

SIFT:

Scale Invariant Feature Transform

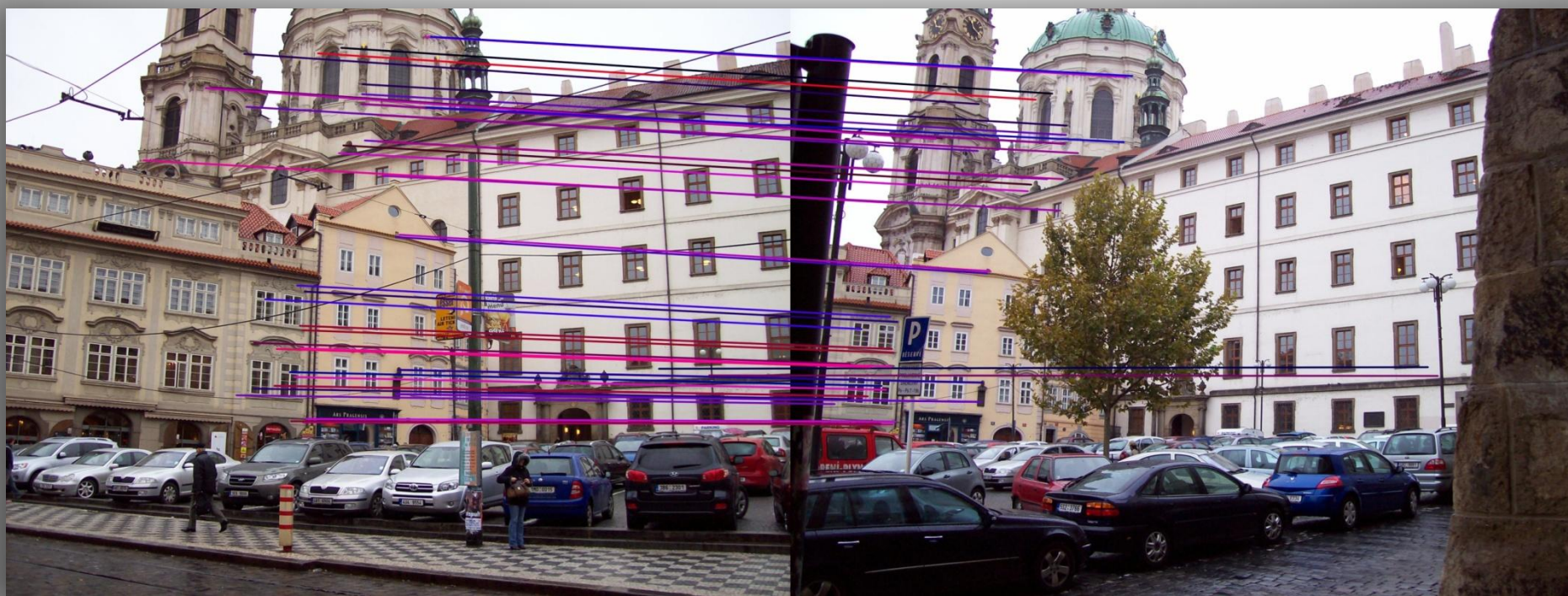
Automatické nalezení korespondencí
mezi dvojicí obrázků

lukas.mach@gmail.com

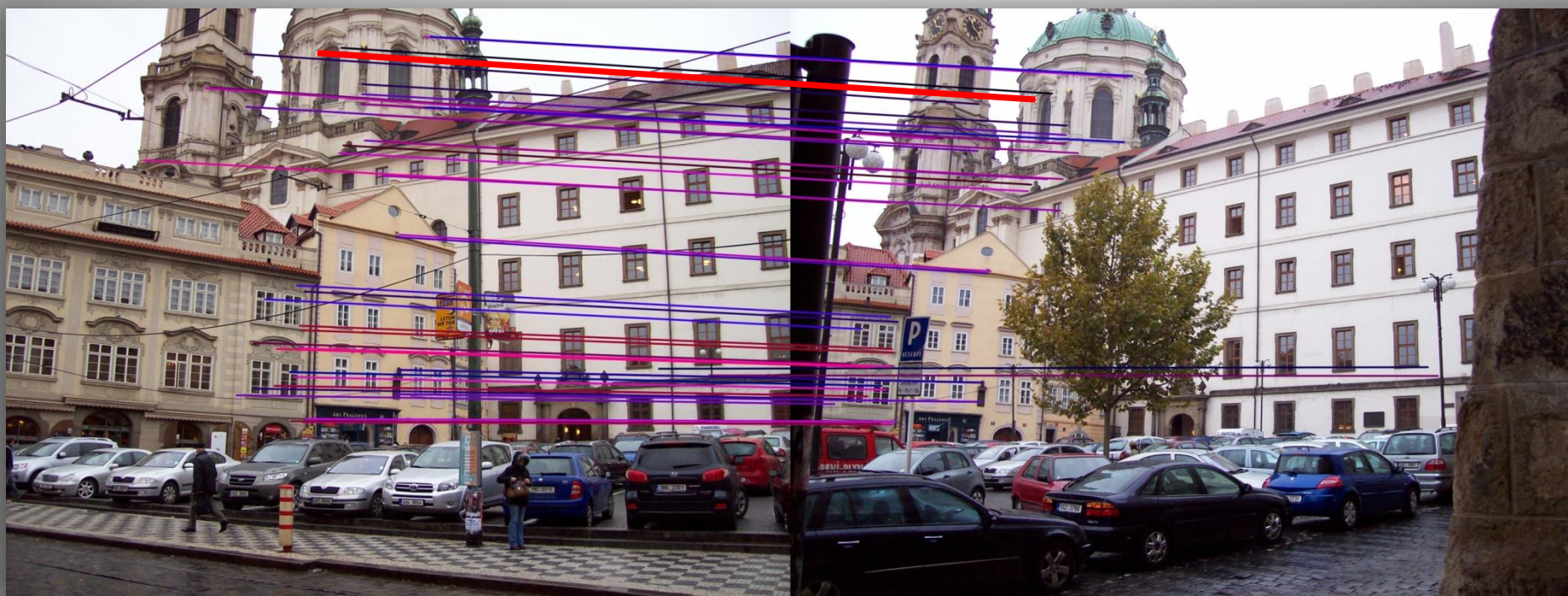
Přílohy (video, zdrojáky, ...) ke stažení na:

<http://mach.matfyz.cz/sift>

Korespondence mezi obrázky



Korespondence mezi obrázky



K čemu je to dobré?

Motivace na začátek

Panorama stitching (a)



Panorama stitching (b)

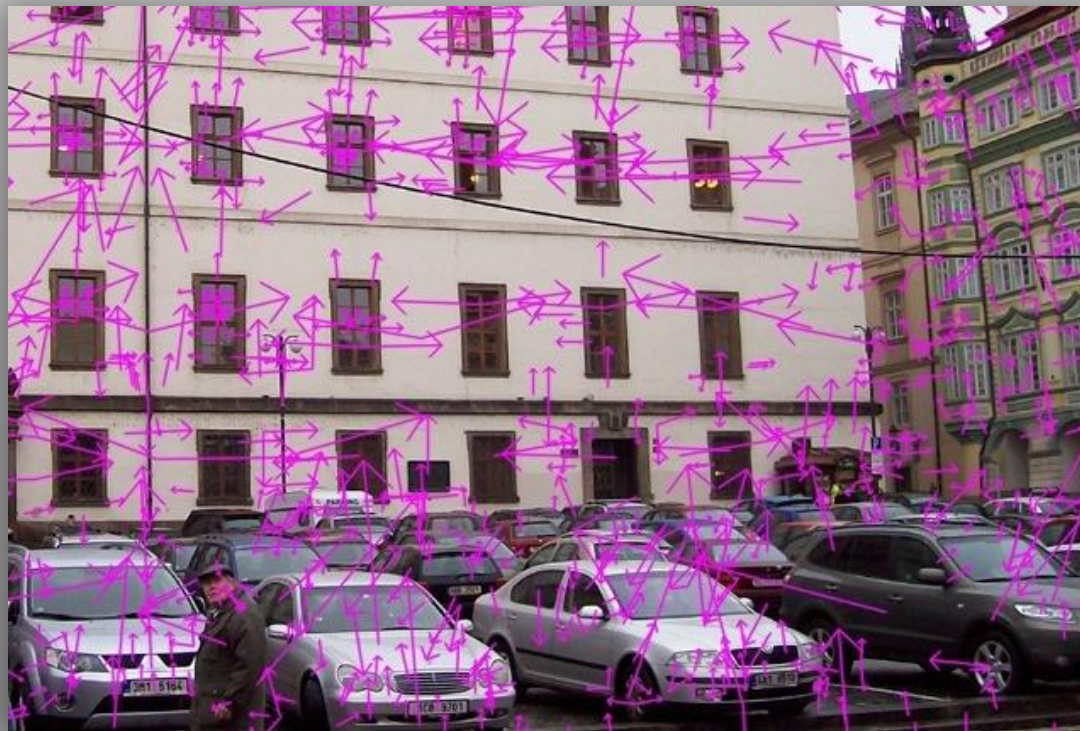


Automatická rekonstrukce 3D scény

- Ukázka

Jak najít korespondence?

- Nejprve na obrázku nalezneme „významná“ místa – **features** (body zájmu)



Deskriptory

- Každou feature popíšeme deskriptorem: **vektorem 128 integerů**
- Deskriptory jsou **invariantní** vůči následujícím operacím s obrazem:
 - zvětšení/zmenšení, posun, 2d rotace, šum, změna kontrastu, jasu, osvětlení
 - částečně invariantní vůči prostorové změně úhlu pohledu
- Stejně body na dalších obrázcích tedy budou mít stejné (nebo velmi podobné) deskriptory

Schéma algoritmu

1. Vybereme velké množství bodů – „kandidátů“
2. Z této množiny **odfiltrujeme nestabilní features** (nekontrastní, na hranách, ...)
3. Vypočítáme **deskriptory**
4. Vše provedeme v různých „měřítcích“ (**scale space**) původního obrázku a všechny features hodíme na jednu hromadu

Pozorování: vezmeme obrázek



Gaussovské rozostření poprvé



Gaussovské rozostření podruhé



Odečteme je od sebe



Odečteme je od sebe

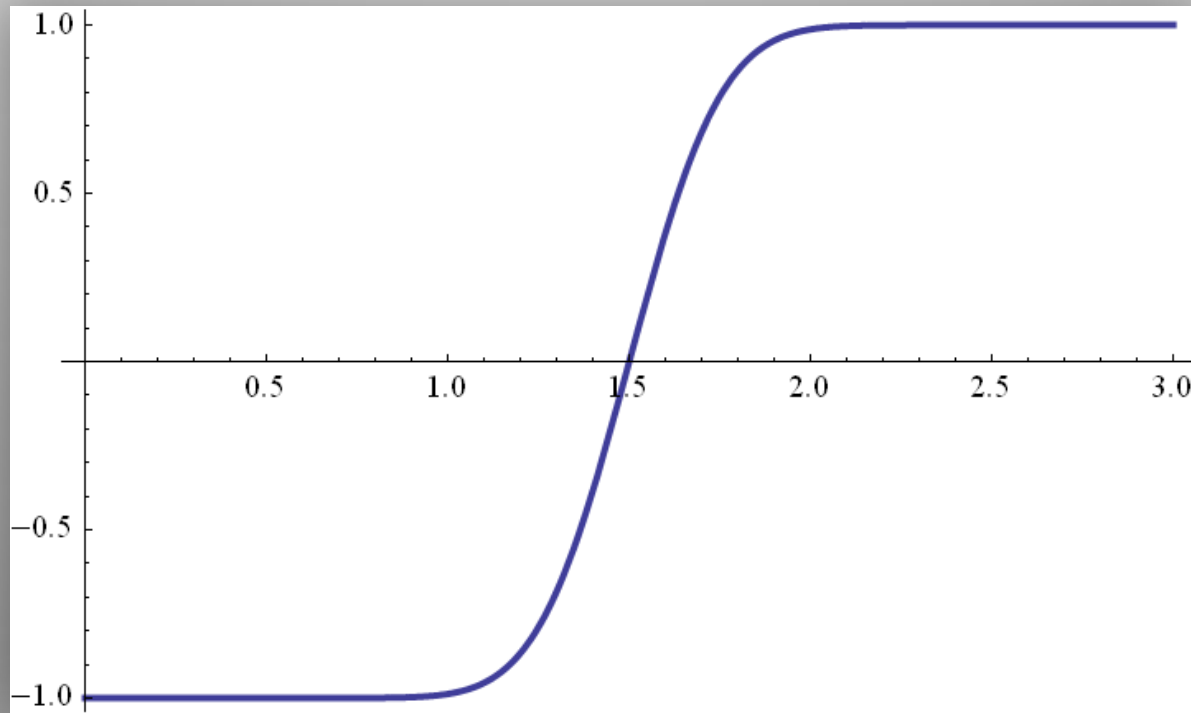


Co se stalo?

- Zajímat nás budou **lokální extrémy** v „rozdílovém“ obrázku (na předchozím obrázku je zobrazený v absolutní hodnotě)
- Černé (= nulový rozdíl) jsou **hrany** a **velké homogenní oblasti** – ty nás nezajímají
- Bílé (= velký rozdíl) jsou oblasti **vedle hran** a **středy malých kruhů**

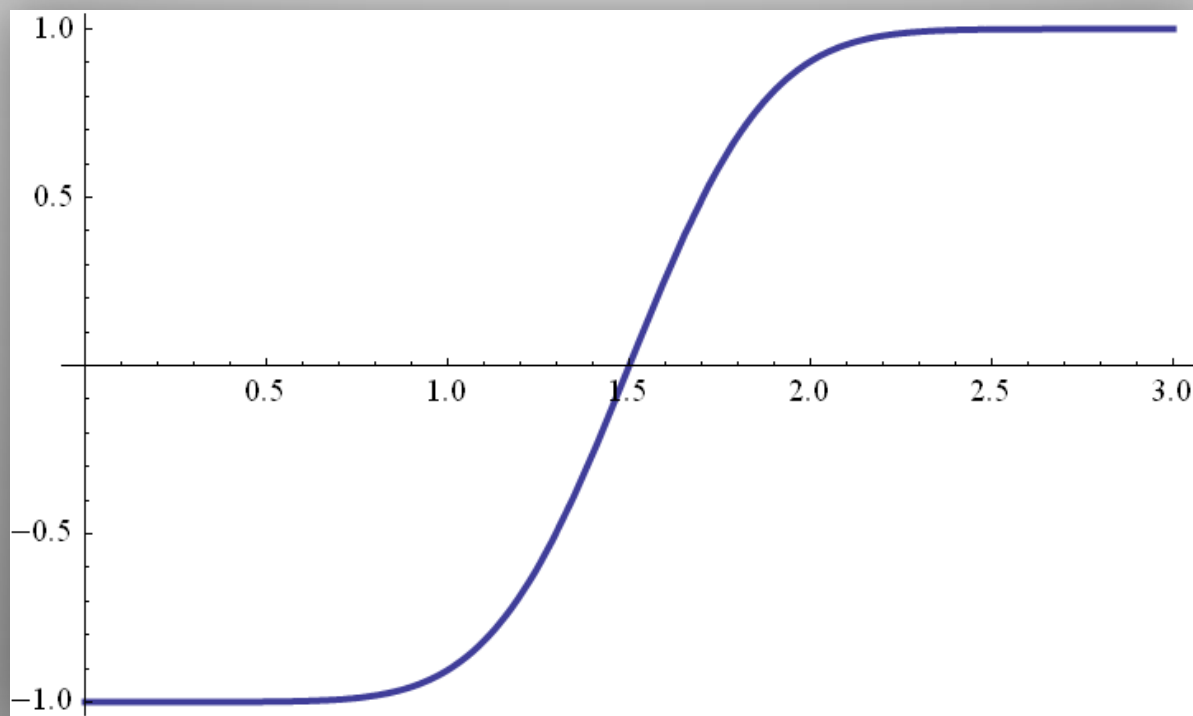
Proč se to stalo?

- Ukážeme si to na příkladu jednorozměrné funkce („1D obrázek“):



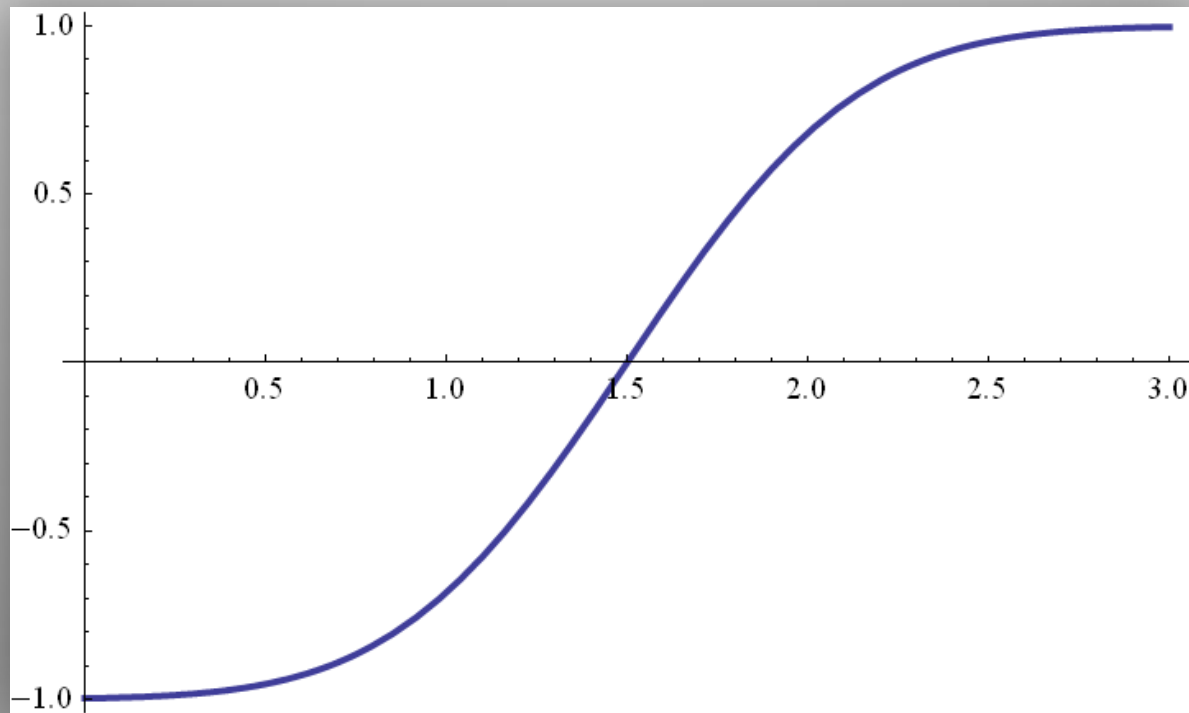
Pokračujeme s funkcí

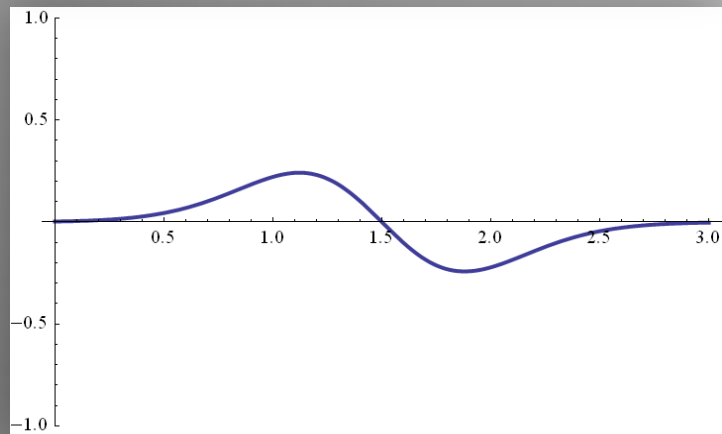
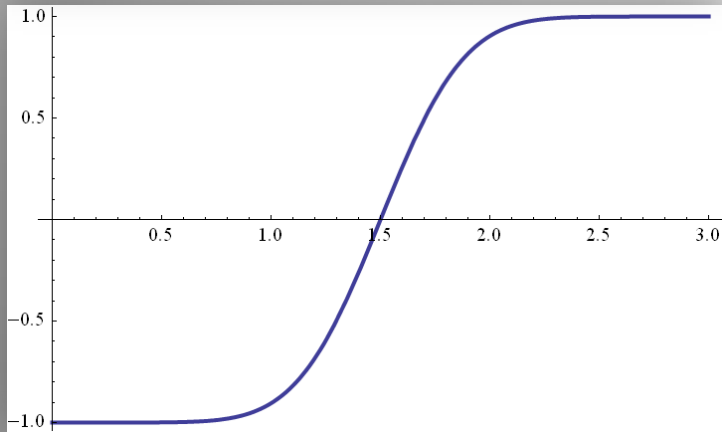
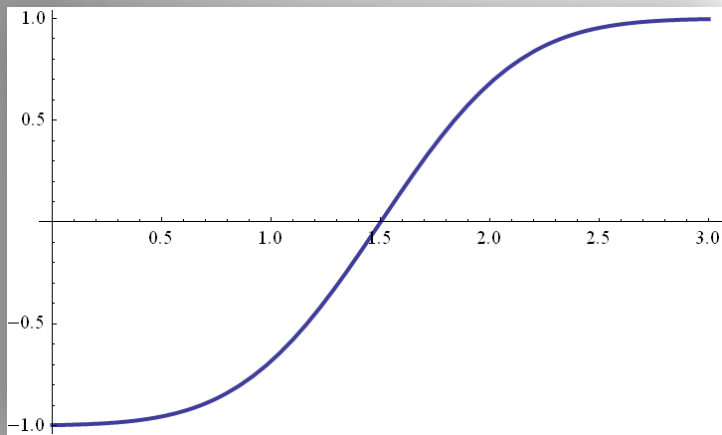
- Funkci rozostříme podobně jako jsme to provedli s obrázkem:



„Rozostření“ podruhé

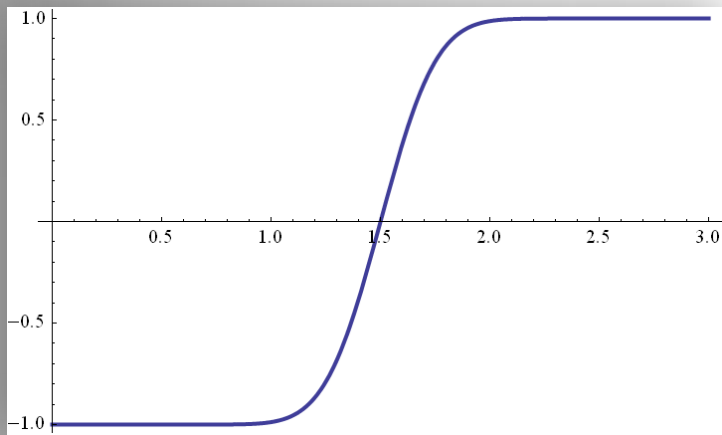
- Zopakujeme konvoluci:



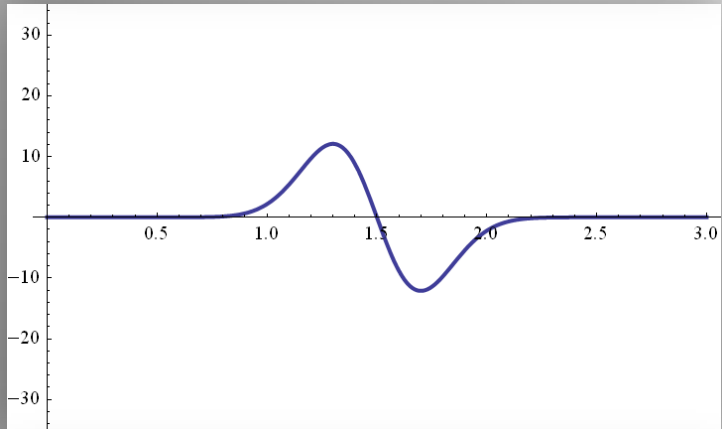


odečteme funkce od sebe

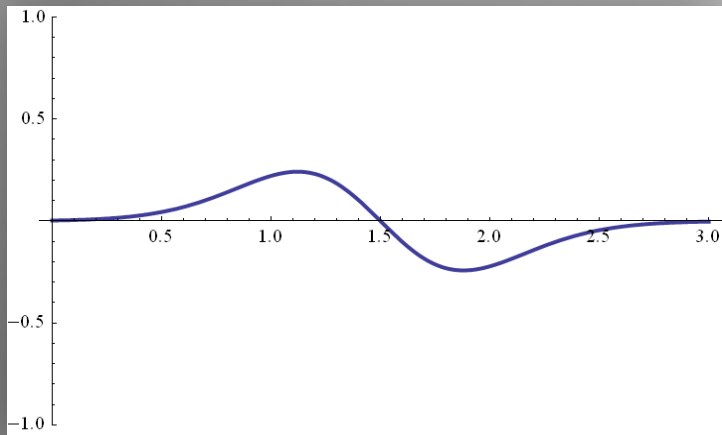
... výsledek



původní funkce



2. derivace této funkce



rozdíl z minulého slidu

Závěr pozorování

- Rozdíl dvou gaussovsky rozostřených obrázků je **aproximací druhé derivace** obrázku (přesněji Laplaciánu)
- Algoritmus na začátku vybere lokální extrémy v takovýchto „rozdílových obrázcích“

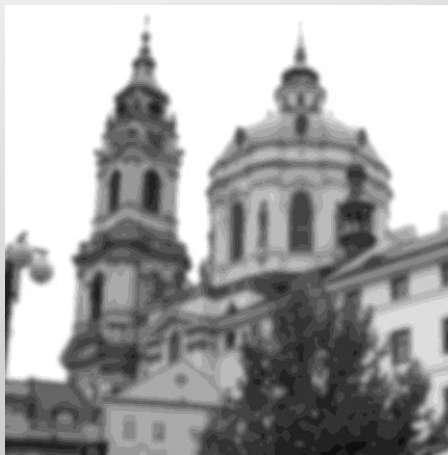
Sestavíme scale space



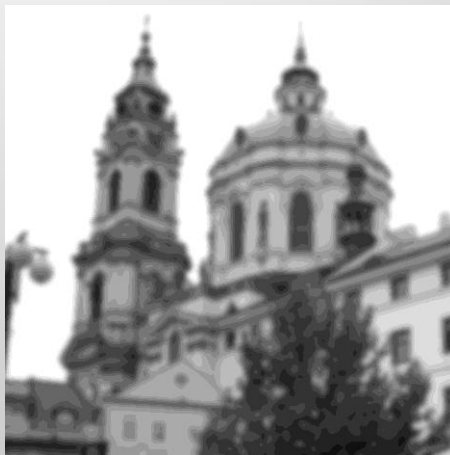
Sestavíme scale space



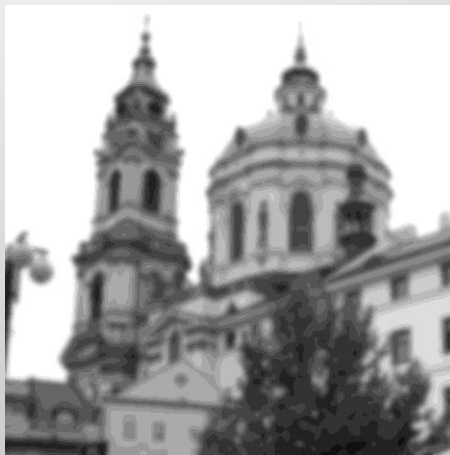
Sestavíme scale space



Sestavíme scale space



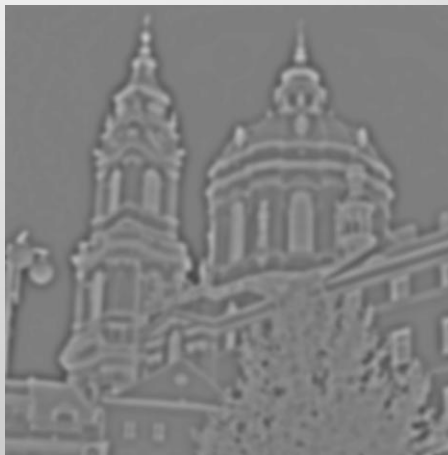
Sestavíme scale space



Sestavíme scale space



Odečteme sousední obrázky



Vybereme lokální extrémy

- V rozdílových obrázcích vybereme **lokální extrémy** – pixely s hodnotou větší (menší) než všichni sousedé



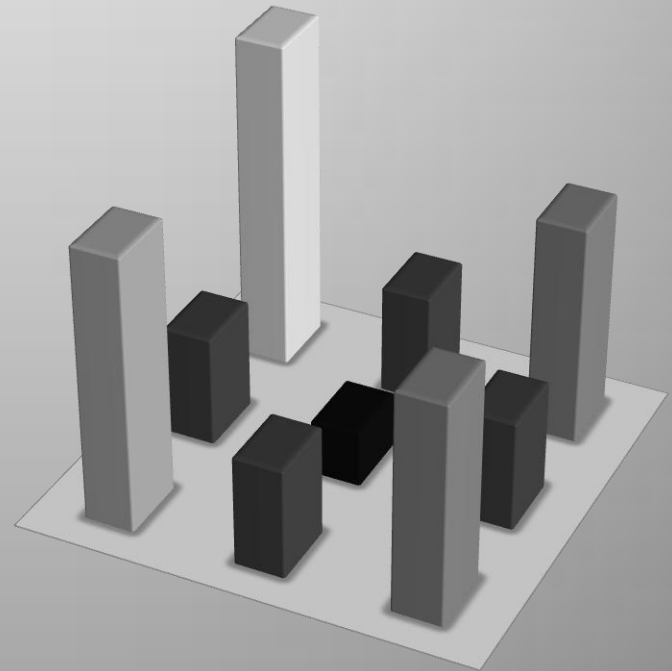
Porovnáváme ovšem i s
pixely na 3x3 okolí na
stejném místě v **sousedních
obrázcích ve scale space**

Ukázka: lokální extrémý ve scale space



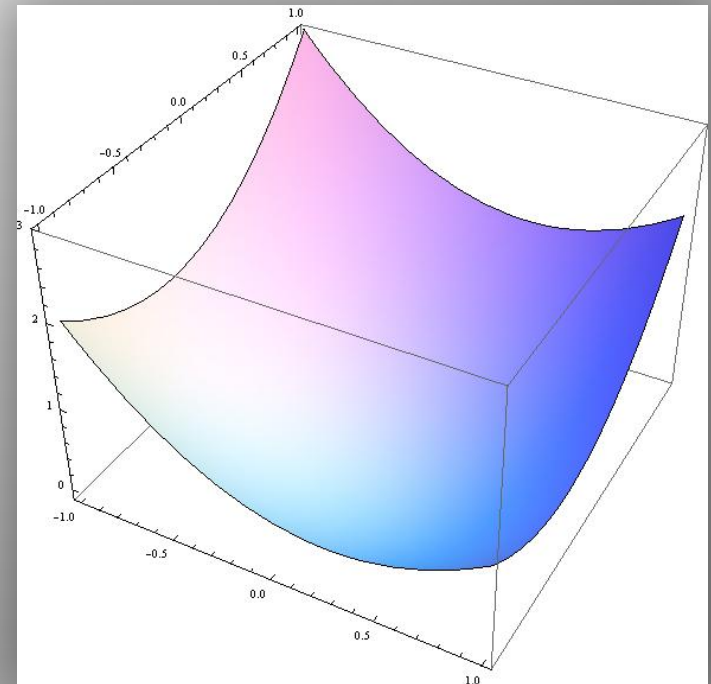
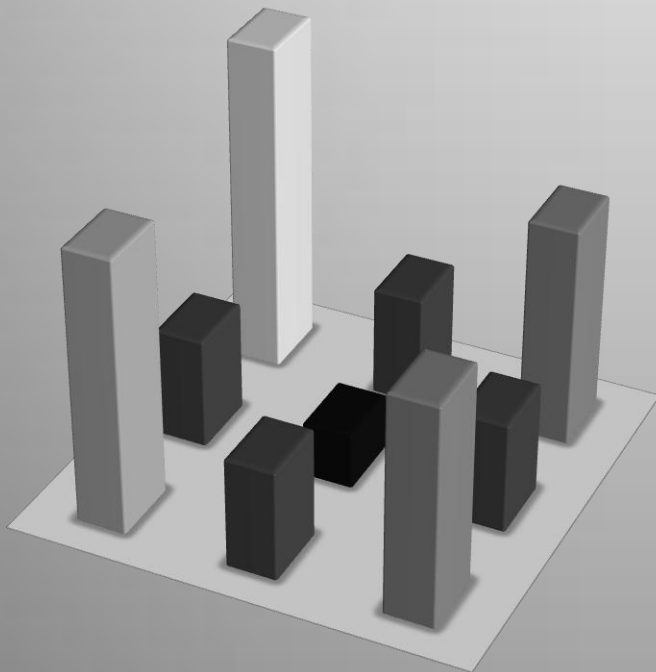
Proložení kvadratickou funkcí (a)

- Pro každý lokální extrém vezmeme jeho 3×3 okolí a představíme si ho jako funkci $\mathbf{R}^2 \rightarrow \mathbf{R}$



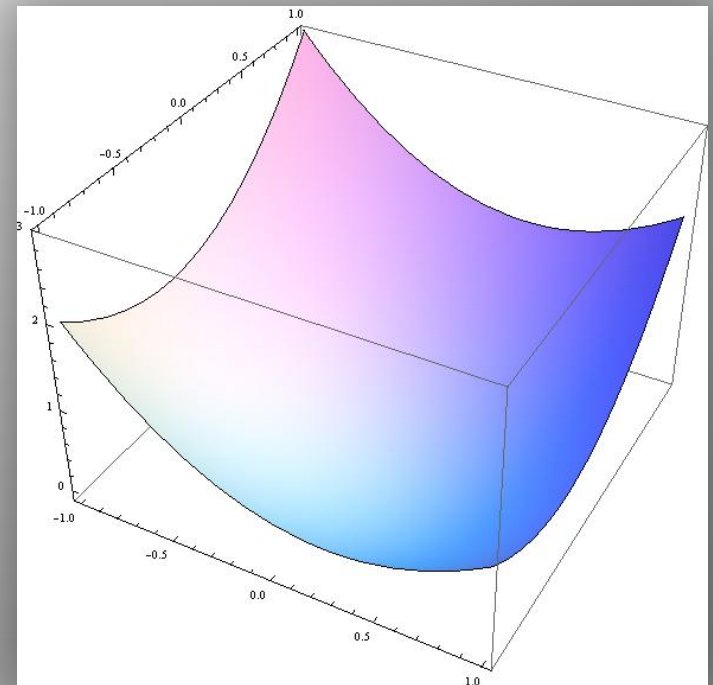
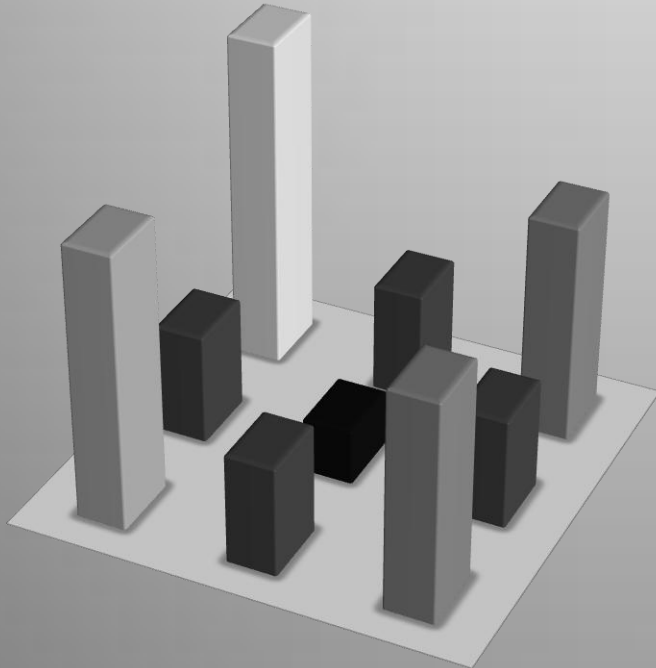
Proložení kvadratickou funkcí (b)

- Těmito hodnotami proložíme trojrozměrnou kvadratickou funkcí:



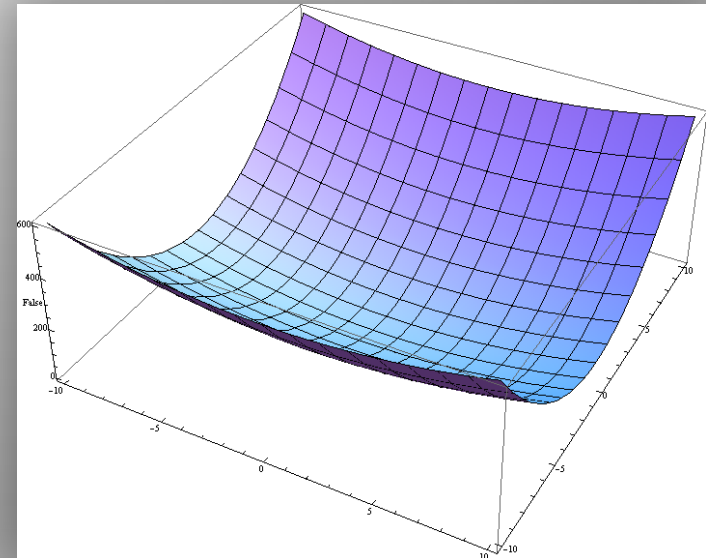
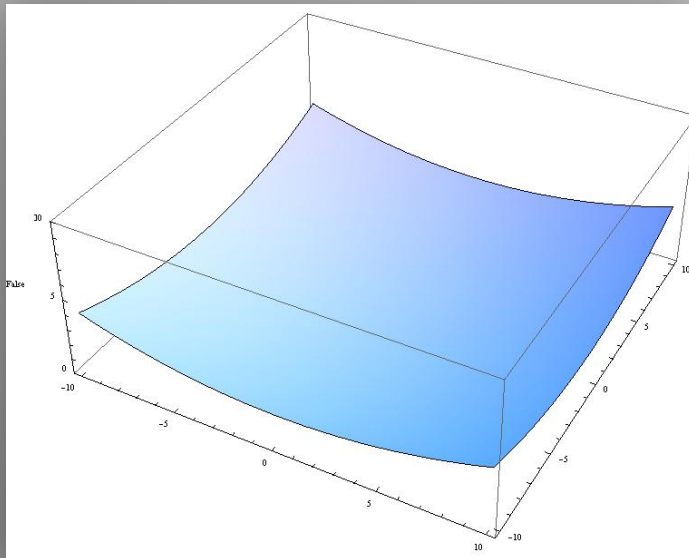
Subpixelová přesnost

- Najdeme minimum/maximum **této funkce** a tak zjistíme polohu feature se subpixelovou přesností



Filtrování features

- Podle tvaru kvadratické funkce odstraníme nestabilní features: **s nedostatečným kontrastem** nebo **ležící podél hran**



Znovu: lokální extrémý ve scale space



Odstranění nekontrastních features



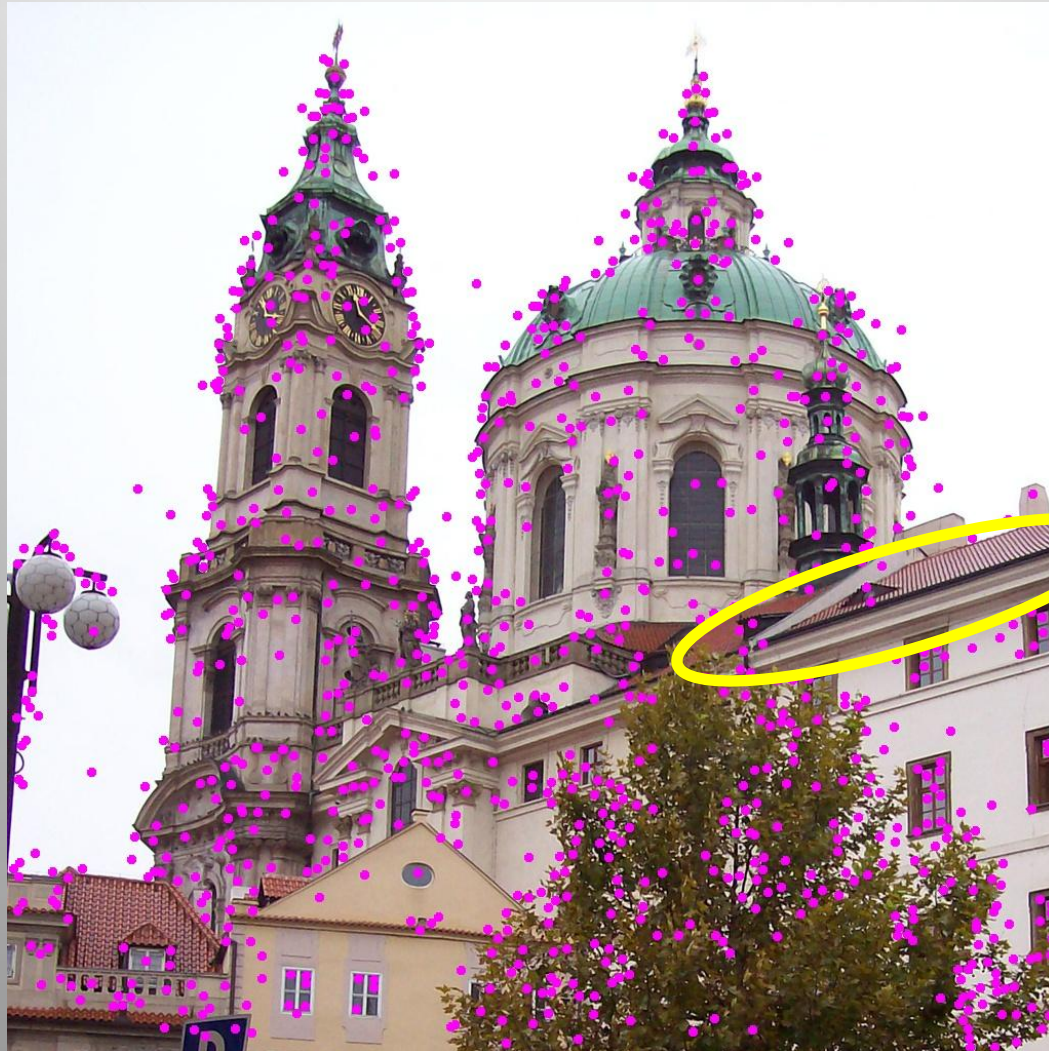
Odstranění features na hranách



Odstranění features na hranách



Odstranění features na hranách



Orientace a velikost

- Bodům zájmu přiřadíme orientaci a velikost
- V **rozostřeném obrázku** vypočítáme 1. derivaci v místě feature (ve směrech os x a y : d_x , d_y)
- Velikost: $L = \text{sqrt}(d_x^2 + d_y^2)$
- Orientace: $\alpha = \text{arctan}(d_y/d_x)$

Orientace a velikost

- Velikost feature: $L = \sqrt{d_x^2 + d_y^2}$
- Orientace feature: $\alpha = \arctan(d_y/d_x)$



Deskriptor

- Chceme vhodně popsat okolí feature (o velikosti **L**)
- Výpočet deskriptoru budeme provádět **relativně** vzhledem k orientaci feature (**α**)
- Tím dosáhneme **invariance vzhledem k rotaci** obrázku

Deskriptor

- Deskriptor je typicky 128-rozměrný vektor vypočítaný na základě **gradientů v okolí** feature
- To vychází z biologického modelu rozpoznávání objektů

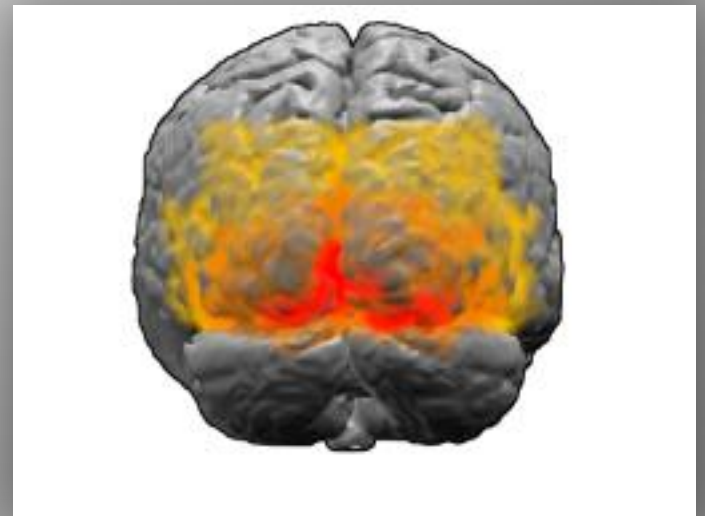
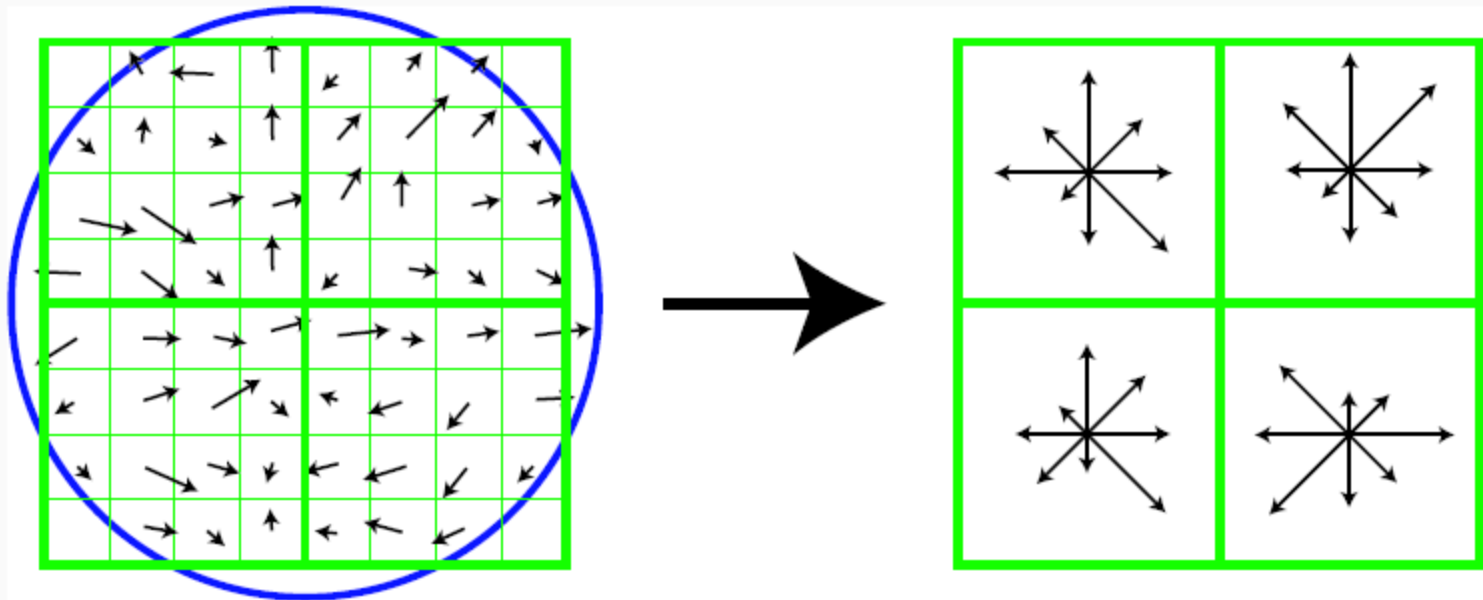


Image licensed under GNU FDL

Deskriptor

- V okolí feature vypočítáme gradienty
- Deskriptor je histogramem těchto gradientů



Feature matching

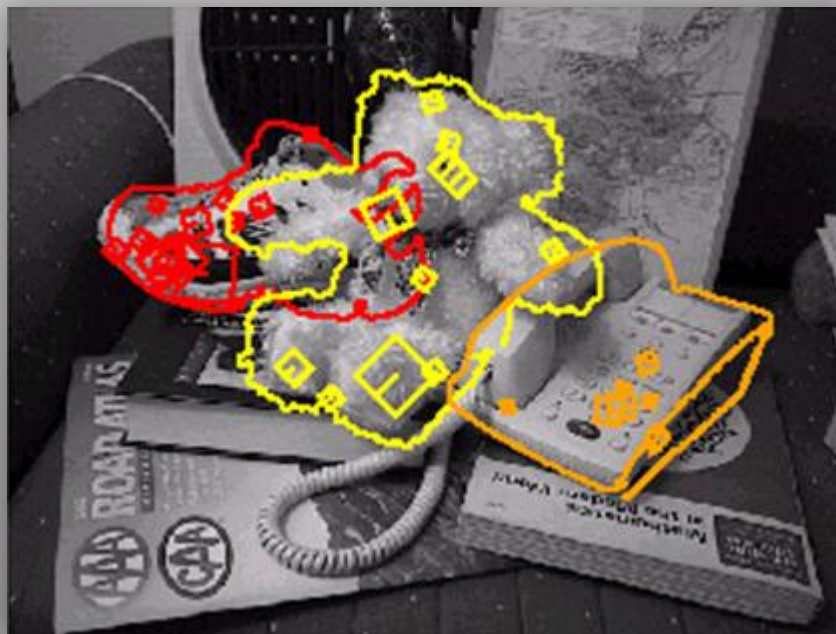
- Každý bod zájmu máme popsáný vektorem
- Na druhém obrázku najdeme bod zájmu, který má tento vektor co **nejpodobnější**
- Podobnost vektorů: použijeme běžnou eukleidovskou metriku
- Feature space outlier rejection
- **Zrychlení: PDS, kd-strom**

Další aplikace

Zdrojáky využívají knihovnu **OpenCV**
a opensource implementaci SIFTů
(viz odkazy na literaturu).

Rozpoznávání objektů

- Hledání korespondencí oproti celé množině fotografií (databáze)



matching.c, pano.c



Kalibrace kamer

- **Kalibrace snímků** videa či fotografií
- U videa ale fungují i primitivnější metody
- Přidávání renderovaných objektů do videa, modelování podle fotografií
- **Nicméně:** SIFT není zcela afině invariantní, hledání korespondencí mezi fotografiemi s velmi odlišnými úhly pohledu je **stále problém**

Literatura

- **Hlavní článek:**

- David G. Lowe: „**Distinctive image features from scale-invariant keypoints**,“ International Journal of Computer Vision, 60, 2 (2004), pp. 91-110
- <http://www.cs.ubc.ca/~lowe/keypoints/>

Další informace

- Filtrování chybných korespondencí (RANSAC, guided matching) a mnoho dalšího:
 - R. I. Hartley, A. Zisserman, „**Multiple View Geometry in Computer Vision, 2nd Edition,**“ Cambridge University Pr., 2004.
- Automatická rekonstrukce 3D scén pomocí SIFTů:
 - Brown, Lowe, „**Unsupervised 3D Object Recognition and Reconstruction in Unordered Datasets**“
 - <http://research.microsoft.com/~brown/papers/3dim05.pdf>
- Jiný algoritmus (stabilnější při změně polohy pozorovatele):
 - J. Matas, O. Chum, M. Urban, T. Pajdla, „**Robust Wide Baseline Stereo from Maximally Stable Extremal Regions,**“ British Machine Vision Conference, 2002.

Odkazy

- Open source implementace:
 - Seznam na:
[http://people.csail.mit.edu/albert/ladypack/wiki/index.php/Known implementations of SIFT](http://people.csail.mit.edu/albert/ladypack/wiki/index.php/Known_implementations_of_SIFT)
 - Zejména:
<http://web.engr.oregonstate.edu/~hess/index.html>
- Komerční software (kalibrace kamer):
 - <http://www.realviz.com/>
 - <http://www.2d3.com/>

Děkuji za pozornost



[Licensed under Creative Commons BY 3.0](https://creativecommons.org/licenses/by/3.0/)