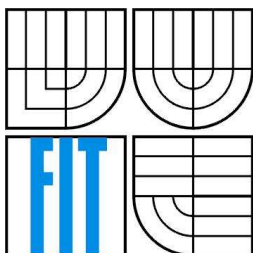


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ
FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND
MULTIMEDIA

KŘIVKY VYPLŇUJÍCÍ PROSTOR

SPACE FILLING CURVES

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

MAREK STAŇO

VEDOUČÍ PRÁCE
SUPERVISOR

ING. ONDŘEJ ŠILER

BRNO 2008

Zadanie

1. Zoznámte sa s rôznymi druhmi kriviek, ktoré vyplňujú priestor.
2. Pokúste sa zovšeobecniť tieto krivky do viacdimeziionálnych priestorov.
3. Implementujte niektoré algoritmy kriviek.
4. Navrhните u fraktálnych kriviek nerekurzívne riešenie.
5. Zhodnoťte výsledky a analyzujte možnosti využitia týchto kriviek.

Licenční zmluva

Licenční zmluva je uložená v archíve Fakulty informačných technológií Vysokého učení technického v Brne.

Abstrakt

Cieľom mojej práce je zoznámiť sa s krivkami vyplňujúcimi priestor a pokúsiť sa zovšeobecniť tieto krivky do viacdimezióálnych priestorov. Ďalej sa v práci nachádzajú informácie o rekurzívnom a nerekurzívnom riešení týchto kriviek a využitie kriviek v praxi. Hlavným cieľom je zhotoviť program, ktorý dokáže vykresliť niektoré krivky.

Pomocou získaných znalostí navrhujem program, ktorý vypočíta súradnice zadanej krivky a druhý program, ktorý pomocou týchto súradníc vykreslí obrázok s krivkou. K vytvoreniu použijem programovací jazyk C a grafickú knižnicu Gd.

Kľúčové slova

Fraktál, Hausdorffova dimenzia, Topologická dimenzia, sebedobnosť, rekurgia

Abstract

The objective of my work is introduce space filling curves, and try to generalize them to multidimensional spaces. Below are in the work informations about recursive and non-recursive solving of these curves, and usage curves in practice. The main objective is to create program which can draw these curves.

Based on theory and practical knowledge I design program, which can compute the coordinates of the curve, and another program which uses these coordinates to draw the curve. To make this I will use programming language C and graphics library Gd.

Keywords

Fractal, Hausdorff dimension, Topological dimension, self-similarity, recursion

Citace

Staňo M.: Krivky vyplňujúce priestor, bakalárska práca, Brno, FIT VUT v Brne, 2008

Krivky vyplňujúce priestor

Prehlásenie

Prehlasujem, že som túto bakalárskou prácu vypracoval samostatne, pod vedením Ing. Ondřeja Šilera a že som uviedol všetky literárne pramene a publikácie, z ktorých som čerpal.

.....
Meno Priezvisko
Dátum

Pod'akovanie

Ďakujem vedúcemu mojej práce, pánovi Ing. Ondřejovi Šilerovi, za jeho odbornú pomoc a jeho veľmi cenné rady pri spracovaní tejto témy.

Zároveň by som chcel poďakovať celému ústavu UPGM na Fakulte informačných technológií

Vysokého učení technického v Brne za organizovanie odborných seminárov.

© Marek Staňo, 2008.

Tato práca vznikla ako školské dielo na Vysokého učení technického v Brne, Fakulte informačných technológií. Práce je chránená autorským zákonom a jej užitie bez vykonania oprávnenia autorom je nezákonné, s výnimkou zákonom definovaných prípadov..

OBSAH

1	Úvod	2
2	Fraktálna geometria	3
2.1	Vznik fraktálnej geometrie	3
2.2	Euklidovská vs. Fraktálna geometria	3
2.3	Fraktál	3
2.4	Sebepodobnosť	4
3	Oboznámenie s krivkami	4
3.1	Kochova krivka	4
3.2	Peanova krivka	5
3.3	Wunderlichove modifikácie Peanovej krivky	7
3.4	Hilbertova krivka	8
3.5	Moorova krivka	9
3.6	Mortonova (Z-order) krivka	10
4	Krivky vyplňujúce viacdimenziálny priestor	12
4.1	Hilbertova krivka v 3D	12
4.2	Mortonova krivka v 3D	14
4.3	Krivky v 4-dimenziálnom priestore	15
5	Rekurzia a nerekurzívne riešenie kriviek vyplňujúcich priestor.....	17
5.1	Čo je rekurzia	17
5.2	Nerekurzívne riešenie kriviek vyplňujúcich priestor	18
6	Využitie kriviek vyplňujúcich priestor v praxi	19
6.1	Počítačová grafika	19
6.2	Využitie fraktálnej analýzy pri hodnotení kvality tlače	20
6.3	Ďalšie využitie kriviek vyplňujúcich priestor	20
7	Tvorba programu	22
8	Záver	23
9	Literatúra	24
10	Zoznam príloh	25

1 Úvod

Krivky vyplňujúce priestor používame v súčasnej dobe na rôzne účely v počítačovej technológii, vo viacdimenziálnych priestoroch na indexovanie, alebo vo wireless sieťach. Medzi najznámejšie patria Peanova krivka, Hilbertova krivka, Moorova krivka alebo Kochova vložka. Tieto krivky môžu byť tvorené v dvojdimenziálnom priestore, niektoré majú aj trojdimenziálne verzie, napríklad Hilbertova krivka, alebo Kochova vložka. Patria medzi fraktálne krivky.

Začiatok mojej práce je o fraktálnej geometrii, pri ktorej vysvetľujem Topologickú a Hausdorffovu dimenziu. Zaoberám sa krivkami vo viacdimenziálnom priestore. Spomeniem nerekurzívne riešenie kriviek a využijem ich v praxi.

V praktickej časti vysvetľujem tvorbu niektorých algoritmov kriviek.

2 Fraktálna geometria

2.1 Vznik fraktálnej geometrie

Rozvoj fraktálnej geometrie sa datuje v 60. rokoch 20. storočia a za jej zakladateľa je považovaný vedec a matematik poľského pôvodu Benoit B. Mandelbrot, ktorý ako prvý matematik definoval pojem fraktál. Už pred Mandelbrotom sa vedci a matematici zaoberali geometrickými útvarmi nazývanými fraktálmi, ale nikto z nich nedefinoval obecné tento pojem, preto je prvenstvo pripisované práve Mandelbrotovi. Až do 19. storočia sa využívalo geometrických útvarov pravidelných a hladkých (priamka, štvorec, kružnica), ale ak vezmeme do úvahy tvary prírody skúsme ich týmito útvarmi popísať. Práve koncom 19. storočia sa začali objavovať tvary, ktoré medzi tie pravidelné nezapadali. V roku 1872 ako prvý Karl Theodor Wilhelm Weierstrass zostrojil spojitú funkciu, ktorá nemá nikde deriváciu, nasledoval ho Georg Cantor so svojím diskontinuom (1884) Giuseppe Peano s krivkou, ktorá spojitاً zobrazuje úsečku na štvorec (1890) a Niels Fabian Helge von Koch s jeho nekonečne dlhou úsečkou ohraničujúcou konečnú plochu.

2.2 Euklidovská vs. Fraktálna geometria

Klasická Euklidovská geometria popisuje tvary pomocou tzv. topologickej dimenzie. O dimenzii ako takej môžeme jednoducho povedať, že definuje, koľkými parametrami môžeme dané teleso popísať. Topologická dimenzia je celočíselná a popisuje geometricky hladké a pravidelné tvary, napríklad dimenzia bodu je rovná 0, pre priamku je dimenzia rovná 1, pre rovinné teleso 2 a priestorové teleso má dimenziu 3. U fraktálnej geometrie využívame neceločíselnú dimenziu. Podľa nemeckého profesora Felixa Hausdorffa sa nazýva Hausdorffova dimenzia. V mnohej literatúre sa však stretávame s pojmom fraktálna dimenzia. Hodnota tejto dimenzie nám udáva mieru členitosti objektu. Čím je hodnota Hausdorffovej dimenzie medzi dvoma číslami väčšia, tým je tvar členitejší a z pohľadu vyplnenia priestoru lepší.

2.3 Fraktál

Termín fraktál pochádza z latinského slova fractus, prekladom rozbitý. Presná definícia, čo to fraktál je, nebola úplne definovaná. Používa sa definícia Benoita Mandelbrota, ktorý definoval fraktál ako

množinu, či geometrický útvar, ktorého Hausdorffova dimenzia je väčšia ako dimenzia topologická.
[1]

2.4 Sebepodobnosť

Sebepodobnosť je jednou z hlavných vlastností fraktálu a zjednodušene by sa dalo povedať, že sebepodobnosť znamená opakovanie seba samého napríklad zmenenou veľkosťou, posunutím, rotáciou, alebo skosením. [2]

3 Oboznámenie s krivkami

V tejto časti budú spomenuté niektoré najznámejšie krivky vyplňujúce priestor. Medzi tieto krivky patria Kochova krivka, Hilbertova krivka, Peanova krivka a rôzne iné. Budú popísané ich vlastnosti, história a spôsob akým ich vytvárame. Viac informácií v [3].

3.1 Kochova krivka

Kochova krivka je jedna z prvých popísaných fraktálnych kriviek. Známejšia je ako súčasť Kochovej vložky, vytvorenej z troch spojených Kochových kriviek. Pomenovaná je podľa Helge von Kocha (1870-1924). Bol to švédsky matematik, napísal niekoľko kníh o číselnej teórii, ale jeho najznámejšie dielo bolo *Sur une courbe continue sans tangente, obtenue par une construction géométrique élémentaire* z roku 1904, v ktorom popísal práve túto krivku.

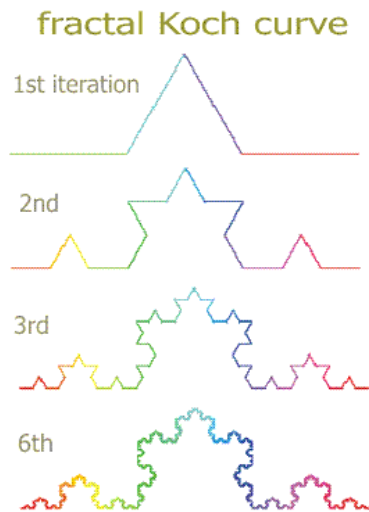
Tvorba Kochovej krivky

Kochova krivka vznikne nekonečným opakovaním jednoduchého postupu. Na začiatku je prostá úsečka (v prípade Kochovej vložky rovnostranný trojuholník tvorený tromi takými úsečkami).

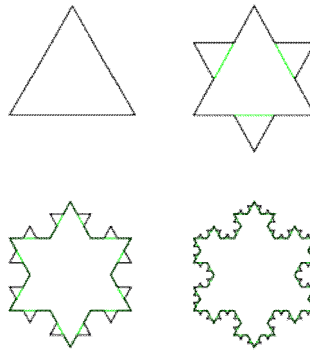
V každom kroku sa potom spraví nasledujúce:

1. Úsečka sa rozdelí na tretiny
2. Nad prostrednú tretinu sa zostrojí rovnostranný trojuholník
3. Základňa trojuholníka (bývalá stredná tretina úsečky) sa odstráni

Tým sa z pôvodnej úsečky stane krivka zložená zo štyroch úsečiek (respektíve z trojuholníka sa stane šesťcípá hviezda) a postup sa rekurzívne opakuje s každou takto vzniknutou úsečkou.



Obrázok 1: Kochova krivka



Obrázok 2: Kochova vločka(Kochov ostrov)

Kochova krivka vznikne ako limita pri opakovaní tohto postupu do nekonečna. Jej dĺžka je nekonečná, lebo sa v každom kroku predĺži vždy o tretinu – z troch častí úsečky vzniknú štyri rovnako dlhé. Z toho vyplýva, že v kroku n bude dĺžka krivky $(4/3)^n$ dĺžky pôvodnej úsečky. Hausdorfova dimenzia Kochovej krivky je teda $\log 4 / \log 3 \cong 1,26$ (t.j. krivka zaplňuje priestor viac ako obyčajná priamka s dimenziou 1, ale nezaplňuje ho úplne ako iné krivky s dimenziou 2).

Ako už bolo povedané Kochova vločka (niekedy ju voláme aj Kochov ostrov) vzniká tým, že sa na počiatku pracuje s rovnostranným trojuholníkom miesto jedinej úsečky. Výsledkom je teda plošný fraktálny útvar. Obsah takéhoto útvaru je (na rozdiel od jeho obvodu) konečný. V každom kroku sa síce plocha zväčšuje, ale pridávané trojuholníky sú čím ďalej tým menšie a výsledkom je konvergentná geometrická rada. Obsah takejto Kochovej vločky je rovný $8/5$ obsahu pôvodného trojuholníka. U Kochovej vločky teda nekonečne dlhá krivka ohraničuje konečnú plochu.

Miernou modifikáciou pravidiel možno vytvoriť mnoho podobných kriviek, napríklad použitím štvorcov namiesto trojuholníkov, alebo Kochova antivločka, pri ktorej trojuholníky smerujú dovnútra pôvodného trojuholníka a obsah fraktálu tak znižujú.

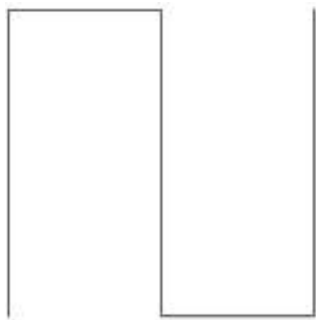
3.2 Peanova krivka

Krivky vyplňujúce priestor (Peanove krivky), sú krivky, ktoré ako prvý popísal Giuseppe Peano žijúci v rokoch (1858 – 1932).

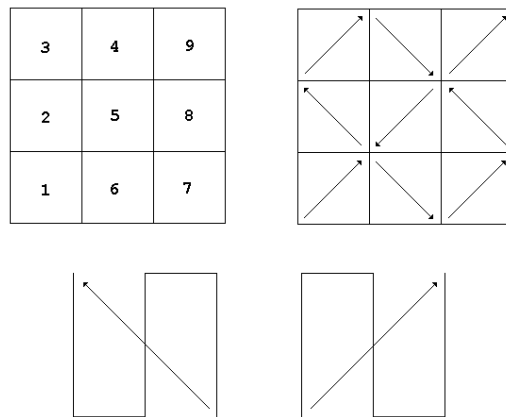
Peanove krivky, na rozdiel od kriviek popísaných vyššie vyplňujú celý (2-dimenzionálny) štvorec alebo celú (3-dimenzionálnu) kocku. Je to vlastne spojitá krivka, ktorá môže byť chápaná ako neprerušená cestička pohybujúceho sa bodu. Lepšia definícia je: „Krivka (s koncovými bodmi) je spojitá funkcia, ktorej oblasť je jednotkový interval $[0,1]$ “. Podľa Peanovej krivky vznikli ďalšie dve. Jednu odvodil David Hilbert a druhú E.H.Moore, tieto krivky budú popísané nižšie.

Tvorba Peanovej krivky

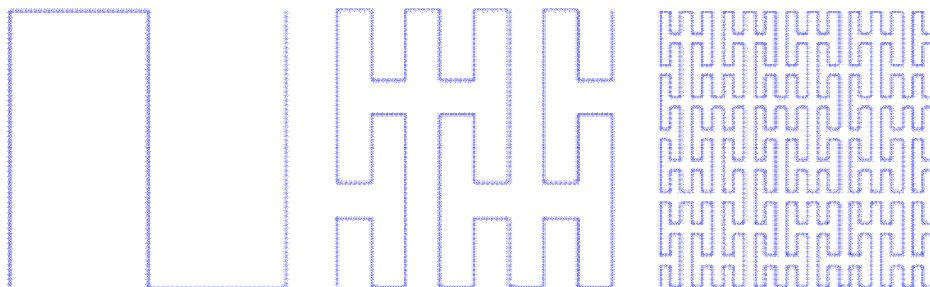
Peanova krivka vzniká (ako skoro všetky) opakovaním jedného postupu. Na začiatku krivka vyzerá ako na obr. 3. Potom sa každý z deviatich štvorcov rozdelí na ďalších 9 a do nich sa kopírujú zmenšené krivky z predchádzajúcej iterácie podľa pravidiel z obr. 4. Tým sa z nej stane krivka, ktorá vyzerá oveľa zložitejšie, ale je to krivka, pospájaná z 9 kópií základnej krivky. Všimnime si, že krivky z obrázka 4 dolu ktoré kopírujeme hore podľa pravidiel, vznikli z jednej základnej krivky z 0 iterácie jednoduchým vertikálnym prevrátením. Tento postup sa potom opakuje s každou takto vzniknutou krivkou. Dôležité je aj to, že Peanovu krivku vytvárame v štvorci $3^{-n} \times 3^{-n}$ zatiaľ čo jej odvodenia od Hilberta a Moora vytvárame v štvorci $2^{-n} \times 2^{-n}$.



Obrázok 3: Základný tvar Peanovej krivky



Obrázok 4: Tvorba Peanovej krivky

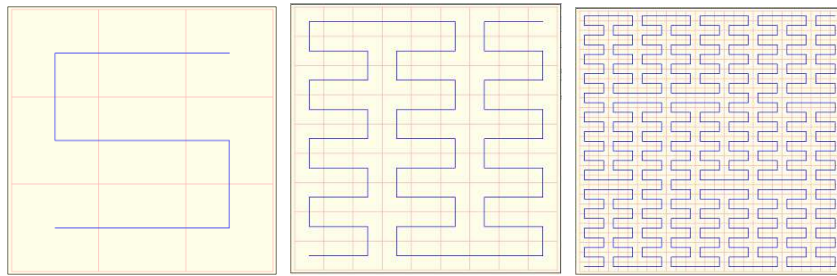


Obrázok 5: Prvé dve iterácie Peanovej krivky

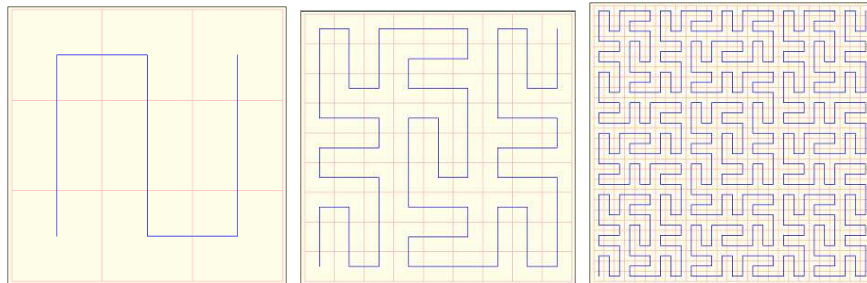
Na obr. 5 vidíme, že z pohľadu vyplnenia priestoru je Peanova krivka jedna z najlepších, pretože jej Hausdorfova dimenzia je rovná 2 čo znamená, že vyplňuje celý priestor.

3.3 Wunderlichove modifikácie Peanovej krivky

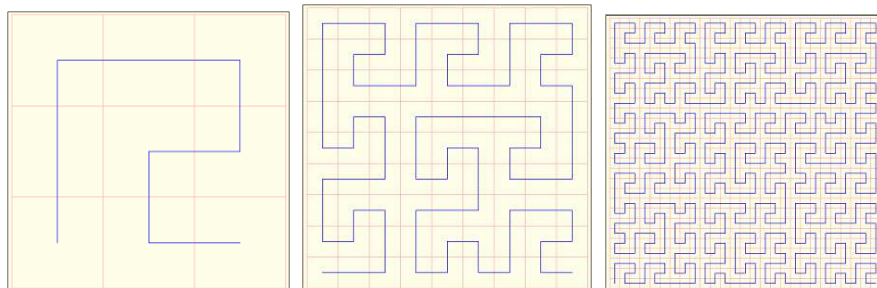
Walter Wunderlich vytvoril 3 modifikácie Peanovej krivky. Popísal všetky možné krivky vyplňujúce priestor, ktorých konštrukcia je v intervale 3^{-2n} tj. štvorec o veľkosti $3^{-n} \times 3^{-n}$. Poznáme dva typy týchto kriviek, a to serpentínovú a zákrutovitú (červovitú). Vytváranie troch Wunderlichových kriviek nebude podrobne popísané, môžete ich však vidieť na obrázkoch 6,7 a 8. Bližšie informácie o Wunderlichových modifikáciách Peanovej krivky na [4].



Obrázok 6: Wunderlichova krivka 1



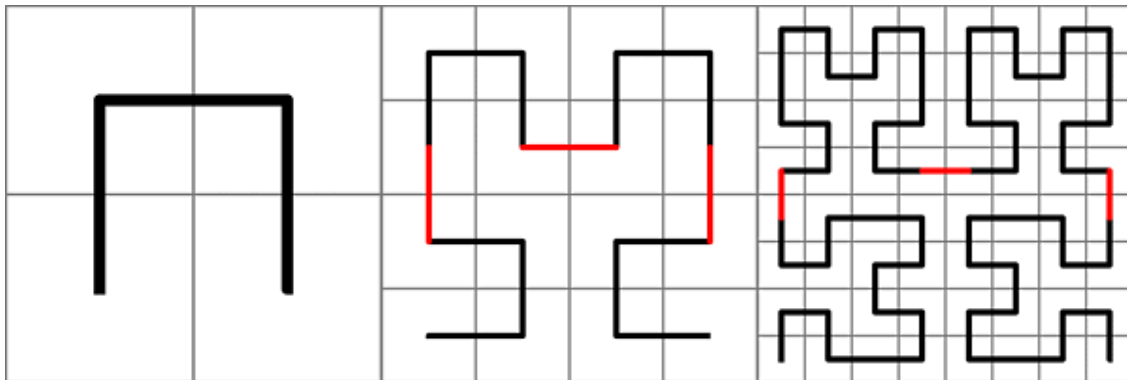
Obrázok 7: Wunderlichova krivka 2



Obrázok 8: Wunderlichova krivka 3

3.4 Hilbertova krivka

Tvorcom Hilbertovej krivky je Nemecký matematik David Hilbert. Žil v rokoch 1862-1943. Hilbertova krivka je spojitá fraktálna krivka vyplňujúca priestor. Hausdorfova dimenzia tejto krivky je 2, čo znamená, že dokonale vyplňuje 2D priestor. Dĺžka krivky je $2^n - 1/2^n$ a exponenciálne rastie s n , n je číslo iterácie (alebo krok vykreslenia).



Obrázok 9: Tvorba Hilbertovej krivky

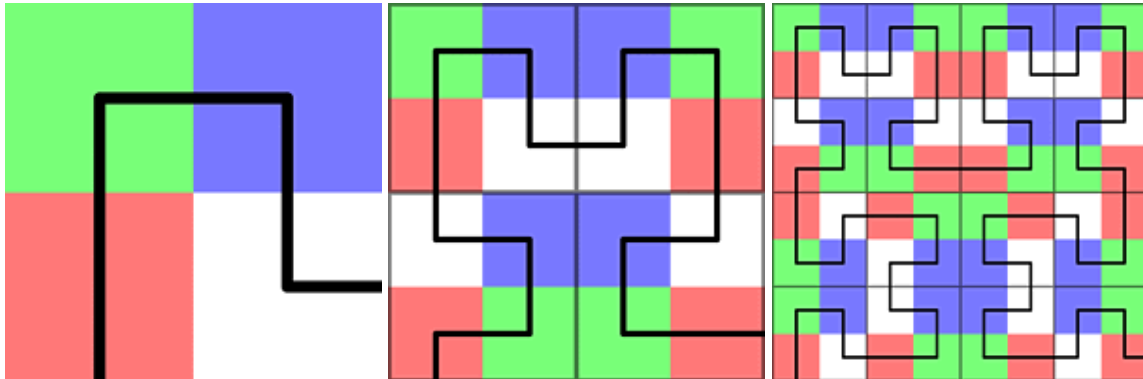
Tvorba Hilbertovej krivky

Hilbertovu krivku tvoríme z neúplného štvorca (má len 3 strany, alebo tvar obráteného U). Na obrázku 9 v strede vidíme 1. iteráciu Hilbertovej krivky. Každý zo štyroch štvorcov bol rozdelený do ďalších štyroch. Krivka v tvare obráteného U z prvého obrázka sa zmenší na polovicu svojej veľkosti a kopíruje sa do každého zo štyroch sektorov. V ľavom hornom rohu je jednoducho skopírovaná, v pravom hornom je horizontálne prevrátená. V ľavom dolnom je otočená o 90 stupňov v smere hodinových ručičiek a v pravom dolnom je otočená o 90 stupňov proti smeru hodinových ručičiek. Tieto 4 diely sú spojené tromi časťami, z ktorých každá je rovnakej veľkosti, ako zmenšené diely obráteného U tvaru. Na obrázku 9 ich vidíme nakreslené červenou farbou. Viac informácií o tvorbe Hilbertovej krivky na [8].

Na obrázku 9 vpravo vidíme druhú iteráciu. Každý zo šesťnástich štvorcov z iterácie 1 je rozdelený na ďalšie štyri, a tvar vzniknutý v iterácii 1 je zmenšený a kopírovaný. Znova, tri spájacie časti sú pridané na dokončenie krivky. Všetky iterácie pokračujú týmto istým spôsobom. Krivka z predchádzajúcej iterácie sa zmenší na polovicu a štyrikrát sa skopíruje. V ľavom hornom rohu sa skopíruje, v pravom hornom prevrátená, v ľavom dolnom otočená a v pravom dolnom otočená opačne. Krivka vždy začína v ľavom dolnom rohu a končí v pravom dolnom, nikdy sa neprekríži, ani nedotkne.

Na obrázku 10 sú prvé dve iterácie s farebným pozadím na lepšie predstavenie konštrukcie krivky. V každom 2x2 bloku štvorcov krivka začína v červenom, ide cez zelený, modrý a končí

v bielom. V príkladoch pre iteráciu 1 a 2 boli mriežkové čiary pridané na lepšie rozpoznanie rozdelenia do štvorcov 2x2. Tu môžeme pekne vidieť, ako je základný tvar krivky len veľakrát zmenšovaný a otáčaný.



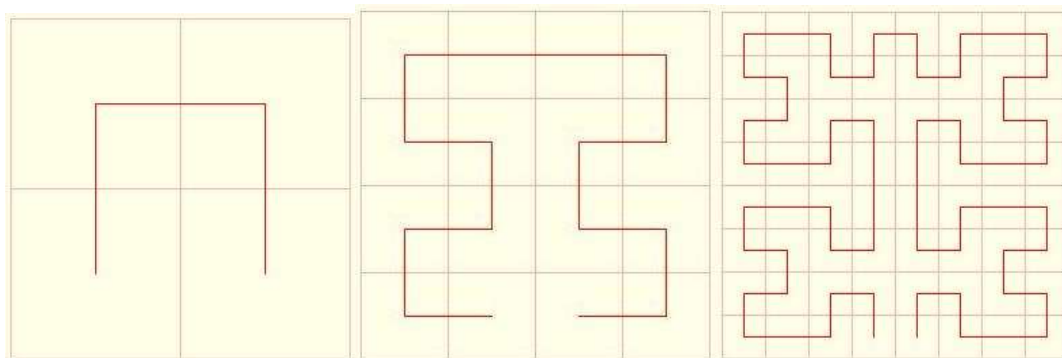
Obrázok 10: Prvé dve iterácie Hilbertovej krivky s farebným pozadím

3.5 Moorova krivka

Moorova krivka je takisto ako aj Hilbertova odvodená od Peanovej krivky. Jej tvorcom je **Eliakim Hastings Moore** (1862-1932). Moorova krivka je spojitá fraktálna krivka vyplňujúca priestor. Hausdorfova dimenzia tejto krivky je 2, čo znamená, že dokonale vyplňuje 2D priestor. Dĺžka krivky je $2^n - 1/2^n$ a exponenciálne rastie s n , n je počet opakovaní rekurzie, alebo krok vykreslenia.

Tvorba Moorovej krivky

Moorovu krivku tvoríme takisto ako Hilbertovu z neúplného štvorca v tvare obráteného U. Na tejto krivke je oproti ostatným zvláštne, že každá ďalšia iterácia sa netvorí z predchádzajúcej iterácie tejto krivky, ale je tvorená z predchádzajúcej iterácie krivky Hilbertovej. Prvú iteráciu Moorovej krivky môžeme vidieť na obrázku 11 v strede.

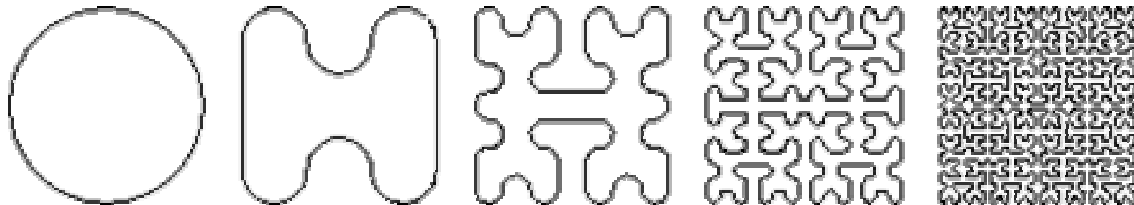


Obrázok 11: Prvé 2 iterácie Moorovej krivky

Vzniká tak, že každý zo štyroch štvorcov sa rozdelí do ďalších štyroch. Tvar obráteného U z obrázka 11 vľavo sa zmenší na polovicu a postupne sa kopíruje do každého zo štyroch sektorov. Do ľavého dolného sa otočí o 90 stupňov proti smeru hodinových ručičiek, do pravého dolného sa otočí o 90 stupňov smere hodinových ručičiek a horizontálne sa prevráti, do ľavého horného sa otočí o 90 stupňov v protismere a tiež sa horizontálne prevráti a do pravého horného sa len otočí o 90 stupňov v smere hodinových ručičiek. Tieto 4 diely sú spojené tromi časťami, z ktorých každá je rovnakej veľkosti, ako zmenšené diely obráteného U tvaru tak isto ako u Hilbertovej krivky. Ďalšiu (čiže druhú) iteráciu vidíme na obrázku 11 vpravo. Každý zo šesnástich štvorcov bol opäť rozdelený na 4, ale tento krát nebudeme zmenšovať tvar, ktorý sme dostali v predchádzajúcej iterácii, ale musíme použiť tvar, ktorý vznikne po prvej iterácii Hilbertovej krivky (napr. obr. 10 v strede) a tento zmenšíme na polovicu a kopírujeme podľa rovnakých pravidiel ako pri prvej iterácii. Pri tretej iterácii opäť zoberieme krivku z druhej iterácie Hilbertovej krivky a túto potom kopíruje podľa daných pravidiel.

Vznikne nám krivka, ktorá začína i končí dolu v strede štvorca, a takisto ako Hilbertova sa nikdy neprekríži ani nedotkne.

Poznáme tiež modifikovanú verziu Moorovej krivky, ktorá sa tvorí z kruhu, a môžeme ju vidieť na obrázku 12.



Obrázok 12: Modifikovaná (kruhová) verzia Moorovej krivky

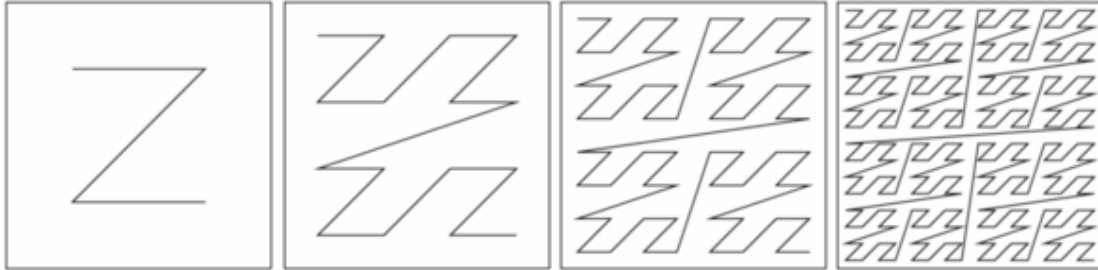
3.6 Mortonova krivka (Z-order)

Mortonovu krivku, alebo Z-order krivku popísal ako prvý G. M. Morton. Veľmi často sa používa v počítačových technológiách. Jej názov (Z-order krivka) je odvodený od tvaru jej základnej krivky, ktorá vyzerá ako písmeno Z. Zobrazená je na obrázku 13 vľavo.

Tvorba Mortonovej krivky

Mortonovu krivku tvoríme tak, že na začiatku vpíšeme do štvorca krivku v tvare písmena Z. Prvú iteráciu vytvoríme tým, že rozdelíme štvorec na 4 menšie štvorce, základnú krivku zmenšíme na polovicu a do každého z týchto štyroch novovzniknutých štvorcov skopírujeme zmenšenú neotočenú základnú krivku. Pri Mortonovej krivke sú dôležité spojovacie čiary, ktoré spájajú tieto 4

krivky do jednej. Prvá spojovacia čiara začína v spodnom konci krivky z ľavého horného rohu, a smeruje k hornému koncu krivky z pravého horného rohu. Zo spodného konca tejto krivky začína druhá spojovacia čiara a smeruje k hornému koncu krivky v ľavom dolnom rohu, a z jej spodného konca ide tretia krivka do horného konca krivky z pravého dolného rohu. Pre lepšie predstavenie je prvých pár iterácií zobrazených na obrázku 13.



Obrázok 13: Prvé 3 iterácie Mortonovej krivky

Ďalšie iterácie, ako môžeme vidieť na obrázku 13 vznikajú rovnako, ako pri iných krivkách. Zoberieme krivku z predchádzajúcej iterácie, zmenšíme na polovicu, skopírujeme ju do každého zo 4 štvorcov a spojíme pomocou spojovacích čiar.

Mortonova krivka je spojitá fraktálna krivka vyplňujúca priestor, nikdy sa nepretne, ani nedotýka samej seba, používa sa v počítačových technológiách napríklad na indexovanie. Krivka vždy začína v ľavom hornom rohu, končí v pravom dolnom a zabezpečuje spoľahlivý prechod celým štvorcom. Hodnota jej hausdorffovej dimenzie je rovná 2, takže aj Mortonova krivka dokonale vyplňuje 2D priestor. Viac na [5].

4 Krivky vyplňujúce viacdimezióálny priestor

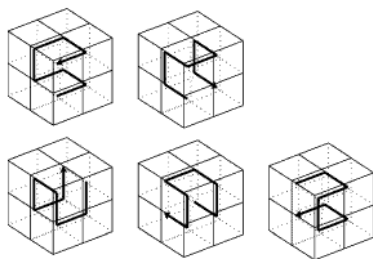
V časti krivky vyplňujúce viacdimezióálny priestor budú popísané niektoré krivky, ktoré vyplňujú trojdimezióálnu kocku. Medzi tieto krivky patrí napríklad 3D verzia Hilbertovej krivky, alebo trojdimezióálna verzia Z krivky (Mortonovej krivky). Na záver kapitoly bude popísané niečo o štvordimezióálnych krivkách. Viac informácií k tejto kapitole na [6,12].

4.1 Hilbertova krivka v 3D

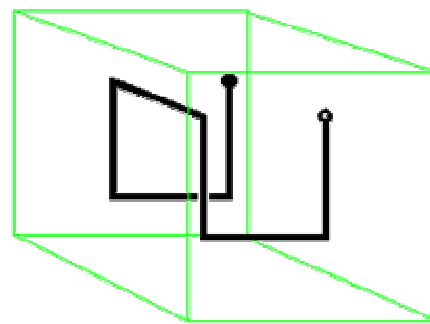
Trojdimezióálna Hilbertova krivka je takisto ako jej 2D verzia spojitá fraktálna krivka. Tvoríme ju v kocke $2 \times 2 \times 2$ a krivka ju vyplní celú, takže jej Hausdorffova dimenzia je rovná 3. Z hľadiska vyplnenia priestoru je teda Hilbertova krivka v 3D priestore ako aj v 2D dokonalá.

Tvorba Hilbertovej krivky v 3D

Trojdimezióálna Hilbertova krivka sa tvorí podobne ako dvojdimenziálna z jednej základnej krivky pomocou rekurzie. Základná krivka je zobrazená v kocke $2 \times 2 \times 2$, čiže nám vznikne 8 menších kociek. Keď sa na kocku pozeráme spredu, základná krivka začína v pravom hornom prednom rohu, ďalej smeruje dolu, potom doľava a odtiaľ hore. Z ľavého horného predného rohu prejdeme do ľavého zadného horného, odtiaľ prejdeme dolu, potom doprava a skončíme v pravom hornom zadnom rohu. Pre lepšiu ilustráciu si môžeme tvar základnej kocky pozrieť na obrázkoch 14.a a 14.b.



Obrázok 14.a: Základná Hilbertova 3D krivka z rôznych pohľadov



Obrázok 14.b: Základná Hilbertova 3D krivka pri pohľade spredu.

Keďže už máme základný tvar krivky, môžeme prejsť na prvú iteráciu. Vytvoríme ju tak, že každú z ôsmich kociek, ktoré vznikli z pôvodnej kocky rozdelíme na ďalších 8, takže nám vznikne

kocka 4x4x4. Pôvodnú krivku zmenšíme na polovičnú veľkosť a postupne otáčame a kopírujeme do každej z 8 pôvodných kociek podľa zadaných pravidiel. Pre lepšiu prehľadnosť sú tieto pravidlá uvedené v bodoch. Pravidlá popisujú ako v jednotlivých kockách otáčame pôvodný tvar krivky. Treba pripomenúť, že na kocku sa pozeráme spredu, aby sme si nepomýlili smer otáčania. Dozadu znamená, že predná strana sa stane vrchnou, a doprava znamená, že predná strana sa stane pravou stranou. Označenie jednotlivých kociek bude pomocou *n, s, e, w, b, f* z anglických názvov pre východ západ sever, juh, prednú a zadnú stranu. (napríklad pravá horná predná bude *enf*)

enf- v smere hodinových ručičiek a potom prevrátime dozadu.

esf- doprava a potom dopredu.

wsf- proti smeru hodinových ručičiek a potom doprava.

wnf- v smere hodinových ručičiek o 180°.

wnb- v smere hodinových ručičiek o 180°.

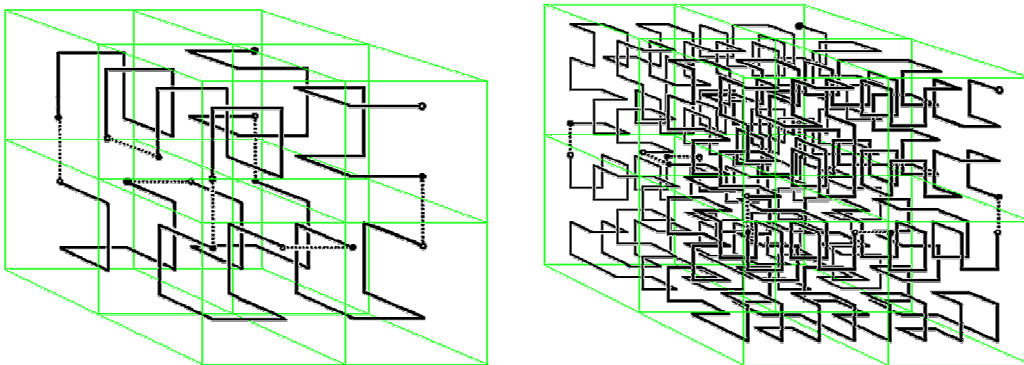
wsb- proti smeru hodinových ručičiek a potom doľava.

esb- proti smeru hodinových ručičiek a potom doľava.

wnb- v smere hodinových ručičiek a dopredu.

Všetky tieto časti sú pospájané siedmimi dielmi, ktoré majú veľkosť, ako jedna úsečka zmenšenej základnej krivky. Môžeme ich vidieť na obrázku 15 prerušovanou čiarou. Poradie spájania je presne také isté ako smer základnej krivky. Ďalšiu iteráciu, čiže druhú, spravíme tak, že zoberieme krivku z prvej iterácie, zmenšíme ju na polovicu, a postupne ju kopírujeme a otáčame podľa hore uvedených pravidiel. Každá ďalšia iterácia sa robí presne rovnako. Zoberie sa krivka z predchádzajúcej iterácie, zmenší sa na polovičnú veľkosť, potom sa kopíruje a otáča podľa pravidiel.

Krivku môžeme nazvať aj ako prechod pohybujúceho sa bodu celou kockou, pretože ako už bolo hore spomenuté, krivka je spojitá, vyplňuje celú kocku, a pritom sa nikdy nepretína ani nedotýka. Ďalšie iterácie krivky si môžete pozrieť na obrázku 15.



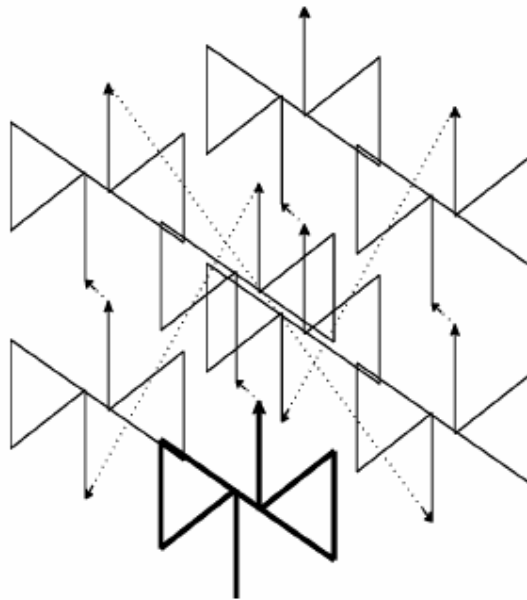
Obrázok 15: Ďalšie iterácie trojdimenzionálnej Hilbertovej krivky

4.2 Mortonova krivka v 3D

Trojdimenzionálna Mortonova krivka je takisto ako jej 2D verzia spojitá fraktálna krivka. Tvoríme ju v kocke $2 \times 2 \times 2$ a krivka ju vyplní celú, takže jej Hausdorffova dimenzia je rovná 3. Z hľadiska vyplnenia priestoru je teda Mortonova krivka v 3D priestore ako aj v 2D dokonalá.

Tvorba Mortonovej krivky v 3D

Mortonovu krivku tvoríme v kocke $2 \times 2 \times 2$ zo základnej krivky opakovaním jedného postupu. Základná krivka začína vpredu v ľavom dolnom rohu, potom prechádza do ľavého horného, odtiaľ ide dozadu do ľavého dolného rohu, potom hore do ľavého horného zadného. Odtiaľ prechádza krížom cez celú kocku do pravého dolného predného rohu, potom do pravého horného pokračuje v pravom dolnom zadnom, a končí vzadu v pravom hornom rohu.



Obrázok 16. Mortonova 3D krivka

Prvú iteráciu vytvoríme tak, že základná krivka sa zmenší na polovicu, kocka sa rozdelí na 8 menších kociek, a do každej z týchto kociek sa skopíruje neotočená zmenšená základná krivka. Tieto krivky sú spojené pomocou spojovacích čiar a to v poradí, v akom smeruje základná krivka. Vždy zvrchu zmenšenej krivky ide spojovacia čiara do začiatku, čiže spodku nasledujúcej krivky. Každá ďalšia iterácia sa robí tým istým spôsobom pomocou krivky z predchádzajúcej iterácie.

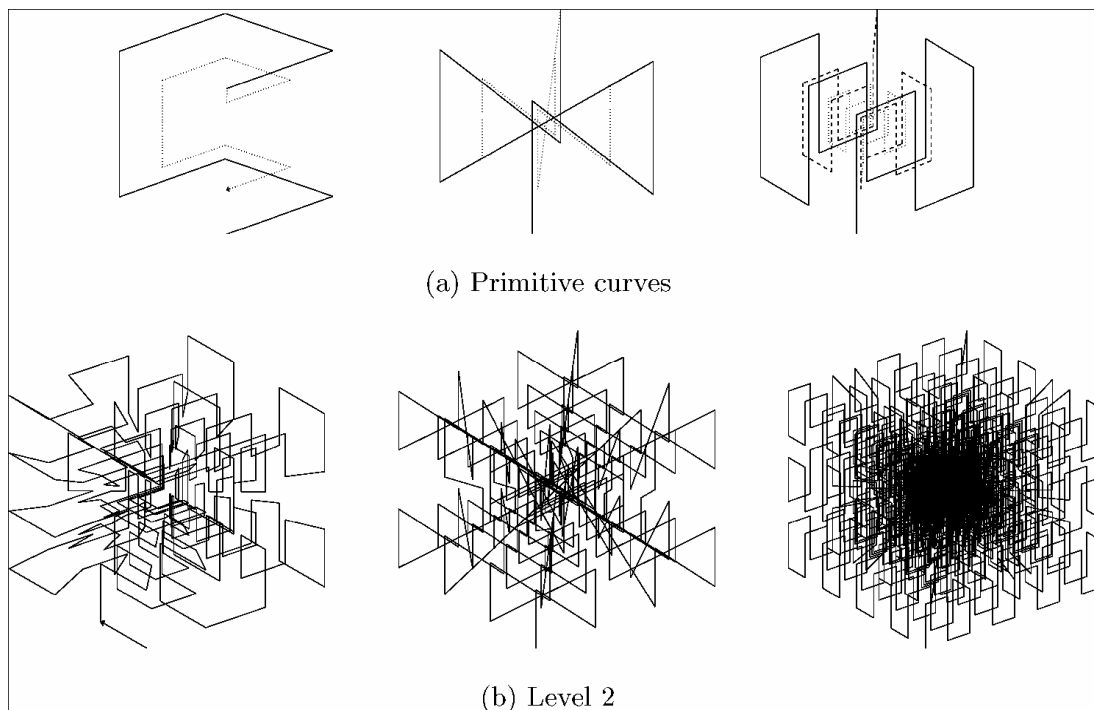
4.3 Krivky v 4-dimenzionálnom priestore

Štvorrozmerný priestor

Štvorrozmerný priestor je priestor, v ktorom je každý bod definovaný štyrmi rozmermi. Človek si svojím myslením nevie štvorrozmerný priestor predstaviť, lebo žije iba v trojrozmernom priestore. V štvorrozmernom priestore je možné niečo také nepredstaviteľné, že napríklad 2 roviny majú iba jeden spoločný bod. Takýto štvorrozmerný priestor nazývame aj priestoročas alebo časopriestor. Je to priestor zjednocujúci fyzikálny 3D priestor a čas. Pri popise pohybu musíme zaznamenať nielen polohu, ale aj čas zaznamenávanej udalosti. Záznam o každej udalosti sa skladá vždy zo štyroch čísel, kde tri z nich udávajú polohu udalosti v priestore a jeden údaj udáva čas jej nastátia. Túto skutočnosť môžeme povedať aj inými slovami:

“Všetky objekty aj my sa pohybujeme nielen v priestore, ale aj v čase”. [6] Alebo by sme mohli povedať skrátene v časopriestore. Všetky udalosti, dianie okolo nás, ale aj vo vesmíre sa odohráva v časopriestore. Na prvý pohľad by sa mohlo zdať, že takáto myšlienka dávať dokopy čas a priestor, len tým, že sme vymysleli nový skrátenejší názov neprinesie nič nového. Ukázalo sa však, že v relativite je to jedna z kľúčových myšlienok. Navyše v teórii relativity je dokonca nevyhnutné uvažovať vždy dianie v rámci časopriestoru, nikdy nie osobitne v priestore a osobitne v čase.

Krivky vyplňujúce 4-rozmerný priestor



Obrázok 17: 4D Hilbertova, Mortonova, a Peanova krivka

Krivky vyplňujúce štvorrozmerný priestor sú veľmi zložité na pochopenie, lebo človek si nevie predstaviť štvorrozmerný priestor, preto nebudem podrobne popisovať tvorbu týchto kriviek. Niektoré môžete vidieť na obrázku 17. Krivky na obrázku vznikajú pomocou koncentrovania trojdimenzionálnych kriviek do jedného bodu, ktorý sa nachádza v strede trojdimenzionálneho priestoru. Krivky sa dajú vykresliť s rôznou koncentráciou prvých troch dimenzií do stredu kocky. Na obrázku sú vykreslené krivky po druhej iterácii s koncentráciou prvých troch dimenzií do stredu 25%. Samozrejme je nemožné sledovať smer krivky na obrázkoch, ich účelom je len lepšie predstavenie týchto 4D kriviek.

5 Rekúzia a nerekurzívne riešenie kriviek vyplňujúcich priestor

Väčšina algoritmov kriviek vyplňujúcich priestor je vytvorených pomocou rekúzie. V tejto kapitole sa budem zaoberať tým, čo je rekúzia a ako sa vytvárajú algoritmy týchto kriviek bez rekúzie.

5.1 Čo je rekúzia

Rekúzia znamená sebaopakovanie. Používa sa veľmi často v matematike a informatike. V oblasti matematiky pojem rekúzia chápeme ako definovanie objektu pomocou volania samého seba. Využíva sa napríklad pre definíciu prirodzených čísiel, stromových štruktúr a niektorých funkcií.

V programovaní rekúzia predstavuje opakované vnorenie volanie rovnakej funkcie (podprogramu), v takom prípade sa hovorí o rekurzívnej funkcii. Neoddeliteľnou súčasťou rekurzívnej funkcie musí byť ukončujúca podmienka určujúca, kedy sa má vnáranie zastaviť. Keďže toto býva najčastejším zdrojom chýb, je treba ju navrhnúť dostatočne silným spôsobom a preveriť všetky možné stavy. Pre uplatnenie rekurzívnych algoritmov je treba, aby programovací jazyk umožňoval volanie podprogramu ešte pred ukončením jeho predchádzajúceho volania. Každý algoritmus používajúci rekúziu je možné prepísať do nerekurzívneho tvaru pri použití zásobníku alebo inej pamäťovej štruktúry.

Základné rozdelenie rekúzií

Rekurzívne chovanie môže byť rôzne v závislosti na tom, koľko podprogramov sa jej účastní.

Rozlišujeme dva základné typy delenia:

- **Priama rekúzia** nastáva, pokiaľ podprogram volá priamo sám seba.
- **Nepriama (vzájomná) rekúzia** je situácia, kedy vzájomné volanie podprogramov vytvorí „kruh“. Napríklad v príkazovej časti funkcie A je volaná funkcia B, vo funkcii B voláme funkciu C, ktorá volá funkciu A.

Druhé delenie:

- **Lineárna rekúzia** nastáva, pokiaľ podprogram pri vykonávaní svojej úlohy volá sám seba iba raz. Vytvára sa tak lineárna štruktúra postupne volaných podprogramov.
- **Stromová rekúzia** nastáva, pokiaľ sa funkcia alebo procedúra v rámci jedného vykonania svojej úlohy vyvolá viackrát. Vzniknutú štruktúru je možné znázorniť ako strom. Pre dve volania v jednom priechode vzniká binárny strom, pre tri ternárny strom atď.

Pri kombinácii spomenutých typov rekúzie je možné docieľiť veľmi komplikovaných štruktúr. Je treba pripomenúť, že už z charakteru takéhoto volania podprogramov sa nedá docieľiť inej ako symetrickej štruktúry. Rekurzívne algoritmy sú výhodné najmä vtedy, keď problém, ktorý sa rieši,

alebo údaje, s ktorými sa má manipulovať, sú definované rekurzívne. To však neznamená, že rekurzívne definície automaticky zaručujú, že použitie rekurzívneho algoritmu bude najlepším spôsobom riešenia daného problému. Skutočne, objasnenie pojmu rekurzívny algoritmus prostredníctvom nevhodných príkladov, bolo najväčším dôvodom vzniku všeobecných obáv a antipatií proti použitiu rekurzcie pri programovaní. Súčasne vznikla domienka, že rekurzia je neefektívna. To samozrejme neznamená, že by sme sa mali zbaviť rekurzcie za každú cenu. Práve algoritmy kriviek vyplňujúcich priestor sú vhodné na aplikáciu pomocou rekurzcie, pretože sú tvorené pomocou predchádzajúceho kroku. Napríklad Hilbertova krivka je tvorená štyrmi krivkami z predchádzajúceho kroku, zmenšenými na polovicu.

5.2 Nerekurzívne riešenie kriviek vyplňujúcich priestor

Ako už bolo spomenuté krivky vyplňujúce priestor sa tvoria pomocou rekurzívnych algoritmov, pretože každý nasledujúci krok používa krivku z predchádzajúceho kroku. Preto je veľmi ťažké vytvoriť tieto algoritmy bez rekurzcie.

Pri tvorbe algoritmov bez rekurzcie si musíme krivku označiť indexami, od začiatku až po koniec krivky na každom bode, kde sa mení smer krivky. Algoritmy bez rekurzcie potom používajú tieto indexy na vypočítanie súradníc daného bodu. Keď chceme krivku pomocou takéhoto algoritmu nakresliť musíme najskôr všetky súradnice vypočítať, zapísať ich do tabuľky a až potom vykresliť. Príklad na takýto algoritmus bez rekurzcie môžete nájsť na [7].

6 Využitie kriviek vyplňujúcich priestor v praxi

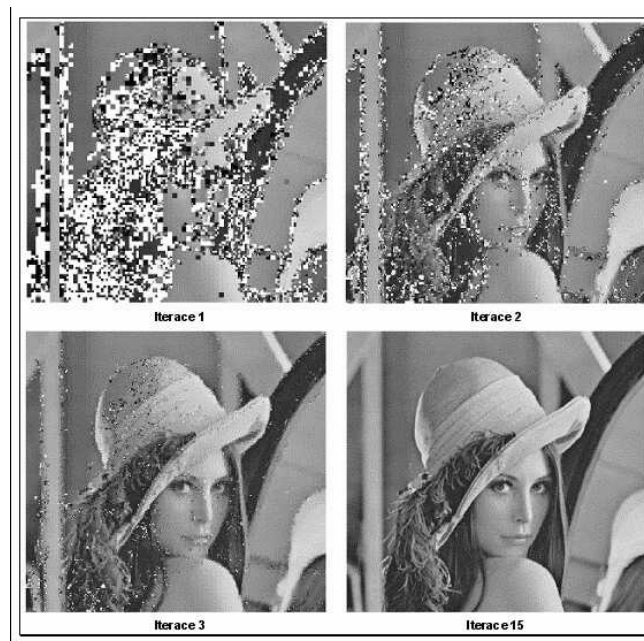
Krivky vyplňujúce priestor používame v súčasnej dobe na rôzne účely, hlavne v počítačových technológiách, napríklad v počítačovej grafike, viacdimenziálnych priestoroch na indexovanie[9], na preskupovanie výpočtov[10], alebo vo wireless sieťach [11]. Táto kapitola bližšie popisuje jednotlivé odvetvia využitia týchto kriviek.

6.1 Počítačová grafika

Krivky vyplňujúce priestor majú veľké zastúpenie v počítačovej grafike. Našli veľa uplatnenia v oblastiach pre prácu s informáciami. Kompresia dát je typickou oblasťou využitia týchto kriviek. Ďalej môžeme nimi generovať textúry, ktoré sú pamäťovo veľmi náročné, (niekedy až 10x menší objem dát) a netrpia skreslením, pri zväčšení ako bitmapové textúry. Sú vhodné aj pre kompresiu dát, keďže jedna transformácia je v tomto prípade zapísaná pomocou ôsmich čísel. Túto metódu využíva grafický formát FIF. Táto fraktálna kompresia je stratová a dosahuje pri vyšších stupňoch vizuálne lepšie výsledky, aj keď porovnanie nie je jednoduché z dôvodu odlišných prejavov straty kvality oproti napríklad JPEG. Príklad rekonštrukcie fotografie na obrázku 18.a pomocou čiernej plochy vidíme na obrázku 18.b. Pôvodný obrázok bol analyzovaný pomocou hierarchického členenia a boli vyhladané podobné pod oblasti pre jeho komprimáciu.



Obrázok 18.a: Pôvodná Lena



Obrázok 18.b: Rekonštrukcia obrázku Lena

6.2 Využitie fraktálnej analýzy pri hodnotení kvality tlače

Keďže krivky vyplňujúce priestor patria medzi fraktálne krivky, spomeniem niečo o fraktálnej analýze pri hodnotení kvality tlače.

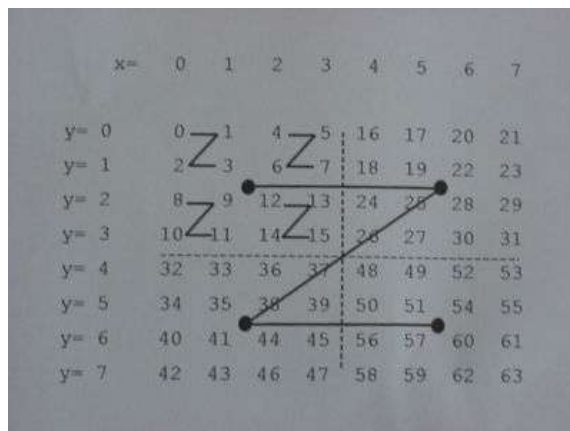
Pri hodnotení kvality tlače hrá v dnešnej dobe stále väčšiu úlohu obrazová analýza. Súvisí to so stále sa zvyšujúcou kvalitou a rozlišovacích schopností zariadení, ktoré vedú prevádzať obrazy do digitálnej podoby. Kvalita zaznamenaných obrazov závisí predovšetkým na kvalite záznamového zariadenia. Takže pre bezchybnú obrazovú analýzu je potreba mať čo najlepšie záznamové zariadenie, a taktiež záznam musí byť uložený v čo najlepšej kvalite, preto je treba použiť len bezstratovú kompresiu. Len pri splnení týchto podmienok môžeme vlastnou obrazovou analýzou získať parametre obrazov, ktoré objektívne vypovedajú o vlastnostiach povrchov potlačených materiálov a plôch.

Programy na spracovanie digitálnych fotografií používajú integrálne transformácie obrazových dát (a to buď periodických, resp. harmonických alebo waveletových). Tieto transformácie môžeme využiť k zisťovaniu fraktálneho charakteru čiar, plôch, respektíve povrchov a objemov 3D objektov. Pri fraktálnej analýze hrajú dôležitú úlohu dva parametre: Fraktálna dimenzia D a fraktálna miera K . Fraktálna dimenzia leží v intervale $D = \langle 0, E \rangle$, kde E je rozmer euklidovského priestoru, je to vlastne nám už známa Hausdorffova dimenzia. Pre $D \rightarrow 0$ je zmena štruktúry ako funkcie veľkosti mierky najväčšia (to je charakteristické pre obrazy, v ktorých analyzovaná štruktúra zaberá veľmi malý priestor, napríklad bod v priestore, na priamke) Pre $D \rightarrow E$ sa štruktúra nemení v závislosti na veľkosti mierky (analyzovaný objekt zaberá celý analyzovaný priestor). Fraktálna miera leží v intervale $K = \langle K_0 - K_{max} \rangle$, kde K_{max} je celkový počet elementárnych buniek v objekte. Pomocou fraktálnej miery sa dá určiť, ako je priestor zaplnený daným objektom. Napríklad Peanova krivka pokrýva 100% plochy obrazu to znamená, že Fraktálna miera bude rovná: $K/K_{max} = 1$. K určeniu fraktálnych parametrov sa veľmi často používa metóda „box-counting“. Túto metódu možno použiť k analýze binárnych dát, čiernobielych obrazov (farebných obrazov vytvorených prahovaním, t.j. priradením čiernej, resp. bielej farby podľa určitých kritérií v RGB, HSV priestore). Výsledkom obrazovej analýzy je tzv. „box-counting“ fraktálna dimenzia a fraktálna miera.

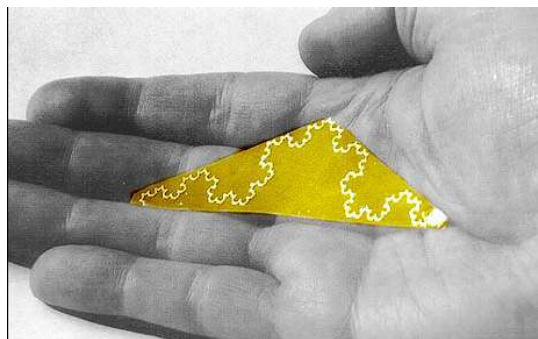
6.3 Ďalšie využitie kriviek vyplňujúcich priestor

Ako bolo uvedené v úvode tejto kapitoly, krivky vyplňujúce priestor majú veľké využitie v počítačovej technológii. Ďalšie využitie, ktoré nebolo v tejto kapitole popísané je napríklad indexovanie. Na to sa najčastejšie používa Mortonova (Z-order) krivka, aj pretože má jednoduchý priechod viacdimenziálnymi útvarmi [5]. Pekný príklad na to, ako sa indexuje v štvorci pomocou z-krivky je na obrázku 19. Ďalšie využitie kriviek je v projektovaní antén. Kvôli ich vlastnostiam

ako sú napríklad pri Kochovej krivke, že nekonečne dlhá krivka ohraničuje konečnú plochu sú využívané práve na tvorbu antén, pretože čím má anténa väčší povrch, tým ma lepšiu schopnosť vysielať alebo prijímať elektromagnetický signál. Anténa, ktorá vyzerá ako Kochova krivka je na obrázku 20.



Obrázok 19.: indexovanie pomocou Z-order krivky



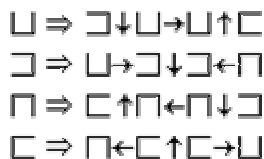
Obrázok 20.: Fraktálna anténa

7 Tvorba programu

Jedným bodom zo zadania bolo pokúsiť sa implementovať niektoré algoritmy kriviek. Vytvoril som krivky Hilbertovu, Peanovu, Moorovu a Z_order krivku. Program je vytvorený v programovacom jazyku C. K vykreslovaniu kriviek som použil grafickú knižnicu Gd. Je to jednoduchá knižnica, ktorá slúži na vytváranie jednoduchých obrázkov.

Program je rozdelený na dve časti. Prvý program vypočítava súradnice krivky podľa zadaných parametrov a vypíše ich na štandardný výstup. Druhý program spracuje tieto súradnice, pomocou grafickej knižnice Gd vykreslí obrázok s krivkou a uloží ho pod názvom „menokrivky_levelkrivky.gif“.

Prvý program sa volá krivky. Spúšťa sa s parametrami – meno krivky a level krivky. Po spracovaní týchto parametrov sa program podľa názvu krivky rozhodne, ktorú procedúru spustí. Súradnice všetkých kriviek sa vypočítavajú pomocou rekurzívnych algoritmov. Krivky sú tvorené rotáciou kriviek z predchádzajúcej iterácie a spojovacími čiarami. Na tom istom princípe funguje aj prvý program - krivky. Z hlavnej funkcie sa zavolá napríklad funkcia hilbert() s parametrami n, a UP kde n je level krivky, a UP je otočenie základnej krivky. UP preto, lebo prvý level krivky je otočený hore. Potom v tele funkcie sa opäť volá táto funkcia ale už s parametrami n-1 a otočením podľa toho ako je na rade. Medzi týmito malými krivkami sú ešte spojovacie čiary. Pravidlá otáčania a spojovacích čiar pre Hilbertovu krivku sú na obrázku 21.



Obrázok 21.: Pravidlá otáčania pre Hilbertovu krivku

Vo funkcii pre Moorovu krivku môžeme pekne vidieť ako sa tvorí z predchádzajúcej iterácie Hilbertovej krivky. Z hlavnej funkcie main je zavolaná funkcia moore s parametrami n a UP, ale v tele tejto funkcie sa nevolá rekurzívne funkcia moore, ale hilbert.

Program po vypočítaní súradníc vypíše na štandardný výstup najskôr meno krivky a level, aby program na vykreslenie vedel o akú krivku ide a potom vypíše samotné súradnice vo formáte [x,y].

Druhý program sa volá kresli. Spúšťa sa súčasne s programom krivky pomocou pipy. Program - kresli prečíta všetky údaje, ktoré vypíše program krivky na štandardný výstup, pomocou prvých dvoch údajov, ktoré sú názov krivky a level vytvorí súbor „menokrivky_levelkrivky.gif“, do ktorého pomocou načítaných súradníc vykreslí požadovanú krivku.

8 Záver

V mojej práci som uplatnil získané vedomosti z oblasti kriviek vyplňujúcich priestor a zhotovil som program, ktorý dokáže vykresliť niektoré krivky podľa zadaných parametrov. Je to jednoduchý program, ktorý sa môže rozšíriť pridaním ďalších kriviek, grafickým rozhraním a. i.

K vytvoreniu som použil programovací jazyk C a grafickú knižnicu Gd.

Štúdiom odbornej literatúry som si značne rozšíril vedomosti v oblasti počítačovej grafiky, fraktálnej geometrie a hlavne v oblasti kriviek vyplňujúcich priestor. Naučil som sa pracovať s grafickou knižnicou Gd.

Vzhľadom k tomu, že k tejto téme je málo dostupnej literatúry v podobe kníh, viac informácii som získaval priamo z internetu, alebo z článkov z rôznych vedeckých časopisov dostupných tiež na internete. Väčšina odborných článkov je uvedená v anglickom jazyku, preto bolo náročnejšie tieto informácie spracovať.

Verím, že táto práca bude prínosom aj pre študentov počítačovej grafiky a ľudí, ktorí sa zaoberajú fraktálnymi krivkami a krivkami vyplňujúcimi priestor.

9 Literatúra

- [1] Mandelbrot, B.B. (1982).: *The Fractal Geometry of Nature*. W.H. Freeman and Company. ISBN 0-7167-1186-9.
- [2] WWW stránky. Sebepodobnosť – wikipedia. URL: <http://en.wikipedia.org/wiki/Self-similarity>. Vyhľadane v januári 2008.
- [3] Sagan, H. (1994).: *Space-Filling Curves*. Springer-Verlag. ISBN 0387942653.
- [4] WWW stránky. Plane Filling Curves: Peano's & Wunderlich's – Cut The Knot. URL: <http://www.cut-the-knot.org/Curriculum/Geometry/Peano.shtml>. Vyhľadane v januári 2008.
- [5] WWW stránky. Z-order krivka – wikipedia. URL: http://en.wikipedia.org/wiki/Z-order_%28curve%29. Vyhľadane v januári 2008.
- [6] Guohua, Jin and Mellor-Crummey, John.: *SFCGen: A Framework for Efficient Generation of Multi-Dimensional Space-Filling Curves by Recursion*. Dokument dostupný online na URL: <http://portal.acm.org/citation.cfm?doid=1055531.1055537>. Vyhľadane v decembri 2007.
- [7] Moore, D. WWW stránky.: *Fast Hilbert Curve Generation, Sorting, and Range Queries* URL: <file:///c:/SKOLA/BP/materialy/Fast%20Hilbert%20Curves%20in%20C,%20without%20Recursion.htm>. Vyhľadane v januári 2008.
- [8] WWW stránky: The Hilbert Curve – CompuPhase. URL: <http://www.compuphase.com/hilbert.htm>. Vyhľadane v decembri 2007.
- [9] Lawder, J.K. and King, P.J.H.: *Using Space-filling Curves for Multi-dimensional indexing*. Dokument dostupný online na URL: <http://www.dcs.bbk.ac.uk/TriStarp/pubs/bncod17.pdf>. Vyhľadane v januári 2008.
- [10] Guohua, J. and Mellor-Crummey, J.: *Using Space-filling Curves for Computation Reordering*. Dokument dostupný online na URL: <http://www.cs.rice.edu/~jin/papers/lacsi05.pdf>. Vyhľadane v januári 2008.
- [11] Nasipuri, A.: *Serial Data Fusion Using Space-filling Curves in Wireless Sensor Networks*. Dokument dostupný online na URL: <http://www.cs.sunysb.edu/~samir/Pubs/serial.pdf>. Vyhľadane v januári 2008.
- [12] WWW stránky: Multidimensional Space-Filling Curves. URL: <http://www-swiss.ai.mit.edu/~jaffer/Geometry/MDSFC>. Vyhľadane v januári 2008.

10 Zoznam príloh

Príloha 1. Manuál.

Príloha 2. CD so zdrojovými textami.

Príloha 1

Táto príloha obsahuje návod ako pracovať s programom

Program krivky sa spúšťa s parametrami: -názov_krivky level krivky:

- Pre Hilbertovu krivku –hilbert
- Pre Peanovu krivku –peano
- Pre Moorovu krivku –moore
- Pre Z-order krivku –z_order

Príklad: `krivky.exe –hilbert 2` : vypíše súradnice pre Hilbertovu krivku level 2

`krivky.exe –z_order 5` : vypíše súradnice pre Z-order krivku level 5

Program kresli sa spúšťa bez parametrov ale musí sa spúšťať súčasne s programom krivky

Príklad: `krivky.exe –moore 3 | kresli.exe` vykreslí Moorovu krivku level 3 a uloží obrázok do súboru `moore_3.gif`

Programy s nastavené aby vykreslovali tieto 4 krivky až do levelu 6. Keď zadáte vyšší level program vypíše chybu a skončí.