

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

MĚŘENÍ VÝŠKY POSTAVY V OBRAZE

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. ADAM OLEJÁR

BRNO 2015



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ**

ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

MĚŘENÍ VÝŠKY POSTAVY V OBRAZE

HEIGHT MEASUREMENT IN DIGITAL IMAGE

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. ADAM OLEJÁR

VEDOUČÍ PRÁCE

SUPERVISOR

doc. Ing. KAMIL ŘÍHA, Ph.D.

BRNO 2015



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav telekomunikací

Diplomová práce

magisterský navazující studijní obor
Telekomunikační a informační technika

Student: Bc. Adam Olejář

ID: 134576

Ročník: 2

Akademický rok: 2014/2015

NÁZEV TÉMATU:

Měření výšky postavy v obraze

POKYNY PRO VYPRACOVÁNÍ:

Nastudujte teoretický princip měření výšky postavy umístěné v rovině, tedy proceduru kalibrace kamery, detekce podstavné roviny, detekce postavy a změření její výšky vzhledem k dané rovině. Naprogramujte samostatnou aplikaci, která bude realizovat kalibrační proceduru a automatické určování parametrů podstavné roviny v testovacích snímcích. Tato aplikace bude dále umožňovat označení postavy, jejích koncových bodů a finální výpočet výšky.

Doporučený nástroj pro implementaci: knihovny OpenCV a vývojové prostředí MS Visual C++.

DOPORUČENÁ LITERATURA:

- [1] Hartley R., Zissermann A.: Multiple View Geometry in Computer Vision. Cambridge: Cambridge University Press, 2003.
- [2] Gonzalez R.C., Woods R.E.: Digital Image Processing. New Jersey: Prentice-Hall, 2002.
- [3] PRATA S.: Mistrovství v C++, Computer Press, Brno 2004, ISBN 80-251-0098-7.

Termín zadání: 9.2.2015

Termín odevzdání: 26.5.2015

Vedoucí práce: doc. Ing. Kamil Říha, Ph.D.

Konzultanti diplomové práce:

doc. Ing. Jiří Mišurec, CSc.

Předseda oborové rady

UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

ABSTRAKT

Cieľom tejto práce bolo zhrnutie teórie nutnej pre úpravu, detekciu postavy a výpočet výšky detekovanej postavy v obraze. Následne boli tieto informácie aplikované pri implementácii algoritmu. Prvá polovica práce teoreticky rozoberá skúmanú problematiku. Pojednáva o základných metódach predspracovania obrazu, rovinnej a projektívnej geometrie a ich transformácií. Popisuje skreslenie, ktoré do obrazu vnášajú nedokonalosti optických sústav kamier a možnosti jeho odstránenia. Následne vysvetľuje algoritmus HOG a opisuje samotnú metódu určenia výšky postavy. V Druhej polovici práce je zdokumentovaný vytvorený algoritmus spolu s jeho štatistickým vyhodnotením.

KLÚČOVÉ SLOVÁ

HOG, SVM, nevlastný bod, histogram orientovaných gradientov, OpenCV, výška postavy, kalibrácia kamery, výška postavy v obraze

ABSTRACT

The aim of this paper is a summary of the theory necessary for a modification, detection of person and the height calculation of the detected person in the image. These information were then used for implementation of the algorithm. The first half reveals theoretical problems and solutions. Shows the basic methods of image preprocessing and discusses the basic concepts of plane and projective geometry and transformations. Then describes the distortion, that brings into the picture imperfections of optical systems of cameras and the possibilities of removing them. Explains HOG algorithm and the actual method of calculating height of person detected in the image. The second half describes algorithm structure and statistical evaluation.

KEYWORDS

HOG, SVM, vanishing point, histogram of oriented gradients, OpenCV, person's height, camera calibration, person's height in the image

PREHLÁSENIE

Prehlasujem, že som svoju diplomovú prácu na tému „Měření výšky postavy v obraze“ vypracoval samostatne pod vedením vedúceho diplomovej práce, využitím odbornej literatúry a ďalších informačných zdrojov, ktoré sú všetky citované v práci a uvedené v zozname literatúry na konci práce.

Ako autor uvedenej diplomovej práce ďalej prehlasujem, že v súvislosti s vytvorením tejto diplomovej práce som neporušil autorské práva tretích osôb, najmä som nezasiahol nedovoleným spôsobom do cudzích autorských práv osobnostných a/nebo majetkových a som si plne vedomý následkov porušenia ustanovenia § 11 a nasledujúcich autorského zákona č. 121/2000 Sb., o právu autorskom, o právach súvisejúcich s právom autorským a o zmene niektorých zákonov (autorský zákon), vo znení neskorších predpisov, vrátane možných trestnoprávných dôsledkov vyplývajúcich z ustanovenia časti druhej, hlavy VI. diel 4 Trestného zákoníka č. 40/2009 Sb.

Brno

.....

(podpis autora)

POĎAKOVANIE

Rád bych poděkoval vedoucímu diplomové práce panu doc. Ing. Kamilovi Říhovi, Ph.D. za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Brno

.....

(podpis autora)

OBSAH

Úvod	11
1 Predspracovanie obrazu	12
1.1 Histogram	12
1.1.1 Ekvalizácia histogramu	12
1.2 Odstránenie šumu	14
1.2.1 Mediánový filter	15
1.2.2 Gaussov filter	15
2 Geometria obrazu	16
2.1 Rovinná geometria	16
2.1.1 Stĺpcové a riadkové vektory	16
2.1.2 Bod a priamka	16
2.1.3 Priesečník priamok	17
2.1.4 Priamka prechádzajúca dvoma bodmi	18
2.1.5 Stupne voľnosti	18
2.2 3D → 2D Projektívna geometria	19
2.2.1 Obrazová rovina	19
2.2.2 Priesečník rovnobežiek	20
2.2.3 Nevlastné body	20
2.2.4 Nevlastná priamka, Vanishing line	20
2.3 Geometrické transformácie	23
2.3.1 Euklidovské transformácie	24
2.3.2 Afinné transformácie	25
2.3.3 Projektívna transformácia	26
3 Model kamery	27
3.1 Skreslenie obrazu	27
3.1.1 Perspektívne skreslenie	27
3.1.2 Radiálne skreslenie	28
3.2 Potlačenie skreslenia	30
3.2.1 Potlačenie perspektívneho skreslenia	30
3.2.2 Potlačenie radiálneho skreslenia – kalibrácia kamery	30
4 Algoritmus pre detekciu osoby - HOG	32
4.1 Princíp HOG detektoru	32
4.1.1 Hrana, Gradient	33
4.1.2 Predspracovanie obrazu	34

4.1.3	Výpočet orientovaných gradientov	34
4.1.4	Normalizácia orientovaných gradientov	35
4.1.5	Výpočet histogramu buniek	36
4.1.6	Normalizácia blokov	36
4.1.7	Support Vector Machine - SVM	37
5	Metóda výpočtu výšky postavy	40
6	Algoritmus pre výpočet výšky postavy	44
6.1	Úvod	44
6.1.1	OpenCV	44
6.2	Algoritmus	44
6.2.1	Metóda Main()	45
6.3	Odstránenie skreslenia	45
6.3.1	Kalibrácia kamery	45
6.3.2	Korekcia obrazu	47
6.4	Detekcia postavy	48
6.5	Detekcia vanishing pointov	50
6.5.1	Metóda computeVanishingPoints()	50
6.5.2	Metóda getBeams()	53
6.5.3	Metóda getVanishingPoint()	54
6.6	Výpočet výšky postavy	55
6.6.1	Metóda setVariables()	56
6.6.2	Metóda compute()	56
6.7	Vstupy a výstupy	56
7	Vyhodnotenie	59
7.1	Závislosť 1	60
7.2	Závislosť 2	62
7.3	Zhrnutie	64
7.4	Podmienky pre správne fungovanie metódy	65
8	Záver	66
	Literatúra	67
	Zoznam symbolov, veličín a skratiek	69

ZOZNAM OBRÁZKOV

1.1	Príklad ekvalizácie histogramu	14
2.1	Priesečník 2 priamok	18
2.2	Priamka prechádzajúca 2 bodmi	18
2.3	Sústava pozorovateľ - projektívna rovina - snímaná scéna	19
2.4	Grafické znázornenie kolajníc	21
2.5	Obrazy z reálneho sveta	21
2.6	Vzájomné vzťahy scény a projektívnej roviny.	22
2.7	Centrálne transformácie bodu $\mathbf{x} \rightarrow \mathbf{x}'$ podľa bodu C	23
3.1	Vznik Perspektívneho Skreslenia	28
3.2	Obraz so súdkovitým skreslením	29
3.3	Obraz s poduškovým skreslením	29
3.4	Obraz s komplexným skreslením	29
4.1	Schéma algoritmu metódy HOG	32
4.2	Vzťah gradientu a hrany	33
4.3	Druhy hrán	34
4.4	Spôsob určenia gradientu vektora pre jednotlivé pixely.	35
4.5	Lineárne neodeliteľné triedy [14]	37
4.6	Lineárne oddelenie triedy nadrovinou [14]	38
4.7	Pohľad na kolmo na rovinu	38
4.8	Proces učenia SVM	39
4.9	Proces detekcie SVM	39
5.1	Náčrt priestorovej konfigurácie dvoch výšok voči rovine	40
5.2	Zostrojenie horizontálneho a vertikálneho VP, horizontálnej VL a bodu \mathbf{t}_1'	41
5.3	Vytvorenie bodu \mathbf{T}_1'' a \mathbf{T}_2'	42
6.1	Jednotlivé body kalibračného obrazca	46
6.2	Ukážka korekcie obrazu	48
6.3	Diagram metódy <code>computeVanishingPoints()</code>	51
6.4	Binárny obraz hrán po detektore <code>Canny()</code>	52
6.5	Priamky detekované detektorom <code>HoughLinesP()</code>	53
6.6	Diagram metódy <code>getBeams()</code>	54
6.7	3 rôzne smery priamok pre 3 rôzne VP;	55
6.8	Dva označené body	57
6.9	Konzolový vstup a výstup	58
7.1	Vzdialenosť referenčných výšok z tab. 7.4	60
7.2	Závislosť chyby výpočtu od percentuálnej vzdialenosti referenčnej výšky a výšky osoby voči celkovej šírke obrazu	61

7.3	Zvolené priamky	62
7.4	Zavislosť hodnôt z tab. 7.5 pre priamku č.1 z obrazu 7.3	63
7.5	Zavislosť hodnôt z tab. 7.6 pre priamku č.2 z obrazu 7.3	64

ZOZNAM TABULIEK

7.1	Hodnoty vypočítané pre jeden referenčný rozmer v rôznej vzdialenosti	59
7.2	Hodnoty namerané na priamke č.2 na obr 7.3	59
7.3	Hodnoty namerané na priamke č.1 na obr 7.3	60
7.4	Hodnoty ku grafu 7.2	61
7.5	Skutočné výšky jednotlivých bodov pre priamku č.1	62
7.6	Skutočné výšky jednotlivých bodov pre priamku č.2	63

ÚVOD

Pri pohľade na neznámy objekt náš mozog spustí zložitý proces merania a skúmania priestoru ktorý vidíme. Deje sa to automaticky a vďaka skúsenostiam a vlastnostiam objektov, ktoré si pamätáme, dokážeme pomerne presne odhadovať vzdialenosti bez toho, aby sme tieto rozmery poznali.

Oproti väčšine digitálnych obrazových záznamov je obrovskou výhodou mozgu stereoskopický obraz. Vďaka nemu je schopný odhadnúť dĺžku s obrovskou presnosťou. V poslednej dobe rastie úsilie predať túto schopnosť stroju.

Čoraz častejšie sa stretávame s požiadavkou získania čo najpresnejších fyzikálnych údajov z rôznych druhov obrazu. S nástupom stále výkonnejších počítačov, nových nástrojov, metód a programovacích jazykov stúpajú schopnosti algoritmov získavať tieto dáta často už aj v reálnom čase. Bohužiaľ, väčšina obrazových záznamov stereoskopickým obrazom nedisponuje, čo proces výpočtu značne komplikuje.

Cieľom tejto práce bolo zhrnúť problematiku detekcie a výpočtu výšky osoby v obraze a následne vytvoriť algoritmus, ktorý by takouto schopnosťou disponoval. K tomuto účelu je v prvej polovici práce teoreticky vysvetlená problematika predspracovania obrazu a základnej obrazovej geometrie. Následne je vysvetlený princíp metódy HOG pre vyhľadávanie objektov v obraze a popísaný samotný spôsob výpočtu.

V druhej polovici práce je popísaný vytvorený algoritmus spolu s vyhodnotením jeho vlastností.

1 PREDSPRACOVANIE OBRAZU

Obraz, ktorý chceme analyzovať obsahuje často šum, rozostrenie, alebo je nevhodne exponovaný. Pred samotným použitím metód, ktoré budú popísané v nasledujúcich kapitolách, na daný obraz, je vhodné aplikovať naň niekoľko jednoduchých úprav, ktoré môžu zvýšiť ich účinnosť a presnosť.

Najčastejšími problémami, kvôli ktorým je nutné obraz takto upravovať, je zlá expozícia scény alebo šum. Dôsledkom zlej expozície vznikajú plochy v obraze ktoré sú preexponované alebo podexponované. Hodnoty jasů v daných miestach dosahujú maximálnych hodnôt a preto nenesú skoro žiadnu užitočnú informáciu.

1.1 Histogram

Histogram je jedna zo základných charakteristík snímku, ktoré berieme do úvahy. Z matematického hľadiska je to pravdepodobnostná funkcia ktorá priraduje každému stĺpcu v ňom jednu hodnotu pravdepodobnosti. Obrazový histogram vyjadruje hodnotu pravdepodobnosti zastúpenia jednotlivých farebných alebo jasových hodnôt v obraze. Vzhľadom na hodnotu ktorú histogram vyjadruje môžeme hovoriť o:

1. Histogram jednotlivých farebných zložiek: v tomto prípade sa pre každú farebnú zložku určí vlastný histogram.
2. Histogram jasovej zložky: vyjadruje rozloženie jasů v obraze s ohľadom na citlivosť ľudského oka pre jednotlivé farby. Hodnota jasů Y pre farebný model RGB je vypočítaná zo vzťahu

$$Y(r, g, b) = 0,299r + 0,587g + 0,114b \quad (1.1)$$

kde r označuje red (červená), g označuje green (zelená), b označuje blue (modrá). Histogram obrazu s jasovým rozsahom $\langle 0, L - 1 \rangle$ definujeme ako diskrétnu funkciu

$$h(r_k) = n_k \quad (1.2)$$

kde r_k je k -tá úroveň jasů a n_k je počet bodov s touto úrovňou jasů.

Pre potreby tejto práce budeme ďalej hovoriť iba o jasovom histograme ktorého úpravou je možné meniť kontrast a jas v obraze. [5][9]

1.1.1 Ekvalizácia histogramu

Ekvalizáciou chceme upraviť histogram na stav, pri ktorom dosiahneme rovnomerné zastúpenie jasových zložiek, čím dôjde aj ku zvýšeniu kontrastu.

Aby bolo možné odvodiť transformáciu ekvalizovaného histogramu, musíme ho považovať za spojitú funkciu. Následne prevedieme obe premenné jasových hodnôt

s a r do intervalu $\langle 0, 1 \rangle$, v ktorom $r = 0$ predstavuje úroveň čiernej a $r = 1$ úroveň bielej. Potom hľadáme takú transformáciu

$$s = T(r) \text{ kde } 0 \leq r \leq 1 \quad (1.3)$$

pomocou ktorej môžeme realizovať prevod medzi pôvodnými úrovňami jasu r na výsledné hodnoty s , pre ktorú musí platiť:

1. $T(r)$ je monotónne rastúca v intervale $0 \leq T(r) \leq 1$
2. $T(r)$ je jednoznačná v intervale $0 \leq T(r) \leq 1$
3. $0 \leq T(r) \leq 1$ pre $0 \leq r \leq 1$

Podmienka monotónnosti 1 zaisťuje správne usporiadanie hodnôt s rastom od čiernej po bielu. Ak by transformačná funkcia túto podmienku nespĺnila, bude v určitých hodnotách jasu produkovať inverzný obraz. Podmienka jednoznačnosti 2 definuje existenciu spätnej transformácie. Podmienka 3, zaisťuje že vstupné aj výstupné hodnoty budú spadať do jedného intervalu funkčných hodnôt.

Ak sú všetky tieto podmienky splnené, je možné nájsť inverznú transformáciu T^{-1} hodnôt s do r

$$r = T^{-1}(s) \text{ kde } 0 \leq r \leq 1. \quad (1.4)$$

Ak histogram pôvodného obrazu $H(r)$ s jasovým rozsahom $r = \langle r_0, r_k \rangle$ transformujeme na obraz s ekvalizovaným histogramom $G(s)$ s jasovým rozsahom $s = \langle s_0, s_k \rangle$ a chceme aby s a r bolo monotónne, môžeme napísať

$$\sum_{i=0}^k G(s_i) = \sum_{i=0}^k H(r_i) \quad (1.5)$$

kde k je maximálny jasový rozsah. Sumy v rovnici 1.5 sú diskkrétne distribučné funkcie. Histogram $G(s)$ má rovnomerné rozdelenie f s konštantnou hustotou pravdepodobnosti.

$$f = \frac{N^2}{s_k - s_0} \quad (1.6)$$

Ak v spojitom prípade dosadíme za ľavú stranu 1.5 rovnicu 1.6, potom

$$N^2 \int_{s_0}^s \frac{1}{s_k - s_0} ds = \frac{N^2(s - s_0)}{s_k - s_0} = \int_{r_0}^r H(s) ds \quad (1.7)$$

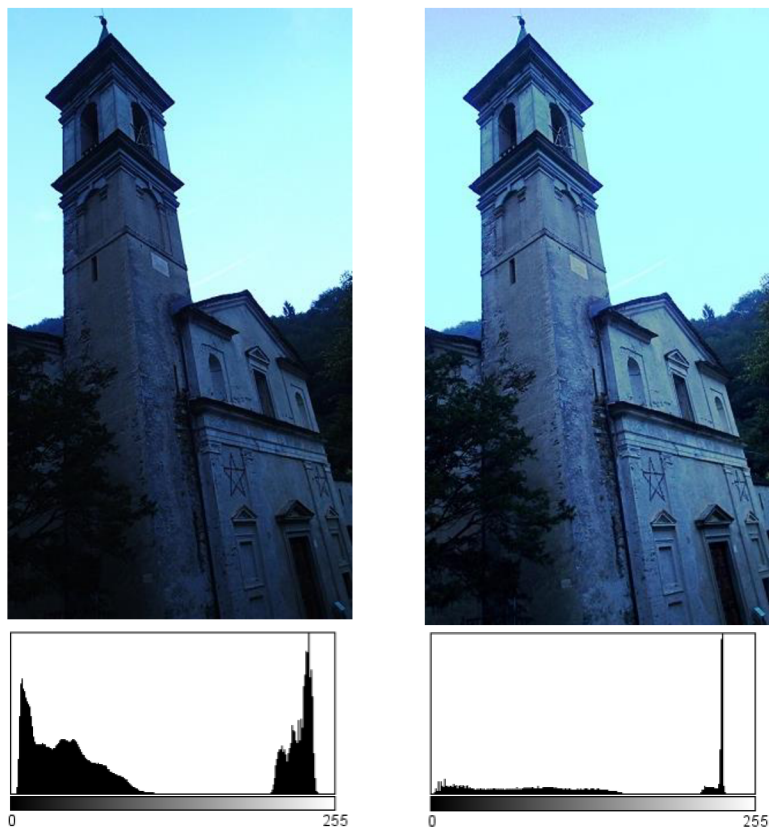
Transformáciu T je následne možné vyjadriť ako

$$s = T(r) = \frac{s_k - s_0}{N^2} \int_{r_0}^r H(s) ds + s_0 \quad (1.8)$$

a z nej aproximácia bude

$$s = T(r) = \frac{s_k - s_0}{N^2} \sum_{i=0}^r H(i) + s_0 \quad (1.9)$$

Na obr. 1.1 je zrejmy účinok ekvalizácie na podexponovaný obraz. Budova na obr 1.1a ktorému nebol ekvalizovaný histogram je tmavá a niektoré kontúry splývajú s tmavými oblasťami v tieni. Obrazu 1.1b bol histogram ekvalizovaný a na prvý pohľad je zrejme že detaily budovy sú výraznejšie a lepšie čitateľné. Ekvalizáciou sme dosiahli zvýšenie kontrastu pod-exponovanej snímky.



(a) Obraz pred ekvalizáciou

(b) Obraz po ekvalizácii

Obr. 1.1: Príklad ekvalizácie histogramu

Účinok ekvalizácie je dobre viditeľný aj na samotných histogramoch oboch obrazov. Obraz 1.1a má dve výrazné maximá na oboch stranách intervalu, avšak v stredných hodnotách sa nenachádzajú skoro žiadne. U ekvalizovaného histogramu 1.1b je jas rozložený rovnomerne vo väčšine intervalu. [9] [1] [5]

1.2 Odstránenie šumu

Odstránenie šumu je jednoduchou a účinnou metódou pre zlepšenie čitateľnosti obrazu pre algoritmus. V tejto kapitole budú popísané iba základné možnosti redukcie šumu.

1.2.1 Mediánový filter

Jedná sa o nelineárny štatistický filter. Odozva závisí od štatistickej kvalifikácie bodov daných filtračnou maskou. Body, ktoré do filtračnej masky spadajú, budú usporiadané podľa veľkosti. Z tohto usporiadania sa vyberie medián¹, a jeho hodnota bude použitá ako nová hodnota daného pixelu.

Cieľom mediánového filtra je eliminovať veľké jasové rozdiely na malej ploche spôsobené najčastejšie impulzným šumom (napr. typu „sůl a pepř“). [1] [9] [5]

1.2.2 Gaussov filter

Filter, ktorého odozva je definovaná priemerom hodnôt bodov v okolí definovaným filtračnou maskou. Maska je definovaná vzťahom 1.10

$$G(x, y; \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (1.10)$$

kde σ reprezentuje štandardnú odchýlku hodnôt pre Gaussovu funkciu a x, y je horizontálna a vertikálna vzdialenosť od pôvodného bodu.

Gaussov filter je lineárny vyhladzovací filter. Filtruje vysoké frekvencie v obraze, ktoré sú charakterizované veľkou zmenou jasů na malej ploche. Tieto často charakterizujú šum, ale aj ostré hrany a rohy. Aplikáciou takéhoto filtra sa preto odstráni šum, ale aj ostrosť hrán a preto sa obraz javí mierne rozostrený, čo je niekedy nežiadúce. [1] [9] [5]

¹hodnota, ktorá je poradovým číslom približne v strede

2 GEOMETRIA OBRAZU

Kapitola obsahuje popis základných pojmov a metód rovinnej geometrie a projektívnych transformácií v rovine. Tieto transformácie vytvárajú v zachytenom obraze určité geometrické skreslenie. Toto skreslenie závisí od optickej sústavy kamery/fotoaparátu a jej/jeho perspektívy vzhľadom na zachytenú scénu. Pri skreslení sa stále zachovávajú isté vlastnosti, napr. kolinearita (rovná čiara je zobrazená ako rovná čiara), avšak napr. dve rovnobežné priamky už po transformácii rovnobežné nie sú. V tejto kapitole budú popísané základné pojmy nutné pre objasnenie problematiky ktoré budú používané v nasledujúcich kapitolách.

2.1 Rovinná geometria

Na geometriu je možné nahliadať z 2 pohľadov:

1. Geometrický: útvary nie sú definované súradnicami, ale vzájomnou vzdialenosťou a základnými geometrickými útvarmi bez použitia algebry
2. Algebraický: využíva matematické definície geometrických útvarov a ich súradníc pre výpočet požadovaných skutočností.

Pre naše účely je väčšinou vhodnejší algebraický prístup, ktorý je vhodnejší pre algoritmizáciu a bude použitý aj v tejto kapitole pre definíciu základných vlastností jednotlivých útvarov.

2.1.1 Stĺpcové a riadkové vektory

Ak zvážime, že lineárne prevody medzi vektorovými priestormi môžeme reprezentovať formou matice, zvyčajne je výsledok operácie vektorového súčinu – matica \times vektor – ďalší vektor. Preto musíme uvažovať o rozdielnom použití stĺpcových a riadkových matíc, keďže nie každú maticu a vektor je možné spolu vynásobiť.

Geometrické entity budú primárne reprezentované stĺpcovým vektorom, napr. \mathbf{x} a ich transponovaná riadková forma bude označená \mathbf{x}^\top . Z toho vyplýva že môžeme

zapísať stĺpcový vektor $\begin{pmatrix} x \\ y \end{pmatrix}$

aj tvarom $(x, y)^\top$ pričom oba zápisy reprezentujú tú istú formu.[6]

2.1.2 Bod a priamka

Bod a priamka sú dva základné geometrické útvary, ktoré budú v nasledujúcej časti popísané a bude vysvetlený ich zápis používaný v nasledujúcich kapitolách.

Priamka

Každú priamku je možné algebraicky reprezentovať rovnicou

$$ax + by + c = 0 \quad (2.1)$$

v homogénnej sústave súradníc, pričom a , b , a c sú parametre, ktoré ju jednoznačne charakterizujú a x, y sú súradnice ľubovoľného bodu ležiaceho na nej. Priamku teda môžeme zapísať ako vektor $(a, b, c)^\top$, pričom za ekvivalentnú priamku je považovaná aj akákoľvek iná priamka $k(a, b, c)^\top$ pre $k \in \mathbb{R} - \{0\}$.

Bod a jeho homogénna reprezentácia

Z predchádzajúcej definície priamky vychádza aj reprezentácia bodu v homogénnych súradniciach. Bod $\mathbf{x} = (x, y)^\top$ leží na priamke $\mathbf{l} = (a, b, c)^\top$ iba ak $ax + by + c = 0$.

Algebraické riešenie tohto problému môžeme napísať v podobe skalárneho súčinu dvoch vektorov, pričom bod $(x, y)^\top$ v \mathbb{R}^2 bude reprezentovaný vektorom $\mathbf{x} = (x, y, 1)$.

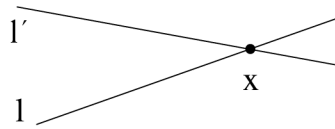
Rozšírením vektora o poslednú súradnicu 1 prevedieme vektor do homogénneho zápisu. Napriek 3 súradniciam stále zostávame v rovine. Dôvod tohto kroku spočíva v probléme transformácie bodu $(0,0)$, ktorý bude po transformácii ľubovoľnou maticou zase iba bod $(0,0)$. Dôsledky tejto operácie sú:

1. Možnosť zapísať bod nekonečným množstvom spôsobov, pričom musí platiť $(x, y, 1)^\top = (kx, ky, k)^\top$ kde $k \neq 0$. Prevod na nehomogénny zápis spočíva v delení alebo násobení vektora tak, aby posledná súradnica bola 1, potom sa už uvažujú iba prvé dve súradnice.
2. Vznik tzv. nevlastných bodov – bodov v nekonečne. Viac o týchto bodoch v kapitole 2.2.2 a 2.2.3

Následne môžeme napísať $(x, y, 1)(a, b, c)^\top = (x, y, 1)\mathbf{l} = 0$. Podobne ako pri priamke, musí platiť že $k \in \mathbb{R} - \{0\}$ a pre priamku \mathbf{l} platí že $(kx, ky, k)\mathbf{l} = 0$ iba ak platí $(x, y, 1)\mathbf{l} = 0$. Bod je teda podobne ako priamku možné reprezentovať homogénnou formou vektoru $\mathbf{x} = (x_1, x_2, x_3)^\top$ a jeho nehomogénnou formou v \mathbb{R}^2 $(\frac{x_1}{x_3}, \frac{x_2}{x_3})^\top = (x, y)^\top$. [6]

2.1.3 Priesečník priamok

Je to konkrétny bod, v ktorom sa pretínajú aspoň dve priamky. Získanie polohy priesečníka dvoch priamok bude v ďalších kapitolách zásadné pre získanie informácií o skreslení a snahe o jeho odstránenie. Následne aj pri odhadovaní skutočných tvarov a rozmerov v obraze.



Obr. 2.1: Priesečník 2 priamok

Algebraicky je definovaný ako vektorový súčin dvoch priamok. Zapísať je to možné

$$\mathbf{x} = \mathbf{l} \times \mathbf{l}' \quad (2.2)$$

kde \mathbf{x} je výsledný bod a \mathbf{l}, \mathbf{l}' sú dané priamky. [6]

2.1.4 Priamka prechádzajúca dvoma bodmi

Podobne ako hľadanie priesečníku priamok, aj nájdenie priamky prechádzajúcej dvoma konkrétnymi bodmi je veľmi často používaná operácia hlavne pri odhadovaní skutočných pomerov v obraze. Analogicky ku priesečníku je definovaná ako vektorový súčin

$$\mathbf{l} = \mathbf{x} \times \mathbf{x}' \quad (2.3)$$

kde \mathbf{l} je vektor výslednej priamky a \mathbf{x}, \mathbf{x}' sú dané body. [6]



Obr. 2.2: Priamka prechádzajúca 2 bodmi

2.1.5 Stupne voľnosti

Stupne voľnosti definujú počet jednoznačne definujúcich a nezávislých parametrov objektu.

Bod je jednoznačne definovaný 2 súradnicami (x, y) , má teda dva stupne voľnosti. Pre priamku v projektívnej rovine táto vlastnosť platí analogicky, avšak priamka je definovaná podľa vzťahu 2.1 dvoma nezávislými parametrami – 2 dvoma pomermi $\{a : b : c\}$, má teda tiež 2 stupne voľnosti.

2.2 3D \rightarrow 2D Projektívna geometria

2D projektívna geometria skúma vlastnosti projektívnej roviny, ktoré sú invariantné voči projektívnym transformáciám. V tejto kapitole budú vysvetlené a popísané základné pojmy a javy projektívneho zobrazenia. [13]

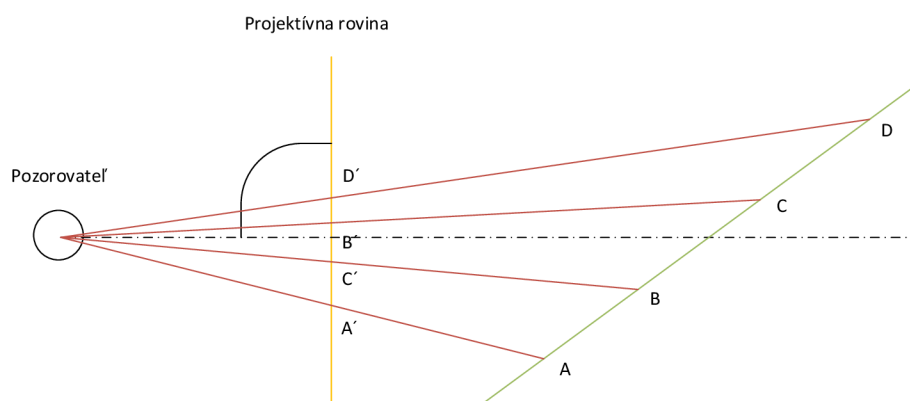
2.2.1 Obrazová rovina

Pojem projektívna rovina zavedieme kvôli spôsobu, akým je zachytená 3D scéna do 2D obrazu a snahe o jeho popis. V našom prípade bude táto rovina kolmá na optickú os a rovnobežná s rovinou optického senzoru a premietne sa do nej snímaná scéna so skreslením, ktorého charakter závisí od kamery a perspektívy pozorovateľa. Je definovaná nasledovne:

Projektívnu rovinu definujeme ako množinu bodov s istými podmnožinami, ktoré nazývame priamkami, a s množinou bodov ležiacich na priamkach, pričom platia tieto pravidlá:

1. Každými dvoma bodmi prechádza práve jedna priamka
2. Každé dve priamky majú práve jeden spoločný bod
3. Existujú také štyri body, z ktorých žiadne tri neležia na jednej priamke.

Na obr. 2.3 je znázornený spôsob, akým sa do projektívnej roviny prevádzajú body zo snímanej scény. Bod A sa transformuje do bodu A', B do B', ..., pričom body A', B', C', D' ležia na priesečníku priamky prechádzajúcej bodmi pozorovateľ – pôvodný bod a projektívnej roviny. Zachovanie pomerov obrazu transformovaného do projektívnej roviny bude popísané v nasledujúcich kapitolách.[4] [13]



Obr. 2.3: Sústava pozorovateľ - projektívna rovina - snímaná scéna

2.2.2 Priesečník rovnobežiek

Priesečník 2 rovnobežiek je špeciálny prípad priesečníka dvoch priamok.

Algebraicky je priesečník definovaný vzťahom 2.2. Môžeme teda napísať, že priesečník priamok $\mathbf{l} = (a, b, c)^\top$ a $\mathbf{l}' = (a, b, c')^\top$, ktoré sú rovnobežné (líšia sa iba parametrom c), leží v bode $\mathbf{x} = \mathbf{l} \times \mathbf{l}' = (c' - c)(b, -a, 0)^\top$, kde rozdiel c -parametrov charakterizuje iba zmena veľkosti, ktorú môžeme v tomto prípade ignorovať.

Ak sa pokúsime previesť takýto bod do nehomogénnych súradníc, výsledkom bude bod $\mathbf{x} = (\frac{b}{0}, \frac{-a}{0})^\top$, kde pri delení 0 dostaneme nekonečne veľké súradnice. Tým sme potvrdili postulát klasickej geometrie o priesečníku dvoch rovnobežiek, ktorých priesečník leží v nekonečne.

Aj keď takýto bod leží v nekonečne, stále ho považujeme za súčasť roviny a nazývame ho „nevlastný bod“. [6]

2.2.3 Nevlastné body

Je to množina bodov, ktoré v homogénnej forme zápisu majú na mieste poslednej súradnice hodnotu 0. Ako bolo povedané v predchádzajúcich kapitolách, ležia v nekonečne a nie je možný ich prevod na nehomogénnu formu, avšak stále **sú** súčasťou projektívnej roviny.

Vanishing point je obrazom nevlastného bodu transformovaného do projektívnej roviny. Pre názornosť viď obr. 2.6. Vznik a existencia takéhoto bodu bude vysvetlená na príklade koľajníc.

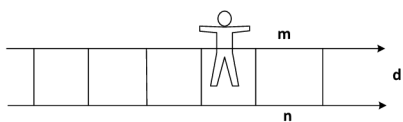
Koľajnice sú z geometrického hľadiska dve rovnobežné priamky ktoré na obr. 2.4 znázorňujú vektory \mathbf{m} a \mathbf{n} . Ich priesečník \mathbf{x} leží v nekonečne, čo dokladá aj obr. 2.4a kde je vidieť, že sú rovnobežné. Na obr. 2.4b je ale zrejme, že skreslením došlo ku prieniku priamok \mathbf{m} a \mathbf{n} v bode V, ktorý je obrazom nevlastného bodu \mathbf{x} v projektívnej rovine. Nazveme ho „vanishing point“ a ďalej bude označovaný už iba VP. Bod C na obr. 2.4b označuje ohnisko kamery a vektor \mathbf{d} je predĺžením úsečky C-V, pričom je rovnobežný s priamkami \mathbf{m} a \mathbf{n} . [4]

Na obr. 2.5 sú vyššie popísané skutočnosti zobrazené na obrazoch z reálneho sveta.

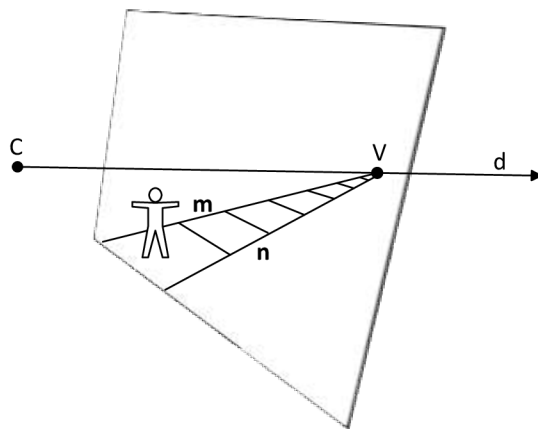
Množinu všetkých nevlastných bodov VP možno zapísať tvarom $(x, y, 0)$, kedy každý bod bude definovaný pomerom $x : y$. [4]

2.2.4 Nevlastná priamka, Vanishing line

Cez všetky nevlastné body prechádza jedna priamka, ktorá leží v nekonečne a má tvar $l_\infty = (0, 0, 1)^\top$. Takúto priamku nazývame „nevlastná priamka“. Znázornená je na obr. 2.6 hore. Nevlastná priamka definuje zároveň aj horizont roviny. [4]

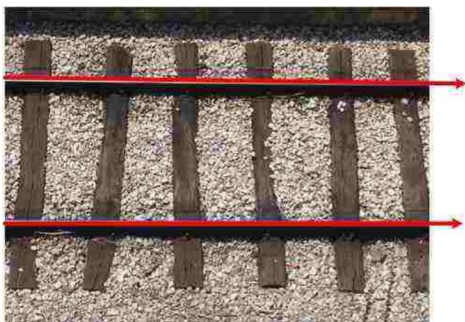


(a) Reálne koľajnice

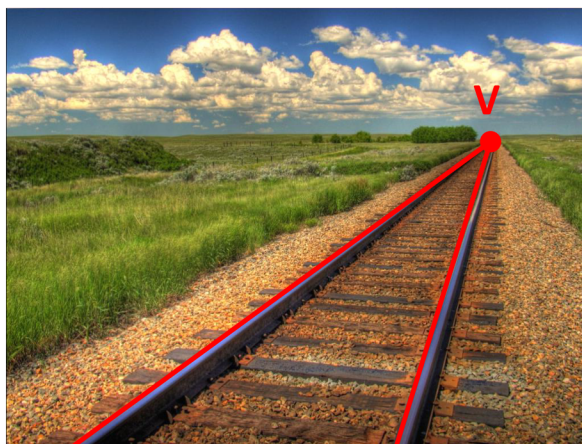


(b) Teoretická schéma

Obr. 2.4: Grafické znázornenie koľajníc

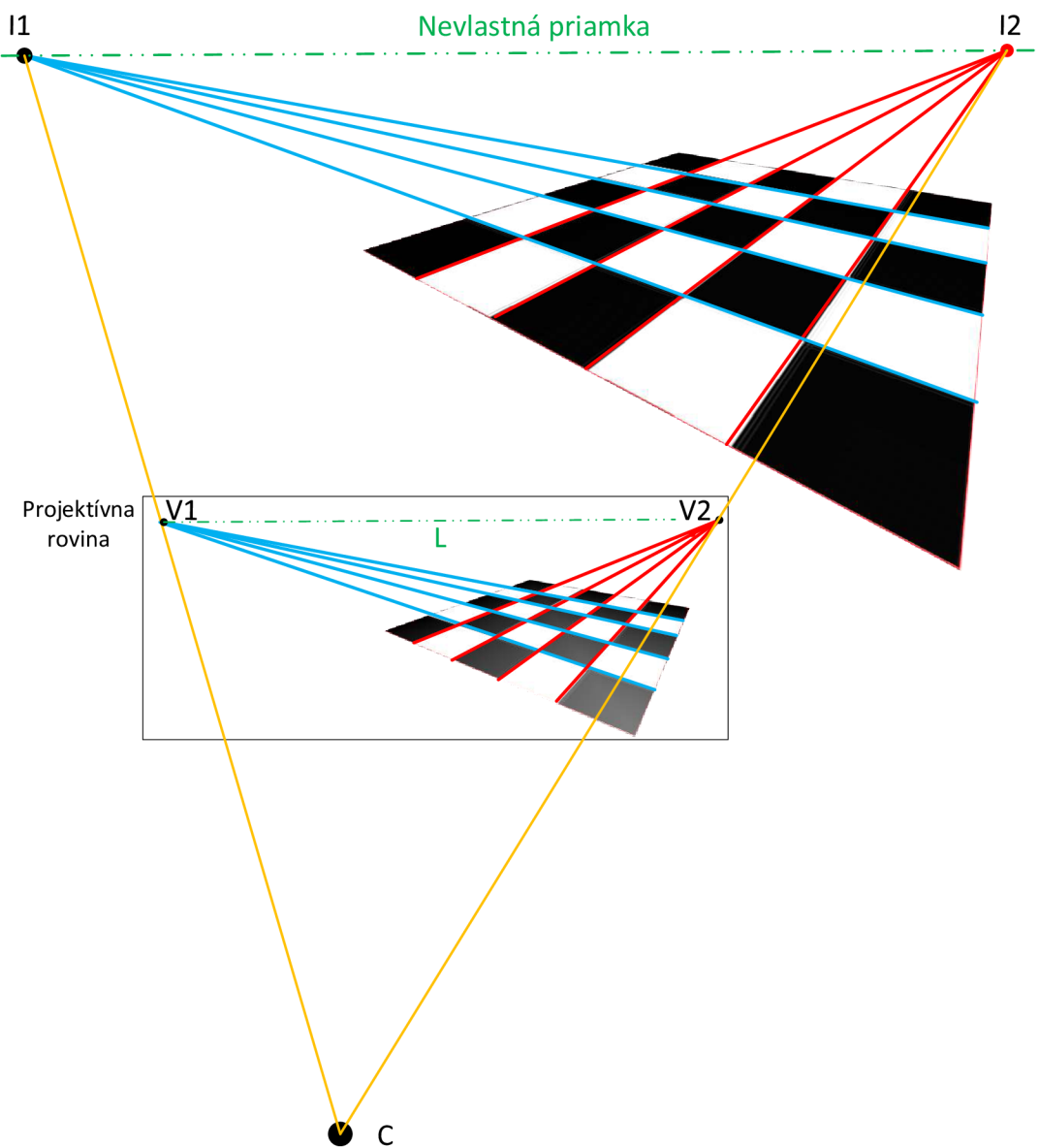


(a) Reálne koľajnice



(b) Vanishing point v obraze

Obr. 2.5: Obrazy z reálneho sveta



Obr. 2.6: Vzájomné vzťahy scény a projektívnej roviny.

Obr. 2.6 znázorňuje vzťahy medzi:

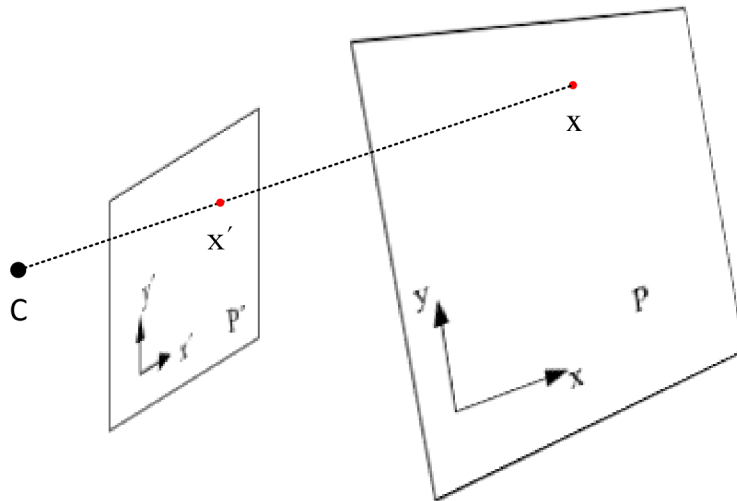
1. nevlastnými bodmi (I1 a I2) a VP (V1 a V2)
2. nevlastnou priamkou a vanishing line (VL)- označená L
3. objektmi z bodov 1,2 a ohniskom kamery C

[4]

2.3 Geometrické transformácie

Zachytenie priestorového objektu do roviny je možné považovať za geometrickú transformáciu. Takáto transformácia generuje geometrické skreslenie, ktoré je možné odvodiť z princípu zobrazovania perspektívou kamery.

Podľa definície je to invertibilné zobrazenie $h: \mathbb{P}^2 \rightarrow \mathbb{P}^2$, ktoré platí iba v prípade, že existuje taká nesingulárna (determinant $\neq 0$) matica \mathbf{H} o veľkosti 3×3 , že pre každý bod v \mathbb{P}^2 definovaný vektorom \mathbf{x} platí $h(\mathbf{x}) = \mathbf{H}\mathbf{x}$.



Obr. 2.7: Centrálne transformácie bodu $\mathbf{x} \rightarrow \mathbf{x}'$ podľa bodu C

Z hľadiska spôsobeného skreslenia môžeme rozdeliť podľa jednoduchosti geometrické transformácie na 3 základné druhy:

1. Euklidovské transformácie
2. Afinné transformácie
3. Projektívne transformácie

Všetky geometrické transformácie mapujú bod z jedného vektorového priestoru do iného. Vzhľadom na skutočnosť, že nie všetky výsledky transformácie súradníc

majú za výsledok celé číslo (nezapadajú do pôvodného celočíselného súradnicového systému), je nutné takéto body vhodne interpolovať.[9] [13]

2.3.1 Euklidovské transformácie

Najjednoduchšie a najpoužívanejšie lineárne transformácie. Je možné s nimi možné realizovať dva druhy operácií:

1. Transláciu (posunutie): jedná sa o posunutie bodu $\mathbf{z} = (x, y)$ do bodu $\mathbf{z}' = (x', y')$ kedy

$$\begin{aligned}x' &= x + t_x \\y' &= y + t_y\end{aligned}\tag{2.4}$$

kde hodnoty t_x, t_y predstavujú veľkosti posunutia v jednotlivých osách. Vektorovo je to možné zapísať ako

$$\mathbf{z}' = \mathbf{z} + \mathbf{t}\tag{2.5}$$

kde vektor $\mathbf{t} = \begin{bmatrix} t_x \\ t_y \end{bmatrix}$ je vektor posunutia.

2. Rotáciu (otočenie): jedná sa o otočenie bodu, pričom os otáčania leží v počiatku sústavy súradníc. Zjednodušene je rotáciu možné zapísať ako

$$\begin{aligned}\cos \beta &= \frac{x}{r} \\ \sin \beta &= \frac{y}{r}\end{aligned}\tag{2.6}$$

kde (x, y) su súradnice pôvodného bodu, r je priama vzdialenosť k počiatku súradníc a β je uhol otočenia voči vodorovnej osi. Pri rozpísaní goniometrickými funkciami a dosadením dostaneme konečné rovnice pre jednotlivé súradnice

$$\begin{aligned}x' &= x \cdot \cos \alpha - y \cdot \sin \alpha \\y' &= x \cdot \sin \alpha - y \cdot \cos \alpha\end{aligned}\tag{2.7}$$

prícom α je uhol otočenia oproti pôvodnému uhlu voči vodorovnej osi. Následne môžeme napísať

$$\mathbf{z}' = \begin{bmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{bmatrix} \mathbf{z}\tag{2.8}$$

Prevodom do homogénnych súradníc je možné zjednodušiť obe operácie a zlúčiť ich pomocou jednej tzv. transformačnej matice \mathbf{T} , ktorá má tvar

$$\mathbf{T} = \begin{bmatrix} \cos \alpha & -\sin \alpha & t_x \\ \sin \alpha & \cos \alpha & t_y \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}_2^T & 1 \end{bmatrix}\tag{2.9}$$

kde \mathbf{R} je matica rotácie 2.8, \mathbf{t} je vektor posunutia 2.5, $\mathbf{0}_2^T$ je nulový riadkový vektor, ktorý vznikol pri prevode súradníc na homogénnu formu.[9]

Počet DoF euklidovskej transformácie je definovaný počtom nezávislých premenných v transformačnej matici, ktoré sú tri - jeden rotačný a dva translačné.

2.3.2 Afinné transformácie

Afinné transformácie, ináč nazývané aj podobnostné, sú lineárne transformácie, pri ktorých je zachovaná rovnobežnosť 2 priamok v obraze, ale ich rovnobežnosť medzi obrazmi zachovaná byť nemusí. Euklidovské transformácie z predchádzajúcej kapitoly 2.3.1 je možné zaradiť podľa tejto podmienky ako podskupinu, avšak oproti nim sú afinné rozšírené o operácie:

1. Zmena mierky: Zväčšenie alebo zmenšenie je asi najčastejšie používanou operáciou. V praxi sa prejavuje ako zmena veľkosti celého obrazu. Rovnako ako v kap. 2.3.1 je možné vyjadriť takúto transformáciu pomocou transformačnej matice, ktorá má v homogénnych súradniciach tvar

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (2.10)$$

a hodnoty s_x, s_y sú koeficienty zmeny veľkosti v jednotlivých osiach. Podľa ich hodnoty je možné dosiahnuť nasledujúce vlastnosti:

- (a) Zväčšenie: pre $s > 1$.
- (b) Zmenšenie: pre $0 < s < 1$.
- (c) Prevrátenie: $s < 0$.

Pokiaľ $s_x = s_y$, jedná sa o proporcionálne zmeny kedy nedochádza ku zmenám v pomeroch strán.

2. Skosenie: Časté skreslenie, ktoré vzniká keď objekt nie je posunutý ako celok, ale len jedna jeho časť, zatiaľ čo druhá zostáva na pôvodných súradniciach. Transformačná matica má tvar

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & k_x & 0 \\ k_y & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (2.11)$$

kde hodnoty k_x, k_y vyjadrujú mieru skosenia v smere jednotlivých osí.

[9]

Skladanie afinných transformácií

V obraze často dochádza ku kombinácii vyššie spomenutých skreslení na jeden objekt. Aby nebolo nutné tieto operácie realizovať separátne, je možné výslednú trans-

formáciu \mathbf{T} zapísať ako aplikáciu čiastkových transformácií napr. $\mathbf{T}_1, \mathbf{T}_2, \mathbf{T}_3$

$$\mathbf{x}' = \mathbf{T}_3 \cdot (\mathbf{T}_2 \cdot (\mathbf{T}_1 \cdot \mathbf{x})) = \mathbf{T}_3 \cdot \mathbf{T}_2 \cdot \mathbf{T}_1 \cdot \mathbf{x} = \mathbf{T} \cdot \mathbf{x} \quad (2.12)$$

Ak by takýchto operácií bolo viac, takýmto zjednodušením dosiahneme značnej úspory pri množstve výpočtov. Ako sme hovorili už v predchádzajúcich kapitolách, jednotlivé transformácie majú vlastné transformačné matice. Tieto matice je možné zlúčiť do všeobecnej afinnej transformačnej matice, ktorá má tvar

$$\mathbf{T} = \begin{bmatrix} a_{1,1} & a_{1,2} & t_x \\ a_{2,1} & a_{2,2} & t_y \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{t} \\ \mathbf{0}_2^\top & 1 \end{bmatrix} \quad (2.13)$$

kde matica \mathbf{A} je nesingulárna matica, \mathbf{t} je translačný vektor a $\mathbf{0}_2^\top$ je riadkový nulový vektor. Počet DoF takejto transformácie je 6. [9] [13]

2.3.3 Projektívna transformácia

Rovinná projektívna (perspektívna) transformácia je lineárna nesingulárna transformácia homogénnych súradníc. Je zovšeobecnením afinnej transformácie. Transformačnú maticu je možné zapísať ako:

$$\mathbf{T} = \begin{bmatrix} a_{1,1} & a_{1,2} & t_x \\ a_{2,1} & a_{2,2} & t_y \\ v_1 & v_2 & v \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{t} \\ \mathbf{v}^\top & v \end{bmatrix} \quad (2.14)$$

kde matica \mathbf{A} je nesingulárna matica, \mathbf{t} je translačný vektor a \mathbf{v}^\top je koeficient podobný ako pri homogénnej reprezentácii bodu, ktorý nie je možné normovať.

Aj tu pri homogénnej forme zápisu platí, že násobením transformačnej matice \mathbf{T} akýmkoľvek $k \neq 0$ sa nezmení charakter transformácie. Ten je definovaný vzájomnými pomermi členov v matici ktoré zároveň definujú 8 DoF.

Túto transformáciu je možné definovať znalosťou 4 korešpondujúcich bodov v oboch rovinách, pričom tri nesmú ležať na jednej priamke. [9][6] [13]

3 MODEL KAMERY

V kapitole budú objasnené základné vlastnosti zachytávania obrazu pomocou kamery. Tento proces chápeme ako mapovanie 3D priestoru do 2D priestoru projekívnej roviny. Vlastnosti kamery môžeme vyjadriť maticovým zápisom známym z kapitoly 2.3.

3.1 Skreslenie obrazu

V tejto časti budú prebrané najčastejšie prípady skreslenia obrazu vznikajúce usporiadaním scény, optickou sústavou kamery alebo perspektívou.

Kvôli skutočnosti, že iba minimum digitálnych obrazov tieto skreslenia vôbec neobsahuje, stretneme sa s nimi skoro pri každom analyzovanom obraze. Pre potreby tejto práce je preto vhodné objasniť základné príčiny vedúce k najčastejším druhom skreslenia obrazu kvôli pochopeniu výskytu a princípov jeho odstránenia. K jednotlivým skresleniam bude vysvetlená príčina vzniku a v kap. 3.2 spôsob ich potlačenia.

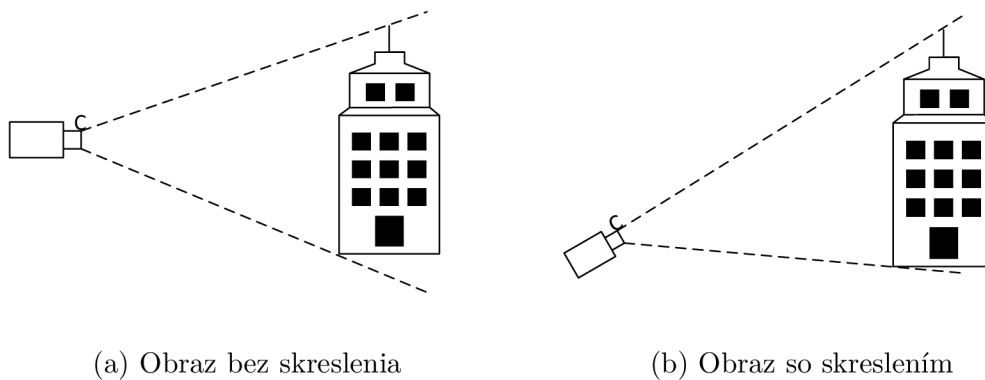
3.1.1 Perspektívne skreslenie

Medzi najčastejšie prípady skreslenia obrazu patrí perspektívne skreslenie. Na rozdiel od ostatných typov, nevzniká v dôsledku optického skreslenia, ale vďaka perspektíve kamery v momente, keď je priestorový objekt transformovaný do roviny snímača.

Súvisí s fyzickou vzdialenosťou objektov od roviny snímača v kamere. Ak je objekt veľmi blízko, môže vyzeráť neproporcionálne veľký alebo deformovaný v porovnaní s objektmi v pozadí. Skreslenie vzniká pomerom vzdialenosti objektu v popredí ku vzdialenosti objektu v pozadí. Tento jav sa prejavuje aj pri jedinom väčšom objekte, ktorého jedna časť je vzdialenejšia ako druhá.

Ako bolo spomenuté vyššie, perspektívne skreslenie súvisí hlavne s fyzickou vzdialenosťou. Pri bežných fotografiách je často takéto skreslenie zosilnené použitím širokouhlých objektívov ktoré majú veľký zorný uhol. Vďaka tomu sú schopné zachytiť množstvo vzdialených a zároveň blízkych objektov v jednom obraze.

Obr. 3.1a znázorňuje spôsob, pri ktorom nedochádza ku perspektívnemu skresleniu, lebo snímaný objekt je kolmý na optickú os kamery a teda je jeho rovina rovnobežná s plochou snímača. Na obr. 3.1b stena budovy nie je kolmá ku optickej osi kamery a vo výslednom obraze bude budova „pretiahnutá“ a smerom nahor sa bude zužovať.



Obr. 3.1: Vznik Perspektívneho Skreslenia

Napriek častým domnienkam perspektívne skreslenie vôbec nesúvisí s ohniskovou vzdialenosťou objektívu, ktorá teoreticky mení iba výrez pôvodného obrazu.

3.1.2 Radiálne skreslenie

Na rozdiel od perspektívneho skreslenia sa jedná o optické skreslenie, ktoré vzniká kvôli chybám a nedokonalostiam lomu lúčov v optických sústavách objektívov. Rozlišujeme 3 hlavné druhy radiálneho skreslenia:

1. Súdkovité skreslenie
2. Poduškovité skreslenie
3. Komplexné skreslenie

Najvýraznejšie sa tieto druhy skreslenia prejavujú u širokouhlých objektívov kvôli použitiu šošoviek so silným zakrivením. [10][6]

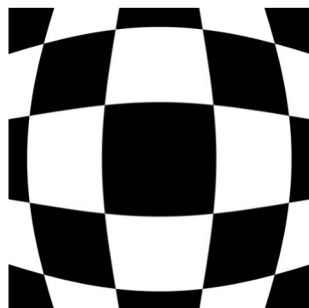
Súdkovité skreslenie

V prípade súdkovitého skreslenia (z angl. barrel distortion) sa znižuje zväčšenie pôvodného obrazu s rastúcou vzdialenosťou smerom od stredu obrazu ako je znázornené na obr. 3.2.

Vzniká hlavne pri použití širokouhlých objektívov, kedy veľkosť zorného poľa objektívu je podstatne väčšia ako veľkosť optického senzoru. Preto treba obraz „stlačiť“. Vďaka tomu vyzerá obraz ako premietnutý na guľu alebo sud. [10][6]

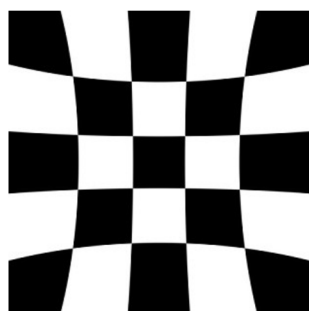
Poduškovité skreslenie

Opačným prípadom od súdkovitého je poduškovité skreslenie (z angl. pincushion distortion) obr. 3.3, kde sa s rastúcou vzdialenosťou od stredu obrazu zvyšuje zväčšenie pôvodného obrazu.



Obr. 3.2: Obrázok so súdkovitým skreslením

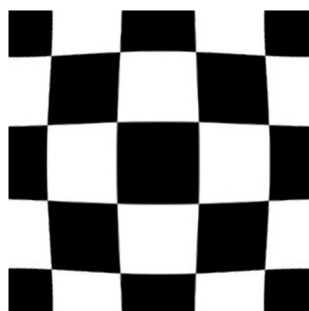
Vzniká hlavne pri použití teleobjektívov, kedy veľkosť zorného poľa objektívu je podstatne menšia ako veľkosť optického senzoru. Preto je treba obrázok „natahnúť“, vďaka čomu vyzerá ako poduška. [10][6]



Obr. 3.3: Obrázok s poduškovým skreslením

Komplexné skreslenie

Špeciálnym prípadom je komplexné skreslenie, ktoré vzniká kombináciou poduškovitého a súdkovitého skreslenia. Od optického stredu začína ako súdkovité a plynule prechádza do poduškovitého skreslenia smerom ku krajom obrázku vid' obr. 3.4. Vyskytuje sa pri objektívoch s veľkým rozsahom ohniskovej vzdialenosti. [10][6]



Obr. 3.4: Obrázok s komplexným skreslením

3.2 Potlačenie skreslenia

Skreslenie je možné vyjadriť pomocou projektívnej transformácie pôvodného ne-skresleného obrazu. Projektívna transformácia je invertibilná operácia a preto je možné takéto skreslenie vrátiť pomocou inverznej transformácie. Výsledkom bude obraz, v ktorom majú objekty zachované pôvodné geometrické vlastnosti.

Je nutné zdôrazniť, že skreslenia nezachovávajú pôvodné geometrické tvary. Preto je odstránenie skreslenia pre realizáciu tejto práce kritické a bez neho nebude možné z obrazu získať presné údaje nutné pre ďalšie spracovanie.

3.2.1 Potlačenie perspektívneho skreslenia

Na potlačenie projektívneho skreslenia sa používa transformácia z kap 2.3.3. Ako už bolo povedané, maticu takejto transformácie je možné určiť pomocou 4 bodov, z ktorých 3 neležia na jednej priamke. Takýmto bodmi sa najčastejšie volia rohy pôvodne pravouhlých objektov, ktorých poloha je prerátaná do správneho pravouhlého tvaru. Zo vzťahov skreslených a nových bodov sa určí transformačná matica ktorá je následne aplikovaná na celý obraz. [6]

3.2.2 Potlačenie radiálneho skreslenia – kalibrácia kamery

Keďže radiálne skreslenie vzniká v dôsledku optických nedokonalostí objektívu, každý typ objektívu v závislosti od svojej aktuálnej konfigurácie zanechá v obraze iné skreslenie. Proces, pomocou ktorého definujeme vzťah medzi súradnicami snímaného priestoru a súradnicami v projektívnej rovine, sa nazýva kalibrácia kamery. Je možné ho realizovať viacerými spôsobmi:

- Fotogrametricky: Kalibrácia prebieha snímaním objektu so známymi vlastnosťami. Väčšinou ide o rovinný vzor, ktorý je snímaný v rôznych orientáciách. Najčastejšie sú vzory šachovnice, sústredných kružníc, či rôzne druhy čiar na objekte.
- Autokalibráciou: Typ kalibrácie, ktorý nevyžaduje znalosť snímanej scény. Využíva viac snímok danej scény, pričom sa nemenia vnútorné parametre kamery. Následne sa medzi nimi zisťuje korešpondencia bodov medzi jednotlivými snímkami.
- Špeciálne typy kalibrácie: využívajú konkrétne vlastnosti objektívov, funkciu zoom a iné. Z praktického hľadiska nie je vhodné ich použiť pre účely tejto práce, preto nebudú detailnejšie rozobraté.

Pre potreby tejto práce bude najvhodnejšia fotometrická kalibrácia.

Kalibrácia kamery

Proces kalibrácie spočíva v určení kalibračnej matice \mathbf{K} . Kalibračná matica \mathbf{K} vzniká súčinom extrinsickej \mathbf{G} a intrinsickej matice \mathbf{P} . Matica \mathbf{P} je definovaná ako

$$\mathbf{P} = \mathbf{A}\mathbf{P}_0\mathbf{G} \quad (3.1)$$

kde \mathbf{A} je matica intrinsických parametrov s tvarom

$$\mathbf{A} = \begin{bmatrix} a_u & 0 & u_0 \\ 0 & a_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.2)$$

kde a_u, a_v určujú zväčšenie v smere osí obrazovej roviny a u_0, v_0 definujú polohu polohu počiatku obrazovej roviny.

\mathbf{P}_0 matica štandardnej normalizovanej perspektívnej projekcie

$$\mathbf{P}_0 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (3.3)$$

Matica \mathbf{G} zahŕňa extrinsické parametre.

$$\mathbf{G} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}_3^\top & 1 \end{bmatrix} \quad (3.4)$$

kde matica \mathbf{R} určuje natočenie kamery voči svetovému súradnicovému systému a \mathbf{t} je vektor popisujúci posunutie stredu premietania voči počiatku.[9] [6]

Po dosadení do vzťahu pre výpočet kalibračnej matice \mathbf{K}

$$\mathbf{K} = \mathbf{P}\mathbf{G} \quad (3.5)$$

dostaneme

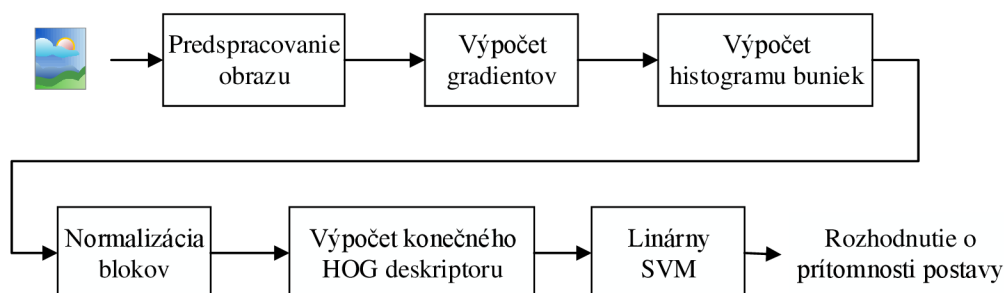
$$\mathbf{K} = \begin{bmatrix} a_u & 0 & u_0 & 0 \\ 0 & a_v & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{r}_1^\top & 0 & 0 & \mathbf{t}_x \\ 0 & \mathbf{r}_2^\top & 0 & \mathbf{t}_y \\ 0 & 0 & \mathbf{r}_3^\top & \mathbf{t}_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} a_u\mathbf{r}_1^\top + u_0\mathbf{r}_3^\top & a_u\mathbf{t}_x + u_0\mathbf{t}_z \\ a_v\mathbf{r}_2^\top + v_0\mathbf{r}_3^\top & a_v\mathbf{t}_y + v_0\mathbf{t}_z \\ \mathbf{r}_3^\top & \mathbf{t}_z \end{bmatrix}. \quad (3.6)$$

4 ALGORITMUS PRE DETEKCIU OSOBY - HOG

Pre detekciu osoby v tejto práci bola vybratá metóda histogramov orientovaných gradientov alebo angl. Histograms of Oriented Gradient (HOG). Hlavná myšlienka metódy spočíva v skutočnosti že objekt, v obraze definovaný istou podobou a tvarom, je možné charakterizovať intenzitou gradientov a vďaka tomu je ho možné detekovať.

4.1 Princíp HOG detektoru

HOG je v tejto metóde použitý ako deskriptor, ktorého hlavným účelom je generalizovať objekt tak, aby rovnaký objekt generoval čo najpodobnejší deskriptor vždy, bez závislosti na podmienkach v obraze (natočenie, svetelné podmienky). Jednotlivé kroky algoritmu znázorňuje bloková schéma na obr. 4.1 a budú detailnejšie popísané v nasledujúcich častiach.



Obr. 4.1: Schéma algoritmu metódy HOG

Výhodou HOG detektoru je jeho jednoduchosť v porovnaní s inými metódami pre detekciu. Hlavným dôvodom je skutočnosť, že na popis hľadaného objektu používa globálny deskriptor ktorý popisuje celý objekt, na rozdiel od metód používajúcich množinu lokálnych deskriptorov.

4.1.1 Hrana, Gradient

Gradient

Gradient sa využíva v oblasti spracovania obrazu na detekciu hrán objektov a smeru v danom bode. Je definovaný ako vektor ∇f parciálnych derivácií v smere osí x a y

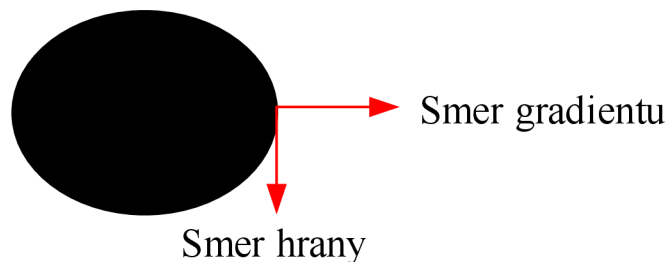
$$\nabla f(x, y) = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right) \quad (4.1)$$

Po získaní vektora je možné vypočítať jeho modul

$$|\nabla f(x, y)| = \sqrt{\left(\frac{\partial f}{\partial x} \right)^2 + \left(\frac{\partial f}{\partial y} \right)^2} \quad (4.2)$$

a smer, daný uhlom ϕ voči horizontálnej osi

$$\phi = \arctan \left(\frac{\frac{\partial f}{\partial x}}{\frac{\partial f}{\partial y}} \right) \quad (4.3)$$



Obr. 4.2: Vzťah gradientu a hrany

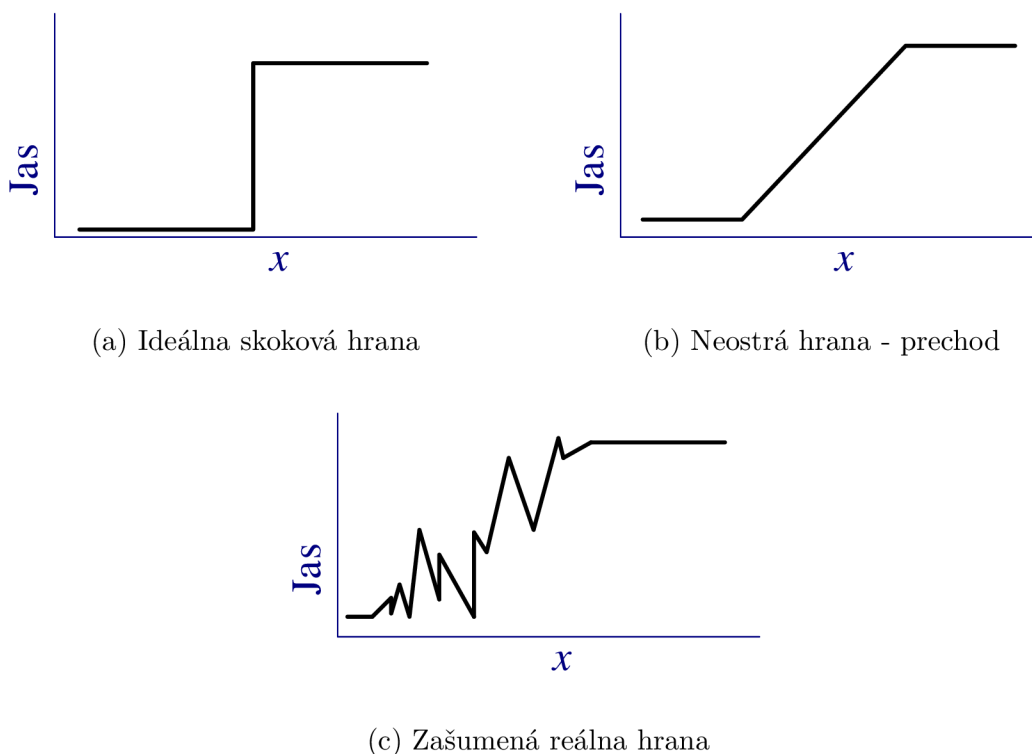
[7]

Hrana

Hrana je oblasť, v ktorej dochádza k výraznej zmene jasových hodnôt. Tieto oblasti obsahujú viac informácií než oblasti s monotónnym jasom.

Pojem hrana vyjadruje zmenu rýchlosti a smeru rastu obrazovej funkcie $f(x, y)$. Podľa priebehu jasovej funkcie môžeme rozdeliť hrany do niekoľkých jasových profilov viď obr. 4.3.

Na obr. 4.3a je idealizovaná hrana, ktorej priebeh je skoková zmena jasou rozdeľujúca susedné pixely na dve skupiny s rozdielnymi hodnotami. V reálnych fotografiách



Obr. 4.3: Druhy hrán

sa skoro nevyskytuje. Priebeh jasů na obr. 4.3c je výsledný obraz po prieniku šumu do priebehu 4.3b.

[7][9]

4.1.2 Predspracovanie obrazu

Autori metódy HOG skúšali rôzne ďalšie možnosti úpravy vstupného snímku, napr. prevod na stupne šedi, použitie farebných priestorov RGB a LAB, gamma ekvalizáciu. Ukázalo sa však ze na výslednú úspešnosť detektoru mali zanedbateľný vplyv. [3]

4.1.3 Výpočet orientovaných gradientov

Ako vyplýva z názvu samotnej metódy, výpočet orientovaných gradientov bude kritický pre ďalšie operácie v detektore. Orientovaný gradient vyjadruje zmeny susedných pixelov v horizontálnom a vertikálnom smere. Kvôli jednoduchosti bude výpočet popísaný na čiernobiely obraz. Prebieha v nasledujúcich krokoch:

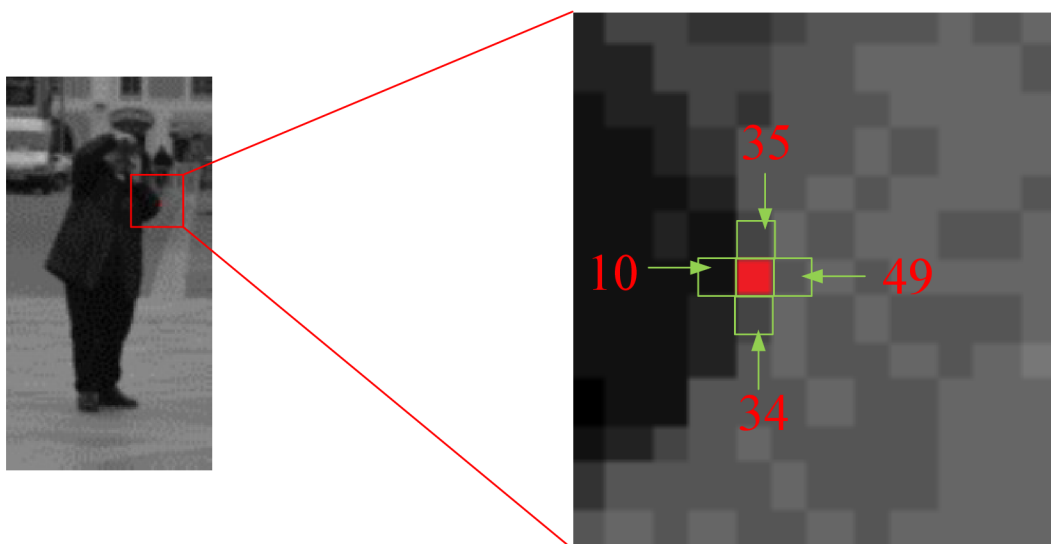
1. Výber pixelu.
2. Zistenie hodnôt jasů horizontálnych a vertikálnych susedných pixelov.

3. Výpočet gradientu pomocou vzťahu 4.1. Najjednoduchšie je tento výpočet aproximovať rozdielom hodnôt susedných pixelov. Na obr. 4.4 je to v horizontálnom smere $49 - 10 = 39$ a vo vertikálnom smere $35 - 34 = 1$. Tým dostávame vektor gradientu $\begin{pmatrix} 39 \\ 1 \end{pmatrix}$ pre daný bod.
4. Výpočet pre modul gradientu je daný vzťahom

$$v = \sqrt{(x)^2 + (y)^2} = \sqrt{(39)^2 + (1)^2} = 40 \quad (4.4)$$

a uhol gradientu voči vodorovnej osi x je definovaný vzťahom

$$\phi = \arctan \frac{y}{x} = \arctan \frac{1}{39} = 0,0263 \text{ rad} = 9,471^\circ \quad (4.5)$$



Obr. 4.4: Spôsob určenia gradientu vektora pre jednotlivé pixely.

Veľkosť uhla gradientu sa môže pohybovať v rozsahoch $0^\circ - 360^\circ$ alebo $0^\circ - 180^\circ$. Výkon deektoru je citlivý na masky, pomocou ktorých sa gradient počítal. Najlepším riešením pre autorov [3] sa ukázala najjednoduchšia maska $[-1,0,1]$, ktorá je použitá aj v príklade.

[3] [12]

4.1.4 Normalizácia orientovaných gradientov

Pri zmenách jasu a kontrastu vplyvom napr. predspracovania môže dôjsť ku zmenám, ktoré nebudú ekvivalentné pre celý obraz. Pri výraznejšom zvýšení jasu násobením môže dôjsť ku stavu, kedy sa hodnota svetlých pixelov zväčší viac ako hodnota

tmavých pixelov. V tomto prípade vzrastie kontrast a hrany sa stanú výraznejšie. To zmení aj veľkosť vektorov orientovaných gradientov, čo je nežiadúce.

Preto je nutné vektor previesť na normalizovaný vektor. Toto dosiahneme vydelením vektora jeho veľkosťou. Pre príklad z kap.4.1.3 bude mať normalizovaný vektor hodnotu $\begin{pmatrix} 0,975 \\ 0,025 \end{pmatrix}$. Takýto vektor je už invariantný voči zmenám kontrastu.

4.1.5 Výpočet histogramu buniek

Po výpočte z kapitoly 4.1.3 pre každý pixel sa celý obraz rozdelí do blokov pomocou masky 8×8 px. Tento rozmer a tvar bol zvolený autormi metódy [3] a dosiahol najlepšie výsledky. Následne sú veľkosti jednotlivých gradientov z bunky roztriedené do binov(tried) histogramu podľa smeru. Veľkosť histogramu bola zvolená na 9 binov, z čoho vyplýva citlivosť 20° pre rozsah $0^\circ - 180^\circ$. Táto konfigurácia sa ukázala ako najlepšia[3]

Prínos vektora do histogramu závisí od veľkosti vektora(väčší vektor ovplyvní histogram viac). Ak vektor nemá presne smer binu, je jeho veľkosť pomerne rozdelená medzi oba susedné biny. [3].

4.1.6 Normalizácia blokov

Veľkosť gradientov v jednom obraze sa výrazne líši vďaka zmenám v nasvietení a kontrastu objektu s pozadím. Preto je lokálna normalizácia kontrastu kľúčová pre správne fungovanie detektoru.

Normalizácia je bloková operácia, ktorá nad bunkami rozvinie vzájomne sa prekrývajúcu oblasť 3×3 blokov, každý o rozmere 6×6 px. Vďaka tomu, že sa bloky prekrývajú, každý blok prispeje do výsledného deskriptoru viac krát, čo výrazne zvyšuje presnosť.

Autori metódy HOG [3] vyvinuli 4 rozdielne normalizačné schémy.

Z každého bloku získame vektor histogramu. Tento vektor následne normalizujeme L1 alebo L2 normalizáciou.

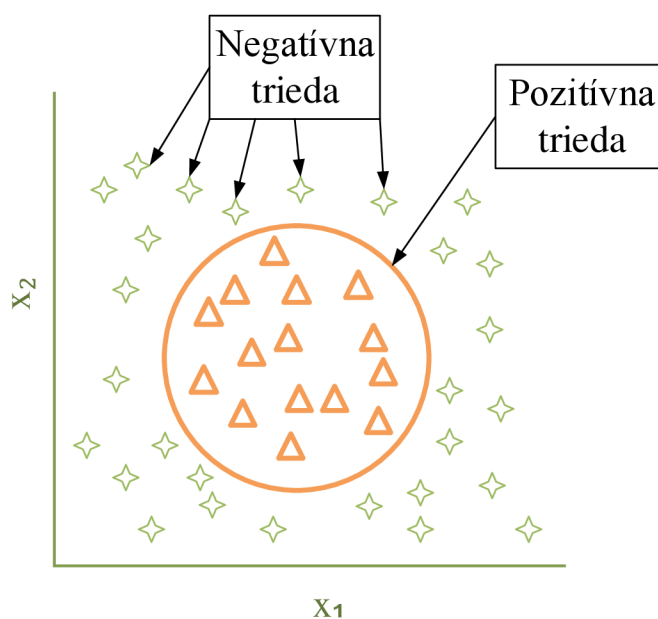
$$\text{L1-norm: } \mathbf{v} \rightarrow \frac{\mathbf{v}}{\sqrt{\|\mathbf{v}\|_1 + \epsilon}} \quad (4.6)$$

$$\text{L2-norm: } \mathbf{v} \rightarrow \frac{\mathbf{v}}{\sqrt{\|\mathbf{v}\|_2^2 + \epsilon^2}} \quad (4.7)$$

kde \mathbf{v} je normalizovaný vektor, ktorý nazývame **deskriptor**. Zvyšné dve normalizačné schémy pozostávajú z kombinácie uvedených dvoch schém a limitáciou výsledku. ϵ je malá konštanta, ktorá nemá veľký vplyv až do veľkých hodnôt.

4.1.7 Support Vector Machine - SVM

Support Vector Machine (SVM) je výkonný učiaci sa algoritmus slúžiaci pre klasifikáciu dát. Ide o tzv. jadrový algoritmus, ktorý sa snaží využiť výhody efektívnych algoritmov pre nájdenie lineárnej hranice a zároveň je schopný reprezentovať zložité nelineárne funkcie. Jednou zo základných myšlienok je prevod pôvodného vstupného priestoru obr. 4.5



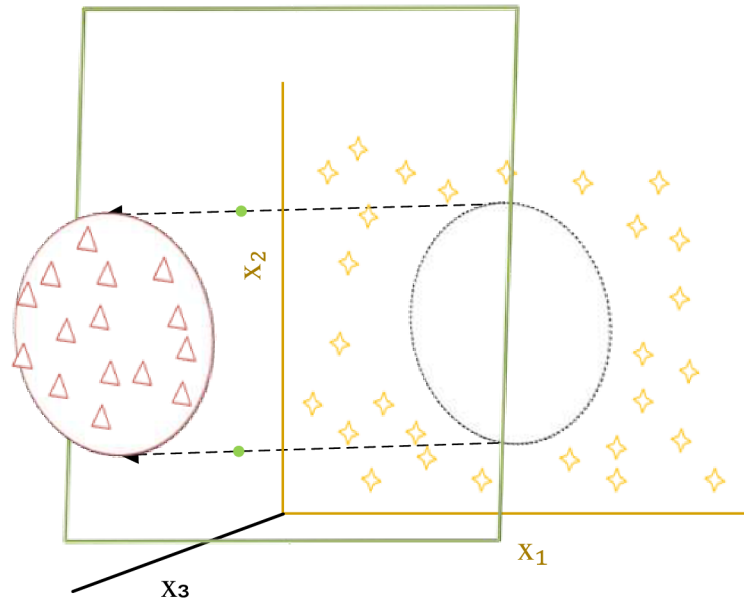
Obr. 4.5: Lineárne neodeliteľné triedy [14]

do viac-dimenzionálneho, kde je možné oddeliť triedy lineárne. Znázornenie takejto myšlienky je na obr. 4.6

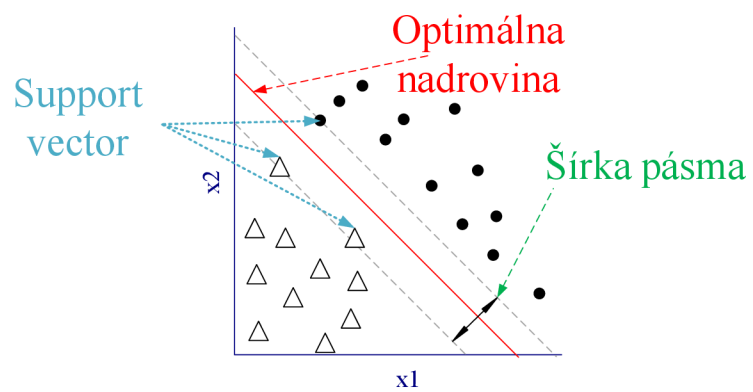
Pridaním novej roviny je nutné nahradiť pôvodné označenie 2 atribútov (x_1, x_2) novým označením definovaným funkciami f_1, f_2, f_3 definovanými

$$f_1 = x_1^2 \quad f_2 = x_2^2 \quad f_3 = \sqrt{2x_1x_2} \quad (4.8)$$

Toto riešenie je komplikované skutočnosťou, že v d -rozmernom priestore bude oddeľovač definovaný rovnicou, ktorá má d -parametrov. Hrozí strata všeobecnosti klasifikátoru „pretrénovaním“. Preto sa táto metóda snaží nájsť taký lineárny oddeľovač, ktorý má čo najširšie písmo medzi pozitívnymi a negatívnymi príkladmi.



Obr. 4.6: Lineárne oddelenie triedy nadrovinou [14]



Obr. 4.7: Pohľad na kolmo na rovinu

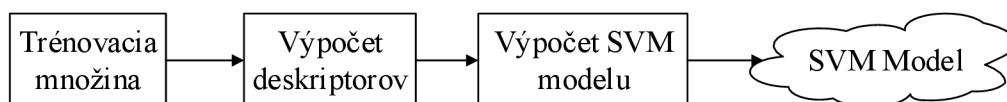
Učenie

Aby bolo možné z množiny deskriptorov určiť či sa v obraze hľadaný objekt nachádza, je nutné najprv vytvoriť dve tréningové sady obrazov:

- Pozitívne: tie, ktoré obsahujú hľadaný objekt
- Negatívne: tie, ktoré neobsahujú hľadaný objekt

Sady musia mať primeranú veľkosť, aby nedošlo ku pretrénovaniu, ani ku stavu, kedy nie je dostatok vzorov pre vytvorenie silného modelu. Obe tieto možnosti znižujú účinnosť detektoru.

Pre tréningové množiny obrazov, u ktorých je vopred známe či obsahujú alebo neobsahujú hľadaný objekt, sú vyrábané deskriptory ktorými sa SVM naučí vyhodnotiť hranicu medzi obrazom, ktorý objekt obsahuje a tým, ktorý nie. Učenie spočíva vo vytvorení SVM klasifikačného modelu – matematickej hyper-plochy – ktorá umožní správne a lineárne rozdeliť tréningovú množinu do vopred známych tried. [8] [2] [14]



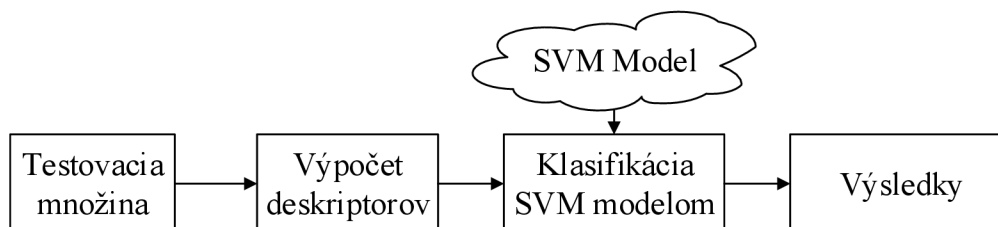
Obr. 4.8: Proces učenia SVM

Detekcia

Detekcia prebieha v 2 hlavných krokoch:

1. Výpočet deskriptoru.
2. Klasifikácia s SVM modelom vytvoreným v časti **Učenie**.

[8] [2] [14]

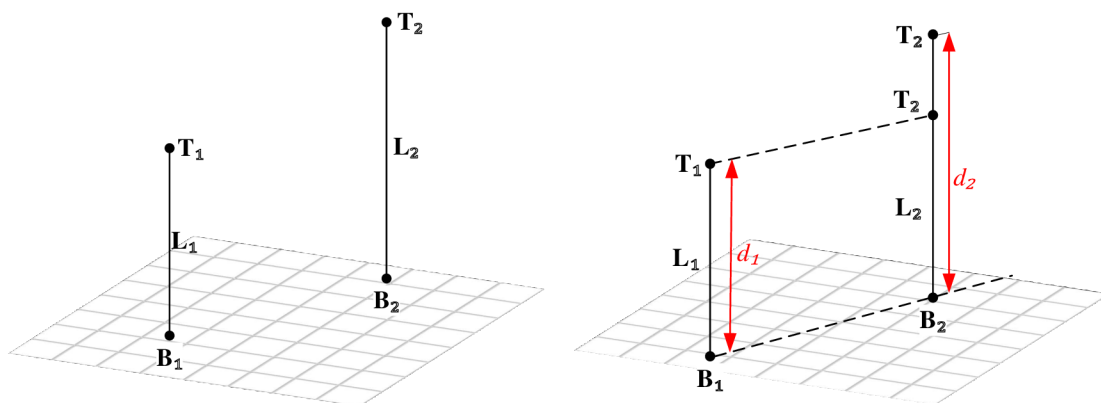


Obr. 4.9: Proces detekcie SVM

5 METÓDA VÝPOČTU VÝŠKY POSTAVY

V tejto kapitole bude popísaná použitá metóda pre výpočet výšky. Metóda bola zvolená z publikácie [6] str.220-223 a využíva pomer známej veľkosti objektu v obraze ku výške detekovanej postavy. Vzhľadom na charakter metódy je nutné aby, známy objekt stál na spoločnej rovine s meranou osobou.

Skutočná situácia v 3D priestore je znázornená na obr. 5.1



(a) Vertikálne segmenty

(b) Vytvorenie bodu \mathbf{T}'

Obr. 5.1: Náčrt priestorovej konfigurácie dvoch výšok voči rovine

Vertikálne segmenty $\mathbf{L}_1 = \langle \mathbf{B}_1, \mathbf{T}_1 \rangle$ a $\mathbf{L}_2 = \langle \mathbf{B}_2, \mathbf{T}_2 \rangle$ z obr. 5.1a reprezentujú dva objekty, z toho jeden má známu výšku. Ich výšky označíme d_1 a d_2 viz. obr. 5.1b. Body $\mathbf{B}_1, \mathbf{B}_2$ ležia v jednej rovine podstavy.

Pre výpočet pomeru $d_1:d_2$ je potrebné vytvoriť obraz \mathbf{T}_1' bodu \mathbf{T}_1 v segmente \mathbf{L}_2 . Tento obraz dostaneme zostrojením priamky prechádzajúcej bodom \mathbf{T}_1 , ktorá je rovnobežná s rovinou podstavy. Jej prienik so segmentom \mathbf{L}_2 je bod \mathbf{T}_1' . Potom vzdialenosť $\langle \mathbf{B}_2, \mathbf{T}_1' \rangle = d_1$.

Tým sme preniesli dĺžku segmentu \mathbf{L}_1 do segmentu \mathbf{L}_2 a môžeme určiť ich pomer.

V projektívnej rovine je tento postup mierne zložitejší kvôli skresleniu, ktoré do obrazu vnáša perspektíva. Vďaka tomu budú mať, v reálnom svete paralelné priamky, v obrazovej rovine kamery priesečník. Tento postup bude popísaný na obr. 5.2 a 5.3 a v nasledujúcich bodoch.

1. Určíme horizontálnu VL I pomocou 2 VP vyrátaných ako priesečník 2 párov rovnobežných priamok v rovine podstavy vid' obr. 2.6.

2. Vyrátame vertikálny vanishing point \mathbf{v} ako priesečník 2 rovnobežných vertikálnych priamok podobne ako v predchádzajúcom bode.
3. Určíme priamky \mathbf{l}_1 a \mathbf{l}_2 ako

$$\mathbf{l}_1 = \mathbf{v} \times \mathbf{b}_1 \quad (5.1)$$

$$\mathbf{l}_2 = \mathbf{v} \times \mathbf{b}_2 \quad (5.2)$$

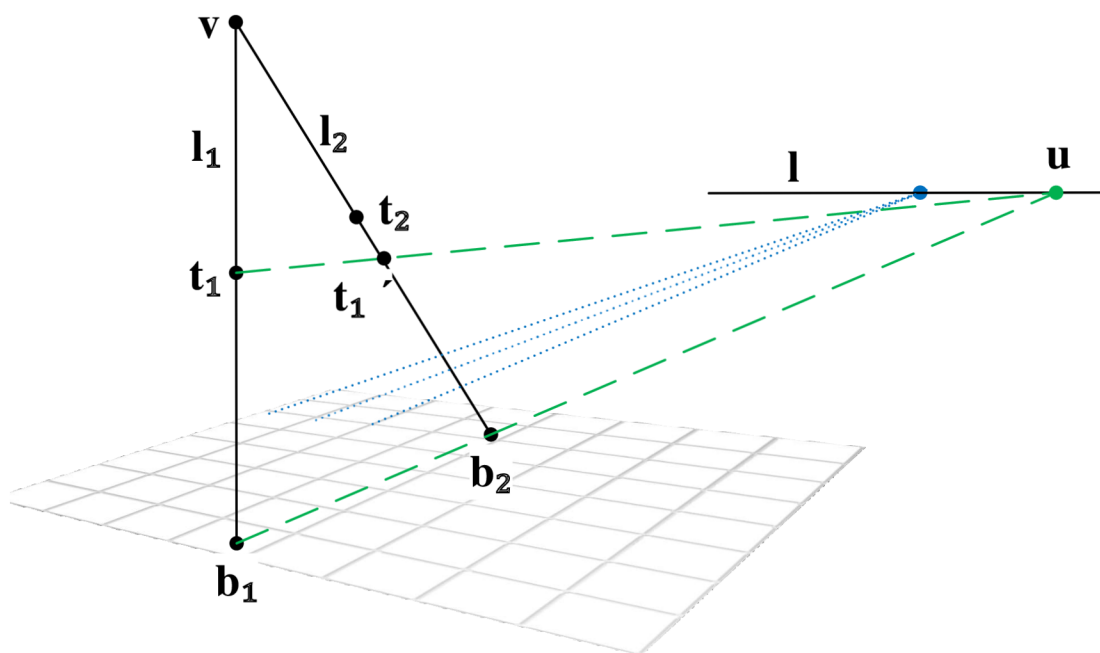
4. Vyrátame vanishing point \mathbf{u} , ako priesečník priamky $\langle \mathbf{b}_1, \mathbf{b}_2 \rangle$ a vanishing line \mathbf{l}

$$\mathbf{u} = (\mathbf{b}_1 \times \mathbf{b}_2) \times \mathbf{l} \quad (5.3)$$

Keďže bod \mathbf{u} je VP a teoreticky leží v nekonečne, každá priamka prechádzajúca ním bude rovnobežná s priamkou $\langle \mathbf{b}_1, \mathbf{b}_2 \rangle$.

5. Vyrátame bod \mathbf{t}_1' , ako priesečník priamky \mathbf{l}_2 s priamkou $\langle \mathbf{t}_1, \mathbf{u} \rangle$

$$\mathbf{t}_1' = (\mathbf{t}_1 \times \mathbf{u}) \times \mathbf{l}_2 \quad (5.4)$$



Obr. 5.2: Zostrojenie horizontálneho a vertikálneho VP, horizontálnej VL a bodu \mathbf{t}_1'

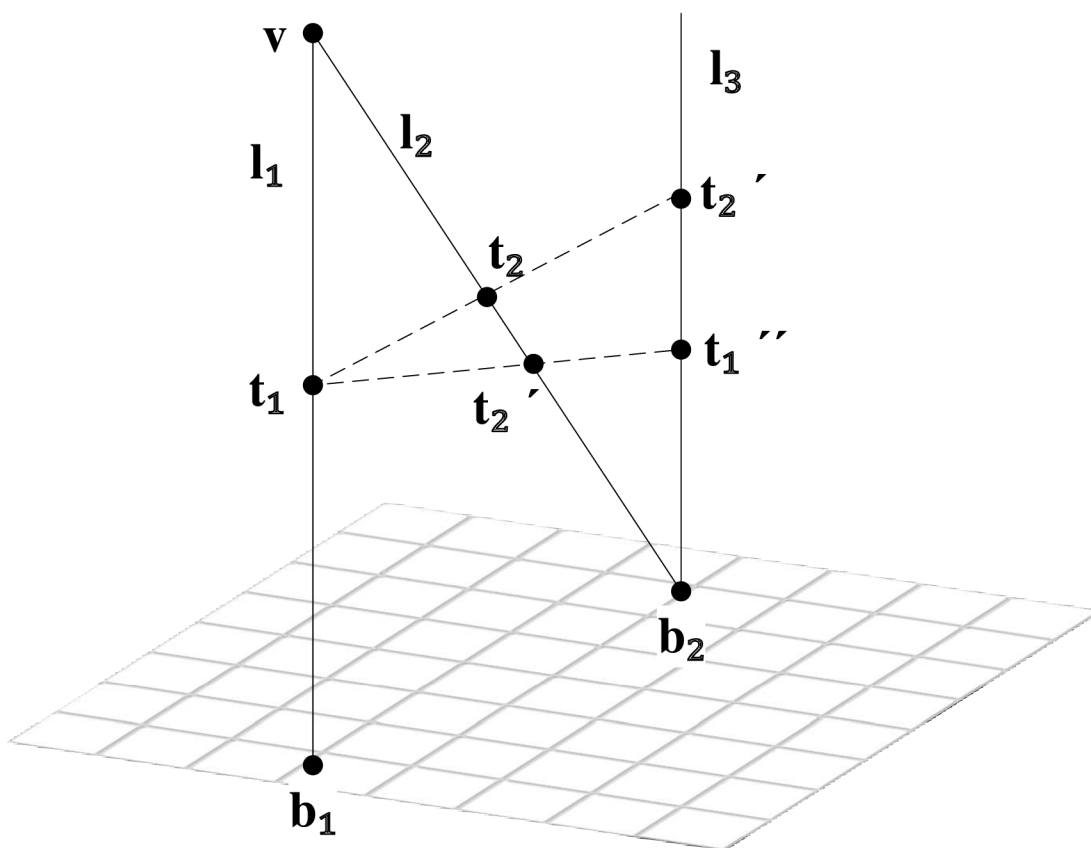
Po získaní bodu \mathbf{t}_1' môžeme prejsť k mapovaniu bodov \mathbf{t}_1' a \mathbf{t}_2 do priamky \mathbf{l}_3 vid obr. 5.3.

6. Zostrojíme priamku l_3 , paralelnú s priamkou l_1 , prechádzajúcou bodom b_2
7. Do priamky l_3 premietneme bod t_1'' ako priesečník priamky $\langle t_1, t_1' \rangle$ s priamkou l_3

$$t_1'' = (t_1 \times t_1') \times l_3 \quad (5.5)$$

8. Zostrojíme bod t_2' , ako priesečník priamky $\langle t_1, t_2 \rangle$ s priamkou l_3

$$t_2' = (t_1 \times t_2) \times l_3 \quad (5.6)$$



Obr. 5.3: Vytvorenie bodu T_1'' a T_2'

9. Vytvorením priamky **13** sme transformovali bod **v** do nekonečna a tým dostali reálny pomer vzdialeností ako

$$\frac{d_1}{d_2} = \frac{\langle t_1'', b2 \rangle}{\langle t_2', b2 \rangle} \quad (5.7)$$

10. Následne môžeme určiť podľa ich pomeru a známej veľkosti jedného z objektov výšku neznámeho objektu, v našom prípade postavy ako

$$\text{výška postavy} = \frac{d_1 * \text{známa výška objektu}}{d_2} \quad (5.8)$$

6 ALGORITMUS PRE VÝPOČET VÝŠKY POSTAVY

Cieľom práce bolo vytvoriť algoritmus ktorý načíta digitálny obraz, vyhľadá v ňom postavu a vypočíta jej výšku na základe známeho zadaného rozmeru ktorý sa v obraze nachádza.

6.1 Úvod

Algoritmus bol implementovaný v jazyku C++ vo vývojovom prostredí Microsoft Visual Studio 2013. Na spracovanie obrazových informácií sme použili knižnicu OpenCV.

Ako kamera bola použitá GoPro HD HERO2, pre ktorú sme na začiatku vytvorili a uložili kalibračnú maticu.

6.1.1 OpenCV

Jedná sa o open source knižnicu špecializovanú na počítačové videnie a spracovanie obrazu (CV=Computer Vision). Je vydaná pod voľnou licenciou BSD, vďaka čomu je možné ju využívať aj na komerčné účely. Podporuje viacero programovacích jazykov rôznych úrovní (Java, C++, C, Python) v 32 aj 64 bitovej architektúre.

V dnešnej dobe je OpenCV jednou z najrozšírenejších knižníc pre spracovanie obrazu na svete. Vďaka vysokej úrovni optimalizácie algoritmov a možnosti využitia výkonu grafickej karty výrazne skracuje výpočtový čas. Toto je aj dôvod prečo bola knižnica vybraná pre potreby tejto práce. V práci je použitá verzia OpenCV 3.0 RC1.[11]

6.2 Algoritmus

Program je fyzicky členený na 4 súbory *.cpp a ich hlavičkové súbory *.h, pričom každá vykonáva jeden logický krok v algoritme.

Program ďalej obsahuje hlavnú funkciu `main` nachádzajúcu sa v súbore `Main.cpp`, v ktorej sú inicializované a volané jednotlivé inštancie tried. Obsah tried bude rozobratý v nasledujúcich kapitolách.

Algoritmus pracuje v týchto základných krokoch:

1. Načítanie obrazu.
2. Trieda

`CameraCalibration`

Odstránenie skreslenia pomocou kalibračnej matice.

3. Trieda

`PersonDetector`

Detekcia osoby v obraze pomocou metódy HOG a SVM.

4. Trieda

`VanishingPointDetector`

Výpočet VP:

- (a) Detekcia hrán Canny detektorom.
- (b) Detekcia priamok pomocou Hough pravdepodobnostnou transformáciou.
- (c) Analýza a výber vhodných priamok.
- (d) Výpočet 3 VP v 3 rozmeroch.

5. Trieda

`PersonMeasurement`

Výpočet výšky detekovanej osoby:

- (a) Výpočet VL.
- (b) Výpočet bodov a pomerov.
- (c) Výpočet výšky z pomeru.

6.2.1 Metóda `Main()`

Hlavná metóda celého algoritmu. Pre užívateľa vytvára jednoduché textové menu s hierarchickou štruktúrou. Načíta vstupný obraz, vytvára a inicializuje jednotlivé inštancie potrebných tried a predáva ich produkty medzi sebou.

6.3 Odstránenie skreslenia

O kalibráciu obrazu sa stará trieda *CameraCalibration*, ktorá obsahuje metódy slúžiace na vytvorenie, uloženie a nahranie kalibračnej matice. Následne umožňuje aj jej aplikáciu na skreslený obraz a jeho korekciu.

6.3.1 Kalibrácia kamery

Kalibračnú maticu je možné vytvoriť dynamicky pomocou napr. webkamery alebo nahratím snímok kalibračného vzoru.

Pre oba spôsoby kalibrácie sme vytvorili vhodné metódy. Pri kalibrácii webkamerou v reálnom čase však vznikali časté problémy s rozmazaním a skreslením

obrazu pri pohybe kalibračného vzoru pred kamerou. Preto sa táto metóda javí ako nevhodná.

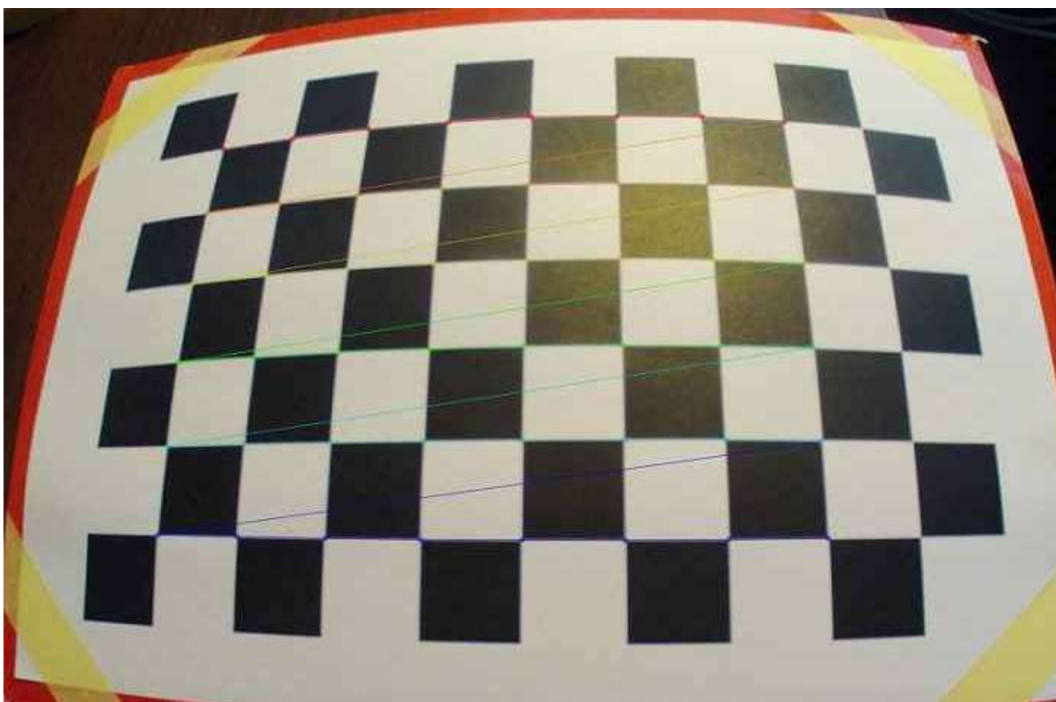
V našom prípade bola ako kalibračný vzor použitá šachovnica. O detekciu jednotlivých bodov kalibračného vzoru sa stará metóda `findChessboardCorners()`

```
bool findChessboardCorners( InputArray image ,  
Size patternSize , OutputArray corners , int flags )
```

kde

- `InputArray image`
-Vstupný obraz v ktorom chceme kalibračný obrazec vyhľadať.
- `Size patternSize`
-Rozmery šachovnice v kalibračnom obrazei.
- `OutputArray corners`
-Súradnice detekovaných bodov šachovnice v obraze.
- `int flags`
-Parametre určujúce doplnkové vlastnosti vyhľadávania.

Výstup z tejto metódy sú body známej šachovnice posunutú o radiálne a perspektívne skreslenie ako napr. na obr. 6.1. . Tieto detekované body sú následne



Obr. 6.1: Jednotlivé body kalibračného obrazca

odoslané metóde ktorá sa stará o samotnú kalibráciu – `calibrateCamera()`:

```
double calibrateCamera(InputArrayOfArrays objectPoints ,
InputArrayOfArrays imagePoints , Size imageSize ,
InputOutputArray cameraMatrix , InputOutputArray distCoeffs ,
OutputArrayOfArrays rvecs , OutputArrayOfArrays tvecs ,
int flags=0)
```

kde

- `InputArrayOfArrays objectPoints`
- predstavuje vektor vektorov súradníc bodov v kalibračných obrazoch. Každý vektor predstavuje jednu fotografiu kalibračného obrazca.
- `InputArrayOfArrays imagePoints`
- predstavuje vektor vektorov súradníc bodov po projektívnej transformácii v danej snímke. Každý vektor predstavuje jednu fotografiu kalibračného obrazca.
- `Size imageSize`
- Veľkosť obrazu.
- `InputOutputArray cameraMatrix`
- Výstupná matica intrinsických parametrov.
- `InputOutputArray distCoeffs`
- Výstupný vektor koeficientov skreslenia.
- `OutputArrayOfArrays rvecs`
- Výstupný vektor rotačných vektorov určených pre každú fotografiu kalibračného obrazca zvlášť.
- `OutputArrayOfArrays tvecs`
- Výstupný vektor translačných koeficientov určených pre každú fotografiu kalibračného obrazca zvlášť.
- `int flags=0`
- Parameter, ktorý v práci nebol využitý, hodnota bola nastavená na 0.

Táto metóda pomocou zadaných parametrov a znalosti vlastností kalibračného vzoru vypočíta skreslenie, ktoré do obrazu vniesla optická sústava kamery. Z tohoto skreslenia následne určí kalibračnú maticu kamery.

Dĺžka kalibrácie môže trvať od niekoľko sekúnd až do niekoľko hodín v závislosti od rozlíšenia a počtu kalibračných snímok.

6.3.2 Korekcia obrazu

Na odstránenie skreslenia slúži metóda OpenCV `undistort()`, ktorá je volaná v metóde `RemoveDistortion()` triedy *CameraCalibration*.

```
undistort(InputArray src , OutputArray dst ,
InputArray cameraMatrix , InputArray distCoeffs ,
```



```
int flags=0)
```

kde

1. `InputArray src, OutputArray dst`
-Vstupný (src) a výstupný (dst) obraz.
2. `InputArray cameraMatrix`
-Matica intrinsických parametrov (výstup z metódy `calibrateCamera()`).
3. `InputArray distCoeffs`
- Vektor koeficientov skreslenia (výstup z metódy `calibrateCamera()`).

Na obr. 6.2a je zjavné zakrivenie hlavne v krajných oblastiach, avšak zreteľné je aj na zakrivení pôvodne rovných hrán. Vzniká kvôli vlastnostiam objektívu typu „rybie oko“ ktorým je použitá kamera GoPro vybavená.

V obr. 6.2b je toto skreslenie odstránené rovnako ako aj zakrivenie reálne rovných hrán v obraze.



(a) Pred korekciou



(b) Po korekcii

Obr. 6.2: Ukážka korekcie obrazu

6.4 Detekcia postavy

Pre vyhľadávanie postavy bola v práci zvolená metóda HOG v kombinácii s SVM. Princíp tejto metódy je popísaný v kap. 4.1 a kap. 4.1.7.

Knižnica OpenCV obsahuje natívne detektor postáv s vytvoreným deskriptorom, preto nie je potrebné vytvárať nový. Na tento účel slúži metóda `detectMultiScale()` objektu *HOGDescriptor*.

```
HOGDescriptor.detectMultiScale(const Mat& src,  
vector<Rect>& found_locations, double hit_threshold=0,
```

```
Size win_stride=Size(), double scale0=1.05,  
int group_threshold=1)
```

kde

1. `Mat& src`
-Zdrojový obraz pre detektor.
2. `vector<Rect>& found_locations`
-Výstupný vektor typu `Rect` (obdĺžnik) ohraničuje hľadaný objekt.
3. `double hit_threshold=0`
-Hraničná vzdialenosť medzi objektom a klasifikačnou rovinou SVM z kap.-4.1.7. V práci bola použitá hodnota 0.0.
4. `Size win_stride=Size()`
-Strieda vyhľadávacieho okna. V práci použitá hodnota (8,8).
5. `double scale0=1.05`
-Koeficient zväčšovania vyhľadávacieho okna. V práci bola použitá prednastavená hodnota 1.05.
6. `int group_threshold=1`
-Koeficient regulujúci podobnosť pri detekcii toho istého objektu pri rôznych veľkostiach vyhľadávacieho okna. V práci sme použili hodnotu 1 čo znamená, že pri nájdení viacerých prekrývajúcich sa objektov budú tieto obdĺžniky reprezentované ako jeden.

Vyššie uvedená metóda vráti vektor obdĺžnikov pozitívne identifikovaných objektov. Vzhľadom na spôsob metódy je vhodné skontrolovať daný vektor a odstrániť objekty ktoré sú falošne identifikované, prípadne malé natoľko, že nemá zmysel uvažovať ich pri ďalších výpočtoch. Na to sme použili kritérium plochy, ktoré objekt zaberá. Plocha nájdeného objektu musí byť minimálne 1/10 z celkovej plochy obrazu.

```
if(foundFiltered.at(i).area() <  
    ((_frame.cols*_frame.rows) / 10) ) {  
    foundFiltered.erase(foundFiltered.begin() + i);  
}
```

HOG detektor vracia mierne väčšie ohraničenia ako reálne sú, preto je ešte vhodné zmenšiť nájdený objekt hlavne kvôli presnosti matematických výpočtov v ďalších krokoch algoritmu.

```
//Found rectanlge r  
Rect r = foundFiltered[i];  
r.x += cvRound(r.width*0.1);  
r.width = cvRound(r.width*0.8);  
r.y += cvRound(r.height*0.09);
```

```
r.height = cvRound(r.height*0.73);
```

Konkrétne hodnoty zmenšenia boli čiastočne prebraté z tutoriálov OpenCV a mierne upravené pre potreby tejto práce.

6.5 Detekcia vanishing pointov

Automatická detekcia VP je problematická hlavne pre priradenie správnych priamok ku správne VP. V tejto práci boli vzhľadom na typ scény zvolené kritéria popísané v nasledujúcich kapitolách. Pre iný druh scény však tieto kritériá platiť nemusia.

O výpočet súradníc VP sa stará trieda *VanishingPointDetector*. Obsahuje metódu `computeVanishingPoints()` ktorá postupne vykonáva jednotlivé dielčie kroky.

6.5.1 Metóda `computeVanishingPoints()`

Táto metóda zastrešuje celý proces výpočtu VP. Jej diagram je znázornený na obr 6.3.

Vstupom metódy je farebný 3-kanálový obraz, ktorý je najskôr konvertovaný na šedotónový 1-kanálový. Vďaka tomu je možné ekvalizovať jeho histogram a zlepšiť čitateľnosť pre detektory rozobraté v nasledujúcich odstavcoch.

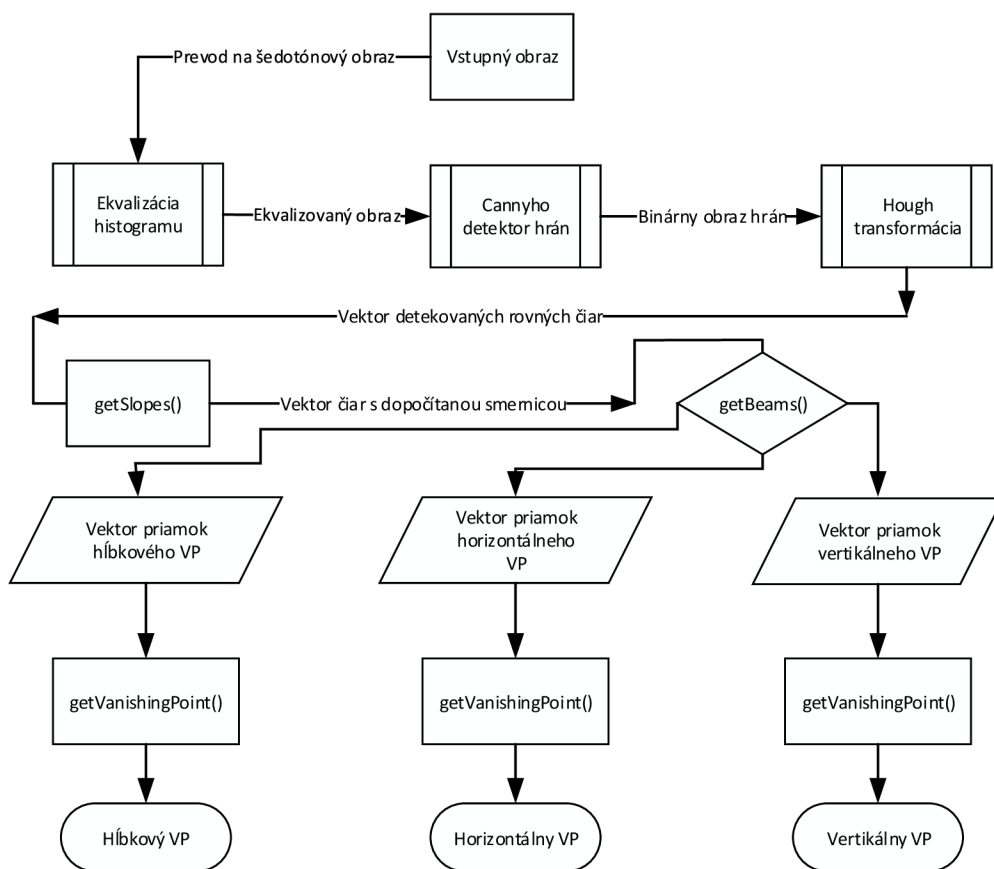
Správne fungovanie a výpočet kriticky závisí na výstupných hodnotách detektorov `Canny()`,

```
void Canny(InputArray image, OutputArray edges,  
double threshold1, double threshold2, int apertureSize=3)
```

kde

- `InputArray image`
-Zdrojový šedotónový obraz pre detektor.
- `OutputArray edges`
-Výstupný binárny obraz.
- `double threshold1`
-Prvá prahová hodnota pre hysteréznú procedúru. V práci bola použitá hodnota 90.
- `double threshold2`
-Druhá prahová hodnota pre hysteréznú procedúru. V práci bola použitá hodnota **$3 * threshold1$** .
- `int apertureSize=3`
-Veľkosť Sobelovho operátora. V práci bola použitá hodnota 3.

ktorý v obraze vyhledá hrany a vráti binárny obraz iba týchto hrán. Binárny obraz vid' obr. 6.4, je následne pomocou detektoru `HoughLinesP()`



Obr. 6.3: Diagram metódy computeVanishingPoints()



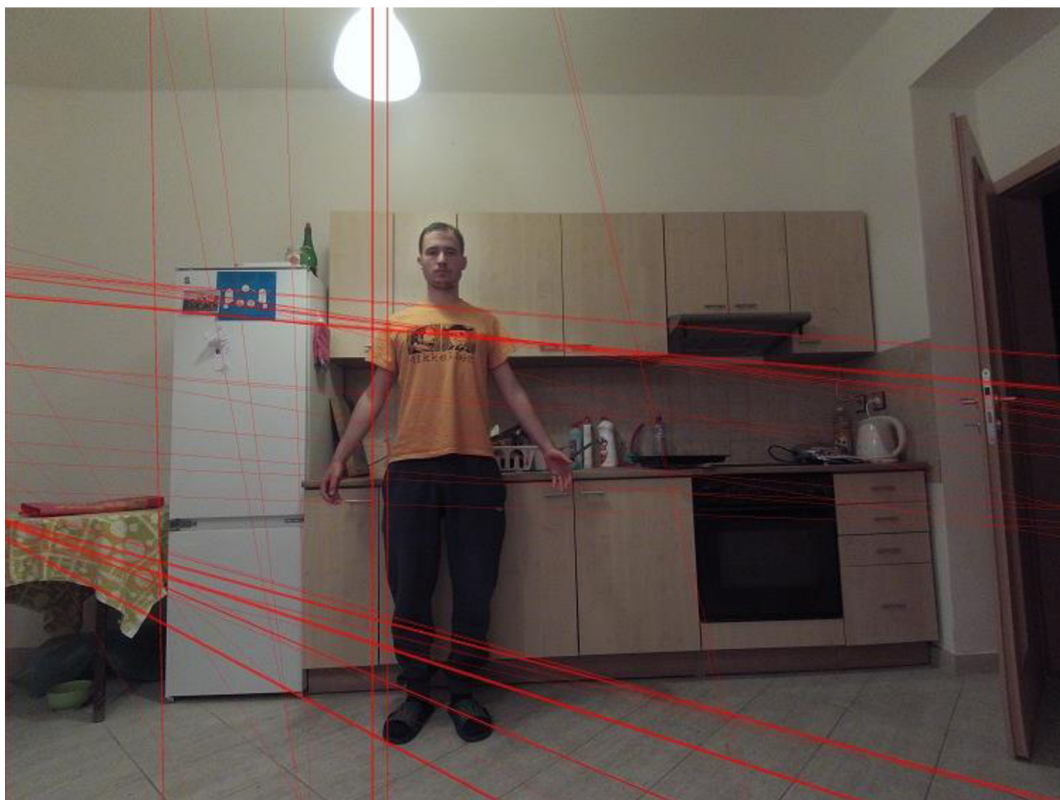
Obr. 6.4: Binárny obraz hrán po detektore *Canny()*

```
HoughLinesP(InputArray image, OutputArray lines,  
double rho, double theta, int threshold,  
double minLineLength=0, double maxLineGap=0 )
```

kde

- `InputArray image`
-Zdrojový binárny obraz pre detektor.
- `OutputArray edges`
-Výstupný vektor pozitívne detekovaných rovných čiar.
- `double rho`
-Rozlíšenie vzdialenosti v px. V práci bola použitá hodnota 1.
- `double theta`
-Rozlíšenie uhlu v radiánoch. V práci bola použitá hodnota $\pi / 180$ čo v praxi znamená hodnotu 1° .
- `int threshold`
-Prahová hodnota prienikov pre detekciu čiary. V práci bola použitá hodnota 150.

- `double minLineLength=0`
-Minimálne množstvo bodov ktoré môžu tvoriť čiaru. V práci bola použitá hodnota 200.
- `double maxLineGap=0`
-Maximálna vzdialenosť medzi dvoma bodmi ktoré môžeme považovať za jednu čiaru. V práci bola použitá hodnota 50.



Obr. 6.5: Priamky detekované detektorom *HoughLinesP()*

analyzovaný na segmenty priamok s parametrami definovanými vyššie. Tieto parametre boli zvolené experimentálne s najlepšimi výsledkami pre testovaný obraz. Výsledky detektoru *HoughLinesP()* môžeme vidieť na obr. 6.5.

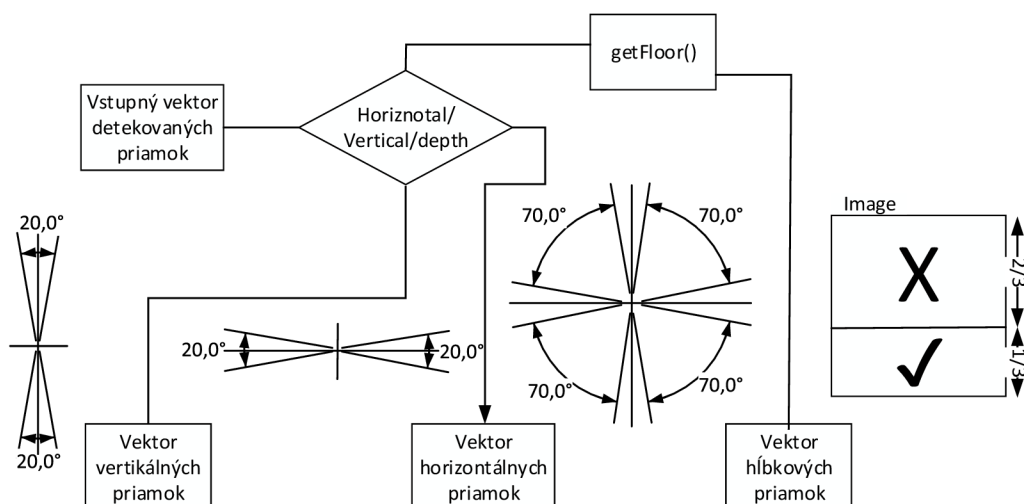
6.5.2 Metóda *getBeams()*

Táto metóda slúži na priradenie jednotlivých detekovaných čiar do jediného zväzku, z ktorého bude možné vypočítať súradnice VP.

Vstupom metódy je vektor priamok z detektoru *HoughLinesP()*, ktorý triedi do 3 nových vektorov podľa kritérií a predpokladov, kladených na polohu a smer každého VP zvlášť. Ak je priamka priradená niektorému z VP, je odstránená z pôvodného

vektoru a priradená do vektoru daného VP. Týmto je zamedzené možnému použitiu jednej priamky pre dva VP.

Na obr. 6.6 je znázornený diagram metódy `getBeams()` spolu s kritériami priradenia do daných vektorov.



Obr. 6.6: Diagram metódy `getBeams()`

Ako základné kritérium delenia sme použili uhol od horizontálnej alebo vertikálnej osi. Pri detekcii čiar definujúcich hĺbkový VP sme však okrem kritéria uhlu museli zaviesť aj kritérium polohy danej čiary, pričom sa táto čiara musela vyskytovať v spodnej tretine obrazu. Po tejto úprave boli výsledky detekcie hĺbkového VP uspokojivé.

Obr. 6.7 znázorňuje priradenie jednotlivých priamok ku konkrétnym VP. Z obrazu je zrejmé prevažne 3 smerov čiar ku 3 rôznym VP:

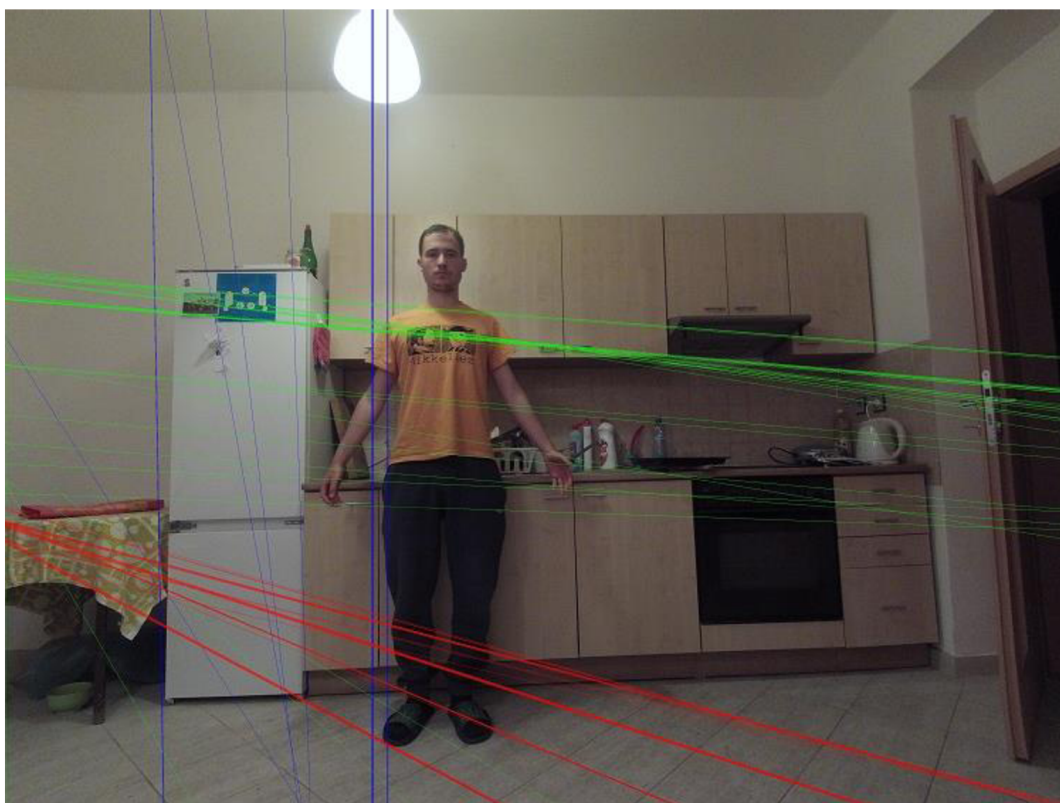
1. Vertikálny (modrá)
2. Horizontálny (zelená)
3. Hĺbkový (červená)

Výstupom metódy su vyššie spomenuté 3 vektory priamok.

6.5.3 Metóda `getVanishingPoint()`

Vstupným parametrom metódy je vektor priamok z výstupu metódy `getBeams()` popísanej v predchádzajúcej kapitole.

Pre každý pár priamok je vypočítaný priesečník. Následne je zo všetkých priesečníkov vypočítaná hodnota mediánu v smere jednotlivých osí (x,y) . Výstupom tejto metódy je bod (x,y) , ktorý považujeme za VP.



Obr. 6.7: 3 rôzne smery priamok pre 3 rôzne VP;

Pôvodne uvažovaná hodnota priemeru sa v praxi ukázala ako nevhodná, kvôli príliš veľkému vplyvu krajne extrémnych hodnôt, ktoré znehodnotili výsledok.

6.6 Výpočet výšky postavy

Pre finálny výpočet výšky detekovanej postavy slúži trieda *PersonMeasurement*. Vstupnými parametrami sú výstupné hodnoty z ostatných tried:

1. Vertikálny VP z triedy *VanishingPointDetector*.
2. Horizontálny VP z triedy *VanishingPointDetector*.
3. Hĺbkový VP z triedy *VanishingPointDetector*.
4. Vektor obdĺžnikov, ktoré ohraničujú detekované osoby z triedy *PersonDetector*.
5. Obraz bez skreslenia pre vstup od užívateľa z triedy *CameraCalibrator*.

Vstupnou metódou triedy *PersonMeasurement* je `setVariables()`, ktorá inicializuje všetky potrebné premenné.

6.6.1 Metóda `setVariables()`

Metóda, ktorá sa stará správnou inicializáciu premenných. Ďalej vykonáva 2 druhy normalizácie:

1. Extrahuje 2 body z obdĺžnika reprezentujúceho detekovanú postavu a načíta ich do typu *Vec4i*. Zároveň zaistí, že prvý uložený je bod ležiaci v rovine podstavy a druhý reprezentuje výšku. Táto operácia je nutná pre zachovanie správneho postupu v ďalších krokoch.
2. Prepočíta horný bod tak, aby ležal na spojnici podstavného bodu a vertikálneho VP. Musí však zachovať pôvodnú vzájomnú vzdialenosť oboch bodov. Tento krok je z hľadiska funkčnosti metódy výpočtu nevyhnutný.

Metóda následne zobrazí obraz s vyznačenými detekovanými osobami a aktivuje možnosť označenia výšky známeho predmetu priamo v obraze.

6.6.2 Metóda `compute()`

Jedná sa o privátnu metódu, ktorá realizuje samotný výpočet výšky detekovanej osoby podľa postupu popísaného v kap. 5. Pre každú detekovanú osobu sa vždy po zadaní referenčného rozmeru v obraze, vypočíta nová výška podľa aktuálnych parametrov.

Po vyrátaní následne metóda zobrazí v konzolovom okne výšku jednotlivých postáv a čaká na ďalší vstup.

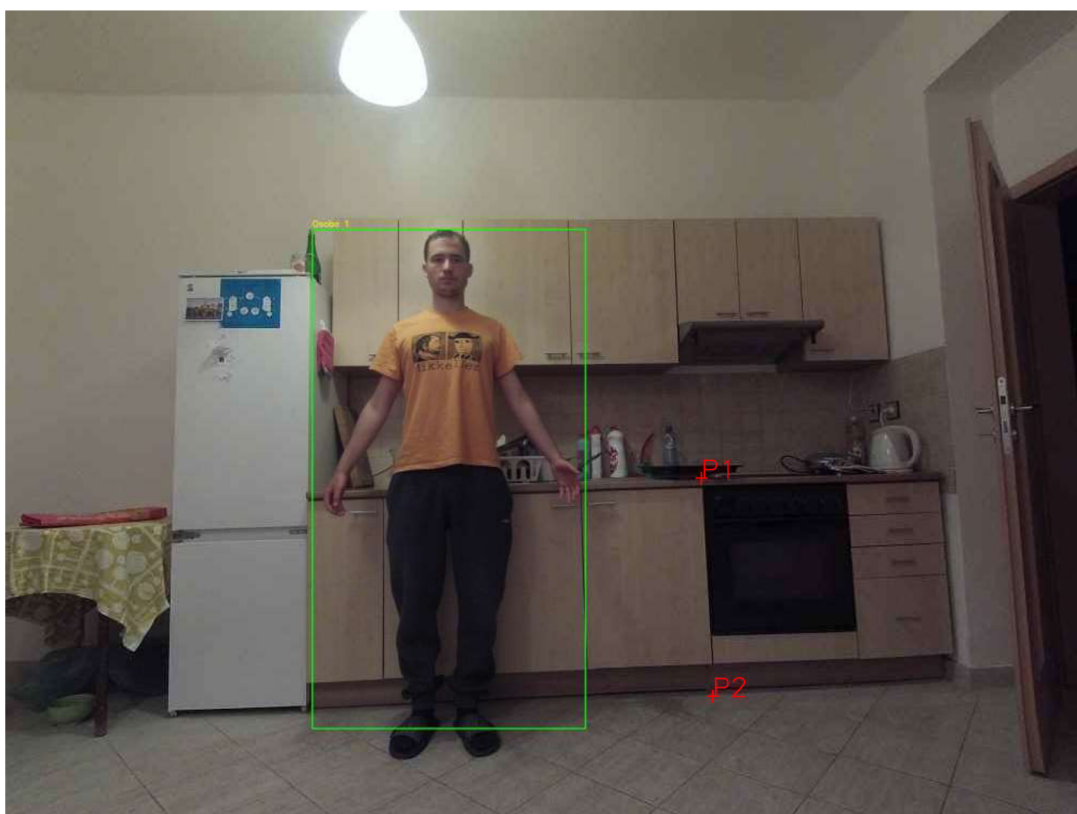
6.7 Vstupy a výstupy

Obmedzené možnosti GUI knižnice OpenCV dovoľujú iba základné interakcie s obrazom, napr. označenie bodu/bodov. Vytvárať plnohodnotné GUI však neumožňujú.

Pre potreby tejto práce je to však postačujúce a pre užívateľa je vytvorené jednoduché stromové textové menu:

1. krok: Voľba módu Meranie/Kalibrácia:
 - (a) mód Meranie: nahrá kalibráciu zo súboru a zobrazí zadaný obraz.
 - (b) mód Kalibrácia: realizuje kalibračný proces a výsledky uloží do nového súboru.
2. krok: Vloženie názvu kalibračného súboru:
 - (a) mód Meranie: ak súbor existuje, je načítaný. Ak nie, program skončí.
 - (b) mód Kalibrácia: ak súbor neexistuje, je vytvorený nový súbor. Ak existuje, bude prepísaný.
3. krok: Vloženie názvu obrazu:
 - (a) mód Meranie: obraz načíta a následne

- i. aplikuje korekciu podľa načítanej konfigurácie
 - ii. prehľadá ho detektorom postáv
 - iii. vyhledá VP
 - iv. zobrazí obraz s vyznačenými detekovanými osobami
- (b) mód Kalibrácia: program očakáva názvy kalibračných obrazov, pre ukončenie vstupu očakáva výraz „kalibruj“.
4. krok:
- (a) mód Meranie: užívateľ označí dva body – P1 a P2 vid' obr. 6.8 – reprezentujúce jeden dĺžkový segment, ktorý predstavuje známy rozmer. Tieto body sú označené užívateľom pomocou myši v obraze. Voľba prvého bodu prebieha ľavým tlačidlom myši, voľba druhého bodu prebieha pravým tlačidlom. Na poradí voľby nezáleží. Pre potvrdenie voľby je nutné stlačiť ľubovoľnú klávesu.

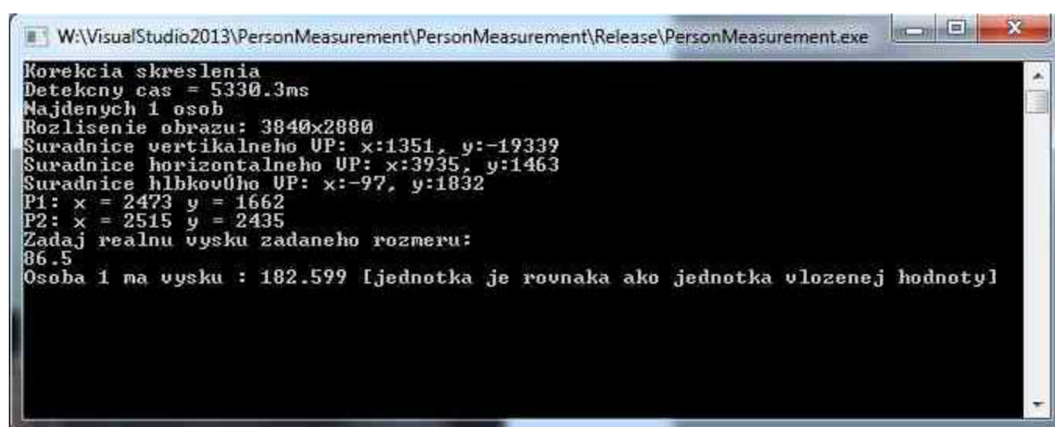


Obr. 6.8: Dva označené body

- (b) mód Kalibrácia: začne kalibračný proces na vložených snímkach.
5. krok:
- (a) mód Meranie: užívateľ vloží numerický vstup do konzolového okna reprezentujúci reálnu výšku rozmeru zvoleného v kroku 4a vid' obr. 6.9.

Pri zadávaní desatinných miest je nutné používať desatinnú bodku, nie čiarku.

- (b) mód Kalibrácia: po skončení kalibračnej procedúry je kalibrácia uložená do kalibračného súboru a program skončí.
- 6. mód Meranie: Následne prebehnú výpočty pre všetky detekované postavy a na výstup konzolového okna sú vypísané všetky vypočítané výšky jednotlivých postáv - obr. 6.9. Jednotka výsledku je vzhľadom na spôsob výpočtu zhodná s jednotkou, v akej bola vložená vstupná hodnota.



```
W:\VisualStudio2013\PersonMeasurement\PersonMeasurement\Release\PersonMeasurement.exe
Korekcia skreslenia
Detekcny cas = 5330.3ms
Najdenych 1 osob
Rozlisenie obrazu: 3840x2880
Suradnice vertikálneho UP: x:1351, y:-19339
Suradnice horizontálneho UP: x:3935, y:1463
Suradnice hĺbkového UP: x:-97, y:1832
P1: x = 2473 y = 1662
P2: x = 2515 y = 2435
Zadaj realnu vysku zadaneho rozmeru:
86.5
Osoba 1 ma vysku : 182.599 [jednotka je rovnaka ako jednotka vlozenej hodnoty]
```

Obr. 6.9: Konzolový vstup a výstup

Po tomto kroku je možné znova zvoliť referenčnú dĺžku a celý výpočet opakovať.

7 VYHODNOTENIE

Na zhodnotenie vlastností metódy sme vybrali dva hlavné princípy výpočtu hodnotiacich parametrov, ktoré budú popísané v nasledujúcich kapitolách. Pre každý rozmer sme meranie realizovali tri krát. Z nameraných hodnôt sme určili priemer, ktorý považujeme za správnu hodnotu.

Zvolené závislosti:

1. Závislosť chyby merania pre jednu referenčnú hodnotu v rôznych vzdialenostiach od meraného rozmeru. Namerané hodnoty sme vyniesli do tab. 7.1.

Číslo merania	Meranie 1 [cm]	Meranie 2 [cm]	Meranie 3 [cm]	Priemer [cm]
0	164,236	168,493	172,803	168,510
1	173,217	173,745	171,936	172,966
2	176,604	176,942	176,556	176,700
3	181,144	179,855	180,794	180,597
4	184,235	185,065	182,564	183,954
5	188,788	189,659	187,361	188,602
6	192,594	193,198	194,457	193,416
7	193,949	194,959	194,708	194,538

Tab. 7.1: Hodnoty vypočítané pre jeden referenčný rozmer v rôznej vzdialenosti

2. Závislosť chyby merania pre rôzne referenčné výšky z obr. 7.3 v dvoch vzdialenostiach od meranej výšky. Namerané hodnoty sme vyniesli do tab. 7.2 a 7.3.

Bod	Meranie 1 [cm]	Meranie 2 [cm]	Meranie 3 [cm]	Priemer [cm]
1	177,057	179,118	180,191	178,789
2	193,227	193,696	194,756	193,893
3	195,28	194,702	194,05	194,677
4	195,558	196,335	196,541	196,145
5	199,206	196,848	198,38	198,145
6	199,096	198,818	199,557	199,157
7	200,741	201,821	201,382	201,315
8	202,34	203,037	202,63	202,669
9	217,589	219,151	220,547	219,096

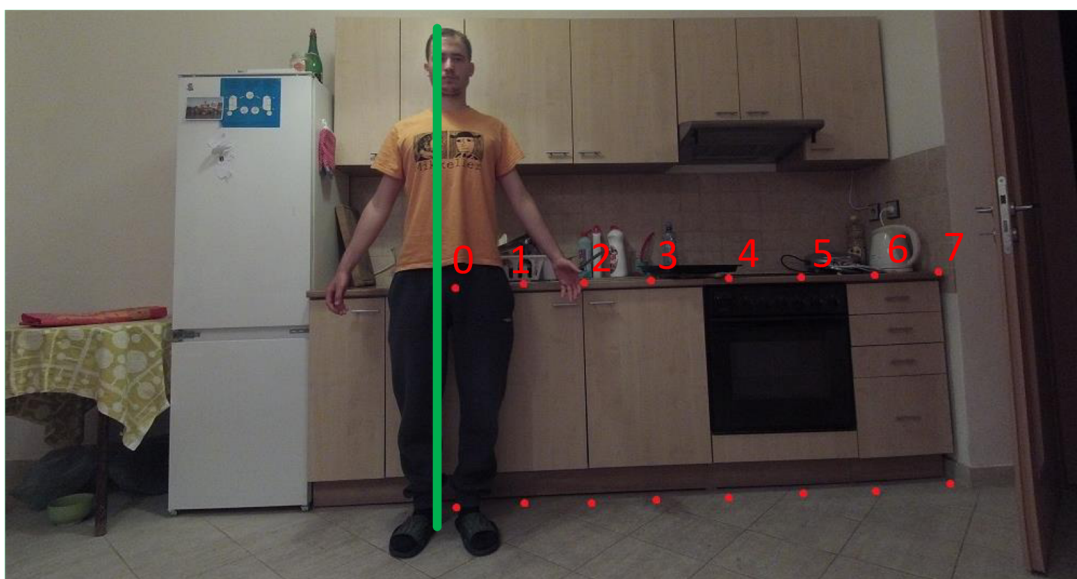
Tab. 7.2: Hodnoty namerané na priamke č.2 na obr 7.3

Bod	Meranie 1 [cm]	Meranie 2 [cm]	Meranie 3 [cm]	Priemer [cm]
1	183,857	194,803	195,135	191,265
2	186,976	184,382	186,212	185,857
3	186,974	186,364	186,563	186,634
4	187,554	188,902	189,595	188,684
5	190,943	190,821	192,1	191,288

Tab. 7.3: Hodnoty namerané na priamke č.1 na obr 7.3

7.1 Závislosť 1

Pre ten istý referenčný rozmer počítame chybu merania v rôznych vzdialenostiach od meraného rozmeru. Výška referenčného rozmeru činila 86 cm. Na obr. 7.1 je znázornená červenými bodmi (reálne ide o výšku kuchynskej linky v testovanom obraze).

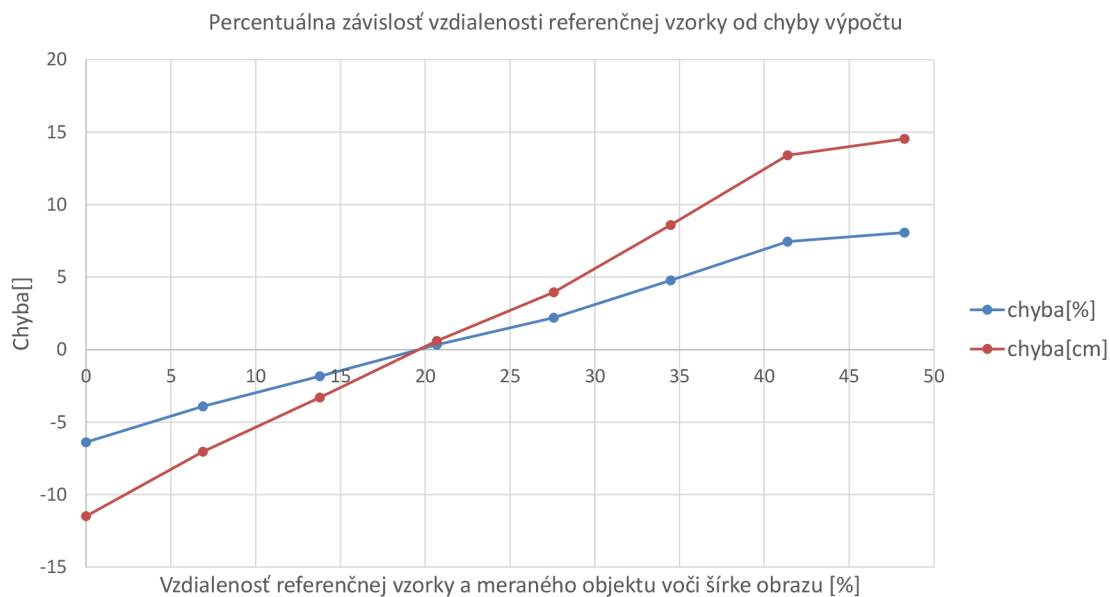


Obr. 7.1: Vzdialenosť referenčných výšok z tab. 7.4

Výsledky merania sme vyniesli do tab.7.4 a grafu 7.2.

Z grafu 7.2 je zrejmá hodnota závislosti pri prechode 0. Táto hodnota reprezentuje najvyššiu presnosť výpočtu pri definovaní referenčnej výšky vo vzdialenosti niekde medzi 18-20% šírky obrazu od meranej osoby.

Priemerná nameraná hodnota činí **182,41 cm** čo predstavuje chybu **2,41 cm** oproti reálnej výške osoby, ktorá je 180 cm.



Obr. 7.2: Závislosť chyby výpočtu od percentuálnej vzdialenosti referenčnej výšky a výšky osoby voči celkovej šírke obrazu

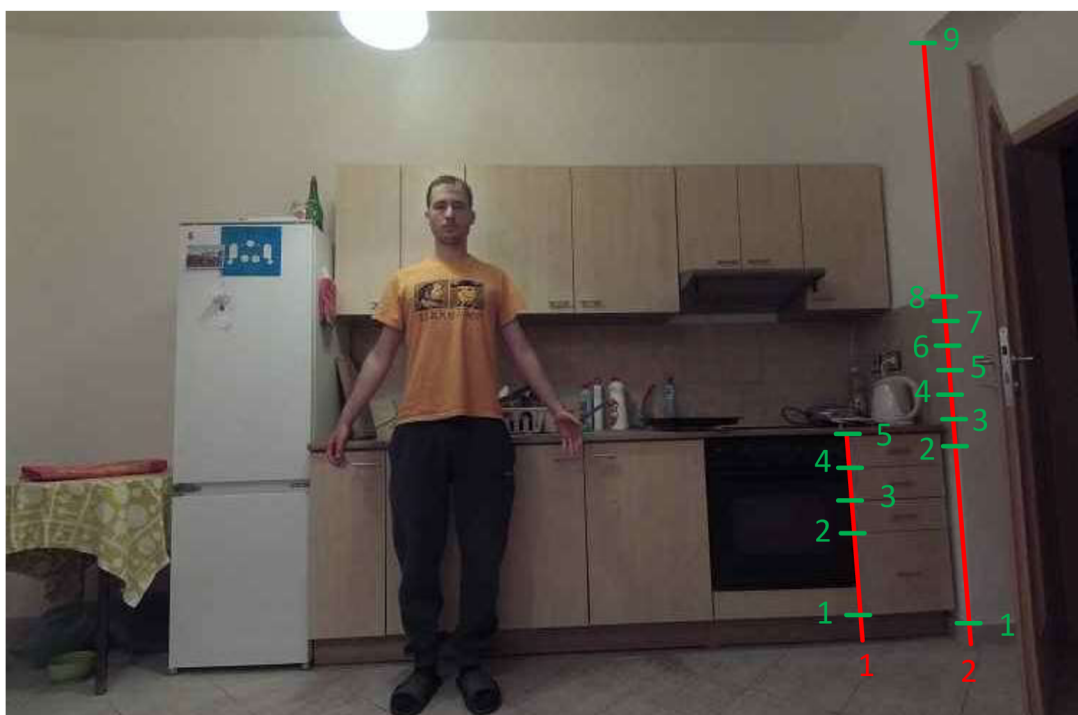
Bod jednotka	Meraná výška [cm]	Chyba		Vzdialenosť	
		[cm]	[%]	[px]	[%]
0	168,510	-11,489	-6,383	0	0
1	172,966	-7,034	-3,908	264	6,896
2	176,700	-3,299	-1,833	529	13,793
3	180,597	0,597	0,332	794	20,689
4	183,954	3,954	2,197	1059	27,586
5	188,602	8,602	4,779	1324	34,482
6	193,416	13,416	7,454	1588	41,379
7	194,538	14,538	8,077	1853	48,275
Priemer	182,410	2,410	1,339	-	-

Tab. 7.4: Hodnoty ku grafu 7.2

7.2 Závislosť 2

Pre dve vzdialenosti od meraného rozmeru určujeme chyby pri rôznych definovaných referenčných výškach. Zvolené referenčné výšky je možné vidieť na obr. 7.3. Tieto rozmery boli zvolené na základe dobrých záchytných bodov v obraze ktoré, sú od seba vo väčšine prípadov vzdialené v pravidelných intervaloch.

Pre priamku č.1 sú to hrany zásuviek, pre priamku č.2 kachličky obkladu so známym rozmerom. Vypočítané hodnoty pre priamku č.1 sú v tab. 7.5 a pre priamku č.2 v tab. 7.6.



Obr. 7.3: Zvolené priamky

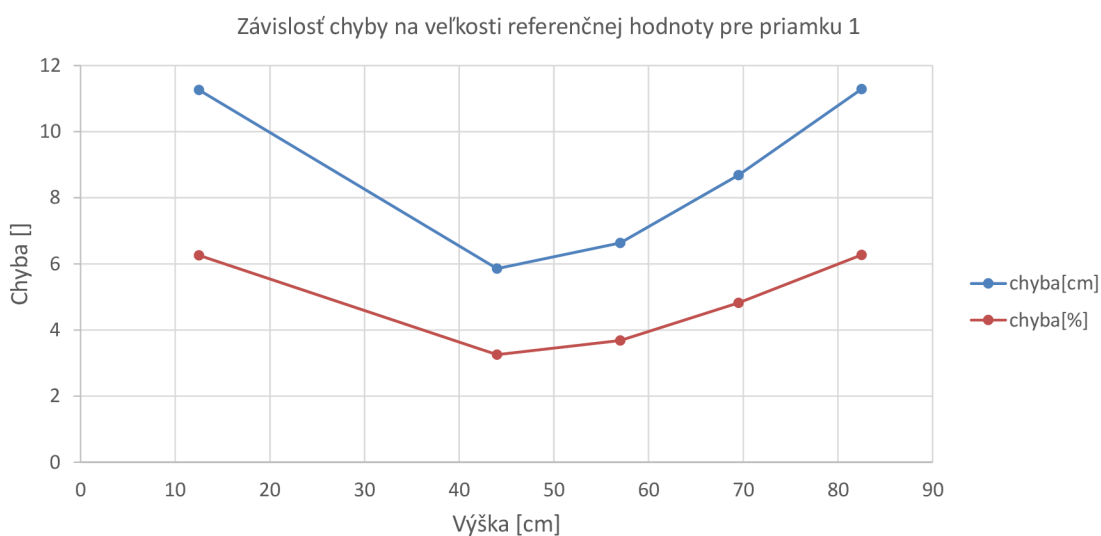
Bod	Meraná výška[cm]	Referenčná výška[cm]	Chyba [cm]	Chyba [%]
1,000	191,265	12,500	11,265	6,258
2,000	185,857	44,000	5,857	3,254
3,000	186,634	57,000	6,634	3,685
4,000	188,684	69,500	8,684	4,824
5,000	191,288	82,500	11,288	6,271
Priemer	188,745	-	8,745	4,859

Tab. 7.5: Skutočné výšky jednotlivých bodov pre priamku č.1

Bod	Meraná výška[cm]	Referenčná výška[cm]	Chyba [cm]	Chyba [%]
1	178,789	8	-1,211	-0,673
2	193,893	77,5	13,893	7,718
3	194,677	87,7	14,677	8,154
4	196,145	97,9	16,145	8,969
5	198,145	108,1	18,145	10,080
6	199,157	118,3	19,157	10,643
7	201,315	128,5	21,315	11,841
8	202,669	138,7	22,669	12,594
9	219,096	250	39,096	21,720
Priemer	198,209	-	18,209	10,116

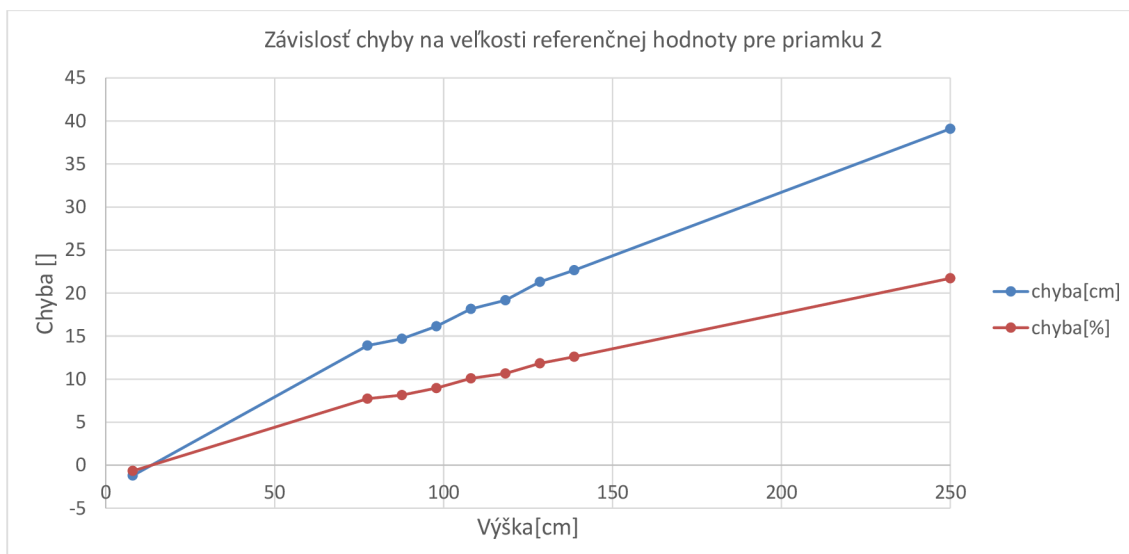
Tab. 7.6: Skutočné výšky jednotlivých bodov pre priamku č.2

Závislosti vypočítaných hodnôt sme vyniesli do grafov 7.4 pre priamku č.1 a 7.5 pre priamku č.2.



Obr. 7.4: Závislosť hodnôt z tab. 7.5 pre priamku č.1 z obrazu 7.3

Pre priamku č.1 na obr. 7.4 je zjavný pokles chyby v rozmedzí hodnôt 40-60 cm referenčnej výšky. Tento jav pravdepodobne pozorujeme aj v prípade priamky č.2 na obr.7.5 v rozmedzí hodnôt 75-100 cm, kde sa prejaví ako dočasné zmiernenie stúpania krivky. Posunutie môže byť spôsobené mierne väčšou vzdialenosťou od meranej výšky a tiež fakt, že obe priamky neležia v jednej rovine.



Obr. 7.5: Závislosť hodnôt z tab. 7.6 pre priamku č.2 z obrazu 7.3

7.3 Zhrnutie

Metódu sme podrobili 2 testom a zistili nasledujúce:

1. Najväčšiu presnosť má metóda pri zvolení referenčnej výšky vo vzdialenosti cca 18-20 % šírky obrazu od meraného objektu.
2. Chyba merania rapídne stúpa pri zvolení väčšieho referenčného rozmeru ako bol skutočný meraný rozmer.
3. Pri zvolení rovnakého referenčného rozmeru, sa pri zvyšovaní vzdialenosti od meraného rozmeru zvyšuje chyba približne lineárne. Pri vzdialenosti 50% šírky obrazu dosahuje chyba hodnotu 15 cm čo predstavovalo cca 8 %. Po prekročení 40% vzdialenosti sa nárast chyby mierne spomalí.
4. Pri volení rôzneho referenčného rozmeru v rovnakej vzdialenosti od meranej výšky, sa v oboch prípadoch prejavil mierny pokles chyby v určitom rozsahu. Pre priamku č.1 na obr. 7.4 je pokles chyby v rozmedzí hodnôt 40-60 cm . V prípade priamky č.2 na obr. 7.5 je v rozmedzí hodnôt 75-100 cm. Posunutie tohoto rozmedzia je spôsobené pravdepodobne mierne väčšou vzdialenosťou od meranej výšky a skutočnosť, že obe priamky neležia v jednej rovine.
5. Závislosť na obr. 7.4 má tvar hyperboly, avšak predpokladáme lineárne pokračovanie so stúpajúcou vzdialenosťou.
6. Závislosť na obr. 7.5 je lineárna čo odpovedá aj výsledkom z prechádzajúcich meraní.
7. Metóda je veľmi citlivá na drobné zmeny súradníc referenčného rozmeru hlavne vo väčších vzdialenostiach od meraného rozmeru.

Ako najväčší problém sa počas merania ukázalo problematické presné označenie jednotlivých bodov hlavne pri menších rozmeroch, kde drobné rozdiely v referenčnej výške predstavovali obrovské rozdiely vo vypočítaných hodnotách.

7.4 Podmienky pre správne fungovanie metódy

Základné predpoklady pre správne fungovanie metódy vyplývajúce zo spôsobu výpočtu:

1. Obraz musí byť kalibrovaný a bez skreslenia.
2. Meraná osoba musí stáť v podstavnej rovine miestnosti.
3. Referenčný objekt musí stáť v podstavnej rovine miestnosti.
4. Musí byť prítomný dostatočný počet hrán so správnou orientáciou, aby bolo možné vypočítať VP.
5. V obraze sa musí nachádzať aspoň jedna postava.
6. Označenie referenčného rozmeru musí byť čo najpresnejšie s reálnou zadanou výškou. Pri drobných zmenách, hlavne u menších rozmerov, je potom výsledná chyba takéhoto výpočtu obrovská.

8 ZÁVER

Cieľom práce bolo naštudovať metódy detekcie a merania výšky postavy v obraze. Následne bola vytvorená samostatná aplikácia schopná realizovať kalibračnú procedúru, detekciu parametrov podstavnej roviny, detekciu osoby a následne výpočet jej výšky. Aplikácia je implementovaná v jazyku C++ vo vývojovom prostredí Microsoft Visual Studio 2013. Na spracovanie obrazu bola zvolená a použitá knižnica OpenCV.

Prvá polovica práce je zameraná na teoretický rozbor problematiky. Vysvetľuje metódy a princípy zlepšujúce rôzne vlastnosti obrazu. Prvá kapitola popisuje základné operácie s histogramom, použiteľné hlavne pre zlepšenie čitateľnosti obrazu. Následne sú vysvetlené základné pojmy rovinnej, priestorovej a projektívnej geometrie. Tretia časť objasňuje pojem „model kamery“, druhy skreslení a možnosti ich korekcií. Vo štvrtej kapitole je vysvetlený princíp metódy HOG a piata teoreticky popisuje aplikovanú metódu výpočtu.

Druhá polovica práce popisuje praktickú implementáciu v jazyku C++ pomocou metód popísaných v teoretickom rozbere. V siedmej kapitole je popísaný algoritmus programu spolu s najdôležitejšími metódami a nastaveniami. Kapitola sedem následne hodnotí presnosť algoritmu na základe dvoch zvolených kritérií a zhŕňa podmienky, ktoré musia byť dodržané pre správne fungovanie algoritmu.

Podľa výsledkov získaných z hodnotenia pri prvom kritériu bola priemerná chyba výpočtu 2,41 cm, čo predstavuje 1,339%.

Podľa druhej závislosti dosiahla chyba merania až 18,209 cm, čo odpovedá 10,116 %. Výrazný rozdiel oproti prvému meraniu je pravdepodobne spôsobený pomerne veľkou vzdialenosťou referenčného rozmeru od meranej osoby a vysokou citlivosťou na presnosť označenia referenčného rozmeru. Pre presnejšie meranie chyby by bolo vhodné vložiť do obrazu vysoký objekt s pravidelnými zachytnými bodmi vo väčšom rozsahu. Tým by bolo možné presnosť hodnotenia zlepšiť.

Metóda sa ukázala ako pomerne presná pri dodržaní správnych podmienok a optimálnej veľkosti/polohe referenčného rozmeru. Pri nevhodnej veľkosti/polohe referenčného rozmeru však vykazovala značné chyby. Vzhľadom na fakt, že celý proces kriticky závisí na výsledkoch detektoru `Canny()`, pre zlepšenie by bolo vhodné vytvoriť algoritmus so spätnou väzbou, ktorý by automaticky nastavoval parametre detektoru podľa vlastností výstupného obrazu.

LITERATÚRA

- [1] Blazsovits, G.: DIP - Digital Image Processing. In *Interaktívna ucebnica spracovania obrazu, FMFI UK*, FMFI UK, Bratislava, 2006, str. 100.
URL <http://dip.sccg.sk/predspra/predspra.htm>
- [2] Chang, C.-C.; Lin, C.-J.: LIBSVM: A Library for Support Vector Machines. *ACM Trans. Intell. Syst. Technol.*, may 2011: s. 27:1–27:27, ISSN 2157-6904, doi:10.1145/1961189.1961199.
URL <http://doi.acm.org/10.1145/1961189.1961199>
- [3] Dalal, N.; Triggs, B.: Histograms of Oriented Gradients for Human Detection. In *International Conference on Computer Vision & Pattern Recognition*, editace C. Schmid; S. Soatto; C. Tomasi, INRIA Rhône-Alpes, ZIRST-655, av. de l'Europe, Montbonnot-38334, June 2005, s. 886–893.
URL <http://lear.inrialpes.fr/pubs/2005/DT05>
- [4] Drakos, N.: *Computer Based Learning Unit*. University of Edinburgh, 1993.
URL <http://homepages.inf.ed.ac.uk/rbf/CVonline/>
- [5] Gonzalez, R. C.; Woods, R. E.: *Digital Image Processing*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 2006, ISBN 013168728X.
- [6] Hartley, R. I.; Zisserman, A.: *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, 2004.
- [7] Hlavac, V.: Hledani hran.
URL <http://cmp.felk.cvut.cz/~hlavac/TeachPresCz/11DigZpr0br/22EdgeDetectionCz.pdf>
- [8] wei Hsu, C.; chung Chang, C.; jen Lin, C.: A practical guide to support vector classification. 2010.
URL <http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf>
- [9] Ing. Kamil Riha, P.: *Pokrocile techniky zpracovani obrazu*. VUT v Brne, 2012.
- [10] Mansurov, N.: What is Distortion? *Photography Life*, 8 2013.
URL <https://photographylife.com/what-is-distortion>
- [11] OpenCV Developers Team: *The OpenCV Reference Manual*. April 2015.
URL <http://docs.opencv.org/opencv2refman.pdf>

- [12] Satpathy, A.; Jiang, X.; Eng, H.-L.: Extended Histogram of Gradients with Asymmetric Principal Component and Discriminant Analyses for Human Detection. In *Computer and Robot Vision (CRV), 2011 Canadian Conference on*, May 2011, s. 64–71, doi:10.1109/CRV.2011.16.
- [13] Szeliski, R.: *Computer Vision: Algorithms and Applications*. New York, NY, USA: Springer-Verlag New York, Inc., 2010, ISBN 1848829345, 9781848829343.
- [14] Zizka, J.: SVM. In *Studine materialy predmetu Strojoveho uceni*, FI, 2005, str. 4. URL http://is.muni.cz/el/1433/podzim2006/PA034/09_SVM.pdf

ZOZNAM SYMBOLOV, VELIČÍN A SKRATIEK

VL vanishing line

VP vanishing point

DoF degrees of freedom - stupne voľnosti

HOG Histograms of Oriented Gradient

px pixel

SVM Support Vector Machine