

Analitikus MIP grafikus hardveren

Józsa Péter, Csébfalvi Balázs

Budapesti Műszaki és Gazdaságtudományi Egyetem
Irányítástechnika és Informatika Tanszék
cseb@iit.bme.hu

Absztrakt. Angiográfiás CT/MR adatokról legegyszerűbben úgy lehet vetületi képet készíteni, hogy minden vetítő sugár mentén megkeressük a 3D sűrűségfüggvény maximumát (Maximum Intensity Projection - MIP), és ezt az értéket rendeljük az adott sugárhoz tartozó pixelhez. A maximumot a gyakorlatban úgy határozzák meg, hogy a sugarak mentén kelőképpen sűrű trilineáris újramintavételezést végeznek, majd kiválasztják a maximális sűrűségű trilineáris mintát. Ezzel a módszerrel viszont potenciálisan elkerülhetjük a tényleges maximumot, ha az éppen két szomszédos mintavételi pont közé esik. Ennek valószínűségét csökkenteni lehet a mintavételi frekvencia növelésével, viszont minél több trilineáris mintát veszünk, annál lassabb lesz a megjelenítés. Ebben a cikkben egy olyan algoritmust mutatunk be, amely a maximumot analitikusan határozza meg, majd részletesen leírjuk a módszer hatékony GPU implementációját, amely interaktív megjelenítést biztosít.

1. Bevezetés

A CT/MR-angiográfia egy olyan diagnosztikai eljárás, amely az érrendszer vizsgálatára alkalmas. A vérerekbe kontrasztanyagot fecskendeznek, majd a testről CT/MR felvétel készül. Mivel a kontrasztanyag sűrűsége jóval nagyobb, mint az ereket környező szöveteké, ezért az érhálózat jól elkülöníthető a térfogati adatokban. Ennek köszönhetően az érhálózatról úgy kaphatunk átfogó vetületi képet, ha a vetítő sugarak mentén megkeressük a CT/MR adat által reprezentált 3D sűrűségfüggvény maximumát (Maximum Intensity Projection - MIP) [6]. Ehhez az eredeti folytonos függvényt rekonstruálni kell a rendelkezésre álló diszkrét mintavételezett reprezentációból, azaz a voxelek között interpolálni kell. A térfogatvizualizáció gyakorlatában a de facto standard rekonstrukciós eljárás a trilineáris interpoláció, melyet a mai modern grafikus processzorok (Graphics Processing Unit - GPU) hardveresen támogatnak. A MIP vizualizációt ezért kézenfekvő CPU helyett GPU-n implementálni, amely több nagyságrenddel nagyobb sebességet biztosít. A maximumot a gyakorlatban úgy közelítik, hogy a vetítő sugarak mentén egyenlő lépésközzel véges számú trilineáris mintát vesznek, és ezek közül kiválasztják a legnagyobbat. Ezzel a módszerrel viszont még akkor is elkerülhetjük a globális maximumot, ha a mintavételi távolságot relatíve kicsire állítjuk. A megjelenítés hatékonysága szempontjából a szűk keresztmetszetet a 3D textúraolvasások száma jelenti. Elvileg egy kocka alakú cellán átmenő sugárköteg

ugyanazt a nyolc sarokvoxelt veszi alapul a lokális maximum meghatározásához. Ennek ellenére sűrű mintavételezést és nagy képfelbontást feltételezve nyolcnál jóval több textúraolvasást vesz igénybe a hagyományos implementáció. Ahhoz, hogy elkerüljük a túlmintavételezésből adódó teljesítménycsökkenést, azaz minimalizáljuk a felesleges textúraolvasások számát, érdemes a lokális, adott cellán belüli maximumokat analitikusan kiszámolni. Ebben a cikkben erre a feladatra adunk egy hatékony GPU alapú megoldást.

2. Korábbi módszerek

A MIP vizualizáció legfőbb előnye, hogy ellentétben a hagyományos direkt térfogat-vizualizációval (Direct Volume Rendering - DVR) [8,3] nem igényli a vizualizációs paraméterek időigényes finomhangolását [5]. Az eredeti sűrűségfüggvény értékeit legfeljebb egy könnyen paraméterezhető átviteli függvénnyel képzik le intenzitásértékekre, melyek adott esetben jobban kihangsúlyozzák a vizsgálni kívánt tartományt. Erre a célra rendszerint egy kétparaméteres úgynevezett ablakozó függvényt (windowing function) használnak.

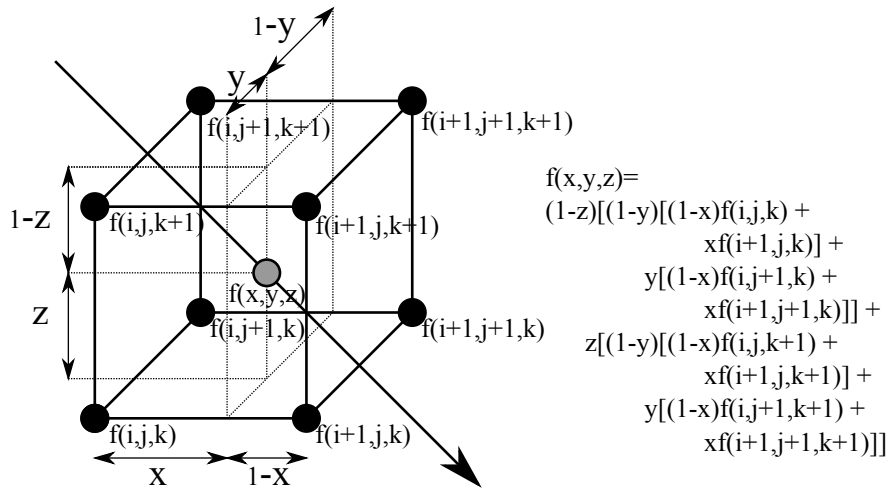
A klasszikus MIP modell egyik hátránya a mélységinformáció hiánya. A módszer alternatív változatai, mint a DMIP (Depth-Shaded Maximum Intensity Projection), LMIP (Local Maximum Intensity Projection) és a MIDA (Maximum Intensity Difference Accumulation) ezt a hiányosságot igyekeznek kompenzálni. A DMIP [6] egy mélységi intenzitásmodulációt alkalmaz a vetítés előtt, míg az LMIP [9] egy előre definiált küszöbérték felett megkeresi a sugár mentén az első lokális maximumot. A MIDA [1] lényegében ötvözi a hagyományos DVR és a MIP egymást kiegészítő előnyös tulajdonságait.

A MIP vizualizáció másik hátránya, hogy csak túlmintavételezéssel ad megbízható eredményt. A hagyományos DVR modellhez számos módszert kidolgoztak a túlmintavételezés elkerülésére. Ezek közül a legnépszerűbb a preintegrálás [4], amely a térfogat-vizualizációs integrál értékét két mintavételi pont között, a két pont sűrűségének kétdimenziós függvényeként írja fel. Ezt a függvényt egy előkalkulált 2D táblázattal lehet reprezentálni. A túlmintavételezés helyett ezért elegendő minden szomszédos pontpárra kiolvasni a táblázatban tárolt integrált. Egy másik megközelítés szerint a relatíve alulmintavételezett sugarak mentén célszerű a folytonos sűrűségprofil egy magasabb rendű konvolúciós szűrővel reprodukálni, ami szintén helyettesíti a drága textúraolvasásokat [7]. MIP vizualizációra egyik megközelítést sem lehet adaptálni. Trilineáris rekonstrukciót feltételezve a sugármenti sűrűségprofil egy szakaszonként harmadfokú polinom. Az egyes polinomiális szakaszok együtthatóit négy minta alapján lehet meghatározni. Ebből következik, hogy a lokális maximumokat egy 4D táblázatban kéne előkalkulálni, ami aránytalanul nagy tárolási kapacitást igényelne. A sugármenti magasabb rendű rekonstrukciós szűrést sem célszerű a MIP modellre alkalmazni, mivel a kernel simító hatása miatt potenciálisan csökkenthetjük a globális maximum értékét.

Módszerünk kifejlesztéséhez az inspirációt a szintfelületek direkt térfogatvizualizációjának analitikus kiértékelése adta [2]. Ennek lényege, hogy a sugarak mentén az első felületi dőfspontot úgy keressük meg, hogy a szakaszonként harmadfokú sugárprofil mentén detektáljuk azt a pontot, ahol a polinom éppen a szintfelületet definiáló küszöbértékkel egyenlő. Ez valójában nem más, mint egy harmadfokú polinom gyökeinek kiszámítása. Az analitikus kiértékelést MIP modellre úgy lehet adaptálni, hogy a klasszikus módon meghatározzuk a polinomiális sugárprofil szélsőértékeit. A potenciális maximumot ott keressük, ahol a profil deriváltja nulla. A harmadfokú polinom deriváltja másodfokú, ezért a feladat visszavezethető egy másodfokú egyenlet megoldására. A sugárprofil csak a gyököknek megfelelő pozíciókban értékeljük ki, és ezek maximuma adja az egyes cellákon belül a lokális maximumot.

3. Analitikus MIP

A hagyományos sugárkövetés és MIP kompozitálás során általában a sugár mentén egyenletes távolságokra lévő pontokban mintavételezzük a térfogati adatot. A mintavételi pontokhoz tartozó sűrűségértékeket leggyakrabban a mintavételi pontot magában foglaló cella sarkaiban ismert értékek trilineáris interpolációjával közelítjük (1. ábra).



1. ábra: Trilineáris interpoláció.

A módszer hátránya, hogy egyáltalán nem biztos, hogy a cellán belüli sűrűségmaximum pont a mintavételi pontba esik, akár az is elképzelhető, hogy a sugármenti maximumkeresés közben elhaladunk egy globális maximumérték mellett. A sűrűségérték cellán belüli eloszlásának lineáris közelítése azonban analitikus

formában is felírható és ennek felhasználásával a maximális érték a sugár mentén pontosan meghatározható. Az eloszlás lineáris közelítése a következő formában írható fel:

$$f(x, y, z) = Axyz + Bxy + Cyz + Dxz + Ex + Fy + Gy + H, \quad (1)$$

ahol x, y, z a térkoordináták, $A - H$ pedig a cella sarkaiban ismert értékekből számítható konstansok. A sugár paraméteres egyenlete:

$$\mathbf{p}(t) = \mathbf{p}_0 + \mathbf{d}t, \quad (2)$$

ahol t a sugárparaméter, \mathbf{d} a sugár irányvektora és \mathbf{p}_0 a sugár kezdőpontja. A második egyenletet skaláris formába átírva és az első egyenletbe helyettesítve megkapjuk a cellán belüli sűrűségeloszlás lineáris közelítését a sugár mentén a t sugárparaméter függvényében:

$$f(t) = at^3 + bt^2 + ct + d \quad (3)$$

ahol a, b, c, d egyelőre ismeretlen konstansok. Az a, b, c, d együtthatókat meghatározhatjuk például az első és a második egyenlet ismert együtthatói alapján, az első egyenlet $A - H$ együtthatóinak kiszámításához azonban szükségünk van a cella csúcspontjaiban ismert sűrűségértékekre. A gyakorlatban ez összesen nyolc legközelebbi szomszéd (nearest-neighbour - NN) textúralekérdezést jelentene. Egy harmadfokú polinom négy együtthatója azonban egyértelműen meghatározható akkor is, ha ismerjük a polinom helyettesítési értékét négy különböző változóérték esetén. Esetünkben ez azt jelenti, hogy ismernünk kell a mintavételi értéket négy különböző t pozícióban. Mint korábban láttuk, a mintavételi értéket megkaphatjuk egyetlen trilineáris textúralekérdezéssel, így mindössze négy trilineáris mintavétellel meghatározhatjuk a sugármenti sűrűségeloszlás analitikus alakját egy cellán belül. A korszerű GPU-k a trilineáris textúraszűrést hardveresen implementálják, ezért egy 3D-s textúra olvasásánál egy interpolált minta olvasásának sebessége gyakorlatilag megegyezik a legközelebbi szomszéd olvasásának sebességével. A szükséges textúraolvasások számának feleződése mellett az együtthatók meghatározásához szükséges algebrai műveletek száma is jelentősen csökken. A mintavételi pontok megválasztásakor a következő szempontokat kell szem előtt tartanunk:

- A négy pontnak különbözőnek kell lennie.
- A pontoknak ugyanazon cellán belül kell elhelyezkedniük, mivel az egyenletet egy cellán belüli interpolációból vezettük le.
- A numerikus stabilitás miatt célszerű a pontokat egyenletes távolságban felvenni.
- A $t = 0$ és $t = 1$ értékekre az egyenletek egyszerűbb alakot öltenek.

Ezen szempontok figyelembe vételével vegyük t értékét a sugár cellába történő belépési pontjában nullának, a kilépési pontban egynek, és válasszuk mintavételi pontnak ezeket a pontokat, valamint a két pont közötti szakasz harmadolópontjait ($t = 1/3$ és $t = 2/3$ helyen). A négy mintavételi értéket rendre f_1, f_2, f_3 ,

f_4 -gyel jelölve a következő négyismeretlenes lineáris egyenletrendszert kapjuk:

$$\begin{aligned}f_1 &= f(0) = d, \\f_2 &= f\left(\frac{1}{3}\right) = \frac{1}{27}a + \frac{1}{9}b + \frac{1}{3}c + d, \\f_3 &= f\left(\frac{2}{3}\right) = \frac{8}{27}a + \frac{4}{9}b + \frac{2}{3}c + d, \\f_4 &= f(1) = a + b + c + d,\end{aligned}\tag{4}$$

amelynek megoldása:

$$\begin{aligned}a &= -\frac{9}{2}f_1 + \frac{27}{2}f_2 - \frac{27}{2}f_3 + \frac{9}{2}f_4, \\b &= 9f_1 - \frac{45}{2}f_2 + 18f_3 - \frac{9}{2}f_4, \\c &= -\frac{11}{2}f_1 + 9f_2 - \frac{9}{2}f_3 + f_4, \\d &= f_1.\end{aligned}\tag{5}$$

Az együtthatók ismeretében már egyszerűen meghatározhatjuk az eloszlás maximumát a sűrűségfüggvény deriváltjának segítségével:

$$f'(t) = 3at^2 + 2bt + c.\tag{6}$$

Az eloszlásfüggvény maximumát a kezdőpontjában, a végpontjában, illetve a deriváltfüggvény zérushelyein veheti fel. Mivel a deriváltfüggvény másodfokú, ezért legfeljebb két zérushelye lehet, melyeket az $f'(t) = 0$ másodfokú egyenlet megoldásával kaphatunk meg. Ennek alapján a sűrűségfüggvény maximumára négy lehetőség adódik: f_1 , f_4 illetve $\frac{-b \pm \sqrt{b^2 - 3ac}}{3a}$.

3.1. A cellák bejárása

A cellán belüli analitikus sugárprofil meghatározására az előző fejezetben bemutatott módszer feltételezi, hogy a sugárkövetés során a sugár mentén haladva egyszerre egy cellát vizsgálunk, és ismerjük a sugárnak az adott cellára vonatkozó belépési és kilépési pontjait. A szokásos sugárkövetési módszerek viszont általában a sugár mentén egyenlő lépésközként vesznek mintát, így a mintavételi pontok legfeljebb véletlenül esnek cellahatárookra. Az analitikus MIP mintavételezéshez szükségünk volt egy olyan sugárkövetési módszerre, amely a sugár mentén celláról cellára haladva megadja a sugár metszéspontjait az adott cellával.

Az általunk alkalmazott algoritmus a számítógépes grafikában közismert DDA (Digital Differential Analyzer) elven alapszik [10]. A DDA algoritmusok diszkrét lépésekben haladva egy változó értékét lineárisan interpolálják két érték között úgy, hogy a változó pillanatnyi értékéhez minden lépésben hozzáadják az előre

kiszámított lépésenkénti növekményt. A naiv implementációhoz képest így jelentős mértékben javítható a teljesítmény, mivel lépésenként csak egyetlen összeadást kell kiértékelni. A DDA alapú sugárkövetés alapgondolata az, hogy a sugár mentén végighaladva inkrementálisan követjük a sugár metszéspontjait az egyes voxelsíkokkal, és minden lépésben az aktuális pozícióhoz legközelebb eső metszéspontot választjuk mintavételi pontnak. A bejárás során az egyes axiális síkokkal való legutolsó metszéspontokhoz tartozó sugárparaméter értékeket egy háromdimenziós vektorban tároljuk. Két szomszédos voxelsíkkal vett metszéspontokhoz tartozó paraméterkülönbség értékét (azaz a két voxelsík sugármenti távolságát) mindhárom koordinátatengely mentén előre kiszámítjuk, és szintén egy vektorban tároljuk. A bejárás során a legutolsó metszéspontokhoz minden lépésben hozzáadjuk a távolságvektort, így megkapjuk az egyes tengelyekhez tartozó következő síkmetszetek sugármenti távolságát. Az így kapott távolságok közül a legkisebb felel meg a sugár és a voxelrács következő metszéspontjának. A bejárás elején a kiindulási pont mindig egy metszéspont valamelyik voxelsíkkal és egyúttal egy belépési pont egy cellába, a következő metszéspont pedig megadja az ugyanazon cellához tartozó kilépési pontot. A következő lépésben ez a kilépési pont (amely egyúttal a következő cellába történő belépési pont is) válik az aktuális ponttá, és így tovább. Az algoritmus megértését segíti az alábbi pszeudokód:

```
void function TraceInit(vec3 start, vec3 dir,
    vec3 sampleDist, vec3 tAccum, vec3 tStep)
{
    for each axis do
        if (dir equals 0)
            tAccum = INFINITY
            tStep = 0
        else if dir is negative
            pos = round start up to nearest multiple of sampleDist
            tAccum = (start - pos) / dir
            tStep = -dir / sampleDist
        else
            pos = round start down to nearest multiple of sampleDist
            tAccum = (start - pos) / dir
            tStep = dir / sampleDist
    }

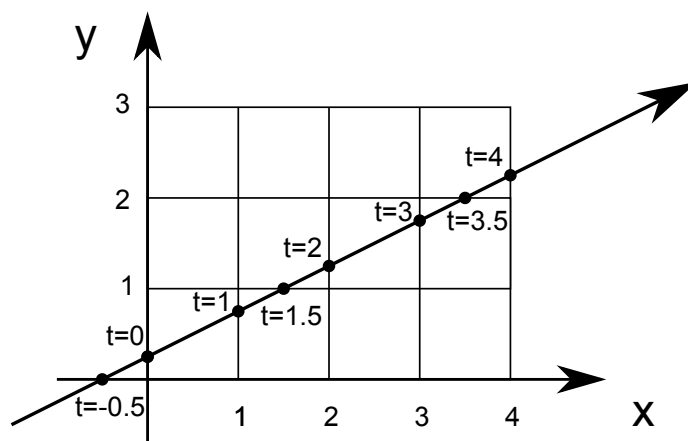
float function TraceStep(vec3 tAccum, vec3 tStep)
{
    tNext = tAccum + tStep
    tMin = min(tNext)
    for each axis do
        if tNext[axis] equals tMin
            tAccum[axis] = tNext[axis]
    return tMin
}
```

```
function RayTraversal(rayStart, rayDir, tEnd, sampleDist)
{
    t = 0
    entryPoint = rayStart
    TraceInit(rayStart, rayDir, sampleDist, tAccum, tStep)
    do
        tNext = TraceStep(tAccum, tStep)
        exitPoint = rayStart + tNext * rayDir

        // perform cell processing here

        t = tNext;
        entryPoint = exitPoint
    while t < tEnd
}
```

Kövessük végig az algoritmus működését egy egyszerűsített kétdimenziós példán, amely a 2. ábrán látható:



2. ábra: DDA sugárkövetés két dimenzióban.

Legyen a sugár belépési pontja a rácsba $[0, 0.25]$, irányvektora $[1, 0.5]$. A rács-távolság minden irányban 1. Ha a `TraceInit` függvény segítségével elvégezzük az inicializálást, akkor `tAccum` kezdeti értéke $[0, -0.5]$ lesz, míg `tStep`-re $[1, 2]$ adódik. Induljunk el a sugár mentén a kezdőpontból. Az állapotváltozók értékét az egyes lépések során az 1. táblázat tartalmazza. Az első lépésben `tAccum` kiindulási értékéhez hozzáadva `tStep` értékét $[1, 1.5]$ -öt kapunk, tehát a következő x-tengelymetszet a sugár mentén $t=1$, míg a következő y-tengelymetszet $t=1.5$ értéknél lesz. A kettő közül az x-tengelymetszet van közelebb, tehát ez lesz a következő mintavételi ponthoz tartozó `tNext` paraméterérték. Mivel az

x-tengelymetszetet választottuk, ezért t_{Accum} x-koordinátáját felülírjuk t_{Next} új értékével. A következő lépésben az így kiszámított t_{Next} értéket tesszük t aktuális értékévé, és megismételjük az előző lépésben leírtakat, azzal a különbséggel, hogy t_{Accum} megnövelése után most az y-tengelymetszet lesz közelebb, ezért azzal kell tovább számolnunk. A további lépések a táblázat alapján már egyszerűen követhetők.

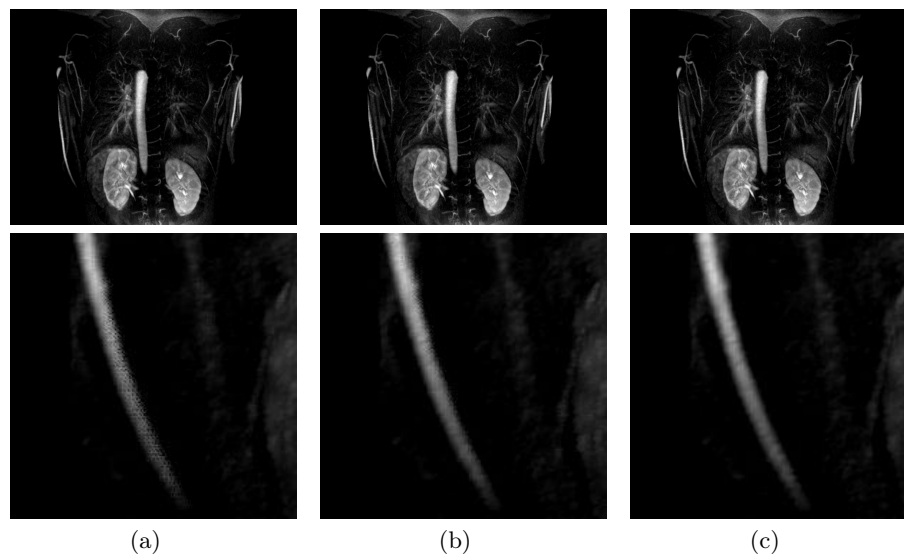
Lépés	t	t_{Accum}	$t_{\text{Accum}} + t_{\text{Step}}$	t_{Next}
1.	0	[0, -0.5]	[1, 1.5]	1
2.	1	[1, -0.5]	[2, 1.5]	1.5
3.	1.5	[1, 1.5]	[2, 3.5]	2
4.	2	[2, 1.5]	[3, 3.5]	3
5.	3	[3, 1.5]	[4, 3.5]	3.5

1. táblázat: Az állapotváltozók értékei a sugárkövetés során.

A cellák bejárása könnyen kiterjeszthető három dimenzióra, néhány további szempontra azonban érdemes felhívni a figyelmet. A legfontosabb, hogy a legközelebbi metszéspont vizsgálatakor a távolságokra több azonos érték is adódhat a különböző irányokban, ha a sugár pont egy rácsvonalon halad keresztül. Ilyenkor a metszéspont frissítésénél vigyázni kell, hogy minden olyan irányban frissítsük a az állapotvektort, amelynél a távolság megegyezik a minimummal. A gyakorlati implementációban ez egy kritikus pont, ugyanis előfordulhatnak elfajuló esetek, amikor például a sugár egy cella élén halad keresztül, és a numerikus pontatlanságok miatt több nagyon közeli, de nem azonos metszéspontot kapunk. Az implementációban ezért a minimummal azonosnak tekintünk egy távolságot akkor is, ha nagyon közel van hozzá, így az algoritmus ezekben az esetekben is robusztus módon működik. A DDA típusú sugárkövetési algoritmus inicializálásához ismernünk kell a sugárnak a teljes voxelstruktúrára vonatkozó belépési és kilépési pontjait is. Az algoritmus bemutatásánál erre külön nem térünk ki, mivel ezek hagyományos módszerekkel (például CPU-n algebrai úton, GPU-n proxy geometria renderelésével) egyszerűen meghatározhatók.

4. Eredmények

Módszerünket gyakorlati angiográfiás CT és MR adatokon teszteltük. A 3. ábra az analitikus és a hagyományos MIP eredményeit mutatja. Az első kép (a) renderelésénél a sugármenti mintavételi távolságot úgy állítottuk be, hogy az megegyezzen a voxelek közötti távolsággal. Az alulmintavételezés miatt jól látható, hogy a vérerek szélén nem sikerül a tényleges maximumot detektálni. Ha a mintavételi frekvenciát nyolcszorosára növeljük (b), akkor pontosabb képet kapunk, viszont így is észre lehet venni hibás pixeleket. Ezzel szemben az analitikus MIP kiértékelésnél (c) nem is kell a radiológusnak mintavételi frekvenciát megadni,



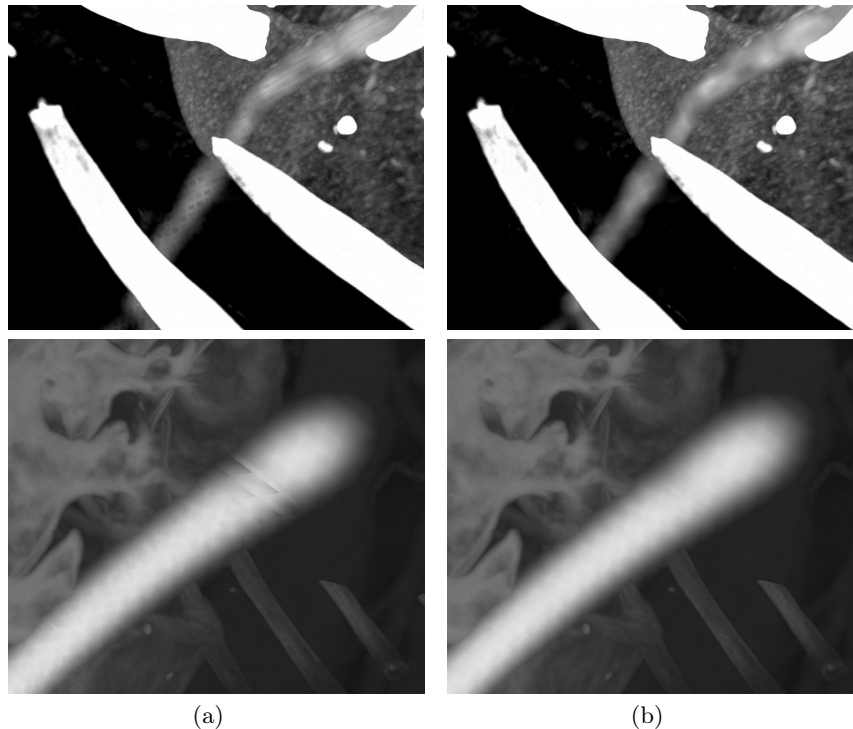
3. ábra: Különböző MIP implementációkkal renderelt képek. (a): Hagyományos MIP alacsony mintavételi frekvencia mellett. (b): Hagyományos MIP nyolcszor nagyobb mintavételi frekvencia mellett. (c) Analitikus MIP.

mivel az algoritmus minden pixelre eleve a pontos maximumot számolja ki, nem pedig annak egy numerikus közelítését. A GPU implementáció interaktív renderelést biztosít, a teljesítményt pedig főként a textúraolvasások száma határozza meg. Az analitikus MIP-hez cellánként három mintavételezésre van szükség, a hagyományos MIP viszont ennél rendszerint jóval több mintából tudja csak az analitikus megoldást megfelelően közelíteni.

A pontos maximumhelyek elkerülésén túl a hagyományos MIP kiértékelésénél a trilineáris mintavételezés alacsony mintavételi frekvencia mellett vizuális artifaktumokat eredményezhet. A 4. ábrán jól látható, hogy ezeket az artifaktumokat az analitikus megoldás jelentősen csökkenti. Tapasztalatunk szerint hagyományos MIP implementációt alkalmazva legalább négyszeres, de esetenként csak nyolcszoros felülmintavételezéssel lehet hasonló képminőséget elérni. Fontos megjegyezni, hogy a hagyományos MIP már négyszeres felülmintavételezéssel is lassabb mint az analitikus MIP. A másodpercben mért futási időket a 2. táblázat mutatja. A méréshez AMD Radeon HD5670 grafikus kártyán rendereltünk 1200x1024-es felbontású képeket.

5. Konklúzió

Ebben a cikkben megmutattuk, hogy a mai grafikus kártyákon MIP képeket analitikusan is ki lehet számolni. Módszerünk egyrészt kihasználja a sugarak inkrementális kiértékelését, másrészt a behuzalozott trilineáris interpolációt. A



4. ábra: Hagyományos (a) és analitikus (b) MIP összehasonlítása vizuális artifaktumok szempontjából.

hatékony implementációnak köszönhetően a megjelenítés interaktív, és a felhasználónak nem kell külön mintavételi frekvenciát beállítani.

6. A továbbfejlesztés lehetőségei

Az analitikus MIP kiértékelést akkor lehetne igazán hatékonyá tenni, ha egy objektumsorrendi megközelítést alkalmazva, a cellákat sorra egymás után tudnánk a képsíkra vetíteni. A memóriaolvasások száma ezáltal jelentősen csökkenne, mivel az összes adott cellán átmenő sugár lokális maximumát ugyanabból a nyolc sarokvoxelből lehetne számítani. Ezt a megközelítést különösen akkor érdemes alkalmazni, ha a renderelt kép nagyfelbontású, azaz a térfogati adatot jelentősen túlmintavételezzük a képtérben.

Köszönetnyilvánítás

A kutatás a TÁMOP-4.2.2.B-10/1-2010-0009 és az OTKA K-101527 projektek keretén belül valósult meg. Módszerünket a Voreen (voreen.uni-muenster.de)

Tesztadat	Méret	MIP 1x	MIP 4x	MIP 8x	AMIP
cenovix-ct1-abdomen	512x512x361	0,375	1,225	2,252	0,654
quanredeusix	384x22x512	0,106	0,14	0,188	0,103
felix-cemra-hires	576x118x576	0,131	0,313	0,62	0,202
mri	256x256x124	0,249	0,72	1,408	0,335

2. táblázat: Hagyományos és analitikus MIP összehasonlítása futási idők alapján.

térfogat-vizualizációs keretrendszer felhasználásával implementáltuk. A tesztadatokat az OsiriX DICOM Viewer (www.osirix-viewer.com/datasets/) adatbázisából vettük.

Irodalom

1. Stefan Bruckner and Eduard Gröller. Instant volume visualization using maximum intensity difference accumulation. *Computer Graphics Forum*, 28(3):775–782, 2009.
2. Yu-Tai Ching Chyi-Cheng Lin. An efficient volume- rendering algorithm with an analytic approach. *The Visual Computer*, 12(10):515–526, 1996.
3. Robert A. Drebin, Loren Carpenter, and Pat Hanrahan. Volume rendering. *SIGGRAPH Computer Graphics*, 22(4):65–74, 1988.
4. Klaus Engel, Martin Kraus, and Thomas Ertl. High-quality pre-integrated volume rendering using hardware-accelerated pixel shading. In *Proceedings of the ACM SIGGRAPH/EUROGRAPHICS workshop on Graphics hardware*, pages 9–16, 2001.
5. Elliot K. Fishman, Derek R. Ney, David G. Heath, Frank M. Corl, Karen M. Horton, and Pamela T. Johnson. Volume Rendering versus Maximum Intensity Projection in CT Angiography: What Works Best, When, and Why1. *Radiographics*, 26(3):905–922, 2006.
6. Wolfgang Heidrich, Michael McCool, and John Stevens. Interactive maximum projection volume rendering. In *Proceedings of the 6th conference on Visualization '95*, pages 11–18, 1995.
7. Byeonghun Lee, Jihye Yun, Jinwook Seo, Byonghyo Shim, Yeong-Gil Shin, and Bohyoung Kim. Fast high-quality volume ray casting with virtual samplings. *IEEE Transactions on Visualization and Computer Graphics*, 16:1525–1532, 2010.
8. Marc Levoy. Display of surfaces from volume data. *IEEE Computer Graphics Applications*, 8(3):29–37, 1988.
9. Yoshinobu Sato, Yoshinobu Sato, Shin Nakajima, Nobuyuki Shiraga, Shinichi Tamura, and Ron Kikinis. Local maximum intensity projection (LMIP): A new rendering method for vascular visualization. *Computer Assisted Tomography*, 22(6):912–917, 1998.
10. László Szirmay-Kalos. *Theory of Three Dimensional Computer Graphics*. Akadémiai Kiadó, Budapest, 1996.