

Optimalizace

7. Volné lokální extrémymy

Tomáš Werner, Tomáš Kroupa

FEL ČVUT

Analytické podmínky na volné lokální extrémy

Věta (nutná podmínka prvního řádu)

Nechť funkce $f: \mathbb{R}^n \rightarrow \mathbb{R}$ je v bodě $\mathbf{x} \in \mathbb{R}^n$ (totálně) diferencovatelná.

Jestliže \mathbf{x} je lokální extrém funkce f , pak $f'(\mathbf{x}) = \mathbf{0}$.

Bod $\mathbf{x} \in \mathbb{R}^n$ splňující $f'(\mathbf{x}) = \mathbf{0}$ se nazývá **stacionární bod** funkce f .

Věta (podmínky druhého řádu)

Nechť funkce $f: \mathbb{R}^n \rightarrow \mathbb{R}$ je v bodě $\mathbf{x} \in \mathbb{R}^n$ dvakrát diferencovatelná.

- Jestliže \mathbf{x} je lokální minimum [maximum] funkce f , pak $f'(\mathbf{x}) = \mathbf{0}$ a Hessova matice $f''(\mathbf{x})$ je pozitivně [negativně] semidefinitní.
- Jestliže $f'(\mathbf{x}) = \mathbf{0}$ a Hessova matice $f''(\mathbf{x})$ je pozitivně [negativně] definitní, pak \mathbf{x} je ostré lokální minimum [maximum] funkce f .

Tedy když $f''(\mathbf{x})$ je indefinitní, pak \mathbf{x} není lokální extrém.

Bod \mathbf{x} , ve kterém $f'(\mathbf{x}) = \mathbf{0}$ a $f''(\mathbf{x})$ je indefinitní, se nazývá **sedlový bod** funkce f .

Iterační metody na volné lokální extrémů

Rychlost konvergence iteračních algoritmů

Nechť posloupnost bodů $\mathbf{x}_k \in \mathbb{R}^n$ konverguje k bodu \mathbf{x}^* .

Pak posloupnost čísel $r_k = \|\mathbf{x}_k - \mathbf{x}^*\| \geq 0$ konverguje k nule.

Definice: Pokud existuje limita

$$\rho = \lim_{k \rightarrow \infty} \frac{r_{k+1}}{r_k}$$

řekneme, že posloupnost \mathbf{x}_k (příp. r_k) konverguje

- **sublineárně**, pokud $\rho = 1$,
- **lineárně**, pokud $0 < \rho < 1$,
- **superlineárně**, pokud $\rho = 0$.

Příklady:

- Posloupnost $r_k = \frac{1}{k} = (1, \frac{1}{2}, \frac{1}{3}, \frac{1}{4}, \dots)$ konverguje sublineárně.
- Posloupnost $r_k = 2^{-k} = (\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \dots)$ konverguje lineárně.
- Posloupnost $r_k = 2^{-2^k} = (\frac{1}{4}, \frac{1}{16}, \frac{1}{256}, \dots)$ konverguje superlineárně.

Sestupné metody, názvosloví

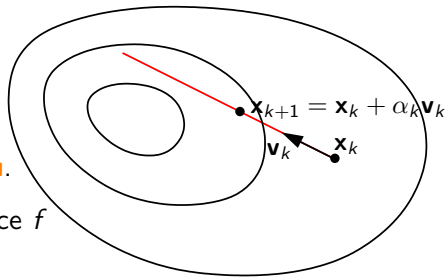
Hledáme lokální minimum diferencovatelné funkce $f: \mathbb{R}^n \rightarrow \mathbb{R}$.

- Zvolíme **počáteční odhad** $\mathbf{x}_0 \in \mathbb{R}^n$
- Tvoříme posloupnost $\mathbf{x}_1, \mathbf{x}_2, \dots$ pomocí iterace

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{v}_k$$

kde \mathbf{v}_k je **směr hledání** a $\alpha_k > 0$ určuje **délku kroku**.

- \mathbf{v}_k, α_k mohou záviset na hodnotách a derivacích funkce f v bodech $\mathbf{x}_k, \mathbf{x}_{k-1}, \mathbf{x}_{k-2}, \dots$ (často závisí jen na \mathbf{x}_k).
Řád metody je nejvyšší řád použitých derivací.



Sestupné metody: pro každé k platí $f(\mathbf{x}_{k+1}) < f(\mathbf{x}_k)$.

- \mathbf{v}_k je **sestupný směr** jestliže

$$f_{\mathbf{v}_k}(\mathbf{x}_k) = f'(\mathbf{x}_k) \mathbf{v}_k < 0.$$

Tvrzení: Je-li \mathbf{v}_k sestupný, pak existuje $\alpha_k > 0$ takové, že $f(\mathbf{x}_{k+1}) < f(\mathbf{x}_k)$.

Volba součinitele délky kroku α_k

- **Optimální délka kroku (exact line search):**

α_k je globální minimum funkce $\varphi(\alpha) = f(\mathbf{x}_k + \alpha \mathbf{v}_k)$ na intervalu $(0, \infty)$.

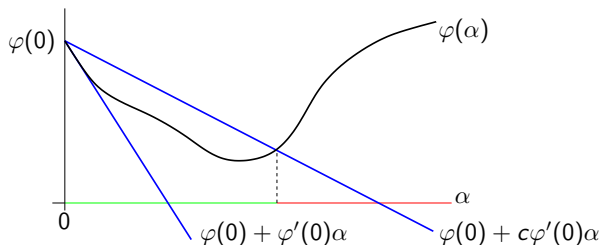
- **Suboptimální délka kroku (approximate/inexact line search):**

Hledáme nějaké $\alpha > 0$ tak, aby $\varphi(\alpha) < \varphi(0)$.

Backtracking line search: Začneme od nějakého velkého $\alpha > 0$ a zmenšujeme ho (násobením konstantou), dokud nesplňuje **Armijo-Goldsteinovo pravidlo**

$$\varphi(\alpha) \leq \varphi(0) + c\varphi'(0)\alpha$$

kde $c \in (0, 1)$ je libovolná konstanta (stejná pro všechny iterace).



- **Pevná posloupnost délek kroku** $(\alpha_k)_{k=0}^{\infty}$. Obvykle splňují

$$\lim_{k \rightarrow \infty} \alpha_k = 0, \quad \sum_{k=1}^{\infty} \alpha_k = \infty.$$

- **Konstantní délka kroku:** $\alpha_k = \alpha > 0$ pro každé k

Tyto dvě volby negarantují monotónní pokles funkce f .

Gradientní metoda

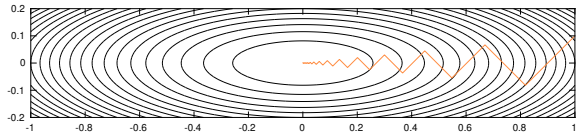
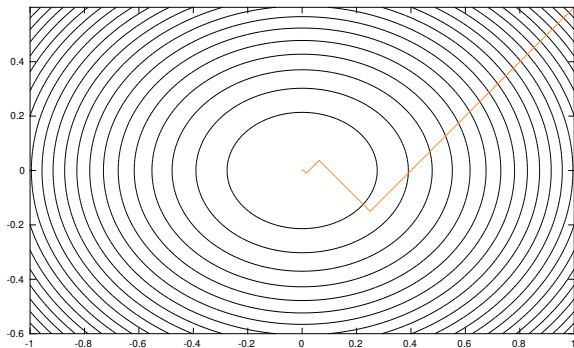
Gradientní metoda (Gradient Descent)

Iterace: $\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k f'(\mathbf{x}_k)^T$

- Směr $\mathbf{v}_k = -f'(\mathbf{x}_k)^T = -\nabla f(\mathbf{x}_k)$ je vždy sestupný:

$$f'(\mathbf{x}_k)\mathbf{v}_k = -f'(\mathbf{x}_k)f'(\mathbf{x}_k)^T = -\|\nabla f(\mathbf{x}_k)\|^2 < 0$$

- Robustní: za velmi obecných předpokladů konverguje, a to obvykle **lineárně**.
- Pomalá konvergence je-li funkce okolo minima protažená (cik-cak chování).



Newtonova metoda

Newtonova metoda na řešení soustavy rovnic

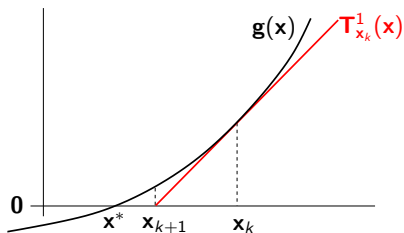
Dáno diferencovatelné zobrazení $\mathbf{g}: \mathbb{R}^n \rightarrow \mathbb{R}^n$, řešíme soustavu rovnic $\mathbf{g}(\mathbf{x}) = \mathbf{0}$.

Iterace: Pro známé \mathbf{x}_k hledáme \mathbf{x}_{k+1} splňující

$$\mathbf{T}_{\mathbf{x}_k}^1(\mathbf{x}_{k+1}) = \mathbf{g}(\mathbf{x}_k) + \mathbf{g}'(\mathbf{x}_k)(\mathbf{x}_{k+1} - \mathbf{x}_k) = \mathbf{0}$$

To je lineární soustava s řešením

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \mathbf{g}'(\mathbf{x}_k)^{-1} \mathbf{g}(\mathbf{x}_k)$$



- Konverguje, jestliže počáteční odhad \mathbf{x}_0 je 'dost' blízko nějakého kořene.
- **Tvrzení:** Jestliže konverguje, pak konverguje **superlineárně**.

Příklad pro $n = 1$: Babylónská metoda na výpočet druhé odmocniny

Řešíme

$$g(x) = x^2 - a = 0$$

Iterace:

$$x_{k+1} = x_k - g'(x_k)^{-1}g(x_k) = x_k - \frac{x_k^2 - a}{2x_k} = \frac{1}{2} \left(x_k + \frac{a}{x_k} \right)$$

Matlab:

```
a=4; x=a;
for i=1:5
    x=(x+a/x)/2;
    fprintf('x=%.12g  g=%.12g\n',x,x^2-a);
end
```

x=2.5 g=2.25

x=2.05 g=0.2025

x=2.0006097561 g=0.00243939619274

x=2.00000009292 g=3.71689187872e-07

x=2 g=8.881784197e-15

Příklad pro $n = 2$: Průsečík dvou rovinných křivek

Najdi řešení soustavy dvou rovnic o dvou neznámých:

$$(x - 1)^2 + y^2 = 1$$

$$x^4 + y^4 = 1$$

Iterace:

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \end{bmatrix} = \begin{bmatrix} x_k \\ y_k \end{bmatrix} - \underbrace{\begin{bmatrix} 2(x_k - 1) & 2y_k \\ 4x_k^3 & 4y_k^3 \end{bmatrix}}_{g'(x_k, y_k)}^{-1} \underbrace{\begin{bmatrix} (x_k - 1)^2 + y_k^2 - 1 \\ x_k^4 + y_k^4 - 1 \end{bmatrix}}_{g(x_k, y_k)}$$

```
x = [1;1];  
for iter = 1:5  
    g = [(x(1)-1)^2+x(2)^2-1; x(1)^4+x(2)^4-1];  
    x = x - [2*(x(1)-1) 2*x(2); 4*x.^3] \ g;  
    fprintf('x=(%.12g,%.12g) g=(%.12g,%.12g)\n', x, g);  
end
```

x=(0.75,1) g=(0.0625,0.31640625)

x=(0.678779069767,0.950944767442) g=(0.00747883674452,0.0300334591266)

x=(0.671937746776,0.944701508411) g=(8.57819836066e-05,0.000339082408219)

x=(0.671859761262,0.944629025098) g=(1.13355711484e-08,4.46057650816e-08)

x=(0.671859751039,0.944629015546) g=(0,8.881784197e-16)

Použití Newtonovy metody na minimalizaci funkce

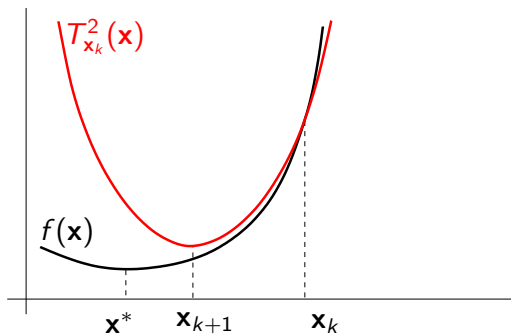
Hledáme lokální minimum dvakrát diferencovatelné funkce $f: \mathbb{R}^n \rightarrow \mathbb{R}$.

Iterace: Řešíme stacionární podmínku $\mathbf{g}(\mathbf{x}) = f'(\mathbf{x})^T = \mathbf{0}$ Newtonovou metodou:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - f''(\mathbf{x}_k)^{-1} f'(\mathbf{x}_k)^T$$

Jiné odvození: Minimalizujeme Taylorův polynom $T_{\mathbf{x}_k}^2(\mathbf{x}_{k+1})$ funkce f stupně 2 v okolí \mathbf{x}_k

$$T_{\mathbf{x}_k}^2(\mathbf{x}_{k+1}) = f(\mathbf{x}_k) + f'(\mathbf{x}_k)(\mathbf{x}_{k+1} - \mathbf{x}_k) + \frac{1}{2}(\mathbf{x}_{k+1} - \mathbf{x}_k)^T f''(\mathbf{x}_k)(\mathbf{x}_{k+1} - \mathbf{x}_k)$$



Pro lepší konvergenci lze přidat součinitel délky kroku:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k \underbrace{f''(\mathbf{x}_k)^{-1} f'(\mathbf{x}_k)^T}_{-\mathbf{v}_k}$$

Pro $\alpha_k = 1$ se metoda nazývá **čistá** Newtonova metoda.

- **Newtonův směr** $\mathbf{v}_k = -f''(\mathbf{x}_k)^{-1} f'(\mathbf{x}_k)^T$ je sestupný, když $f''(\mathbf{x}_k)$ je pozitivně definitní:

$$f'(\mathbf{x}_k) \mathbf{v}_k = -f'(\mathbf{x}_k) f''(\mathbf{x}_k)^{-1} f'(\mathbf{x}_k)^T < 0$$

- Pak je výhodné soustavu $f''(\mathbf{x}_k) \mathbf{v}_k = -f'(\mathbf{x}_k)^T$ řešit Choleského rozkladem.

Kvasi-newtonovské metody

- Počítání Hessiánu $f''(\mathbf{x}_k)$ a řešení soustavy $f''(\mathbf{x}_k)\mathbf{v}_k = -f'(\mathbf{x}_k)^T$ je drahé.
- Myšlenka kvasi-newtonových metod:
 $f''(\mathbf{x}_k)$ aproximujeme pomocí minulých hodnot $f(\mathbf{x}_i)$ a $f'(\mathbf{x}_i)$, $i \leq k$.
- Speciálně pro $n = 1$ proměnnou: metoda sečen (vs. metoda tečen = Newtonova metoda)
- Populární metoda: **Broyden–Fletcher–Goldfarb–Shanno (BFGS)**
Matlabská funkce `fminunc`.

Nelineární úloha nejmenších čtverců

Nelineární úloha nejmenších čtverců

Pro dané diferencovatelné zobrazení $\mathbf{g}: \mathbb{R}^n \rightarrow \mathbb{R}^m$ minimalizujeme funkci

$$f(\mathbf{x}) = \|\mathbf{g}(\mathbf{x})\|^2 = \mathbf{g}(\mathbf{x})^T \mathbf{g}(\mathbf{x}) = \sum_{i=1}^m g_i(\mathbf{x})^2$$

- Tedy minimalizujeme součet čtverců daných funkcí.
- Speciální případ je lineární úloha nejmenších čtverců: $\mathbf{g}(\mathbf{x}) = \mathbf{Ax} - \mathbf{b}$ je afinní.

Gaussova-Newtonova metoda

Iterace: Známe \mathbf{x}_k a najdeme \mathbf{x}_{k+1} které minimalizuje funkci

$$\|\mathbf{T}_{\mathbf{x}_k}^1(\mathbf{x}_{k+1})\|^2 = \|\mathbf{g}(\mathbf{x}_k) + \mathbf{g}'(\mathbf{x}_k)(\mathbf{x}_{k+1} - \mathbf{x}_k)\|^2$$

To je lineární úloha nejmenších čtverců s řešením

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \mathbf{g}'(\mathbf{x}_k)^+ \mathbf{g}(\mathbf{x}_k)$$

Můžeme přidat délku kroku:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k \underbrace{\mathbf{g}'(\mathbf{x}_k)^+ \mathbf{g}(\mathbf{x}_k)}_{-\mathbf{v}_k}$$

Pro $\alpha_k = 1$ se metoda nazývá **čistá** Gaussova-Newtonova metoda.

Gaussův-Newtonův směr

$$\mathbf{v}_k = -\mathbf{g}'(\mathbf{x}_k)^+ \mathbf{g}(\mathbf{x}_k) = -(\mathbf{g}'(\mathbf{x}_k)^T \mathbf{g}'(\mathbf{x}_k))^{-1} \underbrace{\mathbf{g}'(\mathbf{x}_k)^T \mathbf{g}(\mathbf{x}_k)}_{\frac{1}{2} f'(\mathbf{x}_k)^T}$$

je vždy sestupný:

$$f'(\mathbf{x}_k) \mathbf{v}_k = -\frac{1}{2} f'(\mathbf{x}_k) (\mathbf{g}'(\mathbf{x}_k)^T \mathbf{g}'(\mathbf{x}_k))^{-1} f'(\mathbf{x}_k)^T < 0$$

Rozdíl oproti Newtonově metodě

Minimalizace funkce $f(\mathbf{x}) = \|\mathbf{g}(\mathbf{x})\|^2$ Newtonovou vs. Gaussovou-Newtonovou metodou:

- Gaussův-Newtonův směr je

$$\mathbf{v}_k = -\frac{1}{2}(\mathbf{g}'(\mathbf{x}_k)^T \mathbf{g}'(\mathbf{x}_k))^{-1} f'(\mathbf{x}_k)^T$$

- Newtonův směr je

$$\begin{aligned} \mathbf{v}_k &= -f''(\mathbf{x}_k)^{-1} f'(\mathbf{x}_k)^T \\ &= -\frac{1}{2} \left(\mathbf{g}'(\mathbf{x}_k)^T \mathbf{g}'(\mathbf{x}_k) + \sum_{i=1}^m g_i(\mathbf{x}_k) g_i''(\mathbf{x}_k) \right)^{-1} f'(\mathbf{x}_k)^T \end{aligned}$$

Gaussova-Newtonova metoda zanedbává **členy druhého řádu** v Hessiánu $f''(\mathbf{x})$.

- Proto obvykle konverguje o něco pomaleji než Newtonova metoda.
- Ale výhoda: nemusíme počítat Hessiány $g_i''(\mathbf{x}_k)$

Příklad: Přibližný průsečík tří rovinných křivek

Najdi přibližné řešení soustavy tří rovnic o dvou neznámých:

$$(x - 1)^2 + y^2 = 1$$

$$x^4 + y^4 = 1$$

$$x^2 + (y - 1)^2 = \frac{1}{2}$$

Tedy minimalizujeme funkci

$$f(x, y) = \mathbf{g}(x, y)^T \mathbf{g}(x, y) = ((x - 1)^2 + y^2 - 1)^2 + (x^4 + y^4 - 1)^2 + (x^2 + (y - 1)^2 - \frac{1}{2})^2$$

Iterace čisté Gaussovy-Newtonovy metody:

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \end{bmatrix} = \begin{bmatrix} x_k \\ y_k \end{bmatrix} - \underbrace{\begin{bmatrix} 2(x_k - 1) & 2y_k \\ 4x_k^3 & 4y_k^3 \\ 2x_k & 2(y_k - 1) \end{bmatrix}}_{\mathbf{g}'(x_k, y_k)} \underbrace{\begin{bmatrix} (x_k - 1)^2 + y_k^2 - 1 \\ x_k^4 + y_k^4 - 1 \\ x_k^2 + (y_k - 1)^2 - \frac{1}{2} \end{bmatrix}}_{\mathbf{g}(x_k, y_k)}$$

Matlab:

```
x = [1;1];
for iter = 1:10
    g = [(x(1)-1)^2+x(2)^2-1; x(1)^4+x(2)^4-1; x(1)^2+(x(2)-1)^2-.5];
    x = x - [2*(x(1)-1) 2*x(2); 4*x'.^3; 2*x(1) 2*(x(2)-1)] \ g;
    fprintf('x=(%.12g,%.12g) g=(%.12g,%.12g,%.12g)\n',x,g);
end

x=(0.75,1) g=(0,1,0.5)
x=(0.696777860013,0.945770115246) g=(0.0625,0.31640625,0.0625)
x=(0.691092552216,0.940578214706) g=(-0.0135752229284,0.0358061117489,-0.0115597333957)
x=(0.691002680826,0.94054818438) g=(-0.0198888107249,0.0107820331379,-0.0188601357041)
x=(0.691002154829,0.940548357781) g=(-0.0198897696029,0.0105634499047,-0.0189807767107)
x=(0.691002152527,0.940548357855) g=(-0.0198891183559,0.0105633328127,-0.0189815242591)
x=(0.691002152516,0.940548357857) g=(-0.0198891167934,0.0105633300224,-0.0189815274491)
x=(0.691002152516,0.940548357857) g=(-0.0198891167821,0.0105633300147,-0.0189815274653)
x=(0.691002152516,0.940548357857) g=(-0.019889116782,0.0105633300146,-0.0189815274654)
x=(0.691002152516,0.940548357857) g=(-0.019889116782,0.0105633300146,-0.0189815274654)
```

Levenbergova-Marquardtova metoda

Zopakujme lineární úlohu nejmenších čtverců s regularizací:

- $\|\mathbf{Ax} - \mathbf{b}\|^2 + \mu\|\mathbf{x}\|^2$ (kde $\mu > 0$), nabývá minima pro $\mathbf{x} = (\mathbf{A}^T\mathbf{A} + \mu\mathbf{I})^{-1}\mathbf{A}^T\mathbf{b}$.
- Matice $\mathbf{A}^T\mathbf{A} + \mu\mathbf{I}$ je regulární pro každou \mathbf{A} .

Iterace L-M metody: Známe \mathbf{x}_k a najdeme \mathbf{x}_{k+1} které minimalizuje funkci

$$\|\mathbf{g}(\mathbf{x}_k) + \mathbf{g}'(\mathbf{x}_k)(\mathbf{x}_{k+1} - \mathbf{x}_k)\|^2 + \mu_k\|\mathbf{x}_{k+1} - \mathbf{x}_k\|^2.$$

Ta nabývá minimum pro

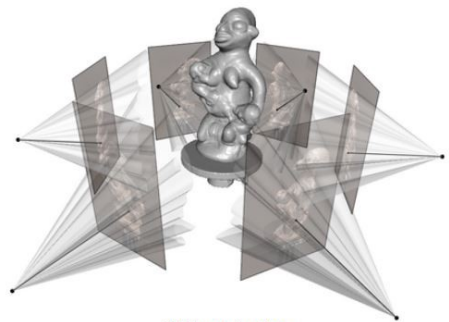
$$\mathbf{x}_{k+1} = \mathbf{x}_k - (\mathbf{g}'(\mathbf{x}_k)^T\mathbf{g}'(\mathbf{x}_k) + \mu_k\mathbf{I})^{-1}\mathbf{g}'(\mathbf{x}_k)^T\mathbf{g}(\mathbf{x}_k).$$

- Pro $\mu_k \approx 0$ se blíží Gaussově-Newtonově iteraci.
- Pro $\mu_k \gg 0$ se blíží iteraci gradientní metody: $\mathbf{x}_{k+1} \approx \mathbf{x}_k - \frac{1}{\mu_k} \underbrace{\mathbf{g}'(\mathbf{x}_k)^T\mathbf{g}(\mathbf{x}_k)}_{\frac{1}{2}f'(\mathbf{x}_k)^T}$

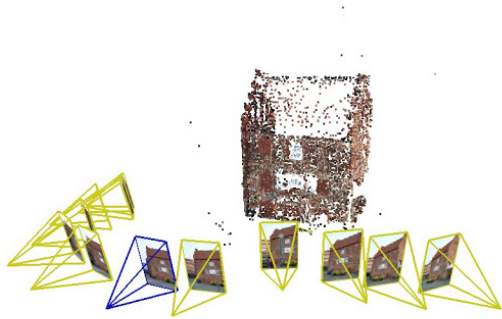
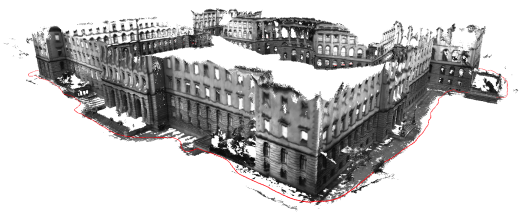
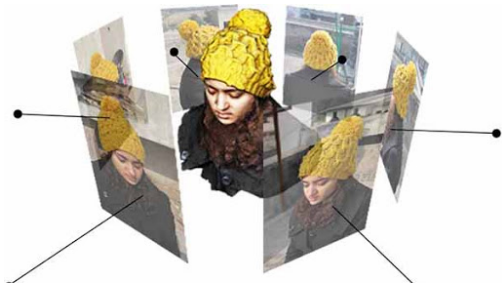
Heuristika pro volbu μ_k (zároveň nahrazuje line search):

- Pokud $f(\mathbf{x}_{k+1}) < f(\mathbf{x}_k)$, iteraci přijmeme a μ_k zmenšíme (např. 2 krát).
- Pokud $f(\mathbf{x}_{k+1}) \geq f(\mathbf{x}_k)$, iteraci odmítneme a μ_k zvětšíme (např. 2 krát).

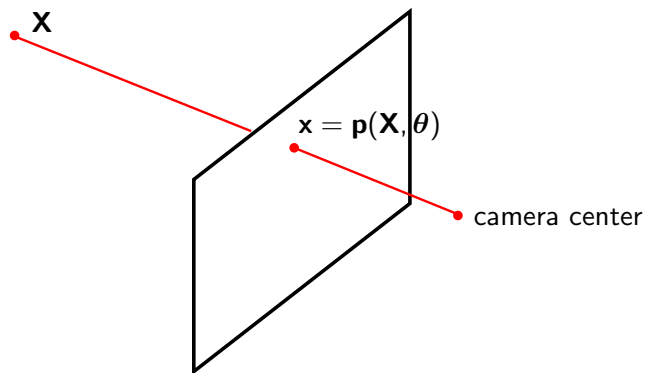
Příklad: Rekonstrukce scény z fotografií



3D Reconstruction



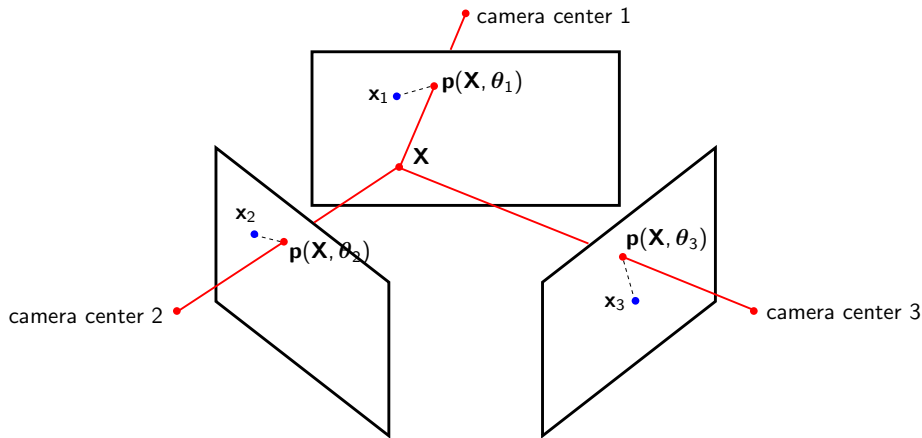
Zobrazení bodu v prostoru kamerou



- $\mathbf{X} \in \mathbb{R}^3$ je bod ve scéně
- $\mathbf{x} \in \mathbb{R}^2$ je obraz bodu \mathbf{X} v kameře
- θ jsou parametry kamery (poloha, natočení, zoom, radiální zkreslení, ...)

Rekonstrukce jednoho bodu

Známe obrazy $\mathbf{x}_1, \dots, \mathbf{x}_m$ bodu \mathbf{X} v kamerách a parametry kamer $\theta_1, \dots, \theta_m$. Odhadni bod \mathbf{X} .



Řešení:

1. Najdi počáteční odhad bodu \mathbf{X} vhodnou lineární metodou.

2. Upřesni odhad \mathbf{X} minimalizací funkce $\sum_{i=1}^m \|\mathbf{p}(\mathbf{X}, \theta_i) - \mathbf{x}_i\|^2$.

Rekonstrukce bodů a kamer

Známe obrazy \mathbf{x}_{ij} bodů \mathbf{X}_j v kamerách θ_i (kde $i = 1, \dots, m$, $j = 1, \dots, n$).
Odhadni body $\mathbf{X}_1, \dots, \mathbf{X}_n$ a parametry $\theta_1, \dots, \theta_m$.

Řešení:

1. Najdi počáteční odhad $\mathbf{X}_1, \dots, \mathbf{X}_n$ a $\theta_1, \dots, \theta_m$ vhodnou lineární metodou.

2. Upřesni odhad minimalizací funkce
$$\sum_{i=1}^m \sum_{j=1}^n \|\mathbf{p}(\mathbf{X}_j, \theta_i) - \mathbf{x}_{ij}\|^2$$

Tento krok je známý jako **bundle adjustment**.

Přehled probraných metod

úloha	metoda	iterace metody
$\min_{\mathbf{x}} f(\mathbf{x})$	obecná sestupná	$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{v}_k$
$\min_{\mathbf{x}} f(\mathbf{x})$	gradient	$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k f'(\mathbf{x}_k)^T$
$\mathbf{g}(\mathbf{x}) = \mathbf{0}$	Newton	$\mathbf{x}_{k+1} = \mathbf{x}_k - \mathbf{g}'(\mathbf{x}_k)^{-1} \mathbf{g}(\mathbf{x}_k)$
$\min_{\mathbf{x}} f(\mathbf{x})$	Newton	$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k f''(\mathbf{x}_k)^{-1} f'(\mathbf{x}_k)^T$
$\min_{\mathbf{x}} \underbrace{\ \mathbf{g}(\mathbf{x})\ ^2}_{f(\mathbf{x})}$	Gauss-Newton	$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k \mathbf{g}'(\mathbf{x}_k)^+ \mathbf{g}(\mathbf{x}_k)$ $= \mathbf{x}_k - \alpha_k \frac{1}{2} (\mathbf{g}'(\mathbf{x}_k)^T \mathbf{g}'(\mathbf{x}_k))^{-1} f'(\mathbf{x}_k)^T$
$\min_{\mathbf{x}} \underbrace{\ \mathbf{g}(\mathbf{x})\ ^2}_{f(\mathbf{x})}$	Levenberg-Marquardt	$\mathbf{x}_{k+1} = \mathbf{x}_k - \frac{1}{2} (\mathbf{g}'(\mathbf{x}_k)^T \mathbf{g}'(\mathbf{x}_k) + \mu_k \mathbf{I})^{-1} f'(\mathbf{x}_k)^T$

Neučte se vzorečky nazpaměť, ale naučte se je odvodit!

Hledejte v tabulce souvislosti!