Pfad $\mathbf{p}^{2^{J-j}}$ verwendet wurden, nicht leer ist

dann:

- Schreibe, falls $\mathbf{p}^{2^{J-j}}(l) + 1 \in M$, $\mathbf{p}^{2^{J-j}}(l) + 1$ in einen Vektor \mathbf{n} .
- Falls $\mathbf{p}^{2^{J-j}}(l) - 1 \in M$, so schreibe $\mathbf{p}^{2^{J-j}}(l) - 1$ an die nächste Stelle des Vektors \mathbf{n} .
- Schreibe die restlichen Nachbarindizes aus M aufsteigend nach der Größe geordnet an die folgenden Stellen des Vektors \mathbf{n} .
- Nehme die Komponente von $\mathbf{n}(\mu^*)$ von $\mathbf{n} = (\mathbf{n}(0), \mathbf{n}(1), \dots)^T$ mit dem kleinsten Index μ^* so dass

$$|f^{2^{J-j}}(\mathbf{p}^{2^{J-j}}(l)) - f^{2^{J-j}}(\tilde{\mathbf{n}}(\mu^*))| \leq S \quad (3.5)$$

erfüllt ist und setze $\mathbf{p}^{2^{J-j}}(l+1) := \mathbf{n}(\mu^*)$, $\tilde{\mathbf{p}}^{2^{J-j}}(l+1) := \mu^*$.

Wenn keine Komponente aus \mathbf{n} die Ungleichung (3.5) erfüllt, so nehme

$$\mu^* := \operatorname{argmin}_{\mu} \{|f^{2^{J-j}}(\mathbf{p}^{2^{J-j}}(l)) - f^{2^{J-j}}(\mathbf{n}(\mu))|\}$$

und setze $\mathbf{p}^{2^{J-j}}(l+1) := \mathbf{n}(\mu^*)$, $\tilde{\mathbf{p}}^{2^{J-j}}(l+1) := \mu^*$;

- **Sonst:** Wähle $\mathbf{p}^{2^{J-j}}(l+1)$ gemäß einer im Vorhinein festgelegten Strategie, zum Beispiel gemäß der Strategie, für $\mathbf{p}^{2^{J-j}}(l+1)$ immer den kleinsten Index aus $\{0, \dots, \frac{N_1 N_2}{2^j} - 2\}$ zu wählen, der noch nicht für den Pfad verwendet wurde. Wenn nach dieser Strategie verfahren wird, muss für $\tilde{\mathbf{p}}^{2^{J-j}}$ kein Eintrag gespeichert werden, um die Rekonstruktion zu gewährleisten. Hat man beispielsweise die Strategie festgelegt, dass man 7 gleichmäßig über das Bild verteilte Indizes betrachtet und aus der Menge \tilde{M} dieser Indizes den Index m_i als Nachfolger nimmt,

für den

$$m_i := \operatorname{argmin}_{m \in \tilde{M}} \{|f^{2J-j}(\mathbf{p}^{2J-j}(l)) - f^{2J-j}(m)|\}$$

gilt, also $\mathbf{p}^{2J-j}(l+1) = m_i$, so muss man natürlich, damit man später wieder rekonstruieren kann, in $\tilde{\mathbf{p}}^{2J-j}$ abspeichern, welchen dieser 7 Indizes man genommen hat. Dies kann in ähnlicher Weise wie oben geschehen, nämlich derart, dass man, wenn man den kleinsten dieser 7 Indizes genommen hat, $\tilde{\mathbf{p}}^{2J-j}(l+1) := 0$ speichert, wenn man den zweitkleinsten nimmt, $\tilde{\mathbf{p}}^{2J-j}(l+1) := 1$ speichert, usw.

Zur Rekonstruktion des Pfades im $j+1$ -ten Level wird nur $\tilde{\mathbf{p}}^{2J-j}$ benötigt, \mathbf{p}^{2J-j} kann nach Durchführung des obigen Algorithmus gelöscht werden. Für $S = 0$ erhalten wir auch in den folgenden Leveln die strikte EPWT. Der Pfad $\tilde{\mathbf{p}}^{2J-j}$ besteht dann nicht mehr zwangsläufig nur aus den Zahlen 0 bis 7 sondern kann auch aus größeren Zahlen bestehen, da es in den weiteren Leveln passieren kann, dass man mehr als 8 Nachbarn hat.

3.4.2 Direkte-Nachbarn-EPWT

Eine andere Möglichkeit in weiteren Leveln den nächsten Pfadeintrag zu wählen wäre, Nachbarindexmengen zu bevorzugen, die einen größeren „gemeinsamen Rand“ mit der betrachteten Indexmenge besitzen. Dazu ordnet man die Nachbarindexmengen einer Indexmenge $I_{\mathbf{p}^{2J-j}(l)}^{2J-j}$ für Level $j+1$ mit $j \geq 0$ folgendermaßen. Man zählt für jede Nachbarindexmenge I_n^{2J-j} , durch die der Pfad noch nicht geführt wurde, die Anzahl a_n der Elemente $k \in I_n^{2J-j}$, für die ein $q \in I_{\mathbf{p}^{2J-j}(l)}^{2J-j}$ existiert, so dass $k \in N(q)$. Anschließend ordnet man diese Nachbarindexmengen gemäß ihrer jeweiligen Anzahl a_n , so dass in dieser Ordnung eine Nachbarindexmenge I_n^{2J-j} vor einer anderen Nachbarindexmenge $I_{\tilde{n}}^{2J-j}$ steht, wenn $a_n > a_{\tilde{n}}$. Für den Fall, dass $a_n = a_{\tilde{n}}$ ist, so ordnet man I_n^{2J-j} vor $I_{\tilde{n}}^{2J-j}$ wenn $n < \tilde{n}$ und $I_{\tilde{n}}^{2J-j}$ vor I_n^{2J-j} wenn $\tilde{n} < n$. Die nächste Indexmenge des Pfades wird nun die erste Nachbarindexmenge gemäß der vorher bestimmten Ordnung, der ein Wert zugewiesen wurde, der sich von dem Wert $f^{2J-j}(\mathbf{p}^{2J-j}(l))$, der zu der Indexmenge $I_{\mathbf{p}^{2J-j}(l)}^{2J-j}$ korrespondiert, betragsmäßig nur um eine vorher festgelegte Größe S unterscheidet. Sei diese Indexmenge die Indexmenge $I_{n^*}^{2J-j}$, die die r -te Indexmenge gemäß der vorher bestimmten Ordnung sei, so speichert man im Pfadvektor nun $\mathbf{p}^{2J-j}(l+1) := n^*$ bzw. $\tilde{\mathbf{p}}^{2J-j}(l+1) := r-1$.

Wenn alle noch nicht für den Pfad verwendeten Nachbarindexmengen von $I_{\mathbf{p}^{2J-j}(l)}^{2J-j}$ einen Wert besitzen, der sich von $f^{2J-j}(\mathbf{p}^{2J-j}(l))$ um mehr als S unterscheidet, so wird unter den noch nicht für den Pfad verwendeten Nachbarindexmengen die mit dem ähnlichsten Wert ausgewählt. Wenn mehrere solcher Indexmengen I_n^{2J-j} bestehen, so wird diejenige genommen, für die das n am kleinsten ist.

Beispiel Das folgende Beispiel veranschaulicht diese Prozedur. Wir betrachten ein Bild der Größe 8×8 . Nach Anwendung von drei Leveln der EPWT sucht man im vierten Level einen Pfad durch 8 Indexmengen, von denen jede aus 8 Indizes besteht. Die Abbildung (3.3) stellt diese Indexmengen dar. Es seien $I_0^3, I_1^3, I_2^3, I_3^3, I_4^3, I_5^3, I_6^3$ bzw. I_7^3 die Werte $f^3(0) = 20, f^3(1) = 4, f^3(2) = 4, f^3(3) = 25, f^3(4) = 65, f^3(5) = 30, f^3(6) = 60$ bzw. $f^3(7) = 63$ zugeordnet. Man fängt den Pfad zum Beispiel mit der ersten Indexmenge I_0^3 an und betrachtet jetzt die Nachbarindexmengen I_1^3, I_3^3, I_4^3 und I_5^3 . Die Nachbarindexmenge I_1^3 beinhaltet 4 Indizes, die Nachbarn von Indizes aus I_0^3 sind, in I_3^3 sind zwei Nachbarn, in I_4^3 ein Nachbar und in I_5^3 fünf Nachbarn. Also werden die Nachbarindexmengen in der Reihenfolge $I_5^3, I_1^3, I_3^3, I_4^3$ geordnet. Nun wird der Wert $f^3(0)$, der zu der Indexmenge I_0^3 korrespondiert, mit dem Wert $f^3(5)$ der Indexmenge I_5^3 verglichen. Sofern die absolute Differenz kleiner oder gleich einer vorgegebenen Schranke (z.B. $S = 20$) ist, d.h.

$$|f^3(0) - f^3(5)| \leq 20,$$

so wird $\mathbf{p}^3(1) := 5$ bzw. $\tilde{\mathbf{p}}^3(1) = 0$ gesetzt. Nun werden die Nachbarindexmengen von I_5^3 betrachtet - mit Ausnahme von I_0^3 , durch die der Pfad schon führt. Das heißt, wir zählen für I_3^3, I_4^3 und I_6^3 wieviele Elemente sie jeweils besitzen, die direkte Nachbarn zu Elementen aus I_5^3 sind. Für I_3^3 sind es ein, für I_4^3 ein und für I_6^3 fünf Elemente, also ordnet man entsprechend I_6^3, I_3^3, I_4^3 . Nun testet man, ob für die absolute Differenz

$$|f^3(5) - f^3(6)| \leq 20$$

gilt. Wegen $f^3(5) = 30$ und $f^3(6) = 60$ ist dies nicht erfüllt, so dass man nun

$$|f^3(5) - f^3(3)| = |30 - 25| \leq 20$$

testet. Dies ist erfüllt, weswegen man nun $\mathbf{p}^3(2) := 3$ bzw. $\tilde{\mathbf{p}}^3(2) := 1$ setzt. Die noch nicht für den Pfad verwendeten Nachbarindexmengen von I_3^3 sind I_1^3, I_2^3, I_4^3 und I_7^3 . Die Ordnung lautet hier $I_2^3, I_1^3, I_4^3, I_7^3$, die zugeordneten Werte sind $f^3(2) = 4, f^3(1) = 4, f^3(4) = 65$

sowie $f^3(7) = 63$. Da sich jeder dieser Werte von $f^3(3) = 25$ um mehr als 20 unterscheidet, wird die Indexmenge aus $\{I_1^3, I_2^3, I_4^3, I_7^3\}$ gewählt, deren zugeordneter Wert betragsmäßig am nächsten an $f^3(3)$ ist. Es gilt

$$|f^3(1) - f^3(3)| = |f^3(2) - f^3(3)| \leq |f^3(7) - f^3(3)| \leq |f^3(4) - f^3(3)|,$$

so dass sowohl I_1^3 und I_2^3 in Frage gekommen. Aufgrund von $1 < 2$ setzen wir $\mathbf{p}^3(3) := 1$. Gemäß der vorher bestimmten Ordnung ist I_1^3 die erste Nachbarindexmenge, weshalb $\tilde{\mathbf{p}}^3(3) := 0$ gesetzt wird. Der restliche Pfad wird analog bestimmt.

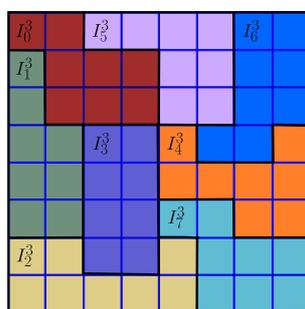


Abbildung 3.3: Beispiel zur Direkte-Nachbarn-EPWT.

3.4.3 Mittelpunkt-EPWT

Die „Mittelpunkt-EPWT“ versucht ebenfalls, die Pfadkosten in den höheren Leveln zu verringern. Hier werden die Mittelpunkte der Indexmengen berechnet und die Nachbarindexmengen nach der euklidischen Entfernung ihrer Mittelpunkte zu dem Mittelpunkt der Indexmenge, für die gerade ein Nachfolger gesucht wird, geordnet. Man beginnt die Ordnung mit der Nachbarindexmenge, die den kleinsten euklidischen Abstand hat. Die Berechnung der Mittelpunkte erfolgt hier durch die Bildung des arithmetischen Mittels über alle zweidimensionalen Indizes, die in der jeweiligen Indexmenge erhalten sind. Als nächste Indexmenge auf dem Pfad wird nun nicht zwangsläufig die Nachbarindexmenge genommen, die die Differenz der ihnen zugeordneten Werte minimiert, sondern die erste Nachbarindexmenge, so dass die Differenz der Werte kleiner als eine gewisse Schranke ist. Erst, wenn dies für keine der Nachbarindexmengen der Fall ist, wird die Nachbarindexmenge mit der kleinsten Differenz genommen. Falls gar keine Nachbarindexmengen, die noch nicht für den Pfad verwendet wurden, existieren, muss wie bei der strikten EPWT ein neues Pfadstück gestartet werden.

Beispiel Die Funktionsweise wird nun anhand des auch schon im vorigen Abschnitt verwendeten Beispielbildes (3.3) erklärt. Die zweidimensionalen Indizes, die hier zu Indexmenge I_0^3 gehören, sind $(0, 0)$, $(0, 1)$, $(1, 1)$, $(1, 2)$, $(1, 3)$, $(2, 1)$, $(2, 2)$, $(2, 3)$. Der Mittelpunkt liegt dementsprechend bei $(\frac{0+0+1+1+1+2+2+2}{8} = \frac{9}{8}, \frac{0+1+1+2+3+1+2+3}{8} = \frac{13}{8})$. Die Nachbarindexmenge I_1^3 besteht aus den Indizes $(1, 0)$, $(2, 0)$, $(3, 0)$, $(3, 1)$, $(4, 0)$, $(4, 1)$, $(5, 0)$ und $(5, 1)$. Somit liegt der Mittelpunkt bei $(\frac{27}{8}, \frac{3}{8})$. Der Mittelpunkt von I_3^3 , die aus den Indizes $(3, 2)$, $(3, 3)$, $(4, 2)$, $(4, 3)$, $(5, 2)$, $(5, 3)$, $(6, 2)$ und $(6, 3)$ besteht, liegt bei $(\frac{36}{8}, \frac{20}{8})$ und der Mittelpunkt von I_2^3 bei $(\frac{53}{8}, \frac{15}{8})$, der von I_4^3 bei $(\frac{32}{8}, \frac{46}{8})$, von I_5^3 bei $(\frac{6}{8}, \frac{32}{8})$, von I_6^3 bei $(\frac{12}{8}, \frac{50}{8})$ und von I_7^3 bei $(\frac{49}{8}, \frac{45}{8})$. Somit beträgt der euklidische Abstand vom Mittelpunkt von I_0^3 zu dem von I_1^3 $\sqrt{\frac{(27-9)^2}{64} + \frac{(13-3)^2}{64}} = \frac{\sqrt{424}}{8}$, der zum Mittelpunkt von I_3^3 beträgt $\frac{\sqrt{778}}{8}$, zu dem von I_4^3 beträgt der euklidische Abstand $\frac{\sqrt{1618}}{8}$, der Mittelpunkt von I_5^3 ist $\frac{\sqrt{370}}{8}$ entfernt. Somit ist die Ordnung der Nachbarindexmengen durch $I_5^3, I_1^3, I_3^3, I_4^3$ gegeben. Wenn wir hier die Schranke ebenfalls $S = 20$ setzen, so ist $\mathbf{p}^3(1) := 5$, bzw. $\tilde{\mathbf{p}}^3(1) := 0$. Die Indexmenge I_5^3 hat nur drei Nachbarindexmengen, die noch nicht für den Pfad verwendet wurden, nämlich I_3^3, I_4^3 und I_6^3 . Wegen

$$\frac{\sqrt{(6-12)^2 + (32-50)^2}}{8} < \frac{\sqrt{(6-32)^2 + (32-46)^2}}{8} < \frac{\sqrt{(6-36)^2 + (32-20)^2}}{8}$$

werden die Indexmengen I_6^3, I_4^3, I_3^3 geordnet. Da allerdings sowohl die Differenz $|30 - 60|$ als auch die Differenz $|30 - 65|$ größer als unsere Schranke 20 ist, wird I_3^3 gewählt, da $|30 - 25| < 20$. Somit ist $\mathbf{p}^3(2) := 3$ bzw. $\tilde{\mathbf{p}}^3(2) = 2$. Die restlichen Pfadeinträge werden analog bestimmt.

3.4.4 Einfach-EPWT und Halbe EPWT

Eine weitere Möglichkeit, in höheren Leveln Pfadkosten zu sparen, ist natürlich, gar nicht so viele Pfade zu berechnen. So könnte man zum Beispiel nur den ersten Pfad berechnen und dann mehrere Level der eindimensionalen Wavelet-Transformation entlang dieses ersten Pfades anwenden. Diese Methode wird im Folgenden „Einfach-EPWT“ genannt. Die „Halbe EPWT“ besteht darin, nur in jedem zweiten Level einen neuen Pfad zu bestimmen.

3.5 Numerische Ergebnisse

In diesem Unterkapitel werden wir uns den Ergebnissen der einzelnen im vorigen Abschnitt vorgestellten Verfahren widmen. Es wurden stets Bilder der Größe $256 \times 256 = 65536$ verwendet und 11 Level der EPWT angewandt. Zuerst stellen wir die Resultate vor, die die strikte EPWT liefert. Es wird also in der Regel bei der Bestimmung des nächsten Pfadeintrags $\mathbf{p}(n)$ der Nachbar als Nächster genommen, dessen Funktionswert dem von $\mathbf{p}(n-1)$ am ähnlichsten ist. Als Testbilder wurden CLOCK, LENA und ein Ausschnitt von SAILS verwendet. Nach Anwendung der EPWT wurden die betragsmäßig kleinen Waveletkoeffizienten zu Null gesetzt und dann aus den verbliebenen Waveletkoeffizienten eine Approximation des Originalbildes berechnet. Im Fall des CLOCK-Bildes wurden die 1000 relevantesten Koeffizienten behalten, im Fall des LENA-Bildes die betragsmäßig größten 700 Koeffizienten und beim SAILS-Bild sogar nur 200 von 65536 Koeffizienten.

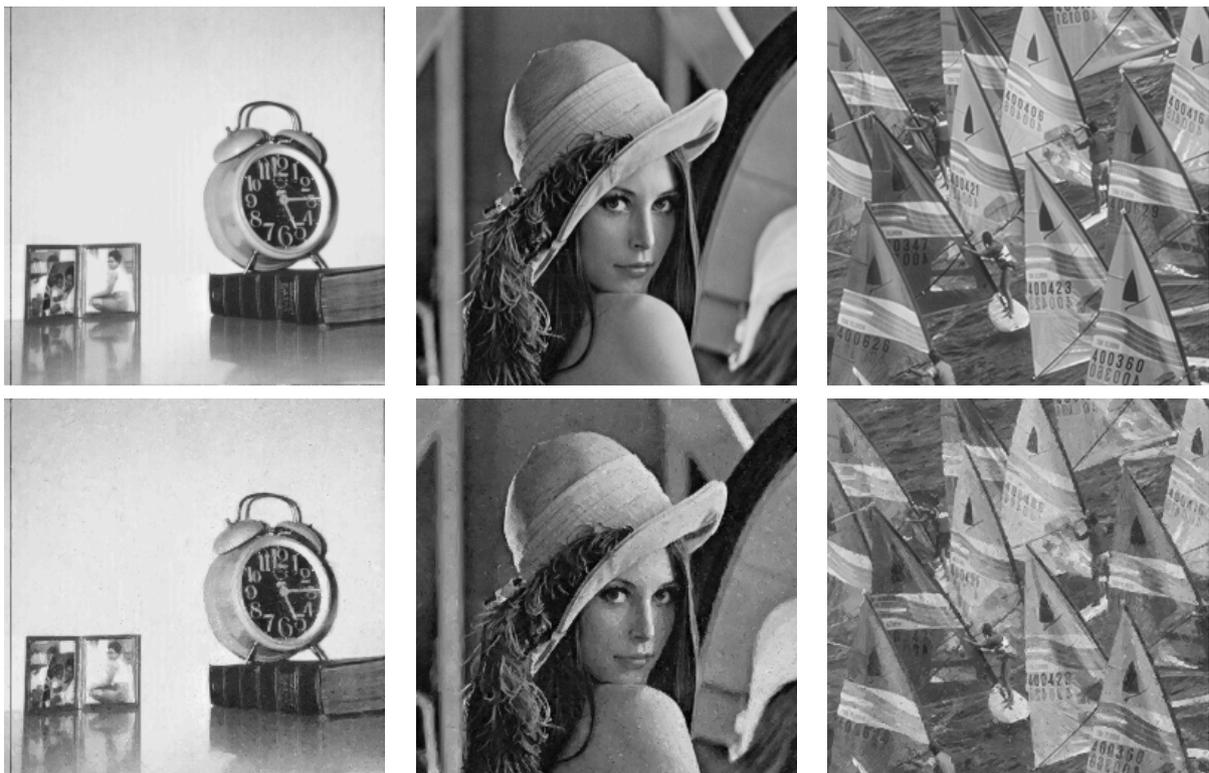


Abbildung 3.4: Erste Zeile: Originale der Bilder CLOCK, LENA und SAILS.

Zweite Zeile: Approximation von CLOCK mit 1000 Koeffizienten, von LENA mit 700 und von SAILS mit 200 Koeffizienten.

Der PSNR-Wert der Approximation des CLOCK-Bildes liegt bei 33,25 dB, die Entropie,

die versucht, eine obere Schranke für die Pfadkosten zu bestimmen, liegt bei 5,14 *bpp*, wovon die Kosten für die Speicherung des ersten Pfades 2,05 *bpp* betragen. Die Approximation des LENA-Bildes hat eine PSNR von 28,32 *dB* und der zu speichernde Pfad eine Entropie von 5,28 *bpp*, wovon 2,21 *bpp* für den ersten Pfad benötigt werden. Die Ergebnisse für die Approximation des SAILS-Bildes sind eine PSNR von 27,31 *dB* und eine Entropie von 5,39 *bpp*, wovon die Entropie für den ersten Pfad 2,31 *bpp* beträgt. Man erkennt, dass es in der Tat notwendig ist, Möglichkeiten zur Reduzierung der Pfadkosten zu finden. Als Nächstes werden darum die Ergebnisse der relaxierten EPWT vorgestellt. Hier wurden auch 1000 Koeffizienten für das CLOCK-, 700 Koeffizienten für das LENA- und 200 Koeffizienten für das SAILS-Bild gespeichert. Für die Bestimmung des ersten Pfades wurde die relaxierte EPWT mit einer Schranke von 5, 13, 30 bzw. 60 angewendet. Es wurde also versucht, die Richtung des Pfades möglichst lange beizubehalten, solange die Differenz der Funktionswerte kleiner als 5, 13, 30 bzw. 60 ist. Die resultierenden Bilder für die Schranken 5, 13 und 60 sind in der Abb. (3.5) zu sehen, die PSNR- und Entropieergebnisse in der Tabelle (3.1).

Bild	Nnu	Schranke	PSNR	Entropie	Ent. 1.Pfad
CLOCK	1000	0	33,25dB	5,14bpp	2,05bpp
CLOCK	1000	5	33,18dB	4,19bpp	0,87bpp
CLOCK	1000	13	33,26dB	3,83bpp	0,43bpp
CLOCK	1000	30	33,11dB	3,75bpp	0,23bpp
CLOCK	1000	60	32,30dB	3,60bpp	0,10bpp
LENA	700	0	30,49dB	5,28bpp	2,21bpp
LENA	700	5	30,39dB	4,63bpp	1,35bpp
LENA	700	13	30,32dB	4,13bpp	0,70bpp
LENA	700	30	29,95dB	3,87bpp	0,32bpp
LENA	700	60	29,13dB	3,70bpp	0,10bpp
SAILS	200	0	27,31dB	5,39bpp	2,31bpp
SAILS	200	5	27,25dB	4,99bpp	1,81bpp
SAILS	200	13	27,22dB	4,43bpp	1,13bpp
SAILS	200	30	26,78dB	4,03bpp	0,49bpp
SAILS	200	60	26,02dB	3,76bpp	0,14bpp

Tabelle 3.1: Vergleich der relaxierten EPWT bei verschiedenen Schranken bei der Bestimmung des ersten Pfades. Nnu bezeichne die Anzahl der Koeffizienten, die nicht zu Null gesetzt werden.



Abbildung 3.5: Approximation von CLOCK mit 1000 Koeffizienten, von LENA mit 700 und von SAILS mit 200 Koeffizienten. Linke Spalte: Schranke= 5, mittlere Spalte: Schranke= 13, rechte Spalte: Schranke= 60.

Durch eine kleine Änderung der Schranke kann die Entropie des ersten Pfades schon wesentlich verringert werden ohne zu großen Verlusten bei der Bildqualität zu führen.

Die Tabelle (3.2) zeigt für das Bild SAILS, wie häufig die einzelnen Ziffern 0 bis 7 im ersten Pfadvektor vorkommen, wenn man als Schranke im ersten Pfad 0, 30 bzw. 60 wählt und wieviele Sprünge der erste Pfad hat. Man sieht, dass die einmal eingeschlagene Richtung

	Schranke 0	Schranke 30	Schranke 60
Ziffer 0	28238	60988	64527
Ziffer 1	12888	2334	662
Ziffer 2	8871	1168	220
Ziffer 3	6691	576	93
Ziffer 4	4424	249	23
Ziffer 5	2556	133	7
Ziffer 6	1853	87	4
Ziffer 7	15	1	0
Sprünge	4581	2040	817

Tabelle 3.2: Anzahl der Ziffern 0 bis 7 und der Sprünge im ersten Pfadvektor $\tilde{\mathbf{p}}^8$ bei den Schranken 0, 30, 60.

des Pfades mit zunehmender Schranke immer länger beibehalten wird. So folgt man zum Beispiel bei der Schranke 60 insgesamt 64527 Mal der eingeschlagenen Richtung. Es fällt ebenfalls auf, dass selbst dann, wenn man die Schranke gleich 0 wählt, die Richtung schon 28238 Mal dieselbe bleibt. Dies zeigt, dass die Speicherung des Pfadvektors derart, dass man nur die eingeschlagenen Richtungen und nicht die Indizes speichert, zu einer enormen Verminderung der Pfadkosten führt.

Allerdings sind die Pfadkosten in den höheren Leveln noch zu reduzieren. Die Tabelle (3.3) zeigt die Ergebnisse für die oben als erstes vorgestellte Methode aus [43]. Der Pfad des ersten Levels wurde gemäß des Algorithmus zur relaxierten EPWT mit der Schranke 13 bestimmt, in den weiteren 10 Leveln der EPWT wurde als Schranke 5, 13, 30 und 60 gewählt. Außerdem wurde dreimal eine auf das Level angepasste Schranke gewählt, nämlich $5 \times Level$, $\frac{60}{Level}$ und $\frac{200}{Level}$. So wären zum Beispiel die Schranken im zweiten Level 10, 20 bzw. 100 und im dritten Level 15, 20 bzw. $66\frac{2}{3}$. Dadurch können Ersparungen von ein bis zwei Bits per Pixel erreicht werden, wobei dann aber auch der PSNR-Wert um ein bis zwei dB sinkt. Zum Vergleich sind in den letzten sechs Zeilen von Tabelle (3.3) die Ergebnisse dargestellt, die man erhält, wenn man fünf Level der CDF-9/7-Wavelet-Transformation anwendet. Man sieht, dass die relaxierte EPWT Ergebnisse liefert, die für die gleiche Anzahl von Nichtnullelementen um 4 bis 8 dB besser sind. Allerdings muss man natürlich noch berücksichtigen, dass bei der relaxierten EPWT noch Pfadkosten hinzukommen, die man bei der CDF-9/7-Wavelet-Transformation nicht hat.

Bild	Nnu	Schranke	PSNR	Entropie	Ent. 1.Pfad
CLOCK	1000	5	33,55dB	2,59bpp	0,43bpp
CLOCK	1000	13	33,51dB	1,98bpp	0,43bpp
CLOCK	1000	30	32,37dB	1,61bpp	0,43bpp
CLOCK	1000	60	31,39dB	1,28bpp	0,43bpp
CLOCK	1000	$5 \times Level$	33,49dB	1,93bpp	0,43bpp
CLOCK	1000	$\frac{60}{Level}$	33,53dB	1,80bpp	0,43bpp
CLOCK	1000	$\frac{200}{Level}$	30,82dB	1,31bpp	0,43bpp
LENA	700	5	30,46dB	3,63bpp	0,70bpp
LENA	700	13	30,46dB	2,91bpp	0,70bpp
LENA	700	30	30,10dB	2,27bpp	0,70bpp
LENA	700	60	28,46dB	1,78bpp	0,70bpp
LENA	700	$5 \times Level$	30,49dB	2,85bpp	0,70bpp
LENA	700	$\frac{60}{Level}$	30,23dB	2,52bpp	0,70bpp
LENA	700	$\frac{200}{Level}$	27,88dB	1,79bpp	0,70bpp
SAILS	200	5	27,19dB	4,27bpp	1,13bpp
SAILS	200	13	27,18dB	3,83bpp	1,13bpp
SAILS	200	30	26,95dB	3,14bpp	1,13bpp
SAILS	200	60	25,63dB	2,47bpp	1,13bpp
SAILS	200	$5 \times Level$	27,19dB	3,78bpp	1,13bpp
SAILS	200	$\frac{60}{Level}$	26,88dB	3,39bpp	1,13bpp
SAILS	200	$\frac{200}{Level}$	25,09dB	2,49bpp	1,13bpp
CDF 9/7					
CLOCK	1000		29,93dB		
CLOCK	3000		32,48dB		
LENA	700		24,28dB		
LENA	3000		29,95dB		
SAILS	200		19,58dB		
SAILS	3000		25,86dB		

Tabelle 3.3: Vergleich der relaxierten EPWT bei verschiedenen Schranken für die Berechnung der Pfadvektoren in den weiteren Leveln. Nnu bezeichne die Anzahl der Koeffizienten, die nicht zu Null gesetzt werden. In den letzten 6 Zeilen sind die Ergebnisse zu sehen, die man nach Anwendung von fünf Leveln der CDF-9/7-Wavelet-Transformation erhält.

Die Tabelle (3.4) zeigt die Ergebnisse der Mittelpunkt-EPWT. Auch hier wurde im ersten Level als Schranke 13 gewählt. In den höheren Leveln wurden verschiedene Schranken gewählt. Die Ergebnisse für die Schranken 5 und 60 sind nicht nur in der Tabelle nachzulesen, sondern auch in der Abbildung 3.6 (in der zweiten und dritten Zeile) dargestellt. Zum Vergleich sind in der ersten Zeile die Ergebnisse, die die Anwendung von 5 Leveln der CDF-9/7-Wavelet-Transformation bei der gleichen Anzahl an Nichtnullelementen (1000 bzw. 700 bzw. 200) liefert, dargestellt.

Bild	Nnu	Schranke	PSNR	Entropie	Ent. 1.Pfad
CLOCK	1000	5	33,29dB	2,49bpp	0,43bpp
CLOCK	1000	13	32,99dB	1,95bpp	0,43bpp
CLOCK	1000	30	32,37dB	1,61bpp	0,43bpp
CLOCK	1000	60	30,61dB	1,35bpp	0,43bpp
CLOCK	1000	$5 \times Level$	32,94dB	1,92bpp	0,43bpp
CLOCK	1000	$\frac{60}{Level}$ Level	32,76dB	1,78bpp	0,43bpp
CLOCK	1000	$\frac{200}{Level}$ Level	30,26dB	1,35bpp	0,43bpp
LENA	700	5	30,24dB	3,42bpp	0,70bpp
LENA	700	13	30,16dB	2,79bpp	0,70bpp
LENA	700	30	29,89dB	2,16bpp	0,70bpp
LENA	700	60	27,82dB	1,73bpp	0,70bpp
LENA	700	$5 \times Level$	29,99dB	2,74bpp	0,70bpp
LENA	700	$\frac{60}{Level}$	29,54dB	2,42bpp	0,70bpp
LENA	700	$\frac{200}{Level}$	27,69dB	1,79bpp	0,70bpp
SAILS	200	5	27,05dB	4,13bpp	1,13bpp
SAILS	200	13	26,86dB	3,63bpp	1,13bpp
SAILS	200	30	26,50dB	2,97bpp	1,13bpp
SAILS	200	60	25,35dB	2,38bpp	1,13bpp
SAILS	200	$5 \times Level$	26,82dB	3,60bpp	1,13bpp
SAILS	200	$\frac{60}{Level}$	26,70dB	3,22bpp	1,13bpp
SAILS	200	$\frac{200}{Level}$	24,81dB	2,40bpp	1,13bpp

Tabelle 3.4: Vergleich der relaxierten EPWT bei verschiedenen Schranken für die Berechnung der Pfadvektoren in den weiteren Leveln mit Mittelpunkt-EPWT. Nnu bezeichne die Anzahl der Koeffizienten, die nicht zu Null gesetzt werden.

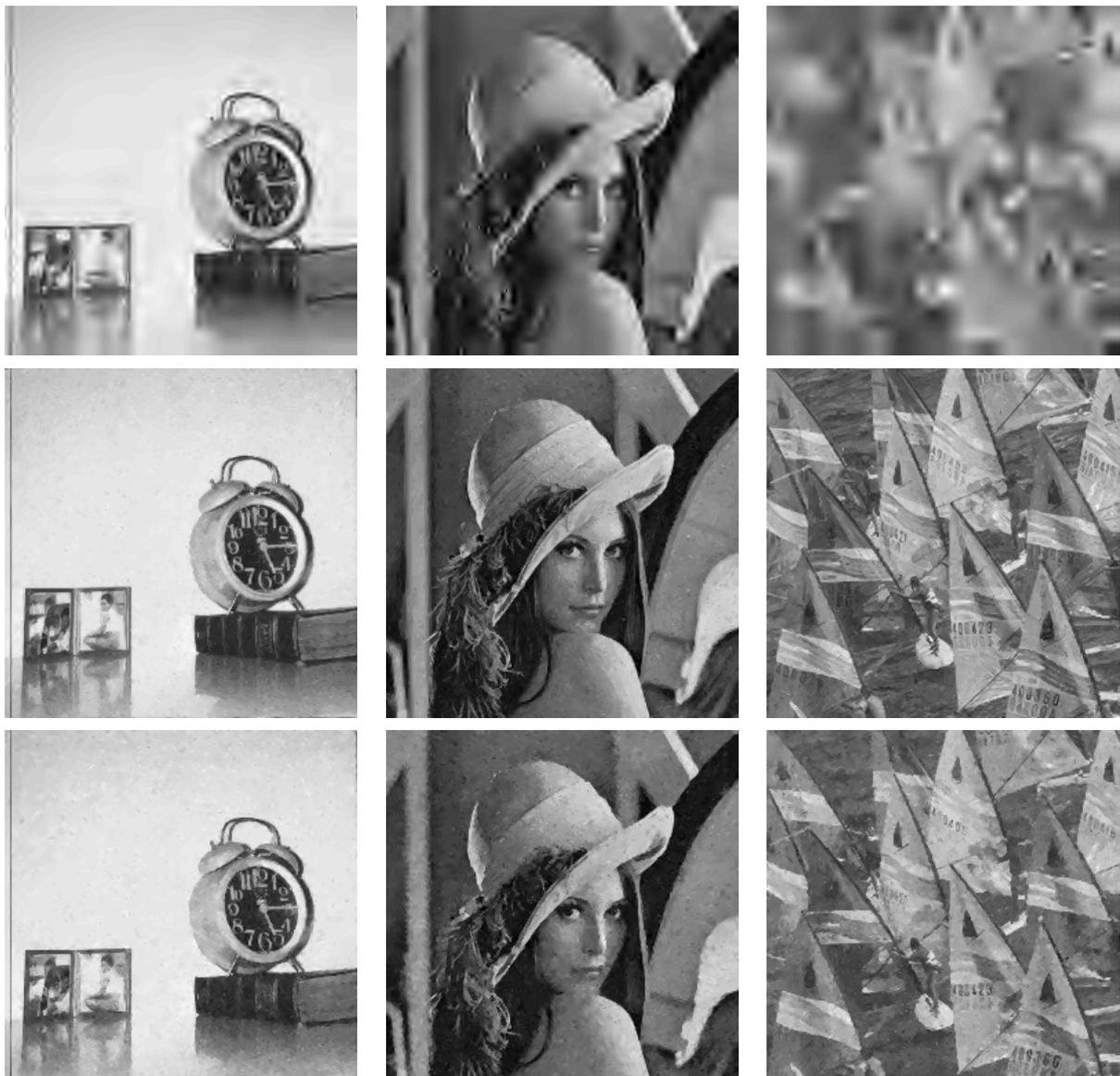


Abbildung 3.6: Approximation von CLOCK mit 1000 Koeffizienten, von LENA mit 700 und von SAILS mit 200 Koeffizienten. Erste Zeile: CDF-9/7, zweite Zeile: Mittelpunkt-EPWT mit Schranke 5 in höheren Leveln, dritte Zeile: Mittelpunkt-EPWT mit Schranke 60 in höheren Leveln.

In der Tabelle (3.5) sind die Ergebnisse der Einfach-EPWT zu sehen, die man für die Bilder CLOCK, LENA und SAILS erreicht, wenn für die Berechnung des Pfades die Schranken 0, 5, 13, 30 bzw. 60 gewählt werden.

Bild	Nnu	Schranke	PSNR	Entropie
CLOCK	1000	0	26,56dB	2,05bpp
CLOCK	1000	5	27,40dB	0,88bpp
CLOCK	1000	13	28,37dB	0,43bpp
CLOCK	1000	30	27,20dB	0,23bpp
CLOCK	1000	60	25,85dB	0,10bpp
LENA	700	0	22,28dB	2,21bpp
LENA	700	5	22,63dB	1,35bpp
LENA	700	13	23,04dB	0,70bpp
LENA	700	30	21,25dB	0,33bpp
LENA	700	60	19,43dB	0,10bpp
SAILS	200	0	21,23dB	2,31bpp
SAILS	200	5	18,36dB	1,81bpp
SAILS	200	13	18,43dB	1,13bpp
SAILS	200	30	18,48dB	0,49bpp
SAILS	200	60	18,02dB	0,14bpp

Tabelle 3.5: Vergleich der relaxierten EPWT bei verschiedenen Schranken für die Berechnung der Pfadvektoren in den weiteren Leveln mit Einfach-EPWT. Nnu bezeichne die Anzahl der Koeffizienten, die nicht zu Null gesetzt werden.

3.6 Ausblick

Wir haben in den letzten Abschnitten die Funktionsweise der EPWT für zweidimensionale Daten erläutert und anhand von Bildern veranschaulicht. Allerdings lässt sich die EPWT in etwas verallgemeinerter Form auch auf höherdimensionale Daten anwenden. Diese Idee ist eng mit der von Ram et al. [50] vorgestellten Methode verbunden. Ram et al. konstruieren aus vorgegebenen, möglicherweise höherdimensionalen Punkten $\{\gamma_j\}_{j=1}^N$, $N = 2^{L-1}$, $L \in \mathbb{N}$, mit $\gamma_j \in \mathbb{R}^M$, einen vollständigen binären Baum. Dieser Baum hat als Blätter die Datenpunkte $\{\mathbf{c}_i^\ell\}_{i=1}^N := \{\gamma_i\}_{i=1}^N$ und wird sukzessive Ebene für Ebene bis zur Wurzel c_1^1 aufgebaut. Dies geschieht, indem man versucht, die Beziehungen zwischen den einzelnen Knoten \mathbf{c}_j^ℓ , $j = 1, \dots, \frac{N}{2^{L-\ell}}$, $\ell = 1, \dots, L$ zu berücksichtigen, beispielsweise den euklidischen Abstand $w(\mathbf{c}_i^\ell, \mathbf{c}_j^\ell) := \|\mathbf{c}_i^\ell - \mathbf{c}_j^\ell\|_2$, $i, j \in \{1, \dots, \frac{N}{2^{L-\ell}}\}$ zwischen den einzelnen Knoten. Die schon bestimmten Knoten einer Ebene des Baumes werden derart sortiert, dass in dieser Sortierung aufeinanderfolgende Knoten \mathbf{c}_i^ℓ und \mathbf{c}_j^ℓ einen möglichst kleinen Wert $w(\mathbf{c}_i^\ell, \mathbf{c}_j^\ell)$ (also z.B. einen möglichst kleinen euklidischen Abstand) haben. Man speichert die Sortierungsreihenfolge und ordnet je zweien in dieser Sortierung aufeinanderfolgenden Knoten \mathbf{c}_i^ℓ und \mathbf{c}_j^ℓ denselben Vaterknoten $\mathbf{c}_k^{\ell-1}$ zu, der für den Fall, dass als Wavelet-Transformation

die Haar-Wavelet-Transformation verwendet wird, gleich $\mathbf{c}_k^{\ell-1} := \frac{\mathbf{c}_i^\ell + \mathbf{c}_j^\ell}{\sqrt{2}}$ gesetzt wird. Wenn man alle Ebenen des Baumes bis zur Wurzel bestimmt hat, wird auf die zu den $\{\gamma_j\}_{j=1}^N$ gehörenden Funktionswerte $\{f(\gamma_j)\}_{j=1}^N$, $N = 2^{L-1}$, $f : \{\gamma_1, \dots, \gamma_N\} \rightarrow \mathbb{R}$ in der vorher bestimmten Sortierungsreihenfolge eine diskrete Wavelet-Transformation (z.B. die diskrete Haar-Wavelet-Transformation) angewendet. Die resultierenden Tiefpasskoeffizienten werden den entsprechenden Vaterknoten zugewiesen. Nun wird ein weiteres Level einer Wavelet-Transformation auf die Tiefpasskoeffizienten, sortiert gemäß der zuvor bestimmten Reihenfolge der Vaterknoten, angewendet, usw. Man hofft, dass Datenpunkte, die einen kleinen euklidischen Abstand haben, ähnliche Funktionswerte besitzen und man deshalb durch die neue Sortierung erreicht, dass ähnliche Funktionswerte aufeinander folgen und man somit nach der Anwendung der Wavelet-Transformation viele kleine Waveletkoeffizienten erhält.

Die Anwendung der EPWT auf höherdimensionale Daten kann auf ähnliche Art und Weise geschehen. Sei $\Gamma := \{\gamma_1, \dots, \gamma_N\}$ eine diskrete endliche Menge von Punkten, die in \mathbb{R}^d , $d > 2$, enthalten sind. Die Funktionswerte an den einzelnen Punkten seien mit $\{f(\mathbf{x}_1), \dots, f(\mathbf{x}_n)\}$ bezeichnet. Die Idee besteht nun, wie auch schon oben beschrieben, darin, einen Pfad durch diese höherdimensionalen Punkte zu bestimmen, so dass auf diesem Pfad benachbarte Punkte ähnliche Funktionswerte besitzen. Am Ende wendet man eine eindimensionale Wavelet-Transformation auf die Funktionswerte, die gemäß des Pfades geordnet sind, an. Man startet den Pfad z.B. mit γ_1 und wählt nun als nächsten Punkt auf dem Pfad einen Punkt γ_i aus der Nachbarschaft, der von allen Punkten aus der Nachbarschaft den zu $f(\gamma_1)$ ähnlichsten Funktionswert hat. Nun sucht man in der Nachbarschaft von diesem Punkt (γ_1 ausgenommen) den Punkt mit dem ähnlichsten Funktionswert zu $f(\gamma_i)$, usw. Man sieht, dass die Pfadbestimmung im Prinzip genauso wie im Zweidimensionalen abläuft. Die Frage ist nur, wie hier Nachbarschaften definiert werden. Dies kann zum Beispiel geschehen, in dem man alle Punkte aus Γ , die in einer ε -Kugel um γ_i liegen, als Nachbarn von γ_i definiert. Das ε muss hier natürlich geeignet gewählt sein.

In weiteren Leveln kann man ähnlich wie bei Ram, Elad und Cohen [50] jeweils zwei auf dem Pfad des vorigen Levels benachbarte Punkte durch einen neuen Punkt approximieren, der auf der Mitte der Strecke zwischen den beiden Punkten liegt. Diesem Punkt wird nun der entsprechende Mittelwert zugeordnet. So würden im zweiten Level γ_1 und γ_i keine Indexmenge bilden (wie bei der EPWT), sondern durch $\mathbf{c}_1^{L-1} := \frac{\gamma_1 + \gamma_i}{2}$ repräsentiert werden, dem der Mittelwert zugewiesen wird, der sonst der Indexmenge zugewiesen wurde. Die Bestimmung des Pfades durch die neu festgelegten Punkte mit den dazugehörigen Mit-

telwerten wird nun analog zum ersten Level durchgeführt. Logischerweise muss man nun allerdings die Größe der Kugel, die die Nachbarschaft definiert, anpassen, da mit jedem Level weniger Punkte vorhanden sind und somit unter Umständen die Distanz zu dem nächstgelegenen Punkt steigt.

Da man hier für den ersten Pfad nicht mehr nur 8 Richtungen hat, die der Pfad einschlagen kann, sondern unter Umständen ziemlich viele, lässt sich die relaxierte EPWT so nicht auf den höherdimensionalen Fall übertragen. Es ist somit überaus wichtig, sich Gedanken über eine andere Vereinfachung der Speicherung des ersten Pfades bzw. eine andere Wahl der Nachbarschaften zu machen.

4 Hybrid-Algorithmen

Die Hybridmethode, die in diesem Kapitel vorgestellt wird, basiert auf der im vorigen Kapitel eingeführten Easy-Path-Wavelet-Transformation. Die erhaltenen Resultate sind in der Publikation [48] zusammengefasst. Unser Ziel ist es, eine weitere Verringerung der Adaptivitätskosten gegenüber den Adaptivitätskosten der leicht abgewandelten Version der EPWT bei gleich guter Approximation des Originals zu erreichen. Hierbei soll ausgenutzt werden, dass sich die EPWT gut für die Approximation von Kanten und Texturen eignet, während $2d$ -Tensorprodukt-Wavelet-Transformationen wie z.B. die 9/7-Transformation von Cohen, Daubechies und Feauveau optimale N -Term-Approximationsresultate für glatte Gebiete liefert, siehe [38]. Die Idee ist nun, grob gesagt, das Bild in zwei Teilbilder zu zerlegen und zwar in ein Teilbild mit hoher Regularität und in ein Teilbild, das die Kanten und Texturen enthält. Das glatte Teilbild wird dann mittels einer gut geeigneten $2d$ -Tensorprodukt-Wavelet-Transformation approximiert, während die Kanten und Texturen mit Hilfe der EPWT approximiert werden. In dem folgenden Abschnitt wird diese Zerlegung detaillierter erklärt.

4.1 1. Schritt: Zerlegung des digitalen Bildes

Zuerst glätten wir unser gegebenes digitales Bild $u = (u(i, j))_{i=1, j=1}^{N_1, N_2}$. Eine Variante ist zum Beispiel, einen Glättungsfilter anzuwenden, der auf linearer Diffusion beruht, und zwar

$$u^{k+1}(i, j) = u^k(i, j) + \tau (u^k(i+1, j) + u^k(i-1, j) + u^k(i, j-1) + u^k(i, j+1) - 4u^k(i, j))$$

für $i = 1, \dots, N_1$, $j = 1, \dots, N_2$ und mit $u^0 := u$ sowie Neumann-Randbedingungen. Der Parameter τ ist für die Stärke der Glättung in jedem Schritt verantwortlich. Nach K Iterationen, wobei K passend abhängig vom gegebenen Bild gewählt sei, erhält man so ein geglättetes Bild $u^{sm} := (u^K(i, j))_{i=1, j=1}^{N_1, N_2}$. Auf das Differenzbild $d := u^0 - u^{sm}$ wird nun

eine harte Thresholdfunktion S_θ mit

$$S_\theta x := \begin{cases} x & \text{für } |x| \geq \theta \\ 0 & \text{für } |x| < \theta \end{cases}$$

angewandt, wobei θ so gewählt wird, dass $\tilde{d} = \left(\tilde{d}(i, j) \right)_{i=1, j=1}^{N_1, N_2}$ mit $\tilde{d}(i, j) := S_\theta d(i, j)$ exakt $2^{2\zeta} < N_1 N_2$ Bildwerte besitzt ($\zeta \in \mathbb{N}$), die nicht Null sind.

Berechne nun die leicht veränderte geglättete Version des Originalbildes, nämlich $\tilde{u}^{sm} := u - \tilde{d}$.

4.2 2. Schritt: Approximation des geglätteten Bildes

Auf das geglättete Bild \tilde{u}^{sm} wird nun eine Tensorprodukt-Wavelet-Transformation angewandt. Dies ist sinnvoll, wie im Folgenden gezeigt wird.

Wir betrachten die zweidimensionalen Wavelets

$$\psi^1(x_1, x_2) = \phi(x_1)\psi(x_2), \quad \psi^2(x_1, x_2) = \psi(x_1)\phi(x_2), \quad \psi^3(x_1, x_2) = \psi(x_1)\psi(x_2)$$

und die dualen Wavelets

$$\tilde{\psi}^1(x_1, x_2) = \tilde{\phi}(x_1)\tilde{\psi}(x_2), \quad \tilde{\psi}^2(x_1, x_2) = \tilde{\psi}(x_1)\tilde{\phi}(x_2), \quad \tilde{\psi}^3(x_1, x_2) = \tilde{\psi}(x_1)\tilde{\psi}(x_2),$$

die durch die dualen Paare ϕ, ψ und $\tilde{\phi}, \tilde{\psi}$ eindimensionaler Skalierungsfunktionen und Waveletfunktionen erzeugt werden. Dann sind $\{\psi_{j,n}^1, \psi_{j,n}^2, \psi_{j,n}^3\}_{(j,n) \in \mathbb{Z}^3}$ und $\{\tilde{\psi}_{j,n}^1, \tilde{\psi}_{j,n}^2, \tilde{\psi}_{j,n}^3\}_{(j,n) \in \mathbb{Z}^3}$ nach [38] biorthogonale Rieszbasen des $L^2(\mathbb{R}^2)$.

Es wurde in [38] gezeigt, dass Tensorprodukt-Waveletbasen optimal für die dünnbesetzte Darstellung von glatten Bildern sind. Genauer gesagt, gilt für eine in einem Gebiet Ω zum Exponenten α hölderstetige Funktion f und die Waveletapproximation f_M , die unter Verwendung einer genügend glatten Waveletbasis und eines harten Shrinkage (d.h., nur die betragsmäßig größten M Waveletkoeffizienten werden behalten) entstanden ist, dass

$$\|f - f_M\|_2^2 < CM^{-\alpha},$$

wobei $\|\cdot\|_2$ die L^2 -Norm bezeichnet und der Exponent $-\alpha$ optimal ist.

Somit ist die Tensorprodukt-Wavelet-Transformation besonders gut zur Approximation von glatten Bildern geeignet.

Nach Anwendung der Tensorprodukt-Wavelet-Transformation auf unser geglättetes Bild werden nur die betragsmäßig größten M Waveletkoeffizienten behalten, die anderen werden Null gesetzt. Wir nennen die resultierende dünn besetzte Darstellung v_M . Das Bild, das man nach der Rekonstruktion unter Verwendung der betragsmäßig größten M Waveletkoeffizienten erhält, wird nun für den weiteren Verlauf mit \tilde{u}_M^{sm} bezeichnet.

4.3 3. Schritt: Approximation der Kanten und Texturen

Zuerst berechnen wir das Differenzbild $u^r = u - \tilde{u}_M^{sm}$, das die Kanten und Texturen sowie die kleinen Differenzen vom Originalbild zur Approximation des geglätteten Bildes beinhaltet. Unser Ziel ist es nun, die Kanten und Texturen vom Originalbild, die bei \tilde{u}_M^{sm} verloren gegangen sind, ebenfalls zu approximieren, so dass beide Approximationen, die des geglätteten Bildes und die der Kanten und Texturen, zusammen eine gute Approximation des Originalbildes ergeben. Wir wenden nun auf das Differenzbild u^r eine harte Thresholdfunktion S_θ an, so dass, wie im ersten Schritt, nur die betragsmäßig größten $2^{2\zeta}$ Werte, die sich an den Stellen befinden, an denen in u die Kanten und Texturen sind, nicht Null gesetzt werden. In den numerischen Experimenten haben wir 2ζ meist so gewählt, dass $2^{2\zeta}$ genau ein Viertel der Gesamtbildwerte N_1N_2 ist. Auf die nicht Null gesetzten Werte (und nur diese) wird nun die Easy-Path-Wavelet-Transformation angewandt. Wir wenden die EPWT somit auf die $2^{2\zeta}$ Bildwerte von $(u^r(i, j))_{i,j=1}^{N_1, N_2}$ an, für die $(i, j) \in I_{2\zeta}$ mit

$$I_{2\zeta} := \{(i, j) \mid 1 \leq i \leq N_1, 1 \leq j \leq N_2, |u^r(i, j)| \geq \theta\}.$$

Der Teil des Bildes, auf den wir die EPWT anwenden, sei hier mit \tilde{u}^r bezeichnet.

Die Easy-Path-Wavelet-Transformation auf diesem Teilbild funktioniert ähnlich wie die Easy-Path-Wavelet-Transformation auf einem vollständigen Bild, die schon im vorigen Kapitel beschrieben wurde. Auch hier wird ein Pfad gesucht, so dass benachbarte Werte auf dem Pfad ähnliche Werte besitzen, und dann wird auf die auf diese Weise sinnvoll geordneten Werte eine eindimensionale Wavelet-Transformation angewandt.

Der Begriff der Nachbarschaft $N(i, j)$ des Pixels $(i, j) \in I_{2\zeta}$ ist hier analog zum vorigen

Kapitel definiert, d.h., auch hier gilt

$$N(i, j) := \{i_1, j_1 \in \{(i, j) \mid 1 \leq i_1 \leq N_1, 1 \leq j_1 \leq N_2\} \mid |i - i_1| \leq 1, |j - j_1| \leq 1\}.$$

Wir starten den Pfad mit einem beliebigen Pixel $(i, j) \in I_{2\zeta}$, zum Beispiel mit dem Pixel aus $I_{2\zeta}$, so dass $i + N_1(j - 1)$ minimal ist, und setzen $p^{2\zeta}(1) := (i, j)$. Sind schon n Komponenten des Pfades $p^{2\zeta} = (p^{2\zeta}(n))_{n=1}^{2\zeta}$ durch das Teilbild bestimmt, so wird der auf $p^{2\zeta}(n) = (\tilde{i}, \tilde{j})$ folgende Eintrag des Pfadvektors folgendermaßen bestimmt. Es werden alle Nachbarn des Pixels (\tilde{i}, \tilde{j}) betrachtet, die auch im Teilbild liegen, aber durch die der Pfad noch nicht geht, genauer gesagt betrachten wir die Menge

$$\tilde{N}(p^{2\zeta}(n)) := (N(p^{2\zeta}(n)) \cap I_{2\zeta}) \setminus \{p^{2\zeta}(1), \dots, p^{2\zeta}(n)\}.$$

Sofern $p^{2\zeta}(n)$ in der Nachbarschaft von $p^{2\zeta}(n - 1)$ lag, wird nun versucht, die einmal eingeschlagene Richtung des Pfades beizubehalten. Das heißt, wir ändern die Richtung des Pfades nur in dem Fall, dass das Pixel $q \in N$, das so liegt, dass die Richtung beibehalten würde, eine zu große Differenz zu dem Wert von $p^{2\zeta}(n)$ aufweist oder in dem Fall, dass dieses Pixel nicht in $\tilde{N}(p^{2\zeta}(n))$ liegt. Mit einer „zu großen Differenz“ bezeichnen wir hier eine absolute Differenz, die größer als eine gewisse im Vorhinein festgelegte Schranke $\tilde{\theta}$ ist. Wenn einer der beiden Fälle vorliegt und wir somit die Richtung des Pfades ändern müssen, betrachten wir im Uhrzeigersinn ab q die Pixel aus $\tilde{N}(p^{2\zeta}(n))$. Das erste Pixel, das einen Wert besitzt, der betragsmäßig höchstens $\tilde{\theta}$ von dem Wert von $p^{2\zeta}(n)$ entfernt ist, wird nun als nächstes Pixel auf dem Pfad, $p^{2\zeta}(n + 1)$, genommen. Wenn alle Differenzen größer als $\tilde{\theta}$ sind, so wird das Pixel genommen, für das die Differenz minimal wird.

In dem Fall, dass die Menge $\tilde{N}(p^{2\zeta}(n))$ leer ist, muss ein neues Pfadstück angefangen werden. Eine Möglichkeit wäre, einen Index aus $I_{2\zeta} \setminus \{p^{2\zeta}(1), \dots, p^{2\zeta}(n)\}$ zu nehmen, der einen möglichst kleinen euklidischen Abstand zu $p^{2\zeta}(n)$ aufweist. Auf diese Weise wird ein Pfad durch alle Pixel aus $I_{2\zeta}$ bestimmt, der jedes Pixel genau einmal enthält. Die Abbildung 4.1 dient zur Illustration dieses Schritts. Das erste Bild zeigt das Originalbild. Im mittleren Bild ist das geglättete Bild zu sehen. Das letzte Bild zeigt das Differenzbild, wobei die Pixel, die nicht zu unserem zu approximierenden Teilbild gehören, grau dargestellt sind, während die hellen Pixel für positive Differenzen im Teilbild und die dunklen Pixel für negative Differenzen im Teilbild stehen.

Auf die Pixel des Teilbildes des Differenzbildes wird nun, angeordnet in der Reihenfolge des Pfades, eine eindimensionale Wavelet-Transformation wie z.B. die 9/7-Wavelet-

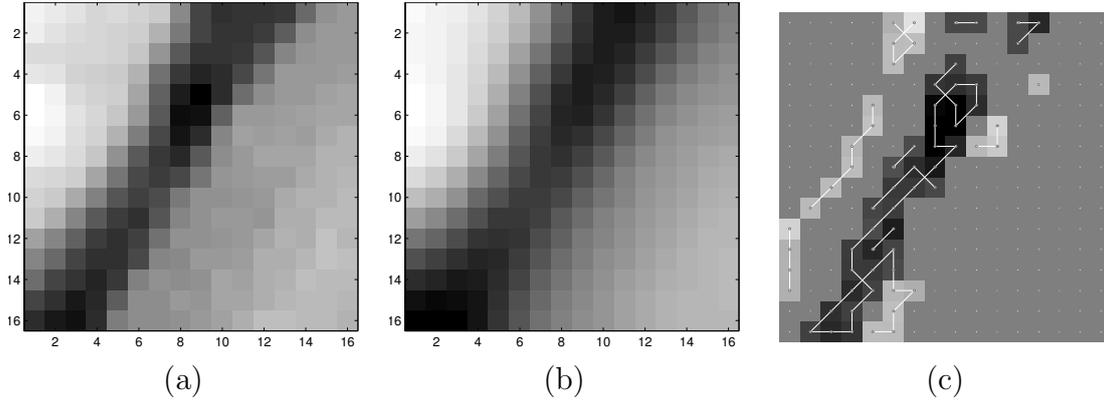


Abbildung 4.1: (a) Originalbild 16×16 , (b) Geglättetes Bild, (c) Teilbild des Differenzbildes mit 64 von Null verschiedenen Werten und der erste Pfad der EPWT.

Transformation angewandt. Auf den Tiefpassanteil der resultierenden Waveletkoeffizienten können nun noch weitere Level der Easy-Path-Wavelet-Transformation angewandt werden. Dies geschieht wieder analog zum vorigen Kapitel. Sei die Menge, die aus den beiden Indizes $p^{2\zeta}(2n)$ und $p^{2\zeta}(2n+1)$, $n \in \{1, \dots, \frac{2^{2\zeta}}{2}\}$, besteht, als Indexmenge $I_n^{2\zeta-1}$ bezeichnet. Ihr sei der n -te Tiefpasswert zugeordnet. Nun wird ein Pfad durch die so resultierenden $\frac{2^{2\zeta}}{2}$ Indexmengen gesucht, so dass auf diesem Pfad benachbarte Indexmengen einen möglichst ähnlichen Tiefpasswert besitzen und man somit, wenn man eine Wavelet-Transformation auf die auf diese Weise geordneten Tiefpasswerte anwendet, wiederum viele kleine Hochpasswerte erhält. Der Pfad des nächsten Levels wird nun mit einer beliebigen Indexmenge gestartet. Man betrachtet nun alle an diese Indexmenge angrenzenden Indexmengen, die noch nicht für den Pfad verwendet wurden und nimmt entweder die angrenzende Indexmenge als nächste Indexmenge des Pfades, die den Tiefpasswert mit der geringsten Differenz zu dem Tiefpasswert der letzten Indexmenge auf dem Pfad aufweist oder benutzt eine der anderen im vorigen Kapitel vorgestellten Techniken für die Wahl der nächsten Indexmenge. Der Pfad p^j , den man so schließlich im Level j erhält, ist eine Permutation der Indizes n der Indexmengen I_n^j , $n = 1, \dots, 2^j$. Auf die zu den Indexmengen korrespondierenden Tiefpasswerte, angeordnet gemäß des Pfades, wird wiederum eine eindimensionale Wavelet-Transformation angewandt. Nun werden neue Indexmengen gebildet, wobei die Indexmenge I_n^{j-1} eine Vereinigung der beiden Indexmengen I_{2n-1}^j und I_{2n}^j ist und den n -ten Tiefpasswert, der bei der Wavelet-Transformation im letzten Level entstanden ist, zugewiesen bekommt, $n = 1, \dots, 2^{j-1}$. Wiederum wird ein Pfad durch diese Indexmengen gesucht, und so weiter. Insgesamt kann man, bei einem Teilbild mit $2^{2\zeta}$ Werten bis zu $2\zeta - s$ Level der EPWT anwenden, sofern die Länge der Waveletfilter kleiner oder gleich 2^s ist.

4.3.1 Beispiel

Die Abbildung 4.3 zeigt die Pfade in den einzelnen Leveln für ein Beispielbild 4.2. Die weißen, unnummerierten Kästchen stehen für die Pixel, die nicht zu dem Teil des Differenzbildes, den wir betrachten, gehören. Die grauen, nummerierten Kästchen stellen die betragsmäßig größten $16 = \frac{8 \times 8}{4}$ Werte des Differenzbildes dar. Der Einfachheit halber haben wir die Pixel des Teilbildes mit eindimensionalen Indizes versehen, wobei wir die bijektive Abbildung γ mit $\gamma(i, j) := i + N_1(j - 1)$ verwendet haben. Die Nachbarschaft $N(i, j)$ eines Pixels (i, j) entspricht somit der Menge

$$\mathcal{N}(i + N_1(j - 1)) := \{i_1 + N_1(j_1 - 1) \mid (i_1, j_1) \in N(i, j)\}.$$

Für unser Beispiel setzen wir die Schranke $\tilde{\theta} := 0$, das heißt, wir nehmen, wenn wir n Komponenten des Pfades, $n \in \{1, \dots, 15\}$, bestimmt haben, und $p^{2\zeta}(n) = q$ ($q \in \{1, 9, 17, 20, 21, 22, 25, 28, 30, 31, 34, 35, 38, 39, 43, 51\}$) ist, als nächstes Pixel auf dem Pfad immer das Nachbapixel aus $\tilde{N}(q)$, dessen Wert die Differenz zum Wert von q minimiert. D.h., wir wählen $p^{2\zeta}(n + 1)$ derart, dass $|\tilde{u}^r(\gamma^{-1}(q)) - \tilde{u}^r(\gamma^{-1}(p^{2\zeta}(n + 1)))|$ minimal wird. Hier bezeichne γ^{-1} die Umkehrabbildung von γ . Wir beginnen den Pfad mit $p^4(1) = 1$. In der Nachbarschaft $\tilde{N}(p^4(1))$ liegt nur ein Pixel, nämlich das Pixel mit der Nummer 9, somit folgt $p^4(2) = 9$. Nachdem die nächsten drei Komponenten des Pfades, $p^4(3) = 17$, $p^4(4) = 25$ und $p^4(5) = 34$, bestimmt sind, haben wir für die Wahl der nächsten Pfadkomponente zwei Möglichkeiten, da in $\tilde{N}(p^4(5))$ zwei mögliche Indizes enthalten sind und zwar die Indizes 35 und 43. Man sieht, dass der Grauwert von dem Pixel mit der Nummer 35 näher an dem Grauwert des Pixels mit der Nummer 34 ist, als der Grauwert des Pixels mit der Nummer 43. Deshalb setzen wir $p^4(6) = 35$. Auch für $p^4(7)$ gibt es zwei mögliche Kandidaten, nämlich die Pixel mit den Nummern 28 und 43. Wenn man die Bildwerte betrachtet, stellt man fest, dass das Pixel mit der Nummer 43 einen ähnlicheren Wert besitzt als das Pixel mit der Nummer 28 und somit ist $p^4(7) = 43$. Da das Pixel mit der Nummer 43 nur einen Nachbarn besitzt, der in $\tilde{N}(p^4(7))$ ist, nämlich das Pixel mit der Nummer 51, ist $p^4(8) = 51$. Nun sind wir in einer Sackgasse gelandet und müssen also ein neues Pfadstück beginnen. Wir wählen hier das Pixel mit dem kleinsten euklidischen Abstand zu $p^4(8)$, d.h., wir setzen $p^4(9) = 28$. Das erste Bild der Abbildung 4.3 zeigt den vollständigen Pfad p^4 . Nachdem eine eindimensionale Wavelet-Transformation auf $\{\tilde{u}^4(p^4(n))\}_{n=1}^{16}$ angewandt wurde, erhält man die 8 Waveletkoeffizienten $|g^3(n)|_{n=1}^8$, die gespeichert werden und die 8 Tiefpasskoeffizienten $(\tilde{u}^3(n))_{n=1}^8$. Die Werte der Tiefpasskoeffizienten werden nun

der Reihe nach den Indexmengen $I_n^3 = p^4(2n - 1) \cup p^4(2n)$, $n = 1, \dots, 8$ zugewiesen, die auch im zweiten Bild eingezeichnet sind, wobei die Grauwerte der Indexmengen die Werte der Tiefpasskoeffizienten darstellen. Nun suchen wir einen Pfad durch diese Indexmengen. Eine simple Lösung wäre natürlich, $p^3 = (1, 2, 3, 4, 5, 6, 7, 8)$ zu wählen. Dann würde das zweite Level der Wavelet-Transformation einfach auf $(\tilde{u}^3(n))_{n=1}^8$ angewandt werden und es würden uns keine weiteren Pfadkosten entstehen. Andererseits sind auf diese Weise die entstehenden neuen Waveletkoeffizienten größer als wenn man die Tiefpasswerte vorher so angeordnet hätte, dass ähnlich große Werte nah beieinander stehen. In dem vorletzten Bild der Abbildung, 4.3(c), ist ein Pfad p^3 dargestellt, der für kleinere Waveletkoeffizienten sorgt, weil immer die angrenzende Nachbarindexmenge als nächste Indexmenge auf dem Pfad genommen wird, die den ähnlichsten Wert zu der momentan letzten Indexmenge auf dem Pfad aufweist. Da der Wert von $p^3(3) = 3$ näher an dem Wert von der fünften Indexmenge als an dem Wert der vierten Indexmenge ist, wird hier zum Beispiel $p^3(4) = 5$ gesetzt. Die fünfte Indexmenge hat nur eine benachbarte, noch nicht zum Pfad gehörende Indexmenge, nämlich die sechste Indexmenge, so dass $p^3(5) = 6$ ist. Die sechste Indexmenge hat wieder zwei benachbarte Indexmengen, durch die der Pfad noch nicht geführt hat. Man sieht allerdings, dass der Wert der siebten Indexmenge näher an dem Wert der sechsten Indexmenge ist, als der Wert der achten Indexmenge, so dass wir $p^3(6) = 7$ und $p^3(7) = 8$ wählen. Jetzt müssen wir ein neues Pfadstück starten, weil die achte Indexmenge keine angrenzenden Indexmengen, die noch nicht Teil des Pfades sind, besitzt. Die einzige Indexmenge, die noch übrig ist, ist die vierte Indexmenge und somit ist $p^3(8) = 4$. Der komplette Pfad lautet $p^3 = (1, 2, 3, 5, 6, 7, 8, 4)$. Im letzten Bild 4.3(d) sind die neu gebildeten Indexmengen I_n^2 , $n = 1, 2, 3, 4$ zu sehen, die jeweils aus I_{2n-1}^3 und I_{2n}^3 bestehen.

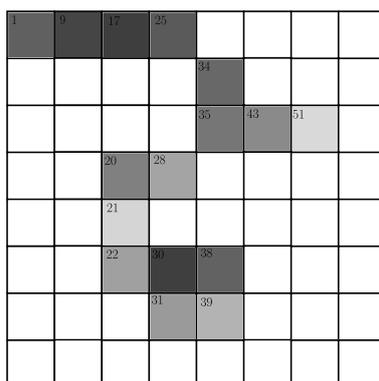


Abbildung 4.2: Die nummerierten Pixel gehören zu dem Teil des Differenzbildes, den wir betrachten.

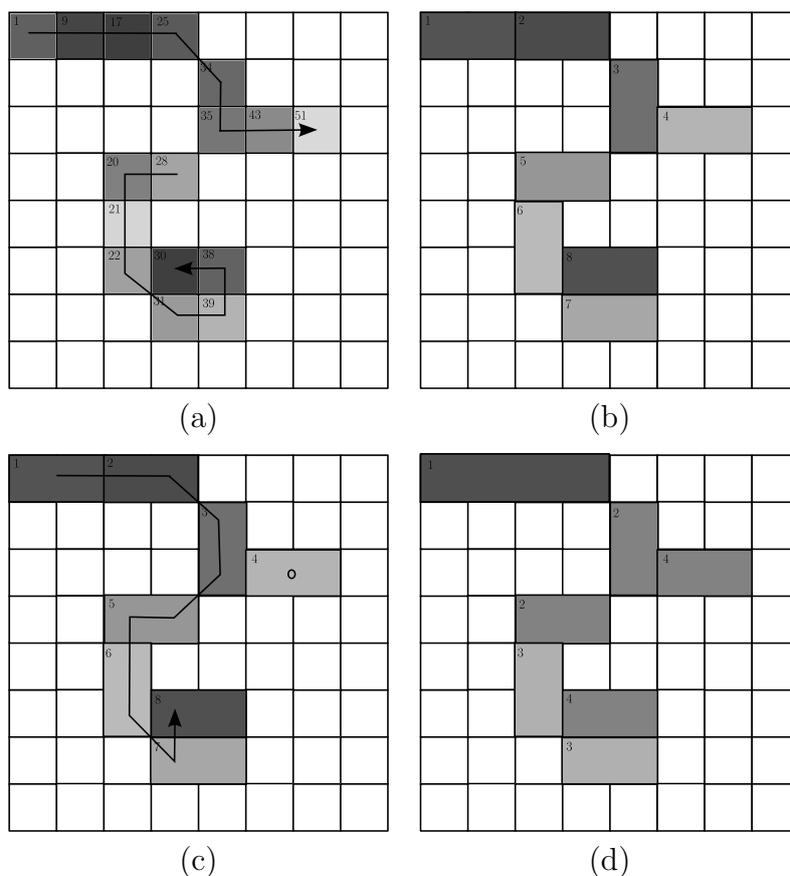


Abbildung 4.3: (a) der erste Pfad, (b) die Indexmengen nach dem ersten Level EPWT, (c) der zweite Pfad, (d) die neuen Indexmengen.

Damit wir eine noch dünner besetzte Darstellung von \tilde{u}^r erhalten, wenden wir auf die resultierenden Waveletkoeffizienten ein hartes Shrinkage an, so dass nur die betragsmäßig größten N Waveletkoeffizienten nicht Null gesetzt werden. Die dünn besetzte Darstellung, die wir nun erhalten, bezeichnen wir mit v_N . Außerdem haben wir selbstverständlich noch den Pfad, der bei der Anwendung der EPWT bestimmt wurde, zu speichern. Mit Hilfe des Pfades können wir nun aus v_N eine Approximation von \tilde{u}^r rekonstruieren. Diese Rekonstruktion bezeichnen wir mit \tilde{u}_N^r .

4.4 4. Schritt: Zusammenfügen

Die Approximation des geglätteten Bildes, \tilde{u}_M^{sm} , die mit Hilfe der Tensorprodukt-Wavelet-Transformation erreicht wurde, und die Approximation der Kanten und Texturen, \tilde{u}^r , die

mit Hilfe der EPWT erzeugt wurde, liefern zusammengefügt eine gute Approximation des Originalbildes u , die unter Verwendung von $M + N$ Waveletkoeffizienten erreicht werden kann. Man darf allerdings nicht vergessen, dass man zusätzlich zu v_M und v_N , die insgesamt aus $M + N$ von Null verschiedenen Waveletkoeffizienten bestehen, noch den Pfad, der bei der EPWT entstanden ist, und unter Umständen, je nach Speicherungsmethode des Pfadvektors, die Position des Teilbildes, d.h., die Information, welche $2^{2\zeta}$ Bildwerte des Differenzbildes u^r approximiert wurden, zu speichern hat. Insbesondere der erste Pfad lässt sich, wie im Kapitel zur Easy-Path-Wavelet-Transformation beschrieben, effizienter speichern, wenn man statt der Reihenfolge der Indizes nur die Richtungsänderungen des Pfades speichert. Das heißt, man speichert, solange die Richtung beibehalten wird, Nullen. Erst wenn die Richtung geändert wird, speichert man, welches Pixel ausgehend von dem Pixel, das in Richtung des Pfades liegt, genommen wurde. Das heißt, wenn man den Pfad bis zu einem Pixel p bestimmt hat, so weist man den bis zu acht „geeigneten“ Nachbarpixeln des Pixels p , durch die der Pfad noch nicht geführt hat, die Zahlen 0 bis höchstens 7 zu. Dabei beginnt man im Uhrzeigersinn bei dem ersten „geeigneten“ Pixel ab dem Nachbarpixel, das so liegt, dass die Richtung des Pfades beibehalten wird. Nun speichert man im Pfadvektor nicht den Index des nächsten Pixels, das genommen wird, sondern nur die Zahl zwischen 0 und 7, die ihm zugewiesen wurde. Ein Pixel wird hier „geeignet“ genannt, wenn es noch nicht auf dem Pfad liegt und zu \tilde{u}^r gehört. Man beachte, dass um eine Rekonstruktion gewährleisten zu können, auch die Position des Differenzbildes gespeichert werden muss. Wenn man unter allen „geeigneten“ Pixeln die Pixel versteht, die noch nicht auf dem Pfad liegen und somit auch die Nachbarpixel von p , die nicht Bestandteil von \tilde{u}^r sind, bei der Nummerierung (aber natürlich nicht bei der Verlegung des Pfades) miteinbezieht, so ist die Position des Differenzbildes schon automatisch mit in dem Pfadvektor gespeichert. Allerdings enthält der Pfadvektor bei dieser Speichermethode größere Einträge als bei der ersten Variante.

In dem Beispiel, das in Abbildung (4.3) vorgestellt wurde, könnte man den ersten Pfad zum Beispiel folgendermaßen

$$((1, 1), 0, 0, 0, 1, 1, 6, 0; (1, -3), 4, 6, 0, 7, 7, 6, 6)$$

speichern. Hier ist $(1, 1)$ der Startindex und $(1, -3)$ die Differenz der nichtbenachbarten Pixel mit den Indizes $p^4(8)$ und $p^4(9)$. Wenn man die erste Komponente eines neuen Pfadstücks bestimmt hat (also in diesem Fall zum Beispiel $(1, 1)$), so bedeutet eine 0 im

nächsten Eintrag, dass der Pfad die rechte Richtung eingeschlagen hat.

Auf diese Weise ist die Position des Differenzbildes im ersten Pfad schon mitgespeichert.

4.5 Ein Beispiel zur Erläuterung

In dem folgenden Beispiel werden die einzelnen Schritte anhand eines 256×256 Pixel großen Ausschnitts des Bildes SAILS verdeutlicht. Das erste Bild der Abbildung 4.4 zeigt das Original u , für das eine dünnbesetzte Darstellung gefunden werden soll. Dieses Bild wird nun mit dem oben beschriebenen Glättungsfilter mit den Parametern $\tau = 0,17$ und $K = 5$ geglättet. Das Resultat ist in Abbildung 4.4(b) zu sehen. Wir berechnen als Nächstes das Differenzbild, von dem wir anschließend außer den betragsmäßig größten $16384 = \frac{256 \times 256}{4}$ Werten alle Werte zu Null setzen und so das Differenzbild \tilde{d} erhalten, das aus Kanten und Texturen besteht. Wenn man dieses Differenzbild \tilde{d} vom Original abzieht, so erhält man das geglättete Bild \tilde{u}^{sm} , das in Abbildung 4.4(c) dargestellt ist. Dieses Bild wird nun mittels der biorthogonalen 9/7-Tensorprodukt-Wavelet-Transformation von Cohen, Daubechies und Feauveau approximiert. Von den $256 \times 256 = 65536$ Waveletkoeffizienten werden nur die betragsmäßig größten 1200 Waveletkoeffizienten nicht zu Null gesetzt. Abb. 4.4(d) zeigt diese Approximation \tilde{u}_{1200}^{sm} . In Abb. 4.4(e) ist das Differenzbild $u - \tilde{u}_{1200}^{sm}$ dargestellt. Hier wurden die Farben zur besseren Darstellung invertiert, so dass weiß für Null und Schwarz für den Wert 255 steht. Die kleinen Differenzen des Differenzbildes werden nun zu Null gesetzt, so dass nur die betragsmäßig größten Differenzen, die zu den Kanten und Texturen des Originalbildes korrespondieren, erhalten bleiben. In unserem Beispiel wurden nur die betragsmäßig größten 16384 Werte behalten. Eine ebenfalls invertierte Version dieses neuen Differenzbildes erhalten wir in Abbildung 4.4(f). Diese 16384 Werte des Differenzbildes werden nun mittels der Easy-Path-Wavelet-Transformation approximiert. Die eindimensionale Wavelet-Transformation, die entlang des Pfades angewandt wird, ist ebenfalls die 9/7-Wavelet-Transformation. Auch hier werden die resultierenden Waveletkoeffizienten nicht alle für die Rekonstruktion des Differenzbildes verwendet, sondern nur die 800 betragsmäßig größten. Die Approximation des Differenzbildes, die man so nach der Rekonstruktion erhält, zeigt Abbildung 4.4(g), wobei die Grauwerte wieder, wie auch schon zuvor, zur besseren Darstellung invertiert sind. Wenn wir nun die zuletzt berechnete Approximation der Kanten und Texturen zu der Approximation des geglätteten Bildes, also zu Abbildung 4.4(d), addieren, erhalten wir Abbildung 4.4(h), das eine sehr gute Approximation des Originalbildes liefert. Selbst kleine Details wie die Zahlen auf den Segeln,

die Muster der Segel, Wellen und Segler sind klar zu erkennen. Für diese Approximation wurden insgesamt nur 2000 Waveletkoeffizienten verwendet.

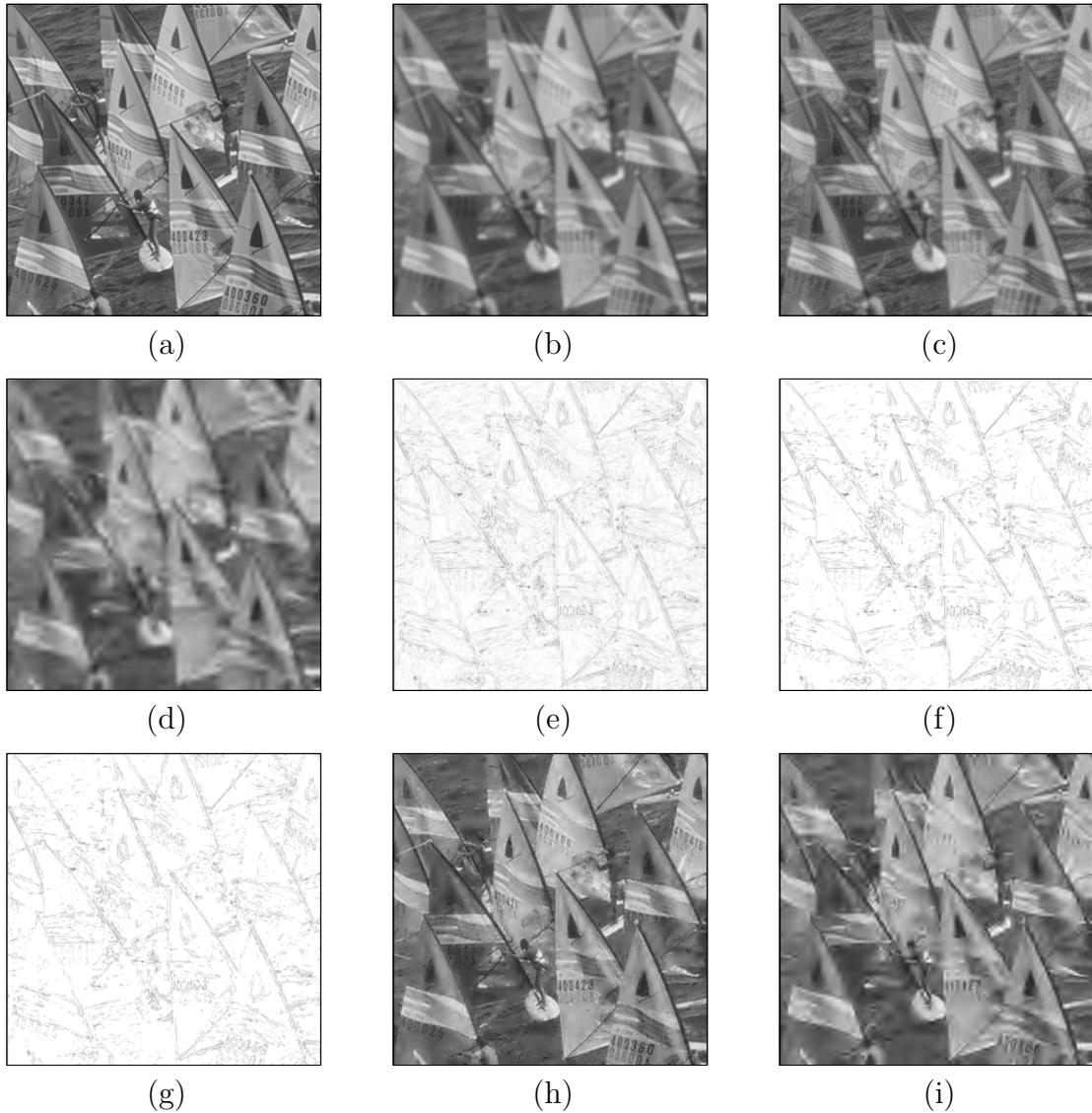


Abbildung 4.4: Illustration der einzelnen Schritte. (a) Originalbild, (b) geglättetes Bild u^{sm} , (c) geglättetes Bild \tilde{u}^{sm} , (d) Waveletapproximation \tilde{u}_{1200}^{sm} , (e) Differenzbild u^r , absolute Werte, invertiert, (f) \tilde{u}^r , invertiert, (g) EPWT-Approximation \tilde{u}_{800}^r , invertiert, (h) Approximation $\tilde{u}_{1200}^{sm} + \tilde{u}_{800}^r$ mit 2000 Koeffizienten, (i) biorthogonale 9/7-Tensorprodukt-Waveletapproximation mit 2000 Koeffizienten.

Eine Approximation des Originalbildes mittels einer $9/7$ -Tensorprodukt-Wavelet-Transformation, ebenfalls unter Verwendung der betragsmäßig größten 2000 Waveletkoeffizienten, liefert Abbildung 4.4(i). Dieses Bild ist um Einiges undeutlicher und verwischter als die Approximation mit der Hybridmethode. Hier sind die Zahlen auf den Segeln nicht mehr lesbar und auch die Muster der Segel sind nur noch verschwommen. Die Segler sind ebenfalls nicht mehr so klar zu erkennen. Man muss aber beachten, dass bei dem letzten Bild außer den 2000 Waveletkoeffizienten (und deren Positionen) keinerlei weitere Informationen gespeichert werden müssen während man für das Bild, das die Hybridmethode ergeben hat, zusätzlich noch den Pfad für die EPWT zu speichern hat.

4.6 Numerische Ergebnisse

In diesem Abschnitt präsentieren wir numerische Ergebnisse des Hybridverfahrens. Wir wenden es auf diverse Bilder der Größe 256×256 Pixel wie z.B. LENA, PEPPER, CAMERAMAN, CLOCK sowie auf 256×256 Pixel große Ausschnitte der Bilder SAILS, GOLDHILL und BARBARA an und vergleichen die Ergebnisse mit denen einer $9/7$ -, $7/9$ - und $D4$ -Tensorprodukt-Wavelet-Transformation. Außerdem betrachten wir den Einfluss von unterschiedlichen Zeitschrittweiten τ und Iterationsanzahlen K im ersten Schritt des Verfahrens sowie von unterschiedlichen Wavelet-Transformationen zur Approximation des geglätteten Bildes bzw. bei der Easy-Path-Wavelet-Transformation. Wir betrachten auch unterschiedliche Methoden, um den Pfad in weiteren Leveln günstiger zu speichern. Desweiteren vergleichen wir die Speicherkosten der Hybridmethode mit denen von Curvelets und der $9/7$ -Transformation.

Im ersten Experiment berechnen wir eine dünnbesetzte Darstellung der Bilder, die aus nur 500 Waveletkoeffizienten besteht. Im ersten Schritt unseres Verfahrens wenden wir 5 Iterationen des Glättungsfilters mit einer Zeitschrittweite von $\tau = 0,17$ an. Die Parameter θ im ersten und im dritten Schritt werden so gewählt, dass nur die $16384 = \frac{256 \times 256}{4}$ betragsmäßig größten Werte von d bzw. u^r nicht zu Null gesetzt werden. Auf den glatten Teil des Bildes werden 5 Level der biorthogonalen $9/7$ -Tensorprodukt-Wavelet-Transformation angewandt und nur die 300 betragsmäßig größten Werte behalten. Desweiteren wenden wir 11 Iterationen der EPWT zur Approximation des Teilbildes \tilde{u}^r , das die Kanten und Texturen enthält, an, wobei wir als Wavelet-Transformation auch hier die - allerdings eindimensionale - $9/7$ -Transformation anwenden. Wir setzen nur die 200 betragsmäßig größten Waveletkoeffizienten nicht zu Null. Tabelle 4.1 zeigt die Anzahl der Nichtnulle-

Bild	N _{nu}	9/7	Hybrid		MPkt-Hybrid		Einfach-Hybrid	
		PSNR	PSNR	Entropie	PSNR	Entropie	PSNR	Entropie
BARBARA	500	23.33	27.34	1.0497	27.28	1.0070	24.42	0.4010
CAMERAMAN	500	22.54	27.61	1.0714	27.49	0.9893	23.79	0.3794
CLOCK	500	24.61	31.06	1.0163	30.87	0.8742	26.69	0.3014
GOLDHILL	500	24.18	28.18	0.9918	28.19	0.8408	25.98	0.3300
LENA	500	23.21	28.02	1.0343	27.91	0.9022	24.66	0.3313
PEPPER	500	23.41	28.07	1.0286	28.03	0.8795	24.89	0.3143
SAILS	500	21.32	25.52	1.0179	25.42	0.9190	22.95	0.3664
BARBARA	2000	26.07	30.50	1.1097	30.50	1.0950	28.12	0.4411
CAMERAMAN	2000	27.17	31.46	1.1033	31.35	1.0472	28.36	0.4153
CLOCK	2000	29.93	35.48	1.0416	35.55	0.9329	32.49	0.3266
GOLDHILL	2000	27.82	31.41	0.9860	31.37	0.8986	29.90	0.3556
LENA	2000	28.16	32.66	1.0715	32.52	0.9699	29.97	0.3790
PEPPER	2000	28.84	32.97	1.0488	33.01	0.9385	30.62	0.3632
SAILS	2000	24.57	28.30	1.0195	28.26	0.9666	26.77	0.3916

Tabelle 4.1: Vergleich der 9/7-Wavelet-Transformation mit der Hybridmethode.

mente (N_{nu}, zweite Spalte) und die PSNR-Werte, die man bei Anwendung der Hybridmethoden „Hybrid“ „MPkt-Hybrid“ und „Einfach-Hybrid“ erhält. Hierbei besteht der einzige Unterschied zwischen diesen Methoden in der Wahl des Pfades in den weiteren Leveln 2, ..., 11. Ihre genaue Funktionsweise wurde bereits im Kapitel über die EPWT erläutert. Außerdem ist in der Tabelle die Entropie des Pfades der EPWT für die unterschiedlichen Methoden angegeben. Um einen Vergleich zu haben, ist in der dritten Spalte der Tabelle der PSNR-Wert angegeben, den man erhält, wenn man eine 9/7–Tensorprodukt-Wavelet-Transformation auf das Originalbild anwendet und die betragsmäßig größten 500 Waveletkoeffizienten behält.

Die Originalbilder sowie deren Approximationen, die bei Anwendung der Hybridmethode „Hybrid“ bzw. bei der Anwendung einer 9/7–Tensorprodukt-Wavelet-Transformation auf das Originalbild entstehen, sind in den Abbildungen 4.6 und 4.5 zu sehen. Das Original ist jeweils das in der linken Spalte dargestellte Bild, während das mittlere Bild durch die 9/7–Transformation und das rechte Bild durch die Hybridmethode entstanden ist.

In einem weiteren Experiment werden die Hybridmethoden auf die gleichen Bilder und mit den gleichen Parametern angewendet, wobei wir diesmal die betragsmäßig größten 1200 Waveletkoeffizienten behalten, die bei der Approximation des glatten Teils entstehen sowie die 800 betragsmäßig größten Waveletkoeffizienten, die bei der EPWT entstehen. Die Resultate sind ebenfalls in Tabelle 4.1 zu sehen.



Abbildung 4.5: (Links) Originalbild; (Mitte) 9/7-Tensorprodukt-Wavelet-Transformation mit 500 von Null verschiedenen Waveletkoeffizienten (Nnu); (Rechts) Hybridmethode mit 500 von Null verschiedenen Waveletkoeffizienten.



Abbildung 4.6: (Links) Originalbild; (Mitte) $9/7$ -Tensorprodukt-Wavelet-Transformation mit 500 von Null verschiedenen Waveletkoeffizienten (Nnu); (Rechts) Hybridmethode mit 500 von Null verschiedenen Waveletkoeffizienten.

Im Folgenden noch ein paar Anmerkungen zur Entropie bzw. zum Pfad: Wie schon im vierten Schritt angedeutet, lässt sich das Differenzbild im ersten Pfad mitspeichern. Die Anzahl der unterschiedlichen Werte im ersten Pfadvektor lässt sich stark begrenzen, wenn man in der Regel nur speichert, welche der acht möglichen Richtungen man einschlägt (siehe dazu auch das Kapitel zur EPWT). Wenn der Pfad einer Kante folgt, so wird oft die

eingeschlagene Richtung beibehalten. Dadurch, dass man die Richtung nur wechselt, wenn die Differenz zwischen den Werten zu groß (größer als eine im Vorhinein festgelegte Schranke) ist - und somit ziemlich häufig Nullen speichert - erreicht man ebenfalls eine geringere Entropie. Andere Werte als die acht Richtungen erhält man nur, wenn man einen Sprung im Pfad hat. Dann muss der Index des ersten Pixels des neuen Pfadstücks oder seine Differenz zum vorigen Pixel gespeichert werden, um später die Rekonstruktion gewährleisten zu können. Dies funktioniert natürlich nur, wenn der Index des Pixels (oder die Differenz zum vorigen Pixel, je nachdem was man zur Speicherung benutzt) größer als sieben ist, da man ansonsten nicht mehr genau sagen kann, ob die Richtung des Pfades oder der Index des Pixels (bzw. die Differenz zum vorigen Pixel) gemeint waren. Um dieses Problem zu vermeiden, könnte man standardmäßig auf jeden Index, der gespeichert wird (bzw. jede Differenz zum vorigen Pixel), die Zahl sieben aufaddieren. Wüsste man die Position des Differenzbildes, würde es sogar reichen, zum Beispiel 7 gleichmäßig über das Bild verteilte Pixel auszusuchen, die noch nicht für den Pfad verwendet wurden und im Pfad nur mit einer Zahl zwischen 1 und 7 abzuspeichern, welches der 7 Pixel letztendlich als erstes Pixel des neuen Pfadstücks gewählt wurde. In den weiteren Leveln kommen bei der Suche nach einer „Nachfolgerindexmenge“ auf dem Pfad alle angrenzenden Nachbarindexmengen in Frage, die noch nicht für den Pfad verwendet wurden. Man hat nun also nicht mehr nur bis zu 8 mögliche Richtungen, sondern unter Umständen ziemlich viele Nachbarindexmengen zur Auswahl. Auch das „Richtung-beibehalten“ lässt sich jetzt, wie auch bei der „normalen“ EPWT in höheren Leveln, nicht mehr so einfach definieren. Es stellt sich also die Frage, wie wir den Pfad in weiteren Leveln günstig speichern. Die Hybridmethode „Hybrid“ speichert in den weiteren Leveln einfach die Reihenfolge der Nummern der Indexmengen. Hier wird die nächste Indexmenge auf dem Pfad danach ausgesucht, dass sie angrenzend an die momentan letzte Indexmenge auf dem Pfad ist und noch nicht für den Pfad verwendet wurde. Außerdem soll sie von allen noch nicht für den Pfad verwendeten angrenzenden Indexmengen den Wert mit der kleinste Differenz zum Wert der momentan letzten Indexmenge des Pfades aufweisen. Da somit in den Pfaden der weiteren Level der Methode „Hybrid“ keine Zahl mehr als einmal vorkommt, erreicht man keine besonders gute Entropie.

Die Methode „Mittelpunkt-Hybrid“ („MPkt-Hybrid“), versucht, eine geringere Entropie als die Methode „Hybrid“ zu erzielen. Das erste Level funktioniert hier genauso wie das der normalen „Hybrid“-Methode. In den weiteren Level gehen wir analog zu den weiteren Leveln der Mittelpunkt-EPWT vor, die im Kapitel zur EPWT erläutert wurde. Auch hier nimmt man nicht die Nachbarindexmenge als nächste Komponente im Pfadvektor, der der ähnlichste Wert zugeordnet ist, sondern die erste Nachbarindexmenge gemäß der vorher bestimmten Ordnung, deren Wert nur um weniger als eine vorher festgelegte Schranke abweicht. In den ersten beiden Experimenten wurde diese Schranke gleich 13 gewählt. In dem Pfad wird nun nur die Position der gewählten Indexmenge in der geordneten Menge aller möglichen Nachbarindexmengen gespeichert. Dadurch erreicht man, dass in einem Pfad einige ähnliche Indizes/Positionen vorkommen. Weil man davon ausgehen kann, dass Indexmengen, deren Mittelpunkte am nächsten liegen, auch ähnliche zugewiesene Werte aufweisen, ist es wahrscheinlich, dass schon die Indexmengen, die in der geordneten Menge weit vorne liegen und somit einen nahen Mittelpunkt besitzen, die Anforderung erfüllen, dass die Differenz der Werte kleiner als eine gewisse Schranke ist. Das heißt, dass die Werte 0, 1, 2, 3 im Pfad besonders oft gespeichert werden.

Die Methode „Einfach-Hybrid“ besteht (wie auch schon die „Einfach-EPWT“ des EPWT-Kapitels) darin, nur den ersten Pfad zu berechnen und auf diesen Pfad mehrere Level einer eindimensionalen Wavelet-Transformation anzuwenden. In den weiteren Leveln werden also keine neuen Pfade mehr berechnet und dementsprechend fallen dann auch keine adaptivitätsbedingten Speicherkosten mehr an. Der Nachteil ist natürlich, dass man so in der Regel nicht mehr so viele ähnliche Werte nebeneinander angeordnet hat und somit größere Waveletkoeffizienten als in den ersten beiden Methoden erhält.

Die Tabellen 4.2 und 4.3 zeigen Resultate, die durch Anwendung verschiedener Wavelet-Transformationen zur Approximation des glatten Bildes und (des Teiles) des Differenzbildes entstanden sind. Hier wurde ebenfalls $\tau = 0,17$ verwendet, und es wurden 5 Iterationen des Glättungsprozesses und ebenfalls 5 Iterationen der Tensorprodukt-Wavelet-Transformation zur Approximation des glatten Bildes angewandt. Die Art der verwendeten Filterbank ist in der ersten Spalte angegeben. Die Schranke für die Differenzen der Pixelwerte im ersten Level der EPWT wurde hier gleich 13 gewählt. Für die weiteren Level wird entweder die benachbarte Indexmenge mit dem ähnlichsten Wert genommen (siehe „Hybrid“ Spalten 3 und 4), die „MPkt-Hybrid“-Methode mit der Schranke 13 angewandt (Spalten 5 und 6) oder die „Einfach-Hybrid“-Methode (Spalten 7 und 8) angewandt. Welche eindimensionale Wavelet-Transformation für die EPWT verwendet wurde, ist in der zweiten Spalte zu

lesen. Es wurden 11 Iterationen dieser Wavelet-Transformation verwendet. Zum Vergleich sind auch die PSNR-Werte angegeben, die bei Approximierung des Originalbildes mit der $D4-$, $7/9-$ bzw. $9/7-$ Tensorprodukt-Wavelet-Transformation erreicht wurden. In allen Fällen wurden nur die betragsmäßig größten 500 Waveletkoeffizienten zur Rekonstruktion verwendet. Als Testbilder wurden hier die Bilder PEPPER und LENA genutzt.

		Hybrid		MPkt-Hybrid		Einfach-Hybrid	
glatt	diff	PSNR	Entropie	PSNR	Entropie	PSNR	Entropie
D4	D4	27.23	1.0274	27.05	0.8921	24.41	0.3262
D4	7-9	26.86	1.0295	26.92	0.8929	24.13	0.3262
D4	9/7	27.23	1.0248	27.18	0.8909	24.36	0.3262
7-9	D4	26.82	1.0200	26.76	0.8806	24.02	0.3175
7-9	7-9	26.65	1.0132	26.62	0.8820	24.16	0.3175
7-9	9/7	26.90	1.0125	26.86	0.8735	24.30	0.3175
9/7	D4	28.01	1.0273	27.92	0.8772	24.96	0.3143
9/7	7-9	27.67	1.0253	27.65	0.8824	24.68	0.3143
9/7	9/7	28.07	1.0286	28.03	0.8795	24.89	0.3143
Tensorprodukt		PSNR					
D4		22.51					
7-9		22.15					
9/7		23.41					

Tabelle 4.2: Unterschiedliche Wavelet-Transformationen zur Approximation des glatten Teils und der Kanten und Texturen des PEPPER-Bildes.

glatt		Hybrid		MPkt-Hybrid		Einfach-Hybrid	
	diff	PSNR	Entropie	PSNR	Entropie	PSNR	Entropie
D4	D4	26.60	1.0097	26.47	0.8775	23.74	0.3216
D4	7-9	26.34	1.0044	26.42	0.8851	23.75	0.3216
D4	9/7	26.70	1.0051	26.57	0.8768	23.97	0.3216
7-9	D4	26.51	1.0016	26.47	0.8771	23.94	0.3191
7-9	7-9	26.33	1.0081	26.31	0.8779	23.88	0.3191
7-9	9/7	26.63	1.0137	26.55	0.8742	23.99	0.3191
9/7	D4	27.90	1.0318	27.75	0.9043	24.66	0.3313
9/7	7-9	27.62	1.0307	27.54	0.9067	24.57	0.3313
9/7	9/7	28.02	1.0343	27.91	0.9022	24.66	0.3313
Tensorprodukt		PSNR					
D4		22.13					
7-9		22.02					
9/7		23.21					

Tabelle 4.3: Unterschiedliche Wavelet-Transformationen zur Approximation des glatten Teils und der Kanten und Texturen des LENA-Bildes.

In den Tabellen (4.4) und (4.5) ist die Abhängigkeit der Approximationsergebnisse von der Wahl des Parameters τ bzw. von der Anzahl der Iterationen im Glättungsprozess dargestellt. Hier wurde als Testbild das Bild `CLOCK` verwendet. In der Tabelle 4.4 wurden 5 Iterationen des Glättungsfilters mit unterschiedlichen Zeitschrittweiten angewandt. Die anderen Parameter wurden genauso wie beim ersten Experiment gewählt. Zum Vergleich: der PSNR-Wert den man nach Anwendung der 9/7–Tensorprodukt-Wavelet-Transformation auf das Originalbild erhält, beträgt 24,61 dB wenn nur die betragsmäßig größten 500 Waveletkoeffizienten behalten wurden. Man sieht, dass die Ergebnisse nur leicht von τ abhängen. In der Tabelle (4.5) sind die PSNR- und Entropiewerte zu sehen, die bei fester Zeitschrittweite $\tau = 0,17$ aber unterschiedlichen Iterationsanzahlen im Glättungsprozess entstanden sind. Alle anderen Parameter wurden wie oben gewählt. Unsere numerischen Ergebnisse zeigen, dass fünf Iterationen des Glättungsfilters mit fester Zeitschrittweite $\tau = 0,17$ für alle betrachteten Bilder zufriedenstellende Ergebnisse liefern.

In Tabelle 4.6 sind die Ergebnisse für verschiedene Schranken für den ersten Pfad der EPWT zusammengestellt. Die restlichen Parameter wurden wie oben gewählt.

τ	Hybrid		MPkt-Hybrid		Einfach-Hybrid	
	PSNR	Entropie	PSNR	Entropie	PSNR	Entropie
0.04	30.95	1.0203	30.81	0.8656	26.65	0.3008
0.08	30.94	1.0209	30.95	0.8738	26.66	0.3047
0.12	30.89	1.0156	30.77	0.8650	26.58	0.2988
0.14	30.90	1.0224	30.67	0.8688	26.68	0.2958
0.18	30.93	1.0186	30.92	0.8744	26.62	0.2978
0.20	30.91	1.0197	31.03	0.8820	26.67	0.3016
0.22	31.09	1.0256	31.14	0.8841	26.74	0.3041
0.24	31.02	1.0235	30.98	0.8798	26.46	0.2982

Tabelle 4.4: PSNR- und Entropieresultate für verschiedene Werte für τ .

Iterationen	Hybrid		Mpkt-Hybrid		Einfach-Hybrid	
	PSNR	Entropie	PSNR	Entropie	PSNR	Entropie
1	30.81	1.0186	30.84	0.8688	26.56	0.3072
3	30.92	1.0221	30.80	0.8691	26.67	0.3033
5	31.06	1.0163	30.87	0.8742	26.69	0.3014
8	31.02	1.0321	30.99	0.8784	26.42	0.3008
12	30.86	1.0297	31.25	0.8795	26.43	0.2939
20	31.03	1.0269	31.09	0.8871	26.15	0.2912
24	31.17	1.0318	31.02	0.8878	26.19	0.2883
28	30.94	1.0298	31.03	0.8891	26.04	0.2929

Tabelle 4.5: PSNR- und Entropieergebnisse für verschiedene Iterationsanzahlen im Glättungsprozess.

Schranke	Hybrid		MPkt-Hybrid		Einfach-Hybrid	
	PSNR	Entropie	PSNR	Entropie	PSNR	Entropie
0	31.04	1.2176	31.04	1.2138	25.65	0.5342
1	31.05	1.2048	31.19	1.1966	25.59	0.5193
3	30.94	1.1678	30.88	1.1437	25.65	0.4704
6	31.11	1.0936	31.02	1.0390	25.73	0.3817
9	31.10	1.0670	31.14	0.9676	26.17	0.3335
12	31.25	1.0407	31.05	0.9017	25.96	0.2992
13	31.06	1.0163	31.03	0.8891	26.04	0.2929
15	31.17	1.0140	31.05	0.8483	26.12	0.2736
18	31.05	0.9956	30.84	0.7953	26.05	0.2901
21	31.06	0.9944	30.87	0.7678	26.42	0.2376
30	30.97	0.9644	30.72	0.6849	26.13	0.2029
90	29.79	0.8796	28.28	0.3910	24.25	0.0360

Tabelle 4.6: Unterschiedliche Schranken im ersten Level der EPWT.

Obwohl die Adaptivitätskosten der Hybridmethode, das heißt die Kosten, die für die Speicherung des Pfades anfallen, geringer als bei der EPWT sind, so sind diese Kosten doch immer noch nicht vernachlässigbar.

Um die Kodierungskosten der Mittelpunkt-Hybridmethode („MPkt-Hybrid“) mit denen der 9/7–Tensor– produkt–Wavelet–Transformation bzw. mit denen von Curvelets zu vergleichen, wenden wir folgendes vereinfachtes Schema an. Wir berechnen die Kosten für die Kodierung der Position der M Waveletkoeffizienten, die nicht Null gesetzt sind, mit

$$-\frac{M}{N} \log_2 \left(\frac{M}{N} \right) - \frac{N-M}{N} \log_2 \left(\frac{N-M}{N} \right),$$

wobei $N = 256 \times 256$ die Anzahl der Koeffizienten des Bildes ist. Für die Kodierung jedes einzelnen Waveletkoeffizienten werden jeweils 16 Bit verwendet. Für die Tensorprodukt–Wavelet–Transformation bestehen die Speicherkosten (in bpp) aus den Kosten für die Kodierung der Position der Waveletkoeffizienten, die nicht Null gesetzt sind, und den Kosten für die Speicherung dieser Koeffizienten, d.h. $16 \frac{M}{N}$. Für die Hybridmethode haben wir sowohl die Speicherkosten für die Approximation des glatten Bildes (die genauso wie oben berechnet werden) als auch die Kosten für die Speicherung der Approximation von Kanten und Texturen. Für die Speicherung der Kanten und Texturen müssen wir nicht nur

Bild	9/7			Curvelets			MPkt-Hybrid mit 9/7		
	Nnu	PSNR	Kosten	Nnu	PSNR	Kosten	Nnu	PSNR	Kosten
BARBARA	2000	26,07	0,6853	2000	22,84	0,7028	500	27,28	0,6880
CAMERAMAN	2000	27,17	0,6853	2000	22,69	0,7028	500	27,49	0,6880
CLOCK	2000	29,93	0,6853	2000	21,25	0,7028	500	30,87	0,6880
GOLDHILL	2000	27,82	0,6853	2000	23,25	0,7028	500	28,19	0,6880
LENA	2000	28,16	0,6853	2000	22,04	0,7028	500	27,91	0,6880
PEPPER	2000	28,84	0,6853	2000	21,77	0,7028	500	28,03	0,6880
SAILS	2000	24,57	0,6853	2000	20,75	0,7028	500	25,42	0,6880

Tabelle 4.7: Vergleich der Speicherkosten von 9/7-Wavelet-Transformation, Curvelets und Hybridmethode. PSNR in dB, Kosten in bpp.

die Positionen und die Werte der bei Anwendung der EPWT entstandenen Waveletkoeffizienten speichern, sondern auch den Pfad. Die Kosten für die Kodierung des Pfades seien hier mit 0,5 bpp geschätzt. Wenn man die Curvelet-Transformation (siehe MATLAB toolbox von www.curvelet.org) anwendet, erhält man aufgrund ihrer Redundanz 184985 Curveletkoeffizienten für ein 256×256 -Bild. Diese Redundanz führt zu ziemlich schlechten Kompressionsergebnissen verglichen mit anderen Methoden. In der Tabelle 4.7 sind die Kodierungskosten zusammengefasst, so dass man die Methoden einigermaßen fair vergleichen kann. Bei der Tensorprodukt-Wavelet-Transformation wurden die 2000 wichtigsten Waveletkoeffizienten nicht Null gesetzt, während bei der Hybridmethode nur 500 Waveletkoeffizienten nicht Null gesetzt wurden. Wir stellen fest, dass die Kodierung des Pfades der EPWT (0,5 bpp) den Großteil der kompletten Speicherkosten (0,6880 bpp) ausmacht. Man sieht, dass man sich noch Gedanken über eine bessere Speicherung des Pfades machen muss.

5 Approximationsresultate

In diesem Kapitel, das auf unseren Arbeiten [47] und [46] basiert, widmen wir uns der Frage, wie gut man Bilder oder, genereller gesagt, zweidimensionale Signale mittels der EPWT approximieren kann. Sei das Signal $F \in L^2([0, 1) \times [0, 1))$ auf endlich vielen Gebieten $\{\Omega_i\}_{1 \leq i \leq K}$ gleichmäßig stetig. Für die Gebiete Ω_i gelte, dass sie zusammenhängende Teilmengen von $[0, 1) \times [0, 1)$ sind, wobei die Ränder dieser Gebiete Ω_i stetig und von endlicher Länge sind und alle Gebiete $\{\Omega_i\}_{1 \leq i \leq K}$ zusammen eine disjunkte Partition von $[0, 1) \times [0, 1)$ bilden, d.h.

$$\bigcup_{i=1}^K \Omega_i = [0, 1) \times [0, 1) \quad \text{und} \quad \Omega_i \cap \Omega_j = \emptyset \text{ für } i \neq j, \forall i, j \in \{1, \dots, K\}.$$

Außerdem sei F in jeder Region Ω_i , $i \in \{1, \dots, K\}$, Hölder-glatt von der Ordnung $\alpha > 0$, das heißt, dass für alle μ -ten Ableitungen von F , $|\mu| = \lfloor \alpha \rfloor$,

$$|F^{(\mu)}(x) - F^{(\mu)}(y)| \leq C \|x - y\|_2^{\alpha - |\mu|} \quad (5.1)$$

gilt, wobei x und y im selben Gebiet Ω_i liegen und die Konstante $C > 0$ von i unabhängig ist. An der Grenze zweier verschiedener benachbarter Gebiete sei es erlaubt, dass f unstetig ist. Wir werden zeigen, dass, wenn obige Voraussetzungen erfüllt sind und die Pfadvektoren so gewählt wurden, dass die Indexmengen gewisse Eigenschaften erfüllen, eine optimale N -Term-Approximation von F erreicht werden kann.

Zuerst beschäftigen wir uns mit dem Fall, dass $0 < \alpha \leq 1$ ist, danach zeigen wir, dass die Approximationsresultate auch allgemein für $\alpha > 0$ gelten.

5.1 Approximationseigenschaften der EPWT - der Fall

$$0 < \alpha \leq 1$$

Im Folgenden sei die bivariate Funktion F^{2^J} mit $J > 0$ wieder die stückweise konstante Approximation des Bildes F , das auf dem äquidistanten Gitter

$$\left\{ \left(\frac{n}{2^J}, \frac{m}{2^J} \right) \mid 0 \leq n, m \leq N - 1 \right\}$$

abgetastet wurde und die Form

$$F^{2^J}(x) := \sum_{n \in I_{2^J}} F\left(\frac{n}{2^J}\right) \chi_{[0,1) \times [0,1)}(2^J x - n) \quad \text{für } x \in [0, 1) \times [0, 1) \quad (5.2)$$

hat mit $I_{2^J} := \{(i, j)\}_{i,j=0}^{2^J-1}$, d.h., für $x \in [2^{-J}n_1, 2^{-J}(n_1 + 1)) \times [2^{-J}n_2, 2^{-J}(n_2 + 1))$ sei $F^{2^J}(x) = F(2^{-J}n)$ mit $n = (n_1, n_2)$.

Die Indexmenge der Gitterpunkte, die in Ω_i liegen, wird im weiteren Verlauf mit

$$\Gamma_i^J := \left\{ n \in I_J \mid \frac{n}{2^J} \in \Omega_i \right\}$$

bezeichnet. Zur besseren Unterscheidung, welche Indexmenge gemeint ist, wird sie auch im weiteren Verlauf *Bildgebietsindexmenge* genannt. Offensichtlich gilt $\bigcup_{i=1}^K \Gamma_i^J = I_{2^J}$ und die Anzahl der Elemente der Bildgebietsindexmenge Γ_i^J beträgt für jedes $i \in \{1, \dots, K\}$ $\#\Gamma_i^J = C2^{2^J}$, wobei $C \leq 1$.

5.1.i Lemma. *Aus der Hölderstetigkeit von F erhält man, dass*

$$\left| F^{2^J}(2^{-J}q) - F^{2^J}(2^{-J}m) \right| \leq C2^{-J\alpha} \|q - m\|_2^\alpha \quad (5.3)$$

gilt, falls $q, m \in \Gamma_i^J$.

Beweis Aus der Definition von F^{2^J} folgt

$$\begin{aligned} \left| F^{2^J}(2^{-J}q) - F^{2^J}(2^{-J}m) \right| &= \left| F(2^{-J}q) - F(2^{-J}m) \right| \stackrel{(5.1)}{\leq} C \|2^{-J}m - 2^{-J}q\|_2^\alpha \\ &= C2^{-J\alpha} \|q - m\|_2^\alpha \end{aligned}$$

□

Der Einfachheit halber wird bei der Ausführung der EPWT die eindimensionale Haar-Wavelet-Transformation verwendet, die im ersten Kapitel eingeführt wurde. Der Pfad, der im ersten Level der EPWT bestimmt wird, hat nur eine sehr begrenzte Anzahl an Sprüngen, die insbesondere dann entstehen, wenn ein Pfadstück schon alle Indizes aus Γ_i^J , $i \in \{1, \dots, K\}$, „abgearbeitet“ hat und nun ein Pfadstück durch ein neues Bildgebiet beginnt. Die eindimensionale Funktion \tilde{f}^{2^J} ist entlang des Pfades $p^{2^J} = \left((p_1^{2^J})^T, \dots, (p_k^{2^J})^T \right)^T$ durch

$$\tilde{f}^{2^J}(t) := \sum_{\ell=0}^{2^{2^J}-1} F^{2^J} \left(\frac{p^{2^J}(\ell)}{2^J} \right) \varphi(2^{2^J}t - \ell) = \sum_{\ell=0}^{2^{2^J}-1} 2^{-J} F^{2^J} \left(\frac{p^{2^J}(\ell)}{2^J} \right) \varphi_{2^J, \ell}(t), \quad t \in [0, 1),$$

mit (siehe auch die Definition der Haar-Wavelets im Grundlagen-Kapitel, (2.2))

$$\varphi_{2^J, \ell}(t) = 2^J \varphi(2^{2^J}t - \ell) := \begin{cases} 2^J & t \in \left[\frac{\ell}{2^{2^J}}, \frac{\ell+1}{2^{2^J}} \right), \\ 0 & \text{sonst} \end{cases}$$

definiert. Mit dem obigen Lemma folgt, dass sie die Abschätzung

$$\begin{aligned} \left| \tilde{f}^{2^J} \left(\frac{2r}{2^{2^J}} \right) - \tilde{f}^{2^J} \left(\frac{2r+1}{2^{2^J}} \right) \right| &= \left| 2^{-J} \sum_{\ell=0}^{2^{2^J}-1} F^{2^J} \left(\frac{p^{2^J}(\ell)}{2^J} \right) \left(\varphi_{2^J, \ell} \left(\frac{2r}{2^{2^J}} \right) - \varphi_{2^J, \ell} \left(\frac{2r+1}{2^{2^J}} \right) \right) \right| \\ &= \left| F^{2^J} \left(\frac{p^{2^J}(2r)}{2^J} \right) - F^{2^J} \left(\frac{p^{2^J}(2r+1)}{2^J} \right) \right| \\ &\leq C \left\| \frac{p^{2^J}(2r)}{2^J} - \frac{p^{2^J}(2r+1)}{2^J} \right\|_2^\alpha \\ &= C 2^{-J\alpha} \|p^{2^J}(2r) - p^{2^J}(2r+1)\|_2^\alpha = C 2^{-J\alpha + \frac{1}{2}\alpha} \end{aligned} \tag{5.4}$$

für $r \in \{0, \dots, 2^{2^J-1} - 1\}$ erfüllt, wenn $p^{2^J}(2r)$ und $p^{2^J}(2r+1)$ benachbart in der selben Indexmenge Γ_i^J , $i \in 1, \dots, K$ liegen und somit höchstens den Abstand $\sqrt{2}$ haben. Nach der Anwendung einer eindimensionalen Haar-Wavelet-Transformation auf die ursprünglichen Daten $\{F^{2^J}(2^{-J}n)\}_{n \in I_j} = \left\{ \tilde{f}^{2^J}(2^{-2^J}\ell) \right\}_{\ell=0}^{2^{2^J}-1}$, angeordnet gemäß des Pfades p^{2^J} , erhält

man die Skalierungskoeffizienten

$$\begin{aligned}
f^{2^J-1}(\ell) &:= \langle \tilde{f}^{2^J}, \varphi_{2^J-1, \ell} \rangle = \sum_{k=0}^{2^{2^J}-1} F^{2^J} \left(\frac{p^{2^J}(k)}{2^J} \right) \langle \chi_{[\frac{k}{2^{2^J}}, \frac{k+1}{2^{2^J}}]}, \varphi_{2^J-1, \ell} \rangle \\
&= 2^{J-\frac{1}{2}} \sum_{k=0}^{2^{2^J}-1} F^{2^J} \left(\frac{p^{2^J}(k)}{2^J} \right) \langle \chi_{[\frac{k}{2^{2^J}}, \frac{k+1}{2^{2^J}}]}, \chi_{[\frac{2\ell}{2^{2^J}}, \frac{2\ell+2}{2^{2^J}}]} \rangle \\
&= 2^{J-\frac{1}{2}} F^{2^J} \left(\frac{p^{2^J}(2\ell)}{2^J} \right) \langle \chi_{[\frac{2\ell}{2^{2^J}}, \frac{2\ell+1}{2^{2^J}}]}, \chi_{[\frac{2\ell}{2^{2^J}}, \frac{2\ell+2}{2^{2^J}}]} \rangle \\
&\quad + 2^{J-\frac{1}{2}} F^{2^J} \left(\frac{p^{2^J}(2\ell+1)}{2^J} \right) \langle \chi_{[\frac{2\ell+1}{2^{2^J}}, \frac{2\ell+2}{2^{2^J}}]}, \chi_{[\frac{2\ell}{2^{2^J}}, \frac{2\ell+2}{2^{2^J}}]} \rangle \\
&= 2^{-J-\frac{1}{2}} \left(F^{2^J} \left(\frac{p^{2^J}(2\ell)}{2^J} \right) + F^{2^J} \left(\frac{p^{2^J}(2\ell+1)}{2^J} \right) \right)
\end{aligned} \tag{5.5}$$

und die Waveletkoeffizienten (für die Definition von $\psi_{2^J-1, \ell}$ siehe Grundlagen-Kapitel (2.1))

$$\begin{aligned}
g^{2^J-1}(\ell) &:= \langle \tilde{f}^{2^J}, \psi_{2^J-1, \ell} \rangle = \sum_{k=0}^{2^{2^J}-1} F^{2^J} \left(\frac{p^{2^J}(k)}{2^J} \right) \langle \varphi(2^{2^J}t - k), \psi_{2^J-1, \ell} \rangle \\
&= 2^{J-\frac{1}{2}} \sum_{k=0}^{2^{2^J}-1} F^{2^J} \left(\frac{p^{2^J}(k)}{2^J} \right) \langle \chi_{[\frac{k}{2^{2^J}}, \frac{k+1}{2^{2^J}}]}, \chi_{[\frac{2\ell}{2^{2^J}}, \frac{2\ell+1}{2^{2^J}}]} - \chi_{[\frac{2\ell+1}{2^{2^J}}, \frac{2\ell+2}{2^{2^J}}]} \rangle \\
&= 2^{J-\frac{1}{2}} \left(F^{2^J} \left(\frac{p^{2^J}(2\ell)}{2^J} \right) \frac{1}{2^{2^J}} - F^{2^J} \left(\frac{p^{2^J}(2\ell+1)}{2^J} \right) \frac{1}{2^{2^J}} \right) \\
&= 2^{-J-\frac{1}{2}} \left(F^{2^J} \left(\frac{p^{2^J}(2\ell)}{2^J} \right) - F^{2^J} \left(\frac{p^{2^J}(2\ell+1)}{2^J} \right) \right).
\end{aligned}$$

Die Waveletkoeffizienten kann man, wenn $p^{2^J}(\ell)$ und $p^{2^J}(2\ell+1)$ benachbart sind und in derselben Indexmenge liegen, mit Hilfe von (5.4) wie folgt abschätzen

$$\begin{aligned}
|g^{2^J-1}(\ell)| &= 2^{-J-\frac{1}{2}} \left(F^{2^J} \left(\frac{p^{2^J}(2\ell)}{2^J} \right) - F^{2^J} \left(\frac{p^{2^J}(2\ell+1)}{2^J} \right) \right) \\
&\stackrel{(5.4)}{\leq} 2^{-J-\frac{1}{2}} C 2^{-J\alpha+\frac{1}{2}\alpha} = \frac{1}{2} 2^{-J+\frac{1}{2}} C 2^{-J\alpha+\frac{1}{2}\alpha} = \frac{1}{2} C 2^{-(J-\frac{1}{2})(1+\alpha)}.
\end{aligned}$$

Im nächsten Level der Easy-Path-Wavelet-Transformation betrachten wir eine Zerlegung der Indexmenge Γ_i^J in Indexmengen $I_\ell^{2^J-1}$, $\ell = 0, \dots, 2^{2^J-1} - 1$, wobei diese Zerlegung durch den Pfad p^{2^J} der Art bestimmt wird, dass

$$I_\ell^{2^J-1} := \{p^{2^J}(2\ell), p^{2^J}(2\ell+1)\} \quad \text{mit} \quad \ell = 0, \dots, 2^{2^J-1} - 1$$

ist. Desweiteren sei

$$F^{2J-1}(x) := \sum_{\ell=0}^{2^{2J-1}-1} f^{2J-1}(\ell) h_{2J-1, \ell}(x), \quad x \in [0, 1) \times [0, 1) \quad (5.6)$$

mit

$$h_{2J-1, \ell}(x) := 2^{J-\frac{1}{2}} \left(\chi_{[0,1) \times [0,1)}(2^J x - p^{2J}(2\ell)) + \chi_{[0,1) \times [0,1)}(2^J x - p^{2J}(2\ell + 1)) \right),$$

d.h., dass den beiden Pixeln einer Indexmenge der Mittelwert dieser beiden Pixel zugeordnet wird. Wir suchen nun einen Pfad $p^{2J-1} = (p^{2J-1}(\ell))_{\ell=0}^{2^{2J-1}-1}$ durch diese Indexmengen I_{ℓ}^{2J-1} , $\ell = 0, \dots, 2^{2J-1} - 1$. Dieser Pfad soll möglichst zusammenhängend sein, das heißt, es soll für alle $r \in \{0, \dots, 2^{2J-2} - 1\}$ bis auf eine endliche Menge, deren Größe nicht von J abhängig ist, gelten, dass $I_{p^{2J-1}(2r+1)}^{2J-1} \in N \left(I_{p^{2J-1}(2r)}^{2J-1} \right)$ ist. Anschließend wenden wir ein Level der eindimensionalen Haar-Wavelet-Transformation auf

$$\tilde{f}^{2J-1}(t) := \sum_{\ell=0}^{2^{2J-1}-1} f^{2J-1}(p^{2J-1}(\ell)) \varphi_{2J-1, \ell}(t) \quad t \in [0, 1)$$

an. Wir erhalten so die Skalierungs- und Waveletkoeffizienten

$$f^{2J-2}(\ell) := \langle \tilde{f}^{2J-1}, \varphi_{2J-2, \ell} \rangle, \quad g^{2J-2}(\ell) := \langle \tilde{f}^{2J-1}, \psi_{2J-2, \ell} \rangle, \quad \ell = 0, \dots, 2^{2J-2} - 1.$$

Im nächsten Level wird dann das „Tiefpassbild“

$$F^{2J-2}(x) := \sum_{\ell=0}^{2^{2J-2}-1} f^{2J-2}(\ell) h_{2J-2, \ell}(x), \quad x \in [0, 1) \times [0, 1)$$

betrachtet, mit

$$h_{2J-2, \ell} := \frac{h_{2J-1, p^{2J-1}(2\ell)} + h_{2J-1, p^{2J-1}(2\ell+1)}}{\sqrt{2}}, \quad \ell = 0, \dots, 2^{2J-2} - 1,$$

wobei $h_{2J-2, \ell}$ in den Gebieten, die durch die Indexmengen $I_{p^{2J-1}(2\ell)}^{2J-1}$ und $I_{p^{2J-1}(2\ell+1)}^{2J-1}$ bestimmt sind, konstant 2^{J-1} ist und in den anderen Gebieten verschwindet. Dies kann man nun noch weiter fortsetzen. Im $(2J - j)$ -ten Level, $j = 2J - 1, 2J - 2, \dots, 0$ wird dement-

sprechend ein Pfadvektor $p^{j+1} = (p^{j+1}(\ell))_{\ell=0}^{2^{j+1}-1}$ durch die Indexmengen

$$I_\ell^{j+1} := I_{p^{j+2}(2\ell)}^{j+2} \cup I_{p^{j+2}(2\ell)}^{j+2}, \quad \ell = 0, \dots, 2^{j+1} - 1,$$

bestimmt. Als Nächstes wird nun eine Haar-Wavelet-Transformation auf

$$\tilde{f}^{j+1}(t) := \sum_{\ell=0}^{2^{j+1}-1} f^{j+1}(p^{j+1}(\ell)) \varphi_{j+1, \ell}(t), \quad t \in [0, 1)$$

angewandt, die die Skalierungs- und Waveletkoeffizienten

$$f^j(\ell) := \langle \tilde{f}^{j+1}, \varphi_j, \ell \rangle, \quad g^j(\ell) := \langle \tilde{f}^{j+1}, \psi_j, \ell \rangle, \quad \ell = 0, \dots, 2^j - 1$$

liefert.

5.1.1 Adaptive Haarwaveletbasis

Sei V_{2^J} der Raum der stückweise konstanten Funktionen

$$V_{2^J} = \text{span} \{ h_{2^J, \ell} \mid \ell = 0, \dots, 2^{2^J} - 1 \},$$

wobei die $h_{2^J, \ell}$ in L^2 normalisierte Funktionen sind, die auf einem Quadrat der Länge 2^{-J} konstant 2^J sind und ansonsten verschwinden, wobei für keine zwei Funktionen $h_{2^J, m}$ und $h_{2^J, n}$ mit $m \neq n$ gilt, dass sie auf dem gleichen Quadrat oder auch nur auf Teilen des gleichen Quadrats existieren. Genauer gesagt, es gelte

$$h_{2^J, \ell} := 2^J \chi_{[0,1) \times [0,1)}(2^J \cdot -p^{2^J}(\ell)), \quad \ell = 0, \dots, 2^{2^J} - 1.$$

Aus der Definition der $h_{2^J, \ell}$, $\ell = 0, \dots, 2^{2^J} - 1$, ergibt sich, dass sie eine orthonormale Basis des V_{2^J} bilden. Außerdem ist ersichtlich, dass die Funktion F^{2^J} von oben (5.2) auch folgendermaßen

$$F^{2^J}(x) = \sum_{\ell=0}^{2^{2^J}-1} F\left(\frac{p^{2^J}(\ell)}{2^J}\right) \frac{h_{2^J, \ell}(x)}{2^J}, \quad x \in [0, 1) \times [0, 1)$$

geschrieben werden kann.

Nach einem Level der EPWT, d.h. nachdem man ein Level der Haar-Wavelet-Transformation

auf die gemäß des Pfades $(p^{2J}(\ell))_{\ell=0}^{2^{2J}-1}$ angeordneten Funktionswerte angewendet hat, erhält man die neuen Funktionenräume

$$V_{2J-1}(F) := \text{span} \left\{ h_{2J-1, \ell} := \frac{h_{2J, 2\ell} + h_{2J, 2\ell+1}}{\sqrt{2}} \mid \ell = 0, \dots, 2^{2J-1} - 1 \right\},$$

$$W_{2J-1}(F) := \text{span} \left\{ \Psi_{2J-1, \ell} := \frac{h_{2J, 2\ell} - h_{2J, 2\ell+1}}{\sqrt{2}} \mid \ell = 0, \dots, 2^{2J-1} - 1 \right\}.$$

Diese Funktionenräume bestehen aus stückweise konstanten Funktionen, deren Träger sich über zwei Quadrate der Länge 2^{-J} erstreckt, die normalerweise benachbart sind. Die Funktionenräume $V_{2J-1}(F)$ und $W_{2J-1}(F)$ sind nicht nur Teilräume des oben definierten Funktionenraums V_{2J} . Es gilt ebenfalls

$$\begin{aligned} \langle h_{2J-1, \ell}, \Psi_{2J-1, m} \rangle &= \frac{\langle h_{2J, 2\ell} + h_{2J, 2\ell+1}, h_{2J, 2m} - h_{2J, 2m+1} \rangle}{2} \\ &= \begin{cases} 0 & \ell \neq m \\ \frac{\langle h_{2J, 2\ell}, h_{2J, 2\ell} \rangle - \langle h_{2J, 2\ell+1}, h_{2J, 2\ell+1} \rangle}{2} = 0 & \ell = m \end{cases} \end{aligned}$$

und

$$\begin{aligned} \frac{1}{\sqrt{2}} (h_{2J-1, \ell} + \Psi_{2J-1, \ell}) &= h_{2J, p^{2J}(2\ell)}, & \ell = 0, \dots, 2^{2J-1} - 1 \\ \frac{1}{\sqrt{2}} (h_{2J-1, \ell} - \Psi_{2J-1, \ell}) &= h_{2J, p^{2J}(2\ell+1)}, & \ell = 0, \dots, 2^{2J-1} - 1. \end{aligned}$$

Somit liefert das erste Level der EPWT eine orthogonale Zerlegung von F^{2J} in $F^{2J-1} \in V_{2J-1}(f)$ und $G^{2J-1} \in W_{2J-1}(f)$ mit F^{2J-1} wie in (5.6) definiert und

$$G^{2J-1}(x) = \sum_{\ell=0}^{2^{2J-1}-1} g^{2J-1}(\ell) \Psi_{2J-1, \ell}(x), \quad x \in [0, 1) \times [0, 1).$$

In den nächsten Leveln wird analog verfahren, d.h., für $j = 2J - 1, 2J - 2, \dots, 0$ werden die adaptiven Skalierungs- und Waveleträume bezüglich der Pfadvektoren $(p^{j+1}(\ell))_{\ell=0}^{2^{j+1}-1}$, nämlich

$$V_j(F) := \text{span} \left\{ h_{j, \ell} := \frac{h_{j+1, p^{j+1}(2\ell)} + h_{j+1, p^{j+1}(2\ell+1)}}{\sqrt{2}} \mid \ell = 0, \dots, 2^j - 1 \right\} \text{ und}$$

$$W_j(F) := \text{span} \left\{ \Psi_{j, \ell} := \frac{h_{j+1, p^{j+1}(2\ell)} - h_{j+1, p^{j+1}(2\ell+1)}}{\sqrt{2}} \mid \ell = 0, \dots, 2^j - 1 \right\},$$

bestimmt. Die Träger der orthogonalen Basisfunktionen $h_{j, \ell} \in V_j(F)$ bzw. $\Psi_{j, \ell} \in W_j(f)$ bestehen jeweils aus 2^{2J-j} Quadraten der Länge 2^{-j} , die normalerweise zusammenhängend angeordnet sind. Analog wie oben im ersten Level gilt auch hier, dass $V_j(F) \perp W_j(F)$ und dass $V_j(F) + W_j(F) = V_{j+1}(F)$. Die Funktionsmenge

$$\{h_{0,0}\} \cup \{\Psi_{j,\ell} \mid \ell = 0, \dots, 2^j - 1, j = 2J - 1, \dots, 0\}$$

bildet also eine orthonormale Basis von V_{2J} . Somit liefern $2J$ Level der EPWT, angewandt auf eine Funktion $F^{2J} \in V_{2J}$, die orthonormale Zerlegung

$$F^{2J} = F^0 + \sum_{j=0}^{2J-1} G^j \quad (5.7)$$

mit $G^j(x) = \sum_{\ell=0}^{2^j-1} g^j(\ell) \Psi_{j, \ell}(x)$ für $x \in [0, 1] \times [0, 1]$ und $F^0 = f^0(0)h_{0,0} = f^0(0)\chi_{[0,1] \times [0,1]}$, wobei

$$f^0(0) = \int_{[0,1] \times [0,1]} F^{2J}(x) dx = 2^{-2J} \sum_{\ell=0}^{2^{2J}-1} F^{2J}(p^{2J}(\ell)).$$

5.1.2 Anforderungen an die Pfade

Sei J eine beliebig große, feste natürliche Zahl. Wir fordern, dass unsere Pfade $(p^j(\ell))_{\ell=0}^{2^j-1}$, $j = 1, \dots, 2J$ in jedem Level „sinnvoll“ festgelegt wurden. Genauer gesagt fordern wir, dass die Indextmengen I_ℓ^j , $\ell = 0, \dots, 2^j - 1$, die folgendermaßen iterativ durch die Pfade bestimmt wurden

$$I_\ell^{2J} := \{p^{2J}(\ell)\}, \quad \ell = 0, \dots, 2^{2J} - 1,$$

$$I_\ell^j := I_{p^{j+1}(2\ell)}^{j+1} \cup I_{p^{j+1}(2\ell+1)}^{j+1} \quad \ell = 0, \dots, 2^j - 1, \quad 1 \leq j < 2J,$$

die folgenden zwei Eigenschaften erfüllen.

(i) Gebietsbedingung

Nach Anwendung des $(2J - j)$ ten Levels ($j \in \{0, \dots, 2J - 1\}$) der EPWT gibt es höchstens $C_1 K$ Indextmengen I_ℓ^j , deren Komponenten sich nicht vollständig in einer Gebietsindexmenge Γ_i^j , $i \in \{1, \dots, K\}$ befinden. Die Konstante C_1 soll von der Größe des Bildes und der Anzahl der Level unabhängig sein, d.h. also, sie darf weder von J

noch von j abhängen.

(ii) **Durchmesserbedingung**

Nach Anwendung des $(2J-j)$ ten Levels ($j \in \{0, \dots, 2J-1\}$) der EPWT besitzen fast alle Indextmengen I_ℓ^j , $\ell \in \{0, \dots, 2^j-1\}$, deren Komponenten allesamt in derselben Indextmenge Γ_i^j , $i \in \{1, \dots, K\}$ liegen, einen optimalen „Durchmesser“. Das heißt, wir fordern, dass eine von J und j unabhängige Konstante $D < \infty$ existiert, so dass

$$\text{diam } I_\ell^j = \max_{k_1, k_2 \in I_\ell^j} \|k_1 - k_2\|_2 \leq D2^{J-\frac{j}{2}} \quad (5.8)$$

für fast alle Indextmengen $I_\ell^j \subset \Gamma_i^j$ gilt und die Anzahl der Indextmengen $I_\ell^j \subset \Gamma_i^j$, die diese Durchmesserbedingung nicht erfüllen, durch eine von J und j unabhängige Konstante $C_2 < \infty$ nach oben beschränkt ist.

Die Gebietsbedingung wird erfüllt, wenn der Pfad zunächst alle Indextmengen eines Gebietes „abarbeitet“, bevor er durch eine Indextmenge aus einem anderen Gebiet läuft. Die Durchmesserbedingung sorgt dafür, dass der Durchmesser der konstruierten Indextmengen in jedem Level nur höchstens um den Faktor $\sqrt{2}$ wächst, so dass langgezogene Indextmengen vermieden werden. Zum Beispiel ist die Durchmesserbedingung erfüllt, wenn man die 2^{2J-j} Indizes, die in der Indextmenge I_ℓ^j (mit j gerade) enthalten sind, in einem Quadrat der Länge $2^{J-\frac{j}{2}}$ anordnet. In diesem Fall wäre (5.8) mit $D = \sqrt{2}$ erfüllt.

Wir bezeichnen mit

$$\Lambda^j$$

die Menge der Indizes $\ell \in \{0, \dots, 2^j-1\}$, für die die Indextmenge I_ℓ^j für ein i komplett in einem Γ_i^j enthalten ist und die Durchmesserbedingung (5.8) erfüllt. Wenn sowohl die Gebietsbedingung als auch die Durchmesserbedingung in jedem Level $2J-j$ erfüllt sind, ist die Anzahl der Indizes ℓ , die nicht in Λ^j enthalten sind, nach oben durch $C_1K + C_2$ beschränkt.

5.1.3 Beispiel

Das folgende Beispiel soll die oben eingeführten Anforderungen, die *Gebietsbedingung* und die *Durchmesserbedingung* verdeutlichen und die Effizienz der EPWT untermauern. Wir

betrachten eine stückweise hölderstetige Funktion F und ihre Interpolante F^{2^J} mit $J = 4$, siehe Abbildung 5.1(a). Unser Bild F^{2^J} besteht aus drei Regionen, und wir nehmen an, dass in jeder dieser drei Regionen die Hölderbedingung (5.3) mit einer geeigneten Konstante C erfüllt sei.

Unser Ziel ist es, den EPWT-Algorithmus so anzuwenden, dass sowohl die Gebietsbedingung als auch die Durchmesserbedingung mit kleinen Konstanten C_1 , D und C_2 erfüllt sind. Zuerst bestimmen wir einen Pfad p^8 der Länge $16 \times 16 = 256$ durch Aneinanderreihung der drei Pfade p_1^8 (Pfad durch den Hintergrund), p_2^8 (Pfad durch den Kreis) und p_3^8 (Pfad durch den schrägen Balken), siehe Abbildung 5.1(b). Der jeweils erste Index der Pfade p_1^8 , p_2^8 und p_3^8 ist mit einem kleinen Kreis markiert. In diesem Beispiel sind die Pfade p_ν^8 , $\nu = 1, 2, 3$ sogar alle vollständig zusammenhängend, d.h., es gilt $p_\nu^8(\ell + 1) \in N(p_\nu^8(\ell))$ für $\nu = 1, 2, 3$ und für $\ell = 0, \dots, \#\Gamma_\nu^4 - 2$, wobei jedes Γ_ν^4 , $\nu = 1, 2, 3$ jeweils eins der Gebiete *Hintergrund*, *Kreis* oder *schräger Balken* ist. In Abbildung 5.1(c) ist das Tiefpassbild F^7 dargestellt. Ferner wurden dort die Indexmengen I_ℓ^7 , $\ell = 0, \dots, 127$, die durch p^8 festgelegt wurden, kenntlich gemacht. Nur eine Indexmenge erfüllt die Gebietsbedingung nicht, nämlich die Indexmenge, die aus dem Index $(0, 15)$ (von p_1^8) in der rechten oberen Ecke und dem ersten Index von p_2^8 , Index $(2, 4)$, besteht, siehe Abbildung 5.1(c). Also gilt $C_1 = \frac{1}{3}$. Es existieren Indexmengen I_ℓ^7 der Form $\{m, n\}$ mit $\|m - n\|_2 = \sqrt{2}$, also ist die Durchmesserbedingung (5.8) mit $D = 1$ erfüllt und wir erhalten $C_2 = 0$. Im zweiten Level der EPWT erhält man das Tiefpassbild F^6 , siehe Abbildung 5.1(d), - auch hier sind wieder die einzelnen Indexmengen, die jetzt jeweils 4 Elemente besitzen, gekennzeichnet. Zwei Indexmengen erfüllen die Gebietsbedingung nicht, d.h. also $C_1 = \frac{2}{3}$. Am unteren Rand des Bildes gibt es zwei Indexmengen, die die Durchmesserbedingung (5.8) nur mit $D = \frac{3}{2}$ erfüllen würden - die restlichen Indexmengen, die vollständig in einem Gebiet enthalten sind, erfüllen (5.8) mit $D \leq \frac{\sqrt{5}}{2}$. Das Tiefpassbild, das nach Anwendung des dritten Levels der EPWT entsteht, ist in Abbildung 5.1(e) zu betrachten. Die Indexmengen der Größe 8 sind eingezeichnet. Wenn man den Pfadvektor so gewählt hat, dass die beiden Indexmengen, die nach dem letzten Level die Gebietsbedingung nicht erfüllt haben, nun zusammen eine neue Indexmenge bilden, so erhält man nun nur noch eine Indexmenge, die die Gebietsbedingung verletzt. Es gilt somit $C_1 = \frac{1}{3}$. Die restlichen Indexmengen erfüllen die Durchmesserbedingung (5.8) mit $D = \frac{\sqrt{18}}{2\sqrt{2}} = \frac{3}{2}$. Die letzte Abbildung 5.1(f) stellt das Tiefpassbild F^4 dar. Hier verletzt nur eine Indexmenge die Gebietsbedingung, d.h., $C_1 = \frac{1}{3}$. Die Durchmesserbedingung ist mit $D = \frac{\sqrt{45}}{4}$ für alle Indexmengen, die die Gebietsbedingung erfüllen, erfüllt. Die levelunabhängigen Konstanten sind $C_1 = \frac{2}{3}$, $D = \frac{\sqrt{45}}{4}$ und $C_2 = 0$.

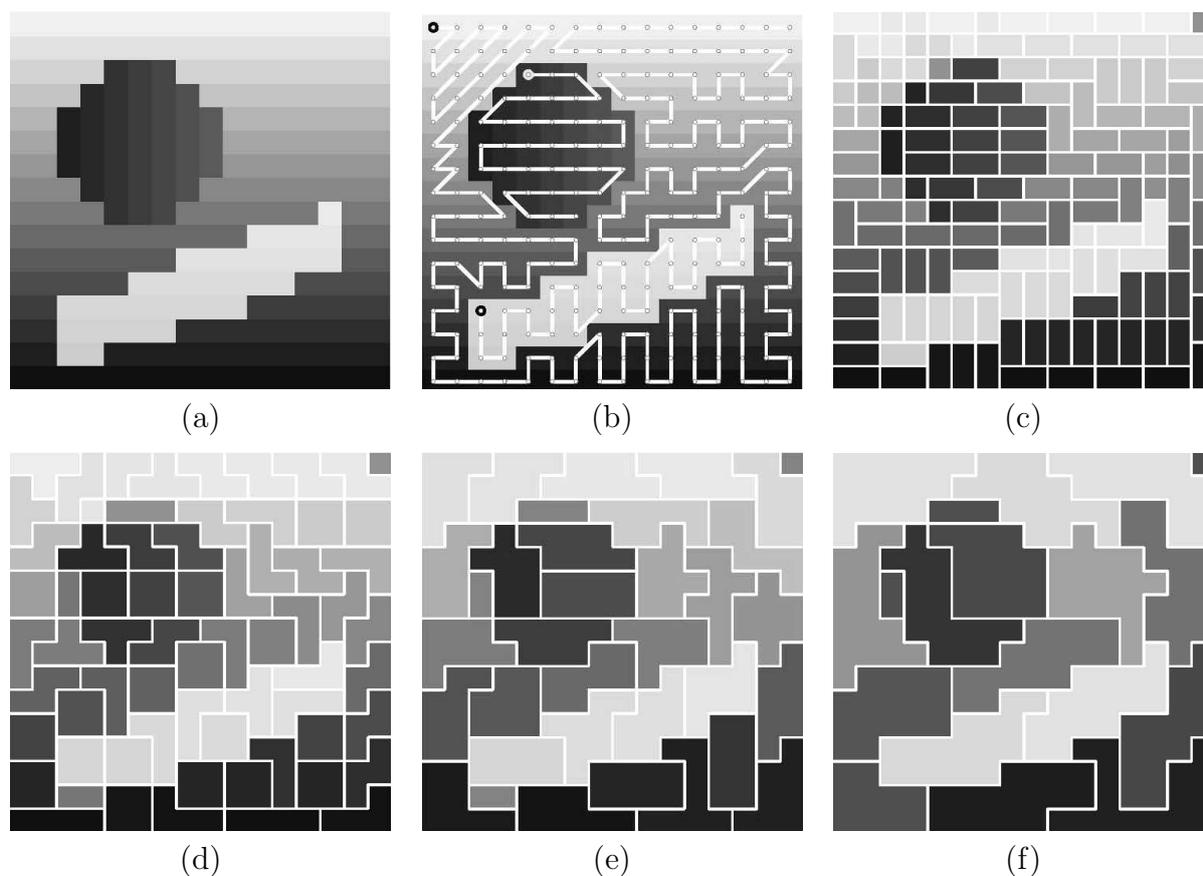


Abbildung 5.1: Anwendung der EPWT auf ein 16×16 -Bild. (a) F^8 , (b) F^8 mit p^8 eingezeichnet, (c) Tiefpassbild F^7 mit Indextmengen eingezeichnet, (d) Tiefpassbild F^6 mit Indextmengen eingezeichnet, (e) Tiefpassbild F^5 mit Indextmengen eingezeichnet, (f) Tiefpassbild F^4 mit Indextmengen eingezeichnet.

5.1.4 Algorithmus zur Bestimmung des Pfades

Nun widmen wir uns der Bestimmung eines Pfades, der die oben eingeführten Anforderungen, die Gebietsbedingung und die Durchmesserbedingung, erfüllt. Der Pfad, der bei Anwendung der relaxierten EPWT entsteht, erfüllt in der Regel (d.h. bei sinnvoll gewählter Schranke) automatisch die Gebietsbedingung, da die vorgegebene Schranke dafür sorgt, dass die Funktionswerte aufeinander folgender Pfadkomponenten nicht stark voneinander abweichen, d.h., dass der Pfad lange in einem Gebiet bleibt. Wenn man in den höheren Leveln die Mittelpunkt-EPWT anwendet, so ist auch die Durchmesserbedingung erfüllt, da auf dem Pfad benachbarte Indextmengen bei dieser Methode in der Regel nicht weit entfernte Mittelpunkte besitzen.

Nachfolgend stellen wir eine etwas andere Möglichkeit zur Pfadbestimmung vor. Wir nehmen an, dass das gegebene Bild aus einer endlichen Anzahl von glatten Gebieten besteht und wenden einen Kantendetektor auf das Bild an, um diese Gebiete Ω_i , $i = 1, \dots, K$, und die zugehörigen Indexmengen Γ_i^J zu bestimmen. Grob gesagt, funktioniert der Algorithmus wie folgt. Wir starten mit einer Indexmenge eines Gebietes, die die wenigsten Nachbarindexmengen besitzt, und nehmen nun als nächste Pfadkomponente die (zulässige) Indexmenge aus demselben Gebiet, für die gilt, dass der Durchmesser der Vereinigung der beiden Indexmengen minimal wird. Nun nehmen wir die nächste zulässige Indexmenge aus demselben Gebiet, die am wenigsten noch nicht für den Pfad verwendete Nachbarindexmengen besitzt, und wählen als nächste Pfadkomponente wieder die (zulässige) Indexmenge aus demselben Gebiet, für die gilt, dass der Durchmesser der Vereinigung der beiden Indexmengen minimal wird. Dies wiederholt man, bis keine Indexmenge des Gebiets mehr eine Nachbarindexmenge besitzt, die noch nicht für den Pfad verwendet wurde. Wenn dies eintritt, nimmt man sich das nächste Gebiet vor, in dem es noch Indexmengen mit zulässigen Nachbarindexmengen gibt, und verfährt analog. Am Ende, wenn es in jedem Gebiet nur noch zulässige Indexmengen ohne zulässige Nachbarindexmengen gibt, fasst man je zwei der noch nicht für den Pfad verwendeten Indexmengen zusammen. Eine genauere Beschreibung folgt unten.

Es seien $I_{n_1+2^j n_2}^{2^j} := \{n\}$ mit $n = (n_1, n_2)$, $0 \leq n_1 \leq 2^j - 1$, $0 \leq n_2 \leq 2^j - 1$. Die Menge aller Indexmengen im $(2^j - j + 1)$ -ten Level sei mit M^j bezeichnet. Im $(2^j - j + 1)$ -ten Level wird die Konstruktion des Pfades $p^j = (p^j(\ell))_{\ell=0}^{2^j-1}$ durch die Indexmengen I_ℓ^j wie folgt durchgeführt.

Algorithmus

Sei $i = 1$, $\ell = 0$.

1. **Wenn** $\ell = 0$, wähle $p^j(0)$ derart, dass $I_{p^j(0)}^j$ die Indexmenge ist, die in Γ_i^J enthalten ist
und die die kleinste positive Anzahl an vollständig in Γ_i^J enthaltenen Nachbarindexmengen besitzt und gehe nach 2.

Sonst: Wenn keine in $\Gamma_i^J \setminus \{I_{p^j(0)}^j, \dots, I_{p^j(2^{\ell-1})}^j\}$ enthaltene Indexmenge Nachbarindexmengen, die in $\Gamma_i^J \setminus \{I_{p^j(0)}^j, \dots, I_{p^j(2^{\ell-1})}^j\}$ enthalten sind, besitzt, prüfe:

wenn $i < K$ ist, so setze $i = i + 1$ und fange wieder vorne bei 1. an.

sonst gehe nach 3.

sonst wähle $p^j(2\ell)$ so, dass die Indexmenge $I_{p^j(2\ell)}^j \in M^j \setminus \{I_{p^j(0)}^j, \dots, I_{p^j(2\ell-1)}^j\}$ eine Teilmenge von Γ_i^J ist und die Indexmenge ist, die die kleinste positive Anzahl an Nachbarindexmengen, die in $\Gamma_{i,z}^J := \Gamma_i^J \setminus \{I_{p^j(0)}^j, \dots, I_{p^j(2\ell-1)}^j\}$ enthalten sind, besitzt und gehe nach 2.

2. Wähle $p^j(2\ell + 1)$, so dass

$$I_{p^j(2\ell+1)}^j = \operatorname{argmin}_{I_k^j \in \Gamma_{i,z}^J} \left(\operatorname{diam} \left(I_{p^j(2\ell)}^j \cup I_k^j \right) \right).$$

Setze

$$I_\ell^{j-1} = I_{p^j(2\ell)}^j \cup I_{p^j(2\ell+1)}^j$$

und $\ell = \ell + 1$ und gehe nach 1.

3. Wähle zwei Indexmengen, durch die der Pfad noch nicht geht und speichere, welche ausgewählt wurden, in $p^j(2\ell)$ und $p^j(2\ell + 1)$. Setze

$$I_\ell^{j-1} = I_{p^j(2\ell)}^j \cup I_{p^j(2\ell+1)}^j$$

und $\ell = \ell + 1$.

wenn $2\ell + 2 \neq 2^j$ fange wieder vorne bei 3 an.

sonst stoppe.

5.1.5 Approximationseigenschaften der EPWT

Für ein gegebenes $J > 0$ sei die Funktion F^{2J} die stückweise konstante Approximation des Bildes F , die (5.3) erfüllt. In diesem Abschnitt werden wir zeigen, dass die EPWT

eine optimale N -Term-Approximation von F liefert, wenn die Pfade so gewählt wurden, dass sowohl die oben im Abschnitt „Anforderungen für die Pfade“ eingeführte Gebiets- als auch die Durchmesserbedingung erfüllt ist. Zunächst zeigen wir Abschätzungen für die Skalierungskoeffizienten und Waveletkoeffizienten, die im weiteren Verlauf für den Beweis der optimalen N -Term-Approximation benötigt werden.

5.1.ii Satz. *Sei $F^{2^J} \in V_{2^J}$ ein stückweise konstantes Bild, das (5.3) für jede Indexmenge Γ_i^J , $i = 1, \dots, K$ erfüllt. Außerdem gelte für die Pfade $(p^{j+1}(\ell))_{\ell=0}^{2^{j+1}-1}$, $j = 2J-1, \dots, 0$, dass sie die Gebietsbedingung und die Durchmesserbedingung erfüllen. Sei*

$$f^{2^J}(p^{2^J}(\ell)) := 2^{-J} F^{2^J} \left(\frac{p^{2^J}(\ell)}{2^J} \right), \quad \ell = 0, \dots, 2^{2^J} - 1$$

und seien

$$f^j(\ell) = \langle \tilde{f}^{j+1}, \varphi_{j,\ell} \rangle, \quad j = 2J-1, 2J-2, \dots, 0, \quad \ell = 0, \dots, 2^j - 1$$

die Skalierungskoeffizienten, die man bei Anwendung der EPWT erhält. Dann gilt für alle $j = 2J, \dots, 0$ und $\ell \in \Lambda^{j-1}$ die Abschätzung

$$|f^j(p^j(2\ell)) - f^j(p^j(2\ell+1))| \leq 2^{\frac{\alpha}{2}} C D^\alpha 2^{-\frac{j(\alpha+1)}{2}}. \quad (5.9)$$

Hierbei ist $D > 1$ die Konstante aus der Durchmesserbedingung, C ist die Konstante aus (5.3) und α ist der Hölderexponent.

Außerdem erhält man für alle $\ell \in \{0, \dots, 2^{j-1} - 1\} \setminus \Lambda^{j-1}$ die Abschätzung

$$|f^j(p^j(2\ell)) - f^j(p^j(2\ell+1))| \leq \tilde{C} 2^{-\frac{j}{2}} \quad (5.10)$$

mit einer von J und j unabhängigen Konstante \tilde{C} .

Beweis Für $j = 2J$ folgt die Abschätzung (5.9) direkt aus der Ungleichung (5.4), denn

$$\begin{aligned} |f^{2^J}(p^{2^J}(2\ell)) - f^{2^J}(p^{2^J}(2\ell+1))| &= \frac{\left| F^{2^J} \left(\frac{p^{2^J}(2\ell)}{2^J} \right) - F^{2^J} \left(\frac{p^{2^J}(2\ell+1)}{2^J} \right) \right|}{2^J} \\ &\leq \frac{C 2^{(-J+\frac{1}{2})\alpha}}{2^J} \\ &= 2^{\frac{\alpha}{2}} C 2^{-J(\alpha+1)} \stackrel{D>1}{<} 2^{\frac{\alpha}{2}} C D^\alpha 2^{-J(\alpha+1)}. \end{aligned}$$

Es gilt

$$\begin{aligned} f^{2J-1}(\ell) &\stackrel{(5.5)}{=} 2^{-J-\frac{1}{2}} \left(F^{2J} \left(\frac{p^{2J}(2\ell)}{2^J} \right) + F^{2J} \left(\frac{p^{2J}(2\ell+1)}{2^J} \right) \right) \\ &= 2^{-J-\frac{1}{2}} \sum_{r \in I_\ell^{2J-1}} F^{2J} \left(\frac{r}{2^J} \right). \end{aligned}$$

Also folgt mit (5.3)

$$\begin{aligned} &|f^{2J-1}(p^{2J-1}(2\ell)) - f^{2J-1}(p^{2J-1}(2\ell+1))| \\ &= 2^{-J-\frac{1}{2}} \left| \sum_{n \in I_{p^{2J-1}(2\ell)}^{2J-1}} F^{2J} \left(\frac{n}{2^J} \right) - \sum_{n \in I_{p^{2J-1}(2\ell+1)}^{2J-1}} F^{2J} \left(\frac{n}{2^J} \right) \right| \\ &\leq 2^{-J-\frac{1}{2}} \left(\left| F^{2J} \left(\frac{m_1}{2^J} \right) - F^{2J} \left(\frac{n_1}{2^J} \right) \right| + \left| F^{2J} \left(\frac{m_2}{2^J} \right) - F^{2J} \left(\frac{n_2}{2^J} \right) \right| \right) \\ &\stackrel{(5.3)}{\leq} 2^{-J-\frac{1}{2}} 2C \left(\frac{2D}{2^J} \right)^\alpha = 2^{\frac{\alpha}{2}} CD^\alpha 2^{-(J-\frac{1}{2})(\alpha+1)}, \end{aligned} \tag{5.11}$$

wobei $m_1, m_2 \in I_{p^{2J-1}(2\ell)}^{2J-1}$ und $n_1, n_2 \in I_{p^{2J-1}(2\ell+1)}^{2J-1}$ gelte und wir für die letzte Ungleichung ebenfalls benutzt haben, dass zwei Elemente aus

$$I_\ell^{2J-2} = I_{p^{2J-1}(2\ell)}^{2J-1} \cup I_{p^{2J-1}(2\ell+1)}^{2J-1}$$

aufgrund der Durchmesserbedingung höchstens $2D$ voneinander entfernt sind. Man kann induktiv zeigen, dass

$$f^j(\ell) = 2^{-J} \frac{1}{2^{J-\frac{j}{2}}} \sum_{n \in I_\ell^j} F^{2J} \left(\frac{n}{2^J} \right) \tag{5.12}$$

allgemein für $j \in \{1, \dots, 2J-1\}$ gilt, denn wenn (5.12) für alle j mit $2J-1 \geq j \geq m+1$ erfüllt ist, so gilt für $j=m$, dass

$$\begin{aligned} f^j(\ell) &= \langle \tilde{f}^{j+1}, \varphi_{j,\ell} \rangle = \sum_{k=0}^{2^{j+1}-1} f^{j+1}(p^{j+1}(k)) \langle \varphi_{j+1,k}, \varphi_{j,\ell} \rangle \\ &= 2^{\frac{j+1}{2}} 2^{\frac{j}{2}} \sum_{k=0}^{2^{j+1}-1} f^{j+1}(p^{j+1}(k)) \langle \chi_{[2^{-(j+1)k}, 2^{-(j+1)(k+1)}]}, \chi_{[2^{-(j+1)(2\ell)}, 2^{-(j+1)(2\ell+2)}]} \rangle \end{aligned}$$

und somit

$$\begin{aligned}
f^j(\ell) &= 2^{j+\frac{1}{2}} \left(\frac{f^{j+1}(p^{j+1}(2\ell))}{2^{j+1}} + \frac{f^{j+1}(p^{j+1}(2\ell+1))}{2^{j+1}} \right) \\
&= 2^{-\frac{1}{2}} \left(2^{-j} \frac{1}{2^{J-\frac{j+1}{2}}} \sum_{n \in I_{p^{j+1}(2\ell)}^{j+1}} F^{2J} \left(\frac{n}{2^J} \right) + 2^{-j} \frac{1}{2^{J-\frac{j+1}{2}}} \sum_{n \in I_{p^{j+1}(2\ell+1)}^{j+1}} F^{2J} \left(\frac{n}{2^J} \right) \right) \\
&= 2^{-j} \frac{1}{2^{J-\frac{j}{2}}} \sum_{n \in I_\ell^j} F^{2J} \left(\frac{n}{2^J} \right)
\end{aligned}$$

ist. Man kann $f^j(\ell)$ also als ein gewichtetes Mittel der Funktionswerte $F^{2J}(2^{-j}n)$ mit $n \in I_\ell^j$ sehen.

Für $\ell \in \Lambda^{j-1}$ erhalten wir analog zu (5.11) wegen $\#I_\ell^j = 2^{2J-j}$ und aufgrund der Durchmesserbedingung $\text{diam } I_\ell^{j-1} \leq D2^{J-\frac{j-1}{2}}$, dass

$$\begin{aligned}
|f^j(p^j(2\ell)) - f^j(p^j(2\ell+1))| &= 2^{-2J+\frac{j}{2}} \left| \sum_{n \in I_{p^j(2\ell)}^j} F^{2J} \left(\frac{n}{2^J} \right) - \sum_{n \in I_{p^j(2\ell+1)}^j} F^{2J} \left(\frac{n}{2^J} \right) \right| \\
&\stackrel{(5.11)}{\leq} 2^{-2J+\frac{j}{2}} 2^{2J-j} C \left(\frac{D2^{J-\frac{j-1}{2}}}{2^J} \right)^\alpha \\
&= 2^{\frac{\alpha}{2}} CD^\alpha 2^{-j\frac{\alpha+1}{2}}
\end{aligned}$$

Für die Indizes $\ell \in \{0, \dots, 2^{j-1} - 1\} \setminus \Lambda^{j-1}$, für die die Indexmengen I_ℓ^j die Durchmesserbedingung nicht erfüllen, gilt wegen der Beschränktheit von F^{2J} mit der Dreiecksungleichung und $\#I_\ell^j = 2^{2J-j}$

$$\begin{aligned}
|f^j(p^j(2\ell)) - f^j(p^j(2\ell+1))| &= 2^{-2J+\frac{j}{2}} \left| \sum_{n \in I_{p^j(2\ell)}^j} F^{2J} \left(\frac{n}{2^J} \right) - \sum_{n \in I_{p^j(2\ell+1)}^j} F^{2J} \left(\frac{n}{2^J} \right) \right| \\
&\leq 2^{-2J+\frac{j}{2}} 2^{2J-j} \tilde{C} = \tilde{C} 2^{-\frac{j}{2}}.
\end{aligned}$$

□

Nun können wir die Waveletkoeffizienten, die wir nach Anwendung der EPWT erhalten haben, wie folgt abschätzen.

5.1.iii Satz. Seien $g^j(\ell) = \langle \tilde{f}^{j+1}, \psi_{j,\ell} \rangle$, $\ell = 0, \dots, 2^j - 1$, die Waveletkoeffizienten, die entstehen, wenn man den EPWT-Algorithmus auf $F^{2^j} \in V_{2^j}$ anwendet. Wir setzen voraus, dass F^{2^j} (5.3) erfüllt und dass die Pfade $(p^{j+1}(\ell))_{\ell=0}^{2^{j+1}-1}$, $j = 2J - 1, \dots, 0$, die bei Anwendung der EPWT benutzt werden, die Gebietsbedingung und die Durchmesserbedingung erfüllen. Dann erhält man für alle $j = 2J - 1, \dots, 0$ und $\ell \in \Lambda^j$ die Abschätzung

$$|g^j(\ell)| \leq \frac{1}{2} CD^\alpha 2^{-\frac{j(\alpha+1)}{2}}, \quad (5.13)$$

wobei $D > 1$ die Konstante aus der Durchmesserbedingung (5.8) ist sowie C und α die Hölderkonstante bzw. der Hölderexponent aus (5.3) sind. Außerdem gilt die Abschätzung

$$|g^j(\ell)| \leq \frac{1}{2} \tilde{C} 2^{-\frac{j}{2}} \quad (5.14)$$

für alle $\ell \in \{0, \dots, 2^j - 1\} \setminus \Lambda^j$, wobei die Konstante \tilde{C} von J und j unabhängig ist.

Beweis Für die eindimensionalen Haarwavelets gilt

$$\begin{aligned} \psi_{j,\ell} &= 2^{\frac{j}{2}} \psi(2^j \cdot -\ell) \\ &= 2^{-\frac{1}{2}} 2^{\frac{j+1}{2}} \left(\chi_{[2^{-(j+1)}(2\ell), 2^{-(j+1)}(2\ell+1)]} - \chi_{[2^{-(j+1)}(2\ell+1), 2^{-(j+1)}(2\ell+2)]} \right) \\ &= 2^{-\frac{1}{2}} (\varphi(2^{j+1} \cdot -2\ell) - \varphi(2^{j+1} \cdot -(2\ell+1))) \\ &= 2^{-\frac{1}{2}} (\varphi_{j+1,2\ell} - \varphi_{j+1,2\ell+1}). \end{aligned} \quad (5.15)$$

Außerdem folgt

$$\langle \varphi_{j+1,r}, \varphi_{j+1,\ell} \rangle = 2^{\frac{j+1}{2}} 2^{\frac{j+1}{2}} \langle \chi_{[2^{-(j+1)}r, 2^{-(j+1)}(r+1)]}, \chi_{[2^{-(j+1)}\ell, 2^{-(j+1)}(\ell+1)]} \rangle = \delta_{r,\ell}. \quad (5.16)$$

Wir erhalten nun für alle $\ell \in \Lambda^j$ folgende Abschätzung für die Waveletkoeffizienten

$$\begin{aligned} |g^j(\ell)| &= \left| \langle \tilde{f}^{j+1}, \psi_{j,\ell} \rangle \right| \stackrel{(5.15)}{=} \left| \sum_{r=0}^{2^{j+1}-1} f^{j+1}(p^{j+1}(r)) \langle \varphi_{j+1,r}, 2^{-\frac{1}{2}} (\varphi_{j+1,2\ell} - \varphi_{j+1,2\ell+1}) \rangle \right| \\ &\stackrel{(5.16)}{=} 2^{-\frac{1}{2}} |f^{j+1}(p^{j+1}(2\ell)) - f^{j+1}(p^{j+1}(2\ell+1))| \\ &\stackrel{(5.9)}{\leq} 2^{-\frac{1}{2}} 2^{\frac{\alpha}{2}} CD^\alpha 2^{-\frac{(j+1)(\alpha+1)}{2}} = \frac{1}{2} CD^\alpha 2^{-\frac{j(\alpha+1)}{2}}. \end{aligned}$$

Für alle $\ell \in \{0, \dots, 2^j - 1\} \setminus \Lambda^j$ folgt analog mit (5.10), dass

$$|g^j(\ell)| = 2^{-\frac{1}{2}} |f^{j+1}(p^{j+1}(2\ell)) - f^{j+1}(p^{j+1}(2\ell + 1))| \stackrel{(5.10)}{\leq} 2^{-\frac{1}{2}} \tilde{C} 2^{-\frac{(j+1)}{2}} = \tilde{C} 2^{-\frac{(j+2)}{2}}$$

□

Das gesamte Bild ist nach Anwendung von $2J$ Leveln der EPWT allein durch die Waveletkoeffizienten $g^j = (g^j(\ell))_{\ell=0}^{2^j-1}$, $j = 2J - 1, \dots, 0$, den Mittelwert

$$f^0(0) = 2^{-2J} \sum_{n \in I_{2J}} F^{2J}(2^{-J}n)$$

und die Pfade, die in jedem Level bestimmt werden,

$$\mathbf{p} = ((p^{2J})^T, \dots, (p^1)^T) \in \mathbb{R}^{2(2^{2J}-1)}$$

wieder rekonstruierbar. Um eine noch dünner besetztere und somit noch effizientere Darstellung des digitalen Bildes F^{2J} zu erlangen, wenden wir folgende harte Thresholdfunktion

$$s_\sigma(x) = \begin{cases} x & |x| \geq \sigma \\ 0 & |x| < \sigma \end{cases}$$

auf die Waveletkoeffizienten $g^j(\ell)$ an und setzen bis auf die N betragsmäßig größten Waveletkoeffizienten alle Waveletkoeffizienten gleich Null. Nun ist das Original F^{2J} nicht mehr exakt rekonstruierbar. Im Folgenden bezeichne F_N^{2J} die sogenannte N -Term-Approximation von F^{2J} , die entsteht, wenn man zur Rekonstruktion nur die betragsmäßig größten N Waveletkoeffizienten verwendet. Außerdem sei A_N^{2J} die Menge der Indizes (j, ℓ) , $j = 0, \dots, 2J - 1$, $\ell = 0, \dots, 2^j - 1$, für die $g^j(\ell)$ zu den größten N Waveletkoeffizienten gehört. Die Größe des Fehlers, der entsteht, wenn man nur die N betragsmäßig größten Waveletkoeffizienten zur Rekonstruktion verwendet, lässt sich wegen der Orthogonalität der verwendeten adaptiven

Haar-Wavelet-Basis in der Form

$$\begin{aligned} \epsilon_N &= \|F^{2J} - F_N^{2J}\|_2^2 \stackrel{(5.7)}{=} \left\| \left(F^0 + \sum_{j=0}^{2J-1} \sum_{\ell=0}^{2^j-1} g^j(\ell) \Psi_{j,\ell}(x) \right) - \left(F^0 + \sum_{(j,\ell) \in A_N^{2J}} g^j(\ell) \Psi_{j,\ell}(x) \right) \right\|_2^2 \\ &= \left\| \sum_{(j,\ell) \notin A_N^{2J}} g^j(\ell) \Psi_{j,\ell}(x) \right\|_2^2 = \sum_{(j,\ell) \notin A_N^{2J}} |g^j(\ell)|^2, \end{aligned}$$

darstellen und wird im folgenden Satz abgeschätzt.

5.1.iv Satz. Sei F_N^{2J} die N -Term-Approximation von F^{2J} und sei weiterhin vorausgesetzt, dass alle Voraussetzungen von Satz (5.1.iii) erfüllt seien. Dann gilt für alle $J \in \mathbb{N}$ die folgende Abschätzung

$$\epsilon_N = \|F^{2J} - F_N^{2J}\|_2^2 \leq \tilde{C} N^{-\alpha}, \quad (5.17)$$

wobei $\tilde{C} < \infty$ nicht von J abhängt.

Beweis Seien die Waveletkoeffizienten $g^j(\ell)$, $j = 0, \dots, 2J - 1$, $\ell = 0, \dots, 2^j - 1$ und der Mittelwert $g^{-1} = g^{-1}(0) := f^0(0)$ absteigend geordnet, so dass wir eine Folge $(g_\mu)_{\mu=0}^{2^{2J}-1}$ mit $|g_\mu| \geq |g_{\mu+1}|$ für $\mu = 0, \dots, 2^{2J} - 2$ erhalten.

Als Erstes zeigen wir nun, dass $\sum_{\mu=0}^{2^{2J}-1} |g_\mu|^p$ mit $\frac{1}{2} < \frac{1}{p} < \frac{\alpha+1}{2}$ unabhängig von der Wahl von $J > 1$ beschränkt ist. Um dies zu zeigen, werden wir Satz (5.1.iii) anwenden. Wie bereits im Abschnitt über die Anforderungen an die Pfadvektoren erwähnt, gibt es in jedem Level höchstens $C_1 K + C_2$ Indexmengen, die nicht gleichzeitig die Gebietsbedingung und die Durchmesserbedingung erfüllen und somit gibt es in jedem Level höchstens $C_1 K + C_2$ Waveletkoeffizienten, die (5.13) nicht erfüllen. Diese Waveletkoeffizienten seien im Folgenden als „Waveletkoeffizienten vom Typ-II“ bezeichnet, während die Waveletkoeffizienten, die (5.13) erfüllen, als „vom Typ I“ bezeichnet werden. Die Summe aller Typ-II-Waveletkoeffizienten

kann man wie folgt unabhängig von J für alle $p > 0$ abschätzen

$$\begin{aligned}
\sum_{g_\mu \text{ vom Typ II}} |g_\mu|^p &\stackrel{(5.14)}{\leq} |g^{-1}(0)|^p + (C_1K + C_2) \sum_{j=0}^{2J-1} \left(\tilde{C} 2^{-\frac{j+2}{2}} \right)^p \\
&= |g^{-1}(0)|^p + (C_1K + C_2) \left(\frac{\tilde{C}}{2} \right)^p \sum_{j=0}^{2J-1} 2^{-\frac{jp}{2}} \\
&= |g^{-1}(0)|^p + (C_1K + C_2) \left(\frac{\tilde{C}}{2} \right)^p \frac{1 - (2^{-\frac{p}{2}})^{2J}}{1 - 2^{-\frac{p}{2}}} \\
&\leq |g^{-1}(0)|^p + (C_1K + C_2) \left(\frac{\tilde{C}}{2} \right)^p \frac{1}{1 - 2^{-\frac{p}{2}}}.
\end{aligned}$$

Für die Summe aller Typ-I-Waveletkoeffizienten erhält man die Abschätzung

$$\begin{aligned}
\sum_{g_\mu \text{ vom Typ I}} |g_\mu|^p &\stackrel{(5.13)}{\leq} 2^j \left| \frac{1}{2} CD^\alpha 2^{-\frac{j(\alpha+1)}{2}} \right|^p \\
&= \left(\frac{CD^\alpha}{2} \right)^p \sum_{j=0}^{2J-1} 2^{-j \left(\frac{p(\alpha+1)}{2} - 1 \right)} \\
&= \left(\frac{CD^\alpha}{2} \right)^p \sum_{j=0}^{2J-1} \left(\left(\frac{1}{2} \right)^{\left(\frac{p(\alpha+1)}{2} - 1 \right)} \right)^j,
\end{aligned}$$

wobei 2^j die Anzahl der Waveletkoeffizienten vom Typ I ist, die es höchstens pro Level gibt, nämlich dann, wenn es in dem jeweiligen Level keine Waveletkoeffizienten vom Typ II gibt. Die Summe ist, unabhängig von J , endlich, wenn $\frac{p(\alpha+1)}{2} - 1 > 0$, d.h., $\frac{1}{p} < \frac{\alpha+1}{2}$, gilt. Somit ist die Folge $(g_\mu)_{\mu=0}^{2^{2J}-1}$ für $\frac{1}{p} < \frac{\alpha+1}{2}$ in l^p und wir können das folgende Resultat von Stechkin, siehe [22], anwenden:

Sei \mathcal{F} eine endliche oder abzählbare Indexmenge, $0 < p \leq q$ und $\alpha = (\alpha_\mu)_{\mu \in \mathcal{F}}$ eine Folge aus $l^p(\mathcal{F})$. Wenn \mathcal{F}_N die Menge der N Indizes ist, für die die zugehörigen $|\alpha_\mu|$ s am größten sind, so gilt

$$\sum_{\mu \notin \mathcal{F}_N} |\alpha_\mu|^p \leq \|\alpha\|_{l^p}^q N^{-rq} \tag{5.18}$$

mit $r := \frac{1}{p} - \frac{1}{q} \geq 0$.

In unserem Fall wählen wir $\mathcal{F} = \{0, \dots, 2^{2J} - 1\}$, $q = 2$ und p wie zuvor. Dann folgt aus (5.18) mit $rq = 2r = \frac{2}{p} - 1$ für unsere Folge $(g_\mu)_{\mu=0}^{2^{2J}-1}$, dass

$$\sum_{\mu=N+1}^{2^{2J}-1} |g_\mu|^2 \leq \left\| (g_\mu)_{\mu=0}^{2^{2J}-1} \right\|_{l^p}^2 N^{1-\frac{2}{p}}$$

gilt. Mit $p \rightarrow \frac{2}{\alpha+1}$ folgt die Behauptung. \square

5.1.v Satz. Sei $F \in L^2([0, 1) \times [0, 1))$ stückweise hölderstetig auf einer endlichen Menge von Gebieten $\{\Omega_i\}_{1 \leq i \leq K}$. Die Ränder $\partial\Omega_i$ seien stetig sowie von endlicher Länge und es gelte

$$\bigcup_{i=1}^K \Omega_i = [0, 1) \times [0, 1), \quad \Omega_i \cap \Omega_j = \emptyset \quad \text{für } i \neq j,$$

wobei jeder Abschluss $\bar{\Omega}_i$, $i = 1, \dots, K$, eine zusammenhängende Teilmenge von $[0, 1)^2$ sei. Dann existiert für jedes $\epsilon > 0$ eine ganze Zahl $J(\epsilon)$, so dass für alle $J \geq J(\epsilon)$ die N -Term-Abschätzung

$$\|F - F_N^{2J}\|_{L^2}^2 < \tilde{C}N^{-\alpha} + \epsilon$$

gilt. Hierbei sei \tilde{C} die Konstante aus (5.17).

Beweis Für ein gegebenes $J > 0$ und $n = (n_1, n_2) \in I_{2J}$ sei

$$S_n^J := [2^{-J}n_1, 2^{-J}(n_1 + 1)) \times [2^{-J}n_2, 2^{-J}(n_2 + 1))$$

Dann gilt

$$\|F - F^{2J}\|_{L^2}^2 = \sum_{n \in I_{2J}} \int_{S_n^J} |F(x) - F(2^{-J}(n))|^2 dx.$$

Wir teilen nun die Indizes $n \in I_{2J}$ in zwei Mengen auf, und zwar

$$I'_{2J} := \{n \in I_{2J} \mid S_n^J \cap \partial\Omega_i = \emptyset \text{ für alle } i \in \{1, \dots, K\}\}$$

und

$$I''_{2J} := I_{2J} \setminus I'_{2J}.$$

Aufgrund der endlichen Länge von $\bigcup_{i=1}^K \partial\Omega_i$ existiert eine von J unabhängige Konstante L , so dass $\#I''_{2J} \leq L2^J$ für alle J gilt. Für die Indizes aus I'_{2J} können wir die Hölderstetigkeit (5.1) verwenden und erhalten wegen $\#I'_{2J} \leq \#I_{2J} = 2^{2J}$ somit

$$\begin{aligned} \sum_{n \in I''_{2J}} \int_{S_n^J} |F(x) - F(2^{-J}n)|^2 dx &\leq \sum_{n \in I'_{2J}} \left(C(2^{-J}\sqrt{2})^\alpha \right)^2 \int_{S_n^J} dx \\ &= \sum_{n \in I'_{2J}} 2^{-J} 2^{-J} \left(C2^{(-J+\frac{1}{2})\alpha} \right)^2 \\ &\leq C^2 2^{(-2J+1)\alpha}. \end{aligned}$$

Weil F beschränkt ist, d.h. $|F(x)| < \tilde{C}$ für ein $C' > 0$, erhalten wir außerdem

$$\sum_{n \in I''_{2J}} \int_{S_n^J} |F(x) - F(2^{-J}n)|^2 dx < \sum_{n \in I'_{2J}} \int_{S_n^J} (2C')^2 dx = \sum_{n \in I'_{2J}} 2^{-2J} (2C')^2 < 2^{-J+2} LC'.$$

Also gilt

$$\|F - F^{2J}\|_{L^2}^2 < C^2 2^{(-2J+1)\alpha} + 2^{-J+2} LC'.$$

Für jedes $\epsilon > 0$ gibt es somit eine ganze Zahl $J(\epsilon)$, so dass $\|F - F^{2J}\|_{L^2}^2 < \epsilon$ für alle $J \geq J(\epsilon)$ gilt. Aus diesem Grunde gilt mit (5.17)

$$\|F - F_N^{2J}\|_{L^2}^2 \leq \|F - F^{2J}\|_{L^2}^2 + \|F^{2J} - F_N^{2J}\|_{L^2}^2 < \epsilon + \tilde{C}N^{-\alpha}.$$

□

Nachdem unter gewissen Voraussetzungen eine optimale N -Term-Approximation für $0 < \alpha \leq 1$ gezeigt wurde, stellt sich die Frage, ob man auch eine N -Term-Approximation für größere α beweisen kann. Der nächste Abschnitt widmet sich diesem Problem.

5.2 Approximationseigenschaften der EPWT - der Fall

$$\alpha > 0$$

Wir nehmen wieder an, dass wir von unserer bivariaten Funktion F nur die Funktionswerte kennen, die bei gleichmäßiger Abtastung von F entlang eines endlichen quadratischen Gitters entstehen. Das heißt, wir haben die Werte $\{F(\frac{n}{2^J})\}_{n \in I_{2J}}$ gegeben, wobei 2^J , $J > 1$, die Anzahl der Gitterpunkte pro Reihe (und also auch pro Spalte) ist und I_{2J} wie vorher die Indexierung der Gitterpunkte beschreibt

$$I_{2J} := \{n = (n_1, n_2) \mid 0 \leq n_1 \leq 2^J - 1, 0 \leq n_2 \leq 2^J - 1\}.$$

Außerdem bezeichne

$$\Gamma_i^{2J} := \left\{ n \in I_{2J} \mid \frac{n}{2^J} \in \Omega_i \right\} \quad \text{für } 1 \leq i \leq K$$

die Menge der Indizes aus I_{2J} , die in dem Gebiet Ω_i liegen, $1 \leq i \leq K$, so dass

$$\bigcup_{i=1}^K \Gamma_i^{2J} = I_{2J}$$

gilt und somit in Γ_i^{2J} höchstens genauso viele Elemente wie in I_{2J} , nämlich 2^{2J} , enthalten sein können, $1 \leq i \leq K$.

Wir berechnen nun anhand der gegebenen Funktionswerte $\{F(\frac{n}{2^J})\}_{n \in I_{2J}}$ eine stückweise glatte Approximation von F . Dies geschieht hier mit polyharmonischer Spline-Interpolation, die wir separat auf jede einzelne Region Ω_i , $1 \leq i \leq K$, anwenden, das heißt, wir bestimmen eine Funktion der Form

$$F^{2J}(x) := \sum_{i=1}^K \left(\sum_{n \in \Gamma_i^{2J}} c_n^i \phi_m \left(\left\| x - \frac{n}{2^J} \right\|_2 \right) + p_m^i(x) \right) \chi_{\Omega_i}(x) \quad \text{für } x \in [0, 1] \times [0, 1] \quad (5.19)$$

die die Interpolationsbedingungen

$$F^{2J} \left(\frac{n}{2^J} \right) = F \left(\frac{n}{2^J} \right) \quad \text{für alle } n \in I_{2J}$$

erfüllt. Mit χ_{Ω_i} sei hier die charakteristische Funktion von Ω_i bezeichnet, $\phi_m(r) := r^{2m} \log(r)$, $m := \max(\lfloor \alpha \rfloor, 2)$, sei eine polyharmonische Spline-Basisfunktion und p_m^i sei ein bivariates

Polynom, das höchstens den Grad m besitzt. Die Interpolierende (5.19) ist durch die Interpolationsbedingungen eindeutig bestimmt. Wir können auf diese Art und Weise Polynome bis zum Grad m approximieren, die lokale Approximationsordnung liegt bei $m+1$. Näheres ist auch in [33] und [34] zu finden.

Die Bedingung (5.1) der Hölder-Glattheit an F ist (siehe [2]) äquivalent zu folgender Bedingung: Es existiert für alle $x_0 \in \Omega_i$ ein bivariates Polynom q_α , das den Grad $[\alpha]$ besitzt (in der Regel das Taylorpolynom von F an der Stelle x_0 vom Grad $[\alpha]$) und

$$|F(x) - q_\alpha(x - x_0)| \leq C \|x - x_0\|_2^\alpha$$

für alle $x \in \Omega_i$ in einer Nachbarschaft von x_0 erfüllt, wobei $C > 0$ eine von x und x_0 unabhängige Konstante sei.

5.2.i Definition. Für die Definition des Besovraums $B_{\infty,\infty}^\alpha$ benötigen wir das *Stetigkeitsmodul N -ter Ordnung*, das (siehe [13]) durch

$$\omega_N(f, t)_\infty = \sup_{|h| \leq t} \|\Delta_h^N f\|_{L^\infty(\Omega_{h,N})},$$

mit einem Vektor $h \in \mathbb{R}^d$, dessen euklidische Norm kleiner oder gleich t ist, definiert ist. Hier ist $\Omega \subset \mathbb{R}^d$,

$$\Omega_{h,N} := \{x \in \Omega \mid x + kh \in \Omega, k = 0, \dots, N\}$$

und

$$\Delta_h^1 f(x) := f(x+h) - f(x) \quad \text{bzw.} \quad \Delta_h^N f(x) := \Delta_h^1 (\Delta_h^{N-1}) f(x).$$

5.2.ii Definition. Der *Besovraum* $B_{\infty,\infty}^\alpha$ besteht aus den Funktionen $f \in L^\infty(\Omega)$, $\Omega \subset \mathbb{R}^d$, für die

$$\{2^{\alpha j} \omega_N(f, 2^{-j})_\infty\}_{j \geq 0} \in \ell^\infty$$

für $\alpha < N$ ist.

Die Besovnorm ist durch

$$\|f\|_{B_{\infty,\infty}^{\alpha}} := \|f\|_{L^{\infty}} + |f|_{B_{\infty,\infty}^{\alpha}}, \quad |f|_{B_{\infty,\infty}^{\alpha}} := \left\| \left\{ 2^{\alpha j} \omega_N(f, 2^{-j})_{\infty} \right\}_{j \geq 0} \right\|_{\ell^{\infty}}$$

gegeben, siehe [13].

Der Hölderraum $C^{\alpha}(\Omega_i)$ der Ordnung $\alpha > 0$ mit der Norm

$$\|F\|_{C^{\alpha}(\Omega_i)} := \|F\|_{C^{[\alpha]}(\Omega_i)} + \sum_{|\mu|=m} \sup_{x \neq y} \frac{|F^{(\mu)}(x) - F^{(\mu)}(y)|}{\|x - y\|_2^{\alpha-m}}$$

ist identisch mit dem Besovraum $B_{\infty,\infty}^{\alpha}(\Omega_i)$ wenn α keine natürliche Zahl ist (siehe [54], Korollar 2.5.12, 3.4.2.(2)). Der Einbettungssatz für Besovräume liefert $B_{\infty,\infty}^{\alpha} \subset B_{2,2}^{\alpha}(\Omega_i)$ (siehe [13], Seite 163 unten) Da $B_{2,2}^{\alpha}(\Omega_i)$ äquivalent zum Sobolevraum $H^{\alpha}(\Omega_i)$ ist (siehe [13], Seite 163, Remark 3.2.4) können wir die Abschätzung

$$\|F - F^{2J}\|_{L^2(\Omega)} \leq C_F \sum_{i=1}^K h_{\Omega_i}^{\alpha} \|F\|_{B_{2,2}^{\alpha}(\Omega_i)} \quad (5.20)$$

des Interpolationsfehlers in Sobolevräumen, die in [41] gezeigt wurde, verwenden. Hierbei bezeichne

$$h_{\Omega_i} := \sup_{x \in \Omega_i} \inf_{n \in \Gamma_i^{2J}} \left\| x - \frac{n}{2^J} \right\|_2 \leq 2^{-J} \quad \text{für } 1 \leq i \leq K$$

die Distanz, die ein beliebiger Punkt aus Ω_i höchstens zu dem nächstgelegenen Punkt $\tilde{n} = \frac{n^*}{2^J}$ aus Ω_i hat, für den gilt, dass $n^* \in \Gamma_i^{2J}$ ist.

5.2.1 Eine leichte Abänderung der EPWT

Zum Beweis des gewünschten optimalen N -Term-Approximationsresultates für $\alpha > 1$ benötigen wir eine im Vergleich zu Kapitel 3 leicht abgeänderte Form der EPWT. Insbesondere werden wir für die Festlegung der Pfade in höheren Leveln keine Indexmengen definieren. Der Pfad des ersten Levels soll wieder so beschaffen sein, dass zunächst alle Elemente einer Bildgebietsindexregion Γ_i^{2J} in den Pfadvektor geschrieben werden, bevor man einen Index einer anderen Bildgebietsindexregion im Pfad speichert. Ist dies geschehen, schreibt man alle Indizes der nächsten Bildgebietsindexregion in den Pfad, usw. Genauer gesagt, nehmen wir für den ersten Eintrag $p^{2J}(0)$ des Pfadvektors einen Index aus der er-

sten Bildregion $\Gamma_1^{2^J}$. Wenn wir die ersten $n + 1$ Komponenten des Pfadvektors bestimmt haben und die $(n + 1)$ -te Komponente $p^{2^J}(n)$ in der Bildregion $\Gamma_i^{2^J}$ ($i \in \{1, \dots, K\}$) liegt, so wird die nächste Komponente $p^{2^J}(n + 1)$ aus der Menge

$$M := (N(p^{2^J}(n)) \cap \Gamma_i^{2^J}) \setminus \{p^{2^J}(0), \dots, p^{2^J}(n)\}$$

gewählt. Wenn die Menge M leer ist, so wird der Index $p^{2^J}(n + 1)$ aus der Menge

$$\Gamma_i^{2^J} \setminus \{p^{2^J}(0), \dots, p^{2^J}(n)\}$$

gewählt, sofern diese nicht auch leer ist. In diesem Fall setzen wir $p^{2^J}(n + 1)$ gleich einem Index aus der Menge $I_{2^J} \setminus \{p^{2^J}(0), \dots, p^{2^J}(n)\}$. Wenn der Pfad vollständig bestimmt ist, so wird (genau wie bei der strikten EPWT) auf die Bildwerte, angeordnet gemäß des soeben bestimmten Pfades, d.h. auf $\left(F^{2^J} \left(\frac{p^{2^J}(\ell)}{2^J}\right)\right)_{\ell=0}^{2^{2^J}-1}$, eine eindimensionale Wavelet-Transformation angewendet, die die Hochpasswerte $(d^{2^J-1}(\ell))_{\ell=0}^{2^{2^J-1}-1}$ und die Tiefpasswerte $(c^{2^J-1}(\ell))_{\ell=0}^{2^{2^J-1}-1}$ liefert. Diese Vorgehensweise sichert ab, dass die Gebietsbedingung für den Pfadvektor p^{2^J} erfüllt ist. Im zweiten Level werden im Gegensatz zur in Kapitel 3 betrachteten EPWT keine Indexmengen mehr betrachtet, das heißt, man fasst hier nicht die auf dem Pfad p^{2^J} aufeinanderfolgenden Indizes $p^{2^J}(n)$ und $p^{2^J}(n + 1)$ ($n = 0, \dots, 2^{2^J} - 2$) zu einer Indexmenge zusammen und weist dieser den Tiefpasswert $c^{2^J-1}(n)$ zu. Stattdessen weist man jedem Index $p^{2^J}(2\ell)$ jeweils einen Tiefpasswert $c^{2^J-1}(\ell)$ zu ($\ell = 0, \dots, 2^{2^J-1} - 1$). Nun sucht man einen Pfad durch diese Indizes, die in der Menge

$$\Gamma^{2^{J-1}} := \{p^{2^J}(2\ell) \mid \ell = 0, \dots, 2^{2^J-1} - 1\} = \bigcup_{i=1}^K \Gamma_i^{2^{J-1}}$$

enthalten sind. Wie im ersten Level versucht man nun ebenfalls, zuerst alle Indizes einer Bildregion „abzuarbeiten“, bevor man den Pfad durch Indizes einer anderen Bildregion legt. Wenn wir also die ersten $n + 1$ ($n \in \{0, \dots, 2^{2^J-1} - 3\}$) Einträge des neuen Pfadvektors $p^{2^{J-1}}$ bestimmt haben und der Index $p^{2^{J-1}}(n)$ in der Menge $\Gamma_i^{2^{J-1}}$ liegt ($i \in \{1, \dots, K\}$), so wählen wir den $(n + 2)$ -ten Eintrag des Pfadvektors aus der Menge aller Nachbarn von $p^{2^{J-1}}(n)$, die in $\Gamma_i^{2^{J-1}}$ liegen und durch die der Pfad noch nicht führt, d.h. aus der Menge

$$\Gamma_i^{2^{J-1}} \cap N(p^{2^{J-1}}(n)) \setminus \{p^{2^{J-1}}(0), \dots, p^{2^{J-1}}(n - 1)\}.$$

Falls diese Menge leer ist, so wird der $(n + 2)$ -te Eintrag des Pfadvektors aus der Menge

$$\Gamma_i^{2^{J-1}} \setminus \{p^{2^{J-1}}(0), \dots, p^{2^{J-1}}(n - 1)\},$$

und falls auch diese Menge leer ist, aus der Menge $\Gamma^{2^{J-1}} \setminus \{p^{2^{J-1}}(0), \dots, p^{2^{J-1}}(n - 1)\}$ gewählt. Der Begriff der Nachbarschaft von $p^{2^{J-1}}(n)$, kurz gesagt $N(p^{2^{J-1}}(n))$, sei in diesem zweiten Level wie folgt definiert. Wir sagen, dass $p^{2^{J-1}}(n + 1)$ genau dann ein Nachbar von $p^{2^{J-1}}(n)$ ist, wenn

$$\|p^{2^{J-1}}(n) - p^{2^{J-1}}(n + 1)\|_2 \leq 2$$

gilt. Auf die Einträge von $c^{2^{J-1}}$ entlang des Pfades, d.h. auf $(c^{2^{J-1}}(p^{2^{J-1}}(\ell)))_{\ell=0}^{2^{2^{J-1}}-1}$, wird nun ebenfalls eine eindimensionale Wavelet-Transformation angewandt, so dass man wiederum einen Vektor mit Hochpasskoeffizienten $(d^{2^{J-2}}(\ell))_{\ell=0}^{2^{2^{J-2}}-1}$ und einen Vektor mit Tiefpasskoeffizienten $(c^{2^{J-2}}(\ell))_{\ell=0}^{2^{2^{J-2}}-1}$ erhält. In den weiteren Leveln wird wie folgt verfahren. Wenn im vorigen Level die Tiefpasskoeffizienten c^j , mit $j \in \{2^J - 2, \dots, 1\}$ berechnet wurden, so wird, wie nachfolgend beschrieben, eine Permutation p^j dieser Tiefpasskoeffizienten berechnet und auf die Tiefpasskoeffizienten, angeordnet gemäß dieser Permutation p^j , eine eindimensionale Wavelet-Transformation angewandt, die nun wiederum Hochpasskoeffizienten $\{d^{j-1}(\ell)\}_{\ell=0}^{2^{2^{j-3}}-1}$ und Tiefpasskoeffizienten $\{c^{j-1}(\ell)\}_{\ell=0}^{2^{2^{j-3}}-1}$ liefert. Die Berechnung der Permutation erfolgt analog zu den vorangegangenen Leveln. Wir ordnen jedem $c^j(\ell)$, $\ell = 0, \dots, 2^j - 1$, den Index $p^{j+1}(2\ell)$ zu und suchen jetzt einen Pfad p^j durch die Teilmenge

$$\Gamma^j := \{p^{j+1}(2\ell) \mid \ell = 0, \dots, 2^j - 1\} \subset I_{2^j}$$

der Indizes, so dass auf diesem Pfad aufeinanderfolgende Indizes möglichst in der gleichen Bildregion Γ_i , $i = 1, \dots, K$, liegen und benachbart sind. „Benachbart“ heißen hier zwei Indizes k und l aus

$$\Gamma^j = \bigcup_{i=0}^K \Gamma_i^j := \bigcup_{i=0}^K \{p^{j+1}(2\ell) \mid \ell = 0, \dots, 2^j - 1, p^{j+2}(2\ell) \in \Gamma_i\}$$

genau dann, wenn

$$\|k - l\|_2 \leq D2^{J-\frac{j+1}{2}}$$

gilt, wobei $D \geq \sqrt{2}$ eine geeignet festgelegte Konstante ist, die möglichst klein (z.B. zwischen $\sqrt{2}$ und 2) gewählt werden sollte, bei der Bestimmung von p^{2^J} und $p^{2^{J-1}}$ hatten wir $D = \sqrt{2}$ gewählt. Wir schreiben dann $k \in N(l)$ (bzw. $l \in N(k)$). Der Pfad p^j wird mit einem beliebigen Index aus Γ^j gestartet, z.B. $p^j(0) = p^{j+1}(0)$. Wenn die Einträge $p^j(0), \dots, p^j(m)$, $m = 0, \dots, 2^j - 2$, bestimmt sind und der letzte Index $p^j(m)$ in der Bildregion Γ_i^j liegt, so wird der folgende Index $p^j(m+1)$ aus der Menge

$$(N(p^j(m)) \cap \Gamma_i^j) \setminus \{p^j(0), \dots, p^j(m)\} \quad (5.21)$$

gewählt, sofern diese nicht leer ist. Wenn die Menge leer ist, so wird der Index aus

$$\Gamma_i^j \setminus \{p^j(0), \dots, p^j(m)\}$$

gewählt, und wenn diese Menge ebenfalls leer ist, so wird der nächste Eintrag von p^j an der Stelle $m+1$ aus der Menge

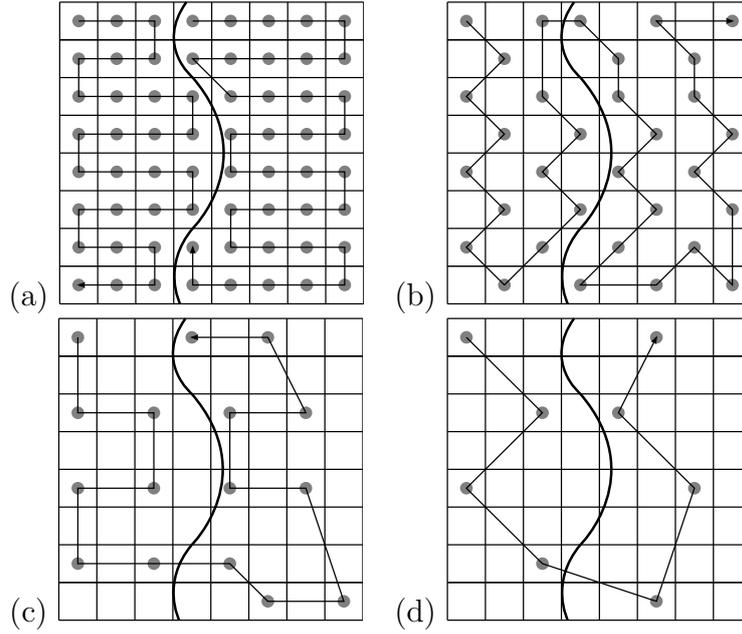
$$\Gamma^j \setminus \{p^j(0), \dots, p^j(m)\}$$

gewählt. Immer dann, wenn die Menge $(N(p^j(m)) \cap \Gamma_i^j) \setminus \{p^j(0), \dots, p^j(m)\}$ leer ist, kann es passieren, dass der Pfad unterbrochen wird (d.h., dass die Einträge $p^j(m)$ und $p^j(m+1)$ nicht benachbart sind). Die Zahl der Unterbrechungen sei durch $C_1 K$ beschränkt, wobei C_1 eine von J unabhängige Konstante ist. Somit ist auch die Zahl der betragsmäßig hohen Waveletkoeffizienten durch $C_1 K$ beschränkt, wohingegen die Anzahl der betragsmäßig hohen Waveletkoeffizienten bei der Tensorprodukt-Wavelet-Transformation in der Regel in Beziehung zu der Gesamtlänge der Ränder der glatten Gebiete steht und somit auch von dem Level j der Wavelet-Transformation abhängt.

Beispiel

Das folgende Beispiel soll diese Prozedur verdeutlichen. Das gegebene Bild bestehe aus zwei Regionen, die jeweils Hölder-glatt sind, aber möglicherweise unstetig entlang der Kurve, die den Rand zwischen den beiden Regionen beschreibt. Dieses Bild sei in diskreter Form durch $8 \times 8 = 64$ Datenpunkte gegeben, d.h. $J = 3$.

Die Pfadbestimmung des ersten Levels besteht darin, einen Pfad durch alle Datenpunkte zu finden, wobei erst die eine und danach die andere Region „abgearbeitet“ wird und wenn möglich immer der Index eines benachbarten Datenpunkts als nächster Pfadeintrag ge-

Abbildung 5.2: Pfadkonstruktion. (a) p^6 , (b) p^5 , (c) p^4 , (d) p^3 .

nommen wird, sofern dieser in derselben Region liegt. Das erste Bild stellt die Position der Grenzkurve zwischen den beiden Gebieten dar und liefert Information darüber, welche der 64 Datenpunkte welchem Gebiet zugeordnet wurden. Außerdem ist der Pfad p^6 eingezeichnet. Im zweiten Bild ist zu erkennen (graue Punkte), welche Indizes der Datenpunkte im zweiten Level betrachtet werden - nämlich jeder zweite Index auf dem Pfad p^6 , beginnend mit dem ersten. Der neue Pfad p^5 ist ebenfalls dargestellt. Bild 5.2(c) und 5.2(d) zeigen die Mengen $\Gamma^4 = \Gamma_1^4 \cup \Gamma_2^4$ bzw. $\Gamma^3 = \Gamma_1^3 \cup \Gamma_2^3$ sowie die Pfade p^4 bzw. p^3 .

In diesem Beispiel gilt für zwei auf dem Pfad aufeinanderfolgende Indizes - sofern sie in demselben Γ_i^j ($j \in \{6, 5, 4, 3\}$, $i \in \{1, 2\}$) liegen,

$$\begin{aligned} \|p^6(n+1) - p^6(n)\|_2 &\leq \sqrt{2} \leq D, & \|p^5(n+1) - p^5(n)\|_2 &\leq 2 \leq \sqrt{2}D, \\ \|p^4(n+1) - p^4(n)\|_2 &\leq \sqrt{10} \leq 2D, & \|p^3(n+1) - p^3(n)\|_2 &\leq \sqrt{10} \leq \sqrt{8}D. \end{aligned}$$

Dies gilt für $D := \frac{\sqrt{10}}{2} \approx 1,5811$. Man sieht in diesem Beispiel, dass die Indizes, die in den einzelnen Mengen Γ_i^j liegen ($j \in \{6, 5, 4, 3\}$, $i \in \{1, 2\}$), quasi-uniform verteilt sind. Dies wird dadurch erreicht, dass die Konstante D , die bestimmt, welche Indizes noch zu der Nachbarschaft eines Indizes gehören, genügend klein ist.

5.2.2 Einige Abschätzungen für das erste Level

Zunächst widmen wir uns einigen Abschätzungen (insbesondere der Waveletkoeffizienten) bezüglich des ersten Levels.

Der Pfad, der im ersten Level berechnet wird, besteht aus einer Aneinandereiung der Indizes aller gegebenen Datenpunkte. Mit anderen Worten: der Pfad durch das Bild ist in einer diskreten Form gegeben. Wir sind an einer kontinuierlichen Darstellung des Pfades interessiert. Deshalb betrachten wir eine genügend glatte parametrische Kurve $\tilde{p}^{2J}(t)$, $t \in [0, 1]$, die den Pfad p^{2J} interpoliert. Es soll gelten, dass $\tilde{p}^{2J}(\frac{\ell}{2^{2J}}) = \frac{p^{2J}(\ell)}{2^{2J}}$ für $\ell = 0, \dots, 2^{2J} - 1$ sowie $\tilde{p}^{2J}(t) \in \Omega_i$ für $t \in [\frac{\ell}{2^{2J}}, \frac{\ell+1}{2^{2J}}]$ sofern $\frac{p^{2J}(\ell)}{2^{2J}}$ und $\frac{p^{2J}(\ell+1)}{2^{2J}}$ in Ω_i liegen. Die eindimensionale Funktion \tilde{f}^{2J} beschreibe die Bildwerte entlang dieser parametrischen Kurve \tilde{p}^{2J} , d.h.

$$\tilde{f}^{2J}(t) := F^{2J}(\tilde{p}^{2J}(t)) \quad \text{für } t \in [0, 1)$$

Immer dann, wenn $p^{2J}(\ell)$ und $p^{2J}(\ell + 1)$ nicht in derselben Bildregion Γ_i^{2J} ($i \in \{1, \dots, K\}$) liegen, oder wenn $p^{2J}(\ell)$ und $p^{2J}(\ell + 1)$ nicht benachbart sind, kann es vorkommen, dass die Funktion \tilde{f}^{2J} in dem entsprechenden Teilintervall $[\frac{\ell}{2^{2J}}, \frac{\ell+1}{2^{2J}}]$ unstetig ist. Der Einfachheit halber nehmen wir an, dass der Pfad nur unterbrochen wird, wenn er die eine Bildregion verlässt und in die andere führt.

Aus dem Spursatz für Hölder- bzw. Besovräume, siehe [53], folgt, dass für $F_{|\Omega_i}^{2J} \in B_{\infty, \infty}^\alpha(\Omega_i)$ die Funktion \tilde{f}^{2J} in jedem Teilintervall von $[0, 1)$, das ausschließlich aus Zahlen der Menge $T_i := \{t \in [0, 1) \mid \tilde{p}^{2J}(t) \in \Omega_i\}$ besteht, ebenfalls Hölder-glatt von der Ordnung α ist. Wir erhalten in den Teilintervallen, in denen \tilde{f}^{2J} glatt ist, für das Stetigkeitsmodul N -ter Ordnung, das (siehe auch [13], Seite 160-161) durch

$$\omega_N(\tilde{f}^{2J}, h)_\infty := \sup_{|\tilde{h}| \leq h} \left\| \Delta_{\tilde{h}}^N \tilde{f}^{2J} \right\|_{L^\infty(T_{i,h,N})}$$

mit

$$\Delta_{\tilde{h}}^1 \tilde{f}^{2J}(t) = \tilde{f}^{2J}(t + \tilde{h}) - \tilde{f}^{2J}(t), \quad \Delta_{\tilde{h}}^N \tilde{f}^{2J}(t) = \Delta_{\tilde{h}}^1 \left(\Delta_{\tilde{h}}^{N-1} \right) \tilde{f}^{2J}(t),$$

und

$$T_{i,h,N} := \{t \in [0, 1) \mid \tilde{p}^{2J}(t + kh) \in \Omega_i, k = 0, \dots, N\}$$

definiert ist, folgende Abschätzung

$$\omega_N(\tilde{f}^{2J}, h)_\infty \leq C (2^J h)^\alpha \left\| \tilde{f}^{2J} \right\|_{B_{\infty, \infty}^\alpha}.$$

Wir betrachten nun die L^2 -Projektion f^{2J} von \tilde{f}^{2J} auf den Skalierungsraum

$$V^{2J} := \overline{\text{span} \{ \varphi_{2J, n} \mid n = 0, \dots, 2^{2J} - 1 \}},$$

wobei φ eine genügend glatte Skalierungsfunktion ist und der Abschluss bezüglich der L^2 -Norm gemeint ist. Dann ist $f^{2J} := P_{2J} \tilde{f}^{2J} = \sum_{n=0}^{2^{2J}-1} \langle f^{2J}, \tilde{\varphi}_{2J, n} \rangle \varphi_{2J, n}$ auch Hölder-glatt von der Ordnung α . Das Skalarprodukt $\langle \cdot, \cdot \rangle$ sei hier und im Folgenden, sofern nicht anders angegeben, das Standard- L^2 -Skalarprodukt. Wir erhalten nun in den Teilintervallen folgende Abschätzung

$$\begin{aligned} \left\| \tilde{f}^{2J} - f^{2J} \right\|_{L^\infty(T_{i, 2^{-2J}, N})} &= \left\| \tilde{f}^{2J} - P_{2J} \tilde{f}^{2J} \right\|_{L^\infty(T_{i, 2^{-2J}, N})} \leq C \omega_N(\tilde{f}^{2J}, 2^{-2J}) \\ &\leq C (2^{-J})^\alpha \left\| \tilde{f}^{2J} \right\|_{B_{\infty, \infty}^\alpha}, \end{aligned} \quad (5.22)$$

wobei die erste Ungleichung gemäß [13], Theorem 3.3.3, folgt und die Konstanten C nicht zwangsläufig die selben sein müssen. Insbesondere gilt somit

$$\left| \tilde{f}^{2J} \left(\frac{\ell}{2^{2J}} \right) - f^{2J} \left(\frac{\ell}{2^{2J}} \right) \right| = \left| F^{2J} \left(\frac{p^{2J}(\ell)}{2^{-J}} \right) - f^{2J} \left(\frac{\ell}{2^{2J}} \right) \right| \leq \tilde{C} 2^{-J\alpha}.$$

Nun wenden wir ein Level einer eindimensionalen Wavelet-Transformation an, um

$$f^{2J} = \sum_{\ell=0}^{2^{2J}-1} c_p^{2J}(\ell) \varphi_{2J, \ell} \quad \text{mit } c_p^{2J}(\ell) := \langle f^{2J}, \tilde{\varphi}_{2J, \ell} \rangle$$

in einen Tiefpassanteil

$$f^{2J-1} := \sum_{n=0}^{2^{2J-1}-1} c_p^{2J-1}(n) \varphi_{2J-1, n} \quad \text{mit } c_p^{2J-1}(n) := \langle f^{2J}, \tilde{\varphi}_{2J-1, n} \rangle$$

und einen Hochpassanteil

$$g^{2^J-1} := \sum_{n=0}^{2^{2^J-1}-1} d_p^{2^J-1}(n) \psi_{2^J-1,n} \quad \text{mit } d_p^{2^J-1}(n) := \langle f^{2^J}, \tilde{\psi}_{2^J-1,n} \rangle$$

zu zerlegen. Da f^{2^J} Hölder-glatt in T_i ist, gilt für $t \in T_i$ die Darstellung

$$f^{2^J}(t) = q_\alpha(t-t_0) + R(t-t_0), \quad t_0 \in \{2^{-2^J}k \mid k = 0, \dots, 2^{2^J} - 1\} \cap T_i, \quad |t - t_0| \leq 2^{-2^J},$$

wobei q_α das Taylorpolynom vom Grad $[\alpha]$ von f^{2^J} an der Entwicklungsstelle t_0 bezeichne und das Restglied R die Ungleichung $|R(t-t_0)| \leq c_\varphi 2^{-J\alpha}$ erfüllt. Wenn $\text{supp}(\tilde{\psi}_{2^J-1,n}) \in T_i$ für $i \in \{1, \dots, K\}$ ist, so kann man die Waveletkoeffizienten mit

$$\begin{aligned} |d_p^{2^J-1}(n)| &= \left| \langle q_\alpha(\cdot - t_0) + R(\cdot - t_0), \tilde{\psi}_{2^J-1,n} \rangle \right| = \left| \langle R(\cdot - t_0), \tilde{\psi}_{2^J-1,n} \rangle \right| \\ &\leq c_\varphi 2^{-J\alpha} \left\| \tilde{\psi}_{2^J-1,n} \right\|_1 = \tilde{c}_\varphi 2^{(-J+\frac{1}{2})(\alpha+1)} \end{aligned}$$

abschätzen, wobei bei dem letzten Gleichheitszeichen benutzt wurde, dass

$$\left\| \tilde{\psi}_{2^J-1,n} \right\|_1 = 2^{-J+\frac{1}{2}} \left\| \tilde{\psi} \right\|_1$$

gilt. Sei nun Λ^{2^J-1} die Menge aller $n \in \{0, \dots, 2^{2^J-1} - 1\}$, für die die obige Abschätzung gilt. Die Anzahl der $2^{2^J-1} - \#\Lambda^{2^J-1}$ Waveletkoeffizienten, für die die Abschätzung nicht gilt, hängt von der Anzahl der Unstetigkeiten von \tilde{f}^{2^J} ab und ist somit durch CK beschränkt (K war die Anzahl der verschiedenen Bildregionen des Bildes F , die Konstante C war von J unabhängig).

Wir betrachten nun die Tiefpassfunktion f^{2^J-1} und konstruieren eine bivariate Funktion F^{2^J-1} . Dies geschieht folgendermaßen: wir nehmen nur die Pfadkomponenten von p^{2^J} , die gerade Indizes haben und setzen

$$\Gamma_i^{2^J-1} := \{p^{2^J}(2n) \mid n = 0, \dots, 2^{2^J-1} - 1, \quad 2^{-J}p^{2^J}(2n) \in \Omega_i\} \quad \text{für } i = 1, \dots, K$$

sowie $\Gamma^{2^J-1} := \bigcup_{i=1}^K \Gamma_i^{2^J-1}$. Wir berechnen nun mittels polyharmonischer Splines eine Interpolierende

$$F^{2^J-1}(x) := \sum_{i=1}^K \left(\sum_{y \in \Gamma_i^{2^J-1}} c_y^i \phi_m \left(\left\| x - \frac{y}{2^J} \right\|_2 \right) + p_m^i(x) \right) \chi_{\Omega_i}(x),$$

die die Interpolationsbedingungen

$$F^{2J-1} \left(\frac{p^{2J}(2n)}{2^J} \right) = f^{2J-1} \left(\frac{2n}{2^{2J}} \right) \quad \text{für alle } n = 0, \dots, 2^{2J-1} - 1$$

erfüllt. Wir erhalten

$$\begin{aligned} \left| F^{2J} \left(\frac{p^{2J}(2n)}{2^J} \right) - F^{2J-1} \left(\frac{p^{2J}(2n)}{2^J} \right) \right| &= \left| F^{2J} \left(\tilde{p}^{2J} \left(\frac{2n}{2^{2J}} \right) \right) - f^{2J-1} \left(\frac{2n}{2^{2J}} \right) \right| \\ &= \left| \tilde{f}^{2J} \left(\frac{2n}{2^{2J}} \right) - f^{2J-1} \left(\frac{2n}{2^{2J}} \right) \right| \\ &= \left| \tilde{f}^{2J} \left(\frac{2n}{2^{2J}} \right) - P_{2J-1} \tilde{f}^{2J} \left(\frac{2n}{2^{2J}} \right) \right| \\ &\leq C 2^{(-J+1)\alpha} = CD^\alpha \left(2^{-J+\frac{1}{2}} \right)^\alpha. \end{aligned}$$

Hierbei ist $D = \sqrt{2}$ und die letzte Ungleichung folgt analog zu (5.22), da f^{2J-1} die orthogonale Projektion von \tilde{f}^{2J} auf

$$V^{2J-1} := \overline{\text{span} \{ \varphi_{2J-1,n} \mid n = 0, \dots, 2^{2J-1} - 1 \}}$$

ist (der Abschluss ist auch hier bezüglich der L^2 -Norm). F^{2J-1} ist, wie man anhand der letzten Ungleichung sieht, immer noch eine gute Approximation an F^{2J} (bzw. an F), da sich die Werte an den Interpolationspunkten von F^{2J-1} nur leicht von den Werten unterscheiden, die F^{2J} und F an diesen Stellen haben. Allerdings hat man zur Konstruktion von F^{2J-1} nur halb so viele Interpolationspunkte wie zur Konstruktion von F^{2J} verwendet. Man kann über die Verteilung der Interpolationspunkte Folgendes feststellen. Zum Einen ist zu jedem beliebigen Punkt einer Bildregion Ω_i , $i = 1, \dots, K$, der nächste Interpolationspunkt höchstens 2^{-J+1} entfernt, d.h.

$$\max_{x \in \Omega_i} \min_{y \in \Gamma_i^{2J-1}} \left| x - \frac{y}{2^J} \right| \leq \frac{2}{2^J} = D 2^{-J+\frac{1}{2}}, \quad D := \sqrt{2}.$$

Zum Anderen sind zwei Interpolationspunkte mindestens $\frac{1}{2^J}$ voneinander entfernt, d.h.

$$\min_{y_1, y_2 \in \Gamma_i^{2J-1}} \left| \frac{y_1}{2^J} - \frac{y_2}{2^J} \right| \geq \frac{1}{2^J};$$

dies folgt aus der Festlegung des Gitters I_{2^J} . Somit erhalten wir mit (5.20)

$$\|F^{2^J} - F^{2^{J-1}}\|_{L^2(\Omega_i)} \leq C \left(2^{-J+\frac{1}{2}}\right)^\alpha \quad \text{für alle } i = 1, \dots, K.$$

5.2.3 Anforderungen an die Pfadvektoren

Wie schon für den Fall $0 \leq \alpha < 1$ müssen auch hier die Pfade gewisse Anforderungen erfüllen. Im Grunde sind diese Anforderungen die gleichen wie im zuvor behandelten Fall. Da wir hier allerdings in den späteren Leveln die Pfade nicht mehr durch Mengen aus Indizes legen sondern durch einzelne unregelmäßig verteilte Indizes, müssen die Bedingungen hier etwas anders formuliert werden. Die **Gebietsbedingung** besteht darin, dass es nur sehr selten passieren soll, dass $p^j(\ell)$ und $p^j(\ell + 1)$ zu verschiedenen Bildregionen Γ_i^j und Γ_k^j gehören - sie ist also erfüllt, wenn zuerst alle Indizes einer Bildregion Ω_i „abgearbeitet“ werden, bevor man die Indizes der nächsten Bildregion abarbeitet. Genauer gesagt, es wird gefordert, dass der Pfad p^j in jedem Level j so gewählt werden soll, dass er höchstens $C_1 K$ Sprünge besitzt, die dadurch entstehen, dass $p^j(\ell)$ und $p^j(\ell + 1)$ keine Nachbarn bzw. nicht in derselben Bildregion sind.

Die **Durchmesserbedingung** besteht darin, dass man fordert, dass fast alle Pfadkomponenten die Abschätzung

$$\left\| \frac{p^j(\ell)}{2^J} - \frac{p^j(\ell + 1)}{2^J} \right\|_2 \leq D 2^{-\frac{j}{2}} \quad (5.23)$$

erfüllen, wobei die Konstante D von J und j unabhängig ist und die Anzahl der Komponenten von p^j , für die die Durchmesserbedingung nicht gilt, durch eine Konstante C_2 beschränkt ist, die nicht von J oder j abhängt. Die Durchmesserbedingung sorgt dafür, dass die Indizes aus Γ^j , skaliert um $\frac{1}{2^J}$, quasi-uniform über das gesamte Gebiet Ω verteilt sind, so dass der Abstand von einem beliebigen Punkt aus einer Bildregion Ω_i zu einem um $\frac{1}{2^J}$ skalierten Index aus Γ_i^j höchstens $D 2^{-\frac{j}{2}}$ beträgt, d.h.

$$\max_{x \in \Omega_i} \min_{y \in \Gamma_i^j} \left\| x - \frac{y}{2^J} \right\|_2 \leq D 2^{-\frac{j}{2}}.$$

Wenn man die oben beschriebene Strategie zur Pfadkonstruktion anwendet, so erhält man automatisch einen Pfad, der diese Bedingungen erfüllt.

5.2.4 Abschätzungen für höhere Level der EPWT

Die Abschätzungen für die höheren Level der EPWT funktionieren analog zu den Abschätzungen des ersten Levels.

Wir beginnen mit der Interpolierenden

$$F^{j+1}(x) := \sum_{i=1}^K \left(\sum_{y \in \Gamma_i^{j+1}} c_y^i \phi_m \left(\left\| x - \frac{y}{2^j} \right\|_2 \right) + p_m^i(x) \right) \chi_{\Omega_i}(x),$$

die die Interpolationsbedingungen

$$F^{j+1} \left(\frac{p^{j+2}(2n)}{2^j} \right) = f^{j+1} \left(\frac{2n}{2^{j+2}} \right) \quad \text{für alle } n = 0, \dots, 2^{j+1} - 1$$

erfüllt. Hier ist wiederum ϕ_m die vorher definierte polyharmonische Splinebasisfunktion und p_m^i ein bivariates Polynom, das höchstens den Grad $m := \max([\alpha], 2)$ besitzt. Nun bestimmen wir zunächst einen geeigneten Pfad p^{j+1} durch die Menge

$$\Gamma^{j+1} = \{ p^{j+2}(2n) \mid n = 0, \dots, 2^{j+1} - 1 \},$$

wobei zu jedem $p^{j+2}(2n)$ ein Datenwert $F^{j+1} \left(\frac{p^{j+2}(2n)}{2^j} \right)$, $n = 0, \dots, 2^{j+1} - 1$ gehört. Dieser Pfad soll so gewählt sein, dass sowohl die Durchmesser- als auch die Gebietsbedingung erfüllt sind. Als Nächstes betrachten wir eine genügend glatte parametrische Kurve $\tilde{p}^{j+1}(t)$, $t \in [0, 1)$ durch die Ebene, die p^{j+1} interpoliert und die Interpolationsbedingungen $\tilde{p}^{j+1} \left(\frac{\ell}{2^{j+1}} \right) = \frac{p^{j+1}(\ell)}{2^j}$ für $\ell = 0, \dots, 2^{j+1} - 1$ erfüllt und derart verläuft, dass $\tilde{p}^{j+1}(t) \in \Omega_i$ ist, wenn $t \in \left[\frac{\ell}{2^{j+1}}, \frac{\ell+1}{2^{j+1}} \right]$ und $\frac{p^{j+1}(\ell)}{2^j}$ und $\frac{p^{j+1}(\ell+1)}{2^j}$ in demselben Gebiet Ω_i liegen.

Die eindimensionale Restriktion von F^{j+1} entlang \tilde{p}^{j+1} wird nun folgendermaßen definiert

$$\tilde{f}^{j+1}(t) := F^{j+1} \left(\tilde{p}^{j+1}(t) \right) \quad t \in [0, 1).$$

Für die Teilintervalle, in denen $\tilde{f}^{j+1}(t)$ Hölder-glatt ist, gilt analog zum ersten Level

$$\omega_N(\tilde{f}^{j+1}, h)_\infty := \sup_{|\tilde{h}| \leq h} \left\| \Delta_{\tilde{h}}^N \tilde{f}^{j+1} \right\|_{L^\infty(T_{i,h,N})} \leq \tilde{c} (2^j h)^\alpha \quad (5.24)$$

mit

$$T_{i,h,N} := \{ t \in [0, 1) \mid \tilde{p}^{j+1}(t) \in \Omega_i, \tilde{p}^{j+1}(t + kh) \in \Omega_i, k = 0, \dots, N \}.$$

Wir betrachten nun die L^2 -Projektion

$$f^{j+1} := P_{j+1}\tilde{f}^{j+1} = \sum_{n=0}^{2^{j+1}-1} c_p^{j+1}(n)\varphi_{j+1,n} \quad \text{mit } c_p^{j+1}(n) := \langle \tilde{f}^{j+1}, \tilde{\varphi}_{j+1,n} \rangle,$$

die \tilde{f}^{j+1} auf den Skalierungsraum

$$V^{j+1} := \overline{\text{span} \{\varphi_{j+1,n} \mid n = 0, \dots, 2^{j+1} - 1\}},$$

projiziert, wobei der Abschluss bezüglich des Raumes L^2 gemeint ist und φ genau wie zuvor eine glatte Skalierungsfunktion ist. Nun erhalten wir

$$\begin{aligned} \left\| \tilde{f}^{j+1} - f^{j+1} \right\|_{L^\infty(T_{i, \frac{1}{2^{j+1}}, N})} &= \left\| \tilde{f}^{j+1} - P_{j+1}\tilde{f}^{j+1} \right\|_{L^\infty(T_{i, \frac{1}{2^{j+1}}, N})} \\ &\leq C\omega_N(\tilde{f}^{j+1}, \frac{2^{J-\frac{j+1}{2}}}{2^{2J}}) \stackrel{(5.24)}{\leq} C2^{-\frac{(j+1)\alpha}{2}}, \end{aligned}$$

wobei die erste Ungleichung folgt, da die Durchmesserbedingung

$$\left\| p^{j+1}(\ell) - p^{j+1}(\ell + 1) \right\|_2 \leq D2^{J-\frac{j+1}{2}} \quad (5.25)$$

erfüllt ist. Die Konstanten C müssen nicht zwangsläufig dieselben sein.

Wir zerlegen nun f^{j+1} in

$$f^{j+1} = \sum_{\ell=0}^{2^{j+1}-1} c_p^{j+1}(\ell)\varphi_{j+1,\ell} = f^j + g^j = \sum_{n=0}^{2^j-1} c_p^j(n)\varphi_{j,n} + \sum_{n=0}^{2^j-1} d_p^j(n)\psi_{j,n}.$$

Aufgrund der Hölder-Glattheit von f^{j+1} in den Teilintervallen

$$T_i := \{t \in [0, 1) \mid \tilde{p}^{2^j}(t) \in \Omega_i\}$$

gilt

$$f^{j+1}(t) = q_\alpha(t - t_0) + R(t - t_0)$$

mit

$$|R(t - t_0)| \leq c_\varphi |t - t_0|^\alpha \stackrel{(5.25)}{\leq} c_\varphi \left(\frac{D2^{\frac{j-(j+1)}{2}}}{2^J} \right)^\alpha = c_\varphi D^\alpha 2^{-\frac{(j+1)\alpha}{2}}$$

für $t, t_0 \in T_i$. Somit können die Waveletkoeffizienten $d_p^j(n)$, für die es ein $i \in \{1, \dots, K\}$ gibt, so dass $\text{supp } \tilde{\psi}_{j,n} \subset T_i$ ist, durch

$$\begin{aligned} |d_p^j(n)| &= \left| \langle R(t - t_0), \tilde{\psi}_{j,n} \rangle \right| \leq c_\varphi D^\alpha 2^{-\frac{(j+1)\alpha}{2}} \left\| \tilde{\psi}_{j,n} \right\|_1 \\ &= c_\varphi D^\alpha 2^{-\frac{(j+1)\alpha}{2}} 2^{-\frac{j}{2}} \left\| \tilde{\psi} \right\|_1 \leq \tilde{c}_\varphi D^\alpha 2^{-\frac{j(\alpha+1)}{2}} \end{aligned} \quad (5.26)$$

Sei Λ^j die Menge, die die Indizes $n \in \{0, \dots, 2^j - 1\}$ enthalte, für die $d_p^j(n)$ die obige Ungleichung erfülle. Die Anzahl der restlichen Waveletkoeffizienten $2^j - \#\Lambda^j$, für die es kein i gibt, so dass $\text{supp } \tilde{\psi}_{j,n} \subset T_i$, ist nach oben durch eine von J und j unabhängige Konstante beschränkt.

Wir erhalten mittels polyharmonischer Splines die Interpolierende

$$F^j(x) := \sum_{i=1}^K \left(\sum_{y \in \Gamma_i^j} c_y^i \phi_m \left(\left\| x - \frac{y}{2^j} \right\|_2 \right) + p_m^i(x) \right) \chi_{\Omega_i(x)},$$

mit

$$\Gamma_i^j := \left\{ p^{j+1}(2n) \mid n = 0, \dots, 2^j - 1, \frac{p^{j+1}(2n)}{2^J} \in \Omega_i \right\}, \quad \Gamma^j := \bigcup_{i=1}^k \Gamma_i^j$$

aus den Interpolationsbedingungen

$$F^j \left(\frac{p^{j+1}(2n)}{2^J} \right) = f^j \left(\frac{n}{2^j} \right) \quad \text{für alle } n = 0, \dots, 2^j - 1.$$

Nun können wir folgendermaßen abschätzen

$$\begin{aligned} \left| F^{j+1} \left(\frac{p^{j+1}(2n)}{2^J} \right) - F^j \left(\frac{p^{j+1}(2n)}{2^J} \right) \right| &= \left| F^{j+1} \left(\tilde{p}^{j+1} \left(\frac{2n}{2^{j+1}} \right) \right) - f^j \left(\frac{n}{2^j} \right) \right| \\ &= \left| \tilde{f}^{j+1} \left(\frac{n}{2^j} \right) - P_j \tilde{f}^{j+1} \left(\frac{n}{2^j} \right) \right| \\ &\leq C 2^{-\frac{(j+1)\alpha}{2}}, \end{aligned} \quad (5.27)$$

so dass wir

$$\begin{aligned} \left| F^{2J} \left(\frac{p^{j+1}(2n)}{2^J} \right) - F^j \left(\frac{p^{j+1}(2n)}{2^J} \right) \right| &\leq \sum_{\nu=j}^{2J-1} \left| F^{\nu+1} \left(\frac{p^{j+1}(2n)}{2^J} \right) - F^\nu \left(\frac{p^{j+1}(2n)}{2^J} \right) \right| \\ &\leq C \sum_{\nu=j}^{2J-1} 2^{-\frac{(\nu+1)\alpha}{2}} \leq \frac{2^{-\frac{(j+1)\alpha}{2}}}{1 - 2^{-\frac{\alpha}{2}}} \end{aligned}$$

erhalten.

5.2.5 Beweis zur N-Term-Approximation für $\alpha > 0$

Die Überlegungen des letzten Abschnitts liefern uns folgenden Satz:

5.2.iii Satz. Seien $d_p^j(\ell) = \langle f^{j+1}, \tilde{\psi}_{j,\ell} \rangle$, $\ell = 0, \dots, 2^j - 1$, $j = 2J - 1, \dots, 0$, die Waveletkoeffizienten, die man erhält, wenn man die EPWT in der zuvor beschriebenen Art und Weise auf ein $F \in L^2([0, 1) \times [0, 1))$ anwendet. Außerdem sei F in jedem Gebiet Ω_i , $1 \leq i \leq K < \infty$, Hölder-glatt von der Ordnung α . Für die Gebiete Ω_i , $1 \leq i \leq K < \infty$, gelte, dass sie einen genügend glatten, endlich langen Lipschitzrand $\partial\Omega_i$ besitzen und ihr Abschluss $\bar{\Omega}_i$ eine zusammenhängende Teilmenge des $[0, 1] \times [0, 1]$ ist, sowie

$$\bigcup_{i=1}^K \Omega_i = [0, 1) \times [0, 1), \quad \Omega_{i_1} \cap \Omega_{i_2} = \emptyset \text{ für } i_1 \neq i_2$$

erfüllt. Desweiteren sei gefordert, dass die Pfade $(p^{j+1}(\ell))_{\ell=0}^{2^{j+1}-1}$, $j = 2J - 1, \dots, 0$, die bei der Anwendung der EPWT entstehen, die Gebietsbedingung und die Durchmesserbedingung, die im vorletzten Abschnitt beschrieben wurden, erfüllen.

Dann gilt für alle $j = 2J - 1, \dots, 0$ und $\ell \in \Lambda^j$ die Abschätzung

$$|d_p^j(\ell)| \leq CD^\alpha 2^{-\frac{j(\alpha+1)}{2}}, \quad (5.28)$$

wobei $D > 1$ die Konstante aus der Durchmesserbedingung (5.23) ist und α der Hölderexponent von F . Die Konstante C hängt von der verwendeten Waveletbasis und der Hölderkonstante aus (5.1) ab. Außerdem lassen sich dann die Waveletkoeffizienten $d_p^j(\ell)$ für die restlichen $\ell \in \{0, \dots, 2^j - 1\} \setminus \Lambda^j$ wie folgt abschätzen

$$|d_p^j(\ell)| \leq \tilde{C} 2^{-\frac{j}{2}}.$$

Die Konstante \tilde{C} ist von J und j unabhängig.

Beweis Der Beweis von (5.28) folgt aus den obigen Betrachtungen, die uns zu (5.26) geführt haben. Für die restlichen Waveletkoeffizienten, für die $\ell \in \{0, \dots, 2^j - 1\} \setminus \Lambda^j$ gilt, erhalten wir die Abschätzung

$$|d_p^j(\ell)| = \left| \langle f^{j+1}, \tilde{\psi}_{j,\ell} \rangle \right| \leq C \left\| \tilde{\psi}_{j,\ell} \right\|_1 = C 2^{-\frac{j}{2}} \left\| \tilde{\psi} \right\|_1 \leq \tilde{C} 2^{-\frac{j}{2}},$$

da wir annehmen können, dass F^j beschränkt ist. \square

Bevor wir die optimale N -Term-Approximation der EPWT zeigen, folgen nun erst einmal einige weitere Betrachtungen. Die Anwendung der EPWT liefert den Vektor

$$\mathbf{d}_p = \left((d_p^{2J-1})^T, \dots, d_p^0, d_p^{-1} \right)^T,$$

der die Waveletkoeffizienten $d_p^j = (d_p^j(\ell))_{\ell=0}^{2^j-1}$ für $j = 0, \dots, 2J - 1$ und den Mittelwert

$$d_p^{-1} = d_p^{-1}(0) := f^0(0) = \frac{1}{2^{2J}} \sum_{n \in I_{2J}} F^{2J} \left(\frac{n}{2^J} \right)$$

enthält sowie die Pfade, die in jedem Iterationsschritt entstehen, nämlich

$$\mathbf{p} = \left((p^{2J})^T, \dots, (p^1)^T \right)^T \in \mathbb{R}^{2(2^{2J}-1)}.$$

Mit Hilfe der Information über die Pfade \mathbf{p} sowie den Waveletkoeffizienten und dem Mittelwert, \mathbf{d}_p , lässt sich eindeutig ein Bild F_{rek}^{2J} rekonstruieren, wobei F_{rek}^{2J} die polyharmonische Spline-Interpolierende ist, die die Interpolationsbedingungen

$$F_{rek}^{2J} \left(\frac{p^{2J}(n)}{2^J} \right) = f^{2J} \left(\frac{n}{2^{2J}} \right), \quad n = 0, \dots, 2^{2J} - 1$$

erfüllt. Analog sind alle „Zwischenergebnisse“, d.h. die polyharmonischen Spline-Interpolierenden eindeutig durch die Interpolationsbedingungen

$$F_{rek}^j \left(\frac{p^j(n)}{2^j} \right) = f^j \left(\frac{n}{2^j} \right), \quad n = 0, \dots, 2^j - 1$$

festgelegt. Wenn man nämlich das $(2J - j)$ -te Level der EPWT betrachtet, so stellt man

fest, dass die Skalierungskoeffizienten $c_p^j(n) = \langle \tilde{f}^{j+1}, \tilde{\varphi}_{j,n} \rangle$ und die Waveletkoeffizienten $d_p^j = \langle \tilde{f}^{j+1}, \tilde{\psi}_{j,n} \rangle$ sowohl f^j als auch g^j und somit auch f^{j+1} eindeutig bestimmen. Also folgt

$$\begin{aligned} \left| F^{j+1} \left(\frac{p^{j+1}(n)}{2^J} \right) - F_{rek}^{j+1} \left(\frac{p^{j+1}(n)}{2^J} \right) \right| &= \left| \tilde{f}^{j+1} \left(\frac{n}{2^{j+1}} \right) - f^{j+1} \left(\frac{2n}{2^{j+1}} \right) \right| \\ &= \left| \tilde{f}^{j+1} \left(\frac{n}{2^{j+1}} \right) - P_{j+1} \tilde{f}^{j+1} \left(\frac{n}{2^{j+1}} \right) \right| \\ &\leq C \left(2^{-\frac{j+1}{2}} \right)^\alpha \end{aligned}$$

für $n = 0, \dots, 2^{j+1} - 1$. Um eine dünner besetztere Darstellung des digitalen Bildes F bzw. F^{2J} zu bekommen, setzen wir die betragsmäßig kleinen Waveletkoeffizienten zu Null, d.h., wir setzen

$$s_\sigma(d_p^j(\ell)) = \begin{cases} d_p^j(\ell) & \text{für } |d_p^j(\ell)| \geq \sigma \\ 0 & \text{für } |d_p^j(\ell)| < \sigma, \end{cases}$$

wobei $\sigma > 0$ eine im Vorhinein festgelegte Konstante ist. Wir betrachten nun den Fehler, der entsteht, wenn man nur die N betragsmäßig größten Waveletkoeffizienten zur Approximation von F^{2J} verwendet. Sei S_N^{2J} die Menge der Indizes (j, ℓ) dieser N betragsmäßig größten Waveletkoeffizienten und $F_{N,rek}^{2J}$ die polyharmonische Spline-Interpolierende, die die Interpolierungsbedingungen

$$F_{N,rek}^{2J} \left(\frac{p^{2J}(n)}{2^J} \right) = f_N^{2J} \left(\frac{n}{2^{2J}} \right) \quad \text{für } n = 0, \dots, 2^{2J} - 1$$

erfüllt, wobei für die Berechnung von $f_N^{2J} = \sum_n c_{p,N}^{2J}(n) \varphi_{2J,n}$ nur die betragsmäßig größten N Waveletkoeffizienten verwendet wurden. Aufgrund der Stabilität der Waveletbasis können wir den Fehler, der dadurch entstanden ist, dass wir nicht alle Waveletkoeffizienten zur Bestimmung von $F_{N,rek}$ verwendet haben, folgendermaßen abschätzen

$$\varepsilon_N = \|F^{2J} - F_{N,rek}^{2J}\|_{L^2(\Omega)}^2 \leq C \sum_{(j,\ell) \notin S_N^{2J}} |d_p^j(\ell)|^2.$$

Somit erhalten wir auch für den Fall $\alpha > 1$ die folgenden Approximationsresultate.

5.2.iv Satz. Sei F_N^{2J} die N -Term-Approximation von F^{2J} , die wie oben beschrieben konstruiert wurde. Außerdem seien die Voraussetzungen von Satz 5.2.iii erfüllt. Dann gilt für alle $J \in \mathbb{N}$, dass

$$\varepsilon_N = \|F^{2J} - F_N^{2J}\|_2^2 \leq \tilde{C}N^{-\alpha}$$

erfüllt ist, wobei $\tilde{C} < \infty$ nicht von J unabhängig ist.

Beweis Der Beweis kann analog zu dem Beweis von Satz 5.1.iv durchgeführt werden. \square

5.2.v Satz. Sei $F \in L^2([0, 1) \times [0, 1))$ eine stückweise Hölder-glatte Funktion auf einer endlichen Menge von Gebieten $\{\Omega_i\}_{1 \leq i \leq K}$. Die Ränder $\partial\Omega_i$ seien stetig sowie von endlicher Länge und es gelte

$$\bigcup_{i=1}^K \Omega_i = [0, 1) \times [0, 1), \quad \Omega_i \cap \Omega_j = \emptyset \quad \text{für } i \neq j,$$

wobei jeder Abschluss $\overline{\Omega}_i$, $i = 1, \dots, K$, eine zusammenhängende Teilmenge von $[0, 1)^2$ sei. Dann existiert für jedes $\varepsilon > 0$ eine ganze Zahl $J(\varepsilon)$, sodass für alle $J \geq J(\varepsilon)$ die N -Term-Abschätzung

$$\|F - F_N^{2J}\|_{L^2}^2 < \tilde{C}N^{-\alpha} + \varepsilon$$

gilt.

Beweis Der Beweis folgt aus Satz 5.2.iv und (5.20). \square

Literaturverzeichnis

- [1] Sergio Amat, Francesc Arandiga, Albert Cohen, Rosa Donat, Gregori Garcia und Markus von Oehsen. Data compression with eno schemes - a case study. *Appl. Comput. Harmon. Anal.*, 11:273–288, 2002.
- [2] Patrik Andersson. Characterization of pointwise hölder regularity. *Appl. Comput. Harmon. Anal.*, 4(4):429–443, 1997.
- [3] Francesc Arandiga, Albert Cohen, Manuel Doblas und Basarab Matei. Edge adapted nonlinear multiscale transforms for compact image representation. *IEEE Int. Conf. Image Proc.*, 2003.
- [4] Francesc Arandiga, Albert Cohen, Rosa Donat und Nira Dyn. Interpolation and approximation of piecewise smooth functions. *SIAM J. Numer. Anal.*, 43:41–57, 2005.
- [5] Francesc Arandiga, Albert Cohen, Rosa Donat, Nira Dyn und Basarab Matei. Approximation of piecewise smooth functions and images by edge-adapted (eno-ea) nonlinear multiresolution techniques. *Appl. Comput. Harm. Anal.*, 24:225–250, 2008.
- [6] Xavier Bresson und Tony F. Chan. Non-local unsupervised variational image segmentation models. Technical Report 08-67, UCLA CAM, 2008.
- [7] Emmanuel J. Candès, Laurent Demanet, David L. Donoho und Lexing Ying. Fast discrete curvelet transforms. *Multiscale Model. Simul.*, 5:861–899, 2006.
- [8] Emmanuel J. Candès und David L. Donoho. Ridgelets: the key to high-dimensional intermittency? *R. Soc. Lond. Philos. Trans. Ser. A Math. Phys. Eng. Sci.*, 357:2495–2509, 1999.
- [9] Emmanuel J. Candès und David L. Donoho. Curvelets and curvilinear integrals. *J. Approx. Theory*, 113:59–90, 2000.

- [10] Emmanuel J. Candès und David L. Donoho. New tight frames of curvelets and optimal representations of objects with piecewise singularities. *Comm. Pure Appl. Math.*, 57:219–266, 2004.
- [11] Tony F. Chan, Selim Esedoglu und Mila Nikolova. Algorithms for finding global minimizers of denoising and segmentation models. *SIAM J. Appl. Math.*, 66:1632–1648, 2006.
- [12] Roger L. Claypoole, Geoffrey M. Davis, Wim Sweldens und Richard G. Baraniuk. Nonlinear wavelet transforms for image coding via lifting. *IEEE Trans. Image Process.*, 12:1449–1459, 2003.
- [13] Albert Cohen. *Numerical Analysis of Wavelet Methods*, volume 32 of *Studies in Mathematics and its Applications*. Elsevier, Amsterdam, 2003.
- [14] Albert Cohen, Wolfgang Dahmen, Ingrid Daubechies und Ronald A. DeVore. Harmonic analysis of the space bv . *Rev. Mat. Iberoam.*, 19(1):235–263, 2003.
- [15] Albert Cohen, Ingrid Daubechies und Jean-Christophe Feauveau. Biorthogonal bases of compactly supported wavelets. *Commun. Pure Appl. Math.*, 45(5):485–560, 1992.
- [16] Albert Cohen, Ronald A. DeVore, Pencho Petrushev und Hong Xu. Nonlinear approximation and the space $bv(\mathbb{R}^2)$. *Amer. J. Math.*, 121:587–628, 1999.
- [17] Albert Cohen und Basarab Matei. Compact representation of images by edge adapted multiscale transforms. Proc. IEEE Int. Conf. on Image Proc. (ICIP), pages 8–11, Thessalonki, Greece, 2001.
- [18] Ingrid Daubechies. *Ten Lectures on Wavelets*. SIAM, Philadelphia, 1992.
- [19] Shai Dekel und Dany Leviatan. Adaptive multivariate approximation using binary space partitions and geometric wavelets. *SIAM J. Numer. Anal.*, 43:707–732, 2006.
- [20] Laurent Demanet und Lexing Ying. Wave atoms and sparsity of oscillatory patterns. *Appl. Comput. Harmon. Anal.*, 23:368–387, 2007.
- [21] Laurent Demaret, Nira Dyn und Armin Iske. Image compression by linear splines over adaptive triangulations. *Signal Processing*, 86:1604–1616, 2006.
- [22] Ronald A. DeVore. Nonlinear approximation. *Acta Numerica*, 7:51–150, 1998.

-
- [23] Minh N. Do und Martin Vetterli. The contourlet transform: an efficient directional multiresolution image representation. *IEEE Trans. Image Process.*, 14:2091–2106, 2005.
- [24] David L. Donoho. Wedgelets: nearly minimax estimation of edges. *Ann. Stat.*, 27:859–897, 1999.
- [25] Mohamed-Jalal Fadili und Jean-Luc Starck. Curvelets and ridgelets. *Encyclopedia of Complexity and Systems Science*, pages 1718—1738, 2009.
- [26] Matan Gavish, Boaz Nadler und Ronald R. Coifman. Multiscale wavelets on trees, graphs and high dimensional data: Theory and applications to semi supervised learning. Proceedings of the 27th International Conference on Machine Learning, Haifa, 2010.
- [27] Kanghui Guo und Demetrio Labate. Optimally sparse multidimensional representation using shearlets. *SIAM J. Math. Anal.*, 39:298–318, 2007.
- [28] Kanghui Guo, Wang-Q Lim, Demetrio Labate, Guido Weiss und Edward Wilson. Wavelets with composite dilations. *Electr. res. Announc. of AMS*, 10:78–87, 2004.
- [29] Alfréd Haar. Zur theorie der orthogonalen funktionensysteme. *Math. Ann.*, 69(3):331–371, 1910.
- [30] Ami Harten. Discrete multiresolution analysis and geralized wavelets. *J. Appl. Num. Math.*, 12:153–193, 1993.
- [31] Ami Harten. Multiresolution representation of data: general framework. *SIAM J. Numer. Anal.*, 33:1205–1256, 1996.
- [32] Henk J.A.M. Heijmans, Béatrice Pesquet-Popescu und Gemma Piella. Building nonredundant adaptive wavelets by update lifting. *Appl. Comput. Harmon. Anal.*, 18(3):252–281, 2005.
- [33] Armin Iske. On the approximation order and numerical stability of local lagrange inter- polation by polyharmonic splines. volume 145 of *Modern Developments in Multivariate Approximation, International Series of Numerical Mathematics*, pages 153–165. Birkhäuser, 1998.

- [34] Armin Iske. *Multiresolution Methods in Scattered Data Modelling*, volume 37 of *Lecture Notes in Computational Science and Engineering*. Springer, Berlin, 2004.
- [35] Laurent Jacques und Jean-Pierre Antoine. Multiselective pyramidal decomposition of images: wavelets with adaptive angular selectivity. *Int. J. Wavelets Multiresolut. Inf. Process.*, 5:785–814, 2007.
- [36] Borislav Karaivanov und Pencho Petrushev. Nonlinear piecewise polynomial approximation beyond besov spaces. *Appl. Comput. Harmon. Anal.*, 15:177–223, 2003.
- [37] Jens Krommweh. Tetrolet transform: A new adaptive haar wavelet algorithm for sparse image representation. *J. Vis. Commun. Image R.*, 21(4):364–374, 2010.
- [38] Stéphane Mallat. *A Wavelet Tour of Signal Processing, Second Edition*. Academic Press, San Diego, 1999.
- [39] Stéphane Mallat. Geometrical grouplets. *Appl. Comput. Harmon. Anal.*, 26:161–180, 2009.
- [40] Martin J. Mohlenkamp und María Cristina Pereyra. *Wavelets, Their Friends, and What They Can Do for You*. Series of Lectures in Mathematics. European Mathematical Society, Zürich, 2008.
- [41] Francis J. Narcowich, Joseph D. Ward und Holger Wendland. Sobolev error estimates and a bernstein inequality for scattered data interpolation via radial basis functions. *Constr. Approx.*, 24(2):175–186, 2006.
- [42] Erwan Le Pennec und Stéphane Mallat. Bandelet image approximation and compression. *Multiscale Model. Simul.*, 4:992–1039, 2005.
- [43] Gerlind Plonka. The easy path wavelet transform: a new adaptive wavelet transform for sparse representation of two-dimensional data. *Multiscale Model. Simul.*, 7:1474–1496, 2009.
- [44] Gerlind Plonka und Daniela Roşca. Easy path wavelet transform on triangulations of the sphere. *Math. Geosci.*, 42(7):839–855, 2010.
- [45] Gerlind Plonka und Stefanie Tenorth. Nonlinear locally adaptive wavelet filter banks. SAMPTA'09 - 8th International Conference on Sampling Theory and Applications, Marseille. online veröffentlicht auf <http://www.latp.univ-mrs.fr/SAMPTA09/>.

- [46] Gerlind Plonka, Stefanie Tenorth und Armin Iske. Optimally sparse image representation by the easy path wavelet transform. *Int. J. Wavelets Multiresolut. Inf. Process.* wird veröffentlicht.
- [47] Gerlind Plonka, Stefanie Tenorth und Armin Iske. Optimal representation of piecewise hölder smooth bivariate functions by the easy path wavelet transform. *Preprint*, 2011.
- [48] Gerlind Plonka, Stefanie Tenorth und Daniela Roşca. A hybrid method for image approximation using the easy path wavelet transform. *IEEE Trans. Image Process.*, 20(2):372 – 381, 2011.
- [49] Duncan D.-Y. Po und Minh N. Do. Directional multiscale modeling of images using the contourlet transform. *IEEE Trans. Image Process.*, 15:1610–1620, 2006.
- [50] Idan Ram, Michael Elad und Israel Cohen. Generalized tree-based wavelet transform. *CoRR*, abs/1011.4615, 2010.
- [51] Rahul Shukla, Pier L. Dragotti, Minh N. Do und Martin Vetterli. Rate-distortion optimized tree structured compression algorithms for piecewise smooth images. *IEEE Trans. Image Process.*, 14:343–359, 2005.
- [52] Jean-Luc Starck, Michael Elad und David L. Donoho. Image decomposition via the combination of sparse representations and a variational approach. *IEEE Trans. Image Process.*, 14:1570–1582, 2005.
- [53] Hans Triebel. *Theory of Function Spaces*. Birkhäuser, Basel, 1983.
- [54] Hans Triebel. *Theory of Function Spaces II*. Birkhäuser, Basel, 1992.
- [55] Vladan Velisavljević, Baltasar Beferull-Lozano, Martin Vetterli und Pier L. Dragotti. Directionlets: anisotropic multidirectional representation with separable filtering. *IEEE Trans. Image Process.*, 15(7):1916–1933, 2006.
- [56] Michael B. Wakin, Justin K. Romberg, Hyeokho Choi und Richard G. Baraniuk. Wavelet-domain approximation and compression of piecewise smooth images. *IEEE Trans. Image Process.*, 15:1071–1108, 2006.
- [57] Joachim Weickert. *Anisotropic diffusion in image processing*. Teubner, Stuttgart, 1998.

Lebenslauf

Persönliche Daten

Name Stefanie Tenorth
Adresse Memeler Ring 72
47495 Rheinberg
E-Mail s.tenorth@math.uni-goettingen.de
Geburtsdatum/-ort 08.06.1983 in Wesel
Familienstand ledig

Schule

08/1993 - 06/2002 **Amplonius-Gymnasium**, Rheinberg.
28.06.2002 Abitur (Note: 2,4)

Studium

10/2002 - 01/2008 **Diplomstudiengang Mathematik (DII) mit Nebenfach Informatik**
Universität Duisburg-Essen, Campus Duisburg (Vordiplom (Note: 1,6)).
29.01.2008 Diplom (Gesamtnote: sehr gut), Titel der Diplomarbeit: Verallgemeinerte Rudin-
Osher-Fatemi-Regularisierung zur Signalentstörung (Note: 1,0)

Berufliche Erfahrungen, Praktika

02/2005 - 04/2005 **Praktikum**, *Forschungszentrum caesar*, Bonn, Arbeitsgruppe Crystal Growth.

10/2004 - 02/2005 **Studentische Hilfskraft**, *Universität Duisburg-Essen*, Fachbereich Mathematik.
10/2005 - 02/2006 Mathematiktutorium für Maschinenbauer, Korrektur von Übungsaufgaben.
04/2006 - 07/2006

04/2008 - 09/2008 **Promotionsstipendium**, *Universität Duisburg-Essen*, Fachbereich Mathematik.

10/2008 - 06/2010 **Wissenschaftliche Mitarbeiterin**, *Universität Duisburg-Essen*,
Fachbereich Mathematik, im Rahmen des DFG-Schwerpunktprogramms 1324
„Extraktion quantifizierbarer Information aus komplexen Systemen“.

04/2008 - 09/2010 **Promotionsstudium**, *Universität Duisburg-Essen*, Fachbereich Mathematik.

07/2010 - 07/2011 **Wissenschaftliche Mitarbeiterin**, *Georg-August-Universität Göttingen*,
Fakultät für Mathematik und Informatik,
im Rahmen des DFG-Schwerpunktprogramms 1324
„Extraktion quantifizierbarer Information aus komplexen Systemen“.

10/2010 - 07/2011 **Promotionsstudium** im Grundprogramm Mathematik,
Georg-August-Universität Göttingen, Fakultät für Mathematik und Informatik.