

VYUŽITIE DIGITÁLNYCH ÚDAJOV O ÚZEMÍ PRE SIMULÁCIU BOJOVEJ ČINNOSTI

por. Ing. Marek SOPKO , plk. Ing. Ľubomír GALBA

Vojenská akadémia v Liptovskom Mikuláši, Fakulta zabezpečenia velenia,
Katedra informatiky a výpočtovej techniky

1. ÚVOD

Rozvoj počítačových technológií a rýchly vývoj grafických systémov umožňuje **realizovať čoraz väčší počet úloh, zameraných na virtualizáciu** činností prostredníctvom dostupného počítačového vybavenia. V súčasnosti je možné realizovať komplexné grafické výstupy, ktoré v dostatočnej miere **napodobujú realitu okolitého prostredia**. Vzniká množstvo **simulátorov, trénažérov a výcvikových pomôcok**, ktoré vytvárajú predpoklady na výrazné zvýšenie kvality výcviku a vzdelávania vojakov i vedeckovýskumnej práce v rezorte MO SR.

Pracovníci Katedry informatiky a výpočtovej techniky Vojenskej akadémie v Liptovskom Mikuláši sa dlhodobo zaoberajú problematikou modelovania a simulácie bojovej činnosti. To znamená, že vo svojej práci používajú aj nástroje na trojrozmerné zobrazovanie scény bojiska. V príspevku je opísaná časť problematiky, ktorá sa zameriava na zobrazovanie bojiska pri simulácii bojovej činnosti. **Zakladá sa na poznatkoch počítačovej grafiky**, simulácie modelovania virtuálneho sveta, **na výstavbe terénu** a jeho objektov, na základe podrobných údajov získaných meraním, syntézou a výskumom. Obsahuje možnosti využitia vybraných metód počítačovej grafiky pri trojrozmernom modelovaní okolitého prostredia, v ktorom môže byť vedená bojová činnosť.

2. DIGITÁLNE INFORMÁCIE O TERÉNE

2.1. Terén

Terén je časť zemského povrchu so všetkými jeho nerovnosťami vytvorenými prírodnými silami, alebo umelo, so všetkými objektmi a javmi, ktoré sa na zemskom povrchu nachádzajú.

Z hľadiska vojenskej topografie, potreby štúdia a hodnotenia terénu, rozdeľujeme terén podľa charakteru jeho základných prvkov na dve základné zložky. Konkrétne sú to **terénne tvary** (vypuklé, vyhlbené alebo rovné časti zemského povrchu rôznych rozmerov, ktoré charakterizujú členitosť terénu, súhrn týchto terénnych tvarov sa nazýva **reliéf terénu**) a **terénne predmety** (všetky predmety, ktoré sa nachádzajú na fyzickom zemskom povrchu (lesy, rieky, močiare..., sídla, komunikácie, kanále...) a charakterizujú jeho **pokrytosť**).

2.2. Digitálny model terénu

Digitálnym modelom terénu sa väčšinou rozumie priestorový geometrický popis reliéfu terénu. Na tomto reliéfe možno ďalej modelovať a popisovať najrôznejšie informácie, ako napr. pri topografických mapách. Možno definovať umiestnenie prírodných i umelých objektov, hranice správnych celkov, hranice povodia... Tieto informácie sú však z pohľadu klasického poňatia digitálneho modelu terénu menej dôležité a hlavne svojím charakterom veľmi komplikujú orientáciu používaných systémov na čistou priestorovú geometriu reliéfu terénu. Tá má totiž veľmi charakteristické prvky, ktoré predurčujú väčšinu súčasných programových systémov k tomu, aby spracovávali iba geometriu reliéfu. *Charakteristické prvky sú predovšetkým tieto:*

- Terénna plocha je veľmi nepravidelná. Vykazuje miesta, kde je priebeh veľmi hladký, inde sú línie, na ktorých je hladkosť narušená. Dokonca sa možno stretnúť s terénymi stupňami, ktoré sú síce väčšinou umelé, avšak k terénu patria. Zvláštny charakter majú také vrcholy, sedlá, ktoré majú často pozdĺžny, hladký priebeh, avšak v kolmom smere sa na nich terénna plocha môže ostro lámať. Tieto javy sa v terminológii DMT nazývajú singularity, ktorých matematickou charakteristikou je nespojitosť funkcie či nespojitosť jej derivácie,

- Modelovaná plocha môže byť veľmi rozsiahla, popisovaná značným počtom dát. Na druhej strane vzhľadom k rozsiahlosti väčšinou dosahuje malých prevýšení, rozmery v smere osi x a y sú väčšie než v smere osi z .
- Valnú väčšinu terénnej plochy možno charakterizovať ako funkciu polohopisných súradníc x , y . Tým je totiž možno vždy priradiť iba jednu výškovú zložku z . Preto sa často prostriedky pre DMT používajú i pre modelovanie a zobrazovanie exaktných matematických funkcií dvoch premenných (niekedy sa uvádza správnejší termín funkcie troch premenných, teda dvoch nezávislých a jednej závislej premennej). Výnimkou môžu byť terénne stupne (zlomy alebo tiež schody), v ktorých je terénna plocha zvislá, niekedy dosahuje charakteru prievisu. Tzv. prievisy sú miesta, ktorými možno viesť kolmicu, pretínajúcu povrch v dvoch alebo viacerých bodoch. Takéto miesta sa vyskytujú veľmi zriedka a pre potreby modelovania nemajú veľký význam.

2.2.1 Princíp delenia plochy:

Konštrukcia **nepравidelnej trojuholníkovej siete** (TIN) je zásadným nástrojom pri vytváraní polyedrického alebo plátového modelu, ktorý používa väčšina systémov DMT. Vstupnými údajmi je množina bodov P_1, P_2, \dots, P_n , ktoré sú dané svojimi priestorovými súradnicami $[x, y, z]$. Vrcholy je nutné pospájať hranami tak, aby vznikla množina trojuholníkov, ktoré k sebe priliehajú a nepretínajú sa. Tieto trojuholníky sa musia maximálnou mierou približovať k aproximovanej ploche reliéfu terénu. Zásadnou charakteristikou algoritmu pre tvorbu siete je teda, ktoré vlastnosti budú určujúce pri optimalizácii maximálneho približovania sa siete k aproximovanej ploche a akou mierou túto inak značne časovo náročnú činnosť urýchliť. Vstupom je vždy množina bodov

$$P_1(x_1, y_1, z_1), P_2(x_2, y_2, z_2), P_3(x_3, y_3, z_3), \dots, P_n(x_n, y_n, z_n)$$

Výstupom z triangulačného algoritmu je množina úsečiek, spájajúcich tieto body tak, aby vznikla vhodná trojuholníková sieť. *Nasledujúci príklad ukazuje niektoré základné typy algoritmov tvorby siete.*

1. vytvor všetky možné spojnice bodov z množiny P a daj ich do množiny K ,
2. porovnaj spojnice v množine K podľa dĺžky od najkratšej po najdlhšiu,
3. vezmi najkratšiu spojnicu z množiny K a premiestni ju do množiny M ,

4. pokiaľ v množine K nezostane žiadna spojnica, algoritmus skončí a v množine M je hľadaná trojuholníková sieť,
5. vezmi najkratšiu spojnicu z množiny K a vykonaj:
6. pokiaľ pretína táto spojnica niektorú zostávajúcu spojnicu v množine M, tak:
7. zruš ju,
8. inak:
9. premiestni ju do množiny M,
10. 6. pokračuj krokom 4.

Vyššie uvedený algoritmus je pre názornosť značne zjednodušený a predovšetkým neobsahuje jednu dôležitú zložku – možnosť včleniť tzv. predurčené hrany. Predurčené alebo tiež preddefinované či pevné hrany sú také spojnice bodov, pomocou ktorých si užívateľ vyžiada už pri vstupe spojenie vybraných bodov, ktoré považuje za definitívne.

Sú to väčšinou singularity a podobné charakteristické miesta terénu, na ktorých by algoritmus mohol vytvoriť trianguláciu inak, ako požaduje užívateľ.

2.2.2 Spôsobu reprezentácie povrchu terénu:

Vizualizáciou (alebo sa používa termín "rendering") sa všeobecne označujú techniky, ktoré prispievajú k vytvoreniu fotorealistického zobrazenia priestorových objektov. Výpočet zabezpečuje prevod súradníc priestorových bodov do roviny. Iba touto funkciou bez ďalších nástrojov možno všeobecne vytvoriť tzv. drôtové zobrazenie.

Aby bolo zobrazenie maximálne vierohodné, je nutné použiť ďalšie nástroje pri vykresľovaní. *Týmito sú predovšetkým:*

- riešenie viditeľnosti,
- priradenie farieb,
- osvetlenie a tieňovanie scény,
- pridanie povrchových textúr plochám.

Takto získaný model v konečnej podobe možno použiť pre ďalšie účely napr. prezentáciu urbanistického celku, zástavbu obytnej štvrti, budúci návrh mestskej komunikácie, zobrazenie bojiska a podobne.

2.3. Reliéf

Reliéf je kontaktnou plochou medzi komponentmi krajinej sféry (medzi atmosférou resp. hydrosférou a litosférou resp. pedosférou). Jeho priebeh je výslednicou rozlične pôsobiacich síl, pričom reliéf zároveň výrazne ovplyvňuje mnohé procesy a javy v krajine ako aj aktivity ľudskej spoločnosti.

2.4. Digitálny model reliéfu

Prvým digitálnym modelom reliéfu bol digitálny model reliéfu prvej generácie (DMR-1). Jeho elementom je charakteristická výška štvorca so stranou 1km. Poloha každého štvorca je určená kódom jeho juhozápadného rohu. Sieť štvorcov zodpovedá sieti pravouhlých súradníc systému S-1942/83.

Digitálny model reliéfu Slovenska s rozlíšením 100 metrov (DMR100-SK) **bol vytvorený** v spolupráci Geografického ústavu SAV a Výskumného ústavu pôdnej úrodnosti v rámci projektu PHARE MERA Land Degradation **v roku 1996.**

Tento údajový súbor bol **spracovávaný v prostredí GIS (GIS GRASS)** s mapovým výstupom v mierke **1:500 000**. Ako vstup pre výpočet DMR boli použité vstupné údaje z dvoch zdrojov - neúplné výškové bodové pole v tvare pravidelnej siete s horizontálnym rozlíšením 100 metrov pochádzajúce z Vojenského topografického ústavu v Dobruške (3.8 mil. bodov pokrývajúcich cca 3/5 územia SR) a bodové pole vytvorené z digitalizovaných vrstevníc zakúpených na GKU Bratislava (59 mapových listov 1:50 000). Súčasťou DMR100-SK (4-250×2-100p) sú gridy nadmorských výšok (m), sklonu a orientácie reliéfu voči svetovým stranám (obe charakteristiky v stupňoch).

Digitálny model reliéfu prešiel svojím vývojom, až bol vytvorený DMR-2 využívaný v súčasnej dobe. DMR-2 obsahuje informácie o terénnych tvaroch, nadmorské výšky uzlových bodov – priesečníkov štvorcovej súradnicovej siete, rovnobežnej s pravouhlým zobrazovacím systémom S-1942/83 o rozmeroch štvorcov 100×100 m. Presnosť výšok odčítaných uzlových bodoch je kontrolnými meraniami stanovená na hodnotu ±3 m až ±8 m vo väčšine územia. Vo vysokohorskom teréne je presnosť nižšia a môže dosiahnuť hodnotu až ±12 m. Vo všetkých prípadoch presnosť definície výšok závisí výrazne na členitosti terénu a jeho zalesnení. Je organizovaný do súborov, obsahujúcich 10 000 16-bitových slov, binárne vyjadrujúcich nadmorskú výšku, t.j. celočíselné hodnoty výšok v štvorci 10×10 km. Pre územie SR je týchto súborov 617, t.j. 12MB dát.

2.5. Digitálny model územia DMÚ-200

Digitálny model územia 200 (DMÚ-200) je komplex údajov a programových prostriedkov pre zber, redakciu a distribúciu digitálnych informácií o území. Rozlišovacia úroveň a obsah zodpovedá topografickej mape mierky 1:200 000.

DMÚ-200 poskytuje **dva druhy finálnych produktov**. Prvým je **programové vybavenie** umožňujúce vlastné zhotovovanie, úpravu a jednoduchý výstup grafickej aj sémantickej zložky užívateľských údajov. Ako užívateľsky nešpecifická aplikácia je ponúkaná nadstavba umožňujúca kombináciu polohopisných údajov DMÚ-200 s výškopisnými údajmi digitálneho modelu reliéfu DMR-2. Umožňuje trojrozmerné znázornenie územia, identifikáciu objektov, riešenie viditeľnosti, rezov, profilov, generovanie vrstevníc.

Druhým produktom sú **topografické údaje**, ktoré možno používať vo vyššie uvedenom programovom prostredí, ale i mimo neho a obsahujú informácie o terénnych predmetoch. Možno ho využívať i po častiach členených územne, podľa druhov objektov, alebo oddelene len grafickú alebo sémantickú zložku.

Vlastnosti DMÚ-200:

- objektovo orientovaný (realita je rozčlenená na **topografické objekty (TO)**, ktoré sa môžu deliť na úseky),
- kódovanie informácií (geometrické, sémantické) je robené jednotným predpisom – **číselníkom TO** (7 vrstiev), je to otvorený systém kódovania, čo umožňuje zavedenie ďalších užívateľských typov a atribútov,
- pre rozširovanie číselníku TO je nutné rešpektovať filozofiu jeho výstavby pre umožnenie ďalšej údajovej komunikácie.

Topografický objekt (TO) je objekt reality zodpovedajúci položkám zoznamu mapových značiek Topo-4-3 pre mierku 1:200 000. Pre jednotlivé položky je konkretizovaný v číselníku TO. TO je obsahovo určený sémantickým typom, určujúcim druh objektu. Jeho vlastnosti sú popísané premenným počtom atribútov objektu (mená, označenie, charakteristiky – do 30 znakov). TO je polohovo určený usporiadanou množinou súradníc x a y , svojou definičnou bodovou množinou (DBM). Súradnica z môže byť uvedená ako atribút. Geometrický typ DBM môže byť bod, línia alebo areál.

Pre zachovanie informačného obsahu údajov je pre výmenu údajov navrhnutý sekvenčný znakový súbor nazvaný **Vstupno-výstupné rozhranie (VVR)**. V číselníku TO

sú registrované nasledovné sémantické typy objektov: reliéf, vodstvo, sídla, komunikácie, vedenia, hranice, pôdny a rastlinný kryt.

Základné údaje sú teda rozčlenené do 7 vrstiev, organizované ako body, línie a oblasti s viac či menej podrobným popisom. Uložené sú vo forme textových súborov pre oblasti mapových listov mierky 1:100 000. Pre územie SR je DMÚ-200 tvorené 444 textovými súbormi, t.j. 18MB dát.

3. OPEN_GL

OpenGL je **softwarová nadstavba grafických hardwarových zariadení** postavených na rýchlych, ale i menej výkonných grafických čipoch. Rozhranie OpenGL je postavené **formou grafickej knižnice**, ktorá plní funkciu rozhrania medzi používanými grafickými zariadeniami s ich programovým vybavením teda s ich grafickými ovládačmi a aplikáciami používanými v súčasných aplikáciách bežiacich na systémoch UNIX, Mac a Windows. Štandard OpenGL je plne otvorený čo vyplýva i zo samotného názvu a znamená to, že všetky jeho funkcie a procedúry sú plne dostupné, zdokumentované, upravovateľné a doplniteľné zo strany používateľa. Toto rozhranie sa ľahko používa a implementuje do danej aplikácie pričom výrazne zvýši jej výkonnosť a funkčnosť bez nutnosti namáhavej práce a zdĺhavého ladenia programu.

Počítačová grafika (špeciálne 3D a interaktívna 3D grafika) sa čoraz viac stáva súčasťou aplikácií nielen na špeciálnych modelovacích a vizualizačných superpočítačoch, ale mnohokrát je súčasťou aj menších a jednoduchších grafických programov „bežiacich“ na **klasických osobných počítačoch triedy SOHO**.

Pre splnenie požiadaviek rýchleho a výkonného behu grafických aplikácií na týchto rozdielnych platformách a pre odstránenie potreby vytvoriť osobitný grafický ovládač s príslušným prístupom k použitému grafickému zariadeniu bolo potrebné vyvinúť **rozhranie pre štandardnú interpretáciu požiadaviek používateľov**. *Takýto ovládací jazyk musí spĺňať niekoľko kritérií a to:*

1. musí byť **implementovateľný** na viacerých platformách bez obmedzenia výkonnosti alebo iných funkčných parametrov,
2. musí byť **jednoduchý, zrozumiteľný** a ľahko pochopiteľný programátorom,
3. musí sa dať ľahko **rozšíriť** bez výraznej zmeny či dopadu na originálne jadro už existujúceho systému.

Tieto požiadavky **spĺňa práve rozhranie OpenGL** a to tým, že je to jednoduchý, priamo operujúci systém vytvárajúci a upravujúci reálne 3D scény. **Podporuje vykresľovanie základných grafických primitív** (bod, čiara, mnohouholník, obraz...) a **základné grafické operácie** (podobnosť, projekcia, osvetlenie, textúry, anti-aliasing...).

3.1. Výhody a nevýhody použitia OpenGL

Jednotlivé výhody použitia rozhrania OpenGL uvádza nasledovný zoznam:

1. je využívané viac ako 5 rokov z čoho vyplýva stabilita a úspešnosť,
2. je vyvinutých viac ako 1000 aplikácií, ktoré ho priamo či nepriamo používajú,
3. je skutočným otvoreným štandardom a teda je použiteľné pre PC, UNIX aj MAC,
4. je úplne škálované a prispôsobené pre všetky podporované platformy, pre akcelerované i neakcelerované PC, je optimalizované aj pre prostredie Infinite Reality SGI Workstations,
5. umožňuje vytvárať obraz výnimočnej kvality pri zachovaní rýchlosti údajového obrazového toku,
6. je dominantným rozhraním pre siete ako také,
7. definuje užívateľsky priaznivý pomer kvality a rýchlosti,
8. má priame previazanie s najkvalitnejšími programovacími jazykmi ako C, C++, Delphi, Java...

Nevýhody používania OpenGL **nie sú pravými nevýhodami**, skôr prispôbením a to hlavne preto, lebo samotné rozhranie bolo vyvíjané iným postupom ako iné rozhrania na urýchľovanie grafických operácií. Je vhodné ukázať odlišnosti postupov. Určitou požiadavkou obvyčajného programátora je rýchlosť zobrazovania 3D scén. To sa dá jednoducho dosiahnuť vypustením zdanlivo nepotrebných prvkov. Tie však dodávajú scéne krásu. Realizuje sa to napríklad vypustením anti-aliasingu, používaním nepravých osvetľovacích modelov či zjednodušovaním tvaru objektov.

Oproti tomu **OpenGL bolo vyvíjané pre zobrazenie všetkých možných prvkov scény vo vysokej kvalite** a toto sa nedalo dosiahnuť bez mierneho zvýšenia nárokov na HW počítača, čo sa zdá byť hlavnou nevýhodou.

Nadá sa povedať, že OpenGL nájde uplatnenie len na kvalitných pracovných staniciach. Je a **dlho bude najlepšou alternatívou pre jednoduché PC platformy a bude zlepšovať ich vlastnosti za predpokladu, že ho daný HW bude môcť využiť**.

4. LOD

Niektoré graficky náročné scény sú stváňované pomocou metódy **Level Of Detail**. Princíp metódy LOD spočíva v detailnom prekreslení objektov až v momente, kedy sa k nim vo virtuálnom prostredí skutočne priblížime.

Skratka LOD (stupeň detailu) znamená prevrat v klasickom poňatí modelov priestorových objektov. Až doposiaľ sme vychádzali z predstavy, že model akéhokolvek telesa by sa mal svojimi vlastnosťami a tvarom čo najviac blížiť skutočnému objektu. Je pravda, že pri zobrazení takéhoto objektu zblízka by mali byť vidieť všetky jeho detaily, jemné prehnutia povrchu, vzory textúr, škálu farebných odtieňov vzniknutá odrazom svetla a podobne.

Potrebuje však mať takýto objekt vymodelovaný so všetkými jeho podrobnosťami i vtedy, keď sme od neho vzdialený desiatky či stovky metrov? V takej vzdialenosti nie sme predsa schopní tieto detaily rozpoznať. Navyiac od určitej vzdialenosti vnímame objekty iba ako farebné škvrny, nič viac. Ak domyslíme tieto skutočnosti do dôsledkov, dôjdeme k záveru, že z hľadiska rýchlosti spracovania virtuálneho sveta by bolo výhodné mať pre každý objekt v zásobe **niekoľko modelov (reprezentácií)**, ktoré by boli zobrazované podľa toho, ako ďaleko sú v danej chvíli od pozorovateľa, resp. ako dôležité sú ich detaily v závislosti na vzdialenosti a tým i veľkosti ich obrazu na obrazovke. Práve takýto spôsob modelovania objektov zaisťuje metóda LOD.

Navonok sa LOD objekt tvári ako jediný objekt. Vnútri však skrýva niekoľko potomkov – reprezentácií toho istého objektu v rôznych stupňoch presnosti. Potomkovia sa zapisujú do parametru *level* postupne – od najdokonalejšie vymodelovanej reprezentácie až po najjednoduchšiu. Akonáhle vzdialenosť prerastie istú úroveň, môže byť zobrazovanie objektu dokonca úplne potlačené. Medzné vzdialenosti, pri ktorých dochádza k prepnutiu medzi dvoma reprezentáciami sa ukladajú do parametru *range*. Vzťažný bod objektu, voči ktorému sa meria aktuálna vzdialenosť k pozorovateľovi, sa zapisuje do parametru *center*.

Šikovným riešením je ponechať parameter *range* prázdny. Prehliadač v takomto prípade nie je odkázaný na predpísané vzdialenosti, v ktorých prepína medzi presnosťou modelu, ale sám volí tú reprezentáciu, ktorú je schopný na konkrétnom počítači vykresliť

v dostatočne krátkom čase. Zdalo by sa, že najlepšie je vždy ponechať záznam *range* prázdny, nech sa prehliadač rozhoduje automaticky. To by však na pomalých počítačoch mohlo znamenať, že návštevník virtuálneho sveta nikdy neuvidí objekty so všetkými ich detailmi.

Preto sa v praxi volí kombinovaný prístup – reprezentácie sú rozdelené napríklad do dvoch skupín obklopených ďalšími objektmi. Hlavný objekt definuje jedinou prepínicu vzdialenosť, čím zaručí povinné prepnutie do jednej z dvoch tried presnosti modelu. Pre objekt nachádzajúci sa pred touto hranicou sa vyberie jeden z “lepších” modelov, a to v závislosti od výkonu počítača, nakoľko podobjekt má prázdny parameter *range*. Podobne pre objekt nachádzajúci sa vo väčšej vzdialenosti sa vyberá z “horších” modelov.

LOD je základným prostriedkom pre docielenie vysokej rýchlosti zobrazovania sveta. Za pomoci tejto metódy možno docieľiť vyváženú zložitosť spracovania virtuálneho prostredia. Zjednodušene možno povedať, že LOD sa postará o pevný počet plôšok, ktoré sú posielané do zobrazovacích algoritmov. Blízke telesá sú reprezentované stovkami až tisíckami plôšok. Akonáhle sa od nich vzdialime, klesne počet plôšok na ich povrchu na desiatky. V rámci daného sveta a danej veľkosti obrazovky potom skutočne zostáva počet plôšok zhruba rovnaký v ktoromkoľvek mieste virtuálneho priestoru. Ľahko tak dokážeme vypočítať, ako výkonný musí byť grafický akcelerátor nášho počítača, aby pohyb vo virtuálnej realite bol dostatočne plynulý. Ak je virtuálny svet efektívne skonštruovaný, nie je potrebné neustále zvyšovať výkonnosť grafického procesoru. Dnes už dosiahnuteľnú hranicu niekoľko miliónov textúrovaných polygónov za sekundu nie je potrebné zásadným spôsobom prekračovať.

LOD by sme teda mali používať čo najčastejšie, i keď tým zvyšujeme celkovú pamäťovú náročnosť. Otvorenou otázkou zostáva, akým spôsobom vygenerovať sadu postupne sa zjednodušujúcich reprezentácií objektu.

5. OCTREE

Ďalším zo známych prístupov k reprezentácii priestorových objektov je použitie tzv. **oktantového stromu**. Je definovaná špeciálna dátová štruktúra vo forme oktantového stromu (maximálne 8 potomkov uzla). Objekt je takto uzatvorený do kocky. Zápis objektu potom pozostáva z rekurzívneho delenia kocky na 8 elementárnych kociek, z ktorých každá môže nadobúdať jeden z troch možných stavov. Objem kocky je plný (Full), prázdny (Void) alebo zmiešaný (Mixed, Partial). Oktantový strom je ďalej

charakterizovaný hodnotou maximálnej a aktuálnej hĺbky. Hodnota **maximálnej hĺbky** charakterizuje maximálnu jemnosť, ktorú možno dosiahnuť pri použití daného oktantového stromu. Hodnota **aktuálnej hĺbky** definuje jemnosť modelu definovaného daným oktantovým stromom. Objekty zapísané pomocou oktantového stromu sú vo všeobecnosti ťažko transformovateľné.

Oktantový strom rozdelí scénu postupne na oktanty, tento strom má v príslušných uzloch či listoch len tie telesá, ktoré do neho zasahujú. Pri hľadani priesečníka sa prechádza scénou na základe tohto stromu. Prázdny uzly či listy sa len prechádza a dôkladnejšie sú testované len neprázdne uzly či listy (oktanty). Použitie tohto spôsobu pri väčšom počte telies spôsobí zrýchlenie metódy o jeden až dva rády.

Vstupom pre vytváranie OCT stromu je sieť trojuholníkov. V oblasti GIS sa pre modelovanie terénu používajú dve dátové štruktúry – nepravidelné trojuholníkové siete (TIN) a pravidelný raster (Grid). Prvá z nich je založená na princípe delenia priestoru pomocou nepravidelnej trojuholníkovej siete, pričom jednotlivé trojuholníky interpolujú povrch. V prípade formátu Grid je terén interpolovaný bodovým poľom. Pre OCT stromy je však potrebné terén reprezentovať pravidelnou sieťou trojuholníkov. OCT stromy sú významne použiteľné pre plnú integráciu modelu terénu a podporu trojrozmerných tvarov, ktoré umožňujú konštruovať i všeobecnejšie trojrozmerné telesá.

6. BSP

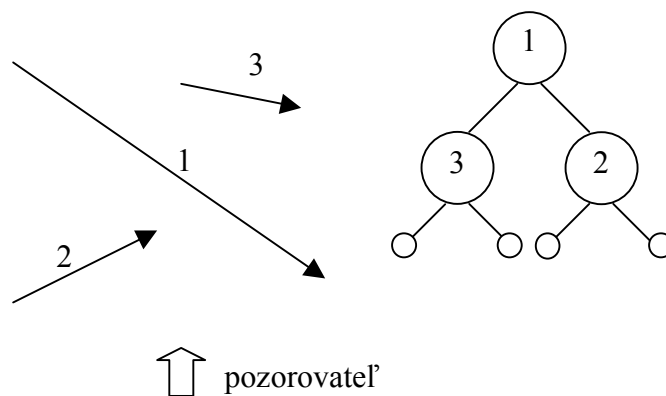
BSP (Binary Space Partitioning) stromy sa používajú obdobne ako oktantové stromy, ale deliaca rovina je vždy len jedna a ňou sa delí priestor scény vždy len na dva polpriestory. Takto vytvorená štruktúra je pamäťovo nenáročná a pomerne rýchlo sa s ňou pracuje.

BSP je metóda hierarchického rozkladu priestoru na **konvexné podpriestory**. BSP stromy sú štruktúry veľmi vhodné na použitie v počítačovej grafike, predovšetkým na rýchle triedenie, vyradovanie odvrátených – neviditeľných plôch a raytracing. Tvorba BSP stromu je výpočtovo náročný proces a spočíva v delení podpriestoru vždy na dve časti, ktoré sa následne rekurzívne delia. Celá štruktúra vzniká v čase predspracovania a pri renderingu v reálnom čase sa (až na výnimky) nemení. Použitie BSP stromu výrazne urýchľuje riešenie viditeľnosti. Podľa toho koľko stupňov voľnosti pri vizualizácii poskytneme (6 alebo menej) v nepriamej úmere ku časovej náročnosti je možné použiť 3D-

BSP stromy (6 stupňov voľnosti t.j. neobmedzený pohyb pozorovateľa) alebo 2D-BSP stromy (menej ako 6 stupňov).

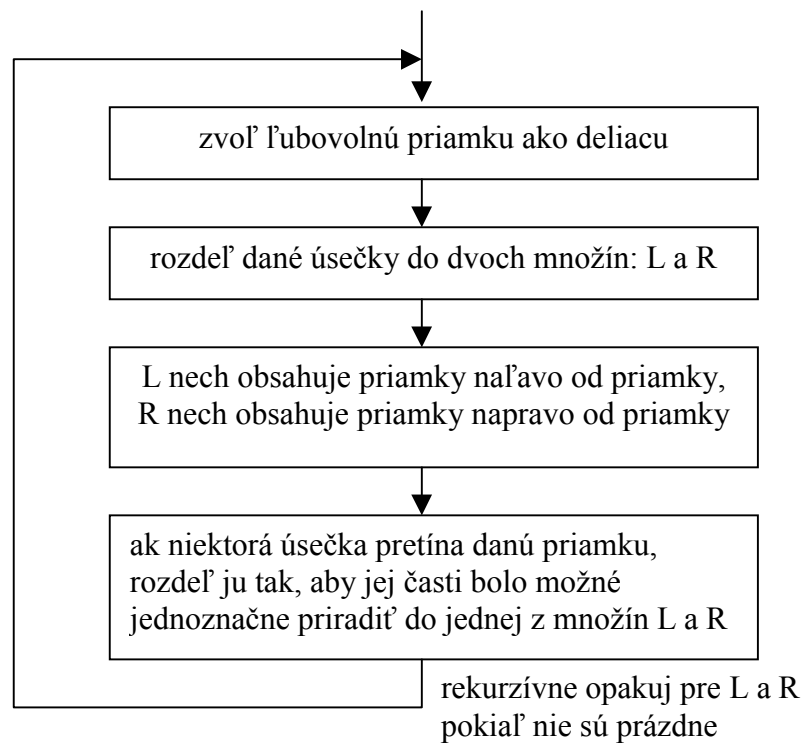
6.1. Tvorba a prechod BSP stromu

Konstruktciu BSP stromu je možné vysvetliť na nasledujúcom príklade riešenia viditeľnosti v rovine (2D-BSP). Majme tri orientované úsečky označené 1, 2 a 3 (vid' nasledujúci obrázok). Šípkou je naznačená poloha a smer pohľadu pozorovateľa na scénu (použitím maliarovho algoritmu by sa úsečky vykreslili zozadu dopredu v poradí 3,1,2).



Obr. 1 2D scéna a jej BSP strom

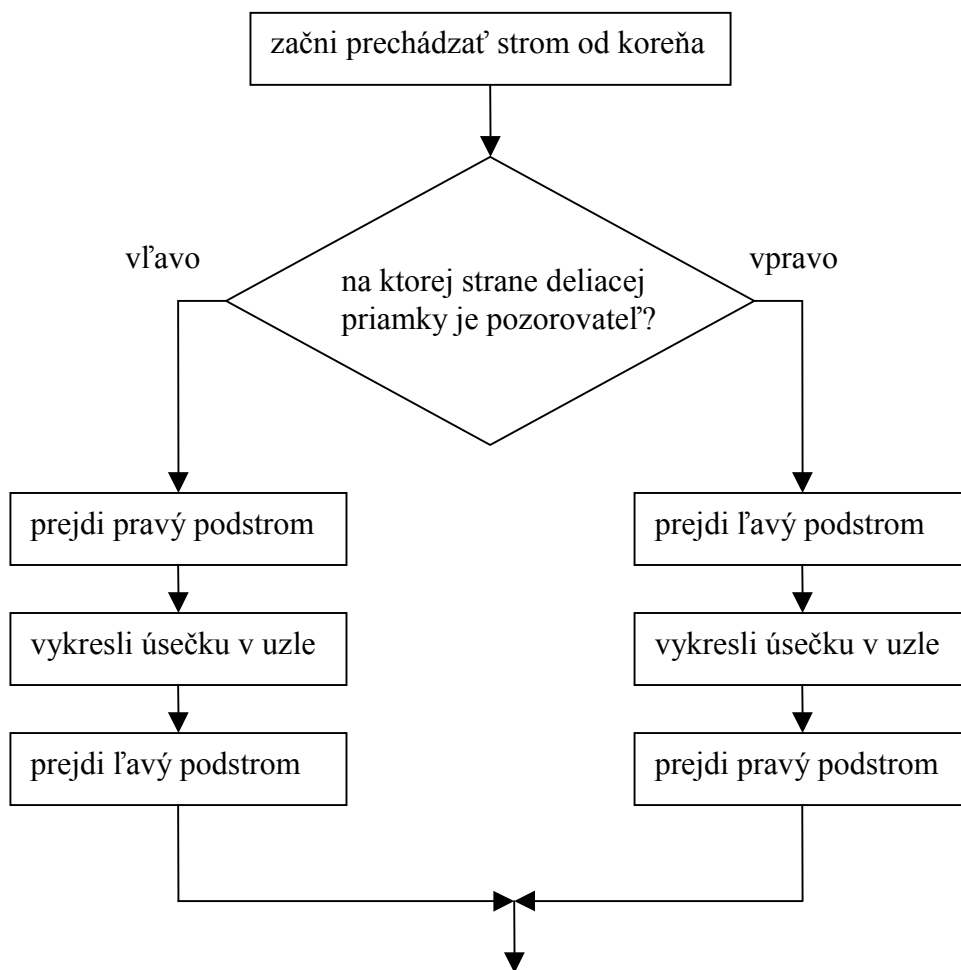
Pri tvorbe 2D-BSP stromu je algoritmus nasledovný:



Obr. 2 Algoritmus tvorby BSP stromu

V prvom kroku je možné vo všeobecnosti voliť ľubovlnnú priamku, pre riešenie viditeľnosti je možné (a zvyčajne aj výhodné) zvoliť jednu z priamok zo scény. V ďalšom texte sa tento postup predpokladá.

Ak by sa zvolila za deliacu priamku úsečka 1, výsledný BSP strom scény na predchádzajúcom obrázku by bol taký, aký je naznačený na Obr. 1. Úsečka 2 leží celá vpravo od úsečky 1 a úsečka 3 leží celá vľavo od úsečky 1.



Obr. 3 Algoritmus prechodu BSP stromom zozadu dopredu

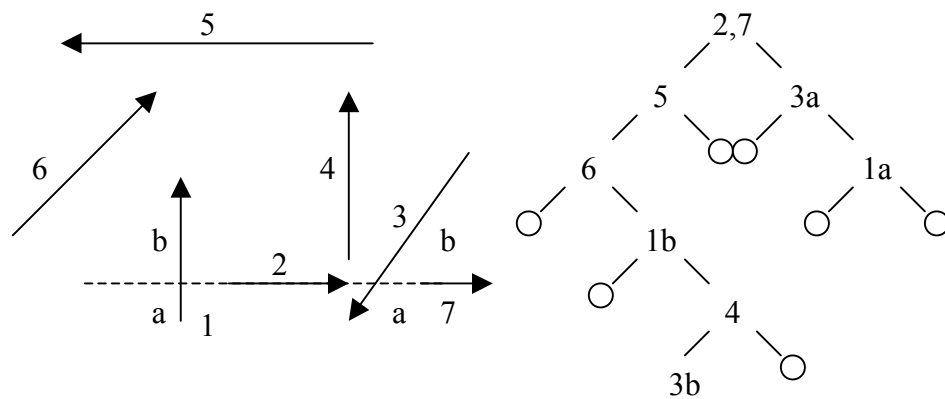
Tento strom sa teraz dá využiť napríklad na vyriešenie viditeľnosti v scéne na Obr. 1. Stačí iba podľa algoritmu na prechádzajúcom obrázku prejsť BSP stromom.

Prechod stromu scény je podľa algoritmu 3,1,2 čo presne súhlasí s postupom kreslenia scény pri maliarovom algoritme. BSP je pri riešení viditeľnosti vlastne rozšírením maliarovho algoritmu s tým, že nie je potrebné úsečky utriediť podľa vzdialenosti od pozorovateľa, ale poradie kreslenia je určené len prejdením BSP stromu.

6.2. Vlastnosti BSP

V scéne na nasledujúcom obrázku je 6 úsečiek a koreň stromu je úsečka 2. Keďže úsečky 1 a 3 neležia celé ani vľavo ani vpravo od úsečky 2, musia sa rozdeliť na dve časti, úsečka 1 na 1a 1b a úsečka 3 na 3a 3b. Týmto sa úsečky rozdelia na množiny $L=\{1b, 3b, 4, 5, 6\}$ a $R=\{1a, 3a\}$. Úsečka 7 je špeciálny prípad, lebo leží na tej istej priamke ako úsečka 2

Neleží ani napravo ani naľavo, leží "paralelne" s ňou. V takomto prípade je na ľubovôli tvorcu stromu, ako s úsečkou 7 naloží. Môže prehlásiť, že leží na jednej zo strán úsečky 2 a môže ju dokonca umiestniť do toho istého uzla stromu. Je to preto, lebo v uzloch sa nachádzajú deliace priamky a nie úsečky a úsečky 2 a 7 ležia na tej istej (deliacej) priamke. Ďalším delením množín L a R môže vzniknúť napríklad BSP strom napr. zobrazený na nasledujúcom obrázku.



Obr. 4 Zložitejšia scéna a jej BSP strom

Ak sa pozorovateľ nachádza v strede scény, tak prechodom BSP stromu podľa algoritmu prechodu BSP stromom zozadu dopredu je postupnosť úsečiek 3a, 1a, 2, 7, 5, 6, 1b, 4, 3b. Je zrejmé, že sa takýmto poradím kreslenia úsečiek docieli prekrytie vzdialenejších úsečiek bližšími a tým aj správna viditeľnosť. Pritom nezávisí na smere pohľadu pozorovateľa! Ak je navyše úsečka viditeľná len z jednej strany (je jednostranná), tak prechodom BSP stromu sa nielenže vyrieši viditeľnosť, ale jeho prechod je vlastne "zadarmo", pretože testovanie, či je pozorovateľ na privrátenej alebo odvrátenej strane úsečky by sa muselo aj tak vykonať (výsledok tohto testu je priamo k dispozícii v kroku 2 algoritmu prechodu BSP stromom).

V BSP strome existujú 2 typy entít: uzly a listy. Uzly sú v diagramoch označené číslicami a listy krúžkami. Uzol predstavuje útvar, ktorým sa priestor delí a list predstavuje konvexný podpriestor. V prípade scény na Obr. 4, keďže sa pracuje v rovine, obsahujú uzly deliace priamky a listy zasa konvexné polygóny, ktoré vznikli delením roviny pri tvorbe stromu. Situáciu znázorňuje predchádzajúci obrázok. V strome je 9 listov, teda rovina je rozdelená na 9 častí. Na samom začiatku je spomenuté "hierarchické" delenie priestoru a táto vlastnosť vyjde na povrch ak si uvedomíme, že uzol (2,7), ktorý je zároveň aj koreňom stromu delí rovinu na dve časti.

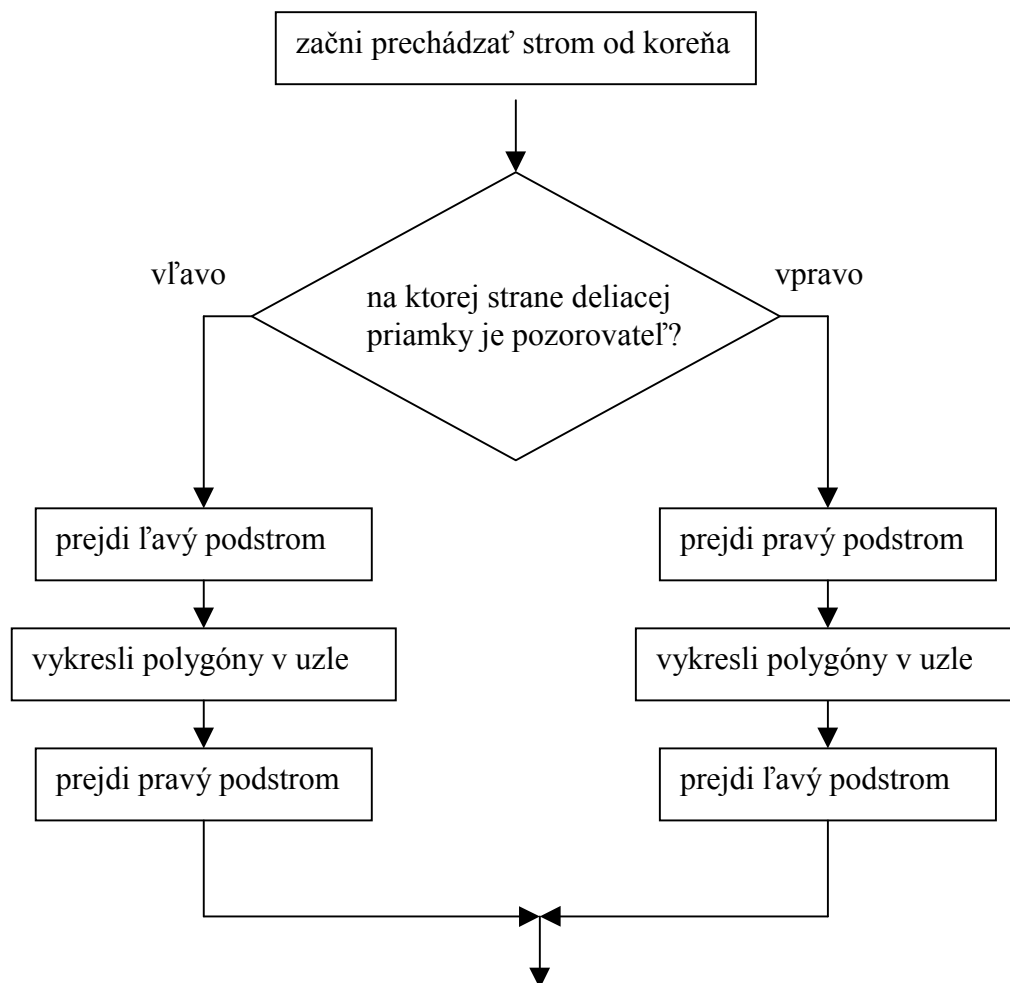
Prvá časť je zjednotením konvexných útvarov z ľavej polovice stromu (polygóny V,VI,VII,VIII a IX) a druhú časť tvorí zjednotenie polygónov z pravej časti stromu (polygóny I,II a III). Teda uzol predstavuje (okrem deliaceho útvaru) aj podpriestor tvorený zjednotením podpriestorov v jeho poduzloch.

6.3. Použitie BSP

Je potrebné zdôrazniť, že na rozdiel od maliarovho algoritmu rieši BSP viditeľnosť vždy korektne. BSP tento problém odstráni rozdelením plôch na časti. Po rozdelení je už možné nájsť správne poradie kreslenia týchto útvarov.

BSP je veľmi vhodná technika na riešenie viditeľnosti v priestore. Pri BSP v 3D sa namiesto úsečiek použijú polygóny a namiesto deliacich priamok deliace roviny. Samotný algoritmus je v 3D rovnaký a vo všeobecnosti použiteľný pre N-rozmerný priestor s N-1 rozmernými deliacimi útvarmi.

Použitím algoritmu (prechod BSP stromom) sa dosiahne rovnaký efekt ako pri maliarovom algoritme, teda vzdialenejšie polygóny budú prekryté bližšími. Je však efektívnejšie, ak sa už nakreslené polygóny neprekresľujú a ak sa toto zabezpečí, je výhodnejšie použiť nasledujúci algoritmus.



Obr. 5 Algoritmus prechodu 3D BSP stromom spredu dozadu

Pri tomto postupe sa budú najprv kresliť polygóny najbližšie k pozorovateľovi. Ošetrenie neprekresľovania je síce práca navyše, ale v konečnom dôsledku stúpne výkon niekoľkonásobne, pretože strom nie je potrebné prejsť celý, ako pri algoritme na Obr. 3, a navyše sa každý pixel obrazovky kreslí iba raz, čo hrá veľkú úlohu pri náročnom výpočte farby pixelu napríklad pri textúrovaní. Samozrejmosťou je, že prechod stromom je výhodné čo najskôr zastaviť, akonáhle je nakreslená celá obrazovka.

Navyše pri prechode stromom je možné za určitých podmienok, vyplývajúcich z konkrétnej aplikácie algoritmu (napríklad použitím hraničných útvarov pre uzly), vynechať celé vetvy stromu naraz a tým algoritmus prechodu ešte zrýchliť.

V procese tvorby BSP stromu vystupuje do popredia problém voľby vhodnej deliacej roviny pričom v zásade existujú dve stratégie. Buď je potrebné mať strom čo najviac vyvážený, alebo je vhodné obmedziť počet delení polygónov na minimum. Sú to protichodné požiadavky a závisí od použitia BSP stromu pre konkrétne účely.

Konštrukcia napríklad maximálne vyváženého stromu je však NP-úplný problém a preto je vo väčšine prípadov potrebné sa uspokojiť pri hľadaní najvhodnejšieho koreňa s rôznymi heuristikami a aproximáciami. Ak sa BSP využíva na riešenie viditeľnosti je výhodné minimalizovať počet delení polygónov na minimum, pretože často je potrebné prejsť veľkú časť stromu a navyše kreslenie štyroch či piatich polygónov (ktoré vznikli rozdelením jediného) je oveľa zdĺhavejšie ako kreslenie polygónu vcelku.

6.4. Výhody a nevýhody BSP

Výhody:

- korektne rieši viditeľnosť pri zložitosti $O(n)$ pri ľubovoľnej polohe pozorovateľa,
- hierarchické rozdelenie priestoru na konvexné podpriestory, čo pomáha pri prechode stromom, kedy je možné vynechať naraz celý podstrom,
- umožňuje kresliť zozadu dopredu aj spredu dozadu.

Nevýhody:

- takmer nulová flexibilita, pretože pri pohybe objektov (úsečiek, polygónov) sa mení ich vzájomná poloha a tým aj usporiadanie v strome, a preto je nutné odznova prepočítať celý BSP strom (prípadne jeho časť), čo pri systémoch pracujúcich v reálnom čase neprichádza do úvahy. Čiastočným riešením (pri veľkých, pohybujúcich sa objektoch) je vkladanie polygónov do stromu v reálnom čase (v predbežne vypočítanom strome sú vtedy len statické objekty),
- pri tvorbe stromu dochádza zákonite k zvýšeniu počtu objektov v scéne spôsobeného delením,
- náročný *preprocessing*.

7. Záver

OpenGL, vrátane opísaných metód, je softwarová nadstavba grafických hardwarových zariadení postavených na rýchlych ale i menej výkonných grafických čípoch. Rozhranie OpenGL je postavené formou grafickej knižnice, ktorá plní funkciu rozhrania medzi používanými grafickými zariadeniami s ich programovým vybavením, teda s ich grafickými ovládačmi a aplikáciami používanými v súčasných aplikáciách bežiacich na systémoch Unix, Mac a Windows. Štandard OpenGL je plne otvorený čo znamená, že všetky jeho funkcie a procedúry sú plne dostupné a zdokumentované a možno ich upravovať a dopĺňať.

Opisovaná problematika nadobúda význam najmä v súvislosti s nasadzovaním pokročilých simulačných systémov v rezorte MO SR. Budúci alianční partneri SR vysoko oceňujú úspešné a rýchle zavedenie systémov konštruktívnej simulácie do výcviku vojakov OS SR vo vojenských akadémiách. Od ich pracovníkov ale čakávajú, že sa zapoja i do sledovateľského výskumu a do aktívneho riešenia čiastkových problémov výcviku s podporou počítačových simulácií. To si vyžaduje systematické osvojovanie najnovších poznatkov z mnohých vedných disciplín a ich aplikovanie vo výcviku vojakov OS SR a v štúdiu vojakov graduálneho i kariérneho vzdelávania. Príslušníci Katedry informatiky a výpočtovej techniky plnia rad významných úloh pri zavádzaní simulačných technológií v rezorte MO SR. V článku je opísaná časť problematiky riešenej na katedre teoreticky i prakticky.

8. LITERATÚRA

- [1] Neider, J., Davis, T., Woo, M.: OpenGL Programming Guide, The Official Guide to Learning OpenGL, Release 1, Addison-Wesley, 1993,
- [2] Watt, A., Watt, M.: Advanced Animation and Rendering Techniques, Addison-Wesley, 1992,
- [3] Foley, J. D., van Dam, A., Feiner, S. K., Hughes, J. F.: Computer Graphics Principles and Practice, Addison-Wesley, 1989,
- [4] Žára, J. a kolektív: Počítačová grafika – princípy a algoritmy, Grada a.s., Praha, 1992,
- [5] Žára, J., Beneš, B., Felkel, P.: Moderní počítačová grafika, Computer Press, Praha, 1998,
- [6] Smernice pre používateľov projektu „Digitálny model reliéfu“, FMO, Praha, 1990,
- [7] Dokumentácia DMU-200,
- [8] Adamjak, M., Piroh, J.: VISÚ Konečný projekt, Topografický ústav Armády SR, Banská Bystrica, 1997.
- [9] Prednášky s predmetov: *Počítačová grafika, Modelovanie a simulácia bojovej činnosti a Vojenské informačné systémy*. Katedra informatiky a výpočtovej techniky, Vojenská akadémia v Liptovskom Mikuláši..