

### 3D grafické akcelerátory

# Grafické karty pod lupou - díl 1.

**Není nijak snadné zorientovat se v tom, co se v poslední době v oblasti grafických akcelerátorů pro osobní počítače děje. Jednotliví distributoři se navzájem trumfují tím, jaké funkce a jak rychle umí jejich hardware poskytovat. Časopisy a internetové diskusní skupiny jsou plné různých, často ostře polemických názorů. Tento článek se pokouší leccos ujasnit. Pokusíme se blíže podívat na jednotlivé funkce, které nové grafické akcelerátory poskytují, a zejména na to, jak fungují a k čemu se mohou hodit.**

Nutno říci, že napsání tohoto článku trvalo pěkně dlouho. Ono prokousat se obrovským objemem dat, která jsou k dispozici, vybrat z nich informace a pokusit se je nějak roztřídit a porozumět jim nebylo snadné. Většina firem i v jinak objektivních "white papers" hodně, a obávám se, že často i záměrně, mate a snaží se představit především sebe jako průkopníky. K tomu je nutno okamžitě dodat, že firmy si své implementace a detaily střeží jako rodinné zlato a tím se situace průzkumníka na poli grafických akcelerátorů komplikuje ještě více. Dozvíte se, že firma něco implementovala hardwarově, ale už se samozřejmě nedozvíte jak, a tak se funkce dají popisovat dosti těžko. Z toho hned na úvod plyne varování: Určitě jsem se v něčem spletl a jistojistě se zde najde velké množství tvrzení, se kterými je možné polemizovat. Přesto, nebo možná právě proto, se do toho tedy pusťme (celý zpracovaný materiál se bohužel nevešel do tohoto vydání Chipu, proto některé teoretičtější části uvedeme příště).

Úvodem by bylo vhodné připomenout, že grafické akcelerátory jsou užívány pouze v rychlé počítačové grafice (pod pojmem rychlá počítačová grafika budeme v dalším textu rozumět tzv. časově kritické aplikace, time critical applications, tj. virtuální realitu, interaktivní grafiku a počítačové hry), a pro skutečně fotorealisticke zobrazení (film, video, kvalitní prezentace apod.) se používají náročnější metody globálního osvětlení (výpočet jediné scény trvá minuty i hodiny), při nichž záleží jen na síle procesoru a propustnosti systému. Fotorealisticke zobrazování není v grafickém hardwaru urychlováno v podstatě nijak, ale i tento čas zajisté jednou přijde.

Obecný trend, který je v rychlé počítačové grafice dneska zřetelný, je jednoznačně diktován standardy typu OpenGL a DirectX. Díky tomu, že jsou jejich funkce všeobecně přijaty jako standardní, poskytovatelé hardwaru se pomalu, ale jistě odhodlávají implementovat do hardwaru stále více funkcí, které dříve musel oddřít procesor. Platí samozřejmě, že jakékoli specializované zařízení poskytuje danou funkci levněji a na podstatně nižší taktovací frekvenci než zařízení obecné. Hezký příklad je k nalezení na stránkách firmy NVIDIA, kde je uvedena následující ilustrace: Pokud chcete sledovat DVD na osobním počítači, který nemá nijak akcelerovanou grafiku a umožňuje tedy pouze nastavit určitou barvu pixelu, potřebujete k tomu Pentium III na 500 MHz. Naproti tomu pokud si koupíte specializovaný DVD přehrávač, věřte, že hardware uvnitř zvládá v pohodě všechno dekódování i zobrazování na bídných 60 MHz (do tzv. GPU – graphics processing unit). Typickým příkladem tohoto trendu je již zmiňované video, další funkce, které se v současné době v GPU realizují, jsou geometrické transformace, osvětlování, textury a jejich komprese, mapování prostředí, simulace mlhy, dvojrozměrné operace, operace s barvami aj. Podívejme se na nejdůležitější z nich. Nejprve popíšeme, co je pro GPU vesměs společné, a potom se podíváme na některé karty podrobněji.

## Zobrazení 3D objektu

Základní operací, která se v rychlé 3D počítačové grafice požaduje, je zobrazení osvětleného trojúhelníku s nanesenou texturou (povrch objektů se pro zobrazení aproximuje sítí trojúhelníků – tessalation). K pochopení nejlépe poslouží zobrazené schéma OpenGL. Trojúhelník je zadán svými souřadnicemi ve 3D, které se zasílají do grafické zobrazovací jednotky. Ta nejprve v části, která se jmenuje per vertex primitive assembly, celý trojúhelník složí a transformuje do požadované polohy. V dalším okamžiku se určí barva vrcholů trojúhelníku. Ta může být buď zadána přímo, nebo se může definovat jako materiál, který je osvětlený světly. Poté se trojúhelník rasterizuje, tedy zjistí se, který pixel na obrazovce by měl být obrazem trojúhelníku pokryt. Od tohoto okamžiku se již pracuje jen s pixely, které jsou však ještě stále obohaceny o souřadnici Z (reprezentuje vzdálenost či hloubku), o souřadnice textury a o koeficient průhlednosti. Souřadnice Z se používá pro výpočet viditelnosti, ale také pro simulaci mlhy, pro mapování textur, pro interpolaci barvy z vrcholů trojúhelníku atd. To vše, stejně jako mapování textur, se děje na úrovni pixelů. Cílem tvůrců grafických čipů je samozřejmě realizovat v GPU vše, tj. celé tzv. grafické proudové zpracování instrukcí (graphics pipelining). Protože je to však neobyčejně tvrdý oříšek, po důkladné analýze se takto dnes realizují především časově nejnáročnější a nejčastěji používané operace.

Pozorný čtenář si možná všiml, že tvrdošjně hovořím o trojúhelnících, avšak OpenGL a D3D poskytují i funkce na zobrazování obecnějších polygonů. O trojúhelnících však můžeme hovořit bez ztráty obecnosti, protože prvním krokem jednotky per vertex primitive assembly je triangulace obecných polygonů.

## Geometrické transformace aneb T z T&L

Většina grafických čipů dnes integruje do jediného hardwarového bloku dvě funkce, transformace a osvětlování (transform and lighting – T&L). Podívejme se nejprve, jak funguje první část.

Každý vrchol trojúhelníku, který je do grafické jednotky zaslán, je podroben mnoha transformacím. Nejprve se musí zorientovat do pohledu virtuální kamery, kterou scénu pozorujeme, to, co není vidět, se musí odříznout (clipping), a konečně se každý vrchol musí přenést do dvourozměrného prostoru, tj. musí se promítnout na virtuální filmové políčko – tomuto kroku se říká projekce (projection) nebo promítání. Naštěstí všechny tyto transformace mohou být realizovány jednotně jako obyčejné násobení matic. Pokud reprezentujeme souřadnice vrcholu trojúhelníku jako vektor, tj. matici  $1 \times 4$ , kde čtyři souřadnice odpovídají  $[X, Y, Z, W]$ , kde  $W$  je tzv. homogenní souřadnice (více viz [1]), můžeme všechny nejčastěji používané transformace (posunutí, změna měřítka, otáčení a promítání) vyjádřit v matici  $4 \times 4$ . Právě násobení matic je operace, kterou se většina dnešních akceleratorů (chtělo by se ulehčeně oddychnout “Konečně”, protože na pracovních stanicích je něco podobného samozřejmostí více než deset let) honosí. Pokud se někde dočtete o geometrických transformacích, věřte, že je za tím zejména obyčejné násobení matic, případně skalární součin.

Další funkce, které současné grafické akcelerátory v hardwaru poskytují, se týkají osvětlování.

## Osvětlování aneb L z T&L

Geometrické modely potřebujeme nějakým způsobem zobrazit. Abychom tento krok mohli učinit, musí mít povrch modelu přiřazené určité optické vlastnosti, k čemuž se v počítačové grafice používají dva základní způsoby. První je přímé přiřazení barev určitým částem modelu – například každá strana krychle může mít jinou barvu, či koule může být natřena různě barevnými proužky. Druhou možností je určit vlastnosti materiálu, ze kterého je objekt vytvořen, a na objekt posvítit. Barva každého bodu na povrchu tělesa se pak určí výpočtem jako výsledek interakce světla s povrchem objektu, případně s objekty dalšími. Způsob výpočtu osvětlení se jmenuje osvětlovací model.

Osvětlovacích modelů existuje velké množství a jeden z nich se ujal zejména v hardwarových akceleratorech, protože je poměrně jednoduchý, snadno se vypočítává a poskytuje překvapivě dobré výsledky – Phongův osvětlovací model (což je něco jiného nežli Phongovo stínování – podrobněji viz příště). Tento empirický model je zaměřen především na rychlost a používá četná zjednodušení, díky kterým je výjimečně vhodný (a byl tak ostatně v roce 1982 navržen) pro implementaci v hardwaru. Phongův osvětlovací model používá v podstatě pouze skalární součin (dot product), tedy sčítání a násobení – opět žádná velká věda. K tomu, aby nějaká jednotka implementovala osvětlování hardwarově, potřebuje znát také polohu světla a jejich vlastnosti, polohu pozorovatele a normálový vektor (kolmý k povrchu) v místě, ve kterém se osvětlení počítá. Současné akcelerátory pro osobní počítače realizují prakticky beze zbytku osvětlování v hardwaru. Nám opět nezbývá, než si oddychnout a dodat “Konečně”, vždyť grafické pracovní stanice touto funkcí disponují již více než deset let.

## Texturování

Nejjednodušší cestou, jak modelovat ve virtuálním světě obrázek (ale také např. kůru stromu či celou fasádu budovy i s okny), je “nalepit” na objekt fotografii – texturu. Jinou možností je mapovat texturu jako prostředí a docílit tak efektů, které byly použity ve filmu Terminátor II. Textura může být rovněž promítnuta do scény jako z promítačky. Jak je zřejmé, textury hrají nesmírně důležitou roli při zvyšování realističnosti virtuálního světa a největším problémem je, že jich je zapotřebí mnoho a že musí být rychle k dispozici. Podívejme se na několik čísel, která jsem opět čerpal ze stránek [www.nvidia.com](http://www.nvidia.com).

Mějme aplikaci, která potřebuje 36 MB textur, a předpokládejme, že tyto textury jsou uloženy v hlavní paměti. S AGP 4X trvá zobrazení celého objemu dat za ideálních podmínek (s ničím se “nepřetahujeme” o sběrnici) 36 ms. Máme-li k dispozici paměť textur přímo na GPU, po prvním načtení trvá celé zobrazení asi 7 ms. Kompresi textur, která se dnes v GPU používá, má konstantní kompresní poměr 1 : 4. Tak se do paměti vejde čtyřikrát více textur a jejich zobrazení je přibližně čtyřikrát rychlejší. Je zřejmé, že komprese textur je vynikajícím prostředkem, jak zvýšit rychlost a v konečném důsledku i kvalitu zobrazování.

V komplikované scéně obsahující mnoho objektů je samozřejmě nutné mít k dispozici velké množství textur. Zároveň je třeba mít tato data k dispozici velice rychle – není možné čekat, až se přečtou z pomalé hlavní paměti, ale musejí být někde poblíž grafické jednotky. Obyčejně je tedy tato takzvaná texturovací paměť fyzicky na grafické kartě. Čím více této paměti je, tím rychlejší je zobrazování scény, protože se data nemusejí složitě a pomalu číst z paměti hlavní. Texturovací paměť má dvě zásadní nevýhody – je drahá a není jí nikdy dost. Například průhledná textura v rozlišení 256 x 256 pixelů v RGB zabere skoro čtvrt megabajtu této drahé paměti.

Firma S3 přišla s velice elegantním řešením, které je snadno dostupné i pro low-end grafické karty a které se dnes používá v různých obměnách i v kartách Voodoo a NVIDIA. S3TC používá ztrátovou kompresi textur (původní obraz je mírně pozměněn), která má konstantní kompresní poměr a komprimace i dekomprimace je velice rychlá a navíc symetrická – tj. doba komprimace i dekomprimace se nijak výrazně

neliší. Textury pak zabírají podstatně méně místa, dopředu se ví, kolik ho bude zapotřebí a jak dlouho trvá komprese a dekomprese. Tyto vlastnosti jsou samozřejmě zcela vynikající pro hardwarovou realizaci (popis viz příště).

## Display list

Jednou ze softwarových technik, které podstatně urychlují zobrazování, je paměť displeje (display list). Na grafické kartě můžeme umístit textury, které jsou žádoucí, ale na druhou stranu, pokud například potřebujeme zobrazit 100 MB geometrických dat, musíme je teoreticky pokaždé přenášet z hlavní paměti do akcelerátoru. Stejně tak pokud často používáme geometrické transformace, musejí se matice mezi sebou pokaždé násobit. To odstraňuje paměť displeje. Data, která se do akcelerátoru přemísťují, je možno seskupit pod jediné číslo a požádat grafiku o jejich zobrazení jako celku. Display list je tedy takové grafické makro. GPU může navíc provést některé výpočty, které je jinak nutné opakovat, jen jednou, a napříště využít pouze výsledku. O zrychlení zobrazování asi není pochyb.

Dnešní akcelerátory používají paměť displeje pro předpočítání transformací. Představme si, že potřebujeme nějaký objekt otočit, posunout, změnit mu měřítko a poté jej znovu otočit. Každé transformaci odpovídá jedna matice, a pokud je budeme změňovat postupně, znamená to násobení a násobení. Vše je ale možné předpočítat do jediné matice a potom násobit pouze jednou. Programátor znalý paměti displeje se o to nemusí starat. Pokud všechny transformace seskupíme pod jedinou hlavičku, bude vše spočítáno pouze jednou, a potom už stačí transformaci pouze používat. Podobným způsobem lze urychlit výpočet osvětlení a některé operace s bitmapami, především zobrazování rastrových fontů.

## Antialiasing

Antialiasing je jedním z největších reklamních taháků, kterými se některé GPU dnes honosí. Mezi základní vlastnosti počítačové grafiky patří to, že pracuje s diskrétním obrazem, který je složen z pixelů. Na rozdíl od ideálního spojitého obrazu tady dochází k tomu, že se určitým nenulovým ploškám – pixelům – přiřazuje konstantní barva obyčejně na základě hodnoty jediného bodu z reálného světa. Tomuto procesu se říká vzorkování a dělá ho například každý digitální fotoaparát. Ze spojitě funkce se odebírají vzorky a ty se poté přiřazují dvojrozměrným intervalům. Samozřejmě tím dochází k nepřesnostem a jevům, které jsou známé jako alias. Dochází k nim zejména u objektů, které jsou velikostí srovnatelné s pixelem. Tyto objekty, pokud se pohybují, jsou jednou vidět a podruhé ne, a tak dochází například k "bublání" textur či k mizení malých objektů. Jiným příkladem jsou tenké čáry, které se vlivem vzorkování mohou přerušit, nehezky vypadá podvzorkované písmo atd. Techniky, které se pokoušejí alias odstranit, se jmenují antialiasing (viz příště).

## HDTV

HDTV (high definition television) je další samozřejmostí, kterou disponují (nebo ji alespoň nějak podporují) dnešní GPU. Opět několik zajímavých čísel. Pokud chcete sledovat digitální televizi, potřebujete k tomu alespoň pět, lépe však osm a více tisíc dolarů. Snahou výrobců GPU je přinést digitální televizi a hlavně DVD do osobních počítačů, zejména kvůli ceně.

HDTV je založena na stejném principu jako MPEG2, tedy na výpočtu rozdílů mezi obrázky a na ukládání obrazu referenčního a rozdílů mezi referenčním a ostatními. Obrázky se komprimují dnes již standardní technikou diskrétní kosinové transformace DCT. Dekomprese spočívá ve výpočtu obrazu z rozdílových vektorů a aplikaci zpětné DCT. Tato operace je výpočetně velice náročná a je samozřejmě snahou výrobců hardwaru implementovat tyto funkce v GPU.

HDTV podporuje poměrně velké množství různých rozlišení a GPU by mělo umožňovat pracovat pokud možno se všemi. Základní problém, který se však při prohlížení na osobním počítači řeší, je odstranění prokládání snímků (interlacing). Televize ani video nepracují s celými snímky, ale s tzv. půlsnímky (fields). Při zobrazování se nejprve zobrazí liché řádky a v dalším kroku sudé, proto se i jednotlivé půlsnímky označují jako sudé a liché. Zobrazený pohyb je pak na obrazovce spojitý a hladší, než kdyby se zobrazovaly celé snímky najednou. Kamery umožňují filmovat přímo v půlsnímcích a televize umějí půlsnímky zobrazovat – ne však počítače. Pokud chceme zobrazovat půlsnímky na obrazovce počítače či – lépe řečeno – v okně nějakého operačního systému, musíme nejprve odstranit prokládání. K tomu se používají v podstatě dvě metody – wave a bob. První z nich jednoduše sloučí dva po sobě jdoucí půlsnímky a zobrazí je po dvojnásobně dlouhou dobu. Nevýhodou je, že pokud je ve scéně příliš mnoho pohybu, dojde k deformaci snímků, protože liché řádky se příliš liší od sudých. Oproti tomu technika bob spočívá v prostém vertikálním roztažení snímku. To je zase nevhodné pro statické objekty, kde je zřetelná "zubatost" hran. Nejzřetelnější je to samozřejmě na textu.

## NVIDIA GeForce2 GTS

Je veřejným tajemstvím, že firma NVIDIA vznikla odchodem skupiny inženýrů z firmy Silicon Graphics (dnes SGI). NVIDIA je dnes patrně největším tahounem v rychlých implementacích standardu OpenGL a

především v podpoře pracovních stanic a osobních počítačů postavených na procesorech Intel a pracujících pod Windows či pod Linuxem. Velmi dobré grafické karty za cenu, za kterou si je může dovolit obyčejný smrtelník, postavené na čipu GeForce 256, mají dnes své nástupce v čipu GeForce2 GTS.

V podstatě vše, co je výše uvedeno jako obecné vlastnosti grafických karet, GPU GeForce2 GTS poskytuje. Malinko vylepšené je HDTV, kde je implementován inteligentní přepínač způsobu odstranění pulsů, který podle dynamiky scény používá techniku wave nebo bob.

## Osvětlování na úrovni pixelu

Podstatným vylepšením v GeForce2 GTS je osvětlování na úrovni pixelu (per pixel lighting), které poskytuje jednotka zvaná NVIDIA Shading Rasterizer (NSR). V příštím Chipu se dozvíte, jak se k výpočtu osvětlení používají vrcholy (vertex) trojúhelníků a barva se interpolací aproximuje na pixely, kde není vypočítána přesně. K výpočtu osvětlení je zároveň nutný vektor, kolmý k povrchu v daném bodě – normálový vektor. Osvětlování na úrovni pixelu v podstatě využívá to, co je v OpenGL běžně k dispozici, a to definice textury jako mapy normálových vektorů. Co to znamená? Textura může být, a nejčastěji je, pouze barva zadaná v daném bodě. Jinou možností je definovat v bodě průhlednost a další možností je definovat v daném bodě normálový vektor. Pokud máme k dispozici osvětlování na úrovni pixelu, a ne interpolaci ze zadaných vrcholů, vypočítá se osvětlení přímo pro daný pixel ze zadané normály. Pokud pracujeme s osvětlením na úrovni vrcholů, dochází při malém počtu trojúhelníků k závažným vizuálním degradacím. Tomu lze sice zamezit pomocí map osvětlení (light map), kde se osvětlení vypočítá zvlášť a nanese se jako textura, s přesným osvětlením na úrovni pixelu. Tato technika je však náročná na texturovací paměť.

Jinou možností využití techniky výpočtu osvětlování na úrovni pixelu je definice tzv. hrbolaté textury (bump texture, viz příště), kde se normála jednoduše pozmění.

## ATI Radeon Graphics

Čip ATI Radeon Graphics se skládá ze dvou zajímavých jednotek – z Charisma Engine a Pixel Tapestry Architecture. První ze jmenovaných má na starosti především osvětlování a geometrické transformace, druhá – gobelínová – se stará zejména o textury.

## Charisma Engine

Po pravdě řečeno, popis nové funkce Charisma Engine – podpory pro animaci skeletonů a jejich potahování kůží – jsem pročítal dost dlouho, a ať jej zkoumám z jedné či z druhé strany, nejde o nic jiného než o aplikaci hardwarově implementovaných matic a jejich násobení, snad malinko vylepšené proti jiným čipům. O tom, že je nutné urychlit zobrazování a animaci postavíček, zejména pro počítačové hry, není asi diskusí, ale je to jen aplikace zmíněné techniky. Možná že se mýlím, prosím čtenáře o názor.

Zajímavou funkcí je interpolace polohy vrcholů. Geometrická jednotka těchto čipů umí interpolovat polohy vrcholů. Patrně jde o lineární interpolaci, která nepůsobí vizuálně příliš realisticky (zejména kvůli nespojitostem prvního řádu), ale i tak je tato funkce, zejména pro počítačové hry, bezesporu užitečná. Jinak tato jednotka samozřejmě podporuje osvětlování a transformace tak, jak je dnes běžným zvykem.

## Pixel Tapestry Architecture

Tato jednotka je patrně nejsilnější zbraní ATI. První ze zajímavých funkcí je aplikace tří textur v jediném hodinovém kroku, což může výrazně zrychlit aplikaci násobných textur (multitexturing). Výhodou implementace v ATI je zejména to, že se žádná data nemusejí opakovaně číst z frame bufferu, což znamená zrychlení.

Vynikající funkcí jsou v GPU implementované trojrozměrné textury. Ty naleznou uplatnění zejména v efektech, jako jsou výbuchy, systémy částic, zobrazování objektů jakoby vyřezaných z jednoho kusu neizotropního materiálu, ale například i při aproximaci osvětlení. Hardwarová implementace 3D textur je, pokud mám dobré informace, vůbec první právě v těchto akcelerátorech.

Pixel Tapestry Architecture poskytuje různé druhy mapování prostředí (environment mapping) a promítané textury. Mapování prostředí je aproximací ray-castingu, hodně zjednodušené globální osvětlovací techniky, a umožňuje zrcadlení zjednodušeného okolí objektu na jeho povrchu. Podrobnější popis je uveden výše, ještě více detailů naleznete v literatuře.

Texturovací jednotka disponuje stejnou funkcí jako GeForce2 GTS pro zobrazování hrbolatých textur, tedy osvětlování na úrovni pixelu. V případě ATI jsou hrbolaté textury rozděleny do tří druhů, kde první dvě jsou v podstatě stejné, klasické hrbolaté textury, a poslední je vylepšená o mapování prostředí (environment mapped bump mapping – EMBM). Umožňuje tak například simulaci pohyblivé se vodní hladiny. Na poloprůhledný obdélník na úrovni pixelu mapujeme měnící se texturu z normálových vektorů a navíc aplikujeme mapování další textury, která reprezentuje prostředí.

Další zajímavou funkcí je hardwarová implementace stínů pomocí techniky využívající pohledu ze zdroje světla pro určení stínů (shadow mapping).

Texturovací jednotka disponuje malinko vylepšenou funkcí pro výpočet mlhy (range based fog).

## 3dfx Voodoo

3dfx je zřejmě nejrozšířenější grafikou, zejména mezi příznivci počítačových her. Před několika měsíci firma vyhlásila konec podpory standardu pro hry Glide a rozhodla se podporovat profesionální OpenGL, nicméně stále udržuje kompatibilitu i s DirectX.

Mezi ne příliš podstatné, i když jistě zajímavé vlastnosti patří vylepšený algoritmus pro generování mlhy. Standardní techniky pracují s mlhou, jejíž intenzita závisí na vzdálenosti lineárně nebo exponenciálně. Implementace v kartách Voodoo pracuje s mlhou, která je definována tabulkou, což má dvě výhody – mlha je realističtější a vypočítává se rychleji. Dalším rozšířením je podpora obdélníkových textur (oproti standardním čtvercovým) a mipmapping na úrovni pixelu (oproti u ostatních výrobců používanému mipmappingu na úrovni mnohoúhelníků). A konečně, Z-buffer může v těchto kartách pracovat buď s celými čísly, což je rychlé, ale ne příliš přesné, nebo s čísly reálnými, což je daleko přesnější, i když pomalejší. Z-buffer pracující s reálnými čísly je, podle materiálů 3dfx, rozšířením, které je vlastní pouze těmto kartám.

Podívejte se, co kromě výše uvedených funkcí 3dfx nabízí dále.

## Komprese textur FXT1

Pročítat dokumentaci ke grafickým akceleratorům je někdy skoro až smutné. V okamžiku, kdy narazíte na něco, co je kvalitní a vymyslela to jiná firma, můžete si být jisti, že se spíše dočkáte popisu metody nežli jejího názvu. Další možností je, že narazíte na Orwellovský newspeak, známý z reklam. A tak se místo toho, že Voodoo podporuje standard S3TC, dočtete, že je implementováno řešení "jiných výrobců grafických karet", ale že se od něj výrazně liší. Věřte, že to, co se jmenuje CC\_MIXED, je, alespoň jak je z dokumentací obou firem patrné, přesná kopie komprese textur od S3 a ostatní režimy, například CC\_HI, jsou pouze logickým rozšířením této techniky. Rozdíl je v tom, že S3TC v základní verzi podporuje pouze 16bitové textury a 3dfx rozšiřuje kompresní algoritmus i na textury 24- a 32bitové, tedy i na průhledné. Tím se tedy "výrazně" liší. Co je zajímavé a jistě chvályhodné, je fakt, že 3dfx dává zdarma k dispozici zdrojové kódy algoritmů komprese i dekomprese. Důvodem je samozřejmě snaha vytvořit standard a přimět ostatní výrobce k implementaci jejich algoritmů. V této oblasti se patrně ještě dočkáme spousty "zábavy". Nepříjemné na celé věci je, že se výrobci nedokáží dohodnout na standardu jednom, a tak si například každý vymýšlí vlastní rozšíření OpenGL.

## T-buffer

Stejně smutnou záležitostí v oblasti matení pojmů je tzv. T-buffer. Na jedenácti stránkách dokumentu popisují autoři textu metodu, kterou v roce 1990 publikovali Haeberli a Akeley od Silicon Graphics pod názvem akumulární buffer (accumulation buffer). Rozdíl v implementaci této metody (kterou zná téměř každý, kdo se jen trochu seznámil s počítačovou grafikou) proti originálu je ten, že v 3dfx sloučili dohromady několik kroků, které jsou v původní metodě rozdělené. O co tedy jde. Princip spočívá v tom, že je k dispozici ještě jeden "jakoby frame buffer" (zde se buffery přesně podle notace OpenGL nazývají zadní a přední), který disponuje malinko jinými funkcemi nežli prostým zobrazením svého obsahu. V okamžiku, kdy do něj zapisujeme, umožňuje tento speciální buffer to, že se nová informace k původní přidá, nikoli přemaže, případně se přidá a zároveň se obsah původního bufferu sníží o nějakou hodnotu. Něco podobného by se jistě dalo implementovat pomocí alfa míchání, ale v tom případě bychom museli celý obsah paměti opakovaně číst.

Aplikací, na kterou se akumulární buffer výtečně hodí, je tzv. rozmazání pohybu (motion blur). Jednoduše namalujete několik snímků pohybujícího se objektu přes sebe a nakonec je přenesete na obrazovku – výsledkem je dojem rozmazání pohybem, podobně jako když jedoucí vlak fotografujete dlouhou expozicí. Jinou aplikací je hloubka pohledu (depth of field), kde se téhož docílí malinkým rozkomínáním kamery, která se však dívá na jediné místo. Výsledkem je pohled, který je zaostřený na cílené místo, a vše ostatní je rozmazané. Důležitou aplikací, pro kterou je možné akumulární buffer použít, je antialiasing celé scény. V tomto případě se obraz namaluje několikrát tak, že se pohne jen malinko – v rozmezí jediného pixelu. Výsledkem je v akumulárním bufferu uložená přesná hodnota pokrytí pixelu příslušnou barvou. Akumulární buffer umožňuje zobrazení měkkých stínů (soft shadows) a mnoho podobných funkcí. Společným problémem všech zmíněných metod je to, že se scéna musí zobrazovat několikrát.

## S3 SAVAGE 2000

Materiály, které jsou k této kartě k dispozici, se soustředí především na popis výkonných funkcí, které tyto GPU poskytují v dekódování videa, které je zde hodně pokročilé. Další zajímavou jednotkou je T&L, o které platí téměř beze zbytku to, co bylo napsáno na začátku článku. Podporuje násobení matic, Phongův osvětlovací model atd.

## Millenium G400 Series

Karta od firmy Matrox je poslední, o níž se v tomto článku zmíníme. Poskytuje většinu standardních

funkcí, podporuje DVD, alfa míchání, stencil buffer (což je poměrně komplikovaný, ale mocný nástroj, na jehož popis bohužel nezbyvá místo) aj. Patrně nejzajímavější funkcí je u ATI popsání mapování prostředí na hrbolaté povrchy – EMBM.

## Závěr

Díky standardům typu OpenGL a DirectX, které jsou všeobecně přijaté a akceptované, se výrobci grafických karet odhodlávají a implementují v hardwaru algoritmy, které jsou mnohdy staré i více než deset let, i techniky zcela nové. Samozřejmostí je podpora násobení matic a operací s vektory, osvětlování, masivní podpora videa a DVD, standardem se stávají texturovací jednotky s poměrně pokročilou kompresí založenou na S3TC a stále více se setkáváme s přesnými výpočty na úrovni pixelů místo výpočtu pro vrcholy trojúhelníků a jejich aproximace lineární interpolací.

Co je tragédií, je zmatek, který producenti GPU vyvolávají přejmenováním existujících názvů a reklamními kampaněmi, které představují skvělé funkce jako skvělejší, než ve skutečnosti jsou. To je však holé konstatování. Změnit to asi půjde těžko.

*Bedřich Beneš  
beda@campus.ccm.itesm.mx*