3 Ein diskretes Modell zur fraktalen Bildkodierung

Die zentrale Idee der fraktalen Bildkompression besteht in der Ausnutzung von Redundanzen durch die Verwendung eines fraktalen Signalmodells. Ein solches Modell ist auf alle Signale anwendbar, die eine Art von Selbstähnlichkeit besitzen. In einem solchen Modell werden “kleinere” Signalteile mit Hilfe von “größeren” transformierten Signalteilen approximiert.

Ziel dieses Kapitels soll die Bereitstellung eines Rahmens zur Beschreibung von Begriffen wie “Signal”, “Signalstück” und “Abbildung von Signalen” sein. Diese Betrachtungen sollen im Wesentlichen in Hinblick auf die Anwendung auf digitale Bilder und die Implementierung in Soft- und Hardware durchgeführt werden.

Für die Bereitstellung solcher Grundlagen existieren in der Literatur verschiedene Ansätze. Einer der wesentlichen Ansätze wurde bereits in Kapitel 2 vorgestellt und basiert auf maßtheoretischen Überlegungen [Barnsley88a] [Barnsley93]. Dieser Ansatz hat den Vorteil, daß man mit ihm auf sehr elegante Art und Weise die Vorgänge bei der Kodierung von Bildern beschreiben kann. Dies gilt insbesondere auch für die Beschreibung der Attraktoren eines IFS. Unter dem Aspekt der Anwendung auf digitale Bilder besteht allerdings darin auch der wesentliche Nachteil dieses Ansatzes. Die benutzten mathematischen Methoden und angenommenen Voraussetzungen sind für konkrete Anwendungen zu abstrakt und lassen sich nur schwer anwenden.

Ein weiterer Ansatz arbeitet mit kontinuierlichen Funktionen auf $\mathbb{R}$ bzw. $\mathbb{R}^2$, wie schon kurz in Kapitel 2 beschrieben. Die in diesem Ansatz verwendeten Methoden sind zwar weniger abstrakt, zeigen aber bei der Anwendung auf diskrete digitale Bilder erhebliche Schwächen in der Beschreibung dieser Bilder auf, weshalb dieser Ansatz hier auch nicht weiter diskutiert werden soll.

Der Ansatz, das Problem mit diskreter Mathematik zu beschreiben und zu bearbeiten, kommt der Problemstellung allerdings am nächsten. Bilder und Signale werden diskret repräsentiert und die Formulierung der Algorithmen wird unter dem Aspekt einer Implementierung auf digitalen Rechnern vorgenommen. Das in diesem Kapitel entwik-

kelte Modell zur fraktalen Bildkodierung benutzt im wesentlichen lineare Algebra zur Beschreibung der Kodierung und Dekodierung und leitet in den folgenden Kapiteln 4 und 5 Eigenschaften und experimentelle Ergebnisse einer auf diesem diskreten Modell basierenden fraktalen Bildkodierung ab.

Das im folgenden entwickelte diskrete Modell stützt sich dabei auf die in Kapitel 2 vorgestellten Ideen von Barnsley und überträgt diese in eine diskrete Notation. Aufgrund der Zielsetzung, ein Modell für eine Implementierung entwickeln zu wollen, werden auch in diesem Kapitel nur jene Aspekte einer fraktalen Kodierung berücksichtigt, die für eine Implementierung relevant sind. Als eine weitere Folge dieser Zielsetzung

$$\left\{ x_{i_0 \ldots i_N} : i_j = 1, 2, \ldots, M_j \; ; \; j = 1, 2, \ldots, N \text{ und } \prod_{j=0}^{N} M_j = M \right\}$$

$$= \{x_{\sigma(1)}, x_{\sigma(2)}, \ldots, x_{\sigma(M)}\} \text{ mit } \sigma \in S_M; \; S_M \text{ symmetrische Gruppe von } \{1, 2, \ldots, M\}$$

σ ist eine bijektive Abbildung, so daß diskrete Signale $\in \mathbb{R}^{M_1 \times M_2 \times M_3}$ immer als endliche diskrete Signale $\in \mathbb{R}^M$ aufgefaßt werden können. Ohne Einschränkung der Allgemeinheit wird im folgenden angenommen, daß $\sigma = id$ ist, wodurch sich die Formulierungen erheblich vereinfachen.

Signalstücke

Nach der Identifizierung eines diskreten Signals mit einem Vektor im $\mathbb{R}^M$ stellt sich die Frage nach der Definition eines Signalstückes. Für den diskreten Fall ist ein *Signalstück* eine Sammlung von Samples, die in einem gewissen Sinne als “benachbart” bezeichnet werden können. In einem Bild wären dies z.B. alle Samples einer Region R. Die Größe M_R eines solchen Signalstückes wird dann durch die Anzahl der Samples festgelegt. In Abhängigkeit von σ besitzen die Samples eines Signalstückes dann eine eindeutig bestimmte Position in der Vektordarstellung:

$$I = \{i_1, i_2, \ldots, i_{M_R}\} \subset \{1, 2, \ldots, M\}$$

Eine solche Menge von Indizes von Samples eines Signalstückes wird *Indexblock* genannt. Der Spaltenvektor bestehend aus den Sample-Werten für die Indizes I wird *Signalblock* genannt:

x_I x_i

/F8 1˝.86ı˝(M)Tjı˝/F8 613.418 Tmı˝(˜.86ı12 278.24 400

Ein Signalblock besteht somit aus dem Inhalt des Signals über dem Indexblock . Bei einem Bild z.B. entspricht ein Signalblock also einem bestimmten Bildbereich.

Abb. 1: *Darstellung eines Pixelblockes als 1-dimensionales Signal*

Die Beschreibung eines Signalstückes ist durch die Angabe des Indexblockes eindeutig.

Partielle Abbildungen

Ein zentraler Aspekt in fraktalen Modellierungsschemata ist die Abbildung eines Signalstückes auf ein (normalerweise) kleineres Signalstück. Eine solche Transformation wird dann eine partielle Abbildung genannt, da eine Transformation nur einen Teil des gesamten Signals abbildet. Für die Abbildung des gesamten Eingangssignals sind i.a. mehrere partielle Transformationen notwendig.

Für den vorliegenden Fall wird eine Abbildung $T_n : \mathbb{R}^M \rightarrow \mathbb{R}^M$ definiert, welche einen Indexblock I des Eingangssignals (Definitionsbereich) auf einen Indexblock J des Ausgangssignals (Wertebereich) abbildet. Diese Abbildung ist durch die folgenden beiden Eigenschaften charakterisiert:

1. $T_n(x)$ ist nur von den Werten von $x \in \mathbb{R}^M$ abhängig, welche Indizes im Indexblock I des Definitionsbereichs besitzen:

$$x = (x_1, \ldots, x_M) \text{ mit } x_i = \begin{cases} \neq 0 \text{ für } i \in I \\ 0 \text{ sonst} \end{cases}$$

2. $T_n(x)$ ist ein Vektor $\in \mathbb{R}^M$, in dem nur jene Werte ungleich null sind, deren Index auch im Indexblock J des Wertebereichs liegen:

$$y = (y_1, \ldots, y_M) = T_n(x) \text{ mit } y_i = \begin{cases} \neq 0 \text{ für } i \in J \\ 0 \text{ sonst} \end{cases}$$

Desweiteren wird an die Transformationen die Anforderung gestellt, daß sie sich in der folgenden Art und Weise zusammensetzen lassen. Die Begründung, warum man diese Anforderung an die Transformationen stellt, wird gegen Ende des Kapitels ersichtlich, wenn die Eigenschaften der Transformationen vor allem im Hinblick auf eine Implementierung des Modells diskutiert werden.

Definition: (Transformation T_n)

Eine Transformation T_n eines Eingangssignalblocks x_I auf einen Indexblock J des Ausgangssignals wird definiert durch:

$$(1)\ T_n(x) = P_J(K(D(F_I(x)))) \text{, wobei gilt:}$$

- ***fetch*-Operator F_I:**

  Dieser Operator wählt die M_D Signalwerte über dem Indexblock des Definitionsbereichs aus:

$$F_I : \mathbb{R}^M \rightarrow \mathbb{R}^{M_D} \text{ mit } F_I(x) = x_I$$

- ***decimation*-Operator D:**
  Reduziert die Größe des Urbildes, d.h. reduziert die Anzahl der Samples, möglichst unter Beibehaltung der Eigenschaften:

$$D: \mathbb{R}^{M_D} \to \mathbb{R}^{M_R} \text{ mit } D(x_I) = x_I'$$

- ***kernel*-Transformation K:**
  Dies ist die eigentliche Transformation, auf die im folgenden noch näher eingegangen wird:

$$K: \mathbb{R}^{M_R} \to \mathbb{R}^{M_R} \text{ mit } K(x_I') = x_J'$$

- ***put*-Operator P_J:**
  Platziert die Ausgabe von K über dem "richtigen" Indexblock des Ausgangssignals:

$$P_J: \mathbb{R}^{M_R} \to \mathbb{R}^{M_R} \text{ mit } P_J(x_J') = x_J$$

Der Aufbau dieser Transformationen legt eine Betrachtung der einzelnen Elemente nahe, was im folgenden auch geschehen soll.

fetch- und *put*-Operator

Der *fetch*- und der *put*-Operator können im vorliegenden diskreten Fall durch zwei Matrizen beschrieben werden:

$$F_I = \begin{bmatrix} 0 & \underbrace{\begin{vmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & & \dots \\ \dots & & 1 & 0 \\ 0 & \dots & 0 & 1 \end{vmatrix}}_{I} & 0 \end{bmatrix} \qquad P_J = \begin{bmatrix} & 0 & \\ \left.\begin{vmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & & \dots \\ \dots & & 1 & 0 \\ 0 & \dots & 0 & 1 \end{vmatrix}\right\} J \\ & 0 & \end{bmatrix}$$

decimation-Operator

Der *decimation*-Operator kann auf verschiedene Art und Weise implementiert werden, wobei die beiden häufigsten Methoden *Subsampling* und *Decimation Averaging* sind. Auch diese Methoden lassen sich durch Matrizen darstellen, wobei, wie oben bereits erwähnt, bei der Darstellung ohne Einschränkung der Allgemeinheit von einer kontinuierlichen Indexmenge ausgegangen wird.

Die Matrix für den *decimation*-Operator hat dann die folgende Gestalt:

$$(2)\ D = \begin{bmatrix} \omega_1\ \omega_2\ \ldots\ \omega_R & & & \\ & \omega_1\ \omega_2\ \ldots\ \omega_R & & \\ & & \ldots & \\ & & & \omega_1\ \omega_2\ \ldots\ \omega_R \end{bmatrix}$$

Die Folge der Gewichte $\omega = \begin{bmatrix}\omega_1\ \omega_2\ \ldots\ \omega_R\end{bmatrix}^T$ wird dann *decimation*-Filter genannt. Mit $\omega = \begin{bmatrix}1\ 0\ \ldots\ 0\end{bmatrix}^T$ wird das Subsampling und mit $\omega = \begin{bmatrix}\frac{1}{r}\ \frac{1}{r}\ \ldots\ \frac{1}{r}\end{bmatrix}^T$ das Averaging durchgeführt. Für ein 2-dimensionales Signal (Bild) erhält man somit:

Abb. 2: *Subsampling und Averaging*

Partielle Transformationen

Obwohl man T_n nur als Operator auf einem Signalstück ansieht, wird er auf dem gesamten Eingangssignal X definiert. Die Definition von T_n bedingt allerdings zuerst die Einführung von zwei neuen Begriffen.

Ein essentieller Schritt bei der fraktalen Bildkodierung ist die *Partitionierung* des Eingangssignals in Wertebereiche (*range blocks*). Besteht diese Partitionierung aus sich nicht überdeckenden gleichgroßen Blöcken der Größe M_R, dann spricht man von einer *uniformen Partitionierung* des Signals. Überdeckt die Partitionierung das gesamte Eingangssignal, so erhält man eine *vollständige disjunkte Partitionierung*. In der Vektordarstellung bedeutet dies eine Partitionierung der Indizes $1, 2, \ldots, M$ in $N = M/M_R$ große Teilmengen $\{J_n\}_{n=1}^{N} = J$. Diese Teilmengen werden die *Wertebereich-Partitionierung* des fraktalen Modells genannt. Neben der uniformen Wertebereich-Partitionierung sind auch adaptive Verfahren anwendbar, wie sie in Kapitel 4 behandelt werden.

Die Wertebereich-Partitionierung enthält die “kleinen Signalstücke”, wenn die Selbsttransformation definiert wird. Zu diesen “kleinen Signalstücken” benötigt man nun noch “große Signalstücke”, aus denen die kleinen hergeleitet werden können. Für ein bestimmtes Ausgangsmodell zur Partitionierung wird auch eine bestimmte Menge von *Definitionsbereichen* (*domain blocks*) gewählt. Diese entsprechende Menge von Definitionsbereichen wird *library* genannt und mit $I = \{I_1, I_2, \ldots, I_Q\}$ bezeichnet.

Existiert eine Wertebereich-Partitionierung und eine entsprechende *library*, so kann eine partielle Transformation folgendermaßen definiert werden:

1. Wähle für jeden Wertebereich $J_n \in J$ einen Definitionsbereich $I_{q(n)} \in I$ und eine kernel-Transformation $K_n \in K$ aus einer vordefinierten Menge K von Transformationen. ($q(n)$ = Auswahlfunktion)
2. Die partielle Transformation ist dann definiert durch:

$$T_n(x) = P_{J_n}(K_n(D(F_{I_{q(n)}}(x))))$$

3. Die Aufsummierung der partiellen Transformationen ergibt dann eine Abbildung des gesamten Eingangssignals:

$$(3)\ T(x) = \sum_{n=1}^{N} T_n(x)$$

Selbstähnlichkeit

Besitzt ein Signalstück die Eigenschaft, durch eine Abbildung $T(x)$ gemäß (1) auf Seite 64 nicht verändert zu werden, dann wird dieses Signalstück *selbsttransformierbar* oder *partiell selbstähnlich* genannt. Ein Signal $x \in \mathbb{R}^M$ ist also selbstähnlich, wenn $T(x) = x$ gilt, daß heißt, wenn x ein Fixpunkt von T ist. Sollte x_T der einzige Fixpunkt von T sein, dann ist mit der Definition von T auch x_T spezifiziert, vorausgesetzt es existiert eine Möglichkeit zur Bestimmung des Fixpunktes eines beliebig gewählten Operators.

2. Formale Beschreibung der fraktalen Bildkodierung

Gegeben seien Signale einer festen Länge M und $S \subset \mathbb{R}^M$ sei die Menge aller mögliche Signale der Länge M einer bestimmten Informationsquelle. Desweiteren sei T_a eine partielle Transformation gemäß (1) auf Seite 64 und (3) auf Seite 67. Dann sei die eine Menge partieller Transformationen definiert durch:

$$T = \{T_a : \mathbb{R}^M \to \mathbb{R}^M : a \in A\}$$

Jede Transformation $T_a \in \mathcal{T}$ sei eindeutig durch einen Parametervektor $a \in \mathcal{A}$, mit einem Parameterraum $\mathcal{A}$, bestimmt. Desweiteren besitze jedes $T_a \in \mathcal{T}$ einen eindeutigen Fixpunkt.

$\Rightarrow$ Der Parametervektor a bestimmt den Fixpunkt eindeutig.

Unter der Voraussetzung, daß S aus Signalen besteht, die durch Transformationen aus $\mathcal{T}$ beschreibbar sind, wird

$$\hat{S} = \{x_t : T(x_t) = x_t \text{ für einige } T \in \mathcal{T}\}$$

das *Fraktale Modell* von S genannt.

Zu einem gegebenen Signal $x \in S$ soll nun ein *fraktales Signal-Modell* $x_T \in \hat{S}$ gefunden werden, so daß x_T eine Approximation von x ist. Dies ist der essentielle Schritt in der fraktalen Signalkodierung, was die fraktale Bildkodierung mit einschließt.

Die Eigenschaft der Transformation, fast überall kontrahierend zu sein, garantiert die Eindeutig- und die Berechenbarkeit des Fixpunktes. Der Fixpunkt kann dann iterativ berechnet werden und wird im allgemeinen *Attraktor* genannt.

3. Collage Theorem

Im allgemeinen verändert ein Operator T im $\mathbb{R}^M$ die Norm eines Vektors, auf den er angewendet wird. Als Angabe dafür, wieviel sich die Norm eines Vektors durch die Anwendung des Operators geändert hat, dient der sogenannte *Lipschitz-Faktor*:

$$s = \sup\left(\frac{\|T(x)\|}{\|x\|}\right)$$

s wird auch die Norm von T genannt und mit $\|T\|$ bezeichnet. Ist der Lipschitz-Faktor kleiner als 1 $(s < 1)$, so wird der Operator T kontrahierend genannt, d.h. der Operator verkleinert die Norm des Vektors. s wird in einem solchen Fall auch *contractivity factor* von T genannt.

Ist die K-te Iterierte T^K $(K \geq 1)$ eines Operators T kontrahierend, dann wird T *kontrahierend in der K-ten Iteration* genannt. Für solche Operatoren gilt die folgende endliche Version des Banach'schen Fixpunktsatzes:

Theorem: (Contractive Mapping Theorem)

Sei $T : \mathbb{R}^M \to \mathbb{R}^M$ ein kontrahierender Operator in der K-ten Iteration mit $K \geq 1$ und $x \in \mathbb{R}^M$.

Dann besitzt T einen durch $x_T = \lim_{n \to \infty} T^n(x)$ eindeutig bestimmten Fixpunkt.

Das Theorem liefert damit sowohl die Eindeutigkeit des Fixpunktes, als auch einen Weg zu dessen Berechnung.

Die Problemstellung bei der fraktalen Signalkodierung ist nun, einen Operator zu finden, dessen Attraktor x_T dem Signal so ähnlich wie möglich ist, was die Minimierung der Entfernung $\|x - x_T\|$ bedeutet. Das Finden dieses Minimums für eine Menge nicht trivialer Operatoren ist i.a. eine schwere Aufgabe.

Für suboptimale, dafür allerdings berechenbare Lösungen existieren verschiedene Methoden. Die meisten dieser Methoden basieren auf dem *Collage Theorem*. Mit diesem Theorem soll die Idee der Collage auf die Signal- und damit auch auf die Bildkodierung übertragen werden.

Unter Verwendung einer zusammengesetzten Transformation T wird ein Teil eines Eingangssignals x ausgeschnitten, transformiert und anschließend als ein Teil des Ausgangssignals wieder eingefügt, so daß auf diese Weise eine Collage $x_C = T(x)$ für das Signal bzw. Bild entsteht. Die Idee des Collage Theorems beim Suchen eines Attraktors x_T ist, daß unter bestimmten Voraussetzungen eine Collage, die "nahe" am Original x ist, auch einen Attraktor liefert, welcher das Original approximiert.

Theorem: (Collage Theorem)

Sei $T: \mathbb{R}^M \to \mathbb{R}^M$ ein kontrahierender Operator in der K-ten Iteration mit $K \geq 1$ und $x_T \in \mathbb{R}^M$ der Fixpunkt von T.

s_k bezeichne den Lipschitz-Faktor der k-ten Iteration T^k.

Desweiteren seien ein Signal $x \in \mathbb{R}^M$, eine Collage $x_C \in \mathbb{R}^M$ und ein $\varepsilon > 0$ gegeben.

Für alle $\|x - x_C\| \leq \varepsilon$ gilt dann:

$$(4)\quad \|x - x_T\| \leq \frac{1 + \sum_{k=1}^{K-1} s_k}{1 - s_K} \cdot \varepsilon$$

$$(5)\quad \Rightarrow \|x - x_T\| \leq \frac{1 - s_1^K}{(1 - s_1)(1 - s_K)} \cdot \varepsilon$$

Für den Spezialfall, daß T strikt kontrahierend $(K = 1)$ ist, folgt:

$$(6)\quad \Rightarrow \|x - x_T\| \leq \frac{1}{(1 - s_1)} \cdot \varepsilon$$

Unter der Voraussetzung, daß die rechte Seite der Gleichung nicht zu groß ist, erhält man als praktische Folgerung aus dem Collage Theorem, daß mit dem Finden eines Operators, dessen Collage nahe bei dem Originalsignal x liegt, dessen Fixpunkt x_T ebenfalls nahe bei x liegt.

4. *kernel*-Transformationen

Die bisher besprochenen Abbildungungen setzen sich gemäß (1) auf Seite 64 aus vier Operatoren zusammen, von denen nun der Kern K der Abbildungen betrachtet werden soll. Neben der Voraussetzung, daß T kontrahierend in der K-ten Iteration ist, ist eine weitere Forderung für ein benutzbares Modell, daß T nicht linear ist.

Dies folgt aus der Tatsache, daß 0 für lineare Transformationen immer ein Fixpunkt ist. Aufgrund der Eindeutigkeit des Fixpunktes muß 0 damit auch der Einzige sein. Dies bedeutet also, daß für die Modellierung eines Signals $x \neq 0$ keine lineare Transformationen verwendet werden können.

Aus der Konstruktion von T gemäß (1) bis (3) wird ersichtlich, daß T genau dann linear ist, wenn die Kerne (*kernels*) der partiellen Abbildungen linear sind. Der nächste Schritt für die fraktale Signalkodierung ist somit die Bestimmung einer Menge von Kernen $\mathcal{K} = \{K : \mathbb{R}^{M_R} \to \mathbb{R}^{M_R}\}$ mit nichtlinearen K's. Aus der Menge aller nichtlinearer Transformationen sei die Klasse der affinen Transformationen gewählt.

Definition: (Affine Transformation)

Es sei L' eine lineare Transformation im $\mathbb{R}^{M_R}$ und $t' \neq \vec{0}$ ein beliebiger Vektor $\in \mathbb{R}^{M_R}$.

Dann ist $K : \mathbb{R}^{M_R} \to \mathbb{R}^{M_R}$ mit $K(x) = L'x + t'$ eine *affine Transformation* im $\mathbb{R}^{M_R}$.

L' ist dann eine $\mathbb{R}^{M_R} \times \mathbb{R}^{M_R}$ Matrix. Wenn $\mathcal{K}$ aus affinen Transformationen des $\mathbb{R}^{M_R}$ besteht, dann besteht

$$\mathcal{T} = \{T = \sum_{n=1}^{N} T_n : T_n(x) = P_{J_n}(K_n(D(F_{I_{q(n)}}(x)))) \text{ mit } K_n \in \mathcal{K}\}$$

aus affinen Transformationen des $\mathbb{R}^M$.

Beispiel

Der lineare Teil der Kerntransformationen aus $\mathcal{K}$ bestehe einfach aus einer Multiplikation mit einer Konstanten β_2, was einer Skalierung der Amplitude entspricht. Der konstante Part der Transformation sei ein Vektor $t = \beta_1 \begin{bmatrix} 1 & 1 & \dots & 1 \end{bmatrix}^T$ mit einer Konstanten β_1. Nach der obigen Definition stellen diese Transformationen damit die einfachsten dar. Eine Transformation aus $\mathcal{T}$ der Form $T(x) = Lx + t$ besitzt damit einen linearen Term L der folgenden Form:

$$(7)\ L = \sum_{n=1}^{N} \beta_2^{(n)} P_{J_n} D F_{I_{q(n)}} = \begin{bmatrix} & & & \beta_2^{(1)} D \\ \beta_2^{(2)} D & & & \\ & \cdots & \cdots & \cdots \\ & & \beta_2^{(N)} D & \end{bmatrix}$$

Die Blockstruktur ergibt sich aus der Annahme, daß die Indexblöcke kontinuierlich sind (s.a. Seite 62). Die horizontale Position einer Teilmatrix $\beta_2^{(n)} D$ entspricht der Position des Indexblocks I_n des zugehörigen Definitionsbereiches. Die vertikale Position ist durch den Indexblock J_n des Wertebereichs bestimmt.

Der Translations-Term hat die folgende Gestalt:

$$t = \left[\underbrace{\beta_1^{(1)}, \beta_1^{(1)}, \ldots, \beta_1^{(1)}}_{\triangleq\ K_1}, \underbrace{\beta_1^{(2)}, \beta_1^{(2)}, \ldots, \beta_1^{(2)}}_{\triangleq\ K_2}, \ldots, \underbrace{\beta_1^{(N)}, \beta_1^{(N)}, \ldots, \beta_1^{(N)}}_{\triangleq\ K_N} \right]^T$$

5. Berechnung der Lipschitz-Faktoren

Da der Lipschitz-Faktor eines Operators sowohl für das Collage Theorem, als auch für die Konvergenz eines iterierten Dekodier-Algorithmus wichtig ist, sollen die folgenden Betrachtungen die Berechenbarkeit desselben zum Ziel haben. Die Berechnung soll für eine Menge affiner Transformationen mit einer uniformen Wertebereichs- und einer vollständig disjunkten Definitionsbereichs-Partitionierung durchgeführt werden. Der lineare Term der Transformation skaliere die Amplitude des Signals, wie im vorherigen Absatz beschrieben. Die *library* bestehe aus Q gleichgroßen Definitionsbereichen der Länge M/Q.

Der Lipschitz-Faktor s einer affinen Transformation ist gleich dem Lipschitz-Faktor seines linearen Terms L und somit gleich der Matrixnorm:

$$s = \|L\|$$

Im Falle der L_2-Norm ist die Matrixnorm $\|L\|_2$ durch die Quadratwurzel des größten Eigenwertes der Matrix $L^T L$ gegeben. Aufgrund der Annahmen über den linearen Term, besitzt $L^T L$ dann die folgende Diagonalstruktur:

$$L^T L = \begin{bmatrix} L_1^* & 0 & \dots & 0 \\ 0 & L_2^* & & \dots \\ \dots & & \dots & \\ 0 & \dots & & L_Q^* \end{bmatrix}$$

Diese Matrix besteht aus Sub-Matrizen L_q^* für $q = 1, \dots, Q$, wobei jedes L_q^* einem Definitionsbereich aus der *library* entspricht. Aufgrund der Blockstruktur von L (siehe (7) auf Seite 71) ergibt sich jede Sub-Matrix L_q^* als Summe über jene Wertebereiche, welche den speziellen Definitionsbereich I_q benutzen:

$$L_q^* = \sum_{I_{q(n)} = I_q} (\beta_2^{(n)} D)^T (\beta_2^{(n)} D) = \sum_{I_{q(n)} = I_q} (\beta_2^{(n)})^2 D^T D$$

Aus der Struktur von D (siehe Gleichung (2) auf Seite 66) folgt weiter, daß jede der Sub-Matrizen L_q^* wiederum eine Block-Struktur besitzt, da

$$D^T D = \begin{bmatrix} \omega\omega^T & 0 & \dots & 0 \\ 0 & \omega\omega^T & & \dots \\ \dots & & \dots & \\ 0 & \dots & & \omega\omega^T \end{bmatrix}$$

gilt.

Die Eigenwerte der Matrix $L^T L$ sind die Lösung von:

$$\det(L^T L - \lambda I) = 0$$

Da die Determinante einer Blockmatrix gleich dem Produkt der Determinanten der Sub-Matrizen ist, folgt:

$$\det(L^T L - \lambda I) = \prod_{q=1}^{Q} (\det(\beta_q^* \omega\omega^T - \lambda I))^{\frac{M}{Q \cdot R}} \text{ mit } \beta_q^* = \sum_{I_{q(n)} = I_q} (\beta_2^{(n)})^2$$

Die Berechnung der Determinaten $\det(\beta_q^* \omega\omega^T - \lambda I)$ hängt somit von der Wahl des *decimation*-Filters ab. Für die eingeführten Filter *subsampling* und *averaging* sieht die Berechnung folgendermaßen aus:

Subsampling:

Es ist $\omega = \begin{bmatrix} 1 & 0 & \dots & 0 \end{bmatrix}^T$, woraus folgt:

$$\Rightarrow \det(\beta_q^* \omega\omega^T - \lambda I) = \begin{vmatrix} \beta_q^* - \lambda & & \\ & \beta_q^* - \lambda & \\ & & \beta_q^* - \lambda \end{vmatrix} = (\beta_q^* - \lambda)^R$$

$$\Rightarrow \det(L^T L - \lambda I) = \prod_{q=1}^{Q} (\beta_q^* - \lambda)^{\frac{M}{Q}}$$

$\Rightarrow$ $L^T L$ hat Q Eigenwerte $\lambda_q = \beta_q^* = \sum_{I_{q(n)} = I_q} (\beta_2^{(n)})^2$, wobei jeder Eigenwert M/Q-mal vorkommt.

Averaging:

Es ist $\omega = \left[\frac{1}{r}\ \frac{1}{r}\ \dots\ \frac{1}{r}\right]^T$, woraus folgt:

$$\Rightarrow \det(\beta_q^* \omega\omega^T - \lambda I) = \begin{vmatrix} \frac{\beta_q^*}{r^2} - \lambda & \frac{\beta_q^*}{r^2} - \lambda & \dots & \frac{\beta_q^*}{r^2} - \lambda \\ \frac{\beta_q^*}{r^2} - \lambda & \frac{\beta_q^*}{r^2} - \lambda & & \dots \\ \dots & & \dots & \\ \frac{\beta_q^*}{r^2} - \lambda & \dots & & \frac{\beta_q^*}{r^2} - \lambda \end{vmatrix} = (-1)^R \cdot \lambda^{R-1} \cdot \left(\lambda - \frac{\beta_q^*}{r^2}\right)$$

$$\Rightarrow \det(L^T L - \lambda I) = \prod_{q=1}^{Q} (-1)^{\frac{M}{q}} \cdot \lambda^{\frac{R-1}{QR} \cdot M} \cdot \left(\lambda - \frac{\beta_q^*}{r^2}\right)$$

$\Rightarrow$ $L^T L$ hat Q Eigenwerte $\lambda_q = \frac{\beta_q^*}{r} = \frac{1}{r} \cdot \sum_{I_{q(n)} = I_q} (\beta_2^{(n)})^2$ ungleich null.

Damit läßt sich der Lipschitz-Faktor für diese beiden Fälle folgendermaßen berechnen:

Subsampling:

$$s = \sqrt{\max\left(\sum_{I_{q(n)} = I_q} (\beta_2^{(n)})^2\right)\Big|_{n=1}^{N}}$$

Averaging:

$$s = \frac{1}{r} \cdot \max\left(\sum_{I_{q(n)} = I_q} (\beta_2^{(n)})^2 \right)\Bigg|_{n=1}^{N}$$

Für die Bestimmung, ob ein Operator kontrahierend ist oder nicht, muß man also für jeden Definitionsbereich die Summe der quadrierten Skalierungsfaktoren für jeden in Frage kommenden Wertebereich berechnen und anschließend das Maximum dieser Summen bestimmen. Beim Subsampling wird dann die Quadratwurzel des Maximums genommen, währenddessen beim Averaging durch das Gewicht dividiert wird. Diese Berechnung der Lipschitz-Faktoren erklärt auch die meist wesentlich langsamere Konvergenz bei der Verwendung des Subsamplings.

Obwohl diese Resultate nur für die L_2-Norm nachgewiesen wurden, können sie auf jede Norm des $\mathbb{R}^M$ angewendet werden, da für endlich dimensionale Vektorräume alle Normen äquivalent sind.

6. Methode zur Fraktalen Signalkodierung

In dem bisher eingeführten Rahmen läßt sich eine Prozedur zur fraktalen Signalkodierung folgendermaßen beschreiben:

```
Führe für jeden Indexblock J_n des Wertebereichs aus:
   Wähle den Indexblock I_{q_n}     und den Kern       , so
   daß                 minimal ist.
```

Nach Beendigung dieser Prozedur hat man jenen Operator gefunden, dessen Collage am nahesten an ist. Gemäß dem Collage Theorem führt dies zu einem Fixpunkt , der das Signal in Abhängigkeit von gemäß (4) bis (6) auf Seite 69 approximiert. Für jeden Wertebereich bedeutet dies die Behandlung der folgenden Aufgabenstellungen:

– Suche des “richtigen” Definitionsbereichs, d.h. Berechnung der Auswahlfunktion
– Bestimmung des “richtigen” Kerns .

Im nachfolgenden Kapitel 4 werden, neben der Definition eines Codec’s (*coder / decoder*) zur fraktalen Bildkodierung, diese beiden Aufgabenstellungen unter praktischen Gesichtspunkten behandelt. Zum einen werden Verfahren zur Partitionierung des Eingangssignals besprochen, zum anderen wird die Bestimmung der Transformationen

und damit des Kerns K_n diskutiert. Die folgenden theoretischen Betrachtungen zur Bestimmung des "richtigen" Kerns dienen der Vorbereitungen praktischer Betrachtungen in Kapitel 4 und schließen dieses Kapitel ab.

Bestimmung eines Kerns $K \in \mathcal{K}$

Im folgenden soll die Bestimmung des "richtigen" Kerns $K \in \mathcal{K}$ behandelt werden. Gegeben seien deshalb ein zu modellierendes Signal x und ein Definitionsbereich I. Da der Definitionsbereich für den Rest des Abschnittes fest ist, werden aus Gründen der Lesbarkeit die Indizes bei P (*put*-Operator) und F (*fetch*-Operator) unterdrückt.

Wie schon in den vorangegangenen Abschnitten bemerkt, muß der Kern keine affine Transformation sein, sondern nur nichtlinear. Darüberhinaus wird an einen Kern die Bedingung gestellt, daß mit $K_1, K_2 \in \mathcal{K}$ auch $K(x) = \beta_1 K_1(x) + \beta_2 K_2(x) \in \mathcal{K}$ für $\beta_1, \beta_2 \in \mathbb{R}$ ist. Aus dieser Forderung folgt aufgrund der Vektorraumeigenschaften, daß $\mathcal{K}$ eine Basis besitzt [Kowalsky79].

Theorem:

$\mathcal{K}$ besitzt eine Basis $\{K_1, K_2, \ldots, K_p\}$, so daß jedes Element $K \in \mathcal{K}$ als Linearkombination geschrieben werden kann: $K = \beta_1 K_1 + \beta_2 K_2 + \ldots + \beta_p K_p$ mit $\beta_1, \ldots, \beta_p \in \mathbb{R}$

Damit hat man ein Mittel zur Beschreibung eines Kerns $K \in \mathcal{K}$ gefunden. Zu jedem Wertebereich muß nun der optimale Definitionsbereich und die optimale Menge von Basis-Koeffizienten $\{\beta_1, \beta_2, \ldots, \beta_p\}$ gefunden werden.

Die Überdeckung eines Wertebereiches J_n durch eine Collage läßt sich somit folgendermaßen formulieren:

$$(8)\ T_n x = PK(DFx) = P \cdot \sum_{p=1}^{P} \beta_p K_p(DFx) = PHb$$

$$\text{mit } H = [K_1(DFx), K_2(DFx), \ldots, K_p(DFx)]^T$$

$$\text{und } b = [\beta_1, \beta_2, \ldots, \beta_p]^T$$

Für jeden Wertebereich muß also das Problem gelöst werden, jenen Vektor zu finden, der die folgende Entfernung minimiert:

$$\left\| x_{J_n} - Hb \right\| \overset{!}{=} \min$$

Unter Verwendung der quadratischen Norm kann das Minimum durch Lösung der folgenden Normalengleichung bestimmt werden:

$$(9)\ H^T H b = H^T x_{J_n}$$

Die Resultate gelten bis jetzt für alle nichtlinearen Operatoren, da bisher nicht gefordert wurde, daß es sich um affine Transformationen handeln soll. Wenn die Auswahlfunktion $q(n)$, welche jedem Werte- einen Definitionsbereich zuordnet, gegeben ist, dann kann der optimale Parametervektor b durch die Gleichung (9) auf Seite 75 bestimmt werden.

Die Bestimmung der besten Zuordnung eines Definitionsbereichs zu einem Wertebereich bedarf eine Suche in der *library*. Diese Methode führt zu einem Operator, der die optimale Collage produziert. Es ist allerdings nicht sicher, daß der gefundene Operator auch kontrahierend ist. Dies muß erst überprüft werden, z.B. durch die Berechnung des Lipschitz-Faktors.

Durch eine erschöpfende Suche in der kompletten *library* wird für jeden Wertebereich der optimale Definitionsbereich gefunden, was sich algorithmisch folgendermaßen beschreiben läßt:

– Für jede Kombination von Werte- und Definitionsbereich finde die optimalen Parameter für den Kern der Transformation.
– Berechne die resultierende Distanz zwischen Wertebereich und Collage durch den folgenden Algorithmus:

```
- Initialisiere q(n):=1 für ∀n.
- Für jeden Indexblock I_q des Definitionsbereichs:
        Berechne die Matrix H gemäß Gleichung (8) auf Seite 75
        Für jeden Indexblock J_n des Wertebereichs:
           Finde den optimalen Vektor b durch die Lösung von
              Gleichung (9) auf Seite 75
           Berechne ||x_{J_n} − Hb|| und aktualisiere q(n), falls das
           Resultat besser, d.h. kleiner, ist.
```

7. Zusammenfassung, Bewertung und Schlußfolgerungen

Das beschriebene diskrete Modell dient im folgenden als Grundlage für die Beschreibung der Eigenschaften von Systemen zur fraktalen Bildkodierung sowie für eine Implementierung und die Durchführung verschiedener Experimente. Dank des Modells ist man in der Lage, die Eigenschaften von Systemen zur fraktalen Bildkodierung so zu bestimmen, daß diese Beschreibung der Eigenschaften unabhängig von einer konkreten Implementierung ist. Dies ist der Inhalt des nachfolgenden Kapitels 4, in welchem Eigenschaften und Fähigkeiten von Systemen zur fraktalen Bildkodierung als Folgerung aus dem Modell des aktuellen Kapitels beschrieben werden.

Ein weiterer Vorteil eines solchen Modells, welches primär unter dem Aspekt der Implementierbarkeit entworfen wurde, ist die Vorgabe einer möglichen Strukturierung für eine Implementierung in verschiedene Teilaspekte. Dabei ergibt sich einerseits eine Aufteilung in technische, implementationsabhängige Details und andererseits erhält man inhaltliche Aspekte, die sich auf die Art der Kodierung beziehen. Eine solche Aufteilung zeigt sich z.B. bei der Konstruktion der Operatoren T_n.

Diese Aufteilung bzw. Strukturierung durch das Modell bietet darüber hinaus die Möglichkeit, bei einer Implementierung gezielt inhaltliche Teilaspekte bezüglich ihrer Wirkung auf das Gesamtsystem zu untersuchen. Basierend auf der Strukturierung aus diesem Kapitel wird dies im abschließenden Kapitel 5 bei der Darstellung und Auswertung verschiedener experimenteller Ergebnisse genutzt, indem z.B. verschiedene Partitionierungsverfahren innerhalb des Gesamtsystems zur fraktalen Bildkodierung implementiert und in ihren Wirkungen verglichen werden.

Mit einem solchen diskreten Modell lassen sich mit Sicherheit nicht alle Aspekte fraktaler Bildkodierung, wie z.B. der Einsatz allgemeiner kontrahierender Abbildungen, erschöpfend betrachten, doch besitzt es im Bezug auf einen Einsatz und eine Implementierung wesentliche Stärken gegenüber anderen Modellen, wie sie z.B. in Kapitel 2 vorgestellt wurden. Dies führte zur Verwendung eines diskreten Modells in dieser Arbeit anstelle anderer Modelle, wie z.B. des Modells von Barnsley.