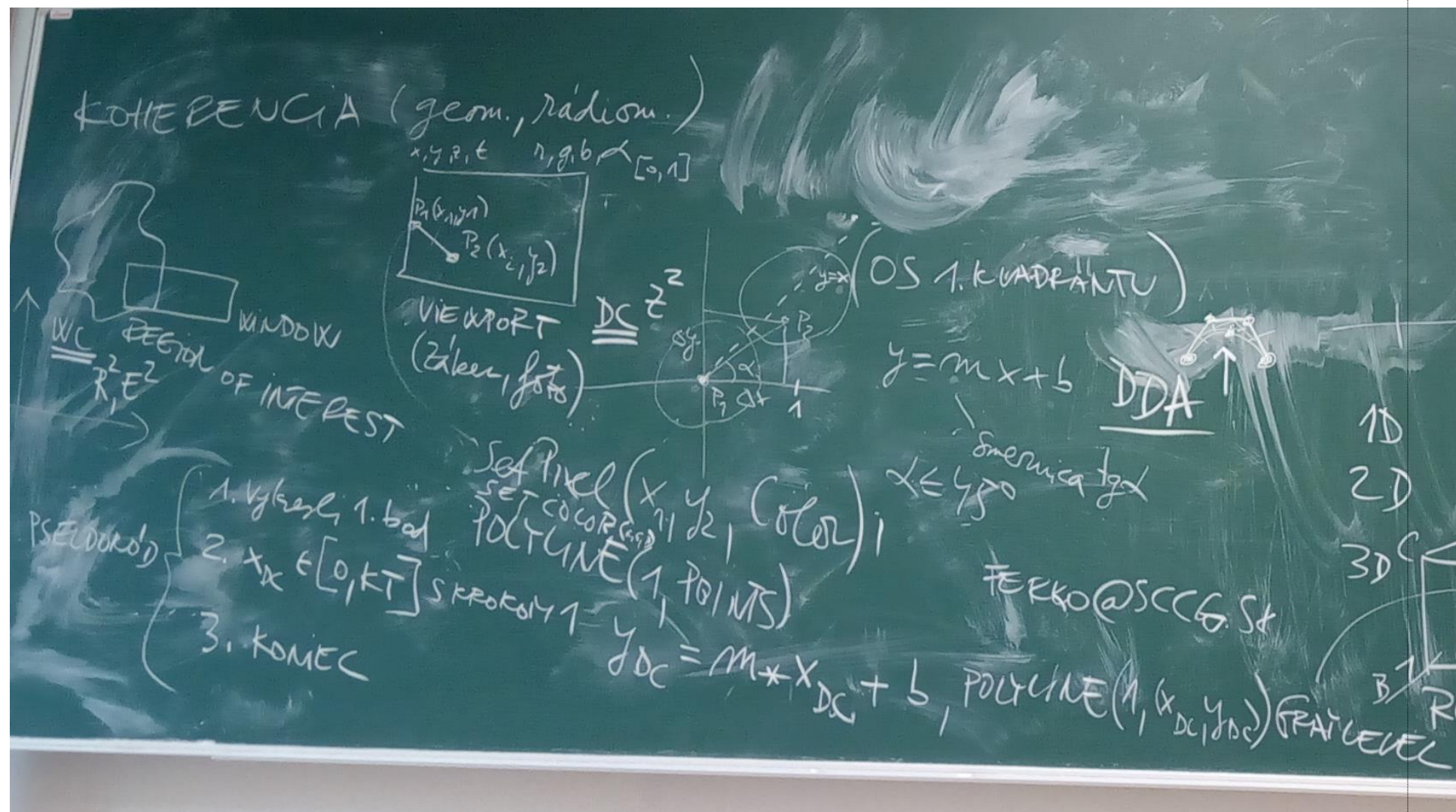


# Koherencia geometrická a rádiometrická UPG, podpora poznámok 6. týždeň LS 2020



Tyžden pred midtermom nadviazeme na videoprednasku doc. Božeka **Mnohouholníky**, v ktorej spojil nam už známe grafické výstupné prvky lomenu čiary a (jednoduchú) vyplňovú oblasť pod označením Mnohouholník, ktorého obsah násiel pomocou triangulácie.

K videoprednaske pridáme jediný ďalší obrázok.

Na obrázku tabule, ktorý v roku 2016 laskavo zdokumentovala Nikola Gibasova, popremysleme spolu o ďalšom princípe v počítačovej grafike, ktorý sme už vlastne použili. Odborne sa nazýva **Koherencia** a použili sme ju mlčky pri rasterizácii úsečky. Algoritmická stratégia, zapísaná v pseudokode vľavo dole, sa nazýva **Iterácia**, išli sme v 1. oktante z bodu (0,0) s krokom 1 a počítali sme y (v súradniciach zariadenia, DC). Pretože už máme zavedený POLYLINE, vykresľujeme bod po bode na logickej úrovni týmto prikazom grafického systému, čo fyzicky znamená SetPixel, zápis do pamäte obrazu, zobrazenie (display). Parameter Color zatiaľ používame, ako keby sme mu rozumeli, že znamená farbu.

Z okna vo svetových súradniciach (Euklidovej roviny) sme transformovali do zaberu úsečky  $P_1P_2$  a rasterizujeme ju po priamke v prvom oktante. Bod za bodom môžeme vykresliť preto, že „svet sa mení nenápadne“, že platí princíp koherencie, že od pixlu k pixlu tu niet rozporov, a preto priatelia informatici hovoria o iterácii, budovaní riešenia krok za krokom. Kým vkladanie diagonal (doc. Božek ich nazýva vnútorná uhlopriečka mnohouholníka) vedie na stratégiu **Rozdeľuj a panuj** (ktorú už poznáme z FloodFill), odrezávanie uší (Ear Cutting) je iterácia, ucho po uchu.

Potrebovali sme orezanie do zaberu (stratégia **Prune & Search**) tzv. **Dervišovu ťavu**, pridali sme k úsečke „niečo viac“, presne dve polpriamky, aby sme mohli využiť smernicový tvar priamky v prvom oktante. Použili sme aj dve zo štyroch geometrických súradníc ( $x, y, z, t$ ) na prepis jazyka bodov do jazyka čísiel podľa Descartesovej metódy. Ak už máme úsečku, môžeme pomocou úsečiek alebo bodov na nich vyčísl'ovať súradnice bodov krivky, a algoritmus de Casteljau znovu využíva koherenciu, bod za bodom, lineárne, kvadraticky, kubicky aj bikvadraticky (quartic).

**Komenského Škola hrou, pre študijný typ Kinesthetic De Casteljau's Algorithm and Bézier Curves**

<http://www.malinc.se/m/DeCasteljauAndBezier.php>

Vieme teda koherentne interpolovať body na úsečke, keď poznáme iba jej krajné body, a na oblúku krivky kvadraticky a kubicky (bikvadratické interpolácie sa v grafike bežne nepoužívajú).

Načo nám to bude?

Od riadkovej koherencie prejdeme ku plošnej koherencii, keď budeme rasterizovať, osvetľovať alebo vyfarbovať trojuholníky, na ktoré vieme rozdeliť každý jednoduchý mnohoúhelník. A práve vyfarbovaním trojuholníkov sa zaoberáme na pravej strane tabule. Riešiť budeme prípad, keď máme vo vrcholoch trojuholníka zadané rôzne farby alebo úrovne šedej (graylevel).

Na ľavej polovici tabule nám ostáva nedovysvetlená farba a email prednášajúceho. To znamená iba podpis kriedou pohybujúceho subjektu.

Čo s farbou? [Ružický, Kapitola 12, od s. 179]

Viditeľná časť elektromagnetického spektra s vlnovou dĺžkou asi od 400 do 700 nanometrov sa dá jednorozmerne matematicky modelovať na úsečke. Ľudia v týchto farbách dúhy rozpoznávajú asi 400-tisíc odtieňov, no programátor v tomto modeli nezapíše bielu farbu.

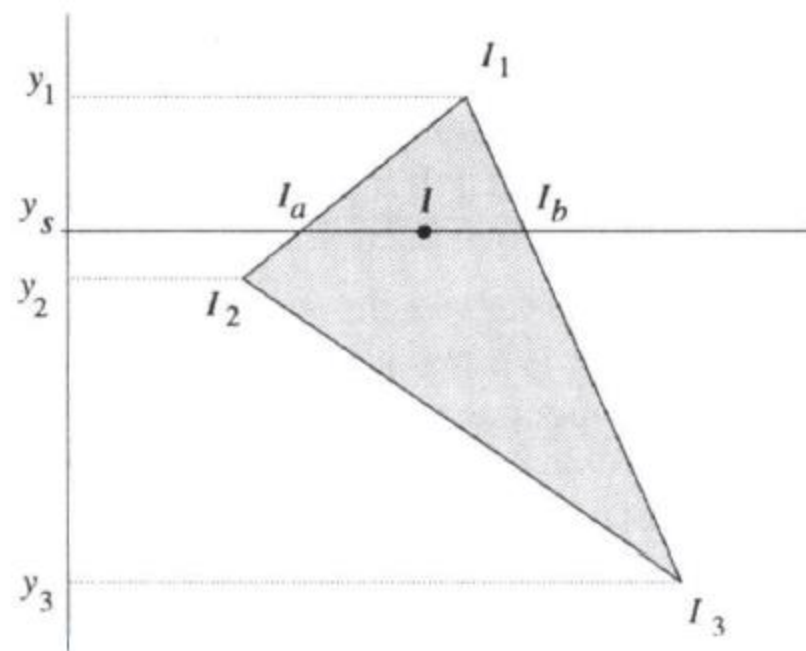
Od roku 1925 zaviedli fyzici dvojrozmerný model, ktorý sa označuje ako CIE diagram, no nezapíšeme v ňom čiernu. Umožňuje zapísať bielu a zaviesť opačné farby. (Trojuholník v CIE diagrame sa nazýva color gamut. Náčrtok je vpravo dole.)

V grafike preto používame na programovanie displejov trojrozmerný model RGB. V bode (0,0,0) kódujeme čiernu, (1,1,1) bielu a na ich spojnici koherentne prechádzajú úrovne šedej, platí  $R=G=B$ . Nočné videnie bez farieb vníma iba odtiene šedej. Ostatných 6 rohov kocky patrí trom primárnym farbám a ich opačným, ktoré sa zas používajú na programovanie farebných tlačiarň. Modrú zložku vidíme slabšie, stačili by 4 bity, no v praxi používame 24 bitov.

Používa sa aj perceptuálne uniformný model HSV, ktorý na rozdiel od displejov a tlačiarň vhodne modeluje ľudské vnímanie farby, [Ružický, s. 188.

A interpolujeme v koherentnej farebnej kocke rovnako ako po úsečke, iba namiesto polohy bodu meníme farbu.

Najšedšiu šedú (0.5,0.5,0.5) v strede kocky získame ako polovicu bielej a polovicu čiernej, po zložkách. Vieme teda interpolovať farby v trojuholníku, po hranách lineárne a vovnútri bilineárne.



*Obr. 13.9 Interpolácia intenzity pre trojuholník*

Na obr. 13.9 máme určené intenzity  $I_1, I_2, I_3$  vo vrcholoch trojuholníka. Najprv vypočítame intenzity  $I_a, I_b$  na zvolenom riadku  $y = y_s$  v krajných bodoch trojuholníka:

$$I_a = I_1 - (I_1 - I_2) \cdot (y_1 - y_s)/(y_1 - y_2) ,$$

$$I_b = I_1 - (I_1 - I_3) \cdot (y_1 - y_s)/(y_1 - y_3) .$$

Výslednú intenzitu  $I$  v bode  $(x, y_s)$  na úseku medzi dvoma krajnými bodmi opäť interpolujeme:

$$I_p = I_b - (I_b - I_a) \cdot (x_b - x)/(x_b - x_a) .$$

Daný postup používame nielen na dopočítanie farby, ale aj na otexturovanie daného trojuholníka, pri jeho zväčšovaní či osvetľovaní.

Na midterme bude príklad takéhoto typu:

**1. Výpočet bilineárnej interpolácie pre šedotónový obrázok, napr. ťažisko rastrového trojuholníka, ktorého intenzity pre vrcholy sú dané dĺžkami, rozdielmi a súčtami počtu znakov Vášho mena.**

Tol'ko doplnok Koherencia po prednáške o mnohouholníkoch a triangulácii.

Druhý doplnok sa bude venovať triangulácii samotnej.  
Naučíme sa skvelú trianguláciu, ktorá má vynikajúce vlastnosti.

Najprv sa s nou pohráme:

<https://demonstrations.wolfram.com/ConvexHullAndDelaunayTriangulation/>

Definície:

<https://mathworld.wolfram.com/DelaunayTriangulation.html>

<https://mathworld.wolfram.com/VoronoiDiagram.html>

<https://mathworld.wolfram.com/ConvexHull.html>

Predbežná štruktúra obsahu:

### **Základy**

- Prehľad problematiky, motivácia a aplikácie.
- Základné pojmy.
- Minimum o hardveri.
- Softverové systémy. Referenčný model počítačovej grafiky. Zobrazovací kanál.
- Základné reprezentácie – pixel, polygon, 2D, 3D, farby.
- Úvodné poznatky o vizuálnom umení a dizajne.

### **Matematika**

- Súradnicové systémy
- Transformácie
- Náhodné čísla
- Geometria v rovine a priestore
- Matice
- Vybrané numerické metódy

## **Grafické systémy, fyzické a logické zariadenia**

## Definícia a referenčný model počítačovej grafiky

Počítačová grafika je podľa definície ISO odvetvie vedy a techniky zaoberajúce sa metódami a postupmi prevodu údajov na grafické zobrazenie a grafického zobrazenia na údaje pomocou počítača.

Počítačovú grafiku delíme na dve časti - pasívnu (zobrazovaciu) grafiku, ktorá z modelových dát vytvára obrázok (picture), a interaktívnu grafiku, kde komunikácia prebieha opačným smerom, zdrojom dát sa v tomto prípade stáva človek (operátor, user, používateľ). Pojmom interaktívna počítačová grafika sa však často označuje aj kombinácia oboch smerov spracovania vizuálnej informácie – zobrazovania aj interakcie.

Základnou predstavou pre medzinárodné (ISO) funkčné normy v oblasti počítačovej grafiky sa v 80. rokoch stal **referenčný model počítačovej grafiky**. Spracovanie informácie sa v ňom rozdeľuje na dve etapy: **modelovanie** a **zobrazovanie**. Obe rieši **aplikačný program**, ktorý ale pre zobrazovanie zvyčajne využíva funkčnosť **grafického systému a pracovnej stanice** (obrazovky alebo iného zobrazovacieho zariadenia a vstupných zariadení). Grafický systém je teda zároveň jazykom, súborom funkcií aj aplikačným programovým rozhraním (API) medzi aplikáciou a hardverom. ISO definovalo pre grafiku normy pre funkčné systémy GKS (Graphical Kernel System) a PHIGS, ktoré sa v praxi nepoužívajú. Grafika sa v praxi najčastejšie programuje pomocou voľne šírenej knižnice OpenGL a v prostredí Windows pomocou firemnej knižnice DirectX, na internete sa najčastejšie používajú jazyky HTML, JAVA, JAVA-3D, VRML a Flash. Knižnicu OpenGL si možno stiahnuť z web stránky spoločnosti Silicon Graphics (SGI):

- <http://www.sgi.com/products/software/opengl/>

Referenčný model počítačovej grafiky vymedzuje aj tri rozdielne ľudské činnosti: aplikačný programátor, autor grafického systému a operátor (používateľ). Základnú štruktúru aplikačného systému, využívajúceho (interaktívnu) počítačovú grafiku možno charakterizovať takto: **aplikačný program** (prípadne s aplikačnou databázou) spracúva model reálnych alebo abstraktných objektov a rozhoduje, čo zobraziť. Popíše teda pre **grafický systém** tvar a vzhľad objektu a grafický systém vykoná požadované zobrazenie na **zariadení**. Autor programu tak nemusí študovať detaily zariadení a aplikačný program sa stáva na zariadení nezávislý a prenositeľný na iné zariadenia.

Analogicky, grafický systém odovzdáva aplikačnému programu vstupy od operátora, ktorý komunikuje s programom cez vstupné zariadenie. Jednou z hlavných myšlienok takéhoto konceptuálneho modelu je už spomenuté **oddelenie modelovania a zobrazovania**. Grafický systém sa stáva nezávislým od aplikácie, rastie jeho univerzálnosť a klesá zložitosť. O hranici medzi modelovaním a zobrazovaním, ktorá vždy vedie „na území“ aplikačného programu, sa vedú spory, napr. kam patrí hierarchia skladania zložitejších objektov z jednoduchších. Grafický systém GKS modelovanie nemá, kým systém PHIGS obsahuje aj štruktúry a ďalšie nástroje na

hierarchické modelovanie. Existuje aj názor, že kým sa neoddeli modelovanie od zobrazovania, ťažko hovoriť o grafickom systéme ako o univerzálnom nástroji na vývoj grafických aplikácií.

Interaktívne editovanie trojrozmerných scén umožňujú viaceré voľne stiahnuteľné animačné programy, napr. od spoločností AliasWavefront, Caligari a Discreet:

- Maya PLE - <http://www.aliaswavefront.com/en/products/maya/ple/index.shtml>
- trueSpace - [www.caligari.com](http://www.caligari.com)
- G max - <http://www.discreet.com/products/gmax/>

PLE znamená personal learning edition, Gmax je zjednodušenou verziou komerčného nástroja 3D Studio Max a trueSpace voľne ponúka verziu 4.0. Na fakulte používame Blender.

Príbuznou oblasťou informatiky je **spracovanie obrazu** (image processing). Spracovanie obrazu má na vstupe obraz (image) a na výstupe tiež obraz, kým grafika obrázkov (picture) vytvára z modelu, geometrického a rádiometrického (osvetlenie, farby, textúry...). Zjednodušene povedané, typický obraz môže byť 2D alebo 3D, kým bežná grafika/animácia pracuje v 7D so súradnicami (x, y, z, t) a (r, g, b), pričom na rastrovej obrazovke sa zobrazujú projektované pixle adresované v dvojrozmernom poli. Postupnosti transformácií z priestoru modelu do priestoru zobrazenia sa hovorí **zobrazovací kanál** (rendering pipeline). Ide o postupnosť súradnicových systémov a transformáciami medzi nimi tak, aby zobrazený bod (v súradniciach zariadenia) zodpovedal bodu v reálnom svete reprezentovanom v modeli (svetové súradnice) a polohe a orientácii kamery (súradnice pohľadu, view coordinates). Z matematického hľadiska sa zobrazovací kanál konštruje pomocou skladania homogénnych matic tak, aby sa bod z okna alebo viditeľného objemu vo svetových súradniciach zobrazil na pixel v zábere v súradniciach zariadenia čo najpresnejšie. Tieto súradnicové systémy sú karteziánske a transformácie medzi nimi sa skladajú z matic posunutia, škálovania, otočenia a perspektívnej alebo paralelnej projekcie.

V troch dôležitých oblastiach sa tento štandardný postup spracovania informácie líši – 1. vizualizácia, 2. objemová grafika (volume graphics) a 3. obrazovo-orientované modelovanie a zobrazovanie (image-based rendering). **Vizualizácia** používa všetky metódy počítačovej grafiky **na podporu porozumenia**, takže nemožno hovoriť o presnosti modelu a farieb, lebo napr. modrá šípka pri predpovedi počasia nemá v reálnom svete reálny vzor. **Objemová grafika** zas používa ako model hodnoty namerané (navzorkované) v trojrozmernom poli, najmä v medicínskej informatike sa zobrazuje trojrozmerný obraz (napr. ľudského tela) s pomocou transparenzie (priehľadnosti). K súradniciam farby pribúda aj priehľadnosť daného pixla (r, g, b, transparenca), vyjadrená hodnotami od nula po jedna. Treťou oblasťou s odlišným postupom spracovania je **obrazovo-orientované modelovanie resp. zobrazovanie** (image based modeling resp. rendering), kde sa z čo najmenšieho počtu obrazov vytvorí kocková, valcová alebo guľová panoráma,

ktorá poskytuje v nej “uprostred” danému pozorovateľovi panoramický obraz daného nafotografovaného reálneho prostredia bez potreby vytvárania trojrozmerného modelu. Autorské nástroje na vizualizáciu, objemovú grafiku a tvorbu panorám sú široko dostupné. Veľmi frekventované vizualizácie pre obchodnú grafiku (koláčový diagram, šípky a iné symboly) možno nájsť v ponukách rozšíreného programu PowerPoint. Voľný nástroj pre objemovú grafiku je VolVis –

- [http://www.cs.sunysb.edu/~vislab/volvis\\_home.html](http://www.cs.sunysb.edu/~vislab/volvis_home.html).

Nevyhnutnou etapou pred grafickým zobrazením je **geometrické modelovanie**. V počítačovej grafike sa používa 5 až 7 súradnicových systémov, ak nerátame ďalšie, ktoré treba pre multimédiá. Nasleduje charakteristika 5 súradnicových systémov pre 2D a 3D grafiku podľa ISO noriem CGRM, PREMO, PHIGS:

- modelové (pre objekty, lokálne súradnice pre každý objekt)
- svetové (scéna, kde sa objekty kombinujú)
- normalizované, interné pre grafický systém
- pohľadu, kamery
- zariadenia (DC, device coordinates)

Šiesty súradnicový systém sa označuje TLC (text local coordinates), v ktorom návrhári fontu reprezentujú krivky pre true type fonty. Tieto súradnice sú obdobou modelových pre grafický výstupný prvok text. Siedmy súradnicový systém sa v prípade potreby používa na textúrovanie povrchu objektov a pre tieto súradnice sa zaužívalo označenie (u, v). Okrem toho sa používajú aj súradnice na vyjadrenie polohy svetelného zdroja pre výpočet viditeľnosti či osvetlenia, alebo sa môže zaviesť pre potreby interakcie rozdelenie obrazovej alebo pracovnej plochy do nejakých vhodných častí. Existujú aj ďalšie, tzv. multimediálne súradnice (napr. na vyjadrenie polohy záznamu v excelovskej tabuľke, spreadsheet). Pri každom súradnicovom systéme treba zvoliť počiatok, osi, orientáciu a pomer strán. Po zavedení súradníc možno už geometriu scény a jej transformácie vhodne vyjadriť pomocou lineárnej alebo nelineárnej algebry a maticového vyjadrenia afinných transformácií a projekcií.

Objekty v scéne sa v počítačovej grafike reprezentujú ako oblak bodov, drôtový model, hraničná reprezentácia (najčastejšie sa povrch objektu aproximuje nepravidelnou trojuholníkovou sieťou, ktorej sa hovorí mesh) alebo už spomenutá objemová reprezentácia. Na všetky tieto reprezentácie treba používať efektívne dátové štruktúry. Priemyselným štandardom pre väčšinu aplikácií počítačovej grafiky a vizualizácie je hraničná reprezentácia a reprezentatívnym kódovaním 3D scén je formát VRML (Virtual Reality Modeling Language, norma ISO), v ktorom môžu byť do scény zahrnuté aj objekty, adresované na inom počítači po internete použitím URL (Uniform Resource Locator). Na zložitejšie aplikácie treba objekty vyjadrovať pomocou povrchov s viacerými úrovňami detailu

(level of details). VRML podporuje 8 úrovní. Pre niektoré operácie ako morphing býva najvhodnejšie mať objekt reprezentovaný funkcionálne (F-rep).

**Všeobecný úvod do hocičoho** na FMFI UK by sa mohol začať spoločným vymýšľaním odpovede na otázku, čo je zaujímavé alebo čo znamená zaujímavosť. Zaujímavosť je jednak pojmom, ktorý sa často používa bez definovania, no zároveň určuje naše potreby alebo priority. Konkrétne, zaujímavosť vyjadruje aj mieru nášho času, ktorý sme ochotní venovať tej-ktorej činnosti, téme (téma je po grécky zázrak), či riešenému problému. Všeobecnú odpoveď na elementárnu zaujímavosť dáva etológia zvierat. Pre každý živý organizmus sú neprestajne zaujímavé 4 základné potreby: 1. útek, únik, ochrana pred nebezpečím, 2. útok na súpera či korisť, 3. hlad a 4. zachovanie druhu. Prvé dva problémy treba často riešiť v čase pod jednu sekundu a zodpovedajú im emócie **strach** a **hnev**, fobia a agresia. **Hlad** a **sex** predstavujú na riešenie „rádovo pomalšie“ problémy, merateľné časovo v hodinách až rokoch. Mimochodom, dve z týchto základných potrieb predstavujú väčšinu problematiky komerčných filmov z Hollywoodu, kým hlad a agresia nie sú natoľko „fotogenické“ ako sebaobrana a rozmnožovanie. Zaujímavosť každého problému možno zvýšiť aj umelo - skrátením času na jeho riešenie. Piatym základným zaujímavým problémom pre každé živé stvorenie je teda **čas**, spotreba času, jej štruktúra, jeho nedostatok, jeho organizácia či jeho straty. Šiestym všeobecne zaujímavým problémom by mohla byť **pamäť**, jej spotreba, jej štruktúra, jej nedostatok, jej organizácia či jej straty. Pamäť súvisí s časom viacerými spôsobmi. Umožňuje čas šetriť, ak poskytuje efektívne rozhodnutia. Umožňuje časom plyvať, ak sa v nej nedarí nájsť správnu odpoveď. Umožňuje čas organizovať a vnímanie času naopak podporuje organizáciu pamäti. Siedmou položkou všeobecnej zaujímavosti by mohlo byť **nové riešenie** niektorého problému, ktoré nám môže sprístupniť kombinácia skúsenosti a tvorivosti alebo iba **ново použitá skúsenosť** alebo iba **tvorivosť**. Novo použitú skúsenosť môžeme dosiahnuť tak, že spravíme pokus, náhodne alebo zámerné. Tvorivosť by sa zas dala definovať ako vyriešenie nového problému, na ktorý dovtedajšia skúsenosť nestačí. V oboch prípadoch – cez skúsenosť či tvorivosť – nás zaujíma **objav**. V oboch prípadoch nás skúsenosť a tvorivosť môžu priviesť k objavu aj tým, že rozpoznáme **omyl**. „Energia omylu“ teda môže byť rovnako zaujímavá ako energia opaku omylu, čiže úspešného pokusu.

Vo „vzťahovom trojuholníku“ čas-pamäť-tvorivosť žijeme tiež neprestajne a ustavične. Čas na riešenie možno skracovať použitím pamäti alebo tvorivým objavom nového postupu. Tvorivý nápad možno neraz úspešne nahradiť skúsenosťou z našej pamäti (starým riešením) alebo ho dosiahnuť či dosahovať vytrvalým investovaním času do hľadania nového riešenia. Pamäť sa rozvíja práve prostredníctvom času a tvorivosti.

Pamäť siaha do minulosti, okamžitej, krátkodobej a dlhodobej. Čas patrí prítomnosti. Tvorivosť odкрýva uskutočniteľné cestičky v možných alebo niekedy iba mysliteľných bludiskách budúcnosti. (Môžeme si myslieť v euklidovskej rovine trojuholník so súčtom vnútorných uhlov iným ako 180 stupňov, dvoch pravých uhlov, ale pravosť mysliteľnosti testujeme uskutočniteľnosťou. A taký trojuholník v rovine neexistuje a v žiadnej z alternatív budúcností ho neuskutočnime.)

Myslíme pritom vždy iba cez pamäť. “Když studujeme, probíráme, analyzujeme nějakou skutečnost, analyzujeme ji takovou, jaká se jeví naši paměti. Známe každou skutečnost jen v minulém čase. Neznáme ji takovou, jaká je v přítomnosti, ve chvíli, kdy se děje, kdy je“ [Kund04, s. 16]. Inak povedané, naše poznanie a poznávanie vždy konfrontuje dva časy. Čas prítomný, keď s pamäťou

stopou skutočnosti zaobchádzame, s časom minulým, keď sa uložila pamäťová stopa, pomocou ktorej študujeme, preberáme, analyzujeme. Keď len hoci klikneme na obrazovke na iné miesto tohto textu, vytvárame hneď naraz dokonca dve minulosti [Qvor01] – 1. minulosť v usporiadaní napísaného textu a 2. minulosť preusporiadania textu podľa polôh kurzora, čiže históriu našej interakcie, riadenej tým, čo bolo pre nás také zaujímavé, že sme tadiaľ riadili kurzor.

A práve čas, ktorý venujeme časovaniu (plánovaniu), pamätaniu (spomínaniu do minulosti) a vytváraniu (“spomínaniu” na budúcnosť), meriame, kvantifikujeme a kvalifikujeme zaujímavosťou. Je to ten čas, ktorý nám ostane či zvýši po riešení štyroch 4 základných biologických potrieb (zachovanie jednotlivca a druhu): 1. útek, 2. útok, 3. hlad a 4. sex.

V čase, ktorý ušetríme v „biologickom štvoruholníku“ (vo sfére biótu, „naš hardver“), sa môžeme vnoriť do „kultúrneho trojuholníka“ (do sféry intelektu, „naš softver“). O týchto sedem typov elementárnych zaujímavostí v čase nášho života sa nedelíme iba so svojou pamäťou, kam sa priebežne ukladá naša celoživotná skúsenosť, ale aj so svojím podvedomím počas snívania, so svojimi deťmi a rodinou („biologickými dedičmi“) počas komunikácie a kooperácie, so svojimi budúcnymi „kultúrnymi dedičmi“ počas písania textov, vyučovania, prednášok a seminárov, komponovania hudby či maľovania obrazov. Znepokojujúci autori ako Rupert Sheldrake či Stanislav Grof pripúšťajú dokonca aj našu obojstrannú komunikáciu s neživou prírodou pomocou tzv. morfických polí či pamätanie si pred narodením a počas pôrodu. Po svojich biologických predkoch dedíme časť svojej skúsenosti a od svojich kultúrnych predkov či predchodcov, ktorých „kultúrnymi dedičmi“ sme my sami sa zase učíme, tvorivo objavujeme ten obsah ich pamäti, ktorý si kopírujeme či modifikujeme do pamäti svojej, lebo ho oni alebo my alebo spoločne vnímame ako zaujímavý. Zaujímajú nás pritom skúsenosti a vedomosti nielen vážne, použiteľné na prácu, ale aj nevážne, „užitočné“ „iba“ na hranie. Johan Huizinga tvrdí, že celá ľudská kultúra má podstatu hry.

Zaujímavosť má teda sedem všeobecných odpovedí, ale na fakulte nás zaujíma **riešenie problémov**, hľadanie odpovede na otázky, ktoré vieme z reality vyňať tak, že si vytvoríme ich myšlienkový **model**, mentálnu mapu, reprezentáciu v pojmoch, ktoré nám umožnia namiesto reálnych alebo nemožných experimentov rozmýšľať. Jan Vodňanský v Radosti z inteligencie poznamenáva, že pred rozmýšľaním spojitú realitu vždy „diskretizujeme“ do pojmov. Zaujímavosť na prednáške alebo na seminári preto poľahky vyvoláme, ak položíme otázku, zadáme problém, sformulujeme hypotézu. V cirkevnej homiletike, náuke o kázni, ktorá sa rozvíja omnoho dlhšie ako univerzity, sa odporúča, aby kázeň odpovedala na otázky, ktoré sú dokonca nevyslovené. Vtedy nastáva ten okamih, keď sa publikum pridruží, asociuje s kazateľom, po nemecky to nazývajú bod pripojenia, Anknupfungspunkt, v teórii NLP [Kupk01] zase kotva (anchor), čas ukotvenia pozornosti, prepnutie do asociovaného stavu.

Na prvej, motivačnej prednáške z počítačovej grafiky sa osvedčuje aj položiť niekoľko základných otázok o pojmoch, čo je problém, čo je model, čo je zaujímavosť. A ešte, ako by ste spočítali zrnká v kope piesku. Ak sa vám tento problém nezdá dosť zaujímavý, predstavte si, že ak ho nevyriešite, budete na úsvite o hlavu kratší a odpudivo zamaskovaný kat si práve začína brúsiť sekeru či leštiť gilotínu. Tým sa po hollywoodsky vystupňuje zaujímavosť pridaním na strachu a skrátením času na riešenie. Odpovede na tieto otázky sa majú písať na tabuľu, spravidla jedným výstižným slovom alebo logom. Odbočiť od témy a vymýšľania možno iba do

príležitostného vtipu. Hodí sa dodať, že máme na to riešenie neobmedzený rozpočet, neobmedzený tím, neobmedzený čas, že prehľadávame priestor možností úplne slobodne a bezbariérovo. Neobmedzený čas máme na riešenie, nie na jeho vymyslenie, objavenie, nápad, ako nám naliehavo a dotieravo pripomína neúprosný zvuk brúsenia a čoraz oslepujúcejší ligot nezadržateľne napredujúceho leštenia usmrcujúceho ostria. Ani jedna odpoveď sa nesmie komentovať, ani neverbálnym vyprsknutím či opovrhlivým zafňkaním či obdivným dýchaním. V publiku má byť aspoň jedna žena, lebo niektorí muži mlčia, ak sa nemajú pred kým vytiahnuť. To sú pravidlá prvej fázy riadeného hrmenia mozgov, brainstormingu. Typické odpovede na spočítanie zrní v kope piesku bývajú

Zrnko po zrnku  
Odvážiť sa odvážiť  
Cez objem  
Presýpať cez lievik a merať čas  
Uhádnuť  
Odhadnúť  
Deliť kopu na rovnaké podkopy  
Blafovať  
Najať toľko počtárov, koľko je zrní  
Počítať po desať, po sto, po tisíc  
A iné (najviac bývalo až 34 nápadov)

A potom, keď v akváriu či posluchárni nastane ticho, ktoré signalizuje, že sme na konci spoločného vymýšľania (out of ideas), dozrie čas, aby sme si ukázali, že v každej z týchto odpovedí sme vymysleli **iný model** problému a že každý model sa potom stáva predpokladom pre **iný algoritmus** (algoritmus znamená postup výpočtu a slovo tak ako algebra pochádza z arabčiny). Zrnko po zrnku nazerá na kopu piesku pomocou klasickej Cantorovej teórie množín. Kopu modeluje množina, zrnko predstavuje jej prvok a hľadáme počet prvkov, kardinalitu konečnej množiny. Váženie, objem a čas presýpania prinášajú fyzikálne vypozerované modely, ktoré do riešenia zapájajú iný hardver (váhu, odmerku, stopky) - a inú chybu výsledku. Uhádnutie by malo byť najrýchlejšie a ak by bolo presné, ťažko by sme ho odovzdali stroju alebo počítaču. Čo je však pri každom z týchto modelov **presnosť**? Pri akej veľkosti zrnko prestane byť zrnkom a stane sa zanedbateľným práškom? Najpresnejší model sa zdá byť ten množinový a algoritmus v cykle identifikuje zrnko, započíta ho a keď sa minie vstupná kopa, tak skončí. To ale oproti tým trom fyzikálnym modelom trvá omnoho dlhšie. Tie okrem hardveru vyžadujú aj zložitejšie počítanie s násobením, ba dokonca delením, ak nám posledná odmerka nevýjde celkom plná.

Z iného hľadiska môžeme prístupy k riešeniu problémov rozdeliť na deterministické, ktoré riešia pôvodný problém s nulovou chybou modelovania, heuristické, ktoré riešia problém s nejakou prípustnou chybou modelu, a empiricko-intuitívne, kde problém rieši človek na základe svojej skúsenosti (empírie) a vhladu (intuície).

1. **Deterministický prístup.** Príkladom deterministického prístupu by bolo spočítanie kopy hracích kociek (aby sme, ako pri piesku, nemali pochybnosť o tom, čo do kopy patrí a čo nie). Latinské slovo determinizmus znamená **určenosť**, presnejšie príčinnú určenosť, predurčenosť či príčinnú podmienenosť. Je to popritom aj jedna z troch vlastností každého algoritmu, ktorý má byť hromadný, deterministický a konečný, aby sme sa výsledkov dožili.

2. **Heuristický prístup.** Keby sme ale kopu hracích kociek alebo jej častí vážili, merali objem či nechali presýpať cez lievik so stopnutím času presýpania, mali by sme heuristický prístup - rýchlejšie riešenie, väčšia chyba výsledku. Grécke slovo Heuréka! znamená Našiel som! a údajne ho kričal Archimedes zo Syrakúz, keď pri kúpaní objavil Archimedov zákon o vytláčaní kvapaliny telesom. Potom ho vraj ihneď použil na dôkaz, či zlatníci kráľovi podstrčili zlatú korunu s vysokou prímесou striebra.

3. **Empiricko-intuitívny prístup** sa používa v oblastiach, kde vhodný model nemáme – ochutnávanie syrov či degustácia vína. Ale aj výnimočne nadaní fenomenálni počtári, ktorí bleskovo násobia mnohociferné čísla, tiež predvádzajú tento prístup. Môže byť bleskurýchly, môže byť úplne presný, ale nevieme ho odovzdať počítačom.

Do výpočtového procesu možno zaradiť aj náhodný krok. Vtedy dostávame tzv. stochastickú procedúru. V tomto zmysle rozlišujeme **výpočty deterministické** a **výpočty stochastické**. Príkladom stochastického postupu môže byť náhodná voľba odbočky pre myš v bludisku. Niektoré bludiská možno preskúmať a nájsť cestu systematickým odbočovaním, deterministicky. Deterministický prístup k riešeniu problémov vedie teda ku deterministickému algoritmu, kým heuristický nemusí byť iba nepresný na výstupe ale môže byť aj neurčitý v poradí vykonávania jednotlivých krokov alebo v tom, ako určí ukončenie výpočtu. Predpokladom každého algoritmu nevyhnutne ostáva **model problému**, ktorý určuje, ako sa pôvodná otázka z reality mení na vstupný dátový záznam. Algoritmus potom pre každý **vstup** vypočíta **výstup**.

Na najlepšej slovenskej fakulte (podľa hodnotenia agentúry ARRA 2006 a 2007) riešime či učíme sa riešiť iba **riešiteľné problémy**, ktoré majú jasne vymedzený vstup a výstup. Ich zaujímavosť pre prax alebo teóriu však môžu určovať aj nematematici, nefyzici a neinformatici. Príkladom myšlienkového zaujímavého problému, ktorý nie je zaujímavý prakticky, je už dávno sprchovanie. Do tohto problému vstupujú prúdiky vody a ľudské telo, ktoré sa pri sprchovaní mierne pohybuje. Za výstup môžeme považovať rozmiestenie kvapiek na tele a stenách sprchovacieho kúta po skončení. Vytvoriť presný model tohto problému a rozumne, efektívne, účinne ho vypočítať vyžaduje niekoľko človekomesiacov tvorivej práce skúseného výskumníka a konštrukciu meracích zariadení na snímanie pohybu vody a tela, ale v praxi sa namiesto presného modelu používa omnoho lacnejšia **imitácia riešenia**, napr. pomocou časticových systémov (particle systems) alebo ešte jednoduchšia a celkom lacná metóda **merania**, kde sa výsledné rozloženie kvapiek ofotografuje a potom sa namapuje ako textúra. Inak povedané, nie na riešenie každého problému existuje dosť investícií, záujemcov a času. Neriešime ani problémy, ktorých výpočet presahuje kapacity všetkých terajších počítačov a nikdy nebudeme riešiť problémy nad Brennemanovou hranicou, ktoré by vyžadovali uložiť viac bitov informácií než je atómov v Slnčnej sústave. Tešíme sa na predpovedaný príchod kvantových počítačov, ktoré by mali niekedy v roku 2022 posunúť hranice našich výpočtových možností do dnes nedosiahnuteľných oblastí. Ignorujeme absolútne aj riešenie „problémov“ z pseudovedy – astrológie, numerológie, chiromantie (veštenia z rúk), ufológie... – lebo celá **pseudoveda** má v skutočnosti iba jediný problém – **ako sa zamaskovať za vedu** a zarobiť na tom peniaze. A to nie je problém pre nás, ale pre divadelníkov, podvodníkov a iluzionistov. Na fakulte sa učíme problémy riešiť

**efektívne**, čiže dostatočne rýchlo a vyhovujúco presne. Zaujímajú nás v tomto zmysle slova zaujímavé, šikovné a vtipné riešenia problémov. A týmto záverom by sa vari mohol skončiť všeobecný úvod do hocičoho, do každého predmetu na FMFI UK.

Konkrétne riešenia zaujímavých vedeckých a technických problémov sa získavajú v postupnosti piatich etáp, ktoré vyjadruje **metodika matematického modelovania**:

1. Analýza problému, ktorej výsledkom je exaktný popis modelu (napr. modelové rovnice).
2. Analýza modelu problému, ktorá poskytuje algoritmus na riešenie problému.
3. Analýza algoritmu, ktorá umožní napísať program.
4. Výpočet zvolených úloh, na výstupe získame údaje-výsledky.
5. Analýza výsledkov, ktorej syntézou by malo byť riešenie problému.

Z každej nasledujúcej etapy sa možno v prípade potreby vrátiť do niektorej predchádzajúcej a spresniť model, algoritmus, program, výstupné údaje... V každej z etáp treba iný spôsob myslenia a spravidla aj inú hlavu, nie vždy našu. Problém spravidla analyzujú odborníci v danej aplikačnej oblasti – fyzici, chemici, geológovia, astronómi, elektronici, ekonómi, biológovia, etológovia, lekári, demografi, psychológovia, jazykovedci, etnografi, encyklopedisti, filozofi... všeobecnejšie povedané prírodovedci, technici a spoločenskovedci. Tí takisto – v úlohe našich zákazníkov - namiesto nás analyzujú výsledky a syntetizujú ich do riešenia. Spájajú mnoho jednotlivých výsledkov do odpovede na pôvodnú otázku. Pracujú tvorivo v etapách prvej a poslednej, kde predovšetkým treba ich odbornosť, nie našu. My, matematici a informatici, spravidla začíname, keď už existuje **aplikačný model**. Výpočet experimentov zas vykonávajú počítače, len niekedy ich kontrolujeme ručným prepočítaním jednoduchších častí zložitého výpočtu. Ostávajú teda dve etapy, kde budeme celý svoj profesionálny život žiť a premýšľať. Etapy 2 a 3 – tvorba algoritmov a tvorba programov – predstavujú dve hlavné činnosti v našom myšlienkovom prekrásnom, neopakovateľne jedinečnom a tvorivom živote.

2. **Analýza modelu problému -> algoritmus.** Algoritmy vytvárajú analytici, vedci, výskumníci, vedúci programátorských tímov, ktorých bibliou bývajú knihy o algoritmoch. Hlavnou z nich je The Art of Computer Programming, monumentálne dielo Donalda Knutha. Numerici čítajú Numerical Recipes, grafici čítajú Graphic Gems. A všetci vedci a výskumníci čítajú samozrejme najnovšie články a knihy. Grafici predovšetkým z každoročných konferencií SIGGRAPH, EUROGRAPHICS, Computer Graphics Interantional, Shape Modeling International, Game Development Conference, IEEE Symposium on Visualization a mnohé iné. Algoritmizáciu mnohí pokladajú nie za vedu, ale za umenie, ako to vyjadruje aj názov Knuthovej trojzväzkovej „biblie“.

3. **Analýza algoritmu -> program.** Programy vytvárajú programátori, ktorí algoritmy a dátové štruktúry z kníh a článkov analyzujú smerom ku konštrukciám v programovacích jazykoch. Tie môžu byť siedmich druhov – imperatívne (príkazové), objektové, logické, deklaratívne... V imperatívnych, príkazových jazykoch sa algoritmus dekomponuje na 1. postupnosť príkazov, 2. vetvenie alebo 3. cyklus. Do každého príkazu možno znovu vnoriť postupnosť, vetvenie alebo cyklus. Tak vznikne štrukturovaný program. V objektovej technológii sa vytvárajú objekty a ich metódy. Prevládajúce profesionálne jazyky sú oi. C, C++, ADA, LISP, FORTRAN,

JAVA... Programátori dostávajú od analytikov špecifikáciu softveru a čítajú predovšetkým referenčné manuály o assembleri jednotlivých hardverových zariadení, vyšších programovacích jazykoch, kompilátoroch, linkeroch a knižniciach. Napríklad, grafický programátor môže ku svojej práci čítať súčasne o programovaní grafickej karty, C++ a knižnici OpenGL. Programovanie sa často označuje ako **druhá gramotnosť**, ako vyšší, aktívnejší stupeň čítania a písania textu. Pojem druhej gramotnosti zaviedol akademik Jeršov a všetci druhogramotní postupne budujú informačnú spoločnosť.

**Informačná spoločnosť** sa podľa rovnomennej fundamentálnej knihy futurológa Alvina Tofflera dá vnímať ako **tretia vlna** technológie, po vlne poľnohospodárskej (výroba potravín) a priemyselnej, industriálnej (predovšetkým výroba energie). Hlavným výrobkom tretej vlny sú a budú **informácie**. Informatici teda znamenajú približne to, čo v 50. rokoch kombajnisti. Prinášajú novú technológiu, objavujú novú organizáciu a deľbu práce, menia spoločnosť. Kombajnisti industrializovali poľnohospodárstvo, informatici informatizujú aj poľnohospodárske aj priemyselné aj svoje vlastné postupy práce. Človek v tretej vlne prestáva byť konzumentom, oddeleným od výrobcu, stáva sa prozumentom, čiže aktívne sa podieľa na tvorbe a spracovaní informácií pre svoju potrebu. Ekonomika informačnej spoločnosti sa stáva vedomostnou (knowledge-based) a najväčším zdrojom bohatstva prestávajú byť polia, bane a továrne. Najcenenejším tovarom sa stávajú objavy, softver, informácie a vedomosti, súhrnne sa označujú ako **symbolický kapitál**. Poliami, na ktorých sa rodí, baňami, v ktorých sa doluje, a fabrikami, v ktorých sa vyrába, sa stávajú tvorivé hlavy vedcov a výskumníkov. Počet informatikov, pracujúcich podľa metodiky matematického modelovania, celosvetovo narastá, ekonomika sa globalizuje, fyzický svet sa virtuálne zmenšuje, internet lidkviduje pojem TAM, lebo všetko (každú potrebnú informáciu a vedomosť) už máme TU, na obrazovke nášho počítača, ak – prirodzene – VIEME, ako na to.

Etapy 2 a 3 – tvorba algoritmov a tvorba programov – predstavujú dve hlavné činnosti v myšlienkovom krásnom, neopakovateľne jedinečnom a tvorivom živote všestranne obdivovaných novodobých kombajnistov, autorov budúcnosti - informatikov.

Kde však nájsť miesto počítačovej grafiky v metodike matematického modelovania? Akoby tam chýbali obrázky... Obrázky, zvuky a symboly tam ale sú. Výsledkami výpočtov nemusia byť iba binárne nuly a jednotky, ale akýkoľvek **multimediálny objekt**. Text, čísla, symboly, glyfy (šípky, noty, dopravné a chemické značky, emotikony...), obrázky statické aj pohyblivé, zvuky (reč, hudba, konkrétne i syntetické zvuky), ba existujú aj pokusy s výstupmi pre čuch, hmat, tepelné senzory... Pre spracovanie vizuálnej informácie sa hľadá spoločné označenie – syntéza a analýza obrazu, grafika a videnie (computer graphics and vision), spracovanie obrazu (image processing), vizuálne výpočty (visual computing).

Za pôvodcu multimediálnej prezentácie vo vyučovaní možno pokladať Jána Amosa Komenského, ktorý v knihe Orbis dictus predkladá deťom české, nemecké a latinské slová, znamenajúce ten istý význam, zobrazený na obrázku. Dnes by sme povedali, že na vstupe kombinuje symbolické a ikonické reprezentácie vo vizuálnom kanáli, ku ktorým pridáva aj akustický vstup. Na výstupe by mala podľa jeho vízie Schola ludus (Škola hrou) pribudnúť radosť z poznávania. Arthur Koestler by ju konkretizoval ako tri bisociácie (spojenia dvoch kontextov v umení, vede a humore), ktoré signalizujeme citoslovcami AH!, AHA! a HAHA! Vo vede si povieme

AHA, keď sa niečo zaujímavé dozvieme a kontext môže byť vtedy len jeden – pred a po okamihu poznania. V jednom kontexte bisociáciu nazývame asociácia a jej informatickým modelom je teleport, hypertextový skok, psychologicky združenie predstáv. Hypertext simuluje index z konca knihy, rozprestretý po celej knihe. Môžeme si dokonca pomyslieť aj na budúci bihypertext, ktorý by bol modelom bisociovania. Moderným modelom multimediálnej komunikácie sa stala prezentácia v PowerPointe a jej komunikačné bohatstvo zmapúva schéma z normy MPEG-4.

## Otvorené problémy v počítačovej grafike

Na históriu matematiky možno nazerať rôzno(z)rako. Medzníkmi môžu byť vydanie Euklidových Základov (Stocheia) s objavom axiomatického systému, Descartesov objav analytickej geometrie, objav diferenciálneho a integrálneho počtu... dôkaz Goedelovej vety o neúplnosti... až po Mandelbrotov objav fraktálnej geometrie. V roku 1900 však publikoval David Hilbert tzv. Erlangenský program, súbor 22 otvorených problémov, a v roku 2000 Claytonova nadácia súbor 8 otvorených problémov. Na vyriešenie každého z nich vypísali prémie milión dolárov. V histórii ľudského poznania teda možno vidieť najvýznamnejšie udalosti ako objavy odpovedí (riešení) alebo ako objavy otázok (problémov). Neinak sa vyvíja poznávanie v počítačovej grafike, ktorého ranú históriu po slovensky spracoval Martin Šperka. Jej vznik mnohí datujú dizertačnou prácou Ivana Sutherlanda, ktorý však v roku 1966 uverejnil prvú desaticu otvorených problémov pre práve vznikajúcu novú disciplínu.

Mnohé dnešné postupy v počítačovej grafike však poznáme aj z obdobia pred počítačmi. Prvý známy návrhár fontu bol sv. Konštantín (glagolica) s použitím dizajnových prvkov trojuholník (symbol Sv. trojice) a kruh (symbol nekonečna). Prvý autor multimedialnej prezentácie bol Ján Ámos Komenský vo svojej fenomenálnej knihe Orbis pictus (Svet v obrazoch), prvej ilustrovanej česko-nemecko-latinskej učebnice jazykov pre deti. Sledovanie lúča (ray tracing) navrhol René Descartes na objasnenie vzniku dúhy a Andrew Glassner predpovedá, že keď pre každý z lúčov čiže pre každý pixel na obrazovke budeme mať v budúcnosti jeden kvantový počítač, tak bude raytracing hotový v jedinom strojovom takte.

Povešť o jednostranovej diplomovke hovorí o tom, že môžete odovzdať jedinú stránku matematického textu, ak na nej podáte dôkaz riešenia významného otvoreného problému. Legenda o najslávnejšej diplomovke v dejinách hovorí o práci, ktorá sa po obhajobe stala štátnym tajomstvom a ihneď podľa nej začali stavať továrne na výrobu rakiet V1 a V2, ktorými neskôr bombardovali Londýn. Jej autora Wernera von Braun koncom 2. svetovej vojny uniesli do USA a téma diplomovky sa tak stala jeho osudom. Tento nemecký študent ako prvý na svete technicky vyriešil reaktívny pohon, ktorého historicky prvý prototyp vyskúšal v Bratislave na potoku v Mlynskej doline nemecký inžinier Andreas Segner a ktorého víziu pre raketovú techniku vo všeobecnosti začiatkom 20. storočia sformuloval ruský vedec Konstantin Ciolkovskij. Na Segnerove práce o dynamike kvapalín zas nadviazal Leonhard Euler, ktorého model (diferenciálnu rovnicu) rozvinuli neskôr Navier-Stokes. Riešenie ich teórie patrí dnes medzi osem otvorených problémov, na ktoré vypísala na prelome tisícročí Claytonova nadácia milióndolárové prémie. Z informatiky tam patrí jednoduchučko zapisateľný problém, či platí  $P=NP$ , ale jeho riešenie bude sotva na jednu stranu.  $P$  znamená triedu polynomiálne a  $NP$  triedu nedeterministicky polynomiálne riešiteľných problémov. Ak sa vo filme Shrek pozeráte na súboj v nehomogénnej zmesi blata a piva, tak na animáciu miešania, rozprskávania a prúdenia týchto dvoch rôznorodých kvapalín bolo treba popti inom vyriešiť sústavu parciálnych diferenciálnych rovníc, ktoré sa označujú ako Navier-Stokesove a patria vo všeobecnosti medzi najprominentnejšie otvorené problémy. Potrebné vedomosti na riešenie konkrétneho zadania tohto problému možno získať v Mlynskej doline – na FMFI UK.

Uzavrieme prehľad počítačovej grafiky práve históriou kladenia otázok, ktorá extrapoluje budúci rozvoj tejto jedinečnej disciplíny modernej informatiky. Ako poznamenáva XXX, najvýznamnejší svetoví grafici spravidla prichádzali s desaticami. V roku 1966 Ivan Sutherland ako prvý položil závažné otázky a ku odpovediam sa vrátil po štyroch desaťročiach v roku 2005. V roku objavu fraktálnej geometrie, 1977, vystúpili s novším súborom problémov Martin Newell a Jim Blinn a po ďalšom desaťročí v roku 1987 Paul Heckbert. Na panelovej diskusii na konferencii SIGGRAPH 1991 identifikovali ďalšie otvorené problémy ôsmi prominentní vizionári. Iné súbory otvorených problémov publikovali nezávislo od seba v roku 1998 Mark Levoy a Jim Blinn, vďaka svojim takmer pravidelným a mimoriadne vtipne písaným „stĺpčekom“ v časopise Computer Graphics and Applications (CGA) vari najpopulárnejší grafik. Históriu na prelome tisícročí uzatvára v roku 2000 Jim Foley, prvý autor „biblie“ počítačovej grafiky. Aj toto uzatváranie histórie kolekcií najzávažnejších otázok však platí iba relatívne. Jednak existuje nepreberné množstvo otvorených problémov v mnohých ďalších publikáciách a čas s vývojom možno ukázu, ktoré z nich sa stanú najdôležitejšími, jednak existuje ďalší zoznam „grafických“ otvorených problémov, ktorý paradoxne nenapísal grafik, ale výpočtoví geometri – David Dobkin a Seth Teller. Prihodilo sa to tak, že v roku 1985 vznikla ďalšia súčasť modernej informatiky – výpočtová geometria (computational geometry), ktorá sa s počítačovou grafikou prelína. Na FMFI UK sa konkrétne prelína tak, že „výpočtovka“ patrí medzi povinné grafické predmety a viaceré jej klasické otázky medzi štátnicové otázky. Prudko rastúca komunita vo výpočtovej geometrii dospela už ku encyklopedizácii svojho poznania a do takejto encyklopédie zozbierali David Dobkin a Seth Teller veľmi podnetný súbor otvorených problémov.