

# AZ ÚJ GENERÁCIÓS GRAFIKUS HARDVEREK BEVONÁSA A MEGMUNKÁLÁS-SZIMULÁCIÓS ELJÁRÁSOKBA

## INVOLVING OF NEW-GENERATION GRAPHICS HARDWARE INTO MACHINING SIMULATION PROCESSES

Tukora Balázs\*, Szalay Tibor\*\*

### ABSTRACT

*The way of using up of the graphics hardware has revolutionary changed in the recent years. The new-generation graphics processors are able to execute high-complexity graphical and non-graphical tasks without the contribution of the central processing unit (CPU) of the computer. The main advantages of general-purpose (i.e. non-graphical) usage appear at the execution of highly parallelized processes: the increasing of speed can reach even one or two orders of magnitude in comparison with the CPU-based solutions. As many of the mechanical simulation processes possesses this nature, the braking through of the new technology is predicted in this field as well. In this article the applicability of the new method is shown by the example of the material removal simulation of free-form objects.*

### 1. BEVEZETÉS

A számítástechnika fejlődése során mindinkább fontossá vált a képi megjelenítés minősége. A személyi számítógépek megjelenése után nem sokkal piacra kerültek az első videokártyák, amelyek a grafikus feladatok terhet átvették a központi egységtől. Ettől fogva a grafikus hardverek fejlődése töretlen. 2003-ra a grafikus kártyák számítási teljesítménye elérte a személyi számítógépek központi egységének teljesítményét, a másodpercenkénti egymilliárd lebegőpontos utasítás (1 GFLOPS) végrehajtásának képességét (dupla pontosságú számítások tekintetében), 2008-ra pedig jelentősen túl is szárnyalta azt: az Intel leggyorsabb, asztali számítógépbe szánt processzorának 70 GFLOPS-os teljesít-

ményéhez képest az ATI grafikus processzora 240 GFLOPS-ot ért el [1][2][3].

A megjelenítés minőségének növekedése magával vonta a grafikus hardverek programozhatóságának igényét is. Az ezredforduló táján találkozhattunk először a *shader* kifejezéssel. Ez kezdetben olyan programcskát jelentett, amely a grafikus hardveren futott, és a képernyőn megjelenő képpontokon végzett utólagos műveleteket (pl. fémes csillogást adva egy jármű színének). A későbbiekben a *shaderek* egyéb feladatok elvégzésére is képessé váltak, mint például a kirajzolandó objektumok térbeli transzformációjára vagy megvilágítási számításokra.

Az elmúlt néhány évben mérföldkövéhez érkezett a grafikus hardverek fejlesztése, az ún. *egységesített architektúrájú eszközök* megjelenésével. Korábban a *shaderek* dedikált hardveregységeken futottak: minden *shader*-típusnak megvolt a saját hardverterülete, amelyet az adott típusú feladat végrehajtására optimalizáltak. A megoldásnak az előnyei mellett volt egy nagy hátránya is: a különböző hardveregységek nem tudtak egymástól átvenni feladatokat, így a gyorsabbak a munkájuk elvégzése után többnyire tétlenül várakoztak a lassabbakra. Egységesített architektúra esetén a különböző *shaderek* ugyanazon hardver-területet használják, az „egységes” hardveren egymással párhuzamosan futnak, így egyensúlyozva ki a részegységek korábbi egyenetlen terhelését. A grafikus hardverből így nagy teljesítményű, általános célú számítási egység vált, ami lehetővé teszi olyan *shaderek* definiálását is, amelyek feladata már nem a képi megjelenítés javítása, hanem például fizikai jellegű szimulációs számítások elvégzése. Az új felépítésű hardver gyorsasága leginkább a nagyfokú párhuzamosítást igénylő számítások végzésénél jelentkezik –, a grafikai jellegű feladatokra optimalizált architektúrájának köszönhetően.

Az egységesített architektúrájú grafikus hardverek megjelenése óta sorra jelennek meg publikációk, amelyek a legkülönbözőbb tudományos területeken bizonyítják az új platformon végzett szimulációs számítások

\* Egyetemi adjunktus, Pécsi Tudományegyetem, Pollack Mihály Műszaki Kar

\*\* Egyetemi docens, Budapesti Műszaki és Gazdaságtudományi Egyetem, Gépészmérnöki Kar

hatékonyságát. A meteorológiai modellezéstől kezdve a molekulák dinamikai viselkedésének vizsgálatán át a kódfejtő algoritmusokig számtalan alkalmazás született az új architektúrára [4][5][6]. A teljesítménynövekedés sok esetben egészen megdöbbentő: az NVidia grafikus hardvergyártó cég honlapját böngészve rengeteg példával találkozhatunk, amelyek közt nem ritka a több százszoros, de akár több ezerszeres sebességnövekedésről tett bejelentés a hagyományos, CPU alapú számításokhoz képest [7].

A Pécsi Tudományegyetem Pollack Mihály Műszaki Karának Műszaki Informatika Tanszékén 2008-ban kezdődtek el az új technológia megismerésére és kiaknázására irányuló kutatások. Olyan területen kívántuk hasznosítani a fejlesztések eredményeit, ahol a fokozott számítási igény nagymértékben párhuzamosított végrehajtási jelleggel párosul, ám a fellelhető megoldások még a hagyományos, CPU-alapú szekvenciális módot követik.

A komplex munkadarabokat eredményező anyageltávolítás szimulációja megfelelő alkalmazási példának ígérkezett. Egyrészt erősen grafikus jellege miatt, ez ugyanis megkönnyíti a számítógépes grafikai eljárásokhoz visszanyúló programozási módszerek használatát, másrészt azért, mert az anyageltávolítást elszenvedő munkadarab geometriájának aktualizálása nagy mennyiségű adaton való gyors számítás igényel a megmunkálás során [8]. A CAM rendszerek gyártói még nem éltek az új technológia adta lehetőségekkel – valószínűleg az igen bonyolult rendszereik gyökeres átalakítással szembeni „tehetetlensége” miatt –, úgy tűnt tehát, hogy az új alapokról elindulva érdemes belevágnunk az ilyen irányú kísérletekbe.

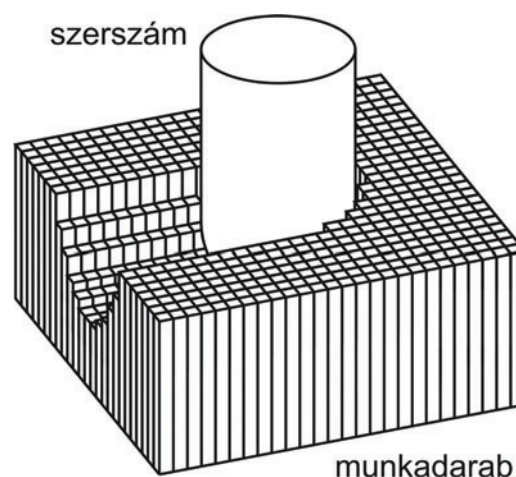
Cikkünk az anyageltávolítás-szimuláció grafikus hardver alapú megvalósítása során eddig elért eredményeinket mutatja be. Tapasztalataink szerint az új technológia jelentős mértékű javulást eredményez a vizsgált szimuláció gyorsaságában és minőségében, és joggal feltételezhető, hogy más gépészeti célú szimulációs eljárások esetén is hasonló javulás érhető el.

## 2. A MEGFELELŐ TÉRFOGAT-REPREZENTÁCIÓS ELJÁRÁS KIVÁLASZTÁSA

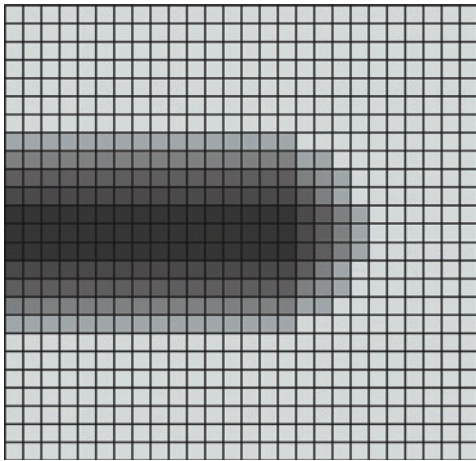
Az anyageltávolítást szimulációjára szolgáló módszerek két csoportba sorolhatók: az analitikus és a közelítő (diszkrét) eljárások közé [9]. Analitikus módszerek esetén a vágószerszám végighalad a szerszám pályán, és bizonyos időközönként a konstruktív szilárdtestgeometria (CSG, Constructive Solid Geometry) szabályai szerinti kivonás művelettel szimuláljuk az anyageltávolítást. Az eljárás legfőbb hátránya az  $O(N^4)$  számítási igény, ahol  $N$  a szerszám pályája pontjainak száma. Komplex feladatoknál ez több tízezer pontot is jelenthet, ami majdhogynem lehetetlenné teszi az alkalmazható-

ságát. Bonyolult szerszámok esetén további problémát jelent azok megfelelő analitikus leírása is. Mindezek ellenére a CAM rendszerek gyártói használják az eljárást, de legfeljebb 3+2 tengelyes megmunkálásoknál, mivel a szabad formájú felületekkel határolt testek reprezentációja gyakorlatilag nem megoldható analitikusan.

Manapság a közelítő eljárások az elterjedtebbek, kisebb erőforrás-igényüknek köszönhetően. Ebben az esetben a munkadarab és a szerszám reprezentációja diszkrét térben történik; a szimuláció minőségét a felbontás határozza meg. *Kép-alapú diszkrét eljárásról* akkor beszélünk, ha a munkadarab geometriáját digitális kép formájában írjuk le: a munkadarabot azonos keresztmetszetű párhuzamos hasábokra bontjuk a megmunkálás irányából (1. ábra), s ezek hosszát egy kétdimenziós adathalmazban, képben tároljuk. A hasáboknak a keresztmetszetükön mért oldalhosszúsága jellemzi a felbontás minőségét. Amennyiben a munkadarab és a képernyő felbontása megegyezik, a hasábok egy mélységi kép (depth image) pixeleiként ábrázolhatók, amely a számítógépes grafika egy általánosan használt formátuma (2. ábra) [10]. (A mélységi képben nincsenek színinformációk, csupán egyetlen adat képpontokként, amely általában a kamerától mért távolságot jelzi.) A vágószerszám söpört térfogatát az előzővel megegyező irányból felvett újabb mélységi képpel írjuk le. Ezek után a munkadarabból kimetszett részt a két mélységi kép összetartozó pixel-értékeinek kivonásával számítjuk. Mivel ez a művelet a grafikus hardverek által széles körben támogatott, igen gyors és egyszerűen megvalósítható eljárásról beszélhetünk.

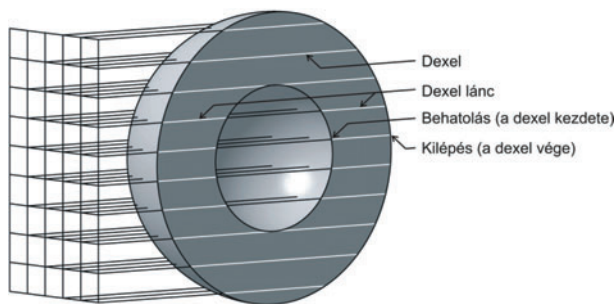


1. ábra. Kép alapú ábrázolás



2. ábra. A munkadarab mélységi képe

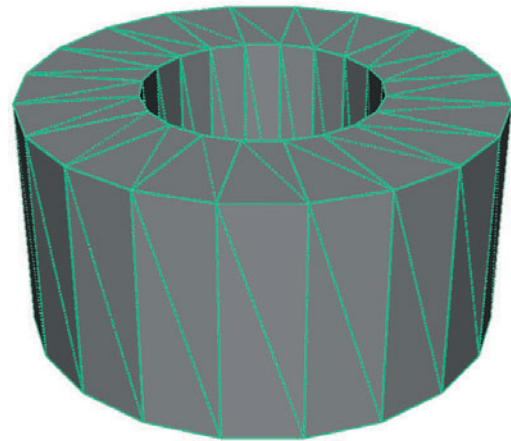
Kép alapú eljárással kizárólag olyan anyageltávolítás-szimulációk végezhetők, amelyeknél a szerszám csak egyetlen irányból végez megmunkálást –, ez pedig nem igaz a többszögletes megmunkálásra. Az így kialakított testeken ugyanis bármely nézőpontból találhatunk takart felületeket, márpedig az egymás mögötti felületek egyetlen mélységi képben nem ábrázolhatók. A probléma megoldására született meg a *dexel-bázisú eljárás* [11]. A dexel elnevezés a „depht element”, „mélységi alkotóelem” kifejezésből származik. Egy kiválasztott nézőpont felől párhuzamos egyeneseket indítunk az objektum felé. Ahol egy egyenes behatol az anyagba (a felénk néző felületen át), egy dexel kezdődik, ahol pedig elhagyja a testet (a hátsó felületen át), véget ér a dexel. A dexeleket kezdő- és végkoordinátaikkal reprezentáljuk a memóriában, az egymás mögötti dexeleket pedig láncba fűzve tároljuk, a legközelebbi felülettől elindulva egészen a leghátsóig.



3. ábra. Dexel-bázisú eljárás

A dexel-bázisú módszer, bár a kép-alapú eljárásokból fejlődött ki, a diszkrét módszerek másik csoportjához tartozik: a *térfogat-alapú (volumetrikus) eljárások* közé. Ebben az esetben a munkadarab által kitöltött tér különböző térfogatelemekeken végzett Boole-algebrai műveletekkel (dexelek vagy apró kockák, ún. *voxelek* uniójaként), bizonyos térfogategységek (pl. tér-nyolcadok) dekompozíciójával, vagy a belső részeket körülvevő héj

leírásával kerül definiálásra. Ez utóbbi eljárást BRep-nek, (Boundary Representation, magyarul „határfelület reprezentáció”) hívják. A BRep a CAD rendszerek legnépszerűbb modell-leíró eljárása, itt a felületeket NURBS-foltok összessége adja ki. CAM rendszerek virtuális megmunkáló moduljaiban is megtalálható a módszer, de a felületeket egyszerűbb módon, háromszögek halmazaként definiálják (4. ábra). Ha az anyag-megmunkálás szimuláció BRep alapú, a geometria tárolásához szükséges memóriaterület folyamatosan növekszik az idő előrehaladtával, mivel megmunkált felületet egyre több és több háromszöggel tudjuk leírni. Emellett a nagyobb mennyiségű adaton végzett műveletek száma miatt a szimuláció sebessége is egyre csökken. Ennek ellenére sok CAM rendszerben találkozhatunk a megoldással – ezekben a felbontás csökkentésével próbálják kiküszöbölni az említett problémákat.



4. ábra. Határfelület reprezentáció (BRep)

Áttekintve a fenti megoldásokat, a dexel-alapú eljárás implementálása mellett döntöttünk. Bár a módszer megfelelő a szabad formájú felületekkel takart testek leírására, az eltérő adat-reprezentációs modell miatt a hagyományos grafikus hardverek nem támogatják. Az egységesített architektúrájú grafikus egységek rugalmas adatkezelése viszont lehetővé teszi a hagyományostól eltérő struktúrájú adatelemek definiálását is. Bebizonyosodott, hogy ha a dexel-bázisú eljárás adatelemeit a grafikus egységek által megkövetelt formátumúvá alakítjuk, akkor megnyílik az út az új hardver kínálta nagyfokú párhuzamosítás kiaknázása felé.

### 3. A DEXEL-BÁZISÚ MEGMUNKÁLÁS-SZIMULÁCIÓ MEGVALÓSÍTÁSA GRAFIKUS HARDVEREN

A hagyományos, CPU-val végzett megmunkálás-szimulációnak a szekvenciális végrehajtáson kívül van egy másik hátránya is: a képernyőn való megjelenítés-

hez a grafikus hardver részére újra és újra el kell küldeni az aktualizált munkadarab geometriáját leíró teljes adathalmazt. Mivel ezt másodpercenként többször is meg kell tenni, a CPU és GPU (Graphics Processing Unit, vagyis a grafikus hardver) között folyamatosan óriási mennyiségű adat áramlik. Tudjuk, hogy az információk továbbításáért felelős adatvonalak sávszélessége véges – ez pedig jelentősen lelassíthatja az egész alkalmazás sebességét. A CPU – GPU közötti szűk keresztmetszet negatív hatásait úgy lehet kiszűrni, ha a munkadarabot leíró adatstruktúra létrehozását, tárolását és feldolgozását teljes egészében a grafikus hardver hatáskörében oldjuk meg.

A fenti folyamat első lépése az előgyártmány geometriájának dexelek formájában történő rögzítése. Szabad formájú testeket könnyedén hozhatunk létre különböző CAD rendszerekben, de ezek objektum-reprezentációja nem dixel-bázisú. Szerencsére akad egy köztes formátum, amelyet minden CAD alkalmazásból ki tudunk exportálni, dexelekké való alakítása pedig egyszerűen algoritmizálható: a gyors prototípusgyártásnál használt stl (stereolithography) fájl-típus. Egy stl fájl nem más, mint az objektum felületének háromszögekre való bontása (tesszelációja) során kapott elemi háromszögek felsorolása. Mivel a grafikus hardverek alapvetően háromszögekből felépített felületek megjelenítésére specializáltak, az stl fájlok feldolgozása nem jelent problémát. A dexelekké való alakítás módja tradicionális, de esetünkben shaderok végzik: a dexelek irányának megválasztása után egyszerre nagyszámú egyenes metszéspontja számítható a háromszögek alkotta felülettel. A párhuzamos számítások végzésére optimalizált hardver igen hatékonyan képes a feladatot elvégezni: a kutatáshoz használt középkategóriás grafikus kártya több mint 1 millió dexelt állított elő egyetlen másodperc alatt.

A dixel-bázisú anyageltávolítás-szimuláció során nagyszámú kivonás-művelet történik a dexelek és a vágószerszám között. Minden egyes szimulációs lépésben meg kell határozni a szerszám és az egyes dexelek metszéspontjait, s ennek alapján kell lerövidíteni vagy változtatlanul hagyni a dixel hosszát.

Az általunk kidolgozott módszer során shaderok végzik a kivonás-műveleteket, két lehetséges módon. Az első esetben analitikusan számolunk: a metszéspontokat vektor-egyenletek algebrai megoldásai adják, melyekben a dexelek mint térbeli egyenes szakaszok, a szerszám pedig mint térbeli felületek (gömbök és hengerek) összessége szerepel. A módszer nagyon gyors, de nyilvánvaló korlátokkal bír, mivel egy bonyolult szerszám vagy söprési térfogat analitikus leírása nagyon nehézkes. Mindazonáltal hatékonyan alkalmazható olyan egyszerű vágószerszámmal végzett szimulációkhoz, mint például a huzalelektrodás szikraforgácsolás.

A másik módszer szerint a dexelek és a szerszám héját leíró háromszög-halmaz metszéspontjait számítjuk ki a shaderok segítségével. (Mivel a szerszám kirajzolása-

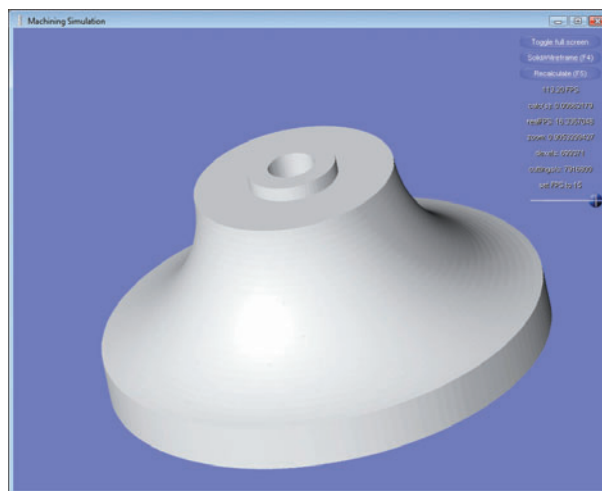
hoz eleve szükség van a felülete háromszög-alapú reprezentációjára, a szükséges adatok rendelkezésünkre állnak.) Az eljárás lehetővé teszi olyan, rendkívül bonyolult szerszámokkal végzett szimulációk elvégzését, mint a tetszőleges formájú elektródával végzett sülyesztékes szikraforgácsolás vagy fogaskerekek lefejtő marásáé.

A grafikus hardver alapú anyageltávolítás-szimuláció eljárás utolsó fázisa az aktualizált geometriájú munkadarab megjelenítése. Mivel a dexelek létrehozása és feldolgozása mellett azok tárolását is a grafikus egység végzi, nem kell igénybe vennünk a központi egység (CPU) és a grafikus hardver közötti szűk sávszélességű adatsatornát az információk eléréséhez. Ez jelentősen növeli a szimuláció sebességét, illetve az azonos sebesség mellett elérhető felbontást, a megjelenítés minőségét.

#### 4. EREDMÉNYEK

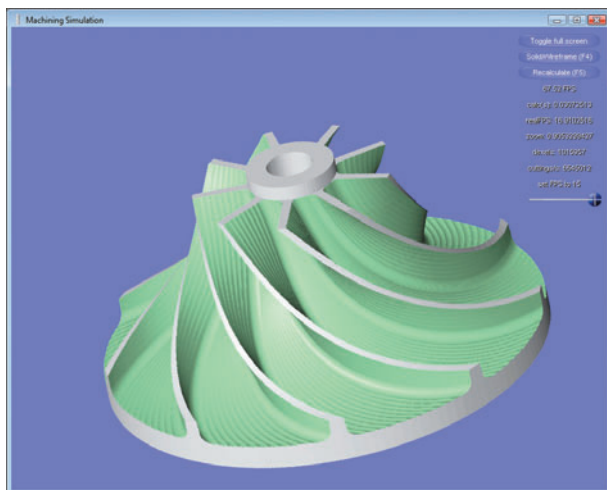
A kidolgozott eljárás demonstrálására elvégeztünk néhány jellegzetes anyageltávolítás-szimulációt az új módszer alapján. A többségük konkrét gépészeti feladatokhoz kötődik, de van közöttük tisztán bemutató jellegű is, fizikai alapok nélkül. Az alkalmazások elkészítésekor nem volt célunk a megmunkálási folyamat paramétereinek optimalizálása, sőt, néhány esetben túlzott minőségi követelményeket állítottunk fel az algoritmus hatékonyságának vizsgálatához. Az eredmények mozgókép formájában megtekinthetők a hivatkozások között feltüntetett weboldalon [12]. Az alábbiakban közülük kettőt részletezünk, a minőségre és gyorsaságra vonatkozó paraméterek felsorolásával.

Az első példában a metszéspont-számítás analitikus elven történik. Az 5. ábrán látható előgyártmány-modell kb. 700 000 dixelből áll, ami 200 milliméter átmérőjű munkadarab esetén kb. 0,2 mm-es felbontást jelent. A vágószerszám gömb és henger uniójaként definiált.



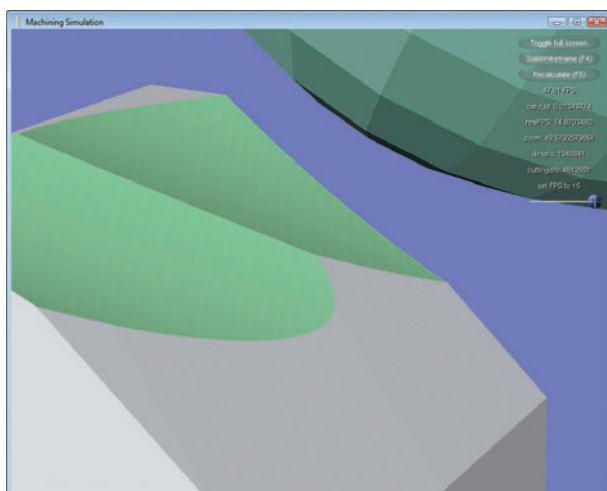
5. ábra. Az előgyártmány modellje

Kb. 210 000 szimulációs lépés után formálódott ki az 1 millió dexelből álló, durván megmunkált turbina-kerék (6. ábra). Az egy szimulációs lépésre jutó számítási idő 0,0065 másodpercről 0,0095 másodpercre nőtt a folyamat alatt, a dexelek számával egyenes arányban. Ez nagyjából 25 perc teljes megmunkálási időt jelent, ami lehetővé teszi a valós idejű nagy felbontású szimulációt.



6. ábra. A megmunkált turbina-kerék

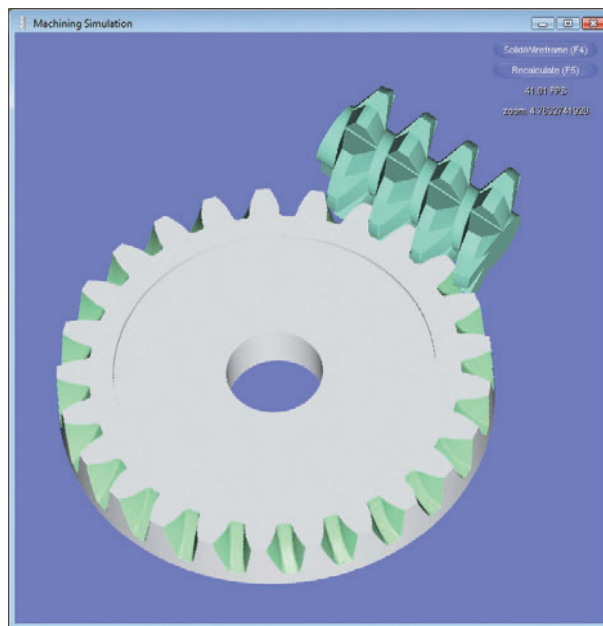
Az eljárás előnye, hogy a grafikus hardver alapú adatfeldolgozással igen gyors adatkonverziót érhetünk el, s ezáltal a munkadarab egy kisebb részlete az eredeti dexelsűrűséggel (tehát megnövelt felbontással) nagyon gyorsan újjáépíthető. A 7. ábrán 50-szeres nagyítás mellett láthatjuk a munkadarab egy részletét, kb. 0,004 mm-es felbontással. A konverzió csupán 0,65 másodpercet vett igénybe.



7. ábra. Munkadarab-részlet 50-szeres nagyítás mellett

A következő példa azt a módszert demonstrálja, amely során a dexelek és a vágószerszám metszéspont-

jaik közvetlenül a vágószerszám felületét leíró héj alapján származtatjuk, gyakorlatilag tetszőleges bonyolultságú szerszám definiálását lehetővé téve. A 8. ábrán látható lefejtő maró modellje tulajdonképpen túl is van részletezve, hiszen a szerszám egyszerűsített spirális burkolómodelljével is ugyanezt az eredményt érünk el –, a bonyolult modell csupán az eljárás hatékonyságának demonstrálására szolgál. 240 000 dexel mellett 0,02 másodperces ciklusidőt értünk el, ami valós idejű szimulációnak felel meg. Hasonló összetettségű szerszámmal mindeddig nem találkoztunk a hagyományos, CPU-alapú eljárások során.



8. ábra. Lefejtő marás szimulációja

## 5. ÖSSZEFOGLALÓ

Az egységesített architektúrájú grafikus hardverek megjelenésével a kis költségvetésű kutató vagy termelő egységekben is lehetővé vált nem grafikus jellegű, nagyfokú párhuzamosított végrehajtást igénylő algoritmusok hatékony végrehajtása. A Pécsi Tudományegyetemen folyó kutatás során az új technológia alkalmazhatóságának vizsgálatát, bevezetését tűztük ki olyan területeken, ahol mindeddig a hagyományos, szekvenciális módszerek uralkodtak. A szabad formájú munkadarabokat eredményező anyageltávolítás szimulációja esetén sikerült olyan eljárást kidolgoznunk, amellyel számottevő minőségi és teljesítménybeli javulás érhető el.

A kutatási projekt interneten publikált eredményei iránt élénk érdeklődés mutatkozik a vezető CAM rendszerfejlesztő cégek felől. Nem kétséges, hogy az olcsó, nagy teljesítményű eszközök használata néhány éven belül általános lesz a számítástechnika egészét tekintve, a játékfejlesztéstől kezdve a gépészeti szimulációs eljárás-

rásokig. Az anyageltávolítás-szimulációs algoritmusok programozását 2008 elején kezdtük el tanszékünkön –, az azóta eltelt idő alatt megduplázódott az azonos áron kapható grafikus hardverek teljesítménye. A technológia alkalmazása előbb-utóbb kikerülhetetlenné válik: a piacképesség megőrzése érdekében idehaza is mielőbb neki kell állni megszokott módszereink felülvizsgálatának, átdolgozásának.

### SUMMARY

With the appearance of the unified-architecture graphics hardware the effective execution of non-graphical, highly parallelized algorithms has become possible even in the low-budget researcher or manufacturer sections. The aim of the research work at the University of Pécs has been the examination of feasibility and applicability of the new technology in such fields where the traditional, sequential methods still rule. In case of the material simulation of free-form objects a new process has been developed, with which significant quality and performance improvement can be reached.

The publication of the project results on the internet has aroused the interest of the leader CAM developers. That is beyond question that the use of these cheap and high-performance devices will be common in the computer technology, from the game development to mechanical simulation processes. The programming of the material removal simulation started at the beginning of 2008 at our department – since that time the performance of the same-priced devices has doubled. The application of the technology is getting unavoidable sooner or later. For reserving the marketability, we have to start revising and reconstruct the traditional methods, here in Hungary as well.

### 6. IRODALOM

- [1] NVIDIA CUDA: Compute unified device architecture programming guide, NVIDIA Corporation, Version 1.1, 2007.
- [2] Intel Core i7 Performance Preview, [http://techgauge.com/article/intel\\_core\\_i7\\_performance\\_preview/9](http://techgauge.com/article/intel_core_i7_performance_preview/9), 2009.8.27.
- [3] AMD FireStream™ 9270, [http://ati.amd.com/technology/streamcomputing/product\\_firestream\\_9270.html](http://ati.amd.com/technology/streamcomputing/product_firestream_9270.html), 2009.8.27.
- [4] MICHALAKES J., VACHHARAJANI M.: GPU Acceleration of numerical weather prediction, submitted to 2008 Workshop on Large-Scale Parallel Processing (LSPP), April 2008., [www.mmm.ucar.edu/wrf/WG2/michalakes\\_lspp.pdf](http://www.mmm.ucar.edu/wrf/WG2/michalakes_lspp.pdf)
- [5] LIU W., SCHMIDT B., VOSS G., MÜLLER-WITTIG W.: Molecular dynamics simulations on commodity GPUs with CUDA, Lecture Notes in Computer Science, Vol. 4873/2007, pp. 185–196.
- [6] MANAVSKI S. A.: CUDA compatible GPU as an efficient hardware accelerator for AES cryptography, Proc. of IEEE International Conference on Signal Processing and Communication, ICSPC 2007, pp.65-68
- [7] NVIDIA CUDA Zone, [http://www.nvidia.com/object/cuda\\_home.html](http://www.nvidia.com/object/cuda_home.html), 2009.8.27.
- [8] TUKORA B., SZALAY T.: Manufacturing simulation in the light of the recent GPU architectures. Proceedings of Gépészet Konferencia (electronical issue): C20
- [9] KARUNAKARAN K.P., SHRINGI R., SINGH A.K.: Virtual machining. 2004. Industry watch 6: 62-68.
- [10] SAITO T., TAKAHASHI T.: NC machining with G-buffer method. 1991. ACM SIGGRAPH Computer Graphics 25(4): 207-216.
- [11] HOOK T. VAN.: Real time shaded NC milling display. 1986. Computer Graphics 20(4): 15-20.
- [12] Machining simulation with GPGPU, [www.youtube.com/user/BalazsTukora#grid/user/08EDE041E671ECD7](http://www.youtube.com/user/BalazsTukora#grid/user/08EDE041E671ECD7), 2010.6.1