Modelowanie proceduralne jako przykład zastosowań matematyki

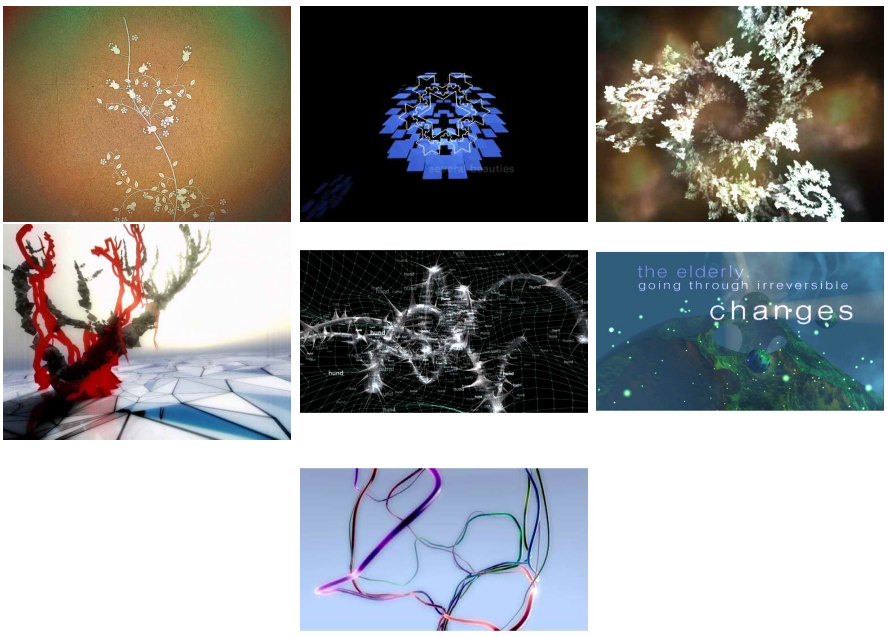
Artur Trzęsiok

10 listopada 2005

Spis treści

Streszczenie

Niniejsza praca jest refleksją na temat zastosowań matematyki w modelowaniu proceduralnym z którym autor spotkał się wielokrotnie w środowisku „Demosceny". „Demoscena" to grupa pasjonatów którzy używają komputera w sposób kreatywny (lecz niekomercyjny) i chcący się podzielić swoimi wynikami pracy z innymi lub (co zdarza się częściej) tworzyć z nimi w kooperacji. Efekty naszej pracy są różnorodne (grafika „per-pixel", grafika „true-color", grafika renderowana, muzyka „trackerowana", muzyka „strumieniowa", grafika ASCII, intra 4kB, intra 64kB, Dema – to tylko niektóre kategorie!) jednak nas najbardziej interesować będą **intra** ponieważ w nich w największym stopniu rozwinęło się zastosowanie modelowania proceduralnego – tutaj obecnie prawie każda nowa produkcja opiera się na zaawansowanych metodach matematycznych. W tej pracy będę chciał jakby „z lotu ptaka" przyjrzeć się w jaki sposób można opisywać za pomocą matematyki obrazy (a nawet animacje!) takie jak poniżej:



Rysunek 1: Przykładowe zrzuty ekranu z produkcji „Demosceny"

1 Wprowadzenie

Opisywać obiekty można na wiele sposobów - wierszem, muzyką, odwzorowywać je na płótnie... można użyć do tego matematyki. Pitagorejczycy wierzyli, że wszystko co wokół nas można opisać za pomocą liczb – już wtedy szukano proporcji, zasad, zapisanych w abstrakcyjnym języku matematyki. Gdy uda nam się przedstawić jakąś rzecz w ten sposób możemy dokonać jej bardzo precyzyjnej analizy za pomocą skomplikowanego aparatu jakiego dostarcza nam matematyka.

Mówiąc o obiektach takich jak *okrąg*, *prosta* czy *punkt* mamy na myśli obiekty idealne, nieistniejące oczywiście w przyrodzie. Jednak kropla wody wpadająca w spokojną tafle wody spowoduje stworzenie „okręgów" – takie uproszczenie (idealizacja) tego modelu do matematycznego pojęcia w większości przypadków jest wystarczająca. Mówiąc o modelowaniu mamy na myśli właśnie taką aproksymacje (z znaną precyzją) obiektów.

Modelować nie trzeba koniecznie przyrody (patrz rysunek 1 z streszczenia pracy). Nic nie stoi na przeszkodzie aby tworzyć modele zupełnie abstrakcyjne, już na poziomie swojej konstrukcji oparte nie na obserwacji a wyobraźni. Modele takie często są bardzo interesujące ze względów formalnych a także czysto estetycznych. Zdarza się, że tak „syntetycznie" stworzony model który jest uznawany jako piękny później znajduje zastosowanie w opisie przyrody.

W tej pracy pragnę przybliżyć zagadnienie modelowania geometrii.

Choć to zbyt duże słowa jak na tę prostą prace pozwolę sobie by korzystając w tym miejscu z pretekstu podzielić się znajomością dwóch cytatów które być może przyświecają ludziom którzy już nie hobbystycznie lecz zawodowo zajmują się modelowaniem:

> „Choć to niewiarygodne, nie ulega wątpliwości, że to, co głęboki umysł postrzega jako piękno, znajduje realizację w zewnętrznej naturze."
>
> S. Chandrasekhar

> „To, co wyobraźnia chwyta jako piękno, musi być prawdą – niezależnie, czy istniało przedtem, czy też nie."
>
> Keats

2 Kilka słów na temat różnych sposobów definiowania geometrii

W grafice komputerowej znamy trzy podejścia aby przechowywać geometrię:

Rastrowo zapisujemy geometrię z dokładnością do rozdzielczości rastra, format uniwersalny (można pod nim zapisać każdy obraz), duża ilość danych do przechowywania, nieprzystosowany do skomplikowanych modyfikacji (strata jakości)

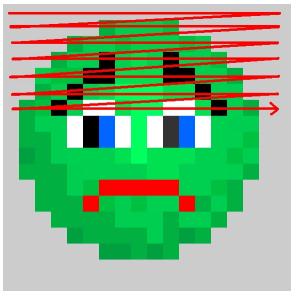
Wektorowo zapisujemy geometrię z dokładnością do złożoności obiektu, format nadający się do opisu kształtów i ewentualnie prostego ich cieniowania, względnie mała ilość danych do przechowania, modyfikacje dokonywane są bez straty jakości

Proceduralnie geometrię zapisujemy z dokładnością do używanego modelu, format nadający się do każdej rzeczy którą potrafimy opisać matematycznie, bardzo mała ilość danych do przechowania, modyfikacje dokonywane są bez straty jakości, a nawet jakość tą można dowolnie powiększać (zmniejszać)

Zrozumienie zasady działania metody rastrowej i wektorowej pomoże w uświadomieniu sobie dlaczego modelowanie proceduralne jest nie tylko interesują z punktu widzenia naukowego ale także praktycznego. Postaram się omówić najistotniejsze kwestie związane z koncepcją pierwszych dwóch by w ten sposób wprowadzić do zagadnienia korzystania z trzeciego.

2.1 Grafika rastrowa

Sposób zapisu obrazu polega na jego aproksymacji poprzez podział obszaru grafiki na uporządkowane jednokolorowe kwadraty (*piksele*). Dzięki uporządkowaniu potrafimy ułożyć poszczególne kwadraty w ciąg, a co za tym idzie musimy zapamiętać tylko kolor poszczególnych kwadratów bez ich położenia (tzn. położenie danego kwadratu jest jednoznacznie określone poprzez pozycje w ciągu). Sposób porządkowania używany przez karty graficzne jest następujący:



Rysunek 2: sposób uporządkowania pikseli w grafice z popularnego komunikatora internetowego w Polsce

Grafika rastrowa przypomina więc mozaikę złożoną z dużej liczby oddzielnych "płytek". Jak zatem widać dokładność aproksymacji obrazu zależeć będzie od dwóch czynników: ilości kwadratów przypadających na dany fragment oraz precyzji w zapisie koloru (będzie o tym w dalszej pracy). Jednak aby uzyskać dobrej jakości obraz do wyświetlenia na monitorze należy na zapamiętanie takiego obrazu poświęcić $2\frac{1}{4}$ mb. Ze względu na tak dużą ilość danych większość formatów rastrowych wykorzystuje algorytmy kompresujące. Przyglądając się np. zdjęciu zachodzącego słońca łatwo zauważymy analizując zmianę koloru w poszczególnych wierszach nieba ze można zadowalająco przybliżyć ją za pomocą jakiegoś wielomianu, w ten sposób zastępujemy przechowywanie np 800 kolorów pamiętaniem kilku współczynników wielomianów (im ich więcej tym lepsza jakość kompresji).

Modyfikacja obrazu rastrowego polega na modyfikowaniu kolorów poszczególnych kwadratów. Zauważmy jednak, że gdy dokonujemy jednak na tym formacie operacji obrotu to będziemy musieli odwrócone kwadraty dostosować do układu ekranu czego nie da się wykonać bez dosyć poważnych przybliżeń, a co za tym idzie straty jakości!

Największym plusem formatu rastrowego jest jego uniwersalność. W formacie tym możemy zapisać zarówno zdjęcia jak i schematy czy nawet tekst. Z powodu tej uniwersalności rozwiązanie takie stosowane jest w telewizorach czy monitorach i dzięki temu możemy na nich oglądać dowolne obrazy.

2.2 Grafika wektorowa

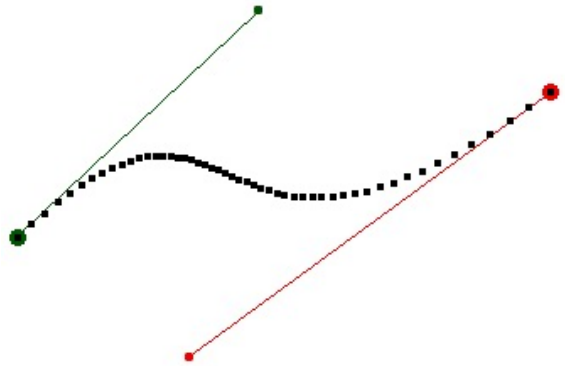
Sposób zapisu obrazu polega na opisaniu go poprzez zbiór tzw punktów kontrolnych. Podstawowym obiektem grafiki wektorowej jest odcinek którego nie musimy aproksymować żadnymi kwadratami lecz wystarczy byśmy podali współrzędne jego końców i za pomocą interpolacji liniowej możemy wskazać wszystkie jego punkty. Gdy zastanowimy się przez chwile jak musiałby wyglądać zapis odcinka przez format rastrowy i jakie komplikacje wynikałyby w momencie gdybyśmy takim obrazem chcieli manipulować (wystarczy, że chcielibyśmy go obrócić o kąt 17 stopni i powiększyć) od razu zrozumiemy jaką przewagę daje nam zapis odcinka w postaci wektorowej. Zauważmy, że taka operacja jest wykonana z precyzją wyłącznie do obliczeń! (a nie do metody). Zapisywanie (a co gorsza tworzenie!) skomplikowanych obiektów za pomocą samych odcinków byłoby bardzo kłopotliwe dlatego w grafice wektorowej wykorzystuje się różnego rodzaju krzywe wielomianowe wyższych stopni. Najbardziej popularne to krzywe Beziera, Hermite'a czy Nurbs. Dla bardziej zaciekawionych podajmy wzory na interpolację Beziera na płaszczyźnie pomiędzy dwoma punktami:

$$x(t) = a_x t^3 + b_x t^2 + c_x t + x_0$$
$$y(t) = a_y t^3 + b_y t^2 + c_y t + y_0$$

gdzie:
$c_x = 3(x_1 - x_0)$
$b_x = 3(x_2 - x_1) - c_x$
$a_x = x_3 - x_0 - c_x - b_x$
$c_y = 3(y_1 - y_0)$
$b_y = 3(y_2 - y_1) - c_y$
$a_y = y_3 - y_0 - c_y - b_y$

$t \in [0, 1]$ – parametr
$(x_0, y_0), (x_1, y_1), (x_2, y_2), (x_3, y_3)$ – współrzędne punktów kontrolnych

Zmieniając liniowo parametr t od 0 do 1 o jakiś stały przyrost otrzymujemy ciąg punktów które można teraz połączyć odcinkami otrzymując krzywą.



Rysunek 3: interpolacja Beziera

Zauważmy, że mamy pełną kontrole precyzji otrzymanej prostej, jeżeli tylko zapragniemy ją zwiększyć wystarczy by zmniejszyć krok interpolacyjny t. Składając teraz takie kawałki krzywych ze sobą możemy opisać z zadawalającą dokładnością każdy kształt. Możemy także stworzyć krzywą zamkniętą która następnie wypełnimy jakimś kolorem (niekoniecznie jednolitym... wypełnianie kolorem może być także interpolowane po jakiejś krzywej).

Uzyskana w ten sposób grafika jest bardzo dobrze przystosowana do modyfikacji gdyż wszelkie obroty, skalowania czy przesunięcia (czy nawet bardziej skomplikowane) sprowadzają się do przesunięcia punktów kontrolnych a następnie wyliczenia na nowo punktów pośrednich.

Technika ta jednak nie nadaje się do wydajnego opisu wielu obiektów np przetworzenie zdjęcia na format wektorowy jest praktycznie niemożliwe (pomysły takie były już dawno - spójrzmy na witraże: próba ujęcia obrazu w kształty „geometryczne"). Zaletą niebywałą tej grafiki jest niewielka ilość

danych potrzebnych do zapisu obrazu. Znalazło to zastosowanie w technologiach internetowych gdzie obecnie coraz więcej stron widzimy wykonanych w np programie „Flash".

2.3 Grafika proceduralna

Zarówno w przypadku grafiki rastrowej jak i proceduralnej zastanawialiśmy się w jaki sposób możemy zachować obraz na komputerze oraz omawialiśmy podstawowe cechy każdego z tych podejść. Grafika proceduralna to już nie tylko zagadnienie przechowywania czy możliwości edytowania grafiki lecz zagadnienie jej generowania.
Co rozumiemy przez generowanie? Przecież w wypadku grafiki proceduralnej komputer także generował punkty na podstawie parametrów zadanych przez człowieka.
Jest to prawda jednak w tamtym przypadku algorytmy operowały na poziomie kształtu, pojedynczej linii. Procedury były używane tylko „lokalnie" za całość obrazu w decydował grafik. Można posłużyć się porównaniem ze algorytm był tutaj tylko narzędziem tworzyła zaś ręka ludzka. Istota grafiki proceduralnej polega na tym, że ostateczny efekt jest w całości stworzony przez algorytmy, a nasz proces tworzenia polega nie na podawaniu parametrów do wzorów (punktów kontrolnych) lecz na samym tworzeniu i projektowaniu algorytmów! Tzn odwołując się do przykładu krzywej Beziera nie zastanawiamy się już za bardzo nad konkretnymi parametrami by krzywa „wyszła ładna" lecz zastanawiamy się jak zmodyfikować ten wzór aby uzyskać porządany efekt.

Wszystko to brzmi może nieco niezrozumiale, podajmy więc jakiś przykład. Dostaliśmy zlecenie by dostarczyć obraz chmury *Cirrocumulus* techniką rastrową, posiedzieliśmy kilka kwadransów nad naszym ulubionym programem do malowania i otrzymaliśmy ładny obrazek. Za jakiś czas jednak znów okazało się, że potrzebne są chmury, ale niestety nie *Cirrocumulus* lecz *Nimbostratus* więc cóż robić - znów trzeba było siąść i rysować. Po kilku podobnych sytuacjach mamy już na naszym koncie rysunki: *Cumulonibus, Cumulus, Stratocumulus* i *Cirrostratus*, a co najgorsze okazało się, że potrzeba jakiś obrazków w dużej ilości egzemplarzy i brakuje czasu więc pracować trzeba w pośpiechu co obniża jakość naszych rysunków...
My, rzecz jasna, się cieszymy bo mamy pracę, ale jak to wygląda ze strony pracodawcy mówić nie trzeba. Co przy takim doświadczeniu przychodzi na myśl? Ano zastanawiamy się czy nie istnieje jakiś sposób by komputer za nas generował te chmury na podstawie zadanych parametrów (stopień kłębiastości, wysokość nad poziomem morza, siły wiatru jaki na nie działa etc) i to najlepiej w taki sposób aby opierał się na jakimś losowym parametrze by każdy wynik różnił się od poprzedniego choć zachowywał te same parametry! Sposób oczywiście istnieje, lecz komputer sam z siebie nic nie wygeneruje – to my musimy przygotować dla niego procedure, i właśnie tworzenie na

poziomie opracowywania tych procedur nazywamy modelowaniem proceduralnym któremu teraz już chciałbym poświęcić resztę pracy.

3 „Fractals everywhere"

Tytuł rozdziału to tytuł książki Michael F. Barnsley'a lecz równie dobrze można było umieścić tytuł publikacji Benoit B. Mandelbrot'a „The Fractal Geometry of Nature". Sugeruje to, że modelowanie proceduralne grafiki jest równoważne zagadnieniu fraktali co nie jest prawdą i chciałbym już zawczasu wytłumaczyć dlaczego właśnie taki tytuł.

Jak ciągle mówiłem w poprzednim rozdziale *modelowanie proceduralne* to dosłownie „modelowanie procedurami" – nikt nie powiedział, że procedury owe prowadzą zawsze do uzyskiwania fraktali. Faktem jednak jest, że te najciekawsze, najbardziej zaskakujące efekty w modelowaniu proceduralnym powstają w oparciu o układy rekurencyjne. Dlaczego? Wytłumaczenie jest proste... dlatego, że w układach dynamicznych najtrudniej jest nam „oszacować" wynik końcowy. Modelowanie (o samym słowie modelowanie będzie jeszcze niebawem) za ich pomocą nie jest zatem jak rozwiązywanie ciągle tej samej zagadki tylko z różnymi danymi lecz bardzo często będzie nas zaskakiwało. Co jest tym zaskoczeniem? To, że za pomocą bardzo prostych operacji matematycznych możemy otrzymać tak zadziwiająco skomplikowane struktury i to nie tylko abstrakcyjne lecz także takie które bezpośrednio odnoszą się do natury...

3.1 Parę słów na temat modelowania

Na pytanie co to jest model mogliśmy się dowiedzieć w wystarczającym stopniu dla zrozumienia tego tekstu z „Wstępu", pozostało jeszcze zagadnienie modelowania. Modelowanie jak się łatwo domyśleć to proces tworzenia modelu wyróżniamy jednak w nim kilka etapów:

1. Wyróżnienie tych parametrów które będziemy uwzględniać w naszym modelu
2. Ustalenie relacji pomiędzy parametrami
3. Przeprowadzanie symulacji
4. Wyciąganie wniosków na podstawie wyników symulacji

Na każdym z tych etapów można dokonać błędu który zniweczy cały nasz wysiłek. Punkt pierwszy można zilustrować w sposób następujący: przeprowadzamy symulacje spadającego jabłka z drzewa, zastanawiamy się co będziemy brali pod uwagę i wybieramy: przyciąganie ziemskie, wysokość jabłka nad ziemią, opór powietrza. Te 3 parametry wystarczą by w sposób zadowalający nas opisać ruch tego jabłka pomimo, że nie uwzględniamy siły

z jaką oddziałowuje na te jabłko Pluton (a wiemy na podstawie praw fizyki, że on także ma wpływ na ruch jabłka). Drugi punkt to w naszym przypadku zastosowanie kilku wzorów z zakresu fizyki z szkoły średniej. Trzeci punkt to uruchomienie odpowiedniego programu (który powstał na bazie punktu drugiego!). Ostatni punkt jest oczywisty.

3.2 Rozejrzyjmy się wokół siebie

Zanim jednak wykonany zalecenie tego podrozdziały wprowadźmy dwa istotne pojęcia:

Model rekurencyjny to taki którego kolejny stan powstaje na bazie swojego poprzedniego stanu (i tylko tego jednego). Algorytmy rekurencyjne są tak obecne w naszym życiu, że ich nie zauważamy np. gdy zbieramy rozrzucone klocki z podłogi to postępujemy w sposób rekurencyjny: zbieramy do ręki tyle klocków ile się w niej mieści i odkładamy a następnie czynność powtarzamy aż do momentu gdy ilość klocków na podłodze równa jest zero.

iteracja to jeden z takich „stanów" modelu rekurencyjnego.

iterowanie to przechodzenie do kolejnych iteracji w modelu rekurencyjnym

Zacznijmy od przykładu... kulinarnego. Benoit B. Mandelbrot podaje jako przykład fraktala w swojej książce kwiat kalafiora!

Rysunek 4: Kwiat kalafiora jest rekurencyjny

Jeżeli przyjrzymy się chwilkę chmurom także zauważymy wiele cech samopodobieństwa

Linia brzegowa, powierzchnia gór, gałęzie drzew, budowa płuc... jeżeli tylko rozejrzymy się wokół siebie odnajdziemy wiele struktur o budowie rekurencyjnej. Wiele zaś z tych struktur przekształca się na skutek działania konkretnych sił które potrafimy opisać za pomocą fizyki...jeżeli umiemy zatem coś „zbudować" i nadać temu właściwości można przypuszczać, że jest możliwe całkiem zadowalające modelowanie rzeczywistości właśnie za pomocą wyłącznie procedur.

Napisałem, że „za pomocą bardzo prostych operacji matematycznych możemy otrzymać tak zadziwiająco skomplikowane struktury" – w kolejnym podrozdziale będę chciał przedstawić kilka konkretnych przykładów które właśnie dla mnie były takim „zadziwieniem".

3.3 Przykłady modeli

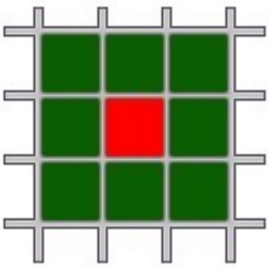
Pozwolę sobie jednak na krótki wstęp który jest co prawda tylko przypomnieniem, ale chciałbym to podkreślić. Zakładam, że każdy z nas zna doświadczenie jakie towarzyszy nam gdy w trakcie rozwiązywania np. zadania z geometrii w pewnym momencie spośród praktycznie nieskończonej ilości możliwości postępowania wyłania się nam ta jedna która poprowadzi do rozwiązania. Jest to bardzo nietrywialne przeżycie. Jego cechą jest to, że z mnogości i skomplikowania wyłania się proste i jasne. Natomiast to „zadziwienie" którym teraz będę próbował się podzielić dotyczy czegoś odwrotnego – gdy z bardzo „prostego i jasnego" będą wyłaniać się niespodziewanie

bardzo skomplikowane struktury.

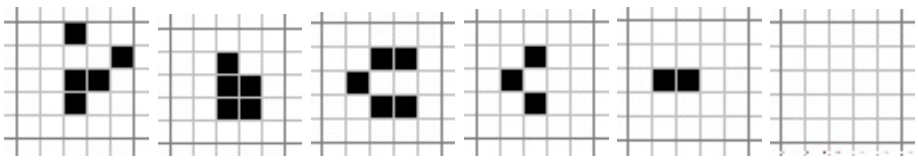
3.3.1 Model gry „Life”

Gra „Life” jest jednoosobowa, rozgrywana na planszy o polach kwadratowych (bardzo często jest to plansza do Go) oraz przy użyciu jednego rodzaju pionków (kamieni). Gra jest iteracyjna, początkowy układy ustala gracz a następnie kolejne uzyskuje poprzez iterowanie. W „Life” są następujące zasady:

1. Na jednym polu może życie być lub nie (zasada binarna, poprzez życie rozumiemy, że jest na danym polu pion)
2. Poprzez komórki sąsiadujące z komórką czerwoną rozumiemy takie komórki zielone:



3. W komórce rodzi się życie jeżeli sąsiaduje z dokładnie 3 żywymi komórkami
4. W komórce podtrzymuje się życie jeżeli sąsiaduje z 2 lub 3 żywymi komórkami
5. W każdym innym przypadku pole pozostaje puste (albo nikogo tam nie było, albo umiera z przeludnienia, albo z samotności)



Rysunek 5: Przykład kolejnych iteracji układu

Model gry „Life” jak widać jest bardzo prosty a jak można bez trudności się przekonać korzystając z ułatwień programów komputerowych rozwój koloni life wcale nie jest taki „prosty do przewidzenia”. Okazuje się ponadto, że

istnieją konfiguracje które nie mają swojego poprzednika. Podałem przykład gry „Life" ze względów historycznych była to jedna z pierwszych symulacji osiągalnych na domowych komputerach a i w ogóle przeprowadzonych przez komputery.

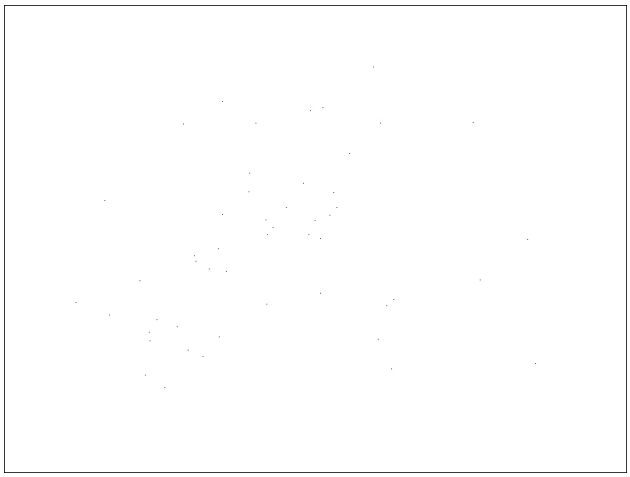
Kto nie miał styczności z tą grą wcześniej może być zdziwiony strukturami które ona generuje.

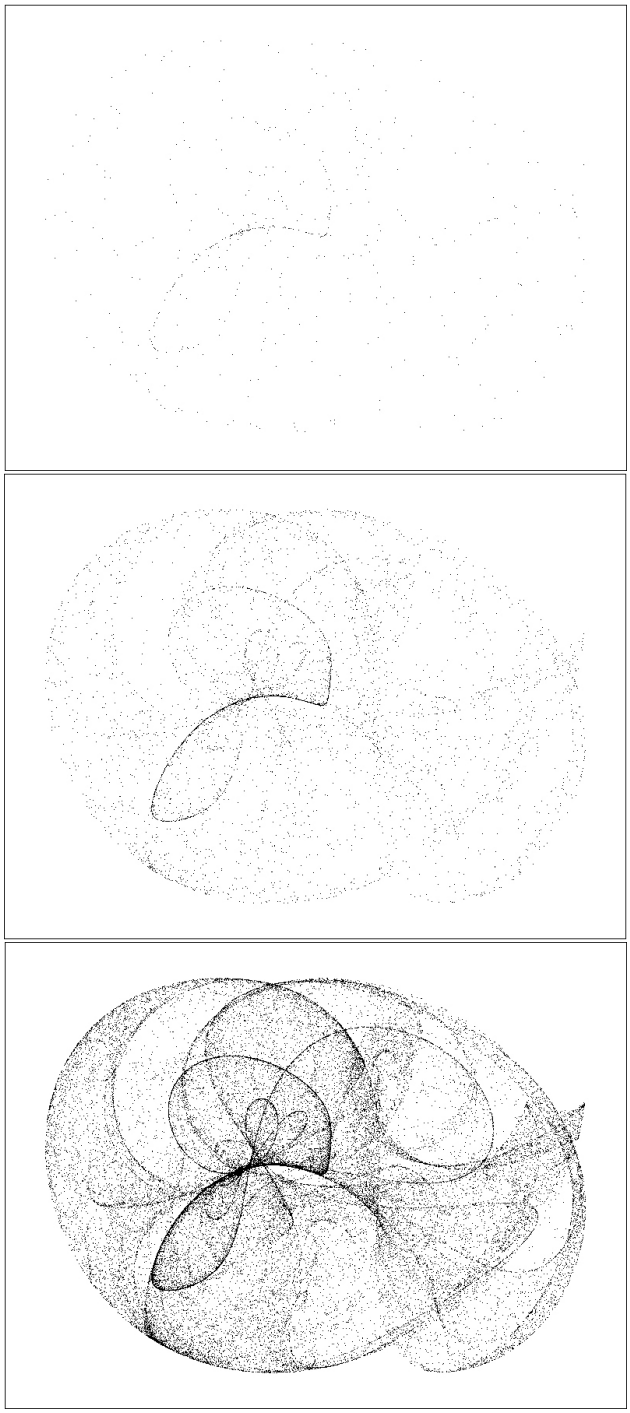
3.3.2 Atraktor Clifford'a

Atraktor Clifford'a powstaje poprzez rozwijanie kolejnych elementów ciągu rekurencyjnego dwóch zmiennych. Uzyskiwane wyrazy ciągu (punkty) zaznaczamy na wykresie.

$$x_{n+1} = \sin(a \cdot y_n) + c \cdot \cos(a \cdot x_n)$$
$$y_{n+1} = \sin(b \cdot x_n) + d \cdot \cos(b \cdot y_n)$$

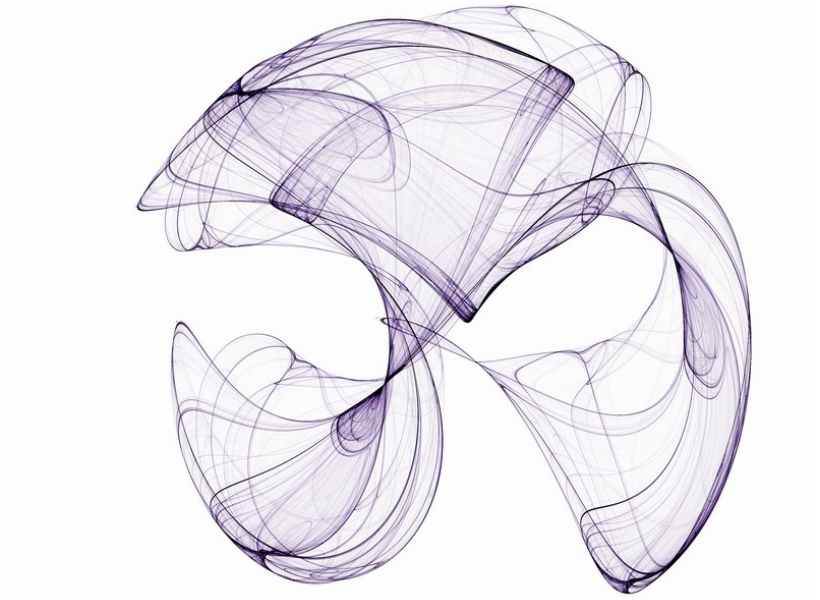
a, b, c, d – parametry definiujące atraktora
x_0, y_0 – pierwszy wyraz ciągu ustalamy $(0, 0)$
Ustalmy:
$a = 1.5$
$b = -1.8$
$c = 1.6$
$d = 0.9$
Zobaczmy jaki otrzymujemy wykres dla kolejno 50,500,5000,50000 iteracji:

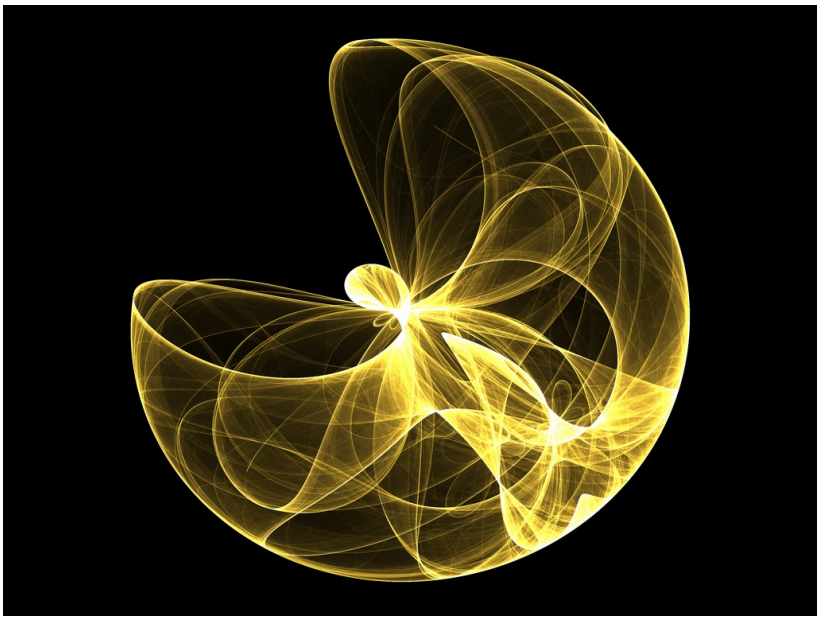




Powiększmy ilość iteracji do kilku miliardów i nie zamiast rysować punkty w danym miejscach będziemy rozjaśniać tło w danym miejscu (coś jakby naświetlanie). Otrzymamy w ten sposób wykres prawdopodobieństwa występowania danego wyrazu ciągu w tych kilku miliardach (wyraz ciągu to punkt

a prawdopodobieństwo będzie stopniem jasności tego punktu) co otrzymamy?





Parametry a, b, c, d definiują zupełnie innego atraktora. Poziom złożoności struktury jaką otrzymaliśmy w zestawieniu z bardzo prostym wzorem rekurencyjnym ukazuje siłę modelowania za pomocą algorytmów.

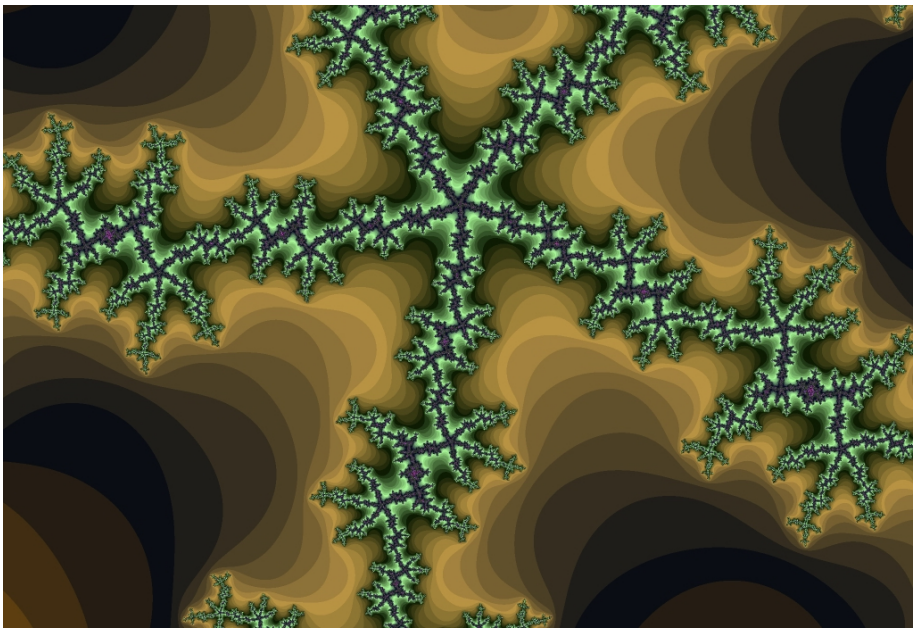
3.3.3 Fraktal Manbelbrota

Przykład bardzo „wyeksploatowany" ale nadal zadziwiający. Algorytm jest następujący:
Początkowo:
$x_0 = y_0 = 0$
Wzór rekurencyjny:

$$x_{n+1} = x_n^2 - y_n^2 + x_w$$

$$y_{n+1} = 2x_n y_n + y_w$$

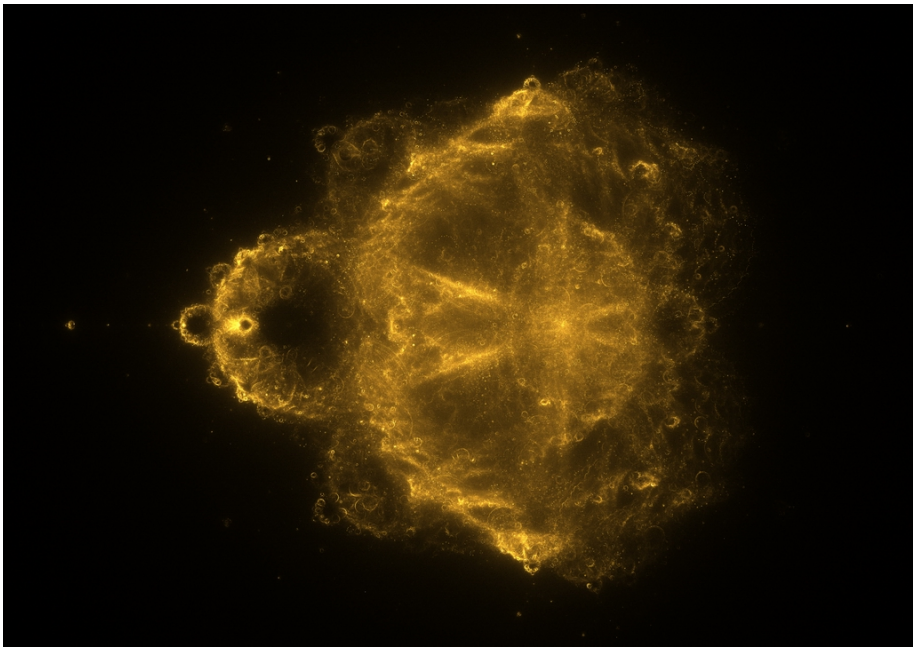
gdzie:
x_w, y_w – współrzędne dla którego poszukujemy wartości
Iterujemy aż do momentu aż nasz punkt nie będzie się znajdować się w kole o promieniu 1 i środku (0,0) – w zależności od ilości iteracji które trzeba wykonać by to osiągnąć ustalamy kolor punktu x_w, y_w



Rysunek 6: fraktal Mandelbrota

3.3.4 Fraktal Buddabrota

Odmiana Fraktala Mandelbrota realizowana w ten sposób, że w czasie iterowania do „ucieczki" z koła rozjaśniamy punktu który wskazuje dany wyraz ciągu.



Rysunek 7: fraktal „Buddabrota”

3.3.5 Osadzanie elektrolityczne

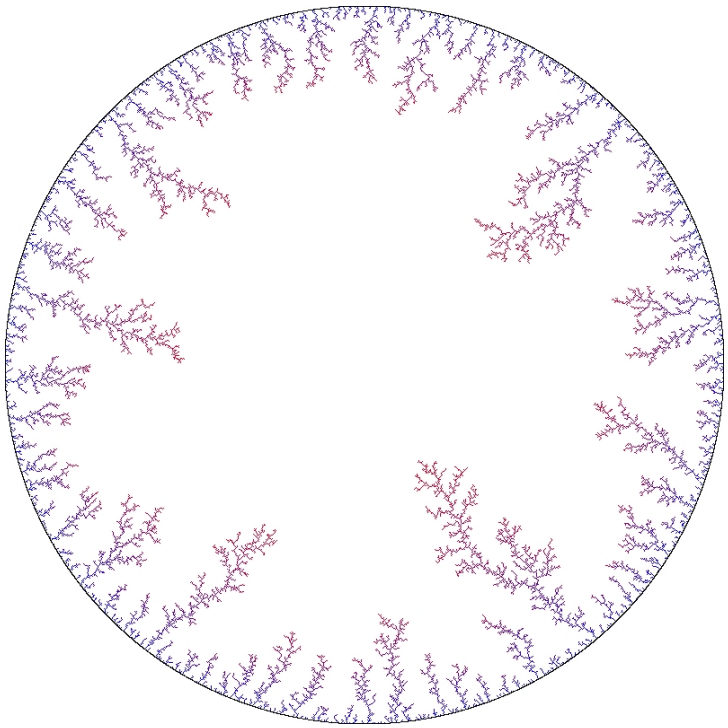
Postępujemy wobec następującego algorytmu:

Start:
Ustaw na środku();
Pętla:
Losuj kierunek(); //jeden z czterech
Przesun sie w danym kierunku();
Sprwadz czy nie jestes kolo osadzonego();
Jeśli tak: osadz sie();
Jeśli nie: idź do pętla();

efekt:

3.3.6 Szum Perlina

...



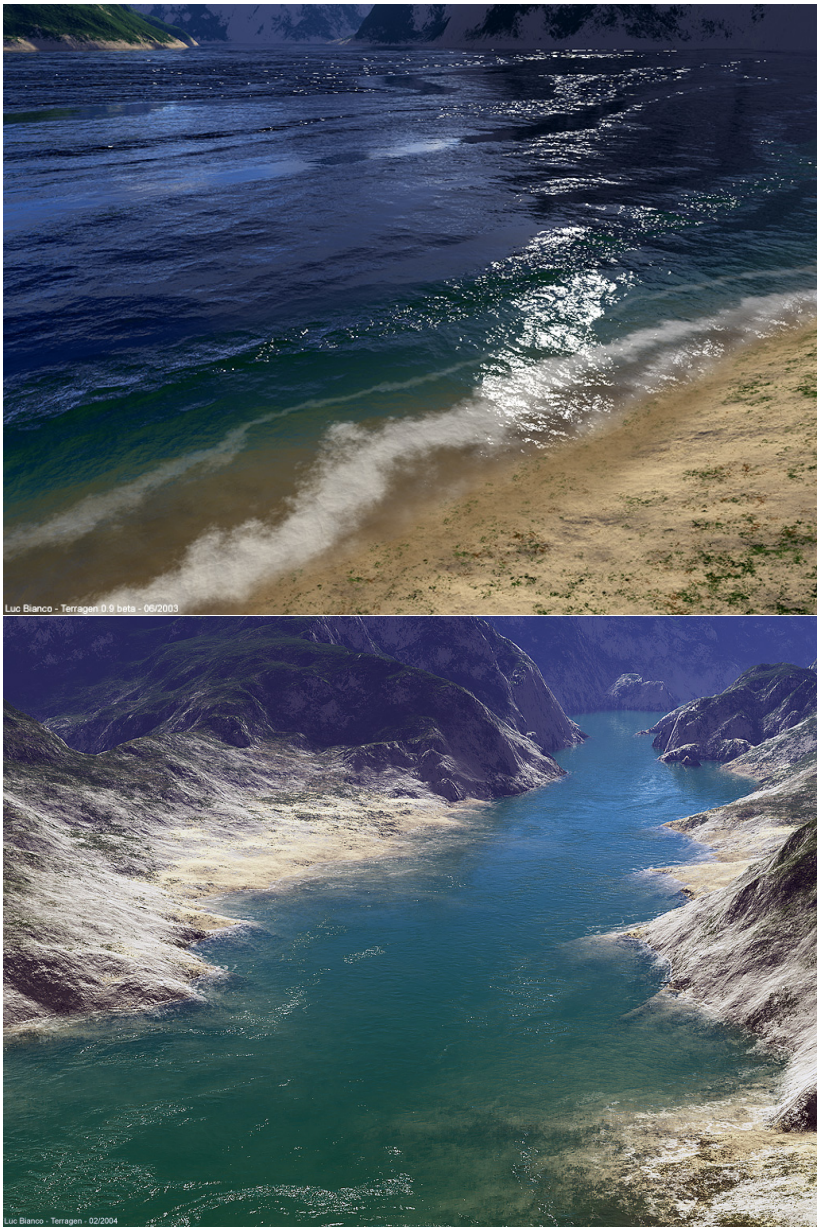
Rysunek 8: Osadzanie elektrolityczne

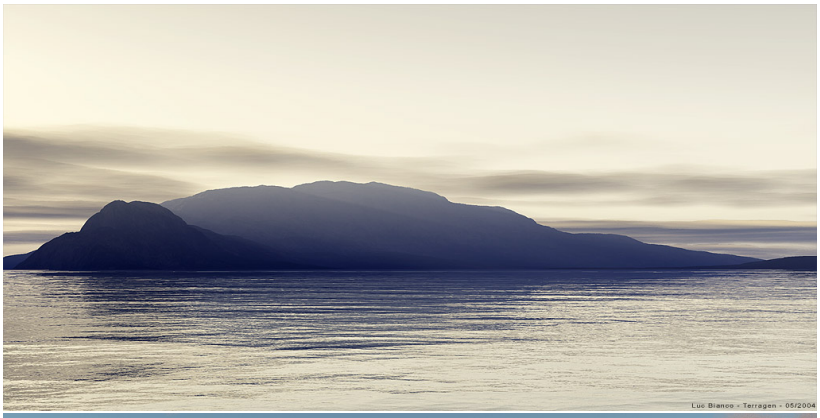
3.4 Zastosowania

Jak się ma to wszystko do tematu modelowania proceduralnego? Pokazaliśmy kilka konkretnych przykładów do czego mogą prowadzić próby modelowania rekurencyjnego, zobaczyliśmy, że efekty są skomplikowane, choć powstawały na bazie prostych przekształceń. Na samym początku mówiliśmy o grafiku który rysował chmury... można mu zatem ułatwić tą prace i jeżeli tak to w jaki sposób? Co prawda tylko ostatni przykład (osadzanie elektrolityczne) był bezpośrednio związany z modelowaniem przyrody to techniki pozostają bardzo zbliżone. Można... oczywiście, chmury są nawet fraktalem więc dają się opisać za pomocą matematyki. Potrafimy stworzyć program który będzie generował chmury, potrafimy stworzyć program który wygeneruje cały las tak by żadne dwa drzewa nie były identyczne a zarazem każde bardzo realistyczne. Potrafimy opisywać świat z coraz zadowalającą precyzją i nie najważniejsze chyba są tutaj wymierne tego korzyści...

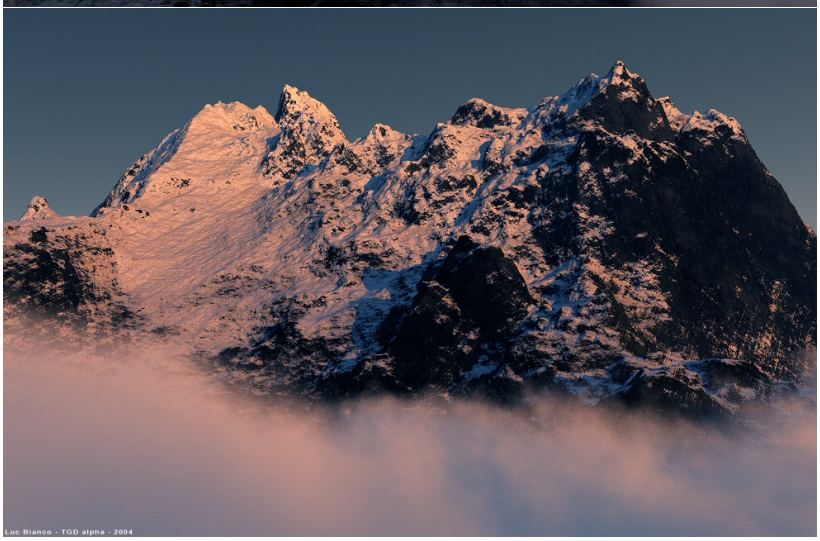
4 Zakończenie

Na zakończenie chciałbym umieścić kilka prac wygenerowanych w całości za pomocą algorytmów aby ukazać jak dalece dziedzina ta rozwinęła się.


Luc Bianco - Terragen 0.9 beta - 06/2003
Luc Bianco - Terragen - 02/2004


Luc Bianco - Terragen - 05/2004


Luc Bianco - TGD alpha - 2004


Luc Bianco - TGD alpha - 2004

Na zakończenie chciałem powiedzieć, że totalnie źle rozplanowałem czas i potem już nad tym przestałem panować i jest to praca która nie powinna być dzisiaj komukolwiek oddana, ale liczę trochę na to, że to informatyka a nie j. polski.

Artur T.