

# Počítačové vidění

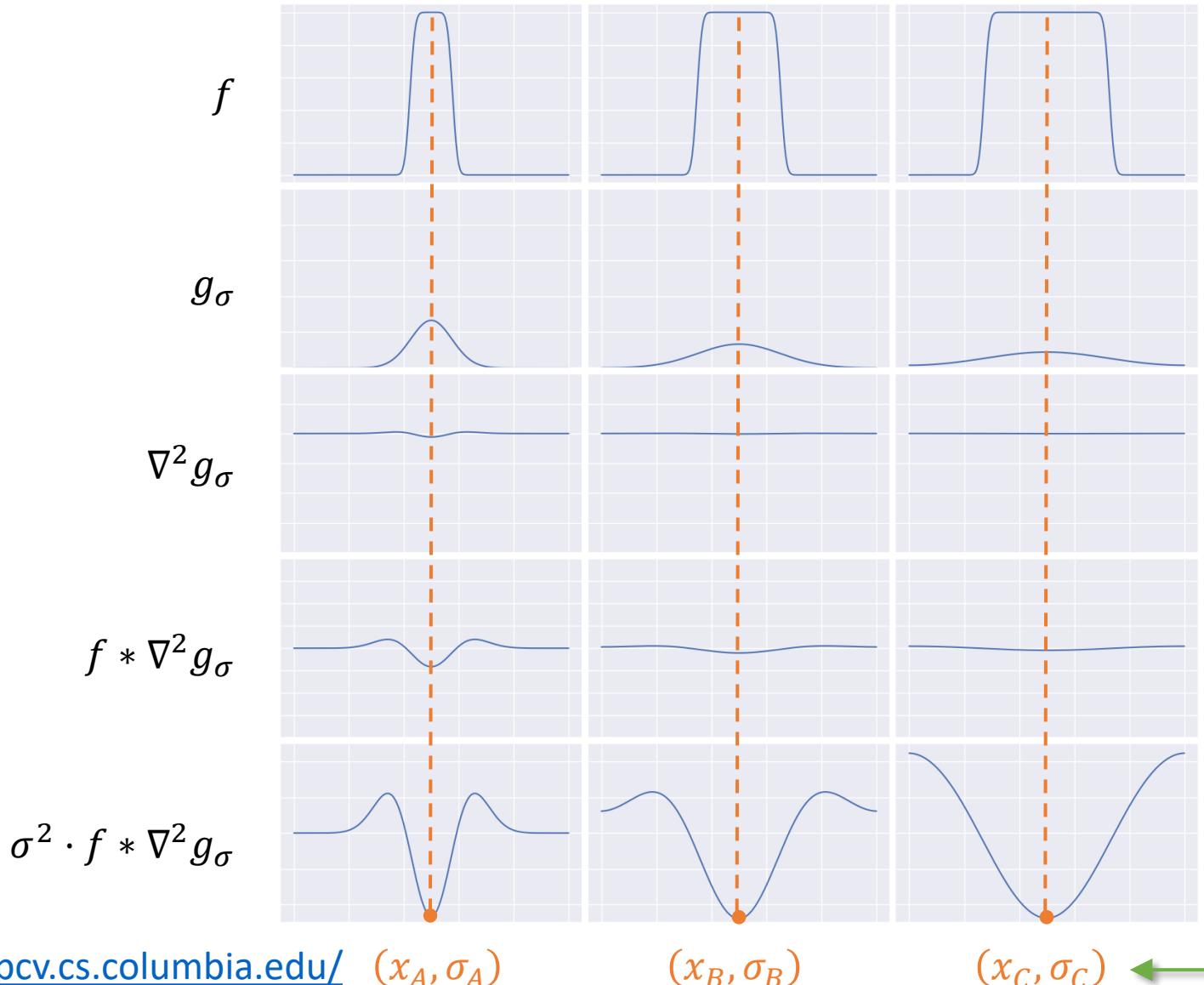
---

Podobnost obrazů na základě lokálních korespondencí

# Minule



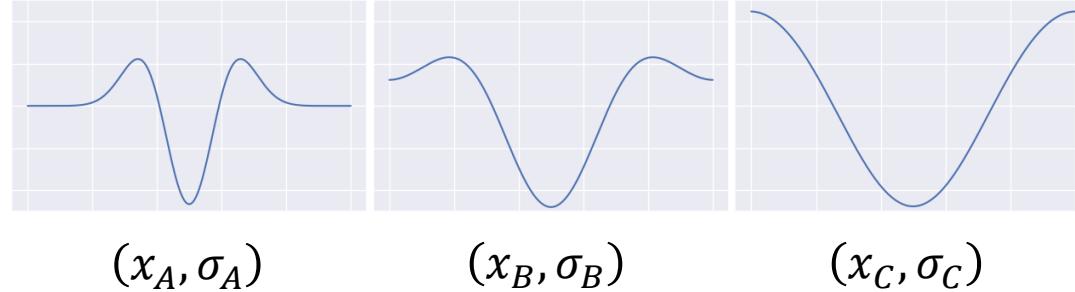
# Minule



Zvýšíme  $\sigma$

Bloby jsou lokalizovány pozicí  
a měřítkem = tzv. prostor  
měřítek (scale space)

# Minule



$x_{\{A,B,C\}}$  ... **Pozice blobu** v prostoru (v obrazu pokud 2D)

$\sigma_{\{A,B,C\}}$  ... **Charakteristické měřítko** blobu, tj.  $\sigma$ , pro kterou měřítkem normalizovaná odezva je maximální

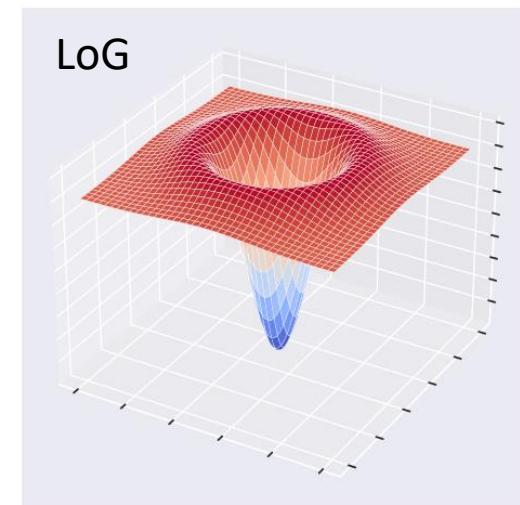
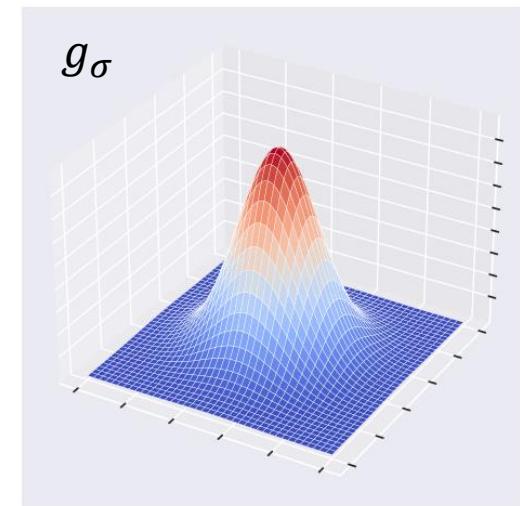
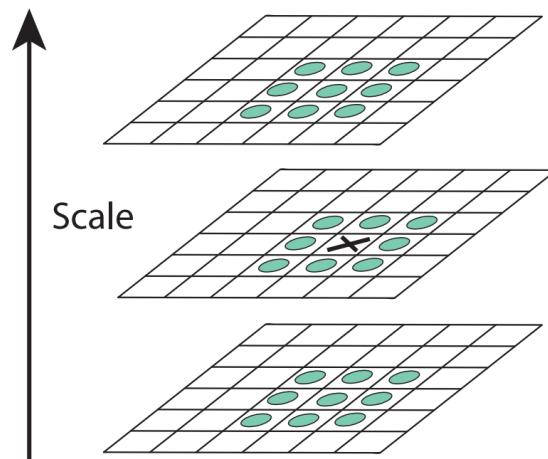
Charakteristické měřítko  $\propto$  velikost blobu

# Minule

- Obrázek  $I(x, y)$  konvolujeme s LoG filtry  $\nabla^2 g_\sigma$  s různými  $\sigma = \sigma_1, \dots, \sigma_S$

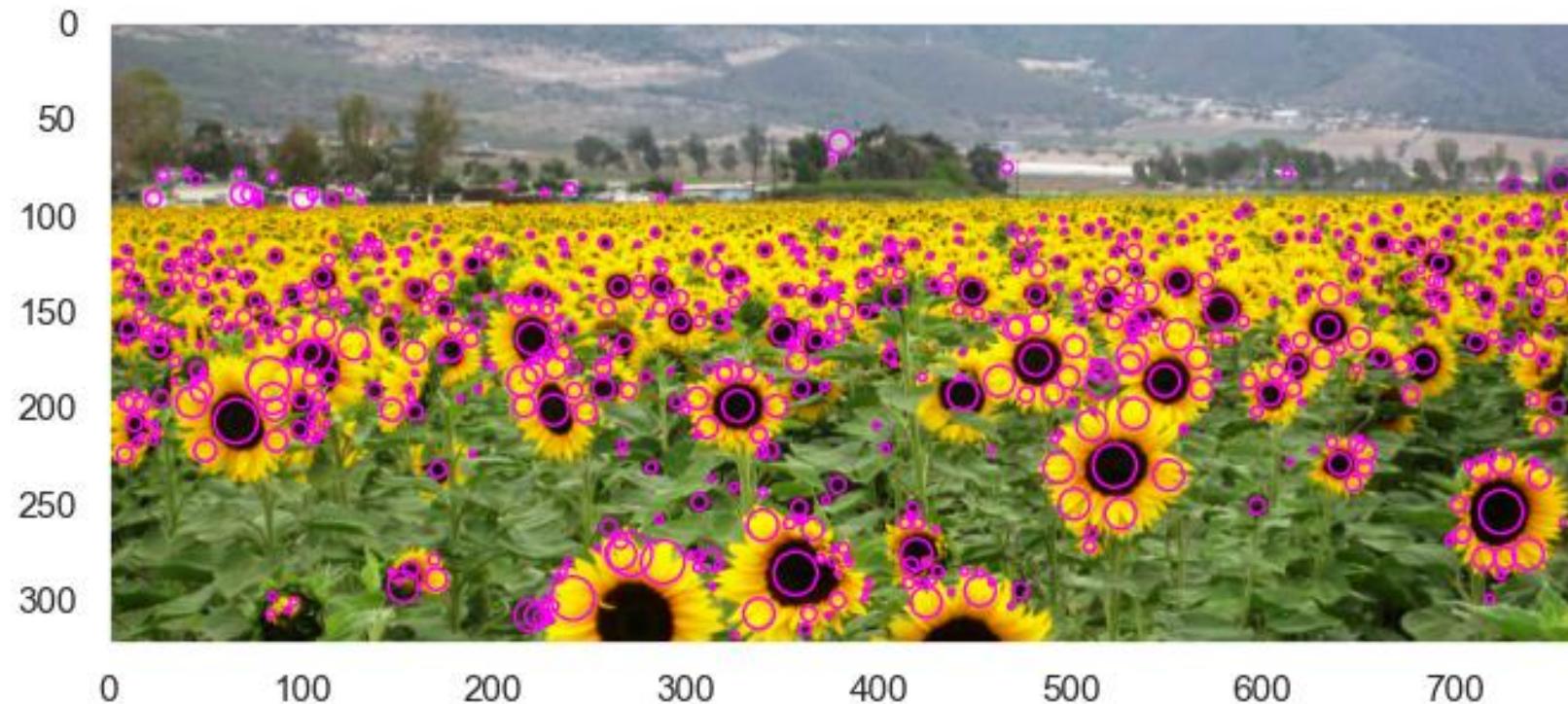
$$\nabla_{\text{norm}}^2 L(x, y, \sigma) = \sigma^2 \cdot I(x, y) * \nabla^2 g_\sigma$$

- Bloby, tj. jejich pozice a velikost, najdeme lokální maxima funkce  $\nabla_{\text{norm}}^2 L(x, y, \sigma)$

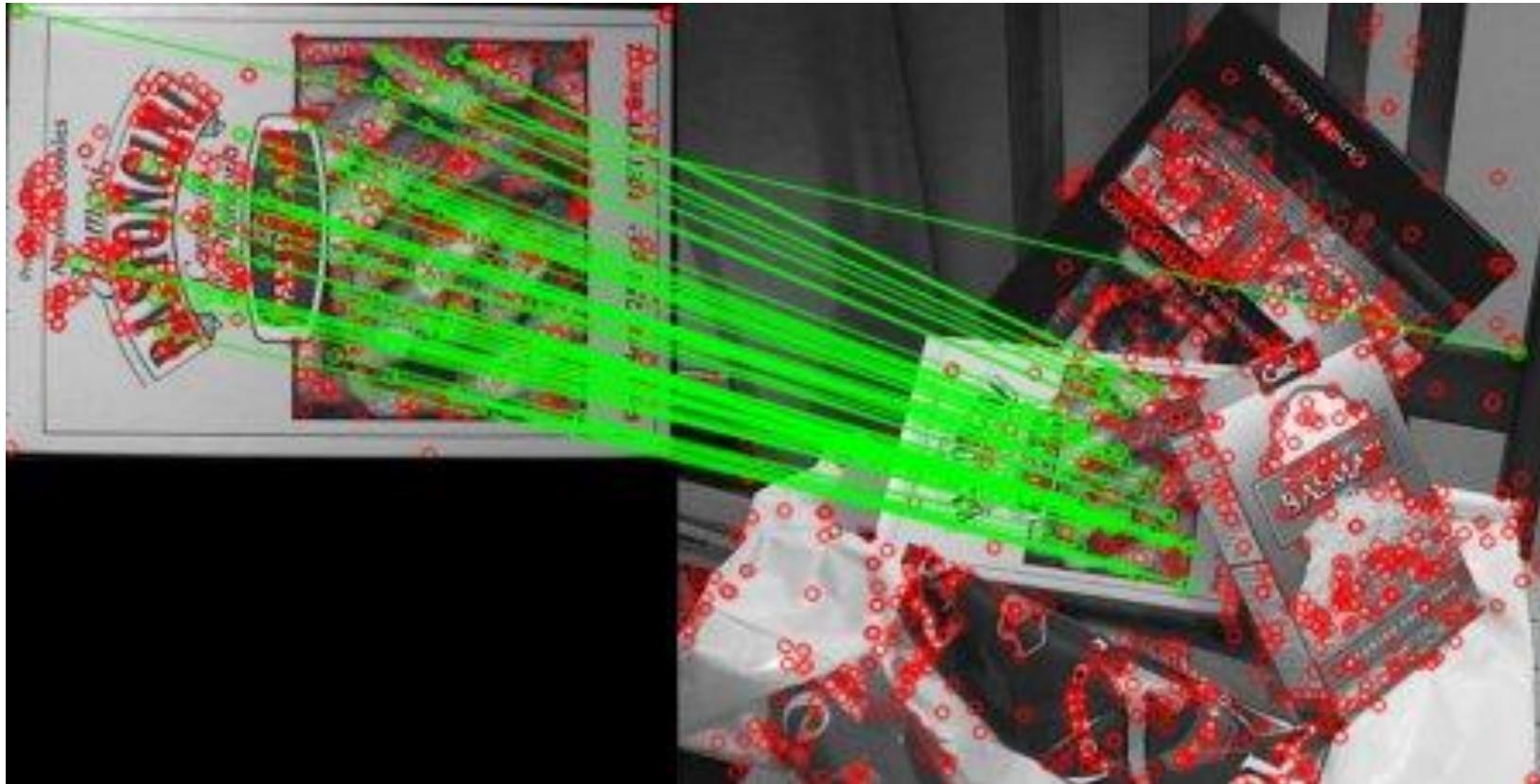


obrázek: <https://www.cs.ubc.ca/~lowe/papers/ijcv04.pdf>

Minule



Dnes



obrázek: [https://docs.opencv.org/3.4/dc/dc3/tutorial\\_py\\_matcher.html](https://docs.opencv.org/3.4/dc/dc3/tutorial_py_matcher.html)

# SIFT detektor

---

# Lokální příznaky SIFT (1999)

- Scale Invariant Feature Transform
- [Distinctive image features from scale-invariant keypoints \(2004\)](#)
- Přes 70000 citací na Google Scholar
- Původní článek z roku 1999 jeden z prvních na téma rozpoznávání podobnosti obrázků na základě korespondence lokálních příznaků
- Velmi úspěšná & robustní metoda

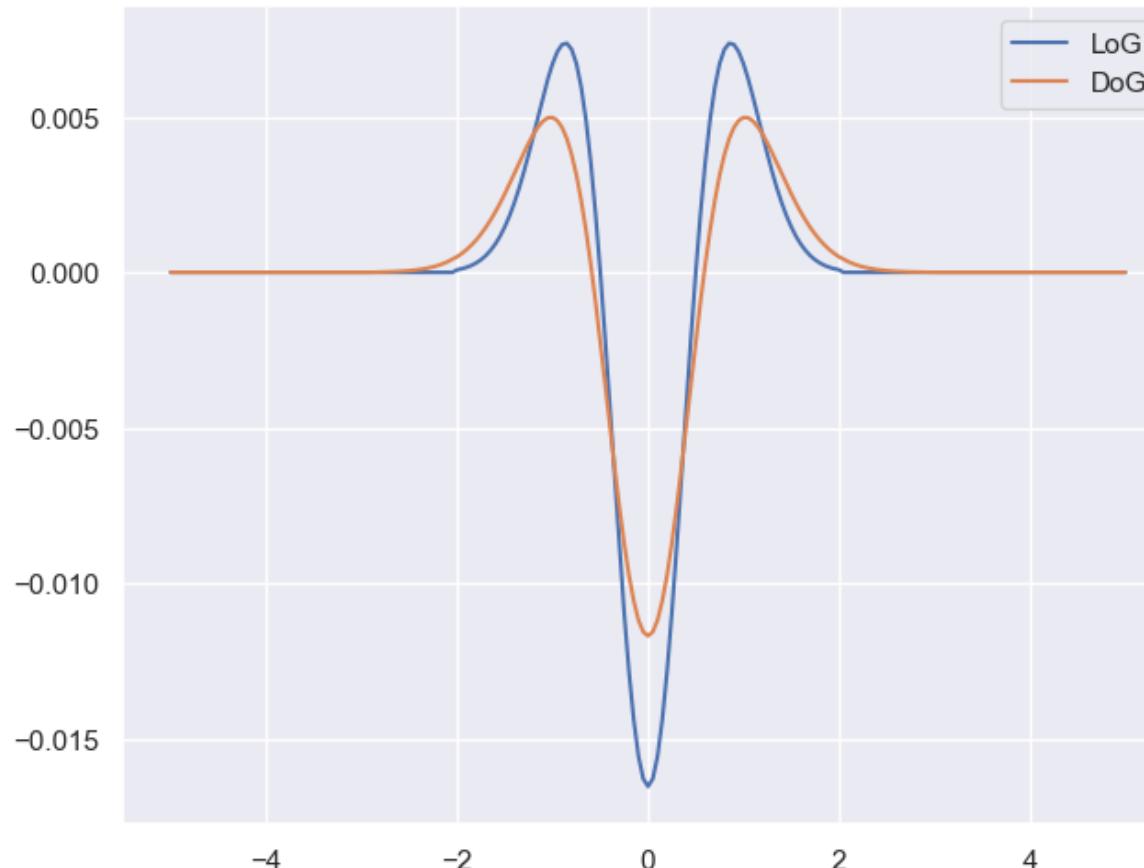
# SIFT přehled

1. Detekce lokálních maxim v prostoru měřítek jako možných zájmových bodů
2. Lokalizace a filtrace zájmových bodů
3. Přiřazení orientace zájmovým bodům
4. Výpočet deskriptorů z okolí zájmových bodů

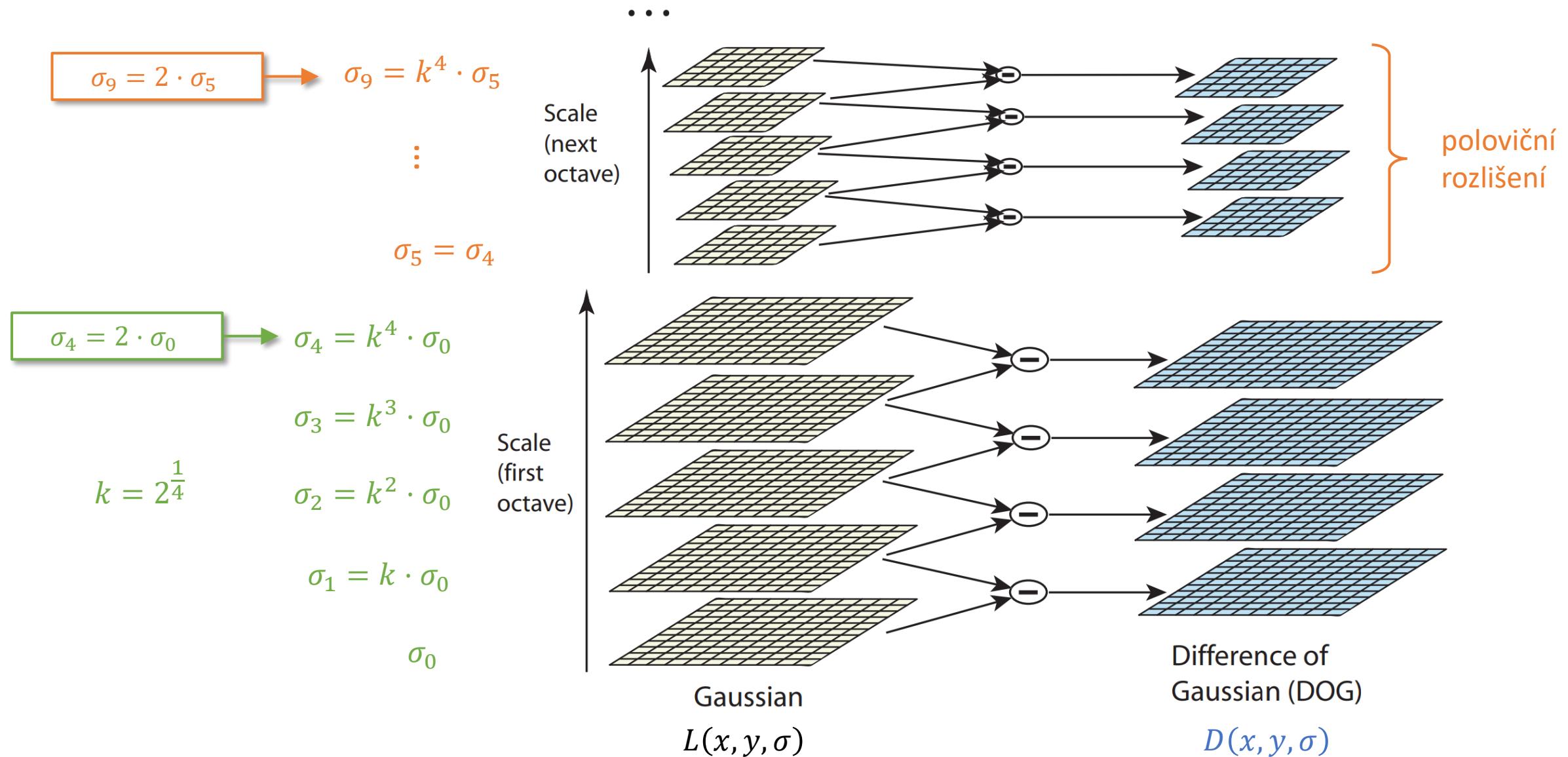
# Detekce: aproximace NLoG rozdílem gaussiánů (DoG)

$$(k - 1) \cdot \sigma^2 \cdot \nabla^2 L(x, y, \sigma) \approx L(x, y, k \cdot \sigma) - L(x, y, \sigma)$$

Laplacian of Gaussians (LoG)      Difference of Gaussians (DoG)

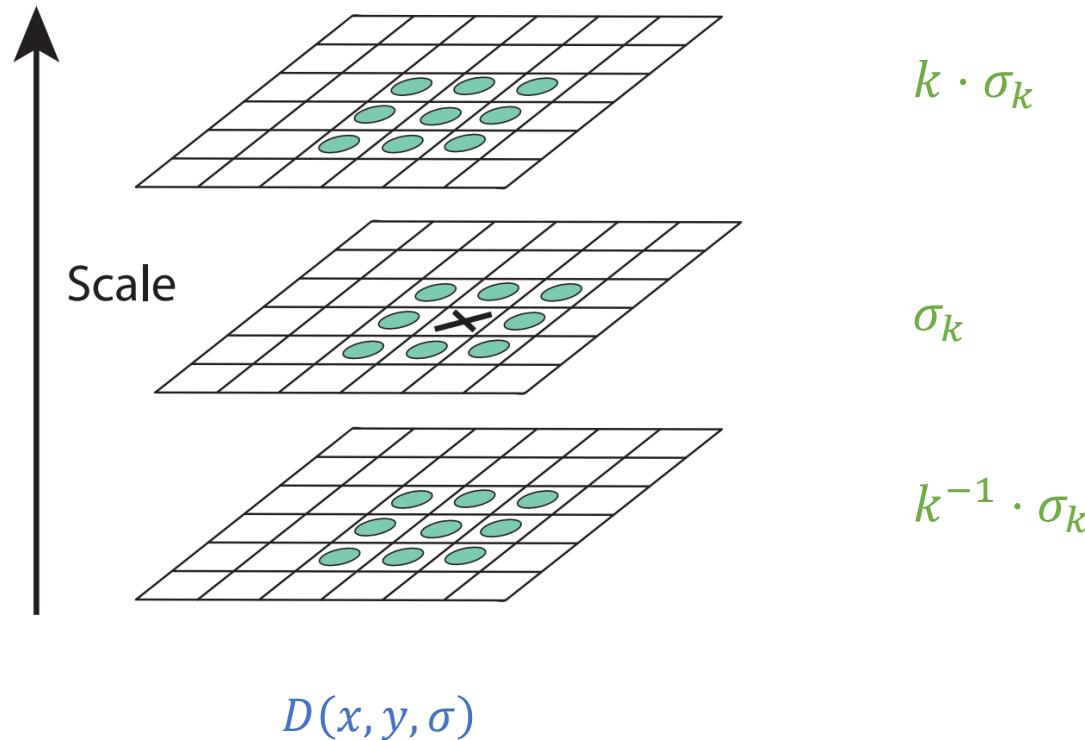


# Detekce: approximace LoG rozdílem gaussiánů (DoG)

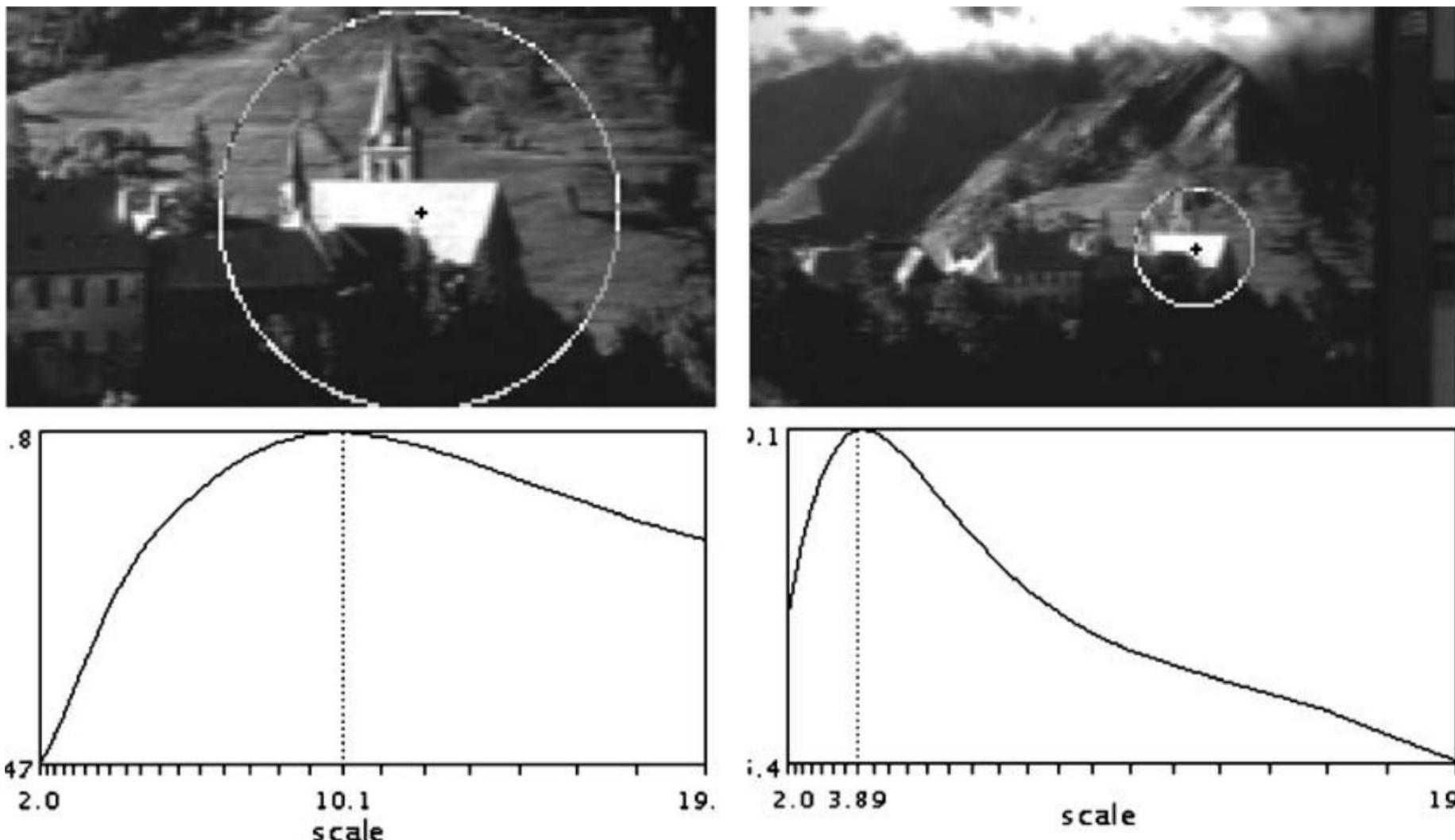


# Detekce lokálních maxim v prostoru měřítek

Bod je označen jako kandidát, pokud představuje lokální maximum nebo minimum v okolí  $3 \times 3 \times 3$

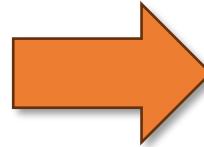
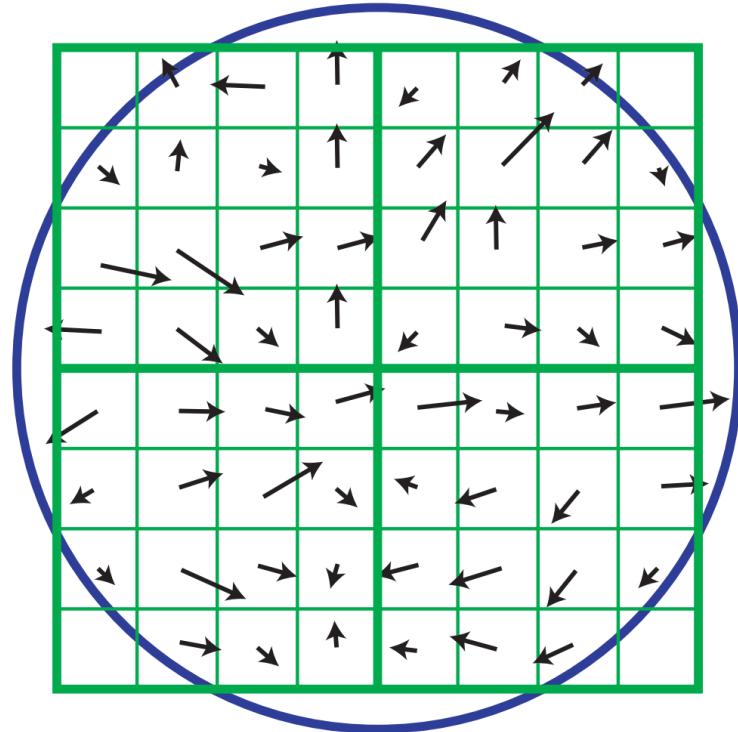


# Invariance vůči měřítku

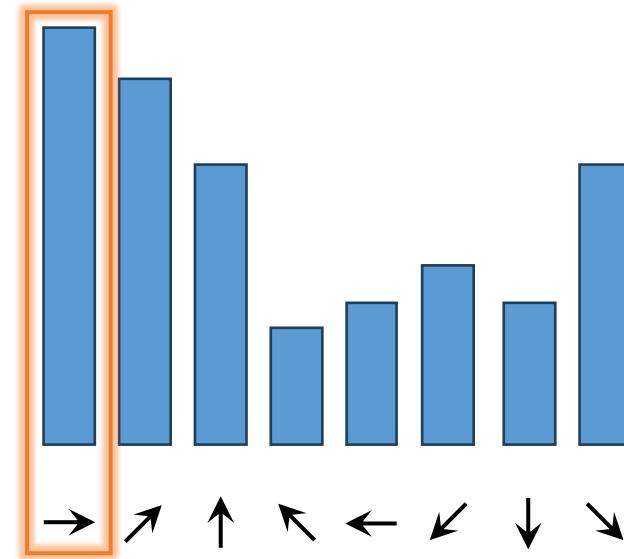


obrázek: [Mikolajczyk & Schmid, "Scale & Affine Invariant Interest Point Detectors"](#)

# Orientace SIFT zájmových bodů (oblastí)

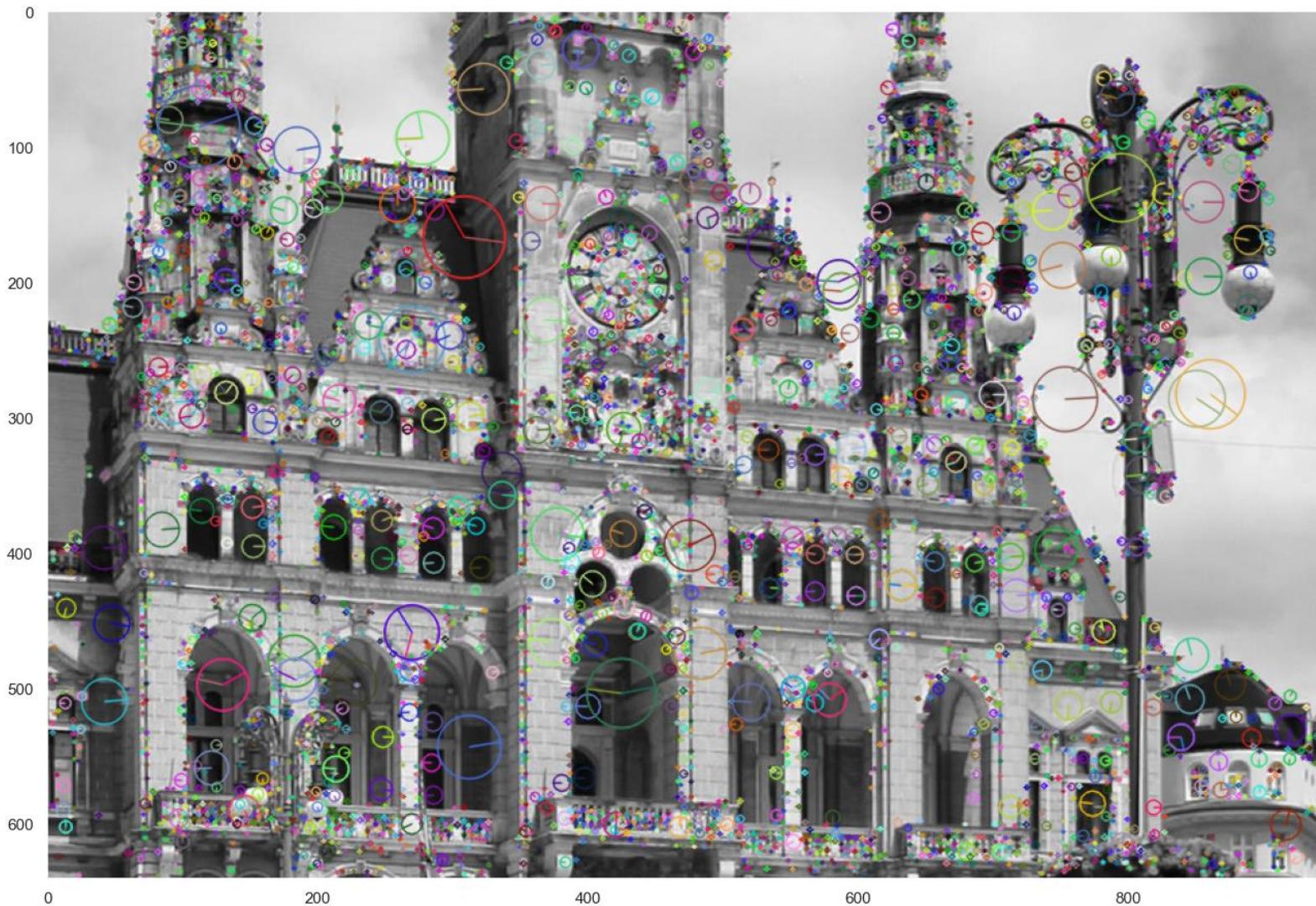


histogram orientací gradientů



položka histogramu s maximální hodnotou určuje orientaci bodu

# Příklad detekce SIFT zájmových bodů



# Implementační detaily a triky v detektoru SIFT

- Počet úrovní měřítek v oktávě  $k = 4$ 
  - Zjištěno empiricky, body stabilní a opakovatelné
  - Navíc kvůli hledání lokálních extrémů se musí vytvořit  $k + 3$  úrovní
  - S každou další oktávou se rozlišení sníží na polovinu, ale přesto nedojde ke ztrátě informace
- Sub-pixelová lokalizace bodů v prostoru i měřítku
  - Okolí každého bodu v  $D$  je approximováno Taylorovým polynomem druhého rádu (paraboloidem)
$$D(\mathbf{x}) = D + \frac{\partial D^\top}{\partial \mathbf{x}} \cdot \mathbf{x} + \frac{1}{2} \cdot \mathbf{x}^\top \cdot \frac{\partial^2 D}{\partial \mathbf{x}^2} \cdot \mathbf{x}$$
  - Paraboloid má minimum v
$$\mathbf{x}^* = -\frac{\partial^2 D^{-1}}{\partial \mathbf{x}^2} \cdot \frac{\partial D}{\partial \mathbf{x}}$$

# Implementační detaily a triky v detektoru SIFT

- Eliminace bodů podél hran

- Založeno na výpočtu Hessiánu (matice druhých derivací) okolí bodu
  - Bod je ponechán, pouze pokud

$$\frac{\text{trace}(\mathbf{H})^2}{\det(\mathbf{H})} < \frac{(r+1)^2}{r}$$

kde  $r = 10$  je empiricky zjištěný práh poměru vlastních čísel Hessiánu  $\mathbf{H}$

- Přiřazení orientace bodům

- Histogram má 36 binů
  - Každý příspěvek do histogramu je vážen gaussovským oknem se středem nad bodem a  $\sigma_g = 1.5 \cdot \sigma$
  - Okolí maxima v histogramu se proloží parabolou a výsledná orientace je pozice jejího maxima
  - Pokud existuje druhé lokální maximum v histogramu, jehož hodnota je 80 % absolutního maxima, je vytvořena kopie bodu na stejně pozici s touto orientací

# Implementační detaily a triky v SIFT

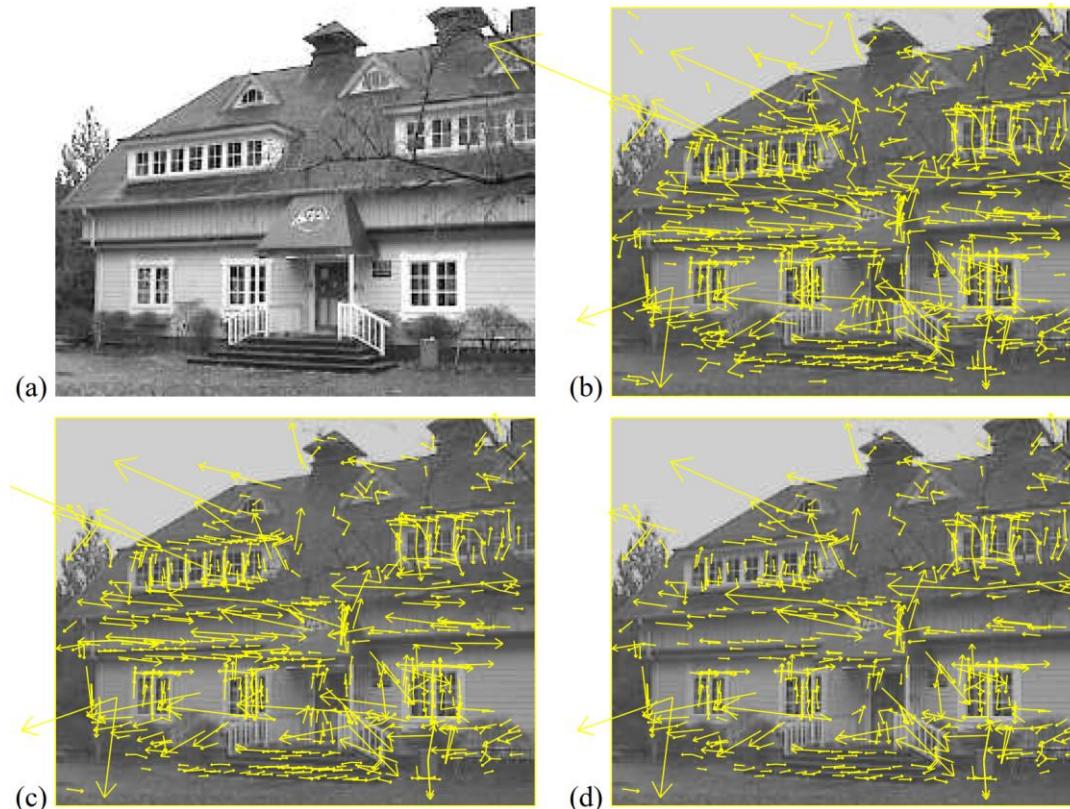
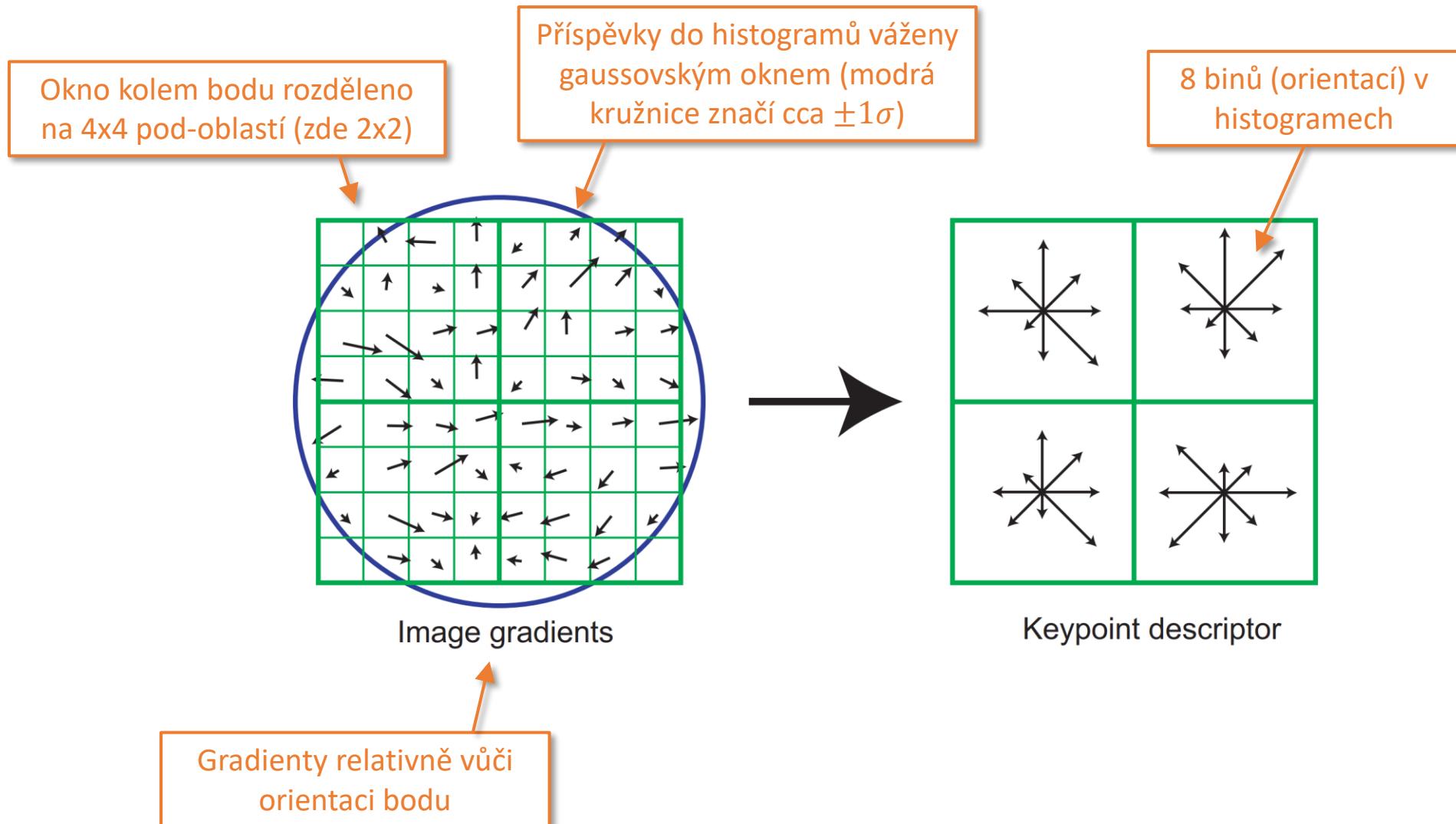


Figure 5: This figure shows the stages of keypoint selection. (a) The 233x189 pixel original image. (b) The initial 832 keypoints locations at maxima and minima of the difference-of-Gaussian function. Keypoints are displayed as vectors indicating scale, orientation, and location. (c) After applying a threshold on minimum contrast, 729 keypoints remain. (d) The final 536 keypoints that remain following an additional threshold on ratio of principal curvatures.

# SIFT deskriptor

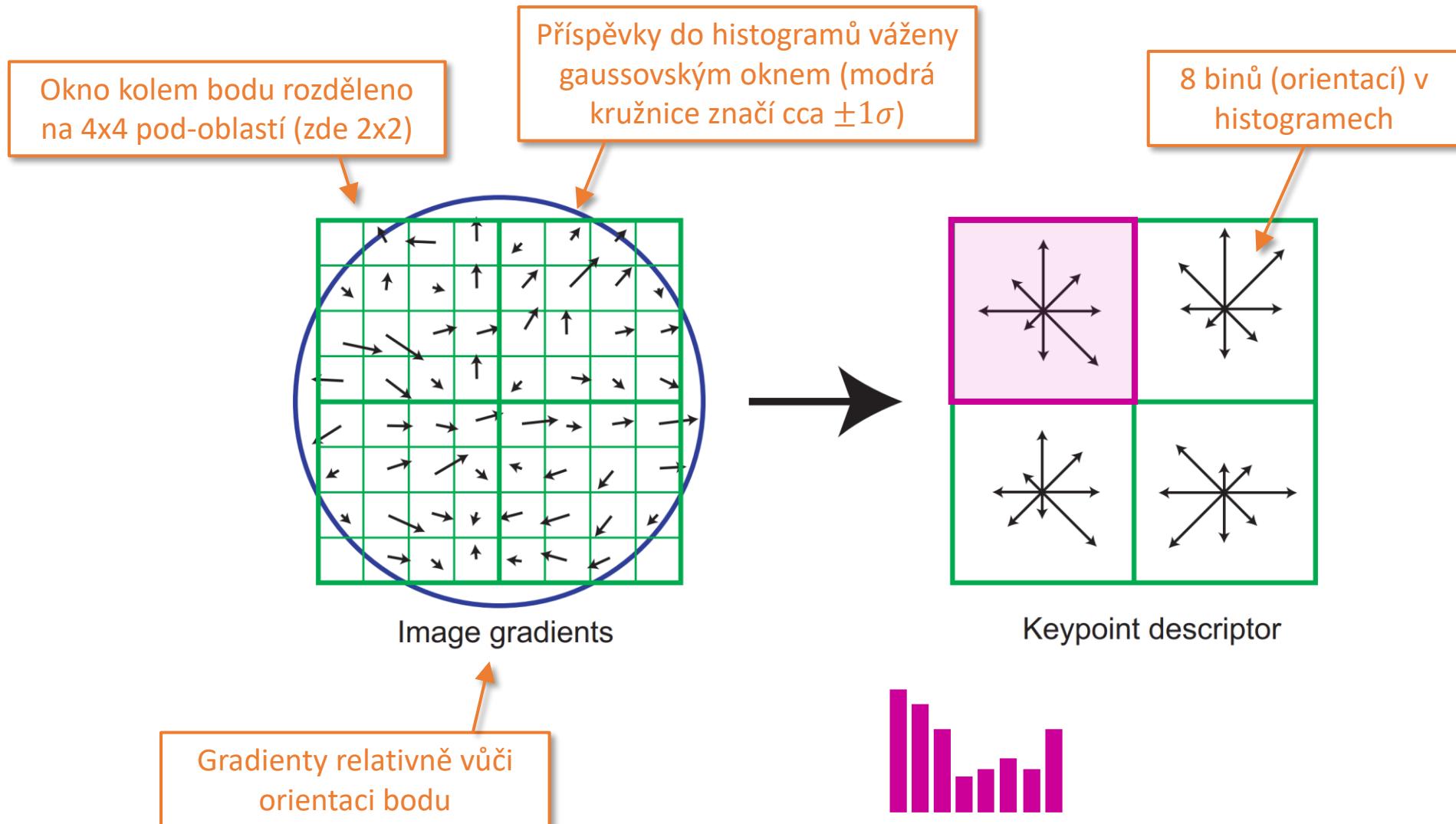
---

# Histogram orientací gradientů



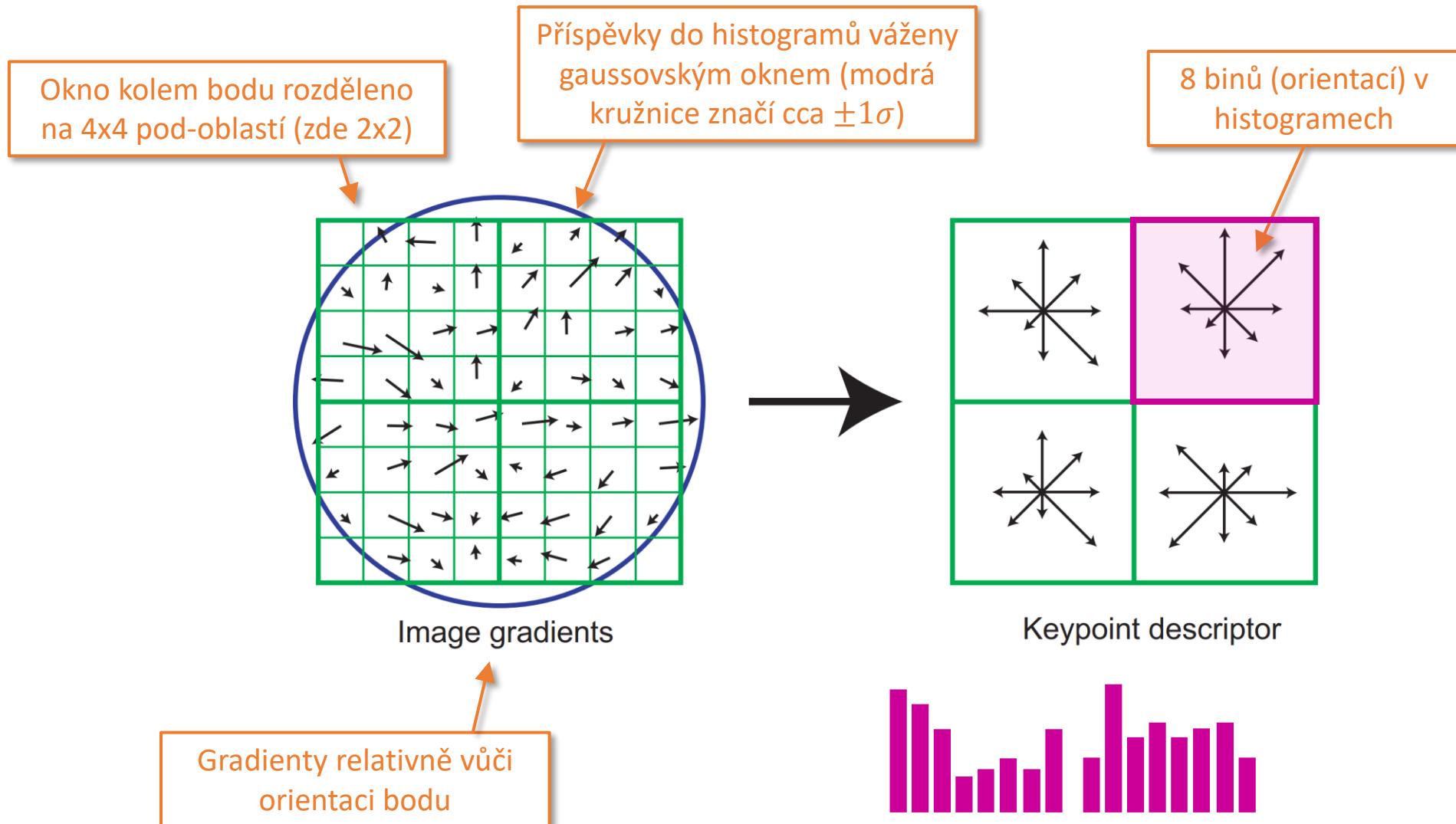
výsledný deskriptor je spojení histogramů pod-oblasti do jednoho vektoru

# Histogram orientací gradientů



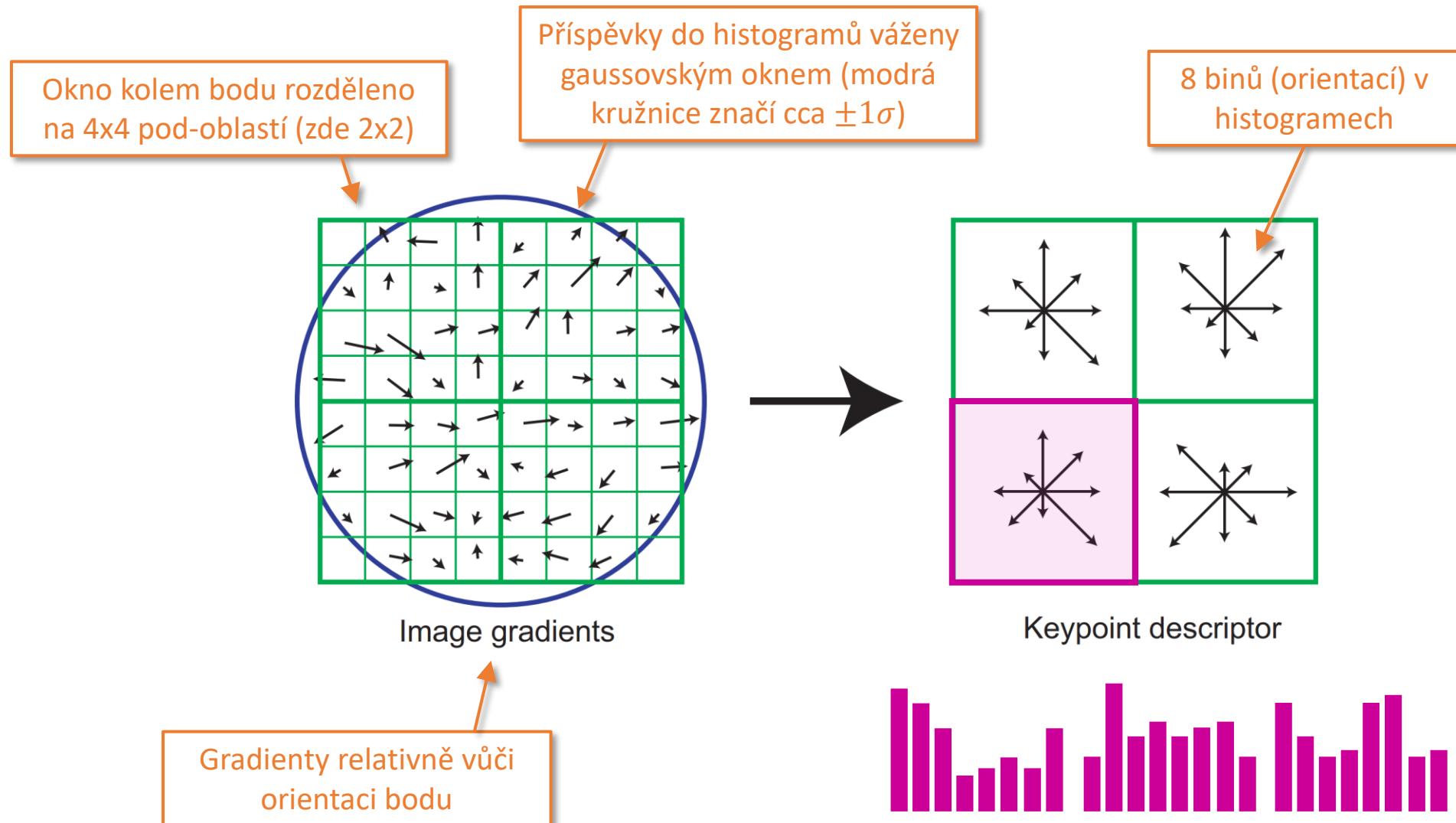
výsledný deskriptor je spojení histogramů pod-oblastí do jednoho vektoru

# Histogram orientací gradientů



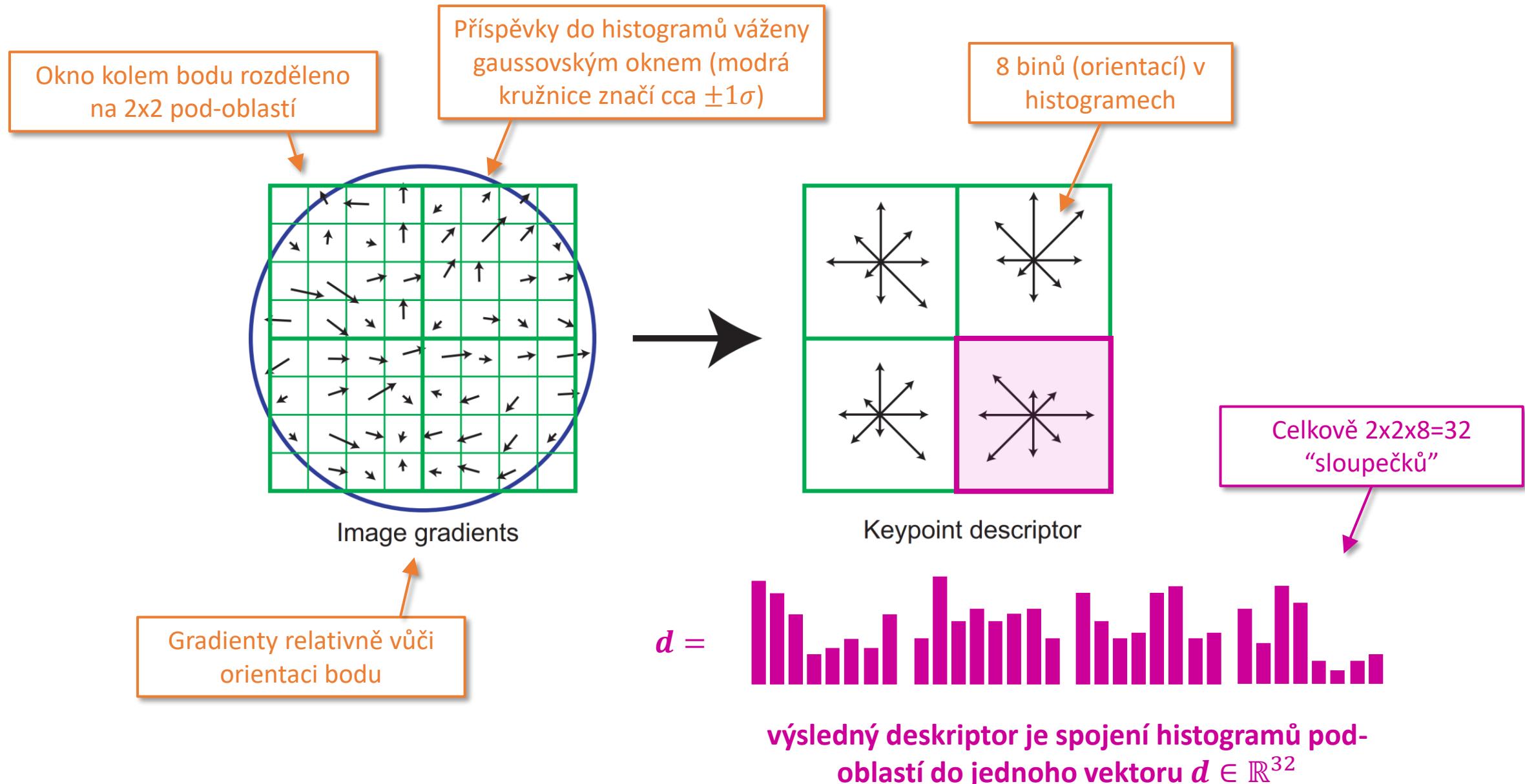
výsledný deskriptor je spojení histogramů pod-oblastí do jednoho vektoru

# Histogram orientací gradientů



výsledný deskriptor je spojení histogramů pod-oblastí do jednoho vektoru

# Histogram orientací gradientů



# Implementační detaily a triky v deskriptoru SIFT

- Ve skutečnosti okno rozděleno na 4x4 pod-obláстí → vektor deskriptoru je dlouhý 4x4x8=128
- Příspěvky do gradientu váženy gaussovským oknem
- Pro snížení citlivosti na změny jasu je deskriptor normalizován na jednotkovou délku (normu)

$$\mathbf{d}_{\text{norm}} = \frac{\mathbf{d}}{\|\mathbf{d}\|}$$

→ deskriptor je invariantní vůči affinní změně osvětlení

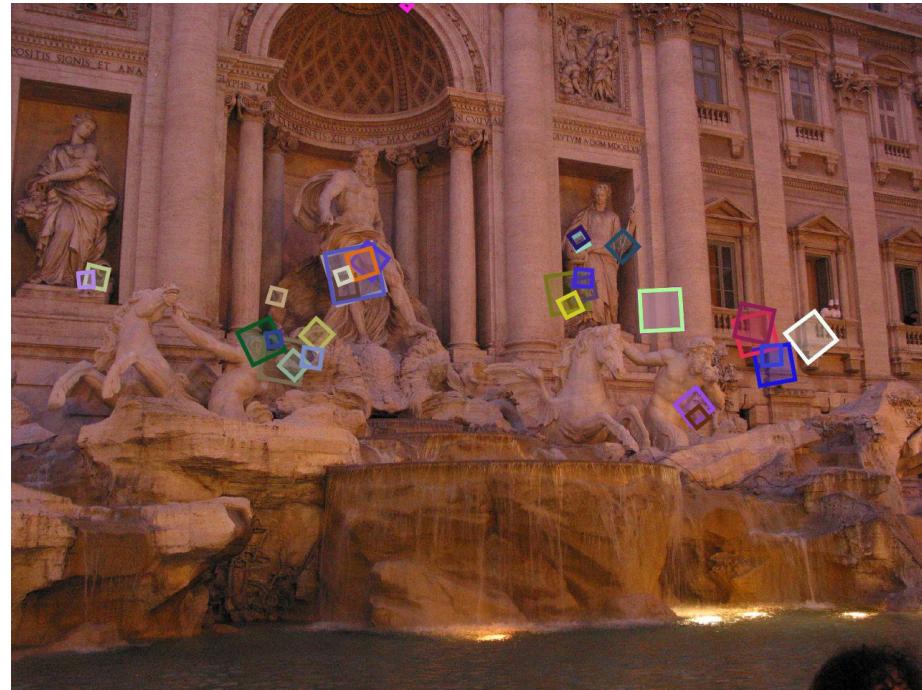
- Limitace vlivu neuniformní změny osvětlení prahováním prvků

$$\mathbf{d}'_{\text{norm}} = \min(\mathbf{d}_{\text{norm}}, 0.2)$$

a poté renormalizováním na jednotkovou normu

$$\mathbf{d}_{\text{renorm}} = \frac{\mathbf{d}'_{\text{norm}}}{\|\mathbf{d}'_{\text{norm}}\|}$$

# Deskriptory SIFT

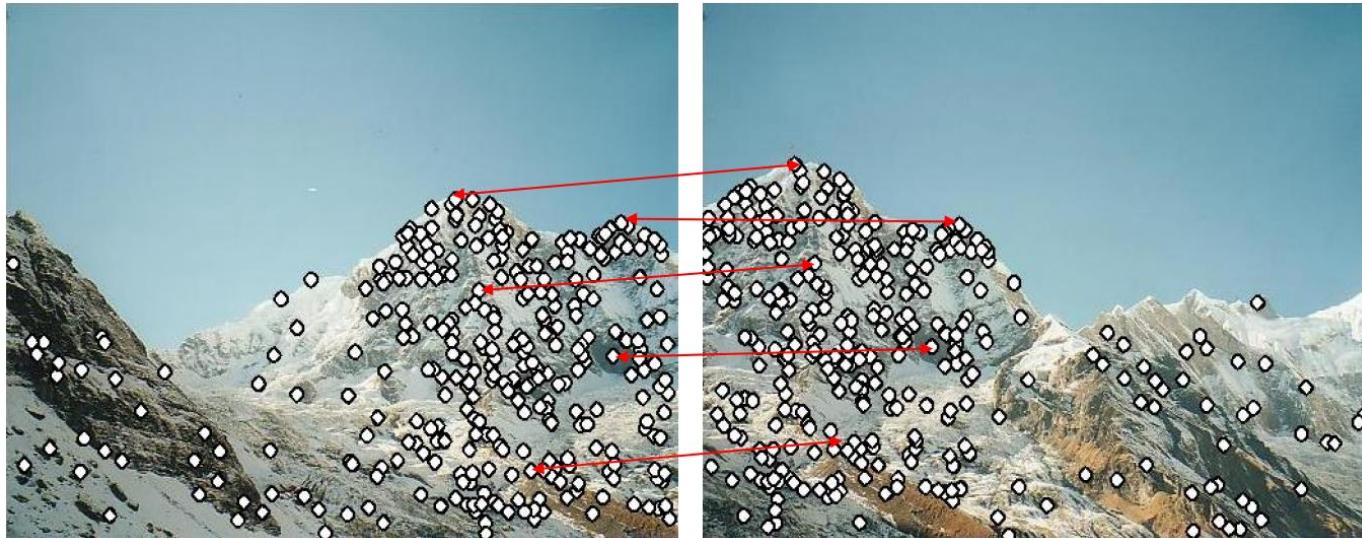


obrázek: <https://www.cs.cornell.edu/courses/cs5670/2023sp/>

## Lokální korespondence mezi obrazy

---

# Lokální korespondence



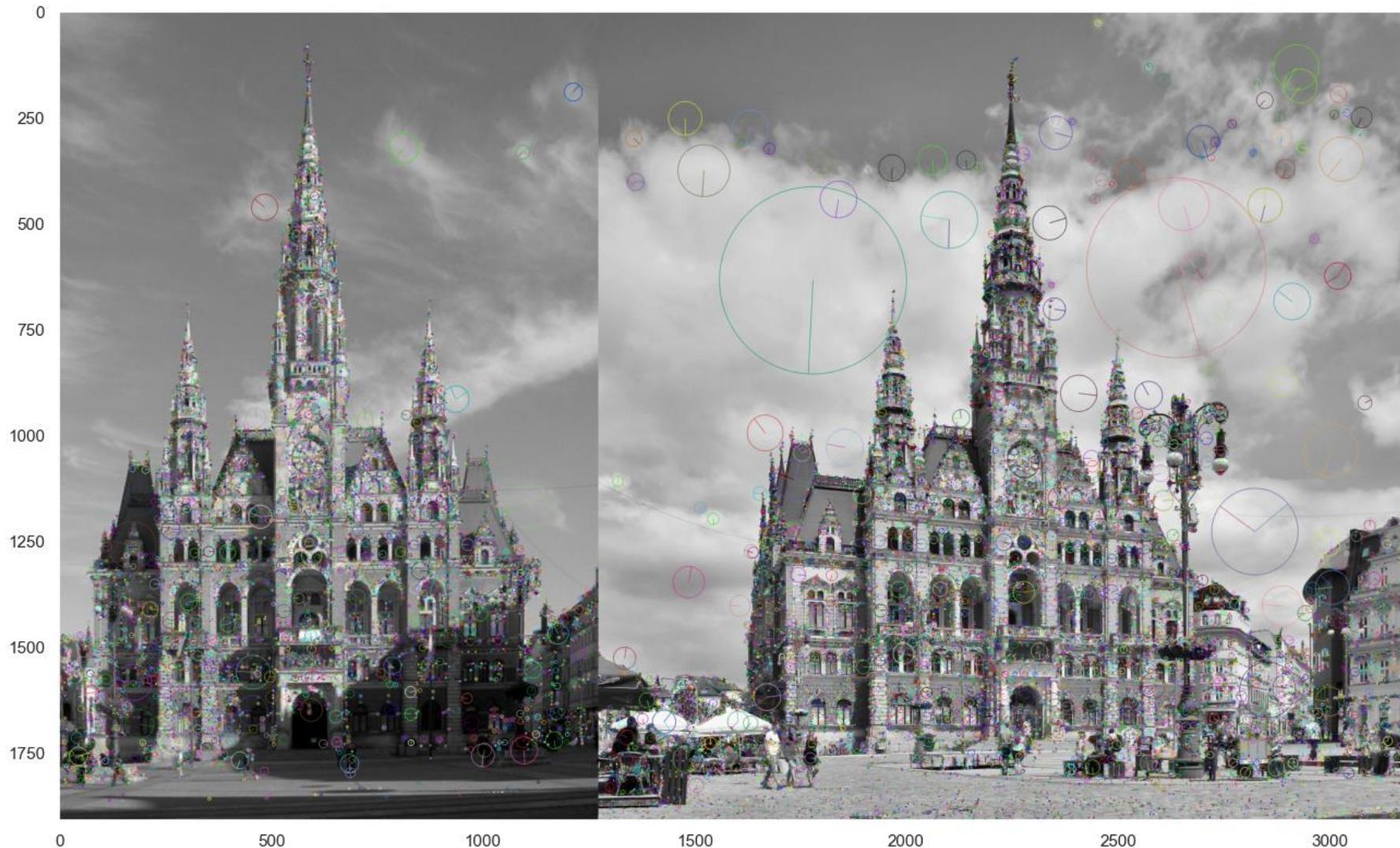
- Abychom mohli např. zarovnat obrazy, potřebujeme vědět, který zájmový bod na obraze 1 odpovídá jakému v obraze 2
- Nejjednodušší postup je hladový algoritmus, kdy ke každému  $x_q$  z "query" obrazu 1 přiřadíme nejpodobnější  $x$  z obrazu 2
- (Ne)podobnost může být např. L2 vzdálenost

$$d(x_q, x) = \|x_q - x\|$$

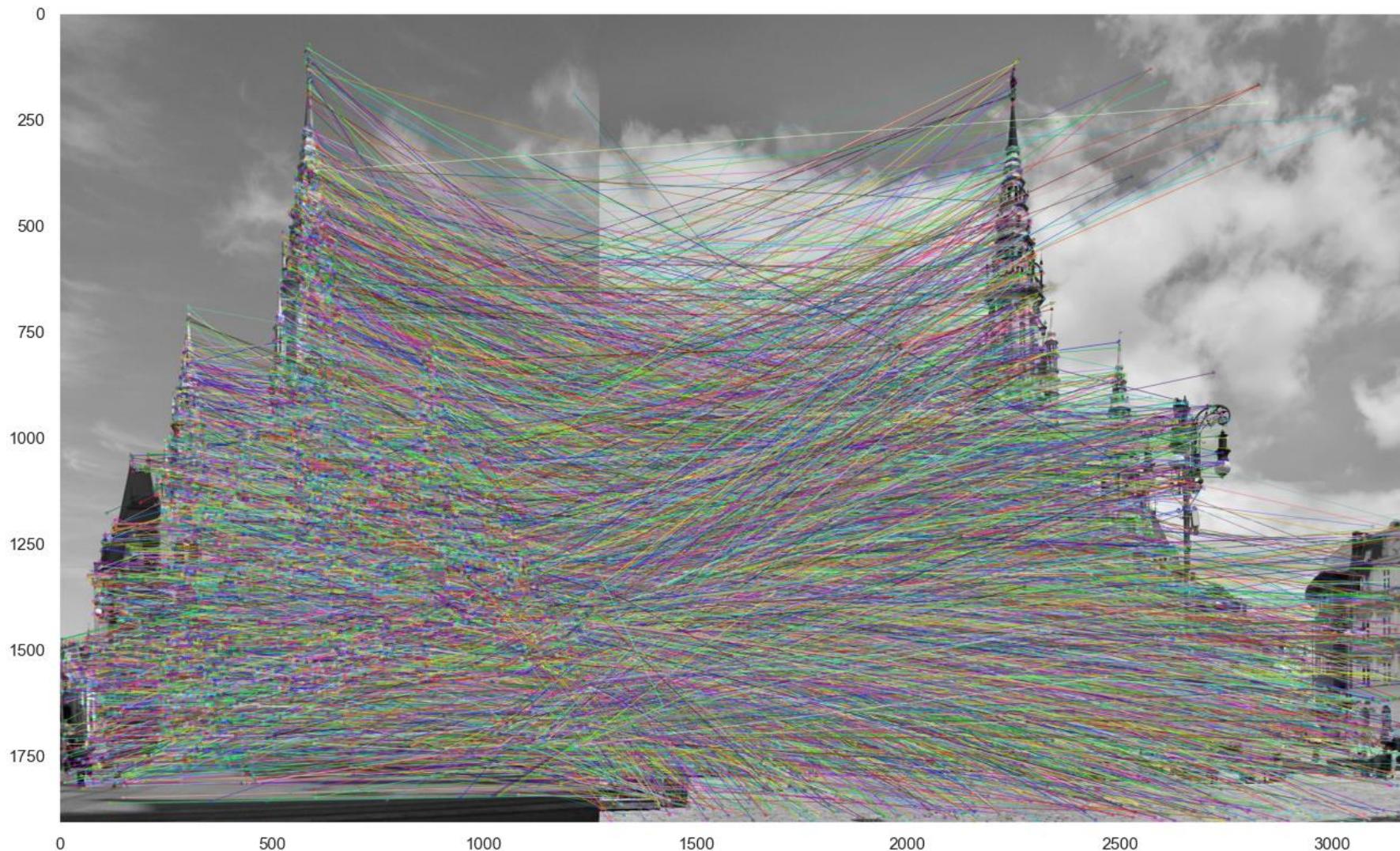
# Lokální korespondence: vstupní obrazy



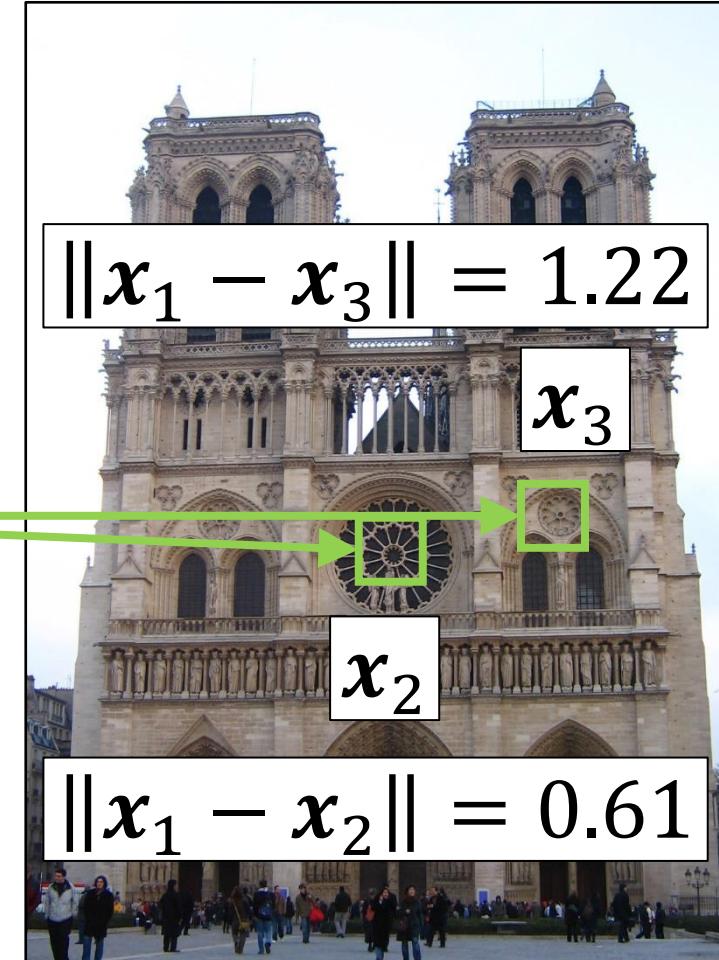
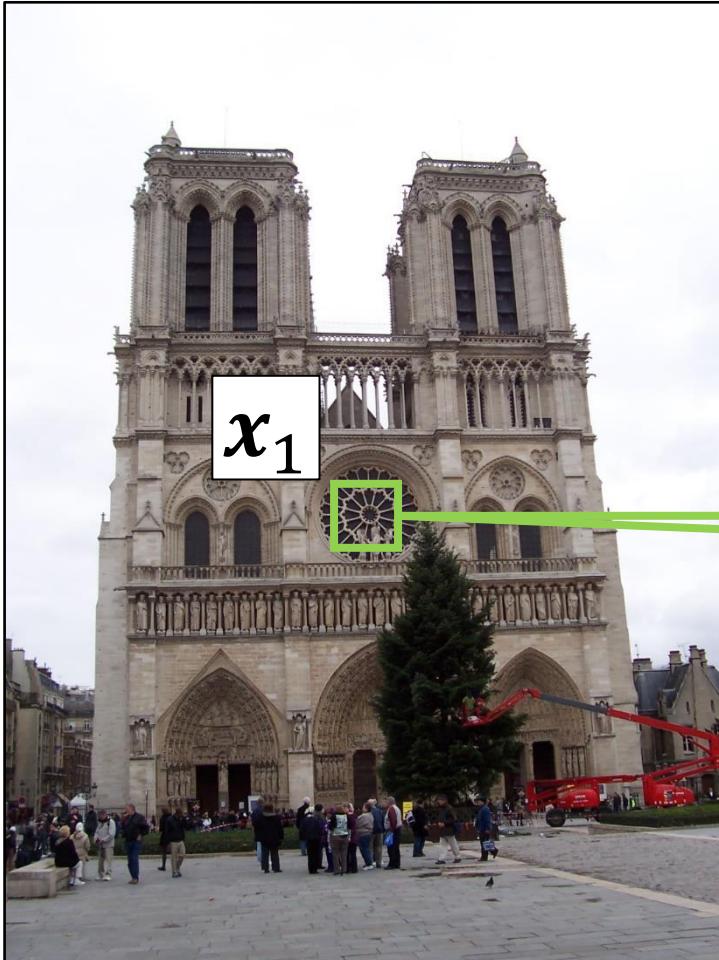
# Lokální korespondence: zájmové body



# Lokální korespondence: přiřazení nejpodobnějších bodů

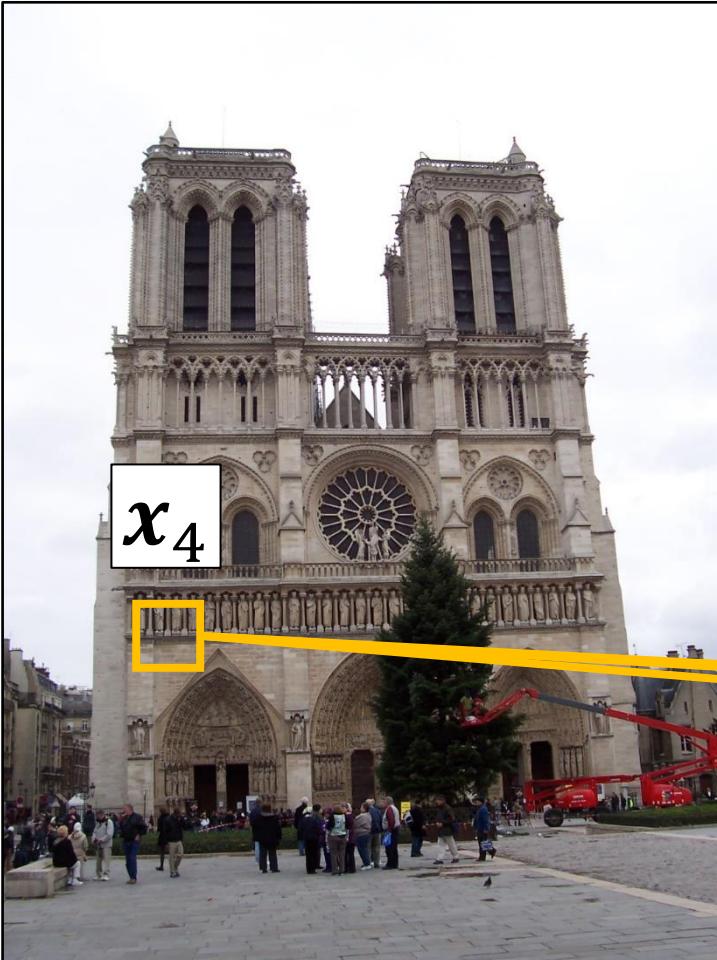


# Jednoznačné přiřazení



$$\|x_1 - x_2\| = 0.61$$

# Nejednoznačné přiřazení



$$\|x_4 - x_5\| = 0.34$$

$$\|x_4 - x_6\| = 0.30$$

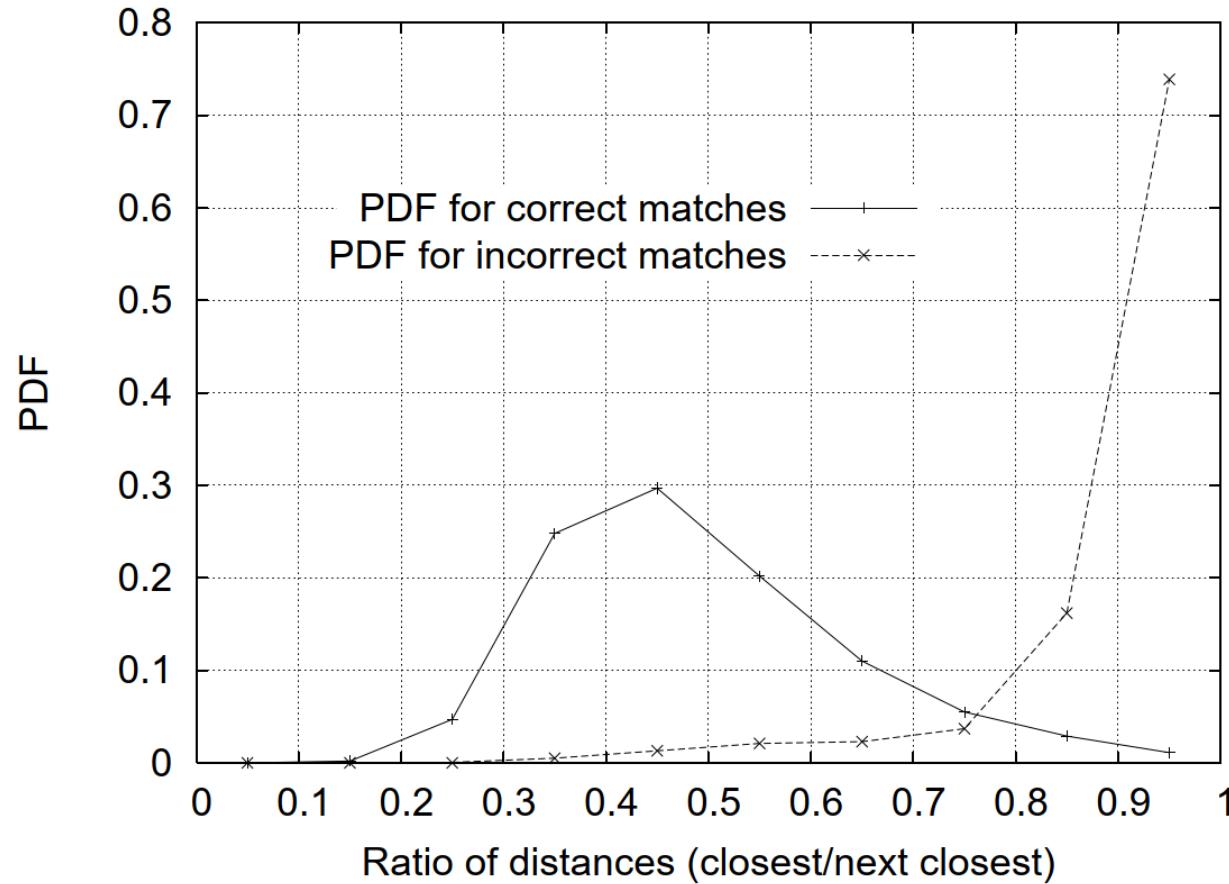
$$\|x_4 - x_7\| = 0.40$$

# Poměr vzdáleností dvou nejbližších sousedů

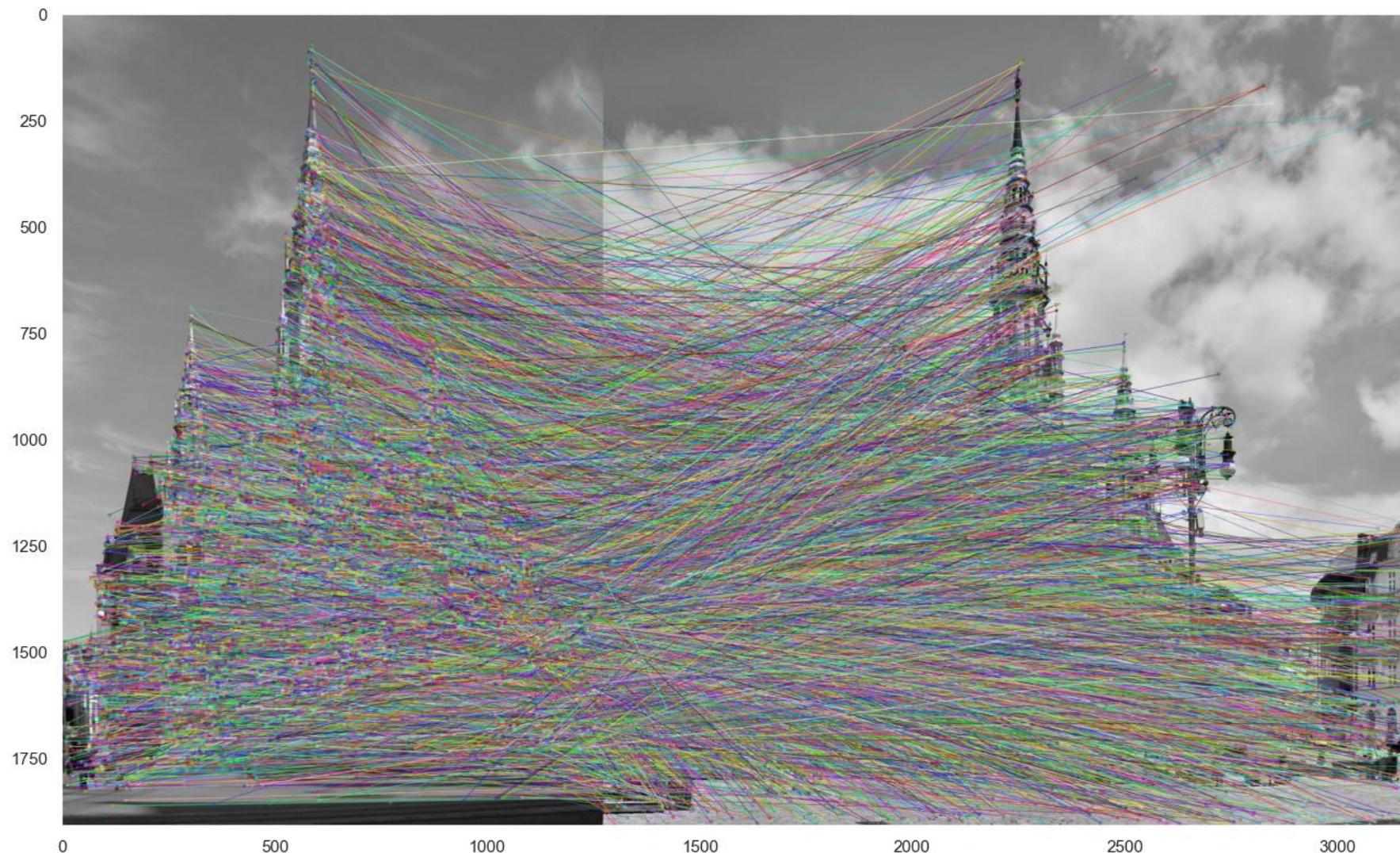
- Je obtížné stanovit práh na vzdálenost mezi  $\mathbf{x}_q$  a nejbližším deskriptorem z druhého obrazu  $\mathbf{x}_{1NN}$
- Namísto toho zvážíme ještě druhý nejpodobnější deskriptor  $\mathbf{x}_{2NN}$
- Dobrý test, zda  $\mathbf{x}_q$  a  $\mathbf{x}_{1NN}$  reprezentují stejný bod, je poměr vzdáleností

$$r = \frac{\|\mathbf{x}_q - \mathbf{x}_{1NN}\|}{\|\mathbf{x}_q - \mathbf{x}_{2NN}\|}$$

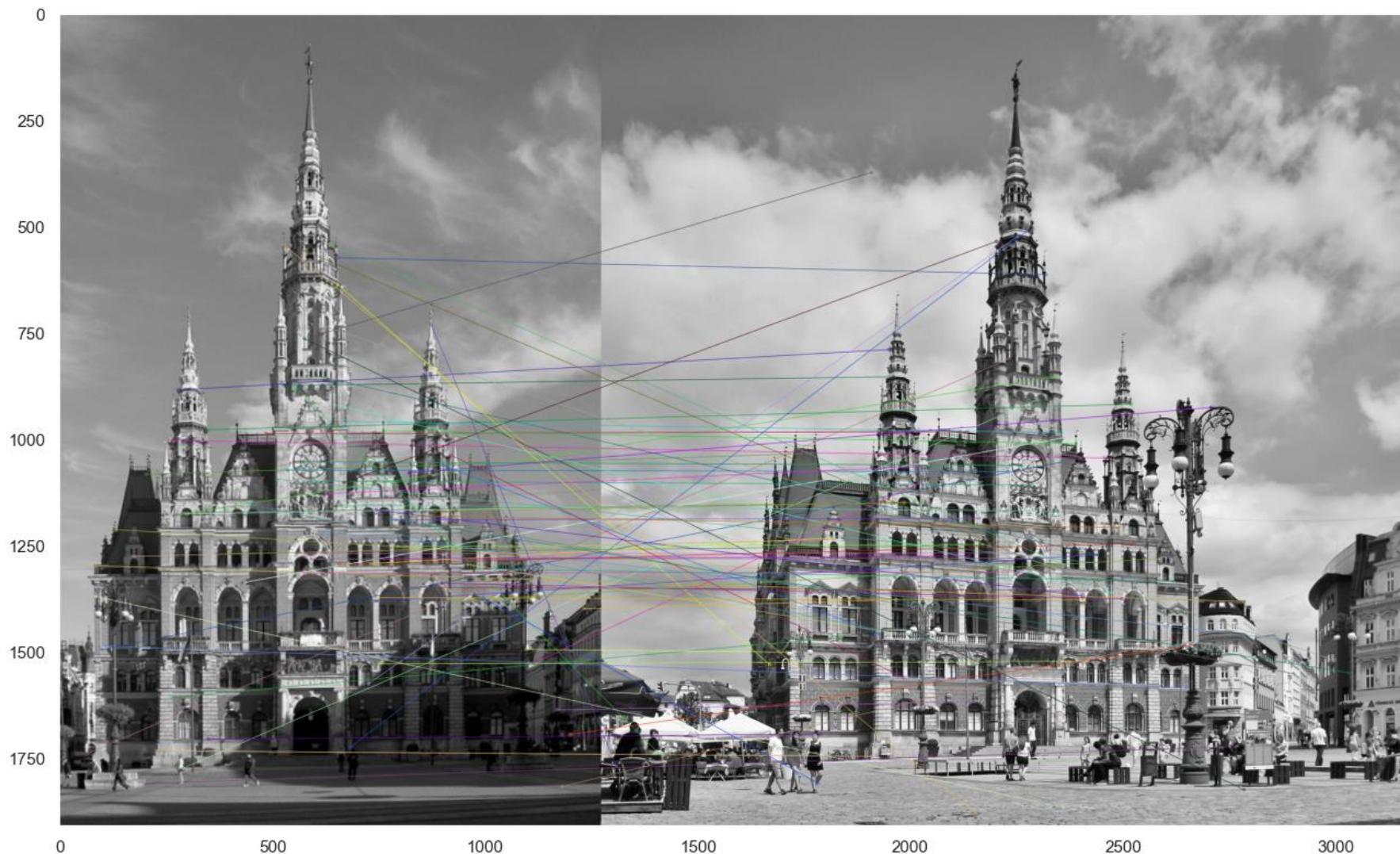
# Poměr vzdáleností dvou nejbližších sousedů



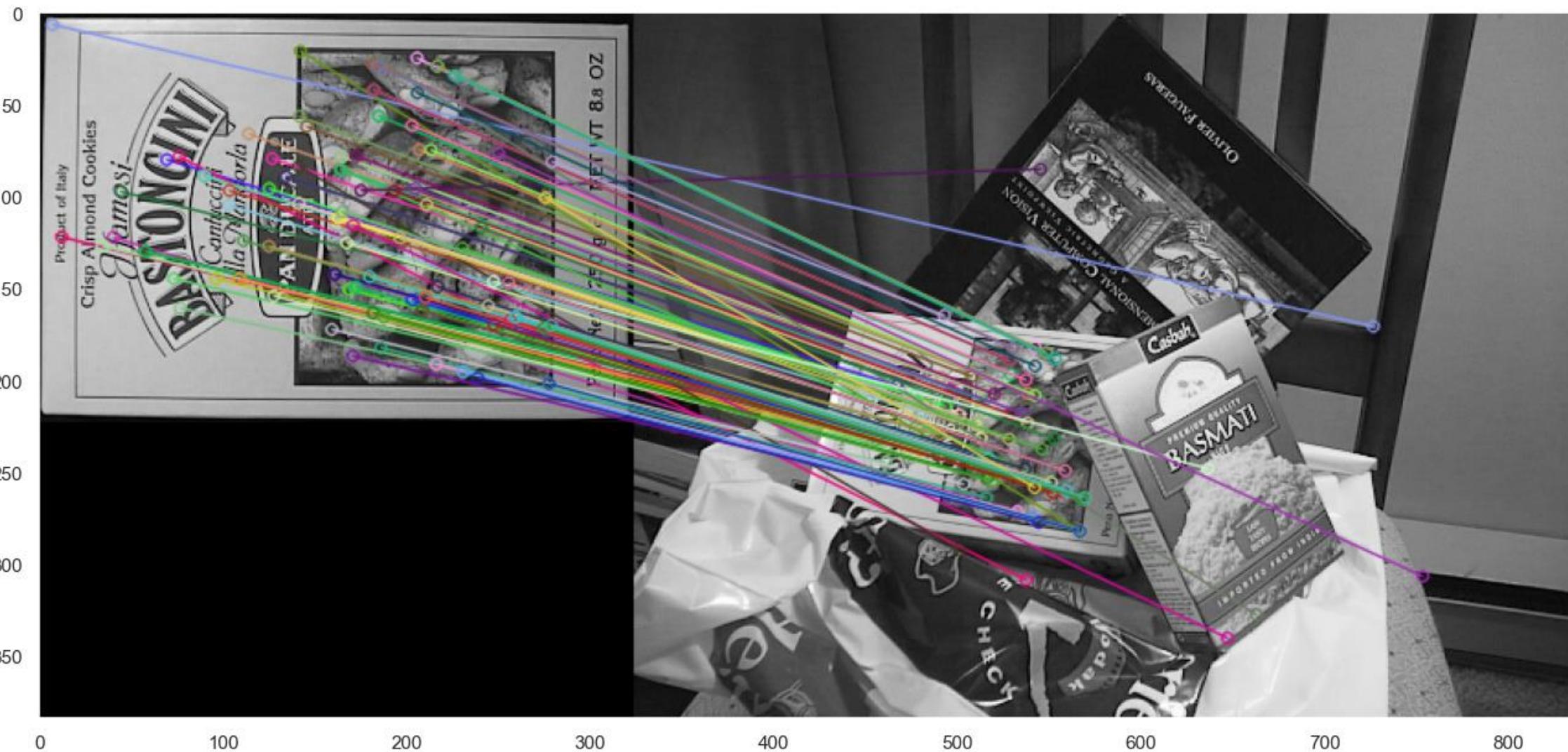
# Poměr vzdáleností dvou nejbližších sousedů: před



# Poměr vzdáleností dvou nejbližších sousedů: po



# Poměr vzdáleností dvou nejbližších sousedů: po



## Odhad geometrické transformace mezi obrazy

---

# Homogenní souřadnice

- Pozici pixelu s kartézskými souřadnicemi  $(x, y)$  můžeme vyjádřit jako vektor

$$\bar{x} = \begin{bmatrix} \bar{x} \\ \bar{y} \\ \bar{z} \end{bmatrix} \in \mathbb{R}^3$$

tzv. homogenních souřadnic, přičemž body lišící se pouze přanásobením považujeme za ekvivalentní, tj.  $\bar{x} \equiv c \cdot \bar{x}$

- Z homogenních souřadnic dostaneme zpět kartézské souřadnice vydělením  $\bar{z}$ , tj.

$$x = \bar{x}/\bar{z}$$
$$y = \bar{y}/\bar{z}$$

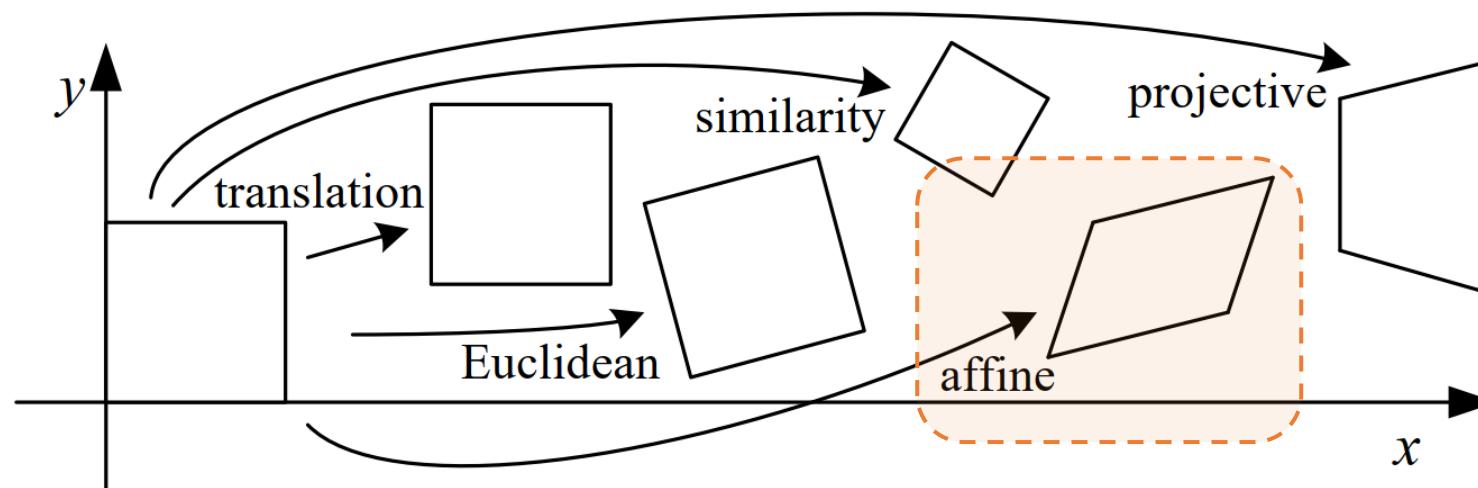
# Afinní transformace

- Afinní transformace nad homogenními souřadnicemi má podobu

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

- Maticově

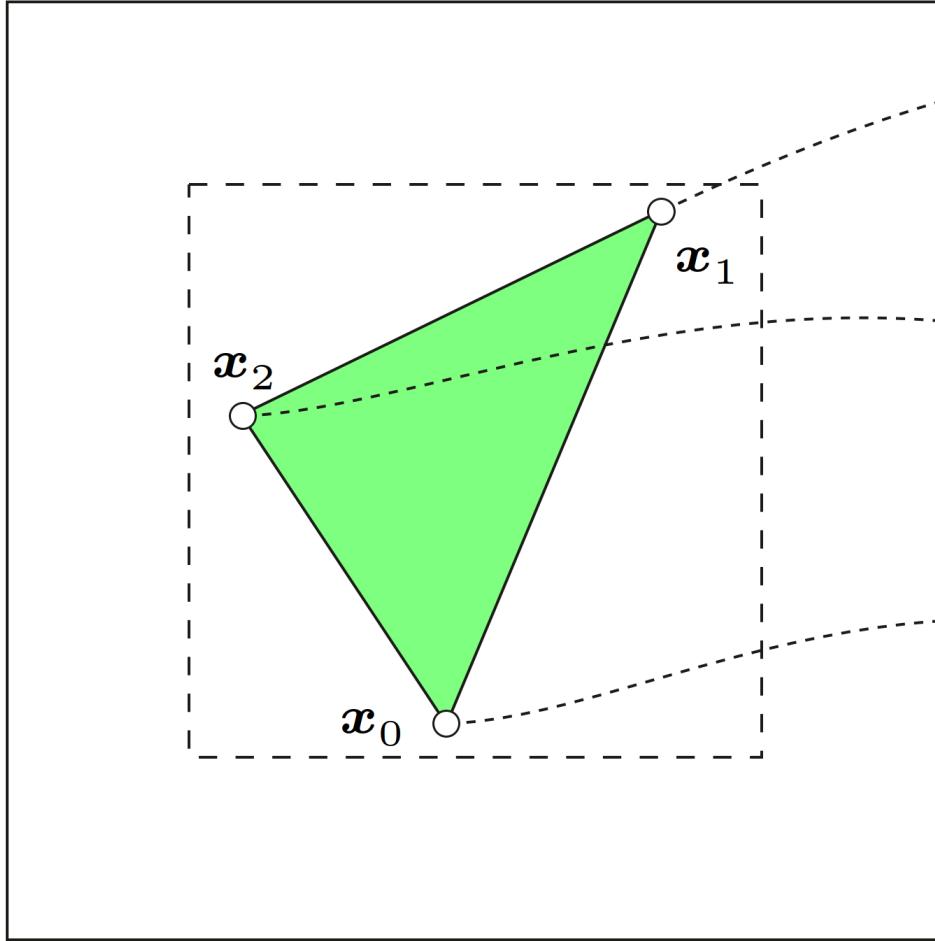
$$\bar{x}' = A \cdot \bar{x}$$



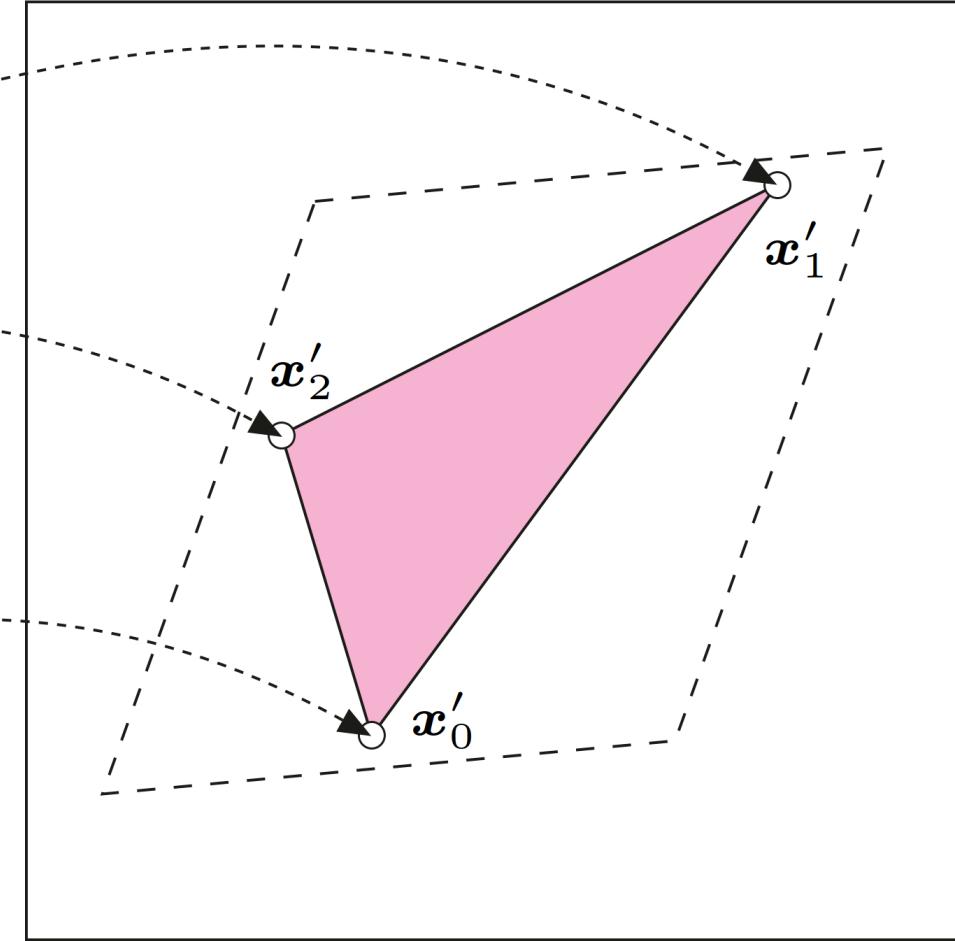
obrázek: <https://szeliski.org/Book/>

# Afinní transformace

$I$



$I'$



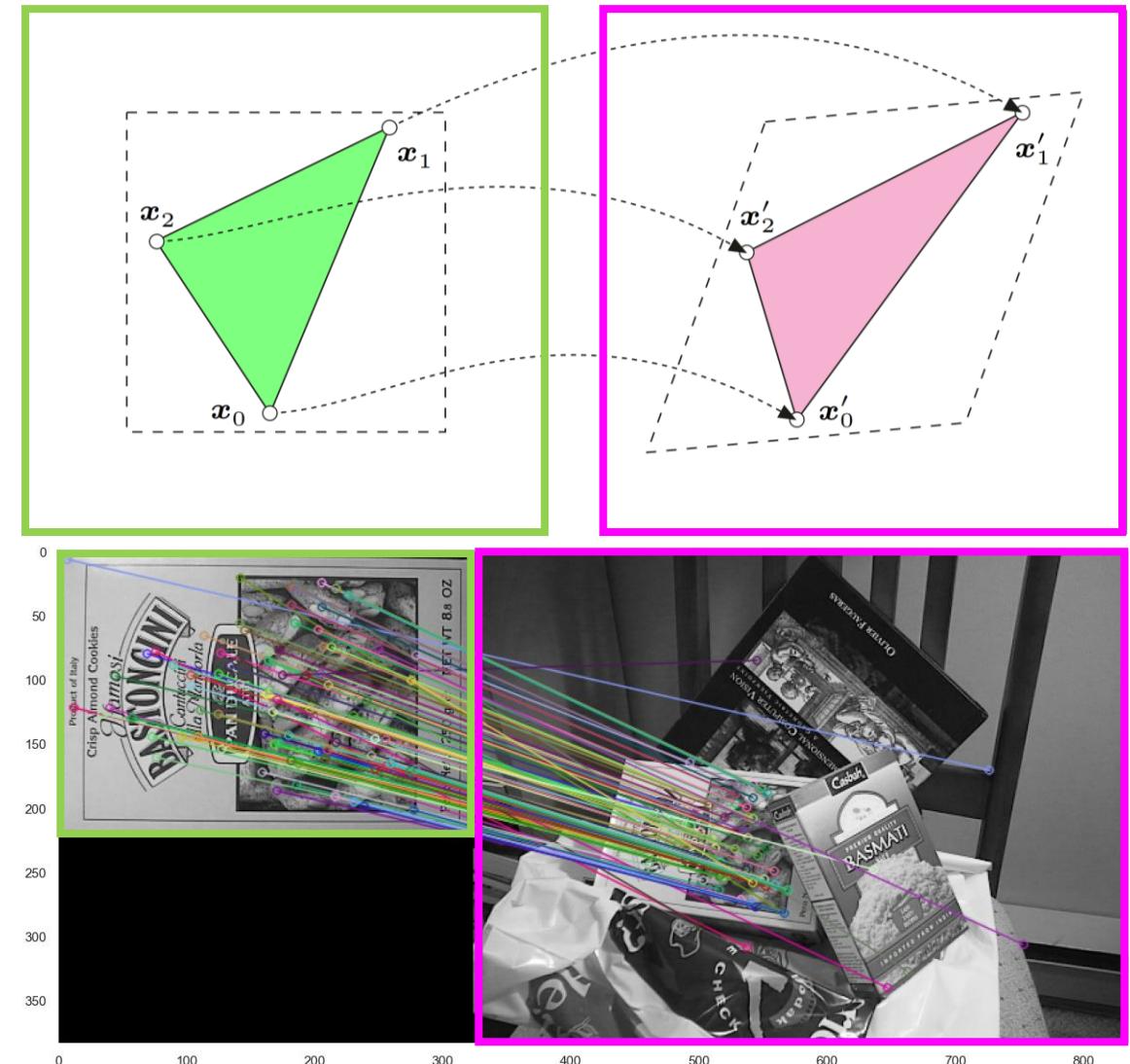
# Odhad affinní transformace z lokálních korespondencí

- Transformace je

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

- Můžeme přepsat jako

$$\begin{bmatrix} x & y & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x & y & 1 \end{bmatrix} \cdot \begin{bmatrix} a_{1,1} \\ a_{1,2} \\ a_{1,3} \\ a_{2,1} \\ a_{2,2} \\ a_{2,3} \end{bmatrix} = \begin{bmatrix} x' \\ y' \end{bmatrix}$$



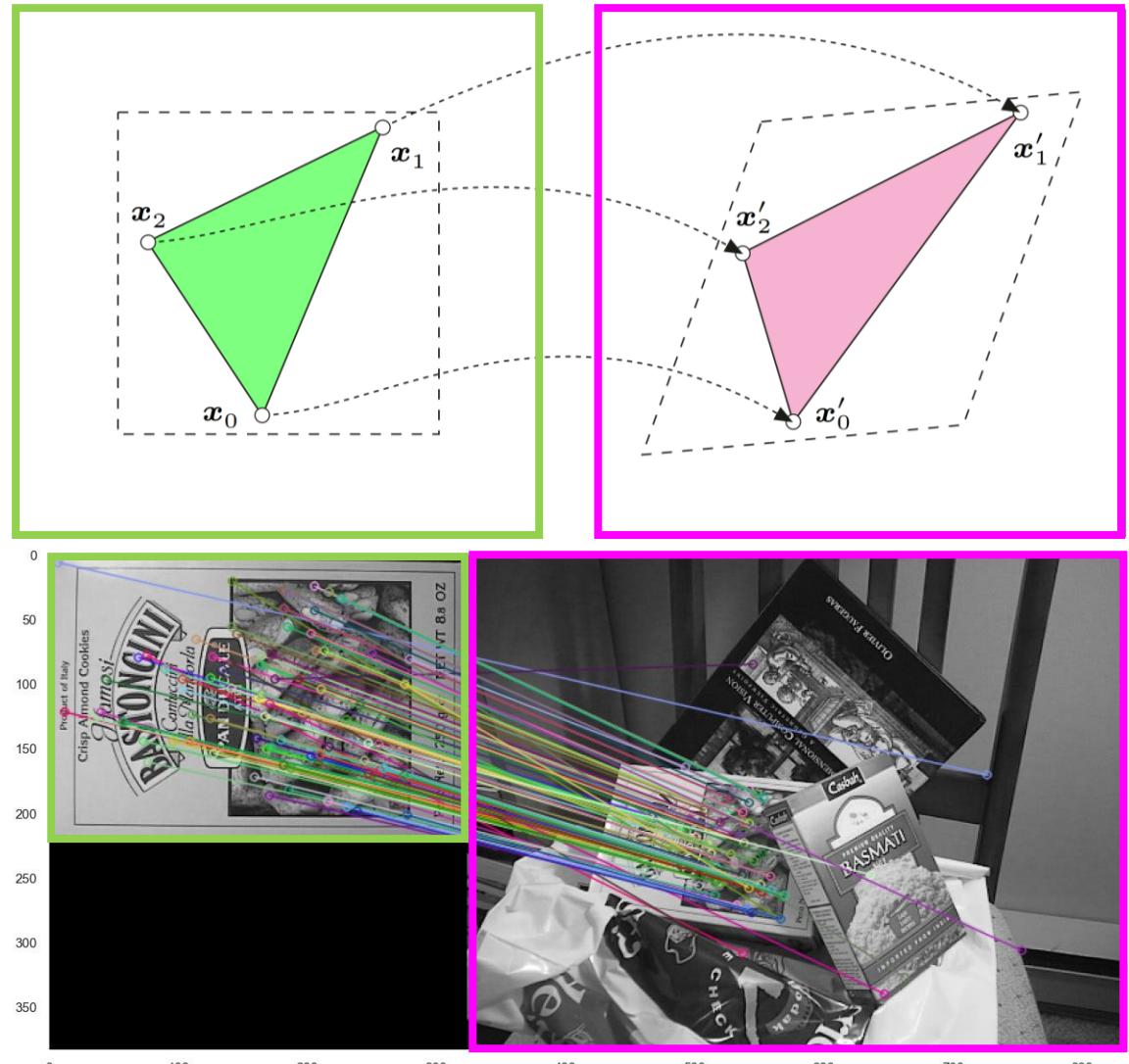
# Odhad affinní transformace z lokálních korespondencí

- Transformace je

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

- Pro trojici bodů zároveň

$$\begin{bmatrix} x_0 & y_0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_0 & y_0 & 1 \\ x_1 & y_1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_1 & y_1 & 1 \\ x_2 & y_2 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_2 & y_2 & 1 \end{bmatrix} \cdot \begin{bmatrix} a_{1,1} \\ a_{1,2} \\ a_{1,3} \\ a_{2,1} \\ a_{2,2} \\ a_{2,3} \end{bmatrix} = \begin{bmatrix} x'_0 \\ y'_0 \\ x'_1 \\ y'_1 \\ x'_2 \\ y'_2 \end{bmatrix}$$



# Odhad affinní transformace z lokálních korespondencí

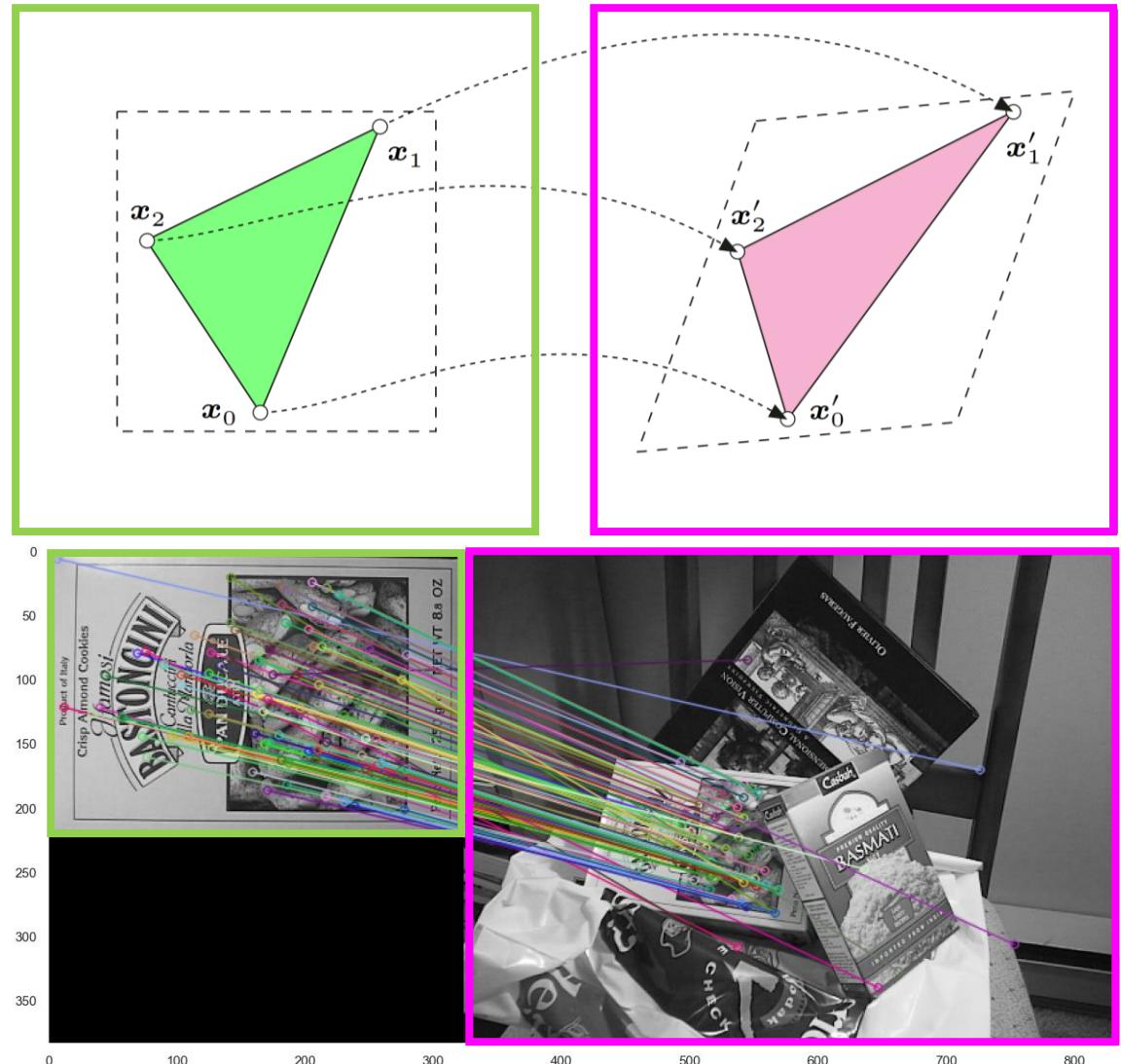
- Transformace je

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

- Pro trojici bodů zároveň

$$\begin{bmatrix} x_0 & y_0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_0 & y_0 & 1 \\ x_1 & y_1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_1 & y_1 & 1 \\ x_2 & y_2 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_2 & y_2 & 1 \end{bmatrix} \cdot \begin{bmatrix} a_{1,1} \\ a_{1,2} \\ a_{1,3} \\ a_{2,1} \\ a_{2,2} \\ a_{2,3} \end{bmatrix} = \begin{bmatrix} x'_0 \\ y'_0 \\ x'_1 \\ y'_1 \\ x'_2 \\ y'_2 \end{bmatrix}$$

- Parametry = řešení soustavy  $X \cdot \mathbf{a} = \bar{x}$



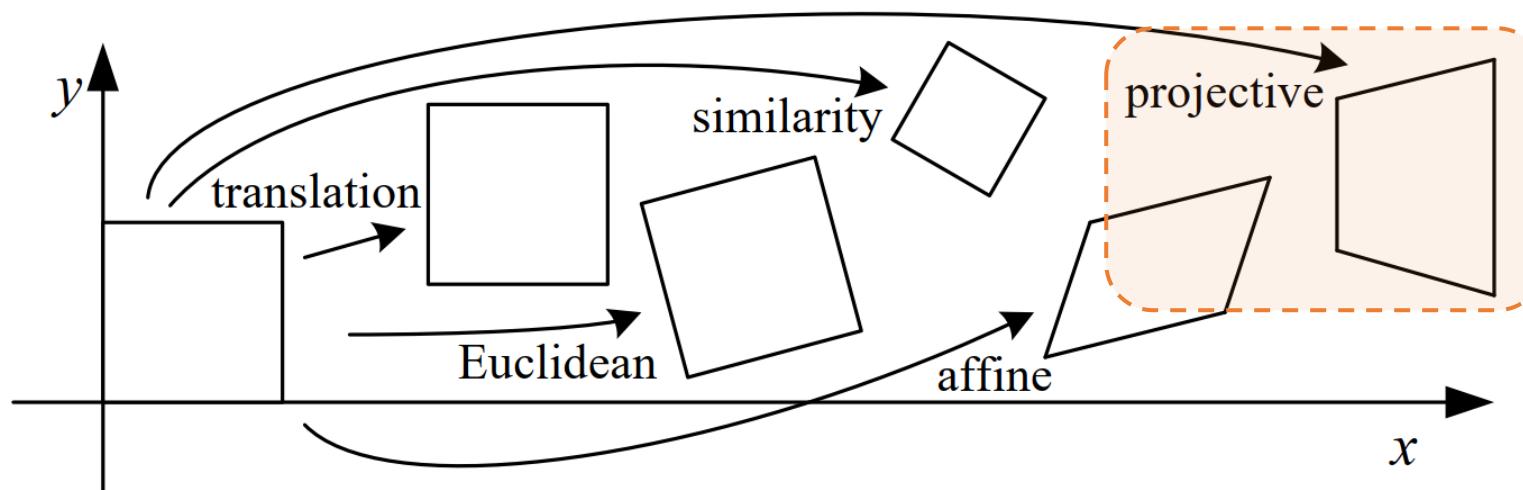
# Projektivní transformace

- Projektivní transformace (homografie) má podobu

$$\begin{bmatrix} \bar{x}' \\ \bar{y}' \\ \bar{z}' \end{bmatrix} = \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ \textcolor{orange}{a_{3,1}} & \textcolor{orange}{a_{3,2}} & \textcolor{orange}{a_{3,3}} \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

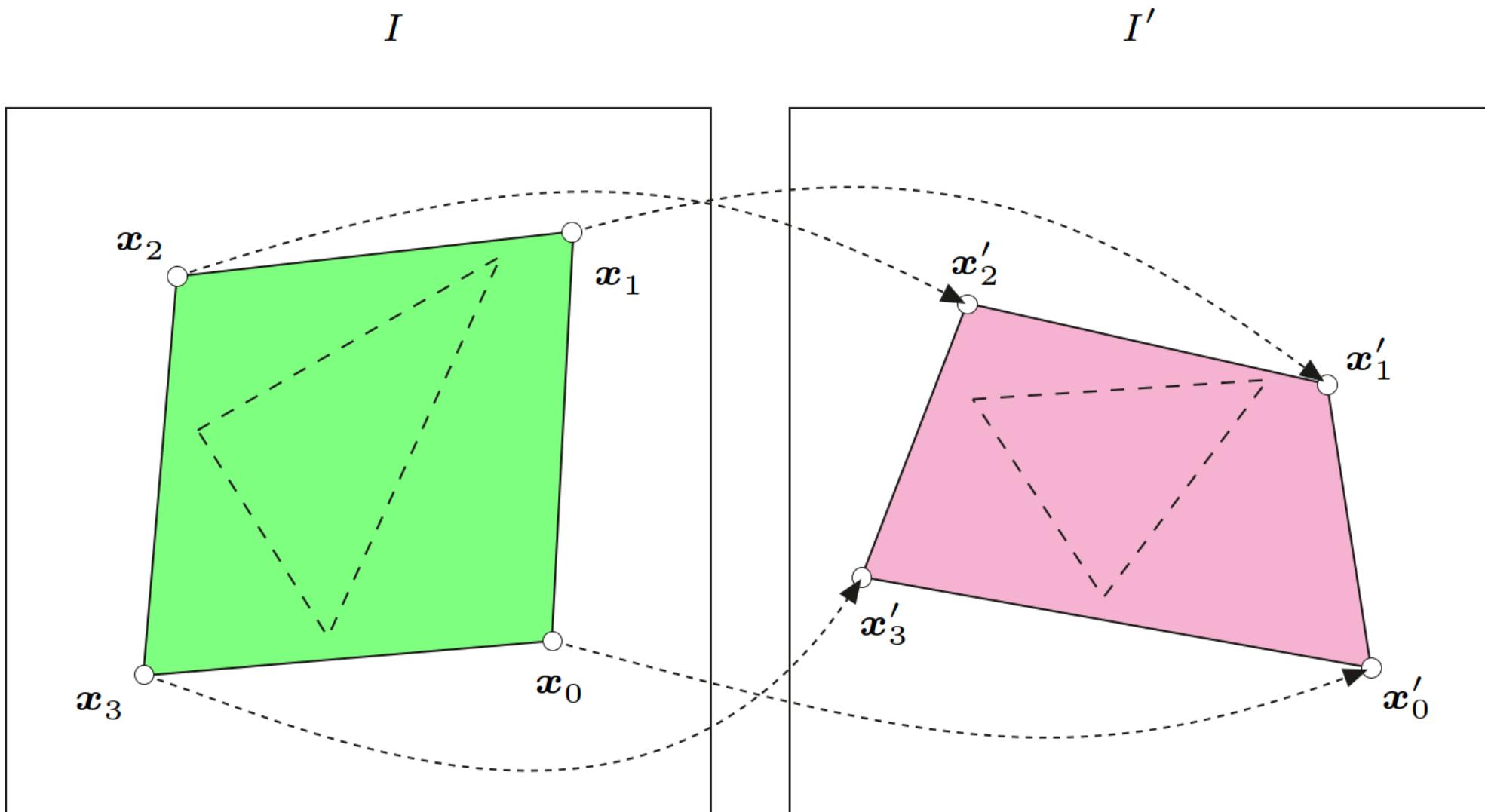
a poté

$$x' = \bar{x}' / \bar{z} \quad y' = \bar{y}' / \bar{z}$$



obrázek: <https://szeliski.org/Book/>

# Projektivní transformace



# Odhad projektivní transformace z lokálních korespondencí

- Transformace je

$$\begin{bmatrix} \bar{x}' \\ \bar{y}' \\ \bar{z}' \end{bmatrix} = \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,1} & a_{3,2} & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

díky dělení při převodu zpět do kartézských souřadnic nezáleží na měřítku

- Můžeme po dělení  $\bar{z}'$  přepsat jako

$$\begin{bmatrix} x_0 & y_0 & 1 & 0 & 0 & 0 & -x_0 \cdot x'_0 & -y_0 \cdot x'_0 \\ 0 & 0 & 0 & x_0 & y_0 & 1 & -x_0 \cdot y'_0 & -y_0 \cdot y'_0 \\ x_1 & y_1 & 1 & 0 & 0 & 0 & -x_1 \cdot x'_1 & -y_1 \cdot x'_1 \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -x_1 \cdot y'_1 & -y_1 \cdot y'_1 \\ x_2 & y_2 & 1 & 0 & 0 & 0 & -x_2 \cdot x'_2 & -y_2 \cdot x'_2 \\ 0 & 0 & 0 & x_2 & y_2 & 1 & -x_2 \cdot y'_2 & -y_2 \cdot y'_2 \\ x_3 & y_3 & 1 & 0 & 0 & 0 & -x_3 \cdot x'_3 & -y_3 \cdot x'_3 \\ 0 & 0 & 0 & x_3 & y_3 & 1 & -x_3 \cdot y'_3 & -y_3 \cdot y'_3 \end{bmatrix} \cdot \begin{bmatrix} a_{1,1} \\ a_{1,2} \\ a_{1,3} \\ a_{2,1} \\ a_{2,2} \\ a_{2,3} \\ a_{3,1} \\ a_{3,2} \end{bmatrix} = \begin{bmatrix} x'_0 \\ y'_0 \\ x'_1 \\ y'_1 \\ x'_2 \\ y'_2 \\ x'_3 \\ y'_3 \end{bmatrix}$$

- A vyřešit soustavu  $X \cdot \mathbf{a} = \bar{x}$

# Odhad pomocí nejmenších čtverců

- Obvykle máme více korespondencí, než potřebujeme

$$2N \left\{ \begin{bmatrix} x_0 & y_0 & 1 & 0 & 0 & 0 & -x_0 \cdot x'_0 & -y_0 \cdot x'_0 \\ 0 & 0 & 0 & x_0 & y_0 & 1 & -x_0 \cdot y'_0 & -y_0 \cdot y'_0 \\ \vdots & \vdots \\ x_N & y_N & 1 & 0 & 0 & 0 & -x_N \cdot x'_N & -y_N \cdot x'_N \\ 0 & 0 & 0 & x_N & y_N & 1 & -x_N \cdot y'_N & -y_N \cdot y'_N \end{bmatrix} \cdot \begin{bmatrix} a_{1,1} \\ a_{1,2} \\ a_{1,3} \\ a_{2,1} \\ a_{2,2} \\ a_{2,3} \\ a_{3,1} \\ a_{3,2} \end{bmatrix} = \begin{bmatrix} x'_0 \\ y'_0 \\ \vdots \\ x'_N \\ y'_N \end{bmatrix} \right\} 2N$$

- V soustavě

$$X \cdot \boldsymbol{a} = \bar{x}$$

je potom více rovnic než neznámých proměnných

- Soustavu lze řešit ve smyslu nejmenších čtverců

$$\boldsymbol{a}^* = \underset{\boldsymbol{a}}{\operatorname{argmin}} \|X \cdot \boldsymbol{a} - \bar{x}\|^2$$

např. pseudo-inverzí jako

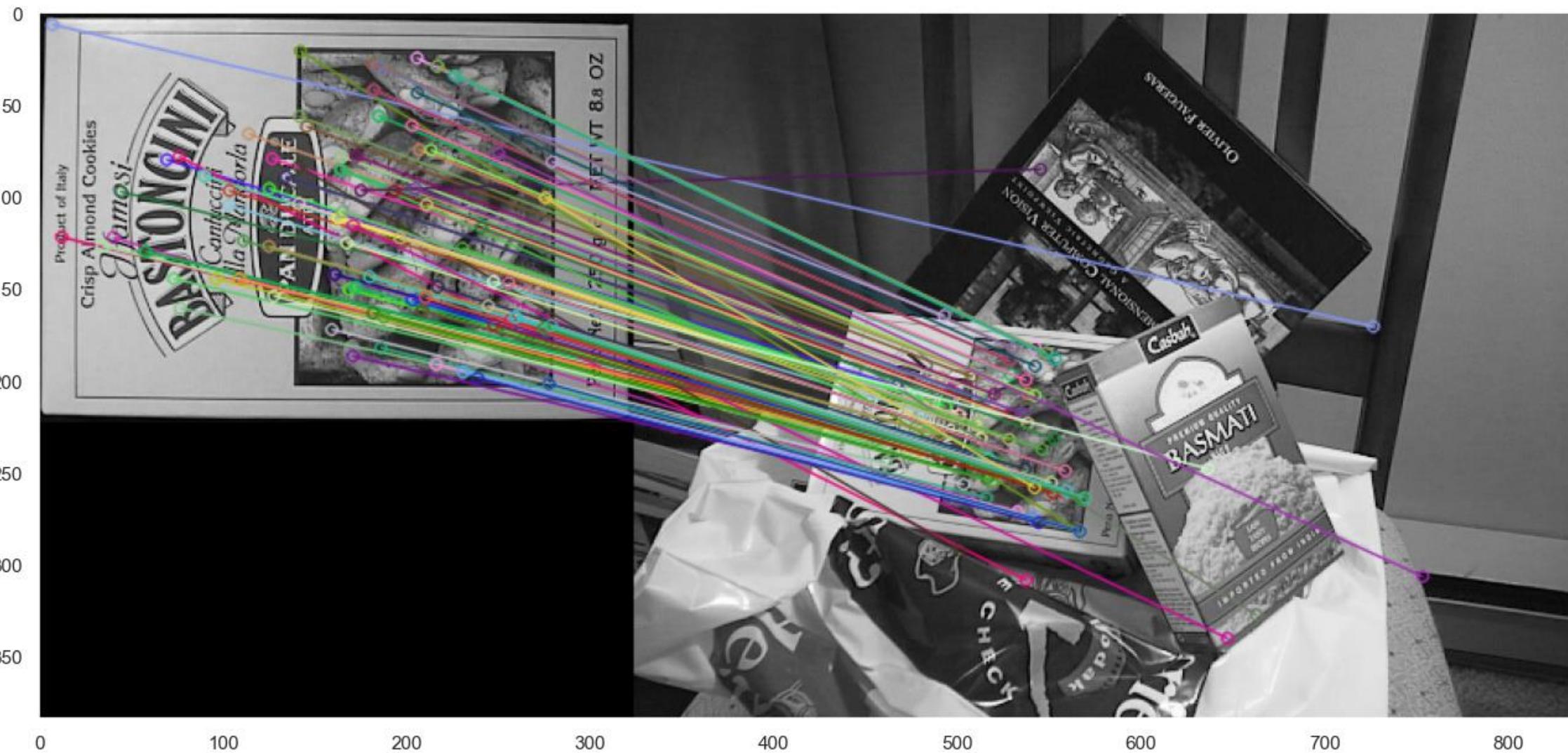
$$\boldsymbol{a}^* = (X^\top \cdot X) \cdot X^\top \cdot \bar{x}$$

# Odhad transformace za přítomnosti chyb v korespondencích

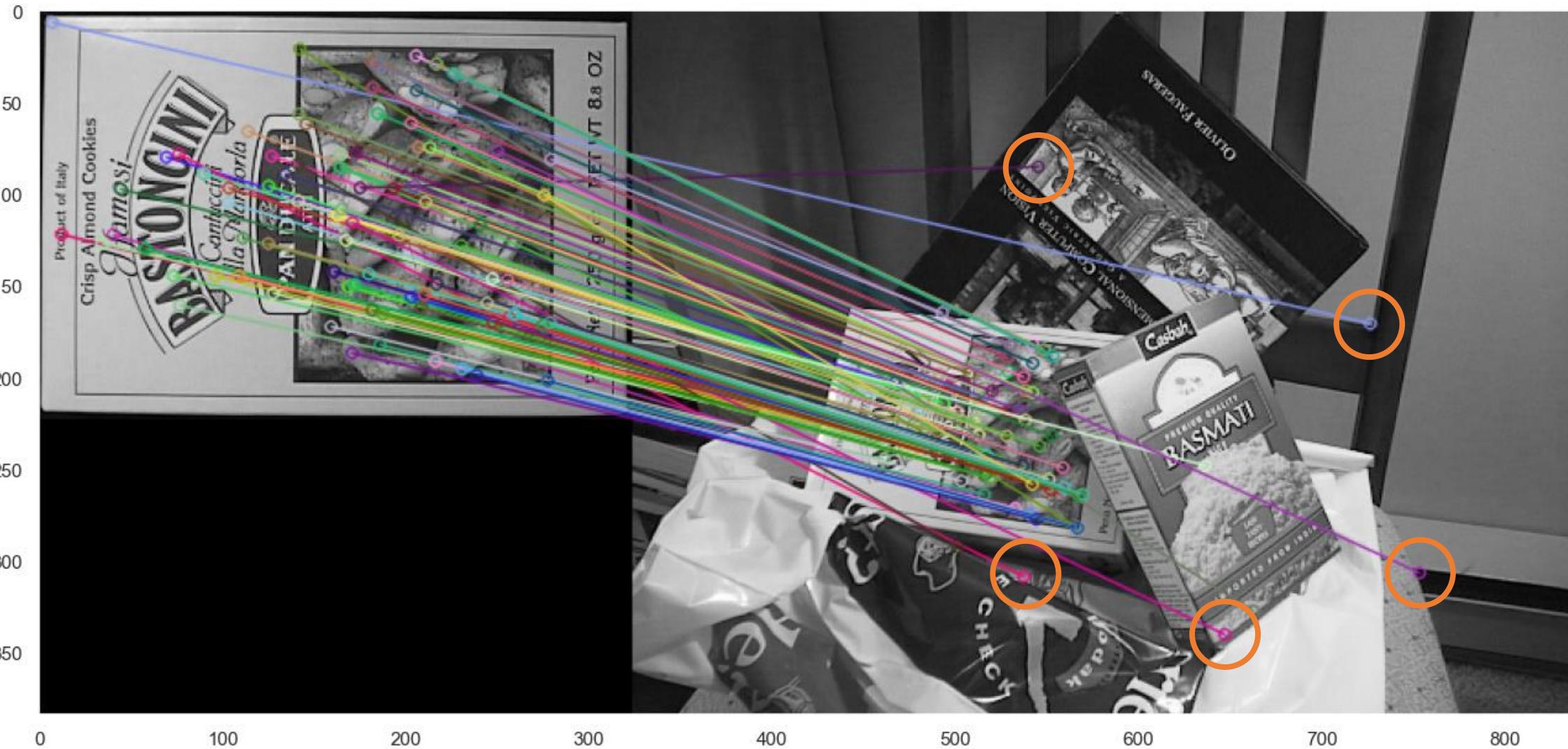
---

Random Sample consensus (RANSAC)

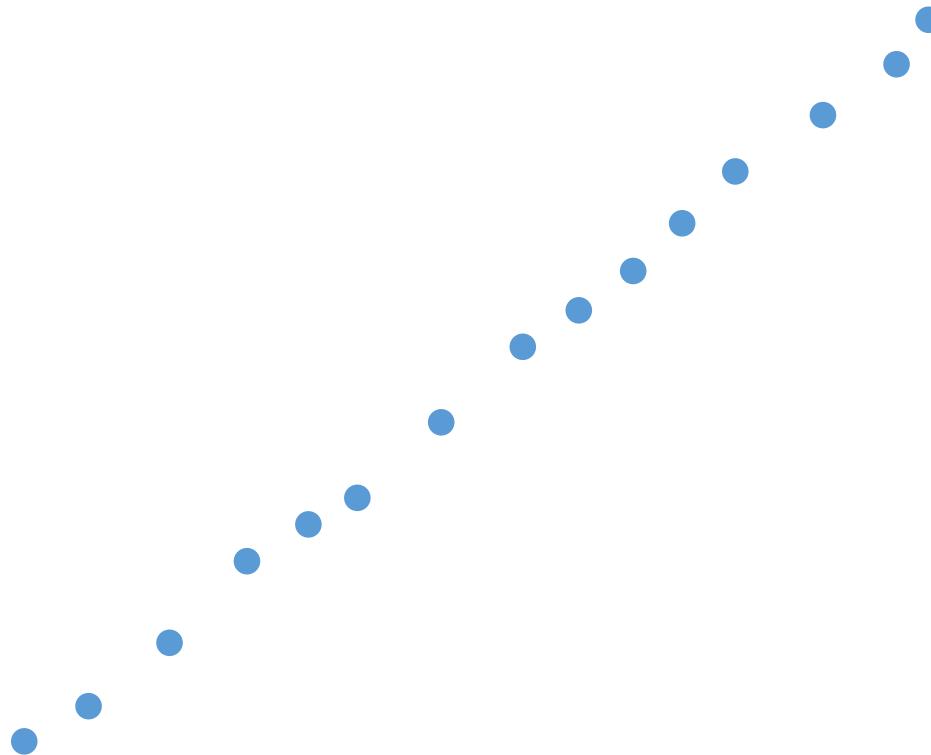
# Chyby v korespondencích



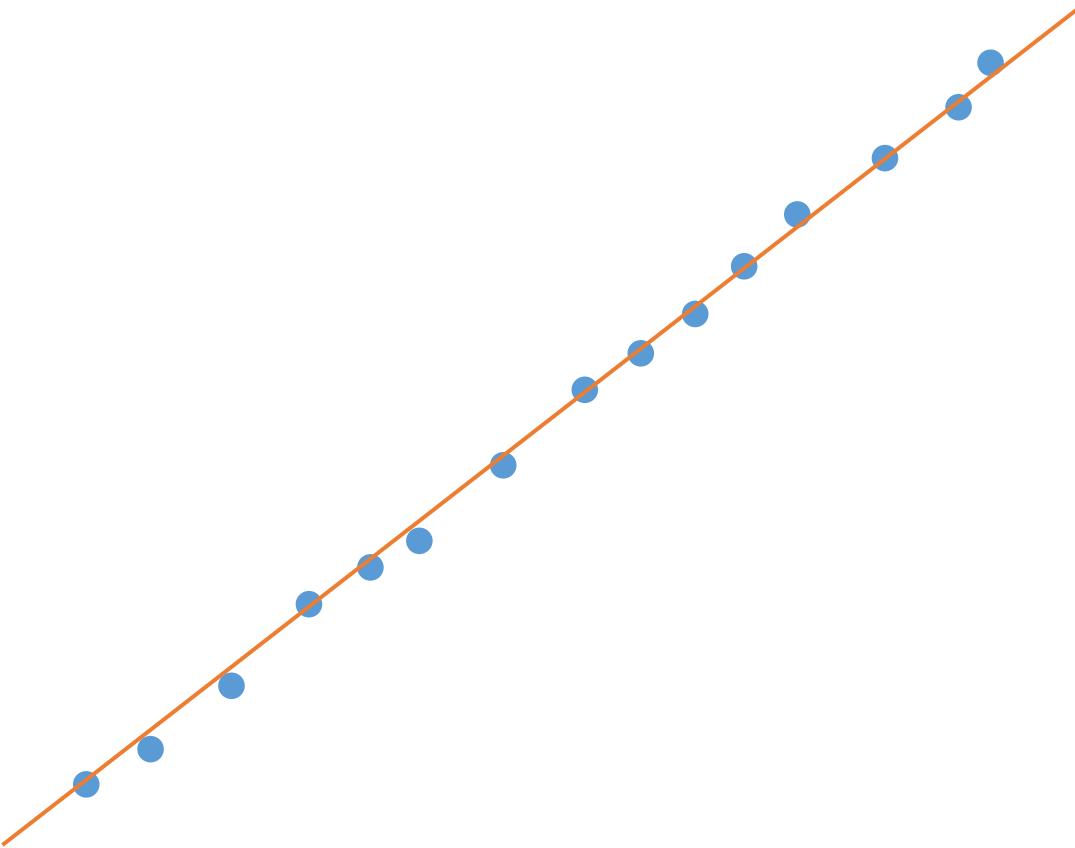
# Chyby v korespondencích



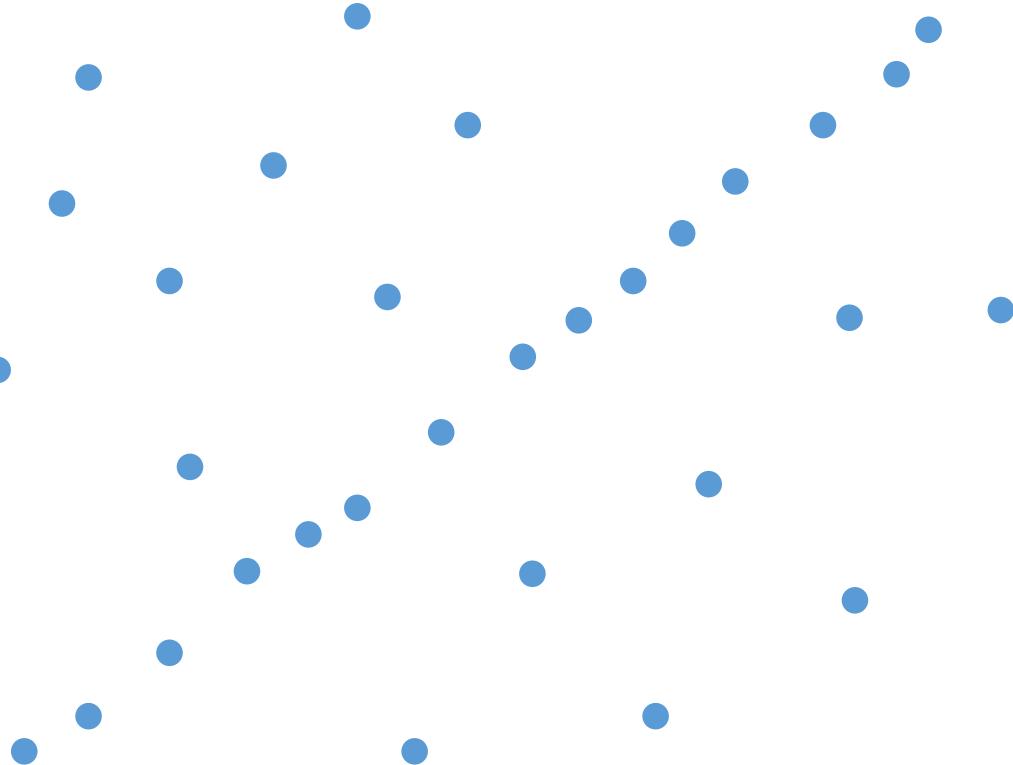
# Lineární regrese metodou nejmenších čtverců



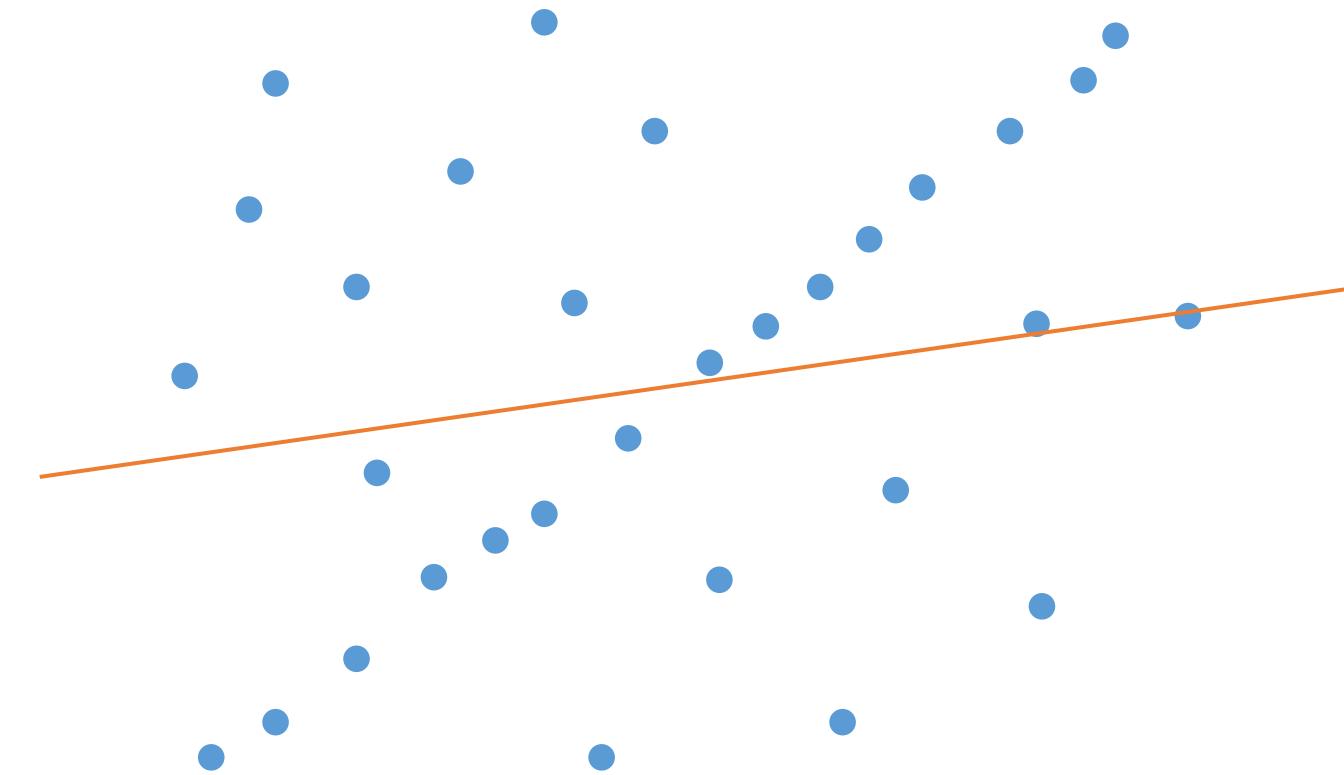
# Lineární regrese metodou nejmenších čtverců



# Lineární regrese metodou nejmenších čtverců s outliersy



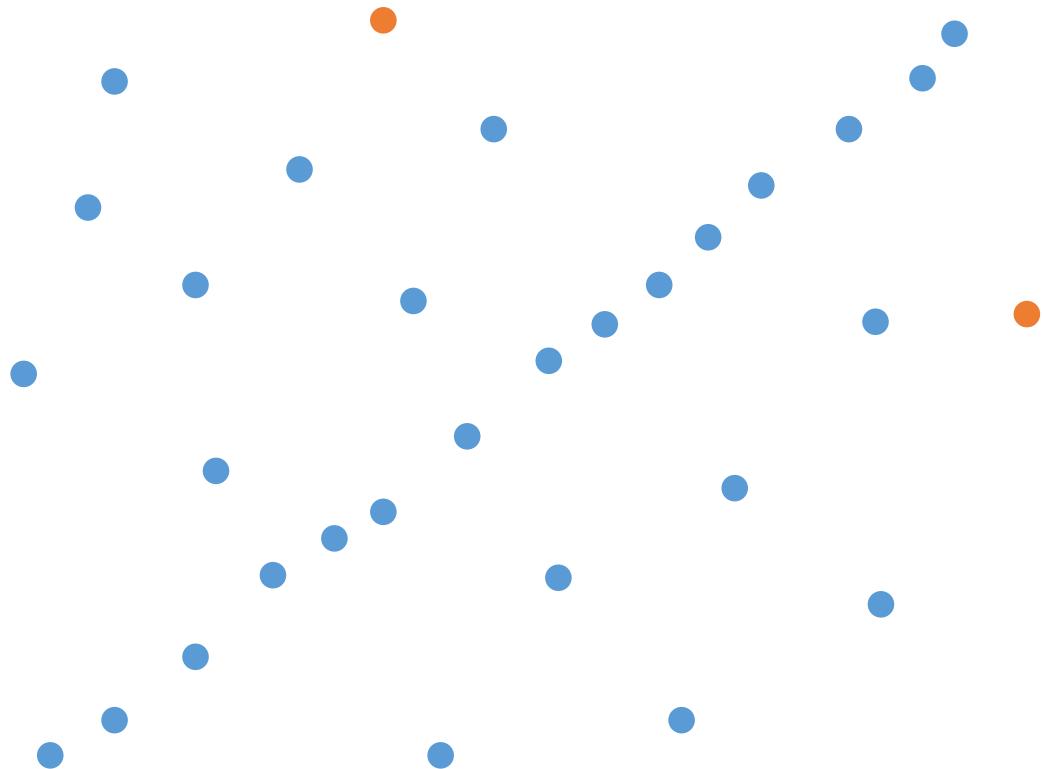
# Lineární regrese metodou nejmenších čtverců s outliersy



# Random Sample Consensus (RANSAC)

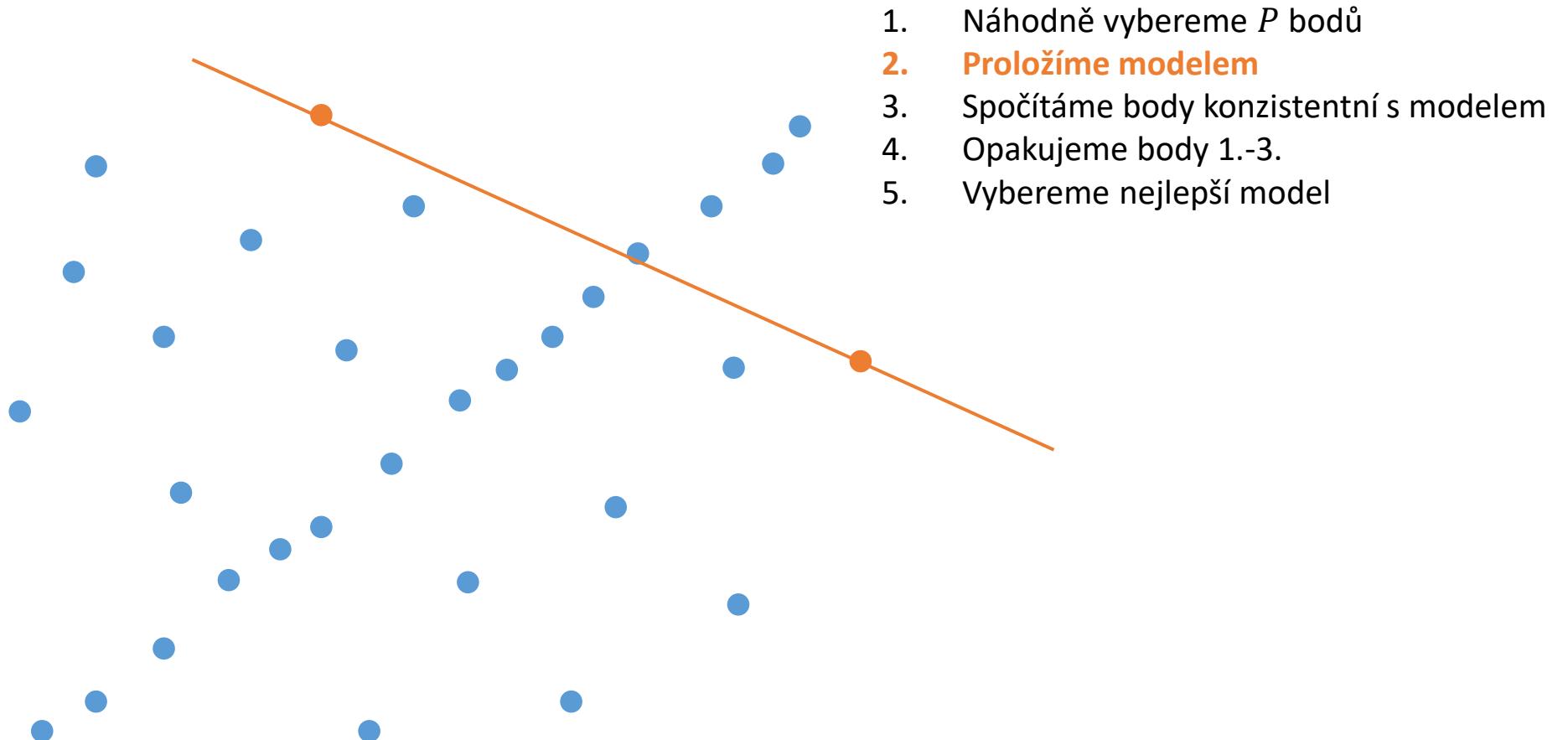
1. Náhodně vybereme  $P$  bodů
2. Proložíme modelem
3. Spočítáme body konzistentní s modelem
4. Opakujeme body 1.-3.
5. Vybereme nejlepší model

# Random Sample Consensus (RANSAC)

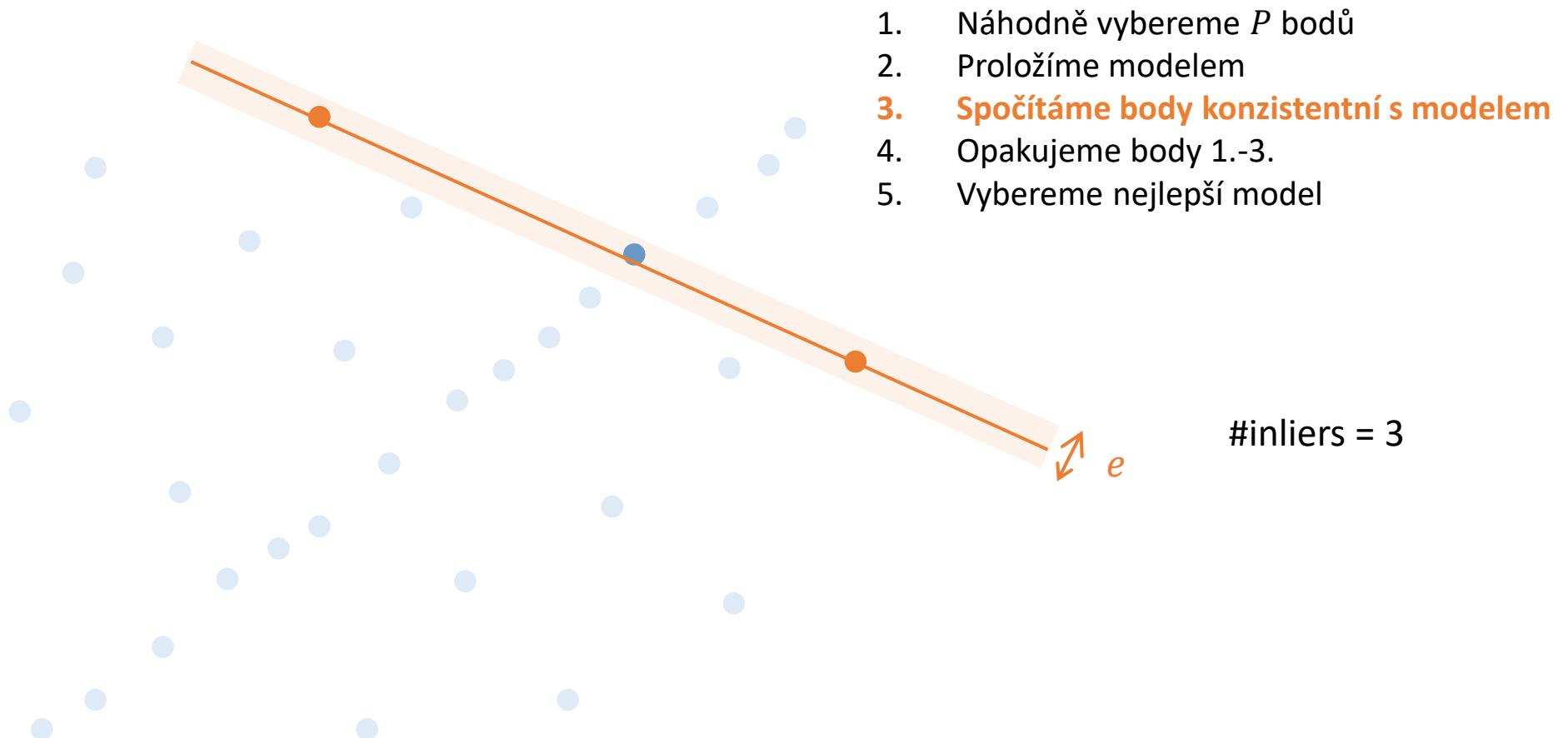


1. Náhodně vybereme  $P$  bodů
2. Proložíme modelem
3. Spočítáme body konzistentní s modelem
4. Opakujeme body 1.-3.
5. Vybereme nejlepší model

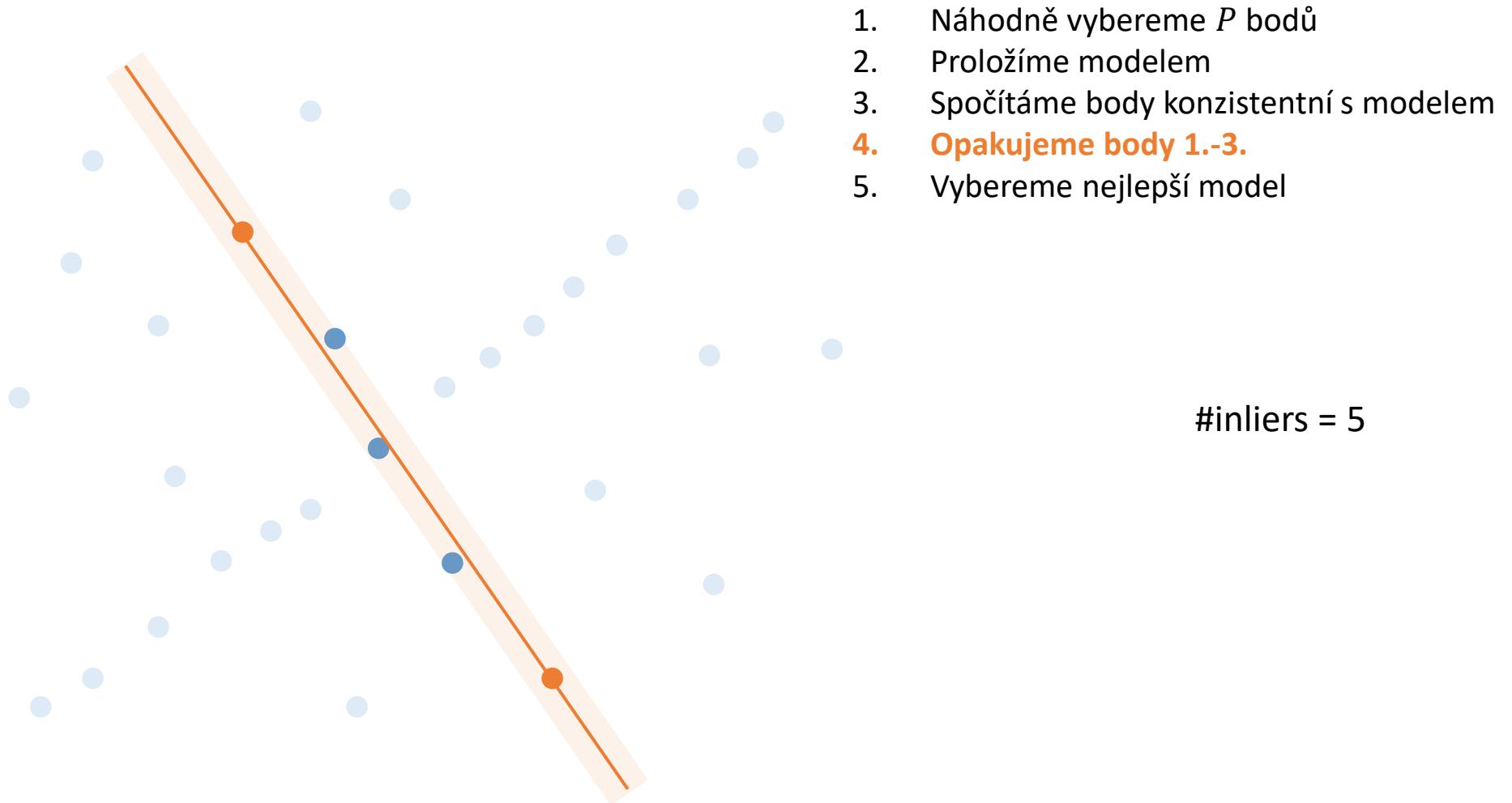
# Random Sample Consensus (RANSAC)



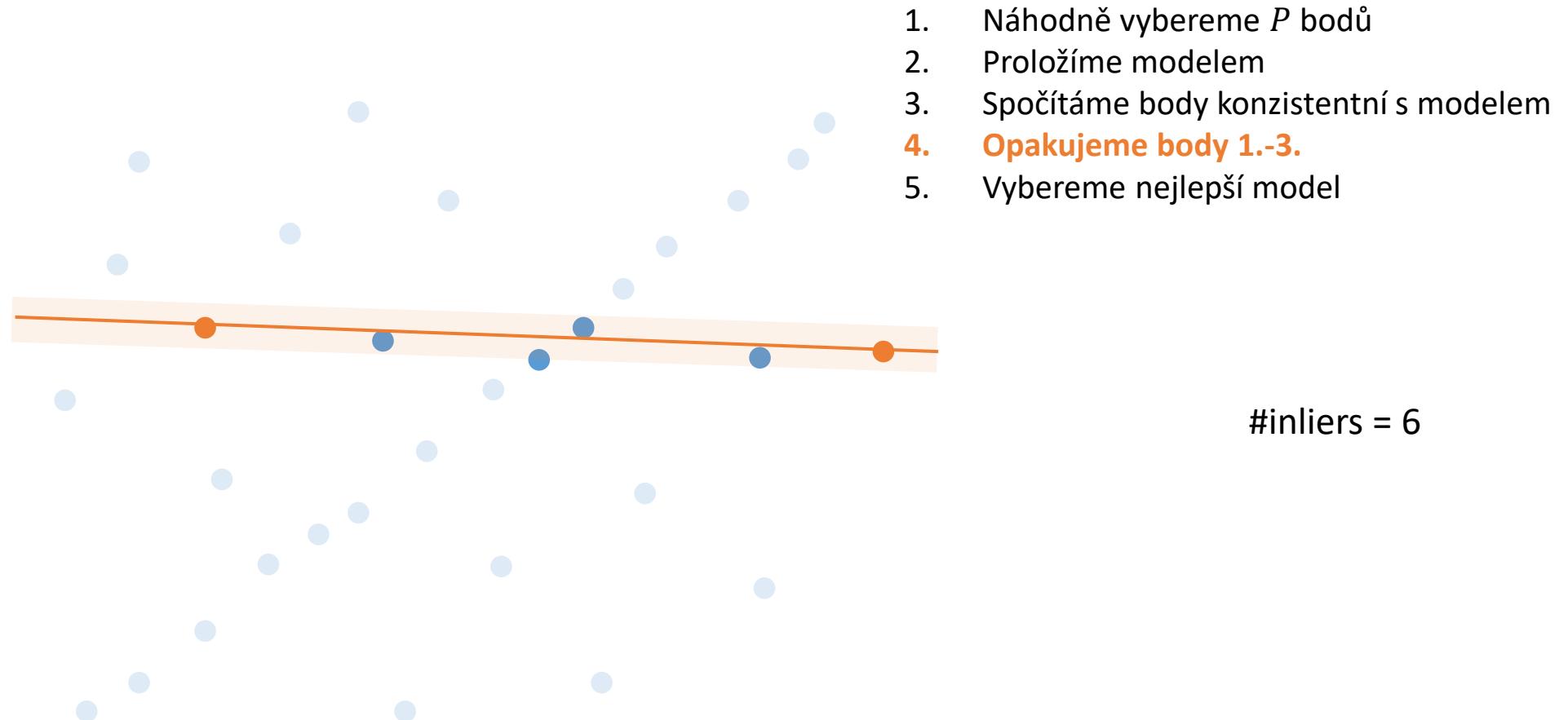
# Random Sample Consensus (RANSAC)



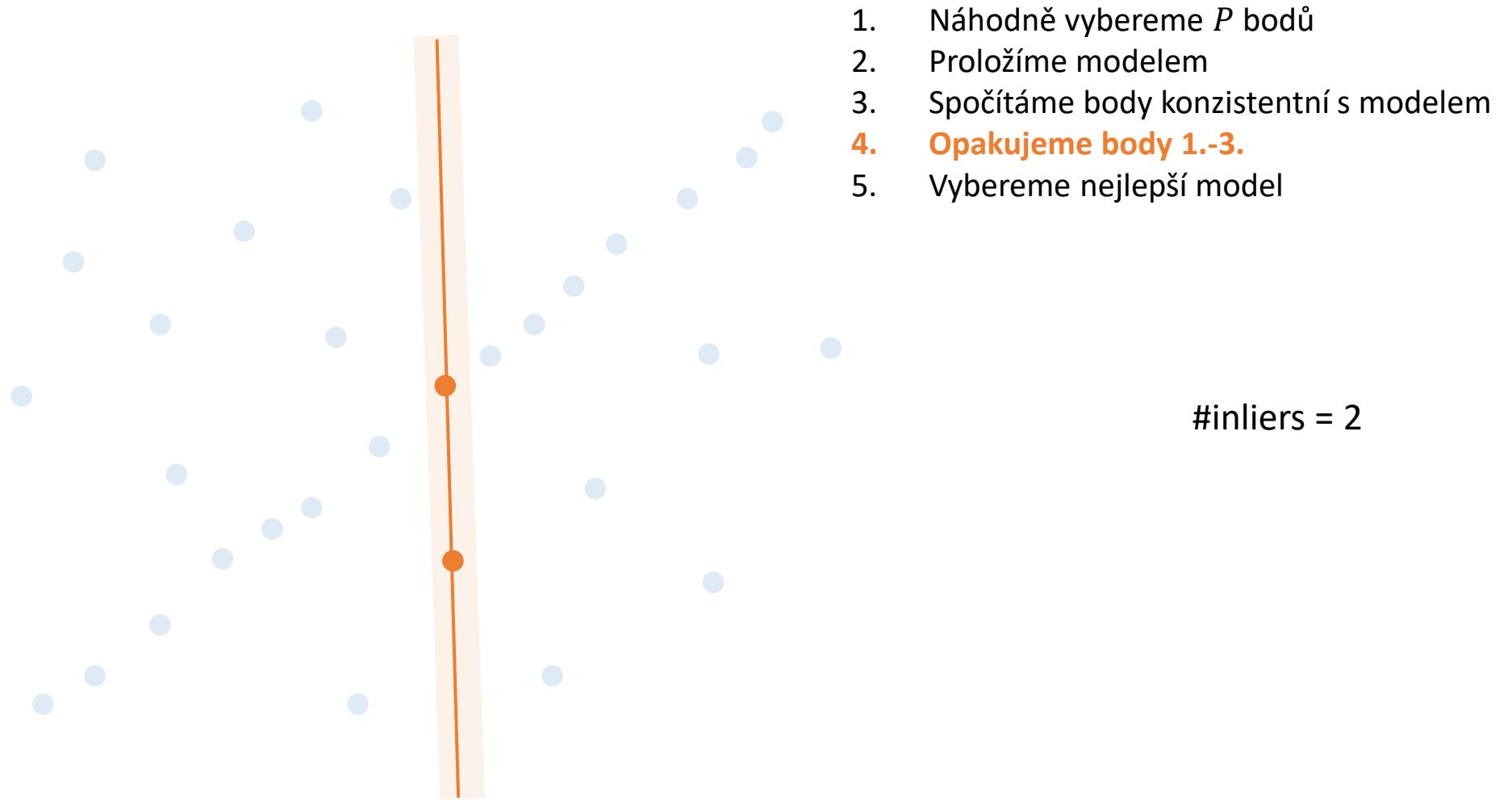
# Random Sample Consensus (RANSAC)



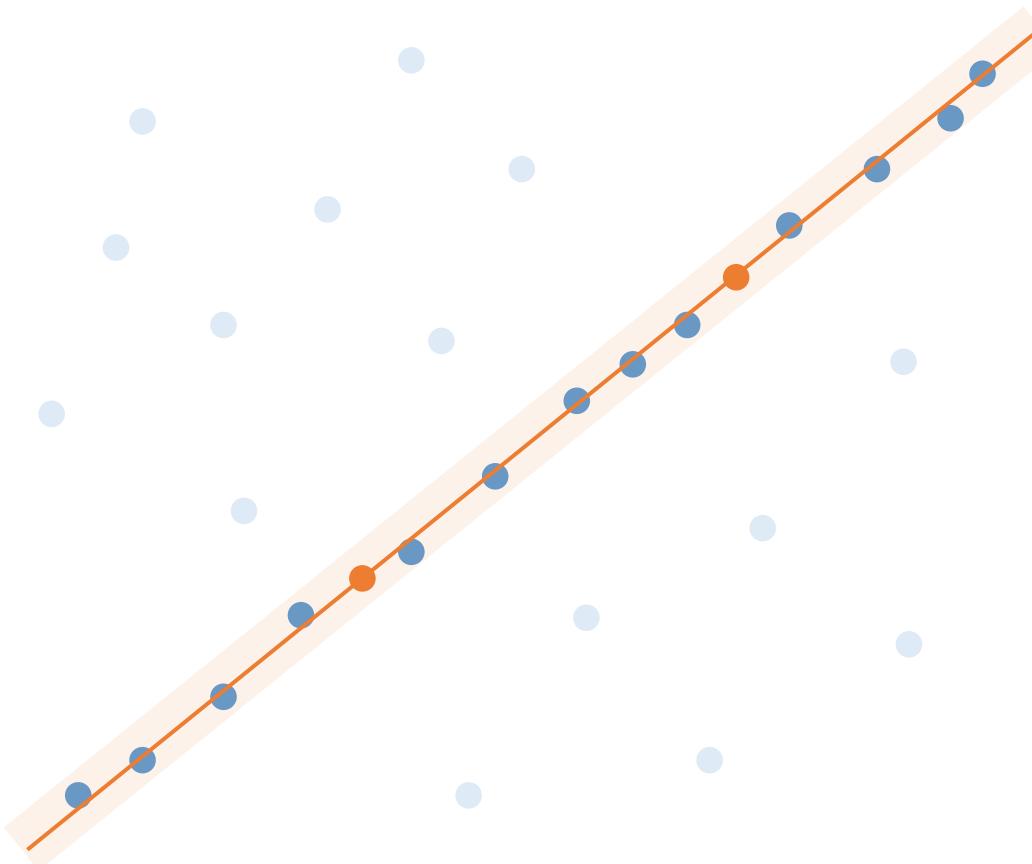
# Random Sample Consensus (RANSAC)



# Random Sample Consensus (RANSAC)



# Random Sample Consensus (RANSAC)



1. Náhodně vybereme  $P$  bodů
2. Proložíme modelem
3. Spočítáme body konzistentní s modelem
- 4. Opakujeme body 1.-3.**
5. Vybereme nejlepší model

#inliers = 15

# Hyperparametry RANSAC

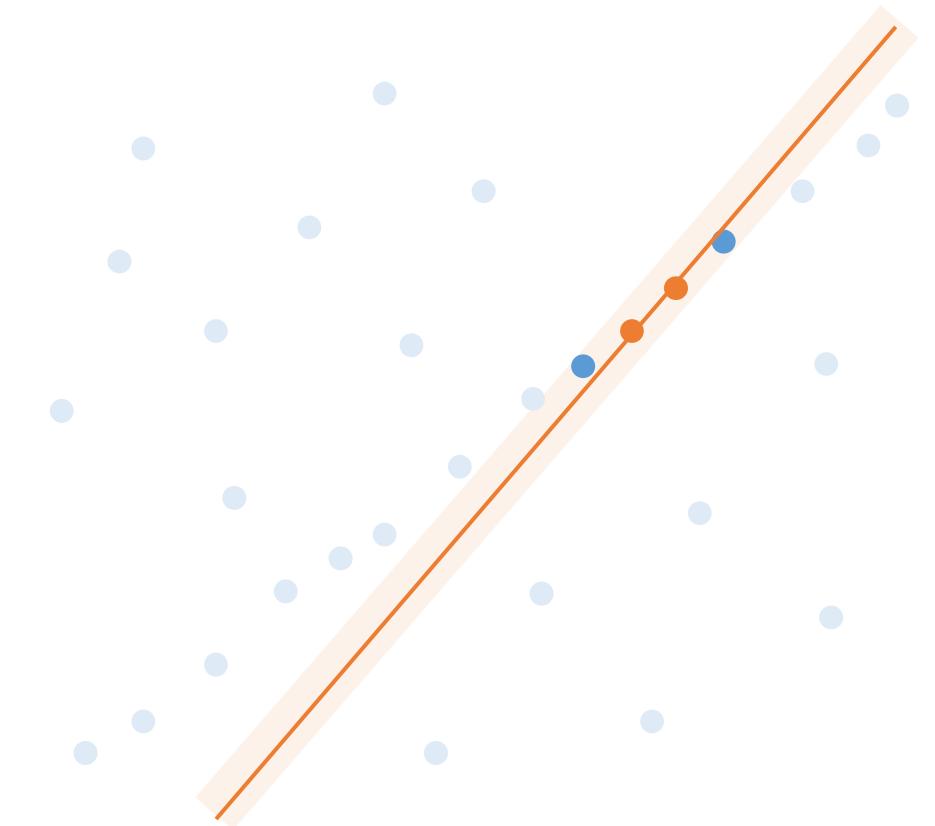
- Počet bodů  $P$ , které budeme náhodně vybírat
  - Nejčastěji nejmenší možný – např. pro přímku 2, pro homografii 4 korespondence
- Práh  $e$  na max. vzdálenost od modelu
  - Např. při předpokladu gaussovské odchylky bodů na základě sigma
- Velikost konsenzu
  - Kolik min. bodů (poměr vůči všem) z celkového datasetu by mělo odpovídat modelu, abychom považovali model za úspěšně nalezený
- Celkový počet iterací  $T$ 
  - Vybereme  $T$  tak, aby např. existovala  $p = 99\%$  pravděpodobnost, že se nám podaří vybrat  $P$  inlierů
  - Za předpokladu poměru outlierů v datasetu  $r$  lze počet iterací stanovit jako

$$(1 - (1 - r)^P)^T = 1 - p$$

$$T = \frac{\log(1 - p)}{\log(1 - (1 - r)^P)}$$

# Výhody a nevýhody RANSAC

- **Výhody**
  - Jednoduché a obecné
  - Obvykle funguje dobře
- **Nevýhody**
  - Závislé na zvolených hyperparametrech
  - Počet potřebných iterací závisí na poměru outlierů  $r$  exponenciálně
  - Někdy min. počet vzorkovaných bodů nestaci pro dobrý počáteční odhad



# Least Median of Squares (LMedS)

## RANSAC

1. Náhodně vybereme  $P$  bodů
2. Proložíme modelem
3. Spočítáme body konzistentní s modelem
4. Opakujeme body 1.-3.
5. Vybereme nejlepší model

## LMedS

1. Náhodně vybereme  $P$  bodů
2. Proložíme modelem
3. Spočítáme medián odchylky všech bodů od modelu
4. Opakujeme body 1.-3.
5. Vybereme nejlepší model

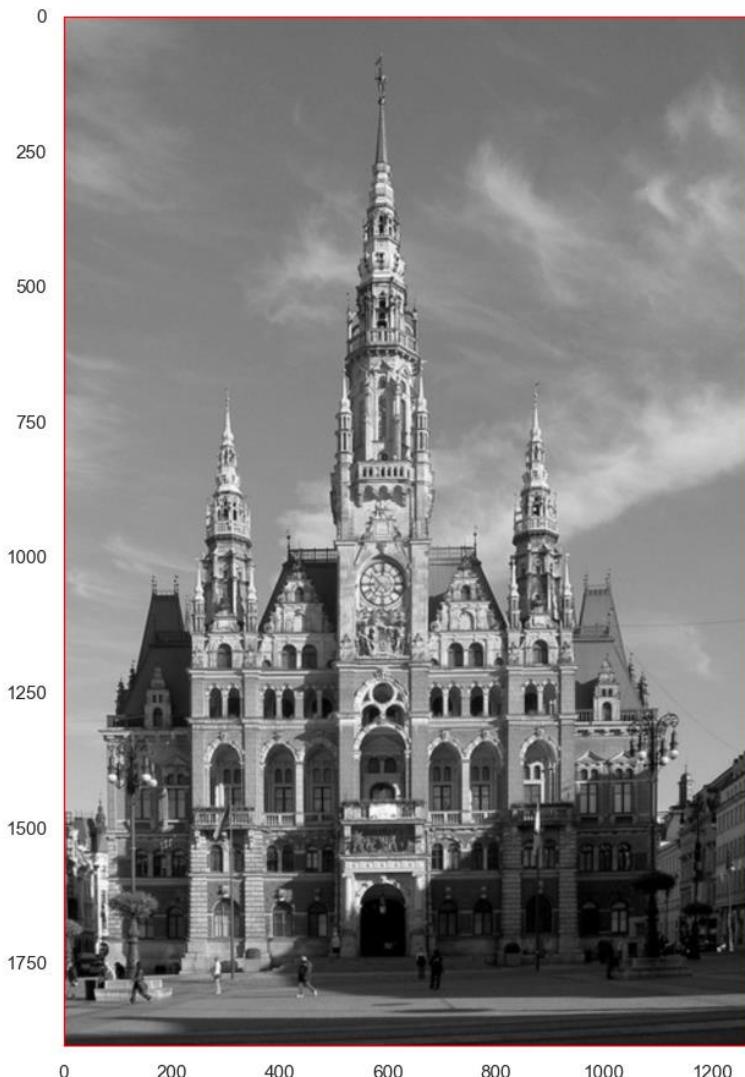
Potřebuje alespoň 50% poměr inlierů



# Odhad transformace: příklad



# Odhad transformace: příklad (horší)



# Shrnutí párování obrazů na základě lokálních korespondencí

1. Detekujeme zájmové body na obou obrazech
2. Vypočteme deskriptory zájmových bodů na obou obrazech
3. Každému bodu z obrazu 1 přiřadíme nejpodobnější bod z obrazu 2
  - Můžeme přitom zohlednit poměr vzdáleností prvního a druhého nejbližšího deskriptoru
4. Odhadneme transformaci robustní metodou
  - Např. RANSAC nebo LMedS
  - Zároveň tím dále odfiltrujeme špatně přiřazené korespondence