

# MODELY PŘÍRODY

I.Kolingerová

## Základní typy modelovaných objektů

- Dendrity, korály
- Pobřeží
- Krajiny
- Planety
- Oblaka
- Rostlinné ekosystémy
- Oheň, kouř, voda

2 hlavní způsoby modelování přírodních prvků:

- simulace fyzikálních procesů
- napodobování výsledného vzhledu

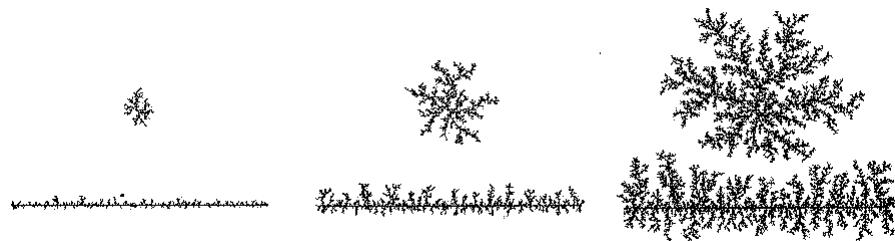
## Modelování dendritů, korálů: Difúzní omezená segregace (DLA)

- Přímá aplikace Brownova pohybu
- El. výboje, obrazce na zamrzlých sklech, korály...
- Roztok s kondenzačním jádrem a plovoucími molekulami
- Pohybující se molekula zachycena jádrem – stane se také kondenzačním jádrem
- Difúzí nová jádra a nový tvar

## DLA

- V 2D vznik fraktál. struktur s dimenzí kolem 1.7
- 2D: na začátku 2D matice, nenulové položky – kondenzační jádra
- Cyklus: částice na okraj matice, Brownův pohyb
- Částice se po doputování ke kondenzačnímu jádru k němu připojí, nastavíme prvek matice jako „obsazen“
- Částice mimo obrazovku: zastavit její trasování, nová částice

## DLA



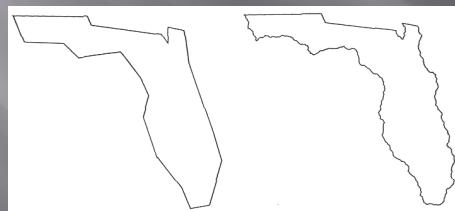
Kondenzace z bodu, kondenzace na úsečku [Zar04]

## DLA

- Modifikace:
  - připojit molekulu k jádru až po několika zásazích (čítač doteků na povrchu)
  - připojit molekulu s určitou pravděpodobností
- Zdlouhavé (exponenciální složitost), i když se zaplňováním oblasti urychlení
- Urychlení: vně minmax boxu jádra rychlejší pohyb

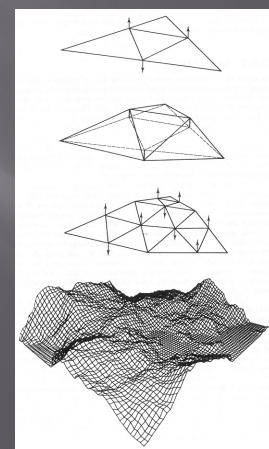
## Modelování pobřeží: náhodné přesouvání (střed.) bodu

- Ostrovy u pobřeží – nutno modelovat samostatnými objekty
- Viz přednáška o fraktálech



## Modelování krajin

- Náhodné přesouvání středního bodu: viz přednáška o fraktálech
- Další možnosti: viz planety



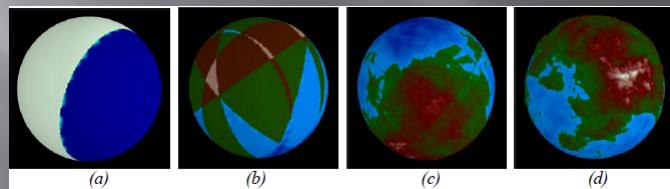
## Modelování krajin

- Výsledky z programu Terragen [TerG]



## Modelování planet: náhodná chyba

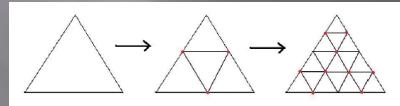
- Vstup: mesh v podobě koule
- 1 iterace: koule se rozřízne náhodně zvolenou rovinou na dvě polokoule, poloměr jedné se náhodně zvětší, druhé zmenší



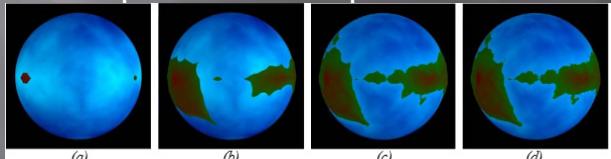
a) Po 1 iteraci, b) po 10 iteracích, c) po 100 iteracích, d) po 1000 iteracích [Lin07]

## Modelování planet: náhodné přesouvání (středního) bodu

- Vstup: mesh v podobě koule (stačí 2 čtyřstěny)
- 1 iterace: trojúhelník se rozdělí na 4 menší,



nový vrchol dostane výšku průměrem rodičů + náhodně vygenerovaný přesun zmenšovaný podle už provedeného počtu dělení



a) Po 1 iteraci, b) po 3 iteracích, c) po 5 iteracích, d) po 7 iteracích [Lin07]

## Modelování planet: multifraktální náhodné přesouvání (středního) bodu

- Pro větší věrohodnost různá úroveň detailu v různých částech – hodnota posunutí se nejen zmenšuje podle provedeného počtu dělení, ale také násobí průměrem výšky rodičů => větší změny ve větších nadmořských výškách

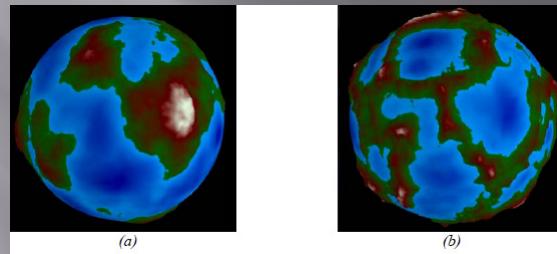
Př.: Offset = random (-Amp,+Amp);  
 $\text{Offset} = \text{Offset}/2^{\uparrow L};$   
 $\text{Offset} = \text{Offset} + \text{Offset} * \text{average (parents)} * k;$

k – změna měřítka výšky

## Modelování planet: Perlinův šum

- Vstup: libovolně definovaná koule (obvykle mesh)
- Stačí jen 1 iterace
- Výška bodu ( $x,y,z$ ) na kouli se změní o hodnotu 3D Perlinova šumu v tomto bodě
- Varianta 1: multifraktál – ve výpočtu Perlinovy funkce se zahrne výška terénu daná zatím spočítanou částí Perlinovy funkce
- Varianta 2: tzv. hřebenový P.šum – mírně upravená funkce produkuje delší, hubenější ostrovy, poloostrovy, hřebeny hor

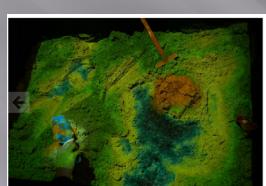
## Modelování planet: Perlinův šum



a) Normální Perlinův šum, b) hřebenový Perlinův šum [Lin07]

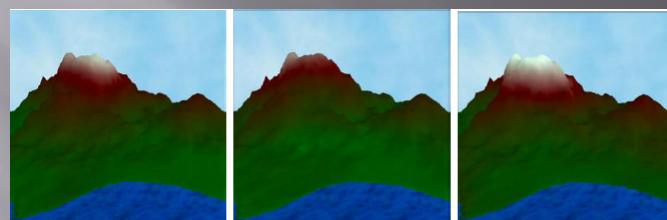
## Obarvení modelu planety

- Základ: obarvení podle výšky, viz zeměpisná škála nebo pískoviště s Kinectem



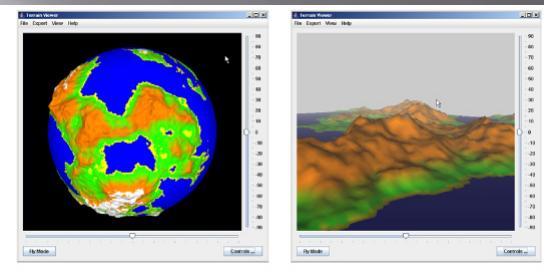
- Mezi barvami interpolace lineární nebo splinem + mírná náhodnost, např. přes Perlin. šum nebo náhodná odchylka, viz částicové systémy

## Obarvení modelu planety

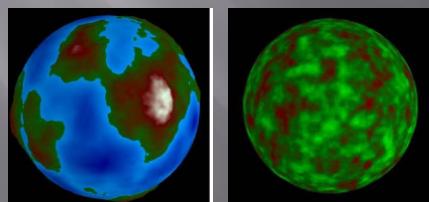


Obarvení podle výšky s různými náhodnými perturbacemi [Lin07]

## Modelování planet



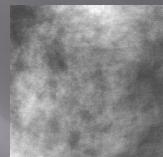
Výsledek z programu TerraJ [TerJ]



Výsledky s užitím Perlinova šumu [Lin07]

## Modelování oblaků: náhodné přesouvání (střed.) bodu

- 2D mraky: kolmý průmět fraktální plochy, výška reprezentována barvou



- 3D: rozšířit o 1 dimenzi, 4.rozměr – hustota
- Zobrazení 3D do 2D: ray tracing nebo animace pro pohybující se mraky
- Oblaky většinou slouží jako pozadí, proto často billboardy a polopřůhledné vrstvy
- Nejobecnější metoda: 3D šumové funkce reprezentované jako objem

## Modelování oblaků 3D buněčným automatem

- [Dob00]
- Jednoduché, rychlé, nepřesné, umí jen mraky typu cumulus

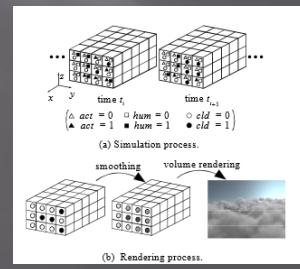


Obr.: Wikipedie

- Mraky - bublinky vzduchu, teplem od země řídnou, stoupají do oblasti nižšího tlaku, kde se bublina rozpíná, tím se ochladí a vzroste v ní vlhkost, nastane fázový přechod na vodní kapky a vzniká mrak

## Modelování oblaků 3D buněčným automatem

- V každé buňce 3 logické proměnné: vlhkost hum, oblak cld, fázový přechod (aktivace) act
- Pravidla: tvorba, zánik, přesun větrem
- Výsledek: stav buňky – mrak je nebo není (cld)
- Zobrazení: vyhlazení výpočtem hustoty, pak voxelová vizualizace



## Modelování oblaků 3D buněčným automatem

- Inicializace:  $cld=0$ ,  $hum$  a  $act$  náhodné hodnoty 0 nebo 1
- $hum=1$  – dost páry na tvorbu mraků
- $act=1$  má dojít k fázovému přechodu z páry na vodu
- $cld=1$  je mrak

## Modelování oblaků 3D buněčným automatem

- Základní pravidla:

$$hum(i, j, k, t_{i+1}) = hum(i, j, k, t_i) \wedge \neg act(i, j, k, t_i)$$

$$cld(i, j, k, t_{i+1}) = cld(i, j, k, t_i) \vee act(i, j, k, t_i)$$

$$act(i, j, k, t_{i+1}) = \neg act(i, j, k, t_i) \wedge hum(i, j, k, t_i) \wedge f_{act}(i, j, k)$$

$$\begin{aligned} f_{act}(i, j, k) = & act(i+1, j, k, t_i) \vee act(i, j+1, k, t_i) \vee act(i, j, k+1, t_i) \vee act(i-1, j, k, t_i) \\ & \vee act(i, j-1, k, t_i) \vee act(i, j, k-1, t_i) \vee act(i-2, j, k, t_i) \vee act(i, j-2, k, t_i) \\ & \vee act(i, j, k-2, t_i) \vee act(i+2, j, k, t_i) \vee act(i, j+2, k, t_i) \end{aligned}$$

## Modelování oblaků 3D buněčným automatem - zánik mraků

- Dodat pravděp. zániku  $p_{ext}$
- If  $cld=1$   
Generate r ,  $0 \leq r \leq 1$ , randomly;  
**if**  $r < p_{ext}$   $cld := 0$  **endif**  
**endif**
- Aby možnost opětovného vzniku mraku v buňce,  
nutno náhodně měnit také act a hum na 1 (pravděp.  
 $p_{act}, p_{hum}$ )
- Pravidla:

$$cld(i, j, k, t_{i+1}) = cld(i, j, k, t_i) \wedge IS(rnd > p_{ext}(i, j, k, t_i))$$

$$hum(i, j, k, t_{i+1}) = hum(i, j, k, t_i) \vee IS(rnd < p_{hum}(i, j, k, t_i))$$

$$act(i, j, k, t_{i+1}) = act(i, j, k, t_i) \vee IS(rnd < p_{act}(i, j, k, t_i))$$

kde IS vrací T/F podle výsledku log.výrazu

## Modelování oblaků 3D buněčným automatem - vliv větru

- Modeluje se posouvání mraků ve směru větru –  
proměnné se adekvátně v buňkách posouvají, rychlosť větru  $v(z_k)$  lze měnit s výškou buňky, celočís. hodnoty
- Pravidla:

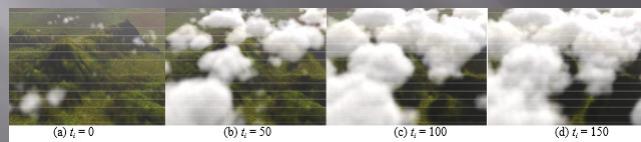
$$hum(i, j, k, t_{i+1}) = \begin{cases} hum(i - v(z_k), j, k, t_i) & pro \ i - v(z_k) > 0 \\ 0 & jinak \end{cases}$$

$$cld(i, j, k, t_{i+1}) = \begin{cases} cld(i - v(z_k), j, k, t_i) & pro \ i - v(z_k) > 0 \\ 0 & jinak \end{cases}$$

$$act(i, j, k, t_{i+1}) = \begin{cases} act(i - v(z_k), j, k, t_i) & pro \ i - v(z_k) > 0 \\ 0 & jinak \end{cases}$$

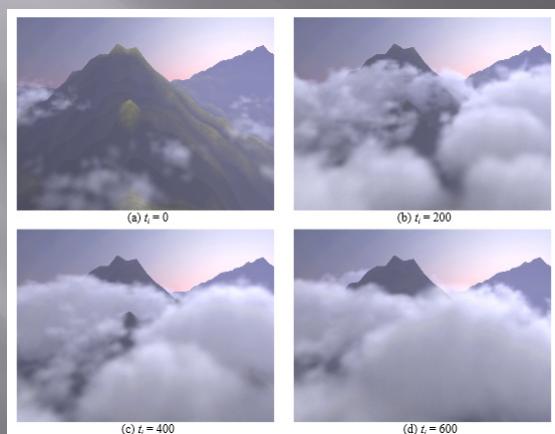
## Modelování oblaků 3D buněčným automatem - vliv větru

- Kontrola pohybu pro animaci elipsoidem -  $p_{act}$ ,  $p_{hum}$   
větší blízko jeho středu než na okrajích,  $p_{ext}$  naopak
- Posouvá se celý elipsoid
- Poloha a rozměr elipsoidů – náhodně nebo zadává animátor



Ukázka výsledků z [Dob00]  
256x128x20 buněk, náhodná generace elipsoidů,  
 $p_{ext} = 0.1$ ,  $p_{act} = 0.001$ ,  $p_{hum} = 0.1$

## Modelování oblaků 3D buněčným automatem



Ukázka výsledků simulace z [Dob00]  
256x128x20 buněk, elipsoidy umístěny ručně kolem hor,  
 $p_{ext} = 0.1$ ,  $p_{act} = 0.001$ ,  $p_{hum} = 0.1$ , uvnitř hory nulové

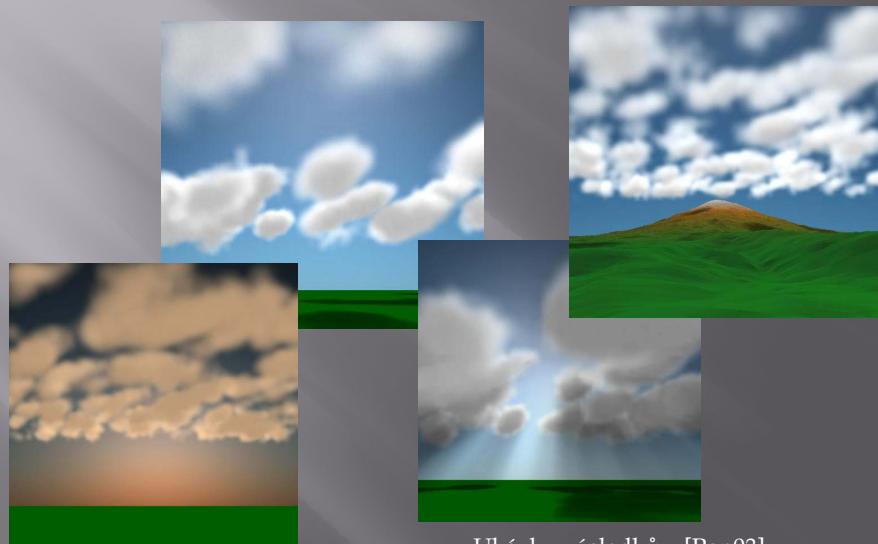
## Modelování oblaků 3D buněčným automatem

- Možnost kombinace se světelnými paprsky



Ukázka výsledků simulace z [Dob00] ,  
256x256x20 buněk, kombinace se slunečními paprsky,  
denní a noční světlo

## Modelování oblaků 3D buněčným automatem



Ukázky výsledků z [Pon03]

## Modelování rostlinných ekosystémů [Deu98]

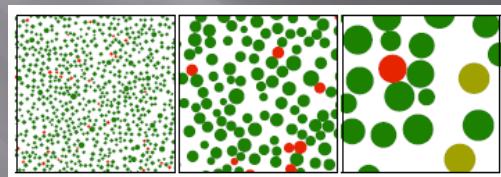
- Napřed vymodelován terén
- Pak specifikována populace rostlin
  - Jak rostliny rozložit po ploše:
    - na základě měření v přírodě
    - nebo simulace interakce mezi rostlinami – často buněčné automaty
    - nebo uživatel zadá interaktivně (např. editací bitmapy)
    - nebo umělá generace na základě nějaké „dobře vypadající“ distribuční funkce

## Modelování rostlinných ekosystémů

- Příklad simulace:
  - Každá z rostlin roste a existuje v tzv. ekologickém okolí – kruh získaný projekcí rostliny na zem
  - Kruhy na počátku simulace umístěny náhodně do rastru, s náhodnými počát. poloměry z daného intervalu
  - S růstem rostliny růst okolí
  - Při kolizi dvou rostlin vítězství silnější, slabší umírá
  - Pokud rostlina dosáhne limitní velikosti, je považována za starou a umírá

## Modelování rostlinných ekosystémů

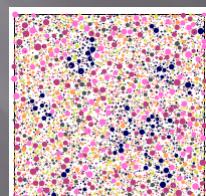
- Po několika iteracích vizuálně věrohodné rozložení rostlin



99., 134. a 164. krok simulace; zelená – běžné rostliny, červená – dominantní, žlutá – staré rostliny [Deu98]

## Modelování rostlinných ekosystémů

- Složitější systém: více druhů rostlin
  - Každý druh popsán parametry – max. velikost, průměrný růst, suchomilnost, průměrný nárůst počtu rostlin v 1 kroku simulace, schopnost přežití oproti jiným druhům apod.
  - Pokud se protínají kruhy různých druhů, silnější dominuje, slabší může zahynout



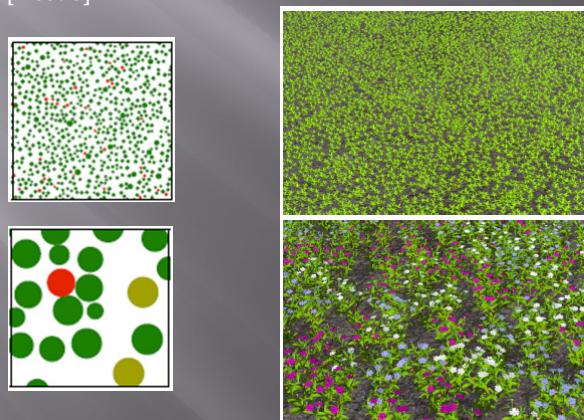
Př.: 8 druhů, modré preferují vlhkko [Deu98]

## Modelování rostlinných ekosystémů

- Rostlina – L-systém nebo částicový systém
- Rostliny generovány procedurálně – úspora paměti pro polygonům
- Pro další úsporu paměti využito **instancování** (více rostlin odvozováno z jedné) a **hierarchie** (skupina rostlin, rostlina, větve, listy, květy...)
- Někdy užit i agentní model: agenti vstupují a vnášejí neklid, např. se pokoušejí odstranit některý druh rostlin v některém místě

## Modelování rostlinných ekosystémů

Př. Výsledek simulace po 99 a 164 iteracích, 7 různých rostlin pro každý typ, ty ještě měněny podle stáří, celkem v obr. 16 tis. rostlin, díky instancování jen 6.7 MB  
[Deu98]



## Modelování rostlinných ekosystémů



Př. Zoom na horskou louku - 8 druhů rostlin,  
100 tis. rostlin ve scéně, jen 151 MB  
(polygonálně by mělo cca 200GB) [Deu98]



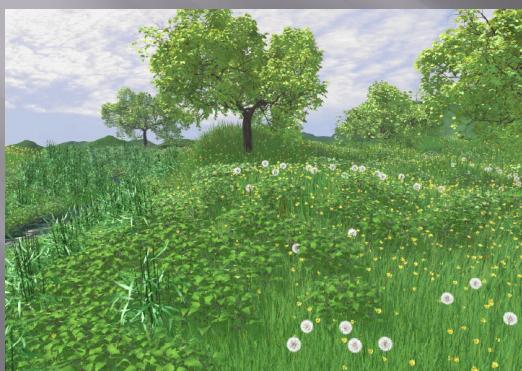
## Modelování rostlinných ekosystémů

Př. Ukázka trávníku z 10 různých instancí  
trsů trávy, koncentrace sedmikrásek řízena  
parametrem [Deu98]



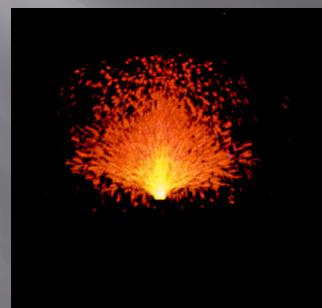
## Modelování rostlinných ekosystémů

Scény se základním rozložením rostlinných společenstev vytvořeným interaktivně [Deu98]



## Modelování ohně, kouře, vody: částicové systémy

- Viz dřívější přednáška



## Literatura

- [Dob00] Dobashi Y., Kanoda K., Yamashita H., Okita T., Nishita T.: A Simple, Efficient Method for Realistic Animation of Clouds, SIGGRAPH 2000, pp.19-28
- [Deu98] Deussen O., Hanrahan P., Lintermann B., Měch R.: Realistic Modeling and Rendering of Plant Ecosystems, s.275-286, SIGGRAPH 1998
- [Lin07] O. Linda: Generation of Planetary Models by Means of Fractal Algorithms, bakalářská práce, vedoucí ing.J.Sloup, ČVUT, 2007
- [Pon03] M. Poneš: Modleování a renderování mraků, bakalářská práce, vedoucí ing.J.Sloup, ČVUT, 2007
- [TerG] Terragen project home page, URL:  
<http://www.planetside.co.uk/terragen/>
- [TerJ] TerraJ project home page, URL: <http://terraj.sourceforge.net/>
- [Zar04] J.Žára, B. Beneš, J. Sochor, P.Felkel: Moderní počítačová grafika, Computer Press, Praha, 2004