Šumy a turbulencie

Čím je zaujímavý?

- Jeden zo základných algoritmov procedurálneho modelovania
- Málo objektov v prírode má pravidelný tvar a vzorku
- Existuje širšia škála šumov – výkonová spektrálna hustota $S(f) \propto 1/f^{\alpha}$ (biely $\alpha = 0$, ružový $\alpha = 1$, červený / Brownov $\alpha = 2$)
- Najjednoduchší prístup generovania = **biely šum**:
  out = random(); // rovnomerné rozdelenie
- **Ružový šum** sa často sa vyskytuje vo fyzikálnych, biologických a ekonomických systémoch

$S(f)$ = rozdelenie priemerného výkonu signálu $x(t)$ vo frekvenčnej oblasti

Čo je šum?

- Definovaný *šumovou funkciou f*
- Vlastnosti šumu:
  - štatisticky invariantný na posunutie (stacionárny)
  - štatisticky invariantný na rotáciu (izotropický)
  - ohraničené frekvenčné spektrum
- Požiadavky na šum:
  - Kompaktnosť (pamäťová náročnosť)
  - Spojitosť (možnosť škálovania)
  - Neperiodičnosť (bez opakovania)
  - Opakovateľnosť (rovnaký výsledok pri násobnom spustení)
  - Ohraničenosť (rozsah, napr. ⟨-1, 1⟩)

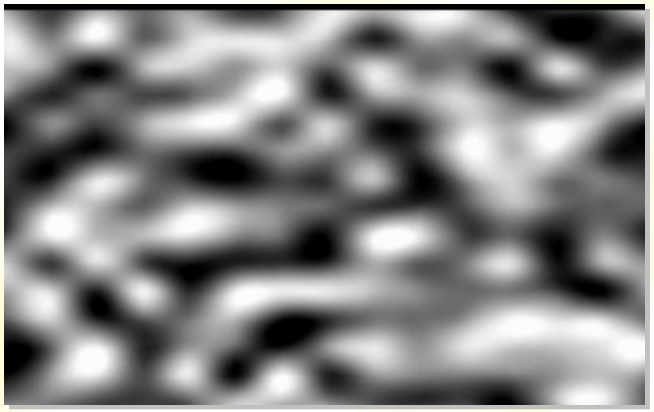
Mriežkový (Lattice) šum

- Využíva celočíselnú mriežku
- Nutnosť prepojiť šum s priestorom

  output = noise(point coord);
- Typy mriežkového šumu:
  - Vygenerovaný v 1D/2D/3D priestorovej mriežke
  - Náhodné hodnoty sú vypočítané v bodoch mriežky
  - Počas renderovania sú body mimo mriežkových bodov interpolované
- Typ interpolácie ovplyvňuje výsledný vzhľad
- Nevýhodou je prejav artefaktov v smere osí

Perlinov šum

- Prvý druh šumu v počítačovej grafike
- Ken Perlin: *An Image Synthesizer*, 1985
- Pôvodne bol definovaný ako “lattice gradient noise“
- Závislý na druhu interpolácie:

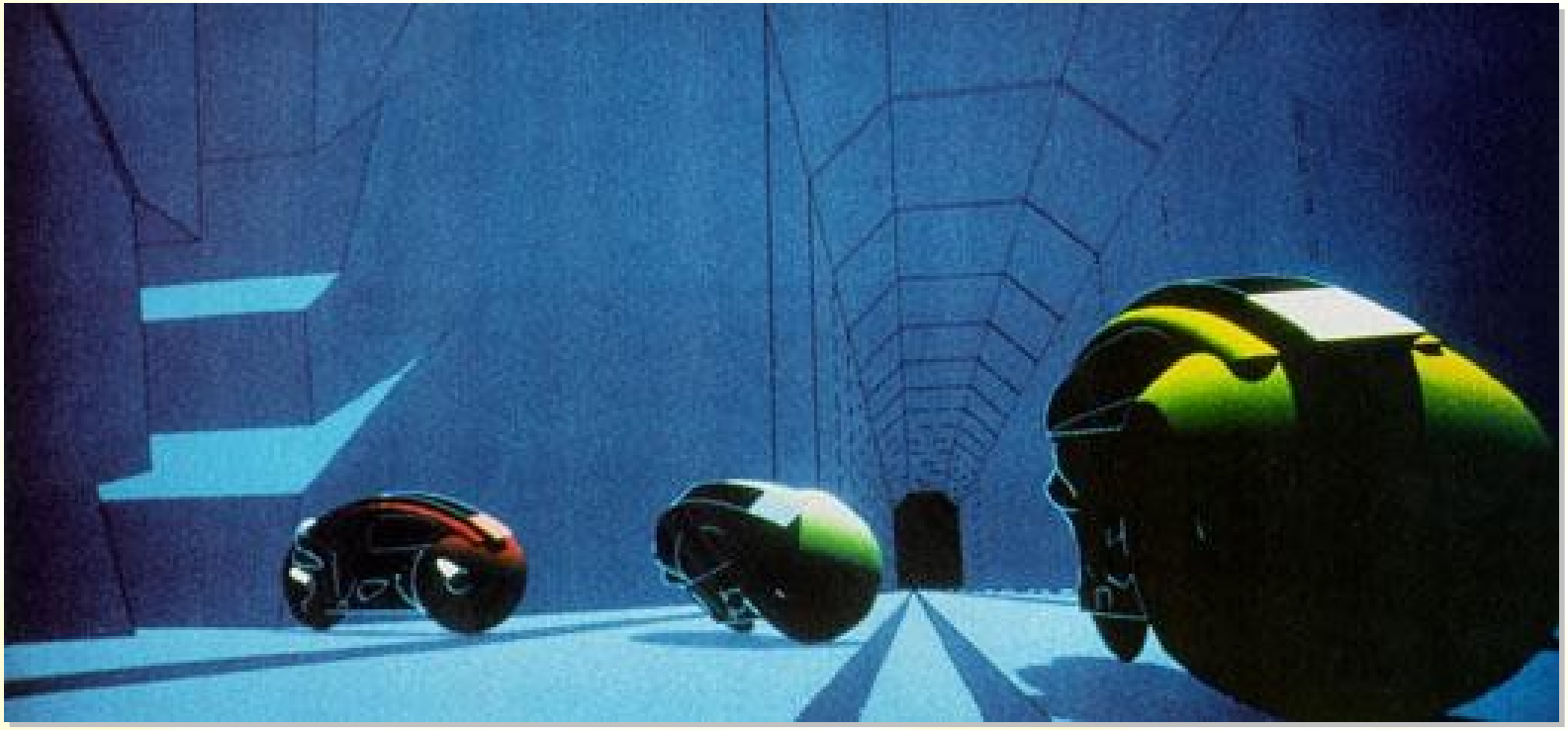


bilineárna

bikubická

História Perlinovho šumu

- Tron 1981



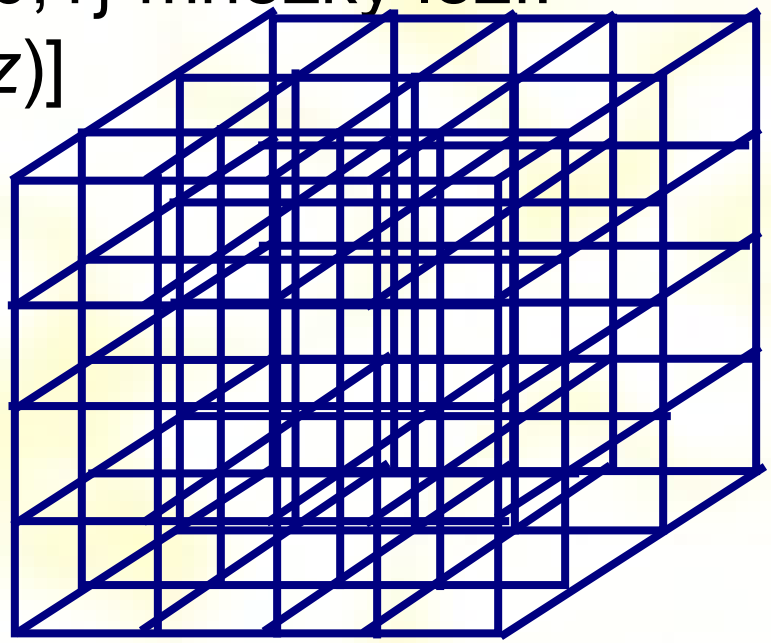
Princíp Perlinovho šumu

- Rozdelenie priestoru do 3D mriežky
- Priradenie „gradientneho“ vektoru s náhodnými hodnotami každému bodu mriežky
  - tento vektor je použitý pri výpočte hodnoty šumu v bode pomocou skalárneho súčinu
- Šumová hodnota v ľubovoľnom bode (x,y,z) je vypočítaná ako vážený priemer 8 hodnôt, prislúchajúcich ôsmym najbližším bodom mriežky

Princíp Perlinovho šumu

- 3D mriežka nemusí mať veľké rozlíšenie (=počet bodov), pretože vďaka permutácií náhodných hodnôt nevzniká opakovanie šumového vzoru
- Pre každý bod (x,y,z) vieme ľahko určiť v ktorej bunke {[i+a,j+b,k+c]; a,b,c=0,1} mriežky leží: [i,j,k]=[floor(x),floor(y),floor(z)]

  kde floor(u) je funkcia, pomocou ktorej určíme najbližší bod mriežky



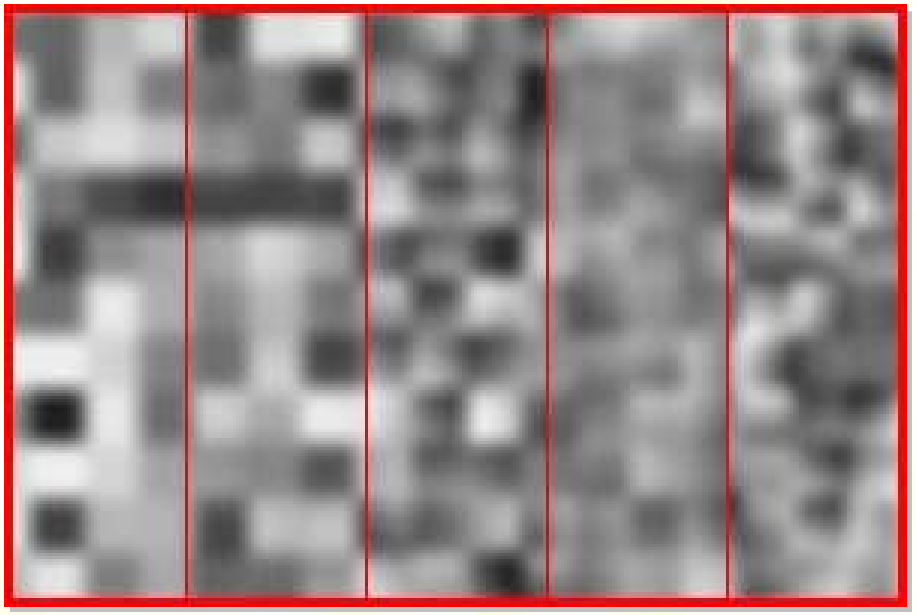
Príklad: mriežkový šum v 3D

- Najskôr vytvoríme pole (tabuľku) G[] 255-tich náhodných vektorov jednotkovej dĺžky a permutačnú tabuľku P[] celých čísel z {1,…,255}.
- Každému [*i*,*j*,*k*] vieme priradiť vektor G[*n*]: n=P[(P[(P[i%256]+j)%256]+k)%256]≡P[P[P[i]+j]+k]
- Pre bod (x,y,z) najskôr vypočítame 8 hodnôt $value_\tau$: (u,v,w)=(x-i,y-j,z-k), [i,j,k]=[floor(x),floor(y),floor(z)] $weight$=drop(u)·drop(v)·drop(w), drop(t)=1-(3t^2+2t^3) $value_\tau$=(G[n]• (u,v,w))·$weight$
- Šumová hodnota v bode (x,y,z) je súčet ôsmych hodnôt $value_\tau$ vypočítaných pre 8 najbližších bodov mriežky

Typy mriežkového šumu

- Šum určený hodnotou (value noise)
  - *n*D-lineárna alebo kubická (napr. Catmull-Rom, Wiener) interpolácia medzi náhodnými hodnotami
- Gradientny šum
  - používa hodnoty gradientov (normované vektory) v uzloch, rovnomerne rozložené na jednotkovej sfére
- Gradientovo-hodnotový šum
  - vážený súčet hodnôt a gradientov
  - Hermitova interpolácia, dotykové vektory splajnu sú tvorené gradientami
- Mriežkovo konvolučný šum
  - využíva interpoláciu pomocou konvolúcie s radiálne symetrickým filtrom

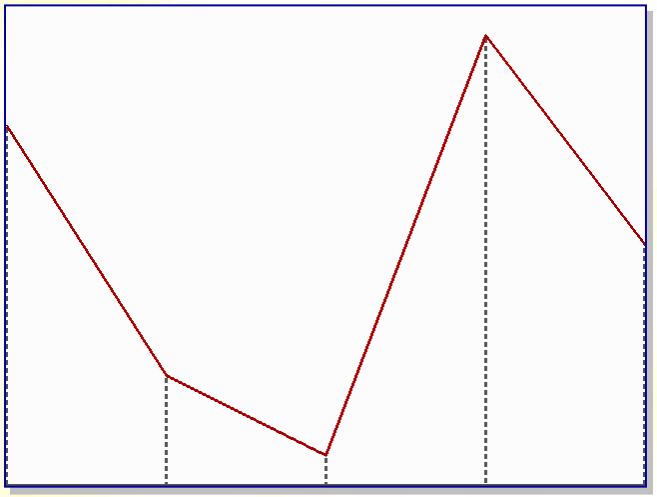
Porovnanie gradientovo-hodnotového šumu



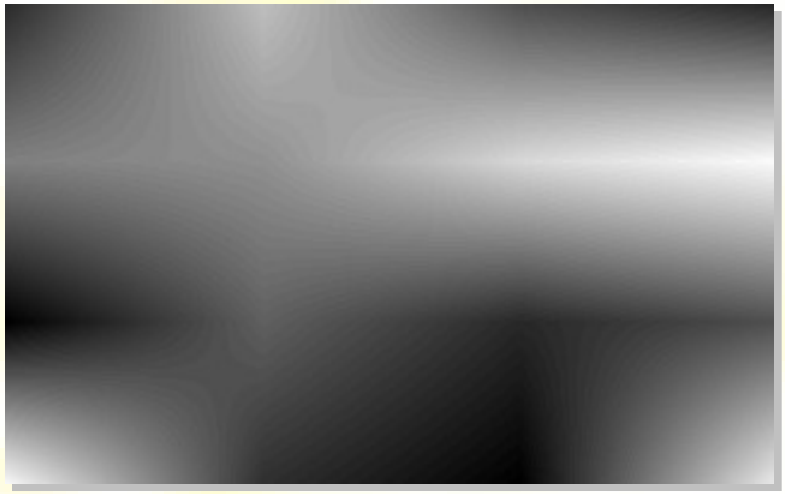
Hodnota / +Grad 0.25 / +Grad 0.5 / +Grad 0.75 / Gradient

Interpolácie

- Najjednoduchší typ = *n*D-lineárna interpolácia



1D lineárna



2D bilineárna

Interpolácia zmiešavacími (blending) funkciami

- Lineárna interpolácia $f(0)$, $f(1)$:

  $f(x) = (1-x)\, f(0) + x\, f(1)$

  používa zmiešavacie funkcie $(1-x)$, x

- Vo všeobecnosti:

  $f(x) = u(x)\, f(0) + v(x)\, f(1)$

- Požiadavky na zmiešavacie funkcie ($x \in \langle 0,1 \rangle$)

  a) $u(x) + v(x) = 1$

  b) $u(0) = 1$, $v(0) = 0$

  c) $u(1) = 0$, $v(1) = 1$

Hladká interpolácia kubikami

- Je treba C^2 spojitosť?
- Ďalšie požiadavky na zmiešavacie funkcie
- Hodnoty derivácií v koncoch:

  d) $u'(0) = 0 \quad v'(0) = 0$

  e) $u'(1) = 0 \quad v'(1) = 0$

- Jedna z možností je použiť Hermitove funkcie H_{03}, H_{23}:

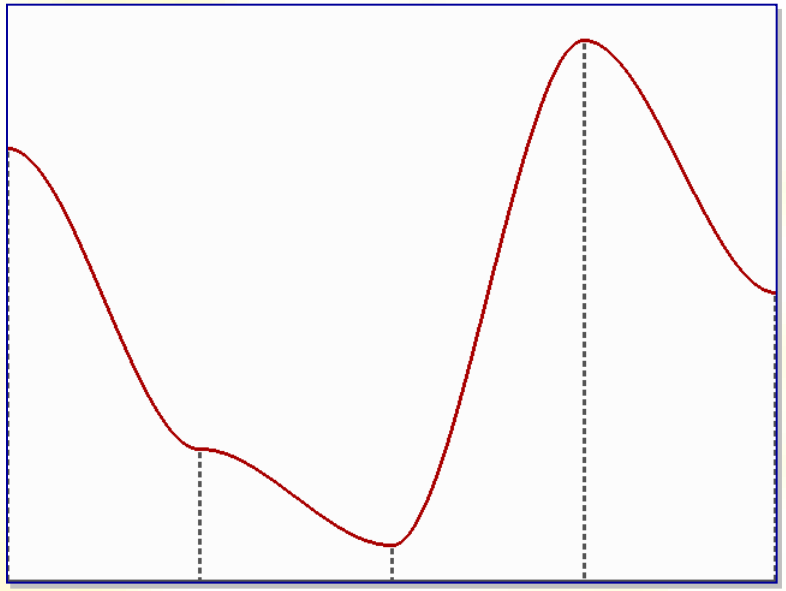
  $u(x) = 2x^3 + 3x^2 + 1$

  $v(x) = -2x^3 + 3x^2$

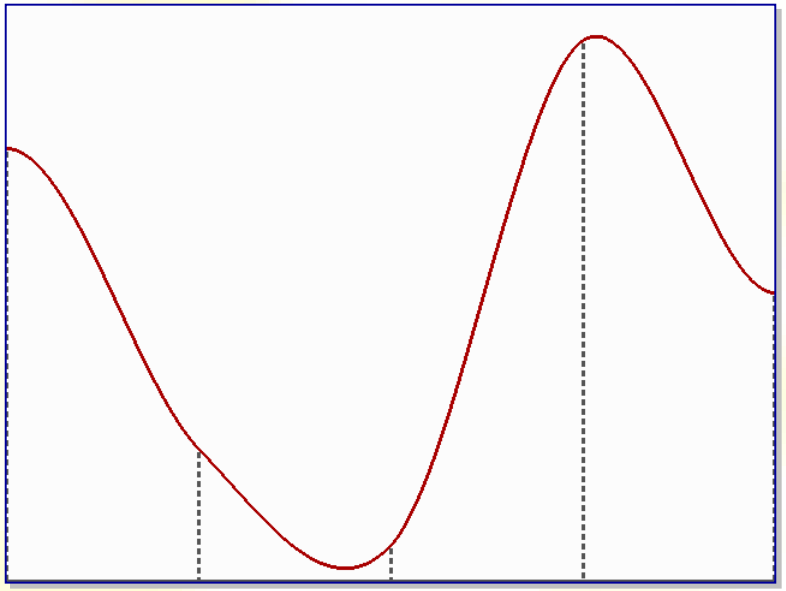
  $f(x) = 2(f(0) - f(1))x^3 + 3(f(1) - f(0))^2 + f(0)$

1D kubická interpolácia

- Nulové derivácie v daných bodoch
- Vypočítané derivácie



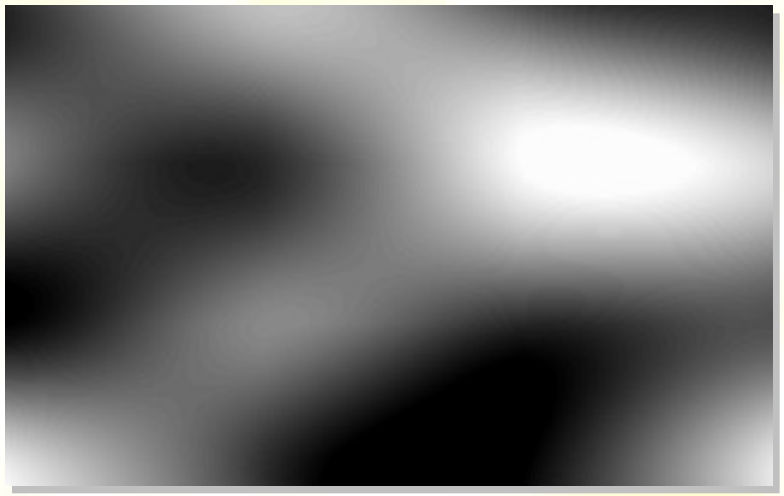
nulové derivácie



vypočítané derivácie

Zložená interpolácia

- Interpolácia v 1D: $f(x) = (1-x)\, f(0) + x\, f(1)$
  - Nahradenie $f(0) = u_0(x)$, $f(1) = u_1(x)$
  - $f(x) = (1-x)\, u_0(x) + x\, u_1(x)$
- „Hladšia“ kubická interpolácia

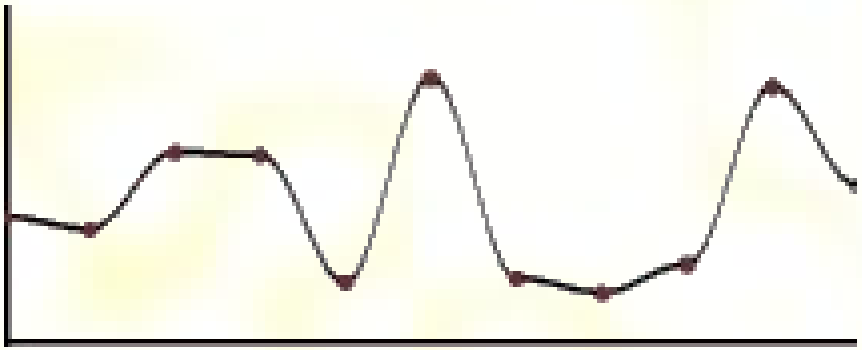


Kosínusová interpolácia

- Zmiešavacie funkcie v tvare:

  $u(x) = (\cos(x\pi) + 1) / 2$

  $v(x) = (1 - \cos(x\pi)) / 2$



- Nevýhoda pomalšieho výpočtu

Konvolučný šum

1. Vygenerujú sa náhodné hodnoty (biely šum) v bodoch mriežky
2. Aplikuje sa konvolučný filter pre každý bod

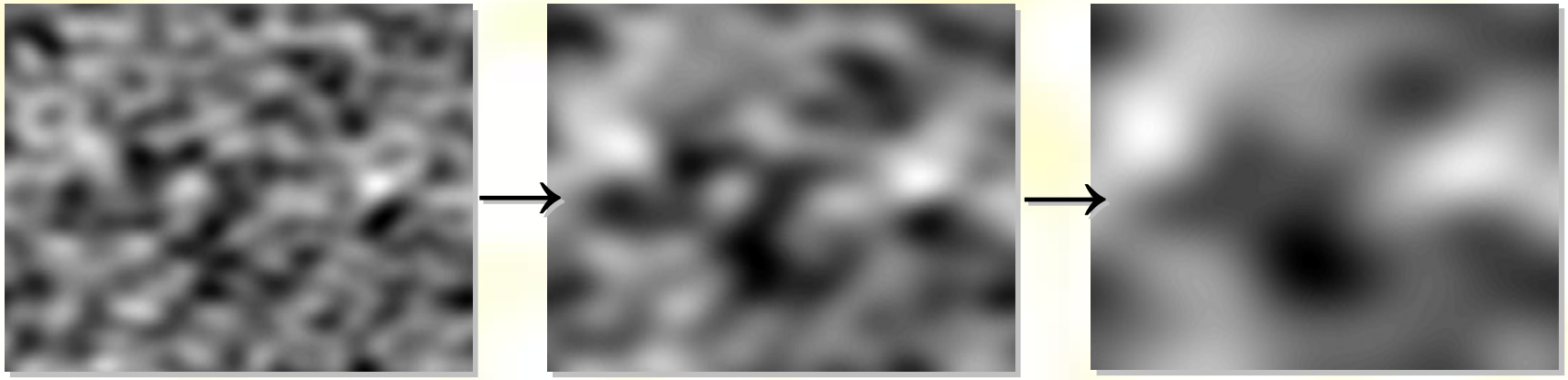
- Gaussova konvolúcia

$$K = \frac{1}{9}\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

- Výsledné hodnoty sú vyčíslené ako vážený priemer ⇒ *spriemerovaný šum*

Spriemerovaný šum

- Ovplyvnenie výsledku počtom iterácií



- Modifikácie: anisotropické filtre

$$K=\frac{1}{55}\begin{pmatrix} 0 & 0 & 5 & 0 & 0 \\ 0 & 1 & 10 & 1 & 0 \\ 1 & 2 & 10 & 2 & 1 \\ 0 & 1 & 10 & 1 & 0 \\ 0 & 0 & 5 & 0 & 0 \end{pmatrix}$$

Iné typy šumov

- Riedky konvolučný šum (sparse convolution noise)
- Bodový šum (spot noise) (van Wijk, 1991)
- fBm (fraktálny Brownov pohyb), posun stredného bodu
- Integrálny šum
- (Fourierova) spektrálna syntéza
- Voronoiov šum
- Simplexný šum
- Turbulencie

Bodový šum (spot noise)

1D funkcia bodového šumu:

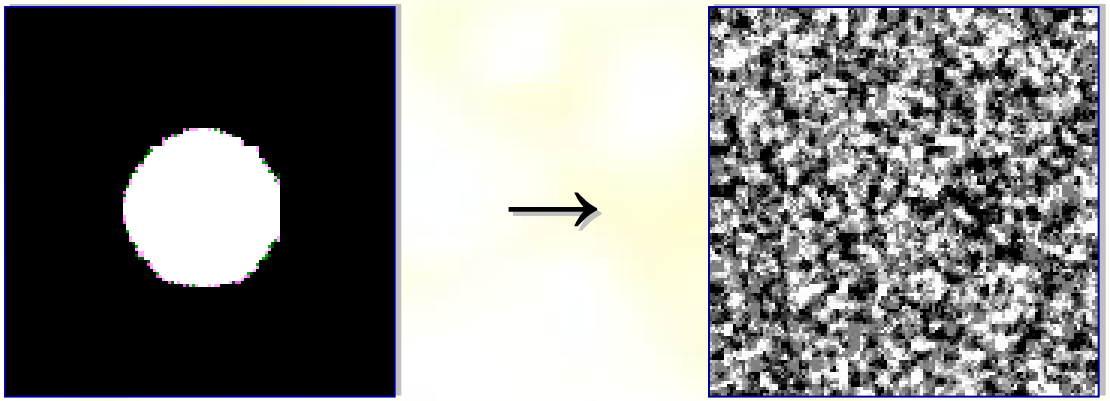
$f(x) = \sum a_i h(x-x_i)$

a_i = náhodná skalárna veličina

x_i = náhodná poloha

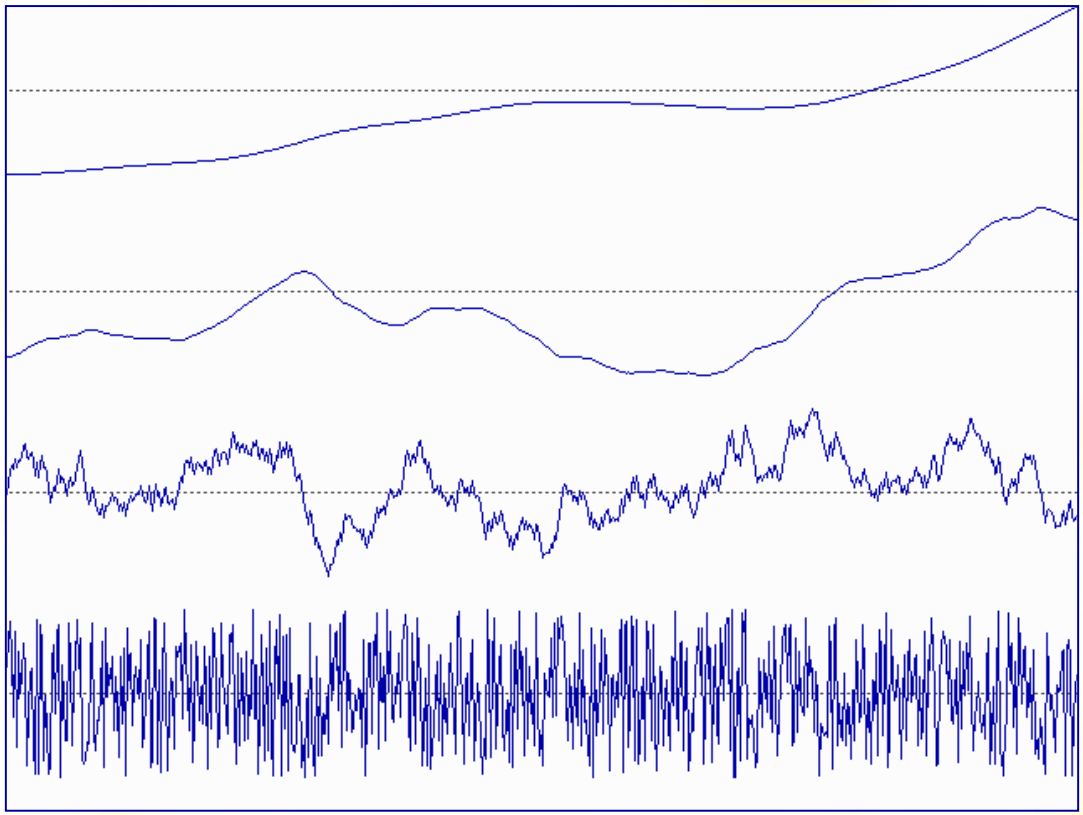
$h()$ je nenulová iba v blízkom okolí $x = 0$

- Mnohonásobné „položenie“ bodovej funkcie $h(x)$:



Integrálny šum

- *n*-té spojenie (integrácia) bieleho šumu
  - noise(*x*) = *f*(noise(*x*-1))



Algoritmus integrálneho šumu

- Náhodné vygenerovanie $f(0), f^{1}(0), \ldots, f^{n-1}(0)$
- Pre hodnotu x:

  0. $f^{n}(x) = \text{random}(x)$

  …

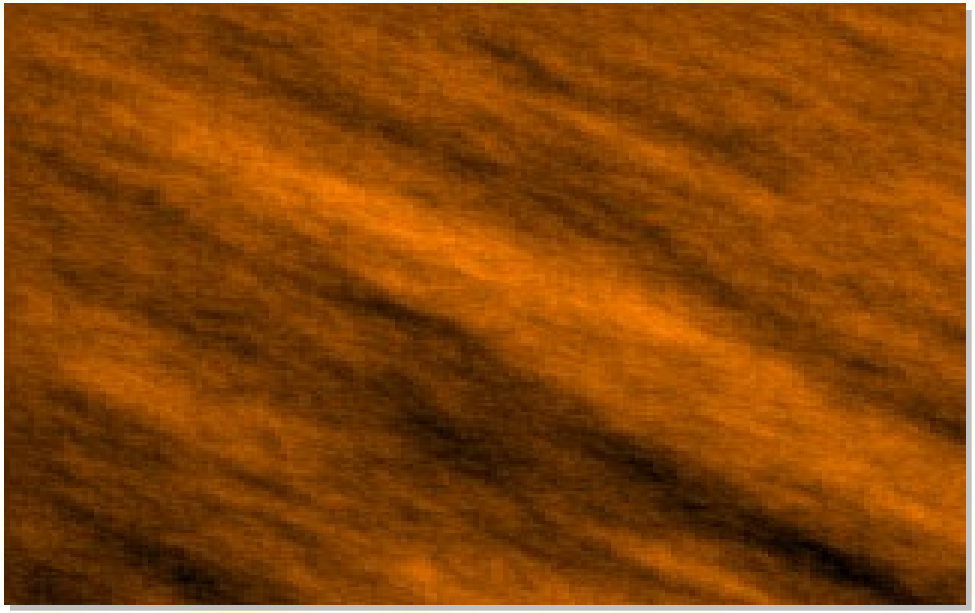
  i. $f^{i}(x) = f^{i}(x\text{-}1) + f^{i+1}(x)$

  …

  n. $f(x) = f^{0}(x)$

  n+1. výsledok $f(x), f^{1}(x), \ldots f^{n-1}(x)$

Príklad 2D Integrálneho šumu



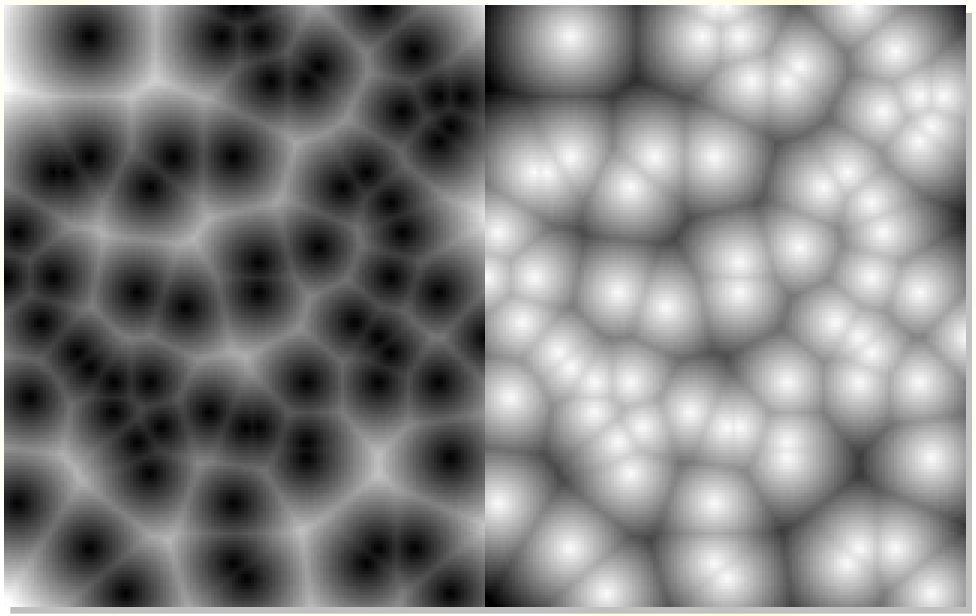
- Vertikálne, horizontálne hrany = 1D šum
- Vnútorné body sú cikcakovito posunuté
  point[x,y] = zloženie point[x-1,y] + point[x,y-1]

Fourierova spektrálna syntéza

- Generuje pseudonáhodné diskrétne frekvenčné spektrum, v ktorom výkon na danej frekvencii má vhodné rozdelenie pravdepodobnosti
- Následne sa vykoná diskrétna inverzná Fourierova transformácia na frekvenčné spektrum, ktorá prevedie šum do priestorovej reprezentácie
- Dá sa využiť napr. na vytvorenie textúry oblakov

Voronoiov šum

1. Náhodne rozložené body
2. Vizualizuje sa vzdialenosť k najbližšiemu susedovi

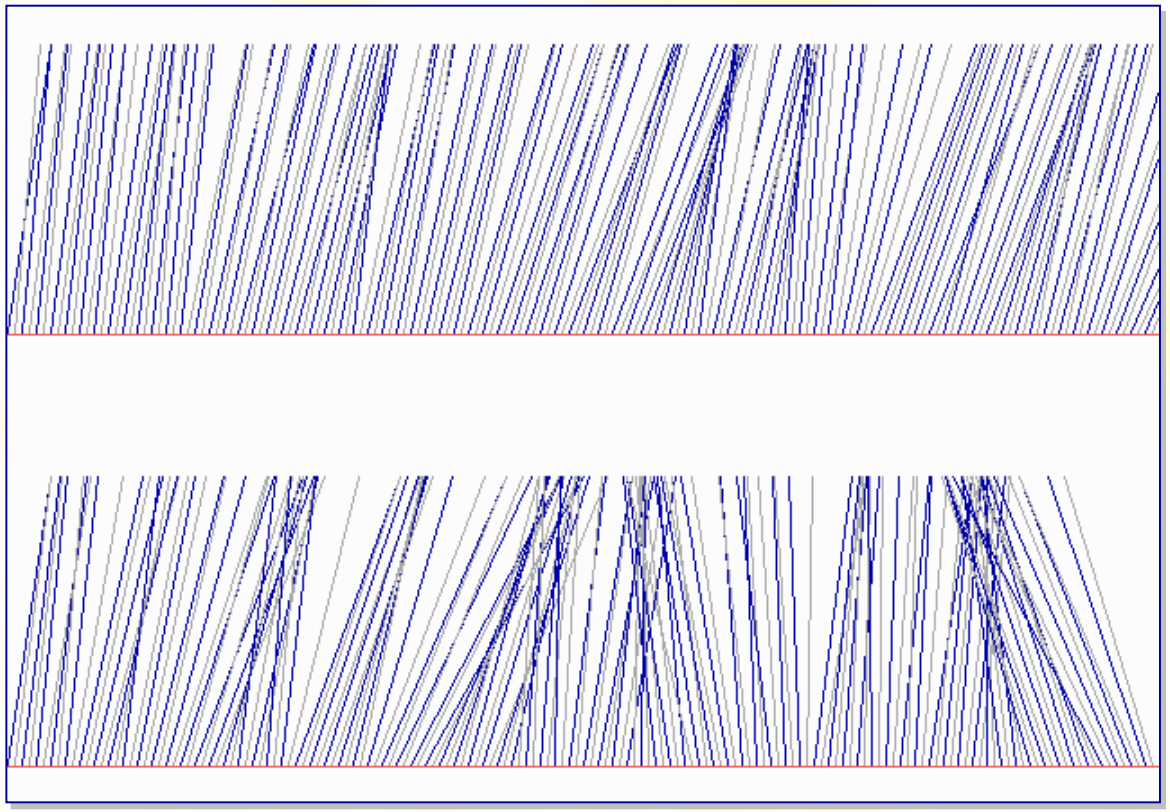


Turbulencie

- Funkcie modulujúce (ovplyvňujúce) pixle obrazu
- Output = $f(x + \text{turb}(x))$
- Častá podmienka funkcie = spojitosť
- Zdroj modulácie:
  - Jednoduchý obraz
  - Explicitná funkcia

Príklad 1D turbulencie

- Pôvodná funkcia = identita



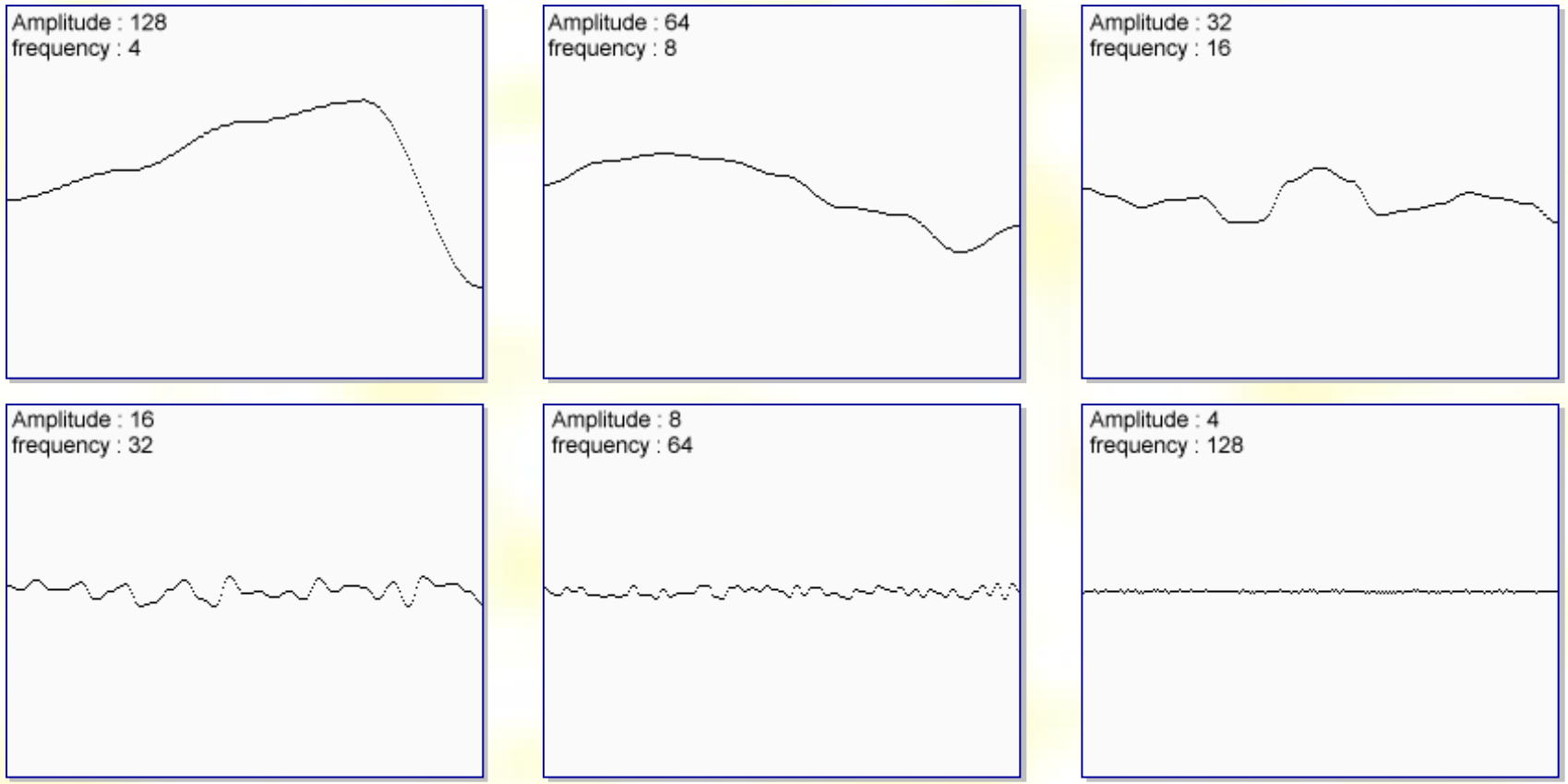
- Dolná turbulencia je 3x väčšia ako horná

Perlinova turbulencia

$$\mathrm{turb}(x) = \sum_{i=0}^{k} \left| \frac{\mathrm{noise}(2^i x)}{2^i} \right|$$

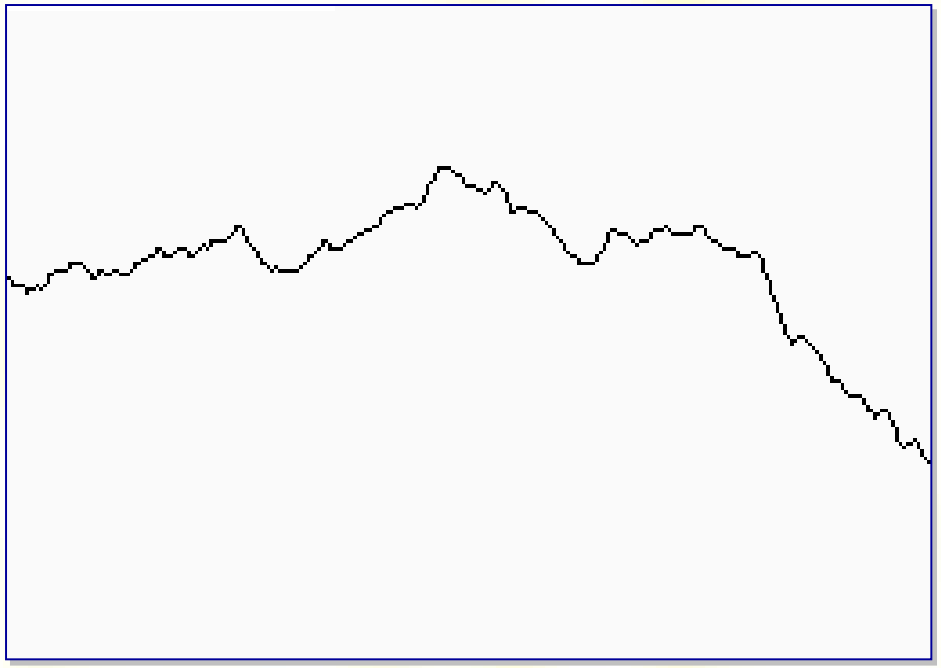
- k = najmenšie číslo vyhovujúce podmienke
  $1/2^{k+1}$ < veľkosť pixla
- Súčet šumových funkcií
- Majú fraktálny charakter:
  - násobná frekvencia
  - polovičná amplitúda
- Šumová funkcia = Perlinov šum

Zloženie Perlinovej turbulencie



- i = číslo oktávy (frekvencia = 2^i)

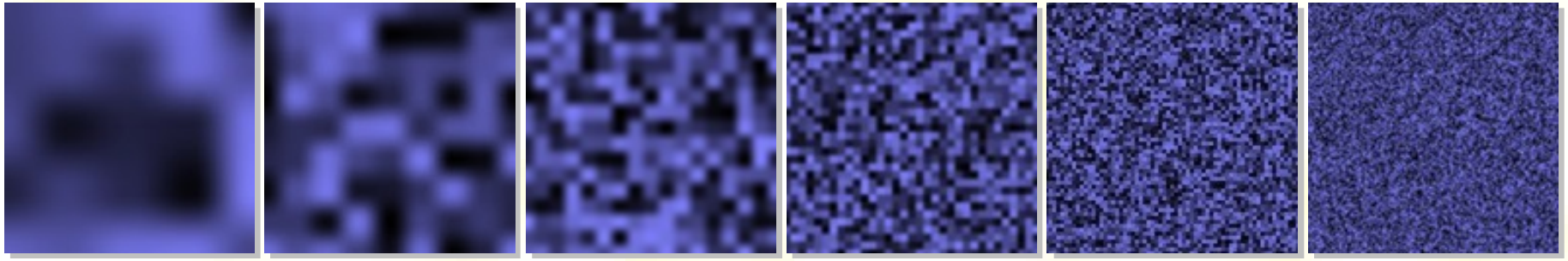
Výsledok 1D Perlinovej turbulencie



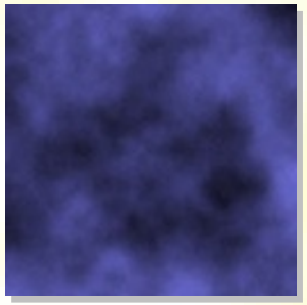
- Frekvencia vykazuje samopodobnosť

2D Perlinova turbulencia

- Šumové funkcie:



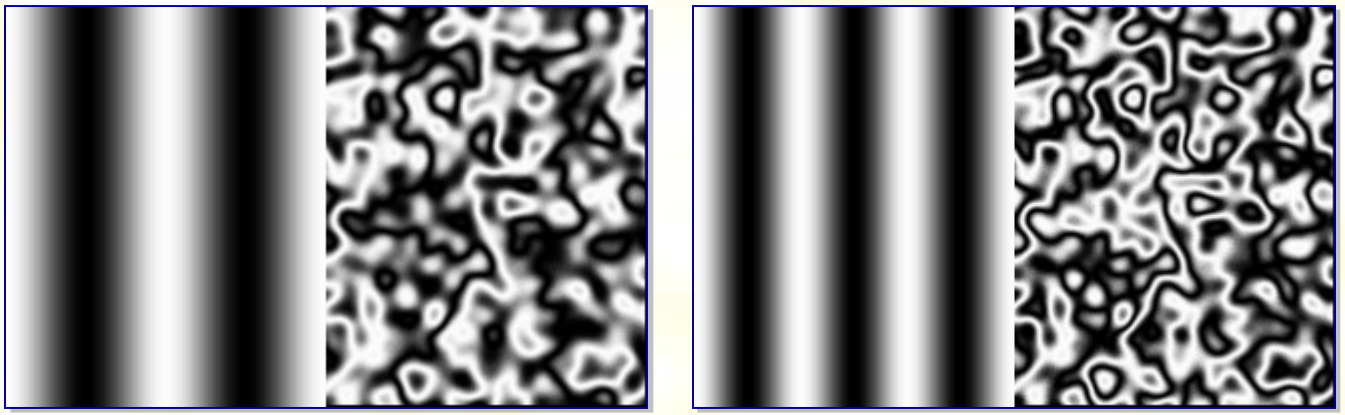
- Zloženie:



Príklad Perlinovej turbulencie

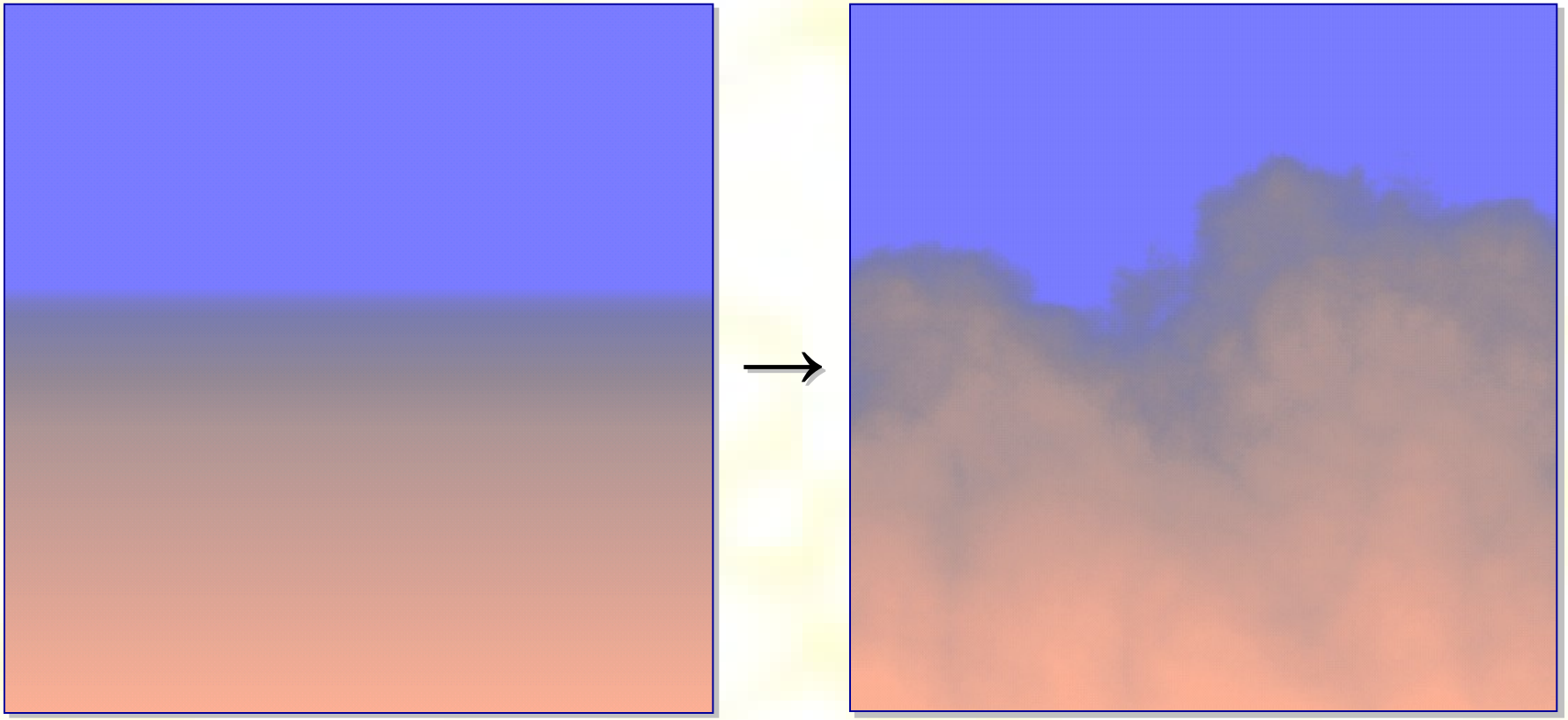
- Fraktálny terén
- Oblaky

Turbulenciou modulované obrázky



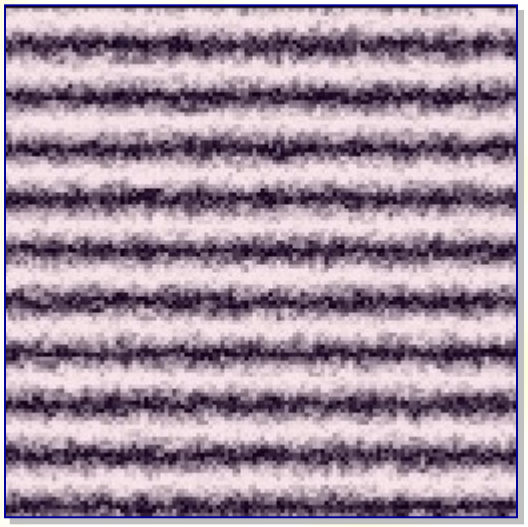
- Zdrojový obrázok = gradientny prechod
- Použitá turbulencia: 2D priemerný šum

Perlinova turbulencia - oblaky

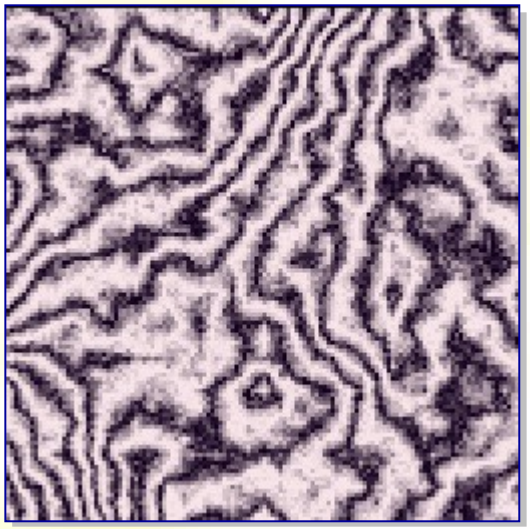


Mramor

- Zdroj $z = \sin(y)$
- Perlinova turbulencia



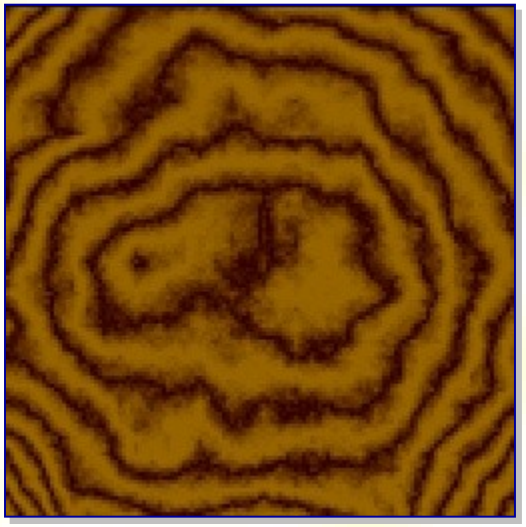
menšia turbulencia

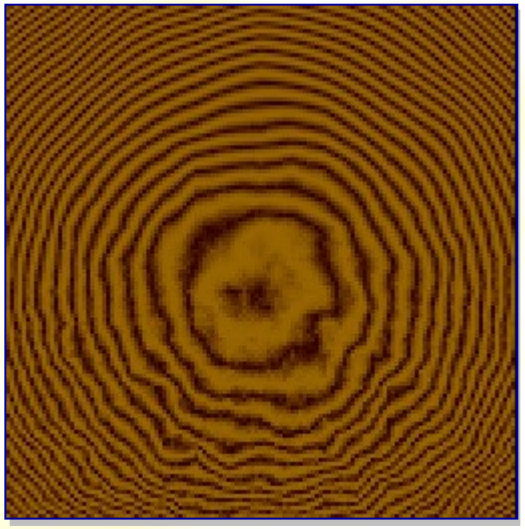


väčšia turbulencia

Drevo

- Zdroj rotačný sínus - $[u \cdot \cos(v), u \cdot \sin(v), \sin(u)]$
- Brownov pohyb





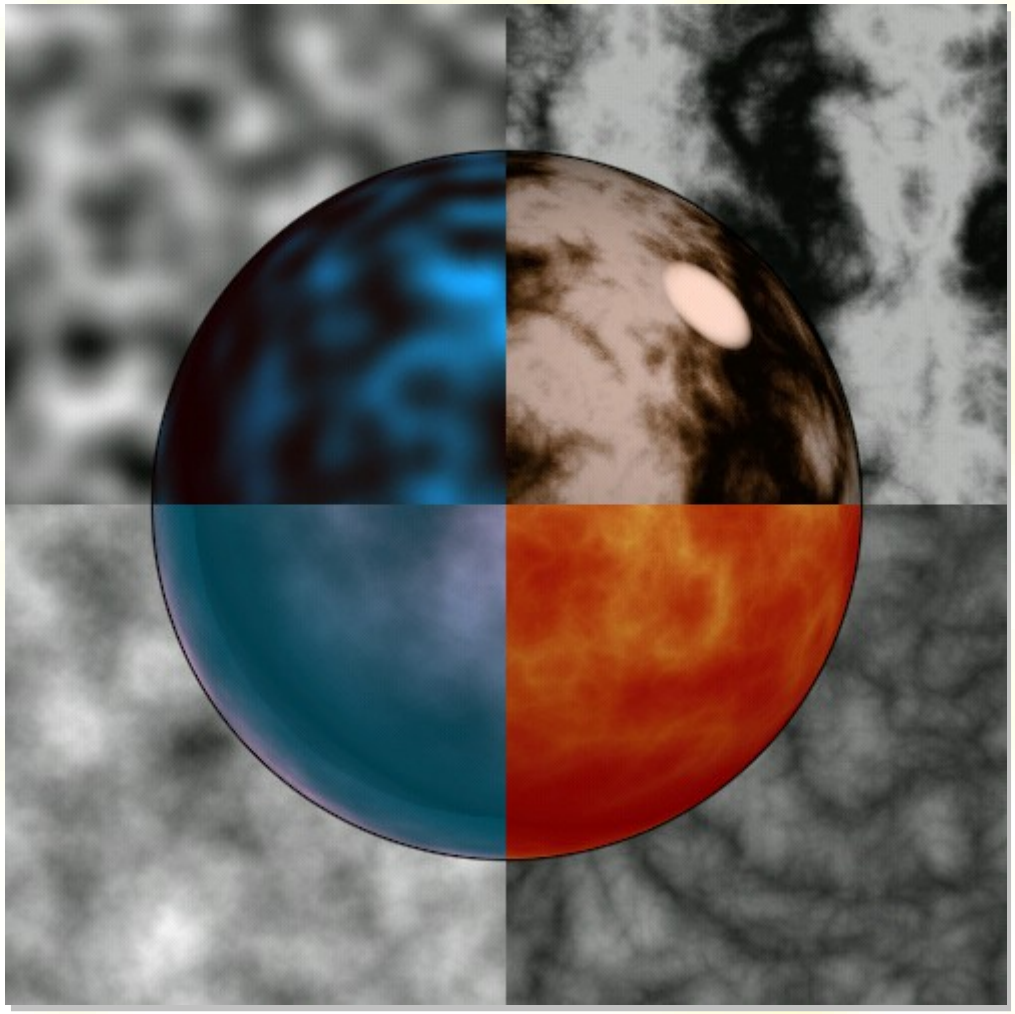
Použitie modulujúcej funkcie

- Output = Ramp($f(\boldsymbol{x})$), kde $\boldsymbol{x}$ je tvaru $\boldsymbol{\alpha}\cdot\boldsymbol{x} + \boldsymbol{\beta}\cdot\boldsymbol{t}(\boldsymbol{\gamma}\cdot\boldsymbol{x})$ a $\boldsymbol{t}(\boldsymbol{x})$ je turbulencia

Príklady modulujúcej funkcie pre $\boldsymbol{\beta} = \boldsymbol{\gamma} = (1, 1, 1)$:

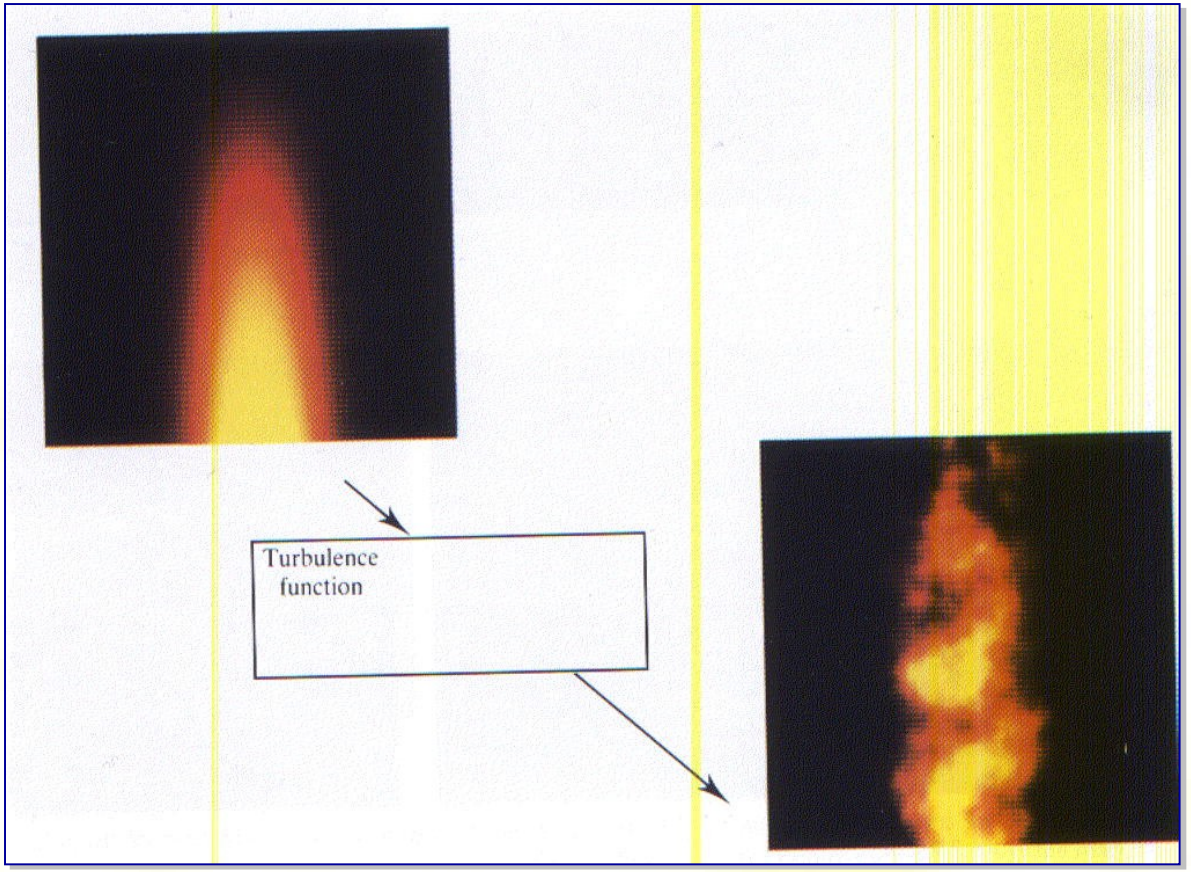
- Drevo $f(\boldsymbol{x})$ = sqrt(x^2+z^2), $\boldsymbol{\alpha} = (1, 1, 0)$
- Drevo $f(\boldsymbol{x})$ = sin(x), $\boldsymbol{\alpha} = (1, 0, 0)$
- Mramor $f(\boldsymbol{x}) = x$, $\boldsymbol{\alpha} = (1, 0, 0)$
- Mramor $f(\boldsymbol{x})$ = abs(sin(x)), $\boldsymbol{\alpha} = (0, 0, 0)$

Príklady ďalších obrázkov turbulencií



Oheň

- 3D turbulencia (2D + čas) vytvárajúca oheň



Animácia ohňa