Transformace obrazu

Pavel Strachota

FJFI ČVUT v Praze

16. listopadu 2012

Obsah

Obsah

Interpolace v 1D

- množina n diskrétních vzorků hodnot funkce (signálu) $f : [a,b] \mapsto \mathbb{R}$ na intervalu $[a,b]$:

$$\left\{ (x_i, f(x_i)) \middle| \, i \in \mathbb{N}_n, x_i \in [a,b] \right\}$$

- **interpolace** - konstrukce funkce $\tilde{f} : [a,b] \mapsto R$ pouze ze znalosti $(x_i, f(x_i))$ tak, aby

$$\tilde{f}(x_i) = f(x_i) \ \forall i \in \mathbb{N}_n \tag{1}$$

- **význam**: snaha **rekonstruovat** přibližnou hodnotu $f(x)$ pro každé $x \in [a,b]$ pomocí $\tilde{f}(x)$
- **aproximace** - obecná rekonstrukce hodnot funkce f bez podmínky (1)

Poznámka: $\mathbb{N}_n = \{k \in \mathbb{Z} | 0 \leq k < n\} = \{0, 1, 2, \ldots, n-1\}$

Interpolace v 1D

Interpolace nejbližším sousedem

- jinak též **po částech konstantní interpolace**
- obvykle *ekvidistantní* rozdělení intervalu

$$x_i = a + ih$$

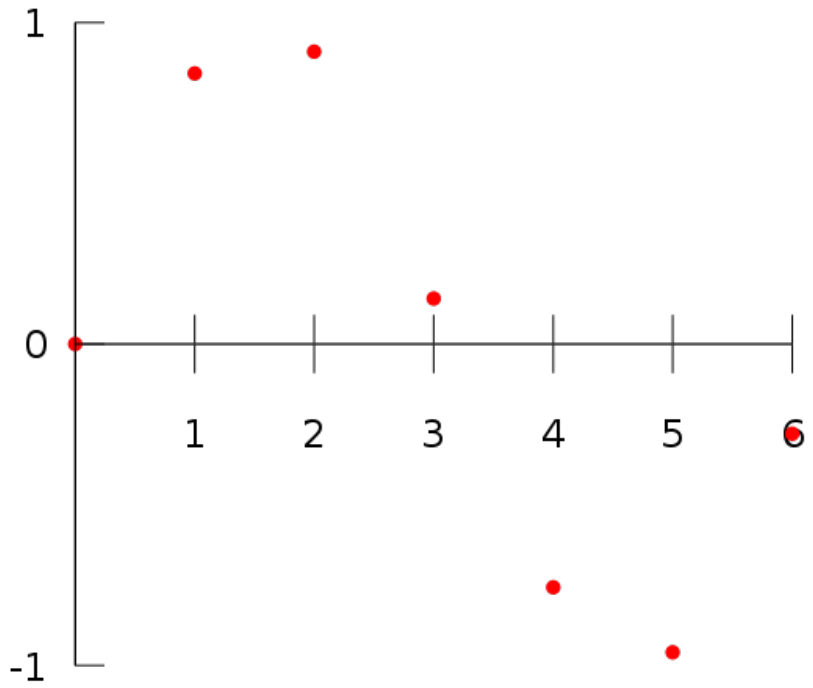
kde

$$h = \frac{(b-a)}{n-1}$$

- definujeme

$$\tilde{f}(x) = f(x_i)$$

$$\forall x \in \left(\left(x_i - \frac{h}{2}, x_i + \frac{h}{2}\right) \cap [a,b]\right)$$



Interpolace v 1D

Interpolace nejbližším sousedem

- jinak též **po částech konstantní interpolace**
- obvykle *ekvidistantní* rozdělení intervalu

$$x_i = a + ih$$

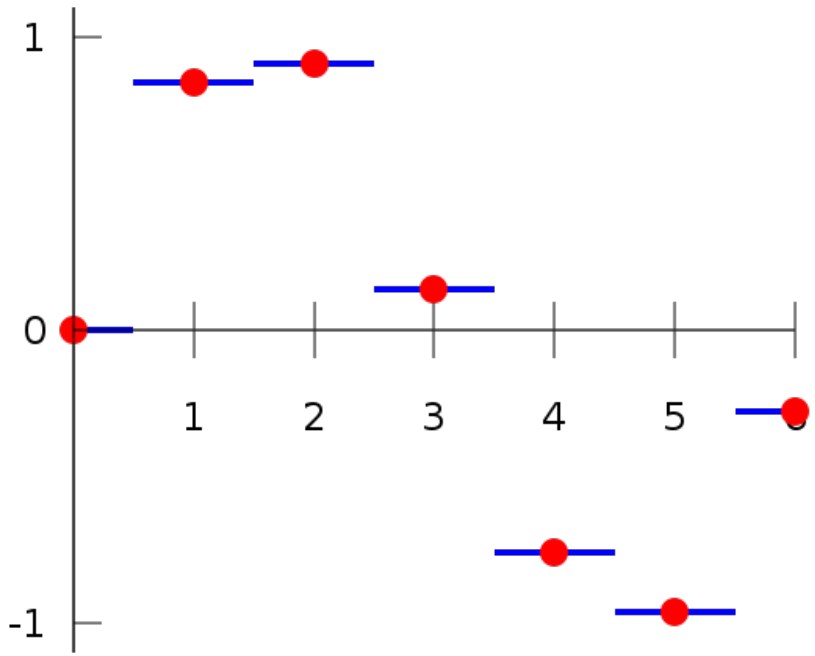
kde

$$h = \frac{(b-a)}{n-1}$$

- definujeme

$$\tilde{f}(x) = f(x_i)$$

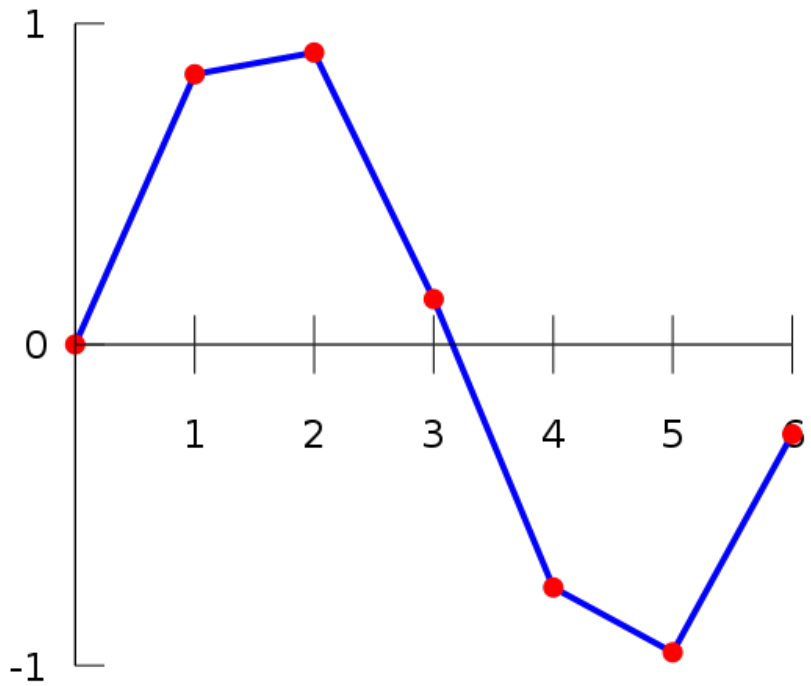
$$\forall x \in \left(\left(x_i - \frac{h}{2}, x_i + \frac{h}{2}\right) \cap [a,b]\right)$$



Interpolace v 1D

Lineární interpolace

- dva sousední body spojeny úsečkou



  - definujeme

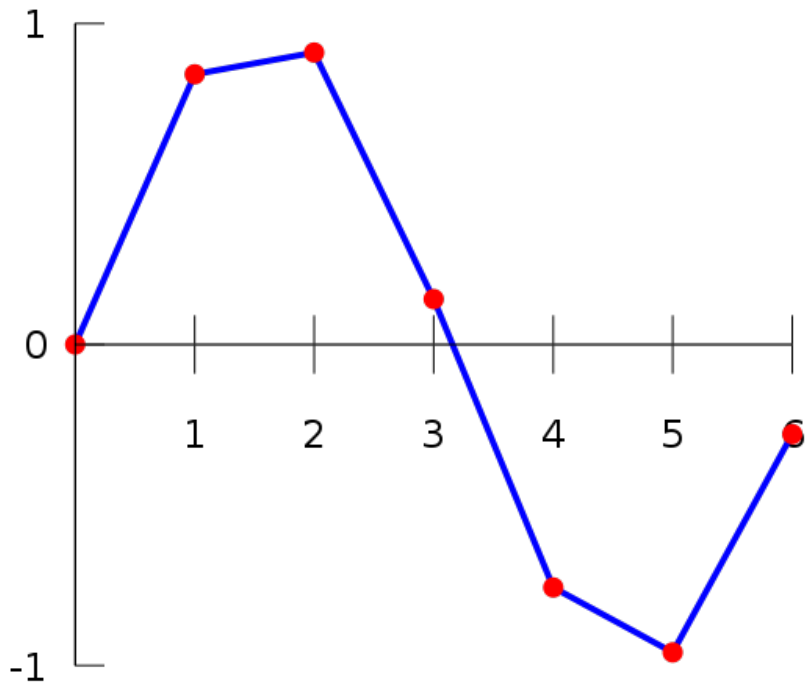
$$\tilde{f}(x) = \frac{x_{i+1} - x}{x_{i+1} - x_i} f(x_i) + \frac{x - x_i}{x_{i+1} - x_i} f(x_{i+1})$$

$$\forall x \in [x_i, x_{i+1}]$$

Interpolace v 1D

Lineární interpolace

- dva sousední body spojeny úsečkou



  - definujeme

$$\tilde{f}(x) = f(x_i) + \frac{x - x_i}{x_{i+1} - x_i}\left(f(x_{i+1}) - f(x_i)\right)$$

$$\forall x \in [x_i, x_{i+1}]$$

Interpolace v 1D

Polynomiální interpolace 1/2

- úsečkou můžeme interpolovat 2 body (lineární interpolace)
- n bodů lze interpolovat polynomem $L(x)$ stupně $n-1$
- definujeme

$$\Phi_i(x) = \frac{\prod_{\substack{k=1 \\ k\neq i}}^{n} (x - x_k)}{\prod_{\substack{k=1 \\ k\neq i}}^{n} (x_i - x_k)},$$

  potom $\Phi_i\left(x_j\right) = \delta_{ij}$
- Lagrangeův interpolační polynom má potom tvar

$$\tilde{f}(x) = L_n(x) = \sum_{i=1}^{n} f(x_i)\,\Phi_i(x)$$

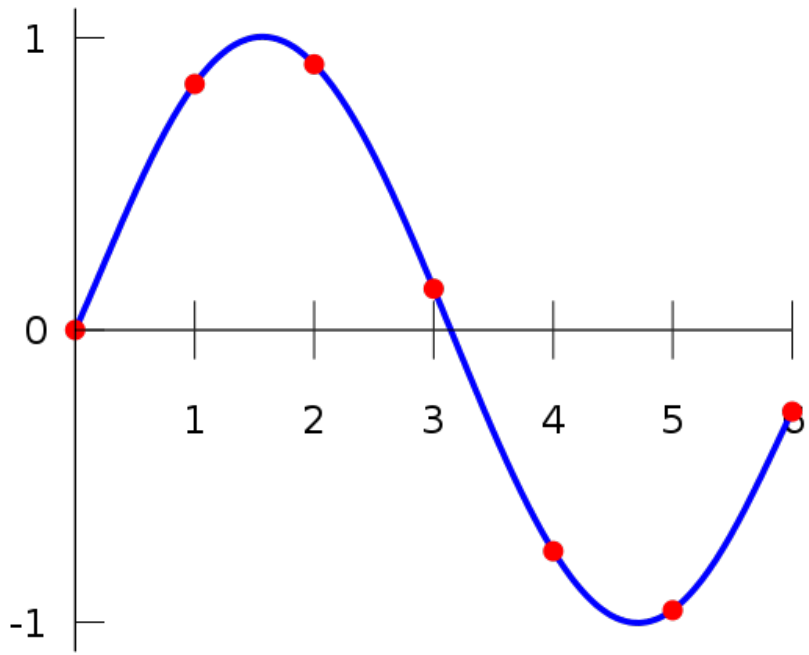
Interpolace v 1D

Polynomiální interpolace 2/2

- Lagrangeův polynom se nekonstruuje přímo, ale efektivně pomocí diferenčního schématu (viz Numerická matematika)
- extrémní hodnoty funkce $\tilde{f}(x)$ mohou ležet výrazně mimo interval

$$\left(\min_{i \in \mathbb{N}_n} f(x_i), \max_{i \in \mathbb{N}_n} f(x_i) \right)$$

$\Longrightarrow$ tzv. *overshoot*



Interpolace v 1D

Interpolace kubikou

- **kubika**, kubický *spline* - křivka popsaná parametricky: $x = x(t)$, $y = y(t)$
- $x(t)$, $y(t)$ jsou polynomy 3. stupně
- interpolace po segmentech (spojení každých dvou bodů)
- koeficienty polynomů lze nalézt tak, aby výsledná křivka měla např. spojitou derivaci

$$\begin{pmatrix} \dot{x}(t) \\ \dot{y}(t) \end{pmatrix}$$

v bodech navazování segmentů

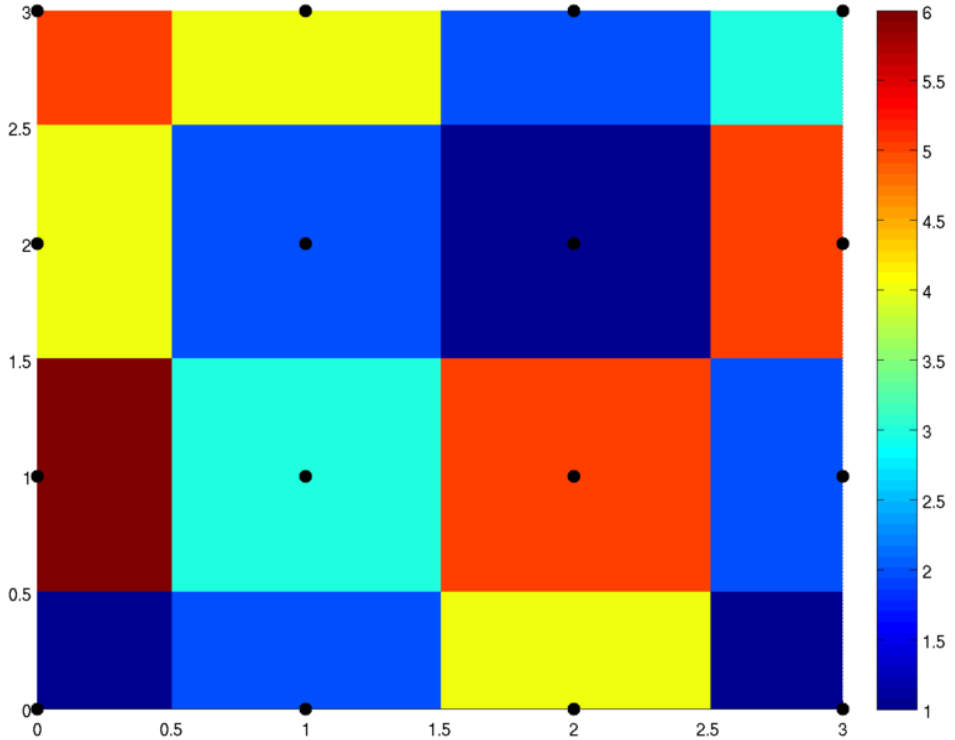
- segment křivky lze definovat pomocí tzv. **řídících bodů**
- mezi řídící body budou patřit i body $[x_i, f(x_i)]$
- detaily viz letní semestr, přednáška *Křivky a plochy*

Interpolace v 2D

Interpolace nejbližším sousedem

- pravidelná mřížka s rozestupem (velikostí oka sítě) $h = 1$
- vzorky funkce $f(x, y)$, tj. např. hodnoty intenzity pixelů $[i, j]$:

$$f_{i,j} = f(i, j)$$



- interpolace:

$$\tilde{f}(x, y) = f_{i,j}$$

$$\forall (x, y) \in \left(i - \frac{1}{2}, i + \frac{1}{2}\right) \times \left(j - \frac{1}{2}, j + \frac{1}{2}\right)$$

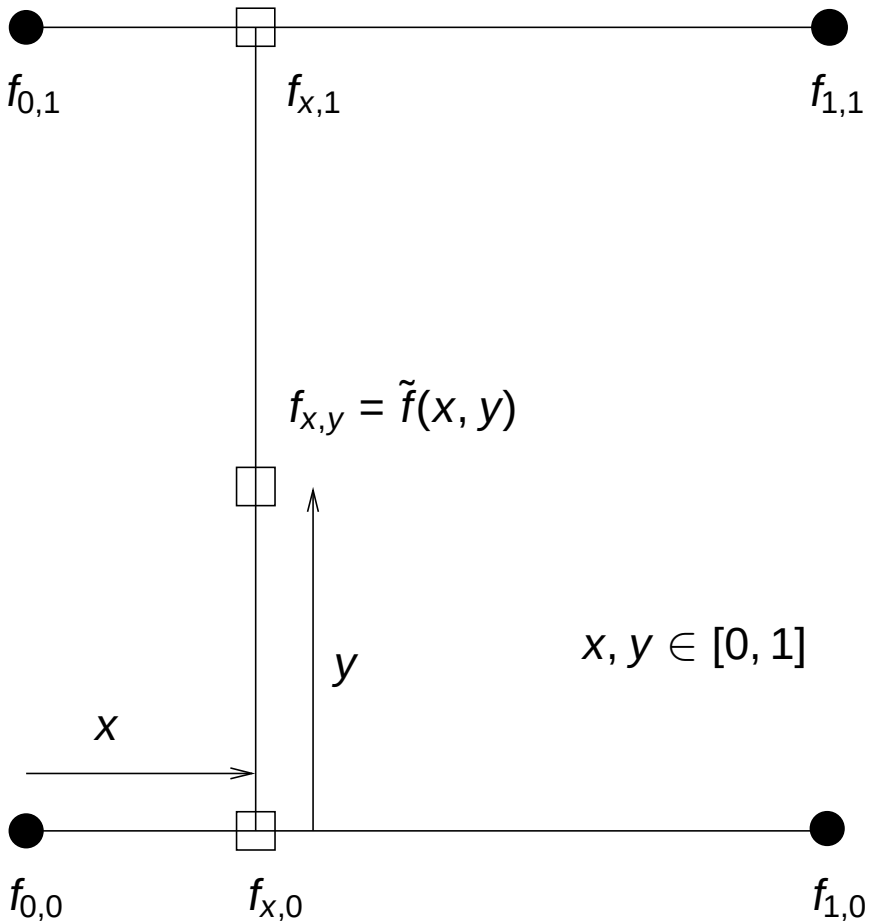
Interpolace v 2D

Bilineární interpolace

- **separabilní** procedura
- nejprve lineární interpolace ve směru osy x:

$$\begin{aligned} f_{x,j} &= f_{i,j} + (x-i)\left(f_{i+1,j} - f_{i,j}\right) \\ f_{x,j+1} &= f_{i,j+1} \\ &+ (x-i)\left(f_{i+1,j+1} - f_{i,j+1}\right) \end{aligned}$$

- poté lineární interpolace získaných hodnot ve směru y



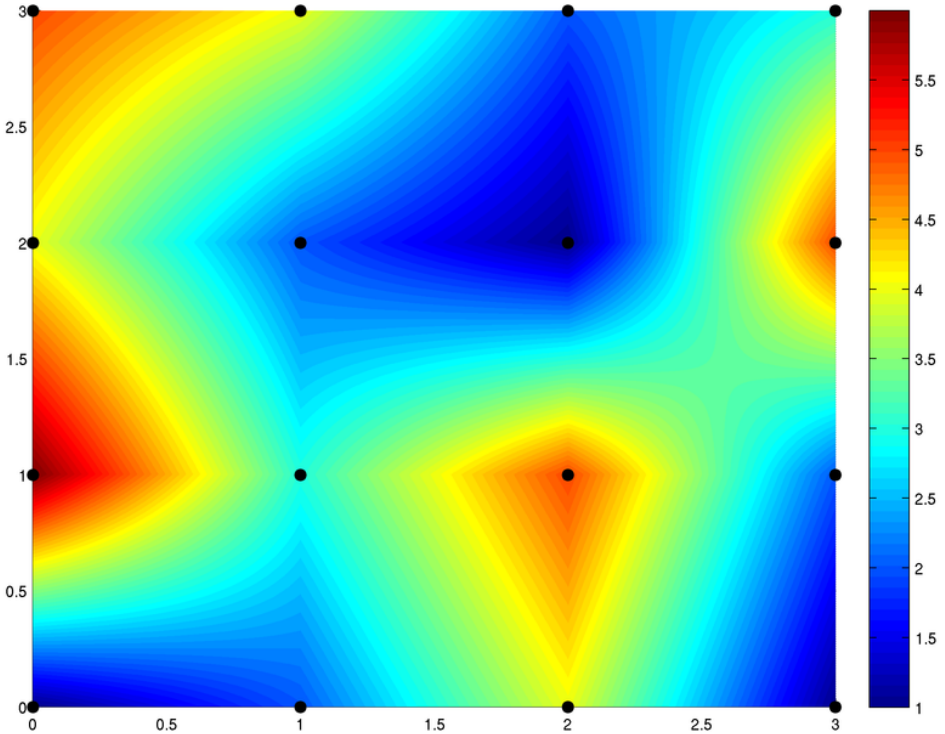
$$\tilde{f}(x,y) = f_{x,j} + (y-j)\left(f_{x,j+1} - f_{x,j}\right)$$

Interpolace v 2D

Bilineární interpolace

- **separabilní** procedura
- nejprve lineární interpolace ve směru osy x:

$$\begin{aligned} f_{x,j} &= f_{i,j} + (x-i)\left(f_{i+1,j} - f_{i,j}\right) \\ f_{x,j+1} &= f_{i,j+1} \\ &+ (x-i)\left(f_{i+1,j+1} - f_{i,j+1}\right) \end{aligned}$$



- poté lineární interpolace získaných hodnot ve směru y

$$\tilde{f}(x,y) = f_{x,j} + (y-j)\left(f_{x,j+1} - f_{x,j}\right)$$

Interpolace v 2D

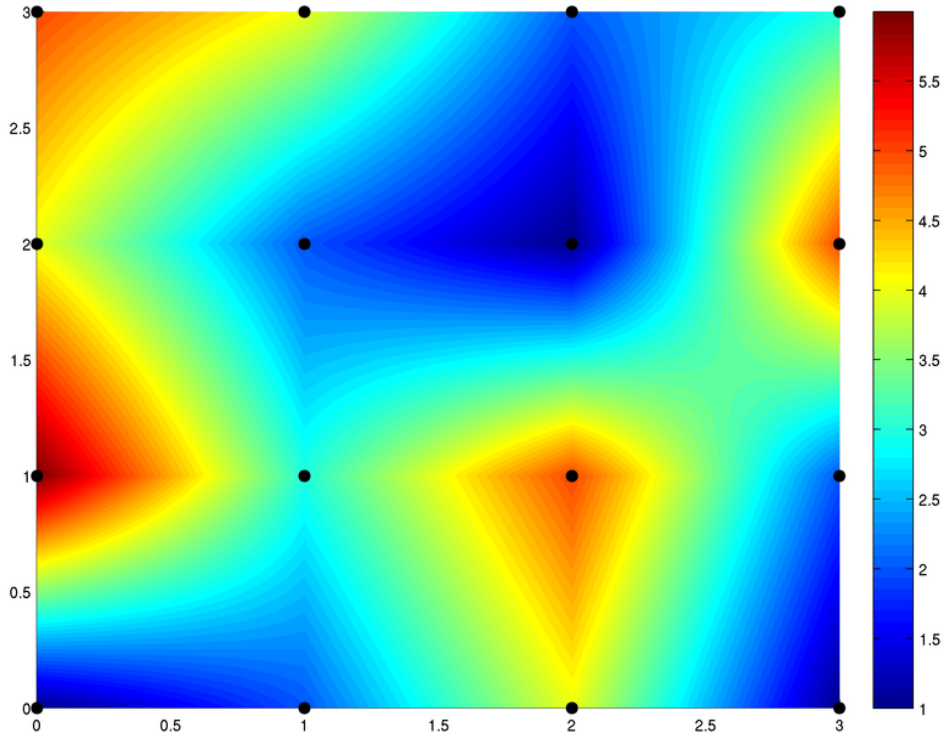
Bikubická interpolace

- interpolace bikubickou plochou mezi 4 body $[i,j],[i+1,j],[i,j+1],[i+1,j+1]$
- obecně

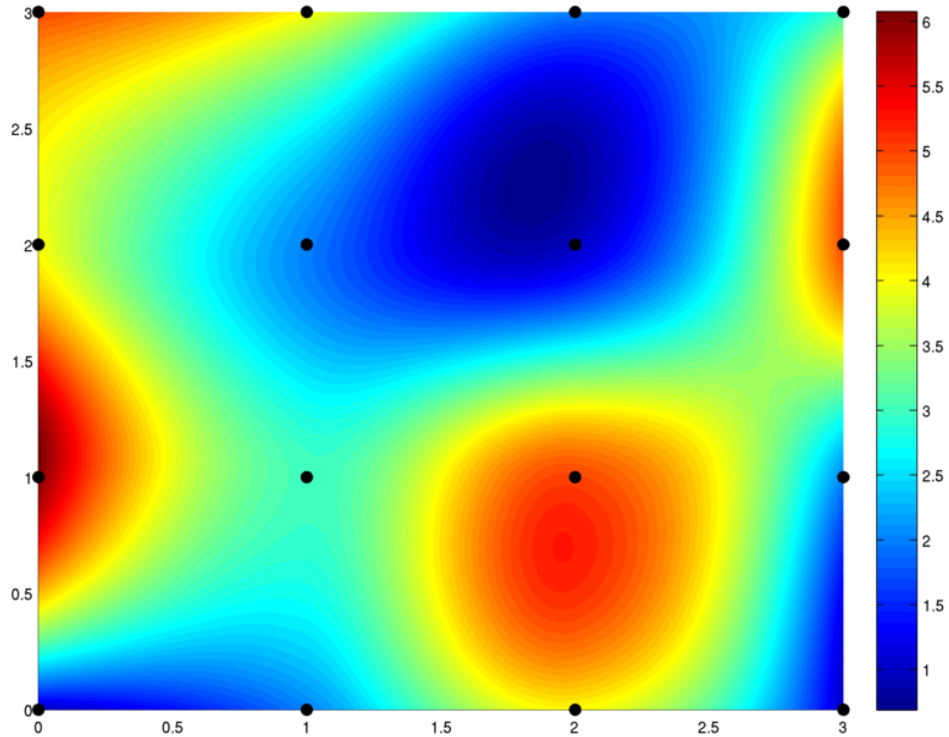
$$\tilde{f}(x,y) = \sum_{k=0}^{3}\sum_{l=0}^{3} a_{kl} x^k y^l$$

- 16 koeficientů a_{kl} lze odvodit z 4×4 podmínek pro 4 krajní body
  - rovnost funkčních hodnot $\tilde{f}(i,j) = f(i,j)$, $\tilde{f}(i+1,j) = f(i+1,j)$ atd.
  - rovnost derivací ve směru x: $\partial_x \tilde{f}(i,j) = \partial_x f(i,j)$, atd.
  - rovnost derivací ve směru y: $\partial_y \tilde{f}(i,j) = \partial_y f(i,j)$, atd.
  - rovnost smíšených derivací: $\partial^2_{xy} \tilde{f}(i,j) = \partial^2_{xy} f(i,j)$, atd.
- pokud derivace nejsou známy, lze je aproximovat konečnými diferencemi

Interpolace v 2D



Bilineární interpolace



Bikubická interpolace

- bikubická interpolace je náročnější, ale generuje hladší funkci

Interpolace barev

- neinterpolujeme skalární funkci, ale barvu
- musíme umět **sčítat barvy, násobit bravy skalárem**
- barva má 3 parametry - lze uvažovat jako vektor
- s vektory pracujeme standardně - po složkách
- vyjádření v různých barevných prostorech + interpolace parametrů po složkách $\Longrightarrow$ různý výsledek
  - viz přednáška *Vnímání a reprezentace barev*
- obvykle si vystačíme s prostorem RGB

Obsah

Základní transformace

- změna velikosti obrazu
- rotace
- zrcadlení

Dva způsoby implementace transformace

- transformace obrazů $T : A \mapsto B$
- **dopředné mapování**: procházíme obrázek A pixelech, hledáme polohu transformovaného pixelu v obraze B (hledáme obraz bodu při zobrazení T)
- **zpětné mapování**: procházíme obrázek B po pixelech, hledáme polohu **vzoru** daného pixelu v obraze A při zobrazení $T \implies$ obvykle potřebujeme najít T^{-1}

Změna velikosti obrazu

problémy:

1. změní se počet pixelů
2. pixely mají po transformaci neceločíselné souřadnice

řešení:

1. procházíme pouze pixely cílového obrazu - použijeme zpětné mapování
2. na původní obrázek A použijeme interpolaci $\Longrightarrow$ intenzita obrazu je definována všude

výběr interpolačního algoritmu:

- **nejbližším sousedem** - nejrychlejší, ale při zvětšení je obraz kostrbatý, při zmenšení se ztrácí tenké objekty (čáry apod.) = aliasing
- **bilineární** - ve většině případů dostačující, při zvětšení rozmazává hrany
- **bikubická** - dostačující hladkost, lépe zachovává hrany

Rotace obrazu

- rotace o násobky 90° triviální
- rotace bodu kolem počátku o úhel α proti směru hodinových ručiček

$$\begin{aligned} x' &= x\cos\alpha - y\sin\alpha \\ y' &= y\cos\alpha + x\sin\alpha \end{aligned}$$

- zpětná transformace je otočení o $-\alpha$

$$\begin{aligned} x &= x'\cos\alpha + y'\sin\alpha \\ y &= y'\cos\alpha - x'\sin\alpha \end{aligned}$$

- opět použijeme interpolaci
- rotovaný obraz se vejde na větší „plátno“ než obraz původní
  - nutnost vypočítat rozměr plátna: pomocí rotace rohů obrazu
  - při zpětném mapování nutnost ošetřit souřadnice vzoru pixelu mimo obraz A

Obsah

Prolnutí obrazů

- **alfa-míchání** (*alpha-blending*) dvou obrazů A_1, A_2 do jednoho obrazu B
- lineární interpolace barev (intenzit) pixelů na totožných pozicích

$$B(i,j) = A_1(i,j) + \alpha(A_2(i,j) - A_1(i,j))$$

- parametr α je průhlednost obrazu A_2 (A_1 je pozadí, A_2 popředí):
- $\alpha = 0 \implies$ zcela průhledný, $\alpha = 1 \implies$ neprůhledný
- v grafických akcelerátorech implementováno hardwarově
- **prolnutí** (*cross dissolve*) - animace sekvence n obrazů, kde

$$B_k(i,j) = A_1(i,j) + \frac{k}{n}(A_2(i,j) - A_1(i,j))$$

$$k = 0, 1, 2, \ldots, n$$

Alfa míchání

Cross-dissolve efekt

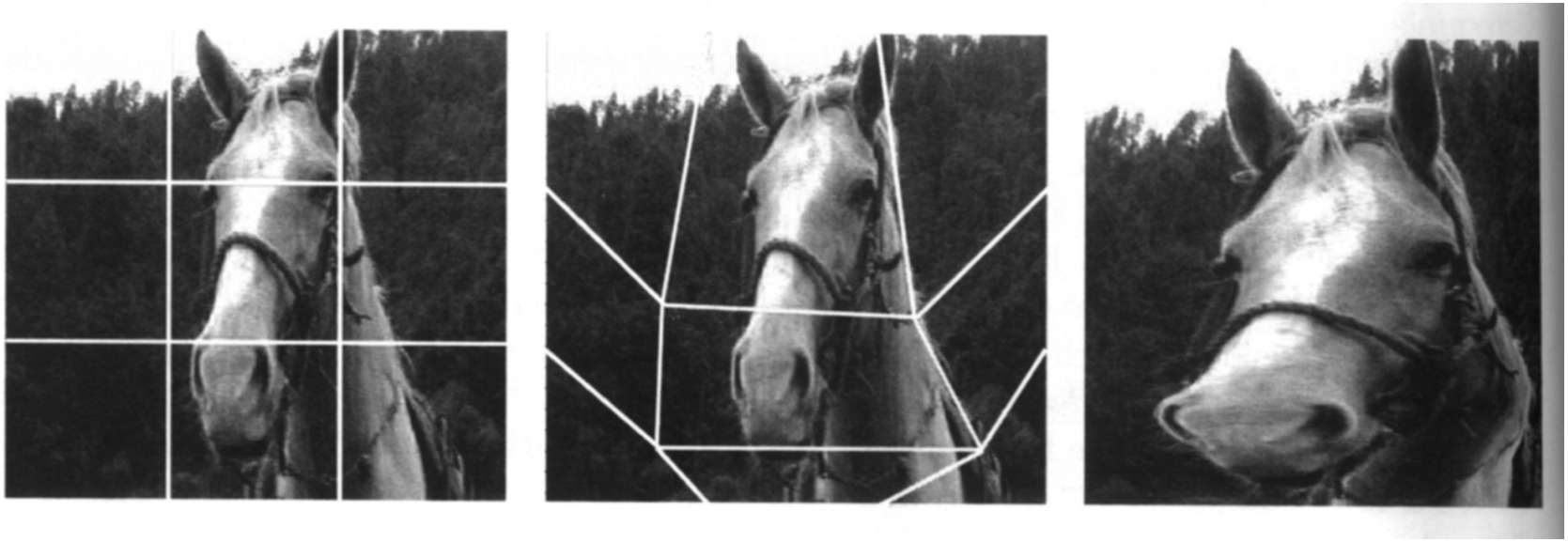
- Slideshow fotografií s efektem dissolve Spustit video

Warping

- deformace obrazu - nelineární transformace
- **síťový warping** - k obrazu je „přilepena“ síť.
  - pohybem uzlových bodů sítě deformujeme obraz
  - v každém oku sítě zůstává stále stejná část obrázku
  - vhodný pro globální deformace
- **úsečkový warping** (*feature based warping*) - místo sítě je v obraze množina nezávislých úseček, které k sobě váží své okolí
  - přemístěním úseček deformujeme obraz
  - vhodnější pro lokální změny

Síťový warping

- na obraz položíme pomyslnou síť
- spojnice uzlových bodů sítě mohou být přímky, kubické křivky (spliny) apod.
- změníme síť editací uzlových, resp. (v případě křivek) řídících bodů

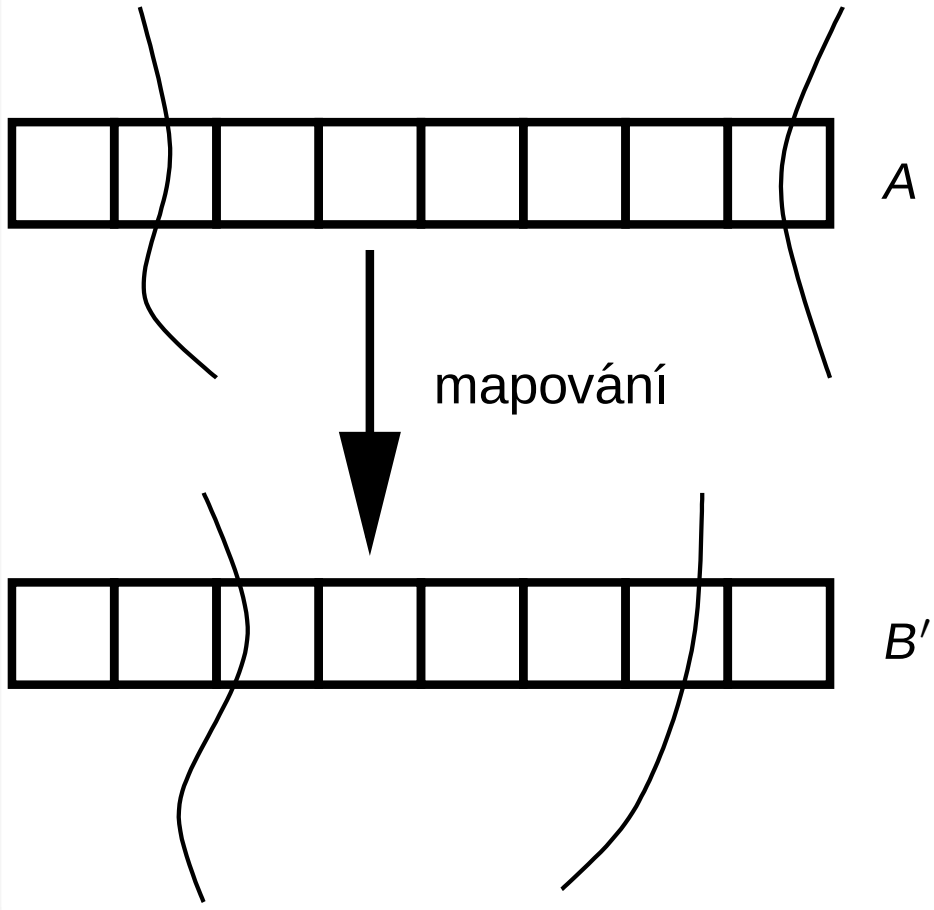


Síťový warping

Algoritmus

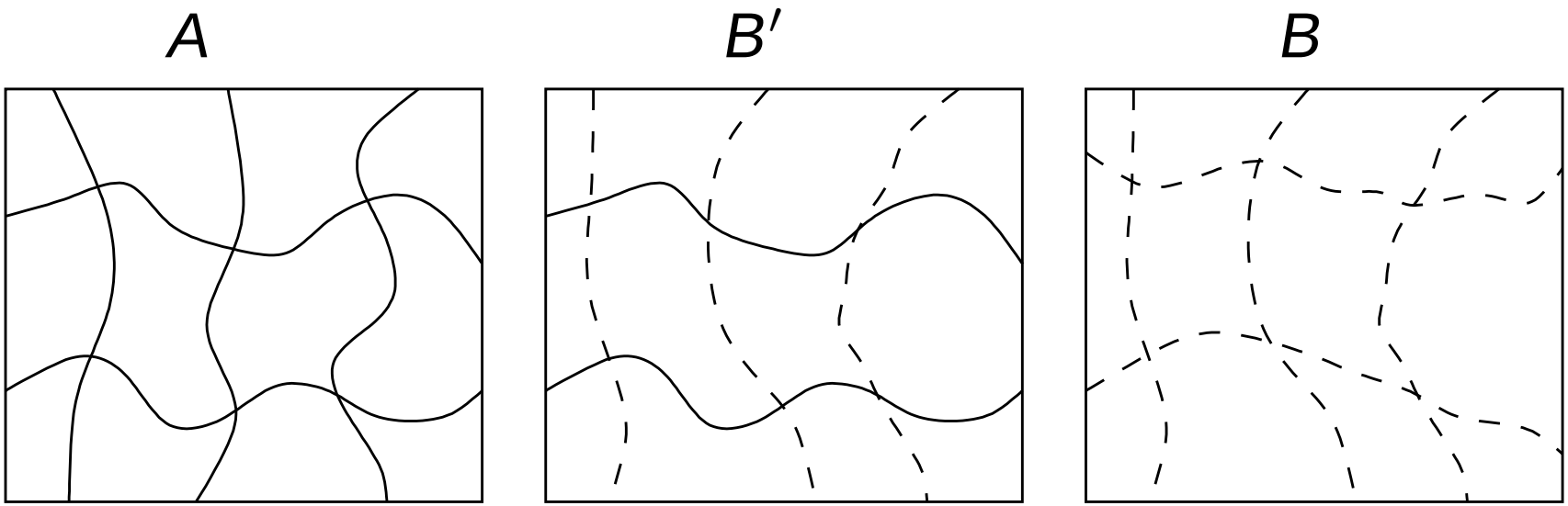
algoritmus je separabilní na dva kroky 1-2, 3-4:

1. určíme průsečíky každého řádku vstupního obrazu A s čarami ve svislém směru
2. získané intervaly se (lineárně) převzorkují na nový rozsah na modifikované síti
   $\Longrightarrow$ vznikne obraz B'
3. určíme průsečíky každého sloupce obrazu B' s čarami ve vodorovném směru
4. získané intervaly převzorkujeme $\Longrightarrow$ vznikne výsledný obraz B



Síťový warping

Algoritmus

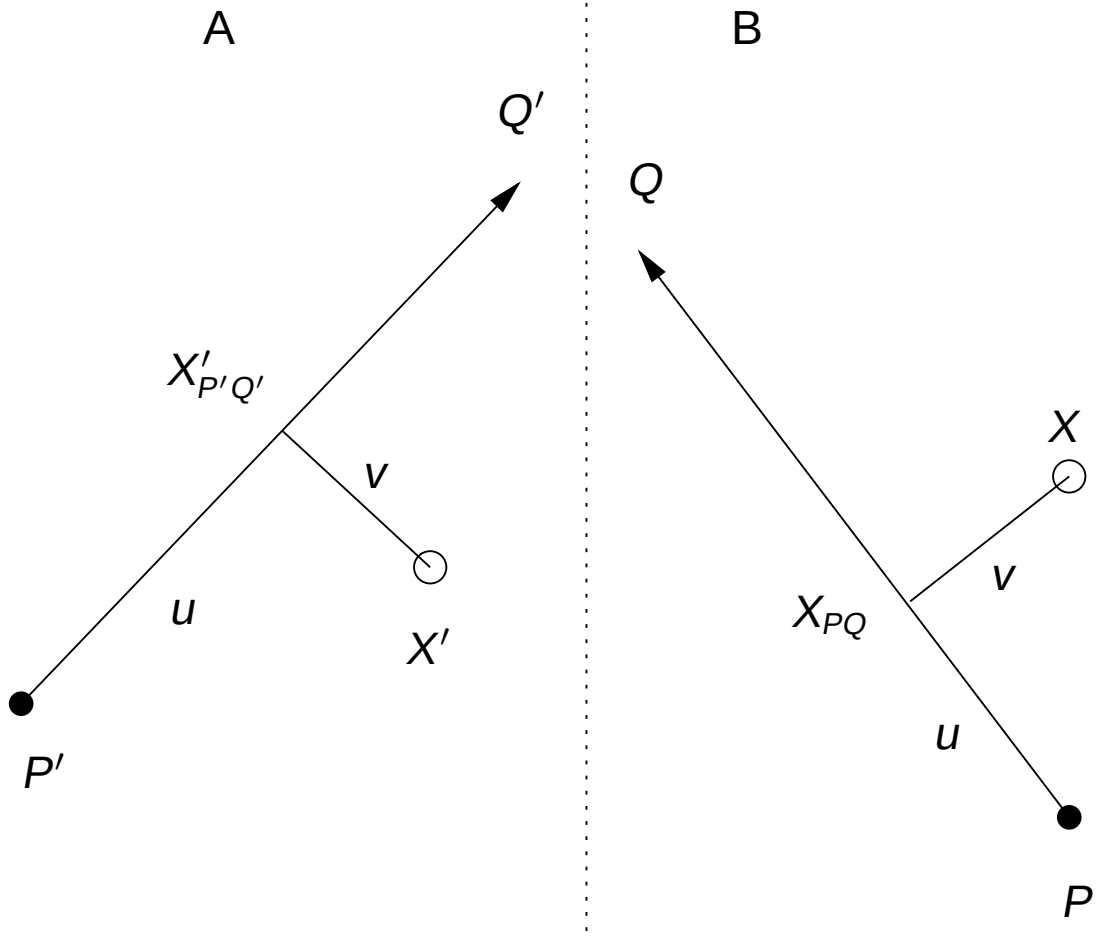


Dvouprůchodové zpracování sítě

Úsečkový warping

Transformace jedním párem úseček 1/2

- používáme **zpětné mapování** bodu X v B na bod X' v A
- úsečka $\overline{PQ}$ definuje v obraze B souřadný systém:
  - u - poloha X_{PQ} podél úsečky škálovaná tak, že v bodě P je $u = 0$, v bodě Q je $u = 1$
  - v - vzdálenost od **přímky** PQ
- bod X' v A najdeme jako bod o souř. (u, v) vzhledem k $\overline{P'Q'}$



Úsečkový warping

Transformace jedním párem úseček 2/2

- souřadnice u: projekce vektoru $(X-P)$ do směru $(Q-P)$

$$u = \frac{(X-P)\cdot(Q-P)}{\|Q-P\|^2}$$

- analogicky souřadnice v:

$$v = \frac{(X-P)\cdot(Q-P)^{\perp}}{\|Q-P\|}$$

kde vektor $\boldsymbol{v}^{\perp} = (-y, x)$ je kolmý k vektoru $\boldsymbol{v} = (x, y)$

- souřadnice bodu X' pak dostaneme snadno jako

$$X' = P' + u(Q'-P') + v\frac{(Q'-P')^{\perp}}{\|Q'-P'\|}$$

- X' nemá celočíselné souřadnice $\Longrightarrow$ **interpolujeme**

Úsečkový warping

Transformace více páry úseček 1/4

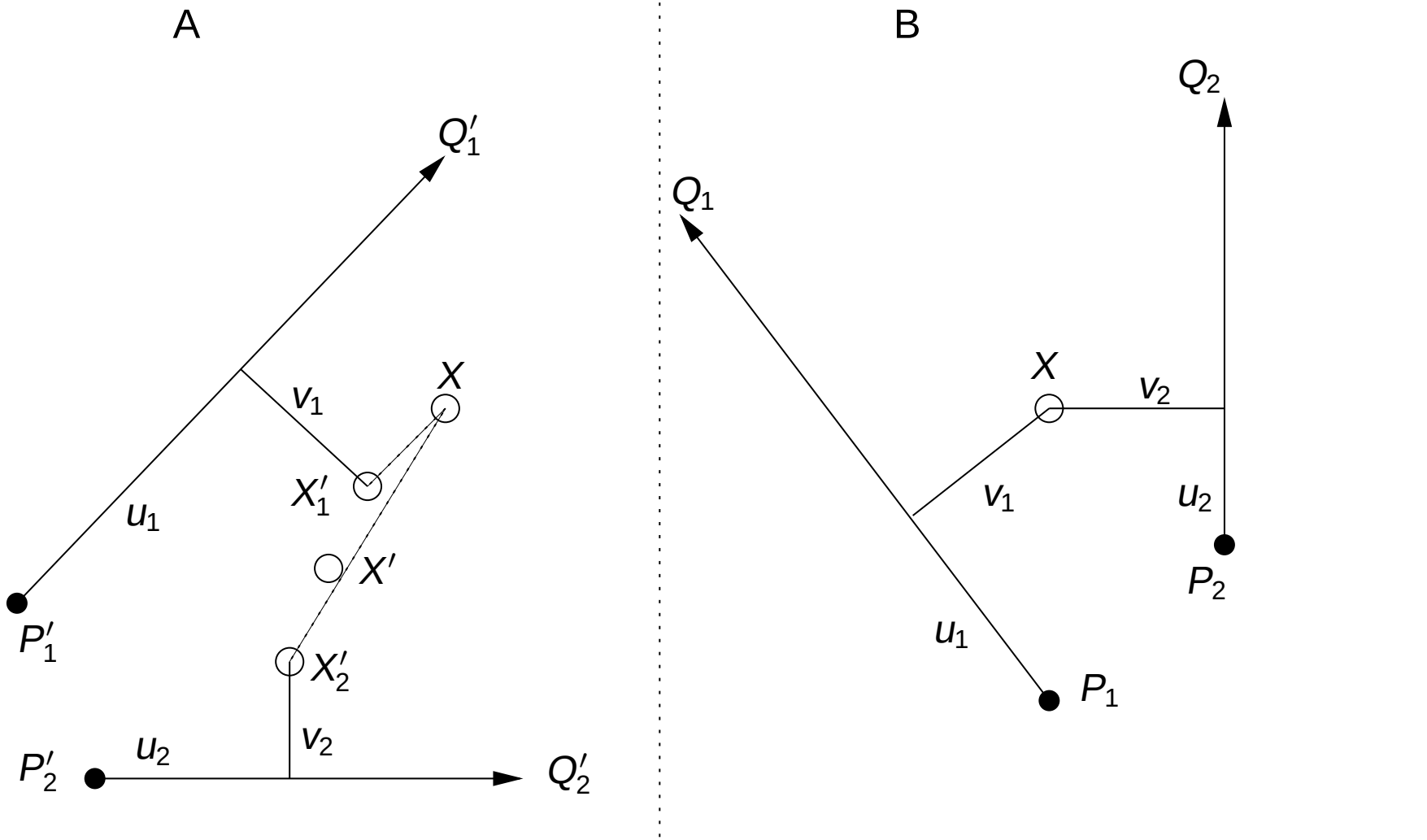
- transformace jedním párem úseček: posunutí, rotace, změna měřítka (afinní transformace)
- více párů úseček $\Longrightarrow$ více bodů X'
- je-li X'_i vzor bodu X při použití i-tého páru úseček a $\boldsymbol{D}_i = X'_i - X$ je odpovídající posunutí bodu X'_i, vypočítáme

$$X' = X + \frac{\sum_{i=1}^{n} w_i \boldsymbol{D}_i}{\sum_{i=1}^{n} w_i}, \text{ resp. přímo } X' = \frac{\sum_{i=1}^{n} w_i X'_i}{\sum_{i=1}^{n} w_i},$$

kde w_i je váha i-tého bodu X'_i závislá na vzdálenosti X od i-té úsečky

Úsečkový warping

Transformace více páry úseček 2/4



Úsečkový warping

Transformace více páry úseček 3/4

- váha w_i se určí jako

$$w_i = \left(\frac{|Q'_i - P'_i|^p}{(a + d_i)} \right)^b,$$

kde d_i je vzdálenost bodu X od **úsečky** $\overline{P_i Q_i}$, tj.

$$d_i = \begin{cases} |v| & u \in [0,1] \\ \|P - X\| & u < 0 \\ \|Q - X\| & u > 1 \end{cases}$$

a $\underline{a, b, p}$ jsou parametry zadané uživatelem

Úsečkový warping

Transformace více páry úseček 4/4

- parametr $a > 0$ určuje vliv úsečky na nejbližší okolí.
  - pokud $a \approx 0$, váha bodu přímo na úsečce je hodně velká $\Longrightarrow$ bod na úsečce se zobrazí opět na bod na úsečce (přesná kontrola)
  - pokud a bude větší, warping bude hladší za cenu menší kontroly
  - lze např. $a = 0.001$.
- parametr b určuje rychlost slábnutí vlivu úsečky se vzdáleností
  - vhodné je $b \in [0.5, 2]$
- parametr $p \in [0, 1]$ určuje vliv délky úsečky na váhu
  - $p = 0 \Longrightarrow$ stejný vliv nezávisle na délce
  - $p = 1 \Longrightarrow$ čím delší, tím silnější vliv

Úsečkový warping

Správné rozmístění úseček

- úsečky by se neměly křížit, jinak vzniknou artefakty („bubliny“) - tzv. *Ghostbustering effect*
  - Ghostbusters:
    - „Don’t cross the streams!“
    - „Why?“
    - „It would be bad.“
- úsečky by se měly dotýkat pouze na koncích, a to v obou obrázcích
- pokud nechceme, aby se X' mapovalo i mimo zdrojový obraz, můžeme definovat páry úseček na okrajích obrázků A, B
  - v opačném případě dodefinujeme okolí obrázku A např. černou

Úsečkový warping

Správné rozmístění úseček

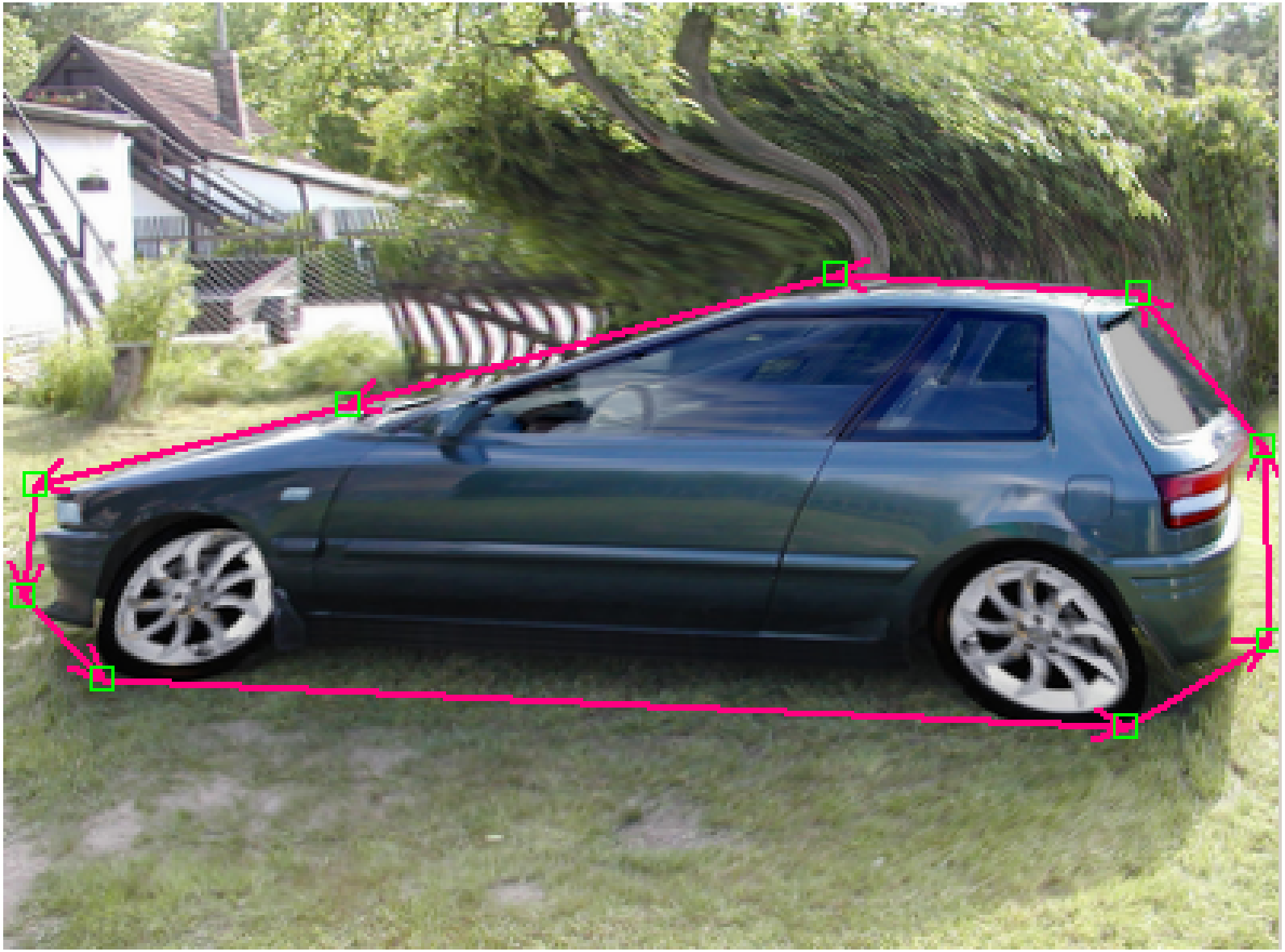
- úsečky by se neměly křížit, jinak vzniknou artefakty („bubliny“) - tzv. *Ghostbustering effect*
  - Ghostbusters:
    - „Don’t cross the streams!“
    - „Why?“
    - „It would be bad.“
- úsečky by se měly dotýkat pouze na koncích, a to v obou obrázcích
- pokud nechceme, aby se X' mapovalo i mimo zdrojový obraz, můžeme definovat páry úseček na okrajích obrázků A, B
  - v opačném případě dodefinujeme okolí obrázku A např. černou

Úsečkový warping - příklady

Spustit demo

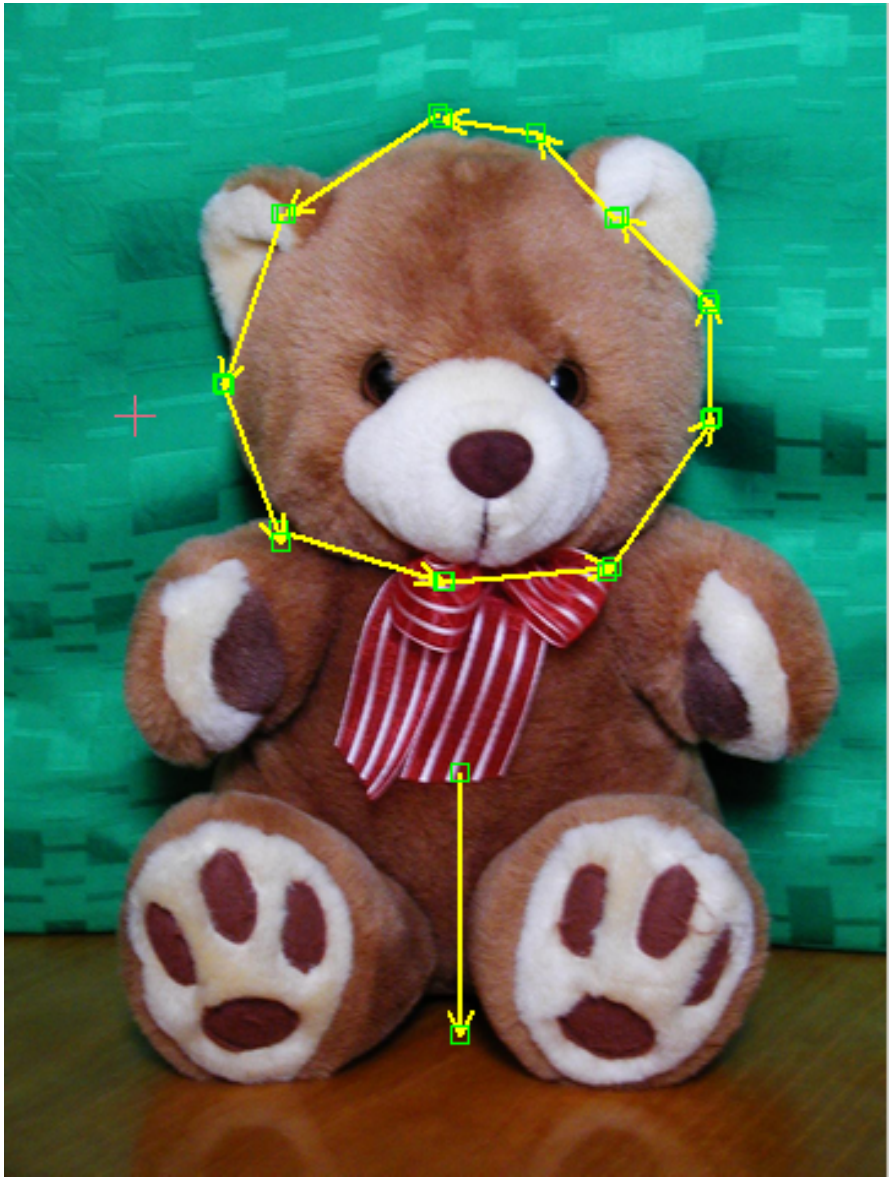
Úsečkový warping - příklady

Spustit demo



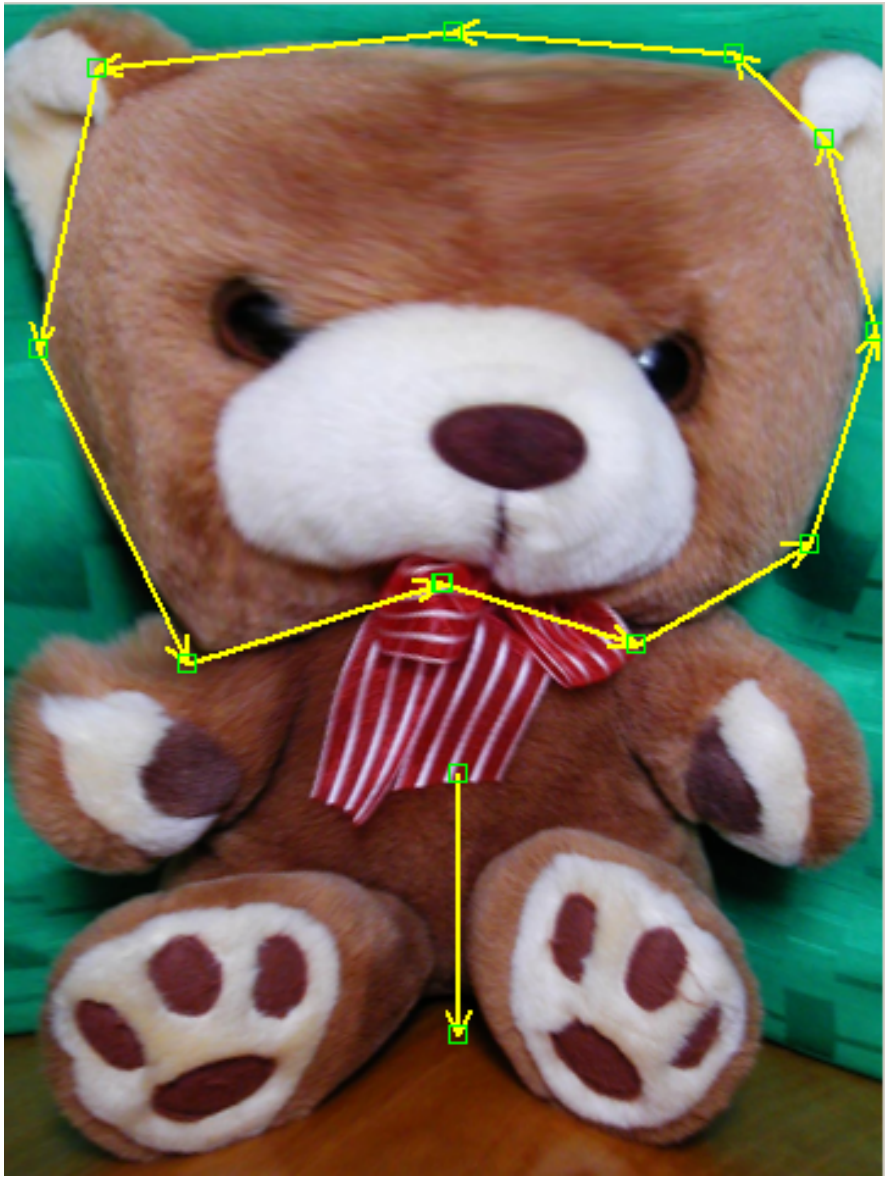
Úsečkový warping - příklady

Spustit demo



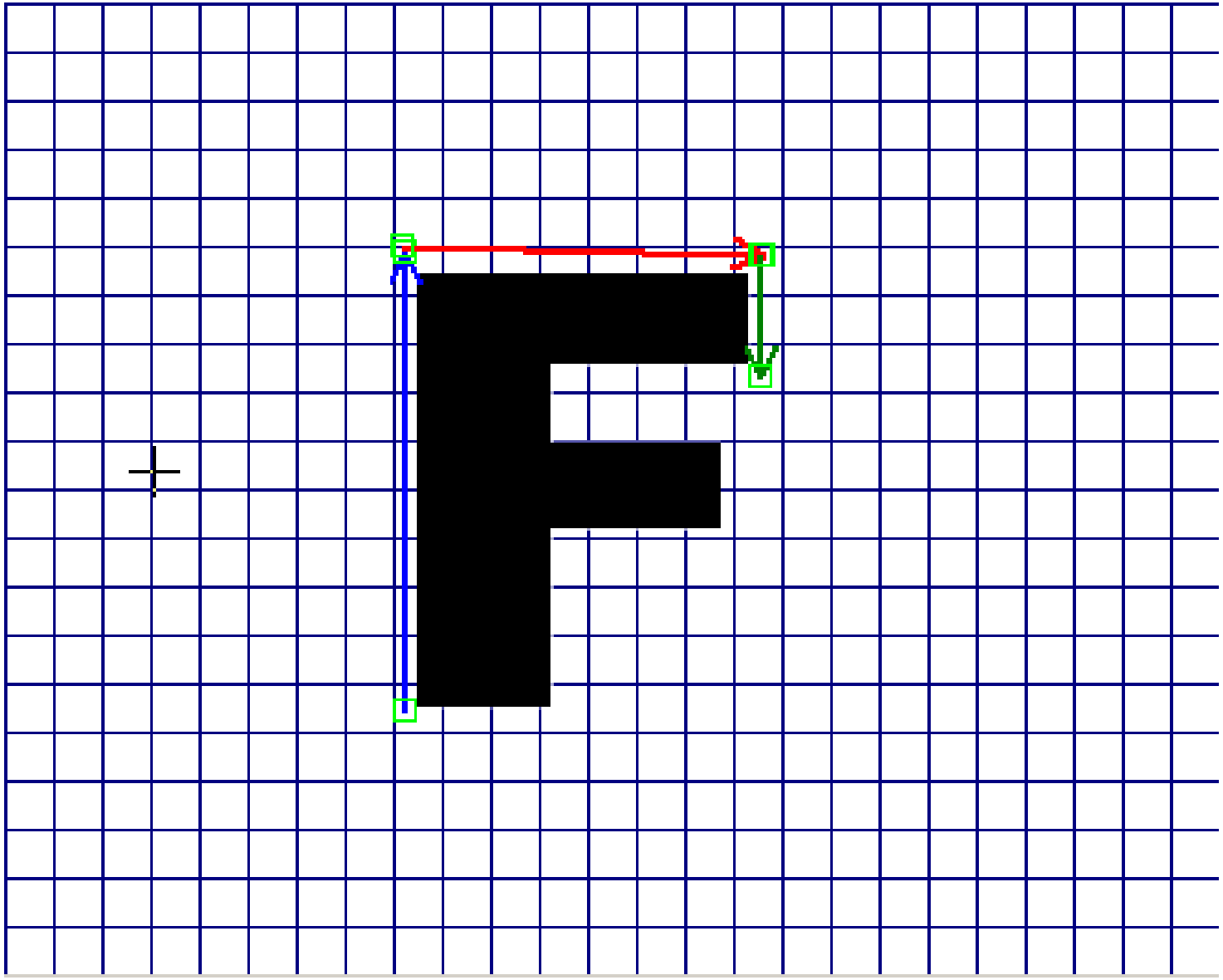
Úsečkový warping - příklady

Spustit demo



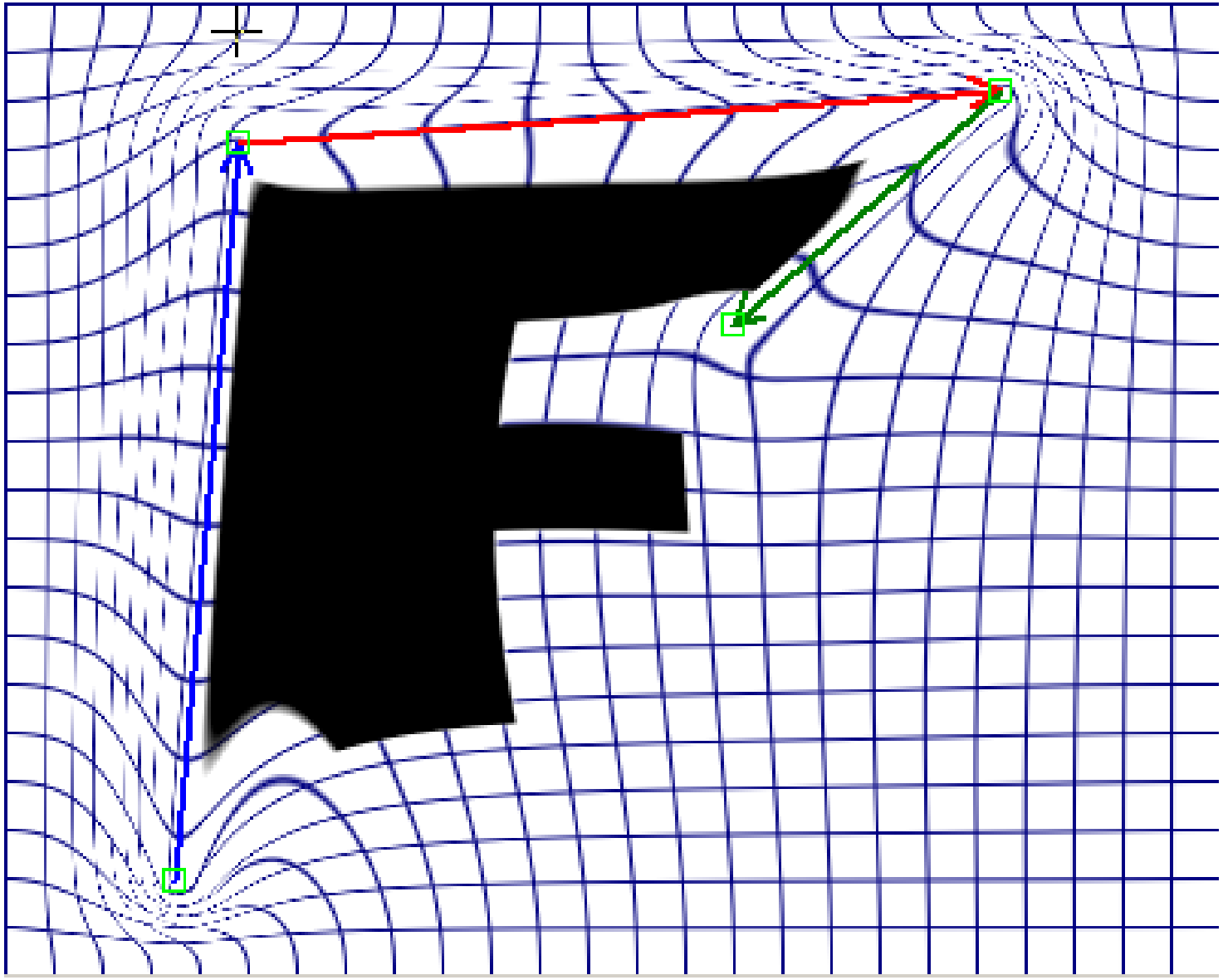
Úsečkový warping - příklady

Spustit demo



Úsečkový warping - příklady

Spustit demo



Morphing

- přeměna obrazu A v obraz B pomocí **warpingu** a **prolnutí**
- deformace obrazu obecně popsána systémem řídících bodů (koncové body úseček, uzlové body sítě)
  - v obrazech A, B systémy bodů $\mathcal{T}_A = \left(p_i^A\right)_{i=1}^m$, $\mathcal{T}_B = \left(p_i^B\right)_{i=1}^m$

Tvorba M_k (k-tého snímku animace), $k = 0, 1, 2, \ldots, n$

1. Vytvoř systém $\mathcal{T}_k$: $\forall i$ lineárně interpoluj souřadnice řídících bodů
$$p_i^k = p_i^A + \frac{k}{n}\left(p_i^B - p_i^A\right)$$
2. warpuj $A \mapsto M_k^A$ přechodem od $\mathcal{T}_A$ k $\mathcal{T}_k$
$B \mapsto M_k^B$ přechodem od $\mathcal{T}_B$ k $\mathcal{T}_k$
3. proveď cross-dissolve

$$M_k = M_k^A + \frac{k}{n}\left(M_k^B - M_k^A\right)$$

Morphing - příklady

- řídící body pro warping vymezují analogické objekty ve zdrojovém a cílovém obrázku
  - při přeměně obličejů: oči, nos, ústa, uši, brada atd.
- Video „Brangelina“ Spustit video
- Video „Michael Jackson“ Spustit video
- Video „Michael Jackson: Black or White“ Spustit video

Literatura

- T. Beier, S. Neely: *Feature-based image metamorphosis*. Computer graphics (SIGGRAPH '92 Proceedings) (1992), 35-42.
- Žára, Beneš, Sochor, Felkel: *Moderní počítačová grafika*. Computer Press, 2005. ISBN: 80-251-0454-0