

# Deskriptory oblastí

Karel Horák, Jan Klečka



Rozvrh přednášky:

1. Úvod.
2. MSER.
3. SIFT.
4. SURF.

# Deskriptory oblastí

Karel Horák, Jan Klečka



Rozvrh přednášky:

- 1. Úvod.**
2. MSER.
3. SIFT.
4. SURF.

# Úvod

- ▶ Detektory významných bodů (rohů - corner features) a oblastí (blob features):
  - ▶ Moravcův operátor
  - ▶ Harrisův detektor
  - ▶ Shi&Tomasi (GFT = Good Features to Track)
  - ▶ FAST (Features from Accelerated Segment Test)
  - ▶ BRISK (Binary Robust Invariant Scalable Keypoints)
  - ▶ FREAK (Fast Retina Keypoint)
  - ▶ HOG (Histogram of Oriented Gradients)
  
- ▶ GLOH (Gradient Location and Orientation Histogram)
- ▶ BRIEF (Binary Robust Independent Elementary Features)
- ▶ ORB (Oriented FAST and Rotated BRIEF)
  
- ▶ MSER (Maximally Stable Extremal Regions)
- ▶ SIFT (Scale-Invariant Feature Transform)
- ▶ SURF (Speeded Up Robust Features)

# Úvod

## ► Literatura chronologicky:

- [1] Moravec, H. P. Towards Automatic Visual Obstacle Avoidance. 1977. (Moravec<sup>2</sup>)
- [2] Harris, C., Stephens, M. A Combined Corner and Edge Detector. 1988. (Harris<sup>2</sup>)
- [3] Shi, J., Tomasi, C. Good Features to Track. 1994. (Shi-Tomasi<sup>2</sup>)
- [4] Matas, J., et al. Robust wide baseline stereo from maximally stable extremal regions. 2002. (MSER<sup>1</sup>)
- [5] Lowe, D. G. Distinctive Image Features from Scale-Invariant Keypoints. 2004. (SIFT<sup>1</sup>)
- [6] Mikolajczyk, K., Schmid, C. A performance evaluation of local descriptors. 2005. (GLOH<sup>3</sup>)
- [7] Rosten, E., Drummond, T. Fusing Points and Lines for High Performance Tracking. 2005. (FAST<sup>2</sup>)
- [8] Dalal, N., Triggs, B. Histograms of oriented gradients for human detection. 2005. (HOG<sup>3</sup>)
- [9] Bay, H., Ess, A., Tuytelaars, T., Gool, L. Van. SURF:Speeded Up Robust Features. 2008. (SURF<sup>1</sup>)
- [10] Calonder, M., et al. BRIEF:Binary Robust Independent Elementary Features. 2010. (BRIEF<sup>3</sup>)
- [11] Leutenegger, S., Chli, M., Siegwart, R. BRISK: Binary Robust Invariant Scalable Keypoints. 2011. (BRISK<sup>3</sup>)
- [12] Rublee, E., et al. ORB: An efficient alternative to SIFT or SURF. 2011. (ORB<sup>3</sup>)
- [13] Alahi, A., Ortiz, R., Vandergheynst, P. FREAK: Fast Retina Keypoint. 2012. (FREAK<sup>3</sup>)

► <sup>1</sup> probráno v tomto materiálu

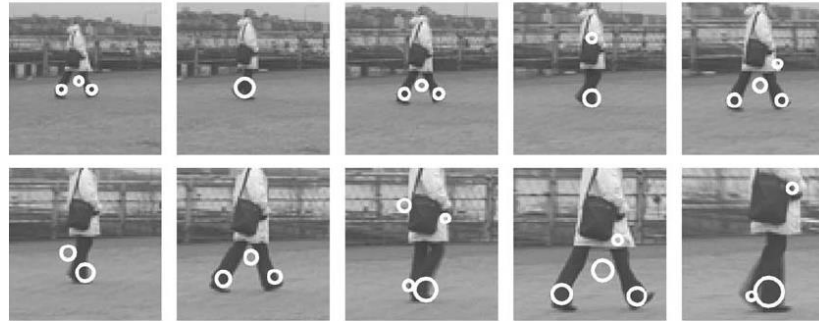
► <sup>2</sup> probráno v jiném materiálu

► <sup>3</sup> neprobráno

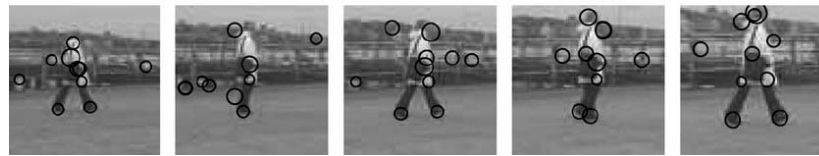
# Úvod

- Co od významných bodů očekáváme?
- Stabilita – nezávislost detekce na obecné afíní transformaci (translace, rotace, zkosení, měřítko),

*Spatio-temporal interest points*



*Spatial interest points*



- Opakovatelnost – detekce stejného znaku ve dvou a více různých snímcích stejné scény



# Deskriptory oblastí

Karel Horák, Jan Klečka

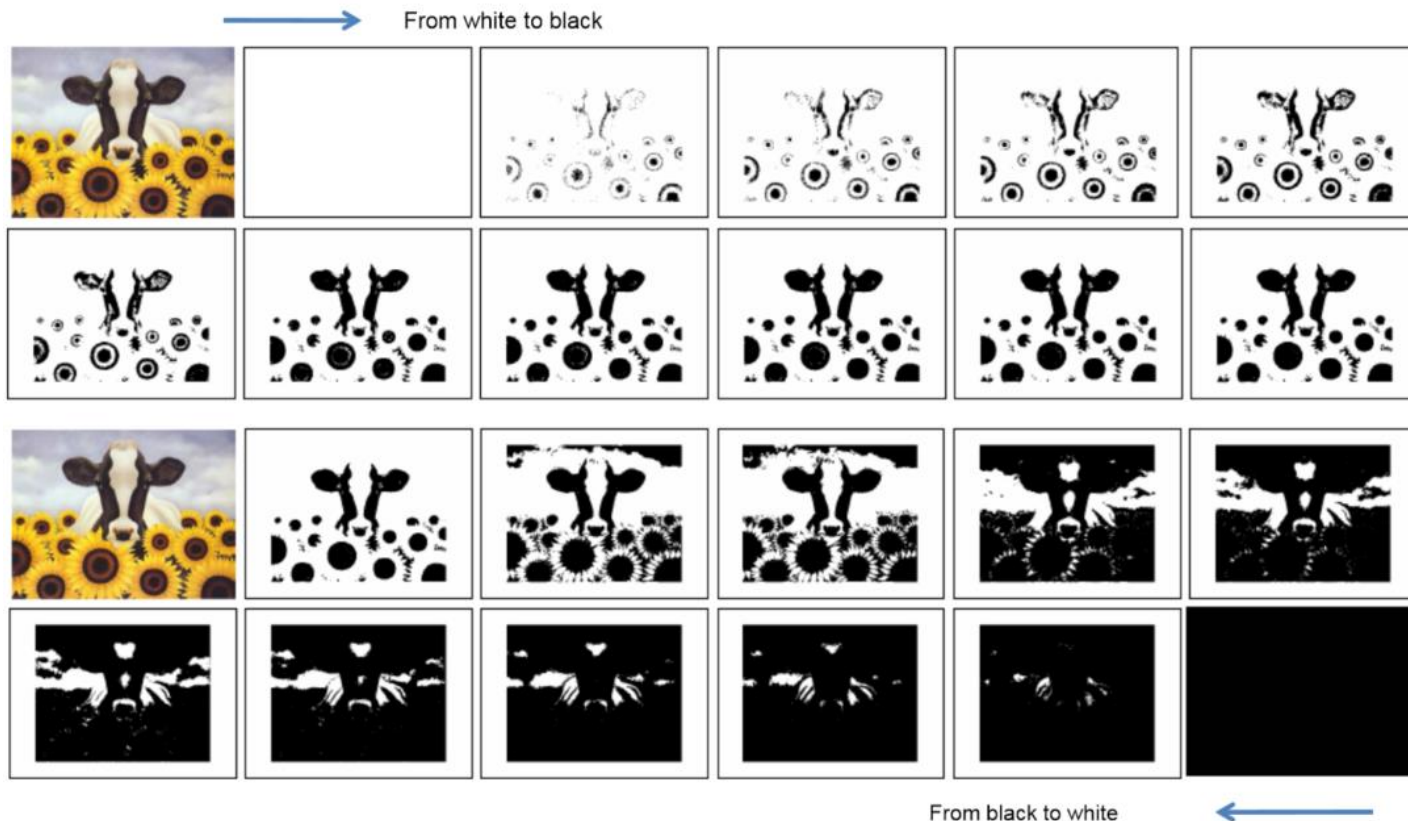


Rozvrh přednášky:

1. Úvod.
- 2. MSER.**
3. SIFT.
4. SURF.

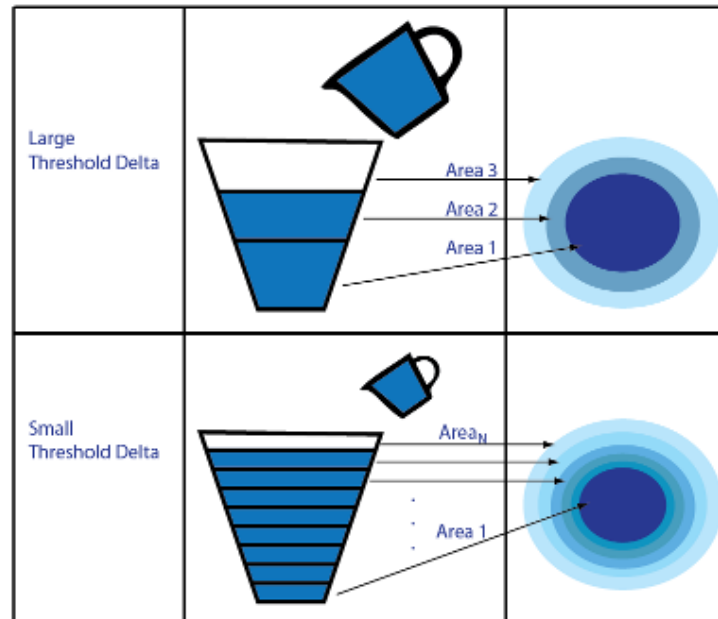
# MSER

- ▶ MSER = Maximally Stable Extremal Region
- ▶ Algoritmus postupně zvyšuje jasovou hodnotu prahu a pro každou takovou uchová prahovaný obraz s objekty oddělenými od pozadí – tzv. extrémální oblasti (Extremal Region).
- ▶ Následně dochází k analýze spojených komponent a ty oblasti, které jsou patrné ve více úrovních prahování jsou označeny jako stabilní (Maximally Stable).



# MSER

- Parametr Delta: řídicí parametr metody - malá hodnota (znázorněno malým šálkem) vyžaduje víc iterací pro průchod všemi jasovými hodnotami oblasti (znázorněno naplněním vědra) vs. velká hodnota (velký šálek) odpovídající většímu kroku a tím menšímu počtu rozlišených oblastí.





# MSER

## MSER – formálně:

**Image**  $I$  is a mapping  $I : \mathcal{D} \subset \mathbb{Z}^2 \rightarrow \mathcal{S}$ . Extremal regions are well defined on images if:

1.  $\mathcal{S}$  is totally ordered, i.e. reflexive, antisymmetric and transitive binary relation  $\leq$  exists. In this paper only  $\mathcal{S} = \{0, 1, \dots, 255\}$  is considered, but extremal regions can be defined on e.g. real-valued images ( $\mathcal{S} = \mathbb{R}$ ).
2. An adjacency (neighbourhood) relation  $A \subset \mathcal{D} \times \mathcal{D}$  is defined. In this paper 4-neighbourhoods are used, i.e.  $p, q \in \mathcal{D}$  are adjacent ( $pAq$ ) iff  $\sum_{i=1}^d |p_i - q_i| \leq 1$ .

**Region**  $Q$  is a contiguous subset of  $\mathcal{D}$ , i.e. for each  $p, q \in Q$  there is a sequence  $p, a_1, a_2, \dots, a_n, q$  and  $pAa_1, a_iAa_{i+1}, a_nAq$ .

**(Outer) Region Boundary**  $\partial Q = \{q \in \mathcal{D} \setminus Q : \exists p \in Q : qAp\}$ , i.e. the boundary  $\partial Q$  of  $Q$  is the set of pixels being adjacent to at least one pixel of  $Q$  but not belonging to  $Q$ .

**Extremal Region**  $Q \subset \mathcal{D}$  is a region such that for all  $p \in Q, q \in \partial Q : I(p) > I(q)$  (maximum intensity region) or  $I(p) < I(q)$  (minimum intensity region).

**Maximally Stable Extremal Region (MSER).** Let  $Q_1, \dots, Q_{i-1}, Q_i, \dots$  be a sequence of nested extremal regions, i.e.  $Q_i \subset Q_{i+1}$ . Extremal region  $Q_{i^*}$  is maximally stable iff  $q(i) = |Q_{i+\Delta} \setminus Q_{i-\Delta}| / |Q_i|$  has a local minimum at  $i^*$  ( $|\cdot|$  denotes cardinality).  $\Delta \in \mathcal{S}$  is a parameter of the method.

# Deskriptory oblastí

Karel Horák, Jan Klečka



---

Rozvrh přednášky:

1. Úvod.
2. MSER.
- 3. SIFT.**
4. SURF.

# SIFT

► SIFT = Scale-Invariant Feature Transform

[1] Lowe, D. G. Object Recognition from Local Scale-Invariant Features. 1999

[2] Lowe, D. G. Distinctive Image Features from Scale-Invariant Keypoints. 2004



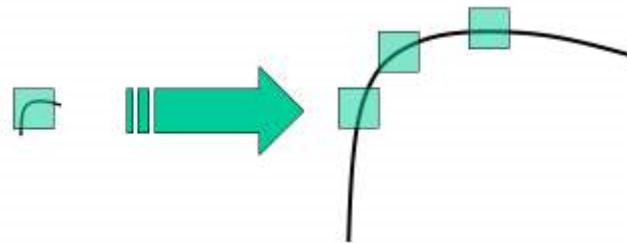
David G. Lowe ()  
Computer Science Department  
University of British Columbia  
Vancouver, Canada

► Algoritmus pro detekci a popis lokálních příznaků v obrazu.

► Obsahuje tedy detektor i deskriptor narozdíl od např. Moravce nebo Harrise (pouze detektory)

► Proč SIFT?

► Invariance na rotaci - nalezení významného bodu v otočeném obrazu nebo pokud se otáčí bod (objekt) samotný, vykazuje většina běžných detektorů jako např. Harris, FAST atd.



► Invariance na měřítko - při aplikaci Harrisova detektoru na jeden rozdílně zvětšený roh nebude jeho odezva stejná pro různá zvětšení tj. není invariantní na změnu měřítka – SIFT do značné míry ano

# SIFT

## ▀ Fáze algoritmu SIFT:

1. Detekce potenciálních významných bodů v tzv. scale-space
2. Určení stability a přesné polohy významných bodů
3. Určení orientace významných bodů
4. Výpočet příznakového vektoru pro každý významný bod

▀ První dvě fáze odpovídají v terminologii CV detektoru (nalezení významného bodu), druhé dvě fáze odpovídají deskriptoru (popis významného bodu tj. souřadnice stabilních významných bodů jsou známé a je potřeba tyto body popsat tj. jednoznačně zakódovat nejbližší okolí).

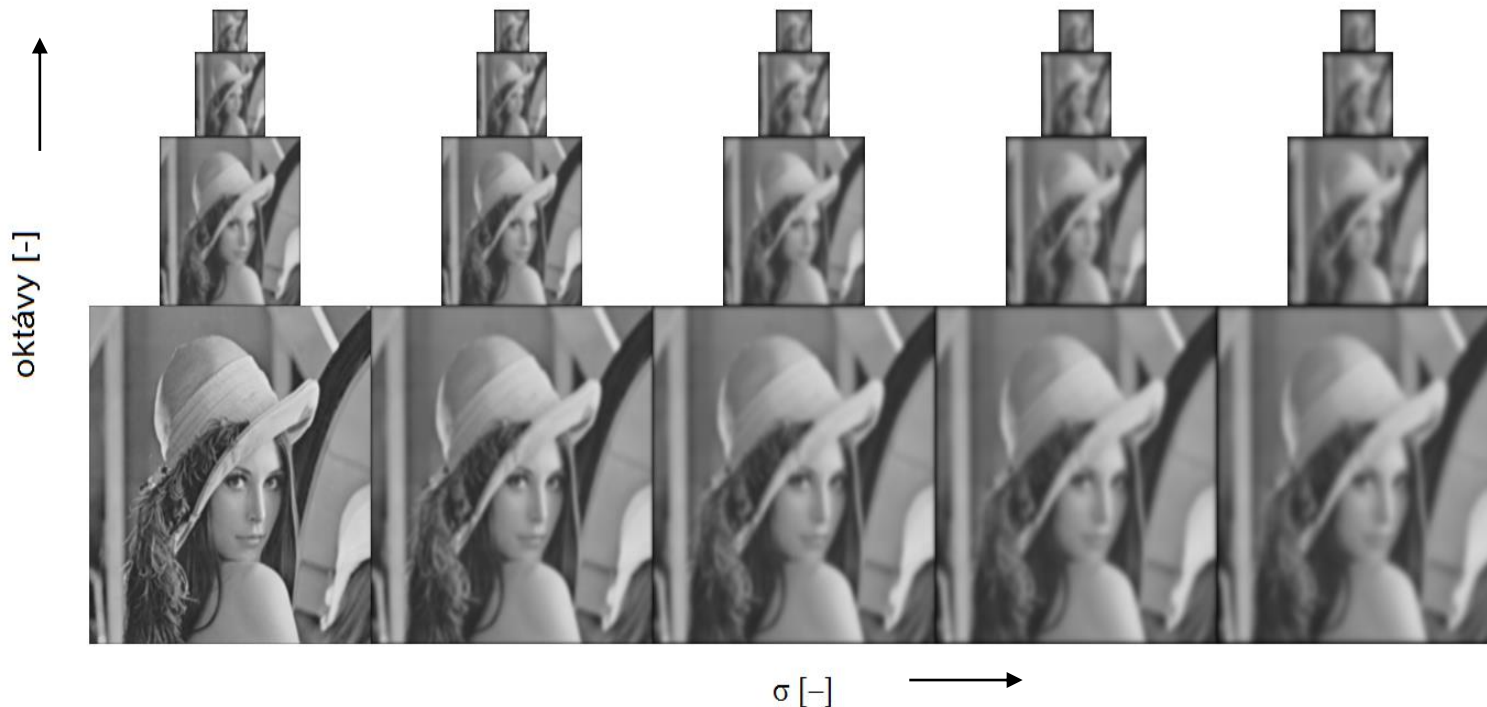
▀ Jednotlivé fáze jsou seřazeny vzestupně podle výpočetní náročnosti, aby se doba výpočtu už tak výpočetně náročného algoritmu co nejvíc zkrátila.

# SIFT

## ▀ Fáze algoritmu SIFT:

1. Detekce potenciálních významných bodů v tzv. scale-space →  $L(x, y, \sigma) = G(x, y, \sigma) * I(x, y)$
2. Určení stability a přesné polohy významných bodů
3. Určení orientace významných bodů
4. Výpočet příznakového vektoru pro každý významný bod

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}}$$



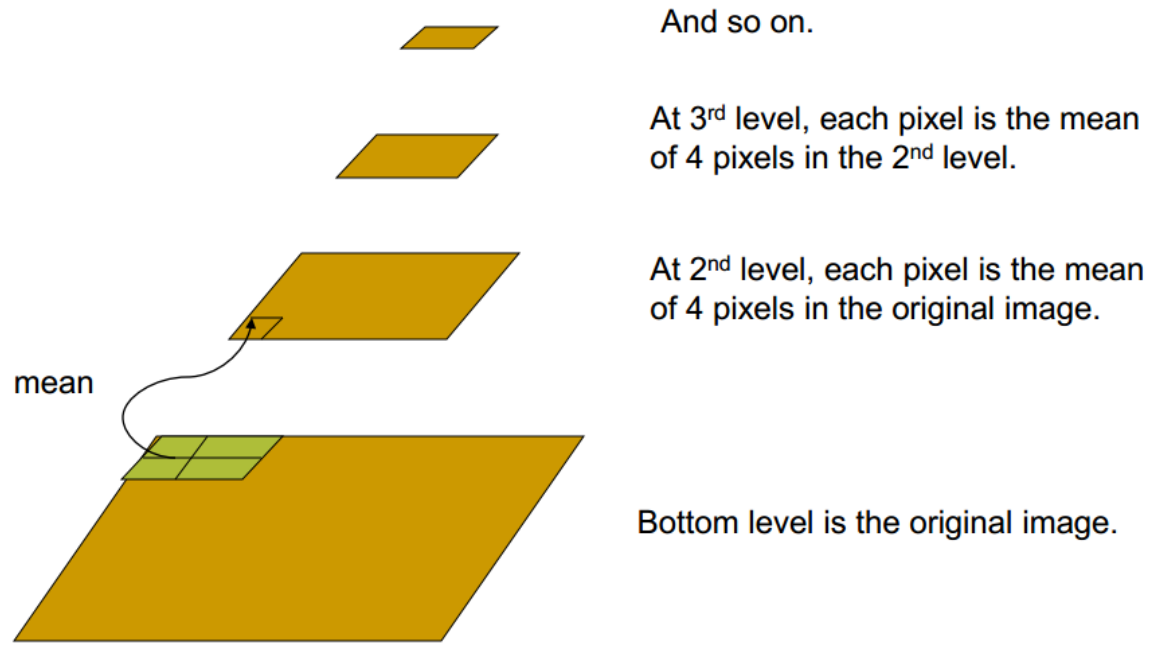
- ▀ Nastavení algoritmu (heuristika): čtyři oktávy s krokem  $\frac{1}{2}$ , pět snímků na oktávu s krokem  $\frac{\sigma_{i+1}}{\sigma_i} = \sqrt{2}$

# SIFT

## ▀ Fáze algoritmu SIFT:

1. Detekce potenciálních významných bodů v tzv. scale-space
2. Určení stability a přesné polohy významných bodů
3. Určení orientace významných bodů
4. Výpočet příznakového vektoru pro každý významný bod

tvorba oktáv

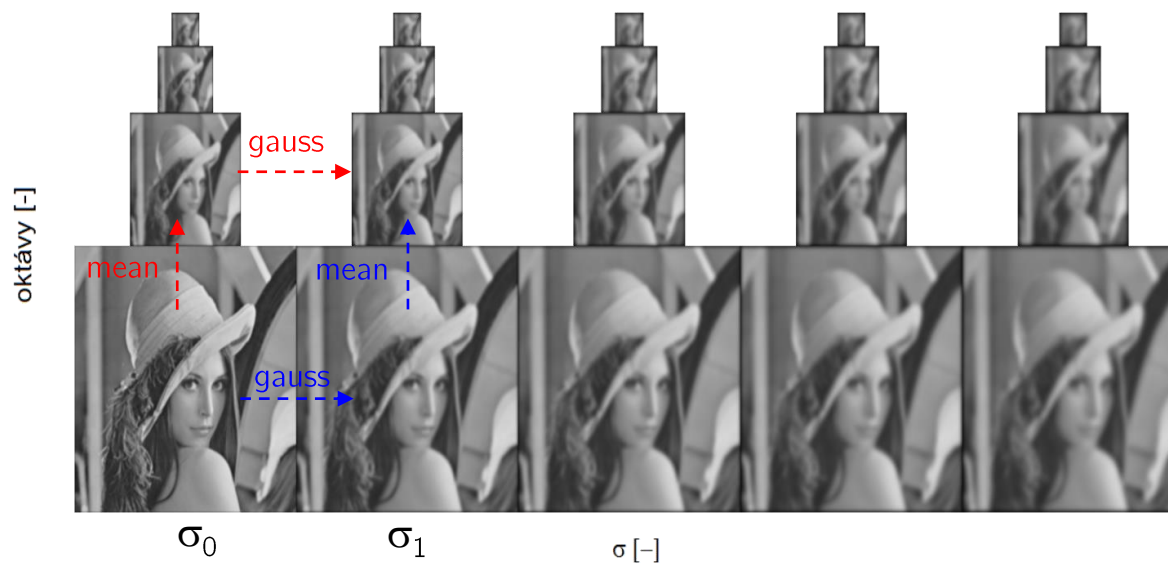


▀ Oktávy se generují prostým podvzorkováním obrazu nižší úrovně (průměr oblasti 2x2 pxl)

# SIFT

## ▮ Fáze algoritmu SIFT:

1. Detekce potenciálních významných bodů v tzv. scale-space
2. Určení stability a přesné polohy významných bodů
3. Určení orientace významných bodů
4. Výpočet příznakového vektoru pro každý významný bod



▮ K zamyšlení: obraz vyšší oktávy se  $\sigma_i$  jde vypočítat dvěma intuitivními způsoby:

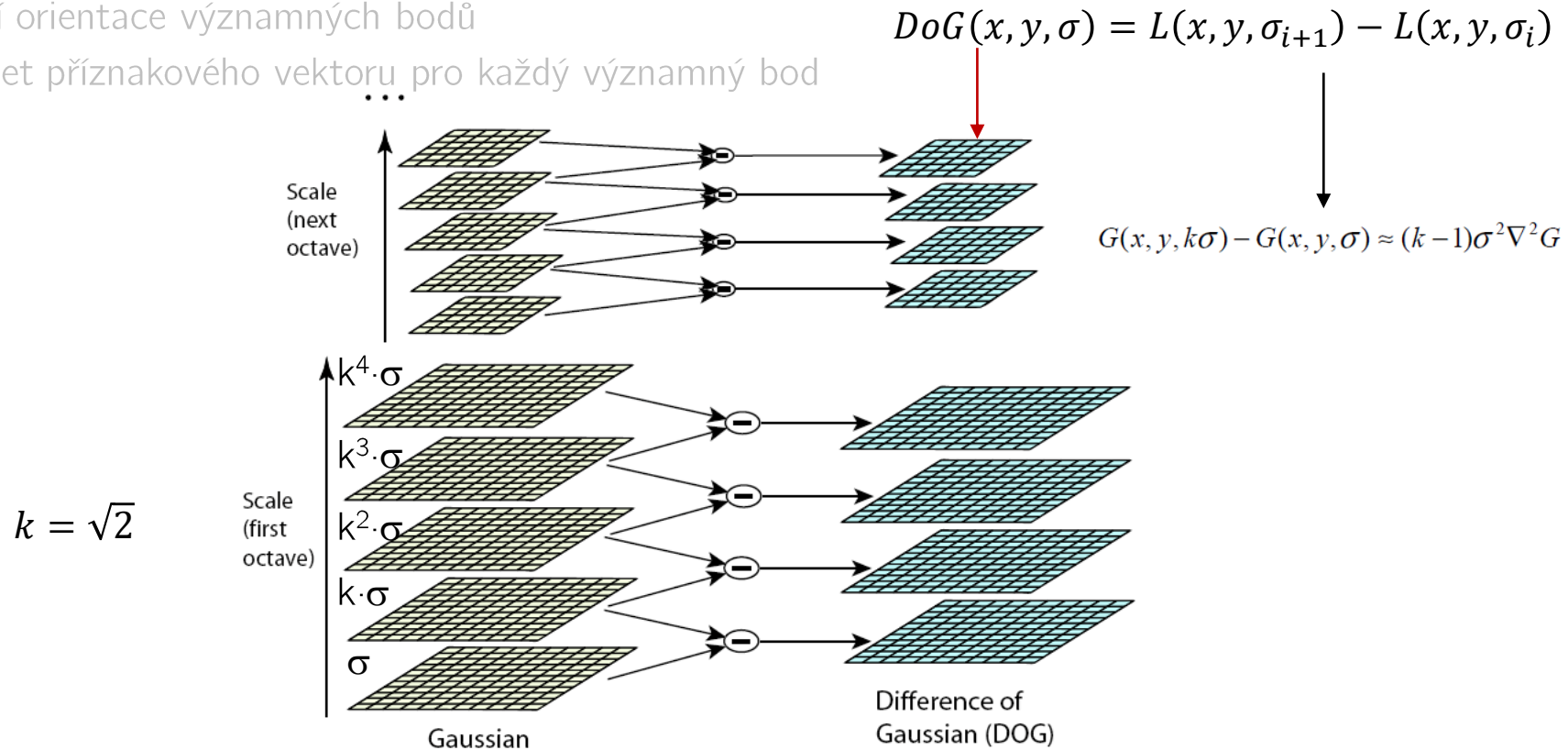
1. podvzorkováním základního obrazu nižší oktávy se  $\sigma_0$  a následnou filtrací gaussem nebo
2. přímým podvzorkováním obrazu nižší oktávy se  $\sigma_i$  bez potřeby další filtrace gaussem

▮ Která varianta je rychlejší, přesnější, vhodná?

# SIFT

## ▀ Fáze algoritmu SIFT:

1. Detekce potenciálních významných bodů v tzv. scale-space
2. Určení stability a přesné polohy významných bodů
3. Určení orientace významných bodů
4. Výpočet příznakového vektoru pro každý významný bod



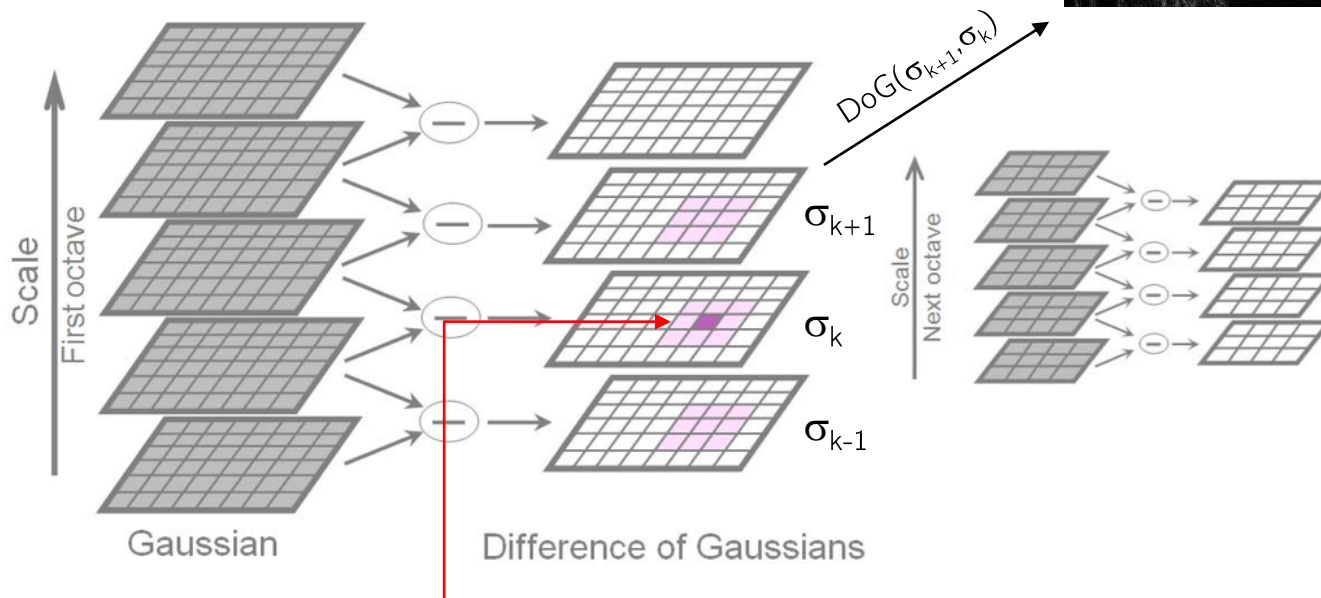
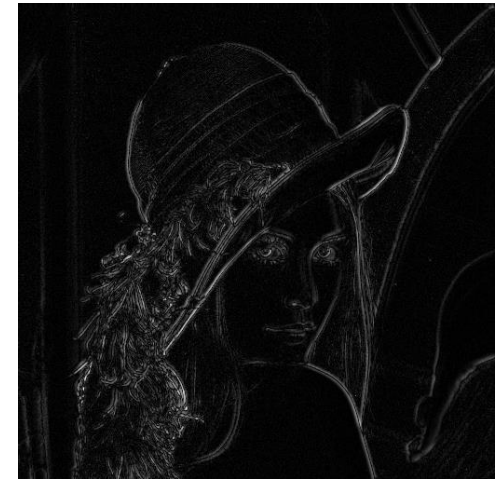
- ▀ První krok detekce: výpočet rozdílových snímků ve všech oktávách  $\Rightarrow$   $n-1$  DoG snímků v oktávě
- ▀ Druhý krok: nemaximální suprese (filtrace lokálních extrémů v DoG)



# SIFT

## ▀ Fáze algoritmu SIFT:

1. Detekce potenciálních významných bodů v tzv. scale-space
2. Určení stability a přesné polohy významných bodů
3. Určení orientace významných bodů
4. Výpočet příznakového vektoru pro každý významný bod



- ▀ Třetí krok: hodnota každého pixelu  $X$  je porovnávána s hodnotami pixelů ležícími v jeho osmiokolí na snímku se  $\sigma_k$  a poté i s hodnotami pixelů se stejnými souřadnicemi na předchozím a následujícím rozdílovém snímku se  $\sigma_{k-1}$  a  $\sigma_{k+1}$ .
- ▀ Pixel  $X$  je označen jako významný bod jen tehdy, pokud je jeho hodnota větší, než hodnota všech porovnávaných pixelů nebo je jeho hodnota menší, než hodnota všech porovnávaných pixelů.

# SIFT

## ► Fáze algoritmu SIFT:

1. Detekce potenciálních významných bodů v tzv. scale-space
2. Určení stability a přesné polohy významných bodů
3. Určení orientace významných bodů
4. Výpočet příznakového vektoru pro každý významný bod

- Stabilita – kromě eliminace bodů v DoG s nízkým kontrastem (viz [2]) jsou dále filtrovány hrany (stabilní jsou rohy, ne hrany) pomocí výpočtu Hessovy matice z hodnot obrazu  $\text{DoG}(x,y,\sigma)$  (podrobněji <http://mathworld.wolfram.com/Hessian.html>):

$$H f(x_1, x_2, \dots, x_n) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \frac{\partial^2 f}{\partial x_1 \partial x_3} & \dots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \frac{\partial^2 f}{\partial x_2 \partial x_3} & \dots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \frac{\partial^2 f}{\partial x_n \partial x_3} & \dots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix} \xrightarrow{\text{pro 2D funkci}} H f(x, y) \equiv \begin{bmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial y \partial x} & \frac{\partial^2 f}{\partial y^2} \end{bmatrix}.$$

- Pro potlačení „nerohových“ lokálních extrémů v obrazech DoG je stanoveno kritérium poměru vlastních čísel matice H:

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

- kde  $\text{tr}(H)$  je stopa (trace) Hessovy matice,  $\det(H)$  je determinant a  $r$  je poměrová konstanta (z [2] doporučeno 10 tj. poměr čtverců součtu prvků hlavní diagonály  $H$  a determinantu  $< 12.1$ )

# SIFT

## ▀ Fáze algoritmu SIFT:

1. Detekce potenciálních významných bodů v tzv. scale-space
2. Určení stability a přesné polohy významných bodů
3. Určení orientace významných bodů
4. Výpočet příznakového vektoru pro každý významný bod

- ▀ Pozn.: elegantní vztah stopy Hessovy matice a nám už známého Laplaciánu (Laplaceův operátor pro filtraci vysokých frekvencí v obrazu viz aproximace druhé derivace obrazové funkce):

$$f(x_1, x_2)$$

$$H(f) = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} \\ \frac{\partial^2 f}{\partial x_1 \partial x_2} & \frac{\partial^2 f}{\partial x_2^2} \end{pmatrix}$$

↓

Hessian matrix

$$\text{Laplace: } \nabla^2 f = \frac{\partial^2 f}{\partial x_1^2} + \frac{\partial^2 f}{\partial x_2^2} + \dots$$

⇓

$$\underline{\nabla^2 f = \text{trace } H(f)}$$

# SIFT

## ▀ Fáze algoritmu SIFT:

1. Detekce potenciálních významných bodů v tzv. scale-space
2. Určení stability a přesné polohy významných bodů
3. Určení orientace významných bodů
4. Výpočet příznakového vektoru pro každý významný bod

- ▀ Přesnost - interpolace souřadnic: okolí nalezeného významného bodu je proloženo kvadratickým polynomem a souřadnice extrému jsou vypočítány z jeho rovnice = zvýšení přesnosti lokalizace nad úroveň vzorkovací frekvence

# SIFT

## ► Fáze algoritmu SIFT:

1. Detekce potenciálních významných bodů v tzv. scale-space
2. Určení stability a přesné polohy významných bodů
3. Určení orientace významných bodů
4. Výpočet příznakového vektoru pro každý významný bod

► Úkolem tohoto kroku je pro každý detekovaný a zatím nevyřazený významný bod nalézt konzistentní směr, vůči němuž bude možné v dalším kroku provést rotačně invariantní popis.

► Pro každý bod v n-pixelovém okolí vyšetřovaného bodu z prostoru  $L(x, y)$  je vypočítána amplituda  $m(x,y)$  a směr  $\theta(x,y)$  gradientu:

$$m(x, y) = \sqrt{(L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) - L(x, y-1))^2}$$

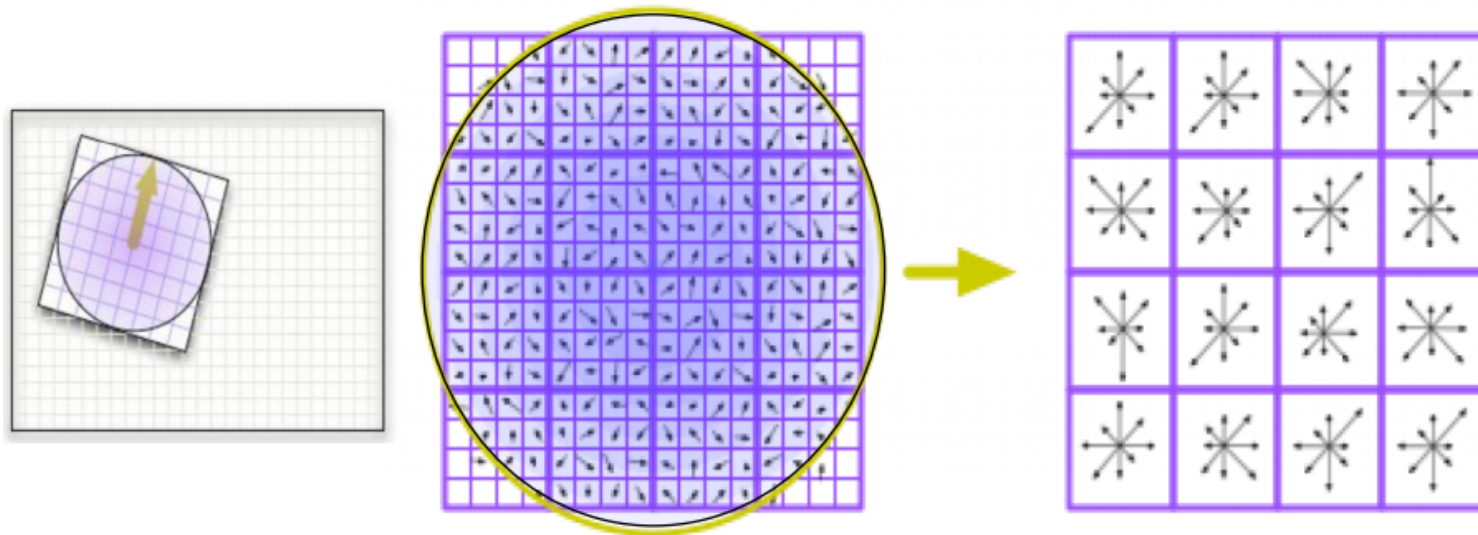
$$\theta(x, y) = \text{atan}((L(x, y+1) - L(x, y-1)) / (L(x+1, y) - L(x-1, y)))$$

- Z vypočítaných hodnot  $\langle \theta_1, \theta_n \rangle$  je sestaven orientovaný histogram s 36 kategoriemi a každý jeho prvek je vážen příslušnou amplitudou a Gaussovým oknem.
- Výsledná orientace  $\theta$  vyšetřovaného významného bodu je dána nejvyšší hodnotou histogramu.
- Pozn.: každý další vrchol histogramu, jehož hodnota dosahuje je alespoň 80% hodnoty maximální hodnoty je uznán jako další určující orientace významného bodu a v následujícím kroku je i pro něj vygenerován příznakový vektor.

# SIFT

## ► Fáze algoritmu SIFT:

1. Detekce potenciálních významných bodů v tzv. scale-space
2. Určení stability a přesné polohy významných bodů
3. Určení orientace významných bodů
4. Výpočet příznakového vektoru pro každý významný bod



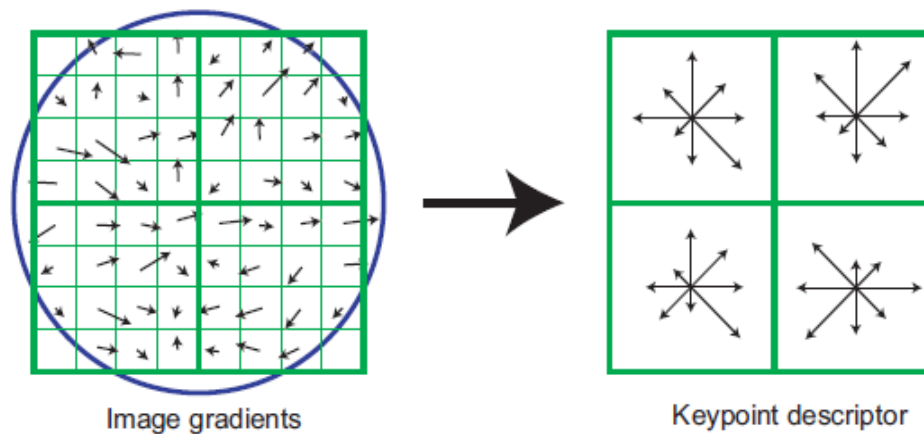
- Pro výpočet deskriptoru významného bodu je vymezena oblast 16x16 pixelů se středem a orientací  $\theta$  v místě detekce významného bodu.
- Oblast je otočena do horizontálního směru ( $\theta \rightarrow 0^\circ$ ), rozdělena na 4x4 oblasti a z každé oblasti čítající 16 pixelů je sestaven histogram orientovaných gradientů.
- každý prvek deskriptoru 2x2 jsou seřazeny jednotlivé prvky histogramu do vektoru, který po normalizaci na jednotkovou délku tvoří výsledný deskriptor.

# SIFT

## ▀ Fáze algoritmu SIFT:

1. Detekce potenciálních významných bodů v tzv. scale-space
2. Určení stability a přesné polohy významných bodů
3. Určení orientace významných bodů
4. Výpočet příznakového vektoru pro každý významný bod

## ▀ Příklad tvorby 2x2 deskriptoru z pole 8x8 pixelů:



▀ Z každého prvku deskriptoru 2x2 (vpravo) jsou seřazeny jednotlivé prvky histogramu do lineární podoby (vektoru), který po normalizaci na jednotkovou délku tvoří výsledný deskriptor.

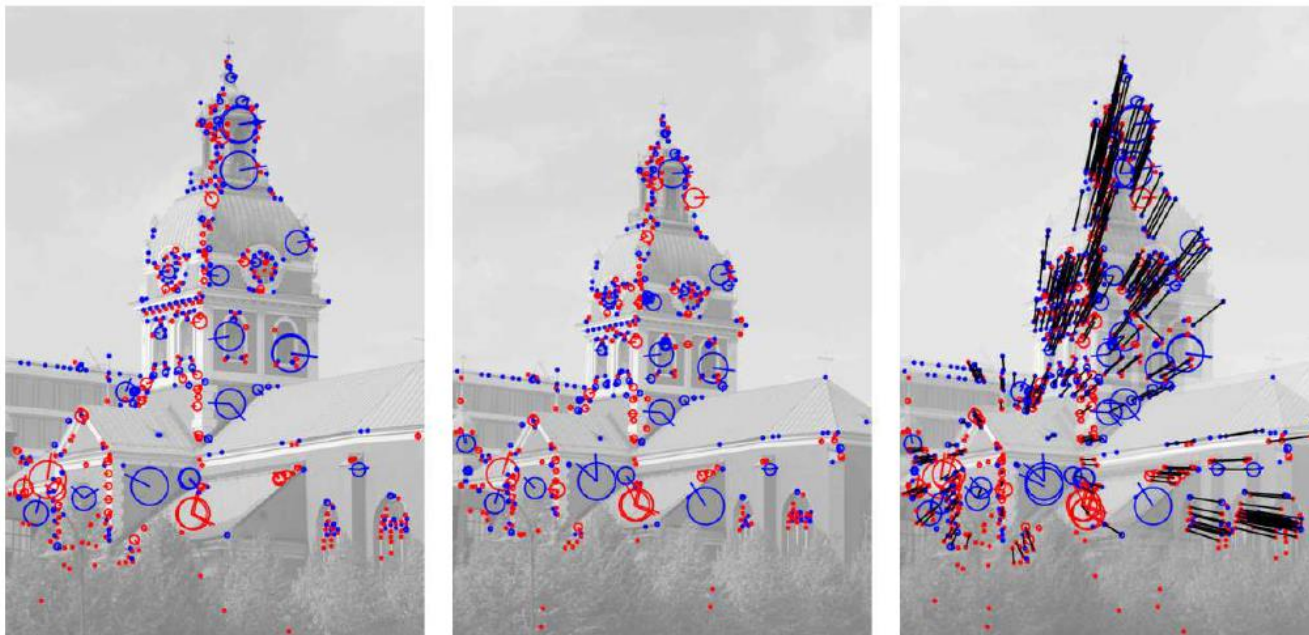
▀ Standardizovaný SIFT deskriptor = 4x4 z pole 16x16 pixelů  $\Rightarrow$  vektor 128 hodnot.



# SIFT

## ▮ Fáze algoritmu SIFT:

1. Detekce potenciálních významných bodů v tzv. scale-space
2. Určení stability a přesné polohy významných bodů
3. Určení orientace významných bodů
4. Výpočet příznakového vektoru pro každý významný bod



▮ SIFT pro korespondenci významných bodů.

▮ SIFT podrobněji: [http://www.scholarpedia.org/article/Scale\\_Invariant\\_Feature\\_Transform](http://www.scholarpedia.org/article/Scale_Invariant_Feature_Transform)



# Deskriptory oblastí

Karel Horák, Jan Klečka



Rozvrh přednášky:

1. Úvod.
2. MSER.
3. SIFT.
- 4. SURF.**

# SURF

► SURF = Speeded Up Robust Features (2006)

[1] Bay, H., Tuytelaars, T., Van Gool, L. SURF: Speeded Up Robust Features. 2006

[2] Bay, H., Tuytelaars, T., Van Gool, L., Ess, A. Speeded-Up Robust Features (SURF). 2008

► Detektor i deskriptor významných bodů, motivovaný SIFT, snaha snížit výpočetní náročnost SIFTu a současně udržet výkonnost (spolehlivost a citlivost detekce Point of Interest)

► Podobně jako SIFT, lze i algoritmus SURF definovat čtyřmi kroky:

1. Detekce významných bodů v scale-space
2. Určení přesné polohy významných bodů
3. Určení orientace významných bodů
4. Popis jednotlivých významných bodů

► Pozn.: formálně velmi podobné se SIFT, procedury použité v jednotlivých krocích jsou ale odlišné (kvůli snížení výpočetní náročnosti)

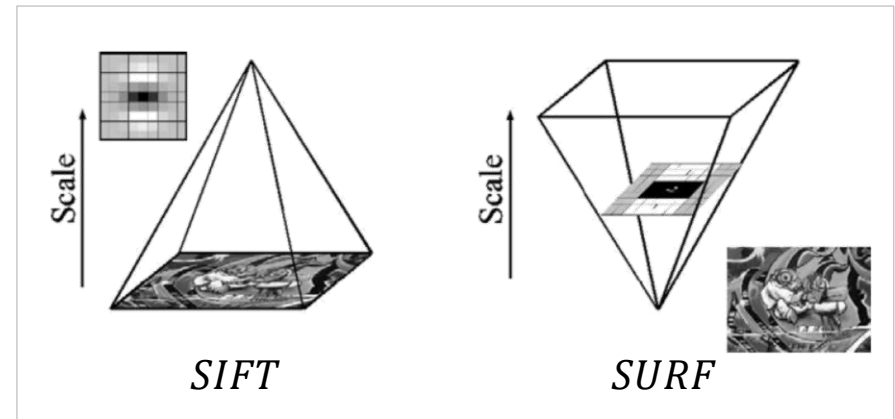
► Podrobněji:

[http://www.scholarpedia.org/article/Scale\\_Invariant\\_Feature\\_Transform#Speeded\\_up\\_robust\\_features\\_.28SURF.29](http://www.scholarpedia.org/article/Scale_Invariant_Feature_Transform#Speeded_up_robust_features_.28SURF.29)

# SURF

## ► Fáze algoritmu SURF:

1. Detekce významných bodů v scale-space
2. Určení přesné polohy významných bodů
3. Určení orientace významných bodů
4. Popis jednotlivých významných bodů



► Pozn.: scale-space se tvoří změnou velikosti masky, nikoliv původního obrazu jako u SIFT —→

► V případě SIFT jsou lokální extrémů hledány pomocí DoG, v případě SURF se pro detekci lokálních extrémů využívá determinantu Hessovy matice:

$$H(\mathbf{x}, \sigma) = \begin{pmatrix} L_{xx}(\mathbf{x}, \sigma) & L_{xy}(\mathbf{x}, \sigma) \\ L_{xy}(\mathbf{x}, \sigma) & L_{yy}(\mathbf{x}, \sigma) \end{pmatrix} \quad \text{kde}$$



$$\det(H(\mathbf{x}, \sigma)) = L_{xx}(x, \sigma) \cdot L_{yy}(x, \sigma) - L_{xy}(x, \sigma)^2$$

$$L_{xx}(\mathbf{x}, \sigma) = I(\mathbf{x}) * \frac{\partial^2 g(\sigma)}{\partial x^2}$$

$$L_{xy}(\mathbf{x}, \sigma) = I(\mathbf{x}) * \frac{\partial^2 g(\sigma)}{\partial xy}$$

$$L_{yy}(\mathbf{x}, \sigma) = I(\mathbf{x}) * \frac{\partial^2 g(\sigma)}{\partial y^2}$$

$$\mathbf{x} = (x \ y)^T$$

►  $g(\sigma)$  je 2D Gaussova funkce,  $\sigma$  její rozptyl (scale detektoru) a  $I(\mathbf{x})$  vstupní obraz.

# SURF

## ▀ Fáze algoritmu SURF:

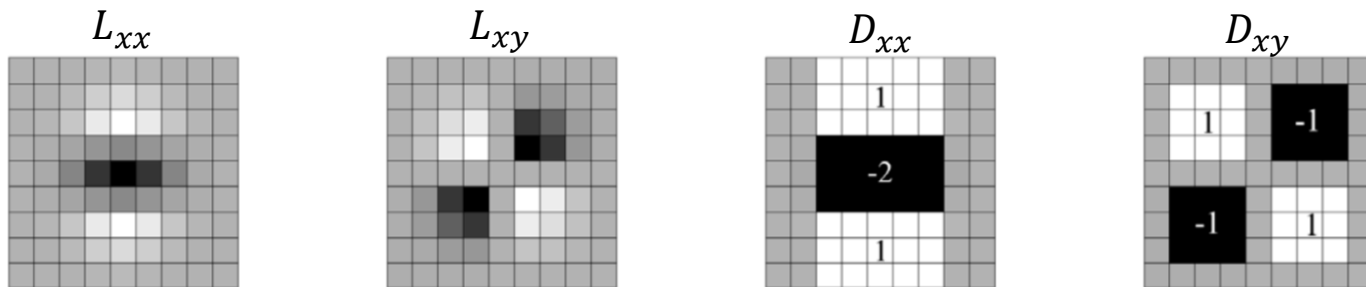
1. Detekce významných bodů v scale-space
2. Určení přesné polohy významných bodů
3. Určení orientace významných bodů
4. Popis jednotlivých významných bodů

„box functions“ umožňují rapidní zrychlení výpočtu konvoluce  $I(x)*g''(\sigma)$  použitím integrálního obrazu  $I_{\Sigma}$  (tři operace  $\pm$  pro výpočet sumy libovolné oblasti  $R$ ).

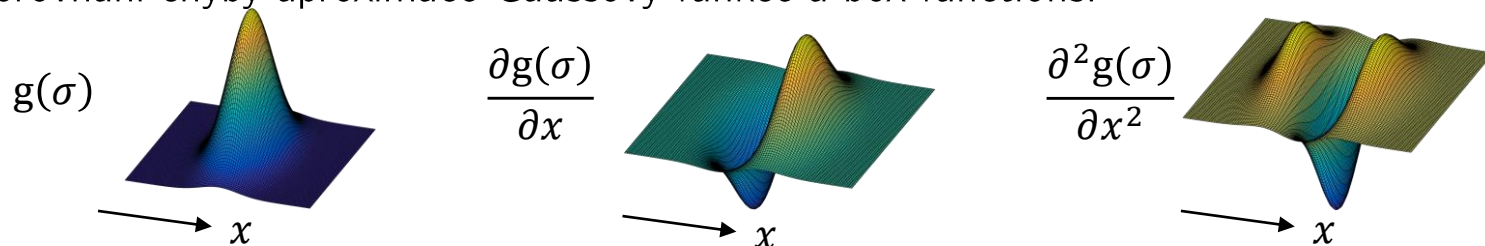
$$\text{Bez } I_{\Sigma} : \Sigma(R) = \sum_{R\_width} \sum_{R\_height} I(x,y)$$

$$\text{S } I_{\Sigma} : \Sigma(R) = I_{\Sigma}(R_{LR}) - I_{\Sigma}(R_{UR}) - I_{\Sigma}(R_{LL}) + I_{\Sigma}(R_{UL})$$

- ▀ SURF používá namísto DoG pro detekci lokálních extrémů Haarovy vlnky (navíc jen jejich aproximaci tzv. „box functions“  $D_{xx}$ ,  $D_{xy}$  a  $D_{yy}$ )



- ▀ Pro porovnání chyby aproximace Gaussovy funkce a box functions:



# SURF

## ▀ Fáze algoritmu SURF:

1. Detekce významných bodů v scale-space
2. Určení přesné polohy významných bodů
3. Určení orientace významných bodů
4. Popis jednotlivých významných bodů

▀ Používá stejný algoritmus jako metoda SIFT, pouze se pro obrazy ve scale-space počítá příslušná hodnota  $\sigma$ .

# SURF

## ▀ Fáze algoritmu SURF:

1. Detekce významných bodů v scale-space
2. Určení přesné polohy významných bodů
3. Určení orientace významných bodů
4. Popis jednotlivých významných bodů

- ▀ SURF používá pro sestavení deskriptoru sumy a absolutní sumy derivací prvního řádu ( $\Sigma d_x$ ,  $\Sigma d_y$ ,  $\Sigma |d_x|$  a  $\Sigma |d_y|$ ) namísto histogramů směrů