Univerzita Pavla Jozefa Šafárika v Košiciach
Prírodovedecká fakulta

ALGORITMIZĂ˘CIA, PARALELIZĂ˘CIA A IMPLEMENTĂ˘CIA SPLAJN MODELOV

ANALĂ˝ZA A NĂ˘VRH RIEĹ˘ENIA PRÁCA

Študijný odbor:	Informatika
Školiace pracovisko:	Ăšstav informatiky
Vedúci záverečnej práce:	doc. RNDr. Csaba TĂ¶rĂ¶k, CSc..
Konzultant:	RNDr. LukĂ˘Ĺ˘ MiĹo

KoĹ˘ice 2015 Bc. Viliam KaÄŤala

Podpísané zadanie práce z AIS2

Abstrakt

Splajny sĂş dĂ´leĹľitĂ˝ sĂşÄŤasĹĄ poÄŤĂtaÄŤovej grafiky. JednĂ˝ sa o matematickĂ˝ model plochy, ktorĂ˝ slĂşĹľi na ÄŤo â€žnajlepĹˇie spojenieâ€ś koneÄŤnej mnoĹľiny bodov. TermĂn â€žnajlepĹˇie spojenieâ€ś v naĹˇom prĂpade znamenĂ˝ hladkĂ˝, matematicky Äľahko vyjadriteÄľnĂ˝ plocha s ÄŤo najmenĹˇĂm zakrivenĂm. VyuĹľitie splajnov v grafike je veÄľmi Ĺˇiroké od rĂ´znych CAD aplikĂ˝ciĂ, v Ĺˇtatistike, alebo v analĂ˝ze dĂ˝t. Splajny existujĂş v mnohĂ˝ch formĂ˝ch, ÄŤi uĹľ vo forme krivky v rovine, rĂ´znych trojrozmernĂ˝ch telies, atÄŹ.. TĂ˝to prĂ˝ca si dĂ˝va za cieÄľ navrhnĂşĹĄ, analyzovaĹĄ a implementovaĹĄ novĂ˝ algoritmus pre bikubickĂş interpolĂ˝ciu v trojrozmernom priestore.

Abstract

Splines are important part of computer graphics. It is a mathematical model of the surface, which is for the "best connection" of any finite set of points. The term "best connection" in this case means smooth, easily calculable mathematical surface 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 desig, analyze and implement new algorithm for counting and generating splines bicubic clamped interpolation in three-dimensional space.

Ăšvod

TĂ©mou mojej diplomovej prĂˇce sĂş priestorovĂ© splajn povrchy, kde naĹˇĂm cieÄľom je preskĂşmaĹĄ novĂ© poznatky o splajnoch, na ktorĂ˝ch istĂ˝ ÄŤas pracuje vedĂşci tejto prĂˇce doc. RNDr. Csaba TĂ¶rĂ¶k, CSc. VĂ˝sledkom tejto prĂˇce je okrem nĂˇvrhu tejto novej metĂłdy vĂ˝poÄŤtu aj porovnanie so znĂˇmymi metĂłdami a implementĂˇcia aplikĂˇcie na vizuĂˇlne porovnĂˇvanie jednotlivĂ˝ch metĂłd. TechnolĂłgia v ktorej budeme tieto poznatky implementovaĹĄ je Microsoft Silverlight. JednĂˇ o veÄľmi schopnĂ˝ nĂˇstroj na tvorbu webovĂ˝ch aplikĂˇcii s plnou podporou hardvĂ©rovej akcelerĂˇcie uĹľĂvateÄľskĂ©ho prostredia. VĂ˝hodou tohto frameworku je, kedĹľe beĹľĂ na platforme Microsoft .NET, moĹľnosĹĄ jednoduchej portĂˇcie na desktopovĂş prĂpadne mobilnĂş aplikĂˇciu. Ĺ truktĂşra prĂˇce je nasledovnĂˇ.

- **Splajn** DefinĂcia pojmu splajn.
- **De Boorova interpolĂˇcia** V tejto ÄŤasti si vysvetlĂme bikubickĂş splajn interpolĂˇciu podÄľa Carla de Boora.
- **NovĂ˝ postup generovania bikubickĂ˝ch splajnov** UkĂˇĹľka modifikovanĂ©ho postupu pre kubickĂ© splajny a jeho rozĹˇĂrenie pre bikubickĂ© splajny.
- **ZrĂ˝chlenie** OÄŤakĂˇvanĂ© zrĂ˝chlenie vĂ˝poÄŤtov novĂ˝m algoritmom.
- **ImplementĂˇcia**

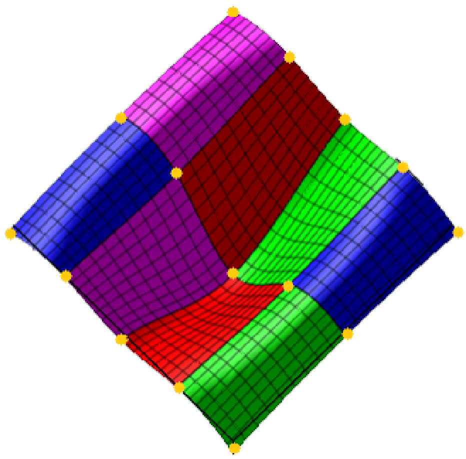
  Podrobnosti implementĂˇcie v MS Silverlight.

FormĂˇlne tĂ©zy diplomovej prĂˇce sĂş:

- AnalĂ˝za modelov interpolaÄŤnĂ˝ch splajn-povrchov a podpory paralelnĂ©ho programovania. .NET Framework
- AlgoritmizĂˇcia, paralelizĂˇciaa vizualizĂˇcia splajn-povrchov.
- ImplementĂˇcia paralelnĂ˝ch algoritmov vybranĂ˝ch modelov splajn-povrchov pomocou viacvlĂˇknovĂ˝ch API pre multiprocesorovĂ© systĂ©my a ich porovnanie.

Splajn

V naĹˇej prĂˇci pracujeme s hermitovskĂ˝mi splajnami, ktorĂ© sĂş Ĺˇtandardne triedy C^1, teda splajnami, ktorĂ˝ch prvĂ© derivĂˇcie v uzloch sa rovnajĂş. V naĹˇej prĂˇci vĹˇak budeme uvaĹľovaĹĄ hermitovskĂ© splajny triedy C^2, teda splajny pri ktorĂ˝ch sa v uzloch rovnajĂş navyĹˇe aj derivĂˇcie druhĂ©ho rĂˇdu.



Obr. 1: Funkcia $sin(\sqrt{x^2+y^2})$ aproximovanĂˇ bikubickĂ˝m splajnom.

FormĂˇlnejĹˇie si popĂĹˇme zĂˇkladnĂ© pojmy s ktorĂ˝mi budem v tomto ÄŤlĂˇnku pracovaĹĄ.

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

Označenie 0.2 Nech $f : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ je nejakĂˇ funkcia a I,J sĂş z $\mathbb{N} \cup \{0\}$. ÄŽalej nech $(u_0, \dots, u_{I+1})$ a $(v_0, \dots, v_{J+1})$ sĂş dve rovnomernĂ© usporiadanĂ© koneÄŤnĂ© postupnosti reĂˇlnych ÄŤĂsel. OznaÄŤme:

- 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 *rohovĂ© uzly.*

- $z_{i,j} = f(u_i, v_j)$.
- $dx_{i,j} = \frac{\partial f(u_i,v_j)}{\partial x}$.
- $dy_{i,j} = \frac{\partial f(u_i,v_j)}{\partial y}$.
- $dxy_{i,j} = \frac{\partial f(u_i,v_j)}{\partial x \partial y}$.
- $\Delta x = u_i - u_{i-1}$ a $\Delta y = v_j - v_{j-1}$ pre ÄľubovoÄľnĂ© $i \in \{1, \ldots, I+1\}$, resp. $j \in \{1, \ldots, J+1\}$.

Ăšlohou je na zĂˇklade vstupnĂ˝ch uzlov nĂˇjsĹĄ *hladkĂş*, po ÄŤastiach definovanĂş funkciu $F : [u_0, u_{I+1}] \times [v_0, v_{J+1}] \to \mathbb{R}$ z so spojitĂ˝mi derivĂˇciami prvĂ©ho a druhĂ©ho rĂˇdu takĂş, Ĺľe pre kaĹľdĂ© $i \in 0, \ldots, I+1$ a $j \in 0, \ldots, J+1$ platĂ $F(u_i, v_j) = z_{i,j}$. Funkcia F predstavuje splajn, priÄŤom jej jednotlivĂ© ÄŤasti nazveme segmenty. Teraz si mĂ´Ĺľeme splajn zadefinovaĹĄ formĂˇlne.

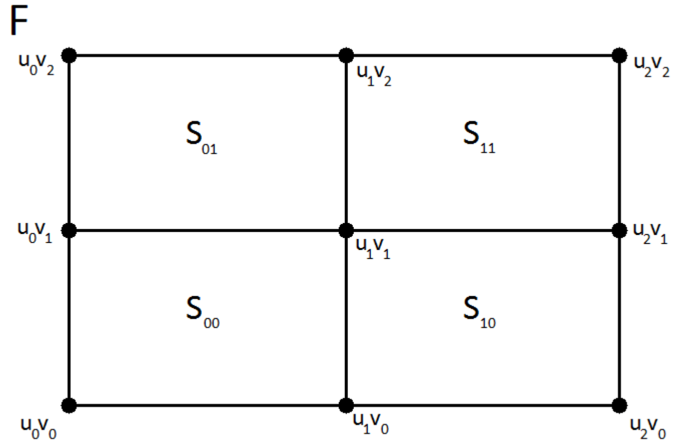
Definícia 0.3 Nech I a J sĂş prirodzenĂ© ÄŤĂsla, $\mathbf{u} = (u_0, \ldots, u_{I+1})$ a $\mathbf{v} = (v_0, \ldots, v_{J+1})$ sĂş reĂˇlne ÄŤĂsla.

- $I \geq 0$ a $J \geq 0$,
- Postupnosti $\mathbf{u}$ a $\mathbf{v}$ sĂş rovnomerne rastĂşce.
- Nech pre kaĹľdĂ© i z $\{0, \ldots, I\}$ a j z $\{0, \ldots, J\}$ je $S_{i,j}$ bikubickĂˇ polynomickĂˇ funkcia z 2-rozmernĂ©ho intervalu $[u_i, u_{i+1}] \times [v_j, v_{j+1}]$ do $\mathbb{R}$.

Definujme funkciu $S_{U,V}$ z 2-rozmernĂ©ho intervalu $[u_0, u_{I+1}] \times [v_0, v_{J+1}]$ do $\mathbb{R}$ vzĹĄahom:

$$S_{\mathbf{u},\mathbf{v}}(x,y) = \begin{cases} S_{0,0}(x,y) & \text{pre } (x,y) \in [u_0,u_1] \times [v_0,v_1], \\ S_{0,1}(x,y) & \text{pre } (x,y) \in [u_0,u_1] \times [v_1,v_2], \\ \vdots & \\ S_{0,J}(x,y) & \text{pre } (x,y) \in [u_0,u_1] \times [v_J,v_{J+1}], \\ S_{1,0}(x,y) & \text{pre } (x,y) \in [u_1,u_2] \times [v_0,v_1], \\ \vdots & \\ S_{1,J}(x,y) & \text{pre } (x,y) \in [u_1,u_2] \times [v_J,v_{J+1}], \\ \vdots & \\ S_{I,0}(x,y) & \text{pre } (x,y) \in [u_I,u_{I+1}] \times [v_0,v_1], \\ \vdots & \\ S_{I,J}(x,y) & \text{pre } (x,y) \in [u_I,u_{I+1}] \times [v_J,v_{J+1}], \end{cases} \tag{1}$$

priÄŤom vĹˇetky susednĂ© segmenty $S_{i,j}$ majĂş derivĂˇcie prvĂ©ho a druhĂ©ho rĂˇdu v uzloch navzĂˇjom rovnĂ©.



Obr. 2: UkĂˇĹľka uzlov pre ĹˇtvorsegmentovĂ˝ splajn.

Ak i je z $\{0,\dots,I\}$ a j je z $\{0,\dots,J\}$, tak Ĺˇtvorice uzlov (u_i,v_j), (u_i,v_{j+1}), (u_{i+1},v_j) a $(u_{i+1},v_{j+1}$ tvoria obdĂşĹľnikovĂ˝ Ăşsek nad ktorĂ˝m sa nachĂˇdza splajnovĂ˝ segment. KaĹľdĂ˝ segment $S_{i,j}$ je bikubickĂˇ funkcia z $[u_i,u_{i+1}]\times[v_j,v_{j+1}]$ do $\mathbb{R}$. VĂ˝slednĂˇ funkcia F teda vznikne zjednotenĂm segmentov $S_{i,j}$. Na vygenerovanie kaĹľdĂ©ho segmentu potrebujeme Ĺˇtyri uzly, a pre kaĹľdĂ˝ uzol prĂsluĹˇne hodnoty z, dx, dy a dxy. Tieto hodnoty vieme zĂskaĹĄ buÄŹ priamo z interpolovanej funkcie f, alebo ich vypoÄŤĂtaĹĄ inĂ˝m spĂ´sobom.

De Boorova interpolĂˇcia

Nech sĂş danĂ© danĂ© hodnoty $(u_0, ...,u_{I+1})$ a $(v_0, ...,v_{J+1})$, kde I,J sĂş z $\mathbb{N}\cup\{0\}$, priÄŤom chceme interpolovaĹĄ funkciu S na $[u_0,u_I]\times[v_0,v_J]$. VĂ˝slednĂ˝ splajn bude tvorenĂ˝ $(I+1)\cdot(J+1)$ segmentami. Ako bolo spomenutĂ©, kaĹľdĂ˝ segment potrebuje Ĺˇtyri uzly a hodnoty z, dx, dy a dxy. Pri de Boorovej interpolĂˇcii nĂˇm treba poznaĹĄ tieto hodnoty:

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

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

Na prĂklade uzlov z obrĂˇzka 2 potrebujeme poznaĹĄ hodnoty takto:

- $z_{0,2},\ z_{1,2},\ z_{2,2},$
  $z_{0,1},\ z_{1,1},\ z_{2,1},$
  $z_{0,0},\ z_{1,0},\ z_{2,0},$

- $dx_{0,2},\qquad ,\ dx_{2,2},$
  $dx_{0,1},\qquad ,\ dx_{2,1},$
  $dx_{0,0},\qquad ,\ dx_{2,0},$

- $dy_{0,2},\ dy_{1,2},\ dy_{2,2},$
  $\qquad ,\qquad ,$
  $dy_{0,0},\ dy_{1,0},\ dy_{2,0},$

- $dxy_{0,2},\qquad ,\ dxy_{2,2},$
  $\qquad ,\qquad ,\qquad ,$
  $dxy_{0,0},\qquad ,\ dxy_{2,0},$

ZvyĹˇnĂ© hodnoty z, dx, dy a dxy vieme jednoznaÄŤne vypoÄŤĂtaĹĄ pomocou $2(I+1)+J+6$ lineĂˇrnych sĂşstav s celkovo $3(I+1)(J+1)+I+J+2$ rovnicami: NiĹľĹˇie uvĂˇdzame modelovĂ© rovnice, pomocou ktorĂ˝ch sĂş zostrojenĂ© tieto sĂşstavy lineĂˇrnych rovnĂc.
Pre $j \in \{0, \ldots, J+1\}$

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

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

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

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

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

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

$$dxy_{i,j+1} + 4dxy_{i,j} + dxy_{i,j-1} = \frac{3}{\Delta y}(dx_{i,j+1} - dx_{i,j-1}),$$
$$i \in \{1, \ldots, J\} \quad (5)$$

KaĹľdĂˇ z tĂ˝chto sĂşstav mĂˇ takĂ˝to maticovĂ˝ tvar:

$$\begin{pmatrix} 4 & 1 & 0 & \ldots & 0 & 0 \\ 1 & 4 & 1 & \ldots & 0 & 0 \\ 0 & 1 & 4 & \ldots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \ldots & 4 & 1 \\ 0 & 0 & 0 & \ldots & 1 & 4 \end{pmatrix} \cdot \begin{pmatrix} D_1 \\ D_2 \\ D_3 \\ \vdots \\ D_{N-1} \\ D_N \end{pmatrix} = \begin{pmatrix} \frac{3}{\Delta}(Y_2 - Y_0) - D_0 \\ \frac{3}{\Delta}(Y_3 - Y_1) \\ \frac{3}{\Delta}(Y_4 - Y_2) \\ \vdots \\ \frac{3}{\Delta}(Y_N - Y_{N-2}) \\ \frac{3}{\Delta}(Y_{N+1} - Y_{N-4}) - D_{N+1} \end{pmatrix}, \quad (6)$$

kde podÄľa toho o ktorĂş z modelovĂ˝ch rovnĂc sa jednĂˇ, hodnoty N, D a Y zadĂˇvame nĂˇsledovne. Nech k z $1, \ldots, K-1$:

- $N = I$, $\Delta = \Delta x$, $D_k = dx_{k,j}$ a $Y_k = z_{k,j}$, pre rovnicu 2.
- $N = I$, $\Delta = \Delta x$, $D_k = dxy_{k,j}$ a $Y_k = dy_{k,j}$, pre rovnicu 3.
- $N = J$, $\Delta = \Delta y$, $D_k = dy_{i,k}$ a $Y_k = z_{i,k}$, pre rovnicu 4.
- $N = J$, $\Delta = \Delta y$, $D_k = dxy_{i,k}$ a $Y_k = dx_{i,k}$, pre rovnicu 4.

Tento splajn jednoznaÄŤne interpoluje funkciu S.

Poznámka 0.4 De Boorova interpolĂˇcia vo vĹˇeobecnosti nepredpokladĂˇ len rovnomerne rastĂşce postupnosti $(u_0, ..., u_I)$ a $(v_0, ..., v_J)$. NĂˇĹˇ postup v ÄŹalĹˇej ÄŤasti ÄŤlĂˇnku ale funguje len s takĂ˝mito postupnosĹĄami. Preto sme v tejto ÄŤasti popĂsali len ĹˇpeciĂˇlny prĂpad interpolĂˇcie pre takto urÄŤenĂ© uzly.

NovĂ˝ postup generovania bikubickĂ˝ch splajnov

V rĂˇmci svojej bakalĂˇrskej prĂˇce som popisoval tento postup pre kubickĂ© splajnovĂ© kriovky triedy C2, teda splajny, kde interpolovanĂˇ funkcia f je typu $\mathbb{R} \to \mathbb{R}$. JednĂ˝m z cieÄľom tejto prĂˇce je postup zovĹˇeobecniĹĄ pre bikubickĂ© splajny, teda pre splajny, kde interpolovanĂˇ funkcia f je typu $\mathbb{R} \times \mathbb{R} \to \mathbb{R}$.

NiĹľĹˇie popĂĹˇeme redukovanĂ˝ systĂ©m podÄľa doc. TĂ¶rĂ¶ka pouĹľĂvanĂ˝ pre kubickĂ© splajny. NeznĂˇme D_0, ..., D_{K+1} sĂş hÄľadanĂ© hodnoty prvĂ˝ch derivĂˇciĂ a Y_0, ..., Y_{K+1}, v prĂpade kubickĂ©ho splajnu, predstavujĂş funkÄŤnĂ© hodnoty interpolovanej funkcie f v uzloch u_0 aĹľ u_{K+1}.

$$
\begin{pmatrix}
-14 & 1 & 0 & \dots & 0 & 0 \\
1 & -14 & 1 & \dots & 0 & 0 \\
0 & 1 & -14 & \dots & 0 & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & 0 & \dots & -14 & 1 \\
0 & 0 & 0 & \dots & 1 & \mu
\end{pmatrix}
\cdot
\begin{pmatrix}
D_2 \\ D_4 \\ D_6 \\ \vdots \\ D_{v-2} \\ D_v
\end{pmatrix}
=
\begin{pmatrix}
\frac{3}{\Delta}(Y_4 - Y_0) - \frac{12}{\Delta}(Y_3 - Y_1) - D_0 \\
\frac{3}{\Delta}(Y_6 - Y_2) - \frac{12}{\Delta}(Y_5 - Y_3) \\
\frac{3}{\Delta}(Y_8 - Y_4) - \frac{12}{\Delta}(Y_7 - Y_5) \\
\vdots \\
\frac{3}{\Delta}(Y_\nu - Y_{\nu-4}) - \frac{12}{\Delta}(Y_{\nu-1} - Y_{\nu-3}) \\
\frac{3}{\Delta}(Y_{\nu+\tau} - Y_{\nu-2}) - \frac{12}{\Delta}(Y_{\nu+1} - Y_{\nu-1} - \theta D_{K+1})
\end{pmatrix}, \quad (7)
$$

kde

$$
\begin{aligned}
&\mu = -15,\ \tau = 0,\ \theta = -4,\ \text{a}\ \nu = K, && \text{ak } K \text{ je pĂˇrne,} \\
&\mu = -14,\ \tau = 2,\ \theta = 1,\ \text{a}\ \nu = K - 1, && \text{ak } K \text{ je nepĂˇrne,}
\end{aligned} \quad (8)
$$

Hodnoty D_k pre nepĂˇrne k vypoÄŤĂtame takto:

$$
D_k = \frac{1}{4}\left(\frac{3}{\Delta}(Y_{k+1} - Y_{k-1}) - D_{k-1} - D_{k+1}\right),\ k \in \{1, 3, \dots, \nu + \tau - 1\} \quad (9)
$$

Pri kubickĂ˝ch splajnoch teda potrebujeme vypoÄŤĂtaĹĄ len prvĂ© derivĂˇcie D_0, ..., D_{K+1} v uzloch u_0, ..., u_{K+1}. Pri bikubickĂ˝ch povrchoch mĂˇme na zĂˇklade de Boora 4 typy sĂşstav, kde postupne rĂˇtame parciĂˇlne derivĂˇcie ∂x a ∂y, priÄŤom dvomi typmi z tĂ˝chto sĂşstav vypoÄŤĂtame parciĂˇlne derivĂˇcie ∂xy.

ZrĂ˝chlenie

Ako pri de Boorovi tak aj pri naĹˇom spĂ´sobe generovania derivĂˇciĂ- dostĂˇvame niekoÄľko systĂ©mov trojdiagonĂˇlnych lineĂˇrnych rovnĂc.

Ĺ tandardnĂ˝ spĂ´sob rieĹˇenia takĂ˝chto rovnĂc

$$\begin{pmatrix} b & 1 & 0 & \dots & 0 & 0 \\ 1 & b & 1 & \dots & 0 & 0 \\ 0 & 1 & b & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & b & 1 \\ 0 & 0 & 0 & \dots & 1 & b \end{pmatrix} \cdot \begin{pmatrix} d_1 \\ d_2 \\ d_3 \\ \vdots \\ d_{K-1} \\ d_K \end{pmatrix} = \begin{pmatrix} r1 - d_0 \\ r2 \\ r3 \\ \vdots \\ r_{K-1} \\ r_K - d_{K+1} \end{pmatrix}, \tag{10}$$

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

$$A = \begin{pmatrix} 1 & 0 & 0 & \dots & 0 & 0 \\ \lambda_2 & 1 & 0 & \dots & 0 & 0 \\ 0 & \lambda_3 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 1 & 0 \\ 0 & 0 & 0 & \dots & \lambda_K & 1 \end{pmatrix}, \begin{pmatrix} \upsilon_1 & 1 & 0 & \dots & 0 & 0 \\ 0 & \upsilon_1 & 1 & \dots & 0 & 0 \\ 0 & 0 & \upsilon_2 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & \upsilon_{K-1} & 0 \\ 0 & 0 & 0 & \dots & 0 & \upsilon_K \end{pmatrix}, \tag{11}$$

Pre k z $\{2,\dots,K\}$ sĂş hodnoty υ_k a λ_k urÄŤenĂ© takto:

$$\upsilon_k = b, \left\{ \lambda_k = \frac{1}{\upsilon_{k-1}}, \upsilon = b - \lambda_k \right\}, k \in \{2,\dots,K\}. \tag{12}$$

Pre priamy a spĂtnĂ˝ chod mĂˇme

$$\text{Priamy: } L\mathbf{y} = \mathbf{r},\ y_1 = r_1,\ \{y_k = r_k - \lambda_k\},\ k \in \{2,\dots,K\}, \tag{13}$$

$$\text{SpĂtnĂ˝: } U\mathbf{d} = \mathbf{y},\ d_K = \frac{y_k}{u_k},\ \left\{ d_k = \frac{1}{u_k}(y_k - d_{k+1}) \right\}, k \in \{K-1,\dots,1\}. \tag{14}$$

LU dekompozĂciou sa rieĹˇi ako de Boorova sĂşstava (6), tak aj naĹˇa redukovanĂˇ (7).

Porovnajme si poÄŤet operĂˇciĂ nĂˇsobenia pri oboch postupoch. PodotĂ˝kam, Ĺľe toto konkrĂ©tne porovnanie sa teraz tĂ˝ka len pri kubickĂ˝ch splajnoch. LU dekompozĂcia a spĂtnĂ˝ prechod obsahujĂş operĂˇciu delenia, ktorĂˇ je v tabuÄľke niĹľĹˇie oznaÄŤenĂˇ γ. Symbol γ udĂˇva, Ĺľe jedna operĂˇcia delenia mĂˇ cenu prĂˇve γ nĂˇsobenĂ. ReĂˇlna hodnota zĂˇvisĂ od procesora, priÄŤom bĂ˝va rovnĂˇ pribliĹľne 3.

MĂ´Ĺľeme vidieĹĄ, Ĺľe vĂ˝poÄŤet neznĂˇmych redukovanĂ˝m algoritmom je rĂ˝chlejĹˇĂ neĹľ de Boorov. ZrĂ˝chlenie je $\frac{4+4\gamma}{5+2\gamma} \approx 1{,}45$, priÄŤom hodnoty d_k, pre nepĂˇrne k (8) mĂ´Ĺľu byĹĄ poÄŤĂtanĂ© paralelne.

	(12)	(6), (7)	(13)	(14)	(9)		
SpĂ´sob	**LU**	**Matica**	**LU priamy**	**LU spĂtnĂ˝**	**Zbytok**	**Celkovo**	**ZrĂ˝chlenie**
de Boor	$\gamma(K-1)$	$1+K$	$K-1$	γK		$2(1+\gamma)K-\gamma$	
RedukovanĂ˝	$\gamma\left(\frac{K}{2}-1\right)$	$2+2\frac{K}{2}$	$\frac{K}{2}-1$	$\gamma\frac{K}{2}$	$1+2\frac{K}{2}$	$\left(\frac{5}{2}+\gamma\right)K+2-\gamma$	$\frac{4+4\gamma}{5+2\gamma}$

Tabuľka 1: PoÄŤet nĂˇsobenĂ s zrĂ˝chlenie pre K pĂˇrne

ImplementĂˇcia

NaĹˇĂm cieÄľom je vytvorenie webovej aplikĂˇcie na bĂˇze Microsoft Silverlight(ÄŹalej len Silverlight). Silverlight je bezplatnĂ˝ nĂˇstroj na tvorbu webovĂ˝ch aplikĂˇciĂ spustiteÄľnĂ˝ch priamo vo webovom prehliadaÄŤi. SamotnĂ˝ vĂ˝voj aplikĂˇcii mĂ´Ĺľe prebiehaĹĄ v ÄľubovoÄľnom programovacom jazyku beĹľiacom pod virtuĂˇlnym strojom Common Language Runtime. V naĹˇom prĂpade sme siahli po jazyku C#.

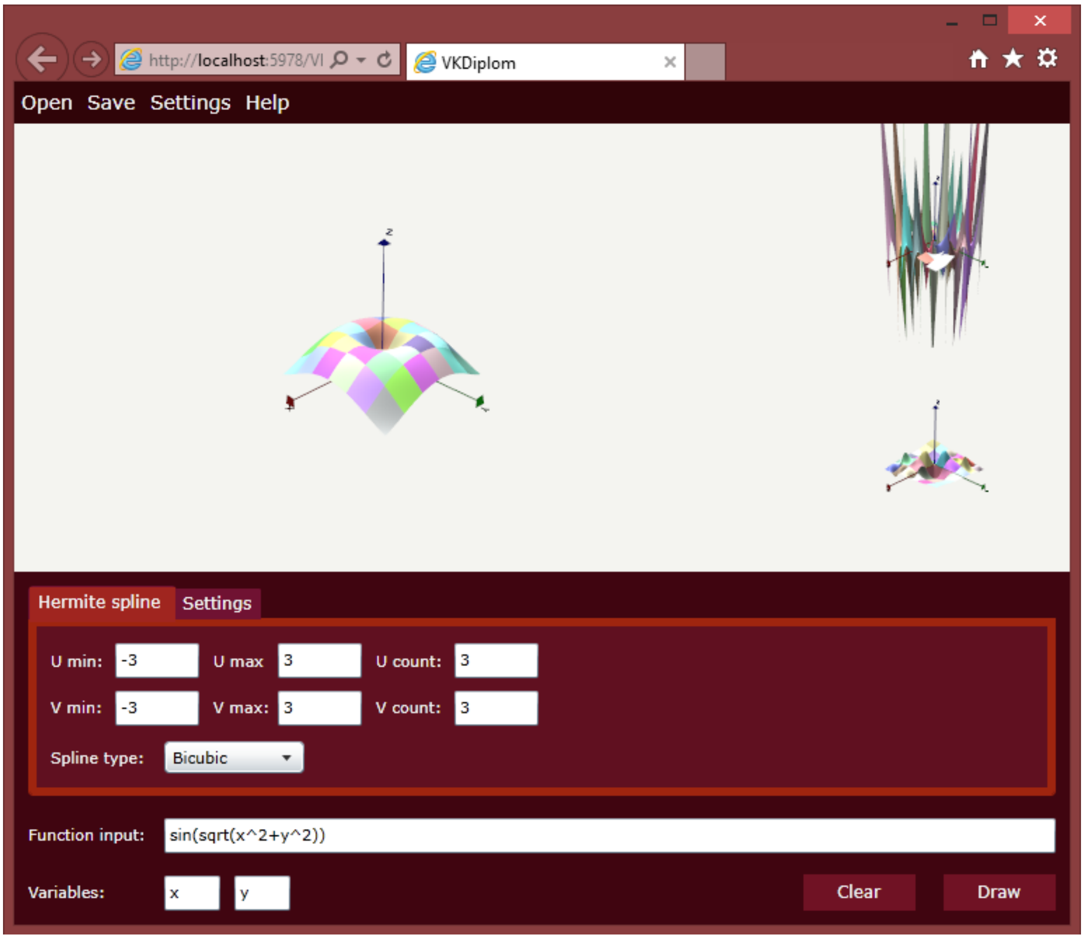
Silverlight je kompatibilnĂ˝ s vĂ¤ÄŤĹˇinou modernĂ˝ch webovĂ˝ch prehliadaÄŤov a operaÄŤnĂ˝ch systĂ©mov vÄŤĂtane Microsoft Windows, Apple OS X a vÄŹaka technolĂłgii Moonlight – open-source implementĂˇcii Silverlight-u aj na vĂ¤ÄŤĹˇine LinuxovĂ˝ch distribĂşciĂ. Framework je moĹľnĂ© pouĹľiĹĄ aj na vĂ˝voj off-line aplikĂˇciĂ v operaÄŤnom systĂ©me Windows Phone, Windows 8 a Windows RT.

Poznámka 0.5 NaĹˇa aplikĂˇcia pouĹľĂva technolĂłgie, ktorĂ© nie sĂş s Moonlight kompatibilnĂ©. V sĂşÄŤasnej dobe teda podporujeme iba MS Windows a Apple OS X.

Silverlight ponĂşka hardvĂ©rovo akcelerovanĂ© API podporujĂşce tvorbu 3D grafickĂ˝ch aplikĂˇciĂ a hier. V dobe pĂsania ÄŤlĂˇnku je grafickĂ˝ engine a funkÄŤnĂ˝ nĂˇvrh uĹľĂvateÄľskĂ©ho rozhrania hotovĂ˝. Pracujem teraz na samotnej funkcionalite, ako je vizuĂˇlne odlĂĹˇenie rozdielov medzi splajnami, benchmarku generovania splajnov a rĂ´znych dodatoÄŤnĂ˝ch funkcionalĂt.

ZĂˇver

Ciele tejto prĂˇce sme spolu so ĹˇkoliteÄľom splnili, novĂ˝ postup sa ukĂˇzal funkÄŤnĂ˝ a rĂ˝chlejĹˇĂ ako predchĂˇdzajĂşce spĂ´soby. V sĂşÄŤas-



Obr. 3: UkĂˇĹľka z aplikĂˇcie

nosti je webovĂˇ aplikĂˇcia takmer hotovĂˇ, implementovali sme vĹˇetky postupy z tejto prĂˇce priÄŤom sa dolaÄŹujĂş rĂ´zne detaily a vylepĹˇuje ovlĂˇdanie uĹľĂvateÄľskĂ©ho rozhrania. PredpokladĂˇme, Ĺľe finĂˇlna verzia by mala byĹĄ hotovĂˇ v priebehu mĂˇja 2014.

Zoznam použitej literatúry

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

[2] I. SzabĂł, L. MiĹo, C. TĂ¶rĂ¶k. Biquarticpolynomials in bicubicspline construction, PF UPJĹ , 2014

[3] L. MiĹo: ParametrickĂ© modelovanie dĂˇt komplexnej ĹˇtruktĂşry, PF UPJĹ , 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