Zeitschrift: Bulletin des Schweizerischen Elektrotechnischen Vereins, des Verbandes Schweizerischer Elektrizitätsunternehmen = Bulletin de l'Association suisse des électriciens, de l'Association des entreprises électriques suisses

Herausgeber: Schweizerischer Elektrotechnischer Verein ; Verband Schweizerischer Elektrizitätsunternehmen

Band: 82 (1991)

Heft: 21

Artikel: Wie Bilder aus sich selbst entstehen : fraktale Bilddatenkompression

Autor: Meneghini, Reto / Stutz, Peter

DOI: https://doi.org/10.5169/seals-903028

Nutzungsbedingungen

Die ETH-Bibliothek ist die Anbieterin der digitalisierten Zeitschriften auf E-Periodica. Sie besitzt keine Urheberrechte an den Zeitschriften und ist nicht verantwortlich für deren Inhalte. Die Rechte liegen in der Regel bei den Herausgebern beziehungsweise den externen Rechteinhabern. Das Veröffentlichen von Bildern in Print- und Online-Publikationen sowie auf Social Media-Kanälen oder Webseiten ist nur mit vorheriger Genehmigung der Rechteinhaber erlaubt. Mehr erfahren

Conditions d'utilisation

L'ETH Library est le fournisseur des revues numérisées. Elle ne détient aucun droit d'auteur sur les revues et n'est pas responsable de leur contenu. En règle générale, les droits sont détenus par les éditeurs ou les détenteurs de droits externes. La reproduction d'images dans des publications imprimées ou en ligne ainsi que sur des canaux de médias sociaux ou des sites web n'est autorisée qu'avec l'accord préalable des détenteurs des droits. En savoir plus

Terms of use

The ETH Library is the provider of the digitised journals. It does not own any copyrights to the journals and is not responsible for their content. The rights usually lie with the publishers or the external rights holders. Publishing images in print and online publications, as well as on social media channels or websites, is only permitted with the prior consent of the rights holders. Find out more

Download PDF: 06.07.2025

ETH-Bibliothek Zürich, E-Periodica, https://www.e-periodica.ch

Wie Bilder aus sich selbst entstehen

Fraktale Bilddatenkompression

Reto Meneghini und Peter Stutz

In vielen Bereichen der elektronischen Datenverarbeitung ist eine rasche Zunahme der zu verarbeitenden Daten festzustellen. Damit gewinnen Verfahren zur Datenkompression zunehmend an Bedeutung. Im Bereich der Bildverarbeitung bietet die Theorie der Fraktale interessante Möglichkeiten, solche Verfahren zu entwickeln.

Dans de nombreux domaines du traitement électronique de l'information on constate une croissance rapide des quantités de données à traiter. Cela fait que les méthodes de compression des données gagnes en importance. Dans le secteur du traitement de l'image, la théorie des fractales offre d'intéressantes possibilités de développer de telles méthodes.

In den letzten Jahren wurden auf dem Gebiet der elektronischen Datenverarbeitung grosse Fortschritte erzielt. In der Wissenschaft, der Technik und im täglichen Leben gibt es kaum mehr Bereiche, die nicht entscheidend durch den Einsatz von moderner Computertechnik geprägt sind. Im Zuge dieser Entwicklung werden die zur Verarbeitung anfallenden Datenmengen zusehends grösser und grösser. Damit wächst auch der Bedarf an Methoden, die den Platzbedarf für eine bestimmte Menge Daten – sei es zur Übertragung oder Speicherung – zu reduzieren erlauben. Insbesondere in der Bildverarbeitung sind solche Komprimierungsverfahren (Bild 1) von grossem Interesse, weil dort Mengen von Daten entstehen, die beispielsweise auch mit modernsten Personal Computern nur schlecht, oft aber auch gar nicht zu bewältigen sind.

Fraktale und Datenkompression

Fraktale sind, grob gesagt, geometrische Gebilde, die unendlich oft vergrössert werden können und dabei immer neue, detailliertere Strukturen zeigen, die im wesentlichen gleich aussehen wie das ursprüngliche Gebilde [1]. Diese zentrale Eigenschaft heisst Selbstähnlichkeit. Ein Beispiel für ein Fraktal ist das sogenannte Sierpinski-Dreieck in Bild 2. Jedes der drei Teildreiecke stellt eine vollständige und verkleinerte Kopie der ganzen Figur dar. Wenn man das Sierpinski-Dreieck mehr und mehr vergrössert, werden immer neue Details sichtbar, in welchen sich unaufhörlich die ursprüngliche Form repetiert.

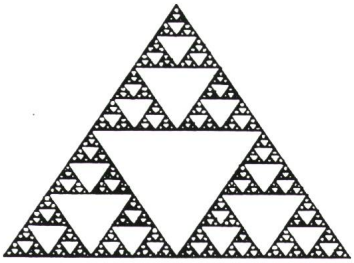
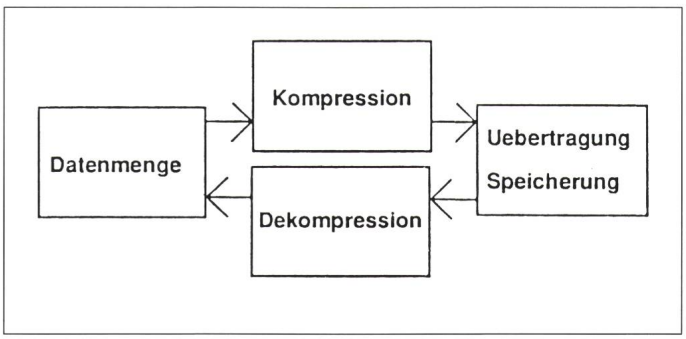


Bild 2 Das Sierpinski-Dreieck

Fraktale Gebilde treten aber auch überall in der Natur auf. So sieht zum Beispiel das Blatt einer Farnpflanze im wesentlichen gleich aus wie die ganze Pflanze, jeder Teil des Blatts gleich wie das ganze Blatt usw. (Bild 3). Auf dieser Erkenntnis beruht die fraktale Datenkompression. Soll nämlich der Farn abgespeichert werden, braucht dies nicht Punkt für Punkt zu geschehen, sondern man kann sich mit

Bild 1
Das Prinzip der Datenkompression



Adresse der Autoren
Reto Meneghini, Dipl. El.-Ing. ETH, und *Peter Stutz*, Dipl.El.-Ing. ETH, Alcatel STR AG, 8055 Zürich.

der Grundstruktur begnügen, die – wie bereits erwähnt – sowohl die Form des ganzen Farns als auch die der einzelnen Blätter bestimmt. Mit dieser Methode der Speicherung wird gegenüber einer konventionellen Methode eine Datenkompression erreicht.

Es bleibt nur noch zu klären, wie die Grundstruktur des Farns mathematisch erfasst werden kann. Eine Möglichkeit hierfür sind die sogenannten IFS-Codes, die im nächsten Abschnitt besprochen werden.

IFS-Codes

Zum besseren Verständnis sollen die Grundlagen der IFS-Codes[1] anhand eines Beispiels erläutert werden. Man betrachte dazu das Bild 4. Jeder der vier markierten Teile des Farns stellt eine verkleinerte Kopie des ganzen Farns dar und kann damit durch eine kontraktive affine Abbildung (Kasten I) aus diesem gewonnen werden. (Der Stengel entspricht der zu einer Geraden transformierten Kopie des Farns [Kasten II]). Mit anderen Worten: jeder Teil des Farns kann als affine Abbildung des ganzen Farns dargestellt werden. Alle vier Abbildungen zusammen bilden den sogenannten *IFS-Code* des Farns. Die Abkürzung IFS steht hierbei für Iterated Function Systems. Der IFS-Code des Farns enthält bereits eine vollständige Beschreibung des Farns. Man kann es aus diesen vier einfachen linearen Abbildungen ohne Informationsverlust zurückgewinnen. Dies lässt sich durch ein Gedankenexperiment plausibel machen:

1. Man markiere sich einen beliebigen Punkt auf dem Farn.
2. Man wähle zufällig eine der vier affinen Abbildungen aus.
3. Man bilde den markierten Punkt mit der ausgewählten Abbildung ab. Da jede der vier möglichen Abbildungen den Farn auf einen Teil von sich selbst abbildet, ist klar, dass der abgebildete Punkt wieder auf dem Farn liegen muss.
4. Man markiere nun den neuen Punkt und wiederhole beliebig oft die Schritte 2–4.

Auf diese Weise entstehen laufend neue Punkte, bis schlussendlich jeder Punkt des Farns markiert ist. Das Bild 5 zeigt das entstehende Farn nach dem Zeichnen der ersten paar hundert Punkte.

Das eben besprochene Dekodier-Verfahren heisst seinem Wesen entsprechend *Random-Walk*; das entstehende Bild wird *Attraktor* genannt. Man beachte besonders folgende zwei Punkte:

1. Jede beliebige (aber genügend lange) Zufallssequenz von ausgeführten Abbildungen überdeckt den Farn vollständig mit markierten Punkten. Ist man allerdings an einem regelmässigen Bildaufbau interessiert, muss berücksichtigt werden, dass die vier Teilbereiche des Farns verschieden gross sind und damit unterschiedlich viele Bildpunkte umfassen. Deshalb muss etwa die Abbildung auf dem oberen Teil des Farns statistisch häufiger als die Abbildungen auf die Blätter ausgeführt werden.
2. Das oben beschriebene Verfahren benutzte zur Dekodierung einen Anfangspunkt im Farn. Der korrekte Attraktor entsteht aber auch mit einem beliebigen Anfangspunkt, wenn man dabei die ersten paar entstehenden Punkte nicht zeichnet. Die vom Random-Walk erzeugte Folge von Punkten strebt dank der sogenannten durchschnittlichen Kontraktionsbedingung (s. Kasten I) immer auf den Attraktor zu und bleibt dann in ihm gefangen.

Bild 3 Farnpflanze

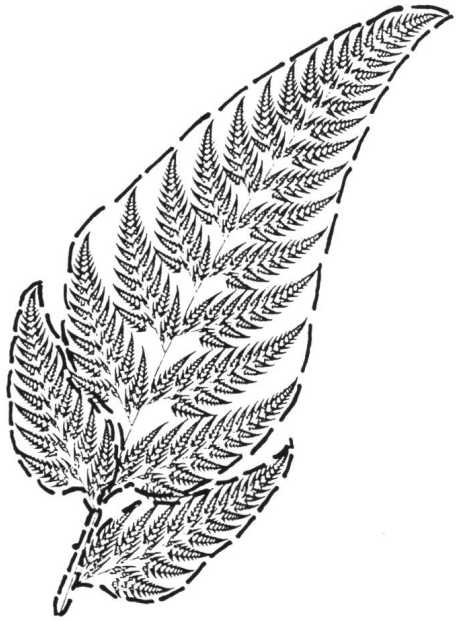


Bild 4 Unterteiltes Farn
Jeder Teil des Farns kann als affine Abbildung des ganzen Farns dargestellt werden

Affine Abbildungen: sind die allgemeinste Form von linearen Abbildungen. Sie zeichnen sich durch Geradentreue aus, was bedeutet, dass bei der Anwendung einer solchen Abbildung auf eine Gerade wieder eine Gerade entsteht. In die Klasse der affinen Abbildungen gehören damit alle linearen Abbildungen wie etwa Rotationen, Streckungen und Translationen. Die allgemeine Form einer affinen Abbildung in der Ebene lautet:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} * \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$$

Kontraktive affine Abbildung: Eine affine Abbildung heisst kontraktiv, wenn sie verkleinernden Charakter hat, das heisst wenn eine Figur durch die Anwendung der Abbildung in eine kleinere Bildfigur transformiert wird.

Durchschnittliche Kontraktionsbedingung: Wendet man auf eine Ausgangsfigur in zufälliger Reihenfolge beliebig viele affine Abbildungen $\{w_i$ aus einer Menge $w_1 \ldots w_n\}$ an, so heissen diese genau dann durchschnittlich kontraktiv, wenn die am Ende resultierende Bildfigur – bei vorgegebener Häufigkeit der einzelnen Abbildungen w_i – kleiner ist, als die Ausgangsfigur.

Kasten I

[1] Die IFS-Codes sind bereits früher im Bulletin SEV/VSE beschrieben worden [2].

Bis hierher wurde beschrieben, was ein IFS-Code ist und wie er dekodiert werden kann. Wie bereits früher erwähnt, erreicht man mit der IFS-Code-Darstellung eines Farns im Vergleich zu einer punktweisen Abspeicherung eine bedeutende Datenkompression. Damit stellen sich die folgenden drei Fragen:

1. Existieren auch zu anderen Bildern IFS-Codes?
2. Wenn ja, wie findet man sie?
3. Erreicht man mit diesen Codes für alle Bilder eine Datenkompression?

Man kann zeigen, dass prinzipiell zu jedem beliebigen Bild ein IFS-Code existiert. Man findet diesen dadurch, dass man die Ausgangsfigur mit verkleinerten Abbildungen von sich selbst vollständig (!) überdeckt und die dazu nötigen Abbildungskoeffizienten berechnet, die dann den gesuchten IFS-Code bilden. Dieses Verfahren wird in der Literatur als *Collage-Theorem* bezeichnet. Im schlechtesten Fall braucht man für ein Bild mit n Bildpunkten n affine Abbildungen, um es vollständig mit verkleinerten Kopien von sich selbst zu überdekken. Man kann nämlich die Ausgangsfigur mittels n affinen Abbildungen, die jeweils das ganze Bild zu einem Punkt transformieren, auf jeden ihrer n Bildpunkte abbilden. Diese Erkenntnis beantwortet die Frage 3. Muss für jeden Bildpunkt eine Abbildung gespeichert werden, so kann man natürlich nicht mehr von Datenkompression sprechen.

Bild 5 Dekodierung des Farns

IFS-Code: Jede Menge $\{w_1 \ldots w_n\}$ von affinen Abbildungen, welche die durchschnittliche Kontraktionsbedingung erfüllen, bildet einen IFS-Code (Iterated Function Systems). Jeder derartige Code beschreibt ein fraktales Bild in der Ebene.
Random-Walk: Dekodierungs-Prozedur für einen IFS-Code, bei der man, in einem beliebigen Startpunkt beginnend, Abbildungen aus dem Code in zufälliger Reihenfolge wiederholt anwendet.
Attraktor: Bei der Dekodierung eines IFS-Codes entstehendes Bild.
Collage-Theorem: Zu einem beliebigen Bild in der Ebene kann ein IFS-Code gefunden werden, indem man es mit affinen Abbildungen auf sich selbst vollständig überdeckt. Alle dazu nötigen Abbildungen zusammen bilden den gesuchten IFS-Code.

Kasten II

IFS-Codes sind, wie man sich leicht überlegen kann, gut geeignet für Bilder, die eine hohe Selbstähnlichkeit besitzen. Fehlt diese jedoch, braucht man sehr schnell eine grosse Anzahl von Abbildungen, die abgespeichert unter Umständen mehr Platz als das Originalbild benötigen. Damit ist klar, dass IFS-Codes auf dem Weg zur Kompression beliebiger Bilder nur von kleinem Nutzen sind. Ein nächster Schritt sind deshalb die sogenannten RIFS-Codes.

RIFS-Codes

Wie bereits erwähnt, ist es im allgemeinen nicht mit vernünftigem Aufwand möglich, ein Bild mit verkleinerten Kopien von sich selbst vollständig zu überdecken. Oftmals tritt jedoch der Fall auf, dass Teile eines Bildes mehrfach in ähnlicher Form auftreten. Diesem Umstand wird mit den RIFS-Codes (Recurrent Iterated Function Systems) Rechnung getragen. Dank ihnen ist es nicht mehr nötig, das Originalbild mit verkleinerten Kopien von sich selbst zu überdecken. Stattdessen genügt es, jeden Teil des Bildes mit verkleinerten Kopien von sich selbst oder von anderen Teilen des Bildes abzudecken. Zu diesem Zweck wird das Original zunächst in geeignete Teilbilder zerlegt. Diese liegen in fiktiven Ebenen, welche dann mit Kopien von anderen Ebenen oder von sich selbst vollständig bedeckt werden müssen. Die Menge der dazu nötigen Transformationen mit ihren jeweiligen Start- und Zielebenen bildet den RIFS-Code des Originalbilds.

Zur Illustration sei im folgenden der RIFS-Code von Bild 6 hergeleitet. Es handelt sich dabei um ein einfaches, aber nicht mehr offensichtlich fraktales Gebilde. Das Bild 7 zeigt die verschiedenen Ebenen und die zugehörigen Transformationen.

Schritt 1: Ebenen wählen

Man zerlegt das Bild in Teile, die ihrerseits mit sich selbst oder mit anderen Teilen abgedeckt werden können. Beim betrachteten Beispiel bieten sich dafür einerseits ein Dreieck und andererseits ein Kreis an. Daraus resultieren insgesamt drei Ebenen:

Bild 6 Kodierungsbeispiel Ausrufezeichen

Ebene 1: Originalbild Ausrufezeichen
Ebene 2: Dreieck
Ebene 3: Kreis

Schlussendlich interessiert nur die Ebene 1 mit dem Ausrufezeichen; die beiden Ebenen 2 und 3 sind fiktive Hilfskonstruktionen, welche bei der Dekodierung nicht gezeichnet werden.

Schritt 2: Abdecken der Ausgangsfigur

Es gilt nun, die verschiedenen Ebenen mit Transformationen von sich selbst oder von anderen Ebenen vollständig zu überdecken. Es ist zweckmässig mit Ebene 1 zu beginnen, welche die Ausgangsfigur, also das Ausrufezeichen, enthält. Man erkennt, dass das Ausrufezeichen aus einem Dreieck und einem Kreis besteht. Es lässt sich demzufolge aus je einer Transformation von Ebene 2 (Dreieck) und Ebene 3 (Kreis) nach Ebene 1 zusammensetzen. Die beiden Transformationen werden im Bild 7 durch die Pfeile 1 und 2 dargestellt.

Die Ausgangspunkte der Pfeile sollen symbolisieren, dass mit der jeweiligen Transformation die *ganze* Ebene abgebildet wird.

Schritt 3:
Abdecken der fiktiven Ebenen

Das Dreieck in Ebene 2 kann losgelöst von den anderen Ebenen mit drei Dreiecken – also mit drei Abbildungen auf sich selbst – ausgelegt und damit vollständig definiert werden. (Pfeile 3, 4, 5). Um das folgende verständlich zu machen, ist an dieser Stelle ein kleiner Einschub nötig: Die Dekodierung von RIFS-Codes funktioniert nach dem gleichen Prinzip wie diejenige von IFS-Codes. Der Random-Walk springt hier zwischen den verschiedenen Ebenen hin und her. Die jeweils nächste Abbildung wird wie bei den IFS-Codes zufällig ausgewählt, allerdings mit der Einschränkung, dass nur Transformationen erlaubt sind, die von der aktuellen Ebene ausgehen. Befindet sich der Random-Walk also gerade in Ebene 2, darf zwar etwa eine Abbildung von Ebene 2 nach Ebene 1, nicht aber eine solche von Ebene 3 nach Ebene 1 angewendet werden. Weil das Originalbild aus den Teilbildern in den fiktiven Ebenen zusammengesetzt wird, muss zusätzlich sichergestellt werden, dass jede fiktive Ebene während des Random-Walks erreicht werden kann.

Damit zurück zur Abdeckung der fiktiven Ebenen: Das Dreieck kann zwar mit den oben beschriebenen drei Abbildungen vollständig aufgebaut werden, es existiert aber noch keine Transformation, die in Ebene 2 hineinführt. Das bedeutet, dass der Random-Walk nicht in Ebene 2 gelangt und deshalb letztlich nie die Abbildung angewendet werden kann, die den oberen Teil des Ausrufezeichens in Ebene 1 aufbaut. Um dies zu vermeiden, führt man eine zusätzliche Transformation ein, die das Ausrufezeichen in das Innere des Dreiecks abbildet (Pfeil 6).

Der Kreis in Ebene 3 lässt sich mit einer Abbildung auf sich selbst (Rotation um einen Winkel $\varphi \neq$ Teiler von 360°) darstellen. Auch hier muss wieder sichergestellt werden, dass der Random-Walk überhaupt die Ebene 3 erreichen kann. Dies geschieht mit der Transformation des Ausrufezeichens ins Innere des Kreises (Pfeile 7 bzw. 8).

Schritt 4: Elemente des RIFS-Codes

Die zur Übermittlung oder Speicherung des Ausrufezeichens notwendigen Daten bestehen aus den benötigten Transformationen und der Information, welche Start- und Zielebene diese Transformationen jeweils besitzen. Ist man an einem regelmässigen Bildaufbau interessiert, müssen den einzelnen Abbildungen wie bei den IFS-Codes bestimmte Wahrscheinlichkeiten zugeordnet werden. Diese Wahrscheinlichkeiten können durch Lösen eines Gleichungssystems bestimmt werden [3; 4].

Abschliessend bleibt noch zu erwähnen, dass das hier behandelte Beispiel dank seiner geometrischen Gestalt natürlich viel einfacher und effizienter kodiert werden könnte. (Es würde genügen, die Koordinaten der drei Eckpunkte des Dreiecks bzw. die Koordinaten des Mittelpunktes und den Radius des Kreises festzuhalten.) Das Beispiel zeigt aber den Aufbau eines RIFS-Codes besonders anschaulich. Der nächste Abschnitt soll nun die Anwendung von RIFS-Codes auf beliebige Bilder erläutern.

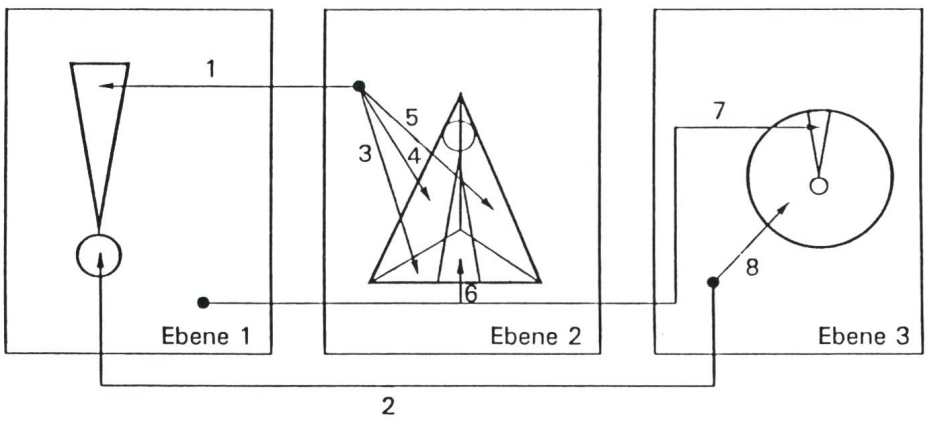


Bild 7 Ebenen und Transformationen des Kodierungsbeispiels

Kodierung beliebiger Bilder

Im folgenden sollen einige Aspekte der Anwendung von RIFS-Codes auf beliebige Monochrom-Bilder betrachtet werden. Detailliertere Informationen zu diesem Abschnitt und der Behandlung von Graustufen- und Farbbildern finden sich in [4].

Bei der Kodierung beliebiger Bilder mittels RIFS-Codes muss man grundsätzlich zwischen manuellen und automatischen Methoden unterscheiden.

Manuelle Kodierung

Die manuelle Kodierung von beliebigen Bildern läuft prinzipiell nach demselben Verfahren, wie dies für das spezielle Beispiel im letzten Abschnitt gezeigt wurde. Zuerst muss das Bild nach einem Teilbereich abgesucht werden, der mehrmals in ähnlicher Form auftritt. Ein Element dieser Menge wird in eine fiktive Ebene gelegt, dann werden die affinen Abbildungen berechnet, die nötig sind, um mit ihm jedes Element der Menge in der Originalbild-Ebene abzudecken. Nach diesem Prinzip definiert man immer weiter neue Ebenen (und Abbildungen von diesen in die Originalbild-Ebene), bis das Originalbild vollständig ausgelegt ist.

Anschliessend müssen die fiktiven Ebenen untereinander ausgelegt werden. Dies ist im allgemeinen der schwierigste Teil der Kodierungsprozedur[2]. Es kommt durchaus vor, dass sich diese Aufgabe für eine bestimmte Wahl der fiktiven Ebenen nicht befriedigend lösen lässt. In diesem Fall müssen diese neu definiert werden, bis ein zufriedenstellendes Resultat erreicht wird.

Es ist offensichtlich, dass mit der beschriebenen Methode nie hundertprozentige Genauigkeit erreicht werden kann, das heisst ein dekodiertes Bild wird immer gewisse Fehler aufweisen. Diese Fehler dürfen von Auge nur schwer auszumachen sein, wenn RIFS-Codes in der Praxis eingesetzt werden sollen. Welche Resultate erreicht werden können, zeigen die Bilder 8 und 9. Sie zeigen ein Originalbild und die durch Dekodierung des RIFS-Codes entstandene Version. Für dieses Bild werden fünf fiktive Ebenen und insgesamt vierzig affine Transformationen benötigt [4]. Man erreicht eine Datenkompression von rund 1:90.

[2] Für das Beispiel im letzten Abschnitt war das einfach, weil die fiktiven Ebenen mit Kopien von sich selbst abgedeckt werden konnten.

Automatische Kodierung

Das vorher beschriebene Verfahren eignet sich nicht für eine automatische Kodierung, weil in hohem Mass Fähigkeiten wie Abstrahieren und Mustererkennung verlangt werden. Genau in diesem Bereich liegen aber auch heute noch die Schwächen des Computers. Man muss also versuchen, die entsprechenden Schritte im manuellen Kodierungsverfahren so zu modifizieren, dass sie leicht algorithmisch dargestellt werden können. Eine Möglichkeit ist etwa die folgende: Statt im Originalbild möglichst oft auftretende ähnliche Teile zu suchen, wird es beispielsweise standardmässig mit einem Raster bedeckt. Die einzelnen Rasterelemente werden dann in ähnlicher Weise wie bei der manuellen Methode mit affinen Transformationen aufeinander abgebildet.

Dieses automatische Verfahren hat den Nachteil, dass damit ein Verlust an Kompressionsfaktor entsteht, weil eben ähnliche Bildelemente nicht mehr als solche erkannt werden. Ausserdem verteilen sich Fehler in nicht beeinflussbarer Weise über das Bild, während bei der manuellen Methode darauf geachtet werden kann, dass man für das Auge besonders wichtige Partien sehr genau kodiert.

Aktueller Stand und Ausblick

Die fraktale Datenkompression befindet sich trotz der sich einstellenden ersten Erfolge noch im Anfangsstadium. Auf dem Gebiet der Erforschung der Fraktale und insbesondere ihrer Anwendung in der Datenkompression hat der amerikanische Mathematiker Michael Barnsley Pionierarbeit geleistet [5; 6; 7]. Mit seiner Firma Barnsley Communications Inc. in Atlanta versucht er seit einigen Jahren die theoretischen Erfolge kommerziell[3] auszunützen. Aus diesem Grund gelangen Informationen über seinen aktuellen Forschungsstand nur spärlich an die Öffentlichkeit. Im Oktober 1990 publizierte Arnaud Jacquin einen Artikel [8], in dem zum ersten Mal genauere Angaben über ein automatisches Kodierungsverfahren gemacht werden. Auch in der Schweiz werden Anstrengungen unternommen, der fraktalen Datenkompression zum Durchbruch zu verhelfen. Am ETH-Institut für Kommunikationstechnik, Zürich, sind entsprechende Forschungsarbeiten im Gange.

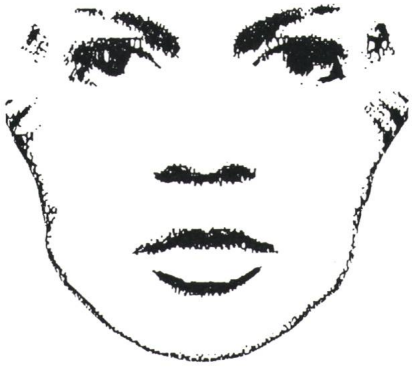


Bild 8 Originalbild

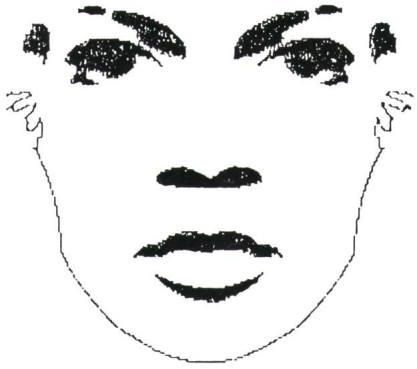


Bild 9 Aus RIFS-Code dekodiertes Bild

Die Erfolgsaussichten für eine Kompressionsmethode auf fraktaler Basis hängen fast ausschliesslich davon ab, ob bessere Verfahren für die automatische Kodierung gefunden werden können.

[3] So wurden beispielsweise eine Hardware-Kodierungskarte und ein Software-Dekoder für IBM-PC und Kompatible entwickelt [7].

Literatur

[1] *Jürgens H., Peitgen H.O., Saupe D.:* Fraktale – eine neue Sprache für komplexe Strukturen. Spektrum der Wissenschaft, Sonderband Chaos und Fraktale 1989.

[2] *Gipser Th.:* Fraktale – eine neue Sprache der Wissenschaft. Bulletin SEV/VSE 81(1990)9.

[3] *Boss R.D.und Jacobs E.W.:* Fractal-Based Image Compression. NTIS Prices, September 1989.

[4] *Meneghini R.* und *Stutz P.:* Fraktale Bilddatenkompression. Diplomarbeit ETH Zürich, Dezember 1990.

[5] *Barnsley M.:* Recurrent Iterated Function Systems. Constructive Approximation 5(1989)1.

[6] *Barnsley M.* und *Jacquin A.:* Application of Recurrent Iterated Function Systems. Proceedings SPIE Int. Soc. Opt. Eng. Vol. 100(1988)1.

[7] *Barnsley M.:* P.OEM-Broschüre, Barnsley Communications Inc. 1990.

[8] *Jacquin A.:* Fractal Image Coding Based on a Theory of Iterated Contractive Image Transformations. SPIE Vol. 1360 Visual Communications and Image Processing 1990 S. 227–239.

Für Unternehmer mit Wachstumspotential. Ascoline.

Denken Sie kurz einmal 5 oder 10 Jahre weiter. Die Unternehmen wachsen und wachsen, und damit steigt auch die Zahl der Telefonanschlüsse. Nun, glücklicherweise sind wir dank dem modularen Konzept von Ascoline darauf vorbereitet. Genauso wie auf die digitale Zukunft der Telekommunikation, sprich ISDN. Denn speziell dafür haben wir unsere neuste Teilnehmervermittlungsanlage entwickelt. Also zu einem grossen Teil auch für die Kinder Ihrer Kunden. Dabei erfüllt Ascoline selbstverständlich bereits heute alle wichtigen Bedürfnisse, wie zum Beispiel optimale Erreichbarkeit, einfache Bedienung oder hohe Flexibilität. Und darum sollten Sie sich auch bereits heute über Ascoline informieren. Senden Sie Ihre Visitenkarte an: Ascom Gfeller AG, Ascoline, Schwarztorstrasse 50, 3000 Bern 14. Oder kontaktieren Sie Ihre zuständige Fermeldedirektion, Tel. 113.

PTT

Teilnehmervermittlungsanlagen: ascom *denkt weiter.*