Univerzita Pavla Jozefa Šafárika v Košiciach
Prírodovedecká fakulta

ALGORITMIZÁCIA, PARALELIZÁCIA A IMPLEMENTÁCIA SPLAJN MODELOV

ANALÝZA A NÁVRH RIEŠENIA PRÁCA

Študijný odbor:	Informatika
Školiace pracovisko:	Ústav informatiky
Vedúci záverečnej práce:	doc. RNDr. Csaba Török, CSc..
Konzultant:	RNDr. Lukáš Miňo

Košice 2015

Bc. Viliam Kačala

Podpísané zadanie práce z AIS2

Abstrakt

plocha

Splajny sú dôležitá súčasť počítačovej grafiky. Jedná sa o matematický model krivky, ktorá slúži na čo „najlepšie spojenie“ konečnej množiny bodov. Termín „najlepšie spojenie“ v našom prípade znamená hladká, matematicky ľahko vyjadriteľná krivka s čo najmenším zakrivením. Využitie splajnov v grafike je veľmi široké od rôznych CAD aplikácií, v štatistike, alebo v analýze dát. Splajny existujú v mnohých formách, či už vo forme krivky v rovine, rôznych trojrozmerných telies, atď.. Táto práca si dáva za cieľ analyzovať, implementovať a porovnať rôzne postupy pri počítaní a generovaní splajnov včítane nového algoritmu pre kubickú clamped interpoláciu v dvojrozmernom priestore, teda v rovine.

Abstract

Splines are important part of computer graphics. It is a mathematical model of the curve, which is for the "best connection" of any finite set of points. The term "best connection" in this case means smooth, easily calculable mathematical curve with minimal curvature. Use of splines in graphics varies from large variety of CAD applications, statistics or in data analysing. Splines exist in many forms, whether in the form of curves in the plane, a variety of three-dimensional bodies, etc. .. This work aims to analyze, implement and compare different procedures for counting and generating splines including new algorithm for cubic clamped interpolation in two-dimensional space, ie in plane.

Úvod

Témou mojej diplomovej práce sú priestorové splajn povrchy, kde naším cieľom je preskúmať nové poznatky o splajnoch, na ktorých istý čas pracuje vedúci tejto práce doc. RNDr. Csaba Török, CSc. Výsledkom tejto práce je okrem návrhu tejto novej metódy výpočtu aj porovnanie so známymi metódami a implementácia aplikácie na vizuálne porovnávanie jednotlivých metód. Technológia v ktorej budeme tieto poznatky implementovať je Microsoft Silverlight. Jedná o veľmi schopný nástroj na tvorbu webových aplikácii s plnou podporou hardvérovej akcelerácie užívateľského prostredia. Výhodou tohto frameworku je, kedže beží na platforme Microsoft .NET, možnosť jednoduchej portácie na desktopovú prípadne mobilnú aplikáciu.

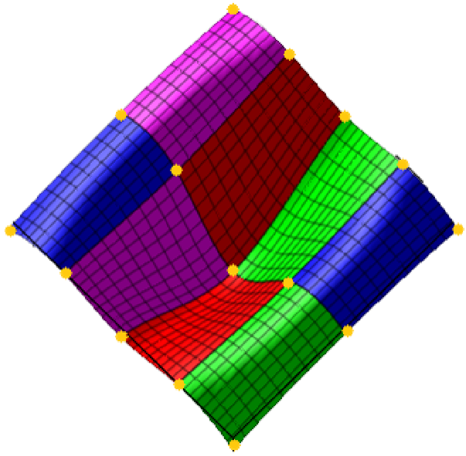
Štruktúra práce je následovná.

- **Splajn**
- **De Boorova interpolácia** V tejto časti si vysvetlíme bikubickú splajn interpoláciu podľa Carla de Boora.
- **Nový postup generovania bikubických splajnov** Ukážka modifikovaného postupu pre kubické splajny a jeho rozšírenie pre bikubické splajny.
- **Zrýchlenie** Očakávané zrýchlenie výpočtov novým algoritmom.
- **Implementácia**

  Podrobnosti implementácie v MS Silverlight.

Formálne tézy diplomovej práce sú:

- Analýza modelov interpolačných splajn-povrchov a podpory paralelného programovania. .NET Framework
- Algoritmizácia, paralelizáciaa vizualizácia splajn-povrchov.
- Implementácia paralelných algoritmov vybraných modelov spline-povrchov pomocou viacvláknových API pre multiprocesorové systémy a ich porovnanie.



Obr. 1: Funkcia $sin(\sqrt{x^2 + y^2})$ aproximovaná bikubickým splajnom.

Splajn

hermitovskymi splajnami, ktore su standardne triedy C1. V nasom pristupe hermitovske splajny budu triedy C2..

V našej práci prac[illegible]mi splajnami triedy C^2, teda so splajnami, ktorých derivácie prvého a druhého rádu sa v segmentoch rovnajú.

Formálnejšie si popíšme základné pojmy s ktorými budem v tomto článku pracovať.

Označenie 0.1 Postupnosť $(a_0, a_1, \dots)$ nazveme **rovnomernou** ak pre ľubovoľné i,j z $\{0, 1, \dots\}$ platí $a_{j+1} - a_j = a_{i+1} - a_i$.

Označenie 0.2 Nech $f : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ je nejaká funkcia a I,J sú z $\mathbb{N} \cup \{0\}$. Ďalej nech $(u_0, \dots, u_{I+1})$ a $(v_0, \dots, v_{J+1})$ sú dve rovnomerné usporiadané konečné postupnosti reálnych čísel. Označme:

- Dvojice (u_i, v_j) nazveme *uzly*.
- Uzly (u_0, v_0), (u_{I+1}, v_0), (u_0, v_{J+1}) a (u_{I+1}, v_{J+1}) nazveme *rohové uzly*.
- $z_{i,j} = f(u_i, v_j)$.
- $dx_{i,j} = \frac{\partial f(u_i,v_j)}{\partial x}$.
- $dy_{i,j} = \frac{\partial f(u_i,v_j)}{\partial y}$.

- $dxy_{i,j} = \frac{\partial f(u_i,v_j)}{\partial x \partial y}$.

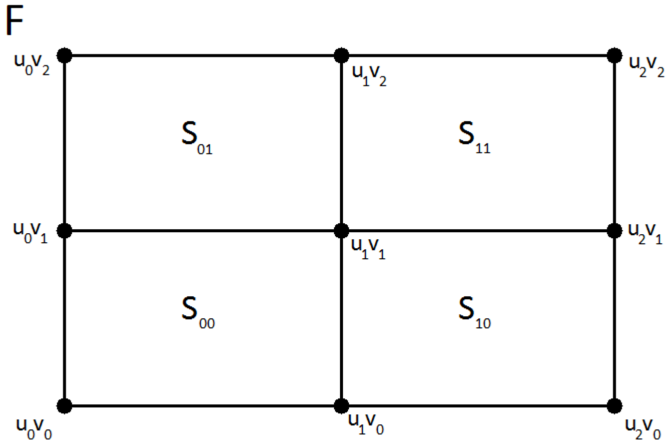
ulohou, je, na zaklade vstupnych hodnot...

- $\Delta x = u_i - u_{i-1}$ a $\Delta y = v_j - v_{j-1}$ pre ľubovoľné $i \in \{1, \ldots, I+1\}$, resp. $j \in \{1, \ldots, J+1\}$.

italic, nie uvodzovky po castiach hladky polynom S

Úlohou je nájsť „hl $, u_{I+1}] \times [v_0, v_{J+1}] \rightarrow \mathbb{R}$ z so spojitými deriváciami prvého a druhého rádu takú, že pre každé $i \in 0, \ldots, I+1$ a $j \in 0, \ldots, J+1$ platí $F(u_i, v_j) = z_{i,j} \cancel{= f(u_i, v_j)}$. Funkcia F predstavuje splajn.

F je bikvarticka funkcia a Sij su splajnove komponenty



tvoria obdlznikovy usek nad ktorym sa nachadza splajnovy segment.

. 2: Ukážka uzlov pre štvor-segmentový splajn.

Ak $i \in 0, \ldots, I$ a $j \in 0, \ldots, J$ tak štvorice uzlov (u_i, v_j), $(u_i, v_{j+1}$, $(u_{i+1}, v_j$ a $(u_{i+1}, v_{j+1}$ tvoria takzvané segmenty. Každý segment $S_{i,j}$ je bikubická funkcia z $[u_i, u_{i+1}] \times [v_j, v_{j+1}]$ do $\mathbb{R}$. Výsledná funkcia F vznikne zjednotením segmentov $S_{i,j}$. Na vygenerovanie každého segmentu potrebujeme štyri uzly, a pre každý uzol príslušne hodnoty z, dx, dy a dxy. Tieto hodnoty vieme získať buď priamo z interpolovanej funkcie f, alebo ich vypočítať iným spôsobom.

S

De Boorova interpolácia

nech su dane hodnoty uzlov

né hodnoty $(u_0, \ldots, u_{I+1})$ a $(v_0, \ldots, v_{J+1})$, kde I,J sú z $\mathbb{N} \cup \{0\}$, pričom chceme interpolovať funkciu f na $[u_0, u_I] \times [v_0, v_J]$. Výsledný splajn bude tvorený $(I+1) \cdot (J+1)$ segmentami. Ako bolo spomenuté, každý segment potrebuje štyri uzly a hodnoty z, dx, dy a dxy. Pri DeBoorovej interpolácii nám treba poznať tieto hodnoty:

- $z_{i,j}$ pre $i \in \{0, \ldots, I+1\}$, $j \in \{0, \ldots, J+1\}$.
- $dx_{i,j}$ pre $i \in \{0, I+1\}$, $j \in \{0, \ldots, J+1\}$.
- $dy_{i,j}$ pre $i \in \{0, \ldots, I+1\}$, $j \in \{0, J+1\}$.
- $dxy_{i,j}$ pre $i \in \{0, I+1\}$, $j \in \{0, J+1\}$.

Na príklade uzlov z obrázka 2 potebujeme poznať hodnoty takto:

- $z_{0,2}$, $z_{1,2}$, $z_{2,2}$,
$z_{0,1}$, $z_{1,1}$, $z_{2,1}$,
$z_{0,0}$, $z_{1,0}$, $z_{2,0}$,
- $dx_{0,2}$, , $dx_{2,2}$,
$dx_{0,1}$, , $dx_{2,1}$,
$dx_{0,0}$, , $dx_{2,0}$,
- $dy_{0,2}$, $dy_{1,2}$, $dy_{2,2}$,
, ,
$dy_{0,0}$, $dy_{1,0}$, $dy_{2,0}$,
- $dxy_{0,2}$, , $dxy_{2,2}$,
, , ,
$dxy_{0,0}$, , $dxy_{2,0}$,

Nizsie uvadzame styri modelove rovnice, pomocou kt. su zostrojene rozne pocty sustav

Zvyšné hodnoty z, dx, dy a dxy vieme jednoznačne vypočítať pomocou $2(I+1)+J+6$ lineárnych sústav o $3(I+1)(J+1)+I+J+2$ rovníc:
Pre $j \in \{0, \ldots, J+1\}$

$$dx_{i+1,j} + 4dx_{i,j} + dx_{i+1,j} = \frac{3}{\Delta x}(z_{i+1,j} - z_{i-1,j}), \qquad i \in \{1, \ldots, I\} \tag{1}$$

Pre $j \in \{0, J+1\}$

$$dxy_{i+1,j} + 4dxy_{i,j} + dxy_{i+1,j} = \frac{3}{\Delta x}(dy_{i+1,j} - dy_{i-1,j}), \qquad i \in \{1, \ldots, I\} \tag{2}$$

Pre $i \in \{0, \dots, I+1\}$

$$dy_{i,j+1} + 4dy_{i,j} + dy_{i,j-1} = \frac{3}{\Delta y}(z_{i,j+1} - z_{i,j-1}), \qquad j \in \{1, \dots, J\} \quad (3)$$

Pre $i \in \{0, \dots, I+1\}$

$$dxy_{i,j+1} + 4dxy_{i,j} + dxy_{i,j-1} = \frac{3}{\Delta y}(dx_{i,j+1} - dx_{i,j-1}), \qquad i \in \{1, \dots, J\} \quad (4)$$

Každá z týchto sústav má takýto maticový tvar:

$$\begin{pmatrix} 4 & 1 & 0 & \dots & 0 & 0 \\ 1 & 4 & 1 & \dots & 0 & 0 \\ 0 & 1 & 4 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 4 & 1 \\ 0 & 0 & 0 & \dots & 1 & 4 \end{pmatrix} \cdot \begin{pmatrix} D_1 \\ D_2 \\ D_3 \\ \vdots \\ D_{N-1} \\ D_N \end{pmatrix} = \begin{pmatrix} \frac{3}{\Delta}(Y_2 - Y_0) - D_0 \\ \frac{3}{\Delta}(Y_3 - Y_1) \\ \frac{3}{\Delta}(Y_4 - Y_2) \\ \vdots \\ \frac{3}{\Delta}(Y_N - Y_{N-2}) \\ \frac{3}{\Delta}(Y_{N+1} - Y_{N-4}) - D_{N+1} \end{pmatrix}, \quad (5)$$

hodnoty D a Y rozlisujeme takto:

kde podľa toho o ktorú z rovníc sa jedná. Nech k z $1, \dots, K-1$:

kde podľa toho o ktorú z modelovych rovníc sa jedná, hodnoty D a Y zadavame nasledovne

- $N = I$, $\Delta = \Delta x$, $D_k = dxy_{k,j}$ a $Y_k = dy_{k,j}$, pre rovnicu 2.
- $N = J$, $\Delta = \Delta y$, $D_k = dy_{i,k}$ a $Y_k = z_{i,k}$, pre rovnicu 3.
- $N = J$, $\Delta = \Delta y$, $D_k = dxy_{i,k}$ a $Y_k = dx_{i,k}$, pre rovnicu 3.

Tento splajn jednoznačne interpoluje funkciu f.

sme popisali

Poznámka 0.3 De Boorova interpolácia vo všeobecnosti nepredpokladá len rovnomerne rastúce postupnosti $(u_0, ..., u_I)$ a $(v_0, ..., v_J)$. Náš postup ale funguje len s takýmito postupnosťami. Preto v tejto časti popíšem len špeciálny prípad interpolácie pre takto určené uzly. Taktiež budeme uvažovať len trojrozmerný priestor.

splajnove krivky

popiseme redukovane system, podla Toroka

Nový postup generovania bikubických splajnov

V rámci svojej bakalárskej práce som popisoval tento postup pre kubické splajny triedy C2, teda splajny, kde interpolovaná funkcia f je typu $\mathbb{R} \to \mathbb{R}$. Jedným z cieľom tejto práce je postup zovšeobecniť pre bikubické splajny, teda pre splajny, kde interpolovaná funkcia f je typu $\mathbb{R} \times \mathbb{R} \to \mathbb{R}$. splajnove povrchy

Nižšie máme redukovaný systém rovníc používaný pre kubické splajny. Hodnoty D_0, ..., D_{K+1} sú hľadané hodnoty prvých derivácií a Y_0, ..., Y_{K+1} v prípade kubického splajnu predstavujú funkčné hodnoty interpolovanej funkcie f v uzloch u_0 až u_{K+1}. neznáme

$$\begin{pmatrix} -14 & 1 & 0 & \dots & 0 & 0 \\ 1 & -14 & 1 & \dots & 0 & 0 \\ 0 & 1 & -14 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & -14 & 1 \\ 0 & 0 & 0 & \dots & 1 & \mu \end{pmatrix} \cdot \begin{pmatrix} D_2 \\ D_4 \\ D_6 \\ \vdots \\ D_{\nu-2} \\ D_\nu \end{pmatrix} =$$

$$\begin{pmatrix} \frac{3}{\Delta}(Y_4 - Y_0) - \frac{12}{\Delta}(Y_3 - Y_1) - D_0 \\ \frac{3}{\Delta}(Y_6 - Y_2) - \frac{12}{\Delta}(Y_5 - Y_3) \\ \frac{3}{\Delta}(Y_8 - Y_4) - \frac{12}{\Delta}(Y_7 - Y_5) \\ \vdots \\ \frac{3}{\Delta}(Y_\nu - Y_{\nu-4}) - \frac{12}{\Delta}(Y_{\nu-1} - Y_{\nu-3}) \\ \frac{3}{\Delta}(Y_{\nu+\tau} - Y_{\nu-2}) - \frac{12}{\Delta}(Y_{\nu+1} - Y_{\nu-1} - \theta D_{K+1}) \end{pmatrix}, \quad (6)$$

kde

$$\begin{aligned} &\mu = -15,\ \tau = 0,\ \theta = -4,\ \text{a}\ \nu = K, && \text{ak } K \text{ je párne,} \\ &\mu = -14,\ \tau = 2,\ \theta = 1,\ \text{a}\ \nu = K - 1, && \text{ak } K \text{ je nepárne,} \end{aligned} \quad (7)$$

Hodnoty D_k pre nepárne k vypočítame takto:

$$D_k = \frac{1}{4}\left(\frac{3}{\Delta}(Y_{k+1} - Y_{k-1}) - D_{k-1} - D_{k+1}\right),\ k \in \{1, 3, \dots, \nu + \tau - 1\} \quad (8)$$

Pri kubických splajnoch teda potrebujeme vypočítať len prvé derivácie D_0, ..., D_{K+1} v uzloch u_0, ..., u_{K+1}. Pri bikubických splajnoch máme na základe de Boora 4 typy sústav, kde postupne rátame parciálne derivácie ∂x a ∂y, pričom dvomi typmi z týchto sústav vypočítame parciálne derivácie ∂xy. povrchoch

Zrýchlenie

Ako pri de Boorovi tak aj pri našom spôsobe generovania derivácií dostávame niekoľko systémov trojdiagonálnych lineárnych rovníc. Štandardný spôsob riešenia takýchto rovníc

$$\begin{pmatrix} b & 1 & 0 & \dots & 0 & 0 \\ 1 & b & 1 & \dots & 0 & 0 \\ 0 & 1 & b & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & b & 1 \\ 0 & 0 & 0 & \dots & 1 & b \end{pmatrix} \cdot \begin{pmatrix} d_1 \\ d_2 \\ d_3 \\ \vdots \\ d_{K-1} \\ d_K \end{pmatrix} = \begin{pmatrix} r1 - d_0 \\ r2 \\ r3 \\ \vdots \\ r_{K-1} \\ r_K - d_{K+1} \end{pmatrix}, \tag{9}$$

spočíva v LU dekompozícia $A\mathbf{d} = L\underbrace{U\mathbf{d}}_{\mathbf{y}} = \mathbf{r}$:

$$A = \begin{pmatrix} 1 & 0 & 0 & \dots & 0 & 0 \\ \lambda_2 & 1 & 0 & \dots & 0 & 0 \\ 0 & \lambda_3 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 1 & 0 \\ 0 & 0 & 0 & \dots & \lambda_K & 1 \end{pmatrix}, \begin{pmatrix} \upsilon_1 & 1 & 0 & \dots & 0 & 0 \\ 0 & \upsilon_1 & 1 & \dots & 0 & 0 \\ 0 & 0 & \upsilon_2 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & \upsilon_{K-1} & 0 \\ 0 & 0 & 0 & \dots & 0 & \upsilon_K \end{pmatrix}, \tag{10}$$

Pre k z $\{2, \dots, K\}$ ~~sú~~ hodnoty υ_k a λ_k sú určené (pozri **??**) takto:

$$\upsilon_k = b, \left\{ \lambda_k = \frac{1}{\upsilon_{k-1}}, \upsilon = b - \lambda_k \right\}, k \in \{2, \dots, K\}. \tag{11}$$

Pre priamy a spätný chod máme

$$\text{Priamy: } L\mathbf{y} = \mathbf{r},\ y_1 = r_1,\ \{y_k = r_k - \lambda_k\},\ k \in \{2, \dots, K\}, \tag{12}$$

$$\text{Spätný: } U\mathbf{d} = \mathbf{y},\ d_K = \frac{y_k}{u_k},\ \left\{ d_k = \frac{1}{u_k}(y_k - d_{k+1}) \right\}, k \in \{K-1, \dots, 1\}. \tag{13}$$

LU dekompozíciou sa rieši ako de Boorova sústava (5), tak aj naša redukovaná (6).

Porovnajme si počet operácií násobenia pri oboch postupoch. Podotýkam, že toto konkrétne porovnanie sa teraz týka len pri kubických splajnoch. LU dekompozícia a spätný prechod obsahujú operáciu delenia, ktorá je v tabuľke nižšie označená γ. Symbol γ udáva, že jedna operácia delenia má cenu práve γ násobení. Reálna hodnota závisí od procesora, pričom býva rovná približne 3.

Môžeme vidieť, že výpočet neznámych redukovaným algoritmom je rýchlejší než de Boorov. Zrýchlenie je $\frac{4+4\gamma}{5+2\gamma} \approx 1{,}45$, pričom hodnoty d_k, pre nepárne k (7) môžu byť počítané paralelne.

	(11)	(5), (6)	(12)	(13)	(8)		
Spôsob	**LU**	**Matica**	**LU priamy**	**LU spätný**	**Zbytok**	**Celkovo**	**Zrýchlenie**
de Boor	$\gamma(K-1)$	$1+K$	$K-1$	γK		$2(1+\gamma)K-\gamma$	
Redukovaný	$\gamma\left(\frac{K}{2}-1\right)$	$2+2\frac{K}{2}$	$\frac{K}{2}-1$	$\gamma\frac{K}{2}$	$1+2\frac{K}{2}$	$\left(\frac{5}{2}+\gamma\right)K+2-\gamma$	$\frac{4+4\gamma}{5+2\gamma}$

Tabuľka 1: Počet násobení s zrýchlenie pre K párne

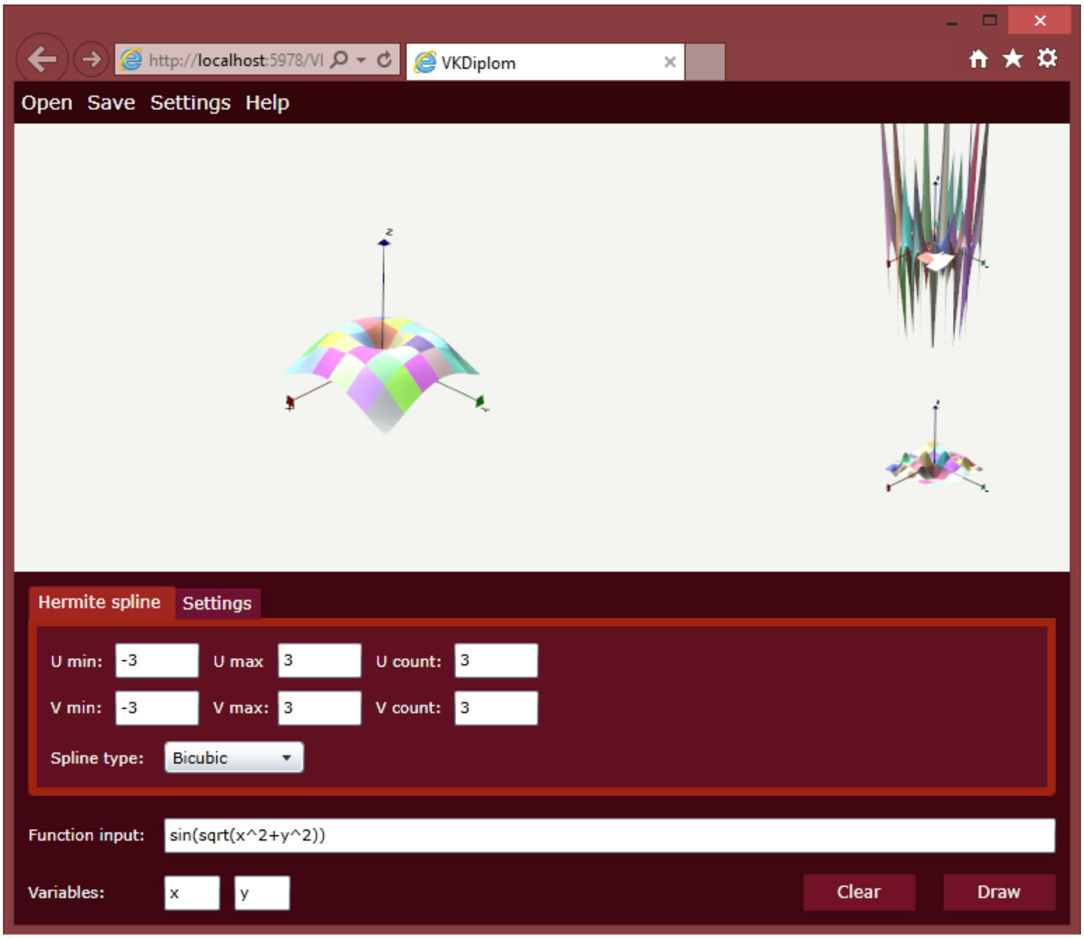
Implementácia

Naším cieľom je vytvorenie webovej aplikácie na báze Microsoft Silverlight(ďalej len Silverlight). Silverlight je bezplatný nástroj na tvorbu webových aplikácií spustiteľných priamo vo webovom prehliadači. Samotný vývoj aplikácii môže prebiehať v ľubovoľnom programovacom jazyku bežiacom pod virtuálnym strojom Common Language Runtime. V našom prípade sme siahli po jazyku C#.

Silverlight je kompatibilný s väčšinou moderných webových prehliadačov a operačných systémov včítane Microsoft Windows, Apple OS X a vďaka technológii Moonlight – open-source implementácii Silverlight-u aj na väčšine Linuxových distribúcií. Framework je možné použiť aj na vývoj off-line aplikácií v operačnom systéme Windows Phone, Windows 8 a Windows RT.

Poznámka 0.4 Naša aplikácia používa technológie, ktoré nie sú s Moonlight kompatibilné. V súčasnej dobe teda podporujeme iba MS Windows a Apple OS X.

Silverlight ponúka hardvérovo akcelerované API podporujúce tvorbu 3D grafických aplikácií a hier. V dobe písania článku je grafický engine a funkčný návrh užívateľského rozhrania hotový. Pracujem teraz na samotnej funkcionalite, ako je vizuálne odlíšenie rozdielov medzi splajnami, benchmarku generovania splajnov a rôznych dodatočných funkcionalít.



Obr. 3: Ukážka z aplikácie

Záver

Ciele tejto práce sme spolu so školiteľom splnili, nový postup sa ukázal funkčný a rýchlejší ako predchádzajúce spôsoby. V súčasnosti je webová aplikácia takmer hotová, implementovali sme všetky postupy z tejto práce pričom sa dolaďujú rôzne detaily a vylepšuje ovládanie užívateľského rozhrania. Predpokladáme, že finálna verzia by mala byť hotová v priebehu mája 2014.

Zoznam použitej literatúry

[1] David Salomon, Curves and Surfaces for Computer Graphics, Springer, 2006

[2] I. Szabó, L. Miňo, C. Török. Biquarticpolynomials in bicubicspline construction, PF UPJŠ, 2014

[3] L. Miňo: Parametrické modelovanie dát komplexnej štruktúry, PF UPJŠ, 2014

[4] C. de Boor:Bicubicspline interpolation, Journal of Mathematics and Physics, 41(3),1962, 212-218.

[5] J. Albahari, B.Albahari: C# 5.0 in a Nutshell, O'Reilly, 2012