

# 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

## URČENÍ POZICE MOBILNÍHO ZAŘÍZENÍ V PROSTORU

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. MICHAL KOMÁR

BRNO 2013



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

# URČENÍ POZICE MOBILNÍHO ZAŘÍZENÍ V PROSTORU

LOCALIZATION OF MOBILE DEVICE IN SPACE

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. MICHAL KOMÁR

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. MAREK ŠOLONY

BRNO 2013

## **Abstrakt**

Tato diplomová práce se zaměřuje na dostupné možnosti lokalizace na současných mobilních telefonech platformy Android. Zkoumá možnosti lokalizace mobilního zařízení nejen s využitím inerciálních senzorů telefonu, ale také možnosti lokalizace s využitím obrazu integrované kamery. V rámci práce jsou popsána provedená měření dostupných inerciálních senzorů, představen algoritmus vizuální lokalizace a navržen systém využívající obou těchto přístupů.

## **Abstract**

This thesis focuses on the current localization options of the Android mobile phone platform. It explores the possibilities of locating mobile devices not only with the use of inertial sensors, but also the possibility of localization using integrated video camera. The work describes the measurements done with available inertial sensors, introduces visual localization algorithm and design a system using these two approaches.

## **Klíčová slova**

pozice mobilního telefonu v prostoru, odhad pozice, Android, OpenCV, hybridní lokalizace v prostoru, sensorová fúze, kalibrace kamery, optický tok

## **Keywords**

mobile device spacial positioning, position estimation, Android, OpenCV, hybrid spacial localization, inertial sensor fusion, camera calibration, optical flow

## **Citace**

Komár Michal: Určení pozice mobilního zařízení v prostoru, diplomová práce, Brno, FIT VUT v Brně, 2013

# Určení pozice mobilního zařízení v prostoru

## Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením Ing. Marka Šolonyho. Konzultantem k této práci mi byl Ing. Pavel Žák.

Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....  
Michal Komár

20.5.2013

## Poděkování

Na tomto místě bych velmi rád poděkoval zejména Ing. Pavlu Žákovi, který mi byl při vytváření této práce konzultantem a poskytl mi mnoho cenných rad, spoustu svého času a zejména pak zpětnou vazbu, bez které by práce nedosahovala takové kvality. Rád bych zde také poděkoval Ing. Marku Šolonymu za vedení práce.

© Michal Komár, 2013

*Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů..*

# Obsah

Obsah .....	1
1 Úvod.....	2
2 Možnosti lokalizace mobilního zařízení .....	3
2.1 GPS.....	3
2.2 Wi-Fi.....	4
2.3 GSM lokalizace .....	6
2.4 Další bezdrátové technologie.....	7
2.5 Vestavěné senzory .....	8
2.6 Akustická lokalizace.....	10
2.7 Existující přístupy / aplikace .....	11
3 Lokalizace s využitím počítačového vidění .....	13
3.1 Sledování změn v obraze .....	13
3.2 Kalibrace kamery .....	19
3.3 Existující přístupy .....	24
4 Dostupné nástroje.....	26
4.1 OpenCV .....	26
4.2 Android OS.....	26
5 Návrh systému .....	28
5.1 Senzorová část .....	28
5.2 Vizualní část .....	33
5.3 Výsledný systém.....	38
6 Závěr .....	43
Literatura .....	44
Seznam příloh .....	47
Příloha A - Obsah DVD.....	48
Příloha B: Manuál k demonstrační aplikaci.....	49

# 1 Úvod

Jedním ze současných trendů ve světě chytrých mobilních telefonů je vývoj aplikací usnadňujících uživateli pohyb ve známém i neznámém prostoru a umožňujících interakci s tímto prostorem skrze obrazovku mobilního zařízení. Společným jmenovatelem těchto aplikací však většinou bývá práce v otevřeném prostoru, kde je velkou výhodou možnost využití GPS přijímače, který v daných podmínkách poskytuje relativně vysokou přesnost. Příkladem zde mohou být různé navigační systémy pro auta nebo osoby nebo různé hry s geo-lokačním základem. V těchto aplikacích však můžeme vyzorovat, že se jedná zpravidla o určování pozice s přesností na celé metry, bez ohledu na to, v jaké poloze se nachází telefon. Pokročilé navigace jsou pak schopny využít i jiných senzorů (například kompasu pro určení směru jízdy), ale opět zde není cílem dosažení velké přesnosti.

Dalším zajímavým odvětvím, kde se do jisté míry můžeme setkat s určením pozice telefonu, je oblast rozšířené reality. Jedná se o pohled do reálného prostředí, který je doplněn o prvky virtuální reality [1]. Zde je již kladem větší důraz na lokální přesnost v určování polohy zařízení. Vezměme si například nějaký program pro mobilní telefony s fotoaparátem, který nám pomocí rozšířené reality umožňuje prezentovat námi zvolený 3D model. Abychom takový model mohli vsadit do reálného prostředí a prohlížet si ho přes obrazovku mobilního telefonu tak, že kolem něj budeme procházet, potřebujeme již znát velmi přesně alespoň relativní změny v poloze zařízení. Pro tyto účely se používají zejména 2 přístupy. Jedním z nich je určování pozice na základě známých značek (angl. marker-based tracking), druhým pak určování pozice bez použití značek (angl. markerless tracking), kdy se využívá například významných bodů v obraze.

Tato práce si za hlavní cíl klade prozkoumat možnosti senzorů v běžných mobilních zařízeních, jejich přesnost a potenciál jejich využití v lokalizaci mobilního telefonu v prostoru. Dále zkoumá možnosti lokalizace v prostoru pomocí zpracování obrazu z kamery telefonu, využitím moderních algoritmů počítačového vidění. Stěžejní částí pak bude návrh a implementace mobilní aplikace využívající pro určení pozice v prostoru fúze počítačového vidění a výstupů sensorových dat. Jako cílová platforma pro demonstrační aplikaci byl zvolen operační systém Android.

Struktura této práce je následující: druhá kapitola je teoretickým základem práce a zaměřuje se na technologie, principy a možnosti využití všech prostředků dostupných na zvolené platformě. Poskytne tak potřebný základ pro návrh celého systému. Třetí kapitola se blíže zaměřuje na lokalizaci s využitím video kamery. Čtvrtá kapitola ve stručnosti přiblíží použité technologie, tj. knihovnu OpenCV pro počítačové vidění a mobilní platformu Android. Pátá kapitola se pak skládá ze tří klíčových částí. Experimentálně zkoumá možnosti dostupných senzorů, představuje jeden z možných přístupů vizuální lokalizace a popisuje navržený systém pracující jak se sensorovými údaji, tak s vizuálním vstupem. V poslední kapitole jsou pak stručně shrnuty dosažené výsledky a navrženo další možné pokračování.

## 2 Možnosti lokalizace mobilního zařízení

Tato kapitola bude zaměřena zejména na teorii spojenou se všemi dostupnými hardwarovými a softwarovými prostředky platformy Android, které mají potenciál použití při lokalizaci zařízení v prostoru. Z toho hlediska se podíváme nejen na vestavěné senzory (akcelerometr, elektronický kompas, gyroskop) a GPS, ale také na bezdrátové technologie (Wi-Fi, Bluetooth, mobilní GSM síť) a lokalizaci pomocí zvuku či zpracování obrazu z kamery. Závěrem kapitoly budou uvedena některá existující řešení.

### 2.1 GPS

Jednou z možností navigace, kterou nabízí moderní mobilní telefony je technologie GPS (zkratka z angl. Global Positioning System). Dle [2] se jedná o družicový navigační systém pracující na principu rádiového dálkoměru, pomocí něhož je možno určit pozici kdekoliv na zemi, s přesností na několik metrů. Systém byl vyvinut pro vojenské účely v USA a plně funkční je od roku 1994. V dnešní době je GPS uvolněno pro civilní použití a nachází velké uplatnění v geodézii, kartografii a samozřejmě v navigaci.

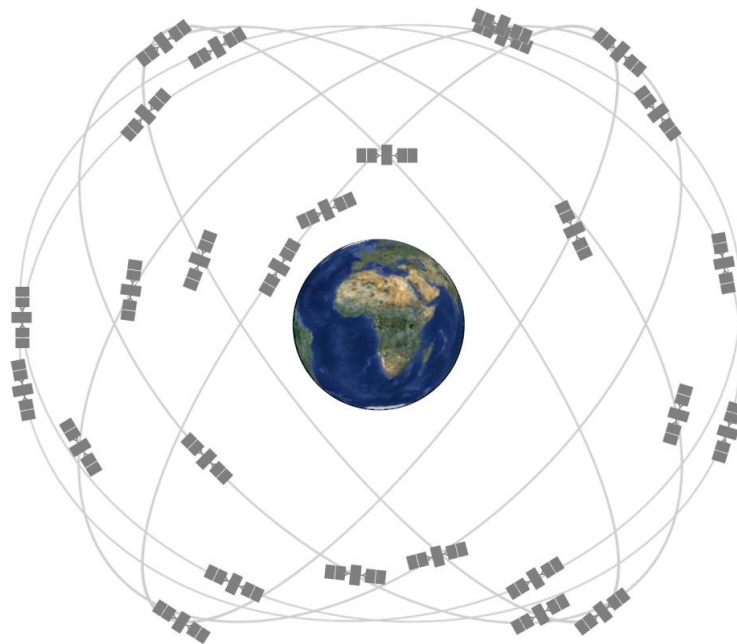
#### 2.1.1 Přehled

GPS systém se skládá ze tří částí: vesmírná část (satelity), kontrolní část (globální síť pozemních kontrolních stanic monitorujících satelity) a uživatelská část (přijímače) [3].

Vesmírná část systému (viz. 2.1) sestává z 24 satelitů obíhajících kolem Země v šesti drahách udržujících si sklon  $55^\circ$  relativně vůči rovníku. Oběžné dráhy jsou v takové poloze, aby bylo zaručeno, že z libovolného místa na planetě bude vždy viditelných minimálně 6 satelitů. Tyto satelity nepřetržitě vysílají zprávy rychlostí 50 bitů za sekundu. Hlavní informací obsaženou ve zprávě je čas, kdy byla zpráva odeslána, přesná orbitální poloha a informace o celkovém stavu systému.

Kontrolní systém, skládající se z mnoha stanic, se používá pro kontrolu stavu satelitů a nezbytnou korekci chyb, jako je například přesné nastavení času satelitu.

Uživatelský segment se pak skládá z přesných vojenských GPS "Precise Positioning Service" a civilních "Standard Position Service". GPS přijímač se obvykle skládá z antény (u mobilních zařízení je často zabudována v těle zařízení, případně integrována v GPS čipu), velmi stabilních interních hodin a softwaru pro výpočet uživatelské polohy a případně dalších veličin, jako je rychlost či orientace.



2.1: GPS rozložení satelitů, 24 slotová konfigurace definovaná ve standardu SPS, viz. [3]

### 2.1.2 Výpočet pozice

GPS přijímače používají geometrické triangulace k výpočtu přesné polohy na základě dat z několika různých satelitů. Jak bylo již zmíněno výše, zpráva satelitu se skládá z přesného času odeslání zprávy, přesné orbitální informace, údaje o stavu (zdraví) systému a hrubé informace o polohách ostatních satelitů.

Údaj o čase vyslání zprávy používá přijímač k zjištění doby přenosu, a tedy i k výpočtu vzdálenosti od satelitu. Z údajů o vzdálenosti a poloze jednotlivých satelitů je pak přijímač schopen lehce spočítat svou polohu s relativně vysokou přesností [4]. Nedostatkem však zůstává závislost na viditelnosti satelitů a proto není možné používat systém GPS v uzavřených prostorách, jako jsou budovy, tunely, atd.

Další detaily o systému GPS jsou k nalezení v [3]. K účelům této práce již však není potřeba zacházet do větších podrobností. Důležité je pouze uvědomit si, že systém GPS má obrovské možnosti použití při lokalizaci v otevřených venkovních prostorech, nicméně v budovách, hlubokých údolích či hustém porostu se může potýkat s odrazy signálu nebo jeho nedostatkem či úplnou ztrátou.

## 2.2 Wi-Fi

Wi-Fi (z angl. Wireless Fidelity) je ustálené označení pro standard IEEE 802.11. Zařízení s podporou Wi-Fi, jako například osobní počítače, herní konzole, chytré telefony, audio-video přehrávače a další, mají možnost připojení k internetu přes přístupový bod (AP - z angl. Access Point). Tato technologie využívá bezlicenčního frekvenčního pásma, a proto je ideální pro budování levné, ale výkonné bezdrátové sítě. Wi-Fi zajišťuje komunikaci na spojové vrstvě, zbytek je záležitostí vyšších protokolů (na rozdíl od Bluetooth, který sám o sobě zajišťuje nejrůznější služby). Typicky se proto přenášejí zapouzdřené ethernetové rámce. Pro bezdrátovou komunikaci na sdíleném médiu (šíření elektromagnetického pole prostorem) je používán protokol CSMA/CA [4] [5].



## 2.2.1 Struktura sítě

Bezdrátová síť může být vybudována různými způsoby v závislosti na požadované funkci. Ve všech případech hraje klíčovou roli identifikátor SSID (Service Set Identifier), což je řetězec až 32 ASCII znaků, kterými se jednotlivé sítě rozlišují. SSID identifikátor je v pravidelných intervalech vysílán jako broadcast, takže všichni potenciální klienti si mohou snadno zobrazit dostupné bezdrátové sítě, ke kterým je možné se připojit (tzv. asociovat se s přístupovým bodem).

### Ad-hoc síť

V ad-hoc síti se navzájem spojují dva klienti, kteří jsou v rovnocenné pozici (peer-to-peer). Vzájemná identifikace probíhá pomocí SSID. Obě strany musí být v přímém rádiovém dosahu, což je typické pro malou síť nebo příležitostné spojení, kdy jsou počítače ve vzdálenosti několika metrů.

### Infrastrukturní síť

Typická infrastrukturní bezdrátová síť obsahuje jeden nebo více přístupových bodů (AP – Access Point), které vysílají své SSID. Klient si podle názvů sítí vybere, ke které se připojí. Několik přístupových bodů může mít stejný SSID identifikátor a je plně záležitostí klienta, ke kterému se připojí. Může se například přepojovat v závislosti na síle signálu a umožňovat tak klientovi volný pohyb ve větší síti (tzv. roaming).

Informace k této podkapitole byly převzaty z [5].

## 2.2.2 Určení pozice pomocí Wi-Fi sítě

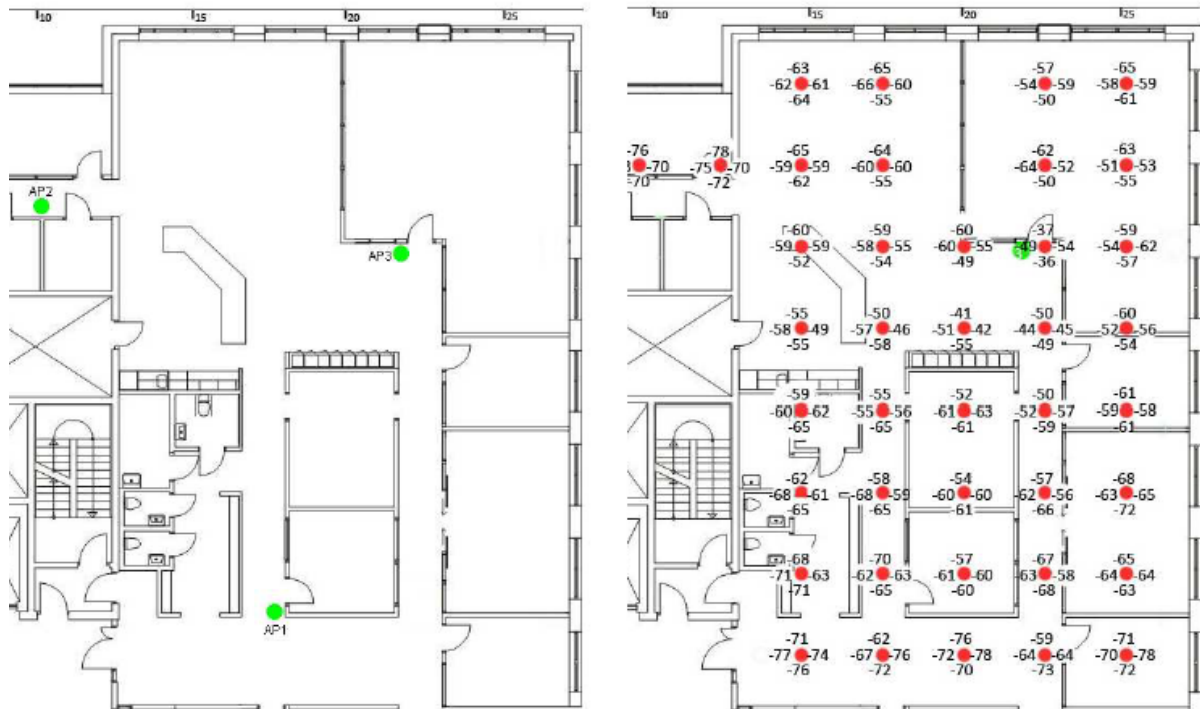
Zjišťování pozice s využitím bezdrátových sítí je v posledních pár letech oblíbeným výzkumným tématem. Hlavní dva faktory, které tuto skutečnost podpořily, jsou nepřesnost nebo nemožnost využití lokalizace za pomoci GPS v uzavřených prostorách a rostoucí využívání Wi-Fi technologie. Přesto, že technologie Wi-Fi nebyla pro lokalizaci předurčena, současné studie ukazují v tomto směru velmi dobré výsledky.

### Metody

Systémy přesného určení pozice pomocí Wi-Fi leží na základech jedné ze dvou metod. Jedna z nich pracuje na principu využití takzvaných fingerprintů<sup>1</sup> a zahrnuje porovnání přístupových bodů a jejich síly signálu s databází fingerprintů, kde hledá místa se stejnou konfigurací přístupových bodů a signálů. Alternativní metodou je pak využití trilaterální lokalizace podobně jako u GPS. Společným základem obou metod je potřebná databáze přesných pozic všech AP (přístupových bodů) a schopnost měřit na zařízení sílu signálu každého AP a to s konzistentní citlivostí. Každá odchylka v databázi nebo při měření se pak znatelně projevuje na výsledku lokalizace. Pro obě dvě metody se tak stává klíčovým problémem pořízení a udržování spolehlivé databáze. Tyto a další podrobnější informace jsou k nalezení v [6].

---

<sup>1</sup> Za fingerprint můžeme považovat nějaký záznam, který přesně popisuje danou situaci, tj. například v našem případě nese informaci o přesném místě a signálu všech dostupných přístupových bodů.



2.2: Vlevo: příklad rozmístění přístupových bodů po budově, vpravo: síla signálu v různých místech budovy měřená vůči AP3. Převzato z [4]

## 2.3 GSM lokalizace

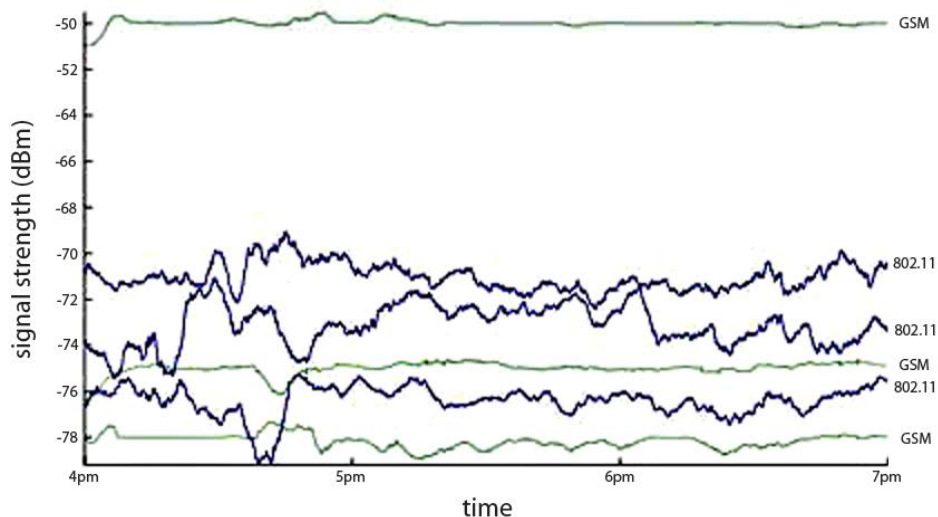
Lokalizace za využití GSM funguje na velmi podobném principu, jako lokalizace pomocí Wi-Fi. Opět zde existuje možnost využití fingerprintů nebo trilaterální lokalizace.

Octavian Manu zmiňuje v [7] některé výhody GSM lokalizace v budovách:

- Větší šířka pokrytí oproti Wi-Fi
- Možnost využití existující hardwarové infrastruktury
- Není závislá na lokálních výpadech elektrické sítě
- GSM, na rozdíl od ISM<sup>2</sup>, využívá licencovaného pásma a není tedy tolik náchylné k rušení blízkými bezdrátovými zařízeními vysílajícími na stejné frekvenci.
- Vysoká cena a komplexnost těchto sítí se projevuje v pomalé a ne příliš časté změně konfigurace.

GSM signál je také zajímavý z hlediska stability v čase. Na obrázku 2.3 je k vidění výstřižek z experimentálního, tří-hodinového měření blízkých Wi-Fi přístupových bodů a signálů šesti nejsilnějších GSM buněk.

<sup>2</sup> Z anglického Industrial, Scientific and Medical Band. Jedná se o označení pro bezdrátové technologie pracující na frekvenci 2.4 GHz (například Wi-Fi, Bluetooth, ZigBee, atd.).



2.3: Porovnání stability signálu GSM a Wi-Fi (802.11) během tří-hodinového měření. Převzato z [7]

Výsledky studií ukazují, že za pomoci GSM lokalizace jsme schopni rozlišit mezi patry budovy a v rámci jednoho patra dosáhnout přesnosti až na 2,5 m.

Hlavní nevýhodou GSM lokalizačních systémů je malá hustota rozmístění BTS<sup>3</sup> (z angl. Base Transceiver Station) stanic v mimoměstských lokalitách, která v těchto místech činí GSM lokalizaci velmi obtížnou.

## 2.4 Další bezdrátové technologie

Kapitola krátce popíše další možnosti lokalizace za využití bezdrátových technologií, které k tomuto účelu nebyly přímo koncipovány. Jejich společnou vlastností je také to, že jsou pro lokalizaci mobilních zařízení používány zřídka.

### 2.4.1 Bluetooth

Bezdrátová technologie Bluetooth je globální bezdrátový standard krátkého dosahu umožňující připojení široké škály elektronických zařízení. Tato technologie se neustále vyvíjí v návaznosti na své hlavní přednosti, kterými jsou nízké energetické náklady, nízká cena, vestavěné bezpečnostní prvky, robustnost, jednoduchost použití a možnost využití jako ad-hoc síť [8]. V současných mobilních zařízeních nachází uplatnění zejména starší Bluetooth třídy 1 podporující maximální rychlost přenosu 1 megabit za sekundu (1 Mbps) a, v současnosti nejběžněji používaný pro mobilní zařízení, Bluetooth třídy 2 s rychlostí až 3 megabity za sekundu (3 Mbps) s maximálním dosahem okolo 10m [4].

Princip využití technologie Bluetooth k lokalizaci mobilního zařízení je podobný, jako je tomu u technologie Wi-Fi a to skenováním dostupných Bluetooth zařízení. Mobilní telefony jsou běžně schopné zjistit seznam dostupných zařízení obsahující následující informace:

- NAME - jméno zařízení
- MAC ADDRESS - MAC adresa zařízení
- TYPE - typ zařízení (osobní počítač, mobilní telefon, atd.)

<sup>3</sup> přijímací a vysílací stanice GSM sítě.

Rozdíl oproti Wi-Fi je ten, že nemáme k dispozici sílu signálu a nemůžeme tedy odhadovat svou přesnou pozici mezi dostupnými zařízeními. Přesto však, vzhledem k tomu, že dosah technologie Bluetooth je limitován svým maximem deseti metrů, můžeme zjistit alespoň lokalitu, ve které se nacházíme, za předpokladu, že máme informaci o přesné poloze dostupných zřízení vůči prostoru, kde se chceme orientovat.

## 2.4.2 NFC

NFC, neboli Near Field Communication, je sada standardů, která definuje architekturu této modulární technologie. Slouží k bezdrátové komunikaci mezi elektronickými zařízeními na krátkou vzdálenost, obvykle přiblížením zařízení do vzdálenosti jednotek centimetrů. Současné a předpokládané využití této technologie je především v bezkontaktních transakcích, výměně dat či třeba ve zjednodušené konfiguraci zařízení (Wi-Fi). S využitím této technologie se počítá především ve vzájemné komunikaci aktivních zařízení, nebo aktivních zařízení s pasivními zařízeními. Pro příklad můžeme uvést vzájemnou komunikaci mobilních telefonů, bezkontaktní karty a její čtečky, či třeba komunikaci mobilního telefonu s tagem (pasivním, nenapájeným NFC zařízením) [9]. Přesto, že tato technologie zatím není běžným vybavením současných mobilních zařízení, proniká do tohoto segmentu velmi rychle a dá se očekávat, že se běžným vybavením chytrých telefonů stane.

Opět jsou možnosti, jak danou technologii využít pro lokalizaci mobilního zařízení v prostoru, nicméně tyto možnosti jsou ještě omezenější, než je tomu v případě Bluetooth. Vezměme si například modelovou situaci, kdy máme budovu (například muzeum, galerii, atd.) pokrytou NFC tagy a v mobilním zařízení máme k dispozici databázi jejich přesných pozic. Načtením libovolného tagu jsme schopni okamžitě a přesně zjistit svou pozici. Podobný přístup však vyžaduje výraznou participaci uživatele a poskytuje nám pouze diskrétní informaci o naší poloze.

## 2.4.3 Další možnosti

Kromě výše zmíněných metod existují ještě mnohé další metody lokalizace za využití bezdrátových technologií. Tyto metody však nejsou příliš zajímavé z pohledu možnosti využití v současných mobilních telefonech buď z důvodu ubývající podpory potřebných senzorů (lokalizace na základě infračerveného záření) nebo obecně neexistující podpory (lokalizace ultrazvukem, RFID lokalizace<sup>4</sup>). Některé z těchto metod jsou popsány v [7].

## 2.5 Vestavěné senzory

S postupným vývojem mobilních telefonů přibývaly také nové hardwarové komponenty. Například akcelerometry se staly běžnou součástí mobilního telefonu potom, co byly představeny jako možné rozšíření pro uživatelské rozhraní a fotoaparát. Jejich schopností je zjistit, v jaké poloze zrovna drží uživatel svůj telefon. Na základě této informace může telefon například automaticky měnit mezi vodorovným a svislým natočením displeje a stejně tak zobrazovat fotografie na displeji se správnou orientací. Běžným vybavením současných chytrých telefonů je gyroskop, akcelerometr, elektronický kompas, senzor vzdálenosti (proximity senzor), světelný senzor a případně další senzory (například softwarová fúze předešlých senzorů), které již nejsou tak obvyklé.

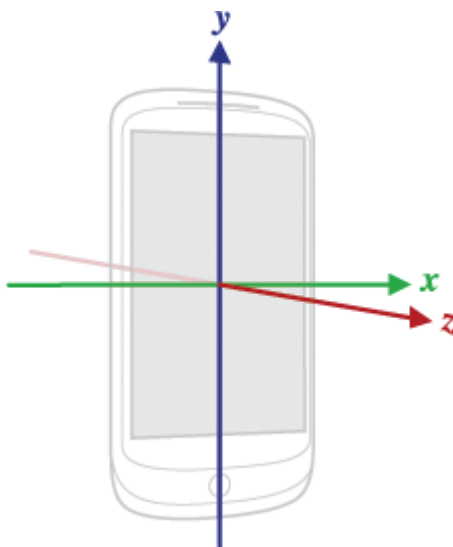
Senzor vzdálenosti a senzor světla umožňují telefonu získat dodatečné informace o kontextu použití telefonu. Senzor vzdálenosti detekuje například situaci, kdy uživatel drží telefon u tváře při

---

<sup>4</sup> Z angl. Radio Frequency IDentification - vysokofrekvenční identifikace.

hovorů a dává tak impuls pro vypnutí dotekové odezvy displeje a tlačítek. Světelný senzor se zase používá pro upravení svítivosti displeje tak, aby například v noci příliš nezářil.

Z hlediska lokalizace a odhadu pohybu je zajímavý zejména akcelerometr, gyroskop a elektronický kompas neboli magnetometr. Následuje stručný popis. [10] [11]



2.4: Souřadný systém (relativně k zařízení) používaný senzorovým aplikačním rozhraním systému Android.

## 2.5.1 Akcelerometr

3-osý akcelerometr nám udává informaci o naměřením zrychlení v jednotkách  $m/s^2$  samostatně pro každou z os X, Y a Z (obrázek 2.4). V případě, že telefon leží na stole displejem nahoru, udává nám akcelerometr ve směru osy Z hodnotu  $g = 9,81 m/s^2$ . To je v souladu se třetím Newtonovým zákonem, který říká, že stůl zrychluje telefon směrem nahoru. V případě volného pádu by akcelerometr ukazoval hodnotu  $0 m/s^2$  ve všech směrech. Akcelerometr může být použit k rozeznání pohybových aktivit. Nejvýznamnějším zdrojem chyb akcelerometru je zkreslení. Za zkreslení akcelerometru můžeme považovat rozdíl výstupního signálu oproti skutečné hodnotě. Zkreslení je možné odhadnout a vyrovnat dlouhodobým měřením výstupu akcelerometru ve stavech, kdy nepodléhá žádnému zrychlení.

## 2.5.2 Elektronický kompas

Elektronický kompas neboli magnetometr je senzor měřící magnetické pole v jednotkách mikro Tesla ve směru osy X, Y a Z. Výstupem senzoru orientace jsou pak tři úhly<sup>5</sup> Azimuth (rotace ve směru osy Z), Pitch (rotace ve směru osy X), Roll (rotace ve směru osy Y) počítané na základě údajů z akcelerometru a magnetometru. To má za následek často vznikající komplikace s přesným odhadem orientace telefonu. Magnetometr je také snadno vychýlitelný magnetickým rušením

---

<sup>5</sup> Vzhledem k nejednoznačnosti českých ekvivalentů uvedeny pouze anglické ekvivalenty.

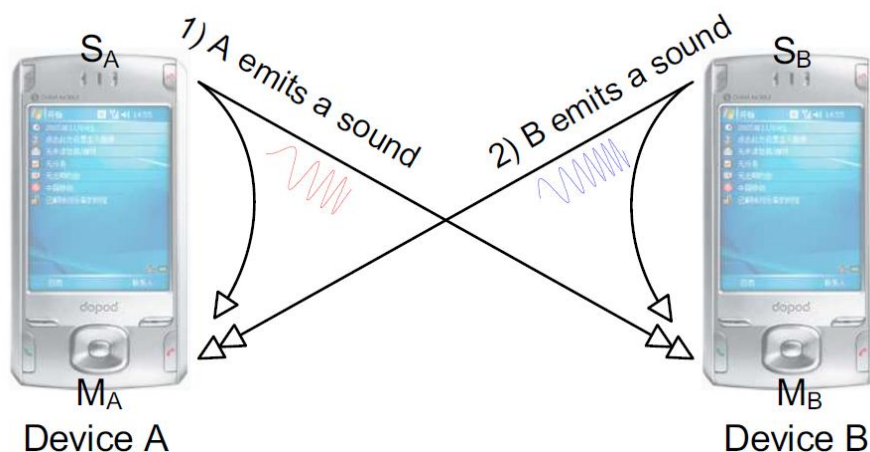
### 2.5.3 Gyroskop

Výstupem gyroskopu je úhlová rotace na všech třech osách měřená v jednotkách radián za sekundu. Důležité je uvést, že gyroskop měří pouze změny v úhlové orientaci. Problémem gyroskopu je častá odchylka a vznikající numerické chyby.

## 2.6 Akustická lokalizace

Jedním ze zajímavých způsobů orientace v prostoru je relativní lokalizace na akustické bázi. Za zmínku stojí zejména metoda BeepBeep popsaná v [12]. Dále pak systém prezentovaný v [13], který navazuje na předchozí metodu, jakožto akustický systém lokalizace s větší přesností pro více mobilních zařízení. Základní myšlenkou obou systémů je využití pouze základního hardwarového vybavené, jako jsou vestavěné reproduktory, mikrofony a Wi-Fi v mobilním zařízení.

Základní myšlenkou obou metod je odhad vzdálenosti na základě takzvaného ETOA (z angl. elapsed time between the two time-of-arrivals), neboli uplynulého času mezi dvěma časy příchodu (znázorněno na obrázku 2.5). Základní schéma tohoto odhadu se skládá ze tří kroků. Předpokladem je, že jsou obě zařízení v nahrávacím stavu. V prvním kroku vyšle zařízení A zvukový signál reproduktorem  $S_A$ . Tento signál zaznamená vlastním mikrofonom a stejně tak jej zaznamená zařízení B. Poté zařízení B vyšle další zvukový signál svým reproduktorem  $S_B$ . Tento signál je také zaznamenán mikrofony obou zařízení. V druhém kroku obě zařízení zkoumají nahraná data a lokalizují části vzorku, kde byly přijaty právě dva předchozí vyslané signály. Rozdíl času mezi těmito dvěma signály je právě výše zmíněný ETOA. Skrze dostupné médium (např. výše zmíněné Wi-Fi spojení) si oba telefony vymění své lokálně změřené ETOA a apriorně změřenou vzdálenost mezi svým reproduktorem a mikrofonom. V posledním kroku se pak na základě těchto informací spočte odhadovaná vzdálenost mezi těmito zařízeními. Zajímavé je, že díky měření ETOA na každém zařízení nezávisle, není potřeba žádné synchronizace času mezi zařízeními.



2.5: Znázornění dvousměrné komunikace při akustickém odhadu vzdálenosti dvou zařízení. Převzato z [12]

## 2.7 Existující přístupy / aplikace

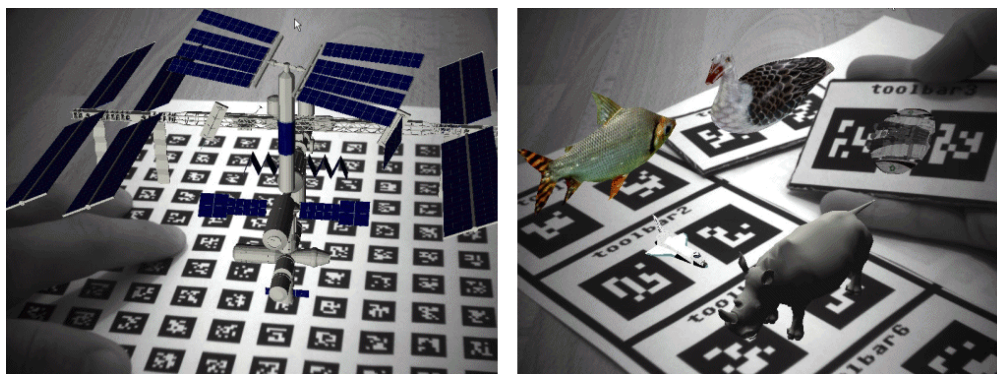
Jak vyplývá z popisu jednotlivých výše zmíněných možností, existuje mnoho způsobů lokalizace chytrého mobilního telefonu v prostoru. Každý z nich je však v různých situacích jinak vhodný. Dle toho se také odvíjí jejich využití v praxi. Pro ukázkou si uvedme některé fungující systémy a aplikace využívající jednu či více ze zmíněných metod:

### Navigace

Asi nejvíce používané aplikace s lokalizací jsou různé mobilní navigace pro automobily, cyklisty či turisty. Tyto navigace využívají k lokalizaci zejména systém GPS pro určení globální polohy zařízení. Pokročilejší aplikace pak dovedou využívat inertních senzorů, zejména pak elektronického kompasu, k upřesnění směru pohybu či natočení mapy do správné polohy na displeji. Bez využití elektronického kompasu se směr pohybu odhaduje na základě informace o současné a předchozí poloze. Těchto aplikací existuje na trhu obrovská spousta, pro ukázkou jmenujme například TomTom, Dynavix nebo Sygic pro mobilní telefony, Becker, Garmin, GoClever nebo Holux jako zástupce přenosných navigací či vestavěné autonavigace firem Blaupunkt, VDO nebo Alpine.

### Rozšířená realita

Jak již bylo zmíněno výše, rozšířená realita je na vzestupu a existuje zde opět celá řada aplikací, ať se jedná o systémy vznikající za akademickými účely nebo o systémy komerční. Tyto aplikace lze zpravidla dělit do dvou kategorií. Globální rozšířená realita s informacemi z GPS. Ta nám umožňuje pohyb po světě, do kterého přidává například různé popisky, fotografie či jiné předměty (například mobilní aplikace Layar, AR Compass a další [14]). Nebo lokálně zaměřené systémy využívající zejména informací získaných z obrazu, například na prohlížení 3D objektů před hledáčkem fotoaparátu či kameru (například systémy Magic Lens nebo Magic Mirror založené na knihovně ARTag [15]).

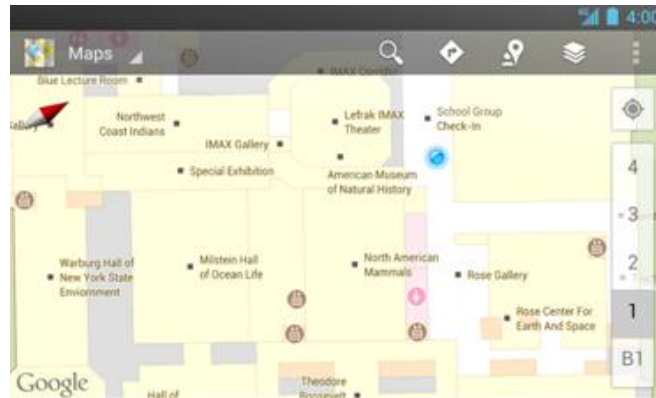


2.6: Ukázka systému Magic Lens knihovny ARTag

### Lokalizace v budovách

Nejvíce v plenkách, nicméně o to zajímavější, jsou systémy lokalizace v budovách nebo určování vzájemné pozice zařízení na krátkou vzdálenost. Vzhledem k omezenému signálu GPS je největším trendem v této oblasti využití technologie Wi-Fi a Bluetooth. Díky systémům, jako je například vyvíjený Google Maps, se myšlenka pohodlné navigace nákupními centry, výrobními halami, památkami nebo veletrhy stává také reálnou a v budoucnu se dá určitě počítat s jejím zdokonalováním. Co se týče vzájemné polohy dvou telefonů, stojí za zmínku například výzkumný

systém BeepBeep (kapitola 2.6) sloužící pro určení vzájemné pozice telefonů na základě zvukové informace.



2.7: Ukázka aplikace Google Maps pro orientaci v budovách. Převzato z [16]



## 3 Lokalizace s využitím počítačového vidění

Tato kapitola se detailněji zaměří na specifické odvětví lokalizace s využitím algoritmů počítačového vidění. Počítačové vidění, jakožto podoblast zpracování obrazu, je odvětvím, které v posledních dvou dekadách zaznamenává velký rozkvět nejen na akademické půdě, ale i v praxi. Uplatnění nachází zejména v oblasti automatického řízení robotů, bezpečnostních systémech, lékařství, ale i v mnoha dalších. Jedním z jeho odvětví je zpracování a sledování pohybu, ať již objektů snímaných kamerou, tak zjišťování polohy kamery vůči snímanému okolnímu světu. Následující podkapitoly se tedy zaměří na klíčové přístupy a algoritmy, které najdou využití v cílovém systému, ke kterému tato práce směřuje.

### 3.1 Sledování změn v obraze

Při práci s videem, na rozdíl od práce s jednotlivými snímky, je často potřeba sledovat dílčí objekty či globální změny pohybu obrazu napříč video-sekvencí. V anglické literatuře je tato oblast nazývána jako "tracking". Cílem není tedy objekt v obraze rozpoznat (identifikovat), nicméně zaměřit se na souvislosti mezi snímky ve video sekvenci a zjišťovat zajímavé informace z pohybu neidentifikovaných objektů. Techniky pro sledování neidentifikovaných objektů tedy typicky zahrnují hledání vizuálně zajímavých bodů v obraze (rohů), spíše než hledání konkrétních objektů. Tyto body jsou pak dále při zpracování párovány a je hledán takzvaný optický tok.

Druhou částí je potom modelování, které vychází z faktu, že předchozí techniky nám poskytují pouze obecnou, ne příliš přesnou představu o aktuální poloze objektu. Vytvořena byla spousta silných matematických nástrojů k odhadu trajektorie objektu na základě těchto získaných údajů. Tyto metody jsou aplikovatelné na dvou nebo tří dimenzionální modely objektů a jejich lokace. Informace k této kapitole byly čerpány převážně z [17], není-li uvedeno jinak.

#### 3.1.1 Hledání rohových bodů

Existuje velké množství lokálních příznaků, které je možno sledovat napříč videem. Přiblížme si tedy trochu, jaké takové příznaky mohou být zajímavé pro náš případ. Je zřejmé, že pokud ve snímku zvolíme bod nacházející se na velké bílé zdi, bude velmi obtížné identifikovat ten stejný bod na jiné pozici na dalším snímku videa. Tj. pokud všechny body na zdi budou stejné nebo i velmi podobné, pravděpodobně nebudeme příliš úspěšní s jejich sledováním napříč videem. Na druhou stranu, zvolíme-li bod, který bude ve snímku unikátní, šance na jeho objevení na následujícím snímku je velmi vysoká. V praxi je tedy důležité zaměřovat se na body v obraze unikátní nebo velmi se blížící tomuto stavu a dokázat je popsat takovým způsobem, abychom je byli schopni mezi sebou porovnávat. Názorná ukázka takových bodů je k vidění na obrázku 3.1.



3.1 Významné body v obraze. Body v označené kolečkem jsou zajímavé z pohledu sledování. Body ohraničené čtvercem jsou pro sledování nevhodné. Zdroj [17]

Vrátíme-li se tedy k příkladu s bílou zdí, mohli bychom usoudit, že jsou ideální body s ostrými změnami, např. ostré přechody barev. Tohle je dobrý začátek, nicméně takové body ještě nejsou pro sledování dostatečné. Je tomu tak proto, že tyto body mohou být například hranou nějakého objektu a všechny ostatní body na této hraně budou pravděpodobně vykazovat velmi podobné vizuální vlastnosti. Nicméně pokud je ale takový přechod v bodě viditelný ze dvou ortogonálních směrů, je pravděpodobnější, že tento bod bude unikátní. Z tohoto důvodu se také mnoho sledovatelných příznaků označuje jako *rohy* (angl. corners). Intuitivně tedy rohy, ne hrany, jsou takové body, které nesou dostatek informace na to, aby byly vybírány právě ony, snímek od snímku.

Nejčastěji používaná definice rohu byla uvedena C. Harrisem [18]. Harris rohy definuje na základě matice derivací druhého řádu ( $\delta^2x$ ,  $\delta^2y$ ,  $\delta x$ ,  $\delta y$ ) intenzity obrazu. Vytvoříme-li z takto derivovaných bodů nový obrázek, vznikne nám takzvaný Hessian obrázek. Tato terminologie vychází z takzvané Hessian matice okolí bodu, která je ve dvou dimenzích definována jako:

$$H(p) = \begin{pmatrix} \frac{\delta^2 I}{\delta x^2} & \frac{\delta^2 I}{\delta x \delta y} \\ \frac{\delta^2 I}{\delta x \delta y} & \frac{\delta^2 I}{\delta y^2} \end{pmatrix} \quad 3.1$$

Pro Harrisovy rohy pak uvažujeme auto-korelační matici druhých derivací obrazu nad malým okolím každého bodu. Tato matice je definována takto:

$$M(x, y) = \begin{bmatrix} \sum_{-K \leq i, j \leq K} w_{i,j} I_x^2(x+i, y+j) & \sum_{-K \leq i, j \leq K} w_{i,j} I_x(x+i, y+j) I_y(x+i, y+j) \\ \sum_{-K \leq i, j \leq K} w_{i,j} I_x(x+i, y+j) I_y(x+i, y+j) & \sum_{-K \leq i, j \leq K} w_{i,j} I_y^2(x+i, y+j) \end{bmatrix} \quad 3.2$$

Podle Harrisovy definice jsou rohy takové body v obraze, jejichž autokorelační matice druhých derivací obsahuje 2 velké vlastní hodnoty. V praxi to znamená, že v takovémto místě je textura (nebo hrana) jdoucí minimálně do dvou různých směrů se středem v tomto bodu. Výhodou této definice je také to, že definuje body invariantní vůči rotaci. To je důležité zejména proto, že sledovaný objekt (případně celý snímek) může během pohybu měnit své natočení. Za povšimnutí stojí také to, že tyto dvě vlastní hodnoty nejen určují, zda je bod vhodný pro sledování, ale také jej popisují.

### 3.1.2 Přesnější určení rohových bodů

Při zpracování obrazu jsme nuceni pracovat s jistým rozlišením snímku, které ale nemusí být dostatečné pro dosažení vysoké přesnosti sledování. Jsme zde omezeni pouze na celočíselný typ vyjadřující souřadnice bodu v obraze. Pro větší přesnost by se však často hodilo využít souřadnic v desetinných číslech, například bod (11,55;128,3).

Uvažujme například nějaký ostrý vrchol, který ve většině případů nebude mít špičku přímo uprostřed obrazového bodu. K získání jeho přesné polohy pak můžeme využít proložení obrazu nějakou křivkou (například parabolou) a s pomocí matematických operací zjistit přesnější polohu tohoto vrcholu. Toto je základní myšlenka většiny algoritmů pro zjištění přesnější polohy rohů.

V předchozím textu bylo naznačeno, jak je možné získat polohu rohů s přesností (celočíselnou) na body obrazu. Uveďme tedy také jak tuto polohu dostatečně zpřesnit. Využívá se matematického faktu, že skalární součin dvou na sebe kolmých vektorů je 0. Tato situace nastává právě u rohů. Podrobnější popis této problematiky a přehled několika možných přístupů je k nalezení například v [19].

### 3.1.3 Stálé příznaky

Stálé příznaky (z angl. invariant features) jsou příznaky, které jsou odolné vůči transformacím, jako je například rotace nebo změna velikosti obrazu. Od prvního představení Harrisových rohů bylo představeno spousta druhů rohů a lokálních příznaků. Pro úplnost zde uveďme seznam příznaků, se kterými se můžeme často setkat v praxi:

#### SIFT

Populární a často používané jsou například SIFT (z angl. "scale-invariant feature transform") příznaky. Tyto příznaky jsou, jak již název napovídá, stálé vůči změně velikosti obrazu. SIFT příznaky jsou detailněji představeny v [20].

#### SURF

(Z angl. Speeded Up Robust Features) je robustní detektor lokálních příznaků, poprvé představený Herbertem Bayem [21]. Tento detektor je částečně inspirován SIFTem a je založen na sumách odezev 2D Haarových vlnek. Uplatnění zde ve velké míře nachází teorie integrálního obrazu<sup>6</sup>.

#### Shi-Thomasi

Za zmínku stojí určitě také rohy podle Shi a Thomasi [22], které jsou vylepšením Harrisových rohů a jsou využívány například v knihovně OpenCV pro počítačové vidění (více o knihovně níže v textu) ve funkci *findGoodFeaturesToTrack()*, které se používá právě pro hledání bodů vhodných ke sledování napříč video sekvencí.

---

<sup>6</sup> Integrální obraz je způsob digitální reprezentace obrazu tak, že každý bod představuje součet hodnot předchozích pixelů doleva a nahoru. Tedy pravý spodní bod obsahuje součet všech pixelů obrázku [48].

## **FAST**

(Z angl. Features from Accelerated Segment Test) je rohový detektor původně vyvinutý Edwardem Rostenem a Tomem Drummondem [23]. Největší výhodou FAST detektoru je jeho výpočetní optimalizace, díky které dosahuje větší rychlosti v porovnání se spoustou ostatních detektorů. Porovnání s výše zmíněnými detektory Harris, Shi-Thomasi a dalšími je rozebráno v [24].

## **MSER**

(Z angl. Maximally Stable Extremal Regions) se používám jako metoda pro detekci blobů v obraze. Představena byla prvně J. Matasem a kolegy [25]. Metoda je výhodná zejména při použití pro stereovizi a rozpoznávání objektů v obraze.

## **ORB**

Jedná se výkonnou a velmi rychlou alternativu k SIFT a SURF detektorům. Jádrem je binární deskriptor vycházející z algoritmu BRIEF. ORB je odolný vůči rotaci a šumu. Více informací a porovnání s jinými detektory je k nalezení v článku [26].

## **BRISK**

(Z angl. Binary Robust Invariant Scalable Keypoints) je relativně novou metodou detekce významných bodů. Metoda je odolná vůči rotaci a změně velikosti a dosahuje dobrých výsledků z hlediska rychlosti a výpočetní náročnosti. BRISK byl představen v [27].

## **FREAK**

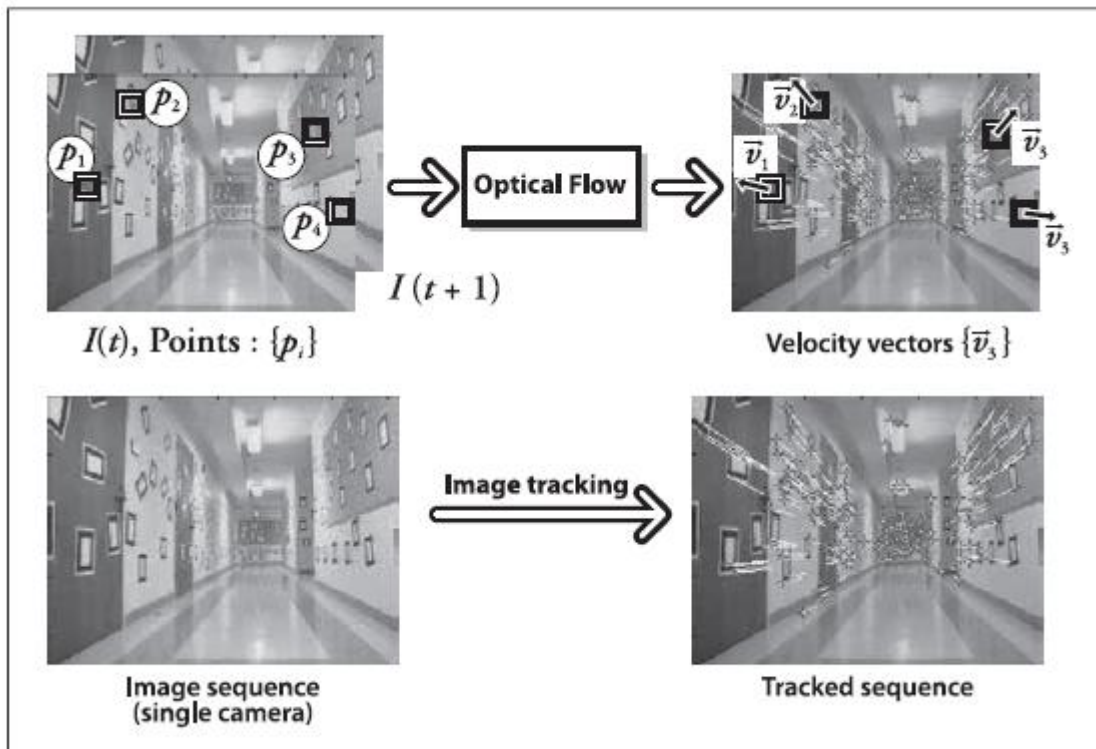
(Z angl. Fast REtinA Keypoint). Jedná se o jeden z novějších algoritmů, inspirovaný lidským vizuálním ústrojím, přesněji sítnicí. Experimentálně bylo ověřeno, že se jedná o obecně rychlejší, paměťově úspornější a robustnější detektor, než je SIFT, SURF nebo BRISK [28].

### **3.1.4 Optický tok**

Jak již bylo naznačeno dříve, častou úlohou počítačového vidění je sledování pohybu mezi dvěma (nebo více) snímky video sekvence bez znalostí o obsahu tohoto snímku. Typicky je tak pohyb samotný indikátorem toho, že se ve videu děje něco zajímavého. Optický tok je znázorněn na obrázku 3.2.

Každému pixelu snímku tedy můžeme nějakým způsobem přiřadit rychlost, případně změnu jeho pozice, která reprezentuje vzdálenost, o kterou se daný pixel pohnul mezi předchozím a současným snímek. Takovýto přístup se obvykle nazývá jako hustý (angl. dense) optický tok, který každému bodu v obraze přiřazuje rychlost.

V praxi není zjišťování takového optického toku jednoduchou záležitostí. Uvažujme například problém se snímáním bílého papíru. Spoustu pixelů, které byly na předchozím snímku bílé, zůstanou bílé i na snímku současném. Změní se pouze hrany papíru a to pouze ty, které nejsou rovnoběžné se směrem pohybu. Proto je potřeba u metod zjišťování hustého optického toku využít nějaké interpolace mezi body, které je jednodušší sledovat a mezi body pro sledování nevhodnými. Tento problém má za následek vysokou výpočetní náročnost těchto metod.



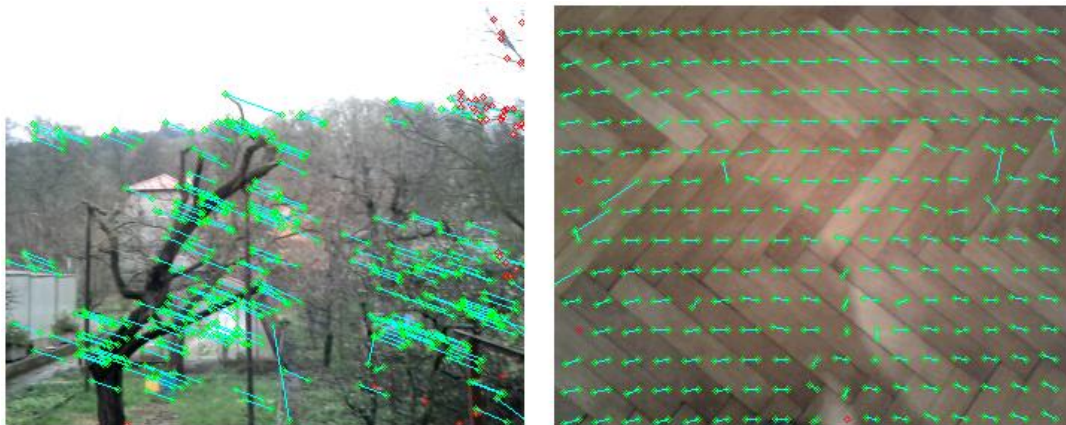
3.2 Optický tok: cílové příznaky (rohy) (vlevo nahoře) jsou sledovány v čase a jejich pohyb je převeden na vektory rychlosti těchto bodů (vpravo nahoře). Dolní snímky ukazují přímý výstup obrázku chodby z kamery (vlevo dole) a vektory toku během pohybu kamery chodbou (vpravo dole). Převzato z [17].

Tento problém nás tedy přivádí k myšlence takzvaného řídkého (angl. sparse) optického toku. Algoritmy pro jeho výpočet již počítají pouze s vybranou skupinou zájmových bodů, pro které je optický tok počítán. Pokud mají tyto body vhodné vlastnosti, jako rohové body zmiňované výše, pak bude jejich sledování relativně robustní a spolehlivé. Ve většině případů je také výpočetně méně náročné než u hustého optického toku.

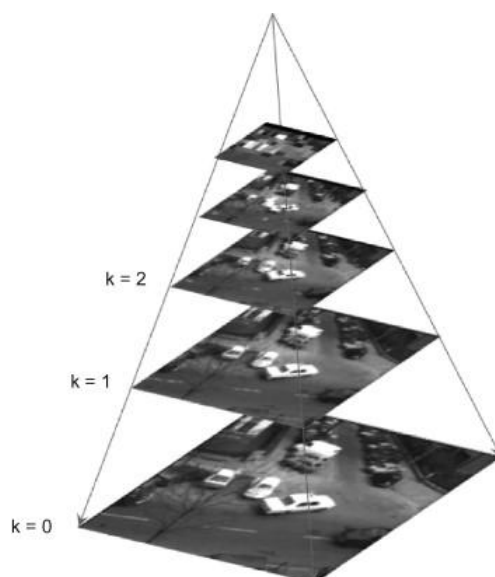
### Metoda Lucas-Kanade

Pro sledování optického toku existuje několik různých metod. Některé z nich jsou popsány například v [17]. Pro představu si v této práci alespoň okrajově představíme algoritmus Lucas-Kanade (LK), který je nejpopulárnější metodou zjišťování řídkého optického toku.

Algoritmus Lucas-Kanade [29], původně představen již v roce 1981, byl pokusem o získání hustého optického toku. Přesto ale, z důvodu, že je snadno aplikovatelný i na podmnožinu obrazových bodů, se stal důležitou metodou řídkého optického toku. Algoritmus může být aplikován na řídkou množinu bodů proto, že pracuje s okny vymezujícími malé okolí každého bodu zájmu (viz. obrázek 3.3). Nevýhodou využití malých oken je, že větší pohyby mohou znamenat, že některý zajímavý bod vypadne z lokálního okna a kvůli tomu je pak pro algoritmus nemožné tento bod najít. Tento problém vedl k návrhu pyramidového algoritmu LK, který začíná se sledováním na nejvyšší úrovni pyramidového modelu obrazu a postupně pokračuje k nižším úrovním (viz. obrázek 3.4). To umožňuje lokálním oknům zaznamenávat i velké pohyby.



3.3 Ukázka výpočtu optického toku algoritmem Lucas-Kanade. Na snímku vlevo je zobrazen optický tok počítaný na základě bodů získaných metodou Shi-Thomasi. Na snímku vpravo jsou použité body pravidelně rozprostřené po celém obraze.



3.4 Pyramidový model snímku používaný v pyramidovém algoritmu Lucas-Kanade. Model obsahuje snímek v několika úrovních detailu, od nejhrubšího po nejjemnější. Převzato z [30].

Základní myšlenka algoritmu Lucas-Kanade spočívá na třech předpokladech:

- **Světelná stálost** - bod obrazu náležící určitému objektu ve scéně nemění svůj vzhled během pohybu ze snímku do snímku. Pro šedotónové snímky to znamená, že jas daného pixelu se nemění během jeho sledování.
- **Časová stálost** - pohyb obrazu je pomalý. V praxi to znamená, že časové přírůstky (snímkování) jsou dostatečně nahuštěné vzhledem k rychlosti pohybu v obraze, takže se sledovaný objekt příliš nehýbe od snímku ke snímku.
- **Prostorová soudružnost** - sousedící body ve scéně náležící stejnému povrchu vykazují podobný pohyb a jsou promítány opět jako blízké od snímku ke snímku.

Za dodržení těchto tří předpokladů potom algoritmus dosahuje dobrých výsledků při sledování optického toku. Podrobnější popis algoritmu, včetně matematického pozadí je uveden například v [17].

### Metoda Horn-Schunck

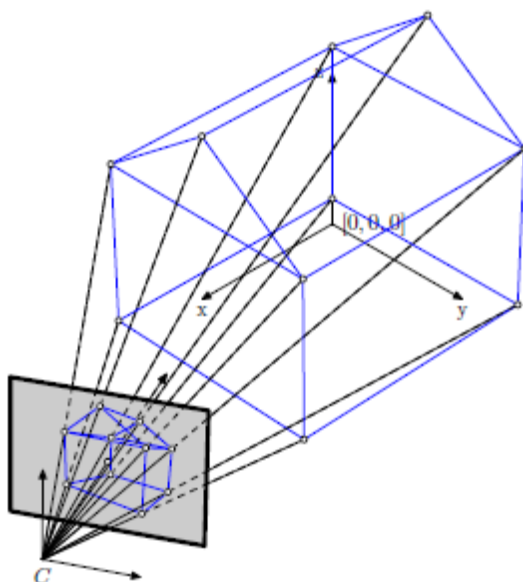
Metody výpočtu hustého optického toku jsou obecně pomalejší, než řídké metody, nicméně pro úplnost je potřeba zmínit algoritmus Horn-Schunck z roku 1981 [31]. Tato technika je zajímavá zejména proto, že jako jedna z prvních úspěšně využívala předpokladu světelné stálosti a derivací jejich základních rovnic.

## 3.2 Kalibrace kamery

Z předchozího textu je zřejmé, že možnosti využití obrazu z kamery mobilního telefonu pro další zpracování, například právě z pohledu lokalizace mobilního zařízení, jsou velké. V závislosti na využitých přístupech a algoritmech jsme schopni dosáhnout zajímavých a velmi přesných výsledků. Předpokladem je však v tomto případě dobře kalibrovaná kamera se známými parametry. Kalibrace nám v tomto případě umožní korigovat nepřesnosti snímače a přesně měřit pozici zařízení. Bude jí věnována následující kapitola, která přiblíží proces kalibrace a nedostatky, které jde správnou kalibrací potlačit.

### 3.2.1 Kalibrace

Chceme-li využívat libovolnou kameru k nějakému přesnému měření, je třeba ji nejdříve kalibrovat. Správně kalibrovaná kamera nám umožňuje pracovat s přesnou pozicí známého objektu vůči pozorovateli (kameře). Kamera provádí mapování mezi 3D světovým metrickým systémem a 2D souřadným systémem obrazu v pixelech (viz. obrázek 3.5). K provedení inverzní operace, tj. získání 3D souřadnic objektu v metrickém systému reálného světa z 2D souřadnic bodů snímku, je nutné znát interní a externí orientace kamery (dále jako interní/externí parametry). Tyto parametry získáme na základě kalibračního procesu.



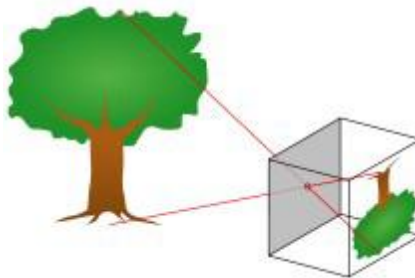
3.5 Projekce 3D objektu reálného světa na 2D projekční plochu kamery [32].

### Interní parametry

Tyto parametry určují vlastnosti optické soustavy kamery, které ovlivňují výsledný pořízený snímek. Parametry slouží k rekonstrukci paprsků středového promítání, transformaci z metrických souřadnic do souřadnic v pixelech a odstranění geometrického zkreslení obrazu. Mezi interní parametry patří ohnisková vzdálenost, poloha hlavního snímkového bodu a koeficienty radiálního a tangenciálního zkreslení kamery.

### Středové promítání

Následující popis vychází z jednoduchého modelu dírkové kamery (angl. pinhole camera) - viz. obrázek 3.6.



3.6 Princip dírkové kamery [32]

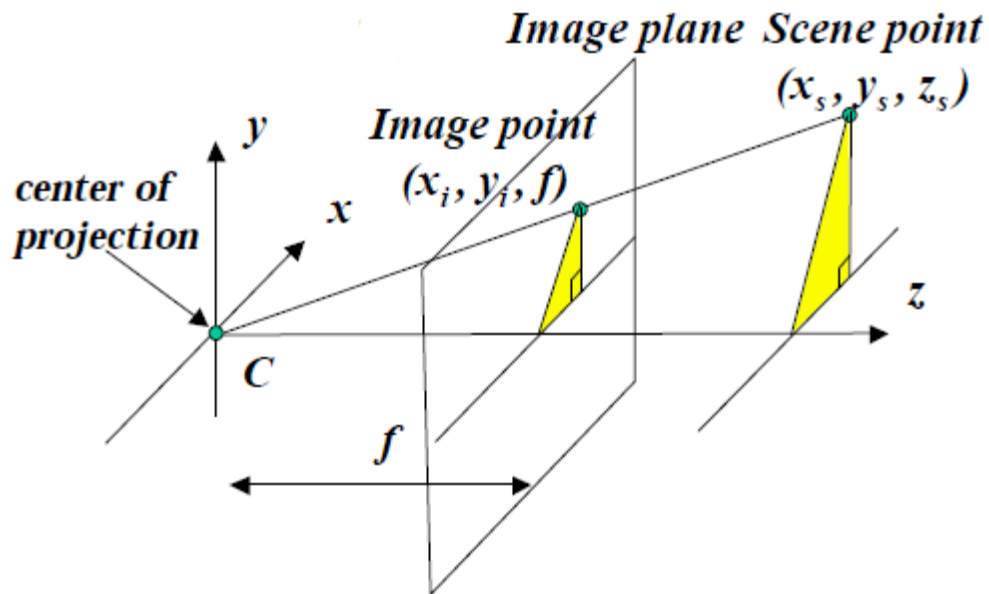
Střed promítání se nachází ve středu kamery. Tento bod se označuje písmenem  $C$  a je výchozím bodem Euklidovského souřadného systému. Pokud jsou obrazové body reprezentovány pomocí homogenních vektorů, středové promítání je lineární transformací bodů z prostoru na obrazovou rovinu. Souřadnice obrazového bodu  $[x_i, y_i, f]$  lze tedy vypočítat ze souřadnic světového bodu  $[x_s, y_s, z_s]$  podle vzorců 3.3 a 3.4. Maticový zápis transformace uvádí vzorec 3.5. Písmeno  $f$  zde vyjadřuje ohniskovou vzdálenost, to je vzdálenost mezi středem promítání a obrazovou/ohniskovou rovinou. U digitálních kamer je to vzdálenost mezi ohniskem čočky objektivu a snímacím čipem. Celá problematika je pak znázorněna na obrázku 3.7.

$$x_i = f * \frac{x_s}{z_s} \quad 3.3$$

$$y_i = f * \frac{y_s}{z_s} \quad 3.4$$

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_s \\ y_s \\ z_s \end{bmatrix}, \text{ kde } x_i = \frac{u}{w} \text{ a } y_i = \frac{v}{w} \quad 3.5$$





3.7 Lineární transformace bodů do obrazové roviny podle paprsků, které procházejí středem promítání a obrazovou rovinou [33].

### Transformace z metrických do pixelových souřadnic

Pro získání absolutních hodnot je třeba uvažovat převod souřadnic obrazových bodů ze systému metrického do systému pixelového. Pro tento převod se využívá následujících interních parametrů:

- **souřadnice hlavního snímkového bodu kamery**
  - $x_0$  - souřadnice na x-ové ose
  - $y_0$  - souřadnice na y-ové ose
- **míra zvětšení/zmenšení** - jedná se o hodnoty vyjadřující počet pixelů na jednotku ohniskové vzdálenosti v obrazových souřadnicích. Používá se v souvislosti s digitálními kamerami, kde se udávají rozměry obrazu v pixelech s různým měřítkem pro osu x a y, protože pixely snímače nemusí mít vždy přesně čtvercový (rovnoustranný) tvar.
  - $m_x$  - množství pixelů odpovídající délkové jednotce na x-ové ose
  - $m_y$  - množství pixelů odpovídající délkové jednotce na y-ové ose
- **úhel mezi osou x a y snímacího senzoru** - obvykle se vyjadřuje koeficientem  $s$ , pro většinu kamer je ale bez odchylky

Převod souřadnic  $[x_i, y_i, f]$  do pixelových souřadnic  $[x_{pix}, y_{pix}]$  je vyjádřen následujícími vztahy (rovnice 3.6, 3.7):

$$x_{pix} = m_x * x_i + x_0 \quad , \text{kde } x_i = f * \frac{x_s}{z_s} \quad 3.6$$

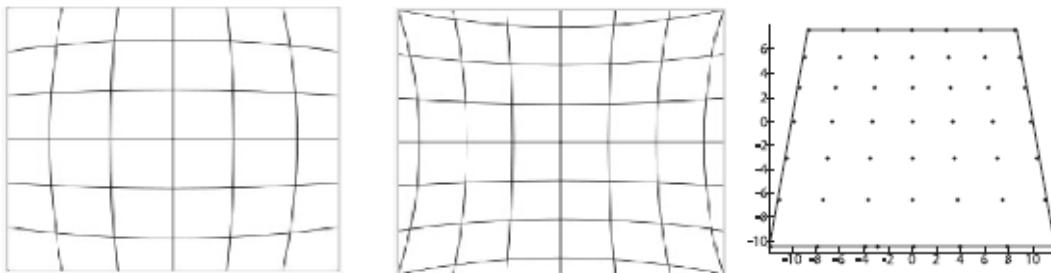
$$y_{pix} = m_y * y_i + y_0 \quad , \text{kde } y_i = f * \frac{y_s}{z_s} \quad 3.7$$

Zobrazení bodu  $X$ , s 3D metrickými souřadnicemi vztaženými k bodu  $C$ , umístěného ve scéně, do obrazové roviny se souřadnicemi v pixelech lze tedy zapsat maticově vztahem 3.8. V tomto vztahu pak  $K$  vyjadřuje matici interních parametrů kamery.

$$u \cong KX \Rightarrow \begin{bmatrix} u^i \\ v^i \\ w^i \end{bmatrix} = \begin{bmatrix} a_x & s & x_0 & 0 \\ 0 & a_y & y_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_s \\ y_s \\ z_s \\ 1 \end{bmatrix}, \text{ kde } a_x = f * m_x \text{ a } a_y = f * m_y \quad 3.8$$

### Zkreslení obrazu

Při průchodu světla ideální čočkou nedochází k žádnému zkreslení obrazu. Bohužel při použití reálných čoček je většinou obraz více či méně zkreslen. To se projevuje tím, že se přímky ve scéně zobrazují jako křivky. Největší podíl na zkreslení má tzv. radiální zkreslení, v menší míře se na něm podílí tzv. tangenciální zkreslení (viz. obrázek 3.8). Tangenciální zkreslení je způsobeno nedokonalým vycentrováním plochy čočky. Jeho koeficienty jsou označovány jako  $p_1$  a  $p_2$ . Radiální zkreslení způsobuje radiální posunutí bodů v obrazové rovině. Velikost posunutí stoupá se vzdáleností od středu snímku – hlavního snímkového bodu. Příklad tzv. polštářkového a soudkovitého radiálního zkreslení je na obrázku 3.8. Koeficienty tohoto zkreslení jsou  $k_1, k_2, k_3, \dots$ . Výsledkem kalibrace bývají pouze koeficienty s indexem 1 až 3. Zkreslení obrazu je možné softwarově odstranit pokud známe uvedené koeficienty.



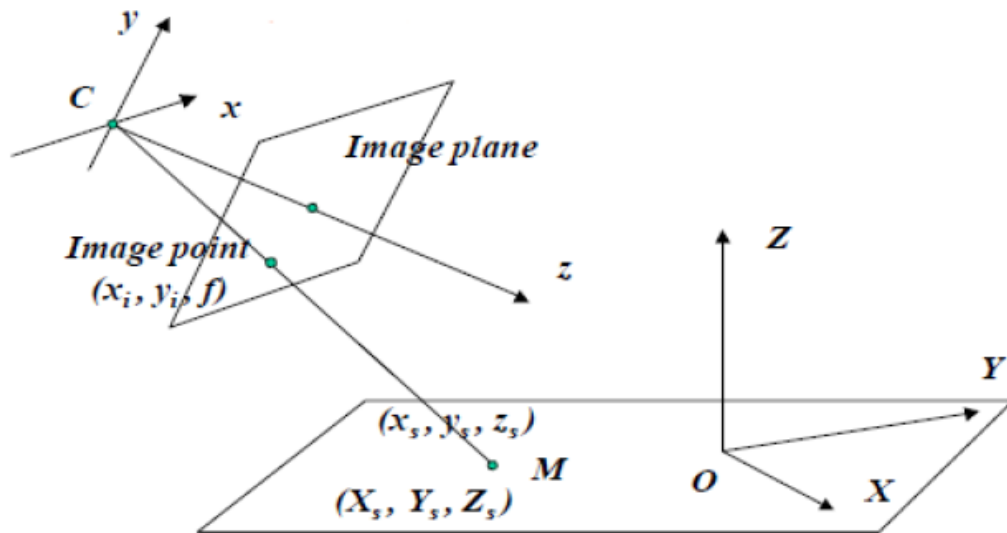
3.8 Ukázky zkreslení obrazu kamery. Zprava radiální zkreslení (polštářkovité), radiální zkreslení (soudkovité), tangenciální zkreslení [34], [17]

### Externí parametry

Tyto parametry slouží k určení polohy a natočení kamery v prostoru vzhledem k nějakému bodu. Parametry konkrétně vyjadřují Euklidovskou transformaci mezi souřadným systémem kamery a světovým souřadným systémem, oproti kterému je kamera kalibrována. Na obrázku 3.9 mají tyto systémy výchozí body  $C$  a  $O$ . Transformace se skládá z rotace  $R$  a posunutí (translace)  $t$  a je vyjádřena pomocí matice v rovnici 3.9. Vztah vyjadřuje transformaci bodu  $M$  ze světového do kamerového souřadného systému. Kvůli maticovému násobení jsou body vyjádřeny v homogenních souřadnicích. Použitá rotační matice  $R$  a translační vektor  $t$  získáme jako výstup kalibračního procesu.

$$X \cong [R_t]M \Rightarrow \begin{bmatrix} x_s \\ y_s \\ z_s \\ 1 \end{bmatrix} = \begin{bmatrix} r_{00} & r_{01} & r_{02} & t_0 \\ r_{10} & r_{11} & r_{12} & t_1 \\ r_{20} & r_{21} & r_{22} & t_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_s \\ Y_s \\ Z_s \\ 1 \end{bmatrix} \quad 3.9$$

Ve vztahu 3.9 pak  $[x_s, y_s, z_s]$  respektive  $[X_s, Y_s, Z_s]$  vyjadřují souřadnice stejného bodu v kamerovém respektive světovém souřadném systému.

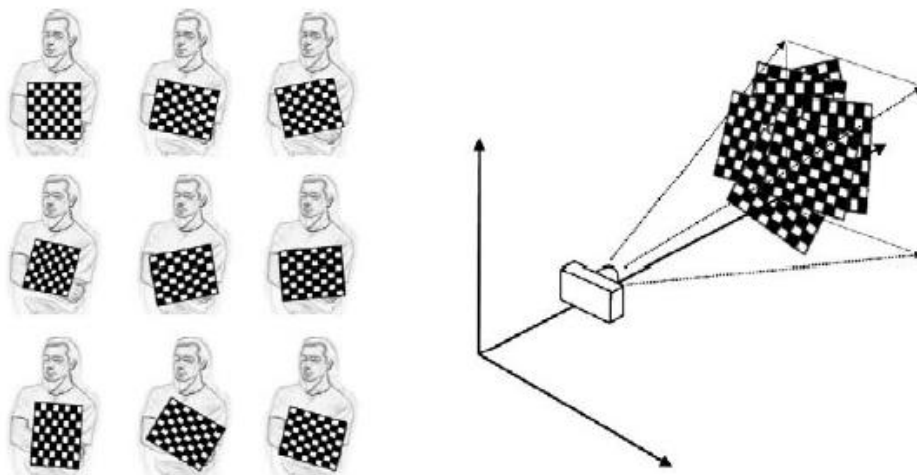


3.9 Transformace ze světových souřadnic do souřadnic obrazové roviny. Převzato z [34].

Předcházející informace ke kalibraci kamery (podkapitola 3.2.1) byly převzaty z [34].

### Šachovnice

Principiálně může být pro kalibraci použit libovolný, dobře charakterizovatelný objekt, přesto se ale v praxi užívá pravidelných, většinou ostrých, vzorů, jako je šachovnice. Literatura zmiňuje i trojrozměrné objekty, jako je třeba kostka pokrytá značkami, nicméně ploché šachovnicovité vzory jsou mnohem vhodnější k snadnému dosažení přesných výsledků. Je tomu tak zejména proto, že je komplikované vytvořit a distribuovat přesné 3D kalibrační objekty. V případě takovéto šachovnice pak kalibrace často probíhá na základě zpracování snímků šachovnice z několika úhlů pohledu (viz. obrázek 3.10).



3.10 Snímky šachovnice v různých polohách poskytují dostatečnou informaci k výpočtu globálních souřadnic šachovnice vůči kameře a zjištění vlastností kamery. [17]

Pro přesnou lokalizaci šachovnice ve snímku se využívá obdobného postupu jako při zpracování optického toku, v tom smyslu, že jsou nejdříve nalezeny rohové body (v tomto případě se jedná body na šachovnici, kde se vždy stýkají 2 černé a 2 bílé čtverce). Po jejich nalezení je potřeba zpřesnit jejich nalezenou pozici. Zjištěné informace pak slouží jako základ pro matematický model kalibrace. Vizualizace nalezené šachovnice je k vidění na obrázku 3.11.



3.11 Vizualizace nalezených rohových bodů při lokalizaci šachovnice

### 3.3 Existující přístupy

Oblast počítačového vidění je oblastí velmi rozšířenou a populární a vzniká zde velké množství vědeckých prací a projektů. V rámci této podkapitoly budou zmíněny vybrané existující přístupy, které více, či méně souvisí s tématem této práce, tj. práce zaměřené na vizuální lokalizaci, případně na fúzi vizuální lokalizace s lokalizací založené na bázi inerciálních senzorů. Mezi ty zajímavé můžeme, mimo jiné, zmínit následující:

- **Inteligentní senzorový systém pro vizuální odhad pohybu v reálném čase** (angl. A real-time smart sensor system for visual motion estimation) [35]. Jedná se o inteligentní senzorovou architekturu vizuálního odhadu pohybu na bázi optického toku. Tento systém pracující v reálném čase byl navrhován na míru pro autonomní vozidla (jako třeba MORIA [36]). Softwarovým jádrem systému je výše zmíněná metoda Horn-Schunk [31] pro efektivní zjišťování optického toku.
- **Optimální odhad pohybu z vizuálních a inerciálních měření** (angl. Optimal motion estimation from visual and inertial measurements) [37]. Stejně jako předchozí práce, i tato byla vytvářena s cílem zdokonalení autonomních vozidel s využitím kamery a inerciálních senzorů. Představuje algoritmus pro výpočet optimálního odhadu pohybu s ohledem na dostupná data z kamery, gyroskopu a akcelerometru. Tato práce ukazuje, že lze dosáhnout velmi přesného odhadu z kamery a inerciálních senzorů i v případě, že samostatně tyto zdroje nemusí dosahovat velké přesnosti.
- **Měření vzdálenosti mobilním telefonem s využitím jedno-kamerové stereovize** (angl. Measuring distance with mobile phones using single-camera stereo vision) [38]. Práce představuje řešení pro mobilní zařízení s jednou kamerou. Vytvořený prototyp

počítá stereo-vizi na základě dvou dostupných snímků ze stejné kamery ale jiné polohy (například snímky z video-sekvence). Výsledkem je schopnost měřit rozměry běžných objektů. Zajímavostí tohoto přístup je přinesení stereovize i na běžná mobilní zařízení s jednou kamerou. (Obdobných řešení je samozřejmě více).

- **Lokalizace kamery s využitím nekompletního šachovnicového vzoru** (angl. Camera localization using incomplete chessboard pattern) [39]. V této práci je představen přístup pro real-time lokalizaci kamery s využitím desky šachovnicového vzoru. Práce popisuje novou metodu pro rekonstrukci vzoru šachovnice z jejích neúplných snímků, což umožňuje správnou lokalizaci kamery i v případech, kdy je šachovnice částečně zakrytá cizím objektem. Systém navržený v této diplomové práci bude možné použít jako vylepšení právě tohoto přístupu a částečnou lokalizaci až za hranici viditelnosti šachovnicového vzoru.

Zmíněné přístupy jsou pouze velmi skromným výběrem z široké palety existujících řešení. Podrobnější výčet by však vydal na celou publikaci a je nad rámec této diplomové práce.

## 4 Dostupné nástroje

Tato kapitola stručně popisuje dva hlavní nástroje (knihovna OpenCV a platforma Android) aktivně využitě při implementaci demonstrační aplikace k této práci. Pro zájemce o podrobnější informace o těch nástrojích se odkáží na svou bakalářskou práci [40], případně na oficiální informační zdroje k těmto nástrojům

### 4.1 OpenCV

OpenCV je otevřená multiplatformní knihovna pro počítačové vidění napsaná v jazyce C a C++. Knihovna byla navržena tak, aby se z ní stal efektivní a rychlý nástroj použitelný při vývoji aplikací pracujících v reálném čase. Největší využití knihovna nalézá v oblasti počítačového vidění, v bezpečnostních a monitorovacích systémech a v umělé inteligenci, zejména v interakci stroj-člověk.

Knihovna nabízí velké množství funkcí pro práci s obrázky nebo sekvencemi obrázků (videem), například jejich vytváření a ukládání, kreslení základních geometrických útvarů a textu, aplikaci široké škály filtrů, převody mezi barevnými modely, matematické operace nad jedním a více obrázky. V hlavní řadě pak implementuje širokou škálu algoritmů počítačového vidění (např. algoritmy segmentace, hledání rohových bodů, fitting, kalibrace kamery, stereovize, práce s pyramidovým modelem obrázku a mnohé další). Umožňuje také snadné připojení kamery. [40]

V době vzniku této práce existuje knihovna OpenCV již ve verzi 2.4. Zajímavou současnou odnoží je pak implementace této knihovny pro mobilní operační systém Android. Ta umožňuje nejen práci s knihovnou přes nativní přístup s využitím Android NDK, ale nabízí také Java rozhraní pro snadnější použití knihovny na zmíněném operačním systému. Při použití aktuální verze OpenCV pro Android je potřeba mít na zařízení nainstalovaný tzv.: *OpenCV Manager*. Jedná se o správce instalace knihovny, který udržuje v systému vždy nejaktuálnější dostupnou verzi knihovny. Výhodou tohoto přístupu (na rozdíl od přibalení knihovny přímo k samostatné aplikaci) je právě zmíněná automatická kontrola nových verzí a také znatelná úspora na velikosti výsledných aplikací, které nemusí knihovnu obsahovat přímo v sobě. Nevýhodou pak je uživatelsky nepohodlná a matoucí nutnost instalovat aplikaci třetí strany.

### 4.2 Android OS

Platforma Android je softwarový balíček určený pro malá přenosná zařízení, jako jsou mobilní telefony, PDA, GPS navigace, notebooky a tablety. Android obsahuje operační systém a klíčové aplikace využívající potenciálu těchto zařízení. Pro více informací o této platformě se odkáží na svou bakalářskou práci [40], kde je podrobněji popsána.

Pro doplnění aktuálních informací jen stojí za zmínku, že OS Android je v současné době již ve verzi 4.2 (Jelly Bean). Tato verze, oproti předchozím, přináší víceuživatelské prostředí a hlavním rysem je sjednocení systému pro použití na široké škále zařízení, zejména pak na mobilních telefonech a tabletech.

#### 4.2.1 Android NDK

Android NDK je sada nástrojů, která umožňuje použití komponent využívajících nativního kódu. Android aplikace běží ve virtuálním stroji Dalvik a jsou psány v jazyce Java. NDK umožňuje implementovat části aplikací v nativním kódu v jazyce C nebo C++ a umožnit tak znovupoužití již

napsaného v těchto jazycích a hlavně v některých opodstatněných případech přináší urychlení zpracování. Vývojářům aplikací nabízí:

- Sadu nástrojů na kompilaci nativních knihoven napsaných v C/C++.
- Způsob, jak přidat nativní knihovny do distribučního .apk balíčku.
- Sadu nativních systémových hlaviček se zaručenou podporou v budoucích verzích.

V současné době není využití nativního kódu pro většinu aplikací žádným přínosem. Je to proto, že v mnoha aplikacích by jeho využití pouze zvýšilo nároky na implementaci, ale výsledná aplikace by nedosahovala většího výkonu.

Vhodnými kandidáty na využití schopností NDK jsou výpočetně náročné operace s vysokými požadavky na paměť, jako zpracování signálů, simulace fyzikálních jevů atd. Další z výhod je možnost efektivního znovu-užití velkých částí existujícího kódu napsaného v jazyce C/C++. V této práci jsem tak využil možnosti relativně snadno přeložit knihovnu OpenCV a těžit z jejích rozsáhlých schopností.

Platforma Android nabízí dva způsoby k použití nativního kódu:

- Napsání aplikace s využitím rozhraní Android a použití JNI (*Java Native Interface*) k přístupu k API poskytovaného Android NDK. Výhodou je možnost využívat pohodlného vývoje pro Android se zachováním možnosti přistupovat k nativním metodám. Tohoto způsobu jsem využil i já při implementaci aplikace k této práci (jedním z hlavních důvodů byla také nemožnost přistoupit ke kameře přímo z nativního kódu).
- Napsání nativní aktivity, která umožňuje implementaci všech metod životního cyklu aktivity. Tato možnost je pouze pro zařízení se systémem Android 2.3 a vyšším.

## 5 Návrh systému

Tato kapitola se zaměří na návrh samotného systému lokalizace. Budou zde uvedeny prvotní představy o výsledném systému, který by měl využívat fúze informací ze sensorových a obrazových dat. V kapitole budou představeny experimentálně zjištěné možnosti jednotlivých přístupů a bude zde prezentována představa o jejich budoucím propojení.

### 5.1 Senzorová část

Vzhledem k představě o systému využívajícím sensorové fúze, je třeba v první kroku ověřit, jaké vůbec možnosti využití sensorů máme na reálném mobilním zařízení. Je také třeba zjistit, s jakou přesností dovede zařízení sensorová data reprodukovat a najít omezení, která by v budoucnu mohla být problémem.

#### 5.1.1 Zkušební zařízení

Pro experimentování se senzory byl využíván převážně telefon **Google Nexus S**<sup>7</sup>, který je společností Google prezentován jako jeden z telefonů pro vývojáře na platformě Android. Nexus S obsahuje jak fyzické, tak softwarové senzory popsané výše. Veškeré grafy a konkrétní naměřené hodnoty v této kapitole pocházejí z testování právě na tomto zařízení.

Nexus S	
<b>Rozměry</b>	Výška: 123,9mm Šířka: 63mm Hloubka: 10,88mm Váha: 129g
<b>Procesor</b>	1 GHz Hummingbird, Cortex A8
<b>Operační systém</b>	Android 4.1.2
<b>Paměť</b>	16GB iNAND flash paměť
<b>Displej</b>	4,0' WVGA (480x800px)
<b>Konektivita</b>	GSM: 850, 900, 1800, 1900 HSDPA, HSUPA A-GPS Fi-Wi (802.11 n/b/g) Bluetooth 2.1 microUSB 2.0 NFC (Near Field Communication)
<b>Senzory</b>	3-osý akcelerometr, 3-osý senzor magnetického pole, 3-osý gyroskop, světelný senzor, proximity senzor, ...

Tabulka 5.1: Specifikace telefonu Google Nexus S

---

<sup>7</sup> Google Nexus S je telefon, který pro společnost Google Inc. vyrábí společnost Samsung. Vzhledem k tomu, že byl telefon v době svého vzniku navržen jako telefon pro vývojáře, obsahuje široké spektrum hardwarového vybavení, které může na jiných telefonech chybět. Telefon také obsahuje čistý operační systém Android bez modifikací a nástaveb výrobce.



## 5.1.2 Test senzorů

Zkušební telefon má zabudovaných několik senzorů. Pro zjištění jejich přesnosti, stability a zvážení jejich možností bylo na počátku provedeno několik testů.

## 5.1.3 Testování akcelerometru

Vzhledem k cíli diplomové práce, vychází testování akcelerometru z předpokladu, že pokud máme dostatečně přesný akcelerometr, měli bychom být schopni z jeho výstupů spočítat směr, rychlost a vzdálenost pohybu. Jak bylo zmíněno výše, akcelerometr měří zrychlení ve třech osách v  $m/s^2$ . Změřená akcelerace je vždy ovlivněna gravitační silou země:

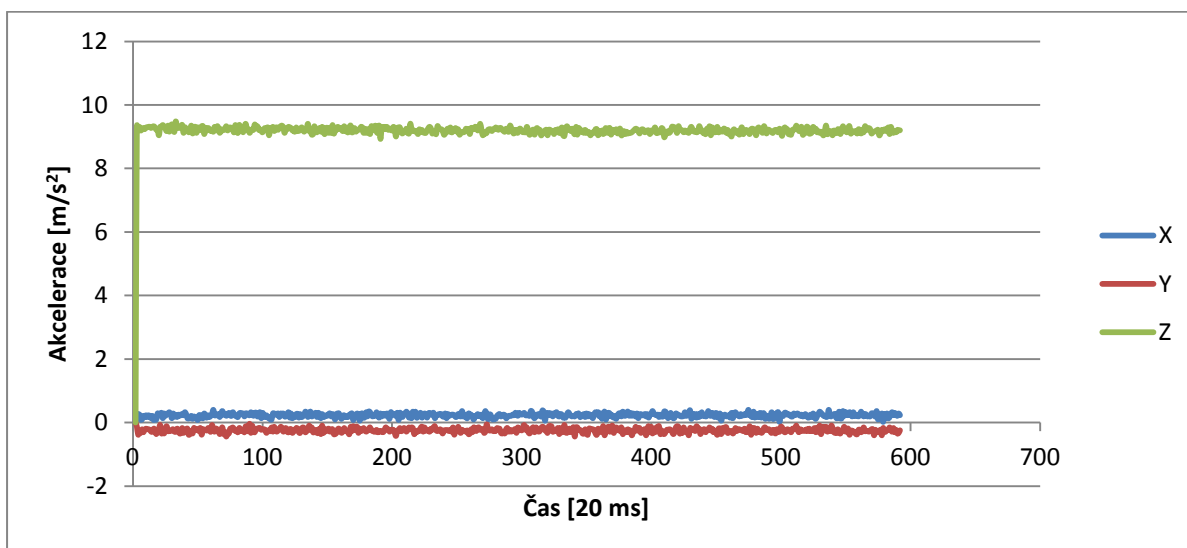
$$aa_d = -g - \sum F/m \quad 5.1$$

kde  $a_d$  je zrychlení zařízení,  $g$  je gravitační síla,  $F$  je síla působící na zařízení a  $m$  je jeho hmotnost. Symbol  $\sum$  reprezentuje součet pro osy X, Y a Z.

Není-li uvedeno jinak, jeden díl na ose x v následujících grafech značí časovou jednotku 20ms. Jedná se o maximální frekvenci odečítání dat na zvoleném testovacím zařízení.

### Měření v klidové poloze

Prvním testem bylo měření v klidové poloze telefonu. Graf tohoto měření je znázorněn na obrázku 5.1. Vypočítáme-li z naměřených dat celkovou akceleraci, vyjde nám číslo okolo  $9.65 m/s^2$  a ne očekávaných  $9,81 m/s^2$ . Již při takto základním měření tedy dostáváme odchylku okolo jednoho procenta, která nám v budoucím systému může generovat nezanedbatelnou chybu.



5.1: Výstup akcelerometru v klidové poloze pro osy X, Y a Z

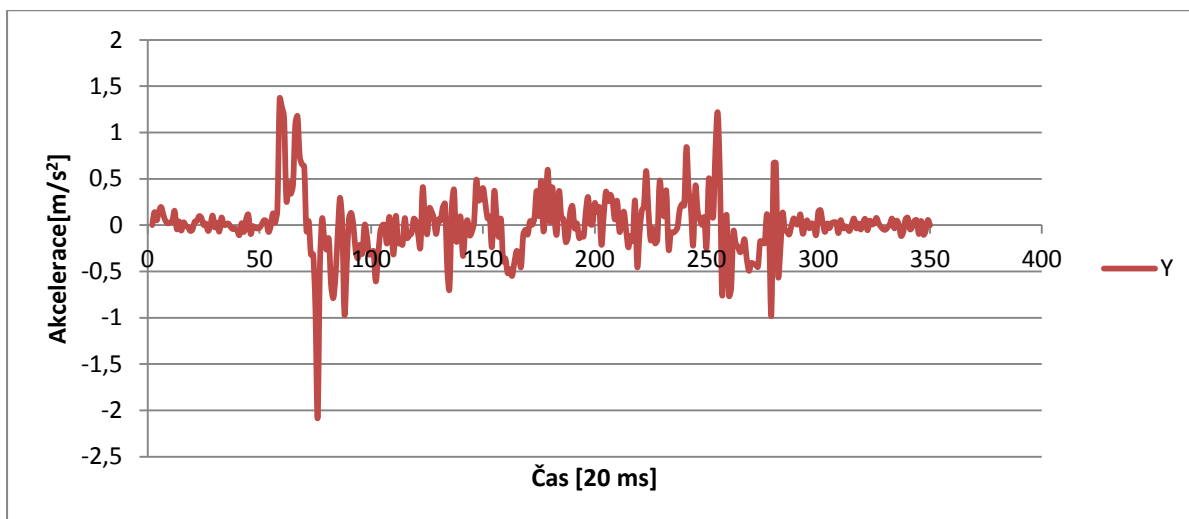
### Posun telefonu po rovné podložce

Dalším navrženým a vyzkoušeným testem byl test posunutí telefonu po rovné podložce na vzdálenost jednoho metru. Z tohoto testu pak dále vycházely experimenty s přepočtem zrychlení na rychlost a vzdálenost.

Pro lepší vypovídací hodnotu tohoto testu byla z výsledku odečtena gravitační složka podle vzorce 5.2:

$$\begin{aligned} x' &= x - w_1 g \\ y' &= y - w_2 g \\ z' &= z - w_3 g \end{aligned} \quad 5.2$$

kde  $x', y', z'$  jsou hodnoty výsledné,  $x, y, z$  jsou hodnoty měřené akcelerometrem  $w_1 + w_2 + w_3 = 1$  jsou váhy vypočtené na základě kalibrace telefonu v klidové poloze. Jedná se o takzvanou **lineární akceleraci**. Po aplikaci tohoto vzorce na naměřená data dostaneme graf znázorněný na obrázku 5.2.

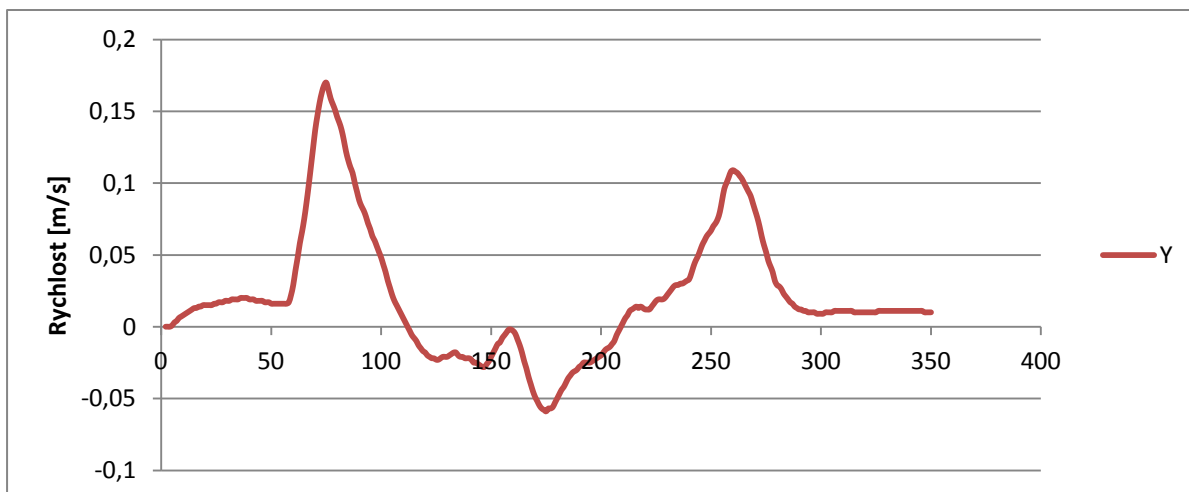


5.2: Posun telefonu po rovné podložce na vzdálenost 1m. Lineární akcelerace.

Na základě výše zjištěné lineární akcelerace je možné dopočítat rychlost  $v$  tak, že integrujeme lineární akceleraci  $a$  v čase  $t$  [4]:

$$v_n = \sum_{i=0}^n (\alpha_i * \Delta t) \quad 5.3$$

Na obrázku 5.3 je znázorněna rychlost spočítaná z dopředné akcelerace v předchozím kroku. Ačkoliv se průběh křivky zdá být v souladu s očekáváním, za povšimnutí stojí hodnoty menší než nula, které by značily zpětný pohyb. Ten však při testu nenastal. Dále se tyto chyby kumulují a ke konci, kdy zařízení již bylo opět v klidu, není naměřená rychlost nulová. Lze tedy očekávat distribuci takové chyby.

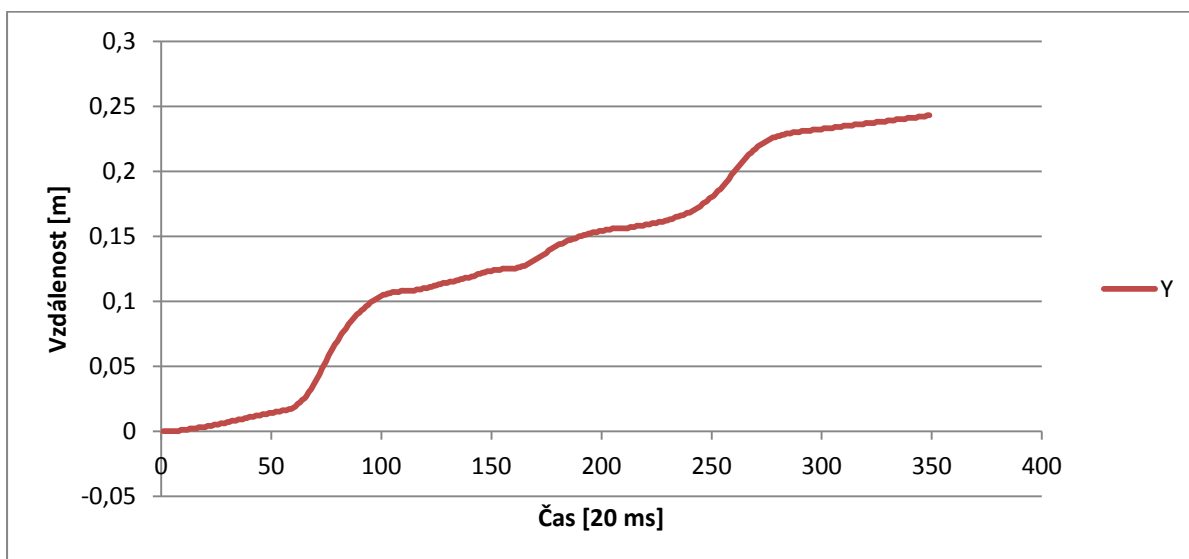


5.3: Rychlost vypočtená z dopředné akcelerace

K vypočtení vzdálenosti  $d$  je třeba integrovat rychlost  $v$  podle času  $t$  [4]:

$$v_n = \sum_{i=0}^n v_i * \Delta t + \frac{1}{2} \alpha_i \Delta t^2 \quad 5.4$$

Obrázek 5.4 ukazuje takto změřenou vzdálenost při stejném testu na 1m dopředného pohybu ve směru osy Y.

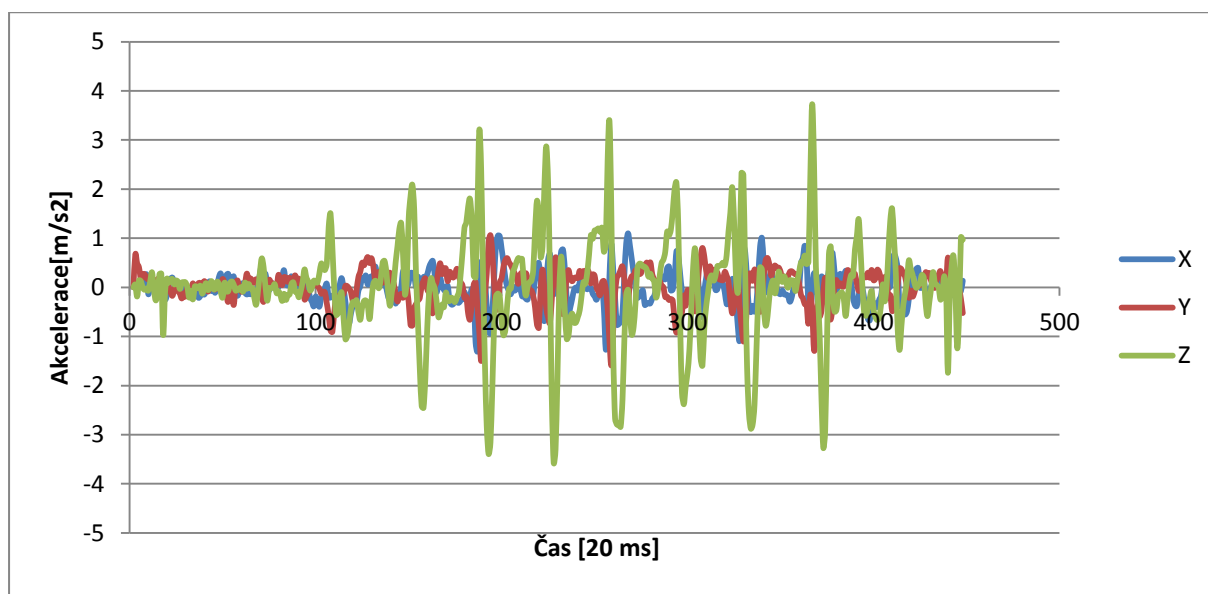


5.4: Vzdálenost vypočtená z dopředné akcelerace

Průběh se opět jeví jako očekávaný. Bohužel je však celková uražená vzdálenost přibližně čtyřnásobná oproti vzdálenosti naměřené akcelerometrem. Je zde také vidět velký postupný růst chyby, projevující se přibývajícím vzdáleností i v klidových polohách (viditelné zejména na konci vynesené křivky).

## Měření při chůzi

Pro možnost porovnání s výsledky z článku [4] bylo provedeno také měření při chůzi. Obrázek 5.5 ukazuje naměřené hodnoty při chůzi.



5.5: Test chůze - 8 kroků s telefonem v ruce

Opět se jedná o výsledky po odečtení gravitační složky. Největší výkyvy jsou v grafu zřejmé na ose Z, jelikož byl telefon během testu držen tak, že osa Y byla kolmá k zemi a zadní strana telefonu směřovala směrem pohybu. Projevy akcelerace na ostatních osách je možné přičíst tomu, že telefon byl držen rukou a nebyl tedy dokonale stabilizovaný v jednom směru. Během tohoto testu bylo při chůzi uděláno osm kroků. Ty se v grafu dají snadno rozeznat velkými výkyvy křivky.

### Orientace telefonu

Díky projevům gravitace na výsledky akcelerometru a distribuci  $g$  do tří směrů akcelerometru jsme také schopni dopočítat orientaci telefonu vůči Zemi. Podle vzorce 5.5 získáme úhel natočení osy x vůči Zemi. Stejně tak vypočítáme i úhly ostatních os telefonu.

$$\alpha_x = \frac{x}{g} * 90 \quad 5.5$$

Provedené testy ukázaly, že takto získané údaje o orientaci telefonu jsou relativně přesné, nicméně pouze za předpokladu, že se telefon nehýbe. V opačném případě se na výstupu akcelerometru projevují i jiné síly, než gravitační.

### 5.1.4 Zhodnocení výsledků testování

Z výsledků provedených testů můžeme vyvodit, že s dodržením vhodných experimentálních podmínek a při správné kalibraci, jsme schopni na základě dat z akcelerometru vypočítat nejen směr pohybu, ale také jeho rychlost a uraženou vzdálenost. Bohužel však pro naše potřeby výpočtu z obecného pohybu bude problematické dosáhnout výsledků, ve kterých by nebyla zanesena velká chyba. Přesto lze ale předpokládat, že získané údaje budou užitečné pro výsledný systém, kde bude možné chybu regulovat na základě informací získaných z počítačového vidění.

## 5.2 Vizuální část

### 5.2.1 Kalibrace kamery

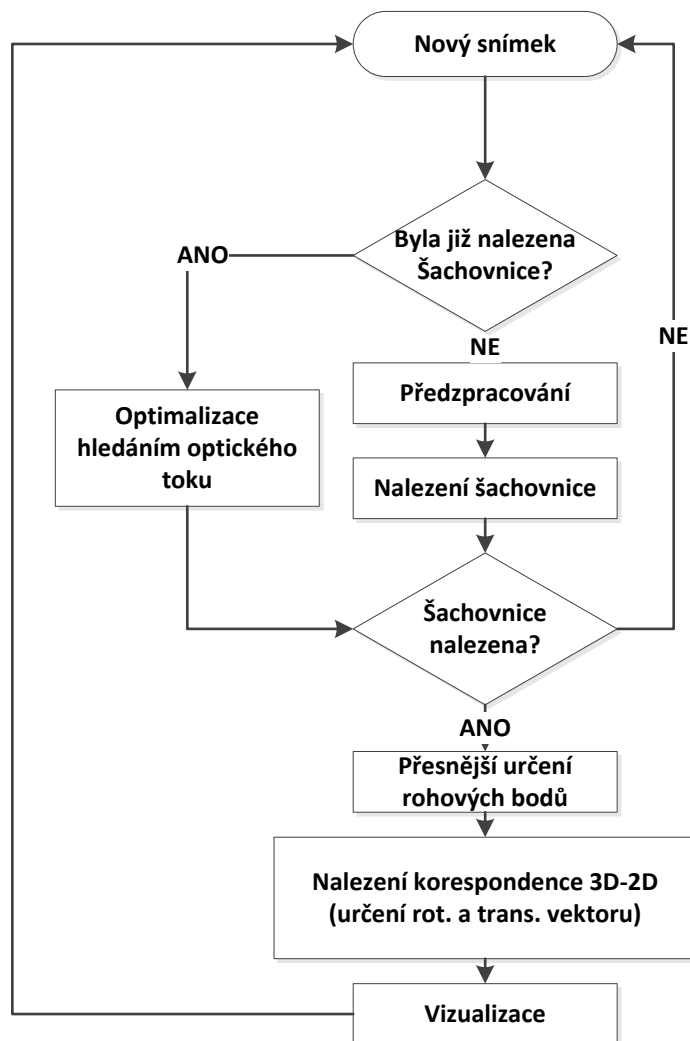
Pro možnost pracovat s reálnými metrickými hodnotami je potřeba nejdříve kalibrovat kameru a získat tak její parametry. Jako nejsnadnější cesta kalibrace kamery testovacího mobilního zařízení, se kvůli výpočetní náročnosti, ukázala kalibrace mimo mobilní zařízení. Na testovacím telefonu byly pořízeny snímky šachovnice o rozměrech 9x6 vnitřních rohů a tyto snímky byly zpracovány kalibrační utilitou na stolním počítači. Jak již bylo zmíněno v teoretickém úvodu, hlavním výstupem kalibrace je projekční matice kamery a koeficienty zkreslení kamery. Výstup po zpracování dostupnými nástroji knihovny OpenCV ve formátu XML může vypadat následovně:

```
<camera_matrix type_id="opencv-matrix">
  <rows>3</rows>
  <cols>3</cols>
  <dt>d</dt>
  <data>
    2.4731298256007667e+003 0. 1.2053301696810206e+003 0.
    2.4774964830008007e+003 9.1844123865234383e+002 0. 0. 1.</data>
</camera_matrix>
<distortion_coefficients type_id="opencv-matrix">
  <rows>5</rows>
  <cols>1</cols>
  <dt>d</dt>
  <data>
    3.2417597129744130e-001 -2.7498030030961269e+000
    -1.0124049316399633e-002 -7.0673321596143902e-003
    1.2079126706676455e+001</data>
</distortion_coefficients>
```

5.6 Matice kamery a koeficienty zkreslení obrazu zjištěné kalibrací testovacího zařízení na šachovnici o rozměrech 9x6 s velikostí čtverce 0.023m

### 5.2.2 Navržený systém

Podsystém lokalizace mobilního zařízení na základě vizuálních dat byl pro tuto práci navržen na základě lokalizace šachovnicovitěho vzoru a určení polohy telefonu vůči němu. Díky údajům získaným při kalibraci jsme tak schopni zjistit přesnou polohu telefonu vůči známé poloze šachovnice. Se správně kalibrovanou kamerou telefonu jsme tuto polohu schopni změřit v metrických jednotkách.

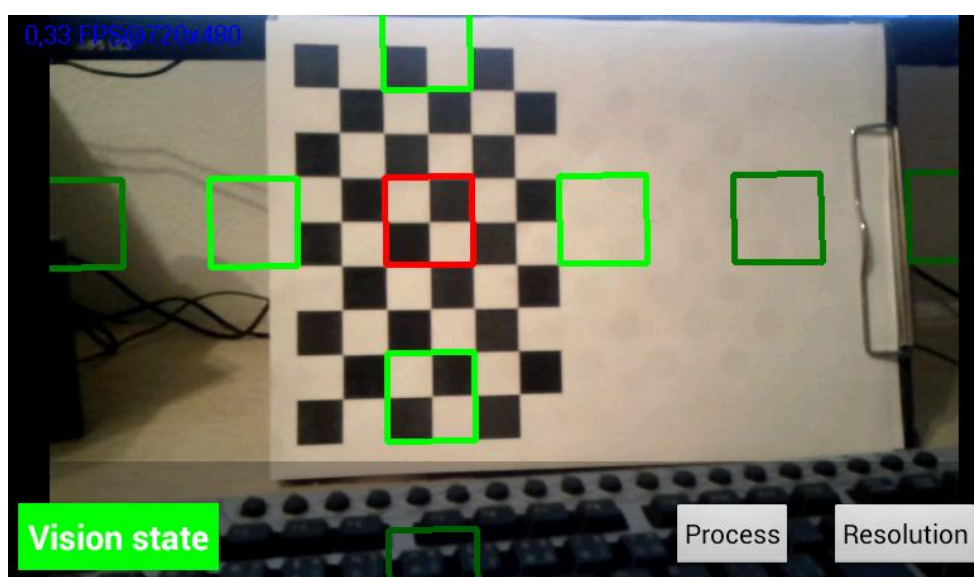


5.7 Diagram navrženého vizuálního systému

Celý vizuální systém (viz. obrázek 5.7) se pak skládá z následujících klíčových kroků:

- **Kalibrace kamery** - provede se pouze jednou pro každou kameru, tj. jednou pro jedno zařízení. Výsledky kalibrace se pak dají uložit a jsou časem neměnné, nedojde-li k nějakému poškození kamery zařízení.
- **Vyhledání rohových bodů šachovnice** - tento krok algoritmu (a každý následující) se pak již provádí pro každý snímek vstupní video-sequence. Ve snímku jsou vyhledány rohové body a zjišťuje se jejich příslušnost šachovnici. Zohledněny jsou přechody mezi vysokou a nízkou intenzitou v okolí bodu (přechody z černé na bílou barvu šachovnice a obráceně) a vzájemná poloha rohových bodů. V OpenCV je možné pro tyto účely využít funkce *findChessboardCorners()*. Jedním ze vstupních parametrů tohoto kroku jsou rozměry hledané šachovnice, konkrétně počet jejich vnitřních rohů. Ten by pro normální herní šachovnici tedy byl 7x7. V praxi je však vhodné využít nějakého asymetrického, obdélníkovitého tvaru šachovnice, s rozměry např. 5x8. Šachovnice s takovýmto rozměrem má pak pouze jednu osu symetrie a její orientace je tak jednoznačně identifikovatelná. (Ukázka zvýrazněných nalezených bodů šachovnice s rozměry 6x9 je na obrázku 3.11)
- **Zpřesnění rohových bodů** - pro přesnější výsledky je potřeba lépe určit pozici rohových bodů bez ohledu na omezení způsobená nižším rozlišením obrazu.

- **Nalezení korespondence** - v tomto kroku je hledána korespondence bodů 3D objektu v souřadnicích reálného světa mezi nalezenými 2D body ve snímku. Známe-li zde parametry kamery získané při kalibraci (matice kamery a koeficienty zkreslení), jsme zde schopni určit velmi přesně rotační a translační vektor polohy kamery vůči šachovnici. Tyto dva vektory nám pak tedy určují přesnou polohu zařízení vůči známé poloze šachovnice.
- **Vizualizace** - posledním krokem algoritmu pro každý snímek je pak vhodná vizualizace naměřených hodnot v demonstrační aplikaci na základě zjištěného rotačního a translačního vektoru. Pro tyto účely byl zvolen jednoduchý kříž se středem na šachovnici s klesající barevnou intenzitou směrem od středu (viz. obrázek 5.8).



5.8 Ukázka vizualizace naměřených hodnoty vykreslením objektu o známých souřadnicích v reálném světě.

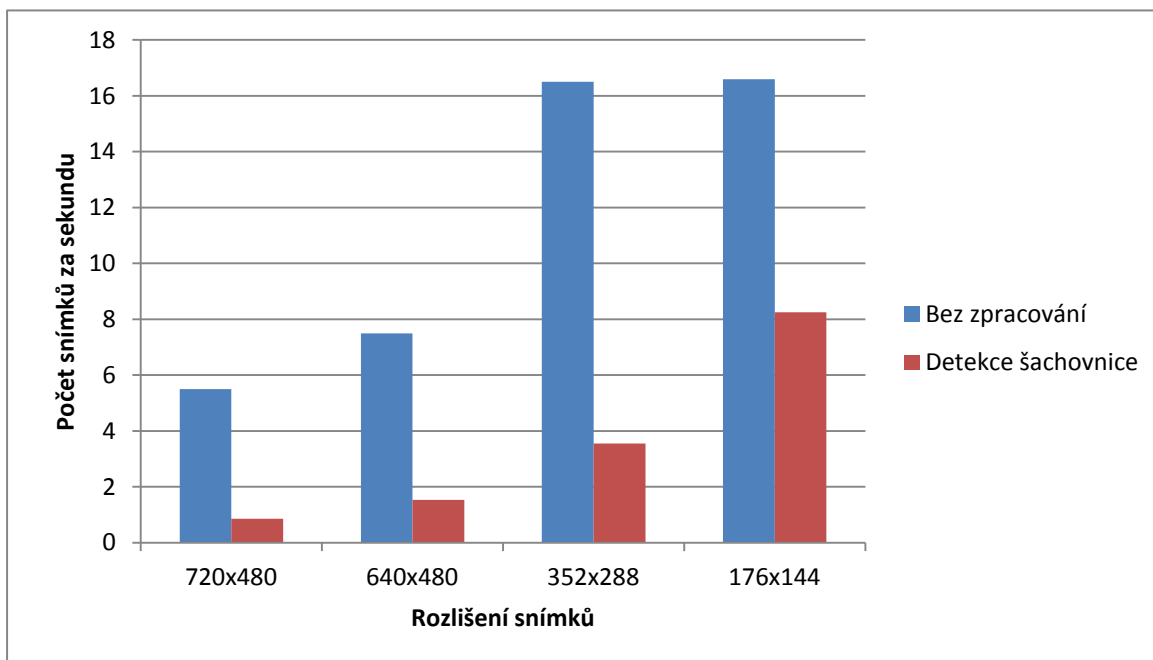
Takto navržený systém pak může sloužit například jako základ aplikace využitím rozšířené reality. Nevýhodou zde však zůstává skutečnost, že jsme stále závislí na viditelnosti sledované šachovnice (případně jiných vizuálních znaků). Tuto skutečnost se pokusí korigovat výsledný fúzní systém (kapitola 5.3).

### 5.2.3 Optimalizace

Během návrhu a implementace části systému, která zpracovává vizuální data z kamery fotoaparátu byl objeven největší nedostatek ve výkonnosti testovacího mobilního zařízení v kombinaci s použitou knihovnou OpenCV. Přesto, že se u testovacího telefonu nejedná o nějaký výpočetně podprůměrný stroj, bylo při prvotním testování zjištěno, že získaný počet snímků za sekundu nemusí být pro výsledný systém dostatečný a proto budou muset být provedeny nezbytné optimalizace na několika stupních zpracování. Pro ukázkou naměřených výkonnostních hodnot slouží graf 5.9. Graf znázorňuje poměr mezi rozlišením vstupního snímku a dosaženým počtem zpracovaných snímků za sekundu a to tak, že modrou barvou jsou vyneseny hodnoty bez dodatečného zpracování (pouze pořízení vstupního snímku přes knihovnu OpenCV a jeho navrácení pro zobrazení) a barva červená znázorňuje průměrnou dosaženou rychlost při detekci šachovnice přes NDK (výstup je k vidění na obrázku 3.11).

Za povšimnutí stojí zejména skutečnost, že i samotný proces pořízení snímku přes rozhraní knihovny OpenCV je relativně dosti pomalý a jak je vidět v nižších rozlišeních, není limitováno

pouze rozlišením, ale u nižších rozlišení se již zdá počet snímků konstantní, bez vlivu klesajícího rozlišení.



5.9 Výsledky zkoušky výkonu testovacího zařízení

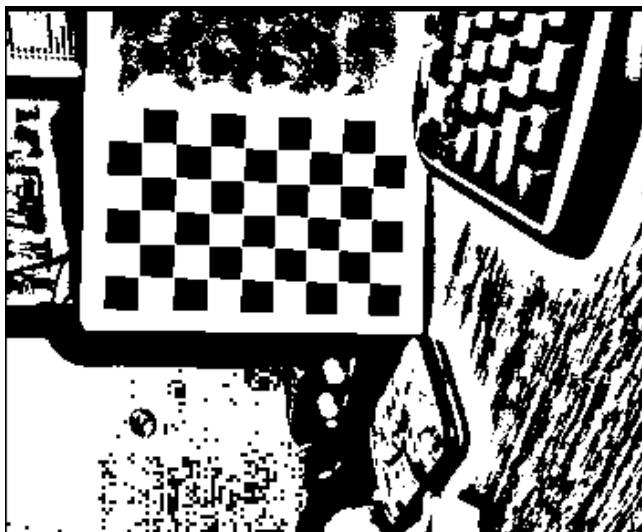
S ohledem na tyto zjištěné skutečnosti bude následující text popisovat některé kroky provedené k tomu, aby výsledný systém dosahoval dostatečné rychlosti pro zpracování v reálném čase.

### Předzpracování snímku

U většiny systémů zpracovávajících obraz je prvním krokem v optimalizaci (ať už směrem k rychlosti nebo spolehlivosti) vhodná před-příprava vstupních dat. Nejčastějšími úpravami jsou změny barevné hloubky snímku (například převod do šedotónové palety, binarizace snímku nebo jiná varianta prahování), morfologické operace nad snímkem (eroze, dilatace), podvzorkování (zmenšení rozlišení) a další.

I v případě vyvíjeného systému se ukázalo, že určité předzpracování vstupního snímku bude pro výsledek velmi přínosné. Prvním krokem v tomto směru je převod na šedotónový obraz. Tento krok je nezbytný už jen z toho důvodu, že některé funkce knihovny OpenCV šedotónový obraz přímo vyžadují. Dalším nezbytným krokem, jak již naznačuje měření na grafu 5.9, bylo podvzorkování. Snížení rozlišení má sice za následek menší rozlišovací schopnost, která se projevuje zejména při lokalizaci šachovnice, na druhou stranu je ale nezbytné kvůli nedostatečnému výpočetnímu výkonu. Poslední úpravou v tomto směru je vhodná binarizace obrazu s využitím algoritmu prahování (angl. thresholding). Morfologické operace nad obrazem v tomto případě nedávají příliš smysl, protože šachovnice je na snímku většinou dostatečně kontrastní a dobře extrahovatelná binarizací. Ukázka snímku po předzpracování je na obrázku 5.10.





5.10 Předzpracovaný (binarizovaný) snímek pro snazší detekci rohových bodů šachovnice

### Optimalizace optickým tokem

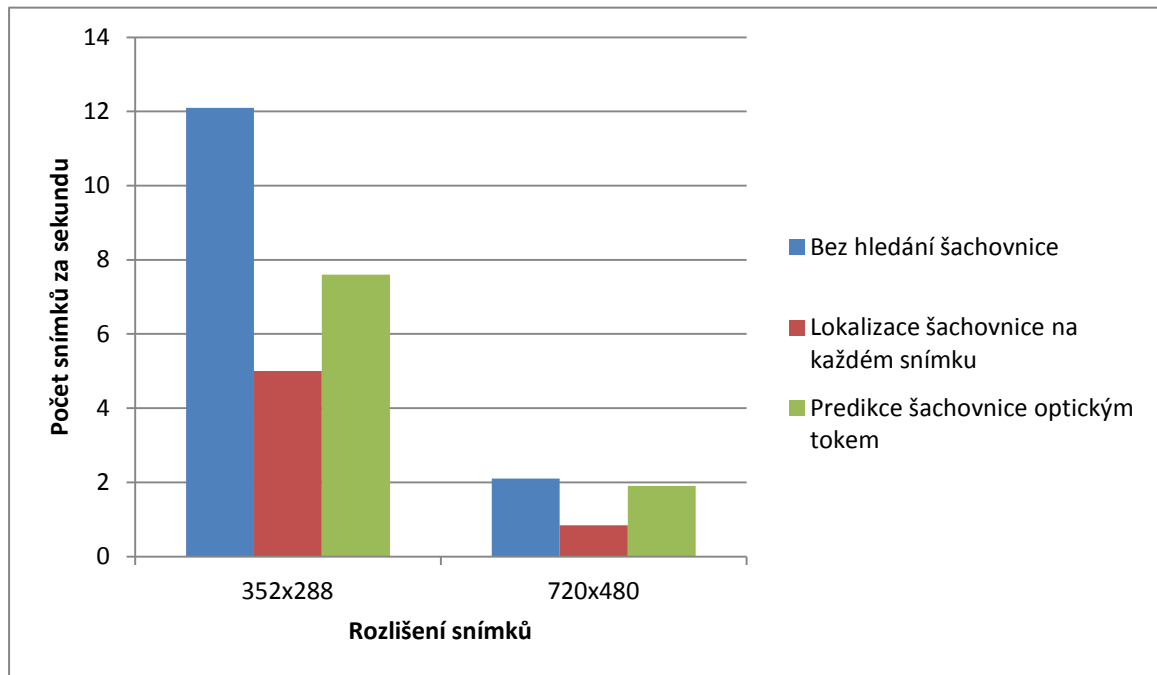
Úzkým hrdlem vytvořeného podsystému lokalizace na bázi počítačového vidění je lokalizace šachovnice. Při zpracování jednotlivých snímků bylo toto místo identifikováno jako místo s největší výpočetní náročností. Na úrovni výsledného systému je v nejjednodušší variantě toto místo řešeno voláním jedné funkce knihovny OpenCV, a to funkce *findChessboardCorners()*. Největší čas tato funkce zkonsumuje v případě, je-li viditelná pouze část šachovnice a algoritmus se ji snaží dohledat. V takovém případě je komplikované identifikovat, kdy má iterativní algoritmus skončit. Funkce umožňuje, jako jeden z parametrů, nastavit příznak *CALIB\_CB\_FAST\_CHECK*, který urychlí návrat z funkce, není-li šachovnice viditelná. V dokumentaci však není nijak detailně popsáno, jak se funkce při použití tohoto příznaku chová. Jako vhodné řešení se tedy jeví vyhnout se této funkci, není-li její použití nezbytné. Právě podobnou optimalizací podobného systému popisuje právě výše zmíněná výzkumná práce [39].

Jedna z uvedených optimalizací ve zmíněné publikaci je založena na využití predikce budoucí pozice rohových bodů šachovnice s využitím optického toku. V praxi tato metoda funguje tak, že je-li jednou lokalizována šachovnice, tak se na dalším snímku algoritmus pokusí o nalezení jejích rohů využitím metody Lucas-Kanade (str. 17). V případě úspěchu je daný snímek zpracován podstatně rychleji. Pokud predikce bodů selže, algoritmus se pokouší o obnovu opět standardním hledáním šachovnice.

Efekt této optimalizace je zobrazen na grafu 5.11. Graf zobrazuje dosažený počet snímků za sekundu na různých úrovních této optimalizace. Modrou barvou je znázorněna situace, kdy se šachovnice vůbec nevyhledává, tj. nalezne se pouze jednou a rohové body se uloží. U každého dalšího snímku se pracuje s body z prvního snímku. Tento stav samozřejmě znemožňuje získání relevantních dat při zpracování, nicméně slouží jako dobrá ukázka podílu lokalizace šachovnice a zbytku zpracování snímku. Červenou barvou je zobrazena situace, kdy se šachovnice lokalizuje (úspěšně) na každém snímku. Zajímavá je pak zejména zelená barva, která ukazuje přínos optimalizace predikcí rohových bodů hledáním optického toku metodou Lucas-Kanade. Měření

probíhalo tak, že na prvním snímku byla šachovnice lokalizována standardním způsobem a na každém dalším snímku pak byly pozice rohových bodů šachovnice obnovovány touto predikcí<sup>8</sup>.

Při reálném použití je však potřeba počítat s věrohodností predikce rohových bodů šachovnice a v případě, že predikce není dostatečně přesná (například pozice bodů byla odhadnuta špatně tak, že nalezené body netvoří mřížku) přistoupit k obnovení opětovnou lokalizací šachovnice standardním způsobem (tj. v našem případě použitím funkce *findChessboardCorners()*).



5.11 Vliv hledání rohových bodů šachovnice na rychlost zpracování snímku a vliv optimalizace predikcí optickým tokem v ideálním případě, kdy by stačilo šachovnici lokalizovat pouze jednou.

### Agregace na úrovni nativního kódu

Jak již bylo popsáno v kapitole 4.1, implementace knihovny OpenCV pro platformu Android funguje jako jakási obálka (angl. wrapper), která přes Java rozhraní volá funkce nativní knihovny napsané v jazyce C s využitím NDK. Z hlediska optimalizace je za dobrý přístup považováno omezení nativních volání na co nejmenší počet. Pokud tedy máme algoritmus volající více knihovnických funkcí, dobrým postupem je napsání vlastní nativní metody, která tento algoritmus agreguje na úrovni nativního kódu. S ohledem na tuto skutečnost bylo přistupováno i při implementaci demonstrační aplikace k této práci. Vzhledem ke komplexnímu propojení s využitím sensorových dat a celkovému konceptu aplikace však tyto optimalizace přinesly pouze minimální zrychlení běhu aplikace.

## 5.3 Výsledný systém

Poslední částí této práce, v návaznosti na lokalizaci telefonu s využitím dostupných hardwarových senzorů nebo lokalizaci na základě vizuální informace, je fúze těchto dvou možných přístupů a návrh systému, který bude robustnější a stabilnější, než systémy využívající pouze jeden z těchto přístupů. Následující kapitola popisuje, jak byl tento systém navržen, jak se jej podařilo implementačně ověřit a jakých bylo dosaženo výsledků.

<sup>8</sup> V případě hledání optického toku také není žádoucí obraz binarizovat, protože tak ztrácíme relevantní informace o predikovaných bodech.

### 5.3.1 Navržený systém

V návaznosti na metody zmíněné v předchozích kapitolách, s ohledem na existující přístupy byl navržen systém lokalizace mobilního telefonu na základě kooperace dvou podsystémů. Jednoho, založeného na zpracování vizuální informace a druhého, založeného na dostupných informacích z inertních senzorů. Tento návrh má výhodu v tom, že jsou jednotlivé podsystémy nahraditelné a výsledný systém je tedy možné vylepšovat a rozvíjet optimalizacemi jeho dílčích částí, například použitím jiného přístupu vizuální lokalizace.

Jak již bylo zmíněno v předchozí kapitole, jako podsystém vizuální lokalizace byl navržen systém lokalizace sledováním šachovnice v obraze prostoru před mobilním zařízením. Tento přístup byl inspirován zejména existujícím řešením popsáním v [39]. Zřejmou nevýhodou tohoto řešení je naprostá ztráta informace o poloze při ztrátě vizuálního kontaktu se šachovnicí. Vzhledem k této skutečnosti byl tedy brán ohled zejména na to, aby výsledný systém této práce byl schopen, alespoň v rámci možností inertních senzorů, udržovat co nejpřesnější informace o změnách polohy i v situacích, kdy se šachovnice ztratí z obrazu. Základní pohled na tento navržený systém je zobrazen na diagramu 5.13. Jak probíhá propojení vizuálního a sensorového subsystému bude popsáno v následující podkapitole.

#### Problémy výsledného návrhu

Jako největší problém výsledného navrženého systému byla velmi omezená možnost vyjádření translačního vektoru na základě dat z akcelerometrů. Jak ukazují experimenty provedené v kapitole 5.1, derivacemi výstupů akcelerometru bohužel nejsme schopni dostat příliš exaktní hodnoty. Není problémem odhadnout, kterým směrem se telefon pravděpodobně pohybuje, nicméně je problém tento pohyb přesně kvantifikovat a určit tak translační vektor mezi snímky. Ukázalo se tedy, že použití akcelerometrů je dostačující například pro využití ve smyslu uživatelského rozhraní pro ovládání her a v podobných případech bez požadavku přesné kvantifikace.

Na druhou stranu se ukázalo, že jsme schopni, na základě senzorů, získat dostatečně přesnou informaci o natočení telefonu vůči gravitačnímu vektoru Země, a tedy přepočítat ji na rotační vektor. Obrázek 5.12 znázorňuje experiment, kdy byla rotace kolem osy z získaná z vizuálního zpracování nahrazena rotací kolem osy z ze sensorového rotačního vektoru.



5.12 Vizualizace objektu s rotačním vektorem částečně složeným z informací získaných ze senzorů (rotace kolem osy z)

## Fúze

Jak již bylo zmíněno, propojení subsystémů je znázorněno na diagramu **5.13**. Hlavní myšlenkou je, aby si systém udržoval povědomí o současné poloze zařízení i v situacích, kdy selhávají údaje z vizuálního subsystému. K tomu může docházet například při zmizení značky z obrazu (v našem případě šachovnice), překrytí značky, nemožnost nalezení rohových bodů u subsystému fungujícího bez využití značek (například přechod přes bílou zeď). Algoritmus výsledného systému implementovaného v rámci demonstrační aplikace a ověřeného v praxi je tedy následující<sup>9</sup>:

- **Kalibrace kamery**
- **Vyhledání rohových bodů a upřesnění jejich pozice** - při prvním nalezení rohových bodů je uložen rozdíl mezi počátečními údaji z vizuálního subsystému (vztažené k šachovnici) a ze sensorů (vztažené k Zemskému povrchu).
- **Nalezení korespondence** - v tomto kroku, za předpokladu, že selže hledání v obraze, je využito rotačního vektoru, který je dopočítán z údajů ze sensorů odečtením rozdílu získaného při inicializaci systému (první nalezení celé šachovnice) následovně:

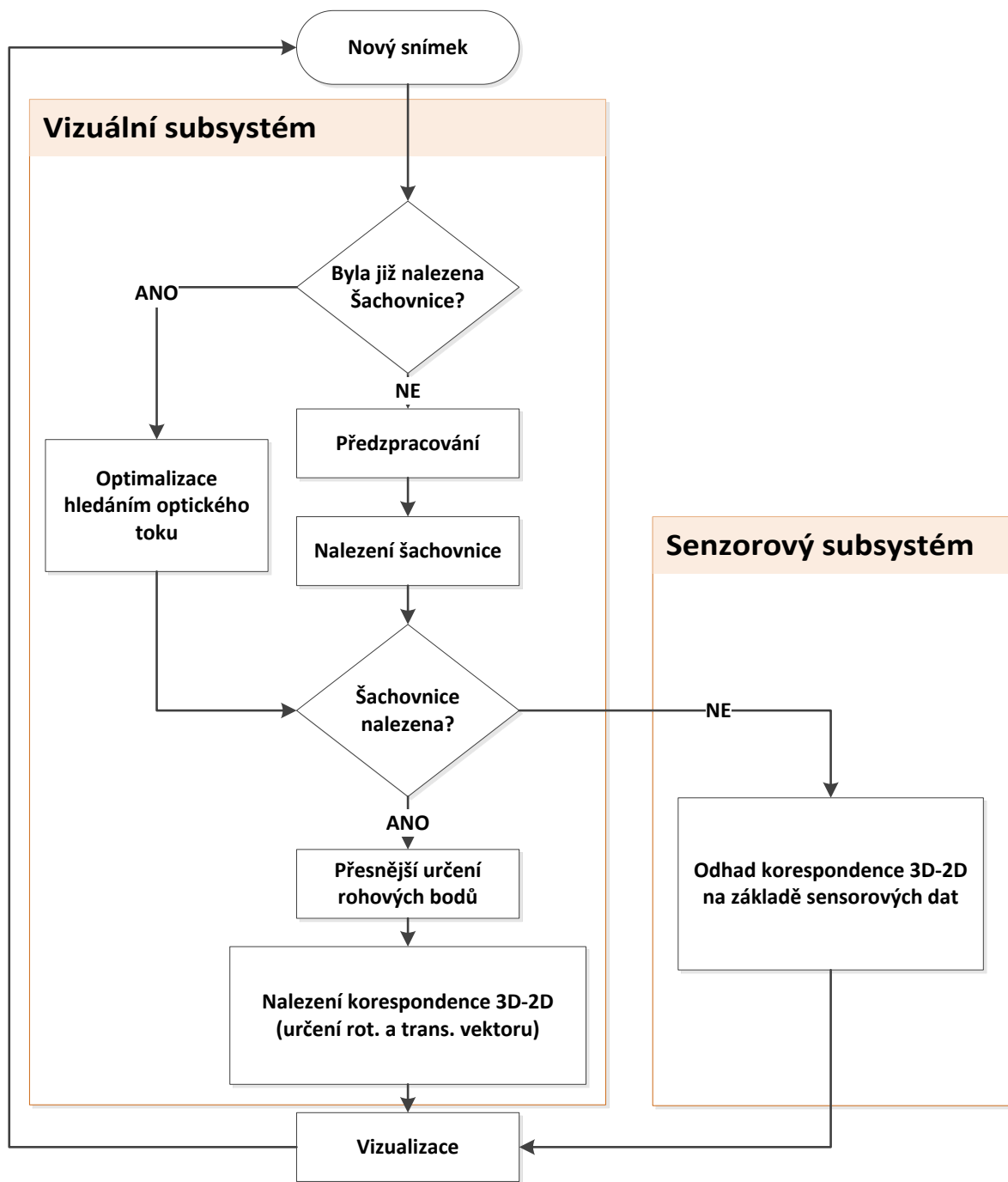
$$\begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} - \begin{bmatrix} X_i \\ Y_i \\ Z_i \end{bmatrix} \quad 5.6$$

$X', Y', Z'$  jsou výsledné rotace kolem os,  $X, Y, Z$  jsou rotace kolem os získané ze sensorů a  $X_i, Y_i, Z_i$  jsou korekce získané při inicializaci, tj. rozdíl mezi údaji z obrazu a ze sensorů. Schopnost systému si v takovém případě zachovat informaci o poloze, i když není šachovnice viditelná (tj. vizuální systém selže), je znázorněna na obrázku **5.14**.

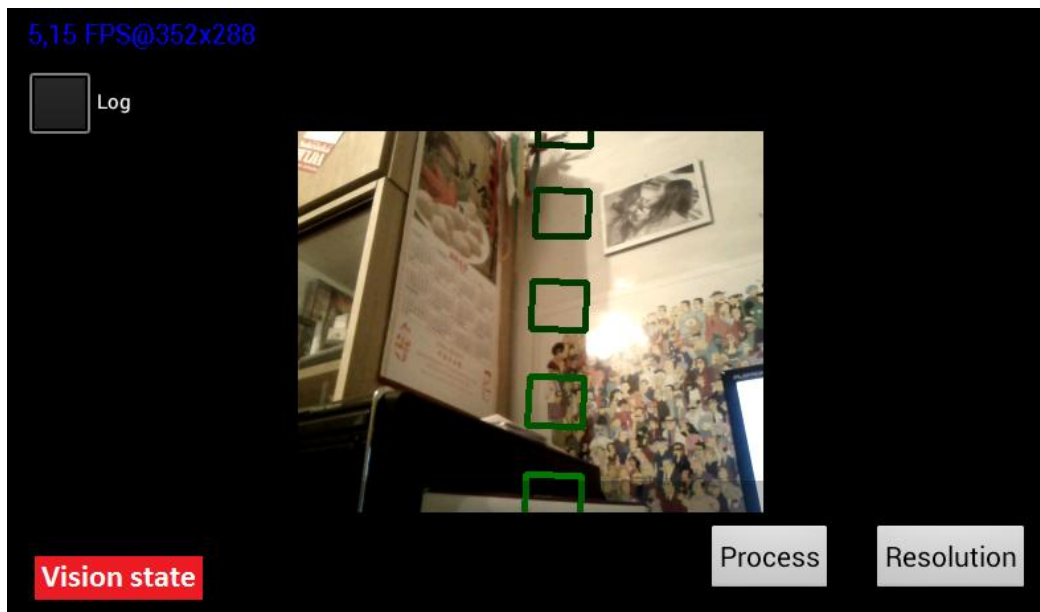
- **Vizualizace**

---

<sup>9</sup> Body rozepsané již v kapitole 5.2.2 zde nejsou znovu podrobně probírány. Pokud je třeba, jsou pouze doplněny o změny ve výsledném systému.



• 5.13 Vizualizace klíčových částí výsledného systému



5.14 Vizualizace na základě sensorových dat při ztrátě vizuálního kontaktu se šachovnicí

Takto navržený systém byl pak testován v modelových situacích a podařilo se ukázat, že je možné relativně dobře doplňovat vizuální informaci o poloze o údaje získané ze sensorů zařízení. Vzhledem k tomu, že však u řešení čistě na bázi sensorů s přibývajícím časem vzrůstá chyba měření, je velmi vhodné tuto chybu korigovat přesnějšími vizuálními měřeními. Problémem výsledného systému je skutečnost, že pro obecný pohyb je třeba znalost rotačního i translačního vektoru a jak již bylo zmíněno výše, dostatečně přesného určení translačního vektoru pro obecný pohyb se v rámci této práce dosáhnout nepodařilo.

Ve výsledku se však podařilo ukázat, že využití sensorů přidává velké možnosti rozšíření systémům založených na počítačovém vidění a existuje zde široký prostor pro budoucí výzkum.

## 6 Závěr

Tato diplomová práce si vzala za úkol prozkoumat různé možnosti lokalizace mobilního zařízení v prostoru a na jejich základě navrhnout vhodný systém lokalizace s využitím vizuální informace z obrazu video kamery, doplněné o údaje ze senzorů telefonu. Po teoretické stránce zde bylo nastíněno velké množství přístupů, které je možné pro lokalizaci telefonu použít. Tyto přístupy byly, vzhledem k jejich rozmanitosti a odlišnostem, charakterizovány tak, aby čtenář po přečtení dovedl lehce vyvodit, jaký přístup by byl vhodný ke konkrétní aplikaci v praxi.

Kromě samotného výzkumu rozličných možností současných mobilních telefonů, je přínosem této práce zaměření se na vestavěné senzory, provedení experimentů s těmito senzory a vyvození závěrů z těchto testů, vzhledem k použití při lokalizaci mobilního zařízení v prostoru, zejména pak při fúzi s informacemi získanými z obrazu. Dále pak představuje systém lokalizace telefonu na základech vizuální informace, který se odráží od článku [39] a tento přístup se posléze snaží zdokonalit o uchování informace o poloze zařízení i v případě, že čistě vizuální zpracování selže.

Pro účely testování vznikly dvě jednoduché aplikace pro platformu Android. První z nich, sloužila jako zdroj dat pro výzkum možností senzorů telefonu a umožňuje zobrazit podrobné informace o dostupných senzorech na daném zařízení a logovat do souboru výstupy ze zvolených senzorů. Tyto výstupy lze pak snadno zpracovat ve vhodném tabulkovém procesoru. Druhá aplikace pak implementuje výsledný navržený systém a vizualizuje zjištěné informace o poloze.

Z experimentů vyplynulo, že díky nepřesnostem a fyzikálním vlastnostem senzorů, nejsme schopni přesně odhadovat pohyb a rotaci v trojrozměrném prostoru, nicméně v určitých časech jsme schopni získat přesné informace například o směru pohybu telefonu či jeho natočení směrem k zemi. Tyto informace byly úspěšně využity jako doplněk k algoritmům počítačového vidění v místech, kde samotné počítačové vidění selhává.

V rámci práce se tedy podařilo dokázat, že fúze počítačového vidění a senzorů mobilních telefonů může vylepšit výsledný systém lokalizace zejména z pohledu robustnosti. Výsledný systém je pak odolnější vůči selhání a umožňuje zotavení nebo alespoň přibližné zachování funkcionality i v situacích, kdy systémy založené pouze na počítačovém vidění selhávají. Podařilo se navrhnout modulární systém, ve kterém bylo ve vizuální části využito lokalizace s využitím značek (v tomto případě šachovnice), nicméně navržený systém je natolik obecný, že využití sensorových informací je aplikovatelné i na jiné přístupy vizuální lokalizace (například přístup bez využití značek zmíněný v [41]).

Tato práce spojuje obecné poznatky z různých odvětví a spojuje je do jednoho funkčního celku. Přesto však bylo cílem ověření navrženého systému a výsledný systém i demonstrační aplikace poskytují velký prostor k optimalizacím, ať už ze strany robustnosti, tak ze strany výpočetní náročnosti. Prostor k optimalizaci také nabízejí využití nástroje, protože se při vývoji ukázalo, že využitá knihovna OpenCV pro Android ještě není zcela optimální a například poskytuje relativně nízký počet vstupních snímků z kamery za jednotku času.

# Literatura

- [1] NOHEJL, P. *Rozšířená realita pro platformu Android*. Brno: 2011. diplomová práce. FIT VUT v Brně.
- [2] POP, P. *Navigační systémy v geografických průzkumných akcích*. Brno: 2009. diplomová práce. FIT VUT v Brně.
- [3] NATIONAL COORDINATION OFFICE FOR SPACE-BASED POSITIONING, N. A. T. O. U. S. A. GPS Overview. *GPS.gov* [online]. verze 12.4.2012 [cit. 2012-12-08]. Dostupné z: <<http://www.gps.gov/systems/gps/>>
- [4] SHALA, U. a A. RODRIGUEZ. *Indoor Positioning using sensor-fusion in Android Devices*. Kristianstad, Sweden: 2011. graduation work. Kristianstad University, Department Computer Science.
- [5] WIKIMEDIA FOUNDATION. Wi-Fi. *Wikipedia* [online]. verze 12.11.2012 [cit. 2012-12-10]. Dostupné z: <<http://cs.wikipedia.org/wiki/Wi-Fi>>
- [6] BELL, S. W. R. JUNG a V. KRISHNAKUMAR. WiFi-based Enhanced Positioning Systems: Accuracy. In: *Proceedings of the 2nd ACM SIGSPATIAL International Workshop on Indoor Spatial Awareness*. San Jose, CA, USA: ACM, 2010, s. 3--9. ISBN 978-1-4503-0433-7.
- [7] MANU, O.. *A Study of Indoor Localization Techniques*. Suceava, Romania: 2009. Ștefan cel Mare University.
- [8] BLUETOOTH SPECIAL INTEREST GROUP. About the Bluetooth SIG: Overview. *The official Bluetooth SIG member website* [online]. verze 2012 [cit. 2012-12-10]. Dostupné z: <[https://www.bluetooth.org/About/bluetooth\\_sig.htm](https://www.bluetooth.org/About/bluetooth_sig.htm)>
- [9] WIKIMEDIA FOUNDATION. Near field communication. *Wikipedie* [online]. verze 6.12.2012 [cit. 2012-12-10]. Dostupné z: <[http://cs.wikipedia.org/wiki/Near\\_Field\\_Communication](http://cs.wikipedia.org/wiki/Near_Field_Communication)>
- [10] AYUB, S. et al. Pedestrian Direction of Movement Determination. In: *2012 Sixth International Conference on Next Generation Mobile Applications, Services and ....* 2012. s. 64-69.
- [11] BARTHOLD, C. K. P. SUBBU a R. DANTU. Evaluation of Gyroscope-embedded Mobile Phones. Denton, Texas: 2011, s. 1632-38. ISBN 978-1-4577-0653-0.
- [12] PENG, C. et al. BeepBeep: A High Accuracy Acoustic Ranging System. In: *Proc. 5th ACM Int'l Conf. Embedded Networked Sensor Systems (SenSys 2007)*. 2007. ACM Press, s. 166-75.
- [13] OH, J.O. M.S. LEE a S. LEE. An Acoustic-based Relative Positioning System for Multiple Mobile Devices. In: *Fourth International Conference on Computer Sciences and Convergence Information Technology*. Seoul, Korea: 2009, s. 1565-70. ISBN 978-0-7695-3896-9.
- [14] IPHONENESS. 40 Best Augmented Reality iPhone Applications [online]. [cit. 2013-01-04]. Dostupné z: <<http://www.iphoneness.com/iphone-apps/best-augmented-reality-iphone-applications/>>
- [15] ARTag [online]. verze 2009 [cit. 2013-01-04]. Dostupné z: <<http://artag.net/>>
- [16] GOOGLE INC. Indoor Google Maps help you make your way through museums. *Google Official Blog* [online]. 2012 [cit. 2013-01-04]. Dostupné z: <<http://googleblog.blogspot.cz/2012/07/indoor-google-maps-help-you-make-your.html>>
- [17] BRADSKY, B. a A. KAEHLER. *Learning OpenCV*. O'Reilly & Associates, Inc. 2008. ISBN 978-0-596-51613-0.
- [18] HARRIS, C. a M. STEPHENS. A combined corner edge detector. In: *Proceedings of the 4th*



- Alvey Vision Conference*. 1988, s. 147-51.
- [19] LUCCHESI, L. a S.K. MITRA. Using saddle points for subpixel feature detection in camera calibration targets. In: *Proceedings of the 2002 Asia Pacific Conference on Circuits and Systems*. 2002, s. 191-95.
- [20] LOWE, D.G. Distinctive image features from scale-invariant keypoints. In: *International Journal of Computer Vision* 60. 2004, s. 91-110.
- [21] HERBERT, B. et al. SURF: Speeded Up Robust Features. In: *Computer Vision and Image Understanding (CVIU)*. 110. 2008, s. 346-59.
- [22] SHI, J. a C. THOMASI. Good features to track. In: *9th IEEE Conference on Computer Vision and Pattern Recognition*. 1994.
- [23] EDWARD, R. a D. TOM. Machine Learning for High-speed Corner Detection. In: *European Conference on Computer Vision*.. s. 430-43.
- [24] ROSTEN, E. P. REID a D. TOM. Faster and better: a machine learning approach. In: *IEEE Trans. Pattern Analysis and Machine Intelligence*. 32. 2010, s. 105-19.
- [25] MATAS, J. et al. Robust wide baseline stereo from maximally stable extremal regions. In: *Proc. of British Machine Vision Conference*. 2002, s. 384-96.
- [26] RUBLEE, E. et al. ORB: an efficient alternative to SIFT or SURF. In: *ICCV 2011*. 2011.
- [27] LEUTENEGGER, S. M. CHLI a R. SIEGWART. BRISK: Binary Robust Invariant Scalable Keypoints. In: *ICCV 2011*. 2011.
- [28] ALAHI, A. R. ORTIZ a P. VANDERGHEYNST. FREAK: Fast Retina Keypoint. In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2012.
- [29] LUCAS, B.D. a T. KANADE. An iterative image registration technique with an application to stereo vision. In: *Proceedings of the 1981 DARPA Imaging Understanding Workshop*. 1981, s. 121-30.
- [30] Journal of Nuclear Materials. *Science Direct* [online]. [cit. 2013-19-04]. Dostupné z: <<http://www.sciencedirect.com/science/article/pii/S0022311510000899#fig1>>
- [31] HORN, B. K. P. a B. G. SHUNCK. Determining optical flow. In: *Artificial Intelligence* 17. 1981, s. 158-203.
- [32] SVOBODA, T. Kalibrace kamery ze (známých) rotací [online]. 2009 [cit. 2013-04-19]. Dostupné z: <[http://cmp.felk.cvut.cz/cmp/courses/Y33ROV/Y33ROV\\_ZS20082009/Lectures/Geometry/calibrotation-print.pdf](http://cmp.felk.cvut.cz/cmp/courses/Y33ROV/Y33ROV_ZS20082009/Lectures/Geometry/calibrotation-print.pdf)>
- [33] DURAISWAMI, R. Lecture for subject Scientific Computing on. *Camera Calibration*. Maryland: Department of computer science, University of Maryland, 2009.
- [34] ZAHŘÁDKA, J. *Rozšířené uživatelské rozhraní*. Brno: 2011. diplomová práce. FIT VUT.
- [35] RÖWEKAMP, T. a L. PETERS. A Real-Time Smart Sensor System for Visual Motion Estimation. In: *Proceedings of the 1997 European Design and Test Conference*. 1997, s. 613.
- [36] SURMANN, H. J. HUSER a L. PETERS. A Fuzzy System for Indoor Mobile Robot Navigation. In: *Proc. IEEE International Conference on Fuzzy Systems*. 1995.
- [37] STRELOW, D. a S. SINGH. Optimal motion estimation from visual and inertial measurements. In: *Proc. of the 6th IEEE Workshop on Applications of Computer Vision*. 2002.
- [38] HOLZMANN, C. a M. HOCHGATENER. Measuring Distance with Mobile Phones Using Single-Camera Stereo Vision. In: *32nd International Conference on Distributed Computing Systems Workshops*. 2012, s. 88-93.

- [39] ŠOLONY, M. et al. Camera localization using incomplete chessboard pattern. In: *Proceedings of the International Conference on Computer*. Algarve, PT: Institute for Systems and Technologies of Information, 2011, s. 415-18. ISBN 978-989-8425-47-8. Dostupné také z: <[http://www.fit.vutbr.cz/research/view\\_pub.php?id=9782](http://www.fit.vutbr.cz/research/view_pub.php?id=9782)>
- [40] KOMÁR, M. *Detektor pohybu pro platformu Android*. Brno: 2011. bakalářská práce. FIT VUT v Brně.
- [41] ŽÁK, P. *Pokročilá uživatelská rozhraní*. Brno: 2010. Teze dizertační práce. Fakulta informačních technologií, VUT v Brně, Ústav počítačové grafiky a multimedií.
- [42] HOLZMANN, C. a M. HOCHGATTERER. Measuring Distance with Mobile Phones Using Single-Camera Stereo Vision. In: *32nd International Conference on Distributed Computing Systems Workshops*. 2012, s. 88-93.
- [43] ROBERT, L. *OpenCV 2 Computer Vision Application Programming Cookbook*. Birmingham: Packt Publishing, 2011. ISBN 978-1-849513-24-1.
- [44] MEIER, R. *Professional Android™ Application Development*. Indianapolis, Indiana: Wiley Publishing, Inc. 2009. ISBN 978-0-470-34471-2.
- [45] GOOGLE INC. API Guides. *Android Developers* [online]. 2012, verze 2012-11-16 [cit. 2012-11-17]. Dostupné z: <<http://developer.android.com>>
- [46] OPENCV DEV TEAM. *OpenCV v2.4.3 documentation* [online]. verze 3.11.2012 [cit. 2012-12-20]. Dostupné z: <<http://docs.opencv.org/>>
- [47] GOOGLE INC. *Nexus S - Android Device Gallery* [online]. [cit. 2012-12-28]. Dostupné z: <<http://www.android.com/devices/detail/nexus-s>>
- [48] Integrovaný obraz. *Wikipedie - otevřená encyklopedie* [online]. [cit. 2013-04-19]. Dostupné z: <[http://cs.wikipedia.org/wiki/Integr%C3%A1ln%C3%AD\\_obraz](http://cs.wikipedia.org/wiki/Integr%C3%A1ln%C3%AD_obraz)>

# Seznam příloh

Příloha A: Obsah DVD

Příloha B: Manuál k demonstrační aplikaci

# Příloha A - Obsah DVD

Obsah DVD přiloženého k diplomové práci:

- **/doc/** - text této technické zprávy
- **/app/src/** - zdrojové soubory demonstrační aplikace k diplomové práci a předcházejícímu semestrálnímu projektu
- **/video/** - demonstrační video běhu aplikace
- **/photos/** - snímky obrazovky vzniklé v průběhu vývoje aplikace
- **/other/sensors/** - výstupy z testování senzorů
- **/other/calib/** - vstupní fotografie pro kalibraci a výstupní kalibrační soubor

# Příloha B: Manuál k demonstrační aplikaci

## Překlad aplikace

Pro úspěšný překlad aplikace je doporučeno postupovat podle standardního návodu nastavení vývojového prostředí pro vývoj pro platformu Android (<http://developer.android.com/sdk/index.html> [12.5.2013]). Dále je potřeba mít na počítači správně nainstalovanou nejnovější verzi knihovny OpenCV (<http://docs.opencv.org/doc/tutorials/tutorials.html> [12.5.2013]).

Při vývoji bylo využito:

- Eclipse SDK verze 3.7.1 + ADT plugin verze 21.1 (včetně NDK)
- Android SDK Tools verze 21.1 a SDK platform API 17
- OpenCV verze 2.4.5

## Ovládání aplikace

Aplikace po spuštění poskytuje následující funkční prvky (viz. obrázek **0.1**) a s nimi spojenou funkcionalitu:

- **Počítadlo snímků za sekundu** - v levém horním rohu se chvíli po spuštění aplikace začne zobrazovat orientační počítadlo zpracovaných snímků za sekundu.
- **Log** - levý horní roh, zaškrtačací "checkbox" umožňující povolit výpis aktuálního stavu rotačních vektorů (senzorového a vizuálního) na obrazovku.
- **Vision state** - levý spodní roh, indikátor vizuálního kontaktu s šachovnicí. Políčko je zelené, pokud se ve snímku podaří správně rozpoznat šachovnici. V opačném případě je políčko červené.
- **Tlačítko Resolution** - pravý dolní roh, tlačítko umožňuje změnu rozlišení na nižší a zpět. Je užitečné, pokud chceme dosáhnout rozumné rychlosti zpracování snímků, tj. zvýšení FPS (z angl. Frames Per Second).
- **Tlačítko Process** - pravý dolní roh, tlačítko pro zapnutí/vypnutí zpracování.
- **Checkbox X, Y, Z** - pravá horní strana obrazovky, zaškrtačací políčka umožňují nastavit, jestli se rotace kolem dané osy má počítat z vizuálního vstupu nebo získávat ze senzorů.
- **Výstup s vykresleným křížem** - pozadí, střed obrazovky, zobrazuje výstup aplikace, tj. v případě správně nalezené šachovnice vykresluje zelený kříž s červeným středem podle získaného rotačního a translačního vektoru. Červenými kroužky je zde také zvýrazněn bod [0,0] šachovnice.



0.1 Uživatelské rozhraní aplikace, popis funkčních prvků