Technische Universität Wien

DIPLOMARBEIT

Diskrete Iterierte Funktionensysteme

ausgeführt am Institut für

Automation

Abteilung für

Mustererkennung und Bildverarbeitung

der Technischen Universität Wien

unter Anleitung von O.Univ.Prof. Dr. W. KROPATSCH

durch

Michael A. Neuhauser

Matrikelnummer 8625686

Storkgasse 13/15, A–1050 Wien

26. November 1992 . . . . . .

Zusammenfassung

Iterierte Funktionensysteme (IFS) sind Mengen kontraktiver Funktionen. Sie definieren einen eindeutigen, fraktalen Attraktor, der als binäres Bild dargestellt werden kann. Ebenso definieren sie ein eindeutiges invariantes Maß, das als Grauwert– oder Farbbild dargestellt werden kann. IFS eignen sich für effiziente Bildcodierung, da bereits IFS mit wenigen Funktionen realistische Bilder natürlicher Objekte erzeugen können. In dieser Arbeit werden die mathematischen Grundlagen von IFS behandelt und die Eigenschaften von affinen Transformationen auf euklidischen Räumen untersucht; eine verständliche Darstellung affiner Transformationen der euklidischen Ebene wird vorgestellt. Danach wird eine Diskretisierung von Transformationen vorgestellt, die es ermöglicht, zwei effiziente Algorithmen zur Berechnung eines diskreten Attraktors zu präsentieren. Einer dieser Algorithmen wird mittels Bildpyramiden erweitert und verbessert. Weiters wird ein effizientes Verfahren zur Berechnung eines diskreten invarianten Maßes angegeben. Durch die Diskretisierung ergibt sich ein Unterschied von Attraktor und invariantem Maß zu ihren diskreten Entsprechungen. Für diesen Unterschied wird eine Fehlerschranke angegeben, aus der hervorgeht, daß eine für Bildcodierung ausreichende Genauigkeit immer erreicht werden kann.

Danksagung

Ich möchte Prof. Dr. Kropatsch herzlich für seine Betreuung und Unterstützung danken. Ebenso bedanke ich mich bei Dr. Pinz und Dipl. Ing. Bischof und allen anderen Mitarbeitern der Abteilung für ihre Hilfe. Weiters möchte ich Frau Regina Melzer und Herrn Andreas Gosch für Rat bezüglich mathematischer Fragen danken.

Abstract

Iterated function systems (IFS) are sets of contractive functions. They define a unique fractal attractor that can be represented as a binary image. They also define a unique invariant measure that can be represented as a greyvalue or color image. IFS are well suited for image coding, because even IFS with few functions are able to generate realistic images of natural objects. Firstly the mathematical basics of IFS will be presented in this work. Then the properties of affine transformation on euclidean spaces will be investigated. An convenient representation of affine transformations on the euclidian plane is developed. A discretization of the transformation is done next, this allows to present two efficient algorithms that compute a discrete attractor. By using image pyramids one of this algorithms is extended and enhanced. An efficient method for calculation of a discrete invariant measure is given. The discrete transformations introduce an error that results in a difference between the attractor, the invariant measure and their discrete counterparts. An error bound for this difference is shown. It follows from this error bound that it is always possible to achieve an accuracy that is sufficient for image coding purposes.

Inhaltsverzeichnis

1 Einleitung **1**

1.1 Bildcodierung 2

1.2 Einige Methoden zur Bildcodierung 4

1.2.1 Pixelcodierung 5

1.2.2 Voraussehende Codierung 6

1.2.3 Transformationscodierung 7

1.2.4 Andere Codierungsmethoden 8

1.2.5 Fraktale Codierung 9

1.3 Eine Einführung in Iterierte Funktionensysteme 10

2 Grundlagen Iterierter Funktionensysteme **13**

2.1 Deterministische IFS 14

2.2 Probabilistische IFS 16

3 Affine Transformationen **21**

3.1 Notation, grundlegende Definitionen und Begriffe 21

3.2 Affine Transformationen auf $\mathbb{R}^d$ 23

3.2.1 Kontraktionsfaktor einer affinen Transformationen 23

3.2.2 Zwei Darstellungen kontraktiver affiner Transformationen 24

3.3 Affine Transformationen auf $\mathbb{R}^2$ 26

3.3.1 Geometrische Transformationen und zugehörige Matrizen 26

3.3.2 Zerlegung eines linearen Operators 27

3.3.3 Kontraktionsfaktor einer affinen Transformation 37

3.3.4 Fixpunkt einer kontraktiven affinen Transformation 38

4 Vom IFS zum Bild **40**

4.1 Abtasten des Attraktors 40

4.2 Diskrete Transformationen 41

4.2.1 Eine Erweiterung der Diskretisierung 43

4.3 Diskreter Attraktor 44

4.3.1 Algorithmus RemoveBackground 44

4.3.2 Algorithmus RemoveBackground mit Bildpyramiden 47

4.3.3 Der Algorithmus SetForeground 51

4.3.4 Unterschied zwischen abgetastetem und diskretem Attraktor 53

4.4 Diskretes invariantes Maß 56

4.4.1 Unterschied zwischen abgetastetem und diskretem invarianten Maß 57

4.5 Verbesserung der diskreten Approximation 58

5 Zusammenfassung und Schluß **59**

Literaturverzeichnis **60**

Abbildungsverzeichnis

1.1 Bild eines Fraktals (Barnsleys Farn [5]) 1

1.2 Ein digitales Bild ist eine rechteckige Anordnung von Pixeln 2

1.3 Die ersten drei Kopien verschiedener Ausgangsbilder einer MCCM 11

2.1 Wirkung des Hutchinsonoperators auf ein Rechteck 16

2.2 Einige IFS mit zugehörigen Attraktoren 17

2.3 Wirkung des Markovoperators auf ein Maß 20

3.1 Orbit der Transformation $w(\bar{x}) = \mathfrak{S}(0.6, 0.9)\bar{x}$ 27

3.2 Orbit der Transformation $w(\bar{x}) = \mathfrak{R}(10°)\mathfrak{S}(0.9, 0.9)\bar{x}$ 27

3.3 Orbit der Transformation $w(\bar{x}) = \mathfrak{R}(10°)\mathfrak{S}(-0.9, 0.9)\bar{x}$ 28

3.4 Orbit der Transformation $w(\bar{x}) = \mathfrak{R}(10°)\mathfrak{S}(0.95, 0.9)\mathfrak{R}(5°)\bar{x}$ 28

3.5 Eigenvektoren der Länge 1 zum Eigenwert λ von $A = \left(\begin{smallmatrix} r & s \\ s & t \end{smallmatrix}\right)$ 33

3.6 Alle Eigenvektoren der Länge 1 zu den beiden Eigenwerten von $A = \left(\begin{smallmatrix} r & s \\ s & t \end{smallmatrix}\right)$ 34

4.1 Verschiedene Methoden der Pixelrepräsentation 43

4.2 Eine mit RemoveBackground generierte Pyramide von Barnsleys Farn 48

4.3 Schema von RemoveBackground mit Bildpyramide 49

4.4 RemoveBackground–Pyramide für IFS $\{w_1, w_2\}$ in $\mathbb{R}$ 50

4.5 Abgetastete und diskrete Farn–Attraktoren 53

Notation

Einige in dieser Arbeit verwendeten Notationen werden hier angeführt.

Notation	*Erläuterung*
$\mathbb{N}$	Die Menge der natürlichen Zahlen (inklusive 0).
$\mathbb{N}^+$	Die Menge der natürlichen Zahlen größer 0.
$\mathbb{Z}$	Die Menge der ganzen Zahlen.
$\mathbb{R}$	Die Menge der reellen Zahlen.
$\mathbb{R}^+$	Die Menge der reellen Zahlen größer 0: $\{x \in \mathbb{R} \mid x > 0\}$
$\mathbb{R}_0^+$	Die Menge der reellen Zahlen größer gleich 0: $\{x \in \mathbb{R} \mid x \geq 0\}$
$[a, b]$	Abgeschlossenes reelles Intervall: $\{x \in \mathbb{R} \mid a \leq x \leq b\}$
$[a, b)$	Rechts halboffenes reelles Intervall: $\{x \in \mathbb{R} \mid a \leq x < b\}$
$(a, b]$	Links halboffenes reelles Intervall: $\{x \in \mathbb{R} \mid a < x \leq b\}$
(a, b)	Offenes reelles Intervall: $\{x \in \mathbb{R} \mid a < x < b\}$
$:=$	Definitionsgleichung, „wird definiert als“
$\leftarrow$	Variablenzuweisung, „erhält den Wert“
$\lfloor x \rfloor$	floor von $x \in \mathbb{R}$: $\lfloor x \rfloor := \max\{n \in \mathbb{Z} \mid n \leq x\}$
$\lceil x \rceil$	ceiling von $x \in \mathbb{R}$: $\lceil x \rceil := \min\{n \in \mathbb{Z} \mid n \geq x\}$
$\emptyset$	Die leere Menge.
$\#A$	Die Anzahl der Elemente der endlichen Menge A.
$\setminus$	Mengendifferenz: $A \setminus B := \{x \in A \mid x \notin B\}$
$\times$	Mengenprodukt: $A \times B := \{(p, q) \mid p \in A \text{ und } q \in B\}$
A^T	Transponierte der Matrix A.
$(f \circ g)(x)$	Hintereinanderausführung von Funktionen: $(f \circ g)(x) := f(g(x))$
$f^{\circ n}(x)$	n-fache Anwendung der Funktion f auf x: $f^{\circ n}(x) := f(f^{\circ n-1}(x))$ für $n \in \mathbb{N}^+$, $f^{\circ 0}(x) := x$

Kapitel 1

Einleitung

Obwohl das Bild in Abb. 1.1 aus ca. 120000 Punkten besteht, kann es mit nur 96 Bytes gespeichert werden. Das ist möglich, weil dieses Bild ein Fraktal ist. Es wurde mit einem Iterierten Funktionensystem (IFS) generiert, einem speziellen Typ von Fraktalcode. Allgemeines zu Fraktalen ist [35] zu entnehmen.

Abbildung 1.1: Bild eines Fraktals (Barnsleys Farn [5])

1.1 Bildcodierung

Ein *digitales Bild* wird hier als rechteckige Anordnung von *Bildelementen* (oder *Pixeln* vom englischen *picture-element*) betrachtet. Auf den Prozeß, der das Bild einer wirklichen Szene über einen bildgebenden Sensor mittels Abtastung im Raum und Quantisierung des Signals in ein digitales Bild umwandelt, soll hier nicht eingegangen werden; siehe dazu [46] und [25], wo auch eine Übersicht über Codierungsmethoden geboten wird. Ein Pixel besteht aus einer Ortsangabe (Zeile und Spalte) und einem Pixelinhalt. In Abb. 1.2 ist das Schema eines solchen digitalen Bildes zu sehen; (z, s, p_{zs}) bezeichnet das Pixel mit Zeile z, Spalte s und Pixelinhalt p_{zs}. Der Inhalt eines Pixel ist vom Typ des Bildes abhängig. Dies kann ein einzelnes Bit

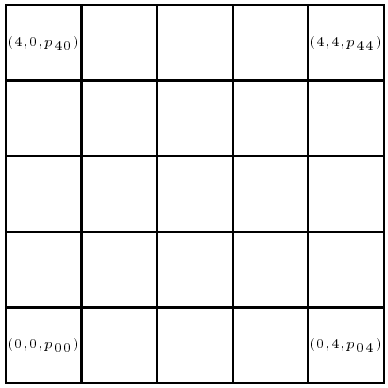


Abbildung 1.2: Ein digitales Bild ist eine rechteckige Anordnung von Pixeln

bei *Binärbildern* sein (also Bildern, in denen nur die Werte Schwarz und Weiß vorkommen); oder ein Grauwert aus einem geeigneten Wertebereich (typisch 0 bis 255, also ein Byte) bei *Grauwertbildern*; oder z.B. ein Farbwert, bestehend aus drei Farbkomponenten (entsprechend drei Bytes), bei *Farbbildern*.

Die *Auflösung* eines Bildes gibt an, wie fein gerade noch unterscheidbare Details im Bild sein können. Hat man zwei Bilder unterschiedlicher Auflösung derselben Szene, so kann man im Bild höherer Auflösung feinere Strukturen erkennen als im Bild niedriger Auflösung. Dort hat man jedoch einen besseren Überblick.

Die *Größe* eines Bildes wird in Anzahl der Pixelzeilen mal Anzahl der Pixelspalten angegeben, z.B. entspricht ein 512×512 Grauwertbild mit 256 Grauwertstufen in etwa einem Schwarz/Weiß Videobild. Wenn man dieses Bild in Form von *rohen Bilddaten* abspeichert (also in Matrixform mit einem Byte pro Pixel), so werden dafür 256 KByte benötigt. Für ein 1024×1024 Farbbild mit 256 Farbstufen pro Farbkanal werden bereits 3 MByte benötigt; diese Anzahl von Farbstufen ist für eine realistische Darstellung ohne Streifenbildung bei kontinuierlichen Farbverläufen notwendig.

Da Bildverarbeitung immer mehr an Bedeutung gewinnt und daher immer mehr digitale Bilder verarbeitet werden, ergeben sich aus den hohen Speicheranforderung dieser Bilder Probleme

für ihre Speicherung, Übertragung und Verarbeitung. Weiters ist die Darstellung von digitalen Bildern durch rohe Bilddaten nicht adäquat für komplexe Anwendungen wie Bildanalyse oder Mustererkennung. Daher müssen geeignete Repräsentationen für digitale Bilder gefunden werden. Unter einem *Bildcode* versteht man eine klar definierte Vorschrift, wie die rohen Bilddaten eines digitalen Bildes in eine codierte Form zu bringen sind (*Codierung*) und wie die originalen Bilddaten (oder die eines Bildes möglichst geringer Abweichung) aus dieser codierten Form zu rekonstruieren sind (*Decodierung*). Meistens wird die codierte Form eines Bildes selbst auch als Code bezeichnet. Sind rekonstruiertes und originales Bild immer identisch, so spricht man von einem *fehlerfreien*, sonst von einem *fehlerbehafteten* Code.

An einen Bildcode werden folgende Anforderungen gestellt:

- hohe Kompressionsraten

  Unter der *Kompressionsrate* eines Codierungsverfahren versteht man das Verhältnis des Speicherplatzbedarfes der rohen Bilddaten und des codierten Bildes; ist dieses Verhältnis groß, so spricht man von einer hohen Kompressionsrate. Manchmal wird diese Größe auch in Bits pro Pixel (BpP) angegeben. Für fehlerfreie Codierung kann man mit Standardmethoden etwa Kompressionsraten von 2:1 bis 3:1 erzielen, bei fehlerbehafteter Codierung 20:1 bis 30:1.

- schnelle Codierung und Decodierung

- gute Qualität des rekonstruierten Bildes

  Die Qualität des rekonstruierten Bildes kann man objektiv messen (eine mathematische Funktion mißt die Abweichung von originalem und rekonstruiertem Bild), oder subjektiv durch einen Menschen beurteilen lassen.

  Um hohe Kompressionsraten zu erzielen, nimmt man einen Informationsverlust in Kauf. Ein Bild kann umso höher komprimiert werden, je geringer sein Informationsgehalt ist. Man verzichtet dabei auf Informationen, die der menschliche Betrachter nicht unbedingt benötigt. Dabei wird die Physiologie des Sehvorganges ausgenutzt; dazu später mehr. Manche Anwendungen, wie z.B. die medizinischen Datenverarbeitung, lassen wegen ihrer Sensitivität diese Vorgehensweise nicht zu.

- universell anwendbar

  Ein Bildcode sollte vergleichbare Resultate bezüglich der ersten drei Punkte für eine große Klasse von Bildern liefern.

- progressive Übertragung

  Wird ein Bild übertragen, so sollte möglichst bald eine Übersicht des Bildes sichtbar sein, genauere Approximationen werden dann Schritt für Schritt aufgebaut. Dies hat den Vorteil, daß ein nicht gewünschtes Bild früh erkannt wird und die Übertragung abgebrochen werden kann.

- anwendungsbezogen

  Die Struktur der codierten Bilddaten sollte die Lösung von Verarbeitungsproblemen unterstützen.

Diese Anforderungen widersprechen sich zum Teil, deshalb muß jeder Bildcode einen Kompromiß eingehen und ist dementsprechend für eine bestimmte Anwendungsklasse geeignet. Die Mehrzahl der im folgenden erwähnten Bildcodes legt keinen Wert auf Anwendungsbezogenheit und versucht nur die anderen Punkte der obigen Auflistung zu erfüllen.

Folgende Eigenschaften werden zur Codierung digitaler Bilder herangezogen:

- Redundanz im Bild

  Es wird ausgenutzt, daß der Bildinhalt in gewissem Sinn glatt, vorhersagbar, nicht zufällig ist. Ein Bild mit konstantem Grauwert kann z.B. durch den Grauwert eines einzelnen Pixels beschrieben werden. Enthält jedoch ein Bild weißes Rauschen, so ist keine Redundanz in ihm enthalten. IFS benützen die Selbstähnlichkeit eines Bildes, um es zu codieren.

- Eigenschaften des menschlichen Sehvorganges

  Das Auge ist z.B. viel empfindlicher für feine Details im Helligkeitssignal, als dies beim Farbssignal der Fall ist [3]. Deshalb kann ohne merklichen Qualitätsverlust ein Farbbild in zwei Farbteile und einen Helligkeitsteil zerlegt werden, wobei die Farbteile mit weniger Stufen quantisiert und mit weniger Werten abgetastet werden als die Helligkeit. Weiters ist das Auge weniger empfindlich gegen Energien hoher räumlicher Frequenz als solcher niedriger räumlicher Frequenz. Dies kann man so ausnutzen: Ein Bild wird in den Frequenzbereich transformiert, die hohen Frequenzen werden mit weniger Stufen quantisiert als die niedrigen Frequnzen; das decodierte Bild wird subjektiv keinen großen Qualitätsverlust zeigen.

- Wissen über das Abgebildete (modellbasierte Codierung)

  Wenn man z.B. weiß, daß man das Bild eines Textes hat, der in einer bestimmten Schriftart gesetzt wurde, so kann man die Zeichen erkennen und nur diese Information codieren. Man muß also ein Modell der abgebildeten Objekte besitzen, was klarerweise eine starke Einschränkung bezüglich der Anwendbarkeit solcher Codes darstellt. Andererseits kann man solch eine Codierung zur Analyse des Bildes heranziehen; im obigen Beispiel wäre die Frage „Wird in diesem Bild Text der bestimmten Schriftart abgebildet?“ IFS können auch auf diese Weise betrachtet werden, da sie Auskunft über das Ausmaß der Selbstähnlichkeit des Bildes geben.

1.2 Einige Methoden zur Bildcodierung

Bildcodes können nach Jain [25] in folgende Klassen eingeteilt werden:

- Pixelcodierung

  Hier wird jedes Pixel für sich behandelt, Abhängigkeiten unter den Pixeln werden ignoriert.

- voraussehende Codierung

  Das zugrunde liegende Prinzip ist, die Redundanz zwischen aufeinander folgenden Pixeln zu eliminieren und nur die neue Information zu codieren.

- Transformationscodierung

  Das Bild wird über eine geeignete Transformation in eine Darstellung übergeführt, sodaß wenige Abtastwerte einen großen Teil der Information beeinhalten.

- andere Methoden

  Generalisierungen oder Kombinationen der zuvor angeführten Methoden. IFS fallen in diese Kategorie, da sie die Redundanz des Bildes auf einer großen Ebene ausnutzen (Selbstähnlichkeit).

Im folgenden werden einige bekannte Bildcodes und jüngste Entwicklungen auf dem Gebiet der fraktalen Bildcodierung vorgestellt.

1.2.1 Pixelcodierung

Bereits die Speicherung der rohen Bilddaten kann als Code dieser Klasse bezeichnet werden.

Entropiecodierung

Es ist dies eine Klasse von Methoden, die ganz allgemein zur Codierung einer beliebigen Nachricht verwendet wird. Eine Nachricht ist dabei eine Kette von Symbolen aus einem bestimmten Alphabet. Jede Nachricht wird durch eine Bitfolge fehlerfrei codiert.

Angenommen, man hat ein Alphabet mit n Symbolen, deren Auftrittswahrscheinlichkeit p_i, $i = 1, \ldots, n$ ist. Werden alle Symbole mit Codewörtern (also Wörtern über dem Alphabet $\{0, 1\}$) fixer Länge (mindestens $\lceil \log_2 n \rceil$) codiert, so werden Nachrichten sicherlich länger sein, als wenn man den Symbolen mit geringerer Auftrittswahrscheinlichkeit längere Codewörter zuweist als solchen hoher Wahrscheinlichkeit. Eine systematische Methode, einen solchen Code variabler Länge zu generieren, wurde von Huffman in [22] vorgestellt; man nennt dieses Verfahren *Huffmancodierung*.

Folgende Methode wurde von Ziv und Lempel in [53] vorgestellt: Während des Codierungsprozesses wird ein Wörterbuch von bereits codierten Subnachrichten angelegt und nochmals auftretende Substrings durch Verweise in dieses Wörterbuch ersetzt. Eine darauf basierende, verbesserte und weit verbreitete Methode, die *LZW–Codierung* wurde von Welch in [50] angegeben.

Eine jüngere Entwicklung ist die *arithmetische Codierung* (Witten et al. [51]), die es erlaubt, Symbole mit weniger als einem Bit zu codieren, und die eine klare Trennung zwischen Codierung und Modell vorsieht. Das Modell liefert die oben angeführten Wahrscheinlichkeiten p_i, die ja notwendig sind, um die Nachricht gut zu codieren. Meistens sind diese Werte nicht a priori bekannt, sondern werden aus den Häufigkeiten der bereits codierten Symbole abgeleitet; jedem Symbol wird dabei die Wahrscheinlichkeit „Häufigkeit dividiert durch Länge der bisher codierten Nachricht“ zugeordnet. Es sind aber durchaus andere Modelle üblich, wie z.B. solche mit

fixen Wahrscheinlichkeiten, oder solche wo die Wahrscheinlichkeit von den unmittelbar zuvor aufgetretenen Symbolen abhängig sind [1].

Die Anwendung der Entropiecodierung auf Bilder ist vielfach. Man kann direkt die Grau- oder Farbwerte eines Bildes auf diese Weise codieren. Entropiecodierung wird aber auch zur Verbesserung der Komprimierungsraten anderer Codes herangezogen.

Lauflängencodierung (Run-Length-Coding)

Das Bild wird zeilenweise abgetastet. Pixel, die den selben Inhalt haben (ein Lauf oder „Run“), werden zusammengefaßt, es werden Länge und Pixelwert codiert. Eventuell wird noch eine Entropiecodierung dieser Werte durchgeführt. Werden Binärbilder auf diese Weise codiert, so muß man nur die Lauflänge codieren, da sich ja schwarze und weiße Läufe immer abwechseln.

Konturcodierung

Das Bild wird in Bereiche mit homogenem Bildwert eingeteilt (segmentiert), diese Bereiche werden dann über ihre Ränder und den Bildwert definiert. Möglichkeiten, die Ränder zu codieren, bieten *Freeman Chain Code*, *Crack Code* und *RULI Code*.

Blockcodierung

Medial (symmetric) axis: Es wird eine Überdeckung des Bildes durch maximale Blöcke (Rechtecke, Scheiben, ...) gleichen Bildwertes gesucht. Codiert werden das Zentrum, Ausdehnung und Bildwert dieser Blöcke.

Quadtree: Ein Quadtree ist ein Baum, dessen Knoten Teile des Bildes repräsentieren. Hat ein solcher Bildteil einen homogenen Bildwert, so hat der Knoten keine Nachfolger, der Bildwert wird im Knoten vermerkt. Ist der Bildteil inhomogen, so wird er geviertelt und der Knoten erhält die Knoten dieser Viertel als Nachfolger. Der so entstehende Baum wird geeignet codiert. Über das Homogenitätskriterium kann man steuern, ob eine fehlerfreie oder fehlerbehaftete Codierung entsteht. Eine neue, schnelle Methode der Quadtree–Codierung wurde von Berger et al. in [10] vorgestellt.

1.2.2 Voraussehende Codierung

Das Bild wird zeilenweise codiert. Aus den bisher codierten Pixelwerten wird der Wert des nächsten Pixels vorhergesagt, die Differenz von vorhergesagtem und tatsächlichem Wert wird codiert. Da die Pixel meistens korelliert sind, werden kleine Fehlerwerte weniger oft vorkommen als große. Über Entropiecodierung kann so eine Bildkompression erreicht werden [41]. Wird der Fehler noch quantisiert, so kann man auf Kosten der Bildqualtität eine größere Komprimierungsrate erzielen.

Die Voraussagefunktion kann entweder nur die vorangehenden Pixel in der aktuellen Zeile oder zusätzlich auch noch die in den vorangehenden Zeilen betrachten. Letzteres ergibt höhere Komprimierungsraten. Adaptive Methoden passen die Voraussagefunktion an das zu codierende Bild an [52].

1.2.3 Transformationscodierung

Blockquantisierung

Das Bild wird über eine (meist separable) orthogonale Transformation vom Orts- in den Frequenzbereich transformiert. Es ergibt sich eine rechteckige Anordnung von Koeffizienten dieser Transformation. Diese Koeffizienten können nach ihrer Wichtigkeit bezüglich der Qualität des rekonstruierten Bildes fix, über ein festgelegtes Schema (*Zonal Coding*) geordnet werden, oder adaptiv nach ihren aktuellen Werten (*Threshold Coding*). Weniger wichtige Koeffizienten können mit weniger Stufen quantisiert oder überhaupt weggelassen werden [16]. Man kann beweisen, daß die Karhunen-Loève-Transformation die für diese Art von Codierung am besten geeignete Transformation ist [25]. Da sie aber einen hohen Rechenaufwand erfordert, werden üblicherweise schnelle unitäre Transformationen wie Fourier-, Hadamard-, Slant- [44] oder Cosinustransformation eingesetzt.

In der Praxis wird die Transformation meist nicht auf das ganze Bild, sondern auf Subblöcke desselben angewendet, deshalb der Name Blockquantisierung. Die Koeffizienten dieser Subblöcke werden dann wie oben beschrieben codiert.

Laplacepyramide

Eine *Bildpyramide* ist eine Sammlung von Bildern unterschiedlicher Auflösung aber gleichen Bildinhalts [27]. Die einzelnen Bilder werden von hoher zu niedriger Auflösung geordnet und numeriert, und ergeben so die *Ebenen* der Pyramide. Das Originalbild wird mit G_0 bezeichnet, das Bild nächst kleinerer Auflösung mit G_1, usw.

Eine *Grauwertpyramide* wird dadurch aufgebaut, daß ein Pixel der Ebene G_{i+1} als Wert das gewichtete Mittel der Pixel innerhalb eines bestimmten *Reduktionsfensters* auf Ebene G_i erhält. Der *Reduktionsfaktor* gibt dabei die Rate an, mit der die Anzahl der Pixel von Ebene zu Ebene abnimmt. Die klassische Pyramide ergibt sich, wenn jeweils zwei mal zwei Pixel ein Pixel der höheren Ebene ergeben; diese Pyramide wird als $2 \times 2/4$–Pyramide bezeichnet, wobei 4 der Reduktionsfaktor ist. Jede Ebene G_{i+1} entspricht einer Tiefpaß gefilterten Version der Ebene G_i, da durch die Mittelung hohe Frequenzen (also feine Details) unterdrückt werden.

Eine *Laplacepyramide* [11] entsteht aus einer Grauwertpyramide, indem man die dort vorhandene Redundanz beseitigt. In der Grauwertpyramide sind ja die tiefen Frequenzen in allen Ebenen vorhanden. Aus einer Ebene G_{i+1} berechnet man sich durch Interpolation eine Approximation $\tilde{G}_i$ der darunterliegende Ebene. Die Ebene i der Laplacepyramide ist dann die Differenz zwischen der tatsächlichen und der approximierten Ebene i der Grauwertpyramide: $L_i := G_i - \tilde{G}_i$. L_i besteht nur aus den Frequenzen, die in G_i enthalten sind, aber in G_{i+1} fehlen;

L_i ist also eine Bandpaß gefiltere Version von G_i. Setzt man noch die Ebene höchster Auflösung $L_n := G_n$, so kann man das Bild aus der Laplacepyramide exakt rekonstruieren. Mit der Laplacepyramide ist eine progressive Übertragung möglich. Indem nacheinander die Ebenen L_n, L_{n-1}, ... übertragen werden, entsteht eine immer besser werdende Approximation des Originalbildes, das nach Übertragung von L_0 fehlerfrei wiedergegeben wird. Da die Interpolationsfunktion eine gute Näherung liefert, kann man die Laplacepyramide durch Entropiecodierung und eventuelle Quantisierung effizient codieren.

Die mit dieser Methode erzielbaren Kompressionsraten werden für die $3 \times 3/2$ (Mayer, Kropatsch) und die $5 \times 5/4$ (Burt [11]) Pyramide in [37, 38, 36] verglichen. Es werden für beide Pyramiden Kompressionsraten von 1:10 bis 1:20 angegeben; aufgrund des niedrigeren Reduktionsfaktors ist aber die $3 \times 3/2$ Pyramide vorzuziehen, da sie mehr Auflösungsstufen bietet.

Neben der Grauwert- und der Laplacepyramide kann man auch Pyramiden definieren, die eine abstraktere Beschreibung des Bildes auf unterschiedlichen Auflösungsebenen liefern. So ordnet z.B. die *Kurvenpyramide* [26] die Kurven eines Bildes nach ihrer Länge.

Wavelets

Die Wavelettransformation [34] verbindet Repräsentation im Frequenz- und Ortsbereich. Ein Bild D der Größe 512×512 wird in vier Bilder der Größe 256×256 zerlegt. Eines dieser Bilder ist ein Ortsbild reduzierter Auflösung. Die drei anderen entsprechen räumlich orientierten, gefilterten Frequenzkanälen, zwei enthalten jeweils die horizontalen und vertikalen hohen Frequenzen (also vertikale und horizontale Kanten), das dritte enthält die hohen Frequenzen in beiden Richtungen (also die Ecken). Durch geeignete Quantisierung (hohe Frequenzen sind nicht so ausschlaggebend wie niedrige) und Entropiecodierung kann wieder eine Komprimierung erreicht werden.

1.2.4 Andere Codierungsmethoden

Hybride Codierung

Hybride Codierung ist eine Kombination von voraussehender und Transformationscodierung. Aus den Koeffizienten des vorangehenden Blockes werden die Koeffizienten des aktuellen Blocks vorhergesagt, der Fehler wird meist gering sein und kann entropiecodiert gespeichert werden. Diese Methode kann noch um eine adaptive Komponente erweitert werden, die die Vorhersagefunktion an das Bild anpaßt.

Der JPEG (Joint Photographics Experts Group) CCITT/ISO Standard für Komprimierung photographischer (Grauwert- und Farb-) Bilder ist eine hybride Codierung [2]. Das Bild wird in 8×8 Blöcken einer diskreten Cosinustransformation (DCT) unterzogen. Die so erhaltenen 64 Koeffizienten pro Block werden nach ihrer Frequenz geordnet und entsprechend quantisiert. Der „Gleichstromkoeffizient“, also jener Wert, der die Grundhelligkeit des Blocks angibt, wird einer voraussehenden Codierung unterzogen, es wird also die Differenz zum vorhergesagten Wert

codiert. Die so gebildeten Koeffizienten werden Huffman- oder arithmetisch codiert. Kompressionsraten von 10:1 bis zu 50:1 werden für diese Methode in [3] angegeben. In [2] wird eine Kompressionsrate von 64:1 bei „brauchbarer“ und 18:1 bei exzellenter Qualität angegeben.

Vektorquantisierung

Das Bild wird in Blöcke fixer Größe zerlegt. Für jeden Block des Bildes wird jener Block in einer gegeben Menge (dem *Codebuch*) von vordefinierten Blöcken gesucht, der die geringste Abweichung zum Bildblock zeigt. Der Index des gefundenen Blockes wird (über Entropiecodierung) codiert. Das Codebuch kann fix sein oder adaptiv über eine Serie von Bildern definiert werden. Hsieh et al. präsentierten kürzlich in [21] einen schnellen Algorithmus zur Erzeugung des Codebuchs.

1.2.5 Fraktale Codierung

Das Gebiet der fraktalen Bildcodierung hat in letzter Zeit viel Aufmerksamkeit auf sich gezogen, nicht zuletzt wegen der immens hohen Kompressionsraten von 10000:1, die von Barnsley in [9] angegeben wurden. Unter anderem wurden Fraktale für die geometrische Modellierung in Systemen zur Erzeugung realistischer, computergenerierter Grafiken verwendet [18, 17, 15].

Neben IFS werden aber auch oft andere Codierungsmethoden als fraktal bezeichnet. Fisher [14] und Jacquin [24] geben eine Methode an, bei der die Selbstähnlichkeit eines Bildes auf Blockebene ausgenützt wird. Es werden kontraktive Funktionen gesucht, die Grauwertblöcke auf kleinere Grauwertblöcke desselben Bildes transformiern. Die Funktionen verändern dabei Lage (Drehen in 90° Grad Schritten, verschieben und Spiegeln) und Grauwerte (lineare Transformation der Grauwerte) der Blöcke. Für einen gegebenen Block wird unter allen möglichen Blöcken jener gesucht, der nach Transformation den gegebenen Block am besten approximiert. So wird jeder Block durch eine Funktion erzeugt; alle diese Funktionen ergeben zusammen eine einzige Funktion. Das rekonstruierte Bild ist der (approximierte) Fixpunkt der so erhaltenen Funktion. Fisher gibt in [14] eine Kompressionsrate von 16.5:1 bei einem Least-Square-Fehler von 10.4 an; Jacquin berichtet in [24] eine Kompressionsrate von 9:1 bei einem Signal/Rauschverhältnis von 27.7 dB. Beide verwenden das bekannte „Lena“ Bild.

Pentland und Horowitz erläutern in [43] eine Methode, bei der ein Bild einer Wavelettransformation unterzogen wird, die drei Frequenzkanäle (horizontal, vertikal und nach beiden Richtungen) getrennt einer Vektorquantisierung unterworfen werden. Dabei wird Selbstähnlichkeit der Frequenzbilder unterschiedlicher Auflösung ausgenutzt, um die Codierung der Vektorquantisierung zu optimieren. Es wird eine Kompressionsrate von 13.3:1 bei einem Signal-/Rauschverhältnis von 30 dB als typisch für diese Methode angegeben.

Wenn man jedoch von den von Barnsley angegeben Resultaten absieht, die näheren Details seiner Methode wurden bisher von ihm nicht angegeben, war bisher niemand in der Lage, eine praktikable Bildcodierung mittels IFS zu publizieren. Dies scheiterte vor allem daran, für ein gegebenes Bild eine Codierung mittels IFS zu finden, sodaß das rekonstruierte Bild von geeigneter Qualität ist. Dieses Problem ist als *inverses Problem* bekannt. Da dies ein schweres

Problem ist, haben verschiedene Autoren [20, 33] einen halbautomatischen Ansatz zu dessen Lösung propagiert. Andere Ansätze versuchen das inverse Problem über die Optimierung eines Fehlerkriteriums zu lösen [31, 32]; Vrscay verwendet in [49] zusätzlich genetische Suchalgorithmen. Für einige eingeschränkte Fälle konnten jedoch bereits automatische Lösungen gefunden werden: Mazel und Hayes geben in [39] zwei Verfahren zur Codierung diskreter, eindimensionaler Signale an, die strikte bzw. partielle Selbstaffinität der Signale ausnutzen; aus den diskreten Funktionswerten werden über eine extensive Suche die am besten geeigneten Transformationen berechnet. In [29, 28] geben Kropatsch et al. eine Lösung des inversen Problems für eindimensionale binäre Pixelzeilen unter Verwendung strikter Selbstaffinität an; der IFS Code wird dabei direkt aus den geometrischen Eigenschaften der Pixelzeile berechnet.

1.3 Eine Einführung in Iterierte Funktionensysteme

Ein IFS ist eine Menge von Funktionen. Die Wirkungsweise dieser Funktionen läßt sich mit einem Gerät veranschaulichen, das wohl jedem vertraut ist: einer Kopiermaschine, die verkleinern kann. Jede Konfiguration einer solchen Maschine entspricht einer Funktion, die in einem IFS verwendet werden kann. Die einzige Bedingung dabei ist, daß die von der Kopiermaschine erzeugte Kopie eines Bildes, also der Output der Kopiermaschine, etwas verkleinert ist.

Wird nun dieser Output als Input für die Maschine verwendet, der resultierende Output wieder als Input usw., zeigt sich folgendes: Die Kopien des Bildes werden kleiner und kleiner, bis früher oder später (abhängig vom Verkleinerungsfaktor) nur noch ein einziger schwarzer Punkt zu sehen ist. Dieses Endresultat ergibt sich immer, egal welches Ausgangsbild ursprünglich als Input für die Kopiermaschine verwendet wurde.

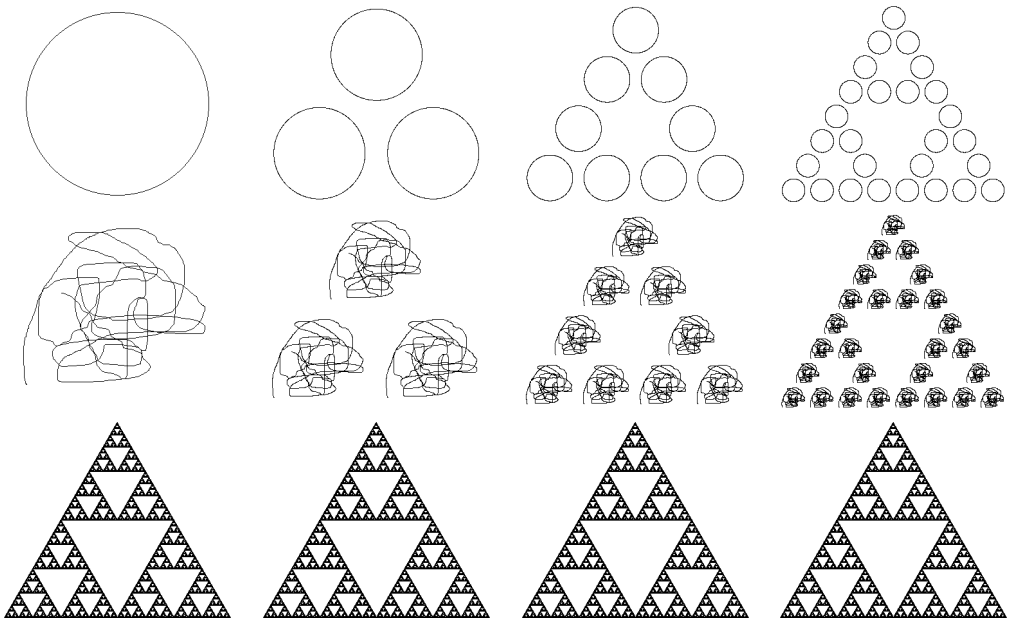
Weit interessantere und vielfältigere Endresultate zeigen sich, wenn die Kopiermaschine zu einer „Multi Contraction Copy Machine“ (MCCM) [42] umgebaut wird:

Statt eines Linsensystems zur Verkleinerung werden nun mehrere unabhängige Linsensysteme verwendet, in Analogie zu IFS, die auch aus mehreren Funktionen bestehen. Sie alle verkleinern das Input–Bild und kopieren es auf das Output–Bild, und zwar abhängig von bestimmten Faktoren: Der erste Parameter der MCCM bestimmt die Anzahl der Linsensysteme, die verwendet werden sollen. Der zweite Parameter hält den Verkleinerungsfaktor jedes einzelnen Linsensystems fest, und der dritte beschreibt die Position des Ergebnisses von jedem einzelnen Linsensystem auf dem Output–Bild.

Betätigt man die MCCM einmal, wird das Input–Bild mit verschiedenen Verkleinerungsfaktoren kopiert und die Ergebnisse in einem bestimmten Muster auf dem Output–Bild angeordnet. Das Output–Bild der MCCM wird wieder als Input–Bild verwendet usw., wie im obigen Experiment für die Kopiermaschine. War bei dieser das Endresultat ein schwarzer Punkt, so ist bei der MCCM das Endresultat ein fraktales Bild, dessen Aussehen nur von den drei Parametern der MCCM abhängt.

Als Beispiel soll eine MCCM mit drei aktiven Linsensystemen genommen werden, die das Input–Bild jeweils um den Faktor $\frac{1}{2}$ verkleinern und die Ergebnisse in Form eines gleichseitigen Dreiecks

anordnen. Abb. 1.3 zeigt die Anwendung der MCCM auf verschiedene Ausgangsbilder. Man sieht, daß das Endresultat immer das gleiche sein wird, unabhängig vom Ausgangsbild.



Ausgangsbild 1. Kopie 2. Kopie 3. Kopie

Abbildung 1.3: Die ersten drei Kopien verschiedener Ausgangsbilder einer MCCM

Das Ausgangsbild in der letzten Zeile zeigt spezielles Verhalten; es ändert sich nicht, egal wie oft die MCCM darauf angewendet wird. Diese Eigenschaft besitzt genau ein Bild, nämlich das Endresultat, das von allen Ausgangsbildern durch Iteration erreicht wird. Das Endresultat der MCCM von Abb. 1.3 ist das sogenannte Sierpinski Dreieck [5].

Nachdem die Funktionsweise einer MCCM beschrieben wurde, kann die Analogie zu IFS gezogen werden. Die IFS, die in dieser Arbeit behandelt werden, bestehen aus endlich vielen kontraktiven affinen Transformationen. Man kann sich die Effekte dieser Transformationen auf geometrischer Ebene vorstellen: Sie beschreiben Rotation, Skalierung, Spiegelung und daraus zusammengesetzte Operationen. Jede Konfiguration der letzten beiden Parameter einer MCCM entspricht einer solchen Transformation[1].

Es werden alle Transformationen auf eine Ausgangsmenge angewendet; die einzelnen Ergebnisse werden vereinigt, wie bei der MCCM die verschiedenen Linsensysteme zusammen ein Output–Bild produzieren. Auf dieses Ergebnis werden wiederum alle Transformationen angewendet usw.

[1]Umgekehrt gilt dies jedoch nicht, da die Skalierung, die eine affine Transformationen durchführt, nicht in allen Richtungen gleichförmig sein muß.

Analog zur MCCM produziert auch ein IFS unabhängig vom Ausgangsbild ein Endresultat, den sogenannten Attraktor. Dieser Attraktor ist eine Menge, die von dem IFS generiert wird.

Nimmt man den Attraktor als Ausgangsmenge, ändert er sich nicht, egal wie oft die Transformationen des IFS darauf angewendet werden.

Jede Transformation eines IFS läßt sich mit sechs reellen Zahlen beschreiben. Das erklärt den geringen Speicherbedarf von Bildern, die durch ein IFS codiert sind. Dies, und die Tatsache, daß fraktale Bilder mit herkömmlichen Methoden nur mit großem Speicheraufwand codierbar sind, läßt es angebracht erscheinen, IFS genauer zu studieren.

Man möchte den Attraktor als digitales Bild darstellen, was mit bisherigen Methoden sehr aufwendig war, da diese Algorithmen den Attraktor in $\mathbb{R}^2$ annähern, ohne die Bildschirmauflösung zu berücksichtigen. Daraus ergibt sich weiters, daß es nicht eindeutig ist, wann der Prozeß beendet werden soll, um ein befriedigendes Ergebnis sowohl hinsichtlich der Bildqualität als auch der Ausgabezeit zu liefern.

In dieser Arbeit werden diskrete IFS eingeführt, die durch eine Diskretisierung der verwendeten Funktionen zustande kommen. Dadurch wird es möglich, das Problem der Attraktordarstellung im diskreten Raum zu behandeln und einen diskreten Attraktor direkt in diesem diskreten Raum zu berechnen. Zwei Algorithmen die einen solchen diskreten Attraktor berechnen werden vorgestellt. Zur weiteren Beschleunigung der Ausgabe werden Bildpyramiden [27] verwendet.

Im Rahmen von grundlegenden Vorbereitungen werden zuerst die mathematischen Grundlagen von IFS vorgestellt, sowie affine Transformationen näher untersucht.

Kapitel 2

Grundlagen Iterierter Funktionensysteme

Es wird hier die grundlegende Theorie Iterierter Funktionensysteme dargelegt, wie sie von Hutchinson in [23] dargestellt und von Barnsley in [5, 6] erweitert wurde. Kürzlich vorgenommene Erweiterung Iterierter Funktionensysteme, wie Lineare Markov IFS [30], Language Restricted IFS [45] oder Recurrent IFS [7] werden hier nicht behandelt, obwohl insbesondere die später vorgenommene Diskretisierung sehr wohl auch auf diese erweiterten IFS anwendbar ist.

Notation und Definitionen folgen teilweise [5, 13]. Wenn nicht anders erwähnt, sind Beweise für die Sätze in [23, 5, 13] zu finden. Eine leicht verständliche Einführung in das Gebiet ist z.B. [42].

(X, d) bezeichne in diesem Kapitel einen vollständigen metrischen Raum mit Distanzfunktion d.

Definition 2.1 Eine Zahl $c \in \mathbb{R}_0^+$, heißt *Kontraktionsfaktor* einer Abbildung w auf X genau dann, wenn

$$d(w(x), w(y)) \leq c d(x, y) \qquad \forall\, x, y \in X \tag{2.1}$$

und es kein $k \in [0, c)$ mit

$$d(w(x), w(y)) \leq k d(x, y) \qquad \forall\, x, y \in X \tag{2.2}$$

gibt.

Definition 2.2 Eine Abbildung w auf X heißt *kontraktiv* genau dann, wenn ihr Kontraktionsfaktor $c < 1$ ist.

Definition 2.3 Ein $x \in X$ heißt *Fixpunkt* der Abbildung $w : X \to X$ genau dann, wenn es $w(x) = x$ (die *Fixpunktgleichung*) erfüllt. Die Fixpunkte von w sind also genau jene Elemente von X, die unter w auf sich selbst abgebildet werden.

Definition 2.4 Für eine kontraktive Abbildung $w : X \to X$ heißt die Folge $\{x_n\}$, $n = 0, 1, 2, \ldots$, mit $x_0 \in X$ und $x_{n+1} = w(x_n)$ für $n \geq 0$, der *Orbit* von x_0 unter w.

Einen Zusammenhang zwischen Fixpunkten einer kontraktiven Abbildung und ihren Orbits liefert der *Banachsche Fixpunktsatz* [19] (hier in eingeschränkter Form):

Satz 2.5 (Banachscher Fixpunktsatz) *Sei $w : X \to X$ eine kontraktive Abbildung auf X, dann hat die Fixpunktgleichung $x = w(x)$ eine eindeutige Lösung $x_F \in X$, und der Orbit jedes Anfangselements $x_0 \in X$ unter w konvergiert zu diesem eindeutigen Fixpunkt.*

Bisher wurden Elemente von X und ihre Bilder unter kontraktiven Abbildungen betrachtet. Nun wenden wir uns Abbildungen zu, die auf speziellen Teilmengen von X definiert sind.

Definition 2.6 Sei $\mathcal{H}(X)$ die Klasse aller nicht leeren kompakten Untermengen von X.

Sei A_δ die *δ-Umgebung* von $A \in \mathcal{H}(X)$, $\delta \in \mathbb{R}_0^+$. Diese besteht aus der Menge der Punkte, die innerhalb einer Distanz δ von A liegen, d.h.

$$A_\delta := \{x \in X \mid d(x, a) \leq \delta \text{ für mindestens ein } a \in A\}. \tag{2.3}$$

Definition 2.7 Die Abbildung $d_h : \mathcal{H}(X) \times \mathcal{H}(X) \to \mathbb{R}_0^+$ mit

$$d_h(A, B) := \inf\{\delta \in \mathbb{R}_0^+ \mid A \subseteq B_\delta \text{ und } B \subseteq A_\delta\} \tag{2.4}$$

für $A, B \in \mathcal{H}(X)$ wird *Hausdorffmetrik* genannt.

Satz 2.8 *Ist (X, d) ein vollständiger metrischer Raum, so ist $(\mathcal{H}(X), d_h)$ ebenfalls ein solcher.*

Der nächste Satz zeigt, daß man ein w zu einer Abbildung auf $\mathcal{H}(X)$ erweitern kann, man also statt einzelnen Punkten ganze Mengen transformiert, und daß dabei die Kontraktivität erhalten bleibt.

Satz 2.9 *Ist $w : X \to X$ eine kontraktive Abbildung auf (X, d) mit Kontraktionsfaktor c, dann ist $w : \mathcal{H}(X) \to \mathcal{H}(X)$ mit*

$$w(A) := \{w(a) \mid a \in A\} \qquad A \in \mathcal{H}(X) \tag{2.5}$$

eine kontraktive Abbildung auf $(\mathcal{H}(X), d_h)$ mit Kontraktionsfaktor c.

2.1 Deterministische IFS

Es werden nun Iterierte Funktionensysteme definiert und ihre Eigenschaften festgestellt.

Definition 2.10 Ein *deterministisches Iteriertes Funktionensystem* (kurz deterministisches IFS oder nur IFS) ist eine endliche Menge von zwei oder mehr kontraktiven Abbildungen auf X.

Die im folgenden definierte Abbildung charakterisiert den Effekt eines IFS auf ein Element aus $\mathcal{H}(X)$.

Definition 2.11 Der *Hutchinsonoperator* $\mathbf{w} : \mathcal{H}(X) \to \mathcal{H}(X)$ eines IFS $\{w_i \mid i = 1, \ldots, n\}$ ist die Vereinigung der Bilder unter den n Abbildungen des IFS:

$$\mathbf{w}(A) := \bigcup_{i=1}^{n} w_i(A) \qquad A \in \mathcal{H}(X) \tag{2.6}$$

Folgende beide Sätze sind zentral in der Theorie von IFS.

Satz 2.12 *Der Hutchinsonoperator* $\mathbf{w} : \mathcal{H}(X) \to \mathcal{H}(X)$ *eines IFS ist eine kontraktive Abbildung auf dem vollständigen metrischen Raum* $(\mathcal{H}(X), d_h)$. *Der maximale Kontraktionsfaktor der Abbildungen des IFS ist der Kontraktionsfaktor von* $\mathbf{w}$ *und wird Kontraktionsfaktor des zugehörigen IFS genannt.*

Wendet man den Banachschen Fixpunktsatz (Satz 2.5) auf den Hutchinsonoperator an, ergibt sich folgender Satz:

Satz 2.13 *Der Hutchinsonoperator* $\mathbf{w}$ *hat einen eindeutigen Fixpunkt* $\mathcal{A} \in \mathcal{H}(X)$. $\mathcal{A}$ *ist die eindeutige Lösung von*

$$A = \mathbf{w}(A). \tag{2.7}$$

Der Orbit $\{A_0, \mathbf{w}(A_0), \mathbf{w}^{\circ 2}(A_0), \ldots\}$ *jedes Anfangselements* $A_0 \in \mathcal{H}(X)$ *unter* $\mathbf{w}$ *konvergiert zu* $\mathcal{A}$.

Da der Fixpunkt des Hutchinsonoperators eine so wichtige Rolle, spielt wird ihm ein spezieller Name gegeben.

Definition 2.14 Der Fixpunkt $\mathcal{A}$ des Hutchinsonoperators wird *Attraktor* des zugehörigen IFS genannt.

Um das soeben Vorgestellte zu illustrieren, werden nun einige Beispiele gebracht. Dabei wird der metrische Raum $\mathbb{R}^2$ mit dem euklidischen Abstand verwendet, da dieser leicht darstellbar ist und dem „natürlichen“ Abstandsbegriff entspricht. Die verwendeten Abbildungen sind kontraktive affine Transformationen, die im nächsten Kapitel noch eingehend studiert werden. Kontraktiv bedeutet nun einfach, daß eine Figur in jeder Richtung kleiner wird, wenn sie einer Transformation unterzogen wird. Um affine Transformationen anschaulich beschreiben zu können, wird eine Notation verwendet, die ebenfalls im nächsten Kapitel genau definiert wird. Hier soll vorerst eine informelle Definition genügen: $\mathfrak{S}(c_x, c_y)$ beschreibt eine Skalierung in x–Richtung mit dem Faktor c_x, c_y entsprechend in y–Richtung. $\mathfrak{R}(\alpha)$ beschreibt eine Drehung um den Winkel α.

Der Effekt des Hutchinsonoperators des IFS $\{w_1, w_2, w_3\}$ auf $\mathbb{R}^2$, mit

$$\begin{aligned} w_1(\bar{x}) &= \mathfrak{S}(0.5, 0.5)\bar{x} \\ w_2(\bar{x}) &= \mathfrak{S}(0.5, 0.5)\bar{x} + \binom{0.25}{\sqrt{3/16}} \qquad \bar{x} \in \mathbb{R}^2 \\ w_3(\bar{x}) &= \mathfrak{S}(0.5, 0.5)\bar{x} + \binom{0.5}{0} \end{aligned} \tag{2.8}$$

ist in Abb. 2.1 illustriert. Die Ausgangsmenge ist das graue Rechteck links oben in der Abbil-

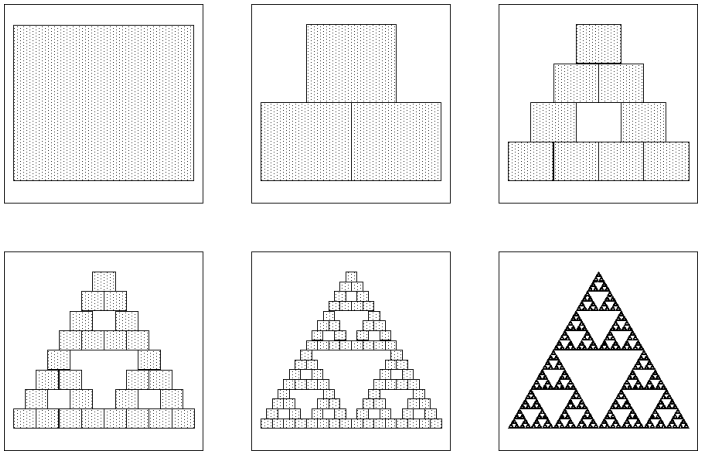


Abbildung 2.1: Wirkung des Hutchinsonoperators auf ein Rechteck

dung. Die einmalige Anwendung des Hutchinsonoperators darauf ergibt die Menge rechts daneben. Wird in der Anwendung des Hutchinsonoperators fortgefahren, ist das Ergebnis eine immer bessere Approximation des Attraktors, in diesem Fall das Sierpinski–Dreieck. Das Sierpinski–Dreieck wird durch den Hutchinsonoperator dieses IFS immer angenähert, egal welche Figur als Ausgangsbild genommen wird, und es ist der einzige Fixpunkt dieses Hutchinsonoperators.

Abb. 2.2 zeigt einige IFS in $\mathbb{R}^2$ und ihre Attraktoren.

2.2 Probabilistische IFS

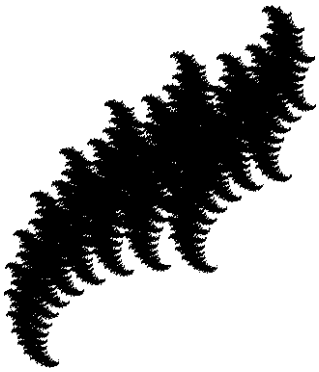
Ein *probabilistisches Iteriertes Funktionensystem* ist ein IFS, das um eine Wahrscheinlichkeit p_i für jede Abbildung w_i erweitert wurde. Die Summe der Wahrscheinlichkeiten ergibt 1.

Sei $\mathcal{B}(X)$ die Borel–σ–Algebra[1] von X [13] und $\mathcal{P}(X)$ der Raum der Wahrscheinlichkeitsmaße[2] auf dem Meßraum $(X, \mathcal{B}(X))$.

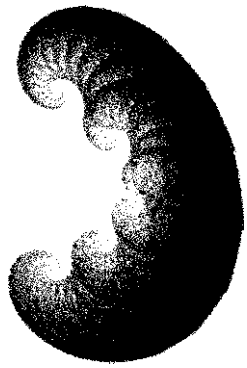
Wenn man $X = \mathbb{R}^2$ setzt, kann man sich ein Wahrscheinlichkeitsmaß $\mu \in \mathcal{P}(\mathbb{R}^2)$ als eine Menge unendlich feinen Sandes mit Gewicht 1 vorstellen, der über der euklidischen Ebene verteilt ist.

[1] Also die vom System der offenen Mengen von X erzeugte σ–Algebra.

[2] Ein Maß μ auf einem Meßraum $(\Omega, \mathfrak{A})$ mit $\mu(\Omega) = 1$ heißt Wahrscheinlichkeitsmaß auf $(\Omega, \mathfrak{A})$.



$$w_1(\bar{x}) = \mathfrak{R}(10^\circ)\mathfrak{S}(0.8, 0.8)\bar{x}$$
$$w_2(\bar{x}) = \mathfrak{R}(20^\circ)\mathfrak{S}(0.64, 0.64)\bar{x} + \binom{5}{3}$$



$$w_1(\bar{x}) = \mathfrak{S}(0.9, 0.8)\mathfrak{R}(-20^\circ)\bar{x}$$
$$w_2(\bar{x}) = \mathfrak{S}(0.8, 0.9)\mathfrak{R}(20^\circ)\bar{x} + \binom{2}{1}$$

$$w_1(\bar{x}) = \mathfrak{S}(0.6, 0.6)\bar{x} + \binom{0.18}{0.36}$$
$$w_2(\bar{x}) = \mathfrak{S}(0.6, 0.6)\bar{x} + \binom{0.18}{0.12}$$
$$w_3(\bar{x}) = \mathfrak{S}(0.5, 0.5)\mathfrak{R}(-53^\circ)\bar{x} + \binom{0.3}{0.45}$$
$$w_4(\bar{x}) = \mathfrak{S}(0.5, 0.5)\mathfrak{R}(53^\circ)\bar{x} + \binom{0.33}{0.09}$$

$$w_1(\bar{x}) = \mathfrak{S}(0.5, 0.5)\bar{x}$$
$$w_2(\bar{x}) = \mathfrak{S}(0.5, 0.5)\bar{x} + \binom{0}{0.5}$$
$$w_3(\bar{x}) = \mathfrak{S}(0.25, 0.8)\mathfrak{R}(45^\circ)\bar{x} + \binom{-0.7}{0}$$
$$w_4(\bar{x}) = \mathfrak{S}(0.25, 0.8)\mathfrak{R}(-45^\circ)\bar{x} + \binom{0.7}{0}$$

Abbildung 2.2: Einige IFS mit zugehörigen Attraktoren

Das Gewicht des Sandes, der auf einer Menge A aus $\mathcal{B}(\mathbb{R}^2)$ lastet, ist dann $\mu(A)$. Aus solchen Maßen können leicht Bilder erzeugt werden [12], denn sie können direkt als Grauwertverteilung interpretiert werden. Über eine Farbtabelle oder durch Überlagerung dreier Maße kann auch ein Farbbild generiert werden. Es wird nun gezeigt, wie ein probabilistisches IFS ein eindeutiges Maß definiert.

Definition 2.15 Jede Abbildung w_i eines probabilistischen IFS erzeugt einen Markovoperator M_i auf $\mathcal{P}(X)$:

$$M_i(\mu)(A) := \mu(w_i^{-1}(A)) \qquad A \in \mathcal{B}(X),\ \mu \in \mathcal{P}(X) \tag{2.9}$$

Der Operator $\mathbf{M} : \mathcal{P}(X) \to \mathcal{P}(X)$ für das gesamte IFS ist dann folgendermaßen definiert:

$$\mathbf{M}(\mu)(A) := \sum_{i=1}^{n} p_i M_i(\mu)(A) \qquad A \in \mathcal{B}(X),\ \mu \in \mathcal{P}(X) \tag{2.10}$$

Der Effekt des Markovoperators $\mathbf{M}$ des IFS

$$\begin{array}{ll} w_1(\bar{x}) = \mathfrak{S}(0.5, 0.5)\bar{x} + \binom{0.5}{0.5} & p_1 = 0.15 \\ w_2(\bar{x}) = \mathfrak{S}(0.5, 0.5)\bar{x} + \binom{0}{0.5} & p_2 = 0.225 \\ w_3(\bar{x}) = \mathfrak{S}(0.5, 0.5)\bar{x} & p_3 = 0.4 \\ w_4(\bar{x}) = \mathfrak{S}(0.5, 0.5)\bar{x} + \binom{0.5}{0} & p_4 = 0.225 \end{array} \qquad \bar{x} \in \mathbb{R}^2 \tag{2.11}$$

auf $\mathbb{R}^2$ soll nun beschrieben werden: Wie man leicht sieht, ist das Einheitsquadrat $[0,1] \times [0,1]$ der Attraktor des IFS, denn es gilt:

$$\begin{aligned} w_1([0,1] \times [0,1]) &= A_1 = [0.5, 1] \times [0.5, 1] \\ w_2([0,1] \times [0,1]) &= A_2 = [0, 0.5] \times [0.5, 1] \\ w_3([0,1] \times [0,1]) &= A_3 = [0, 0.5] \times [0, 0.5] \\ w_4([0,1] \times [0,1]) &= A_4 = [0.5, 1] \times [0, 0.5] \end{aligned} \tag{2.12}$$

und damit

$$\mathbf{w}([0,1] \times [0,1]) = \bigcup_{i=1}^{4} A_i = [0,1] \times [0,1]. \tag{2.13}$$

Es werden nun Grauwertbilder[3] von Maßen gezeigt, deren Träger[4] das Einheitsquadrat ist. Diese Bilder sind so zu interpretieren, daß der Maßwert einer Fläche umso größer ist, je dunkler sie ist. Definiert man μ_0 als das Maß, das jeder Menge A aus $\mathcal{B}(\mathbb{R}^2)$ den Flächeninhalt von $A \cap [0,1] \times [0,1]$ zuordnet, so kann man dieses Maß als ein gleichförmig schwarzes Quadrat darstellen, da

[3] Es wird darauf hingewiesen, daß durch die Reproduktionstechnik bedingt, Grauwertbilder hier als binäre Bilder wiedergegeben werden. Unterschiedliche Grauwerte werden dabei durch unterschiedliche Punktdichten dargestellt.

[4] Der Träger eines Maßes μ auf $(\Omega, \mathfrak{A})$ ist $\Omega \setminus \bigcup\{A \in \mathfrak{A} \mid \mu(A) = 0 \text{ und } A \text{ ist offen}\}$.

gleich große Flächen den gleichen Maßwert haben und damit μ_0 sozusagen homogene Dichte auf $[0,1]\times[0,1]$ besitzt. In Abb. 2.3 sieht man links oben das Maß $\mu_1 = \mathbf{M}(\mu_0)$. Es ergeben sich vier Teilbereiche mit konstanten Grauwerten, die zueinander im Verhältnis der Wahrscheinlichkeiten jener Transformationen stehen, die den entsprechenden Teil des Attraktors erzeugen: Wendet man die w_j auf das Einheitsquadrat an, so erhält man die A_i aus (2.12). Die Maßwerte dieser vier Mengen sind

$$\mu_1(A_i) = \sum_{j=1}^{4} p_j \mu_0(w_j^{-1}(A_i)) \qquad \text{für } i = 1,2,3,4. \tag{2.14}$$

Da aber $\mu_0(w_j^{-1}(A_i)) = 0$ für $i \neq j$ ist, und $\mu_0(w_i^{-1}(A_i)) = \mu_0([0,1]\times[0,1]) = 1$ für $i = 1,2,3,4$ gilt, erhält man aus der obigen Beziehung

$$\mu_1(A_i) = p_i \qquad \text{für } i = 1,2,3,4. \tag{2.15}$$

Rechts oben in Abb. 2.3 sieht man das Bild des Maßes $\mu_2 = \mathbf{M}(\mu_1)$. Da man nun schon jeweils zwei Transformationen auf μ_0 angewendet hat (nämlich $w_i^{-1} \circ w_j^{-1}$ für $i,j \in \{1,2,3,4\}$), ergeben sich nun 16 Teilbereiche konstanter Grauwerte, die im Verhältnis des Produktes der Wahrscheinlichkeiten jener beiden Transformationen stehen, die den entsprechenden Bereich erzeugen. Links unten ist μ_3, rechts unten μ_6 abgebildet. So wie die wiederholte Anwendung des Hutchinsonoperators $\mathbf{w}$ auf ein Bild immer bessere Approximationen des Attraktors liefert, so strebt die Folge $\{\mu_i\}$ gegen ein Maß μ_∞, das von $\mathbf{M}$ nicht mehr verändert wird. Dieses *invariante Maß* ist nur von den Transformationen und Wahrscheinlichkeiten des IFS abhängig.

Definition 2.16 Die Abbildung $d_H : \mathcal{P}(X) \times \mathcal{P}(X) \to \mathbb{R}_0^+$ mit

$$d_H(\mu,\nu) := \sup_{\varphi \in C} \left\{ \int_X \varphi \, d\mu - \int_X \varphi \, d\nu \right\} \tag{2.16}$$

für $\mu, \nu \in \mathcal{P}(X)$, wobei

$$C := \{\varphi : X \to \mathbb{R} \mid \forall\, x,y \in X : |\varphi(x) - \varphi(y)| \leq d(x,y)\} \tag{2.17}$$

wird *Hutchinsonmetrik* genannt.

Satz 2.17 *Ist (X,d) ein vollständiger metrischer Raum, so ist $(\mathcal{P}(X), d_H)$ ebenfalls ein solcher.*

Satz 2.18 *Der Markovoperator $\mathbf{M} : \mathcal{P}(X) \to \mathcal{P}(X)$ eines probabilistischen IFS ist eine kontraktive Abbildung auf dem vollständigen metrischen Raum $(\mathcal{P}(X), d_H)$. $\mathbf{M}$ besitzt einen eindeutigen Fixpunkt $\mu_F \in \mathcal{P}(X)$, der die einzige Lösung der Gleichung*

$$\mu(A) = \mathbf{M}(\mu)(A) \qquad \forall\, A \in \mathcal{B}(X) \tag{2.18}$$

ist und dessen Träger der Attraktor des IFS ist. μ_F wird invariantes Maß *genannt.*

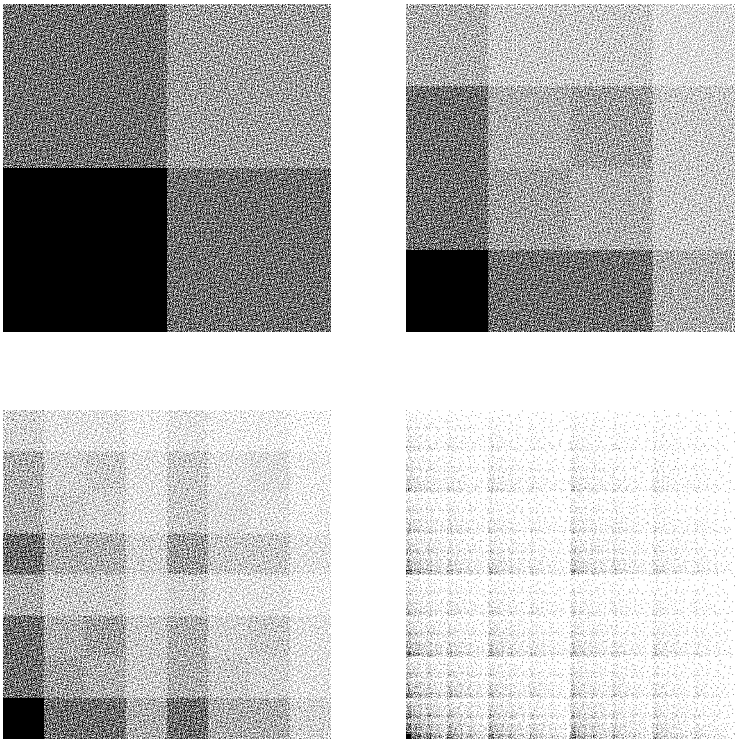


Abbildung 2.3: Wirkung des Markovoperators auf ein Maß

Kapitel 3

Affine Transformationen

IFS, die in $X = \mathbb{R}^d$, $d \in \mathbb{N}^+$ mit dem euklidischen Abstand und affinen Transformationen definiert sind[1], sind von besonderem Interesse, da der euklidische Abstand der „natürliche“ Abstand ist, affine Transformationen geometrische Transformationen sind, deren Wirkungsweise unmittelbar anschaulich ist, und Attraktoren solcher IFS als Bilder dargestellt werden können.

Die Beziehung zwischen einem Bild und einer Teilmenge von $\mathbb{R}^d$ ist wie folgt: Ein (reales) Grauwertbild ist nach [46] eine Funktion $f : \mathbb{R}^2 \to \mathbb{R}_0^+$ deren Träger eine beschränkte Menge ist. Ein Binärbild ist dann analog eine Funktion $b : \mathbb{R}^2 \to \{0, 1\}$ mit beschränktem Träger. Hier wird ein Binärbild b mit dem Abschluß des Trägers von b identifiziert, also der Menge aller Punkte von $\mathbb{R}^2$ die unter b auf 1 abgebildet werden. Man verwendet den Abschluß des Trägers, um Kompaktheit zu erlangen, da dies keine praktische Einschränkung darstellt, und in Bezug auf die verwendete Mathematik von Vorteil ist. Umgekehrt definiert jede kompakte Teilmenge von $\mathbb{R}^2$ damit auch ein Binärbild. Ebenso kann man mit Grauwertbildern verfahren, nur ist es dann notwendig, auch den Grauwert zu codieren; man identifiziert das Grauwertbild f mit dem Abschluß der Menge $\{(x, y, g) \mid f(x, y) = g \text{ mit } x, y \in \mathbb{R} \text{ und } g \in \mathbb{R}^+\}$, sozusagen den Punkten, die eine Helligkeit größer 0 haben. Farbbilder können analog als kompakte Teilmengen von $\mathbb{R}^5$ (zwei Ortskoordinaten und drei Farbwerte) betrachtet werden. Damit die Attraktoren in diesen Fällen für jede Ortskoordinate einen eindeutigen Wert besitzen, dürfen nach [39] die Transformationen nur Schärungen sein, also den Raum in Richtung der Koordinaten der Bildwerte nicht drehen.

Es werden zunächst einige Eigenschaften allgemeiner affiner Transformationen auf $\mathbb{R}^d$ untersucht, dann werden nur mehr affine Transformationen auf $\mathbb{R}^2$ betrachtet.

3.1 Notation, grundlegende Definitionen und Begriffe

Zunächst zur verwendeten Notation:

[1] d bezeichnet hier die Dimension des verwendeten euklidischen Vektorraums, und sollte nicht mit der Metrik $d(x, y)$ verwechselt werden.

- $\mathbb{R}^{d\times d}$ ist die Menge der $d \times d$ Matrizen über $\mathbb{R}$, $d \geq 1$.
- $L, M, S, \ldots \in \mathbb{R}^{d\times d}$ sind $d \times d$ Matrizen über $\mathbb{R}$.
- $I_d \in \mathbb{R}^{d\times d}$ ist die $d \times d$ Einheitsmatrix.
- $\bar{x}, \bar{y}, \bar{z}, \bar{t}, \ldots \in \mathbb{R}^d$ sind Vektoren (Punkte) im d–dimensionalen euklidischen Raum. $\bar{x}_i$ mit $i = 1, \ldots, d$ sind die Komponenten des Vektors $\bar{x}$.
- $\bar{0} \in \mathbb{R}^d$ ist der Nullvektor.

Nun einige gut bekannte Definitionen aus der linearen Algebra, die aber hier bereits auf $\mathbb{R}^d$ als zugrunde liegendem Vektorraum zugeschnitten sind. Für die allgemeinen Definitionen sei auf [40] verwiesen.

Definition 3.1 Ein *linearer Operator* $\varphi : \mathbb{R}^d \to \mathbb{R}^d$ ist eine Abbildung, die die folgenden Eigenschaften hat:

$$\varphi(\bar{x} + \bar{y}) = \varphi(\bar{x}) + \varphi(\bar{y}) \qquad \forall\, \bar{x}, \bar{y} \in \mathbb{R}^d \tag{3.1}$$

$$\varphi(c\bar{x}) = c\varphi(\bar{x}) \qquad \forall\, \bar{x} \in \mathbb{R}^d,\ \forall\, c \in \mathbb{R} \tag{3.2}$$

Die linearen Operatoren auf $\mathbb{R}^d$ sind zu den $d \times d$ Matrizen über $\mathbb{R}$ isomorph [40], dabei gibt es für jeden linearen Operator φ genau eine Matrix $L \in \mathbb{R}^{d\times d}$, sodaß

$$\varphi(\bar{x}) = L\bar{x} \qquad \forall\, \bar{x} \in \mathbb{R}^d. \tag{3.3}$$

Daher werden die Begriffe $d \times d$ Matrix über $\mathbb{R}$ (kurz Matrix) und linearer Operator auf $\mathbb{R}^d$ synonym verwendet.

Definition 3.2 Eine *affine Transformation* $w : \mathbb{R}^d \to \mathbb{R}^d$ ist eine Abbildung, die jedem Punkt $\bar{x} \in \mathbb{R}^d$ den Punkt

$$w(\bar{x}) := L\bar{x} + \bar{t} \tag{3.4}$$

zuordnet, wobei $L \in \mathbb{R}^{d\times d}$ ein linearer Operator auf $\mathbb{R}^d$ ist. Eine affine Transformation ist also durch einen linearen Operator und einen Punkt $\bar{t} \in \mathbb{R}^d$, die *translative Komponente* von w, festgelegt.

Definition 3.3 Als *Inneres Produkt* $\langle\cdot,\cdot\rangle$ zweier Vektoren aus $\mathbb{R}^d$ bezeichnet man die Abbildung:

$$\begin{aligned} \langle\cdot,\cdot\rangle \ :\ \mathbb{R}^d \times \mathbb{R}^d &\to \mathbb{R} \\ (\bar{x}, \bar{y}) &\mapsto \langle\bar{x}, \bar{y}\rangle := \bar{x}^T\bar{y} = \sum_{i=1}^{d} \bar{x}_i\bar{y}_i \end{aligned}$$

Definition 3.4 Die *Norm* (Länge) $\|\cdot\|$ eines Vektors aus $\mathbb{R}^d$ ist folgende Abbildung:

$$\begin{aligned} \|\cdot\| \ :\ \mathbb{R}^d &\to \mathbb{R}_0^+ \\ \bar{x} &\mapsto \|\bar{x}\| := +\sqrt{\langle\bar{x}, \bar{x}\rangle} \end{aligned}$$

Definition 3.5 Der *Euklidische Abstand* (Metrik) d in $\mathbb{R}^d$ ist folgende Abbildung:

$$\begin{aligned} d \;:\; \mathbb{R}^d \times \mathbb{R}^d &\to \mathbb{R}_0^+ \\ (\bar{x}, \bar{y}) &\mapsto d(\bar{x}, \bar{y}) := \|\bar{x} - \bar{y}\| \end{aligned}$$

Definition 3.6 Zwei Vektoren aus $\mathbb{R}^d$ heißen genau dann *orthogonal*, wenn ihr inneres Produkt gleich 0 ist.

$$\bar{x} \perp \bar{y} \;:\Longleftrightarrow\; \langle \bar{x}, \bar{y} \rangle = 0 \qquad \forall\, \bar{x}, \bar{y} \in \mathbb{R}^d \tag{3.5}$$

3.2 Affine Transformationen auf $\mathbb{R}^d$

3.2.1 Kontraktionsfaktor einer affinen Transformationen

Hier werden zwei Sätze bewiesen, die allgemeine Aussagen über den Kontraktionsfaktor einer affinen Transformation auf $\mathbb{R}^d$ machen.

Satz 3.7 *Sei $w(\bar{x}) = L\bar{x} + \bar{t}$ eine affine Transformation auf $\mathbb{R}^d$. Der Kontraktionsfaktor von L ist der Kontraktionsfaktor von w.*

Beweis: Aus der Identität

$$d(w(\bar{x}), w(\bar{y})) = \|w(\bar{x}) - w(\bar{y})\| \tag{3.6}$$

$$= \|L\bar{x} - L\bar{y}\| \tag{3.7}$$

$$= d(L\bar{x}, L\bar{y}) \tag{3.8}$$

die für alle $\bar{x}, \bar{y} \in \mathbb{R}^d$ gilt, folgt der Satz sofort mit Def. 2.1. □

Satz 3.8 *Sei $w(\bar{x}) = L\bar{x} + \bar{t}$ eine affine Transformation auf $\mathbb{R}^d$. $c \in \mathbb{R}_0^+$ ist der Kontraktionsfaktor von w genau dann, wenn*

$$\|L\bar{z}\| \leq c\|\bar{z}\| \qquad \forall\, \bar{z} \in \mathbb{R}^d \tag{3.9}$$

gilt und es kein $k \in [0, c)$ mit

$$\|L\bar{z}\| \leq k\|\bar{z}\| \qquad \forall\, \bar{z} \in \mathbb{R}^d \tag{3.10}$$

gibt.

Beweis: Sei $r \in \mathbb{R}_0^+$ beliebig, dann ist

$$d(w(\bar{x}), w(\bar{y})) \leq r d(\bar{x}, \bar{y}) \qquad \forall\, \bar{x}, \bar{y} \in \mathbb{R}^d \tag{3.11}$$

wahr, genau dann, wenn

$$\|w(\bar{x}) - w(\bar{y})\| \leq r\|\bar{x} - \bar{y}\| \qquad \forall\, \bar{x}, \bar{y} \in \mathbb{R}^d \tag{3.12}$$

$$\iff \qquad \|L\bar{x} - L\bar{y}\| \leq r\|\bar{x} - \bar{y}\| \qquad \forall\, \bar{x}, \bar{y} \in \mathbb{R}^d \tag{3.13}$$

$$\iff \qquad \|L(\bar{x} - \bar{y})\| \leq r\|\bar{x} - \bar{y}\| \qquad \forall\, \bar{x}, \bar{y} \in \mathbb{R}^d \tag{3.14}$$

$$\iff \qquad \|L\bar{z}\| \leq r\|\bar{z}\| \qquad \forall\, \bar{z} \in \mathbb{R}^d \tag{3.15}$$

gilt. Zusammengefaßt ist also für ein beliebiges $r \in \mathbb{R}_0^+$ wahr, daß

$$d(w(\bar{x}), w(\bar{y})) \leq r d(\bar{x}, \bar{y}) \qquad \forall\, \bar{x}, \bar{y} \in \mathbb{R}^d \tag{3.16}$$

genau dann gilt, wenn

$$\|L\bar{z}\| \leq r\|\bar{z}\| \qquad \forall\, \bar{z} \in \mathbb{R}^d \tag{3.17}$$

gilt. Woraus sich, wie man leicht sieht, mit Def. 2.1 der Satz ergibt. □

3.2.2 Zwei Darstellungen kontraktiver affiner Transformationen

Jede kontraktive affine Transformation

$$w(\bar{x}) = L\bar{x} + \bar{t} \tag{3.18}$$

auf $\mathbb{R}^d$ besitzt einen eindeutigen Fixpunkt $\bar{x}_F = w(\bar{x}_F)$ (Satz 2.5). w kann unter Verwendung dieses Fixpunktes als

$$w(\bar{x}) = L(\bar{x} - \bar{x}_F) + \bar{x}_F \tag{3.19}$$

dargestellt werden, da diese beiden Darstellungen eindeutig in einander überführbar sind, wie die Sätze 3.11 und 3.12 beweisen werden.

Aus der Darstellung in Fixpunktform ist ersichtlich, daß L, geometrisch interpretiert, in Bezug auf den Fixpunkt operiert. Das heißt, daß bei einer Drehung $\mathbb{R}^d$ um den Fixpunkt gedreht wird, daß bei einer Kontraktion $\mathbb{R}^d$ zum Fixpunkt hin strebt, etc.

Hilfssatz 3.9 *Sei L ein kontraktiver linearer Operator auf $\mathbb{R}^d$; dann ist $(I_d - L)$ eine invertierbare Matrix.*

Beweis: Halten wir folgendes fest: Wenn 1 ein Eigenwert von L ist, dann sind die Eigenvektoren zum Eigenwert 1 genau die Fixpunkte von L. Denn es gilt $L\bar{x} = 1\bar{x}$ für alle diese Eigenvektoren, dies ist aber zugleich die Fixpunktgleichung für L.

L ist kontraktiv und hat damit nach dem Banachschen Fixpunktsatz (Satz 2.5) genau einen Fixpunkt. Jeder lineare Operator bildet den Nullvektor $\bar{0}$ auf sich selbst ab, deshalb hat jeder kontraktive lineare Operator $\bar{0}$ als einzigen Fixpunkt. Damit folgt aus der obigen Beobachtung,

daß 1 kein Eigenwert von L sein kann, da es sonst von $\bar{0}$ verschiedene Fixpunkte von L geben müßte.[2]

Nach Satz 8.2 [40] gilt: $\lambda \in \mathbb{R}$ ist Eigenwert von $L \iff \det(\lambda I_d - L) = 0$. Wie oben gezeigt wurde, ist 1 kein Eigenwert von L, daher ist $\det(I_d - L) \neq 0$, und damit ist $(I_d - L)$ invertierbar. □

Satz 3.10 *Sei $w(\bar{x}) = L\bar{x} + \bar{t}$ eine kontraktive affine Transformation auf $\mathbb{R}^d$. $\bar{x}_F = (I_d - L)^{-1}\bar{t}$ existiert und ist der einzige Fixpunkt von w.*

Beweis: Nach dem Banachschen Fixpunktsatz (Satz 2.5) existiert ein eindeutiger Fixpunkt $\bar{x}_F$ von w. Berechnung von $\bar{x}_F$:

$$\bar{x}_F = w(\bar{x}_F) \tag{3.20}$$

$$\bar{x}_F = L\bar{x}_F + \bar{t} \tag{3.21}$$

$$\bar{x}_F - L\bar{x}_F = \bar{t} \tag{3.22}$$

$$(I_d - L)\bar{x}_F = \bar{t} \tag{3.23}$$

$$\bar{x}_F = (I_d - L)^{-1}\bar{t} \tag{3.24}$$

Die Multiplikation von links mit $(I_d - L)^{-1}$ ist möglich, weil nach Hilfssatz 3.9 $(I_d - L)$ bei kontraktivem L invertierbar ist. Nach Satz 3.7 ist L kontraktiv, weil w kontraktiv ist. □

Satz 3.11 *Sei $w(\bar{x}) = L(\bar{x} - \bar{x}_F) + \bar{x}_F$ eine kontraktive affine Transformation auf $\mathbb{R}^d$. Es gibt ein eindeutiges $\bar{t} \in \mathbb{R}^d$, sodaß $w(\bar{x}) = L\bar{x} + \bar{t}$ für alle $\bar{x} \in \mathbb{R}^d$ gilt, und zwar $\bar{t} = (I_d - L)\bar{x}_F$.*

Beweis: Die beiden Darstellungen werden gleichgesetzt, und daraus wird $\bar{t}$ berechnet:

$$L\bar{x} + \bar{t} = L(\bar{x} - \bar{x}_F) + \bar{x}_F \tag{3.25}$$

$$L\bar{x} + \bar{t} = L\bar{x} - L\bar{x}_F + I_d\bar{x}_F \tag{3.26}$$

$$\bar{t} = (I_d - L)\bar{x}_F \tag{3.27}$$

□

Satz 3.12 *Sei $w(\bar{x}) = L\bar{x} + \bar{t}$ eine kontraktive affine Transformation auf $\mathbb{R}^d$, $\bar{x}_F = (I_d - L)^{-1}\bar{t}$ ihr einziger Fixpunkt. Es gilt $w(\bar{x}) = L(\bar{x} - \bar{x}_F) + \bar{x}_F$ für alle $\bar{x} \in \mathbb{R}^d$.*

Beweis: Nach Satz 3.10 existiert $\bar{x}_F = (I_d - L)^{-1}\bar{t}$ und ist der einzige Fixpunkt von w. Damit gilt

$$L(\bar{x} - \bar{x}_F) + \bar{x}_F = L(\bar{x} - (I_d - L)^{-1}\bar{t}) + (I_d - L)^{-1}\bar{t} \tag{3.28}$$

$$= L\bar{x} - L(I_d - L)^{-1}\bar{t} + I_d(I_d - L)^{-1}\bar{t} \tag{3.29}$$

$$= L\bar{x} + (I_d - L)(I_d - L)^{-1}\bar{t} \tag{3.30}$$

$$= L\bar{x} + \bar{t} \tag{3.31}$$

$$= w(\bar{x}) \tag{3.32}$$

für alle $\bar{x} \in \mathbb{R}^d$. □

[2] Eigenvektoren sind ja per Definition $\neq \bar{0}$, und wäre 1 ein Eigenwert, hätte er Eigenvektoren.

3.3 Affine Transformationen auf $\mathbb{R}^2$

Dem Fall affiner Transformationen auf $\mathbb{R}^2$ soll nun besondere Aufmerksamkeit geschenkt werden, da im weiteren nur IFS in $\mathbb{R}^2$ behandelt werden. IFS in $\mathbb{R}^2$ zeigen jedoch bereits ein so komplexes Verhalten, daß Grund zur Hoffnung besteht, durch ihr Studium zu einem besseren Verständnis von IFS in höherdimensionalen Räumen zu gelangen.

3.3.1 Geometrische Transformationen und zugehörige Matrizen

Will man verstehen, welche geometrischen Operationen eine affine Transformation auf $\mathbb{R}^2$ durchführt, so ist die Darstellung über eine 2×2 Matrix und eine translative Komponente (bzw. Fixpunkt) nicht sehr hilfreich. Umgekehrt ist es z.B. nicht offensichtlich, welche Werte die Matrix zu einer Transformation haben muß, die eine $30°$ Drehung und eine gleichförmige Skalierung mit dem Wert 0.5 durchführen soll.

Hier werden zunächst zwei Operatoren definiert, die Matrizen generieren, welche bestimmte geometrische Transformationen durchführen.

Definition 3.13 Die Operatoren $\mathfrak{R} : \mathbb{R} \to \mathbb{R}^{2\times 2}$ und $\mathfrak{S} : \mathbb{R} \times \mathbb{R} \to \mathbb{R}^{2\times 2}$ sind wie folgt definiert:

$$\mathfrak{R}(\alpha) := \begin{pmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{pmatrix} \qquad \alpha \in [0, 2\pi) \tag{3.33}$$

$$\mathfrak{S}(c_x, c_y) := \begin{pmatrix} c_x & 0 \\ 0 & c_y \end{pmatrix} \qquad c_x, c_y \in \mathbb{R} \tag{3.34}$$

Wie man leicht sieht, haben diese Operatoren folgende Eigenschaften:

- $\mathfrak{R}(\alpha)$ führt eine Drehung um den Ursprung um einen Winkel α durch.

$(\mathfrak{R}(\alpha))^{-1} = \mathfrak{R}(-\alpha)$	Drehung und Rückdrehung um denselben Winkel heben sich auf.
$\mathfrak{R}(\alpha)\mathfrak{R}(\beta) = \mathfrak{R}(\alpha + \beta)$	Je eine Drehung um α und β ergibt dasselbe wie eine Drehung um $\alpha + \beta$.

- $\mathfrak{S}(c_x, c_y)$ führt eine Skalierung mit c_x in x–, und mit c_y in y–Richtung durch.

$c_x, c_y \in (-1, 1)$	Skalierung ist kontraktiv
$c_x < 0$	Spiegelung um die y–Achse
$c_y < 0$	Spiegelung um die x–Achse

Man kann nun diese beiden Operatoren kombinieren (also die Matrizen multiplizieren), um vielfältigere Transformationen zu erhalten:

- $\mathfrak{R}(\alpha)\mathfrak{S}(c_x, c_y)$ zuerst Skalierung, dann Drehung
- $\mathfrak{S}(c_x, c_y)\mathfrak{R}(\alpha)$ zuerst Drehung, dann Skalierung
- $\mathfrak{R}(\beta)\mathfrak{S}(c_x, c_y)\mathfrak{R}(\alpha)$ Wie später in Satz 3.16 gezeigt wird, kann so *jede* Matrix aus $\mathbb{R}^{2\times 2}$ dargestellt werden. Weiters ist diese Darstellung von Vorteil, da die Kontraktivität der Transformation unmittelbar aus den Werten c_x und c_y ersichtlich ist, was Satz 3.18 beweisen wird.

In Abb. 3.1 – 3.4 wird die Wirkung einiger affiner Transformationen mit Bildern ihrer Orbits illustriert. Der Punkt $\binom{1}{1}$ ist der Startpunkt für jeden Orbit.

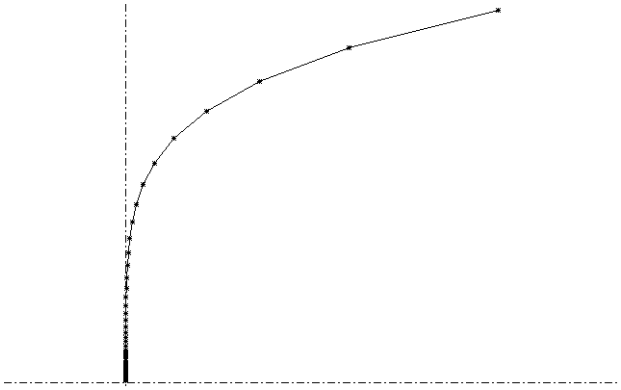


Abbildung 3.1: Orbit der Transformation $w(\bar{x}) = \mathfrak{S}(0.6, 0.9)\bar{x}$

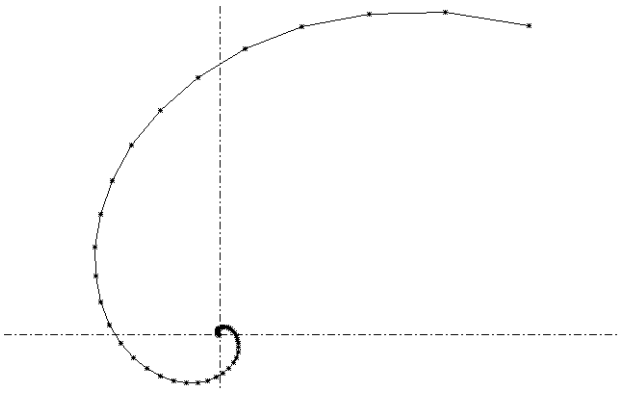


Abbildung 3.2: Orbit der Transformation $w(\bar{x}) = \mathfrak{R}(10°)\mathfrak{S}(0.9, 0.9)\bar{x}$

3.3.2 Zerlegung eines linearen Operators

In diesem Abschnitt wird gezeigt, daß jeder lineare Operator auf $\mathbb{R}^2$ als das Produkt einer Drehung, einer Skalierung/Spiegelung und noch einer Drehung darstellbar ist. Die Beweise sind

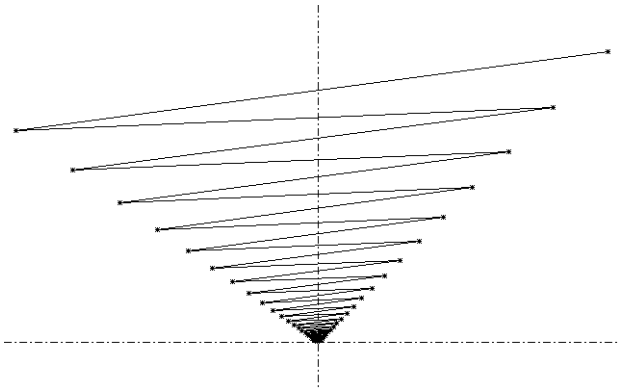


Abbildung 3.3: Orbit der Transformation $w(\bar{x}) = \mathfrak{R}(10^\circ)\mathfrak{S}(-0.9, 0.9)\bar{x}$

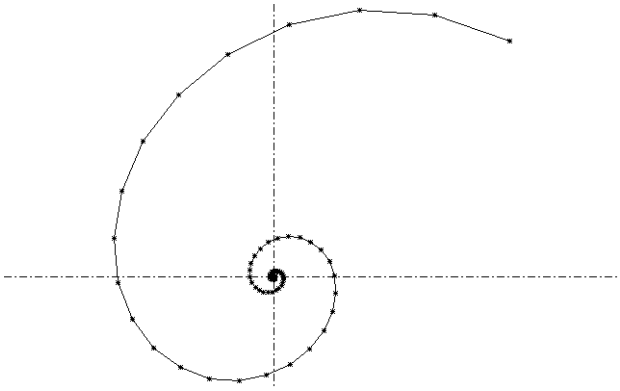


Abbildung 3.4: Orbit der Transformation $w(\bar{x}) = \mathfrak{R}(10^\circ)\mathfrak{S}(0.95, 0.9)\mathfrak{R}(5^\circ)\bar{x}$

konstruktiv in dem Sinn, daß sie angeben, wie die Parameter für die geometrischen Transformationen berechnet werden müssen, um einen linearen Operator so darzustellen. Am Ende des Abschnittes folgt eine Zusammenfassung dieser Berechnungen.

Hilfssatz 3.14 *Jeder lineare Operator L auf $\mathbb{R}^2$ kann als das Produkt einer symmetrischen Matrix und einer Drehungsmatrix dargestellt werden.*

$$\forall\, L = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \in \mathbb{R}^{2\times 2}\ \exists\, \alpha \in [0, 2\pi)\ \exists\, A = \begin{pmatrix} r & s \\ s & t \end{pmatrix} \in \mathbb{R}^{2\times 2} : L = A\mathfrak{R}(\alpha) \tag{3.35}$$

Beweis: Zunächst wird ein Gleichungssystem für r, s, t und α abgeleitet.

$$L = A\mathfrak{R}(\alpha) \qquad | \cdot \mathfrak{R}(-\alpha) \tag{3.36}$$

Da eine Drehung und eine Rückdrehung um denselben Winkel sich aufheben (siehe Abschnitt 3.3.1), ergibt dies:

$$L\mathfrak{R}(-\alpha) = A \tag{3.37}$$

$$\begin{pmatrix} a\cos\alpha - b\sin\alpha & a\sin\alpha + b\cos\alpha \\ c\cos\alpha - d\sin\alpha & c\sin\alpha + d\cos\alpha \end{pmatrix} = \begin{pmatrix} r & s \\ s & t \end{pmatrix} \tag{3.38}$$

also

$$a\cos\alpha - b\sin\alpha = r \tag{3.39}$$

$$a\sin\alpha + b\cos\alpha = s \tag{3.40}$$

$$c\cos\alpha - d\sin\alpha = s \tag{3.41}$$

$$c\sin\alpha + d\cos\alpha = t. \tag{3.42}$$

Aus (3.40) und (3.41) ergibt sich:

$$a\sin\alpha + b\cos\alpha = c\cos\alpha - d\sin\alpha \tag{3.43}$$

$$(a+d)\sin\alpha = (c-b)\cos\alpha \tag{3.44}$$

Fall 1: $a + d = 0$ und $c - b = 0$ (also $a = -d$ und $b = c$)

(3.44) ist für alle α erfüllt. Es wird $\alpha = 0$ gewählt, dann gilt $\mathfrak{R}(\alpha) = \left(\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right)$. Da L die Gestalt $\left(\begin{smallmatrix} a & b \\ b & -a \end{smallmatrix}\right)$ hat, ist der Satz mit $A = \left(\begin{smallmatrix} a & b \\ b & -a \end{smallmatrix}\right)$ für diesen Fall bewiesen.

Fall 2: sonst

Sei α so gewählt, daß

$$\sin\alpha = \frac{c-b}{\sqrt{(c-b)^2 + (a+d)^2}} \tag{3.45}$$

und

$$\cos\alpha = \frac{a+d}{\sqrt{(c-b)^2 + (a+d)^2}} \tag{3.46}$$

gilt. Dann ist (3.44) erfüllt, wie man durch Einsetzen leicht sieht. Die Existenz eines solchen α folgt aus

$$\left(\frac{c-b}{\sqrt{(c-b)^2+(a+d)^2}}\right)^2+\left(\frac{a+d}{\sqrt{(c-b)^2+(a+d)^2}}\right)^2=1. \tag{3.47}$$

Durch Einsetzen von (3.45) und (3.46) in die Gleichungen (3.39), (3.40) und (3.42) werden r, s und t berechnet:

$$r=\frac{a^2+b^2+ad-bc}{\sqrt{(c-b)^2+(a+d)^2}} \tag{3.48}$$

$$s=\frac{ac-bd}{\sqrt{(c-b)^2+(a+d)^2}} \tag{3.49}$$

$$t=\frac{c^2+d^2+ad-bc}{\sqrt{(c-b)^2+(a+d)^2}} \tag{3.50}$$

Es wurde also gezeigt, daß im Fall von $a+d\neq 0$ oder $c-b\neq 0$ Werte für α, r, s und t berechnet werden können, sodaß $L=A\mathfrak{R}(\alpha)$ ist.

Fall 1 und Fall 2 überdecken alle möglichen Werte für a,b,c und d und damit alle $L\in\mathbb{R}^{2\times 2}$. Der Satz ist somit bewiesen. □

Hilfssatz 3.15 *Jede symmetrische Matrix $A\in\mathbb{R}^{2\times 2}$ kann als das Produkt einer Hindrehung, einer Skalierung/Spiegelung und einer Rückdrehung dargestellt werden.*

$$\forall\, A=\begin{pmatrix} r & s\\ s & t\end{pmatrix}\in\mathbb{R}^{2\times 2}\ \exists\, c_1,c_2\in\mathbb{R}\ \exists\,\beta\in[0,2\pi):A=\mathfrak{R}(-\beta)\mathfrak{S}(c_1,c_2)\mathfrak{R}(\beta) \tag{3.51}$$

Beweis: Nach Satz 8.12 aus [40] gibt es zu jeder symmetrischen Matrix A über $\mathbb{R}$ eine orthogonale Matrix S, sodaß $D=SAS^{-1}$ eine reelle Diagonalmatrix ist. Das heißt $D=\mathfrak{S}(c_1,c_2)$ mit $c_1,c_2\in\mathbb{R}$ und $S=\mathfrak{R}(\beta)$ oder $S=\left(\begin{smallmatrix}1&0\\0&-1\end{smallmatrix}\right)\mathfrak{R}(\beta)$, da orthogonale Matrizen genau die Matrizen sind, deren Zeilen– oder Spaltenvektoren ein Orthonormalsystem bilden. Man kann sich leicht davon überzeugen, daß das in $\mathbb{R}^2$ genau die Drehspiegelungsmatrizen sind, die eben alle als $\mathrm{R}(\beta)$ oder $\left(\begin{smallmatrix}1&0\\0&-1\end{smallmatrix}\right)\mathfrak{R}(\beta)$ darstellbar sind. c_1, c_2 sind nach der Bemerkung zu Satz 8.12 [40] die Eigenwerte zu A, die sicher existieren und reell sind, da A symmetrisch und damit normal ist. Die Zeilenvektoren von S sind normalisierte Eigenvektoren zu den Eigenwerten c_1, c_2. Diese Eigenvektoren sind nach Satz 8.9 Korollar 2 [40] bereits orthogonal.

Es werden nun die Eigenwerte c_1, c_2 von A berechnet, um $D=\mathfrak{S}(c_1,c_2)$ zu erhalten. Dann wird gezeigt, daß es einen Winkel β gibt, sodaß die Zeilenvektoren von $\mathfrak{R}(\beta)$ Eigenvektoren der Länge 1 zu den Eigenwerten c_1, c_2 sind.

Eigenwerte c_1, c_2 von $A=\left(\begin{smallmatrix}r&s\\s&t\end{smallmatrix}\right)$: Die Eigenwerte λ einer beliebigen Matrix $M\in\mathbb{R}^{d\times d}$ sind nach Satz 8.2 [40] die Lösungen ihrer charakteristischen Gleichung:

$$\det(M-\lambda I_d)=0 \tag{3.52}$$

Die Eigenwerte λ zu A sind also die Lösungen von

$$\det(A - \lambda I_d) = 0 \tag{3.53}$$

$$\det \begin{pmatrix} r-\lambda & s \\ s & t-\lambda \end{pmatrix} = 0 \tag{3.54}$$

$$(r-\lambda)(t-\lambda) - s^2 = 0 \tag{3.55}$$

$$\lambda^2 - (r+t)\lambda + rt - s^2 = 0. \tag{3.56}$$

Die Lösung dieser quadratischen Gleichung liefert:

$$c_1 = \lambda_1 = \frac{r+t+\sqrt{(r-t)^2+4s^2}}{2} \tag{3.57}$$

$$c_2 = \lambda_2 = \frac{r+t-\sqrt{(r-t)^2+4s^2}}{2} \tag{3.58}$$

Bestimmung von $S = \mathfrak{R}(\beta)$:

Fall 1: $s = 0$

Es gilt $A = \left(\begin{smallmatrix} r & 0 \\ 0 & t \end{smallmatrix}\right)$. Aus (3.57) und (3.58) folgt:

$$c_1 = \frac{r+t+\sqrt{(r-t)^2}}{2} \qquad c_2 = \frac{r+t-\sqrt{(r-t)^2}}{2} \tag{3.59}$$

$$c_1 = \frac{r+t+|r-t|}{2} \qquad c_2 = \frac{r+t-|r-t|}{2} \tag{3.60}$$

Fall 1.1: $r < t$

$$r < t \implies |r-t| = t-r \implies c_1 = t,\ c_2 = r \tag{3.61}$$

$$D = \mathfrak{S}(t,r) = \begin{pmatrix} t & 0 \\ 0 & r \end{pmatrix} \tag{3.62}$$

Eigenwert von A	Eigenvektor der Länge 1
t	$\left(\begin{smallmatrix} 0 \\ 1 \end{smallmatrix}\right)$
r	$\left(\begin{smallmatrix} -1 \\ 0 \end{smallmatrix}\right)$

Daß die angegebenen Vektoren $\bar{x}$ Eigenvektoren zu den entsprechenden Eigenwerten λ sind, kann man leicht durch Einsetzen in $A\bar{x} = \lambda\bar{x}$ überprüfen. Damit ergibt sich S zu

$$S = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} = \mathfrak{R}(\frac{3\pi}{2}) \tag{3.63}$$

$$\beta = \frac{3\pi}{2} \tag{3.64}$$

Fall 1.2: $r \geq t$

$$r \geq t \implies |r-t| = r-t \implies c_1 = r,\, c_2 = t \tag{3.65}$$

$$D = \mathfrak{S}(r,t) = \begin{pmatrix} r & 0 \\ 0 & t \end{pmatrix} \tag{3.66}$$

Eigenwert von A	Eigenvektor der Länge 1
r	$\binom{1}{0}$
t	$\binom{0}{1}$

$$S = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = \mathfrak{R}(0) \tag{3.67}$$

Also gilt

$$\beta = 0. \tag{3.68}$$

Fall 2: $s \neq 0$

Für alle Eigenvektoren $\bar{x}$ zum Eigenwert λ von A gilt:

$$A\bar{x} = \lambda\bar{x} \tag{3.69}$$

$$A\bar{x} - \lambda I_d\bar{x} = \bar{0} \tag{3.70}$$

$$(A - \lambda I_d)\bar{x} = \bar{0} \tag{3.71}$$

$$\begin{pmatrix} r-\lambda & s \\ s & t-\lambda \end{pmatrix} \begin{pmatrix} \bar{x}_1 \\ \bar{x}_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \tag{3.72}$$

Da $\bar{0}$ kein Eigenvektor sein kann, hat das lineare homogene Gleichungssystem (3.72) sicher nichttriviale Lösungen für jeden Eigenwert λ von A. Der Rang von $(A - \lambda I_d)$ kann also nicht 2 sein, weil es dann nur die triviale Lösung $\bar{0}$ gäbe. Wäre der Rang 0, müßte jedes Element der Matrix $(A - \lambda I_d)$ gleich 0 sein. Da dies nicht möglich ist, weil für den hier behandelten Fall ja $s \neq 0$ gilt, folgt, daß der Rang von $(A - \lambda I_d)$ gleich 1 ist für alle Eigenwerte λ von A. Der Lösungsraum für das Gleichungssystem hat damit die Dimension 1 und ist eine Gerade. Wegen $s \neq 0$ ist $\mathcal{L} := \left\{\bar{x} \in \mathbb{R}^2 \mid \bar{x}_2 = \frac{\lambda - r}{s}\bar{x}_1\right\}$ die Menge aller Eigenvektoren zum Eigenwert λ (die möglichen Werte für λ sind c_1, c_2 entsprechend (3.57) und (3.58). $\mathcal{L}$ ist geometrisch interpretiert eine Gerade in $\mathbb{R}^2$, die durch den Nullpunkt geht und die Steigung $\frac{\lambda - r}{s}$ hat. Wegen der Dimension des Lösungsraumes ist die Menge der Eigenvektoren zum Eigenwert λ von A genau $\mathcal{L}$.

Für die Zeilenvektoren von S kommen nur Eigenvektoren $\bar{x}$ der Länge 1 in Frage ($\|\bar{x}\| = 1$); wir wollen sie daher allgemein anschreiben. Aus der Abb. 3.5 sieht man, daß

$$\begin{pmatrix} \cos\arctan\frac{\lambda-r}{s} \\ \sin\arctan\frac{\lambda-r}{s} \end{pmatrix} \quad \text{und} \quad \begin{pmatrix} -\cos\arctan\frac{\lambda-r}{s} \\ -\sin\arctan\frac{\lambda-r}{s} \end{pmatrix} \tag{3.73}$$

die einzigen Eigenvektoren der Länge 1 zum Eigenwert λ sind. Diese beiden Vektoren sind die mit $\bullet$ markierten Punkten in Abb. 3.5.

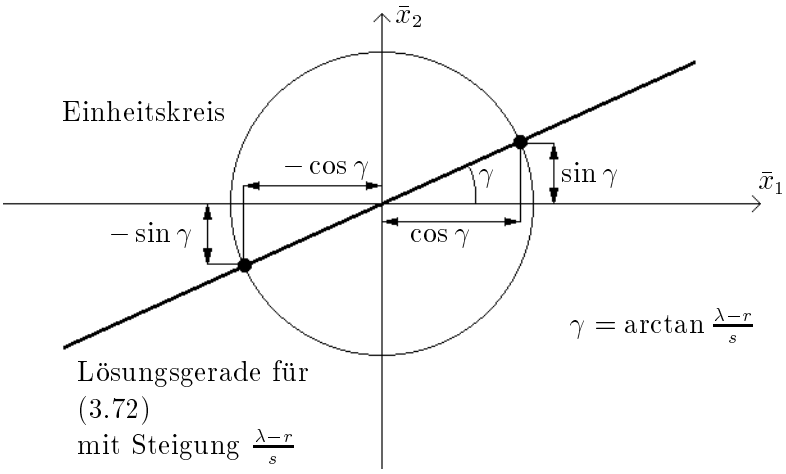


Abbildung 3.5: Eigenvektoren der Länge 1 zum Eigenwert λ von $A = \left(\begin{smallmatrix} r & s \\ s & t \end{smallmatrix}\right)$

Mit

$$\beta_i := \arctan\frac{c_i - r}{s} \qquad i = 1, 2 \tag{3.74}$$

gilt für alle Eigenvektoren $\bar{e}_i$ der Länge 1 zu den Eigenwert c_i von A:

$$\bar{e}_i \in \left\{ \begin{pmatrix} \cos\beta_i \\ \sin\beta_i \end{pmatrix}, -\begin{pmatrix} \cos\beta_i \\ \sin\beta_i \end{pmatrix} \right\} \qquad i = 1, 2 \tag{3.75}$$

Damit sind alle orthogonalen Matrizen S, deren Zeilen Eigenvektoren der Länge 1 zu den entsprechenden Eigenwerten von A in $D = \left(\begin{smallmatrix} c_1 & 0 \\ 0 & c_2 \end{smallmatrix}\right)$ sind, von folgender Gestalt:

$$S = \begin{pmatrix} \bar{e}_1^T \\ \bar{e}_2^T \end{pmatrix} \tag{3.76}$$

Nach Satz 8.9 Korollar 2 [40] sind nämlich die Eigenvektoren zu verschiedenen Eigenwerten einer normalen Matrix orthogonal zueinander. A ist symmetrisch; daher gilt $A^*A = AA^*$ und A ist damit normal. Aus $\bar{e}_1 \perp \bar{e}_2$ läßt sich die Abb. 3.6 ableiten. Jede mögliche Wahl für $\bar{e}_1$ und $\bar{e}_2$ ergibt eine zulässige Matrix S. Setzt man also $\bar{e}_1 = \left(\begin{smallmatrix} \cos\beta_1 \\ \sin\beta_1 \end{smallmatrix}\right)$ und $\bar{e}_2 = -\left(\begin{smallmatrix} \cos\beta_2 \\ \sin\beta_2 \end{smallmatrix}\right)$ fest, dann sieht man wegen $\bar{e}_1 \perp \bar{e}_2$, daß $\bar{e}_2 = \left(\begin{smallmatrix} -\sin\beta_1 \\ \cos\beta_1 \end{smallmatrix}\right)$ gilt. Diesen Schluß kann man folgendermaßen überprüfen: Es gilt $\bar{e}_1 \perp \bar{e}_2 \iff \bar{e}_1^T\bar{e}_2 = 0$. Setzt man die obigen Werte für e_1 und e_2 ein, ergibt sich $\left(\begin{smallmatrix} \cos\beta_1 \\ \sin\beta_1 \end{smallmatrix}\right)^T \left(\begin{smallmatrix} -\sin\beta_1 \\ \cos\beta_1 \end{smallmatrix}\right) = 0$.

Es gilt nun für S:

$$S = \begin{pmatrix} \bar{e}_1^T \\ \bar{e}_2^T \end{pmatrix} = \begin{pmatrix} \cos\beta_1 & \sin\beta_1 \\ -\sin\beta_1 & \cos\beta_1 \end{pmatrix} = \mathfrak{R}(-\beta_1) \tag{3.77}$$

Also erfüllt

$$S = \mathfrak{R}(\beta) \tag{3.78}$$

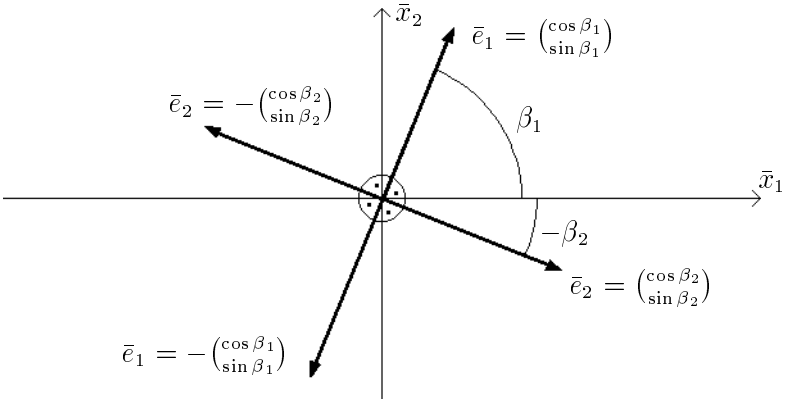


Abbildung 3.6: Alle Eigenvektoren der Länge 1 zu den beiden Eigenwerten von $A = \left(\begin{smallmatrix} r & s \\ s & t \end{smallmatrix}\right)$

mit

$$\beta = -\beta_1 \tag{3.79}$$

die Bedingung

$$D = \begin{pmatrix} c_1 & 0 \\ 0 & c_2 \end{pmatrix} = \mathfrak{S}(c_1, c_2) = SAS^{-1}. \tag{3.80}$$

β liegt modulo 2π im Intervall $[0, 2\pi)$.

Damit wurden alle Fälle für A behandelt, und es wurde gezeigt, daß man zu jeder symmetrischen Matrix A mit den Eigenwerten $c_1, c_2 \in \mathbb{R}$ ein $\beta \in [0, 2\pi)$ finden kann, sodaß mit $S = \mathfrak{R}(\beta)$ und $D = \mathfrak{S}(c_1, c_2)$

$$D = SAS^{-1} \tag{3.81}$$

gilt und daher auch

$$A = S^{-1}DS = \mathfrak{R}(-\beta)\mathfrak{S}(c_1, c_2)\mathfrak{R}(\beta). \tag{3.82}$$

□

Satz 3.16 *Jeder lineare Operator L auf $\mathbb{R}^2$ ist darstellbar als das Produkt einer Drehung, einer Skalierung/Spiegelung und noch einer Drehung.*

$$\forall\, L \in \mathbb{R}^{2\times 2}\ \exists\, \alpha', \beta' \in [0, 2\pi)\ \exists\, c_1, c_2 \in \mathbb{R} : L = \mathfrak{R}(\beta')\mathfrak{S}(c_1, c_2)\mathfrak{R}(\alpha') \tag{3.83}$$

Beweis: Nach Hilfssatz 3.14 gilt:

$$\forall\, L \in \mathbb{R}^{2\times 2}\ \exists\, \alpha \in [0, 2\pi)\ \exists\, A = \begin{pmatrix} r & s \\ s & t \end{pmatrix} \in \mathbb{R}^{2\times 2} : L = A\mathfrak{R}(\alpha) \tag{3.84}$$

Nach Hilfssatz 3.15 gilt:

$$\forall\, A = \begin{pmatrix} r & s \\ s & t \end{pmatrix} \in \mathbb{R}^{2\times 2}\ \exists\, c_1, c_2 \in \mathbb{R}\ \exists\, \beta \in [0, 2\pi) : A = \mathfrak{R}(-\beta)\mathfrak{S}(c_1, c_2)\mathfrak{R}(\beta) \tag{3.85}$$

Daraus folgt, daß für alle $L \in \mathbb{R}^{2\times 2}$ Werte $c_1, c_2 \in \mathbb{R}$ und $\alpha, \beta \in [0, 2\pi)$ existieren, für die folgendes gilt:

$$L = A\mathfrak{R}(\alpha) \tag{3.86}$$
$$= \mathfrak{R}(-\beta)\mathfrak{S}(c_1, c_2)\mathfrak{R}(\beta)\mathfrak{R}(\alpha) \tag{3.87}$$
$$= \mathfrak{R}(-\beta)\mathfrak{S}(c_1, c_2)\mathfrak{R}(\alpha + \beta) \tag{3.88}$$

Mit

$$\beta' = -\beta \quad \text{und} \quad \alpha' = \alpha + \beta \tag{3.89}$$

gilt dann

$$L = \mathfrak{R}(\beta')\mathfrak{S}(c_1, c_2)\mathfrak{R}(\alpha'). \tag{3.90}$$

α' und β' liegen modulo 2π im Intervall $[0, 2\pi)$. □

Zusammenfassung

Hier wird eine Übersicht gegeben, wie man Werte β', c_1, c_2 und α' für einen gegebenen linearen Operator $L = \left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right) \in \mathbb{R}^{2\times 2}$ berechnet, sodaß $L = \mathfrak{R}(\beta')\mathfrak{S}(c_1, c_2)\mathfrak{R}(\alpha')$ ist.

- Berechnung von r, s, t und α aus a, b, c und d:

  Wenn $a = -d$ und $b = c$ ist, dann:

  $$r = a \tag{3.91}$$
  $$s = b \tag{3.92}$$
  $$t = d \tag{3.93}$$
  $$\alpha = 0 \tag{3.94}$$

  Sonst:

  $$r = \frac{a^2 + b^2 + ad - bc}{\sqrt{(c-b)^2 + (a+d)^2}} \tag{3.95}$$
  $$s = \frac{ac - bd}{\sqrt{(c-b)^2 + (a+d)^2}} \tag{3.96}$$
  $$t = \frac{c^2 + d^2 + ad - bc}{\sqrt{(c-b)^2 + (a+d)^2}} \tag{3.97}$$

Um α zu erhalten werden zuerst Werte für $\sin\alpha$ und $\cos\alpha$ berechnet:

$$\sin\alpha = \frac{c-b}{\sqrt{(c-b)^2+(a+d)^2}} \tag{3.98}$$

$$\cos\alpha = \frac{a+d}{\sqrt{(c-b)^2+(a+d)^2}} \tag{3.99}$$

Dann wird eine Fallunterscheidung gemacht:

$\cos\alpha = 0$

$$\alpha = \begin{cases} \frac{3\pi}{2} & \text{wenn } \sin\alpha < 0 \\ \frac{\pi}{2} & \text{sonst} \end{cases} \tag{3.100}$$

$\cos\alpha < 0$

$$\alpha = \pi + \arctan\frac{\sin\alpha}{\cos\alpha} \tag{3.101}$$

$\cos\alpha > 0$

$$\alpha = \begin{cases} \arctan\frac{\sin\alpha}{\cos\alpha} & \text{wenn } \sin\alpha \geq 0 \\ 2\pi + \arctan\frac{\sin\alpha}{\cos\alpha} & \text{sonst} \end{cases} \tag{3.102}$$

- Berechnung von c_1 und c_2 aus r, s, t:

$$c_1 = \frac{r+t+\sqrt{(r-t)^2+4s^2}}{2} \tag{3.103}$$

$$c_2 = \frac{r+t-\sqrt{(r-t)^2+4s^2}}{2} \tag{3.104}$$

- Berechnung von β' aus c_1, r, s, t:

$$\beta' = \begin{cases} \frac{\pi}{2} & \text{wenn } s = 0 \text{ und } r < t \\ 0 & \text{wenn } s = 0 \text{ und } r \geq t \\ \arctan\frac{c_1-r}{s} & \text{sonst} \end{cases} \tag{3.105}$$

- Berechnung von α' aus α und β':

$$\alpha' = \alpha - \beta' \tag{3.106}$$

3.3.3 Kontraktionsfaktor einer affinen Transformation

Der Kontraktionsfaktor ist eine charakteristische Größe einer affinen Transformation. Hier wird eine Methode zur Bestimmung dieser Größe angegeben.

Hilfssatz 3.17 *Für eine affine Transformation $w(\bar{x}) = D\bar{x}$ auf $\mathbb{R}^2$ mit $D = \mathfrak{S}(c_1, c_2)$ und $c_1, c_2 \in \mathbb{R}$ ist $c = \max\{|c_1|, |c_2|\}$ der Kontraktionsfaktor.*

Beweis: Nach Satz 3.8 ist c der Kontraktionsfaktor von w genau dann, wenn erstens

$$\|D\bar{z}\| \leq c\|\bar{z}\| \qquad \forall\, \bar{z} \in \mathbb{R}^2 \tag{3.107}$$

gilt und es zweitens kein $k \in [0, c)$ gibt, sodaß

$$\|D\bar{z}\| \leq k\|\bar{z}\| \qquad \forall\, \bar{z} \in \mathbb{R}^2 \tag{3.108}$$

gilt.

ad (3.107)**:**

$$c = \max\{|c_1|, |c_2|\} \implies c^2 \geq c_1^2 \text{ und } c^2 \geq c_2^2 \tag{3.109}$$

Also gilt

$$c_1^2\bar{z}_1^2 + c_2^2\bar{z}_2^2 \leq c^2(\bar{z}_1^2 + \bar{z}_2^2) \qquad \forall\, \bar{z} = \left(\begin{smallmatrix}\bar{z}_1\\ \bar{z}_2\end{smallmatrix}\right) \in \mathbb{R}^2. \tag{3.110}$$

Da alle Komponenten positiv sind, kann man auf der linken und auf der rechten Seite die Wurzel ziehen:

$$\sqrt{c_1^2\bar{z}_1^2 + c_2^2\bar{z}_2^2} \leq c\sqrt{(\bar{z}_1^2 + \bar{z}_2^2)} \qquad \forall\, \bar{z} \in \mathbb{R}^2 \tag{3.111}$$

Daraus folgt

$$\|D\bar{z}\| \leq c\|\bar{z}\| \qquad \forall\, \bar{z} \in \mathbb{R}^2, \tag{3.112}$$

weil $D = \mathfrak{S}(c_1, c_2) = \left(\begin{smallmatrix} c_1 & 0 \\ 0 & c_2 \end{smallmatrix}\right)$ ist.

ad (3.108)**:**

Fall 1: $|c_1| \geq |c_2|$ (dann gilt $c = |c_1|$)

Angenommen, es gibt ein $k \in [0, |c_1|)$, das (3.108) erfüllt. (3.108) ist für alle $\bar{z} \in \mathbb{R}^2$ erfüllt, also auch für $\binom{1}{0}$:

$$\left\| D\begin{pmatrix}1\\0\end{pmatrix} \right\| \leq k \left\| \begin{pmatrix}1\\0\end{pmatrix} \right\| \tag{3.113}$$

$$\left\| \begin{pmatrix}c_1\\0\end{pmatrix} \right\| \leq k \left\| \begin{pmatrix}1\\0\end{pmatrix} \right\| \tag{3.114}$$

$$\sqrt{c_1^2} \leq k \tag{3.115}$$

$$|c_1| \leq k \tag{3.116}$$

Das ist ein Widerspruch zur Annahme $k \in [0, |c_1|)$.

Fall 2: $|c_1| < |c_2|$ (dann gilt $c = |c_2|$)
Die Beweisführung ist analog zum Fall 1 mit $\bar{z} = \binom{0}{1}$.

Alle Fälle haben auf einen Widerspruch geführt, deshalb gibt es kein k mit den genannten Eigenschaften.

□

Satz 3.18 *Ist $w(\bar{x}) = L\bar{x} + \bar{t}$ eine affine Transformation auf $\mathbb{R}^2$, so gibt es $\alpha, \beta \in [0, 2\pi)$ und $c_1, c_2 \in \mathbb{R}$, sodaß $L = \mathfrak{R}(\beta)\mathfrak{S}(c_1, c_2)\mathfrak{R}(\alpha)$, und $c = \max\{|c_1|, |c_2|\}$ ist der Kontraktionsfaktor von w. (w ist kontraktiv $\iff c < 1$)*

Beweis: Nach Hilfssatz 3.15 existieren α, β, c_1 und c_2 wie angegeben. Nach Hilfssatz 3.17 ist c der Kontraktionsfaktor von $\mathfrak{S}(c_1, c_2)$. Mit Satz 3.8 ist

$$\|\mathfrak{S}(c_1, c_2)\bar{z}\| = \|L\bar{z}\| \qquad \forall\, \bar{z} \in \mathbb{R}^2 \tag{3.117}$$

eine notwendige und hinreichende Bedingung, daß c der Kontraktionsfaktor von w ist. Da $L = \mathfrak{R}(\beta)\mathfrak{S}(c_1, c_2)\mathfrak{R}(\alpha)$ ist und eine Drehung an der Länge eines Vektors (also an seiner Norm) nichts ändert, gilt (3.117) sicherlich. □

3.3.4 Fixpunkt einer kontraktiven affinen Transformation

Der Fixpunkt einer kontraktiven Transformation ist für ihre Wirkung als Teil eines IFS von großer Wichtigkeit, daher wird hier eine explizite Formel für seine Berechnung angegeben.

Satz 3.19 *Ist $w(\bar{x}) = L\bar{x} + \bar{t}$ eine kontraktive affine Transformation auf $\mathbb{R}^2$ mit $L = \left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right)$ und $\bar{t} = \binom{e}{f}$, so ist*

$$\bar{x}_F = \frac{1}{(1-a)(1-d) - bc}\begin{pmatrix} e + bf - de \\ f - af + ce \end{pmatrix} \tag{3.118}$$

der einzige Fixpunkt von w.

Beweis: Nach Satz 3.10 exisitiert $\bar{x}_F = (I_d - L)^{-1}\bar{t}$ und ist der einzige Fixpunkt von w. Damit gilt:

$$\bar{x}_F = (I_d - L)^{-1}\bar{t} \tag{3.119}$$

$$= \left(\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} - \begin{pmatrix} a & b \\ c & d \end{pmatrix}\right)^{-1} \begin{pmatrix} e \\ f \end{pmatrix} \tag{3.120}$$

$$= \begin{pmatrix} 1-a & -b \\ -c & 1-d \end{pmatrix}^{-1} \begin{pmatrix} e \\ f \end{pmatrix} \tag{3.121}$$

$$= \frac{1}{(1-a)(1-d)-bc} \begin{pmatrix} 1-d & b \\ c & 1-a \end{pmatrix} \begin{pmatrix} e \\ f \end{pmatrix} \tag{3.122}$$

$$= \frac{1}{(1-a)(1-d)-bc} \begin{pmatrix} (1-d)e + bf \\ ce + (1-a)f \end{pmatrix} \tag{3.123}$$

$$= \frac{1}{(1-a)(1-d)-bc} \begin{pmatrix} e + bf - de \\ f - af + ce \end{pmatrix} \tag{3.124}$$

□

Kapitel 4

Vom IFS zum Bild

In diesem Kapitel wird erläutert, wie es möglich ist, ein digitales Bild (also ein Binär–, Grauwert– oder Farbbild) eines Attraktors zu erlangen, der von einem IFS in $\mathbb{R}^2$ definiert wird. Zuerst wird eine gut bekannte Methode vorgestellt. Anschließend wird gezeigt, wie es möglich ist, dies mit Hilfe von diskreten Transformationen effizienter durchzuführen. Dabei werden die Begriffe diskreter Attraktor und diskretes invariantes Maß eingeführt. Mittels Bildpyramiden [27] wird eine Effizienzsteigerung erlangt und fraktale Codierung mit Pyramidencodierung kombiniert.

4.1 Abtasten des Attraktors

Ein IFS in $\mathbb{R}^2$ definiert einen Attraktor, der eine kompakte Teilmenge von $\mathbb{R}^2$ ist. Diese Teilmenge kann nun wie folgt als Binärbild auf den Bildschirm gebracht werden. Der Bildschirm wird mit seiner Indexmenge identifiziert, die eine endliche Teilmenge von $\mathbb{N}^2$ ist. Der Teilbereich von $\mathbb{R}^2$, der den Attraktor enthält, wird in kleine Quadrate unterteilt, die mit den Bildschirmpixeln korrespondieren. Ein Pixel wird dann und nur dann auf 1 gesetzt, wenn das korrespondierende Quadrat mindestens einen Punkt des Attraktors enthält. Das resultierende Binärbild wird der *abgetastete Attraktor* genannt. Wenn man jedem Pixel als Grauwert den normierten Maßwert[1] seines korrespondierenden Quadrates zuordnet, erhält man ein Grauwertbild, das *abgetastete invariante Maß*. Das Chaosspiel [5] ist die häufigst angewandte Methode zum Abtasten des Attraktors $\mathcal{A}$ und des invarianten Maßes eines IFS.

[1] Also dividiert durch das Maximum der Maßwerte der Quadrate aller Pixel.

Chaosspiel(IFS $\{w_i \mid i = 1, \ldots, n\}$, $\{p_i \mid i = 1, \ldots, n\}$, $\bar{x}_0$):

- Der Anfangspunkt $\bar{x}_0$ muß in $\mathcal{A}$ liegen, dies ist z.B. für die Fixpunkte der Transformationen des IFS der Fall. j wird auf 0 gesetzt.
- **repeat**

  Das mit $\bar{x}_j$ korrespondierende Pixel wird gesetzt. Nun wird $\bar{x}_{j+1} = w_{i_j}(\bar{x}_j)$ berechnet, wobei $i_j \in \{1, \ldots, n\}$ zufällig, mit Wahrscheinlichkeit p_i gewählt wird.

  until Qualität des Attraktorbildes ist gut genug.

Die Schleife kann abgebrochen werden, wenn keine neuen Pixeln mehr gesetzt werden[2]; dann kann man davon ausgehen, daß die Qualität des berechneten Bildes gut genug ist. Bei der Anzahl der Iterationen wird in der Praxis mit Erfahrungswerten operiert. Da die ermittelte Punktfolge zufällig über dem Attraktor verteilt ist, wird der Attraktor durch diese Punktfolge regelmäßig überdeckt. Nach einer Anzahl von Iterationen, die deutlich größer als die Anzahl der Pixel des Attraktors ist, wird der Attraktor dicht genug überdeckt sein (Abschnitt 2.4 in [8]).

Der Einfluß der Wahrscheinlichkeiten p_i ist wie folgt: Wenn die Wahrscheinlichkeiten entsprechend der Fläche des Attraktors unter den Transformationen gewählt werden, wird der Attraktor regelmäßiger, und daher schneller, mit Punkten überdeckt. Das abgetastete invariante Maß erhält man, wenn gezählt wird, wie oft ein Pixel beim Chaosspiel getroffen wird. Dieser Wert wird mittels Dividieren durch den maximalen Trefferwert normalisiert, diese Größe bestimmt Grauwert (oder über eine Tabelle den Farbwert) des Pixels[3].

Attraktoren mit dem Chaosspiel abzutasten ist aufwendig, denn der Attraktor wird weit genauer berechnet, als zur Darstellung notwendig ist. Dies ist deshalb der Fall, weil durch das Chaosspiel der wirkliche Attraktor angenähert wird, der ja eine Teilmenge von $\mathbb{R}^2$ ist. Da nicht offensichtlich ist, wie viele Iterationen für eine bestimmte Bildschirmauflösung genügen, muß die Zahl der Iterationen entsprechend hoch gewählt werden, sodaß viele Pixel öfter als einmal gesetzt werden.

4.2 Diskrete Transformationen

Zur direkten Berechnung des Attraktors und seines invarianten Maßes für eine gegebene Bildschirmauflösung ist es notwendig, beide als endliche Mengen darzustellen. Es wird der diskrete

[2] Wegen der probabilistischen Natur des Algorithmus kann nicht festgestellt werden wann das der Fall ist, deshalb wird abgebrochen, wenn eine gewisse Zeit keine neuen Pixel mehr gesetzt wurden.

[3] Dies muß von der Möglichkeit unterschieden werden, Grauwert–/Farbbilder durch Attraktoren von IFS in $\mathbb{R}^3$ bzw. $\mathbb{R}^5$ zu erzeugen (siehe Einführung zu Kapitel 3).

Raum $\mathbb{P}^2 \subset \mathbb{N}^2$ der Pixel des Bildschirms verwendet. Der Einfachheit halber wird ein quadratischer Schirm angenommen. Für einen Bildschirm mit der Größe $R \in \mathbb{N}$ wird definiert:

$$\mathbb{P}_R := \{0, 1, \ldots, R-1\} \tag{4.1}$$

und

$$\mathbb{P}_R^2 := \mathbb{P}_R \times \mathbb{P}_R. \tag{4.2}$$

Man identifiziert den Bildschirm mit $[0, R) \times [0, R) \subset \mathbb{R}^2$; jedes Pixel $p = (p_x, p_y) \in \mathbb{P}_R^2$ repräsentiert ein Einheitsquadrat $[p_x, p_x + 1) \times [p_y, p_y + 1) \subset \mathbb{R}^2$. Von nun an wird $\mathbb{P}$ anstelle von $\mathbb{P}_R$ verwendet. Der Einfachheit halber wird angenommen, daß der Attraktor vollständig in $[0, R) \times [0, R)$ liegt, was aber keine Einschränkung darstellt, da jedes IFS in ein IFS transformiert werden kann, dessen Attraktor diese Bedingung erfüllt, und der seine Form bis auf Verkleinerung und Verschiebung behält [5].

Da der Attraktor eines IFS eine Menge ist, muß eine Beziehung zwischen Teilmengen von $\mathbb{P}^2$ und Binärbildern hergestellt werden. Pixelmengen $I \in \mathcal{H}(\mathbb{P}^2)$ und Binärbilder sind isomorph zueinander, wenn die Pixel $\in I$ genau die gesetzten Pixel des Binärbildes sind. Deshalb ist es zulässig, Pixelmengen $I \in \mathcal{H}(\mathbb{P}^2)$ der Einfachheit halber als Bilder zu bezeichnen. Die Grauwerte oder Farben eines Bildes I werden durch Bestimmung der Maßwerte μ_p für alle $p \in I$ definiert. Es muß auch die Bedeutung von $\mathbf{w}$ und $\mathbf{M}$ auf $\mathbb{P}^2$ definiert werden; dazu wird eine Verbindung zwischen $\mathbb{P}^2$ und $\mathbb{R}^2$ hergestellt: Man führt eine Abbildung $\varrho : \mathbb{P}^2 \to \mathcal{B}(\mathbb{R}^2)$ ein[4], d.h. ϱ gibt für jedes Pixel diejenige Teilmenge von $\mathbb{R}^2$ an, durch die es repräsentiert wird. Zwei Möglichkeiten werden behandelt:

1. Jedes Pixel wird durch seinen Mittelpunkt repräsentiert.

$$\varrho^+(p) := \left\{ \begin{pmatrix} p_x + 0.5 \\ p_y + 0.5 \end{pmatrix} \in \mathbb{R}^2 \right\} \qquad p \in \mathbb{P}^2 \tag{4.3}$$

2. Jedes Pixel wird durch eine Teilmenge von $\mathbb{R}^2$ in Form eines Einheitsquadrats repräsentiert.

$$\varrho^\square(p) := [p_x, p_x + 1) \times [p_y, p_y + 1) \qquad p \in \mathbb{P}^2 \tag{4.4}$$

Die erste Wahl führt zu einfachen Algorithmen und ist außerdem exakter, wie in Abschnitt 4.3.4 gezeigt werden wird.

Nach Anwendung einer affinen Transformation auf $\varrho(p)$ für ein Pixel p muß das Resultat wiederum nach $\mathbb{P}^2$ abgebildet werden. Dazu dient die Funktion $\Delta : \mathcal{B}(\mathbb{R}^2) \to \mathcal{H}(\mathbb{P}^2)$:

1. Werden die Pixel durch ihren Mittelpunkt repräsentiert, muß Δ^+ auf Punkten in $\mathbb{R}^2$ operieren. Δ^+ liefert dann das Pixel, das einen gegebenen Punkt enthält.

$$\Delta^+(\{\bar{x}\}) := \{(\lfloor \bar{x}_1 \rfloor, \lfloor \bar{x}_2 \rfloor) \in \mathbb{P}^2\} \qquad \bar{x} = \tbinom{\bar{x}_1}{\bar{x}_2} \in \mathbb{R}^2 \tag{4.5}$$

[4] Es wird $\mathcal{B}(\mathbb{R}^2)$ verwendet, weil ein Pixel ein halboffenes Quadrat repräsentiert. $\mathcal{B}(\mathbb{R}^2)$ ist dabei die Borel–σ–Algebra von $\mathbb{R}^2$.

2. Werden die Pixel als quadratische Teilmengen von $\mathbb{R}^2$ repräsentiert, ist das Resultat der Anwendung einer affinen Transformation auf $\varrho(p)$ ein Parallelogramm. Deshalb bildet $\Delta^\square$ eine Teilmenge von $\mathbb{R}^2$ (ein Parallelogramm) auf jene Pixel ab, die mit ihr eine nicht leere Schnittmenge haben.

$$\Delta^\square(A) := \{p \in \mathbb{P}^2 \mid \varrho^\square(p) \cap A \neq \emptyset\} \qquad A \in \mathcal{B}(\mathbb{R}^2) \tag{4.6}$$

Es ist sicher, daß das Resultat von Δ innerhalb von $\mathbb{P}^2$ liegt; alles außerhalb des Bildschirms wird verworfen. Wenn im folgenden ϱ oder Δ geschrieben wird, ist abhängig von der Pixelrepräsentation ϱ^+ oder Δ^+ bzw. $\varrho^\square$ oder $\Delta^\square$ gemeint.

Für eine affine Transformation $w : \mathbb{R}^2 \to \mathbb{R}^2$ ist ihre zugehörige *diskrete Transformation* $w' : \mathbb{P}^2 \to \mathcal{H}(\mathbb{P}^2)$ wie folgt definiert:

$$w'(p) := (\Delta \circ w \circ \varrho)(p) \qquad p \in \mathbb{P}^2 \tag{4.7}$$

Das Pixel p wird zuerst mittels ϱ nach $\mathbb{R}^2$ abgebildet, dann wird die Transformation w angewendet, und die resultierende Teilmenge von $\mathbb{R}^2$ wird schließlich durch Δ in eine Menge von Pixeln transformiert. Diese Diskretisierung kann auch auf andere als affine Transformationen angewendet werden. In Abb. 4.1 wird gezeigt, wie ein Pixel auf ein anderes/auf andere abgebildet wird. Links wird die Mittelpunktrepräsentation verwendet, rechts die Repräsentation als Quadrat.

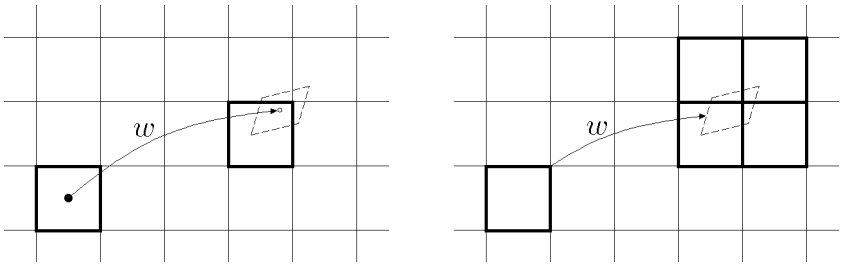


Abbildung 4.1: Verschiedene Methoden der Pixelrepräsentation

4.2.1 Eine Erweiterung der Diskretisierung

Diskrete kontraktive Transformationen mit Mittelpunktsrepräsentation haben nicht notwendigerweise ein einziges Pixel als Fixpunkt. Jedes Pixel dessen Mittelpunkt innerhalb des Pixels bleibt, wenn er transformiert wird, ist ein Fixpunktpixel. Die Eigenschaft des eindeutigen Fixpunktes bleibt erhalten, wenn folgende Erweiterung der Diskretisierung durchgeführt wird: Bildet eine diskrete Transformation w' ein Pixel auf sich selbst ab, dann kann das Pixel den Fixpunkt der Transformation enthalten oder nicht. Im ersten Fall wird nichts geändert, das Pixel wird auf sich selbst abgebildet. Im zweiten Fall wird die Transformation w (nicht w') so oft angewendet, bis ein anderes Pixel erreicht wird. Experimente (Abschnitt 4.3.4) haben die Nützlichkeit dieser Methode bestätigt, sie wurde jedoch noch keiner mathematische Analyse unterzogen.

4.3 Diskreter Attraktor

Der diskrete Hutchinsonoperator $\mathbf{w}' : \mathcal{H}(\mathbb{P}^2) \to \mathcal{H}(\mathbb{P}^2)$ eines IFS $\{w_i \mid i = 1, \ldots, n\}$ bildet eine Pixelmenge auf eine Pixelmenge ab und ist folgendermaßen definiert:

$$\mathbf{w}'(I) := \bigcup_{i=1}^{n} \bigcup_{p \in I} w_i'(p) \qquad I \in \mathcal{H}(\mathbb{P}^2) \tag{4.8}$$

Jedes Pixel des Bildes I wird durch jede Transformation des IFS diskret transformiert.

Definition 4.1 Hat man ein IFS mit diskretem Hutchinsonoperator $\mathbf{w}'$, so wird jede Menge $\mathcal{A}_d \in \mathcal{H}(\mathbb{P}^2)$, mit

$$\mathcal{A}_d = \mathbf{w}'(\mathcal{A}_d) \tag{4.9}$$

ein *diskreter Attraktor* genannt.

Der diskrete Attraktor ist nicht eindeutig, wie es beim tatsächlichen Attraktor der Fall ist (Satz 2.13), da durch die Diskretisierung $\mathbf{w}'$ im allgemeinen keine kontraktive Transformation mehr ist. Es existiert jedoch für jedes IFS genau eine maximale Lösung $\mathcal{A}_d^\bullet$, die alle diskreten Attraktoren beeinhaltet.

Satz 4.2 *Für ein IFS mit diskreten Hutchinsonoperator* $\mathbf{w}'$ *gibt es ein eindeutiges* $\mathcal{A}_d^\bullet \in \mathcal{H}(\mathbb{P}^2)$ *mit*

$$\mathcal{A}_d = \mathbf{w}'(\mathcal{A}_d) \implies \mathcal{A}_d \subseteq \mathcal{A}_d^\bullet \qquad \forall\, \mathcal{A}_d \in \mathcal{H}(\mathbb{P}^2). \tag{4.10}$$

Beweis: Es gibt nur eine endliche Anzahl von diskreten Attraktoren für ein IFS, da $\mathcal{H}(\mathbb{P}^2)$ endlich ist. Bildet man die Vereinigung aller dieser diskreten Attraktoren, so ist das Ergebnis notwendigerweise ein diskreter Attraktor, der jeden anderen diskreten Attraktor enthält. Die Eindeutigkeit folgt aus der Tatsache, daß zwei maximale diskrete Attraktoren jeweils Teilmenge des anderen und somit identisch sind. □

Es folgen nun zwei Algorithmen zur Berechnung eines diskreten Attraktors, weiters wird eine Schranke für den Unterschied zwischen dem tatsächlichen und einem diskreten Attraktor angegeben.

4.3.1 Algorithmus RemoveBackground

Der hier präsentierte Algorithmus berechnet den maximalen Attraktor für ein IFS $\{w_i \mid i = 1, \ldots, n\}$.

Sei

$$G := (\mathbb{P}^2, E) \tag{4.11}$$

ein gerichteter Graph, mit Knotenmenge $\mathbb{P}^2$ (alle Pixel des Bildschirms) und Kantenmenge E. (p, q) ist genau dann eine Kante des Graphen, wenn Pixel p auf Pixel q abgebildet wird:

$$E := \big\{(p,q) \in \mathbb{P}^2 \times \mathbb{P}^2 \mid \exists\, i \in \{1, \ldots, n\} : q \in w_i'(p)\big\} \tag{4.12}$$

Der Effekt der Transformationen des IFS (und hier ist man nicht auf affine beschränkt) kann so vollständig durch einen gerichteten Graphen beschrieben werden.

Man kann jedem $I \in \mathcal{H}(\mathbb{P}^2)$ eindeutig einen Graphen G_I zuordnen, indem man den gerichteten Graphen G auf die Pixel von I beschränkt:

$$G_I := (I, E_I) \quad \text{mit} \quad E_I := E \cap (I \times I) \tag{4.13}$$

Welche Eigenschaften hat nun der Graph G_I, wenn I ein diskreter Attraktor ist? Für die Beantwortung dieser Frage erweist es sich als günstig, die Fixpunktbedingung wie folgt aufzuspalten:

$$I = \mathbf{w}'(I) \iff \mathbf{w}'(I) \subseteq I \text{ und } I \subseteq \mathbf{w}'(I) \tag{4.14}$$

Wir wollen nun diese beiden Bedingungen in Bezug auf den Graphen G untersuchen:

1. $\mathbf{w}'(I)$ ist genau dann eine Teilmenge von I, wenn alle Pixel von I wiederum auf Pixel von I abgebildet werden. Das ist genau dann der Fall, wenn im Graphen G alle Kanten von Knoten aus I wieder zu Knoten aus I führen. Anschaulich kann man sagen, daß dies der Fall ist, wenn bei der Einschränkung von G auf G_I eine Kante genau dann entfernt wird, wenn auch die von ihr verbundenen Knoten entfernt werden; oder umgekehrt, wenn in G_I genau die von Knoten aus I wegführenden Kanten von G enthalten sind.

   $$\mathbf{w}'(I) \subseteq I \iff E_I = E \cap (I \times \mathbb{P}^2) \tag{4.15}$$

2. I ist genau dann eine Teilmenge von $\mathbf{w}'(I)$, wenn alle Pixel von I im Bild mindestens eines Pixels von I unter $\mathbf{w}'$ enthalten sind. Das ist genau dann der Fall, wenn im Graphen G_I alle Knoten mindestens eine einlaufende Kante haben.

   $$I \subseteq \mathbf{w}'(I) \iff \forall\, p \in I : E_I \cap (I \times \{p\}) \neq \emptyset \tag{4.16}$$

Zusammengefaßt gilt also, daß I genau dann ein diskreter Attraktor ist, wenn die Kanten von G_I genau die von Knoten aus I wegführenden Kanten in G sind und alle Knoten in G_I mindestens eine einlaufende Kante haben:

$$I = \mathbf{w}'(I) \iff E_I = E \cap (I \times \mathbb{P}^2) \text{ und } \forall\, p \in I : E_I \cap (I \times \{p\}) \neq \emptyset \tag{4.17}$$

Die maximale Teilmenge von $\mathbb{P}^2$ für die diese Bedingung gilt, ist der maximale diskrete Attraktor. Um diesen maximalen diskreten Attraktor zu erhalten, bildet RemoveBackground anfangs den Graphen G mit den Pixeln des gesamten Bildschirms (also $I = \mathbb{P}^2$), um dann sukkzessive die Knoten aus I zu entfernen, die keine einlaufenden Kanten haben und daher nicht zu einem diskreten Attraktor gehören können. Da somit niemals ein Knoten aus I entfernt wird, der eine einlaufende Kante von einem Knoten aus I hat, kann es auch keine Kante in G geben, die von einem Knoten aus I zu einem Knoten führt, der nicht in I enthalten ist, dieser wäre dann eben nicht entfernt worden.

RemoveBackground(IFS $\{w_i \mid i = 1, \ldots, n\}$):

- Wir starten mit allen Pixeln des Bildschirms.
$$I \leftarrow \mathbb{P}^2$$
- $G = (\mathbb{P}^2, E)$ wird mittels (4.12) berechnet. Es gilt $G = G_I$.
- Die Knoten ohne einlaufende Kante in G_I werden solange aus I gelöscht, bis keine solchen mehr vorhanden sind:
  loop

  Die Knoten in I ohne einlaufende Kante werden bestimmt:
$$K \leftarrow \{p \in I \mid E_I \cap (I \times \{p\}) = \emptyset\}$$
  exitif $(K = \emptyset)$

  Die Knoten von K werden in I gelöscht:
$$I \leftarrow I \setminus K$$
  endloop
- I wird als Ergebnis geliefert.

Satz 4.3 *RemoveBackground berechnet den maximalen diskreten Attraktor eines IFS.*

Beweis: Wie vorher angenommen, liegt jeder diskrete Attraktor vollständig innerhalb des Bildschirms, und RemoveBackground beginnt mit der Menge $\mathbb{P}^2$; deshalb wird die maximale Teilmenge I von $\mathbb{P}^2$ berechnet, für die G_I ein Graph ist, dessen Knoten alle mindestens eine einlaufende Kante haben und wo die Kanten von G_I genau die von Knoten aus I wegführenden Kanten in G sind, da kein Knoten aus I entfernt wird, wenn er eine solche einlaufende Kante besitzt. Deshalb muß I der maximale diskrete Attraktor sein. □

Wird dieser Algorithmus auf einem sequentiellen Computer implementiert, so werden $\mathrm{O}(nR^2)$ Schritte zum Aufbauen des Graphen benötigt. In $\mathrm{O}(R^2)$ Schritten findet man die erste Schicht zu entfernender Pixeln. Da insgesamt $R^2 - \#\mathcal{A}_d$ Pixel entfernt werden, und jedes dieser Pixel maximal n Nachfolger hat, die dann ebenfalls geprüft werden müssen, werden $\mathrm{O}(n(R^2 - \#\mathcal{A}_d))$ Schritte zum Entfernen des Hintergrundes benötigt. Es ergibt sich ein Gesamtaufwand von $\mathrm{O}(nR^2)$ sequentiellen Schritten.

Dieser Algorithmus eignet sich besonders für die Implementierung auf Parallelrechnern, wo jedes Pixel durch einen Prozessor vertreten ist. In n parallelen Schritten wird der Graph G berechnet (jedes Pixel wird n-mal transformiert). In jeweils einem parallelen Schritt werden dann jeweils die Knoten ohne einlaufende Kante entfernt. Wieviele solche Schritte notwendig sind, soll nun untersucht werden. Beim Entfernen der Hintergrundpixel ist I nach dem i-ten Schritt genau $\mathbf{w}'^{\circ i}(\mathbb{P}^2)$: Es werden immer die Pixel ohne einlaufende Kante entfernt. Im ersten Schritt ist $I = \mathbb{P}^2$, die Pixel von $I = \mathbf{w}'(I) = \mathbf{w}'(\mathbb{P}^2)$ werden sicherlich nicht entfernt, aber alle anderen

schon. Im zweiten Schritt haben nun nur mehr die Pixel von $\mathbf{w}'^{\circ 2}(\mathbb{P}^2)$ einlaufende Kanten, usw. Die Größe $\sigma \in (0,1)$ mit

$$\sigma := \frac{\text{area}(\mathbf{w}([0,R) \times [0,R)) \cap [0,R) \times [0,R))}{R^2} \tag{4.18}$$

gibt an, um welchen Faktor die Fläche[5] des gesamten Bildschirms (gleich R^2) bei wiederholter Anwendung von $\mathbf{w}$ auf den Bildschirm (also auf $[0,R) \times [0,R) \in \mathcal{B}(\mathbb{R}^2)$) jeweils vermindert wird. Wenn man den Diskretisierungsfehler vernachlässigt, dann gilt derselbe Faktor auch für $\mathbf{w}'$, wobei die Fläche durch die Anzahl der Pixel ersetzt wird. Es gilt dann

$$\#\mathbf{w}'^{\circ k}(\mathbb{P}^2) = \sigma^k R^2 \qquad k \in \mathbb{N}. \tag{4.19}$$

Aus

$$\#\mathbf{w}'^{\circ m}(\mathbb{P}^2) = \#\mathcal{A}_d \tag{4.20}$$

erhält man dann

$$m = \left\lceil \log_\sigma \frac{\#\mathcal{A}_d}{R^2} \right\rceil \tag{4.21}$$

als die Anzahl an Schritten, die benötigt werden um den diskreten Attraktor $\mathcal{A}_d$ zu erreichen, wenn man den Diskretisierungsfehler vernachlässigt. Also kann man schließen, daß asymptotisch für eine gegen 0 gehende Pixelbreite

$$\mathrm{O}\left(\log_\sigma \frac{\#\mathcal{A}_d}{R^2}\right) \tag{4.22}$$

die Anzahl der benötigten Schritte ist.

4.3.2 Algorithmus RemoveBackground mit Bildpyramiden

In diesem Abschnitt wird eine nicht überlappende $2 \times 2/4$ Pyramidenstruktur benutzt, wie sie in [27] beschrieben ist.

Es ist sinnvoll, möglichst große Bereiche auf dem Bildschirm zu ermitteln, in denen sicher kein Pixel des Attraktors liegt. Diese Bereiche müssen dann im Algorithmus RemoveBackground von Abschnitt 4.3.1 nicht berücksichtigt werden. Im folgenden wird eine Methode zum Auffinden solcher Bereiche beschrieben.

Abb. 4.2 zeigt den Attraktor von Barnsleys Farn [5], der mit RemoveBackground für die Auflösungen 256×256, 128×128, 64×64, 32×32, 16×16 und 8×8 berechnet wurde. Bei einer Auflösung von 16×16 liegt mehr als die Hälfte der Pixel außerhalb des Attraktors; diese müssen deshalb in höheren Auflösungen nicht mehr betrachtet werden.

[5] area(A) bezeichnet den Flächeninhalt (genauer das Lebesgue–Maß) einer Menge $A \in \mathcal{B}(\mathbb{R}^2)$.

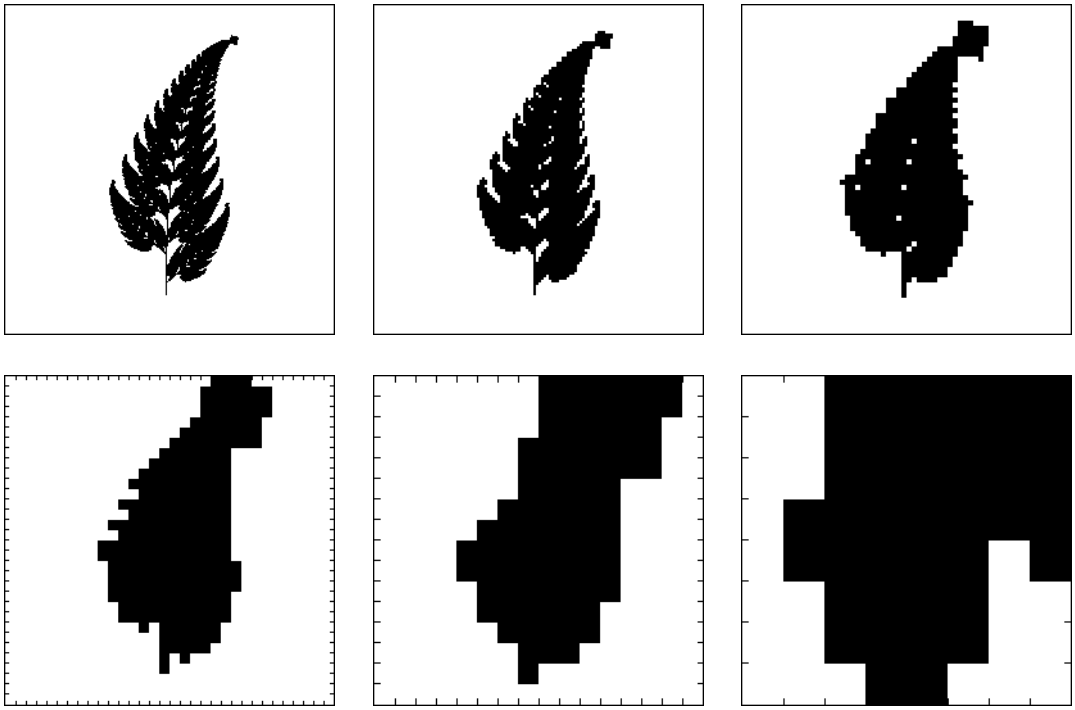


Abbildung 4.2: Eine mit RemoveBackground generierte Pyramide von Barnsleys Farn

Die Bilder werden nach ihrer Auflösung in Ebenen angeordnet und ergeben so die Bildpyramide. I_0 bezeichnet die Ebene des Bildes größter Auflösung (hier 256×256), I_1 die Ebene zweitgrößter Auflösung (hier 128×128), usw. Es wird angenommen, daß die Attraktorpixel und gegebenenfalls ihre Maße auf Ebene I_i der Bildpyramide bekannt sind. Man versieht nun jedes dieser Attraktorpixel mit Zeigern auf seine vier Söhne in der Ebene I_{i-1} und bestimmt somit die Pixel, auf die der Algorithmus RemoveBackground in Ebene I_{i-1} angewendet wird. RemoveBackground wird also auf Ebene I_{i-1} nur auf jene Pixel angewendet, die Sohn eines Attraktorpixels der Ebene I_i sind. Man beginnt in der obersten Ebene I_N der Bildpyramide, siehe Abb. 4.3 wo dieses Verfahren anhand zweier Ebenen einer solchen Pyramide illustriert wird. Attraktorpixel sind dort grau gefärbt. Sind Maße vorhanden, können die Zeiger mit Gewichten versehen werden, die es ermöglichen, Maße für die Söhne aus den Maßen der Vaterpixel zu ermitteln. Wenn der Prozeß in der Ebene I_0 endet, ist die gesamte Bildinformation in den Zeigern enthalten.

Effiziente Bildrekonstruktion mittels Bildpyramiden

Die gewichteten Zeiger der Bildpyramide können dazu verwendet werden, das Originalbild in $O(\log R)$ parallelen Schritten[6] zu generieren. Angenommen, man möchte ein Originalbild mit

[6] Also proportional der Anzahl der Pyramidenebenen, die wiederum logarithmisch mit der Breite des Bildes wächst.

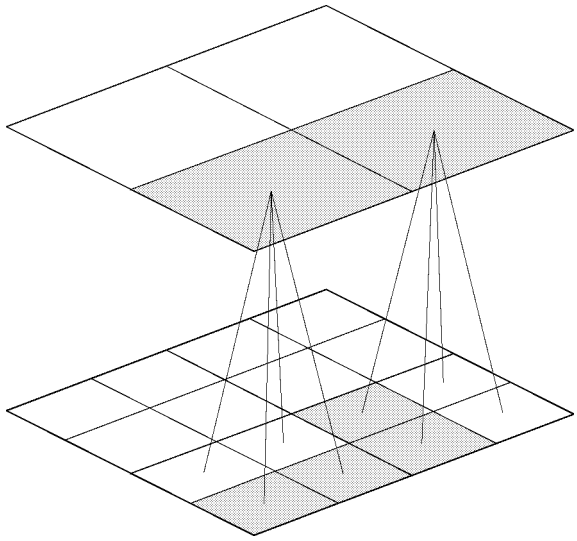


Abbildung 4.3: Schema von RemoveBackground mit Bildpyramide

256×256 Pixeln darstellen: Im ersten Schritt werden, ausgehend von Ebene I_8 der Pyramide die vier Pixel auf Ebene I_7 erzeugt, im zweiten die 16 Pixel auf Ebene I_6, usw., bis mit dem achten Schritt das originale 256×256 Bild erzeugt wird. Man kann sagen, daß alle Punkte innerhalb der Pixel des Originalbildes im Pixel an der Spitze der Pyramide summiert sind. Sie können abhängig von den Gewichten der Zeiger an die Pixel in den unteren Ebenen weitergegeben werden. Die ursprüngliche Punkteverteilung wird in einem parallelen Top–Down–Schritt durch die Pyramide korrekt wiederhergestellt.

Ein illustratives Beispiel in $\mathbb{R}$

Die Arbeitsweise von RemoveBackground mit Pyramiden soll mit einem Beispiel in $\mathbb{R}$ veranschaulicht werden (Abb. 4.4).

Der Attraktor des IFS $\{w_1, w_2\}$ mit

$$w_1(\bar{x}) = \frac{1}{4}\bar{x} \qquad w_2(\bar{x}) = \frac{1}{2}(\bar{x} - 16) + 16 \qquad \bar{x} \in \mathbb{R} \tag{4.23}$$

wird berechnet. Algorithmus RemoveBackground löscht diejenigen Pixel, die keine einlaufende Kante haben, was die vier Graphen in Abb. 4.4 erzeugt. Die Kanten der Graphen sind mit den korrespondierenden Transformationen beschriftet. Pixel bzw. Intervalle werden durch ihre Zentren repräsentiert, die als kleine Kreise gezeichnet sind. Der Attraktor wird mit fetten Linien dargestellt. Ein Pixel p wird auf das Pixel q abgebildet, wenn das Zentrum von p bei einer Transformation in Pixel q fällt.

Im Beispiel beginnt man an der Spitze der Bildpyramide mit zwei Knoten. Nachdem in dieser niedrigen Auflösung alle Pixel einlaufende Kanten haben, sind alle vier Pixel der nächstniedrigeren Ebene Kandidaten für den Attraktor und damit für den Algorithmus RemoveBackground.

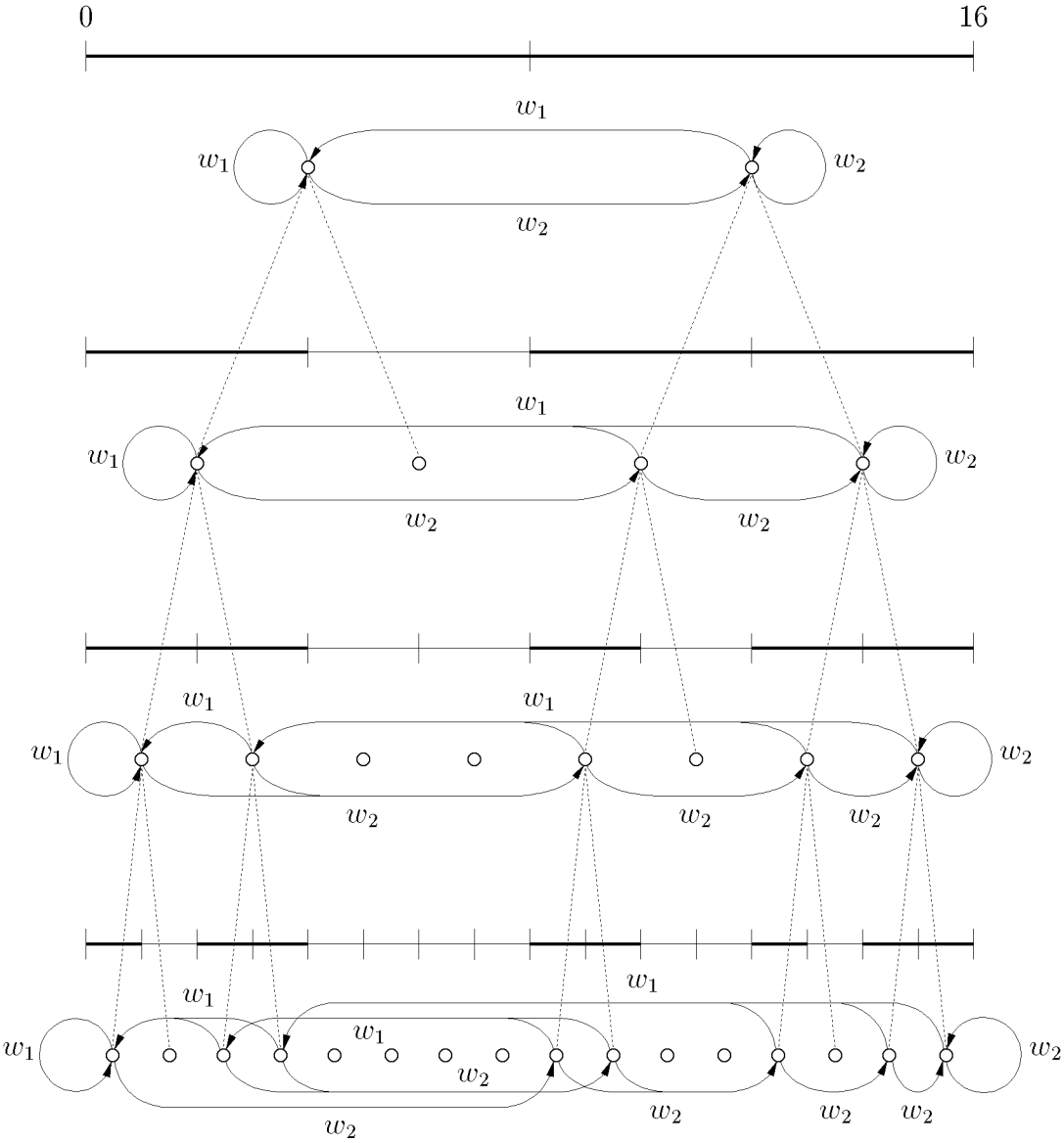


Abbildung 4.4: RemoveBackground–Pyramide für IFS $\{w_1, w_2\}$ in $\mathbb{R}$

Dieses Mal wird ein Pixel entfernt und in der nächstunteren Ebene nur noch die sechs Söhne der verbleibenden drei Pixel betrachtet. Die Vater–Sohn–Zeiger werden in Abb. 4.4 durch strichlierte Linien dargestellt.

4.3.3 Der Algorithmus SetForeground

Eine weitere Möglichkeit einen diskreten Attraktor zu berechnen bietet der hier vorgestellte Algorithmus, der einen nicht notwendigerweise maximalen diskreten Attraktor berechnet. Wenn man die Arbeitsweise von RemoveBackground betrachtet, sieht man, daß dieser Algorithmus, ausgehend von den Pixeln des ganzen Bildschirms, alle Pixel entfernt, die nicht zum maximalen diskreten Attraktor gehören. Der hier beschriebene Algorithmus geht genau den umgekehrten Weg, indem er, ausgehend von einer Startmenge, genau die Pixel zu dieser Startmenge hinzufügt, die notwendig sind, damit ein diskreter Attraktor entsteht. Die Idee ist, ausgehend von einer Startmenge P, die Vereinigung von $\mathbf{w}'(P), \mathbf{w}'^{\circ 2}(P), \ldots$ zu bilden, bis keine neuen Pixel mehr dazukommen. Für bestimmte Startmengen P wird so ein diskreter Attraktor entstehen.

Definition 4.4 Die *transitive Hülle* einer Pixelmenge bezüglich des diskreten Hutchinsonoperators $\mathbf{w}'$ ist wie folgt definiert:

$$[P]^{\mathbf{w}'} := \bigcup_{i \geq 1} \mathbf{w}'^{\circ i}(P) \tag{4.24}$$

Da $\mathbb{P}^2$ ja endlich ist, ist $[P]^{\mathbf{w}'}$ die Vereinigung endlich vieler Mengen.

Satz 4.5 *Sei $P \in \mathcal{H}(\mathbb{P}^2)$ mit $P \subseteq [P]^{\mathbf{w}'}$, dann gilt*

$$[P]^{\mathbf{w}'} = \mathbf{w}'\left([P]^{\mathbf{w}'}\right). \tag{4.25}$$

D.h. $[P]^{\mathbf{w}'}$ ist ein diskreter Attraktor des zu $\mathbf{w}'$ gehörigen IFS.

Beweis: Aus $P \subseteq [P]^{\mathbf{w}'}$ folgt sofort:

$$[P]^{\mathbf{w}'} = P \cup [P]^{\mathbf{w}'}. \tag{4.26}$$

Wendet man $\mathbf{w}'$ auf diese Gleichung an, so erhält man:

$$\mathbf{w}'\left([P]^{\mathbf{w}'}\right) = \mathbf{w}'\left(P \cup [P]^{\mathbf{w}'}\right) \tag{4.27}$$

Mit Def. 4.4 ergibt sich:

$$\mathbf{w}'\left([P]^{\mathbf{w}'}\right) = \mathbf{w}'\left(\bigcup_{i \geq 0} \mathbf{w}'^{\circ i}(P)\right) \tag{4.28}$$

Es ist klar, daß $\mathbf{w}'(I \cup J) = \mathbf{w}'(I) \cup \mathbf{w}'(J)$ für alle $I, J \in \mathcal{H}(\mathbb{P}^2)$ gilt, daher folgt:

$$\mathbf{w}'\left([P]^{\mathbf{w}'}\right) = \bigcup_{i \geq 1} \mathbf{w}'^{\circ i}(P) \tag{4.29}$$

Schließlich erhält man mit Def. 4.4:

$$\mathbf{w}'\left([P]^{\mathbf{w}'}\right) = [P]^{\mathbf{w}'} \tag{4.30}$$

□

SetForeground(IFS $\{w_i \mid i = 1, \ldots, n\}$):

- Die Menge der Startpixel $P \in \mathcal{H}(\mathbb{P}^2)$ muß so initialisiert werden, daß
$$P \subseteq \mathbf{w}'(P)$$
gilt. Z.B. kann P auf die Menge jener Pixel gesetzt werden, die die Fixpunkte der w_i enthalten.
$$P \leftarrow \{p \in \mathbb{P}^2 \mid w_i'(\bar{x}_F) = \bar{x}_F \text{ und } p \in \Delta(\{\bar{x}_F\}) \text{ für } i \in \{1, \ldots, n\}\}$$
- Die Menge der aktiven Pixeln wird mit P initialisiert.
$$A \leftarrow P$$
- $\mathcal{A}_d$ speichert die soweit berechnete transitive Hülle der Ausgangsmenge P bezüglich $\mathbf{w}'$.
$$\mathcal{A}_d \leftarrow \emptyset$$
- S speichert die Knoten, deren direkte Nachfolger bereits berechnet und zu $\mathcal{A}_d$ hinzugefügt wurden.
$$S \leftarrow \emptyset$$
- Die transitive Hülle von P bezüglich $\mathbf{w}'$ wird berechnet.
repeat

  Die direkten Nachfolger der aktiven Pixel werden berechnet.
$$F \leftarrow \mathbf{w}'(A)$$
  Diese werden zu $\mathcal{A}_d$ hinzugefügt.
$$\mathcal{A}_d \leftarrow \mathcal{A}_d \cup F$$
  Die Pixel von A werden als bereits behandelt vermerkt.
$$S \leftarrow S \cup A$$
  Die noch nicht behandelten direkten Nachfolger der aktiven Pixel werden die neuen aktiven Pixeln.
$$A \leftarrow F \setminus S$$

until $(A = \emptyset)$
- $\mathcal{A}_d$ wird als Ergebnis geliefert.

Satz 4.6 *Der Algorithmus SetForeground berechnet einen diskreten Attraktor eines IFS, der nicht notwendigerweise maximal ist.*

Beweis: Es ist leicht zu sehen, daß der Algorithmus ausgehend von P die transitive Hülle von P bezüglich $\mathbf{w}'$ Schritt für Schritt berechnet. P wird vom Algorithmus so gewählt, daß $P \subseteq \mathbf{w}'(P)$ gilt. Damit gilt aber auch trivialerweise $P \subseteq [P]^{\mathbf{w}'}$, und mit Satz 4.5 ist $[P]^{\mathbf{w}'}$ ein diskreter Attraktor. □

Auf regulären sequentiellen Computern braucht SetForeground weniger Speicherplatz und ist schneller als RemoveBackground, da nur jene Pixel transformiert werden, die tatsächlich Teil des berechneten Attraktors sind. Der diskrete Attraktor $\mathcal{A}_d$ eines IFS mit n Funktionen wird von SetForeground in $\mathrm{O}(n\#\mathcal{A}_d)$ sequentiellen Schritten berechnet. Da genau die Pixel des zu berechnenden Attraktors betrachtet werden, ist dieser Algorithmus optimal in dem Sinn, das es keinen anderen Algorithmus geben kann, der asymptotisch weniger Schritte benötigt.

4.3.4 Unterschied zwischen abgetastetem und diskretem Attraktor

Abb. 4.5 zeigt Abbildungen des Farn–Attraktors mit einer Auflösung von 256×256 Pixeln. Die Abbildung oben links wurde mit dem Chaosspiel erzeugt. Das Bild oben rechts wurde

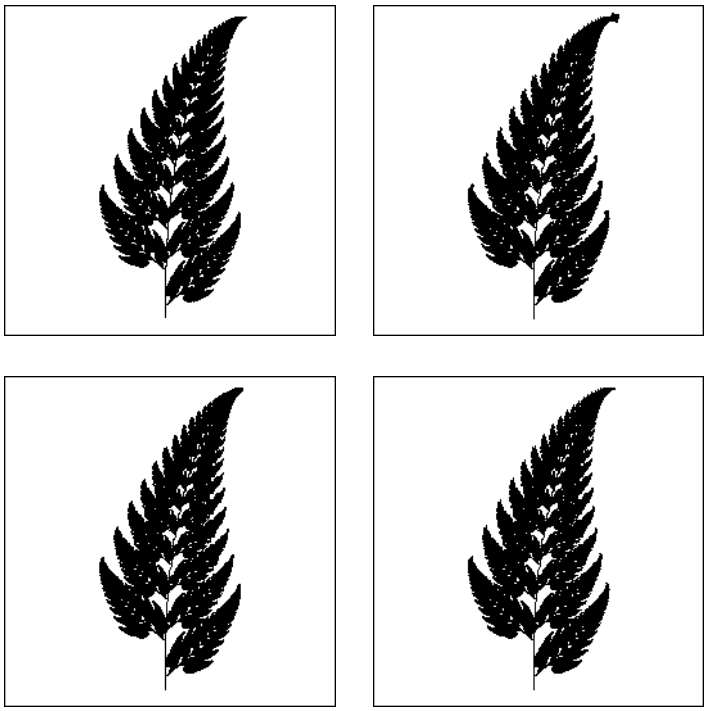


Abbildung 4.5: Abgetastete und diskrete Farn–Attraktoren

mit RemoveBackground, das unten links mit SetForeground generiert, beide mit Mittelpunktsrepräsentation; die Abbildung unten rechts ist das Resultat sowohl von RemoveBackground als auch von SetForeground, wenn die Transformationen mit der im Abschnitt 4.2.1 beschriebenen Erweiterung diskretisiert werden. Der diskrete Attraktor, der mit dieser Methode berechnet wurde, zeigt den kleinsten Unterschied zum abgetasteten Attraktor, was ein überzeugendes

Argument für die Nützlichkeit dieser Methode liefert. Wir wollen im weiteren jedoch die Standardmethode der Diskretisierung untersuchen.

Die Frage ist nun, ob man diese offensichtliche Differenz zwischen tatsächlichen und diskreten Attraktor mathematisch fassen kann. Als quantitatives Maß dieser Differenz bietet sich die Hausdorffmetrik an (Def. 2.7). Da uns der Wert von $d_h(\mathcal{A}, \mathcal{A}_d)$ für den tatsächlichen Attraktor $\mathcal{A} \in \mathcal{H}(\mathbb{R}^2)$ und einen diskreten Attraktor $\mathcal{A}_d \in \mathcal{H}(\mathbb{P}^2)$ interessiert, aber d_h auf Elementen von $\mathcal{H}(\mathbb{R}^2)$ definiert ist, muß der diskrete Attraktor eindeutig mit einem Element aus $\mathcal{H}(\mathbb{R}^2)$ identifiziert werden. Dies erfolgt über die Abbildung ϱ die genau dies für einzelne Pixel liefert (siehe Abschnitt 4.2), und die nun auf $\varrho : \mathcal{H}(\mathbb{P}^2) \to \mathcal{H}(\mathbb{R}^2)$ erweitert wird:

$$\varrho(I) := \bigcup_{p \in I} \overline{\varrho(p)} \qquad I \in \mathcal{H}(\mathbb{P}^2) \tag{4.31}$$

Dabei ist $\overline{\varrho(p)}$ die abgeschlossene Hülle von $\varrho(p)$. Diese Operation ist bei Quadratrepräsentation notwendig, damit $\varrho(I)$ abgeschlossen ist. Nun kann die Definition der Hausdorffmetrik auf $d_h : \mathcal{H}(\mathbb{R}^2) \times \mathcal{H}(\mathbb{P}^2) \to \mathbb{R}_0^+$ erweitert werden:

$$d_h(A, I) := d_h(A, \varrho(I)) \qquad A \in \mathcal{H}(\mathbb{R}^2),\ I \in \mathcal{H}(\mathbb{P}^2) \tag{4.32}$$

Stark hat in [47] folgenden Satz gezeigt:

Satz 4.7 *Sei* $\mathbf{w}$ *der Hutchinsonoperator eines IFS mit Kontraktionsfaktor* λ *und Attraktor* $\mathcal{A}$. *Dann gilt:*

$$d_h(\mathcal{A}, I) > \frac{\mathcal{E}}{1-\lambda} \implies d_h(\mathcal{A}, \mathbf{w}'(I)) < d_h(\mathcal{A}, I) \qquad \forall\, I \in \mathcal{H}(\mathbb{P}^2) \tag{4.33}$$

Der Faktor $\mathcal{E}$ *beeinhaltet die Ungenauigkeit, die durch die Diskretisierung entsteht, ist also von der Pixelrepräsentation abhängig:*

$$\mathcal{E} := \begin{cases} 1/\sqrt{2} & \textit{bei Mittelpunktrepräsentation} \\ \sqrt{2} & \textit{bei Quadratrepräsentation} \end{cases} \tag{4.34}$$

Da der Beweis dieses Satzes in [47] äußerst knapp gehalten ist, wird er hier genauer gebracht:

Beweis: Zuerst wird gezeigt, daß

$$d_h(\mathbf{w}(\varrho(I)), \mathbf{w}'(I)) \leq \mathcal{E} \qquad \forall\, I \in \mathcal{H}(\mathbb{P}^2) \tag{4.35}$$

gilt. Besteht I nur aus einem Pixel, dann kann man dies leicht verifizieren: Die maximale Distanz von $1/\sqrt{2}$ vom Mittelpunkt des Einheitsquadrates zu einem beliebigen Punkt in diesem Quadrat ergibt den Wert bei Mittelpunktrepräsentation. Bei Quadratrepräsentation muß man die Distanz der ganzen Diagonale betrachten, dies ergibt $\sqrt{2}$. Da ganz allgemein für einen vollständigen metrischen Raum (X, d)

$$d_h(A_1 \cup A_2, B_1 \cup B_2) \leq \max(d_h(A_1, B_1), d_h(A_2, B_2)) \tag{4.36}$$

für alle $A_1, A_2, B_1, B_2 \in \mathcal{H}(X)$ gilt, folgt (4.35) durch wiederholte Anwendung dieser Beziehung. Nach Barnsley [5] gilt

$$d_h(\mathbf{w}(A), \mathbf{w}(B)) \leq \lambda d_h(A, B) \qquad \forall\, A, B \in \mathcal{H}(\mathbb{R}^2). \tag{4.37}$$

Aus der Dreiecksungleichung folgt:

$$d_h(\mathcal{A}, \mathbf{w}'(I)) \leq d_h(\mathcal{A}, \mathbf{w}(\varrho(I))) + d_h(\mathbf{w}(\varrho(I)), \mathbf{w}'(I)) \qquad \forall\, I \in \mathcal{H}(\mathbb{P}^2) \tag{4.38}$$

Damit folgt aus (4.37) und (4.35):

$$d_h(\mathcal{A}, \mathbf{w}'(I)) \leq \lambda d_h(\mathcal{A}, I) + \mathcal{E} \qquad \forall\, I \in \mathcal{H}(\mathbb{P}^2) \tag{4.39}$$

Es gilt

$$d_h(\mathcal{A}, I) > \frac{\mathcal{E}}{1-\lambda} \quad \Longrightarrow \quad \lambda d_h(\mathcal{A}, I) + \mathcal{E} < d_h(\mathcal{A}, I) \qquad \forall\, I \in \mathcal{H}(\mathbb{P}^2) \tag{4.40}$$

und mit (4.39)

$$\Longrightarrow \quad d_h(\mathcal{A}, \mathbf{w}'(I)) < d_h(\mathcal{A}, I) \qquad \forall\, I \in \mathcal{H}(\mathbb{P}^2). \tag{4.41}$$

□

Die Anwendung dieses Satzes in [47] ist wie folgt: Dort wird der diskrete Hutchinsonoperator $\mathbf{w}'$ eines IFS so lange auf ein Bild I angewendet, also I auf $\mathbf{w}'(I)$ gesetzt, wie sich das Resultat für das aktuelle I verbessert, also $d_h(\mathcal{A}, \mathbf{w}'(I)) < d_h(\mathcal{A}, I)$ ist. Wie man aus (4.33) sieht, ist dies der Fall, wenn $d_h(\mathcal{A}, I) > \frac{\mathcal{E}}{1-\lambda}$ ist. Also kann man schließen, daß $d_h(\mathcal{A}, I) \leq \frac{\mathcal{E}}{1-\lambda}$ gilt, wenn die Iteration keine Verbesserung mehr ergibt[7]. Die Distanz zwischen dem berechneten I und $\mathcal{A}$ ist also nur vom Kontraktionsfaktor des IFS abhängig. Der nächste Satz wird beweisen, daß dies auch für jeden diskreten Attraktor gilt[8].

Satz 4.8 *Sei* $\mathbf{w}$ *der Hutchinsonoperator eines IFS mit Kontraktionsfaktor* λ *und Attraktor* $\mathcal{A}$*, dann gilt*

$$d_h(\mathcal{A}, \mathcal{A}_d) \leq \frac{\mathcal{E}}{1-\lambda} \tag{4.42}$$

für jeden diskreten Attraktor $\mathcal{A}_d \in \mathcal{H}(\mathbb{P}^2)$.

Beweis: Aus Satz 4.7 folgt sofort:

$$d_h(\mathcal{A}, \mathbf{w}'(I)) \geq d_h(\mathcal{A}, I) \implies d_h(\mathcal{A}, I) \leq \frac{\mathcal{E}}{1-\lambda} \qquad \forall\, I \in \mathcal{H}(\mathbb{P}^2) \tag{4.43}$$

Daraus folgt der Satz, da für jeden diskreten Attraktor $\mathcal{A}_d$

$$d_h(\mathcal{A}, \mathbf{w}'(\mathcal{A}_d)) = d_h(\mathcal{A}, \mathcal{A}_d) \tag{4.44}$$

gilt, wegen $\mathcal{A}_d = \mathbf{w}'(\mathcal{A}_d)$. □

[7] Also wenn $d_h(\mathcal{A}, \mathbf{w}'(I)) \geq d_h(\mathcal{A}, I)$ ist, wobei in [47] die Iteration so lange fortgesetzt wurde, bis zwischen I und $\mathbf{w}'(I)$ kein Unterschied mehr bestand.

[8] Die von Stark angewandte iterative Methode hat kein klar definiertes Abbruchkriterium und muß deshalb nicht notwendigerweise einen diskreten Attraktor liefern.

4.4 Diskretes invariantes Maß

Gegeben sei ein probabilistisches IFS $\{w_i \mid i = 1, \ldots, n\}$ mit Wahrscheinlichkeiten p_i, $i = 1, \ldots, n$. Hat man einen diskreten Attraktor $\mathcal{A}_d$ für dieses IFS berechnet, so kann man auch ein diskretes invariantes Maß dafür berechnen.

Ein diskretes Maß $\bar{\mu} \in \mathcal{P}(\mathbb{P}^2)$ ist durch die Spezifikation der Maßwerte für alle Pixel in $\mathbb{P}^2$ definiert. Es ergibt sich folgender Vektor:

$$\bar{\mu} := (\bar{\mu}_p)_{p \in \mathbb{P}^2} \qquad \bar{\mu}_p \in \mathbb{R}_0^+ \tag{4.45}$$

Der diskrete Markovoperator $\mathbf{M}'$ eines IFS ist:

$$\mathbf{M}'(\bar{\mu}) := \bar{\mu}\Gamma \tag{4.46}$$

wobei die Matrix

$$\Gamma := (\gamma_{pq})_{p,q \in \mathbb{P}^2} \tag{4.47}$$

die Wirkung der Maßwerte beschreibt, die vom Markovoperator von p nach q gesendet werden:

$$\gamma_{pq} := \sum_{i=1}^{n} p_i \pi_{pqi} \qquad p, q \in \mathbb{P}^2 \tag{4.48}$$

p_i ist die Wahrscheinlichkeit für w_i. π_{pqi} zeigt an, welcher Anteil von p unter w_i auf die Fläche von q abgebildet wird:

1. Werden die Pixel durch ihre Mittelpunkte repräsentiert, so wird das gesamte Maß von p zu demjenigen q gesendet, auf dessen Mittelpunkt der Mittelpunkt von p unter w_i abgebildet wird[9]:

$$\pi^+_{pqi} := \begin{cases} 1 & \text{wenn } w_i'(p) = \{q\} \\ 0 & \text{sonst} \end{cases} \qquad p, q \in \mathbb{P}^2,\ i = 1, \ldots, n \tag{4.49}$$

2. Werden die Pixel durch quadratische Teilmengen von $\mathbb{R}^2$ repräsentiert, wird das Maß von p unter denjenigen Pixeln verteilt, auf die p unter w_i abgebildet wird. Die Verteilung erfolgt im Verhältnis des Flächenanteils von p, der auf die jeweiligen Pixel fällt:

$$\pi^\square_{pqi} := \frac{\text{area}((w_i \circ \varrho^\square)(p) \cap \varrho^\square(q))}{\text{area}((w_i \circ \varrho^\square)(p))} \qquad p, q \in \mathbb{P}^2,\ i = 1, \ldots, n \tag{4.50}$$

   area(A) bezeichnet den Flächeninhalt (genauer das Lebesgue–Maß) einer Menge $A \in \mathcal{B}(\mathbb{R}^2)$.

[9] Es wird Mittelpunktsrepräsentation verwendet, daher gilt: $w_i'(p) = (\Delta^+ \circ w_i \circ \varrho^+)(p)$

Ein *diskretes invariantes Maß* $\bar{\mu}_F$ ist ein diskreter Maßvektor, für den

$$\bar{\mu}_F = \mathbf{M}'(\bar{\mu}_F) \tag{4.51}$$

gilt. Mit der Definition von $\mathbf{M}'$ gilt also, daß jede normalisierte Lösung[10] des homogenen Gleichungssystems

$$\bar{\mu}(\Gamma - I) = \bar{0} \tag{4.52}$$

bei der die Maßwerte $\bar{\mu}_p$ der Pixel $p \notin \mathcal{A}_d$ gleich 0 sind, ein diskretes invariantes Maß ist. I ist die Einheitsmatrix.

4.4.1 Unterschied zwischen abgetastetem und diskretem invarianten Maß

Für den Unterschied zwischen abgetastetem und diskretem invarianten Maß kann ebenso eine Schranke angegeben werden (bezüglich der Hutchinsonmetrik), wie für den abgetasteten und den diskreten Attraktor (Abschnitt 4.3.4). Für den Beweis sei auf [47] verwiesen.

Zum Messen des Unterschiedes muß die Definition der Hutchinsonmetrik auf $d_H : \mathcal{P}(\mathbb{R}^2) \times \mathcal{P}(\mathbb{P}^2) \to \mathbb{R}_0^+$ erweitert werden:

$$d_H(\mu, \bar{\mu}) := d_H\left(\mu, \sum_{p \in \mathbb{P}^2} \bar{\mu}_p \delta_p\right) \qquad \mu \in \mathcal{P}(\mathbb{R}^2),\ \bar{\mu} \in \mathcal{P}(\mathbb{P}^2) \tag{4.53}$$

Wobei δ_p das Dirac–Maß[11] im Punkt $\binom{p_x+0.5}{p_y+0.5}$ ist.

Satz 4.9 *Sei* $\mathbf{M}$ *der Hutchinsonoperator eines probabilistischen IFS mit Kontraktionsfaktor* λ *und invarianten Maß* $\mu_F \in \mathcal{P}(\mathbb{R}^2)$. *Dann gilt*

$$d_H(\mu_F, \bar{\mu}) > \frac{\mathcal{F}}{1-\lambda} \implies d_H(\mu_F, \mathbf{M}'(\bar{\mu})) < d_H(\mu_F, \bar{\mu}) \tag{4.54}$$

für alle diskreten Maße $\bar{\mu} \in \mathcal{P}(\mathbb{P}^2)$. *Der Faktor* $\mathcal{F}$ *beeinhaltet die Ungenauigkeit, die durch die Diskretisierung entsteht, ist also von der Pixelrepräsentation abhängig:*

$$\mathcal{F} := \begin{cases} 1/\sqrt{2} & \textit{bei Mittelpunktrepräsentation} \\ (\sqrt{2} + \ln 2)/3 & \textit{bei Quadratrepräsentation} \end{cases} \tag{4.55}$$

Satz 4.10 *Sei* $\mathbf{M}$ *der Hutchinsonoperator eines probabilistischen IFS mit Kontraktionsfaktor* λ *und invarianten Maß* μ_F. *Dann gilt:*

$$d_H(\mu_F, \bar{\mu}_F) \leq \frac{\mathcal{F}}{1-\lambda} \tag{4.56}$$

für jedes diskrete invariante Maß $\bar{\mu}_F \in \mathcal{P}(\mathbb{P}^2)$.

---

[10] Also eine Lösung $\bar{\mu}$, wo $\sum_{p \in \mathbb{P}^2} \bar{\mu}_p = 1$ gilt mit $\bar{\mu}_p \in \mathbb{R}_0^+$ für alle $p \in \mathbb{P}^2$.

[11] Das Dirac-Maß $\delta_{\bar{x}}$ im Punkt $\bar{x} \in \mathbb{R}^2$, ist definiert dadurch, daß $\delta_{\bar{x}}(A) = 1$ genau dann ist, wenn $\bar{x} \in A$ und $\delta_{\bar{x}}(A) = 0$ genau dann, wenn $\bar{x} \notin A$, für alle $A \in \mathcal{B}(\mathbb{R}^2)$.

Beweis: Der Beweis ist analog zu Beweis von Satz 4.8 unter Verwendung von Satz 4.9. □

Hier ergibt sich, daß die Berechnung des diskreten Maßes mit Quadratrepräsentation ein wenig besser ($\mathcal{F} \approx 0.7024$) ist als bei Mittelpunktsrepräsentation ($\mathcal{F} \approx 0.7071$). Da der Unterschied jedoch geringfügig ist, und die Algorithmen bei Quadratrepräsentation ungleich komplexer sind, ist es besser die Mittelpunktsprepräsentation zur Berechnung des diskreten Attraktors und des diskreten invarianten Maßes zu verwenden.

4.5 Verbesserung der diskreten Approximation

Die Distanz zwischen abgestasteten und diskreten Attraktor bzw. invarianten Maß ist also umso größer, je größer der Kontraktionsfakter des IFS ist (also umso weniger kontraktiv es ist). Wenn λ nahe bei 1 ist, kann diese Distanz sehr groß werden, da aber nach [5] jedes IFS durch ein äquivalentes mit geringeren Kontraktionsfaktor ersetzt werden kann, ist dies in Wirklichkeit kein Nachteil: Das IFS $\{w_i \mid i = 1, \ldots, n\}$ mit Wahrscheinlichkeiten $p_1, \ldots, p_n$ und Kontraktionsfaktor λ und jedes IFS $\{w_{\bar{\imath}_1} \circ w_{\bar{\imath}_2} \circ \cdots \circ w_{\bar{\imath}_N} \mid \bar{\imath} \in \{1, 2, \ldots, n\}^N\}$ mit $N \in \mathbb{N}^+$ besitzen denselben Attraktor. Wenn man den zusammengesetzten Funktionen das Produkt der Wahrscheinlichkeiten ihrer Teilfunktionen zuordnet ($p_{\bar{\imath}_1} p_{\bar{\imath}_2} \cdots p_{\bar{\imath}_N}$), ergibt sich auch dasselbe invariante Maß. Der Kontraktionsfaktor des so aufgebauten IFS ist dann λ^N.

Der Unterschied zwischen diskreten und abgetasteten Attraktor wird nach Satz 4.8 maximal ein Pixel betragen, wenn $\lambda < 1 - \mathcal{E}$ ist, also für $\lambda \lessapprox 0.3$ bei Mittelpunktsrepräsentation; bei Quadratrepräsentation kann dieser Distanzwert nicht erreicht werden, da der Fehler $\mathcal{E}$ zu groß ist.

Kapitel 5

Zusammenfassung und Schluß

In dieser Arbeit wurde eine Einführung in die Problemstellung der Bildcodierung gegeben. Es wurde die mathematische Theorie Iterierter Funktionensysteme vorgestellt. Affine Transformationen auf euklidischen Räumen wurden untersucht. Eine leicht verständliche Darstellungsform affiner Transformationen auf $\mathbb{R}^2$ wurde präsentiert; Vorschriften zur Umwandlung in diese Darstellungsform und zur Berechnung des Kontraktionsfaktors und des Fixpunktes solcher Transformationen wurden angegeben. Die Berechnung des Attraktors und des invarianten Maßes wurde von $\mathbb{R}^2$ in einen diskreten Pixelraum durch eine Diskretisierung affiner Transformationen verlagert. Dadurch wurden diskrete Attraktoren und diskrete invariante Maße definiert. Zwei effiziente Algorithmen zur Berechnung des diskreten Attraktors wurden angegeben. Einer dieser Algorithmen wurde mit Bildpyramiden verbessert; eine Brücke zur Pyramidencodierung wurde so geschlagen. Weiters wurde ein effizientes Verfahren zur Berechnung eines diskreten invarianten Maßes angegeben. Die durch die Diskretisierung entstehenden Abweichungen des Attraktors und des invarianten Maßes von ihren diskreten Entsprechungen wurden nach oben beschränkt. Aus diesen Schranken wurde abgeleitet, daß die Mittelpunktsrepräsentation der Quadratrepräsentation zur Diskretisierung affiner Transformationen vorzuziehen ist. Es wurde gezeigt, daß trotz der erwähnten Abweichungen eine für die Bildcodierung ausreichende Genauigkeit immer erreicht werden kann.

Damit IFS für eine effiziente Bildcodierung verwendet werden können, ist es notwendig, das inverse Problem zu lösen. Die hier präsentierten Konzepte stellen jedoch einen Schritt zu einer möglichen Lösung dieses Problems dar, wie einige Publikationen zu diesem Thema bestätigen [29, 28, 48]. Weitere Möglichkeiten für zukünftige Erweiterungen dieser Arbeit stellen die Codierung von Grauwert- bzw. Farbbildern als Attraktoren von IFS in $\mathbb{R}^3$ bzw. $\mathbb{R}^5$, eine mathematische Analyse der in Abschnitt 4.2.1 angegebenen erweiterten Diskretisierung sowie der Ausbau der Diskretisierung auf erweiterte IFS Modelle [7, 45, 30] dar.

Literaturverzeichnis

[1] D. M. Abrahamson. An adaptive dependency source model for data compression. *Communications of the ACM*, 32(1):77–83, 1989.

[2] K. L. Anderson and W. B. Pennebaker. JPEG compression of grayscale and color images. Research report, IBM, IBM Research Division, T. J. Watson Research Center, Yorktown Heights, NY 10598, 1991.

[3] P. H. Ang, P. A. Ruetz, and D. Auld. Video compression makes big gains. *IEEE Spectrum*, pages 16–19, Oct. 1991.

[4] The Association for Computing Machinery. *SIGGRAPH '91 Course Notes—Fractal Modeling in 3-D Computer Graphics and Imaging*, volume C14, 1991.

[5] M. F. Barnsley. *Fractals Everywhere*. Academic Press, 1988.

[6] M. F. Barnsley and S. Demko. Iterated function systems and the global construction of fractals. *Proceedings of the Royal Society of London Ser. A*, 399:243–275, 1985.

[7] M. F. Barnsley, J. H. Elton, and D. P. Hardin. Recurrent iterated function systems. *Constructive Approximation*, 5(1):3, 1989.

[8] M. F. Barnsley, A. Jacquin, F. Malassenet, L. Reuter, and A. D. Sloan. Harnessing chaos for image synthesis. *Computer Graphics*, 22(4):131–140, Aug. 1988.

[9] M. F. Barnsley and A. D. Sloan. A better way to compress images. *BYTE*, pages 215–223, Jan. 1988.

[10] L. Berger, J.-P. Mariot, and C. Launay. A new formulation for fast image coding using quadtree representation. *Pattern Recognition Letters*, 13:425–432, June 1992.

[11] P. J. Burt and E. H. Adelson. The Laplacian pyramid as a compact image code. *IEEE Transactions on Communications*, COM-31(4):532–540, Apr. 1983.

[12] S. Demko, L. Hodges, and B. Naylor. Construction of fractals object with iterated function systems. *Computer Graphics*, 19(3):271–278, 1985.

[13] K. Falconer. *Fractal Geometry, Mathematical Foundations and Applications.* Wiley & Sons, 1990.

[14] Y. Fisher. Fractal image compression. In *Fractals: from folk art to hyperreality. SIGGRAPH 1992 Course Notes*, Chicago, 1992. The Association for Computing Machinery.

[15] E. Gröller. Fractals and solid modeling. *EUROGRAPHICS '92*, 11(3):C–415–C–424, 1992.

[16] A. Habibi and P. A. Wintz. Image coding by linear transformation and block quantization. *IEEE Transactions on Communications*, COM-19(1):50–63, Feb. 1971.

[17] J. C. Hart. Linear fractals in 3-D computer graphics. In *SIGGRAPH '91 Course Notes* [4], pages 145–163.

[18] J. C. Hart and T. A. DeFanti. Efficient antialiased rendering of 3-d linear fractals. *Computer Graphics*, 25(4):91–100, July 1991.

[19] H. Heuser. *Lehrbuch der Analysis.* Teubner Verlag, 1988.

[20] A. N. Horn. IFSs and interactive image synthesis. *Computer Graphics Forum*, 9:127–137, 1990.

[21] C.-H. Hsieh, P.-C. Lu, and J.-C. Chang. Fast codebook generation algorithm for vector quantization of images. *Pattern Recognition Letters*, 12:605–609, Oct. 1991.

[22] D. A. Huffman. A method for the construction of minimum-redundancy codes. *Proc. Inst. Electr. Radio Eng.*, 40(9):1098–1101, Sept. 1952.

[23] J. E. Hutchinson. Fractals and self similarity. *Indiana University Mathematics Journal*, 30(5):713–747, 1981.

[24] A. E. Jacquin. Image coding based on a fractal theory of iterated contractive image transformations. *IEEE Transactions on Image Processing*, IP-1(1):18–30, 1992.

[25] A. K. Jain. *Fundamentals of Digital Image Processing.* Prentice Hall, 1989.

[26] W. G. Kropatsch. Kurvenrepräsentation in Pyramiden. In W. G. Kropatsch and P. Mandl, editors, *Mustererkennung '86*, volume 36, pages 16–51. Oldenbourg, 1986.

[27] W. G. Kropatsch. Image pyramids and curves – an overview. Technical Report 2, Dept. for Pattern Recognition and Image Processing, Technical University of Vienna, Treitlstr. 3/183/2, A-1040 Vienna, 1991.

[28] W. G. Kropatsch, M. A. Neuhauser, and I. J. Leitgeb. Iterated function systems – a direct discrete approach with pyramids. In H. Bischof and W. G. Kropatsch, editors, *Pattern Recognition 1992*, volume 62, pages 108–118. ÖAGM, Oldenburg Verlag, May 1992. Proceedings of the 16th ÖAGM-Meeting.

[29] W. G. Kropatsch, M. A. Neuhauser, I. J. Leitgeb, and H. Bischof. Combining pyramidal and fractal image coding. In *Proceedings of the 11th ICPR*, volume III, pages 61–64. IAPR, IEEE Computer Society Press, Aug. 1992.

[30] J. W. Layman and T. E. Womack. Linear markov iterated function systems. *Computers & Graphics*, 14(2):343–353, 1990.

[31] J. Lévy-Véhel and A. Gagalowicz. Shape approximation by a fractal model. In G. Maréchal, editor, *Proceedings of the EUROGRAPHICS 1987*, pages 159–179. Eurographics Association, Elsevier Science Publishers, Aug. 1987.

[32] J. Lévy-Véhel and A. Gagalowicz. Fractal approximation of 2-D object. In D. A. Duce and P. Jancene, editors, *Proceedings of the EUROGRAPHICS 1988*, pages 297–311. Eurographics Association, Elsevier Science Publishers, Sept. 1988.

[33] R. Libeskind-Hadas and P. Maragos. Application of iterated function systems and skeletonization to synthesis of fractal images. In *SPIE*, volume 845 of *Visual Communications and Image Processing II*, pages 276–284, 1987.

[34] S. G. Mallat. A theory for multiresolution signal decomposition: the wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-11(7):674–693, July 1989.

[35] B. Mandelbrot. *The Fractal Geometry of Nature*. W. H. Freeman and Co., San Francisco, 1982.

[36] H. Mayer and W. G. Kropatsch. Kompakte Bildkodierung mit der $3 \times 3/2$ Pyramide. In A. Pinz, editor, *Wissensbasierte Mustererkennung*, volume 49, pages 195–210. Oldenbourg, 1989.

[37] H. Mayer and W. G. Kropatsch. Progressive Bildübertragung mit der $3 \times 3/2$ Pyramide. In H. Burkhardt, K. H. Höhne, and B. Neumann, editors, *Informatik Fachberichte 219: Mustererkennung 1989*, pages 160–167. Springer Verlag, Hamburg, 1989.

[38] H. F. Mayer. Progressive Übertragung und komprimierte Speicherung von digitalen Bildern. Diplomarbeit, Technical University of Graz, Austria, Oct. 1989.

[39] D. S. Mazel and M. H. Hayes. Using iterated function systems to model discrete sequences. *IEEE Transactions on Signal Processing*, SP-40(7):1724–1734, 1992.

[40] H. Mitsch. *Lineare Algebra und Geometrie*. Prugg Verlag, 1978.

[41] J. B. O'Neal, Jr. Differential pulse-code modulation (DPCM) with entropy coding. *IEEE Transactions on Information Theory*, IT-21(2):169–174, Mar. 1976.

[42] H.-O. Peitgen, H. Jürgens, and D. Saupe. *Fractals for the Classroom*. Springer Verlag, 1991.

[43] A. Pentland and B. Horowitz. A practical approach to fractal-based image compression. In *Data Compression Conference*, pages 176–185. IEEE, IEEE Computer Society Press, 1991.

[44] W. K. Pratt, W. H. Chen, and L. R. Welch. Slant transform image coding. *IEEE Transactions on Communications*, COM-22(8):1075–1093, Aug. 1974.

[45] P. Prusinkiewicz and M. Hammel. Automata, languages, and iterated function systems. In *SIGGRAPH '91 Course Notes* [4], pages 115–143.

[46] A. Rosenfeld and A. C. Kak. *Digital Picture Processing.* Academic Press, Inc., second edition, 1982.

[47] J. Stark. Iterated function systems as neural networks. *Neural Networks*, 4:679–690, 1991.

[48] J. Stark. A neural network to compute the Hutchinson metric in fractal image processing. *IEEE Transactions on Neural Networks*, NN-2(1):156–158, Jan. 1991.

[49] E. R. Vrscay. Moment and collage methods for the inverse problem of fractal construction with iterated function systems. In *SIGGRAPH '91 Course Notes* [4], pages 271–289.

[50] T. A. Welch. A technique for high-performance data compression. *Computer*, pages 8–19, June 1984.

[51] I. H. Witten, R. M. Neal, and J. G. Cleary. Arithmetic coding for data compression. *Communications of the ACM*, 30(6):520–540, 1987.

[52] L. H. Zetterberg, S. Ericsson, and C. Coutrier. DPCM picture coding with two-dimensional control of adaptive quantization. *IEEE Transactions on Communications*, COM-32(4):457–462, Apr. 1984.

[53] J. Ziv and A. Lempel. Compression of individual sequences via variable-rate coding. *IEEE Transactions on Information Theory*, IT-24(5):530–536, Sept. 1978.