

## Absztrakt

A dolgozat célja a fák növekedésének minél valóságosabb modellezése. Ennek során egy kiinduló állapotból a fa biológiából megismert viselkedését követve folyamatosan szimulálni lehet annak várható változásait. Ehhez figyelembe kell venni a fajra jellemző alaki tulajdonságokat, az őt körülvevő fizikai környezetet (környező objektumok elkerülése), illetve a bejövő fény irányát (a növény a fény felé fordul növekedés során). A fény irányának meghatározására videókártyával gyorsított sugárkövető algoritmus szolgál. A rendszer egy időben több fát is képes szimulálni, ezek növekedésük során hatással vannak mind saját magukra, mind a környező fák fejlődésére. A szimuláció végeredményeként létrejövő háromdimenziós famodellek tükrözik az őket meghatározó faji jellegzetességeket a környezeti hatások által torzítva. Így egy valóságos képet adva a fa adott környezetbe való beilleszkedésének menetéről.

## Kulcsszavak

Szimuláció, 3D grafika, GPU programozás, Ray-tracing, Monte-Carlo módszer

## Motiváció

A fák növekedése bonyolult és rengeteg tényező által befolyásolt folyamat. Viszont jó lenne előre tudni, hogy mi lesz egy fával sok évvel azután, hogy elültették valahová. Legyen szó akár egy kertés ház udvarának befásításáról, egy park tervezéséről, vagy akár egész erdők telepítéséről. Továbbá az éghajlati változások hatása is nehezen megjósolható a fákra nézve. Ezekre a problémákra megoldást jelenthet egy szimuláció, amely a fák növekedését modellezi figyelembe véve az őket körülvevő környezetet, és természeti hatásokat.

## Hasonló fejlesztések

- ▶ Számos kutatás foglalkozik a fák növekedésének biológiai és szimulációs háttérével. Ezek főleg az ágak alakjával és az elágazásokkal foglalkoznak. Elterjedtek az L-Systemen alapuló alkalmazások. Ezek többnyire matematikai modellek, gyakran nem is térben működnek.
- ▶ Egy másik hasonló terület a különféle 3D fa modellek generálása. Ezek gyakran megjelennek játékprogramokban és egyéb tájkép generáló rendszerekben. Ezekben az esetekben a szép, valóságos fa generálása a cél, ezek azonban nem veszik figyelembe a környezeti paramétereket, nincs biológiai szimuláció a háttérben.
- ▶ Harmadikként pedig érdemes megemlíteni az erdészeti alkalmazásokat. Ezek tipikusan egy megadott területen a fahozam számítására alkalmasak, nem egy-egy fát szimulálnak. A grafikus megjelenítés pedig nem cél.

## Sikerek

- ▶ Kari TDK 1. helyezett
- ▶ OTDK 2. helyezett
- ▶ 1 konferenciacikk, 1 folyóiratcikk

## Saját módszer bemutatása

A munka első lépéseként fel kellett térképezni a fák növekedését befolyásoló biológiai folyamatokat: fa környezetében lévő akadályok, fény hatása, víz hatása, gravitáció hatása, faji tulajdonságok hatása (elágazások típusa, szöge, darabszáma, hosszanti/szélességi növekedés, stb.).

A környezeti objektumok és fényforrások beállítását követően a fa növesztés egyetlen kezdő pontból (ágból) indul az alábbi szimulációs lépéseket követve:

- ▶ Az új ág(ak) helyének meghatározása a szülő ágon
- ▶ Alap növekedési irány kiszámítása
- ▶ Növekedési irány módosítása a legerősebb látható fényforrás irányába
- ▶ Ütközések vizsgálata, a növekedési irány módosítása szükség szerint
- ▶ A valósághoz is hasonló természetes ívek számítása
- ▶ Leszármazott ágak növesztésére rekurzív hívás

A sugárkövetés egy jól ismert technika a 3D grafika terén, általában egy pixel színének meghatározására használják. Jelen esetben a technika a faágat érő legerősebb fény irányának meghatározására szükséges. A feladat azért összetett, mivel egy faág végén található receptornak nem biztos, hogy közvetlen rálátása van a fényforrásra, figyelembe kell venni a környező objektumok/más fák/saját ágak árnyékolásának hatását, illetve a különböző objektumokról visszaverődő fényeket. Ehhez a vizsgált pont körüli teljes gömbfelületen futó Monte-Carlo módszert használ a modell.

A sugárkövetés alkalmazásával jelentősen megnőtt a rendszer számításigénye, ezért GPU alapú gyorsításra volt szükség. A CUDA alapokon elkészített algoritmus hasonlóan működik mint a CPU megvalósítás, csak az egyes sugarakat nem egymás után, hanem egymással párhuzamosan vizsgálja. Ez jelentősen növelte a rendszer teljesítményét, míg a pontosságot nem rontotta.

A növekedést végző szimuláció egy Blender pluginként lett megvalósítva, ennek megfelelően GUI írására nem volt szükség. A fa környezetét egyszerűen a 3D modellező eszközben kell létrehozni (falak, fényforrások, stb.). Kiegészítő xml állományokban kell megadni a megadott fa fajta paramétereit (pl. egészen más egy fenyő vagy egy nyírfa viselkedése). Ezt követően pedig megadott pontot/pontokat kijelölve indítható a szimuláció.

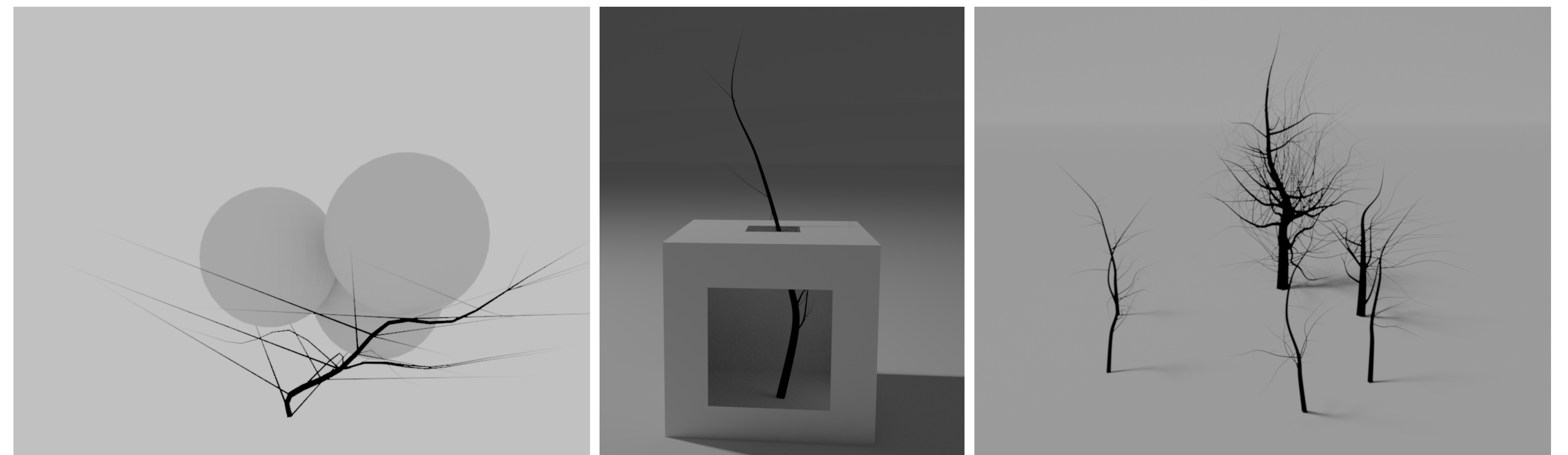
## Eredmények értékelése

Mivel a szimuláció eredménye egy 3D fa modell, ezért a numerikus értékelés nehezen valósítható meg. Az egyes funkciók (akadályok elkerülése, fény hatása, fa jellemzőinek megfelelő növekedés) egyesével külön-külön tesztelhetők. Ezt követően pedig azt érdemes megvizsgálni, hogy az elkészült modellek megfelelnek-e a természetben megszokottaknak.

Az első ábrán látható az objektumoknak való ütközés hatása. A megadott pontból induló fa nem tud az ideális irányba növekedni, mivel ott nekiütközik egy már ottlévő objektumnak. Ilyenkor a természetben megszokott módon folytatja a növekedést elkerülve az akadályt.

A második ábra a fény felé haladást mutatja be. A fényforrás nem látható a képen, baloldalt fent található. A fa kezdetben a kocka belső faláról visszaverődő fény felé kezdett növekedni, később azonban közvetlen rálátást kapott a fényforrásra, ennek hatására megváltozott a növekedés iránya. Ennek hatására automatikusan ki is jött a kocka tetején látható résen keresztül.

A harmadik ábra arra mutat példát, hogy egy időben több fa növekedése során azok egymásra hatással vannak. A nagyobb fa árnyékolja a kisebbeket, azok növekedését lelassítva, így az még nagyobb előnyre tesz szert. Egy fa különböző ágai persze egymásra is hatással vannak.



Mivel a szimuláció nagy erőforrásigénnyel bír, ezért lényeges lépés volt a GPU alapú gyorsítás. A jobboldali grafikon a CPU-GPU futásidőt mutatja a szimulációs időtáv függvényében. Mivel idősebb a fa, annál több ága van, így a számításigény exponenciálisan növekszik.

Ahogy várható, kevés számítás esetén még érdemesebb a CPU-t használni. Nagy méretű fák/nagy számításigény esetén viszont már jól látható a GPU előnye.

