

Debreceni Egyetem  
Informatikai Kar

Fraktál alapú képtömörítés  
Káosz és fraktálok programozói szemmel

Témavezető:  
Dr. Fazekas Gábor  
egyetemi docens

Készítette:  
Tilki Csaba  
programtervező matematikus

Debrecen

2007.

# Tartalomjegyzék

Előszó . . . . .	4
<b>1. Káoszelmélet</b>	<b>5</b>
1.1. Bevezetés . . . . .	5
1.2. Mi a káosz? . . . . .	5
1.2.1. Klasszikus kaotikus modellek . . . . .	7
1.2.1.1. Logisztikus modell . . . . .	7
1.2.1.2. A Lorenz-modell . . . . .	8
<b>2. Fraktálok</b>	<b>10</b>
2.1. Mi az a fraktál? . . . . .	10
2.2. Fraktálok a természetben . . . . .	12
2.3. Történeti áttekintés . . . . .	13
2.4. Klaszikus fraktálok . . . . .	14
2.4.1. Cantor halmaz . . . . .	14
2.4.2. Koch-görbe . . . . .	15
2.4.3. Sierpinski-háromszög . . . . .	15
2.4.4. Sierpinski-szőnyeg . . . . .	16
2.4.5. Menger-szivacs . . . . .	16
2.4.6. Julia halmazok . . . . .	17
2.4.7. Madelbrot halmaz . . . . .	19
2.5. A fraktálok formális leírásának módjai . . . . .	19
2.5.1. L-System fraktálok . . . . .	21
2.5.2. IFS fraktálok . . . . .	22
2.5.3. Komplex fraktálok . . . . .	22
2.5.4. Különös attraktorok . . . . .	23
2.6. Káosz és fraktálok különböző tudományterületeken . . . . .	23
2.6.1. Orvostudomány . . . . .	23
2.6.2. Populációdinamika . . . . .	23
2.6.3. Genetika, posztgenetika: . . . . .	23

2.6.4.	Információelmélet . . . . .	24
2.6.5.	Matematika . . . . .	24
2.6.6.	Informatika . . . . .	25
2.6.6.1.	Alapvető adatszerkezetek . . . . .	25
2.6.6.2.	Számítógépes szimulációk . . . . .	26
2.6.6.3.	Álvéletlenszám generálás . . . . .	27
2.6.6.4.	Kriptográfia . . . . .	27
2.6.6.5.	Szteganográfia . . . . .	27
2.6.6.6.	Digitális képfeldolgozás . . . . .	27
<b>3.</b>	<b>Fraktál alapú képtömörítés</b>	<b>28</b>
3.1.	Matematikai alapok . . . . .	29
3.1.1.	Fraktálok . . . . .	29
3.1.2.	Iterált függvényrendszerek . . . . .	29
3.1.3.	Teljes metrikus terek . . . . .	30
3.1.4.	Kontraktív leképezések és IFS-ek . . . . .	31
3.2.	Az algoritmus javításai, egyéb változatai . . . . .	32
3.2.1.	Klasszifikáció . . . . .	33
3.2.2.	Archetípusok . . . . .	33
3.2.3.	PIFS és neurális hálók . . . . .	33
3.2.4.	PIFS és genetikus algoritmusok(GA) . . . . .	35
3.3.	Egyéb felhasználási területek . . . . .	38
3.3.1.	Képtömörítés . . . . .	38
3.3.2.	Képindelexelés . . . . .	38
3.3.3.	Vízjelzés . . . . .	39
	<b>Bibliográfia</b>	<b>40</b>

# Előszó

A káoszelmélet alig 30 éves múltra tekinthet vissza, mégis gyökereiben felforgatta a világról alkotott képünket. Jelen dolgozat főszereplői a fraktálok. Bár a dolgozat fő témáját a fraktál alapú képtömörítés képezi, szerettem volna egy átfogóbb képet adni a terület mai állásáról, és főleg a különböző alkalmazásokról. Az első fejezet a káosz fogalmával, a kaotikus rendszerek alapvető tulajdonságaival igyekszik megismertetni az olvasót, a második a fraktáloknak nevezett különleges alakzatokkal, ezekenek a kaotikus rendszerekkel való szövevényes kapcsolatával foglalkozik, míg a harmadik egy lehetséges alkalmazási módot mutat be részletesebben.

## Köszönetnyilvánítás

Ezúton szeretnék köszönetet mondani Dr. Fazekas Gábornak, Dr. Gáspár Vilmosnak, Bátfai Norbertnek, Varga Katalinnak, és Seller Péter Andrásnak a dolgozat elkészítésében nyújtott segítségükért.

# 1. fejezet

## Káoszelmélet

### 1.1. Bevezetés

A káosz szó eredetileg tökéletes ürességet, tátongó űrt jelentett, a világ kezdeti állapotát, amiből a Kozmosz, a mindenség származik. A püthagoreusok már másképp gondoltak a káoszra, ők úgy vélték, a káosz összevisszaságot jelent, szerintük a Kozmosz minden összetevője jelen volt már a Káosz korszakában is, csak ezek a teljes rendezetlenség állapotában voltak. A szónak ezen jelentése a mai napig fennmaradt, a mai köznyelv is a (térbeli) rendezetlenség, összevisszaság szinonímjaként használja. A '70-es évek óta azonban egy kifejezetten tudományos jelentéssel is bír, mely jelentősen eltér az előbb említettektől. James Yorke, a Marylandi Egyetem Fizikatudományi és Műszaki Intézetének vezetője egy akkoriban születő új tudományágot nevezett el róla. Az elnevezés egyszerre zseniális és félrevezető. Egyesek szerint a névválasztás az oka a rengeteg félreértésnek, ami ezt a területet jellemzi.

### 1.2. Mi a káosz?

Egészen a 20. századig a tudósok úgy gondolták, a determinisztikus rendszerek viselkedése előre megjósolható. Mindannyian ismerjük Pierre-Simon Laplace híresé vált kijelentését: "A világmindenség jelen állapotát a megelőző állapot hatásaként, és a rákövetkező állapot okaként kell értelmeznünk. Ha adott volna egy értelem, mely képes felfogni a természetet irányító erőket és az alkotóelemek pillanatnyi, egymáshoz viszonyított helyzetét - egy olyan értelem, mely ezt az adattömeget képes feldolgozni és analizálni -, ez az értelem ugyanazon törvények szerint vizsgálná a leghatalmasabb testek és a legkisebb atomok mozgását, akkor nem volna számára bizonytalanság, a jövőt és múltat egyaránt tisztán látná."

Ezt a bizonyos felsőbb értelmet szokás Laplace-démonként emlegetni. A sok szabadsági fokkal rendelkező rendszerekről tudvalevő, hogy véletlenszerű viselkedét mutatnak, ez a sok összetevő bonyolult egymásra hatásának az eredménye. Általános volt viszont a nézet, hogy az egyszerűen

leírható rendszerek csakis egyszerű módon viselkedhetnek. Nem kis meglepetést okozott hát, mikor Edward Lorenz olyan, néhány nemlineáris egyenletből álló modellt talált, amely bizonyos paraméterek mellett teljességgel előrejelezhetetlen időbeli viselkedést produkált. Új paradigma született.

A káoszra különböző definíciók születtek:

- Egyfajta periodikusság nélküli rend.
- Látszólag véletlenszerűen ismétlődő viselkedés egy egyszerű, determinisztikus rendszeren belül.
- Az egyszerű, beépített véletlenszerű vonások nélküli modellek azon képessége, hogy nagyon szabálytalan viselkedést tanúsítsanak. /Ian Stewart/

**A kaotikus mozgás három alapvető jellemzője:**

- *Időben szabálytalan viselkedés*, mely nem állítható elő periodikus mozgások összegeként.
- *A hosszú távú előrejelezhetetlenség*, amely a kezdőfeltételekre való érzékenységből fakad.
- *A fázistérbeli bonyolult fraktálszerkezet*, melynek dimenziója tipikusan alacsony.

**Mi nem káosz?**

- *Nem jelent térbeli rendezetlenséget* annak ellenére, hogy a köznyelv általában épp ilyen értelemben használja.
- *Nem szabályos* mozgás, hiszen előre nem jelezhető.
- *Nem zaj*. A zajos mozgás a sok összetevőből álló rendszerek adott komponensének véletlenszerű viselkedése, mely a többi komponenssel való bonyolult kölcsönhatás, nem pedig (kizárólag) a belső dinamika eredménye.
- *Nem egyszeri instabilitás*. A káosz maga az állandósult instabilitás. Csakis akkor beszélhetünk kaotikus viselkedésről, ha a mozgás egyre újabb és újabb instabil állapotokat közelít meg.
- *Nem turbulencia*. A turbulencia a nagy szabadsági fokú viselkedés szélsőséges példája, mely időben és térben teljesen szabálytalan.

## 1.2.1. Klasszikus kaotikus modellek

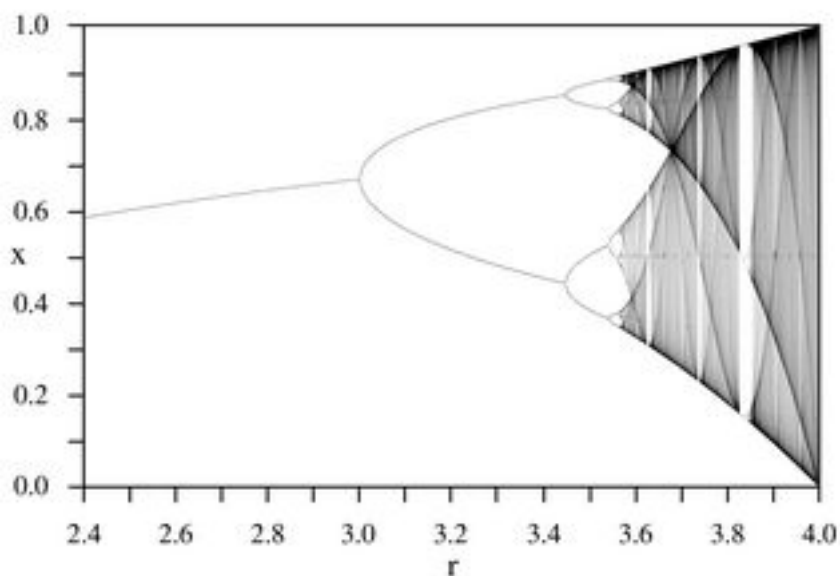
### 1.2.1.1. Logisztikus modell

A logisztikus modellt egy Robert May nevű biológus tette közismertté, az egyik legegyszerűbb – így a leggyakrabban idézett – példa arra, hogyan képes egy végtelenül egyszerű nemlineáris dinamikai egyenlet komplex, kaotikus viselkedésre. Matematikai alakja:

$$x_{n+1} = rx_n(1 - x_n)$$

ahol  $x_n$  0 és 1 közötti valós szám, és valamilyen értelemben az  $n$ . évi populációt reprezentálja. Ennek megfelelően  $x_0$  a kezdeti populációnak megfelelő érték. Az  $r$  pozitív valós szám, amely értékétől a rendszer viselkedése erősen függ. Az  $r$  értékét a  $[0, 4]$  intervallumon végigfuttatva a következő figyelhető meg:

- 0 és 1 közötti  $r$  értékekre a populáció biztosan halálra van ítéelve, a kezdeti populáció mértékétől teljesen függetlenül.
- 1 és 2 közötti értékek esetén a populáció mérete gyorsan megállapodik az  $\frac{r-1}{r}$  értéken.
- 2 és 3 között hasonló figyelhető meg, ám az  $\frac{r-1}{r}$  fixpont instabillá válik, megjelenik az oszcilláció.
- 3 és  $1 + \sqrt{6}$  ( $\approx 3.45$ ) között a populáció 2 ( $r$ -től függő) érték között oszcillál (perióduskettőződés, bifurkáció).
- $\approx 3.45$  és  $\approx 3.54$  között a populáció 4 különböző érték között oszcillál.
- $r = 3.54$  fölött 8, 16, 32, ... periódusú oszcillációk jelennek meg. A perióduskettőződés üteme egyenletesen növekszik, bármely két, egymást követő bifurkációhoz tartozó intervallum aránya  $\delta = 4.669$ , az ún. Feigenbaum konstans.
- $r = 3.57$  Félbeszakad a perióduskettőzések sorozata, felüti fejét a káosz. Semmilyen periodicitás nem figyelhető meg, a kezdeti populáció kis megváltoztatása drámai mértékben változtatja meg a populációszám időbeli alakulását.
- $r = 3.82$  körül háromperiódusú oszcilláció jelenik meg. Ezt 6, 12, ... periódusú oszcillációk követik.
- $r = 4$  A kaotikus tartomány vége, az értékek elhagyják a  $[0, 1]$  intervallumot.



1.1. ábra. Bifurkációs diagram

### 1.2.1.2. A Lorenz-modell

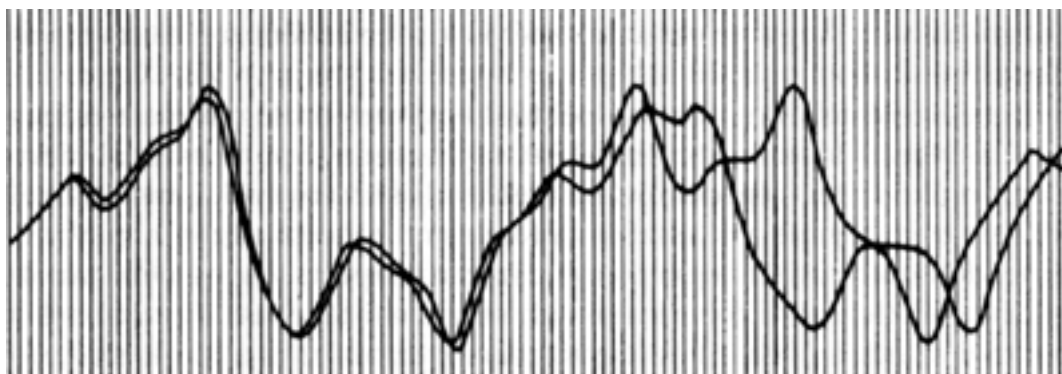
A folytonos idejű kaotikus rendszerek mintapéldája, nevét Edward Lorenz amerikai meteorológusról kapta, aki 1963-ban egy egyszerű időjárási modell felállításával próbálkozott. Az alábbi egyenletrendszert vizsgálta:

$$\dot{x} = \sigma(y - x),$$

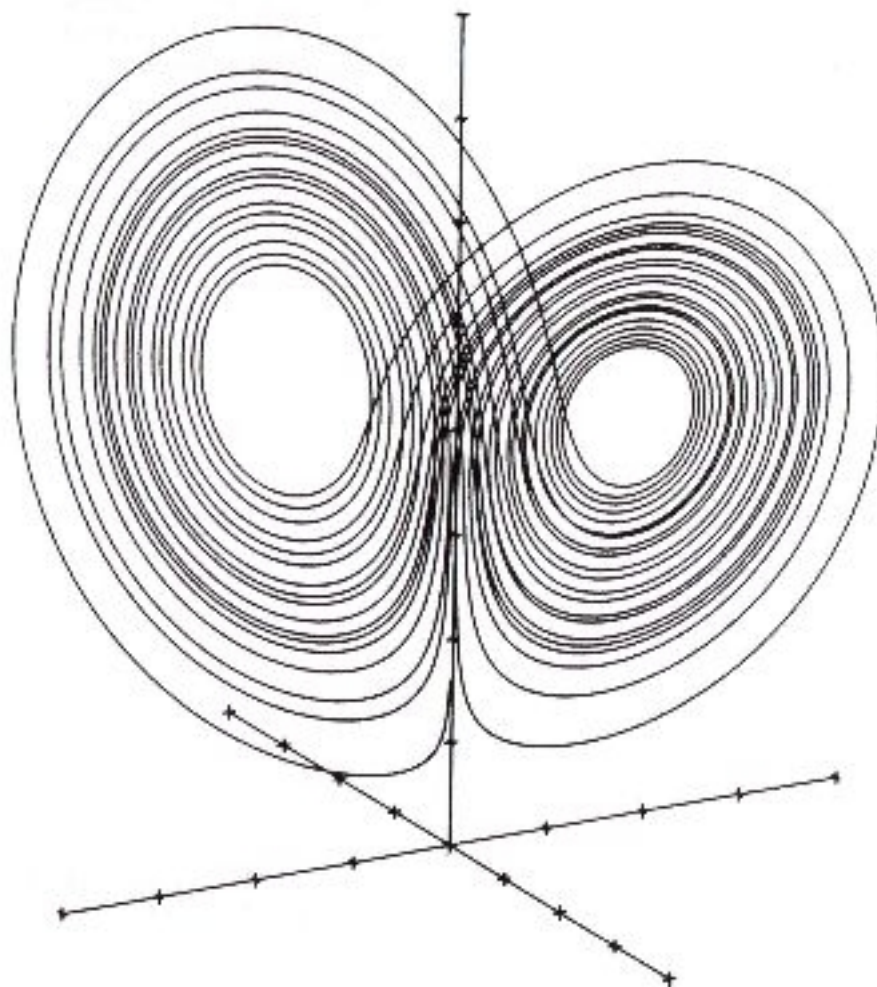
$$\dot{y} = rx - y - xz,$$

$$\dot{z} = -bz + xy$$

Észrevette, hogy  $r = 28$ ,  $\sigma = 10$ ,  $b = 8/3$  paraméterek mellett kis kezdeti feltételekbeli különbség esetén is igen eltérő időfejlődés tapasztalható (1.2-es ábra). Amikor a rendszer viselkedését fázistérben ábrázolta, egy igen furcsa attraktor képe bontakozott ki a szeméi előtt (1.3. ábra). Ez a róla Lorenz-attraktornak elnevezett különös ábra azóta a káosz egyik jelképévé vált. Tulajdonságainak vizsgálatára később még visszatérünk.



1.2. ábra. A kezdeti értékekre való érzékenység mintapéldája



1.3. ábra. A híres Lorenz-attraktor

## 2. fejezet

# Fraktálok

### 2.1. Mi az a fraktál?

A fraktálok törtdimenziós alakzatok. De mit is jelent ez? Minden ember rendelkezik valamiféle naív dimenziófogalommal. Tudjuk, hogy egy pontnak nincs semmilyen irányú kiterjedése, így dimenziója 0. Egy szakasz pontosan egy irányba terjed ki, ezért dimenziója csakis 1 lehet. Hasonló logikával belátható, egy síkidom 2, egy test 3 dimenzióval rendelkezik. Mivel kiterjedések darabszámáról van szó, természetes kikötésnek tűnik, hogy egy objektum dimenziója csakis egész szám lehet. De vajon mit kezdjünk egy olyan görbével (vagyis elvileg egydimenziós matematikai objektummal), amely képes egy (kétdimenziós) síkot kitölteni? Vagy vessünk újra egy pillantást az 1.3.ábrára! Vajon hány dimenziós ez az alakzat? Egyetlen kanyargó vonalból áll, de mégis mintha valamiféle síkidom lenne, sőt egy kicsit a harmadik dimenzióba is behatol. Mintha dimenziója valahol 2 és 3 között lenne, csakhogy 2 és 3 között nincs egyetlen egész szám sem. Naív dimenziófogalmunk megfelelő általánosítására, a valós számokra való kiterjesztésére lenne szükségünk. De előtte ismerkedjünk meg a topológiai dimenzió fogalmával.

**Definíció** (Topológiai dimenzió): Egy csupa izolált pontból álló halmaz topológiai dimenziója 0. Egy  $F$  halmaz topológiai dimenziója  $n$ , ha  $F$  minden pontjának tetszőlegesen kicsi szomszédjai határának topológiai dimenziója  $n - 1$ .

A topológiai dimenzió mindig egész szám. Például tetszőleges hosszúságú intervallum topológiai dimenziója 1, mivel minden egyes pontjához találhatunk egy szomszédos intervallumot, melynek határai izolált pontok, így topológiai dimenziójuk 0. De hogyan terjesztjük ki a dimenzió fogalmát valós számokra? Többféle lehetőség kínálkozik, a legnépszerűbb ún. fraktáldimenziók a Hausdorff- és a boxdimenzió. Ezek közül a gyakorlatban leginkább a másodikat használják.

**Hausdorff-dimenzió:** Induljunk ki egy szakaszból. Ha az eredeti szakaszt az  $N$ -ed részére zsugorítjuk (idegen szóval skálázzuk), akkor ebből az új szakaszból pontosan  $N$  (vagyis  $N^1$ ) darabra van szükség, ha le akarjuk fedni velük az eredeti szakaszt. Ha egy négyzetet zsugorítunk össze az  $N$ -ed részére, akkor pontosan  $N^2$  darab, kocka esetén  $N^3$  darab kicsinyített másra van szükségünk. Könnyen észre vehetjük, hogy a kitevőben lévő szám az objektum euklideszi (vagy topológiai) dimenziójával egyezik meg. Miért ne lehetne a kitevő értéke valós szám? Hisz valós kitevőjű hatványokat „gond nélkül” tuduk kezelni a matematikában. Tehát csak arra kell ügyelnünk, hogy az adott objektum lefedhető legyen a saját kicsinyített másaival, más szóval legyen *önhasonló*. Így tetszőleges önhasonló alakzat dimenziója kiszámítható az alábbi módon:

$$D = \lim_{\epsilon \rightarrow 0} \frac{\log N(\epsilon)}{\log \frac{1}{\epsilon}}$$

ahol  $N(\epsilon)$  darab  $\epsilon$  méretű alakzatra (az eredeti objektum skálázott változataira) van szükség a teljes, eredeti objektum letakarásához. A fent említett Lorenz-attraktor Hausdorff-dimenziója például 2.06

**A boxdimenzió:** A boxdimenzió meghatározásához egy négyzetrácsot (magasabb dimenzióban kockarácsot, stb.) kell a vizsgált alakzatra helyezni. Ezután meghatározzuk azon cellák minimális számát, amelyek segítségével az alakzatunk lefedhető. Ha ezzel megvagyunk, finomítsuk a rácsot, használjunk például fele akkora cellaméretet, mint kezdetben. A lefedéshez szükséges cellák száma így nyilvánvalóan megnő, számunkra most az az érdekes, hogy mennyivel. Egy egyenes szakas esetében a fele akkora cellákból kétszer annyira, míg síkidomok esetén négyszer annyira lenne szükség. Jelöljük  $N$ -nel egy adott alakzat lefedéséhez szükséges cellák számát és jelölje  $r$  az alkalmazott cellaméretet. Ekkor a következő összefüggés érvényes:

$$N = r^{-D_B}$$

ahol a kitevőben szereplő  $D_B$ -t az alakzat boxdimenziójának nevezzük. Rendezve a fenti egyenletet a következőt kapjuk:

$$D_B = \frac{\log N}{\log \frac{1}{r}}$$

Előnye, hogy nem szükséges egzakt önhasonlóság a használatához, így akár ún. önaffin alakzatok dimenziójának mérésére is felhasználható. A boxdimenziót Minkowski-Bouligand dimenzióknak is nevezik.

Ezek ismeretében azt mondhatjuk, hogy fraktálnak nevezünk minden olyan önhasonló alakzatot, amelynek Hausdorff-dimenziója nagyobb, mint a topológiai dimenziója. Ezt elfogadhatnánk akár definíciónak is, de sajnos nem minden fraktál (pl. az ún. térkitöltő görbék) esetén teljesül ez a kitétel. Sajnos ezideáig nem sikerül pontos, egzakt definíciót találni arra, hogy mit

nevezünk fraktálnak, így aztán illik tisztázni, hogy adott helyzetben mit értünk alatta. Ha pontos definíció nem is létezik, általában elég könnyen el tudjuk dönteni egy alakzatról, hogy az fraktál-e vagy sem. A legtöbb fraktál a következő tulajdonságokkal (vagy legalábbis majdnem mindegyikkel) rendelkezik:

- Hausdorff-dimenziójuk nagyobb, mint a topológiai dimenziójuk.
- Skálázással szemben invariánsak, legalábbis néhány mérettartományban.
- Kevés adattal (rövid formulával) leírhatóak.

## 2.2. Fraktálok a természetben

Ha az előző matematikai okfejtés alapján valaki azt gondolná, hogy a fraktálok elborult matematikai alakzatok, és semmi közük a valósághoz, hatalmasat tévedne, a fraktálok megdöbbentő mennyiségben vesznek minket körül. Benois Mandelbrot szavaival élve: „A felhők nem gömbök, a hegyek nem kúpok. A villám nem egyenes utat követ. „ Csak néhány kiragadott példa fraktálstruktúrával bíró természeti képződményre:

- Felhők
- Bolygók, holdak felszíne
- Partvonalak, törésvonalak
- Fák, növények(2.1.ábra)
- Villámok
- Hegyvonulatok
- Folyók
- Tengeri élőlények külső váza
- Hópelyhek

E lista láttán felmerül az emberben, hogy egyáltalán van-e olyan természeti képződmény, amely bír fraktál tulajdonságokkal (a fraktálok eme tobzódására a harmadik fejezetben tárgyalt gyakorlati alkalmazás erősen építeni fog). Ám nem csak a minket körülvevő természet építkezik fraktálokból, hanem mi magunk is, néhány példa, szintén a teljesség igénye nélkül:



2.1. ábra. Egy természetes fraktál: a karfiol

- Érhálózat
- A tüdő
- Az agy felszínének mintázata
- A vesék szűrői

## 2.3. Történeti áttekintés

### **Benois Mandelbrot**

Mandelbrot nevéhez fűződik a fraktál kifejezés (mint gyűjtőfogalom) megteremtése. A szó a latin *fractus* (jelentése: töredezett) szóból ered. Mandelbrot az IBM-nél dolgozott, ahol adatátviteli problémák megoldásával foglalkozott, konkrétan azzal, hogy hogyan lehetne információvesztés nélkül minél hatékonyabban kiszűrni az adatátviteli csatornából a zajt. Azt találta, hogy a zaj egy nagyon különös elrendeződést mutat (lásd Cantor-halmaz). Egy másik munkájában a következő kérdést tette fel: „Milyen hosszú Nagy-Britannia partvonala? „Elsőre talán nem tűnik érdekesnek vagy különösebben nehéznek a kérdés, de ha figyelembe vesszük, hogy Afrika területe lényegesen nagyobb, mint Európáé, mégis Európa partvonala a hosszabb (a partvonal töredezettsége miatt. Fel merült Mandelbrotban a kérdés, hogy egyáltalán lehet-e véges



2.2. ábra. A Cantor halmaz képzésének folyamata

egy ilyen partvonal. Hiszen minél pontosabb mérőműszert használunk, annál több bemélyedés és kiszögellés hossza adódik hozzá az összeghez. A partvonal tehát tipikus példa arra, hogy hogyan zárhat körbe véges területet egy elvileg végtelen hosszú vonal. Jól látható, hogy a fraktálok különleges tulajdonságai egészen eltérő tudományterületeken is felbukkannak, Manedelbrot volt az, aki először foglalta mindezt egységes rendszerbe.

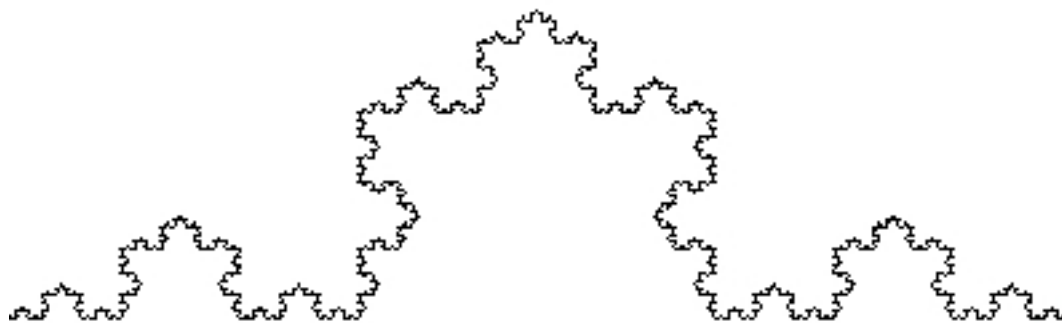
## 2.4. Klaszikus fraktálok

### 2.4.1. Cantor halmaz

*A konstrukció lépései:* Induljunk ki a  $[0, 1]$  (zárt) intervallumból. Most távolítsuk el az  $(\frac{1}{3}, \frac{2}{3})$  nyílt intervallumot (vagyis vegyük a kezdeti és ezen intervallum különbségét). Ennek eredményeképpen az eredeti halmazunk 2 részhalmazra esik szét:  $[0, \frac{1}{3}]$ ,  $[\frac{2}{3}, 1]$ . Ismételjük meg ezekre is a fenti műveletet, azaz távolítsuk el a középső harmadukat. Így már négy, egyenként  $\frac{1}{9}$  hosszúságú részintervallumunk lesz, ezeket tovább alakítva, a fentieket a végtelenségig ismételve az eredményül kapott halmazt nevezzük Cantor-halmaznak (2.2-es ábra).

A Cantor halmaznak van egy nagyon érdekes tulajdonsága, tudniillik ha kiszámoljuk az eltávolított intervallumok összhosszát, az éppen megegyezik a kiindulási szakasz hosszával. Ráadásul ezen szakaszok eltávolítása után még mindig végtelensok pontja maradt az alakzatnak. Mivel a Cantor halmazban egy adott méretű szakaszt harmadára kicsinyített másából pontosan kétszer lehet lefedni, ezért a Cantor halmaz fraktáldimenziójára a következő értéket kapjuk:

$$D = \frac{\ln(2)}{\ln(3)}$$



2.3. ábra. Koch-görbe

ami körülbelül 0.6309-nak felel meg, ez nyilvánvalóan nagyobb, mint a topológiai dimenzió értéke (jelen esetben 0), ezért a Cantor halmaz is természetesen fraktál.

### 2.4.2. Koch-görbe

A Cantor halmaz konstrukciójánál minden szakasz középső harmadát távolítottuk el. Adódik a kérdés, hogy mi történik, ha nem eltávolítunk egy harmadot, hanem hozzáteszünk egyet. Így jutunk az ún. Koch-görbéhez (2.3). Nevét Helge von Koch svéd matematikusról kapta, aki 1904-ben megjelent írásában példaként hozza fel olyan görbére, amelynek semelyik pontjába nem húzható érintő. Számoljuk ki a görbe dimenzióját. Sejtésünk szerint ez egy 1 és 2 közötti érték kell, hogy legyen, és valóban,

$$D = \frac{\ln(4)}{\ln(3)} \approx 1.2619$$

### 2.4.3. Sierpinski-háromszög

*Lehetséges előállítási módok:*

1. Vegyünk egy tetszőleges méretű szabályos háromszöget. Rajzoljuk be a középvonalait. Ezek a szakaszok 4 kisebb (egybevágó) háromszögre osztják fel az eredetit. Ezek közül távolítsuk el a középsőt. A maradék hárommal ismételjük meg a fentieket. Végtelen iterációs lépés után az eredmény a Sierpinski-háromszög (2.4.ábra).
2. Mint később látni fogjuk, a Lindenmayer-Rendszer vonalas (görbeszerű) fraktálok leírására alkalmas. Talán meglepő, de a Sierpinski-háromszög ilyen módon is leírható. A pontos algoritmus a 22. oldalon kerül bemutatásra. Noha kevés számú iteráció után még nem "kitöltöttek" a háromszögek, végtelen számú ismétléssel ez a probléma orvosolható, így az eredmény ismét a Sierpinski-háromszög lesz.

3. "Káoszjáték". Vegyünk fel három pontot a síkon, úgy hogy azok egy egyenlő szárú háromszög csúcsait határozzák meg. Címkezzük fel ezeket a pontokat az 1,2,3 számokkal. Ezeket bázisoknak fogjuk hívni. Ezután kezdetét veszi a játék. Vegyünk fel egy tetszőlegesen kiválasztott pontot a három bázispont által meghatározott háromszögön belül. Ezt játékpontnak fogjuk nevezni. Majd újabb és újabb pontokat veszük fel, a következő szabály szerint. Sorsoljunk véletlenszerűen egy 1 és 3 közötti számot. Ezt könnyedén megtehetjük akár egy hatoldalú dobókocka segítségével is, ha például az egyes és négyes, a kettes és ötös, illetve a hármas és hatos értékű dobás között nem teszünk különbséget. Tegyük fel, hogy a kisorsolt szám az  $x$  volt. Ekkor kössük össze képzeletben a játékpontunkat az  $x$  cimkéjű bázisponttal, és vegyük fel új játékpontként az így kapott szakasz felezőpontját. Mivel a bázisokat minden újabb pont felvételekor véletlenszerűen választjuk, így azt várhatnánk, hogy a játékot néhányszor megismételve mindig más és más ábrát kapunk végeredményül. Ha elegendően sok pontot vettünk fel, akkor tisztán felismerhető lesz a Sierpinski-háromszög jellegzetes alakja.
4. Az első esetben síkidomokból, a másodikban vonalakból, a harmadikban pedig pontokból "építkeztünk". Most egy bizonyos értelemben számokkal tesszük ugyanezt. Induljunk ki a Pascal-háromszögből. Cseréljük le a páratlan számokat adott méretű, fekete színű egyenlő szárú háromszögekere, a páratlan számokat ugyanekkora, "fejük tetejére állított" fehér háromszögekre. Az így kapott alakzat egy Sierpinski-háromszög lesz.

#### 2.4.4. Sierpinski-szőnyeg

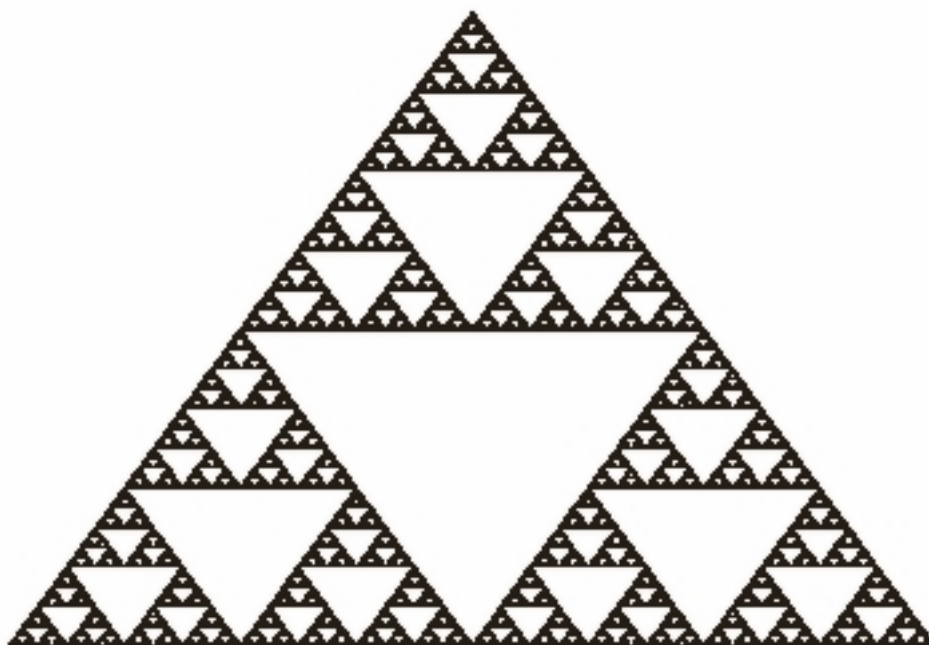
A Sierpinski-szőnyeg (2.5.ábra) a Cantor halmaz egyik lehetséges két dimenziós általánosítása (a másik a Cantor-por). Univerzális görbe, ami azt jelenti, hogy tetszőleges egydimenziós görbe síkra vetített képe homomorf a Sierpinski-szőnyeg egy részhalmazával. Mivel a harmadakkora részekből nyolcat hagyunk meg, ezért a szőnyeg dimenziójára

$$\frac{\ln(8)}{\ln(3)},$$

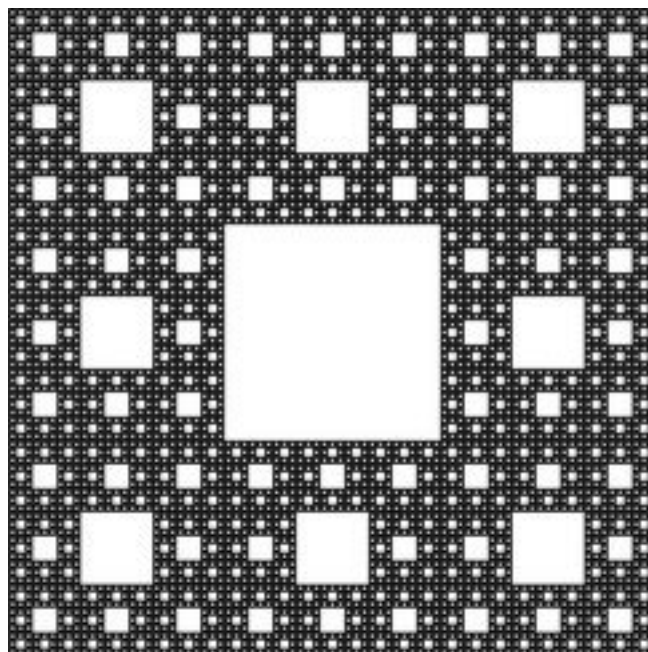
azaz 1.8928 körüli értéket kapunk.

#### 2.4.5. Menger-szivacs

A Menger-szivacs (2.6.ábra) esetén egy (3 dimenziós) kockából indulunk ki, tehát bizonyos értelemben ez a Cantor halmaz 3 dimenziós megfelelője. Rettenetesen érdekes tulajdonságokkal rendelkezik: Végtelen nagy a felszíne, 0 nagyságú a térfogata, emellett ún. univerzális görbe is, hiszen topológiai dimenziója 1 és bármilyen görbe homomorf a Menger-szivacs egy részhalmazával.



2.4. ábra. A Sierpinsi-háromszög

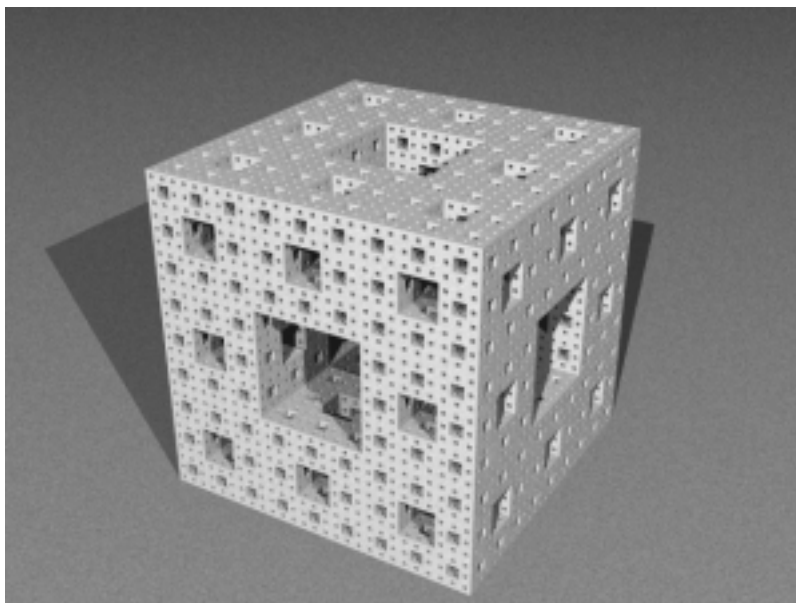


2.5. ábra. Sierpinski-szőnyeg

### 2.4.6. Julia halmazok

A Julia halmazok Gaston Julia (1893-1978) francia matematikusról kapták a nevüket, aki 1918-ban megjelent munkájában ismertette őket. Legyen adott a következő függvény:

$$f : \mathbb{C} \mapsto \mathbb{C}, f(z) = z^2 + c,$$



2.6. ábra. Menger-szivacs

ahol  $c \in \mathbb{C}$  adott paraméter. Iteráljuk  $f$ -et a következő módon:

$$f^0(z) = z$$

$$f^n(z) = f(f^{n-1}(z)), n \geq 1$$

Látható, hogy egy adott  $z \in \mathbb{C}$  értékkel indítva az iterációt két dolog történhet  $z$ -vel, ha  $n \mapsto \infty$ . Vagy elszökik a végtelenbe, vagy egy véges területen marad. Vezessük be a szökési halmaz fogalmát:

**Definíció**(Szökési halmaz) : Legyen  $c \in \mathbb{C}$  adott paraméter. Ekkor azon pontok halmazát, amelyek a végtelenbe szöknek az iteráció során, szökési halmaznak nevezzük, azaz:

$$E_c = \left\{ z \in \mathbb{C} : \lim_{n \rightarrow \infty} |f^n(z)| = +\infty \right\}$$

Most már definiálhatjuk a Julia halmaz fogalmát.

**Definíció**(Julia halmaz) : Legyen  $c \in \mathbb{C}$ . Az  $E_c$  szökési halmaz határát Julia halmaznak nevezzük, és  $J_c$ -vel jelöljük.

### 2.4.7. Madelbrot halmaz

**Definíció**(Mandelbrot halmaz) : Azon  $c$  komplex számok halmaza, amelyekre a  $z \mapsto z^2 + c$  Julia halmaza összefüggő, Mandelbrot halmaznak nevezzük, azaz:

$$M = \left\{ c \in \mathbb{C} : J_c \text{ összefüggő} \right\}$$

Megadható egy alternatív definíció is:

**Definíció**(Mandelbrot halmaz, 2.változat) :

$$M = \left\{ c \in \mathbb{C} : \lim_{n \rightarrow \infty} f_c^n(0) < \infty \right\}$$

Mitsuhiro Shishikura 1991-ben bizonyította, hogy a Mandelbrot-halmaz határának Hausdorff-dimenziója 2, míg a topológikus dimenziója 1, vagyis definíció szerint a Mandelbrot-halmaz határa fraktál. Ezt az állítást egyébként néhány évvel korábban Mandelbrot és Milnor is megsejtette. Korábban Elenbogen és Kaeding fizikusok 1989-ben azt állították, hogy Milnor sejtése hamis, de a bizonyításuk minden bizonnyal hibás volt. A bizonyítás részletei [9]-ben

## 2.5. A fraktálok formális leírásának módjai

A fraktálok végtelenül összetett matematikai alakzatok, amelyek minden (de legalábbis nagyon sok) mérettartományban rendelkeznek részletekkel, így a pusztán lerajzolásuknál sokkal precízebb leírásra van szükségünk, ha egzakt matematikai objektumokként akarjuk kezelni őket. Többféle leírási módszer került kidolgozásra, ám univerzális leíró eszközt (legalábbis ezideáig) nem sikerült megadni. Ezért a fraktálokat az alapján is osztályozhatjuk, hogy mely leíró rendszer segítségével jellemezhetőek (akár annak ellenére is, hogy vannak olyan fraktálok, amelyeket egyszerre több osztályba is besorolhatunk, ilyen például a Sierpinski-háromszög). A főbb osztályok:

- L-System fraktálok
- IFS fraktálok
- Komplex fraktálok
- Különös attraktorok



2.7. ábra. A Mandelbrot halmaz és néhány pontjához tartozó Julia halmaz

### 2.5.1. L-System fraktálok

1968-ban Aristid Lindenmayer, magyar származású elméleti biológus és botanikus alkotta meg a róla Lindenmayer-rendszernek, röviden L-Systemnek nevezett formális leírási módszert. Különböző növények növekedését vizsgálva rájött, hogy közülük igen sok leírható néhány egyszerű formális szabállyal. Ehhez csak egy megfelelő generatív nyelvtant kell rögzíteni.

#### A generatív grammatika:

$$G = \langle V, S, \omega, P \rangle, \text{ ahol}$$

- $V$  : ábécé, változók halmaza (nemterminálisok)
- $S$  : konstansok (terminálisok) halmaza
- $\omega$  : mondatszimbólum
- $P$  : levezetési szabályok halmaza

#### Példák:

##### Cantor-halmaz

- változók : A B
- konstans : nincs
- mondatszimbólum : A szabályok : (A  $\mapsto$  ABA), (B  $\mapsto$  BBB)

Az A (egyenes irányú, előre haladó) rajzolást, a B rajz nélküli haladást jelent.

##### Koch görbe

- változók : F
- konstansok : + -
- mondatszimbólum : F
- szabályok : (F  $\mapsto$  F+F-F-F+F)

Az F rajzolást, a + jel kilencven fokkal balra, a - jel pedig kilencven fokkal jobbra történő fordulást jelent.

##### Sierpinski háromszög

- változók : A B
- konstansok : + -
- mondatszimbólum : A
- szabályok :  $(A \mapsto B-A-B), (B \mapsto A+B+A)$
- fordulási szög:  $60^\circ$

Az A és B is rajzolást jelent, a + jel balra, a - jel jobbra fordulást az adott fordulási szöggel. A szög minden iteráció során előjelet vált.

### Fraktál növény

- változók : X F
- konstansok : + -
- mondatszimbólum : X
- szabályok :  $(X \mapsto F-[X]+X)+F[+FX]-X), (F \mapsto FF)$
- fordulási szög:  $25^\circ$

Az előzőekhez hasonlóan az F rajzolást, a - jel adott szöggel balra, a + jel jobbra fordulást jelent. Az X-hez semmilyen rajzolási művelet nem kapcsolódik, az ő szerepe a különleges forma kialakításában van. Az '[' jel menti az aktuális pozíció és szög értékeket, amelyek az ']' jellel visszatölthetők.

### 2.5.2. IFS fraktálok

Az IFS az Iterated Function System (Iterált függvényrendszer) kifejezés rövidítése. Egy IFS nem más, mint kontraktív,  $\mathbb{R}^2 \mapsto \mathbb{R}^2$  alakú transzformációk kollekcója, mely szintén egy leképezés. Az ilyen típusú leképezéseknek mindig van egy egyedi fixpontja, digitális képekre alkalmazva ez a fixpont általában egy fraktálkép. IFS-sel előállítható például a fent említett fraktálok nagy része, pl: Cantor halmaz, Sierpinski háromszög, és -szőnyeg, Koch-görbe. Az IFS-ekről bővebben a 3.fejezetben lesz szó, ahol a széleskörű alkalmazási lehetőségük egy újabb példáját mutatom be.

### 2.5.3. Komplex fraktálok

Mint ahogy a nevük is mutatja, a komplex számsíkon értelmezett fraktálokról van szó. Példaként említhetjük a korábban már bemutatásra került Julia halmazokat és a Mandelbrot hal-

mazt, vagy az úgynevezet Mágnes-fraktált.

#### **2.5.4. Különös attraktorok**

Különös attraktorra is láttunk már példát, de számos további is létezik, pl: Rössler-attraktor.

### **2.6. Káosz és fraktálok különböző tudományterületeken**

A káoszelmélet igazi transzdiszciplináris tárgy, nem köthető kizárólagosan egyetlen tudományterülethez sem, de talán nincs olyan terület, amelyet ne érintene. Íme néhány példa.

#### **2.6.1. Orvostudomány**

A testünkön belül is jó néhány példát találunk kaotikus folyamatokra, sőt úgy tűnik, ez a jellemző. Így aztán gyakran az okoz problémát, hogy ez a kaotikus viselkedés megszűnik, ilyenkor az orvosok feladata a káosz visszaállítása. Jó példa erre az epilepszia, ahol az agy normális kaotikus működésére való visszaállítás a cél. Az emberi test egyébként zsúfolásig töltve van fraktálokkal. Az élő szervezetekben a természetes szelekció kényszere által meglehetősen energia- és helytakarékos megoldások születtek. A legjellemzőbb a térkitöltő fraktálok jelenléte. Egy átlagos emberi tüdő felszíne például kb. fél teniszpálya nagyságú, mégis elfér a mellkasunkban.

#### **2.6.2. Populációdinamika**

A populációdinamika élőlények időbeli egyedszámváltozását vizsgálja egy adott élőhelyre vonatkozóan. Jellemzően nemlineáris egyenletekkel írható le, hogy mekkora lesz a populáció egyedszáma egy későbbi időpontban, és ezek a nemlineáris egyenletek gyakran mutatnak kaotikus viselkedést. Bizonyos rovarpopulációk esetében ezt laboratóriumi kísérletek is alátámasztják. Az élőlények versengésének dinamikája, valamint táplálékláncok és hálózatok viselkedése is hasonló kaotikus viselkedést mutathat.

#### **2.6.3. Genetika, posztgenetika:**

A modern genetika tudománya alig 100 éves múltra tekinthet vissza, mégis ilyen rövid idő alatt is hihetetlen eredményeket volt képes felmutatni. Ám, ahogy azt a neve is mutatja, az örök-lődési információkat tároló DNS-nek csak egy kizárólagos részével foglalkozik, nevezetesen a génekkel. A természet a DNS-szekvenciának csak egy kis darabját használja a gének kódolásá-ra, a fennmaradó rész – a többség – funkcióját a mai napig nem sikerült tisztázni. Sok szakember gondolta (és gondolja) úgy, hogy a DNS ezen darabjai semmit sem kódolnak, így aztán maguk

közt csak „szemétnek” (angol terminológiával junk DNA-nek) nevezték őket. Nemrég kiderült viszont, hogy nem csak a gének, hanem az eddig feleslegesnek tartott DNS-részletek is fontos információkat kódolnak, csak hogy ezeket más módon lehet kiolvasni, a lineáris feldolgozás ebben az esetben mit sem ér. Az elismert magyar informatikus, Pellionisz András és munkatársai fraktál algoritmusok segítségével fraktál-struktúrát mutató, ismétlődő, önhasonló részeket találtak a DNS szekvenciában. A fraktál-alakzatok használata kitűnő módszer az információ tömörítésére, így nem csoda, ha az öröklődő információ tárolásáért felelős genom szintén ilyen típusú elrendeződést követ.[3] Így aztán az sem túl meglepő, hogy az igazi áttörést egy informatikus munkája hozta meg.

#### 2.6.4. Információelmélet

A káosz információt teremt..... Figyeljük meg a következő számsort:  $0, 0, \dots$  Az sorozat az első elem után semmilyen új információval nem szolgál. A következő sorozat egy fokkal jobb:  $0, 1, 0, 1, \dots$ , hiszen a sorozat első eleme után nem állíthatjuk, hogy ismerjük a sorozat összes elemét, a második után viszont már igen (legalábbis bizonyos értelemben). És most vegyünk egy kaotikusan változó számsort. Nyilvánvaló, hogy eme számsor információtartalma végtelen, hiszen a sorozat bármennyi eleme ismeretében sem tudjuk megjósolni, hogy mi lesz a következő elem. Vajon lehetséges, hogy amikor egy rendszer kaotikus állapotba kerül, akkor információ keletkezik? Norman Packard szavait idézve: „A bonyolult dinamika csúcán a biológiai evolúció vagy a tudat folyamatai állnak. Intuitíve világos feltételezésnek tűnik, hogy ezek a végletesen bonyolult rendszerek információt állítanak elő. Milliárd évekkkel ezelőtt csak protoplazmacseppek léteztek, s most, milliárd évek elteltével már mi is jelen vagyunk. Egyszóval az információ a struktúránkban keletkezett és tárolódott. Ahogyan tehát a gyermekkortól kezdve fejlődik az ember tudata, nemcsak kész információ halmozódik fel benne, hanem új is keletkezik, és pedig korábban nem létezett kapcsolatok révén.”

#### 2.6.5. Matematika

Ahogy azt pl. a Koch-görbe estében is láttuk, az első fraktálokat, mint extrém tulajdonságú objektumoknak, ellenpéldáknak, matematikai „szörnyetegeknek” tekintették. Később kiderült, hogy bizonyos értelemben ezek az alakzatok, sokkal természetesebbek, mint a matematikában addigra széleskörben elterjedt társaik. Mivel a természetben sokkal gyakrabban fordulnak elő fraktálszerű görbék, viszont hagyományos eszközökkel elég nehezen írhatóak le, új módszerek kidolgozására volt szükség. Kiderült, hogy az olyan „hagyományos” görbék is, mint a Bézier-görbék, kezelhetők fraktálként. (Egy igazán jó áttekintése a témának [7]-ben olvasható.)

Egy másik alkalmazási mód a korábban tárgyalt IFS-eken alapul. A harmadik fejezetben ar-

ra láthatunk majd példát, hogy hogyan lehet iterált függvényrendszerek segítségével digitális képeket tömöríteni. Ha jobban meggondoljuk, ez a felhasználási mód nem is annyira újszerű, a matematikusok már elég régen rájöttek, hogy lehet végtelen mennyiségű információt néhány bájtba sűríteni (legalábbis bizonyos értelemben). Vegyük példaként a 2 négyzetgyökét! Mivel  $\sqrt{2}$  irracionális szám, ezért végtelensok számjegyre van szükségünk, hogy egészen pontosan megadjuk az értékét. Vagy választhatunk egy jóval egyszerűbb (és helytakarékosabb), módszert – keresünk egy iterált függvényt, amelynek fixpontja épp a leírni kívánt  $\sqrt{2}$ :

$$\sqrt{2} \quad : \quad x_{n+1} = \frac{1}{2} \left( x_n + \frac{2}{x_n} \right), \quad x_0 > 0$$

Erre az iterált függvényre úgy is tekinthetünk, mint egy végtelen kapacitású információ-tároló berendezésre, amelyből bizonyos sebességgel a  $\sqrt{2}$  számjegyei szekvenciálisan kiolvashatók.

Egyébként a tisztán matematikai fogalmak körében is találunk példát önhasonlóságra.

Példaként álljon itt a mértani sor skálázással szembeni invarianciája:

Induljunk ki a mindeki által ismert mértani sorból:

$$\sum_{k=0}^{\infty} q^k = 1 + q + q^2 + q^3 + \dots$$

Skálázzuk ezt a sort  $q$  faktorral:

$$q \sum_{k=0}^{\infty} q^k = q + q^2 + q^3 + q^4 + \dots$$

Így aztán:

$$\sum_{k=0}^{\infty} q^k = 1 + q \sum_{k=0}^{\infty} q^k$$

Vagyis úgy kaphatjuk meg a mértani sor összegét, hogy 1-hez hozzáadjuk az eredeti sor skálázott változatát.

## 2.6.6. Informatika

### 2.6.6.1. Alapvető adatszerkezetek

Az informatikában két legáltalánosabban használt homogén adatszerkezet minden bizonnyal a láncolt lista és a bináris (vagy többágú) fa. Könnyű belátni, hogy ezek szintén fraktálstruktúrával rendelkeznek. A fa adatszerkezet esetében ez nyilvánvaló is, hiszen éppen a természetben előforduló növények alakjának kellemes tulajdonságait szerették volna az informatikusok át-

ültetni mesterséges környezetbe (külön tudományág foglalkozik a természet „másolásával”, a neve bionika). Az önhasonlóságot használjuk ki rekurzív algoritmusok használatakor is, gondoljunk csak a bináris fák különböző rekurzív bejárési módjaira, ahol egy bináris fa két ágát mint két bináris fát dolgozunk fel.

#### 2.6.6.2. Számítógépes szimulációk

Mivel a természeti képződmények többsége fraktáljellegű, triviálisnak tűnik, hogy virtuális világok létrehozásakor a modellezés eszközeül fraktálokat használjunk. A legújabb számítógépes játékok többsége ki is használja az egyszerű, rövid formulával megadható, természetes hatást keltő fraktálszerű alakzatok által kínált lehetőségeket. Hogy az ily módon előállított képek mennyire természetes hatást keltenek, jól tükrözi a 2.8. ábra is, amely egy ilyen technikával elkészített tájképet mutat be.



2.8. ábra. Fotorealisztikus fraktál-tájkép

### 2.6.6.3. Álvéletlenség generálás

A kaotikus rendszerek alapvető tulajdonsága, hogy hosszú távú időbeli viselkedésük előre gyakorlatilag nem jósolható meg, másrészt a kezdeti feltételek igen csekély eltérése is nagyon különböző időbeli viselkedést (dinamikát) okoz. Ezen tulajdonságaikat akár az előnyünkre is fordíthatjuk. Egy álvéletlenség generátor annál jobb, minél hosszabb periódusú számsort tud előállítani. Ám idővel az ismétlődés elkerülhetetlen. A kaotikus viselkedést a periodicitás teljes hiánya jellemzi, ezért alkalmas lehet véletlen számsorok előállítására. Erre a célra akár az 1.2.1.1. fejezetben tárgyalt logisztikus leképezés bizonyos általánosításait is felhasználhatjuk (az ún.  $B$ -exponenciális leképezéseket, [8]).

### 2.6.6.4. Kriptográfia

A kaotikus rendszerek előző pontban említett tulajdonságait üzenetek titkosítási módszereinek javítására is felhasználhatjuk. Megtehetjük például, hogy a továbbítandó jelhez egy kaotikus jelet adunk hozzá. Egy ilyen elven működő kód feltörése igen nehéz feladat, hiszen még ha ismerjük is a kaotikus jelet létrehozó dinamikai rendszert, a kaotikus jelet létrehozó kezdeti feltételek pontos ismeretének hiányában az eredeti üzenetet nem lennénk képesek visszafejteni.

### 2.6.6.5. Szteganográfia

Az adatrejtés, idegen szóval szteganográfia alapfeladata, hogy úgy rejtse el információt valamilyen hordozó rétegbe (pl. kép, hang, szöveg, stb.), hogy egy illetéktelen számára már a titok puszta létezésének ténye se legyen nyilvánvaló. A képekbe történő információrejtés többféle módja ismert, általános megoldás, hogy a kép egyes pixeleit sorban érintve azok valamilyen paraméterét kismértékben megváltoztatják. Ilyen esetekben célravezetőbb lehet egy térkitöltő görbe által definiált képbejárás algoritmus. Az ilyen módon megváltoztatott képpontok foltszerűen helyezkednek el, amely sokkal kevésbé feltűnő változást okoz, mint a hagyományos, szekvenciális elrendezés.

### 2.6.6.6. Digitális képfeldolgozás

A térkitöltő görbék előbb említett kellemes tulajdonsága miatt célszerűbb lehet egy képet egy térkitöltő görbe mentén bejárni, így pl. több azonos pixelérték kerülhet egymás mellé a bejárás során, ami (különösen egyes tömörítési módszerek esetén) előnyösebb lehet. A fraktálok hasonló jellegű egyéb felhasználási módjaival egy külön fejezet foglalkozik.

## 3. fejezet

# Fraktál alapú képtömörítés

Egy szó többet mond ezer szónál, szoktuk gyakran mondani. Nos, általában jóval több helyet is foglal, egy  $1024 \times 768$ -as felbontású, 32 bites színmélységű tömörítetlen kép mérete közel 3 megabájt tárhelyet foglal el. Így aztán manapság, amikor már a legolcsóbb mobiltelefonokban is beépítet megapixeles kamerát találunk, fontos, hogy mind újabb, nagyobb hatásfokú adattömörítési (elsősorban kép- és hangtömörítési) módszereket alkossunk. Jobb tömörítési arány érhető el, ha a tömörítés során előállt képtől nem várjuk el, hogy pontos mása legyen az eredetinek. Egészen kis eltérésekről van szó, akár olyanokról is, amelyek az emberi szem számára láthatatlanok maradnak. Az ilyen elven működő módszereket veszteséges tömörítőknek nevezzük. Jelenleg három technológia uralkodik a veszteséges tömörítés területén : a vektor kvantálásnak nevezett módszer, a diszkrét koszinusz transzformáció (DCT), és a fraktál alapú képtömörítés. Az első módszer esetén a képet kisebb részekre osztják, és egy ún. kódkönyvből (codebook) választanak hozzá megfelelő reprezentánst. A diszkrét koszinusz transzformáció a képeket egy másik térbe (az ún. frekvenciatérbe) konvertálja, majd az így kapott értékeket megfelelő módon kvantálja. A harmadik esetben a természetben előforduló alakzatok (bizonyos mértékű) önhasonlóságát használja ki. 1988 Barnsley kidolgozott egy módszert, amely digitális képek jó hatásfokú tömörítését tette lehetővé IFS-ek felhasználásával. A módszert sokan továbbfejlesztették, jelenleg is igen sok változata ismert. A technika még csak gyerekcipőben jár, a lehetőségek majdhogynem beláthatatlanok. Barnsley alapgondolata az volt, hogy ha néhány nagyon egyszerű transzformáció segítségével bonyolult, a természetben is előforduló alakzatok sokasága állítható elő, akkor ez felhasználható az ilyen alakzatokat ábrázoló képek tömörítésére, hiszen a képpontok helyett elegendő a képet előállító transzformációkat eltárolni, ami drasztikusan lecsökkenti a szükséges tárhely méretét. Azt is észrevette, hogy az egyes iterációk fixpontja semmilyen módon nem függ a kezdeti képtől, így az iteráció tetszőleges kiindulási képpel elindítható. A kérdés csak az, hogy tetszőleges kép előállítható-e IFS-ek segítségével, vagy meg kell elégednünk bizonyos képek nagyarányú tömörítésének lehetőségével. Sajnos a válasz nemleges, vagyis a képek többségéhez nem adható olyan IFS, amelynek fixpontja maga a kép lenne.

Ez logikus is, hiszen pl. egy emberi arc egyáltalán nem önhaonló, tehát a szerkezete nem fraktálszerű. Márpedig az IFS-ek segítségével leginkább fraktálokat lehet előállítani (vagy csakis azokat, ennek eldöntéséhez pontosan kellene definiálnunk a fraktál fogalmát – pl. fraktál-e egy négyzet?). Szerencsére van megoldás: osszuk fel a képet egyforma méretű kisebb tartományokra, és ezen tartományokat próbáljuk meg összevetni egymással. Ezt a technikát Arnaud Jacquin dolgozta ki, és az ehhez felhasznált speciális iterált függvényrendszereket PIFS-nek (Partitioned Iterated Function System) nevezik. A módszerek jól kidolgozott matematikai alpokkal rendelkeznek, tekintsük most át ezeket.

## 3.1. Matematikai alapok

### 3.1.1. Fraktálok

A továbbiakban egy  $F$  halmazt akkor nevezünk fraktálnak, ha teljesíti az alábbi feltételek (majdnem) mindegyikét:

- (1)  $F$  minden mérettartományban rendelkezik részletekkel.
- (2)  $F$  (pontosan, nagyjából vagy statisztikusan) önhasonló.
- (3)  $F$  fraktáldimenziója nagyobb, mint a topológiai dimenziója.
- (4) Létezik egy igen egyszerű, rövid algoritmus, mely  $F$ -et előállítja/jellemzi.

### 3.1.2. Iterált függvényrendszerek

**Definíció**(Iterált függvényrendszer): Legyen  $X$  teljes metrikus tér. Kontraktív transzformációk egy  $\omega : X \mapsto X$  ( $i = 1, \dots, n$ ) kollekciónak iterált függvényrendszernek nevezzük.

Iterált függvényrendszerek megadásakor ún. affin-transzformációkat használunk.

**Definíció**(Két dimenziós affin-transzformáció): Az

$$\omega_i \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} a_i & b_i \\ c_i & d_i \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} e_i \\ f_i \end{bmatrix}$$

alakú transzformációkat affin transzformációknak nevezzük, ahol az  $a_i, b_i, c_i, d_i, e_i, f_i$  konstanst jelöl.

A képekre mint metrikus terek részhalmazaira fogunk tekinteni, ezért tekintsük át az ide vonatkozó fogalmakat.

### 3.1.3. Teljes metrikus terek

**Definíció**(metrikus tér): Egy  $X$  halmazt metrikus térnek nevezünk, ha adott egy  $d : X \times X \mapsto \mathbb{R}$  valós értékű függvény, mely rendelkezik az akábbi négy tulajdonsággal:

1.  $d(a, b) \geq 0$  minden  $a, b \in X$ -re
2.  $d(a, b) = 0$  akkor és csak akkor, ha  $a = b$  minden  $a, b \in X$ -re
3.  $d(a, b) = d(b, a)$  minden  $a, b \in X$ -re
4.  $d(a, c) \leq d(a, b) + d(b, c)$  minden  $a, b, c \in X$ -re (háromszög-egyenlőtlenség)

A  $d$  függvényt metrikának nevezzük.

#### Hausdorff metrika

Két kép közötti távolság mérésére az ún. Hausdorff-metrikát fogjuk használni. Legyen az  $A$  és  $B$  egy metrikus tér két részhalmaza (azaz legyen kép), ekkor a  $h(A, B)$ -vel jelölt Hausdorff-távolságot a következő módon számolhatjuk ki: Minden  $x \in A$  pontra keressük meg az összes  $y \in B$  pontal vett távolságok minimumát (itt pontok közötti távolságság alatt a „hagyományos” euklideszi távolságot értjük). Számoljuk ki ezeket a minimális távolságokat mind az  $A$ , mind a  $B$  halmazból kiindulva, és vegyük ezek maximumát. Ez lesz az  $A$  és  $B$  halmaz Hausdorff-távolsága.

**Definíció** (Cauchy-sorozat): Egy metrikus térben értelmezett  $a_n$  pontsorozatot Cauchy-sorozatnak nevezünk, ha bármely  $\epsilon > 0$  esetén létezik egy  $n_0$  küszöbszám, melyre  $d(a_m, a_n) < \epsilon$  teljesül  $\forall n, m > n_0$  indexre.

**Definíció** (Teljes metrikus tér): Egy  $(X, d)$  metrikus teret teljesnek nevezünk, ha benne minden Cauchy-sorozat konvergens.

**Definíció** (Kompakt metrikus tér): Egy metrikus tér kompakt, ha zárt és korlátos.

Ezek után megadhatjuk azt a teret, amelyben dolgozni szeretnénk. Legyen

$$\mathcal{H}(X) = \{S \subset X \mid S \text{ kompakt}\}.$$

Tehát a  $\mathcal{H}(X)$  halmaz az  $X$  összes lehetséges kompakt részhalmazainak halmaza. Jelölje tetszőleges  $A \in \mathcal{H}(X)$  esetén  $A_d(\epsilon)$  az  $A$  halmaz pontjaitól legfeljebb  $\epsilon$  távolságra levő pontok

halmazát, azaz legyen  $A_d(\epsilon) = \{x | d(x, y) \leq \epsilon, y \in A\}$ . Ezután az  $A, B \subset \mathcal{H}(\mathcal{X})$  halmazok Hausdorff-távolságán a következőt értjük:

$$h_d(A, B) = \max\{\inf\{\epsilon | B \subset A_d(\epsilon)\}, \inf\{\epsilon | A \subset B_d(\epsilon)\}\}$$

Vegyük észre, hogy  $h_d$  értéke függ a  $d$  metrikától. A gyakorlatban gyakran elhagyják a  $d$  indexet, ha a szövegkörnyezetből világosan kiderül, hogy melyik metrikáról van szó.

**Tétel:** Legyen  $X$  teljes metrikus tér  $d$  metrikával. Ekkor  $\mathcal{H}(X)$  is teljes metrikus tér lesz  $h_d$  metrikával.

### 3.1.4. Kontraktív leképezések és IFS-ek

A célunk az, hogy tetszőleges kezdeti képből kiindulva bizonyos számú iterációs lépés után egy egyértelműen meghatározott fixponthoz érkezzünk, és ez a fixpont maga a kódolandó kép legyen. Az ún. kontraktív leképezések fixpont tétele garantálja nekünk az egyértelmű fixpont létezését, csak hogy valószínűtlen, hogy ez a fixpont egybeessen az eredeti képpel, hiszen ehhez az eredeti képnek tökéletesen önhasonlónak kellene lennie, ami a gyakorlatban igen ritkán teljesül, az  $S$  kódolandó kép helyett a fixpont  $S'$  lesz, ahol az  $S'$ -től csak azt várhatjuk, hogy elegendően közel legyen  $S$ -hez (vagyis a két kép távolsága egy bizonyos hibaérték alatt legyen). Emiatt lesz a tömörítés veszteséges. De az ún. kollázs-tétel ilyen esetekben is biztosít minket arról, hogy minél jobban illeszkedik az eredeti  $S$  képre a megfelelő transzformációkkal összeállított  $W(S)$ , „kollázs”, annál közelebb lesz az  $x_w$  attraktor  $S$ -hez. A matematiai megalapozás:

**Definíció** (Lipschitz-leképezés): Legyen  $(X, d)$  metrikus tér. Egy  $w : X \mapsto X$  leképezést  $s$  Lipschitz faktorú Lipschitz-leképezésnek nevezünk, ha létezik olyan pozitív valós  $s$  érték, hogy:

$$d(w(x), w(y)) \leq sd(x, y)$$

minden  $x, y \in X$  esetén. Ha  $s < 1$ , akkor a  $w$  leképezést  $s$  kontraktivitású kontraktív leképezésnek nevezzük.

**Lemma:** Ha a  $w : X \mapsto X$  leképezés Lipschitz-féle, akkor  $w$  folytonos.

**Tétel:** Ha  $w_i : \mathbb{R} \mapsto \mathbb{R}$   $s_i$  kontraktivitású kontraktív leképezés ( $i = 1 \dots n$ ), akkor a

$$W = \bigcup_{i=1}^n w_i : \mathcal{H}(\mathbb{R}^2) \mapsto \mathcal{H}(\mathbb{R}^2)$$

is  $s = \max_{i=1..n}\{s_i\}$  kontraktivitású kontraktív leképezés a Hausdorff-metrikára nézve.

**Tétel**(Kontraktív leképezések fixpont tétele): Legyen  $X$  teljes metrikus tér és  $f: X \mapsto X$  egy kontraktív leképezés. Ekkor létezik egy egyértelmű  $x_f \in X$  pont, melyre igaz, hogy minden  $x \in X$  esetén

$$x_f = f(x_f) = \lim_{n \rightarrow \infty} f^{o_n}(x).$$

Az  $x_f$  pontot az  $f$  leképezés *fixpontjának* vagy *attraktorának* nevezzük.

**Tétel**(Kollázs-tétel): Az előző tétel jelöléseit megtartva érvényes a következő:

$$d(x, x_f) \geq \frac{1}{1-s} d(x, f(x))$$

Tehát a kódolás menete a következő: Adott egy  $f$  kép, amit kódolni szeretnénk. Osszuk fel a képet értelmezési tartományokra és értékkészletekre. Ezt többféle módon megtehetjük, vagy fix méretű blokkokat használunk, vagy megadhatjuk a képnek az ún. 4-fa reprezentációját. Ennek előnye, hogy változó méretű blokkokat használ, ezzel is növelve az illeszkedés valószínűségét (hiszen lehet, hogy a két képrészlet különböző mérettartományban ugyan, de nagyjából ugyanazt ábrázolja). Ekkor keressük meg minden értékkészlethez azt az értelmezési tartományt és transzformációt, amelyre a transzformáció eredménye és az értékkészlet között az eltérés minimális, és egy bizonyos küszöb alatt van. Ha a második feltétel nem teljesül, a akkor 4-fa felbontást alkalmazó változatban az értékkészletet tovább bontjuk, és újra próbáljuk lefedni valamely értelmezési tartomány és transzformáció segítségével. Ha tovább nem tudjuk bontani, és még mindig hibaküszöb felett vagyunk, akkor kénytelenek leszünk a minimális hibát eredményező transzformációt elfogadni. Ha sikerült minden értékkészletet a megfelelő transzformációk segítségével lefedni, akkor az eltárolt transzformációk reprezentálják a kódolni kívánt képet. A dekódolás ennél sokkal egyszerűbb, induljunk ki egy tetszőleges képből, iteráljuk a PIFS-t, amíg el nem jutottunk a fixponthoz, azaz amíg  $x_f = W(x_f)$  nem teljesül. Ez az  $x_f$  fixpont lesz a dekódolt kép.

## 3.2. Az algoritmus javításai, egyéb változatai

A PIFS eszközrendszert használó fraktál alapú képtömörítés egyetlen nagy hátránya, hogy borzasztóan nagy a kódolás számítási igénye (a dekódolásnál ilyen probléma már nem jelentkezik, tehát ilyen tekintetben felveszi a versenyt az egyéb megvalósításokkal). Ennek oka, hogy minden egyes értékkészlet esetén elképesztően sok értelmezési tartomány jelöltet kell megvizi-

gálni, hogy megtaláljuk a számunkra legmegfelelőbbet. Ezen úgy segíthetünk, hogy vagy valamilyen heurisztikát használunk, vagy a különböző összehasonlításokat párhuzamosan végezzük el. Különböző javított megoldások születtek, ezek közül mutatok be néhányat:

### 3.2.1. Klasszifikáció

Az egyik lehetőség a szükséges összehasonlítások számának drámai csökkentésére a lehetséges értelmezési tartományok osztályokba sorolása. Kódoláskor minden egyes értékkészletet csak a vele egy osztályba tartozó értelmezési tartományokkal hasonlítunk össze. Számos módszer létezik az osztályozás megvalósítására, egy viszonylag egyszerű megoldás a következő. Minden (négyzet alakú) értelmezési tartományt négy egyforma részre osztunk, és az egyes részekhez indexeket rendelünk, úgy, hogy a bal felső négyzet az egyes, a jobb felső a kettes, a bal alsó a hármas, a jobb alsó pedig a négyes index értéket kapja. Ezek után minden negyedhez hozzárendelünk egy számértéket az alábbi módon. Jelöljük az  $i$ . negyedben az egyes pixelértékeket  $p_{i1}, \dots, p_{in}$  -nel ( $i = 1 \dots 4$ ), ekkor legyen  $A_i = \sum_{j=1}^n p_{ij}$  továbbá: Másik képlet Figyelembe véve, hogy kódoláskor az értelmezési tartományok elforgathatóak  $90^\circ$  többszöröseivel, az  $A_i$  értékek három különböző csökkenő sorrendje fordulhat elő:

1. alaposztály :  $A_1 \geq A_2 \geq A_3 \geq A_4$
2. alaposztály :  $A_1 \geq A_2 \geq A_4 \geq A_3$
3. alaposztály :  $A_1 \geq A_4 \geq A_2 \geq A_3$

Mivel az egyes  $V_i$  értékek összes lehetséges sorrendjeinek száma  $4!$  azaz 24, ezért mindhárom alaposztályhoz további 24 alosztály tartozik, így az osztályok száma összesen 72 lesz.

### 3.2.2. Archetípusok

Ebben az esetben is szükség van osztályozásra. Egy archetípus nem más, mint egy adott osztály azon eleme, amellyel a legjobban lefedhető az osztályban lévő összes többi elem. Az archetípusok meghatározása nagyon hasonlít a vektor kvantáláson alapuló képtömörítés codebook-jának meghatározásához, az alapötlet legalábbis ugyanaz. Az eredmények azt mutatják, hogy ezzel a technikával (is) jelentősnek mondható gyorsulás érhető el.

### 3.2.3. PIFS és neurális hálók

A neurális hálók előnye, hogy használatukkal nagyfokú párhuzamosság érhető el. Ez különösen hasznos olyan számításigényes feladatoknál, mint a PIFS alapú képtömörítés. Az elv a következő. Minden egyes pixelhez egy neuront rendelünk hozzá oly módon, hogy az adott

pixel intenzitásértékét a neki megfelelő neuron állapota határozza meg. Az eljárás során a kódolni kívánt képről másolatot készítünk, majd mindkét példányt tartományokra - az egyiket értelmezési tartományokra, a másikat értékkészletekre - osztjuk fel. Ezután minden értelmezési tartománybeli pixelhez egy input és minden értékkészletbelihez egy output neuront rendelünk. Minden  $j$  output neuronhoz 4 input neuron kapcsolódik, legyenek ezek  $i, i + 1, i + 2$  és  $i + 3$ . A  $j$  output neuron  $z_j$  kimeneti értéke a  $Z_i, Z_{i+1}, Z_{i+2}, Z_{i+3}$  értékekből, az ezekhez tartozó  $W_{ji}, W_{ji+1}, W_{ji+2}, W_{ji+3}$  súlyokból, és a  $\theta_j$  küszöbértékből számítható ki. Két különböző aktivációs függvény használatos, ez alapján beszélhetünk lineáris és nemlineáris modellről:

Lineáris modell esetén:

$$z'_j = O_j \left( \sum_{k=i}^{i+3} w_{jk} \times z_k - \theta_j \right)$$

Nemlineáris modell esetén:

$$z'_j = O'_j \left( \frac{1}{B} \sum_{k=i}^{i+3} w_{jk} \times z_k - \theta_j \right)$$

Ahol  $B$  a maximális lehetséges intenzitásérték (leggyakrabban 255).

Általánosan elmondható, hogy a nemlineáris modell sokkal robusztusabb, flexibilisebb, jobb képminőséget eredményez, a lineáris modell viszont egyszerűsége folytán kevésbé számításigényes.

#### Az eljárás lépései a következők:

- (1) Válasszunk egy  $e_c$  hiba-küszöbértéket és egy  $r_m$  minimális méretet az értékkészletekhez.
- (2) Osszuk fel a képet egymást nem átfedő részekre. Ezek lesznek az kiindulási értékkészletek. A kiindulási méretet legyen  $32 \times 32$  pixel.
- (3) Minden  $i$  értékkészlethez keressünk egy olyan (nála négyszer nagyobb méretű)  $j$  értelmezési tartományt, amelynél a az  $i$  és a  $j$  közti hiba legfeljebb  $e_c$  nagyságú. Adjuk meg az  $i$ -hez tartozó transzformációs függvényt (lásd fent), és frissítsük a  $w_{ij}$  súlyokat minden  $i$  és  $j$ -beli pixelre, használjuk a delta tanulási szabályt a transzformációs függvényben szereplő kontraszt és fényességérték kiszámításához, így a hiba minimális lesz.
- (4) Ha adott értékkészlethez nem találunk értelmezési tartományt (vagyis nem találunk olyan tartományt, amelyhez tartozó hibaérték legfeljebb  $e_c$ ), akkor osszuk fel az értékkészletet 4 egyforma nagyságú részre, amelyek mérete még nem kisebb  $r_m$ -nél, és a (3)-as pontnak megfelelően adjuk meg ezen résztartományok transzformációs függvényeit.

- (5) Ha az értékkészletet nem tudjuk további részekre osztani úgy, hogy azok mérete legalább  $r_m$  nagyságú legyen, akkor a legkisebb hibát produkáló értelmezési tartományt kell választanunk még akkor is, ha a hiba mérete meghaladja az előírt  $e_c$  küszöbértéket.

### 3.2.4. PIFS és genetikus algoritmusok(GA)

Az ún. lágyszámítási módszerek egy másik népszerű csoportját a genetikus algoritmusok alkotják. Előnyük, hogy robusztusak, flexibilisek, bizonytalan és pontatlan információs környezetben is eredményesen használhatóak, és a párhuzamos többpontos keresésnek köszönhetően képesek közel optimális megoldást találni akkor is, ha a lehetséges értékek tere hatalmas, és sűrűn tartalmaz lokális optimumokat. Mindezen tulajdonságai teszik különösen alkalmassá a fraktál alapú képtömörítésben való használatra ([1],[2]). A genetikus algoritmusok három alapvető operátora a szelekció, a mutáció és a keresztezés. Ennek megfelelően egy általános genetikus algoritmus felépítését a 3.1. folyamatábra mutatja be.

A genetikus algoritmusok területén kiemelt fontosságú a szkémátétel. A szkémák egyszerű karakterláncok (sztringek), amelyekben 3-féle karakter fordulhat elő: 0, 1, és #. A # jel felfogható ún. Joker-karakternek, egyaránt jelenthet 1-est vagy 0-t. Minden szkéma egy-egy hipersíkot jelöl ki a keresési térben.

Tétel(szkémátétel) : Egy az átlagosnál jobb költséggel rendelkező rövid szkéma exponenciálisan gyakrabban fordul elő a következő generációban, egy átlagosnál rosszabb költségű szkéma pedig exponenciálisan ritkábban.

Ezt kihasználva a megfelelő blokkok kiválasztását gyorsan és hatékonyan tudjuk megtenni. A módszer alapelemei:

- (1) *Kromoszómák*: Minden kromoszóma 19 bitből áll, 8-8-3-as felosztásban, az első tizenhat biten az adott értelmezési tartomány abszolút pozícióját, a maradékon a transzformációt meghatározó információt tároljuk.
- (2) *Fitness-függvény*: Mivel a tartományok közti távolság mérésére a legkisebb négyzetes hibát használjuk, ezért fitness-értéknek tökéletesen megfelel ennek reciproka.
- (3) *Kezdeti populáció*: A kromoszómákat véletlenszerűen állítjuk össze.
- (4) *Szelekció*: A jóság mértéke alapján szelektálunk.
- (5) *Keresztezés*: A populációt a kiszámított fitness-értékek alapján két csoportra bontjuk. Az  $S_b$  jelűben a jobb, az  $S_w$ -ben a rosszabb fitness-értékkel rendelkező kromoszómák kapnak helyet. Mivel az  $S_b$  halmazban található kromoszómák mind jó génekkel rendelkeznek,



3.1. ábra. Egy általános genetikus algoritmus működése

ezért érdemes ezeket egymással keresztezni. Mivel az  $S_w$ -beli kromoszómáknak immáron nem sok hasznát vesszük (hisz kicsi a valószínűsége, hogy közvetlen környezetükben alkalmas kromoszómákat találunk), nyugodtan lecserélhetjük őket az  $S_b$ -beli kromoszómák keresztezése során létrejövő új generációra. A  $P_c$  valószínűség határozza meg, hogy adott helyzetben történjen-e keresztezés.

- (6) *Mutáció:* A természetes képek alapvető tulajdonságait figyelembe véve két mutációs stratégia használatos. A  $S_b$  halmaz bármely kromoszómája esélyes a végső győzelemre. Ebből viszont az következik, hogy az ő közeli szomszédai szintén esélyesnek számítanak, így az értelmezési tartományok abszolút koordinátáinak felső bitjei jó hatásfokú sémákat határozhatnak meg. Belátható, hogy ugyanahhoz a sémához tartozó kromoszómák bizonyos fokig ugyanazt az információt tartalmazzák, ezért egy jó séma minden kromoszómája jó fitness-értékkel bír, így ezt a tulajdonságot megőrzendő az értelmezési tartomány koordinátáinak alsó biteit egy előre rögzített valószínűséggel mutálni fogjuk. Az így kapott leszármazottakat viszont a koordináták felső bitjein és a transzformációt meghatározó biteken érdemes mutálni, hiszen csak így tudjuk garantálni, hogy az algoritmus a teljes keresési teret bejárja..
- (7) *Megállási feltétel:* Befejezzük a keresést, ha elértünk egy előre rögzített iterációs szám-küszöböt, vagy ha ugyanaz a kromoszóma több alkalommal is a legjobbnak bizonyult.

#### **Ezek után a kódolás folyamata a következő :**

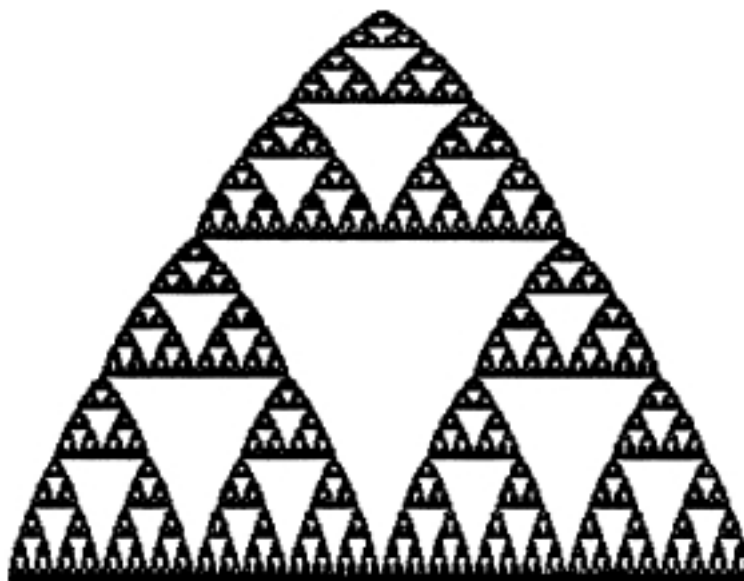
1. Kezdeti lépésként válasszunk egy populációméretet, egy keresztezési maszkot, egy  $P_c$  keresztezési arányt, egy mutációs maszkot, és a  $P_{m,b}$  és  $P_{m,w}$  mutációs rátákat.
2. Generáljunk véletlenszerűen egy kiindulási kromoszóma-populációt.
3. Számoljuk ki ezen kromoszómák fitness-értékeit.
4. Osztályozzuk a kromoszómákat a fitness-értékeiknek megfelelően.
5. Ha elértünk egy előre meghatározott iterációs szám-küszöböt, vagy ha ugyanaz a kromoszóma egymás után többször is legjobbnak bizonyul, akkor jegyezzük fel a neki megfelelő fraktálkódot, és stop. Egyébként folytassuk a következő pontban.
6. A jóságuk alapján soroljuk be a kromoszómákat az  $S_b$  és  $S_w$  halmazokba.
7. Alkalmazzunk egy egyenletes keresztezést az  $S_b$ -beli elemekre, majd az így kapott leszármazottakkal írjuk felül  $S_w$  elemeit.
8. Alkalmazzunk mutációt az  $S_b$ -beli kromoszómákra és azok (immáron  $S_w$ -beli) leszármazottaira.

9. Számoljuk ki az új generáció fitnessz-értékeit, majd ugorjunk a 4-es pontra.

### 3.3. Egyéb felhasználási területek

#### 3.3.1. Képméretezés

#### 3.3.2. Képingdexelés



3.2. ábra. Módosított Sierpinski-háromszög

CF	a	b	c	d	e	f
w1	0.5	0	0	0.4	1	150
w2	0.5	0	0	0.6	-80	1
w3	0.5	0	0	0.6	80	1

3.1. táblázat. A módosított Sierpinski háromszög IFS-kódja

CF	a	b	c	d	e	f
w1	0.5	0	0	0.5	1	150
w2	0.5	0	0	0.5	-80	1
w3	0.5	0	0	0.5	80	1

3.2. táblázat. Az eredeti Sierpinski háromszög IFS-kódja

Hasonló képekhez tarozó IFS is hasonló, mint ahogy az a módosított és az eredeti Sierpinski-háromszög IFS-kódjának összevetéséből is jól látható. Éppen ezért az IFS-kód jól használható

képek globális tulajdonságainak összevetésére, sőt egyfajta kategorizálásra is lehetőséget ad. Bizonyos értelemben tartalom alapú keresést tesz lehetővé képeket tartalmazó adatbázisokban ([5]).

### 3.3.3. Vízjelezés

A demokratikus média (Internet) népszerűsödésével egyre nehezebb feladat a szerzői jogok érvényre juttatása, hiszen például egy digitális képről nagyon nehéz bizonyítani, hogy adott személy készítette. A vízjelezés lényege, hogy olyan (bináris) információkat rejtünk el a képben/képen, amelyek hagyományos eszközökkel nem törölhetőek, ellenálnak különböző geometriai transzformációknak, és a veszteséges tömörítés sem tesz különösebb kárt bennük. Többféle módszer létezik, az egyik a fraktál alapú képtömörítés tulajdonságait használja ki. A módszer lényege, hogy a képhez tartozó IFS-t úgy változtatjuk meg, hogy a kódolás minősége ne változzon. Igen valószínűtlen, hogy egy természetes képen bármely értelmezési tartomány tökéletesen megegyezzen a hozzá tartozó értékkészlettel. Ezt kihasználva néhány  $R$  értékkészletet lecserélünk a hozzájuk tartozó értelmezési tartományok pontos másolataira. Ezek a pontos másolatok reprezentálják (egy rögzített protokoll alapján) az elhelyezni kívánt vízjelet. A legnehezebb feladatot a megfelelő értelmezési tartományok kiválasztása jelenti, hiszen ezeket úgy kell megválasztani, hogy robosztusak legyenek az aluláteresztő szűrőkkel (ilyeneket használnak a tömörítő algoritmusok), a forgatással és az eltolással szemben. A blokkok kiválasztása az alacsony frekvenciás összetevőik alapján is történhet, az algoritmus DCT-alapú változata ez alapján rangsorol. A gyakorlat azt mutatja, hogy a DCT alapú vízjelezési módszer sokkal robosztusabb, mint a képtér-alapú. Egy ilyen alapon működő protokoll részletes leírását találjuk [6]-ban

# Bibliográfia

- [1] Álmos Attila, Győri Sándor, Horváth Gábor, Várkonyiné Kóczky Annamária: Genetikus Algoritmusok
- [2] Ming-Sheng Wua, Jyh-Horng Jeng, Jer-Guang Hsieh : Schema genetic algorithm for fractal image compression
- [3] [origo] Eltűnt a hulladék: új korszak kezdődött a DNS megértésében, 2006.október 16.
- [4] K.T. Sun, S.J. Lee P.Y. Wu : Neural network approaches to fractal image compression and decompression
- [5] M. Nappi, G. Polese, G. Tortora : FIRST: Fractal Indexing and Retrieval SysTem for Image Databases
- [6] Patrick Bas, Jean-Marc Chassery, Franck Davoine : Using the fractal code to watermark images
- [7] Krzysztof Gdawiec : Curves as fractals
- [8] Mahesh C Shastry, Nithin Nagaraj, Prabhakar G Vaidya : The B-Exponential Map: A Generalization of the Logistic Map, and Its Applications In Generating Pseudo-random Numbers
- [9] Mitsuhiro Shishikura : The Hausdorff Dimension of the Boundary of the Mandelbrot Set and Julia Sets
- [10] Jean-Francois Gouyet : Physics and Fractal Structures
- [11] Káosz és nemlineáris dinamika a társadalomtudományokban (Szerkesztette : Fokasz Nikosz)

[12] Tél Tamás - Gruiz Márton : Kaotikus Dinamika

[13] James Gleick : Káosz - egy új tudomány születése

[14] Wikipedia - Fractals - Chaos theory - Hausdorff dimension

[15] Retter Gyula : Fuzzy, Neurális, Genetikus, Kaotikus rendszerek

[16] Bacsosz Sztavrosz : Fraktálok a földrajzban

[17] Heinz-Otto Peitgen Hartmut Jürgens Dietmar Saupe : Chaos and fractals

[18] Kecskés Lajos : Egy ölnyi végtelen

[19] Ziauddin Sardar, Iwona Abrams : Káoszelmélet másKÉPp

[20] Fractal Image compression, Editor : Yuval Fisher