# 5. Detektory "bodů" a oblastí zájmu. Algoritmy detekce a jejich citlivosti na geometrické a fotometrické změny v obraze. Popis oblasti zájmů: metoda lokálních rámců pro zajištění geometrické invariance popisu, deskriptor SIFT (scale invariant feature transform)

# Detektory

# Hessián

Úkolem detektoru je opakovaně najít v obraze **body** nebo **oblasti**, jež jsou dobře lokalizovány a v jejichž **okolí** je dostatek **informace**, aby bylo možné najít pokud možno **jednoznačný** **popis** jeho **okolí**. Detektor lokálních extrémů Hessiánu – determinantu Hessovy matice (matice druhých derivací).

ce37bed45c1a4b5ce414206b4a1d435d.png

hledá centra tzv. blobů, lokálních extrémů intenzitní funkce, které jsou dobře lokalizovány v okolí daném variancí 51131dad50ced65f5818c4c2275a614b.png. Body zájmu jsou pak ty lokální **maxima**, jejichž hodnota je nad určitou **hranicí** danou úrovní šumu v obrázku.

Pro nalezení extrémů Hessiánu je potřeba funkce, která zjistí, zda je bod lokální extrém, toto zjistíme srovnáním hodnoty bodu s jeho osmi **okolím**.

**Výpočet v matlabu:**

|  |  |
| --- | --- |
| function [response]=**hessian\_response**(img,sigma)  smooth = gaussfilter(img,sigma);  dx = conv2(1,[-1 0 1]/2, smooth, 'same');  dxx = conv2(1,[1 -2 1], smooth, 'same');  dxy = conv2([1 0 -1]/2, 1, dx, 'same');  **response** = dxx.\*dyy-dxy.^2; | Detereminant hessianu je ta odezva (response), čím vyšší odezva, tím zajímavější bod. |

# Harris

Algoritmus hledá **rohové** **body (corner points)**, které mají velkou **změnu** **intenzity** jasu (**gradient**) ve všech směrech (x, y) = hledá body v nichž se **gradient** mění ve dvou ortogonálních směrech. K jejich nalezení se používá tzv. **autokorelační** matice C.

autocorelation_matrix.png kde \* je operátor **konvoluce**.

Prostřednictvím okenní funkce G() se akumulují matice vnějších součinů gradientu v oi okolí bodu. Z této matice se poté vypočítá **Harrisova** **odezva** R.

r.png Fce **trace**() sečte prvky na diagonále, alfa je parametr.

**Postup:**

* v každém bodě [x, y] se spočítá derivace podle osy x a podle osy y,
* z toho se sestaví matice C pro konkrétní bod [x, y],
* vypočte se R na tom konkrétním bodě [x, y],
* a jestli je R dostatečně velké, tak tam bude roh

**Parametry:**

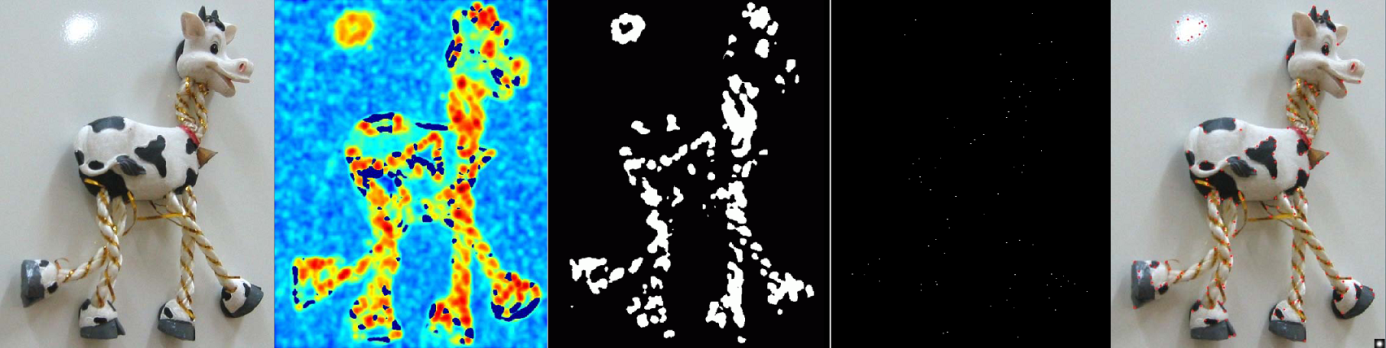
* x, y - souřadnice bodu
* σi - parametr pro rozmazání gaussem (nepřímá úměra)
* σd - parametr pro derivaci (nepřímá úměra)
* threshold - jaká musí být odezva, aby byl bod prohlášen za Harrisův (nepřímá úměra)

**Invariantní:**

* Invariantní k rotaci(geom.)
* Není Invariantní ke změně měřítka(Image scale)(geom.)

*Invariance (česky neměnnost) je označení pro situaci, v níž jsou jisté objekty neměnné při určitých událostech.*

**Z přednášky:**



Harris Workflow: (a) - Original picture, (b) - Compute corner response R, (c) - Find points with large corner response: R > threshold, (d) - Take only the points of local maxima of R, (e) - Finded points

**Popis Algoritmu**

Aby byl Harrisův detektor affině-invariantní znamená, že by měl být invariatní vůči **rotaci** a neuniformnímu (nerovnoměrnému) **zvětšení**. Vůči rotaci, jak známo, Harrisův D. invariatní je.

1. Rozpoznat oblasti pomocí harrisova detektoru a vybrat měřítko.
2. Odhadnout tvar pomocí matice druhých momentů gradientu.
3. Normalizovat oblast do kruhové.
4. Pokud vlastní čísla matice druhých momentů gradientu nejsou stejné, pak jdi na krok 2.

**Poznámka**:

|  |  |
| --- | --- |
| Scale invariance, když použiješ **malé** **okolí**, nemusíš zjistit, že v daném místě je rohový bod. Takže je potřeba zvolit správné **měřítko** a to se dělá tak, že se vypočte harris v různých měřítkách a dostaneš harrisovu odezvu v závislosti na velikosti měřítka a vybereš z toho měřítko, pro které je harrisova odezva **maximální**.  U malého kruhu neuvidíš ten roh a u velkého kruhu ten roh uvidíš, takže přes všechny tyhle měřítka vybereš takové měřítko, kde odezva je největší. |  |

# MSER

Je detektor **Maximalně Stabilních Extremálních Oblastí** by Matas.

Při detekci se sleduje **nárůst oblastí** v prahovém obraze při **zvyšování prahu intenzity**. Se stoupající intenzitou se vkládají do obrazu pixely. Spojují se do **oblastí**, až při maximální intenzitě zůstane **jediná** oblast. Po čase růstu se sledují statistiky jednotlivých oblastí (plocha a délka hranice) a **hledají se rozsahy intenzity, kde se poměr velikosti oblasti ku délce hranice mění nejméně**.

V obrázku postupně zvyšujeme threshold a tím propouštíme pouze pixely **určitého** **jasu**. Při **thresholdu** 0 je celý obrázek **bílý**, se zvyšováním **thresholdu** se objevují **tmavé** body (*lokální minima*). Při určité hodnotě thresholdu se lokální minima **spojí** a **uzavřou** oblast. S dalším zvyšováním thresholdu obrázek tmavne, až je černý.

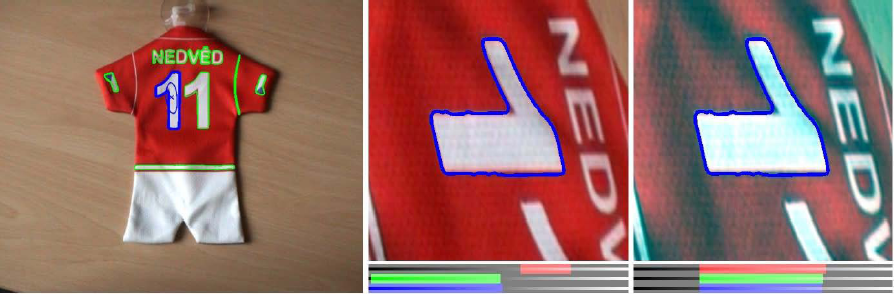
Množina všech spojených oblastí se nazývá množina **maximálních** **regionů**. Množinu **minimálních** **regionů** získáme **inverzí** **jasu** v obrázku.

**Implementace:**

* Pomocí binsortu se setřídí pixely podle jasu a v obrázku se označí svým pořadovým číslem - složitost N.
* Spustí se **union-find** algoritmus pro nalezení oblastí.
* Celková složitost: N\*log(log n)

**Vlastnosti:**

* invariantní vůči **afinní transformaci (rotace, scale a zkosení)**
* kovariantní s průběžnou deformací obrazu
* stabilita - neboť používá jen extremální oblasti, jejichž podpora je prakticky beze změn při rozsahu vybraných prahových hodnot
* multi-scale detection - vzhledem k tomu že není žádné vyhlazování, tak detekujeme jak velmi jemné tak velmi rozlehlé struktury



# Popis oblasti zájmu

# Lokální rámce

Pro získaní lokálního popisu invariantního k geometrickým a fotometrickým transformacím scény, potřebujeme okolí získaných záchytných bodů znormalizovat tak, aby jsme odstranili efekt geometrické resp. fotometrické transformace intenzity. Po odstranění těchto deformací pak můžeme na normalizovaném okolí počítat popis, jenž je nezávislý na těchto deformacích.

**Geometrická normalizace**

Geometrickou normalizací nazýváme proces **geometrické** **transformace** **okolí** bodu v obrázku do kanonického souřadného systému (od 0 do 1). Informace o geometrické transformaci okolí budeme uchovávat ve formě tzv. **rámce** – projekce kanonického souřadného systému do okolí záchytného bodu nebo oblasti v obrázku. Rámec reprezentujeme 3 x 3 transformací A. Transformaci A využijeme k získání výřezu, malého okolí záchytného bodu v kanonickém souřadném systému. Všechny další měření na tomhle typicky čtvercovém výřezu původního obrázku jsou teď již invariantní k odpovídající geometrické transformaci scény.

**Orientace dominantního gradientu**

Na částečně normalizovaném výřezu se v každém bodě odhadne **velikost** a **orientace** gradientu a spočítá se **histogram** gradientů. Protože předpokládáme náhodné natočení výřezu je potřeba gradienty akumulovat jenom za **kruhové** **okolí** např. pomocí váhování okenní funkcí, nebo podmínkou (x-stredx)^2+(y-stredy)^2 < (ps/2)^2, kde ps je velikost hrany výřezu. Jinak by orientaci ovlivnil obsah “rohů” výřezu, jenž ale může být při jiném natočení jiný. Gradienty jsou **váhovány** jejich **velikostí**, **větší** gradient znamená **vyšší** **příspěvek** do odpovídajícího **binu** **histogramu**. Pro zvýšení robustnosti se hlas může rozdělit lineární interpolací do sousedních binů. Na závěr se histogram **vyfiltruje** 1D Gaussiánem a najde lokální **maximum**. Pro přesnou lokalizaci je možné do okolí maxima fitnout parabolu a najít orientaci s přesností větší než 360/počet binů stupňů.

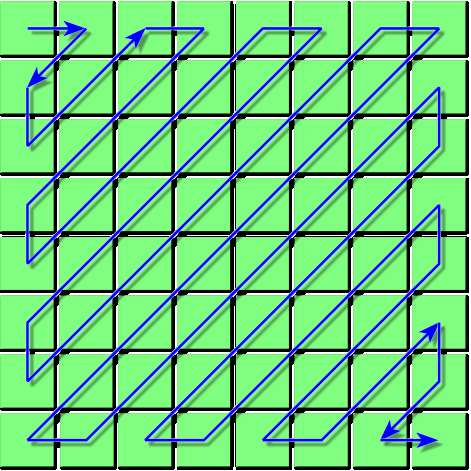
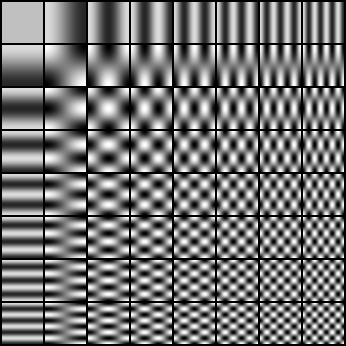
**Fotometrická normalizace**

Fotometrická normalizace má za cíl **potlačit** změny scény způsobeny **změnou** **osvětlení**. Nejjednodušší normalizace je **roztáhnutí** **jasového** **kanálu** tak, aby se využil celý **rozsah** **intenzit**. Další možnost je roztáhnutí jasového kanálu tak, aby po transformaci **střední** **hodnota** odpovídala **půlce** **intenzitního** rozsahu a variance intenzit jeho rozsahu (případně 2x variance). V případě, že chceme využít všechny **tři barevné kanály**, normalizujeme každou složku **zvlášť**.

**Výpočet popisu**

Na **geometricky** a **fotometricky** invariantním **výřezu** obrázku z předchozího kroku můžeme spočítat libovolný nízko-dimenzionální **popis**. Je zřejmé, že proces normalizace není úplně bezchybný a proto je žádoucí, aby výpočet popisu nebyl příliš citlivý k zbývajícím nepřesnostem. Nejjednodušší popis je samotný výřez obrázku, mezi jeho nevýhody patří hlavně vysoká dimenze popisu, citlivost k malým posunům a lokálním změnám intenzity. Vyzkoušíme si proto kromě tohoto popisu lepší varianty.

1. Popis využívající **dvourozměrnou diskrétní kosínovu transformaci** (DCT). Koeficienty této integrální transformace vzniknou skalárním součinem výřezu obrázku s tzv. DCT **bázovými funkcemi** viz obrázek:



Výřez obrázku je takto rozložen na jednotlivé “frekvence” podobně jako u **JPEG komprese**. Podobně jako u komprese obrázků vybereme podmnožinu těchto koeficientů obsahující informaci o nízkých frekvencích. Tím dosáhneme odolnost popisu k vysokofrekvenčnímu šumu a do jisté míry i k nepřesnostem v geometrické normalizaci. Z obrázku bázových funkcí vidíme, že výběr DCT koeficientů budeme provádět v tzv. **zig-zag** pořadí, kdy budeme přidávat **postupně vyšší a vyšší frekvence**.

1. **RGB** histogram, **RG** histogram, histogram **gradientů** jsou metody využívající **globálních vlastnosti výřezu** obrázku. Barevné kanály obrázku (např. R, G, B) se nejdříve zpracují a vypočtou se charakteristiky, jenž se následně kvantizují do binů a spočítají. Výsledný popis je tvořen lineárním vektorem, počtem v jednotlivých binech. RG projekce nebo tzv. **chromacity** (dvou dimenziální RGB prostor) barevný prostor má dvě složky vyjadřující podíl červené a zelené složky v obraze. Výhodou histogramů výřezu je necitlivost k **nepřesnostem geometrické normalizace**, nevýhodou je nižší rozlišovací schopnost tzv. distinktivita.

**Poznámka**: K čemu vlastně ten popis je? Popis ti říká co je v tom rámci (výřezu), můžeš samozřejmě vzít jasové složky naházet do vektoru a máš nějaký popis, ale nebude invariantní např. vůči rotaci a podobně, navíc takový popis je nevhodný protože když se ti to o 1px posune, tak dostaneš úplně jiný vektor, ale obrázek v rámci je defakto stejný, proto se počítá takový popis, aby byl vůči tomuto odolný.

# SIFT (Scale-invariant Feature Transform)

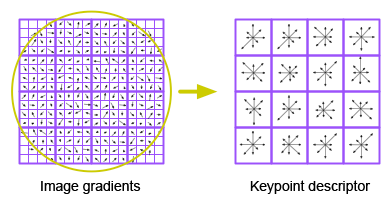
Algoritmus pro **detekci a popis** (*detect and describe*) „**bodů**“ **zájmu**. Díky svým dobrým vlastnostem je **nejčastěji používaným algoritmem pro nalezení lokálních rysů v obraze**.

**Vlastnosti:**

* robustní (i k částečné okluzi)
* invariantní ke změně **měřítka**, **orientace** a **affinímu** **zkreslení**
* částečně invariantní ke změnám v **intenzitě** **osvětlení**
* rychlý (real-time)
* kompaktní

**Popis Algoritmu**

1. **Scale-space extrema detection** V této fázi se prohledává celý **prostor měřítek**. Efektivně se využívá funkce DoG (Difference of Gaussian) pro nalezení **potenciálně zajímavých bodů** zájmu, které jsou invariantní ke změně **měřítka** a **orientace**.
2. **Keypoint Localization** Na pozici **kandidátů** je proložen detailní model pro zjištění **pozice** a **měřítka**. Klíčové body jsou vybírány na základě **míry** jejich **stability**.
3. **Orientation assignment** Ke každému klíčovému bodu je přiřazena **jedna** nebo **více** **orientací** podle lokálních **směrů** **gradientu**. Všechny **další operace** jsou prováděny na obraze transformovaném relativně k přiřazené **orientaci**, **měřítku** a **pozici** každého rysu. Tento krok zaručuje invarianci k těmto transformacím.
4. **Keypoint descriptor** Výřez se rozdělí do **pravidelné** mřížky (nejčastěji 4×4) a v každém čtverci se spočítá **histogram** **gradientů** vážený jejich velikostí pro **8 orientací**. Pro zvýšení robustnosti k nepřesnosti geometrické normalizace se velikosti gradientů **váhují** **okenní** **funkcí**. Dále je každý čtverec mřížky **zvětšen** tak, aby zasahoval do půlky hrany svých sousedů. Hlasování do binů orientací se provádí lineární interpolací do dvou sousedních binů.



SIFT deskriptor je tedy mřížka 4×4 histogramů orientací (kvantizovaných na 8 binů). Jde tedy o 128 prvkový vektor.

|  |  |  |
| --- | --- | --- |
|  |  |  |

After scale space extrema are detected (their location being shown in the uppermost image) the SIFT algorithm discards low contrast keypoints (remaining points are shown in the middle image) and then filters out those located on edges. Resulting set of keypoints is shown on last image.