VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

# DETEKCE A SLEDOVÁNÍ OBJEKTŮ POMOCÍ VÝZNAČNÝCH BODŮ

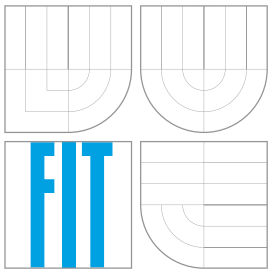
DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. VOJTĚCH BÍLÝ

BRNO 2012

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

# DETEKCE A SLEDOVÁNÍ OBJEKTŮ POMOCÍ VÝZNAČNÝCH BODŮ

OBJECT DETECTION AND TRACKING USING INTEREST POINTS

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE — Bc. VOJTĚCH BÍLÝ
AUTHOR

VEDOUCÍ PRÁCE — Ing. ROMAN JURÁNEK
SUPERVISOR

BRNO 2012

# Abstrakt

Tato práce se zabývá detekcí a sledováním objektů pomocí význačných bodů. Jsou zde popsány existující přístupy k této problematice. Je zde navržená inovovaná metoda detekce objektů založená na Obecné Houghově transformaci a iterativním prohledáváním Houghova prostoru. Na nejrůznějších typech objektu je demonstrována univerzálnost navrženého detektoru. Sledování objektů je řešeno detekcí objektu snímek po snímku.



# Abstract

This paper deals with object detection and tracking using iterest points. Existing approaches are described here. Inovated method based on Generalized Hough transform and iterative Hough-space searching is proposed in this paper. Generality of proposed detector is shown in various types of objects. Object tracking is designed as frame by frame detection.



# Klíčová slova

Detekce objektu, Sledování objektu, Význačný bod, Deskriptor, SIFT, SURF, Shlukování, K-Means, Kalmanův filtr, Mean-Shift sledovaní, Obecná Houghova transformace, Implicit Shape Model, Iterativní prohledávání Houghova prostoru.



# Keywords

Object detection, Object tracking, Interest point, Descriptor, SIFT, SURF, K-Means, Clustering, Kalman filtr, Mean-Shift tracking, Generalized Hough transform, Implicit Shape Model, Iterative Hough-space searching.



# Citace

Vojtěch Bílý: Detekce a sledování objektů pomocí
význačných bodů, diplomová práce, Brno, FIT VUT v Brně, 2012

# Detekce a sledování objektů pomocí význačných bodů

## Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením pana Ing. Juránka

.....
Vojtěch Bílý
22. května 2012

# Obsah


**1 Úvod** **2**

**2 Detekce objektů** **3**
2.1 Význačné body . . . 3
2.2 Deskriptory . . . 10
2.3 Shlukování . . . 12
2.4 Lokalizace objektu . . . 13

**3 Sledování objektů** **17**
3.1 Sledování bodů . . . 18
3.2 Kernel tracking . . . 19
3.3 Sledování obrysů . . . 20

**4 Detekce objektů iterativním hlasováním** **21**
4.1 Tvorba slovníku . . . 21
4.2 Detekce objektů . . . 27
4.3 Sledování objektů . . . 32

**5 Experimenty** **33**
5.1 Implementace . . . 33
5.2 Datové sady . . . 33
5.3 Nastavení parametrů . . . 38
5.4 Výslekdy detekce . . . 46
5.5 Sledování objektů . . . 54

**6 Závěr** **55**

**A Plakát** **61**

# Kapitola 1

# Úvod

Detekce objektů a jejich následné sledování je rozšířeno do mnoha oblastí lidské činnosti. Velký význam na tomto rozšíření je možnost real-time zpracování na běžně dostupných zařízeních jako jsou mobilní telefony, herní konzole či stolní počítače. Velký důraz je přitom kladen na automatizaci detekce a sledování.

Oblastí, kde došlo k posunu od ukázkových projektů k praktickým řešením, je rozšířená realita. Stalo se tak díky rozšíření chytrých mobilních telefonů, které za pomoci kamery, GPS navigace a digitálního kompasu dokáží zobrazovat dodatečné informace o objektech v okolí [19]. Aplikace rozšířené reality také využívají QR kódů, a to jak pro přímé zobrazení popisu objektu, tak i přechod na webovou stánku s informacemi[1]. Také ovládání počítače pomocí gest našlo praktické uplatnění v rozšířeních herních konzolí např. Eye pro PlayStation3 (pouze detekce ovladače) nebo Kinect pro Xbox360 (detekce a rozpoznání jednotlivých lidí, ovládání pomocí celého těla)[2]. V moderních automobilech jsou zabudované systémy pro zjišťování vzdálenosti při couvání nebo zcela automatické systémy pro podélné parkování[3]. Armáda využívá detekci objektů k navádění řízených střel, na letištních kontrolách se detekují nepovolené předměty, v medicíně se detekují mikroorganizmy ze snímků elektronických mikroskopů či nádory na CT snímcích.

Tato práce představuje inovovanou metodu detekce objektů. Metoda se má 2 fáze. První je vytvoření slovníku z anotované sady dat obsahující objekty stejného typu jako hledaný objekt. Slovník se skládá ze seznamu typický částí objektu, pozic jejich výskytu na objektu a jejich věrohodnosti. Druhá fáze slouží k detekci objektů. Na obrázku se pomocí slovníku naleznou možné součásti objektu a tyto součásti hlasují pro střed objektu. Zvláštností navrhované metody je iterativní prohledávání hlasovacího prostoru a vymazávání hlasů pro již detekované objekty. Nalezené objekty mohou být sledovány pomocí Mean-Shiftu .

Kapitola 2 obsahuje souhrn znalostí potřebných pro pochopení detekce objektů, tedy detekci význačných bodů, výpočet deskriptorů a lokalizaci objektu. Dále pak obsahuje úvod do shlukování potřebného k tvorbě slovníku. Kapitola 3 se zabývá technikami sledování již nalezených objektů. Kapitola 4 detailně popisuje hlavní část této práce, tedy kompletní návrh programu pro detekci a sledování objektů pomocí význačných bodů. V kapitole 5 jsou popsány vlastnosti implementovaného programu a vyhodnoceny experimenty na různých datových sadách. Závěrečná kapitola 6 pak rekapituluje obsah práce, komentuje dosažené výsledky a navrhuje další postup práce.

[1] http://www.qr-kody.cz/qr/android-qr-reader.html
[2] http://www.kinect.cz/?p=poznejte
[3] http://www.volkswagen.cz/technika/parkassist/

# Kapitola 2

# Detekce objektů

Detekce objektů může být prováděna nejrůznějšími technikami jako segmenace nebo porovnávání šablon. Tato kapitola neobsahuje obsáhlý výčet metod detekce objektů, ale spíše vysvětluje technologie přístupy, které jsou použity při návrhu aplikace, jež je součástí této práce.

## 2.1 Význačné body

Význačný bod (Bod zájmu, Klíčový bod) je místo v obraze, které má vysokou informační hodnotu vůči svém okolí (změna 2D signálu v jeho okolí). Za význačné body jsou v různém kontextu považovány osamocené body, rohy, hrany nebo výrazné oblasti. Ačkoliv je tato podkapitola dělena na detektory rohů a oblastí, v dalším textu se typ význačného bodu nerozlišuje.

Požadavkem na detektory význačných bodů je detekce odpovídajících si bodů na obrázcích stejné scény při změně pohledu na scénu (posun, rotace, zkosení, změna měřítka či osvětlení).

Na obrázku 2.1 jsou zobrazeny význačné body pořízené různými detektory.

### 2.1.1 Detektory rohů

Roh lze pro potřeby počítačového vidění definovat jako spoj dvou a více různých hran, samostatný bod, konec hrany či vrchol křivky.

#### Moravcův detektor

Moravcův detektor rohů [22, 40] je jedním z nejstarších algoritmů na detekci význačných bodů. Význačný bod je definován jako oblast s malou sebepodobností vůči okolí. Detekce je založena na horizontálním, vertikálním a 2x diagonálním posunu čtvercového okna po obraze a počítání změny jasu (sum of squared differences) mezi posunutými okny. Mohou nastat 3 případy:

1. Jestliže je v okně plocha, všechny posuvy mají malou změnu.
2. Je-li pod oknem hrana, potom se posuv po hraně neprojeví velkou změnou, ale posuv kolmo k hraně se projeví velkou změnou.
3. Je-li pod oknem roh nebo izolovaný bod, potom se všechny posuvy projeví velkou změnou.

Obrázek 2.1: Ukázky význačných bodů pořízených různými detektory. Zleva: Originální obrázek, Harrisův detektor, FAST, DoG (Difference of Gaussians), DoH (Determinant of Hessian), MSER (Maximally stable extremal regions).

Změna $E$ produkovaná posuvem (x,y) v obraze $I$ je dána rovnicí:

$$E_{x,y} = \sum_{u,v} w_{u,v}(I_{u+x,v+y} - I_{u,v})^2 \tag{2.1}$$

kde $w$ specifikuje čtvercové okno (1 uvnitř okna, 0 jinde).

Význačné body u Moravcova detektoru se nacházejí v lokálních maximech $E$ přesahujících nastavený práh.

### Harrisův detektor

Harrisův detektor [22] rohů a hran vychází z Moravcova detektoru a odstraňuje jeho známe nedostatky (posuv okna o diskrétní vzdálenost ve směru $k * 45°$, čtvercové binární okno a odezva pouze na rohy). Na místo posouvání čtvercového okna o diskrétní vzdálenosti se gradienty obrazu počítají pomocí derivací

$$E(x, y) \approx (x, y)M(x, y)^T \tag{2.2}$$

kde M je 2x2 symetrická matice

$$M = \begin{bmatrix} A & C \\ C & B \end{bmatrix} = w * \begin{bmatrix} L_x^2 & L_xL_y \\ L_xL_y & L_y^2 \end{bmatrix} \tag{2.3}$$

$w$ je kruhové gaussovské okno, $L_x^2$ je druhá parciální derivace obrazu podle x, $L_y^2$ je druhá derivace podle y a a $L_xL_y$ derivace podle x a y.

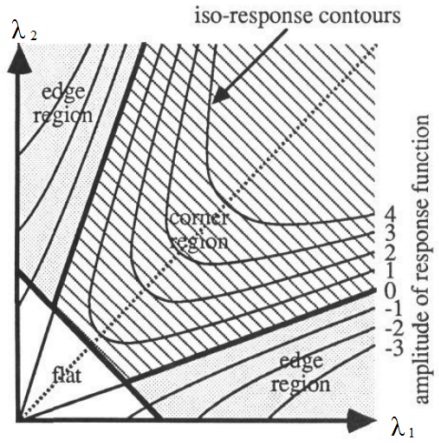
$\lambda_1$ a $\lambda_2$ (vlastní čísla matice M, viz rovnice 2.4 ) přímo určující změny jasů v okolí bodu a tedy určují, zda se v daném bodu nachází plocha, hrana nebo roh (viz obr. 2.2).

$$\lambda_{1,2} = \frac{-(A+B) \pm \sqrt{(A+B)^2 - 4(AB - C^2)}}{2} \tag{2.4}$$

Aby byly nalezeny pouze nejsilnější osamostatnělé rohy nebo hrany, je třeba zachovat body s největší odezvou $R$ v okolí. Používá se vyjádření nevyužívající vlastní čísla

$$R = (\lambda_1 \lambda_2) - k(\lambda_1 + \lambda_2)^2 \tag{2.5a}$$
$$= Det(M) - kTr^2(M) \tag{2.5b}$$
$$= (AB - C^2) - k(A+B)^2 \tag{2.5c}$$

$R$ je pozitivní pro rohy, negativní pro hrany a blízké nule pro rovinné oblasti.




Obrázek 2.2: Okolí bodů v závislosti na vlastních číslech matice. Zdroj: [22].

### Harris-Laplace

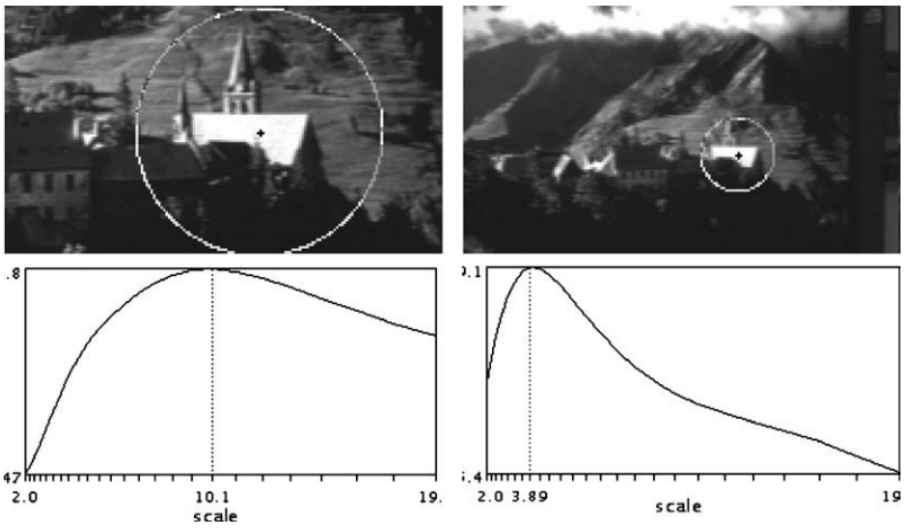
Jedná se o rozšíření [38] Harrisova detektoru, kdy je pro nalezené body vypočítáno jejich charakteristické měřítko a poté aplikován affiní adaptační proces pro získání zkosení.

Nalezení charakteristického měřítka je provedeno iterativním procházením Scale space (viz další podkapitola) v místě význačného bodu a vypočítáním vyhodnocovací funkce v každé úrovni. Maximum této funkce určuje charakteristické měřítko, viz obrázek 2.3.

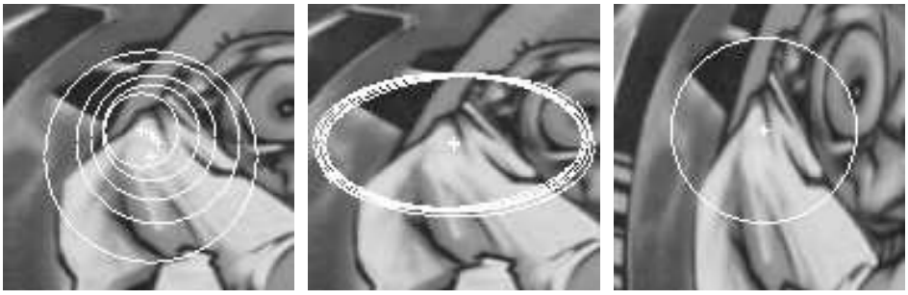
Affiní adaptační proces (viz obrázek 2.4) je iterativní algoritmus pro výpočet zkosení. Postupně se vypočítává integrační a derivační měřítko, matice druhých momentů a vlastní čísla této matice. Výpočet je zastaven jakmile se přestane měnit matice druhých momentů. Vlastní čísla matice určují zkosení.

## 2.1.2 Detektory oblastí

Detektory oblastí detekují oblasti v obraze, které se liší barvou či jasem vůči svému okolí. Narozdíl od detektorů rohů tedy detekují i oblasti bez osamocených bodů, hran, či rohů.




Obrázek 2.3: Nalezení charakteristického měřítka. Zdroj: [38].



Obrázek 2.4: Nalezení zkosení. Pozice význačného bodu v jednotlivých vrstvách scale space; Affiní oblasti získané affiním adaptačním procesem; Normalizování obrazu. Zdroj: [38].

### Laplacian of Gaussian (LoG)

Laplacian of Gaussian [30, 31, 34] je jedna z prvních metod detekce oblastí. Základem této metody je vytvoření Scale space (viz obr. 2.5). Scale space $L$ vzniká filtrací obrazu $I$ 2D Gaussovým filtrem $G$ s různou $\sigma$ (násobky $\sqrt{2}$) (viz rovnice 2.6). Díky separabilitě 2D Gaussova filtru lze filtraci realizovat dvěma průchody 1D Gaussovým filtrem (viz rovnice 2.7) v horizontálním a vertikálním směru.

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \tag{2.6}$$

$$G(x, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-x^2}{2\sigma^2}} \tag{2.7}$$

Poté je spočten Laplacian na jednotlivých vrstvách :

$$LoG(x, y, \sigma) = \frac{\partial^2 L(\sigma)}{\partial x^2} + \frac{\partial^2 L(\sigma)}{\partial y^2} \tag{2.8}$$

jiný možný zápis:

$$\Delta^2_L(x, y, \sigma) = \sigma(L_{xx} + L_{yy}) \tag{2.9}$$

Jako význačný bod je určen extrém v LoG Scale space. Tedy bod, jenž je minimem/maximem ve svém Scale space 26-okolí (viz. obr. 2.6b).

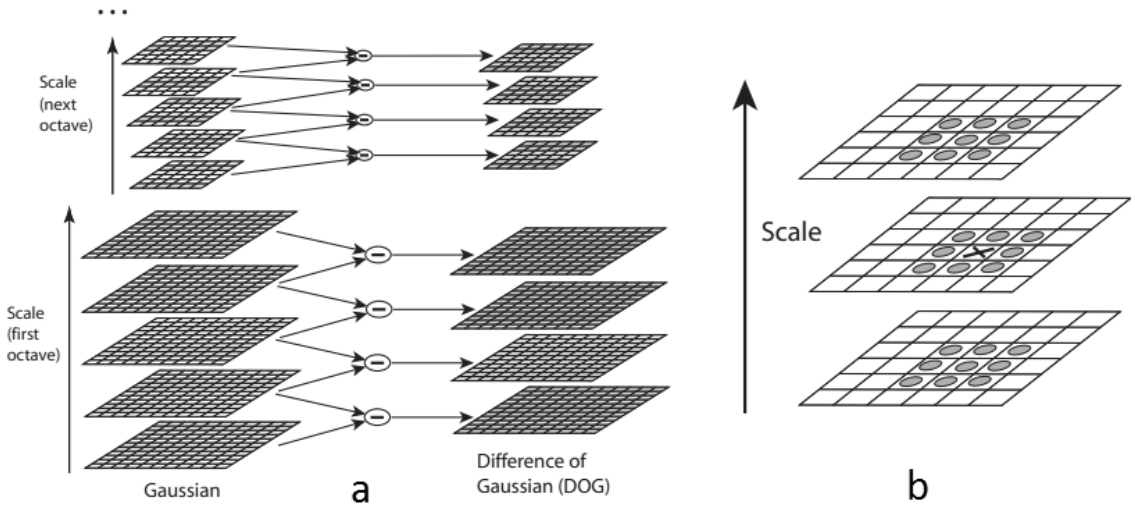
Obrázek 2.5: Příklad Scale space.

### Difference of Gaussians (DoG)

Metoda Difference of Gaussians [34, 30] je aproximací metody Laplacian of Gaussian (LoG). DoG dosahuje srovnatelných výsledků jako LoG a to při menší výpočetní náročnosti [33, 8]. Vytvoření Scale space probíhá obdobně jako u LoG, na závěr však jsou jednotlivé vrstvy zmenšeny bilineární interpolací se vzorkováním 1.5 (nový bod vznikne právě ze 4 původních bodů). Takto vznikají jednotlivé oktávy (osmice obrazů stejných rozměrů s různým rozmazáním). Po vytvoření Scale space se provede postupný odečet 2 sousedních rozmazaných obrazů (viz rovnice 2.10, obr. 2.6 a ) namísto Laplacianu.

$$D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) \tag{2.10a}$$
$$= L(x, y, k\sigma) - L(x, y, \sigma) \tag{2.10b}$$

Význačný bod je lokalizován v minimu či maximu DoG obrazu (viz obr. 2.6 b).




Obrázek 2.6: DoG. a: Vznik DoG z jednotlivých oktáv Scale space; b: Maxima a minima DoG obrazů jsou přímo detekovány porovnáním pixelu X s jeho 26-okolím. Zdroj: [34].

### Determinant of Hessian (DoH)

Determinant of Hessian [31] spočívá ve vyhledání maxima determinantu Hessianovy matice:

$$H(X,\sigma) = \begin{bmatrix} L_{xx}(X,\sigma) & L_{xy}(X,\sigma) \\ L_{xy}(X,\sigma) & L_{yy}(X,\sigma) \end{bmatrix} \tag{2.11}$$

kde $L_{xx}(X,\sigma)$ je konvoluce 2. parciální derivace Gaussovi funkce s obrazem $I$ v bodě $X$.

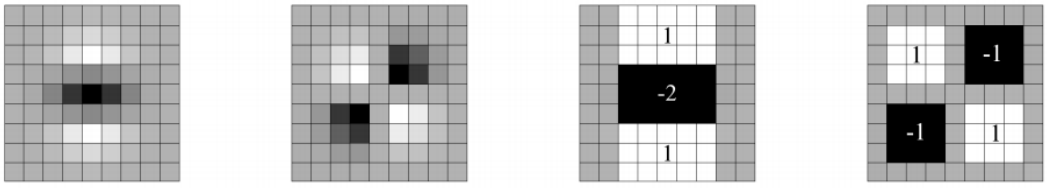
Protože Gaussova funkce musí být diskretizována a tudíž ztrácí svoje ideální vlastnosti, Bay [8] (inspirovaný Loweho úspěšnou aproximací LoG $\rightarrow$ DoG) navrhl metodu založenou na integrálním obrazu (používaný na detekci obličeje v [58]) a hrubých aproximacích Gaussových konvolučních jader na Box filtry (viz obr. 2.7). Hledá proto maximum determinantu aproximované Hessianovy matice (rovnice 2.12) určující pozici význačného bodu.

$$det(H_{approx}) = D_{xx} * D_{yy} - (wD_{xy})^2 \tag{2.12}$$

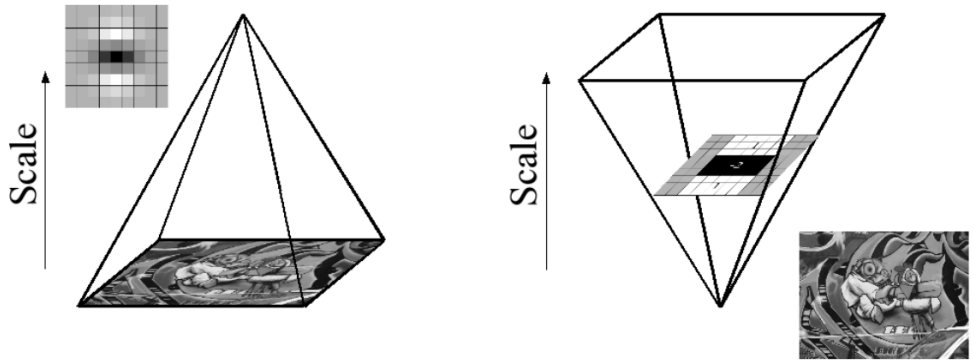
$w = 0.912\cdots \approx 0.9$ slouží k vyvážení energie mezi Gaussovým jádrem a aproximovaným Gaussovým jádrem.
Stejně jako u DoG je potřeba vytvořit Scale space pro detekci velikosti význačného bodu. Zde se však využívá již spočítaného integrálního obrazu a jeho konstantní doby výpočtu konvoluce (nezávislé na velikosti konvolučního jádra). Proto se místo postupného zmenšování obrazu a konvolucí s konstantním jádrem používá postupně zvětšované jádro a konstantní obrázek (viz obr. 2.8). Scale space je rozděleno do oktáv. Každá oktáva reprezentuje jeden obraz získaný konvolucí se vzrůstající velikostí jádra.
Narozdíl od DoG není pozice a scale určen extrémem v Scale space, ale pozice je určena maximem determinantu matice a scale jako maximum funkce interpolované přes jednotlivé hodnoty v nalezené pozici napříč Scale space. Interpolace je potřebná kvůli velkému rozdílu mezi prvními vrstvami jednotlivých oktáv.

### Maximally stable extremal regions (MSER)

MSER [15] vyhledává oblasti (regiony) s největším gradientem jasu na okraji pomocí adaptivního prahování. Obraz $I$ se v každém kroku postupně rozkládá na oddělené regiony $R^g$




Obrázek 2.7: Vlevo: Diskretizované 2. parciální derivace Gaussovi funkce ($G_{yy}$ a $G_{xy}$). Vpravo: Odpovídající aproximace $D_{yy}$ a $D_{xy}$ pro $\sigma = 1.2$. Zdroj: [8].



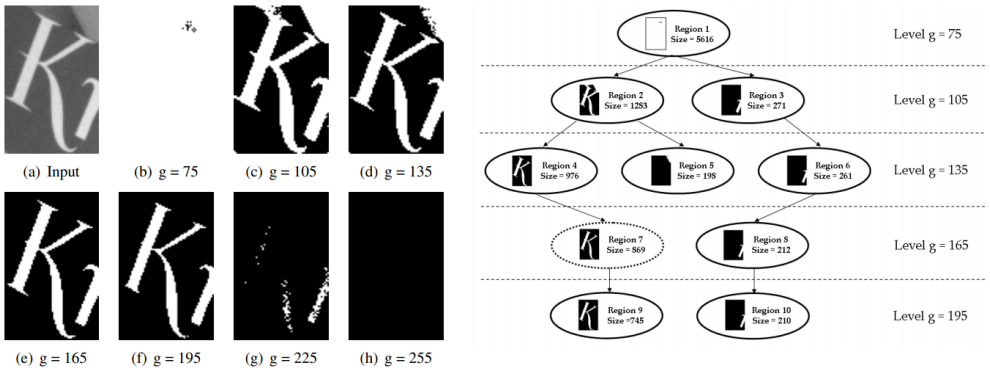

Obrázek 2.8: Rozdíl při budování Scale space. Vlevo:DoG; Vpravo: DoH [8]

pomocí prahování se vzrůstajícím prahem $g$. Tyto regiony se ukládají do Stromu komponent (viz obr. 2.9). V každém kroku se zaznamenává velikost jednotlivých regionů. Jako stabilní je označen region s lokálním minimem funkce

$$\psi(R_i^g) = (|R_j^{g-\Delta}| - |R_k^{g+\Delta}|)/|R_i^g|) \tag{2.13}$$

kde $|.|$ značí kardinalitu, tedy velikost regionu, $R_i^g$ Region $i$ získaný při prahu $g$ a $\Delta$ rozsah stability

V příkladu na obrázku 2.9 je jako MSER označen Region 7 pro nějž $\psi$ dosahuje lokálního minima. O každém detekovaném MSERu se ukládá průměrná hodnota barvy, práh, při kterém byl region detekován, jeho velikost, střed a obalový box a hodnota stability $\psi$. Tyto informace se dále používají pro sledování objektu.




Obrázek 2.9: MSER. Vlevo: Prahování s prahem $g$; Vpravo: Strom komponent. Zdroj: [15].

#### Hessian-Laplace

Jedná se o obdobu Harris-Laplace. Detekují se význačné oblasti pomocí Hessianovy matice, vypočte se charakteristické měřítko a provede se affiní adaptační proces.

## 2.2 Deskriptory

Jakmile je lokalizován význačný bod (pozice, velikost a orientace), je vhodné zjistit informace o jeho okolí. K tomuto účelu se používají deskriptory. Uložení barvy pixelů v okolí význačného bodu se všeobecně nepoužívá kvůli špatné porovnatelnosti dvou stejných bodů ze scény při změně osvětlení či pozice pozorovatele ve 3D scéně.

Mezi používané deskriptory patří Scale-invariant feature transform (SIFT), Speeded Up Robust Feature (SURF), Histogram of Oriented Gradients (HOG) [13], Gradient Location and Orientation Histogram (GLOH)[39], Local Energy based Shape Histogram (LESH) [47], Features from Accelerated Segment Test (FAST) [46], DAISY [55] nebo Binary Robust Independent Elementary Features (BRIEF) [10].

Každá z těchto metod obsahuje komplexní návrh extrakce příznaků (detekce význačných bodů a výpočet jejich deskriptorů). Navrhovaná aplikace však dovoluje použít typ deskriptoru nezávisle na technice detekce význačných bodů. Proto jsou tyto částí v tomto textu odděleny.

### 2.2.1 Patch

Descriptor se skládá z hodnot intenzity jasu ve čtvercovém okolí význačného bodu. Dle použité detekce význačných bodů lze Patch deskriptory rozdělit na několik druhů. Pokud detektor určí pouze pozici význačného bodu (takto funguje většina detektorů rohů), lze deskriptor získat jako čtvercový výřez obrazu o straně 25px se středem ve význačném bodě.

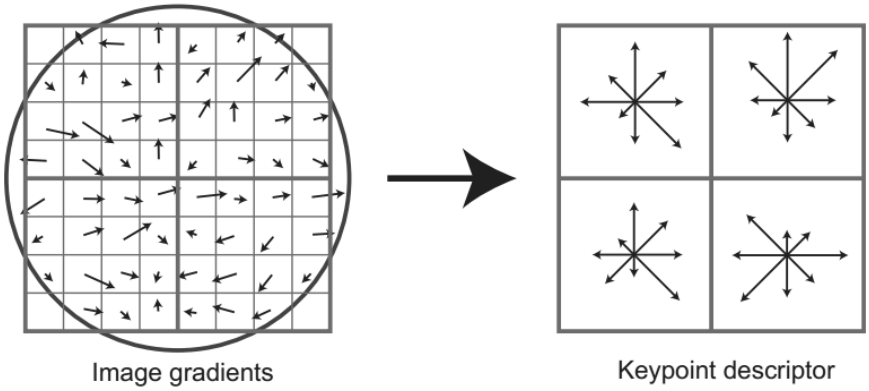
Detektory oblastí poskytují informaci o velikosti význačné oblasti, proto lze deskriptor získat zvětšením oblasti na požadovanou velikost. Rescale má své opodstatnění v zachování dimenzionality a snadnosti porovnávání. Scale patche lze použít pro scale-invariant detekci objektů

Detektory určující dominantí směr oblasti pak umožnují natočení obrázku před získáním deskriptoru. Tyto deskriptory lze použít pro rotation-invariant detekci.

Patche lze jako jediné použít pro vizualizaci jednotlivých etap detekce. Díky své vysoké dimenzionalitě (625 pro patche 25×25) jsou však nevhodné pro rychlé porovnání či shlukování. Dimenzionalitu lze snížit patříklad pomocí Principal Component Analysis (PCA), avšak za cenu ztráty informací.

### 2.2.2 SIFT

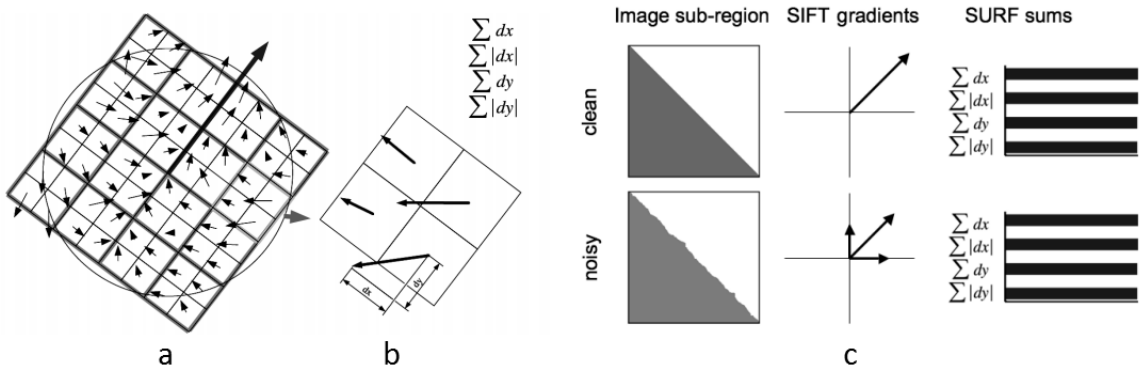
Lowe ve své práci [34, 33] bere okolí význačného bodu dle jeho velikosti, rozdělí ho pravidelnou čtvercovou mřížkou, spočítá gradienty jasu a tyto gradienty normuje 2D Gaussovou funkcí (viz obr. 2.10 Vlevo). Poté je vytvořen histogram gradientů jednotlivých oblastí, viz obrázek 2.10 Vpravo. Na obrázku 2.10 je deskriptor $2 \times 2$ vypočtený z okolí $8 \times 8$. V praxi se používá deskriptor $4 \times 4$ vypočtený z okolí $16 \times 16$. Deskriptor tedy obsahuje velikosti gradientů v 8 hlavních směrech pro každou oblast, tedy $8 * 4 * 4 = 128$ hodnot.




Obrázek 2.10: SIFT. Vlevo: Gradienty v okolí význačného bodu; Vpravo: SIFT deskriptor. Zdroj: [34].

### 2.2.3 SURF

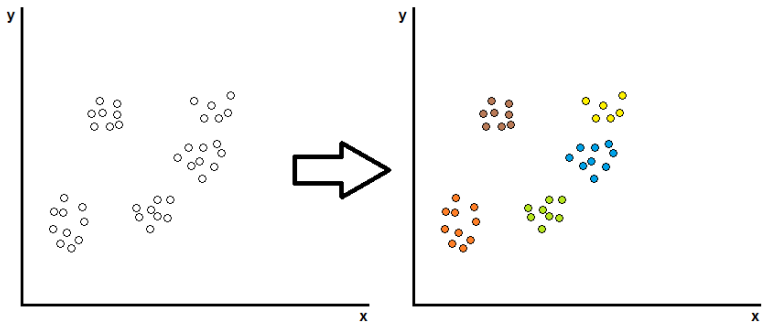
SURF deskriptor [8] vychází z metody SIFT s tím, že snižuje velikost deskriptoru kvůli rychlosti porovnávání při zachování výpočetní rychlosti a informační hodnoty. Pro lokalizaci význačných bodů metodou DoH (kapitola 2.1.2) se využívá integrálního obrazu. Tento integrální obraz se využívá i při výpočtu deskriptoru. Namísto výpočtu gradientů konvolucí obrazu a gaussianu jako u SIFTu se počítají odezvy Haarovy vlnky 1. řádu. Ve směru největší hustoty odezev na vlnky je určena orientace deskriptoru, v jejím směru je pak položena pravidelná čtvercová mřížka o velikosti schodné z velikostí význačného bodu. Mřížka rozděluje plochu význačného bodu na 16 oblastí (viz 2.11 a). Každá z těchto oblastí je dále rozdělena a v každé podoblasti je znovu spočtena odezvy na Haarovu vlnku (viz 2.11 a, b). Poté jsou v každé oblasti spočítány $\sum dx$, $\sum dy$, $\sum |dx|$, $\sum |dx|$ (viz obr. 2.11 b). Výhodou tohoto uložení je větší robustnost vůči obrazovému rušení než u SIFTu (viz obr. 2.11 c) SURF deskriptor obsahuje 4 hodnoty z každé $4 \times 4$ oblasti, dohromady tedy 64 hodnot.




Obrázek 2.11: **a**: Okolí význačného bodu s vyznačenou orientací (velká šipka), $4 \times 4$ mřížka, jejíž jedna část je znázorněna v *b*. **b**: oblast se znázorněným *dx* a *dy*. **c**: rozdíl mezi SIFT a SURF deskriptory zašumělé hrany. Zdroj: [8].

## 2.3 Shlukování

Shlukování (clustering) provádí seskupování množiny objektů do shluků (clusterů). Aplikace navrhovaná v této práci využívá shlukování bodů (vektorů) v n-rozměrném prostoru. Cílem shlukování je vytvoření kompaktních shluků v tomto prostoru, viz obrázek 2.12.




Obrázek 2.12: Ukázka clusteringu bodů ve 2D prostoru. Příslušnost do shluků je rozlišena barvou. Zdroj: [32].

### 2.3.1 K-Means

K-Means [32, 36] patří do skupiny centroidních modelů, kdy je předem daný celkový počet shluků $k$ a každý shluk je reprezentován jedním bodem (centroidem) představující střed shluku. Tento bod nemusí být obsažen v původní množině bodů. Výsledné rozdělení do shluků je charakterizováno kompaktností

$$J = \sum_{j=1}^{k} \sum_{i=1}^{|C_j|} (x_i^j - c_j)^2 \tag{2.14}$$

**Algorithm 1** Algoritmus K-Means

```
X ← {x_1, x_2, ..., x_n}                    // Množina bodů dat
C ← k pseudonáhodných bodů z X               // Množina středů
S ← θ                                        // Inicializace shluků
while e ≤ t1 or i ≤ t2 do                    // Dosažena přesnost nebo počet iterací
  for all x_i ∈ X do
    for all c_j ∈ C do
      if Dis(x_i, c_j) = min then
        S_j ← S_j ∪ x_i                      // Přiřazení bodu do shluku s nejbližším středem
      end if
    end for
  end for
  for all S_i ∈ S do
    c_i ← Avg(S_i)                           // Výpočet nového středu shluku
  end for
end while
```

K-Means je iterativní algoritmus (viz algoritmus 1), v každé iteraci je každý bod přiřazen do shluku s nejbližším středem. Poté je střed vypočten jako těžiště shluku. Cyklus je omezen

počtem iterací nebo přesností, tzn. změnou pozice středů shluků v jednotlivých krocích. Z jednoduchosti algoritmu vyplývá, že rychlost a kvalita shlukování závisí na inicializaci počátečních středů. Jako nejvhodnější inicialize se jeví metoda K-Means++ [3, 4], protože dosahuje nejrychlejší konvergence (logaritmická složitost inicializace $O(log(k))$ a cca 15 cyklů pro dosažení stabilních shluků).

Nevýhodou k-means je nutnost znalosti počtu shluků a různé výsledky v závislosti na počáteční inicializaci. Pro nalezení nejvhodnějšího rozdělení shluků se algoritmus provádí několikanásobně s různou inicializací a ponechává se rozdělení s největší kompaktností (nejmenší $J$ z rovnice 2.14).

### 2.3.2 Aglomerativní shlukování

Jedná se o hierarchické shlukování zdola nahoru, kdy na počátku je každý objekt ve vlastním shluku. Poté se vždy 2 shluky s nejmenší vzdáleností (viz metody níže) sjednotí do společného shluku. Toto se provádí až do vzniku jediného shluku nebo dokud nejmnenší vzdálenost mezi dvěmi shluky nepřekročí daný práh.

Metody pro výpočet vzdálenosti shluků:

- **Single-linkage clustering** $min\{d(a,b) : a \in A, b \in B\}$ [53]
- **Complete-linkage clustering** $max\{d(a,b) : a \in A, b \in B\}$ [14]
- **Average-linkage clustering** $\frac{1}{|A|*|B|} \sum_{a \in A} \sum_{b \in B} d(a,b)$ [50]

kde $d(a,b)$ je vzdálenost dle zvolené metriky, což nejčastěji bývá Euklidovská vzdálenost.

Výhodou tohoto typu shlukování je možnost nastavení maximální vzdálenosti shluků a tedy ponechání osamocených bodů v singletonu, nevýhodou pak velká časová složitost $O(n^3)$, po optimalizaci $O(n^2 * log(n))$ [53, 14].

## 2.4 Lokalizace objektu

Lokalizací objektu je myšleno nalezení korespondence mezi vzorem (modelem) a částí prohledávaného obrazu.

Korespondencí je myšleno, že část jednoho obrazu je částí druhého obrazu. Nalezení korespondencí je nutné pro problémy jako stereomatchting, panoramatické fotografie, či v této práci probíraná detekce objektů.

Metody RANSAC a Houghova transformace se používají k nalezení objektů reprezentovaných parametrickými rovnicemi. Obecná Houghova transformace slouží k nalezení objektů definovaných orientovanými hranami. Všechny tyto metody hledají výskyt konkrétního známého objektu v obrázku. Pro nalezení neznámého objektu ze známého typu slouží například metoda Interleaving object categorization and segmentation.

### 2.4.1 RANSAC

RANdom SAmple Consensus (RANSAC) [18] je metoda používaná na vyhledání nejvhodnějších parametrů pomocí nejlepší shody mezi daty a modelem.

Narozdíl od technik odhadu parametrů, jako metoda nejmenších čtverců počítajících parametry ze všech dat, RANSAC rozděluje data na inliners (data náležející modelu) a outliners (data nenáležející modelu) a parametry vypočítává pouze z inliners, viz algoritmus 2.

**Algorithm 2** Algoritmus RANSAC

```
X ← {x_1, x_2, ..., x_n}                                   // Všechy body dat
m ← stupněm volnosti modelu, m ≤ n.
for i = 1 to K do
    Y ← m náhodných prvků z X
    M ← model(X, Y)                                        // Výpočet modelu M
    for all x_j ∈ X do                                     // Zjištění počtu inliners naležících modelu M
        if x_j ∈ M then
            inliners++
        end if
    end for
    if inliners ≥ t then
        return M.
    end if
end for
return M_i, inliners(M_i) = Max                            // Model s největším počtem inliners
```

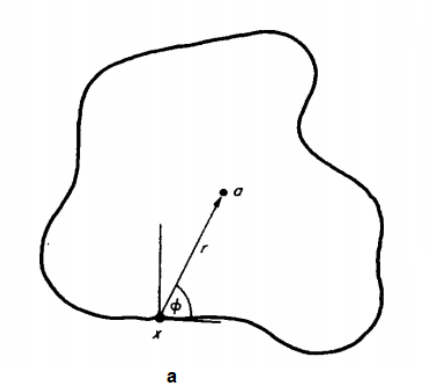
Největší výhoda této metody je již zmiňované rozdělení na inliners a outliners, tedy možnost detekce modelu i v datech s velkým šumem, či s velkým počtem nesouvisejících dat. Nevýhodou metody je neexistence horní hranice časové složitosti. Nastavení horní časové složitosti algoritmu lze provést omezením počtu iterací. Pak však nalezené řešení nemusí být optimální. Časová složitost závisí na počtu parametrů modelu $m$ a na počtu outliners. Protože $m$ omezit nelze, jediná možná optimalizace je proto omezení prohledávané množiny dat dle apriorních informací. Ransac je primárně určen k vyhledání pouze jednoho modelu v množině dat, pro vyhledání více modelů je algoritmus nutné upravit nebo použít jinou metodu, například Houghovu transformaci.

### 2.4.2 Houghova transformace

Houghova transformace [16] je metoda používaná k nalezení parametrů objektu v obraze. Hledaný objekt tedy musí být analyticky popsatelný parametrickou rovnicí. Proto se tato metoda používá zejména pro hledání přímek, kružnic, elips či jednoduchých křivek a objektů popsatelnými těmito útvary. Hlavní výhodou této metody je robustnost vůči nepravidelnostem, šumu a porušení či překrytí hledaného objektu.
Metoda funguje na principu transformace všech bodů původního obrázku na křivky v Houghovu $n$-rozměrného prostoru, kde $n$ je počet parametrů modelu. Houghovu prostoru se též říká akumulátor, protože je inicializován na nulu a hodnota buněk podél každé křivky je inkrementována. Pozice maxima akumulátoru pak určuje parametry nalezeného objektu. Protože v původním obrázku muže být objektů více, lokální maxima větší než práh jsou ohodnocena jako kandidáti na objekt. Kandidáti jsou poté vyhodnoceni dle pravděpodobnosti výskytu. Ten může být spočítán například jako relativní intenzita vůči průměru, mediánu nebo maximu. Protože rychlost algoritmu závisí na počtu bodů v obrazu, je nejprve prováděna detekce hran, např. Sobelovým operátorem nebo Cannyho detektorem hran.

### 2.4.3 Obecná Houghova transformace

Protože Houghova transformace neumí vyhledávat objekty popsané jinak než analytickou parametrickou rovnicí, Bellard představil Obecnou (Generalizovanou) Houghovu transformaci (GHT) [6]. Obrázek i model hledaného objektu jsou reprezentovány formou orientovaných hran. Toto řešení velmi snižuje výpočetní náročnost algoritmu. Protože objekt nelze popsat rovnicí, využívá se R-tabulky. V modelu objektu je zvolen referenční bod $a$. Poté jsou procházeny všechny body $x_i$ modelu a do R-tabulky se ukládá závislost vzdálenosti $r = |x_i - a|$ na úhlu $\phi$ (viz obr. 2.13). Pro jeden úhel $\phi$ může být v tabulce uvedeno více hodnot $r$.




*R*-table format

| $i$ | $\phi_i$ | $R_{\phi_i}$ |
|---|---|---|
| 0 | 0 | $\{\mathbf{r} \mid \mathbf{a} - \mathbf{r} = \mathbf{x},\ \mathbf{x} \text{ in } B,\ \phi(\mathbf{x}) = 0\}$ |
| 1 | $\Delta\phi$ | $\{\mathbf{r} \mid \mathbf{a} - \mathbf{r} = \mathbf{x},\ \mathbf{x} \text{ in } B,\ \phi(\mathbf{x}) = \Delta\phi\}$ |
| 2 | $2\,\Delta\phi$ | $\{\mathbf{r} \mid \mathbf{a} - \mathbf{r} = \mathbf{x},\ \mathbf{x} \text{ in } B,\ \phi(\mathbf{x}) = 2\,\Delta\phi$ |
| ... | ... | ... |

b

Obrázek 2.13: Obecná Houghova transformace. A: Geometrie obecné houghovy transformace. Referenční bod $a$ a libovolný bod $x$, vzdálenost $r = |a - x|$ a úhle $\phi$ mezi osou X a $r$ B: R-Tabulka. Zdroj: [6].

Vyhledávání objektu v obrazu v nejjednodušší formě (fixní velikost a orientace ) pracuje na následujícím principu: Pro každý bod $x$ s úhlem $\phi$, který svírá hrana v bodě $x$ s osou $X$, se akumulátor inkrementuje v bodech $x + r_i$, kde $r_i$ jsou hodnoty R-tabulky pro úhel $\phi$. Všechny lokální maxima akumulátoru přesahující práh jsou ohodnocena jako referenční body (středy) objektů.

Pokud vyhledávané objekty mohou mít v obrázku různou velikost a orientaci, rozměry akumulátoru se zvětší ze 2D (pozice) na 4D (pozice, orientace a velikost). Pokud původní R-tabulku označíme jako $R(\phi)$ pak jednoduchými transformacemi získáme R-tabulky pro detekci zvětšených (rovnice 2.15) nebo natočených (rovnice 2.16) objektů. Tyto tabulky mohou být sloučeny do jedné, kdy každému $\phi$ jsou dány příslušné body ve všech velikostech a natočeních.

$$T_s[R(\phi)] = sR(\phi) \tag{2.15}$$

$$T_\theta[R(\phi)] = Rot\{R[(\phi - \theta) mod 2\pi], \theta\} \tag{2.16}$$

### 2.4.4 Interleaving object categorization and segmentation (IOCS)

Předchozí metody v této podkapitole hledají homografii u obrázků, ve kterých se nachází téměř identický objekt. Pokud však hledáme objekt, jehož přesnou podobu neznáme, ale známe podobu objektů stejného typu, lze použít IOCS navržený v [29] a s obměnou použitý v [59].

IOCS vychází z Obecné Houghovy transformace. Základem této metody je vytvoření slovníku (Implicit Shape Model, ISM) $ISM(C) = (C, P_C)$. Každý záznam je složen z

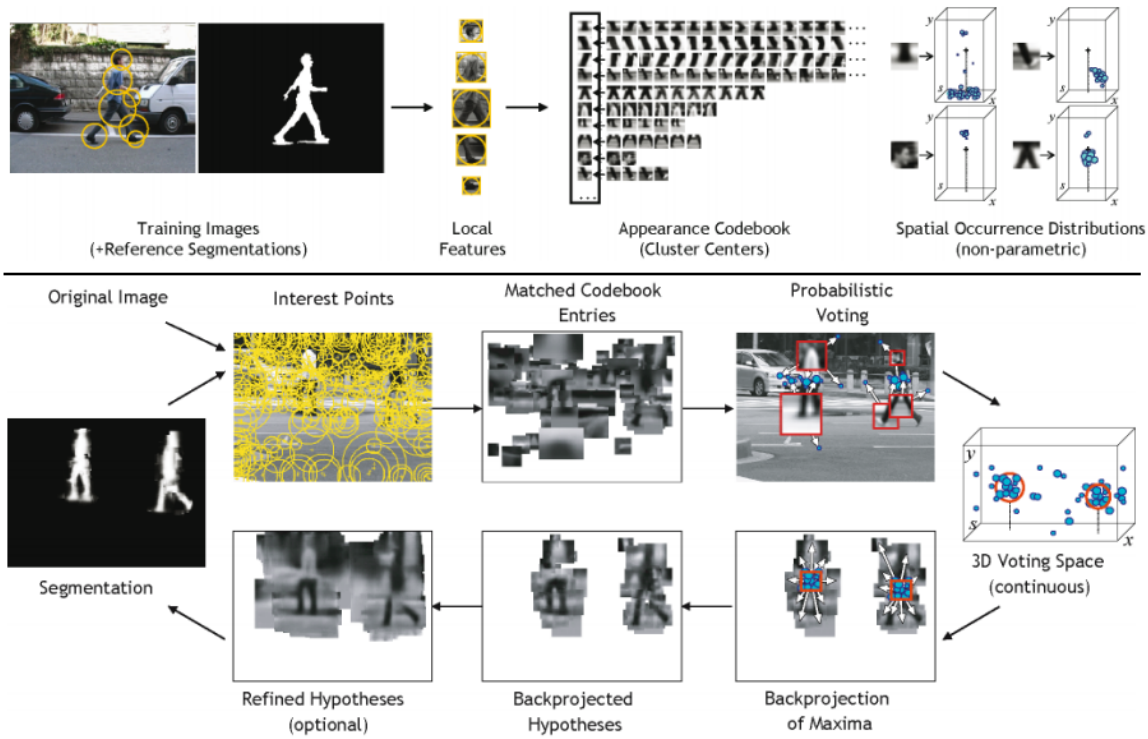
typické části $C$ objektu a rozložením prostorové pravděpodobnosti výskytu $P_C$ této části v objektu. Rozložení pravděpodobnosti je neparametrické a nezávislé pro každý záznam slovníku. Takovýto slovník je dostatečný pro detekci libovolného objektu daného typu. Slovník nahrazuje R-Tabulku z GHT. ISM je použit v aplikaci navrhované v této práci, detaily o jeho tvorbě jsou v kapitole 4.1.

Detekce objektu má několik fází. Nejprve se naleznou význačné body v testovaném obrázku a spočítají se jejich deskriptory. Deskriptory se porovnají se slovníkem. Dostatečně podobné záznamy slovníku pak kopírují své hlasy (rozložení pravděpodobnosti) pro střed objektu do 3D akumulátoru (x, y a velikost). Jednotlivé biny akumulátoru neodpovídají pixelům obrazu, ale pokrývají větší plochu.

Lokální maxima akumulátoru narozdíl od GHT přímo neurčují pozici objektu ale slouží jako výchozí body pro přesné dohledání středu objektu pomocí Mean-shift trackingu ve spojitém prostoru hlasů.

Po nalezení středů objektů je provedena zpětná projekce bodů, hlasujících pro střed. Prostor mezi těmito body je také porovnán se slovníkem.

Všechny body zkopírují masky (dle slovníku) a vytvoří tak segmentaci objektu. Takto segmentovaný objekt je přijat nebo zamítnut pomocí MDL (Minimal Description Lenght) verifikace (viz obr. 2.14 Dolní část).



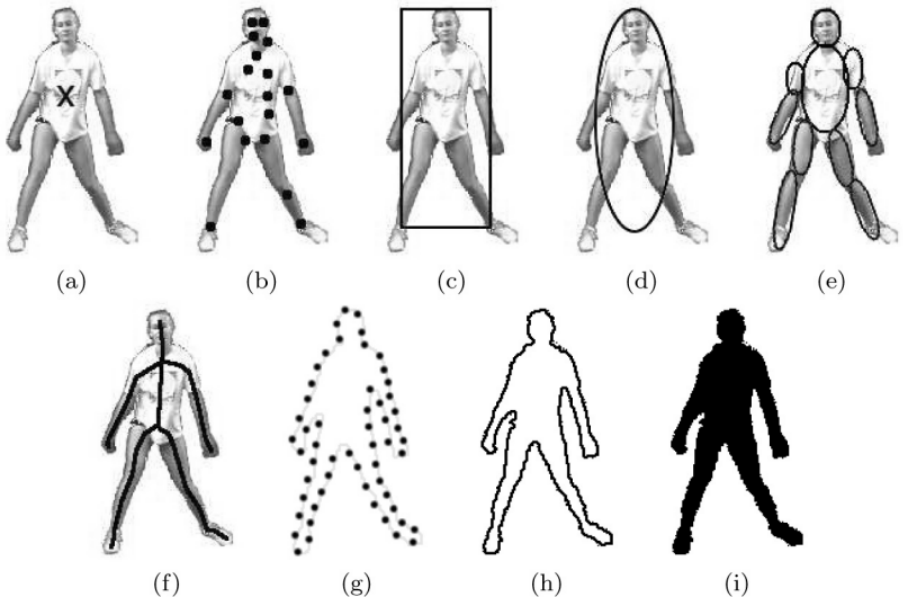

Obrázek 2.14: Horní část: Tvorba slovníku; Dolní část: Detekce a segmentace objektu. Zdroj: [28]

# Kapitola 3

# Sledování objektů

Sledováním objektů je v této práci myšlen proces lokalizace pohybujícího se objektu na souvisejících snímcích videa. Rozsáhlý přehled existujících technik je např. [61].

Metody sledování objektů lze rozdělit podle toho, jak je objekt reprezentován. Nejčastější reprezentace objektu jsou body (střed, význačné body), jednoduchá geometrická primitiva (obalový box a elipsa), obrys & silueta, kostra (ztenčení obrysu) nebo kloubový model (viz obrázek 3.1. Každá reprezentace objektu se hodí k jiné metodě sledování.




Obrázek 3.1: Reprezentace objektu. A: Střed objektu. B: Množina bodů. C: Obalový box. D: Elipsa. E: Model částí. F: Kostra objektu. G: Kontrolní body na obrysu. H: Kompletní obrys. I: Silueta. Zdroj: [61]

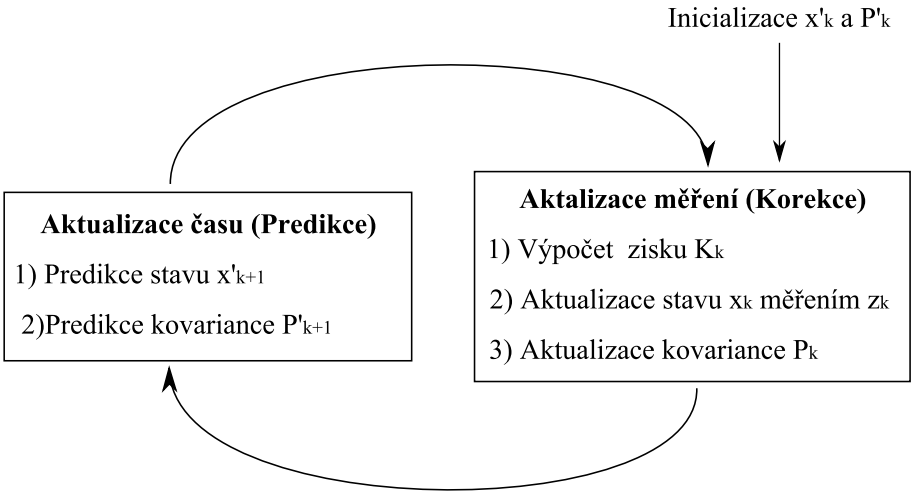
Pro sledování objektů pomocích význačných bodů se nejvíce hodí sledování bodu (středu objektu) nebo sledování geometrického primitiva (obalový box). Sledování kostry či objektu popsaného kloubovým modelem není vhodné pro sledování objektů libovolného, pro aplikaci neznámého, typu.

## 3.1 Sledování bodů

Sledování může být formulováno jako korespondence detekovaných objektů reprezentovaných body napříč jednotlivými snímky. Korespondence bodů je komplikovaný problém kvůli překrytí, špatné detekci, vstupům a výstupům objektů ze scény. Metody sledování bodů lze rozdělit na deterministické a statistické. Deterministické metody používají heuristiky kvalitativního pohybu k omezení korespondenčního problému. Používané deterministické metody jsou například Greedy Optimal Assignment (GOA) Tracker [57] nebo Multi-Frame (MF) [51]. Pravděpodobnostní metody berou při sestavování korespondence v úvahu kromě pozice objektu i chybu vzniklou při lokalizaci tohoto objektu. Mezi pravděpodobnostní metody patří Kalmanův filtr (níže), Částicové filtry [27, 35, 23], Joint Probability Data Association Filtering (JPDAF) [11, 42] a Multiple Hypothesis Tracking (MHT) [43]

### 3.1.1 Kalmanův filtr

Kalmanův filtr [60] je rekurzivní filtr používaný k odhadu lineárních dynamických jevů, kdy nejsou k dispozici přesné hodnoty nebo čas měření je delší než rychlost zobrazování. Stav systému je reprezentován vektorem reálných čísel a kovariancí mezi měřenými a odhadovanými hodnotami. Kalmanův filtr pracuje iterativně ve dvou krocích: Predikce následujícího stavu $x'_{k+1}$ z minulého upřesněného stavu. Upřesnění stavu $x_k$ v čase odpovídající novému stavu. Viz obrázek 3.2




Obrázek 3.2: Princip Kalmanova filtru

Rovnice Kalmanova filtru:

$$x'_{k+1} = Ax_k + Bu_k \tag{3.1}$$

$$P'_{k+1} = AP_kA^T + Q \tag{3.2}$$

$$x_k = y'_k + K(z_k - Hx'_k) \tag{3.3}$$

$$P_k = P'_k(I - KH) \tag{3.4}$$

$$K_k = \frac{P'_kH^T}{HP'_kH^T + R} \tag{3.5}$$

kde $x$ je stav systému, $x'$ odhadovaný stav systému, $P$ kovariance stavu, $P'$ kovariance odhadovaného stavu, $K$ zisk Kalmanova filtru, $M$ matice přechodu stavů, $B$ kontrolní

matice, $Q$ matice nepřesnosti měřeného procesu, $R$ matice nepřesnosti měření, $H$ matice měření a $z$ změřený stav systému.

## 3.2 Kernel tracking

Kernel Tracking je typicky vykonáván jako výpočet pohybu objektu reprezentovaného oblastí primitivního tvaru. Pohyb objektu bývá popsán parametricky (posunutí, rotace, zkosení ...) nebo hustotou toku. Metody detekce lze rozdělit do skupin podle použité reprezentace vzhledu. Tedy na metody používající *modely se vzhledem založeném na šablonách a hustotě* (*Template and Density-Based Appearance Models*) a na *vícepohledové metody* (*Multiview Appearance Models*).

Mezi metody používající *modely založené na šablonách a hustotách* patří například Template matching [49], prohledávající snímek $I_w$, hledající oblast podobnou šabloně objektu $O_t$ definovanou v předchozím snímku. Pozice šablony je počítána například pomocí korelace. Prohledávání snímku je prováděné "hrubou silou"(brute force). Místo šablony se také používají například histogram barev nebo průměrná barva oblasti představující objekt.

Příkladem metody používající vážené histogramy oblastí reprezentující objekt je metoda Real-Time Tracking of Non-Rigid Objects using Mean Shift [12] též nazívaná Mean-shift tracking nebo CAMSHIFT (OpenCV implementace). Výhodou této metody je použití Mean-Shift namísto brute force k nalezení objektu na následujícím snímku.
Další metodou je KLT tracker [52] který iterativně počítá posuv $(du, dv)$ oblasti $25 \times 25$ se středem ve význačném bodě.

$$\begin{pmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{pmatrix} \begin{pmatrix} du \\ dv \end{pmatrix} = \begin{pmatrix} \sum I_x I_t \\ \sum I_y I_t \end{pmatrix} \tag{3.6}$$

Jakmile je získána nová pozice význačného bodu, KLT tracker vyhodnotí kvalitu oblasti spočítáním affiní transformace

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix} \tag{3.7}$$

mezi korespondujícími oblastmi v následujících snímcích. Pokud je SSD (Sum of square difference) mezi současnou oblastí a minulou oblastí malé, je oblast ponechána ke sledování. Oblasti s velkým SSD jsou eliminovány.

Metody, popsané v předchozím odstavci, používají jako model vzhledu histogramy a šablony. Tyto modely jsou generovány online (za běhu programu). Pokud se však sledovaný objekt mění (3D rotace) během sledování, tyto metody selhávají. Řešením tohoto problému jsou *vícepohledové metody*, kdy je model objektu naučen před začátkem sledování z několika různých pohledů. Jednou z těchto metod je Support vector tracking (SVT) [5] používající Support Vector Machine (SVM) klasifikátor pro sledování. SVM je obecné klasifikační schéma, které na množině pozitivních a negativních trénovacích dat najde nejlepší dělící rovinu mezi dvěmi třídami. Během testování SVM uděluje skóre testovaným datům dle stupně příslušnosti do pozitivní třídy. Pro sledovač založený na SVM představují pozitivní data objekty určené ke sledování a negativní data představuje vše ostatní, tedy pozadí. Sledování probíhá maximalizací SVM klasifikačního skóre nad oblastmi snímku pro odhad pozice objektu.

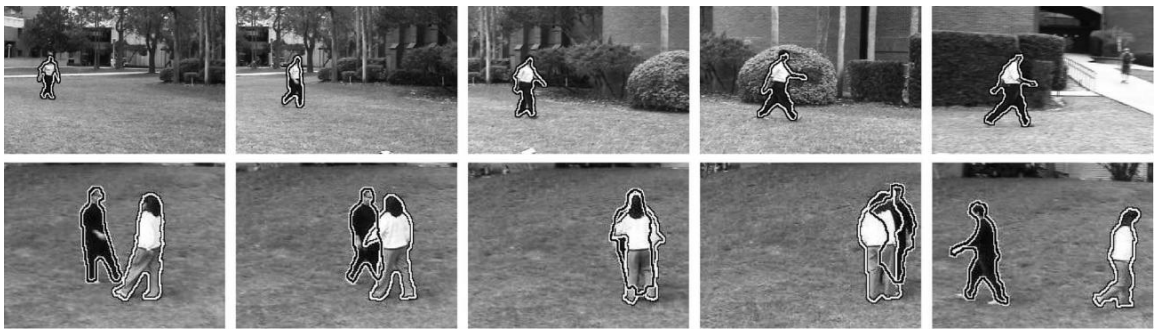
### 3.2.1 Mean-Shift tracking

Tato metoda slouží ke sledováni označené oblasti představující objekt. V označené oblasti referenčního snímku je spočítán histogram barev. Dle tohoto histogramu počítána zpětná projektce všech následujících snímků. Zpětná projekce zvírazňuje barvy obsažené v histogramu a potlačuje barvy nevyskytující se v histogramu. Na každém snímku se zpětnou projekcí je prováděn Mean-shift s počátkem v pozici objektu na předcházejícím snímku. Mean-Shift počítá postupné posuvy okna, představující objekt, ve směru přibývající hustoty.

Výhodou tohoto algoritmu je jeho jednoduchost, rychlost, korektní chování při zmizení objektu z obrazu a potřeba znalosti pouze předchozí pozice objektu. Nevýhodou je sledování pouze barevných objektů, rozlišení více objektů pouze dokud jsou barevně odlišeny či zaniknutí sledování při zakrytí objektu.

## 3.3 Sledování obrysů

Model může být ve formě historgamu barev, hran objektu nebo obrysu objektu. Sledování obrysů je prováděno *Porovnáním tvaru* (*Shapes matching*) [24, 48, 26] nebo *Vývojem obrysu* (*Contur evolution*) [25, 9, 45]. Oba typy provádějí segmentaci použitím znalostí generovaných v předchozích snímcích. Tedy z každého snímku se vypočítává model objektu pro následující snímek. Obrázek 3.3 zobrazuje ukázku sledování obrysů.



Obrázek 3.3: Ukázka sledování obrysů. Horní řada: Detekce objektů s dynamickou kamerou. Dolní řada: Vyhodnocení překrytí 2 objektů. Zdroj: [62]

# Kapitola 4

# Detekce objektů iterativním hlasováním

Aplikace navržena v rámci diplomové práce lokalizuje objekty libovolného typu (určeno trénovací sadou dat) ve videu a tyto objekty poté sleduje. Program je schopen lokalizovat a sledovat více objektů v reálném čase.

Aplikace je rozdělena na tvorbu slovníku a detekci objektů. Pro uživatele slovník představuje "black box", tedy soubor, jež si stačí stáhnout z internetu či vygenerovat pomocí přednastavených konfiguračních souborů a libovolné anotované sady obrázků.

Vstupem programu je cesta ke konfiguračnímu souboru a cesta k testovanému obrázku (či adresáři s obrázky). Výstupem je popis nalezených objektů (počet objektů, středy a velikosti), nebo uložení/zobrazení obrázku s vyznačenými objekty.

## 4.1 Tvorba slovníku

Slovníkem $ISM(c) = (C, P_C, w)$ je v tomto textu myšlen seznam obsahující vždy typickou část $C$ objektu, rozložení prostorové pravděpodobnost výskytu $P_C$ této části (realizováno jako seznam pozic výskytů této části na objektu) a váhou $w$ této části.

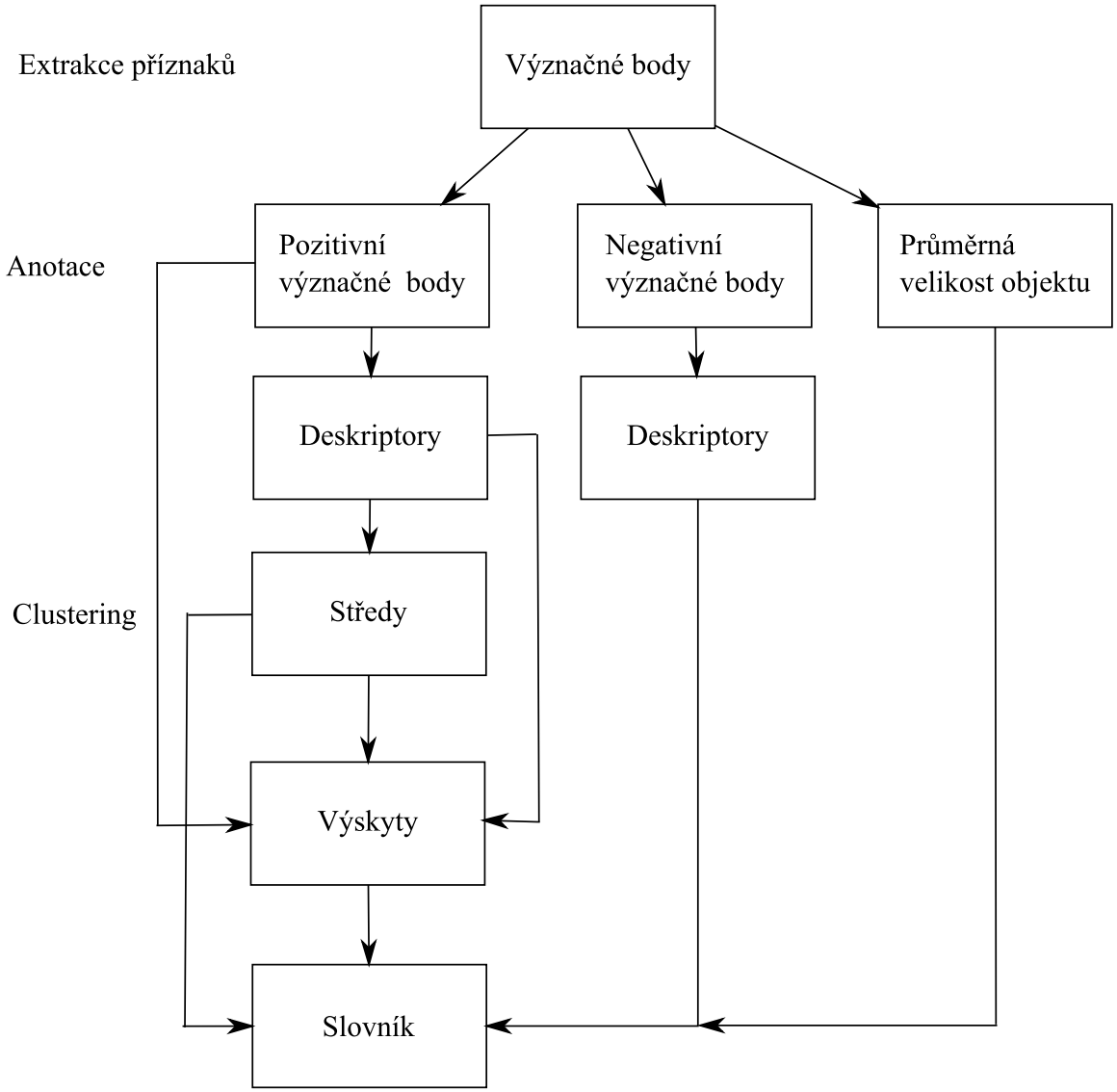
Vytvoření slovníku probíhá v několika krocích. Extrakce příznaků z anotovaných obrázků, shlukování těchto bodů a výhodnocení, kde na objektu se středy shluků vyskytují. Dále je pak možno provést výpočet váhy a ze slovníku odstranit nedůležité záznamy. Viz algoritmus 3.

Schéma na obrázku 4.1 ukazuje datový tok tvorby slovníku. Několik záznamů slovníku je vizualizováno na obrázku 4.2.

### 4.1.1 Extrakce příznaků

Extrakce příznaků je prováděna z jakékoliv anotované obrazové sady. Program předpokládá korektní uvedení cesty ke složce s touto sadou v konfiguračním souboru. Tato sada bude zpracována jako celek, do slovníku nelze přidávat data z dodatečných obrázků.

V každém obrázku se naleznou význačné body (Harris, DoG nebo DoH). Tyto body jsou rozděleny na náležející objektu (pozitivní) a nenáležející objektu (negativní). Rozdělení je možné na základě masky nebo podle anotace obrázku. Pro každý bod se vypočítá deskriptor (Patch, SIFT nebo SURF). Protože Patche mají pro učely shlukování a porovnání 625 dimenzí, je možno použít Principal Component Analysis (PCA). Pomocí PCA lze snížit

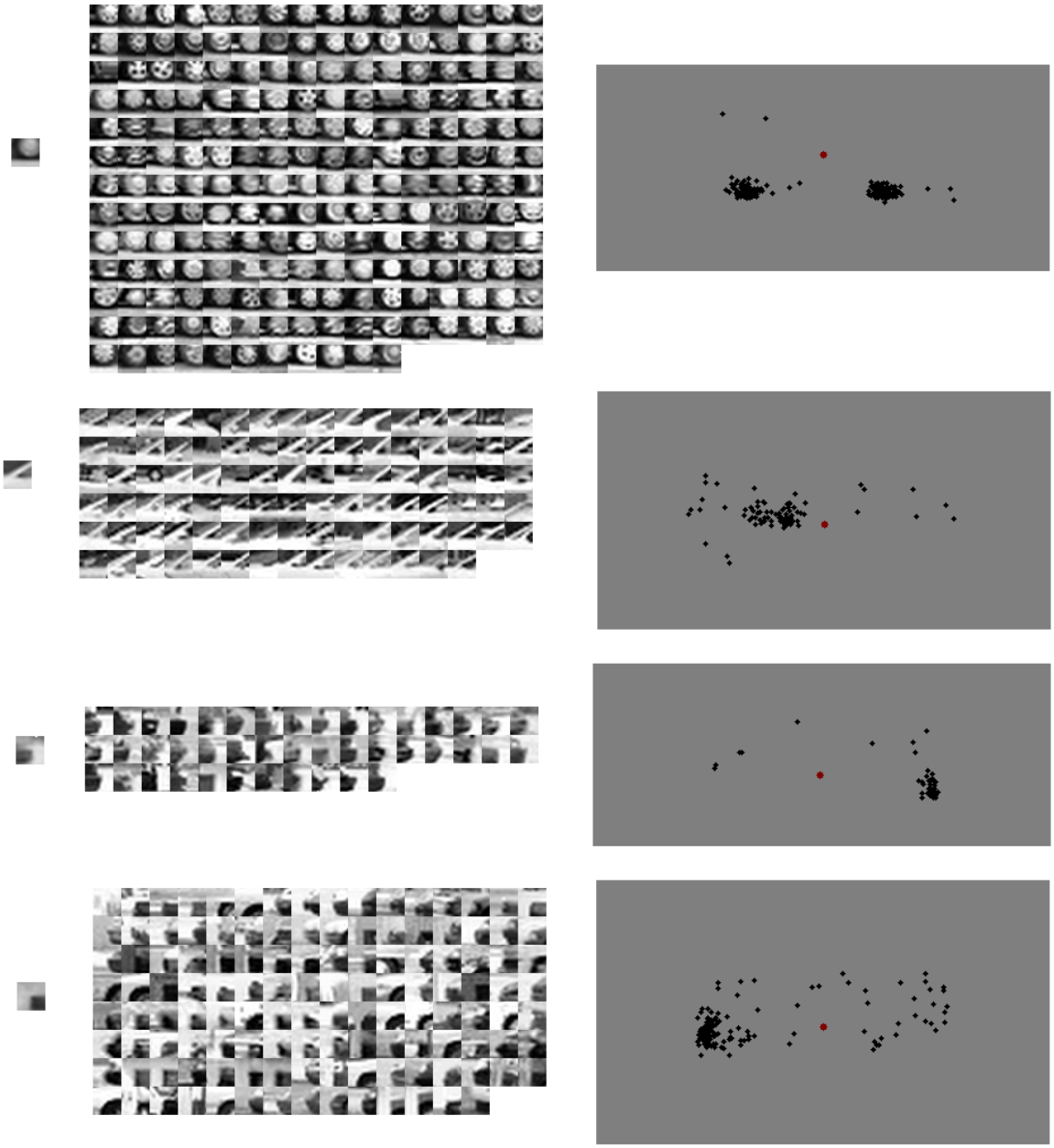



Obrázek 4.1: Datový tok tvorby slovníku.

dimenzionalitu na požadovanou velikost, avšak za cenu ztráty informační hodnoty deskriptoru. Pozice bodů náležících objektu se normalizuje vůči středu objektu.

Pro účely vylepšení detekce jsem se rozhodl začlenit do pravděpodobnostního modelu i negativní body. Do trénovací sady je možno zahrnout i obrazy bez objektů, podmínkou je tak prázdný soubor s anotací či negativní maska. Zahrnutí negativních bodů má význam, pokud tyto body představují pozadí, na kterých se hledaný objekt běžně vyskytuje. Tedy pokud se jedná o detekci chodců či automobilů, jsou fotografie prázdné ulice mnohem přínosnější než fotografie kočky či letadla. Detaily o začlenění negativních bodů do slovníku jsou v podkapitole 4.1.3.

Výstupem této části programu jsou soubory se seznamem pozitivních a negativních bodů včetně jejich deskriptorů, průměrná velikost objektu a případný soubor s PCA (vlastní čísla, vlastní vektory a odchylka).



Obrázek 4.2: Ukázka slovníku pro detekci automobilů: Typická část objektu $C$; Všechny význačné body trénovací sady, jejichž podobnost s touto částí je větší než práh; Rozložení prostorové pravděpodobnosti výskytu této části v objektu se středem v červeném bodě.

### 4.1.2 Nalezení typických částí objektu

Vytvoření typických částí $C$ objektů hledaného typu je docíleno shlukováním pozitivních bodů dle jejich deskriptoru. Výsledné středy shluků pak představují hledané typické části objektů. Není potřeba nalézt nejmenší možný počet středů, ale zajistit, že shluky obsahují vyzuálně podobné části, tedy stejné části objektu. Pro nalezení středů shluků je použit K-Means s K-Means++ inicializací.

Vstupem je seznam pozitivních bodů, výstupem soubor se středy shluků.

### 4.1.3 Rozložení pravděpodobnosti výskytu

Znovu se procházejí všechny pozitivní body a všechny středy shluků. Každý bod, jehož deskriptor má se středem podobnost větší než práh, uloží svojí pozici a velikost do seznamu výskytu tohoto středu. Díky tomuto je možné použít větší počet shluků než minimální. Nadbytečné shluky jsou duplicitní, ale nezhorší vizuální kompaktnost. Navíc je s výskytem duplicit počítáno v pravděpodobnostním modelu detektoru objektů.

Pro patch deskriptory je funkcí podobnosti normalizovaná korelace (Normalized Grayscale Correlation, NGC), popsaná rovnicí 4.1.

$$NGC(A,B) = \frac{\sum((A_i - \overline{A})(B_i - \overline{B}))}{\sqrt{\sum(A_i - \overline{A})^2 \sum(B_i - \overline{B})^2}} \tag{4.1}$$

kde $\overline{A}$ je průměrná hodnota v A. Deskriptory SIFT a SURF jsou porovnávány pomocí Euklidovské vzdálenosti, vyjádřené rovnicí 4.2.

$$EuclidDistance(A,B) = \sqrt{\sum(A_i - B_i)^2} \tag{4.2}$$

Zatímco normalizovaná korelace nabývá hodnot 0 pro zcela odlišné deskriptory až 1 pro identické deskriptory, Euklidovská vzdálenost je rovna 0 pro identické deskriptory, horní mez je pak závislá na počtu dimenzí a rozsahu hodnot deskriptoru. U funkce podobnosti SIFTů a SURFů je tedy snižující se práh zpřísňujícím se kritériem. Podobnost pomocí Euklidovské vzdálenosti je tedy implementována jako $sim(A,B) < t$.

Vstupem je seznam pozitivních bodů a středů shluků, výstupem je slovník.

### 4.1.4 Výpočet váhy

Procházejí se všechny negativní body a všechny středy shluků. Pro každý deskriptor bodu a střed, jejichž podobnost přesahuje hodnotu prahu, se zvyšuje počítadlo negativních výskytů pro střed.

Váha jednotlivých středů lze vyjádřit Bayesovskou statistikou. Váha středu X lze vyjádřit jako:

$$P(objekt|X) = \frac{P(X|objekt)P(objekt)}{P(X|objekt)P(objekt) + P(X|\neg(objekt))P(\neg(objekt))} \tag{4.3}$$

Pokud X obsahuje $m$ pozitivních a $n$ negativních výskytů a a průměrná plocha objektu na pozitivních i negativních obrázcích tvoří $s$, potom lze do rovnice dosadit:

$$P(X|objekt) = \frac{m}{m+n} \tag{4.4a}$$

$$P(object) = s \tag{4.4b}$$

$$P(X|\neg(object) = \frac{n}{m+n} \tag{4.4c}$$

$$P(\neg(object)) = 1 - s \tag{4.4d}$$

Váha určuje, jak důvěryhodný je záznam slovníku. Tedy váha = 1 znamená, že se střed shluku objevuje pouze na objektu, váha blížící se k nule znamená, že se střed vyskytuje často mimo objekt.

Vstupem je slovník, výstupem je slovník s váhou u jednotlivých záznamů.

### 4.1.5 Redukce slovníku

Slovník většinou obsahuje záznamy s několika málo výskyty a záznamy s nízkou vahou. Pokud záznam obsahuje málo výskytu, znamená to, že se nejedná o typickou část objektu. Tyto záznamy tedy budou s velkou pravděpodobností pouze zpomalovat detekci objektu a proto mohou být vymazány. Záznamy s nízkou vahou znamenají, že ač se může jednat o běžnou součást objektu (např. jednobarevnou plochu), tato část se zcela běžně vyskytuje i v okolí objektu. Tyto záznamy neposkytují žádné rozhodující informace, jejich zachování či vymazání tedy nezmění kvalitu detekce. Minimální počet výskytů a váha zavisí na počtu středů shluků a prahu podobnosti. Při obvyklém nastavení těchto parametrů by nemělo být při redukci odstraněno více než 5% až 10% záznamů.

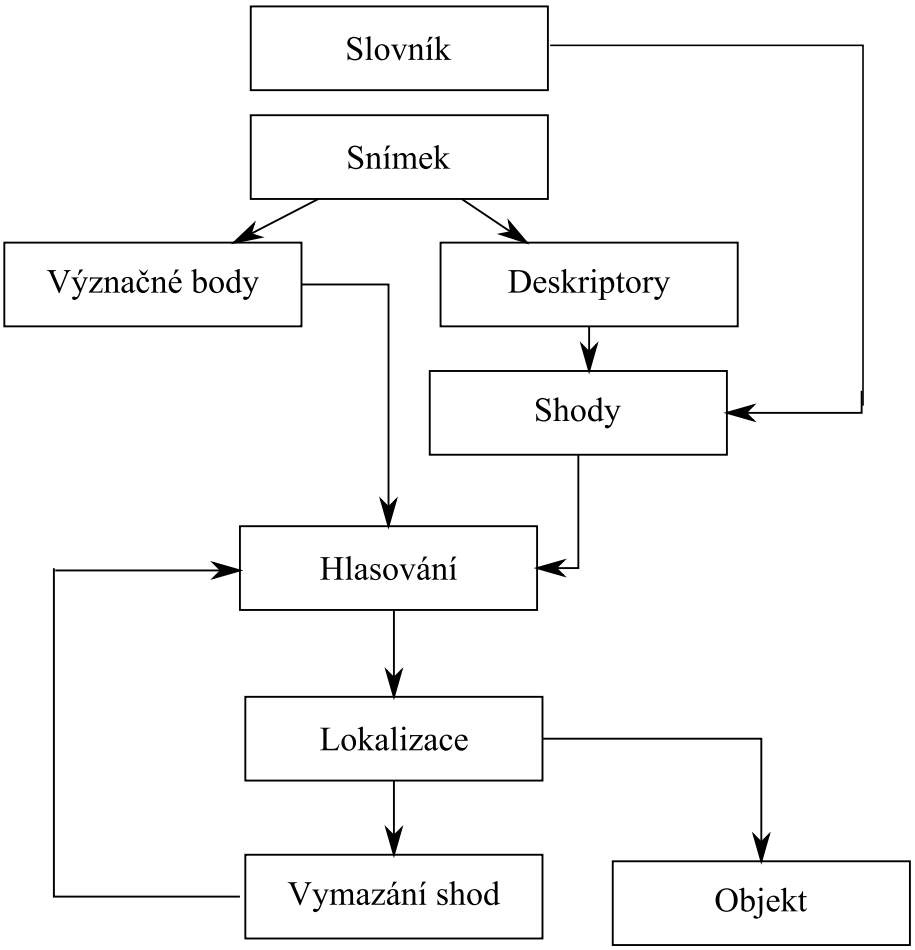
Vstupem je slovník, výstupem je slovník bez nedůležitých záznamů.

**Algorithm 3** Tvorba slovníku

**for all** trénovací obrázky **do**
  Detekce význačných bodů
  $(c_x, c_y) \leftarrow$ střed objektu
  **for all** význačný bod $l_k = (l_x, l_y, l_s)$ a jeho deskriptor $f_k$ **do**
    **if** $l_k \in objekt$ **then**
      $L_p \leftarrow L_p \cup (c_x - l_x, c_y - l_y, l_s)$ // Význačné body na objektu
      $F_p \leftarrow F_p \cup f_k$ // Deskriptory na objektu
    **else**
      $L_n \leftarrow L_n \cup l_k$ // Význačné body mimo objekt
      $F_n \leftarrow F_n \cup f_k$ // Deskriptory mimo objekt
    **end if**
  **end for**
**end for**
Shlukování $F_p$ pro získání středů $C$
$ISM \leftarrow |C|$ // Inicializace slovníku
**for all** středy $C_i$ **do**
  $ISM[i].C \leftarrow C_i$; $ISM[i].Occ \leftarrow \theta$; $ISM[i].w \leftarrow 0$
**end for**
**for all** $Fp_k$ **do** // Přidávání výskytů
  **for all** $ISM_i$ **do**
    **if** $sim(ISM[i].C, Fp_k) \leq t$ **then**
      $ISM[i].Occ \leftarrow ISM[i].Occ \cup (L_k)$
    **end if**
  **end for**
**end for**
**for all** $Fn_k$ **do** // Negativní body
  **for all** $ISM_i$ **do**
    **if** $sim(ISM[i].C, Fn_k) \leq sim$ **then**
      $ISM[i].w \leftarrow ISM[i].w + 1$
    **end if**
  **end for**
**end for**
**for all** $ISM_i$ **do** // Výpočet váhy
  $P_{neg} = ISM[i].w$
  $ISM[i].w \leftarrow \frac{\frac{|ISM[i].Occ|}{P_{neg}+|ISM[i].Occ|}s}{\frac{|ISM[i].Occ|}{P_{neg}+|ISM[i].Occ|}s+\frac{p_{neg}}{P_{neg}+|ISM[i].Occ|}(1-s)}$
**end for**
**for all** $ISM_i$ **do** // Redukce slovníku
  **if** $ISM[i].w < tt$ **or** $|ISM[i].Occ| < MinPocet$ **then**
    $ISM \leftarrow ISM - \{ISM[i]\}$
  **end if**
**end for**

## 4.2 Detekce objektů

Detekce začíná načtením vytvořeného slovníku a průměrné velikosti objektu. Dále pak načtením videa ze souboru, streamu z připojené videokamery nebo obrázku ze souboru. Video je děleno na jednotlivé snímky a každý snímek je zpracováván samostatně. Na snímku se detekují význačné body, porovnají se se slovníkem a provede se hlasování do hlasovacího prostoru. Dostatečně silná maxima hlasovacího prostoru jsou prohlášena za středy objektů. Hledání maxim v hlasovacího prostoru je iterativní. Tento přístup je inspirován [7], avšak zjednodušen. Princip detekce je nastíněn ve schématu na obrázku 4.3.




Obrázek 4.3: Datový tok lokalizace objektu.

### 4.2.1 Hlasování

Na snímku se detekují význačné body, vypočtou se jejich deskriptory a provede se případná PCA. Typ deskriptoru a dimenzionalita PCA se musí schodovat s hodnotami použitými ve slovníků. Deskriptor každého bodu je porovnáván se středy shluků ve slovníku. Pokud podobnost přesáhne práh, pak je záznam slovníku považován za aktivovaný. Aktivované záznamy zapisují své hlasy pro střed dle rovnic 4.5 až 4.7 do diskretizovaného hlasovacího prostoru (akumulátor, Discretized Vote Space, DVS). DVS je třírozměrný (pozice objektu

(x,y) a velikost(s) objektu).

$$s_{hlas} = s_{VyznacnyBod}/s_{VyskytZaznamu} \tag{4.5}$$

$$x_{hlas} = x_{VyznacnyBod} - x_{VyskytZaznamu}s_{hlas} \tag{4.6}$$

$$y_{hlas} = y_{VyznacnyBod} - y_{VyskytZaznamu}s_{hlas} \tag{4.7}$$

Pro detekci objektů s možností jejich rotace by bylo nutné použít 4D hlasovací prostor, kde 4. rozměr představuje natočení objektu. Každý z binů DVS také uchovává spojité hodnoty všech hlasů, které do něj naleží. Uchování originálních hodnot je důležité pro přesné dohledání objektu. Inkrementace DVS není konstantní hodnotou ale váhou konkrétního hlasu. Váha hlasu vychází z pravděpodobnostního modelu. V [29] je suma váh všech hlasů jednoho význačného bodu rovna jedna, váha každého aktivovaného záznamu definována jako $\frac{1}{|N|}$ a váha jednoho hlasu jako $\frac{1}{|N|} * \frac{1}{|Occ|}$. $N$ je počet záznamů, které aktivoval význačný bod, $|Occ|$ je počet hlasů záznamu. Nezáleží tedy, zda slovník obsahuje duplicitní záznamy, protože výsledná váha zůstane stejná. V případě využití negativních bodů se váha jednoho hlasu mění na $\frac{1}{|N|} * \frac{w}{|Occ|}$, kde $w|w \leq 1$ je váha záznamu.

### 4.2.2 Lokalizace

Algoritmus detekce objektů vychází z [7]. Jedná se o iterativní hlasování a následné procházení akumulátoru, viz schéma 4.4 a algoritmus 4. Základní myšlenka tohoto přístupu spočívá v tom, že jeden význačný bod může ležet na maximálně jednom objektu a pro tento objekt také korektně hlasovat. Postup je tedy opakované hlasování do akumulátoru, lokalizace globálního maxima a vymazání takových záznamů ze seznamu shod, jejichž aktivační význačný bod hlasoval pro právě nalezené globální maximum. Tento cyklus běží dokud nalezené maximum je větší než stanovený práh. Postupné odebírání hlasů je zřetelné na 4.5.

V [7] je velikost binu akumulátoru 1px a tedy každý pixel obrazu je považován za potenciální střed objektu. To je možno díky hustému (dense) rozložení význačných bodů. Práce však ukazuje rychlou degradaci detekce při snižování hustoty význačných bodů.

V této práci požívám řídkou (sparse) detekci význačných bodů. Hlasy pro střed jednoho objektu se však nescházejí v 1 pixelu ale v ploše cca 15 pixelů. Proto používám velikost binu akumulátoru větší než jeden pixel (viz výsledky při změně velikosti binu hlasovacího prostoru na obrázu 5.19).

Protože velikost binu větší než několik pixelů způsobuje nepřesné určení obejktu, je počítáno těžiště binu ze spojitých hodnot hlasů náležících tomuto binu.

**Algorithm 4** Lokalizace objektů pomocí iterativního hlasování

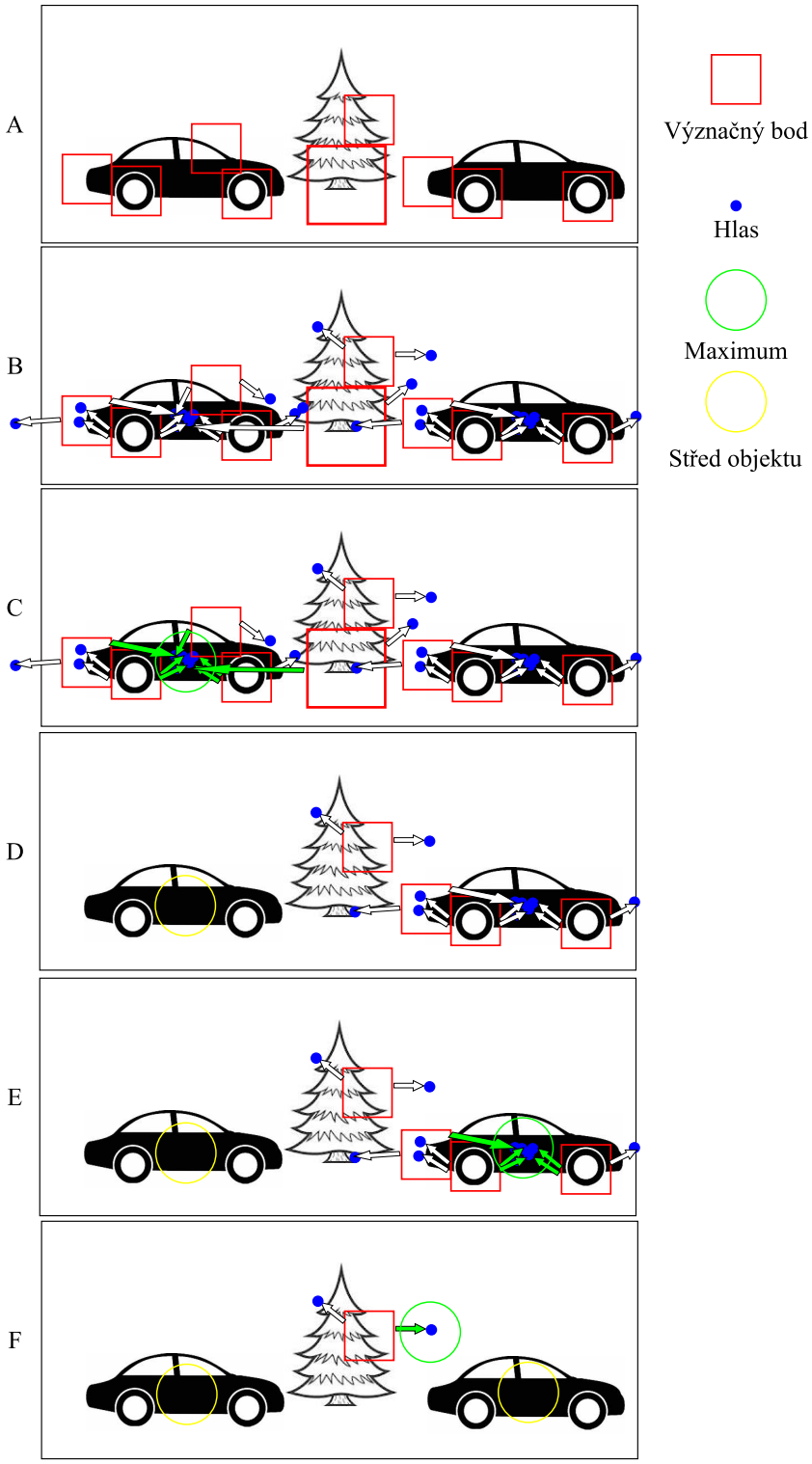
Extrakce význačných bodů $L$ s deskriptory $F$
Vytvoření seznamu shod $M$ // Viz algoritmus 5
$Objects \leftarrow \theta$ // Pozice objektů
**repeat**
  Hlasování // Viz algoritmus 6
  Lokalizace globálního maxima $MaxVal$ na pozici $MaxPos$ v akumulátoru.
  **if** $MaxVal > t$ **then**
    $Objects \leftarrow Objects \cup MaxPos$
    $M \leftarrow M - \{$záznamy s význačnými body hlasující do $MaxPos$ $\}$
  **else**
    break;
  **end if**
**until true**

**Algorithm 5** Tvorba seznamu shod

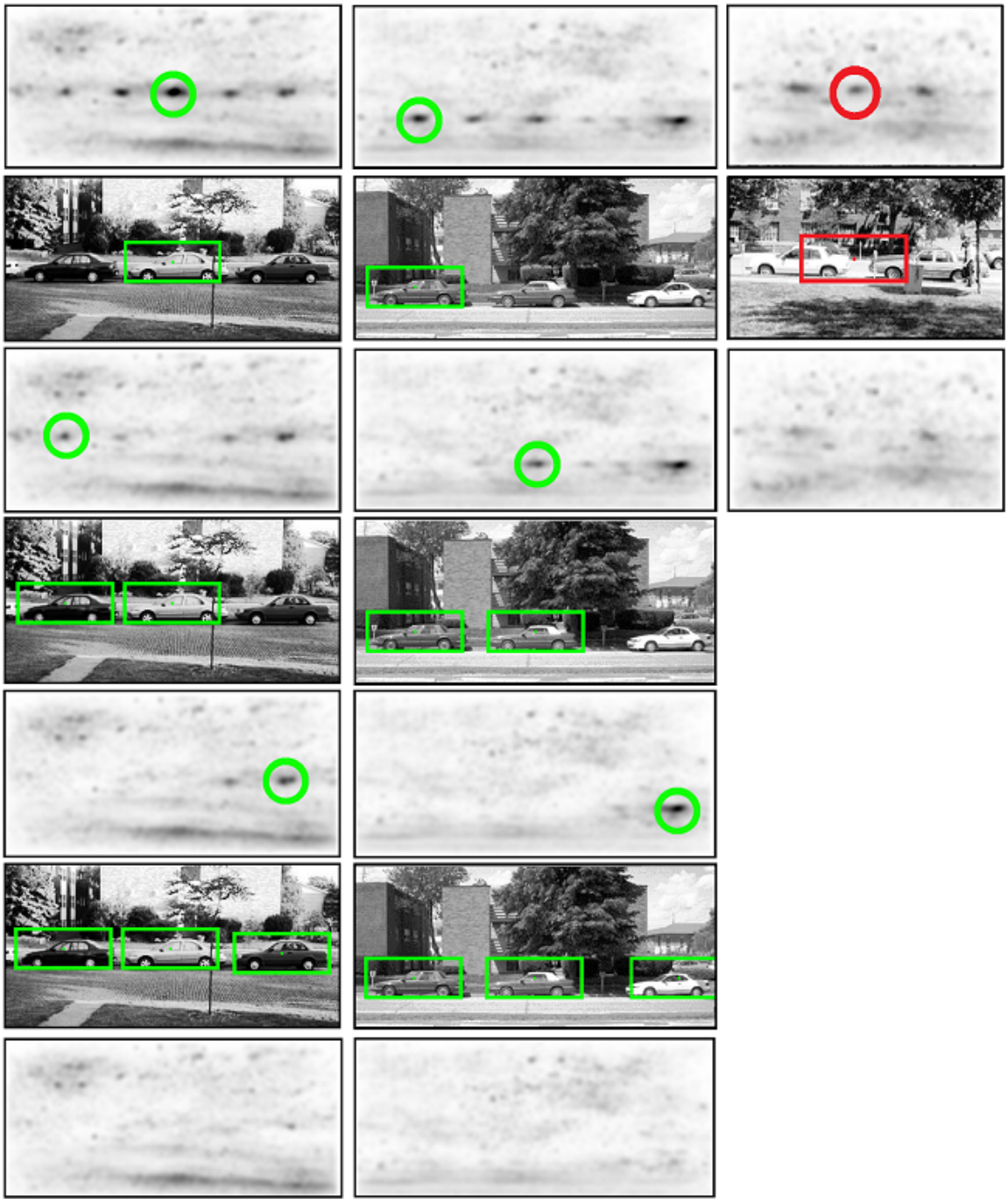
$counter \leftarrow 0$; $total \leftarrow 0$
**for all** deskripror $f_k \in F$ **do**
  **for all** střed $c_l \in ISM$ **do**
    **if** $\mathrm{sim}(f_k, c_l) > t$ **then**
      $M \leftarrow M \cup (k, l, 0)$ // Přidání nalezené shody do seznamu
      $counter++$
    **end if**
  **end for**
  **for** $i = total$ **to** $|M|$ **do**
    $M[i].weight = 1/counter$ // Nastavení váhy
  **end for**
  $total \leftarrow total + counter$
  $counter \leftarrow 0$
**end for**

**Algorithm 6** Hlasování

**for all** Hlasy $m_j(l_k, ISM_l, weight) \in M$ **do** // Shoda bodu $k$ se záznamem $l$
  **for all** $occ \in ISM[l].Occ$ **do**
    $x \leftarrow (l_x - occ_x \frac{l_s}{occ_s}, l_y - occ_y \frac{l_s}{occ_s}, \frac{l_s}{occ_s})$
    **if** $x \notin DVS$ **then**
      $continue$ // Hlasy mimo hlasovací prostory
    **end if**
    $p(o_n, x|C_i, l) \leftarrow \frac{ISM[i].w}{|ISM[i].Occ|}$ // Váha výskytu
    $w \leftarrow p(o_n, x|C_i, l)weight$
    $intX \leftarrow$ Interpolace $x$ do souřadnic akumulátoru
    $DVS[intX] \leftarrow DVS[intX] + w$
    $Votes[intX] \leftarrow Votes[intX] \cup (x, w, k)$ // $x$ a $w$ pro možnost dohledání přesného
      // maxima, $k$ pro odečítání hlasů.
  **end for**
**end for**




Obrázek 4.4: Princip iterativního hlasování. A) Lokalizace význačných bodů; B) Hlasování; C) Detekce globálního maxima (1. iterace); D) Vymazání význačných bodů hlasujících pro maximum; E) Detekce globálního maxima (2. iterace); F) Vymazání význačných bodů hlasujících pro maximum. Další maximum nepřesahuje práh, lokalizace končí.



Obrázek 4.5: Ukázka iterativního odebírání hlasů. Levý a střední sloupec: Požadované chování. Pravý sloupec: Chybné chování způsobené globálním maximem vytvořeným z hlasů dvou objektů.

## 4.3 Sledování objektů

Sledování je primárně řešeno detekcí objektů snímek po snímku, bez návaznosti mezi jednotlivými snímky.

Je možné povolit detekci pouze na každém $m$-tém snímku. Pro každý nalezený objekt je spočítán histogram barev tohoto objektu. V každém následujícím snímku (až do $m$-tého) jsou spočínány zpětné projekce obrázku pro každý histogram (potlačení oblastí s barvami nevyskytujícími se na objektu, zvýraznění oblastí s barvami vyskytujícími se na objektu) a označeny kompaktní zvýrazněné oblasti. Tento algoritmus je CamShift upravený pro detekci více objektů.

Další možností je sledování objektů pomocí Kalmanova filtru. Pozice objektů je na snímcích bez detekce odhadována pouze z pozice, rychlosti a zrychlení objektů, obsah snímku není brán v úvahu. Snímky s detekcí korigují pozici, rychlost i zrychlení. Je nutné detekovat objekty častěji než u CamShiftu.

# Kapitola 5

# Experimenty

## 5.1 Implementace

Projekt je implementován v jazyce C++, použity jsou knihovny *OpenCV_2.3.1* pro práci s obrazem a *dirent* pro přístup k adresářům. Starší verze OpenCV nejsou použitelné kvůli rozdílné implementaci shlukovací funkce kmeans(). Dirent.h je standartní linuxová knihovna, na Windows je potřeba explicitně dodat (je součástí zdrojových textů projektu). Projekt je přeložitelný na platformách Linux a Windows. Překlad je možný pomocí přiloženého makefile nebo .sln pro MS Visual Studio 2010. Projekt je konzolová aplikace bez možnosti zásahu do běhu programu po jeho spuštění. Veškerá nastavení a možnosti jsou načteny z konfiguračního souboru. Cesta k němu je jako jediný argument požadována při spuštění programu. Dalšími volitelnými argumenty jsou pak možnosti změny hodnot konfiguračního souboru, zvláště pak testovaného obrázku / videa. Další informace o parametrech programu viz programovou dokumentaci.

## 5.2 Datové sady

Protože detektor navržený v této práci musí být univerzální, byl testován na několika odlišných datových sadách [1]. Všechny testy jsou vyhodnoceny na základě vyhodnocovacího schéma použitém v [1] založeném na překrývání obalových boxů: hypotéza se středem $(x,y)$ a velikostí$(w,h)$ je porovnána s anotovaným obalovým boxem $(w^*,h^*)$ se středem ve$(x^*, y^*)$ a pokud splňuje

$$\frac{|x - x^*|^2}{(\alpha w^*)^2} + \frac{|y - y^*|^2}{(\alpha h^*)^2} + \frac{|w - w^*|^2}{(\alpha w^*)^2} \leq 1 \tag{5.1}$$

je považována za korektní. Je akceptována pouze jedna hypotéza na jeden anotovaný box, každá další hypotéza na tento box je označena jako False positive. Hodnota $\alpha$ je rovna 0.5 pro všechny sady (při vyhodnocování EER u *UIUC cars side* je $\alpha = 0.25$).

[1] Odkazy na datové sady, aplikace včetně zdrojových souborů, slovníky a výsledky testů jsou k dispozici na http://medusa.fit.vutbr.cz/wiki/index.php/Detekce_objektu_pomoci_vyznacnych_bodu

ETHZ Cars

Tato sada[2] byla vytvořena pro detekci objektů v [29] a obsahuje 50 fotografií bočních pohledů na automobil. Polovina fotografií je unikátních, druhá polovina je pořízena horizontálním překlopením. Automobily tedy "jedou"vždy na obě strany obrázku. Součástí sady jsou anotace a masky. Sada slouží jako trénovací.

Obrázek 5.1: Příklad obrázků z *ETHZ Cars*.

UIUC Cars (Side)

Sada[3] byla vytvořena pro detekci objektů v [1]. Trénovací sada obsahuje 500 obrázku bez objektů (letadla, zvířata, budovy, krajinky a další) a 550 obrázku s objektem. *UIUC Cars* obsahuje 2 testovací sady, *Singlescale test set* obsahující 170 obrázků s 200 objekty v jednotné velikosti. *Multiscale test set* obsahuje 108 obrázků s 139 objekty různých velikostí. Obě sady obsahují částečně viditelné automobily, automobily s nízkým kontrastem vůči pozadí i obrázky s vysoce proměnlivým pozadím. Trénovací i testovací sady byly zvětšeny 2.4 krát, protože původní nízké rozlišení nevyhovovalo nastavení detektorům význačných bodů. Zvětšení pomocí interpolace nepřidá žádné informace a tedy nemá vliv na samotnou detekci. Sada obsahuje anotace, slouží jak pro trénování, tak pro testování.

CalTech Cars (Rear) - Marcus

Sada[4] obsahuje 126 fotografií zadní části automobilu. Z tohoto pohledu je objekt pouze jednobarevnou kompaktní plochou, vyjímku tvoří poznávací značka a zadní světla (u každého automobilu jiná). Díky této jednobarevnosti jsou objekty těžko detekovatelné, protože je na objektech detekováno výrazně méně význačných bodů než na pozadí. Sada obsahuje anotace a masky, slouží jako trénovací.

CalTech Cars (Rear) - Brad

Sada obsahuje 526 fotografií pořízených z pohybujícího se automobilu při běžném provozu. Výzvou této sady je detekce objektů při výrazné změně jejich velikosti. Tato sada neobsahuje anotace ani masky, původně sloužila pro testování přítomnosti/nepřítomnosti objeku.

[2]Informace o sadě ETHZ Cars na http://www.vision.ee.ethz.ch/~bleibe/data/datasets.html#cars-side

[3]Další informace o sadě UIUC cars na http://cogcomp.cs.illinois.edu/Data/Car/

[4]Obě sady CalTech Cars rear a CalTech motorbikes jsou k dispozici na http://www.vision.caltech.edu/html-files/archive.html

Obrázek 5.2: Příklad obrázků z *UIUC Cars (Side)*. Horní řada: Trenovací pozitivní a negativní obrázek . Spodní řada: Testovací obrázky

Obrázek 5.3: Příklad obrázků z *CalTech Cars (Rear) - Marcus.*

### CalTech Motorbikes

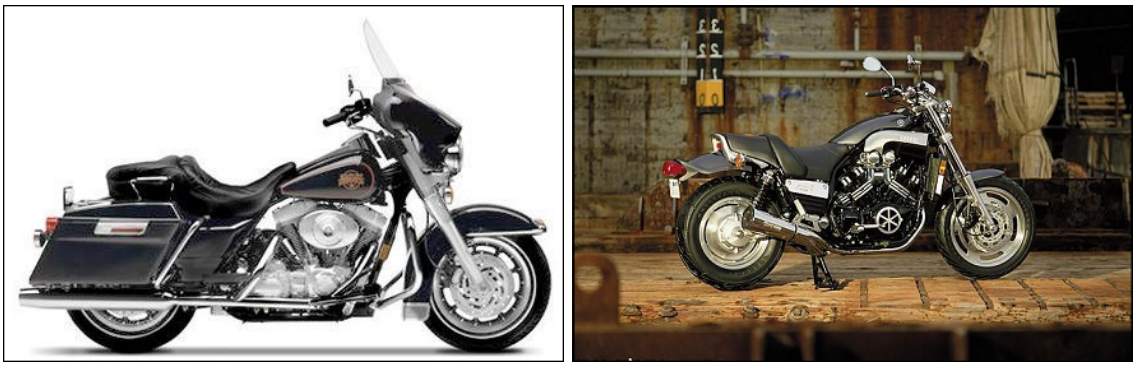
Sada byla použita pro detekci v [17] a obsahuje 335 obrázků s bočním pohledem na motocykl. Pozadí téměř poloviny obrázků je konstantní (bílé, světle modré, černé), zbytek obsahuje reálné pozadí (silnice, krajina). Sada obsahuje pouze anotace a slouží jako trénovací. Pro natrénování detektoru používám pouze obrázky s konstantním pozadím (kvůli porovnatelnosti s ISM[29]).

### ETHZ Motorbikes

Sada[5] obsahuje 115 obrázků obsahující 125 anotovaných motocyklů viditelných převážně z boku. Některé objekty jsou pouze částnečně viditelné. Sada obsahuje i obrázky s mnoha objekty, které nejsou anotované. Sada obsahuje anotace a slouží jako testovací.

[5] ETHZ Motorbikes na http://www.vision.ee.ethz.ch/~bleibe/data/datasets.html#tud-motorbikes

Obrázek 5.4: Příklad obrázků z *CalTech Cars (Rear) - Brad.*



Obrázek 5.5: Příklad obrázků z *CalTech Motorbikes.*

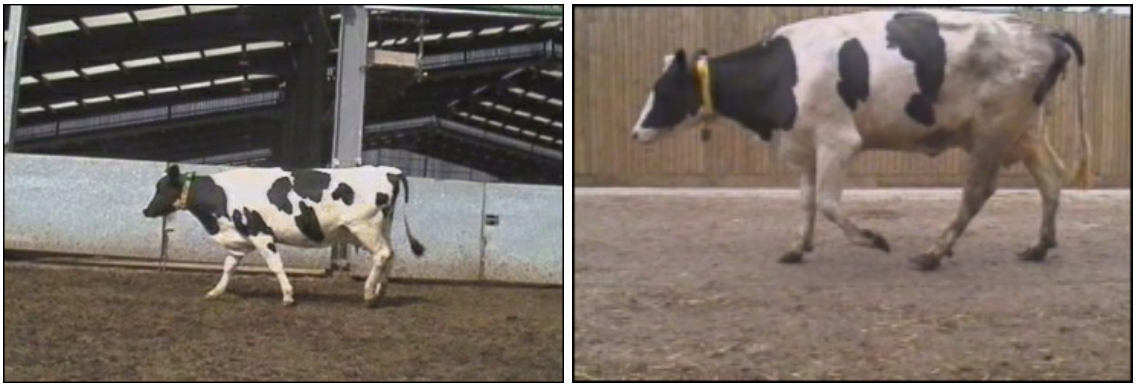
Obrázek 5.6: Příklad obrázků z *ETHZ Motorbikes.*

### Leeds Cows

Sada[6] byla použita v [37]. Trenovací sada obsahuje 111 obrázků krav. Do trénovací sady jsem dodal několik obrázků bez hledaného objektu. Testovací data byly získány jako jednotlivé snímky ze 14 videosekvencí. Sada obsahuje přes 2200 obrázků s více jak 1600 kravami viditelnými z více jak 50%. Některé snímky obsahují šum a mají zřetelné artefakty MPEG-komprese. Trenovací sada obsahuje masky. Testovací sada neobsahuje anotace ani masky,

[6] k dispozici na http://www.vision.ee.ethz.ch/~bleibe/data/datasets.html#leeds-cows

proto není provedena přesná evaluace výsledků. Sada je zahrnuta pro ukázku "univerzálnosti"detektoru. Výsledky detekce krav na testovací sadě jsou zahrnuty v prezentačním videu.



Obrázek 5.7: Příklad obrázků z *Leeds Cows*.

TUD 210 Pedestrians

Sada[7] byla použita v [2] a obsahuje 210 fotografií chodců viditelných z profilu. Fotografie jsou pořízeny pouze před 2 různými pozadími. Tato sada obsahuje anotace i masky, slouží jako trénovací.

Obrázek 5.8: Příklad obrázků z *TUD 210 Pedestrians*.

TUD Crossing Pedestrans

Sada obsahuje 201 fotografií chodců na silničním přechodu s celkem 1008 anotovanými chodci. Sada obsahuje davové scény s částečně viditelnými neanotovanými chodci. Fotografie v jsou rozmazány kvůli dlouhé expozici při fotografování. Sada obsahuje anotace a slouží jako testovací.

[7] Další informace o TUD Pedestrians a TUD Crossing na https://www.d2.mpi-inf.mpg.de/node/382

Obrázek 5.9: Příklad obrázků z *TUD Crossing Pedestrans*.

PennFudan Pedestrans

Datová[8] byla použita pro detekci v [59]. Sada obsahuje 170 obrázků se 423 anotovanými chodci. Některé obrazy obsahují davové scény s mnoha neanotovanými, částečně viditelnými chodci. Díky tomu je detekce na této sadě velmi obtížná. Sada obsahuje anotace i masky. Slouží jako testovací.


Obrázek 5.10: Příklad obrázků z *PennFudan Pedestrans*.

## 5.3 Nastavení parametrů

Tato sekce popisuje možné nastavení parametrů tvorby slovníku a detekce objektů. Je vysvětlen jejich význam, rozsah hodnot a vliv na detekci. Pro každé nastavení programu je vypočítána závislost Recall - Precision a vypočtena plocha pod touto křivkou. Hodnota plochy je značena jako RPC. Tento údaj charakterizující detektor je poté zanesen do grafu

[8] Další informace o sadě na http://www.cis.upenn.edu/~jshi/ped_html/

vlivu parametru na detekci. RPC nabývá hodnot od 0 (nenalezen žádný objekt při libovolném počtu špatných detekcí) do 1 (nalezeny všechny objekty, žádná špatná detekce).

Všechny nastavení programu v této sekci jsou testována na UIUC single-scale test set. Průměrná velikost testovaných obrázků je 500x300 px, počítač obsahuje procesor Intel Core i3 o výkonu 2.1 GHz, paměťová naročnost detekce je max 50MB.

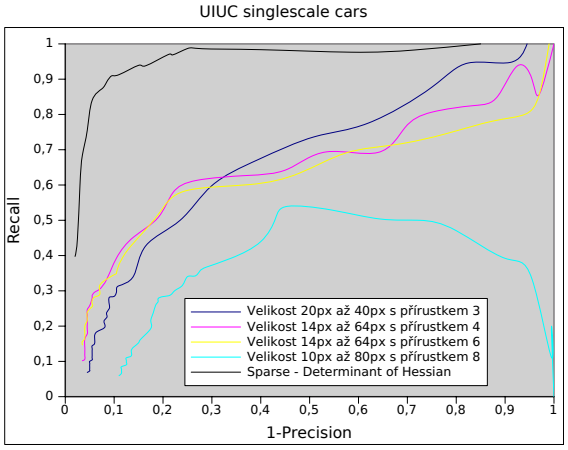
### 5.3.1 Detektor/deskriptor

První z testovaných možností programu je použití různých detektorů význačných bodů posaných v 2.1 a jejich deskriptorů 2.2. Prahy detektorů jsou nastaveny tak, aby bylo nalezeno cca 50000 význačných bodů patřících objektům. Z grafu 5.12 vyplývá, že nejvhodnější je použití detektoru DoH a deskriptoru SURF (oboje OpenCV implementace). SURF díky své nejnižší dimenzionalitě také umožnuje detekci objektů v nejkratším čase.

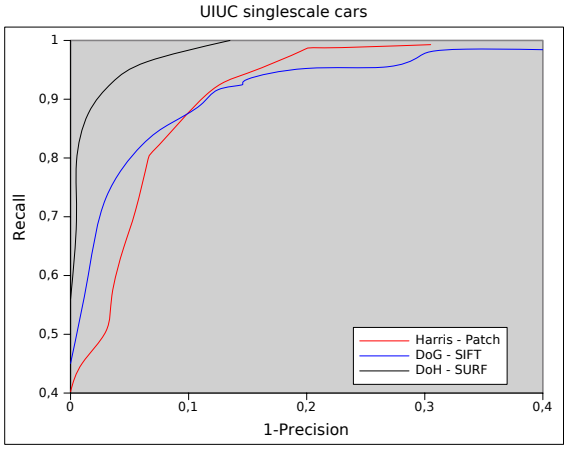
Kromě v práci zmiňovaných detektorů význačných bodů jsem vyzkoušel i Features from Accelerated Segment Test (FAST). Nedostatkem FAST detektoru byl proměnlivý počet detekovaných bodů u relativně podobných obrázků (70 až 700 bodů). Protože každý význačný bod zapisuje do hlasovacího prostoru hlasy o celkové váze 1 (při nepoužití negativních bodů), nelze při výše uvedeném rozsahu stanovit hodnotu detekčního prahu mající smysl pro celou testovací sadu.

V článcích [44, 20] je dosaženo vynikajících výsledků detekce objektů. V obou pracích je použit takzvaný Dense sampling [56, 54] význačných bodů, kdy není aplikována klasická detekce nastíněná v 2.1, ale obraz je souvisle pokryt překrývajícími se význačnými oblastmi. Díky tomu je zajištěn konstantní počet význačných bodů v obraze, nezávisle na osvětlení, konstantní barvě objektu či objektu barevně splývajícím s pozadím. Tento konstantní počet význačných bodů je výhodný pro nalezení a nastavení optimálního prahu detekce. Pokud však uvažujeme dense sampling pouze oblastmi konstantní velikosti, detektor poté není schopen nalézt různě velké objekty. Při užití samplingu postupně zvětšujícími se oblastmi je detekce pomalejší (UIUC singlescale cars: méně než 1 sekunda Sparse, 15 sekund Dense). Pokud bychom uvažovali i o dominantním natočení oblasti nutné pro rotačně-invariantní deskriptory, doba nutná pro detekci by vzrostla nad rámec použitelnosti. Graf 5.11 ukazuje kvalitu detekce při použití dense význačných bodů různou velikostí a hustotou rozložení. Překrytí oblastí význačných bodů je vždy 50%. Mimo SIFT a SURF deskriptory jsou implementovány i patch deskriptory (prostý výřez, výřez se změnou velikosti, výřez se změnou velikosti a natočení). Tyto patche však mají přiliž vysokou dimenzionalitu, proto shlukování i porovnání trvá nepřijatelně dlouhou dobu (oproti ostatním řešením). Shlukování Patchů trvá přibližně 1 hodinu oproti 1 minutě u SURF, detekce objektu 1 minutu oproti méně než 1 sekundě. Prodloužení doby výpočtu však nepřináší zlepšení detekce, viz obrázek 5.12. Řešením problému dimenzionality je použití PCA na Patch deskriptory. Mělo být dosaženo urychlení detekce a zkvalitnění shlukování (zanedbáním nadbytečného množství informací obsaženém v deskriptoru) za cenu mírného snížení kvality detektoru. Ačkoliv po implementaci došlo k výraznému urychlení celého procesu, zhoršení již tak špatné detekce činí toto řešení nepoužitelným. Patch deskriptory jsou tedy vhodné pouze na vizualizace jednotlivých funkcí programu.

V zbytku této práce (pokud není uvedeno jinak) je použit detektor DoH a deskriptor SURF.




Obrázek 5.11: Porovnání Sparse a Dense význačných bodů. U dense samlingu je uvedena nejmenší a největší velikost bodu a přírustek, s jakým jsou body mezi těmito hodnotami zvětšovány.



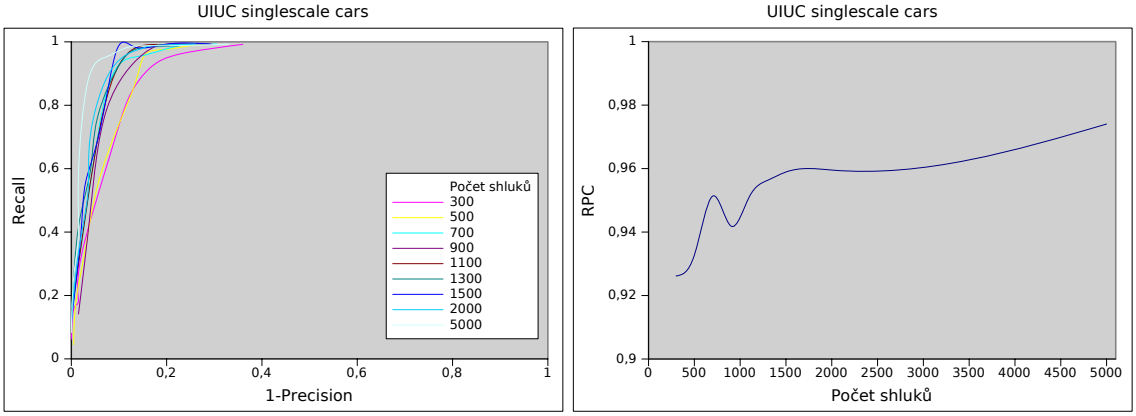

Obrázek 5.12: Porovnání nejlepších nastavení jednotlivých Detektorů/Deskriptorů.

### 5.3.2 Shlukování

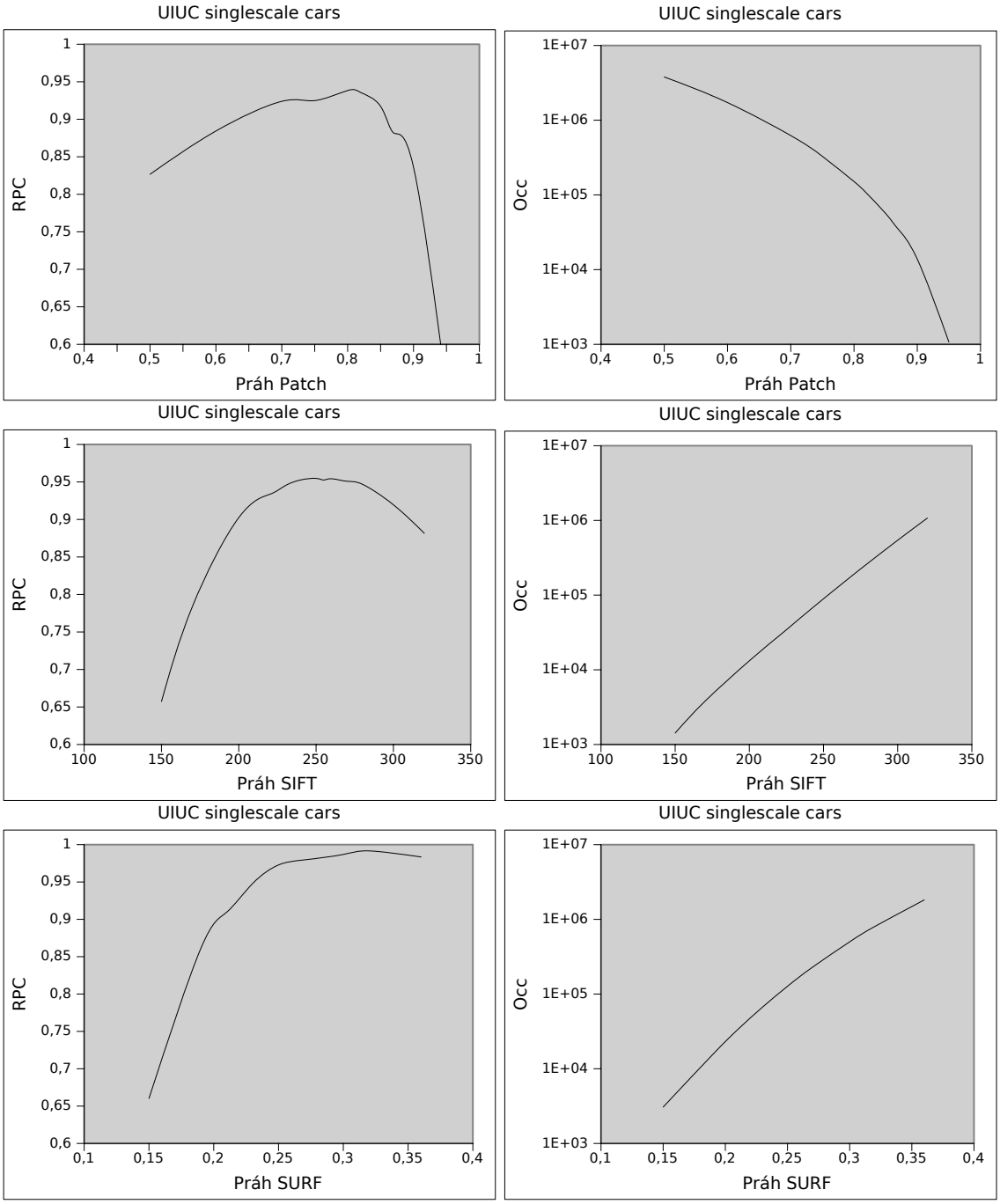
Sklukování deskriptorů je prováděno algoritmem k-means s inicializací k-means++. Dostatečný počet iterací pro nalezení středů je 15 až 20 při cca 10 různých inicializacích. Změny v detekci jsou minimální, se vzrůstajícím počtem shluků mírně stoupá i kvalita detekce. Bohužel však s přibývajícím počtem shluků vzrůstá i doba potřebná pro zpracování jednoho obrazu. Zatímco při použití 300 shluků trvá detekce 0.1 až 0.5 sekundy, při 5000 shlucích už se doba zpracování pohybuje mezi 2 až 5 sekundami.

### 5.3.3 Podobnost deskriptorů

Graf 5.14 na straně 42 ukazuje vliv prahu podobnosti na kvalitu detekce pro jednotlivé deskriptory. Hodnota podobnosti deskriptorů se využívá při sestavování slovníku i při porovnávání význačných bodů testovaného obrázku se záznamy slovníku. Toto možná není uplně nejštastnější řešení, protože při sestavovaní slovníků musí být práh vysoký, aby byl slovník kompaktní. Naproti tomu při vyhledávání podobných záznamů je vhodnější použít




Obrázek 5.13: Vliv změny počtu shluků na kvalitu detekce.

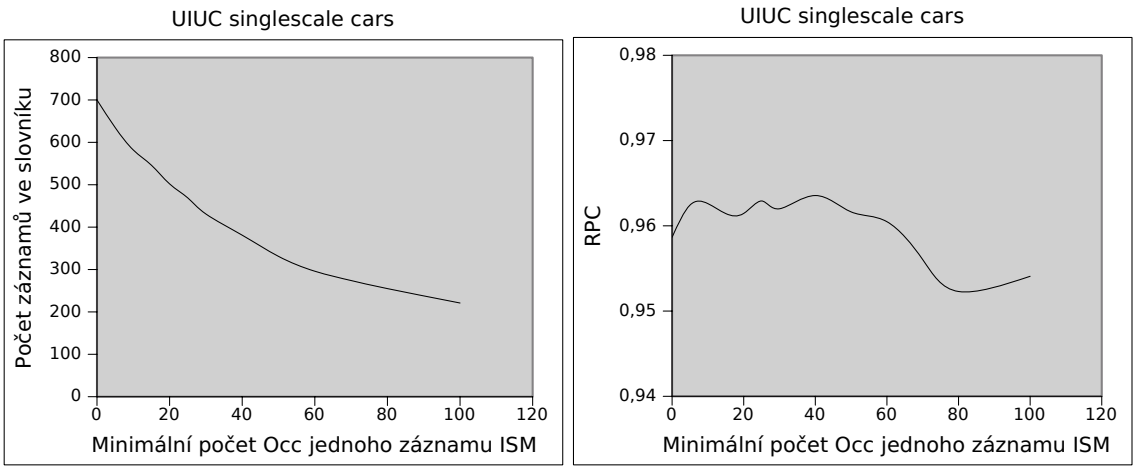



Obrázek 5.14: Kvalita detekce a celkový počet hlasů slovníku v závislosti na změně prahu podobnosti deskriptorů

benevolentnější práh aby byl zajištěn dostatečný počet hlasů v hlasovacím prostoru. Nastavení různých prahů pro sestavení slovníku a detekci objektů se tedy jeví jako vhodná volba.

### 5.3.4 Redukce slovníku

Graf 5.15 ukazuje zvýšení kvality detekce při redukci slovníku mezi 5% a 45% (700 záznamů → 650 až 380 záznamů). Při větší redukci kvalita detekce mírně klesá.




Obrázek 5.15: Vliv redukce slovníku na kvalitu detekce.

### 5.3.5 Scale factor

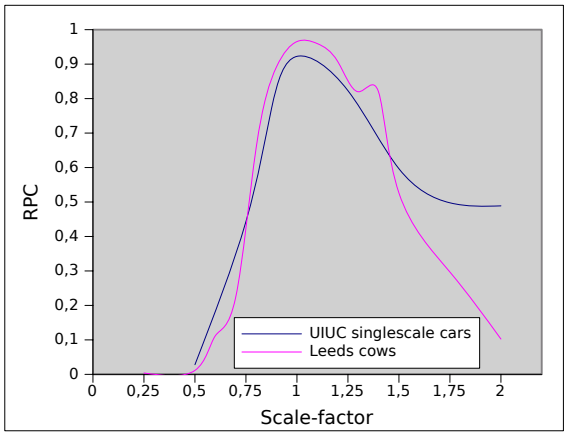
Detektor je navržen tak, aby byl schopen detekovat objekty různých velikostí. Z grafu 5.16 však vyplývá, že detektor je schopen lokalizovat pouze objekty o velikosti 75% až 150% trénovacích objektů. Tento rozsah je neočekávaně malý. Detekce různě velkých objektů závisí na velikosti hlasovacího prostoru (z-souřadnice určující hloubku) a nalezení odpovídajících význačných bodů u zvětšených obrázků.

Nastavení hloubky hlasovacího prostoru u předem známého poměru velikosti trénovacích a testovacích objektů je bezproblémové. Detektory naleznou odpovídající si body i u 9-krát zvětšených obrázků

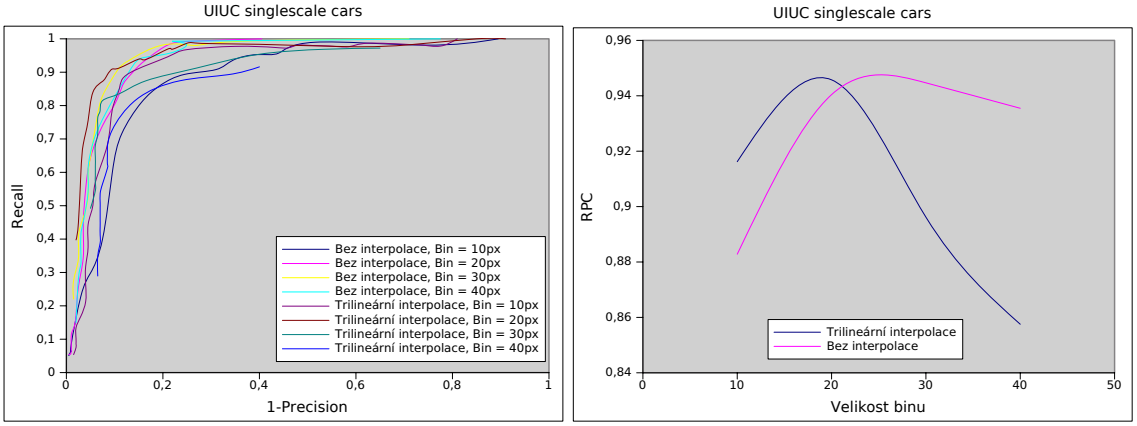
Nenalezení různě velkých obrázků tedy zřejmě znamená špatný návrh nebo implementaci aplikace.

### 5.3.6 Interpolace

Interpolace hlasů do DVS má smysl hlavně při použití 2-fázové lokalizace popsané v 5.3.9. Interpolaci v iterativním vyhledávání lze použít pouze pokud velikost binů DVS je natolik malá, že mezi středy dvou objektů leží nejméně 2 biny. V opačném případě totiž lokalizace 1 objektu zapříčiní vymazání význačných bodů potřebných pro lokalizaci dalších objektů, viz graf 5.17.




Obrázek 5.16: ScaleFactor



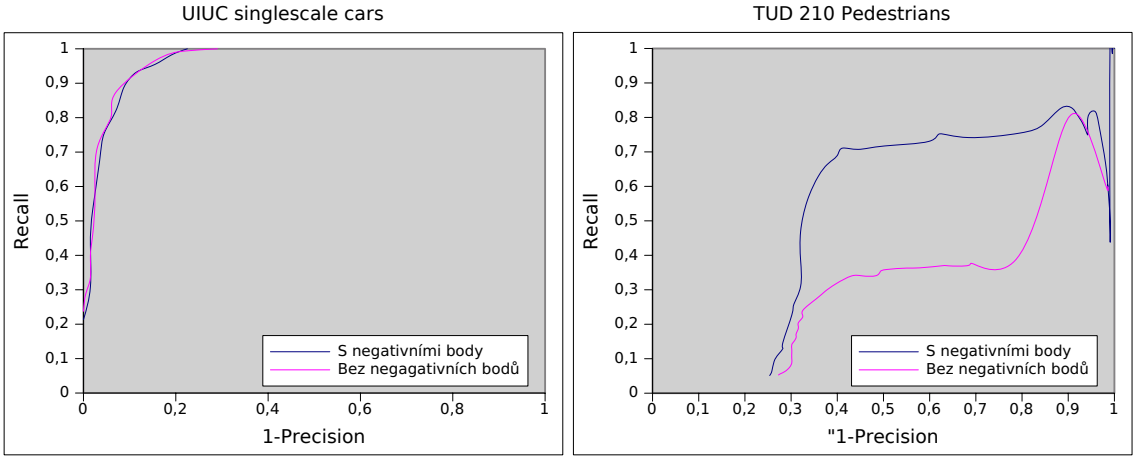

Obrázek 5.17: Vliv interpolace na iterativní detekci objektů.

### 5.3.7 Zahrnutí negativních bodů

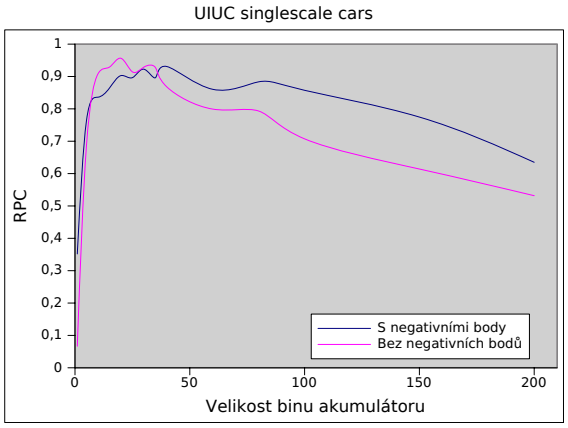
Graf 5.18 ukazuje nulový význam negativních bodů u trenovací sady *UIUC Singlescale Cars*, kde negativní obrázky představují nevyskytující se objekty v testovací sadě. Naproti tomu trenovací sada *TUD 210 Pedestrians* obsahuje chodce před stejným pozadím, před kterým se vyskytují i chodci v testovací sadě. Z grafu je patrné významné vylepšení detekce chodců. Pokud je možno do trénovací sady přidat obrázky s předpokládaným pozadím u testovacích dat, detekce objektu se výrazně zlepší.

### 5.3.8 Velikost hlasovacího prostoru

V ideálním případě by se měly všechny korektní hlasy pro střed objektu scházet do jediného bodu - pixelu. V praxi se však hlasy pro jeden objekt vyskytují na ploše cca 10 až 20 pixelů. Pro lepší detekci je proto vhodné použít hlasovací prostor (akumulátor) s velikostí binu větší než jeden pixel (viz graf 5.19). Zvětšováním velikosti binu hlasovacího prostoru klesá počet False positive. Pokud jsou však biny natolik velké, že se do jednoho binu vejde 2 a více středů objektů, klesá počet korektně nalezených objektů.




Obrázek 5.18: Rozdíl mezi použitím pouze pozitivních a pozitivních & negativních bodů.




Obrázek 5.19: Změna velikosti binu hlasovacího prostoru.

### 5.3.9 2-fázová lokalizace

Dříve než jsem implementoval iterativní lokalizaci, zkoušel jsem 2-fázové vyhledávní inspirované v [29]. První fáze je nalezení všech lokálních maxim v DVS. Hledání probíhá jako výběr binu s největší hodnotou vůči svému 26-okolí. Narozdíl od [29], kde jsou tato lokální maxima použita jako výchozí pozice pro Mean-Shift vyhledávání maxima ve spojitém prostoru, použil jsem výpočetně méně náročnější přístup. Pro každý bin DVS je uchováván seznam hlasů, které pro bin hlasovaly. Každý z hlasů binu lokálního maxima DVS (s možností zahrnutí hlasů od 8 okolních binů) přičtou do 2D prostoru 2D Gaussian s maximem = váha hlasu. Nalezené globální maximum tohoto 2D prostoru určuje pozici objektu.

Sekundární vyhledávání pozice objektu pomocí přičítání gaussianu má nevýhodu v nulovém zpřesnění velikosti objektu.

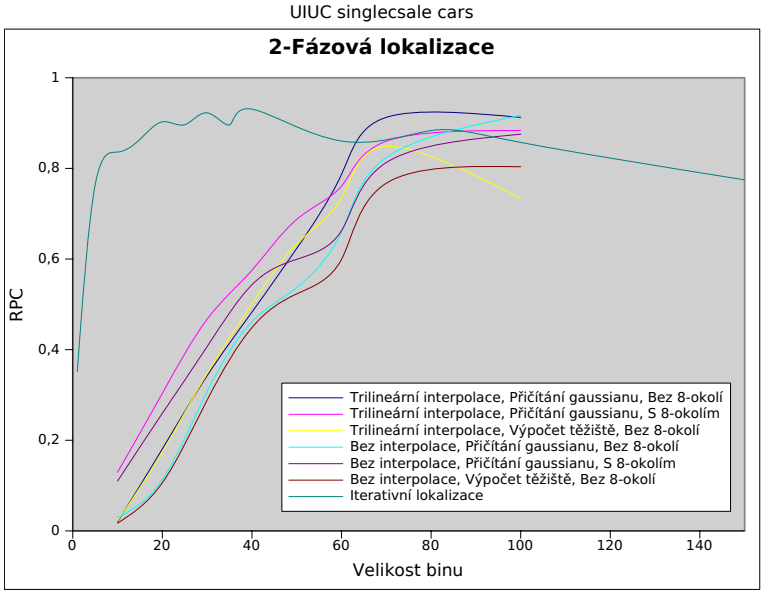
Další implementovanou metodou sekundárního vyhledávání je výpočet těžiště binu obsahující lokální maximum. Těžiště lze nalézt pomocí vzorce

$$CentreOfMass(x,y,z) = \left( \frac{\sum_{i=0}^{N} x_i w_i}{\sum_{i=0}^{N} w_i}, \frac{\sum_{i=0}^{N} y_i w_i}{\sum_{i=0}^{N} w_i}, \frac{\sum_{i=0}^{N} z_i w_i}{\sum_{i=0}^{N} w_i} \right) \tag{5.2}$$

Výsledky při zahrnutí 8-okolí maxima jsou nepoužitelné, pokud se v tomto prostoru vyskytuje více než jeden objekt. Bez zahrnutí 8-okolí však dosahuje tato metoda přesnějších výsledků.

2-fázové vyhledávání lze povolit v konfiguračním souboru, kde lze také nastavit interpolaci, typ sekundárního vyhledávání a zahrnutí hlasů z okolí maxima.

Graf 5.20 ukazuje dosažené výsledky při různých volbách lokalizace. 2-fázová lokalizace je velmi citlivá na kompaktnost hlasovacího prostoru. Pokud DVS není kompaktní je detekováno velké množství lokálních maxim. Kompaktnosti DVS pro použitelnost 2-fázové lokalizace je dosaženo použitím binů o velikosti cca 80 pixelů. Takováto velikost binu však neumožnuje detekovat objekty blízko u sebe. Z tohoto důvodu jsem navrh a implementoval iterativní hlasování s postupným odebíráním hlasů.




Obrázek 5.20: Výsledky různých možností 2-fázové lokalizace objektu. Pro porovnání je uvedena i iterativní lokalizace

## 5.4 Výslekdy detekce

UIUC Cars

Detekce na *UIUC Singlescale Cars* dosahuje výsledků srovnatelných s ostatními přístupy, viz tabulka 5.1. Při vyhodnocování korektní detekce bylo použito evaluační schéma z rovnice 5.1 s $\alpha = 0.25$. Toto zpřísnění vyhodnocovací funkce je nutné kvůli porovnatelnosti s ostatními přístupy. Sledovanou veličinou je Equal Error Rate (EER). ERR je hodnota, pro kterou platí Recall = Precision. Z tabulky vyplývá, že iterativní hlasování dosahuje mírně

| Metoda | Agarwal [1] | Garg [21] | Fergus [17] | ISM bez MDL[29] | **Tato práce** | ISM s MDL[29] | Mutch [41] |
|---|---|---|---|---|---|---|---|
| EER | ~79% | ~88% | ~88.5% | ~91% | **~91.5%** | ~97.5% | ~99.9% |

Tabulka 5.1: Srovnání výsledků detekce na UIUC singlescale car s ostatnímy přístupy z literatury.

lepších výsledků (91% vs. 91.5%) než lokalizace pomocí Mean-Shift trackingu ve spojitém prostoru hlasů.

Detekce na *UIUC Multiscale Cars* naráží na problémovou detekci objektů různých velikostí. Tabulka 5.2 ukazuje srovnání detektorů na multiscale sadě.

| Metoda | Agarwal | **Tato práce DoH+SURF** | ISM DoG+Patch | ISM HesLap+ShapeCont | Mutch |
|---|---|---|---|---|---|
| EER | ∼45% | **∼68%** | ∼85% | ∼95% | ∼90% |

Tabulka 5.2: Srovnání výsledků detekce na UIUC multiscale car s ostatnímy přístupy z literatury.

### TUD 210 Pedestrians

Rozdělením sady na polovinu jsem získal možnost otestovat vliv zahrnutí negativních bodů na detekci (viz obr. 5.18). Z grafu na obrázku 5.21 vyplívá, že se nepodařilo nalézt velkou část objektů i přes jednoduchost této sady. To pravděpodobně způsobuje konstantní pozadí obrázků. Význačné body, které obsahovali pouze malou část objektu a velkou část pozadí, byly detekovány i při nepřítomnosti objektu a hlasovali pro nekorektní pozice. Detekce na této sadě při dosahuje $EER \approx 72\%$ při ohodnocování záznamu slovníku pomocí výskytů mimo objekt. Při konstantní váze záznamů je pak $ERR \approx 40\%$.

### PennFudan Pedestrians

Při detekci na této sadě byl detektor natrénován na *TUD 210 Pedestrians*. Trénovací sada obsahuje chodce viditelné pouze z boku, zatímco testovací sada obsahuje davové scény s chodci viditelných ze všech stran. Na některých snímcích nelze ani rozeznat skutečný počet chodců. Detekce proto nedosahuje dobrý precision/recall poměr ($EER \approx 40\%$) ale obrázek 5.22 ukazuje ucházející detekci i v případě netriviálních obrázků.

### TUD Crossing Pedestrians

Při detekci na této sadě byl detektor natrénován na *TUD 210 Pedestrians*. Narozdíl od sady *PennFudan Pedestrians* obsahuje tato sada chodce natočené bokem v jednotné velikosti. Proto jsem očekával na této sadě lepší výsledky detekce. Bohužel se tak nestalo, viz obrázek 5.21 a tabulka 5.3. Příčinou nepřesvědčivých výsledků detektorů chodců je nejednoznačný obsah hlasovacího prostoru. Hlasovací prostory detekce automobilů 4.5 obsahují zcela jednoznačné maxima ve správných pozicích. Při detekci chodců (obrázky 5.22 a 5.26) tomu tak není. Na obrázku 5.26 je srovnání hlasovacího prostoru vytvořeného pomocí Houghova lesu a hlasovacího prostoru vytvořeného mojí aplikací pomocí slovníku. Toto srovnání ukazuje, že ač je slovník velmi intuitivní struktura, na detekci ve složitějších scénách se více hodí jiné metody.

| Metoda | **Tato práce DoH+SURF** | ISM HesLap+ShapeCont | Iterative HF [7] |
|---|---|---|---|
| EER | **∼22%** | ∼64% | ∼82% |

Tabulka 5.3: Porovnání výsledků detekce na TUD crossing pedestrians.

### CalTech Rear Cars - Markus

Tato sada měla sloužit jako pouze jako trénovací, ale protože se mi nepodařilo získat jinou

anotovanou sadu s pohledem na automobily ze zadu, rozdělil jsem ji na poloviny a použil jako trenovací i testovací. Ačkoliv se zdá tato sada jako triviální, ve skutečnosti je na zadní části automobilu pouze velmi málo význačných oblastí a proto se v akumulátoru často nenasčítá dostatečný počet hlasů odlišující objekt od pozadí. Detekce na této sadě dosáhla výsledku $EER \approx 67\%$.

CalTech Rear Cars - Brad

Tato sada slouží jako testovací pro detektory trénované na sadě CalTech Rear cars - Markus. Bohužel se mi nepodařilo sehnat k této sadě anotace a proto není provedena evaluace výsledků. Na obrázku 5.25 jsou ukázky detekce na této sadě.

TUD Motorbikes

Pro detekci na této sadě byl detektor natrénován na *CalTech Motorbikes* s konstantním pozadím. Detekce motocyklů dosahuje velice špatných výsledků (viz tabulka 5.4 a graf na obrázku 5.21). Příčina je pravděpodobně v bílém pozadí trenovacích obrázků. Tyto bílé oblasti se nevyskytují v testovacích obrázcích a proto nejsou ve slovníku nalezeny záznamy podobném význačným bodům testovaného obrázku. Díky tomuto se v hlasovacím prostoru nevytvoří dostatečně silné hypotézy. Řešením je použití nižšího prahu podobnosti, čímž lze dosáhnout většího počtu nalezených objektů i menšího počtu False positive. Druhým problémem trénovacích obrázků s konstantním pozadí je nulový počet negativních bodů. Díky tomu nelze určit, které záznamy slovníku jsou typickými částmi objektu a které se běžně vyskytují na pozadí.

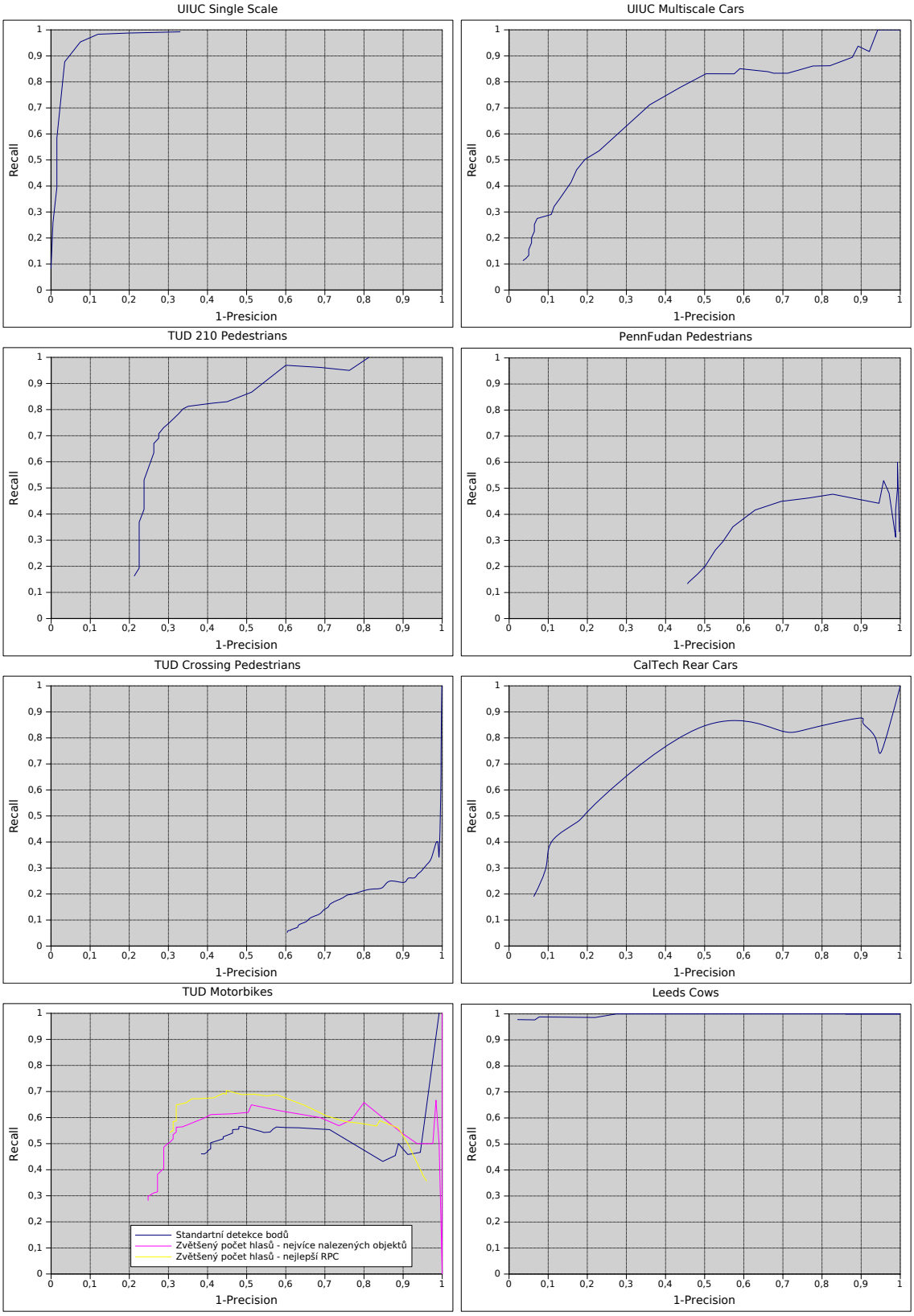
| Metoda | **Tato práce DoH+SURF** | ISM DoG+Patch | ISM HesLap+ShapeCont | ISM DoG+ShapeCont |
|---|---|---|---|---|
| EER | **∼60%** | ∼78% | ∼84% | ∼87% |

Tabulka 5.4: Porovnání výsledků detekce na TUD motorbikes s jinými přístupy.

Leeds Cows

Sada obsahuje anotace pouze k trénovacím datům, testovací data jsou použita jako zdroj pro prezentační video. Trénovací data byla rozdělena 20 trenovacích a 91 testovacích obrázků za účelem alespoň částečné evaluace detektoru na této sadě. Detektor dosáhl vynikajících výsledků ($EER \approx 95\%$), když špatně lokalizoval pouze 2 objekty. Příčinou tohoto úspěchu je pozadí vyskytující se na trénovacích i testovacích datech a také jednotný rozměr objektů.

Tato sada umožňuje stanovení detekovatelnosti objektu v závislosti na jeho viditelnosti. Z obrázků 5.24 je patrná detekovatelnost objektů s 60% viditelností. Detekce objektů se středem mimo obraz není možná, protože hlasy pro střed objektu mimo obraz nejsou brány v úvahu.



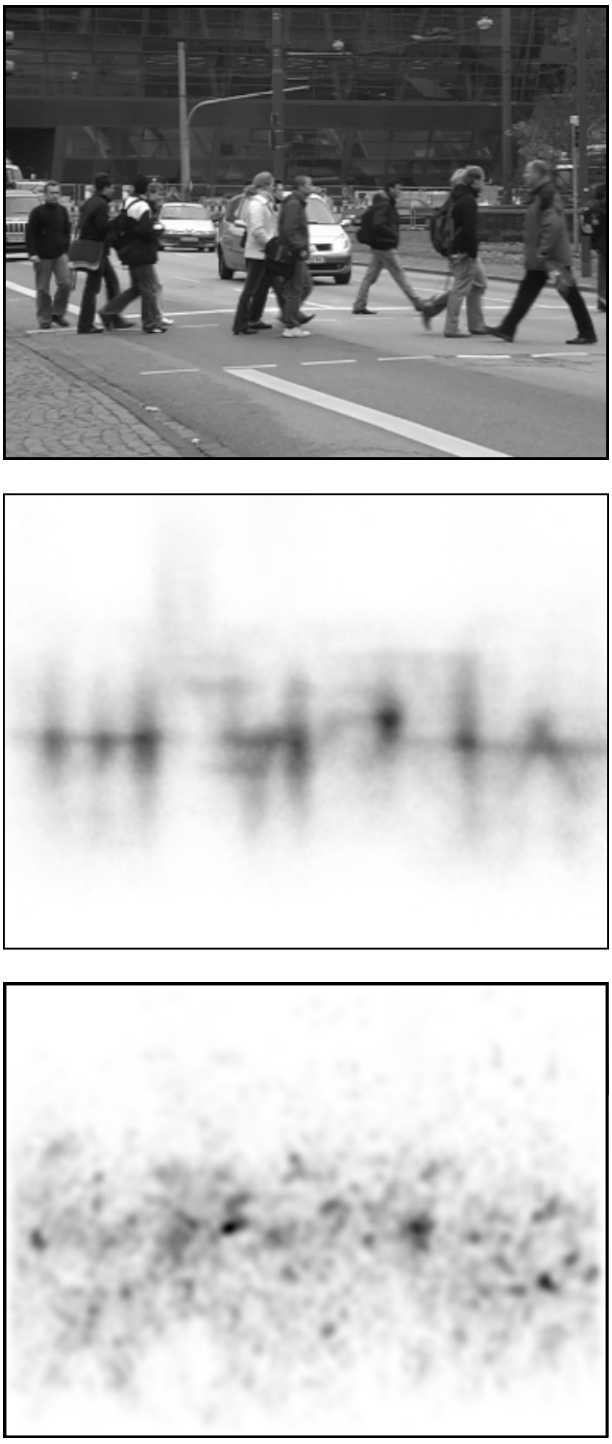

Obrázek 5.21: Porovnání detekce na různých sadách obrázků. UIUC singlescale cars, UIUC multiscale cars, TUD 210 pedestrians, PennFudan pedestrians, TUD Crossing pedestrians, CalTech rear cars, TUD motorbikes, Leeds Cows

Obrázek 5.22: Detekce chodců, PennFudan pedesttians. Zleva: originální obrázek, hlasovací prostor, výsledek detekce. Zeleně True positive, Červeně False positive, Modře False negative

Obrázek 5.23: Detekce chodců, TUD Crossing pedestrians.

Obrázek 5.24: Detekce krav. Objekt je detekovatelný již při 60% viditelnosti.

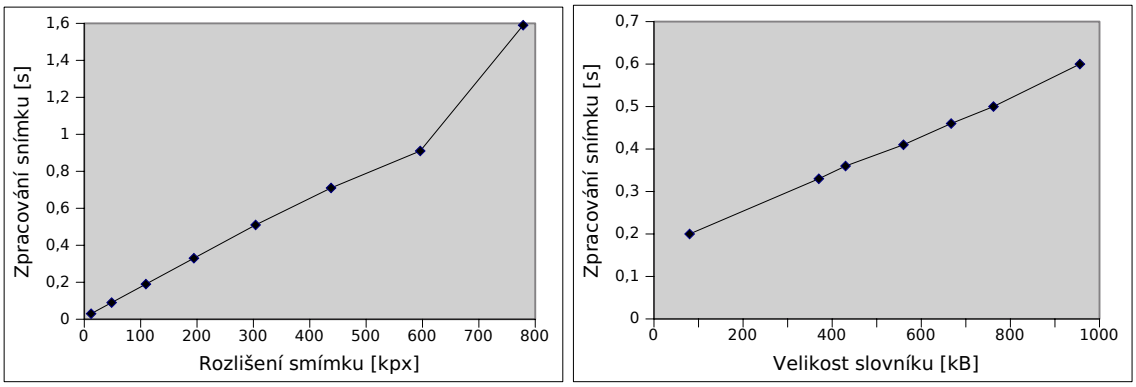
Obrázek 5.25: Detekce zádní části automobilů.



Obrázek 5.26: Srovnání hlasovacích prostorů. Nahoře: Originální obrázek. Uprostřed: Hlasovací prostor vytvořený pomocí Houghova lesu (Hough Forest transform) [7]. Dole: Hlasovací prostor vyvtvořený slovníkem.

## 5.5 Sledování objektů

Graf 5.27 zobrazuje průměrný čas potřebný ke zpracování jednoho snímku. Detektor byl natrénován na 210 snímcích *TUD 210 Pedestrians*, testovaní probíhalo na 170 snímcích *PennFudan Pedestrians*. Velikosti slovníku je měněna vzrůstajícím počtem shluků (38 až 710). Při měnícím se slovníku jsou použity testovací obrázky s průměrnou velikostí $491 \times 396$. Při změně velikosti testovacích obrázků je použit slovník o velikosti 370kB (224 shluků). Kromě




Obrázek 5.27: Doba potřebná ke zpracování jednnoho snímku.

výše zmiňované velikosti slovníku a rozlišení jsem vyzkoušel také vliv interpolace, velikost hlasovacího prostoru a práh podobnosti deskriptorů. Interpolace a velikost hlasovacího prostoru nemá žádný dopad na rychlost detekce. Práh podobnosti se projevuje pouze pokud slovník obsahuje podobné záznamy. V rozsahu hodnot prahů použitých v 5.14 je změna času zpracování jednoho snímku pouze 0.04 sekundy ($0.19 \rightarrow 0.23s$).

Detekce objektů snímek po smímku je možná, pouze pokud je detekce rychlejší než doba zobrazení jednoho snímku videa. Na testovacím počítači je detekce v reálnem čase možná pouze u snímku s rozličením cca $200 \times 120$ pixelů u videa s 25 fps.

Pro sledování objektů na snímcích s větším rozlišením je implementován CamShift tracking. Detekce je prováděna na každém $m$-tém snímků, na ostatní je aplikován sledovací algoritmus. $m$ lze nastavit v konfiguračním souboru. $m = 1$ provádí detekci snímek po snímku.

CamShift sleduje objekt dle jeho dominantí barvy, proto se tento algoritmus nedá použít například u sledování černobílých krav.

# Kapitola 6

# Závěr

V této práci jsem nastínil teorii nutnou k pochopení problematiky detekce a sledování objektů pomocí význačných bodů. Z této teorie jsem vybral vhodné informace k navržení vlastního přístupu. Podle návrhu jsem naprogramoval aplikaci, kterou jsem důkladně otestoval a porovnal s ostatními přístupy.

Z odborné literatury jsem získal informace o extrakci význačných bodů z obrazu a o možnostech popisu těchto bodů. Prostudoval jsem různé techniky detekce objektů pomocí význačných bodů. Zaměřil jsem se na detekci vycházející z Obecné Houghovy transformace, kdy objekty jsou popsany slovníkem ISM.

Další studovanou oblastí jsou techniky sledování objektů. Zaměřil jsem se na sledování bodů Kalmanovým filtrem a sledování oblastí Mean-Shift trackingem.

Navrhl jsem aplikace pro detekci a sledování objektů pomocí význačných bodů. Objekt je reprezentován slovníkem ISM rozšířeným o věrohodnostní váhy získanými z bodů pozadí. Detekce objektů vychází z GHT. Inovací je nevyhledávání lokálních maxim, ale iterativní vyhledávání globálního maxima s postupným vymazáváním hlasovacího prostoru. Sledování je realizováno detekcí na každém snímku nebo Mean-Shift sledovačem.

Implementoval jsem navržený algoritmus. Jedná se o multiplatformní aplikaci napsanou v C++ a využívající knihovny OpenCV.

Aplikaci jsem vyzkoušel při detekci nejrůznějších objektů jako automobilů, chodců, motocyklů či krav. Ukázalo se, že ISM nedělá kompaktní rozložení pravděpodobnosti v hlasovacím prostoru. Iterativní lokalizace tento nedostatek zmírňuje, protože mnoho špatných hypotéz, které se detekují jako lokální maxima hlasovacího prostoru, eliminuje. Eliminace špatných hypotéz však není dostačující. Sledování objektů snímek po snímku je možné ale pouze na videu s nízkým rozlišením či sníženou fps. Mean-Shif tracking sleduje objekty dle jejich barvy, což není ideální řešení.

Výsledky experimentů jsem porovnal s jinými metodami detekce objektů z jediného snímku. Má aplikace nedosahuje špatných výsledků, ale s nejmodernějšími přístupy se zatím nemůže poměřovat. Stejně tak zaostává za metodami lépe kombinující detekci a sledování.

ISM je velmi intuitivní přístup ke slovníku, ale je už více než 5 let starý a nedělá úplně ukázkové plnění akumulátoru. Proto bych jako další směřování této práce viděl vyzkoušení detekce pomocí Houghova lesu [20, 44, 7] namísto slovníku ISM. Další oblastí vhodnou k otestování je maximalizace energie hlasovacího prostoru použitá v [7]. V neposlední řadě by bylo zajímavé srovnání výsledků, pokud by byla implementována verifikace objektu pomocí propojení detekce a sledování použitých např v [2].

# Literatura


[1] Agarwal, S.; Awan, A.; Roth, D.: Learning to detect objects in images via a sparse, part-based representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, ročník 26, č. 11, 2004: s. 1475–1490.

[2] Andriluka, M.; Roth, S.; Schiele, B.: People tracking by detection and people detection by tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, 2008, s. 1–8.

[3] Arthur, D.; Vassilvitskii, S.: On the worst case complexity of the k-means method. *Technical Report*, 2005.

[4] Arthur, D.; Vassilvitskii, S.: k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, Society for Industrial and Applied Mathematics, 2007, s. 1027–1035.

[5] Avidan, S.: Support vector tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, ročník 26, č. 8, 2004: s. 1064–1072.

[6] Ballard, D.: Generalizing the Hough transform to detect arbitrary shapes. *Pattern recognition*, ročník 13, č. 2, 1981: s. 111–122.

[7] Barinova, O.; Lempitsky, V.; Kohli, P.: On detection of multiple object instances using hough transforms. In *IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, 2010, s. 2233–2240.

[8] Bay, H.; Ess, A.; Tuytelaars, T.; aj.: Speeded-up robust features (SURF). *Computer Vision and Image Understanding*, ročník 110, č. 3, 2008: s. 346–359.

[9] Bertalmío, M.; Sapiro, G.; Randall, G.: Morphing active contours. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, ročník 22, č. 7, 2000: s. 733–737.

[10] Calonder, M.; Lepetit, V.; Strecha, C.; aj.: Brief: Binary robust independent elementary features. *Computer Vision–ECCV*, 2010: s. 778–792.

[11] Chang, Y.; Aggarwal, J.: 3d structure reconstruction from an ego motion sequence using statistical estimation and detection theory. In *Proceedings of the IEEE Workshop on Visual Motion*, IEEE, 1991, s. 268–273.

[12] Comaniciu, D.; Ramesh, V.; Meer, P.: Kernel-based object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, ročník 25, č. 5, 2003: s. 564–577.

[13] Dalal, N.; Triggs, B.: Histograms of oriented gradients for human detection. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, ročník 1, Ieee, 2005, s. 886–893.

[14] Defays, D.: An efficient algorithm for a complete link method. *The Computer Journal*, ročník 20, č. 4, 1977: str. 364.

[15] Donoser, M.; Bischof, H.: Efficient maximally stable extremal region (MSER) tracking. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, ročník 1, Ieee, 2006, s. 553–560.

[16] Duda, R.; Hart, P.: Use of the Hough transformation to detect lines and curves in pictures. *Communications of the ACM*, ročník 15, č. 1, 1972: s. 11–15.

[17] Fergus, R.; Perona, P.; Zisserman, A.: Object class recognition by unsupervised scale-invariant learning. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, ročník 2, IEEE, 2003, s. II–264.

[18] Fischler, M.; Bolles, R.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, ročník 24, č. 6, 1981: s. 381–395.

[19] Fritz, G.; Seifert, C.; Paletta, L.: A mobile vision system for urban detection with informative local descriptors. In *IEEE International Conference onComputer Vision Systems*, IEEE, 2006, s. 30–30.

[20] Gall, J.; Lempitsky, V.: Class-specific hough forests for object detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, Ieee, 2009, s. 1022–1029.

[21] Garg, A.; Agarwal, S.; Huang, T.: Fusion of global and local information for object detection. In *16th International Conference on Pattern Recognition*, ročník 3, IEEE, 2002, s. 723–726.

[22] Harris, C.; Stephens, M.: A combined corner and edge detector. In *Alvey vision conference*, ročník 15, Manchester, UK, 1988, str. 50.

[23] Hue, C.; Le Cadre, J.; Perez, P.: Sequential Monte Carlo methods for multiple target tracking and data fusion. *IEEE Transactions on Signal Processing*, ročník 50, č. 2, 2002: s. 309–325.

[24] Huttenlocher, D.; Noh, J.; Rucklidge, W.: Tracking non-rigid objects in complex scenes. In *Fourth International Conference on Computer Vision*, IEEE, 1993, s. 93–101.

[25] Isard, M.; Blake, A.: Condensation-conditional density propagation for visual tracking. *International journal of computer vision*, ročník 29, č. 1, 1998: s. 5–28.

[26] Jinman Kang, I. C.; Medioni, G.: Object Reacquisition Using Invariant Appearance Model. *InInternational Conference on Pattern Recongnition*, 2004: s. 759–762.

[27] Kitagawa, G.: Non-Gaussian state-space modeling of nonstationary time series. *Journal of the American Statistical Association*, 1987: s. 1032–1041.

[28] Leibe, B.; Leonardis, A.; Schiele, B.: Robust object detection with interleaved categorization and segmentation. *International Journal of Computer Vision*, ročník 77, č. 1, 2008: s. 259–289.

[29] Leibe, B.; Schiele, B.: Interleaving object categorization and segmentation. *Cognitive Vision Systems*, 2006: s. 145–161.

[30] Lindeberg, T.: Scale-space theory: A basic tool for analyzing structures at different scales. *Journal of applied statistics*, ročník 21, č. 1-2, 1994: s. 225–270.

[31] Lindeberg, T.: Feature detection with automatic scale selection. *International Journal of Computer Vision*, ročník 30, č. 2, 1998: s. 79–116.

[32] Lloyd, S.: Least squares quantization in PCM. *IEEE Transactions on Information Theory*, ročník 28, č. 2, 1982: s. 129–137.

[33] Lowe, D.: Object recognition from local scale-invariant features. In *The Proceedings of the Seventh IEEE International Conference on Computer Vision*, ročník 2, Ieee, 1999, s. 1150–1157.

[34] Lowe, D.: Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, ročník 60, č. 2, 2004: s. 91–110.

[35] MacKay, D.: Introduction to Monte Carlo methods. *Proceedings of the NATO Advanced Study Institute on Learning in graphical models*, ročník 89, 1998: s. 175–204.

[36] MacQueen, J.; aj.: Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, ročník 1, California, USA, 1967, s. 281–297.

[37] Magee, D.; Boyle, R.: Feature tracking in real world scenes (or how to track a cow). In *IEE Colloquium on Motion Analysis and Tracking*, IET, 1999.

[38] Mikolajczyk, K.; Schmid, C.: Scale & affine invariant interest point detectors. *International journal of computer vision*, ročník 60, č. 1, 2004: s. 63–86.

[39] Mikolajczyk, K.; Schmid, C.: A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence,*, ročník 27, č. 10, 2005: s. 1615–1630.

[40] Moravec, H.: *Obstacle avoidance and navigation in the real world by a seeing robot rover*. Dizertační práce, Department of Computer Science, Stanford University, Září 1980.

[41] Mutch, J.; Lowe, D.: Multiclass object recognition with sparse, localized features. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, ročník 1, IEEE, 2006, s. 11–18.

[42] Rasmussen, C.; Hager, G.: Probabilistic data association methods for tracking complex visual objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, ročník 23, č. 6, 2001: s. 560–576.

[43] Reid, D.: An algorithm for tracking multiple targets. *IEEE Transactions on Automatic Control*, ročník 24, č. 6, 1979: s. 843–854.

[44] Rematas, K.; Leibe, B.: Efficient object detection and segmentation with a cascaded Hough Forest ISM. In *IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, IEEE, 2011, s. 966–973.

[45] Ronfard, R.: Region-based strategies for active contour models. *International Journal of Computer Vision*, ročník 13, č. 2, 1994: s. 229–251.

[46] Rosten, E.; Drummond, T.: Machine learning for high-speed corner detection. *Computer Vision–ECCV 2006*, 2006: s. 430–443.

[47] Sarfraz, M.; Hellwich, O.: Head pose estimation in face recognition across pose scenarios. *VISAPP (1)*, 2008: s. 235–242.

[48] Sato, K.; Aggarwal, J.: Temporal spatio-velocity transform and its application to tracking and interaction. *Computer Vision and Image Understanding*, ročník 96, č. 2, 2004: s. 100–128.

[49] Schweitzer, H.; Bell, J.; Wu, F.: Very fast template matching. *Computer Vision-ECCV 2002*, 2006: s. 145–148.

[50] Seifoddini, H.: Single linkage versus average linkage clustering in machine cells formation applications. *Computers & Industrial Engineering*, ročník 16, č. 3, 1989: s. 419–426.

[51] Shafique, K.; Shah, M.: A non-iterative greedy algorithm for multi-frame point correspondence. *IEEE International Conference on Computer Vision*, 2003: s. 110–115.

[52] Shi, J.; Tomasi, C.: Good features to track. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, IEEE, 1994, s. 593–600.

[53] Sibson, R.: SLINK: an optimally efficient algorithm for the single-link cluster method. *The Computer Journal*, ročník 16, č. 1, 1973: s. 30–34.

[54] Tola, E.; Lepetit, V.; Fua, P.: A fast local descriptor for dense matching. In *IEEE Conference on Computer Vision and Pattern Recognition*, Ieee, 2008, s. 1–8.

[55] Tola, E.; Lepetit, V.; Fua, P.: Daisy: An efficient dense descriptor applied to wide-baseline stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, ročník 32, č. 5, 2010: s. 815–830.

[56] Tuytelaars, T.: Dense interest points. In *IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, 2010, s. 2281–2288.

[57] Veenman, C.; Reinders, M.; Backer, E.: Resolving motion correspondence for densely moving points. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, ročník 23, č. 1, 2001: s. 54–72.

[58] Viola, P.; Jones, M.: Robust real-time face detection. *International journal of computer vision*, ročník 57, č. 2, 2004: s. 137–154.

[59] Wang, L.; Shi, J.; Song, G.; aj.: Object detection combining recognition and segmentation. *Lecture Notes in Computer Science*, 2007: str. 189.

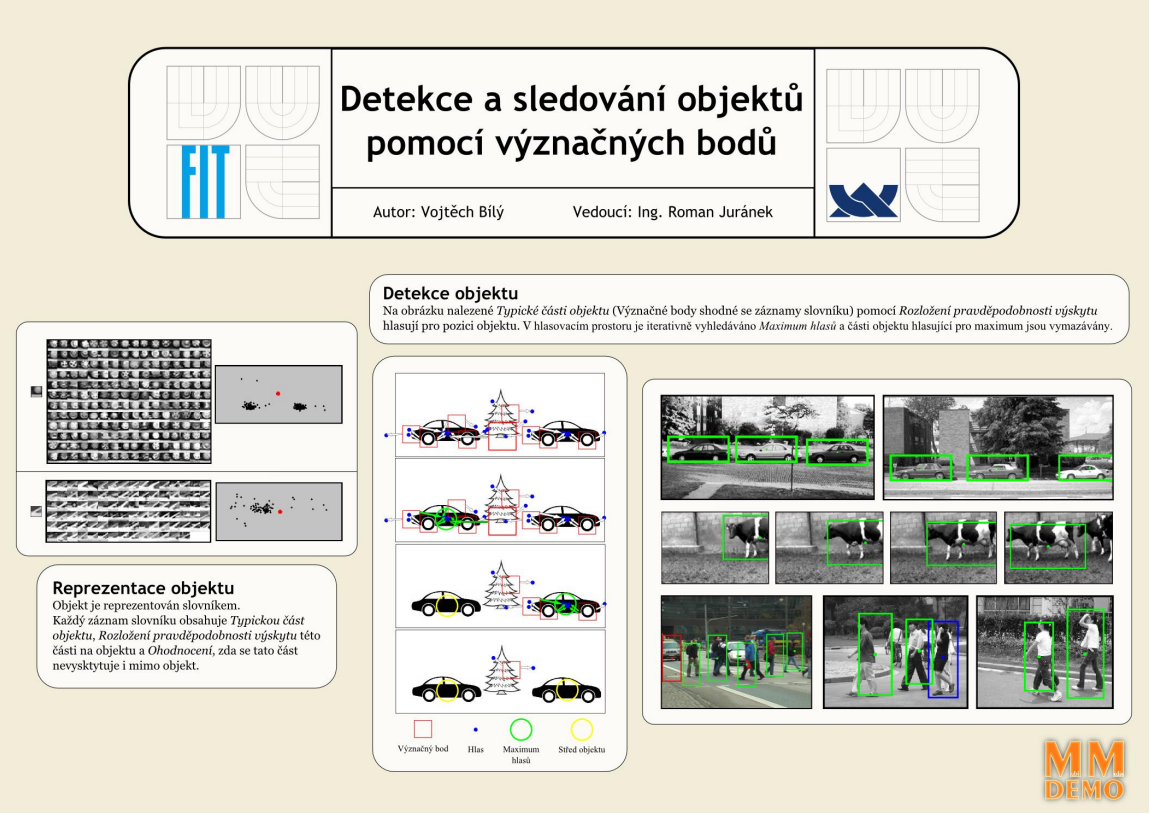
[60] Welch, G.; Bishop, G.: An introduction to the Kalman filter. *Design*, ročník 7, č. 1, 2001: s. 1–16.

[61] Yilmaz, A.; Javed, O.; Shah, M.: Object tracking: A survey. *Acm Computing Surveys (CSUR)*, ročník 38, č. 4, 2006: str. 13.

[62] Yilmaz, A.; Li, X.; Shah, M.: Contour-based object tracking with occlusion handling in video acquired using mobile cameras. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, ročník 26, č. 11, 2004: s. 1531–1536.

# Příloha A

# Plakát




Obrázek A.1: Plakát.