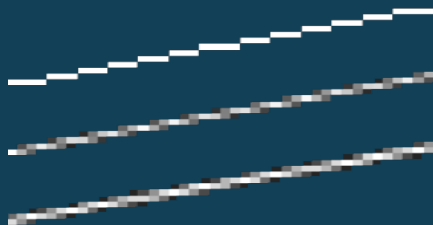


Dóra Varnyú and László Szirmay-Kalos
BME Irányítástechnika és Informatika Tanszék

Thick Line Integration with Filtered Sampling



Az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósult meg.
(EFOP-3.6.2-16-2017-00013)

SZÉCHENYI 2020



Európai Unió
Európai Szociális
Alap

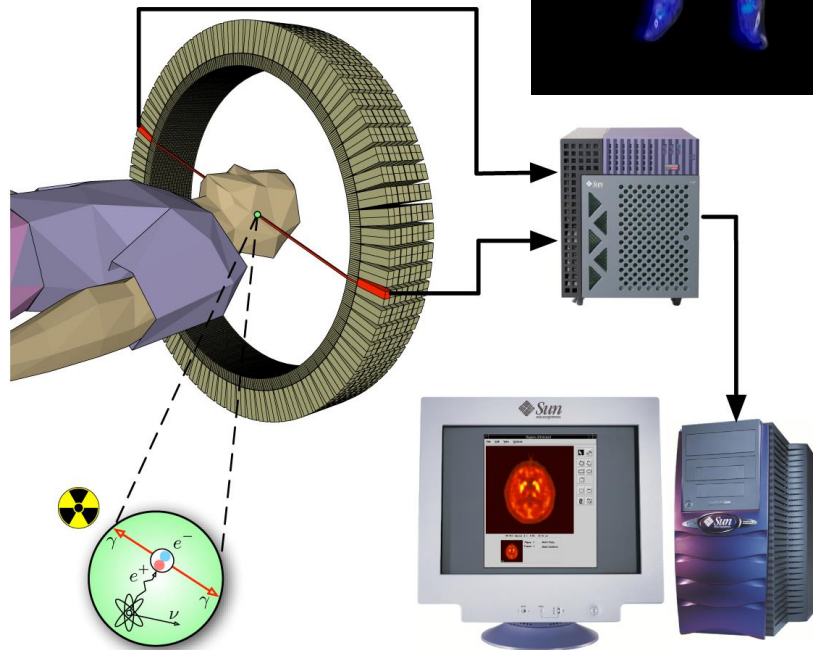
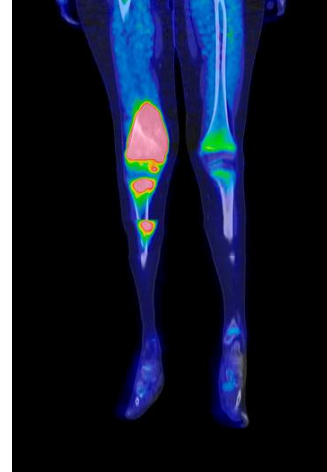


BEFEKTETÉS A JÖVŐBE



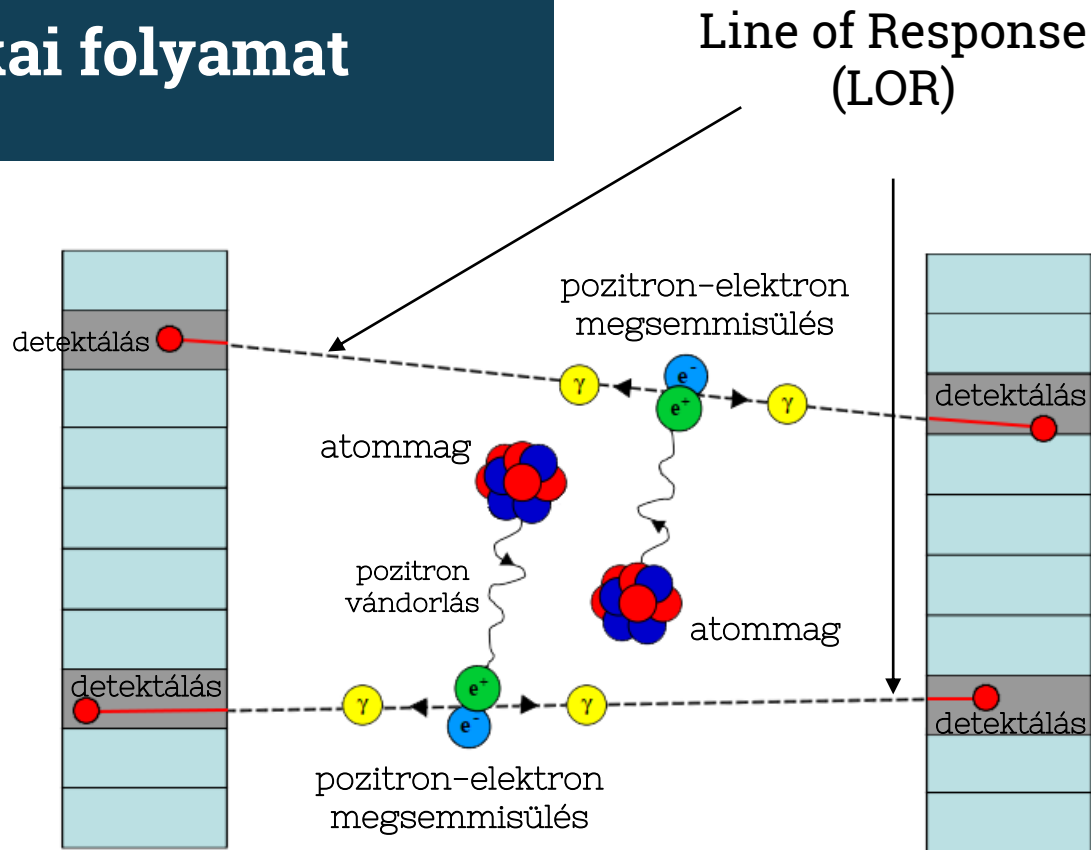
Pozitronemissziós Tomográfia (PET)

- Tumor
- Koszorúér-betegség
- Alzheimer-kór
- Epilepszia
- Parkinson-kór



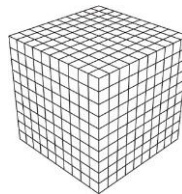


Fizikai folyamat





PET rekonstrukció

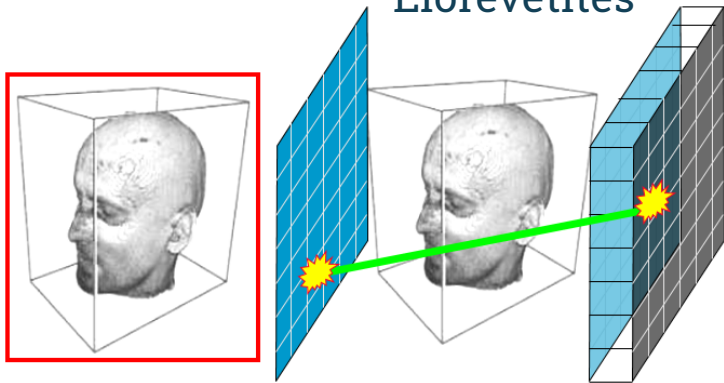


homogén 3D voxelek

Maximum Likelihood alapú iterációs séma:

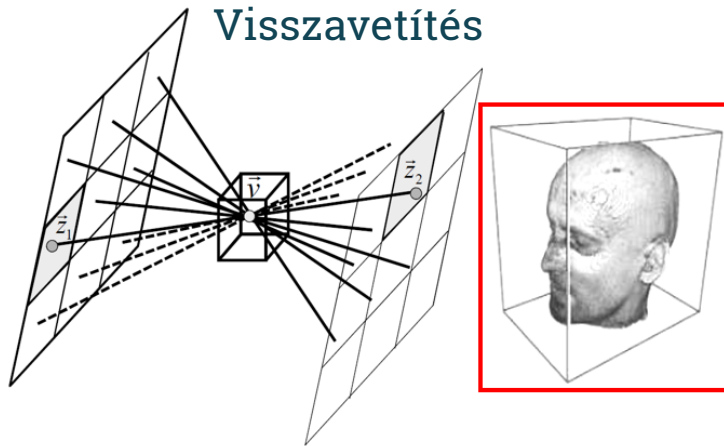
Eloszlás becslés → Várt találatszám

Előrevetítés



Mért találatszám → Korrigált eloszlás becslés

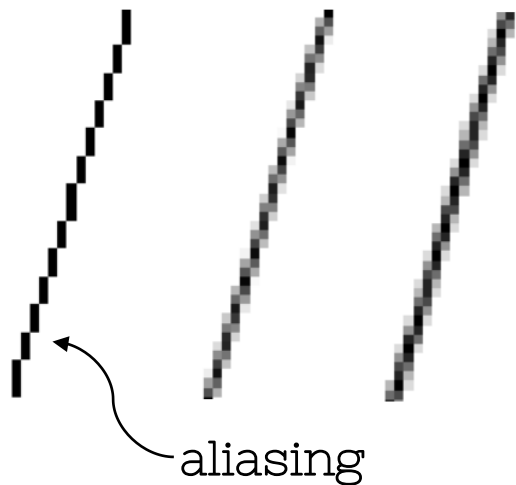
Visszavetítés



10-100 iteráció

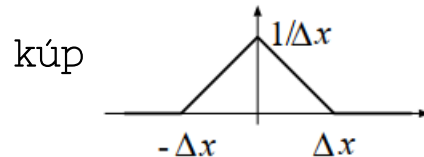
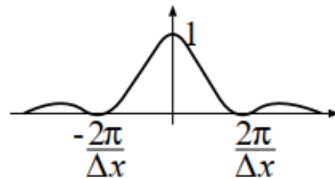
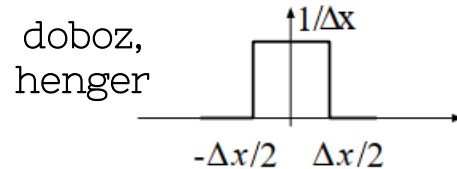
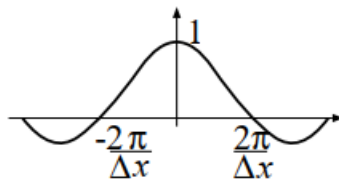
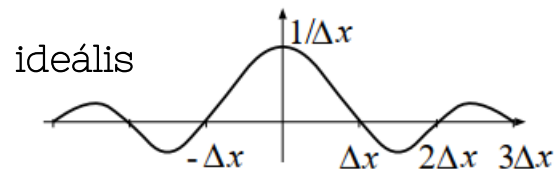
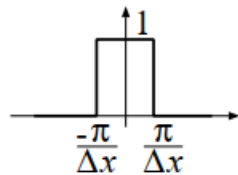
Antialiasing

Szűrjük ki a magas
frekvenciájú komponenseket!



Frekvenciatartomány

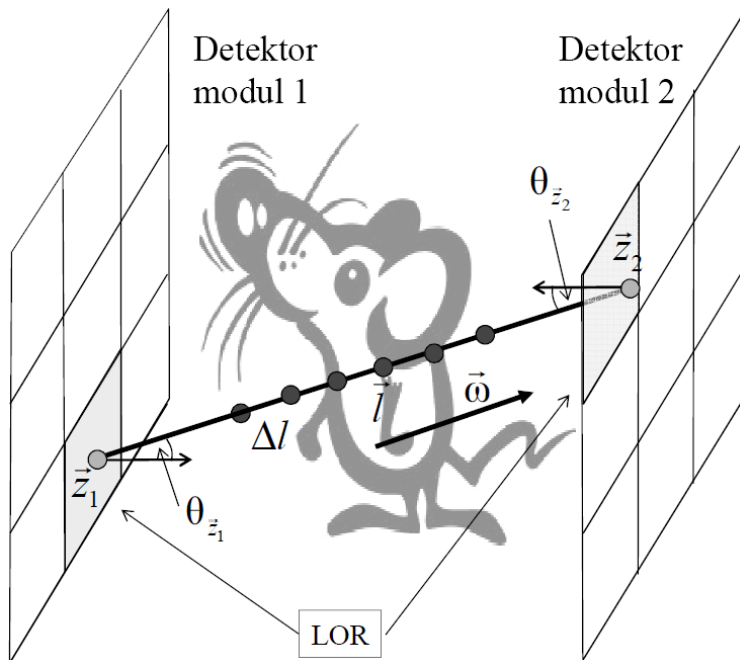
Időtartomány



3D vonalhúzó algoritmusok az előrevetítésben

Előrevetítés: feltételezett nyomjelző eloszlás → várt találatszám a LOR-ban
LOR = Line Of Response: két szemközti detektorkristályt összekötő vonal

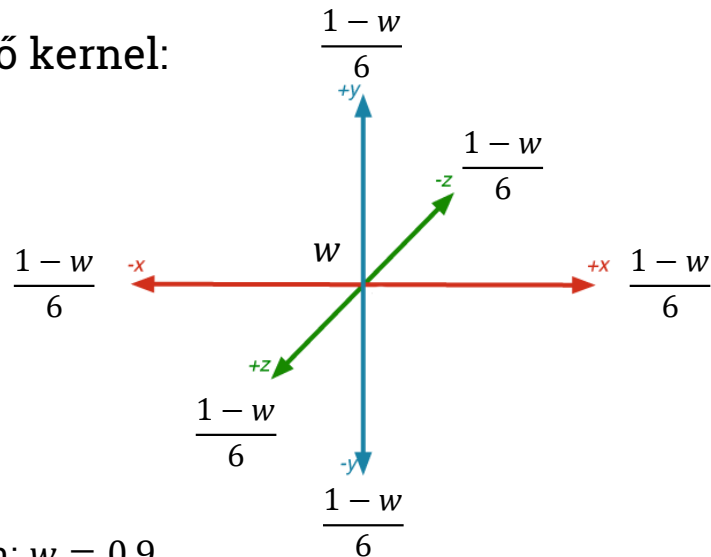
Ray marching



Szűrt ray marching

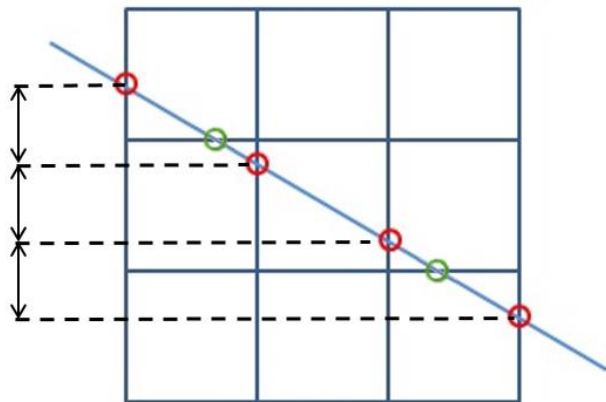
előszűrés + ray marching

Szűrő kernel:



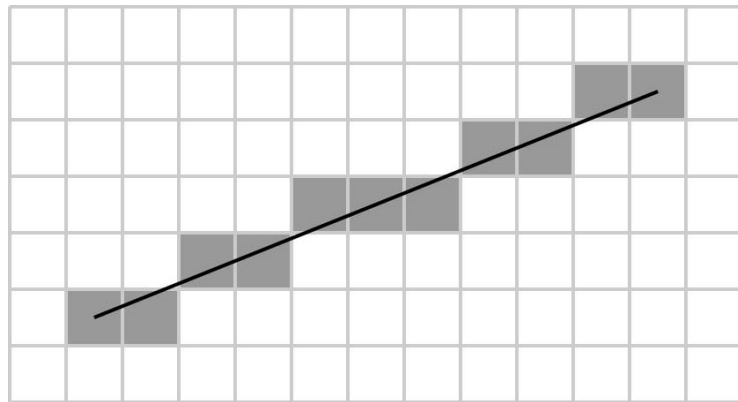
Nálam: $w = 0.9$

Siddon



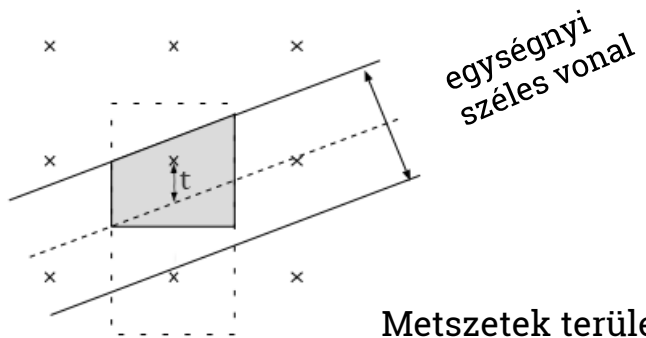
Következő lépés:
 t_x , t_y és t_z sugárparaméterek közül
melyik a legkisebb?

Bresenham



Következő lépés:
 $(x + 1, y)$ vagy $(x + 1, y + 1)$?
 $(x + 1, z)$ vagy $(x + 1, z + 1)$?

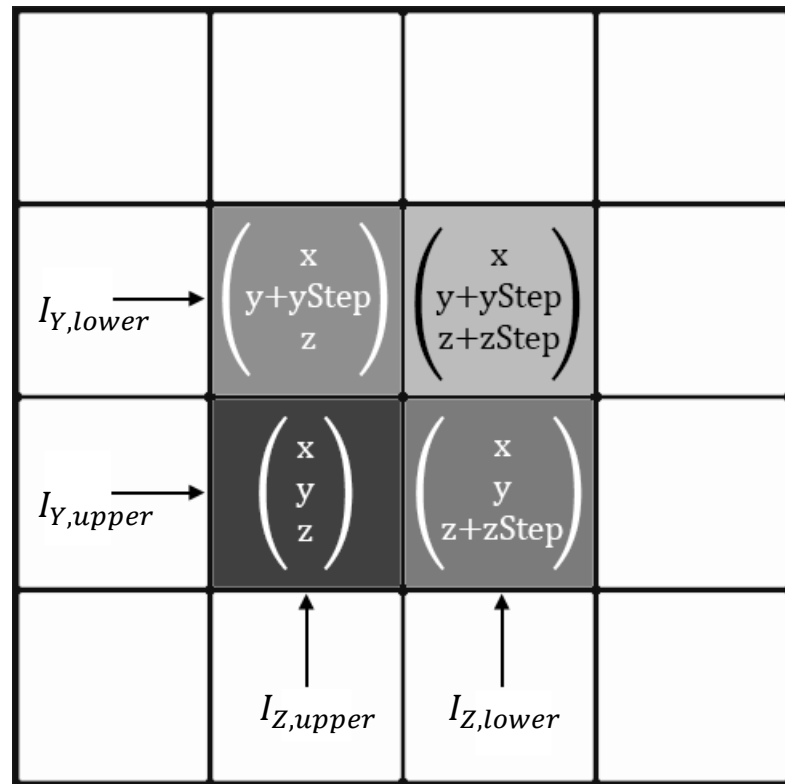
Antialiased Bresenham



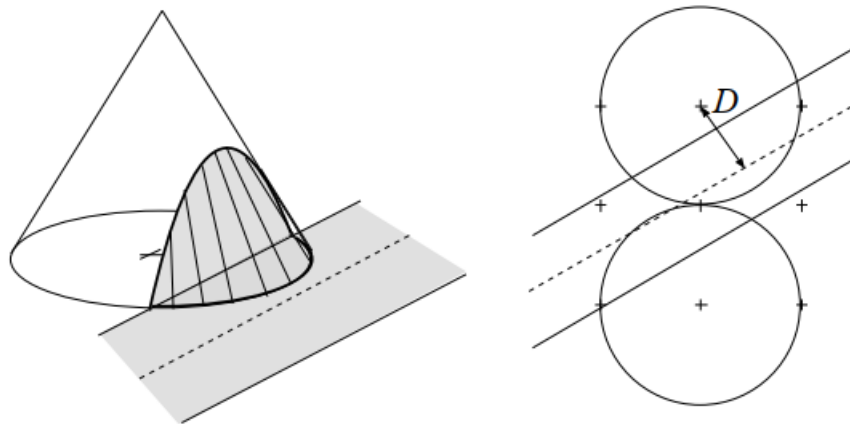
Metszetek területe:

$$I_{upper} \approx (1 - t)$$

$$I_{lower} \approx t$$



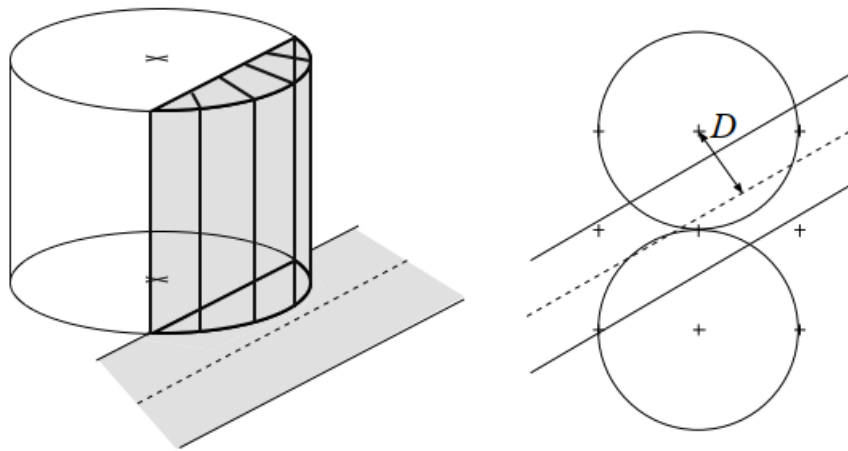
Gupta-Sproull



Előre kiszámolt $V[D]$ keresőtábla

$V(DY_H) \rightarrow$	$\begin{pmatrix} x \\ y+yStep \\ z-zStep \end{pmatrix}$	$\begin{pmatrix} x \\ y+yStep \\ z \end{pmatrix}$	$\begin{pmatrix} x \\ y+yStep \\ z+zStep \end{pmatrix}$	
$V(DY) \rightarrow$	$\begin{pmatrix} x \\ y \\ z-zStep \end{pmatrix}$	$\begin{pmatrix} x \\ y \\ z \end{pmatrix}$	$\begin{pmatrix} x \\ y \\ z+zStep \end{pmatrix}$	
$V(DY_L) \rightarrow$	$\begin{pmatrix} x \\ y-yStep \\ z-zStep \end{pmatrix}$	$\begin{pmatrix} x \\ y-yStep \\ z \end{pmatrix}$	$\begin{pmatrix} x \\ y-yStep \\ z+zStep \end{pmatrix}$	
	$\uparrow$ $V(DZ_L)$	$\uparrow$ $V(DZ)$	$\uparrow$ $V(DZ_H)$	

Hengeres Gupta-Sproull



Előre kiszámolt $V[D]$ keresőtábla

$V(DY_H) \rightarrow$	$\begin{pmatrix} x \\ y+yStep \\ z-zStep \end{pmatrix}$	$\begin{pmatrix} x \\ y+yStep \\ z \end{pmatrix}$	$\begin{pmatrix} x \\ y+yStep \\ z+zStep \end{pmatrix}$	
$V(DY) \rightarrow$	$\begin{pmatrix} x \\ y \\ z-zStep \end{pmatrix}$	$\begin{pmatrix} x \\ y \\ z \end{pmatrix}$	$\begin{pmatrix} x \\ y \\ z+zStep \end{pmatrix}$	
$V(DY_L) \rightarrow$	$\begin{pmatrix} x \\ y-yStep \\ z-zStep \end{pmatrix}$	$\begin{pmatrix} x \\ y-yStep \\ z \end{pmatrix}$	$\begin{pmatrix} x \\ y-yStep \\ z+zStep \end{pmatrix}$	
	$\uparrow$ $V(DZ_L)$	$\uparrow$ $V(DZ)$	$\uparrow$ $V(DZ_H)$	

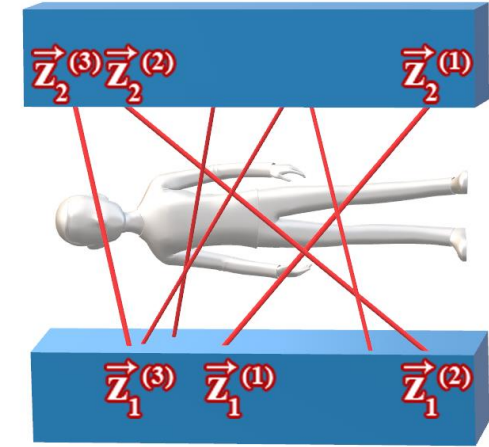
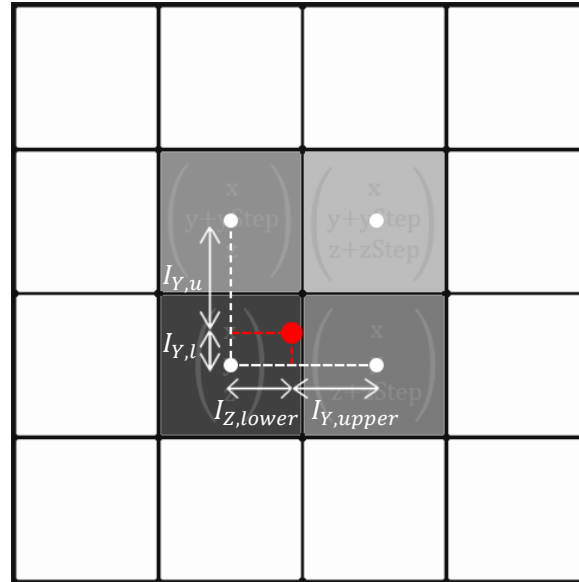
Implementáció

1. Hagyományos CPU program
2. CUDA platform → GPU program



Párhuzamos végrehajtás
a grafikus kártyán:

- ❖ 20.000 szál
- ❖ önállóan számolnak
→ nincs írási ütközés
- ❖ textúra memória
 - gyorsítótár → kevesebb memóiaforgalom
 - beépített hardveres trilineáris interpoláció



6 mérés:

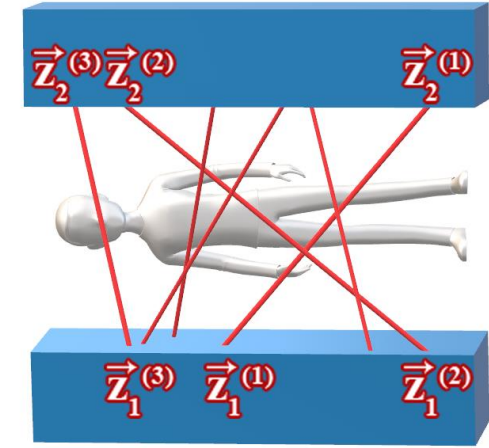
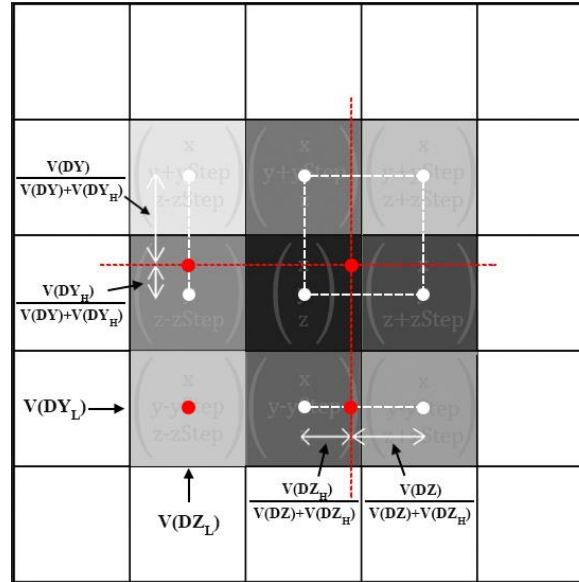
- ❖ 3 féle voxeltömbméret
(8x128x8, 8x256x8, 8x512x8)
- ❖ más-más arányban
aktívak bennük a voxelek
(12.5%, 25%, 50%, 75%)

1. Hagyományos CPU program
2. CUDA platform → GPU program



Párhuzamos végrehajtás
a grafikus kártyán:

- ❖ 20.000 szál
- ❖ önállóan számolnak
→ nincs írási ütközés
- ❖ textúra memória
 - gyorsítótár → kevesebb memóriaforgalom
 - beépített hardveres trilineáris interpoláció



6 mérés:

- ❖ 3 féle voxeltömbméret
(8x128x8, 8x256x8, 8x512x8)
- ❖ más-más arányban
aktívak bennük a voxelek
(12.5%, 25%, 50%, 75%)

Eredmények



01 | Pontosság



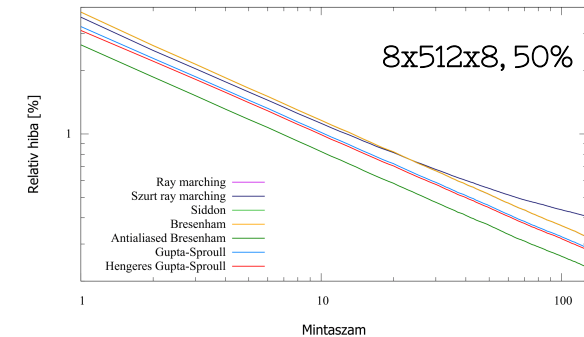
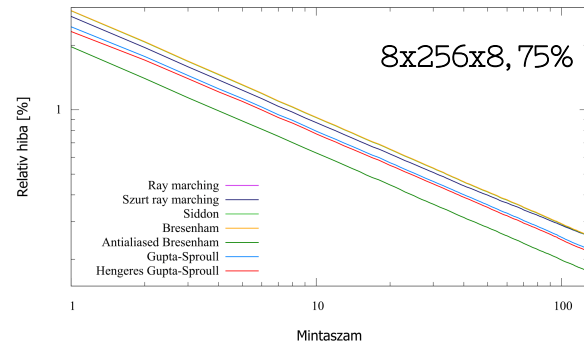
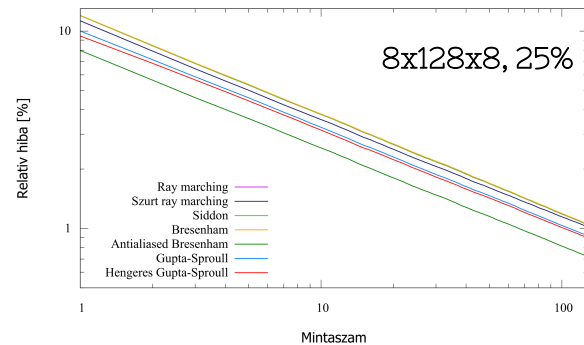
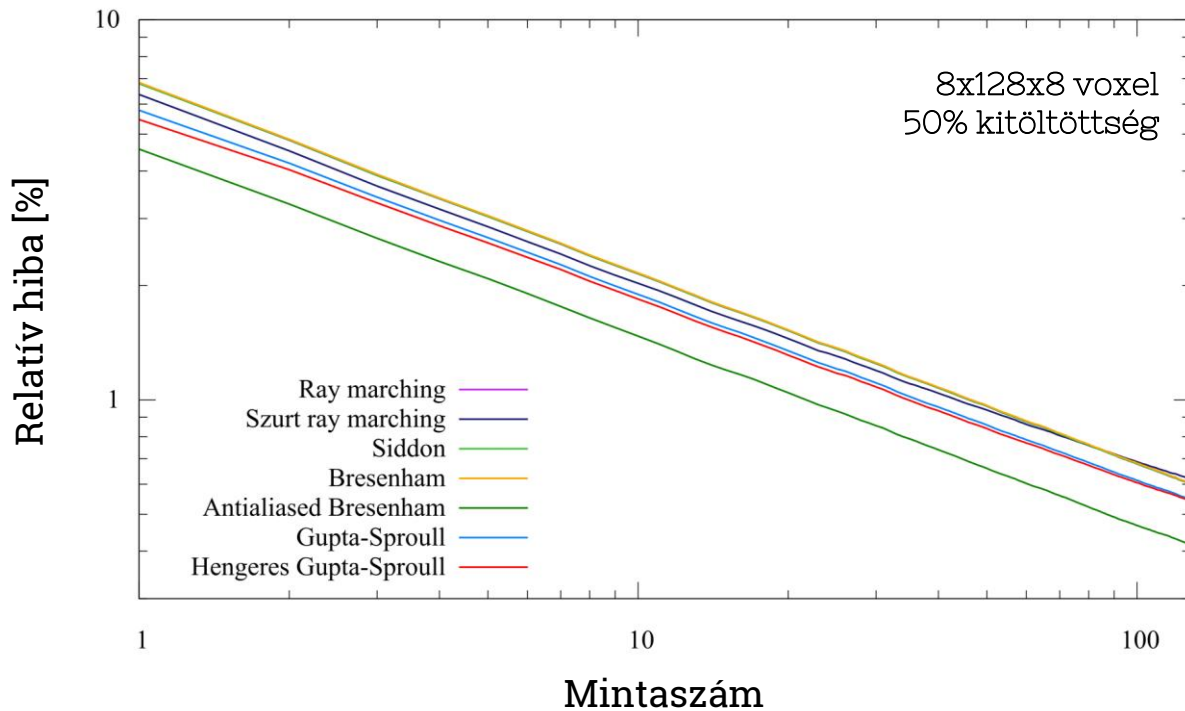
02 | Futásidő



03 | Hatékonyság



Pontosság



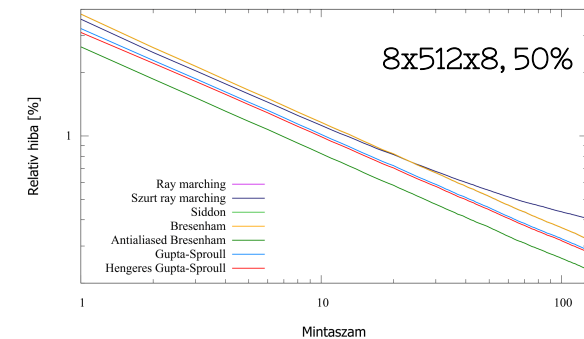
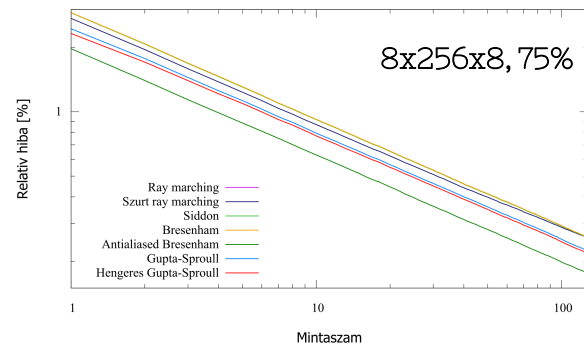
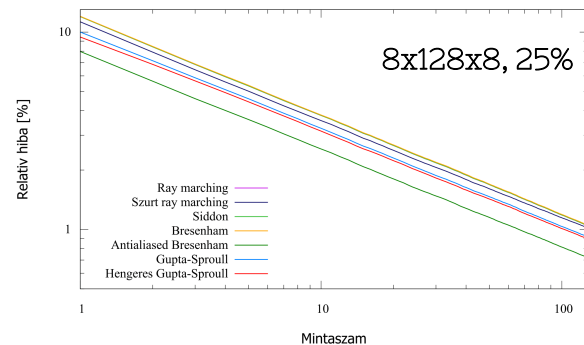


Pontosság

Relatív hiba [%]

Rangsor	Algoritmus	Átlagos hiba 128 minta után
1.	Antialiased Bresenham	0.4695%
2.	Hengeres Gupta-Sproull	0.5867%
3.	Gupta-Sproull	0.5982%
4.	Siddon	0.6760%
5.	Ray marching	0.6765%
6.	Bresenham	0.6784%
7.	Szűrt ray marching	0.6845%

Mintaszám





Futásidő

8x512x8 voxeltömbméreten,
milliszekundumban



Algoritmus	CPU futásidő	GPU futásidő interpoláció nélkül	GPU futásidő interpolációval
Ray marching	10 214	177	228
Szűrt ray marching	10 222	177	228
Siddon	6 671	304	317
Bresenham	6 167	128	114
Antialiased Bresenham	13 344	1 204	115
Gupta-Sproull	41 184	3 161	1 993
Hengeres Gupta-Sproull	41 334	3 257	1 994

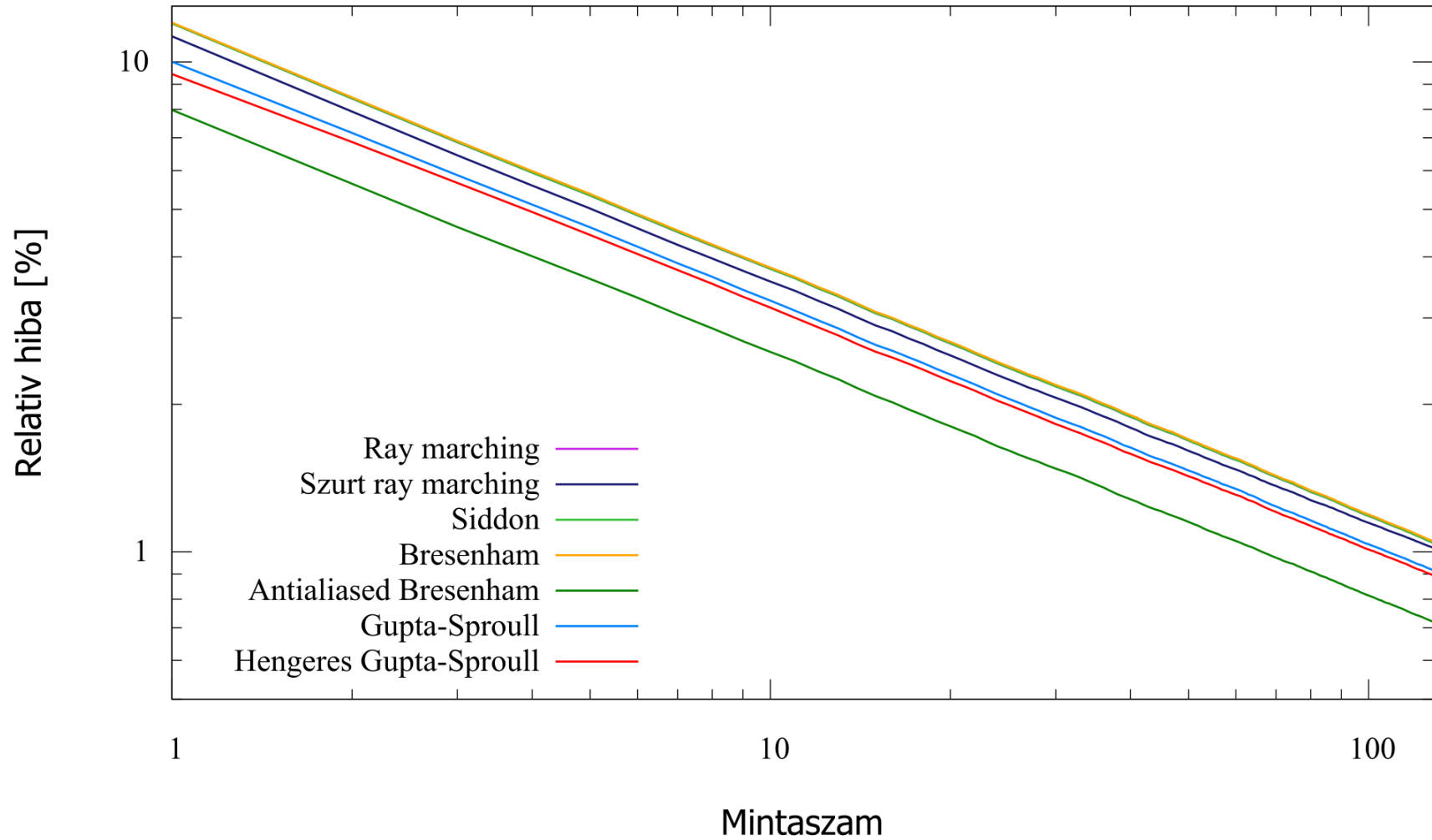


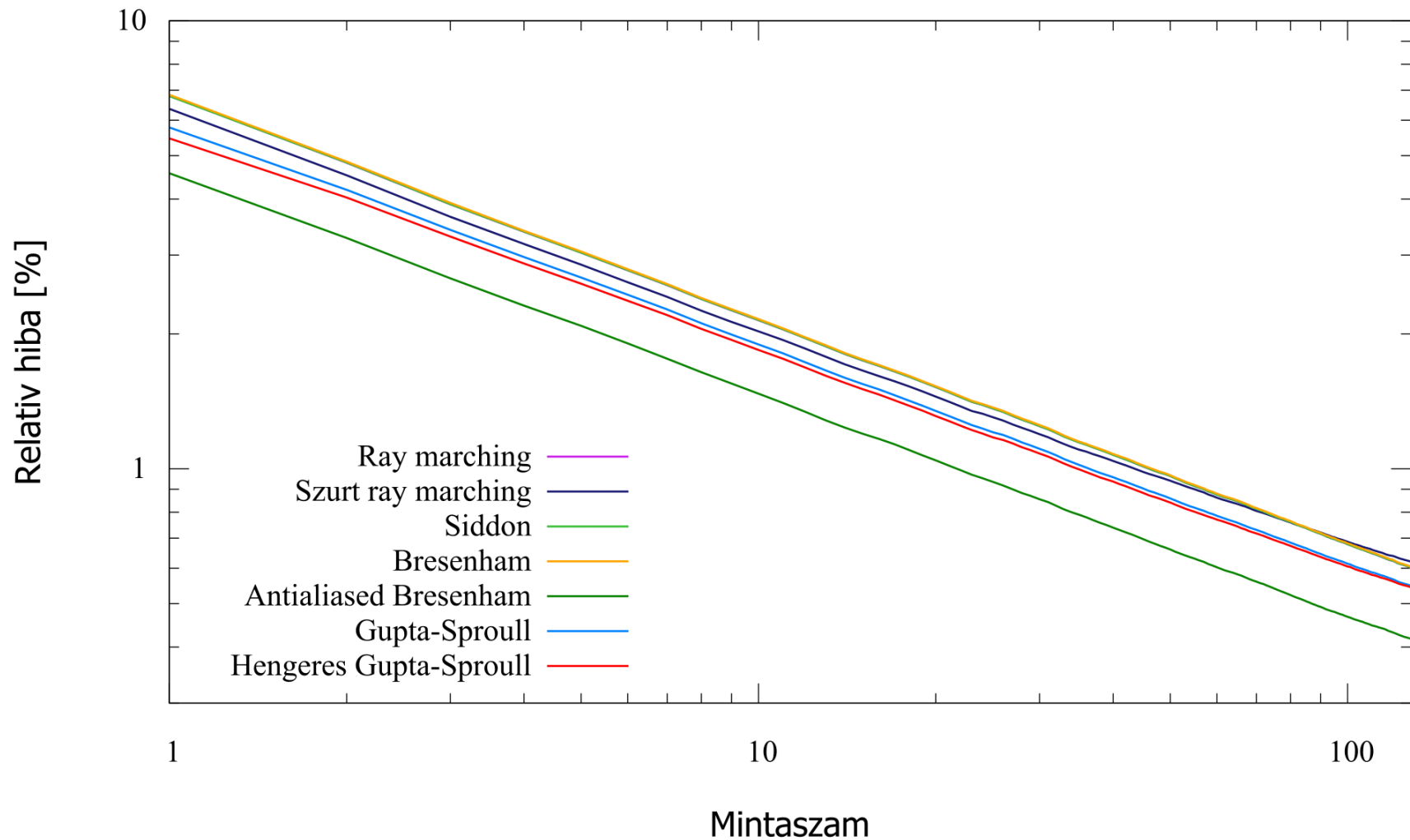
Hatékonyság

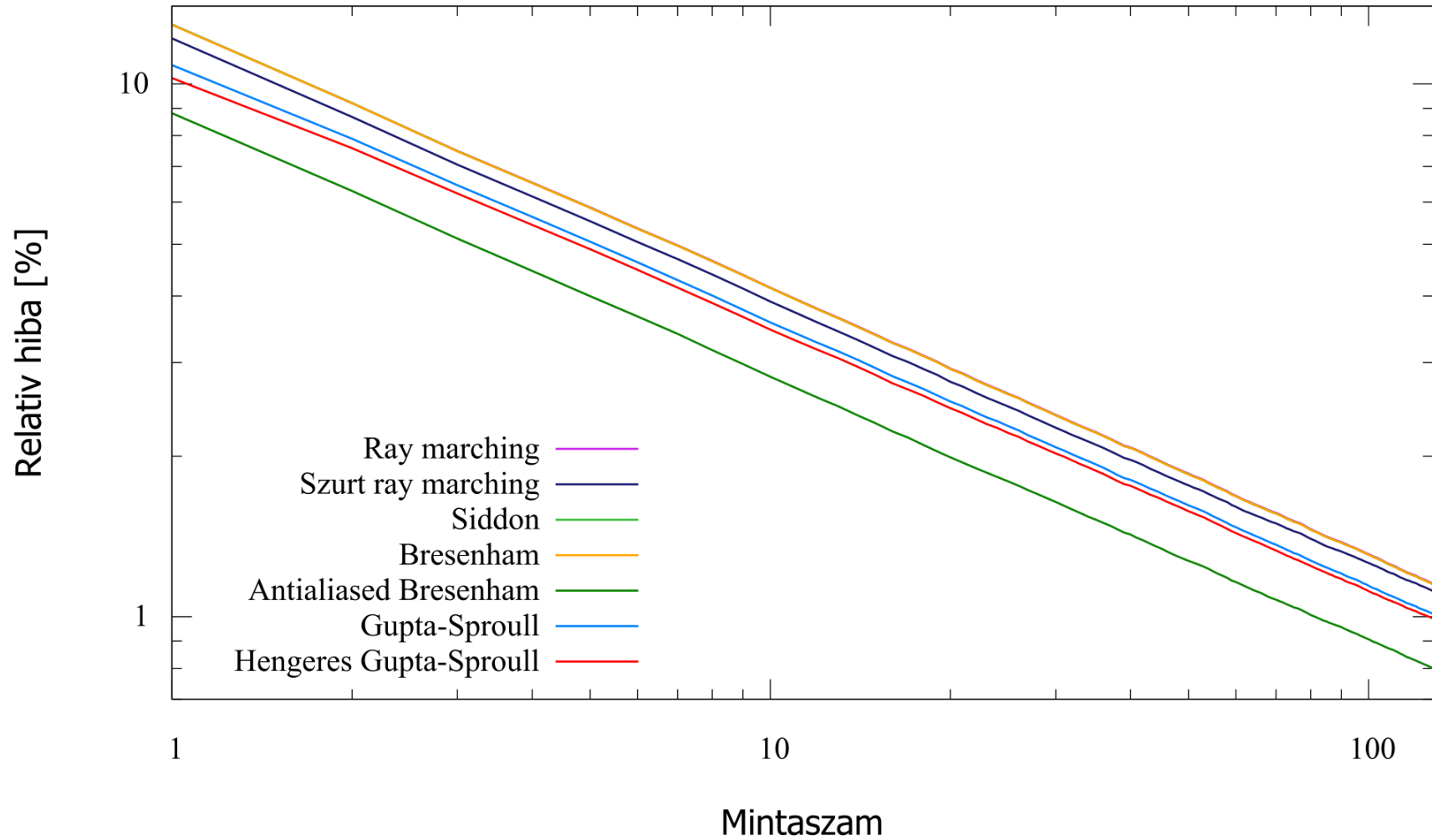
$$\text{hatékonyság} = \frac{1}{\text{hiba}^2 \cdot \text{futásidő}}$$

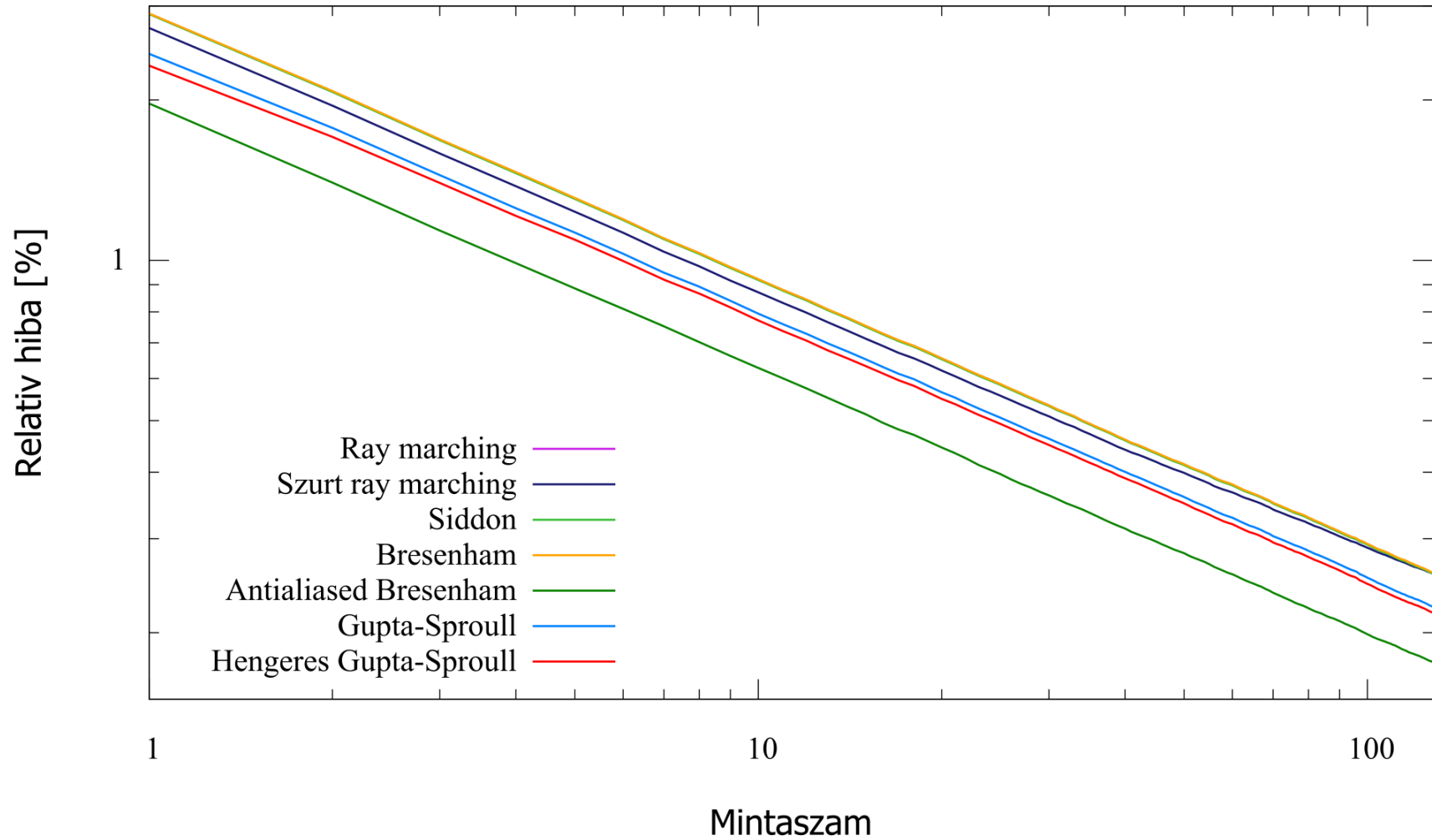
Algoritmus	1. mérés	2. mérés	3. mérés	4. mérés	5. mérés
Ray marching	145	437	60	1 191	417
Szűrt ray marching	152	411	64	1 197	266
Siddon	79	238	43	847	299
Bresenham	308	931	131	2 576	828
Antialiased Bresenham	632	1 879	265	5 472	1 599
Gupta-Sproull	24	66	10	200	60
Hengeres Gupta-Sproull	25	67	10	211	63

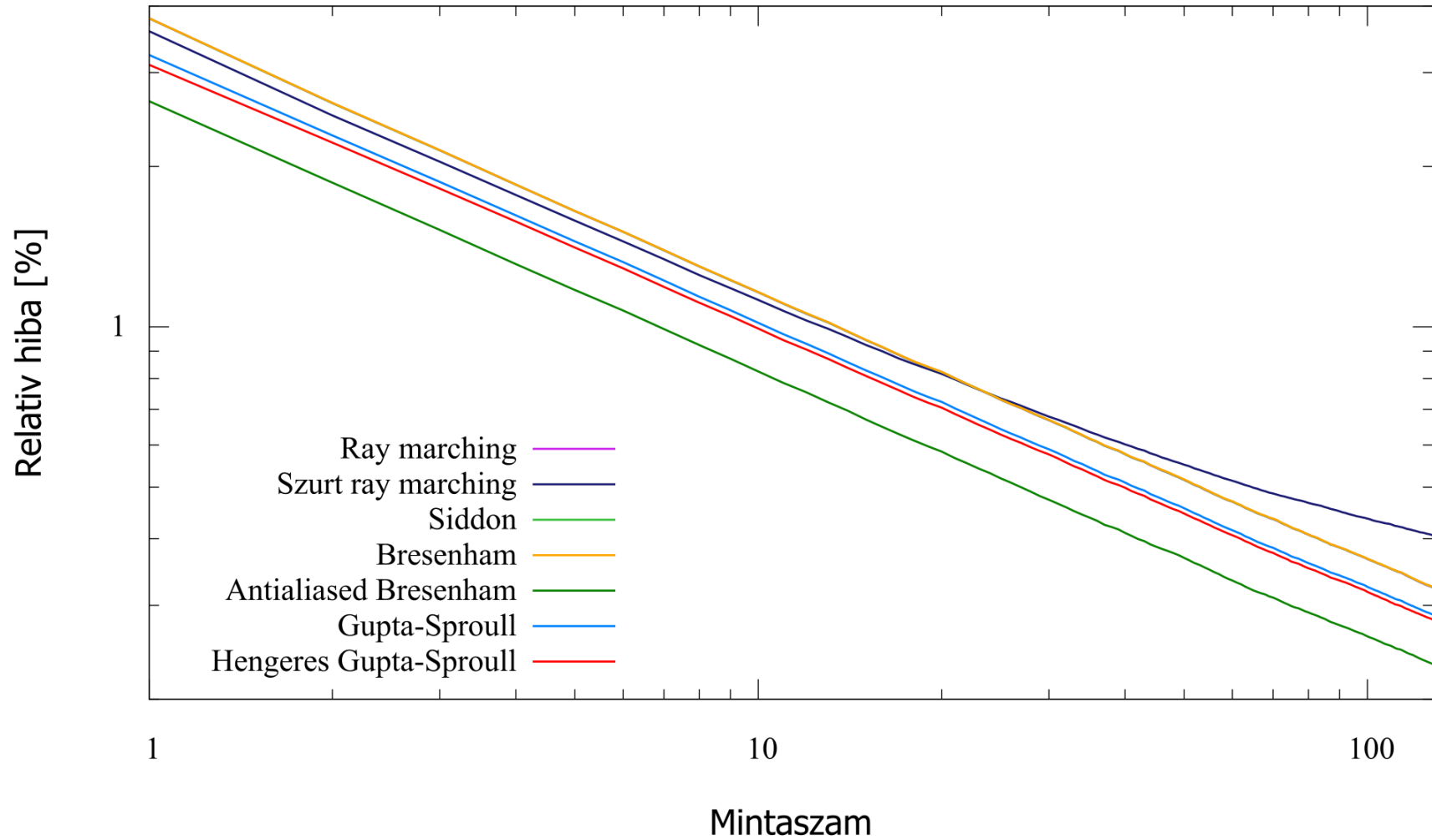
Függelék

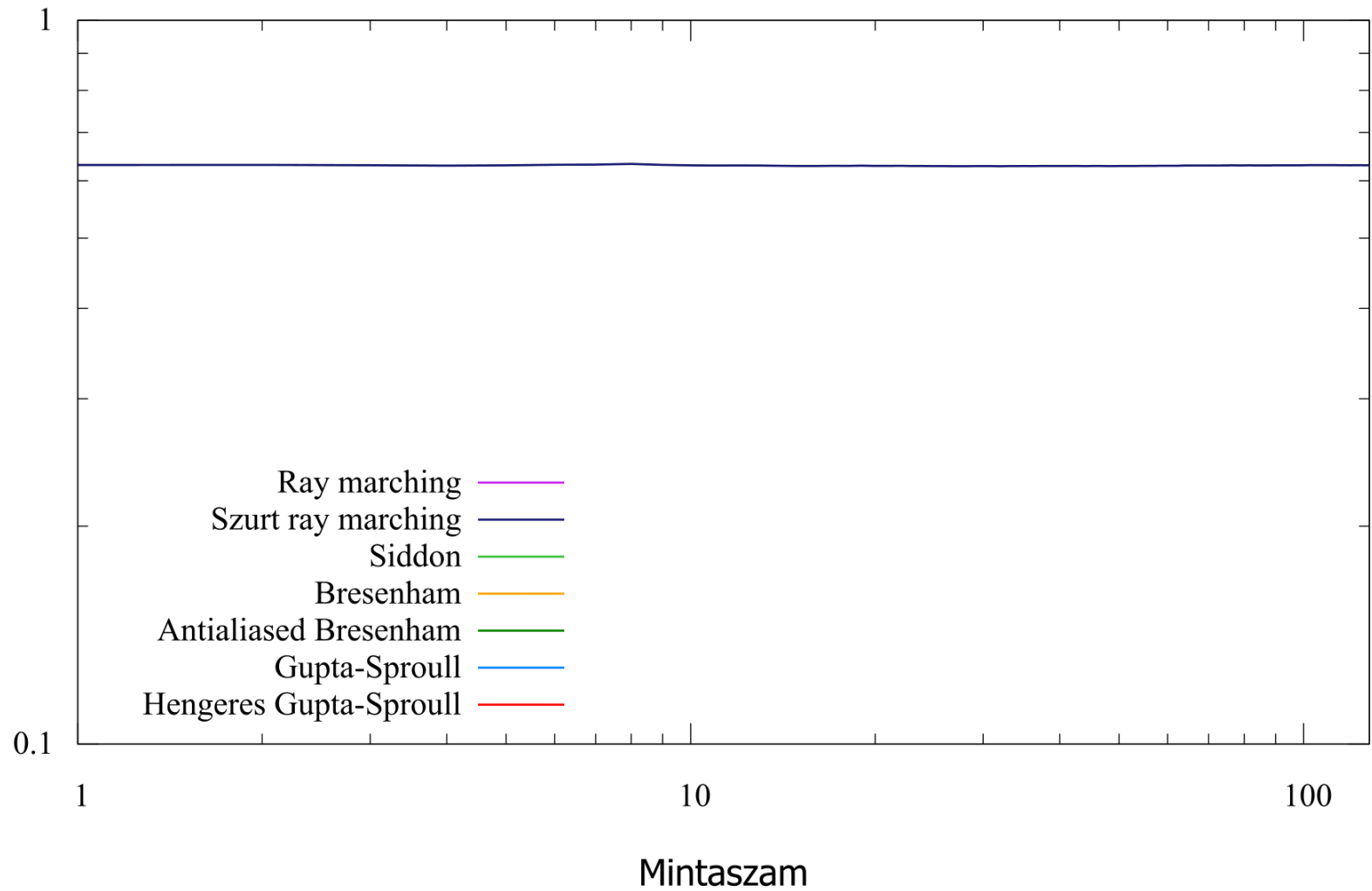


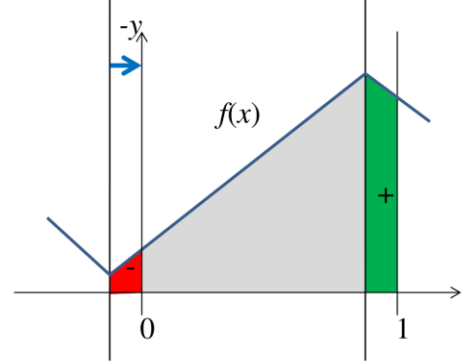
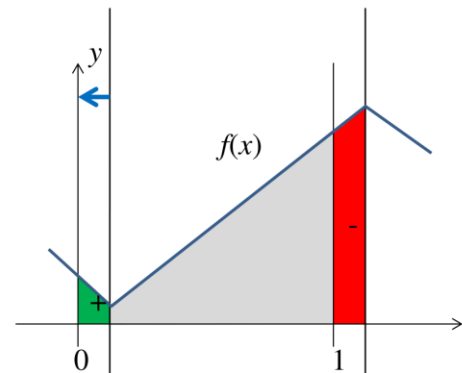
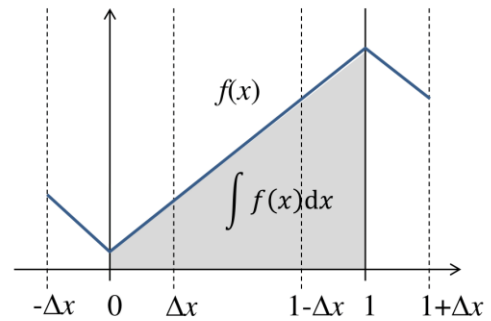
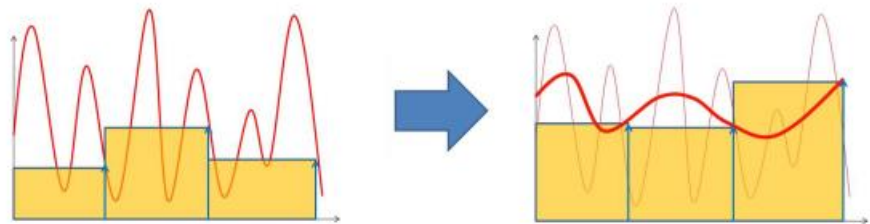




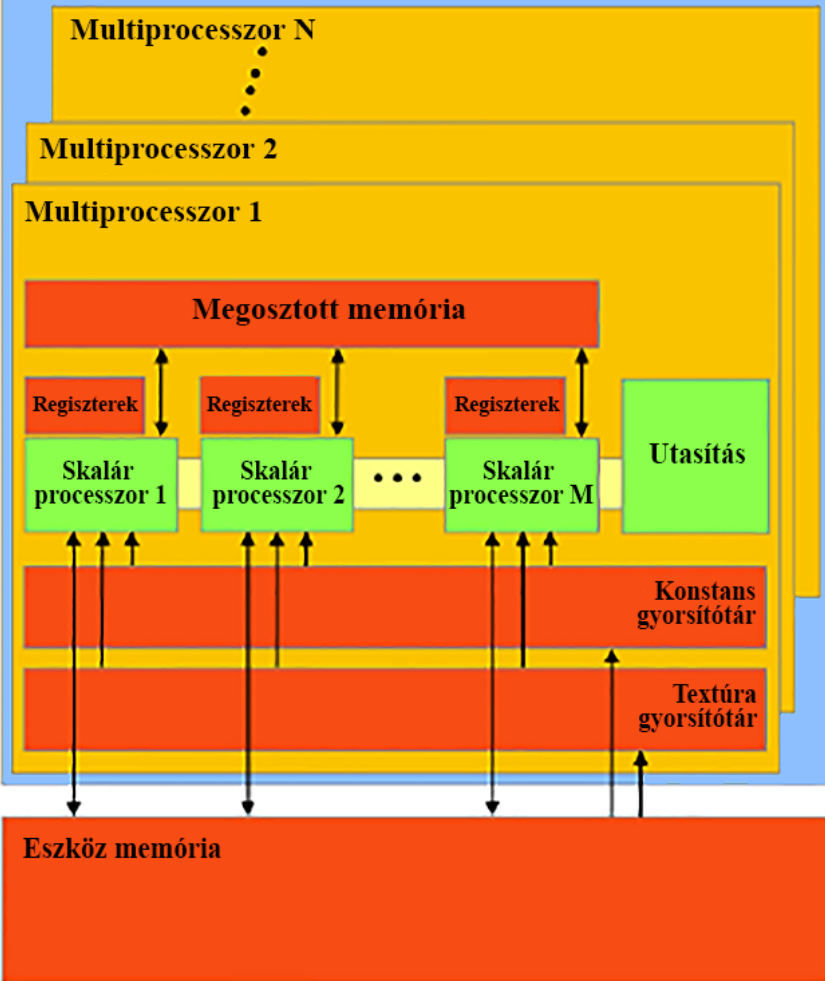








Eszköz



Térbeli lokalitás:

Szál 0

Szál 1

Szál 2

Szál 3

