

# Funkční vzor sledovacího systému – závěrečná zpráva

Technická zpráva - FIT - VG20102015006 – 2015 - 04

Ing. Filip Orság, Ph.D.



Fakulta informačních technologií, Vysoké učení technické v Brně

07. října 2015

## Abstrakt

Tato technická zpráva popisuje algoritmus a hardware použitý v systému pro sledování objektů se zaměřením na aplikaci v bezpečnostních systémech. Konkrétně je uvedeno řešení digitální stabilizace založené na a sledování objektu založené na částicových filtrech, které je optimalizováno z hlediska kvality a rychlosti nasazením grafických akceleračních pro účely zrychlení výpočtu algoritmů. Algoritmus je robustní a odolný vůči pohybu kamery. Poradí si dobře s monochromatickými záběry a je odolný vůči částečnému a úplnému zmizení sledovaného objektu. Po hardwarové stránce se systém skládá z kamerové hlavy vlastní konstrukce s objektivem s proměnnou ohniskovou vzdáleností, který umožní sledování objektů na vzdálenost několika kilometrů a dalšími prvky sloužícími pro řízení kamery. Dále je představena konstrukce nosného mechanismu s manipulátorem, dotekovým panelem a s integrovaným výpočetním systémem.



# Obsah

1 Úvod .....	1
2 Digitální stabilizace obrazu a sledování objektů .....	2
2.1 Digitální stabilizace obrazu.....	2
2.1.1 Stabilizace založená na korelaci .....	2
2.1.2 Stabilizace založená na analýze optického toku .....	3
2.2 Sledování objektů ve videu .....	4
2.2.1 Omezení algoritmů sledování objektů.....	6
2.2.2 Výběr bodů pro sledování.....	7
2.2.3 Metoda odčítání pozadí.....	8
2.2.4 Metoda založená na částicových filtrech .....	8
2.3 Shrnutí .....	10
3 Návrh kamerového systému pro sledování objektů .....	11
3.1 Prvky technického vybavení kamerového systému.....	11
3.1.1 Shrnutí.....	<b>Chyba! Záložka není definována.</b>
3.2 Zhotovení kamerového systému.....	13
3.2.1 Kamera.....	13
3.2.2 Objektiv.....	13
3.2.3 Manipulátor .....	14
3.2.4 Mechanická konstrukce jedno-kamerového systému.....	15
3.3 Elektronický systém kamerového systému.....	15
3.3.1 Elektronika kamerové hlavy .....	15
3.3.2 Hlavní výpočetní systém.....	16
3.3.3 Vlastní návrh hlavního výpočetního systému.....	<b>Chyba! Záložka není definována.</b>
3.3.4 Návrh sestavy výpočetního systému.....	<b>Chyba! Záložka není definována.</b>
3.4 Shrnutí .....	18
3.5 Shrnutí .....	<b>Chyba! Záložka není definována.</b>
3.5.1 Silné stránky algoritmu.....	<b>Chyba! Záložka není definována.</b>
3.5.2 Slabé stránky algoritmu.....	<b>Chyba! Záložka není definována.</b>
4 Závěr .....	20
5 Literatura .....	22

# 1 Úvod

Tento dokument popisuje funkční vzorek, který vznikl v rámci projektu MV ČR Nástroje a metody zpracování videa a obrazu pro boj s terorismem – VG20102015006. Detaily popisující jednotlivé části funkčního vzorku (hardware a software) jsou k dispozici v technických zprávách projektu VG20102015006 – 2013 – 03: *Jednokamerový systém pro sledování objektů* a VG20102015006 – 2014 – 04: *Algoritmus sledování objektů pro funkční vzor sledovacího systému*.

Cílem této práce je ukázat postup výzkumu a vývoje funkčního vzoru zařízení vhodného pro danou aplikační sféru. Základem bude specifikace cílového uživatele a způsobu užití výsledku. Cílovým uživatelem v našem případě bude policie nebo armáda, která má za úkol strážit bezpečnost. Požadavkem policie je vlastnit systém, který by byl schopen, po označení podezřelé osoby ve videu, tuto autonomně sledovat prostřednictvím systému PTZ – tedy otočné kamery s proměnnou ohniskovou vzdáleností. Podobně i z řad armády existují požadavky na pasivní systém schopný sledovat vybraný cíl. Výsledkem by měl být návrh sestavy kamery, manipulátoru, výpočetního hardware a software tak, aby byly splněny všechny požadavky na takový systém kladené. Spolu s požadavky postupně vyplynou i omezení daná především úrovní hardware a dostupností algoritmů. Pro dosažení konvergence požadavků a omezení bude nutné vybrat vhodný algoritmus, který bude odpovídajícím způsobem modifikován a optimalizován. Hardware bude založen na komerčně dostupných elementech a bude nutné navrhnout systém tak, aby byl i cenově přijatelný. Požadavky na algoritmus jsou pak již přesněji specifikovatelné na základě analýzy uživatelských scénářů.

Požadavky na systém lze shrnout do následujících bodů:

- Hardware: kamera s vysokým rozlišením a velkým rozsahem ohniskových vzdáleností (velký = ideálně FullHD, tedy zhruba 2 Mpx) a objektiv s velkou ohniskovou vzdáleností, nejlépe proměnnou a motoricky říditelnou, manipulátor se schopností rotace  $n \times 360^\circ$  a schopností náklonu, komunikace přes vhodné rozhraní, prostředky pro určení polohy (GPS)
- Software: integrované ovládání kamery, Možnost autonomního sledování vybraného objektu, sledování vybraného pohybujícího se objektu (i více objektů současně), odolnost vůči překrytí objektu jinými objekty, částečná odolnost vůči úplnému zmizení objektu, odolnost vůči jasovým změnám a absenci barvy (monochromatický záznam), sledování i při pohybu manipulátoru s kamerou, integrovaná digitální stabilizace obrazu, možnost záznamu videa na disk

Takto definované požadavky byly analyzovány a dále upřesněny dle aktuálního stavu trhu v době řešení. Důležité jsou požadavky dané vybraným algoritmem. V následující kapitole jsou stručně definovány základní pojmy, je zde popsána teorie digitální stabilizace a sledování objektů a algoritmus, který se blíží zadaným požadavkům svým výstupem, více viz technická zpráva projektu VG20102015006 – 2014 – 04. Další kapitola se pak zaměřuje na výběr technických prostředků s tím, že po stručném technickém rozboru současných technologií, byl vybrán takový hardware, který je schopen určený algoritmus spolehlivě provádět za daných podmínek, nebo alespoň dostatečně blízko tak, aby se po optimalizacích co nejvíce přiblížil podmínkám. Detaily popisující rozbor implementace a optimalizací, které zajistí plynulý běh algoritmu spolu se stručným shrnutím experimentálních výsledků je k dispozici v technické zprávě projektu VG20102015006 – 2013 - 03.

## 2 Digitální stabilizace obrazu a sledování objektů

### 2.1 Digitální stabilizace obrazu

Využívání kamerových systémů v různých konkrétních případech užití s sebou přináší mnoho problémů a jedním z nich je i roztřesení, rozhoupání, rotace a jiné nechtěné transformace obrazu. Nechtěné transformace obrazu způsobené mechanickými pohyby celé snímací soustavy nebo jejími částmi lze řešit buď mechanicky, nebo digitálně. Konkrétním příkladem nechtěného pohybu může být například situace, kdy na vysoký stožár umístíme kameru s velkou ohniskovou vzdáleností. V důsledku chvění stožáru, kterému lze těžko zabránit, dochází k malým výkyvům soustavy. V případě, kdy je zorné pole kamery například  $1^\circ$  a dojde-li k výkyvu stožáru o úhel  $0,2^\circ$ , způsobí to nemalé problémy – v podstatě dojde k pohybu v rozsahu  $1/5$  obrazu, což má za následek téměř nesledovatelný záznam na zobrazovacím zařízení. Řešením může být jiná mechanická konstrukce, což nemusí vždy být realizovatelné, nebo využití mechanické stabilizace s gyroskopy, což je velice drahé řešení, anebo se využije prostředků digitální stabilizace.

Digitální stabilizace obrazu je použitelná pouze v situacích, kdy nedochází k velkým výkyvům, které by měly za následek pohyb o více než polovinu zorného pole, neboť v té chvíli nebude dostatek informací k tomu, abychom jakoukoliv stabilizaci prováděli. Digitální stabilizace spočívá ve zpracování a analýze obrazu a následné kompenzaci zjištěného pohybu. Výhodou tohoto přístupu je zcela jistě cena, neboť jde o řešení elegantní a levné. Toto řešení je také značně flexibilní a lze ho dále rozšiřovat a modifikovat, případně integrovat s dalšími algoritmy, což z tohoto přístupu činí ideální metodu stabilizace. Nicméně je nutné myslet i na nevýhody, které toto řešení přináší. Během digitální stabilizace dochází ke ztrátě informace, neboť při korekci mezikamerového pohybu je nutné obraz ořezat, případně i jinak transformovat, což vede k deformaci a ztrátě informace. Dalším negativem je frekvenční omezení. Digitální stabilizace je schopna eliminovat lépe malé, pomalejší změny (nízká frekvence i amplituda) než rychlé změny s velkou dynamikou (vysoká frekvence i amplituda). Snímání obrazu kamerou samo o sobě filtruje obraz a pracuje jako nízkofrekvenční propust, neboť vysoké frekvence pohybu nelze zachytit díky nízké snímkovací frekvenci.

I přes své nevýhody je digitální stabilizace velmi často využívána například v digitálních fotoaparátech, kamerách a dalších zařízeních především pro poměr cena/výkon. Algoritmy řešící problém digitální stabilizace lze rozdělit například podle toho, jaký typ pohybu jsou schopny detekovat a kompenzovat. Obecně mohou činit problém translace, přiblížení/oddálení (změna měřítko), rotace kolem středu, rotace kolem dalších os a jejich kombinace.

#### 2.1.1 Stabilizace založená na korelaci

Digitální stabilizace obrazu založená na korelaci patří mezi algoritmy založené na hledání posunutí vzorové oblasti (*block matching*). Přesnost algoritmu je na úrovni pixelů [12], ale lze dosáhnout i subpixelové přesnosti [14], a algoritmus je schopen rozeznat pouze translační, případně i rotační pohyb, což nám pro účely stabilizace velmi často zcela postačuje. Později si ukážeme i komplexnější způsob stabilizace, který bude integrován spolu se sledováním objektu do jediného algoritmu.

Algoritmus jednoduchého hledání vzoru (*plain matching algorithm*) [12] je založen na definici korelace, z níž vychází. Principem je tedy nalezení vybraného vzoru v následujícím snímku videa pomocí korelace. Pokud je výpočet korelace prováděn v původní (časové) doméně, pak je jeho výpočetní složitost enormní (druhý problém korelace). Proto se často využívá vlastnosti korelace, která umožňuje výpočet provádět ve frekvenční doméně, což umožní výrazné zrychlení celého výpočtu díky možnosti využít rychlou Fourierovu transformaci (pro velmi malé oblasti a rozměry vzoru se však nevyplatí). I přes své nevýhody je korelace

z hlediska svého využití díky dobré implementovatelnosti ve specializovaném hardware velice zajímavá i pro zpracování v časové doméně. Korelaci lze například vypočítat speciálním technickým zařízením. Problémy, které toto řešení přináší, lze řešit různými způsoby (optimalizace hodnot, velikostí registrů, apod.), blíže se však o tomto řešení však zmiňovat nebudeme.

Problémy algoritmů založených na korelaci byly velmi často řešeny hranovými detektory, které obraz ve většině případů konvertují na binární a mají za úkol zvýraznit hrany. Kromě hran však mnohdy zvýrazní i místa, která mají původ především v šumu v obraze a z našeho pohledu mají nulovou vypovídací hodnotu. Nevýhodou tohoto přístupu je také výpočetní režie, kterou přidáváme do celého řetězce zpracování. Řešením jsou například algoritmy, které jsou založeny na hledání vzoru v obraze, který je binárně kódován tak, že nejužitečnější informace (obvykle vyšší bity v monochromatickém obraze) je zachována. Výsledkem je algoritmus [13], který pracuje pouze s bity, což nám umožňuje využít modifikovanou verzi korelační funkce, kde výsledkem je funkce výpočtu chyby, která vznikne nahrazením operace násobení binárním operátorem exkluzivního součtu.

### 2.1.2 Stabilizace založená na analýze optického toku

Kromě stabilizace založené na korelaci, bereme ještě v úvahu stabilizace založenou na analýze optického toku [15]. Tento typ stabilizace je velmi výhodný především proto, že se v rámci řetězce zpracování obrazu často optický tok počítá, ať už je využit ke stabilizaci nebo ne.

Optický tok (*optical flow*) je vektorové pole, které pro každý bod daného snímku v daném čase určuje směr a velikost pohybu vzhledem k následujícímu nebo předchozímu snímku (zde závisí na konkrétní aplikaci). Je nutné si uvědomit, že optický tok vyjadřuje pohyb v obraze, nikoli přímo pohyb objektu ve scéně (podobně jako metody odčítání pozadí). Jakákoliv korespondence detekovaného pohybu a pohybu sledovaného objektu musí být dodatečně zjištěna, tedy korespondence s vektorovým polem pohybu (*motion field*), které reprezentuje skutečný pohyb ve scéně, závisí nejen na povaze scény, ale i na způsobu snímání.

Z uvedeného plyne, že k aplikaci využívající optického toku existují dva základní přístupy. První skupina algoritmů určují takzvaný hustý optický tok (*dense optical flow*) – optický tok se počítá pro všechny body v obraze. Vezmeme-li v úvahu, že se v obraze vyskytují plochy plné podobných bodů, u kterých není snadné určit vektor pohybu (například jednobarevná plocha), je velmi obtížné počítat optický tok pro všechny body. V praxi se tento problém řeší interpolací obtížně počítatelných bodů. Což je cesta ke druhému způsobu řešení optického toku a tím je využití algoritmů ze skupiny řídkého optického toku (*sparse optical flow*). Tyto algoritmy vychází z definované množiny bodů, které by měly být snadno detekovatelná a tím i snadno sledovatelné. Logickou výhodou plynoucí ze snížení počtu sledovaných bodů je zvýšení rychlosti algoritmu. Nejznámějším algoritmem tohoto typu je algoritmus Lucas-Kanade. Jelikož je výpočetní náročnost algoritmů hustých optických toků příliš vysoká, zaměříme se na druhou skupinu algoritmů, konkrétně na algoritmus Lucas-Kanade [28].

Algoritmus Lucas-Kanade je standardním algoritmem výpočtu optického toku. Existují dvě varianty: algoritmus optického toku s rozptýlenými body využívající základní algoritmus Lucas-Kanade nebo iterační pyramidový Lucas-Kanade. První verze algoritmu byla představena v roce 1981 [28]. Problém tohoto řešení je, že počítá optický tok pouze pro malé okolí bodu a tedy je schopen postihnout pouze malé pohyby. Tento problém řeší novější varianta – iterační pyramidový algoritmus, který je schopen zachytit pohyby daleko větší.

Algoritmus Lucas-Kanade patří mezi diferenciální metody výpočtu optického toku. Tento algoritmus předpokládá, že posun mezi dvěma následujícími snímky ve videu (tedy i posun bodu) je relativně malý.

Předpoklady pro správnou funkci algoritmu jsou: konstantní jas (*Brightness Constancy*), časová stálost (*Temporal Persistence*) a prostorová soudržnost (*Spatial Coherence*).

Metodu Lucas-Kanade pro výpočet optického toku aplikujeme na množině bodů vhodných pro sledování (rohy) a typicky s malým okolím (například 5×5), nicméně z matematického hlediska není problém aplikovat metodu i na větší okolí, tedy na větší objekty. Tato metoda je vcelku spolehlivá, dokud jsou splněna omezení – tedy jasová stálost bodů, malý pohyb bodů a koherence pohybu bodů (tj. všechny body okolí se pohybují stejným směrem).

Řešením problémů základní metody je její varianta – iterativní pyramidální metoda Lucas-Kanade [29]. Tato metoda vychází z myšlenky, že dojde-li k pohybu dostatečně velkého objektu, je tento pohyb patrný i na zmenšeném snímku. Zmenšíme-li snímek, jsme schopni zaznamenat větší pohyb, protože jeden bod snímku reprezentuje dva body původního snímku. Snímek je možné zmenšovat tak dlouho, dokud budou změny v obraze (pohyb) dostatečně znatelné – vytvoříme tak pyramidu snímků poloviční velikosti v obou osách  $x$  a  $y$ , tedy celkový počet bodů klesne na čtvrtinu původní velikosti na každé další úrovni. Pokud zvolíme vhodný počet pyramid vzhledem k velikosti sledovaného objektu, je možné sledovat také rychlejší pohyby pokrývající větší prostor. Detailnější popis metody viz [29].

Celkově je nevýhodou metody optického toku neschopnost přizpůsobení se tvarovým změnám objektu způsobeným například změnou měřítka nebo rotací (přizpůsobení lze docílit dodatečnou modifikací vzoru, ale to není ideální dlouhodobé řešení).

Metoda stabilizace obrazu využívající optického toku vychází z předpokladu, že mezi jednotlivými snímky ve videu dochází ke změnám intenzity bodů v obraze v závislosti na pohybu kamery nebo objektu. Dojde-li k pohybu bodu, pak lze změnu pozice tohoto bodu vyjádřit vektorem. Takový vektor lze vytvořit pro každý bod obrazu kamery a na základě matice vektorů pohybu lze provádět další analýzu, například analýzu pohybu kamery. Při pohybu kamery dojde k posunu všech bodů a každý z nich tedy bude posunut v určitém směru v závislosti na druhu pohybu. V případě translačního pohybu, který nás především zajímá, dojde k posunu všech bodů ve stejném směru o vzdálenost závisující na vzdálenosti objektu a parametrech objektivu. Tímto způsobem lze zjistit průměrný pohyb každého bodu (nebo lépe střední hodnotu) a tím i pohyb způsobený pohybem kamery. V případě, že je zorné pole kamery z větší části zakryto blízkým pohybujícím se objektem, bývá tento pohybující se objekt chybně rozpoznán jako pohyb kamery, což je problém, který zatím není algoritmicky řešitelný.

Vektor pohybu bodu lze vyhledat pro každý bod obrazu, ale to není ideální řešení, neboť zabere příliš mnoho výpočetního času. Vhodnější je nalézt významné body v obraze, které posléze sledujeme. Podobného přístupu využíváme i při sledování objektu, kdy nalezneme významné body náležející objektu a snažíme se je nalézt ve snímku, který následuje, proto se jeví využití tohoto přístupu jako vhodnější. Nalezení bodů vhodných pro sledování je důležité například i pro měření vzdálenosti, pokud bychom ji chtěli měřit prostřednictvím dvojice kamer. V této situaci máme k dispozici dva snímky. Pro výpočet disparity bodů je nutné daný bod z prvního snímku nalézt i ve snímku druhém, je tedy účelné použít metodu výběru vhodného bodu, což zvyšuje užitečnost využití metody pro stabilizaci obrazu, protože ji můžeme využít i pro hledání podobných bodů (problém korespondence bodů) pro účely sledování objektu.

## 2.2 Sledování objektů ve videu

Algoritmus sledování objektů v posloupnosti po sobě jdoucích snímků (video) lze popsat velice jednoduše ve dvou krocích:

1. výběr bodů v originálním snímku,
2. nalezení bodů v následujícím snímku (snímcích).

Algoritmy sledování je možné rozdělit do různých skupin podle několika kritérií. Jedním z častých přístupů k dělení těchto algoritmů je rozdělení podle přístupu k řešení problému – například takto podle [9]:

- *segmentační algoritmy* – extrahují pohybující se objekty (popředí) na základě metod segmentace obrazu rozdělením na pozadí a popředí, tj. na základě algoritmické separace toho, co nás zajímá od toho, co nás nezajímá,
- *algoritmy využívající reprezentaci objektu* – slouží robustnímu sledování objektu, který je popsán modelem, tj. každý sledovaný objekt musí být možné popsat modelem, což není vždy splnitelné a někdy i nevhodné (v případě, že nevíme, o jaký typ sledovaného objektu půjde),
- *algoritmy využívající obrazové vlastnosti* – hledají objekt podle určitých obrazových vlastností (příznaků), které jsou extrahovány z původní obrazové informace, tj. objekty jsou vyhledávány například podle svého vzhledu – vzoru, který je následně hledán v následujících snímcích videa,
- *predikční metody* – algoritmy zaměřené na modelování pohybu a z něho vycházející nalezení objektu, využívají se pravděpodobnostní přístupy a některé z předchozích metod.

Dalším způsobem dělení, které nalezneme, je například rozdělení podle směru zpracování obrazu na procesy pracující zdola nahoru a algoritmy pracující shora dolů s dalším dělením podle přístupu, který dále využívají.

Procesy zdola nahoru typicky nějakým způsobem uchovávají reprezentaci objektu a snaží se ho nalézt (lokalizovat) v následujících snímcích. Mezi tyto metody patří například:

- *metody sledující malé oblasti (blob based)* – objekt je segmentován na dílčí části, jejichž pohyb je pak sledován metodami detekce oblastí, korelací nebo optickým tokem,
- *metody založené na podobnosti vzorů (kernel-based)* – iterativní přístup k lokalizaci, kde je cílem maximalizovat míru podobnosti na základě zvolené metriky – vyhledání vzorů,
- *metody založené na sledování kontur (contour tracking)* – je detekována kontura oblasti zájmu a ta je pak sledována například kondenzačním algoritmem [9],
- *metody založené na shodě vizuálních vlastností objektu.*

Procesy shora dolů jsou založeny na filtraci a datové asociaci. Tento způsob se zaměřuje na historii pohybu objektu a snaží se brát v potaz i jeho dynamiku a další vývoj. Tyto metody umožňují sledování komplexnějších objektů a patří mezi ně metody využívající:

- *Kalmanův filtr* – optimální Bayesův filtr pro lineární funkce ovlivněné Gaussovým šumem,
- *částicový filtr* – principiálně založené na rozdělení objektu na částice, jejich ohodnocení a sledování v závislosti na kvalitě částice, tento princip umožňuje sledovat i nelineární procesy.

Základní dělení však není tak podstatné. Mnohem důležitější jsou kritéria hodnocení algoritmů. Takovými kritérii může být například dělení podle odolnosti vůči specifickým druhům obrazových transformací (například odolnost vůči translaci, rotaci a podobně) nebo schopnost kvalitně plnit svůj úkol za různých okolností. Kritérii, která nás tedy budou zajímat, jsou například pohyblivost kamery, odolnost vůči transformacím, rychlost pohybu sledovaného objektu, apod.

Prvním kritériem, které nás bude zajímat, je vhodnost algoritmu pro případy, kdy je kamera statická (například metoda odčítání pozadí) nebo pohyblivá (například metoda vyhledávání vzoru, metoda založená na analýze optického toku, Kalmanův filtr, částicový filtr, apod., viz [9]). Kromě mobility kamery je důležitým údajem například schopnost algoritmu adaptovat se na změny tvaru objektu způsobené rotací, změnou měřítka, případně translací. Mezi metodami schopnými adaptace na transformace je například Kalmanův filtr (především jeho rozšířené varianty) nebo částicový filtr. Metody založené na hledání vzoru



jsou náchylné k rychlé ztrátě objektu [9]. Velmi důležitým kritériem je i schopnost reagovat správně na situace, kdy dojde k částečnému nebo úplnému zakrytí objektu jiným objektem nebo pozadím. S těmito problematickými situacemi se nejlépe vypořádají například částicové filtry. Posledním velmi důležitým kritériem v případě nasazení systému v praxi je i výpočetní náročnost, která bude také velkou měrou rozhodovat o výsledném výběru vhodné metody.

Pokud shrneme tato kritéria, je zřejmé, že ideálu se nejvíce blíží částicové filtry, které umožňují správně reagovat na mnoho podnětů a jsou schopny se zotavit i z chybových stavů.

V následujících kapitolách budou popsány základní definice problému, metody výběru vhodných bodů (resp. oblastí) pro účely sledování objektu a dále budou rozebrány jednotlivé metody sledování objektů.

Cílem sledování objektu je, abychom našli v aktuálním snímku videa objekt (*cíl*), který jsme si zvolili v předchozím snímku (nebo snímcích). Cíl je volen buď manuálně operátorem (například výběrem oblasti, kde se objekt nachází, nebo jiným způsobem popisu případně označení). Také lze využít automatický způsob detekce, který se však hodí spíše pro autonomní dohledové systémy, které kontrolují určitý perimetr a mají za úkol oznámit jeho narušení v případě detekce objektu, který do sledovaného prostoru nepatří. Lze se také setkat s kombinací obou přístupů, kdy operátor (uživatel) vybere oblast, kde se nachází objekt zájmu, který má být dále sledován, a tato oblast je poté analyzována algoritmem, který rozhodne o tom, jaké části z dané oblasti jsou vhodné pro účely sledování. Tato varianta je z mnoha hledisek nejpřínosnější, neboť operátor má možnost zasáhnout do procesu a případně korigovat chyby algoritmu.

Z hlediska uživatelského je tedy cílem objekt, který chceme sledovat. Z pohledu algoritmu je cílem množina bodů, které budou automaticky sledovány = nalezeny v posloupnosti následujících snímků. V následujících kapitolách je popsáno, jakým způsobem lze vybírat body vhodné pro sledování a jak objekty sledovat.

### 2.2.1 Omezení algoritmů sledování objektů

Abychom byli schopni nalézt řešení (například v podobě matic transformace bodů sledovaného objektu), zavádí různé algoritmy různá omezení. Mezi běžná omezení, s nimiž se setkáme, patří:

- pohyb objektu musí být plynulý bez náhlých změn,
- nedochází k náhlým změnám pozadí,
- nedochází k velkým změnám vzhledu objektu,
- kamera je statická,
- omezení počtu a velikosti sledovaných objektů,
- omezení počtu vzájemných interakcí (především překrytí),
- a mnohá další, které přímo souvisí s konkrétní aplikací i algoritmem.

Pro praktické nasazení sledování v reálných aplikacích je také nutné stanovit hranice, které pro nás budou rozhodující při výběru a implementaci vybraného algoritmu. Naše stanovené cíle určují, že *počet sledovaných objektů musí být větší než jeden, algoritmus musí být schopen řešit náhodné zakrytí sledovaného objektu* a důležité také je omezení v podobě nutnosti pracovat pouze s *jasovou informací* – tedy s monochromatickými kamerami, protože pro náš konkrétní případ uvažujeme sledování na velkou vzdálenost, kdy se barevná informace ztrácí. Mnoho algoritmů však vyžaduje pro svou funkci barevný obraz. Takové algoritmy je pak nutné přizpůsobit, je-li to možné. Naším cílem je navrhnout algoritmus tak, aby byl schopen zpracovat jak barevné video (pro účely srovnání s ostatními algoritmy), tak i monochromatické video (pro praktické nasazení).

Na základě těchto omezujících, či spíše (v tomto případě) definujících podmínek, musíme vybírat i vhodné algoritmy. Požadavky kladené na algoritmus tedy jsou:

- schopnost adaptovat se na libovolnou afinní transformaci (rotace, změna velikosti a pozice),
- schopnost pracovat pouze s jasovou informací (nelze tedy využít barevných textur),
- odolnost vůči překrytí a (malé) změně tvaru objektu,
- odolnost vůči jasovým změnám (stíny, osvětlení, apod.),
- rychlost (systém by měl ideálně pracovat v reálném čase, neboť je nutné zachovat plynulost, tedy limitující bude i výpočetní náročnost daného řešení).

## 2.2.2 Výběr bodů pro sledování

Výběr bodů vhodných pro sledování objektu je kritickou částí celého procesu. V případě chybného zvolení bodu v iniciálním snímku dochází k chybnému nalezení bodu v následujících snímcích. Existuje řada vlastností, které by body měly splňovat, aby byly vybrány jako vhodné pro účely sledování.

### Náhodný výběr bodů

Triviální metoda výběru bodů pro účely sledování je náhodný výběr. V inicializační fázi algoritmu se náhodně vybere množina bodů z oblasti náležející objektu, které se poté algoritmus snaží nalézt v jednotlivých po sobě jdoucích snímcích. Zřejmým problémem tohoto postupu je fakt, že vybrané body nemusí mít zrovna vhodné vlastnosti, které by bylo možné sledovat. I přesto se někdy tato metoda používá, především pro svoji rychlost.

### Histogramy

Definice histogramu říká, že histogram je grafické znázornění distribuce dat pomocí sloupcového grafu, přičemž výška sloupců vyjadřuje četnost sledované veličiny v daném intervalu a jeho šířka vyjadřuje šířku intervalů (tříd). Je důležité zvolit správnou šířku intervalu, neboť nesprávná šířka intervalu může snížit informační hodnotu diagramu [16].

Pro účely zpracování obrazu se používají buď barevné, nebo černobílé histogramy. Vycházíme z faktu, že naprostá většina obrázků má barevnou hloubku 8 bitů, každý pixel tedy lze zařadit do jedné z 256 tříd. V případě barevného obrazu se pak využívají histogramy tři – jeden pro každou barevnou složku RGB.

V aplikaci sledování objektů se histogramy využívají tak, že je definována oblast, která nás zajímá. Tato oblast je rozdělena na menší oblasti, přičemž pro každou z těchto malých oblastí je vytvořen vlastní histogram. Je zřejmé, že takováto množina histogramů vcelku jednoznačně reprezentuje vybranou oblast. Množina histogramů pak slouží jako vektor popisující hledaný objekt. Histogramy lze normalizovat nebo jinak upravovat a lze takto získat vcelku robustní příznak pro další postup algoritmu. V praxi se však ukazuje, že metody založené na histogramech, selhávají na černobílých snímcích.

### SURF

Metoda SURF (*Speeded-Up Robust Features*) je metoda, byla představena trojicí Herbert Bay, Tinne Tuytelaars a Luc Van Gool v roce 2006 [16]. Jedná se o novější obdobu metody SIFT (*Scale-Invariant Feature Transform*) z roku 1999 a jejímž autorem je David Lowe [18]. Obě metody slouží k nalezení zajímavých (významných) bodů v obraze. Obrázek popisují pomocí takzvaných deskriptorů vypočítaných pro tyto body. SURF využívá výpočtu determinantu z Hessovy matice (hessián, *DoH – Determinant of Hessian*), naproti tomu SIFT využívá výpočtu rozdílů Gaussiánů (*DoG – Difference of Gaussian*). Takto získané deskriptory jsou invariantní vůči rotaci a vzdálenosti kamery od popisovaného objektu. Algoritmus SURF se využívá v aplikacích počítačového vidění. Používá se pro rekonstrukci 2D a 3D scén, klasifikaci obrázků a rychlý popis obsahu obrázku. Míru podobnosti dvou obrázků lze měřit jako vzdálenost.

## Příznaky vhodné pro sledování

Příznaky vhodné pro sledování (tzv. *Good Features to Track*) nemají jiný specifický název a byly představeny v roce 1994 autory J. Shi a C. Tomasi [19]. Pro účely sledování objektu ve videu se nejlépe hodí body, které mají ve svém okolí nějakou texturu, nebo jsou rohy objektu.

### 2.2.3 Metoda odčítání pozadí

Metoda odčítání pozadí [22][23] je jedna z metod počítačového vidění pro segmentaci obrazu a především pak pro detekci pohybujících se objektů umístěných v popředí scény při statickém snímání obrazu. Uplatní se tedy především v situacích, kdy se kamera snímající scénu nepohybuje, neboť pohyb kamery by vnesl do snímku falešné detekce pohybu. Základní metoda vychází z toho, že máme k dispozici model pozadí scény (tedy snímek objektů, které do scény patří, a které nazýváme souhrnně pozadím), od kterého odčítáme aktuální snímek.

Kromě základní metody odčítání pozadí existují i různé varianty, které se snaží řešit některá problematická místa základní metody, jako je například malá odolnost vůči šumu. Jedna z takových metod modifikuje základní algoritmus tak, že nahrazuje prosté odčítání jasové hodnoty odčítáním hodnoty filtru. Tím může být například jednoduchý průměr nebo medián, jejichž hodnoty získáme z posloupnosti několika předchozích snímků. Tyto metody nutně potřebují určitý „startovací“ čas pro vytvoření dostatečné historie snímků, tedy pro vytvoření kvalitního modelu (tento čas však potřebuje i metoda základní), lze ho však v extrémním případě zkrátit na jediný snímek.

Největší slabinou této metody zůstává nutnost porovnání s prahovou hodnotou, která je zároveň globální a neměnná s časem. To působí velké problémy při změnách osvětlení, což vede k chybným detekcím pohybu. Ukázka metody odčítání pozadí je na obrázku 2.2, kde je znázorněno pozadí bez objektů, cizí objekt (automobil), výsledek odečtení pozadí a prahování.

### 2.2.4 Metoda založená na částicových filtrech

V několika předcházejících letech se stal velmi populárním přístup založený na různých modifikacích a variantách rekurzivního Bayesovského filtrování, nazývaného také částicové filtry (*particle filters*), někdy také označované jako sekvenční metody Monte Carlo (SMC). Částicové filtry poskytují robustní základ pro sledování objektů, který se neomezuje pouze na lineární dynamické systémy a Gaussovo rozložení šumu. Určitou formu částicového filtru použili poprvé na sledování objektů ve videu Isard a Blake a nazvali ho *Condensation* [32], což je zkratka z *Conditional Density Propagation* a v komunitě zaměřené na počítačové vidění je tento algoritmus typickým představitelem částicových filtrů. Tento algoritmus byl použit na sledování objektů v obrazech upravených na konturovou reprezentaci scény [33][34].

Konturové modely jsou relativně robustní vůči změnám osvětlení, ale mohou být výpočetně náročné a náchylné k chybám způsobeným členitostí pozadí scény [35]. Využití informace o barvě je další z mnoha velmi často využívaných technik. Mnohdy je využívána informace ve formě histogramu [31][36]. Částicové filtry využívající barevné informace v podobě histogramů jsou při sledování objektu velmi odolné vůči rigiditě pohybujících se objektů, částečnému zakrytí objektu a šumu. Nicméně tento způsob popisu je omezen tím, že ignoruje prostorovou polohu bodů, což znemožňuje rozeznání objektů s podobným barevným rozložením. Tento problém je pak znásoben v případě nutnosti práce s monochromatickým videem [37].

Jiným jednoduchým a velmi často používaným přístupem k popisu pohybujícího se objektu je využití vzoru objektu (*template*) – tedy malé části snímku, který přímo reprezentuje vzhled daného objektu. Výhodou tohoto přístupu je, že tento popis v sobě nese informaci o barvě i prostorovém vzhledu. Tento přístup je vhodný pro barevné i monochromatické modely a jeho výpočet bývá velmi rychlý. Na druhou stranu může

dojít (a dochází) k tomu, že vybraný vzor se stane nereprezentativní pro sledovaný objekt především z důvodu změny jeho vzhledu v důsledku pohybu (změna měřítka, rotace) a šumu. Aby se předešlo tomuto problému, využívá se například techniky pomalé aktualizace vzoru v průběhu času [38][39][40]. V průběhu aktualizace však hrozí nebezpečí, že vzor postupně začne reprezentovat zcela jiný objekt v důsledku nesprávného určení polohy. Velká pozornost musí být také věnována i (částečnému) zakrytí objektu, které může také vést k selhání algoritmu.

### Nelineární rekurzivní Bayesovské sledování

Nelineární stochastický systém v doméně diskretního času lze popsat následujícími způsobem [17].  
Rovnice

$$x_{n+1} = f(x_n, d_n), \quad (1)$$

se nazývá stavovou (nebo přechodovou) rovnicí a  $x_{n+1}$  popisuje stav systému v čase  $n + 1$ ,  $x_n$  popisuje stav systému v čase  $n$  a  $d_n$  lze považovat za náhodné sekvence v podobě bílého šumu s neznámou statistikou v diskretní časové doméně. Následující rovnice se nazývá rovnice měření (nebo pozorování), kde pozorování v čase  $n$  je označeno jako  $y_n$ ,  $x_n$  je stav systému ve stejném čase a  $v_n$  je, podobně jako u předchozí rovnice, šum:

$$y_n = h(x_n, v_n). \quad (2)$$

Za povšimnutí stojí fakt, že aktuální stav systému závisí pouze na předchozím stavu, což je situace, kterou lze popsat Markovskými modely, a označujeme ji jako Markovský proces. Za určitých okolností lze tyto rovnice redukovat na lineární Markovský model. Teorie zabývající se touto problematikou je velmi dobře popsána například v [17]. Cílem sledování objektu (filtrování) je tedy určení stavu  $x_n$ , který je skrytý, na základě předchozích pozorování  $y_{0:n} = (y_0, y_1, \dots, y_n)$ .

Pokud se na celý problém podíváme z Bayesovského úhlu pohledu, pak je problém sledování objektu vytvoření posteriorní hustoty pravděpodobnosti  $p(x_n | y_{0:n})$  aktuálního stavu  $x_n$ , za předpokladu, že je dána posloupnost pozorování od počátečního stavu 0 do času  $n$  a za dané počáteční hustoty  $p(x_0)$ , přechodové hustoty  $p(x_n | x_{n-1})$  a pravděpodobnosti  $p(y_n | x_n)$ . V principu lze tuto posteriorní hustotu vypočítat rekurzivně ve dvou krocích [17] výpočtem Chapman-Kolmogorovovy rovnice predikčního kroku, která slouží k určení předchozí hustoty  $p(x_n | y_{0:n-1})$ :

$$p(x_n | y_{0:n-1}) = \int p(x_n | x_{n-1}) p(x_{n-1} | y_{0:n-1}) dx_{n-1} \quad (3)$$

a rovnice aktualizací, která vychází z úměrnosti:

$$p(x_n | y_{0:n}) \propto p(y_n | x_n) p(x_n | y_{0:n-1}). \quad (4)$$

Rekurzivní vztah obou rovnic nelze obecně určit analyticky kvůli náročnosti, téměř nemožnosti, vyhodnocení obvykle komplikovaného integrálu (kromě speciálních případů, kdy jde o lineární a Gaussovský stavový prostorový model).

Rekurentní vztahy mezi fázemi vytváří optimální Bayesovské řešení. Abychom se vyhnuli integraci v rekurzivní Bayesovské rovnici, je posteriorní hustota aproximována množinou vzorků, které mají přiřazeny pravděpodobnostní váhy. Nicméně, posteriorní hustota může být nestandardní, a proto může být značně obtížné vygenerovat vzorky z posteriorní hustoty přímo. Abychom tomuto problému předešli, používá se další, tzv. navrhovaná hustota, z níž lze snadněji generovat vzorky, které jsou použity namísto hodnoty pravděpodobnosti posteriorní hustoty. Tento postup je označován jako *Importance Sampling*

[17][25]. Aposteriorní *pdf* tedy vyjádříme například náhodně generovanými váženými vzorky – *částicemi*. S rostoucím počtem částic se přibližujeme skutečnému aposteriornímu *pdf*.

### Bayesovský bootstrap filtr

Bayesovský bootstrap částicový filtr (převzorkovací filtr) je variantou filtru převzorkování na základě významnosti (*Sequential Importance Resampling – SIR*)[26], kde je hustota přechodu využita jako navrhovaná hustota, a kde je krok převzorkování prováděn při každé iteraci. Obecný algoritmus Bayesovského převzorkovacího filtru je složen z následujících kroků:

1. Predikce: každá částice  $\check{\chi}_{n-1}^i, i \in \{1, \dots, N\}$  je využita v modelu systému k výpočtu diskrétní aproximace hustoty pravděpodobnosti

$$\chi_n^i = f(\check{\chi}_{n-1}^i, d_{n-1}). \quad (5)$$

2. Aktualizace: je vyhodnocena pravděpodobnost všech pozorování, každé částici je přiřazena nenormalizovaná váha významnosti:

$$\varpi_n^i = p(y_n | \chi_n^i). \quad (6)$$

3. Normalizace vah:

$$\omega_n^i = \frac{\varpi_n^i}{\sum_1^N \varpi_n^i}. \quad (7)$$

4. Převzorkování: ze stávající množiny částic  $\{\chi_n^1, \dots, \chi_n^N\}$  je vybráno  $N$  částic  $\check{\chi}_n^i$  na základě jejich významnosti, tedy na základě jejich vah  $\omega_n^i$ .

Inicializace filtru vyžaduje  $N$  částic, které jsou vybrány ze známé hustoty pravděpodobnosti, tyto částice pokračují přímo krokem 2, jsou tedy aktualizovány přímo. Jako jednotlivé měření může být odhadovaný stav vypočítán jako střední hodnota nejvýznamnějších částic nebo může být reprezentován přímo nejvýznamnější částicí (tj. jako *Maximum a-posteriori – MAP*).

## 2.3 Shrnutí

V této kapitole jsme shrnuli teoretické poznatky, které mají vztah k problematice digitální stabilizace obrazu a sledování objektu. Kromě základních definic problému byly představeny metody pro výběr bodů vhodných pro sledování a metody sledování objektů. Vybrané metody jsme také implementovali a porovnávali jsme vlastnosti, které dané řešení má, především s přihlédnutím na požadavky aplikací v reálném světě.

Porovnání metod bylo založeno čistě na subjektivním hodnocení kvality výstupu, rychlosti a potřeb pro zvolenou úlohu. Zvážení výhod a nevýhod nás vedlo k tomu, že jako vítězný kandidát pro implementaci byl zvolen algoritmus založený na sledování částic, konkrétně pak varianta Bayesovský bootstrap filtr s tím, že budou provedeny specifické úpravy pro zlepšení vlastností tohoto algoritmu.

### 3 Návrh kamerového systému pro sledování objektů

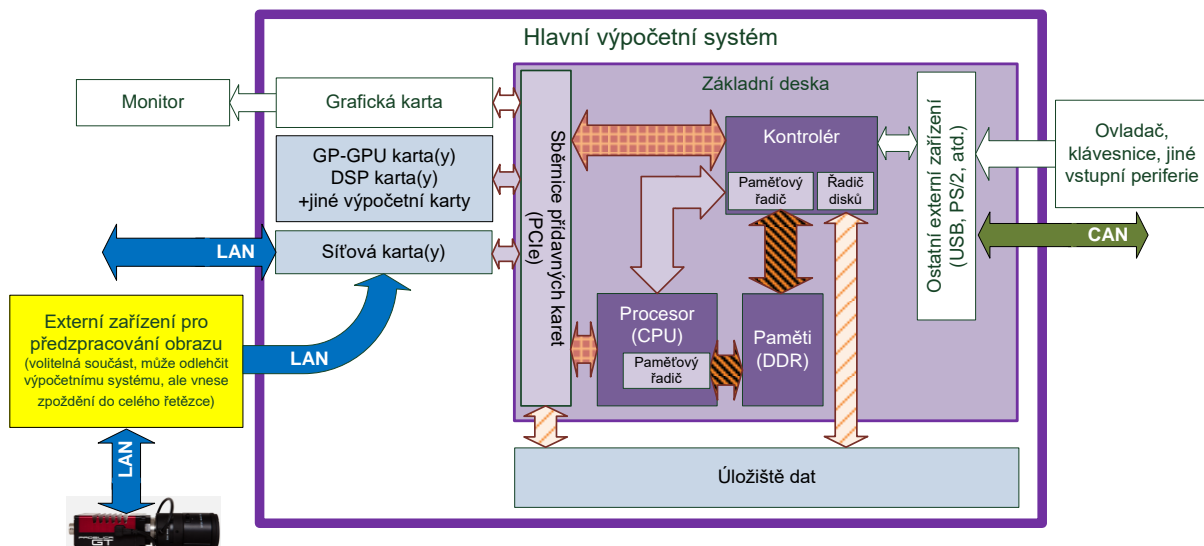
V následující je popsán návrh funkčního vzoru pro sledování objektů v reálném čase. Algoritmus, který je hlavním spotřebitelem výpočetního času, byl popsán v kapitole předchozí. Základem analýzy je rozbor potřeb uživatele, algoritmy vstupující do řetězce zpracování obrazu, ale i empiricky zjištěné nároky samotného algoritmu s tím, že přesné údaje zjistit nepotřebujeme – jde nám pouze o přibližné určení nároků na technické vybavení.

#### 3.1 Prvky technického vybavení kamerového systému

Na obrázku 3.1 je základní koncepční schéma funkčního vzoru kamerového systému. Systém se skládá z několika základních celků – kamerového systému (objektiv, kamera, řídicí elektronika), manipulátoru, a hlavního výpočetního systému. Limitujícím faktorem při výběru prvků systému bude především množství dat, které bude potřeba přenášet a zpracovávat (kamerami počínaje a hlavním výpočetním systémem konče). Celý systém je složen z následujících bloků:

- kamera, objektiv a řídicí elektronika: komunikace přes USB a po ethernetu,
- hlavní výpočetní systém: výkonný výpočetní systém schopný zpracovat všechna potřebná data z kamery,
- manipulátor: umožní rotaci ve dvou osách, což je důležité především z hlediska přenosu dat v místech rotace.

Jednotlivé bloky systému jsou propojeny s hlavním výpočetním systémem prostřednictvím rozhraní USB nebo sítí typu ethernet (systémy vyžadující vysokou datovou propustnost). Z hlediska kalkulace nároků na hardware je pro nás nejdůležitější kamera poskytující obrazová data. Datový tok z kamer přímo ovlivňuje návrh hlavního výpočetního systému.



Obrázek 3.1: Blokové schéma významných funkčních prvků hlavního výpočetního systému.

Rozebereme-li označený blok hlavního výpočetního systému podrobněji, můžeme definovat jednotlivá místa důležitá z hlediska výskytu úzkých hrdel systému a z hlediska výpočetního výkonu. Obrázek 3.1 shrnuje jednotlivé významné celky figurující v hlavním výpočetním systému zpracování obrazu. Jednotlivé důležité celky, které hrají v hlavním výpočetním systému roli, jsou následující:

- externí zařízení pro předzpracování obrazu, síťová karta, GP-GPU karta, datová sběrnice přídavných karet, kontrolér, paměti, procesor, úložiště dat.

## **Síťová karta**

Síťová karta je jednou ze stěžejních částí. Jelikož kamery pořizující záznam jsou připojeny přes síťové rozhraní, byl zvolen síťový port dostatečně kvalitní (použití specializované síťové karty s vlastním procesorem, který odlehčí zátěž hlavního procesoru).

## **GP-GPU akcelerátor**

Využití GP-GPU (*General Purpose – Graphics Processing Unit*, obecné výpočty prováděné na grafických kartách) je velice vhodné právě pro úkoly zaměřené na zpracování obrazu. Grafická karta ve svém procesoru zahrnuje velké množství takzvaných *stream procesorů*. Tyto procesory je možné programovat několika způsoby. Nejpoužívanější jsou přístupy využívající specifikaci firmy nVidia – CUDA a obecnější řešení je pak založeno na knihovně OpenCL. Velké množství malých procesorů v jádru čipu nabízí možnost masivní paralelizace úloh, což je v případě zpracování obrazu přesně ten přístup, který je vhodný. Nespornou výhodou zcela jistě je fakt, že takové řešení nevnáší do celého řetězce zpracování žádné další zpoždění.

## **Kontrolér**

Kontrolér má v moderních systémech za úkol propojit externí zařízení s procesorem. Běžně je v kontroléru integrován řadič pamětí, disků a dalších externích zařízení. Současným trendem je přesun z hlediska rychlosti kritických částí přímo do procesoru (především pak grafická karta a řadič paměti). Při bližším zkoumání trhu docházíme k závěru, že kontroléry již nemají integrovaný paměťový řadič a tento úkol se přesouvá směrem k procesoru. Kontrolér s paměťovým řadičem mohl zajistit různé paměťové přenosy bez zásahu procesoru, což v případě integrace řadiče do procesoru není zcela možné (ačkoliv je procesor koncipován tak, aby tyto režijní úkony byly časově co nejméně náročné). Úkolem kontroléru je zajistit přísun všech dat ze sběrnice směrem k paměťovému řadiči procesoru. Je tedy zřejmé, že nejkritičtější místem z tohoto hlediska bude zajištění dostatečné rychlosti kontroléru výběrem vhodné architektury.

## **Paměti**

Paměti jsou dalším důležitým prvkem ovlivňujícím celkový výkon hlavního výpočetního systému. Propustnost pamětí bude hrát velmi důležitou roli při návrhu a výběru hardware. V dnešní době jsou nejčastěji používány paměti typu DDR SDRAM (*Double Data Rate Synchronous Dynamic Random Access Memory*). Tyto paměti poskytují maximální teoretickou propustnost až 17 GB/s (aktuálně jsou takto rychlé paměti značně drahé a celý systém by prodražily a jejich rychlost je pouze teoretická, nehledě na to, že sběrnice procesoru není pro tento výkon stavěna).

## **Procesor**

Procesor je hlavním výpočetním centrem a bude mít na starost především výpočty podle algoritmů, ale i obsluhu operačního systému, komunikaci a spolupráci s uživatelem. Je zřejmé, že na procesor budou kladeny velké nároky. Moderní procesory jsou běžně vícejádrové, což je vlastnost, kterou budeme využívat. Prvotní úvahy nás vedou k závěru, že bude potřeba 1 jádro pro operační systém a 1 jádro pro zpracování sledování cíle (v kooperaci s dalšími výpočetními prostředky). Problémem, na který zřejmě narazíme, bude sdílení paměti. Sdílená paměť bude brzdit jednotlivé procesory při plnění úloh. V ideálním případě by řešení mělo obsahovat procesory dva, každý se svým paměťovým prostorem.

## **Úložiště dat**

V případě potřeby ukládání příchozího videa z kamery, bylo nutné zvolit dostatečně rychlé datové úložiště. Disky se běžně připojují přes rozhraní kontroléru, v současné době jde především o rozhraní typu SATA (*Serial AT Attachment*). Rozhraní SATA limituje rychlost připojení na teoretické 3 Gb/s, případně 6 Gb/s u novější varianty tohoto rozhraní.

## Ostatní zařízení

Mezi ostatní zařízení patří především periferie pro ovládání systému (například myš, dotykový panel, klávesnice, joystick), grafická karta pro výstup informace na monitor, který je dalším zařízením. Typicky nebudou tato zařízení mít přímý omezující vliv na datovou propustnost nebo vliv na výkon systému. Obsluha těchto zařízení bude probíhat v režii operačního systému a neměla by vyžadovat žádné dodatečné prvky. Uvedené algoritmy poběží na hlavním výpočetním systému s tím, že některé drobnější úlohy jsou řešeny přímo mikrokontroléry příslušných zařízení, případně dalším zařízením určeným pro daný účel.

## 3.2 Prvky kamerového systému

Cílem realizace kamerového systému je výzkum a vývoj technologie pro sledování (i několika) objektů, které se nacházejí ve sledované oblasti pokryté zorným polem kamerového systému. V základní verzi umožní tento systém automatické sledování objektů a přibližné určení jeho zeměpisné pozice v rámci možností, které jedno-kamerové řešení poskytuje. Systém usnadní operátorovi sledování označeného objektu tak, že se celý systém autonomně pokusí objekt sledovat a poskytovat tak prostor a čas pro další práci operátora.

### 3.2.1 Kamera

Výběr digitální kamery a objektivu je svázán s požadavky, které jsou na systém kladeny. Důležitá je především specifikace využití finálního produktu. Od toho se pak odvíjí výběr konkrétního zařízení (jiné je zařízení pro sledování místnosti ze vzdálenosti od 2 do 10 metrů a jiná situace je při sledování objektů ve vzdálenosti několika stovek metrů). To platí jak pro kameru, tak i pro objektiv. Jako nejvýznamnější byly určeny následující skupiny specifických parametrů rozhodujících o výběru kamery:

- rozlišovací schopnost,
- způsob/možnost řízení záznamu,
- způsob připojení,
- možnost přímého řízení objektivu,
- klimatické a provozní podmínky.

Vhodnými kandidáty budou kamery s rozlišením FullHD, tedy 1920×1080 obrazových bodů. Vybrány byly kamery od firmy *Allied Vision Technologies*, které téměř po všech stránkách splňují naše požadavky. Důležitými parametry zvolené kamery GX 1660 jsou například běžné ethernetové připojení, kamera komunikuje pomocí standardu GigE, rozlišení 1600 x 1200 bodů, senzor o velikosti 2/3" s velikostí pixelu 5.5 μm, otvor pro připojení objektivu typ C, obnovovací frekvence: až 63 snímků/s a 14 bitový A/D převodník.

Dalšími podmínkami z hlediska použití kamery je programátorské rozhraní. Výrobce kamery nabízí ovladače pro OS Microsoft Windows i Linux a kompletní SDK ke kameře, což je velmi přínosné z hlediska vývoje software. Kromě toho umí tato kamera i přímo ovládat motory objektivů s proměnnou ohniskovou vzdáleností. Tím tato kamera splňuje téměř maximálně všechny požadavky, které byly stanoveny.

### 3.2.2 Objektiv

Na objektiv budou z hlediska provozního kladeny podobné požadavky jako na kameru. Objektiv by měl umožňovat změnu ohniskové vzdálenosti (tedy zoom) a měnit tak zorné pole v rozmezí ca od 3° do 40°. Dalším požadavkem, který plyne především z výběru kamery, je schopnost pokrytí plochy širokoúhlého senzoru o rozměru 2/3". Jelikož je naším požadavkem relativně extrémní rozsah ohniskových vzdáleností, nezbyvá mnoho objektivů, které by splnily všechny podmínky. Kromě výše uvedených k nim ještě musíme



přidat možnost motorického řízení zoomu, ostření a clony (clonu lze řídit i elektronicky). Všechny požadavky lze shrnout takto:

- rozsah zorného pole ca od 3° do 40° – změnu této hodnoty lze akceptovat,
- připojení typu C (C-mount) – nelze zaměnit za jiný, pouze v případě existence redukce nebo změny výběru kamery,
- motoricky říditelný zoom a ostření,
- říditelná clona (lze brát v úvahu i automatické elektronické řízení),
- schopnost pokrýt celý senzor o rozměrech 2/3“.

Analýza trhu nám poskytne, při zvolených parametrech, pouze malé množství objektivů s tím, že mnohé z nich nesplní pouze některou z podmínek, ale i tak se diskvalifikují z výběru především proto, že některé podmínky nelze zanedbat (například závit a jeho rozměr je jasně stanoven). Vybrán byl objektiv Fujinon C22x17B-Y41, jehož parametry lze shrnout takto:

- ohnisková vzdálenost 17-374 mm, což při rozměru senzoru 2/3“ činí rozsah zorného pole od 1°21' do 29°,
- motoricky říditelný zoom a ostření (typ „motor drive“),
- zpětná vazba s informací o poloze zoomu,
- automatická clona řízená elektronicky video signálem,
- clonové číslo od F2.3 do ca F3000 (ekvivalent T3000),
- C-mount.

Pro tento model jsou v nabídce i extendery, které případně umožní zvýšit ohniskovou vzdálenost na úkor světelnosti objektivu, ale ty zřejmě nebudou pro naše účely potřeba. Navíc objektiv poskytuje i zpětnou vazbu ve formě informace o pozici zoomu a ostření.

### 3.2.3 Manipulátor

Manipulátorem, který naše požadavky na jedno-kamerový systém splnil, je zařízení od firmy EVPÚ Defence – Pan/Tilt MST-01/ST, což je stabilizovaný manipulátor, jehož parametry jsou následující:

- Řízení manipulátoru speciální konzolí operátora nebo prostřednictvím počítače přes ethernetové kabelové spojení.
- TTL vstupy a výstupy, sériová linka pro řízení senzorů.
- Vysoká úhlová přesnost, veliký rozsah úhlových rychlostí.
- Nízká hmotnost a malé rozměry, kompaktní konstrukce.

Standardní technické parametry:

- Rozsah pohybu v azimutu:  $n \times 360^\circ$ , rozsah pohybu v elevaci:  $\pm 60^\circ$
- Úhlová rychlost v azimutu: od  $0,03^\circ \cdot s^{-1}$  do  $120^\circ \cdot s^{-1}$ , úhlová rychlost v elevaci:  $0,03^\circ \cdot s^{-1}$  do  $100^\circ \cdot s^{-1}$
- Přesnost určení pozice: 0,1 mrad
- Řízení: Ethernet 100 Mbit, protokol TCP/IP, externí linky: 1x RS232; 1x RS422/485; 3x RS232/422
- Napájení: stejnosměrný proud, 24 V (20 - 30 V), zatížení: max. 40 kg, rozsah pracovních teplot: od  $-32^\circ C$  do  $+70^\circ C$

Technické parametry stabilizace:

- Max\_E méně než 1,4 mrad, Max\_A méně než 2,4 mrad
- RMS\_E méně než 0,6 mrad, RMS\_A méně než 1,2 mrad

### 3.2.4 Mechanická konstrukce jedno-kamerového systému

Nosná platforma byla navržena s ohledem na jednoduchost a účelnost konstrukce ve spolupráci s firmou Creinos, která hotový produkt také dodala. Požadavky byly:

- možnost montáže manipulátoru a kamerové hlavy,
- nosnost minimálně 30 kg (hmotnost manipulátor + kamerová hlava),
- výška maximálně 90 cm (spolu s manipulátorem by měla být sestava nižší než 150 cm),
- šířka podstavy maximálně 80 cm (je nutné pronášet platformu dveřmi a běžný rozměr dveří je zhruba 80 cm),
- rozměr skříně pro elektroniku minimálně 45 × 30 × 45 cm (skříň pro výpočetní systém rozměru ATX – základní deska má rozměr 30,5 x 24,4 cm, prostor na disky a zdroj, tedy půdorysně alespoň 45 × 45 cm s výškou alespoň 30 cm),
- větrací otvory a otvory pro kabely (minimálně 3 ks ethernetového kabelu CAT 5, 2 ks USB kabelu, 1 ks video kabelu HDMI nebo DVI, napájecí kabel – nutný je otvor průměru alespoň 2 cm),
- aretační nožky.

### 3.3 Elektronický systém kamerového systému

Elektronický systém kamerového systému se skládá z kontroléru kamery, ze zařízení pro čtení zpětnovazební informace zoomu a ostření, zařízení s GPS a z hlavního výpočetního systému. Další hardware, který se bude podílet na činnosti, tvoří vstupně-výstupní periferie v podobě klávesnice, dotekového ovládacího zařízení spojeného s monitorem pro dotekové ovládání systému a joystick nebo joypad.

#### 3.3.1 Elektronika kamerové hlavy

Elektronika kamerové hlavy je zastoupena především kamerou samotnou, která snímá video a řídí nastavení ohniskové vzdálenosti a ostření. Čtení zpětnovazební informace o poloze zoomu a ostření je potřeba vyřešit dodatečným zařízením, které je potřeba vyvinout.

Pro účely měření stavu potenciometrů objektivu byl použit vývojový kit STM32F4DISCOVERY s mikrokontrolérem STM32F407 MCU. Tento kit umožňuje relativně snadné zprovoznění úloh, jako je například porovnání napětí na výstupu stavového potenciometru vůči dodanému referenčnímu napětí, což je přesně úkol, který bylo potřeba vyřešit. Na desce kitu je osazen čip STM32F407VGT6, který umožňuje ladit aplikace nástrojem ST-LINK/V2, také obsahuje dva ST MEMS, digitální akcelerometr, digitální mikrofon, jeden DAC převodník, LED, tlačítka a USB OTG mikro-AB konektor pro připojení k externímu zařízení. Mezi hlavní vlastnosti procesoru STM32F407VGT6 patří:

- 32bitový mikrokontrolér s jádrem ARM Cortex-M4F, 1 MB paměti flash, 192 kB RAM,
- ST-LINK/V2 s možností nastavení pro účely ladění,
- Napájení přes USB sběrnici nebo z externího zdroje 5 V,
- Pohybový senzor, 3osý digitální akcelerometr,
- Zvukový snímač a všesměrový mikrofon,
- DAC převodník se zesilovačem třídy D.

Způsob připojení objektivu a kitu není příliš komplikovaný. Vývody objektivu poskytují vše potřebné pro měření stavu potenciometrů – vstup referenčního napětí  $V_{ref}$ , které nastavíme dle možností desky STM32F4, výstup poskytující napětí odpovídající poloze ostření a napětí odpovídající poloze přiblížení objektivu (zoomu). Programově zjistíme číselnou hodnotu v rozsahu 16 bitů úměrnou velikosti napětí.

### 3.3.2 Hlavní výpočetní systém

Hlavní výpočetní systém je uložen v prostoru podstavce manipulátoru. Volný prostor uvnitř krytu podstavce je dostatečný, není tedy nutné se příliš prostorově omezovat. Z hlediska návrhu a celkového řešení je nejjednodušší variantou centralizované řešení (*all-in-one*), kdy jeden systém bude obsluhovat ukládání a zpracování veškerých dat. Výhodou je především to, že vše bude obslouženo v rámci jednoho systému, náklady budou nižší, z hlediska programátorského bude vše snazší a celé řešení bude elegantnější (jedno zařízení na všechno).

#### **Základní deska**

Jde o základní hardware, který bude tvořit výpočetní systém. Jedním z parametrů desky je její velikost. Z hlediska vybavení nás bude zajímat především počet a typ ethernetových a USB přípojek, neboť není potřeba připojovat téměř jiná periferní zařízení. Vzhledem k tomu, že většina základních desek poskytuje maximálně 2 ethernetové konektory, bylo nutné vybavit systém i další síťovou kartou. Podstatná pro nás také je podpora vícejádrových procesorů, případně možnost osazení více procesory, množství osaditelné paměti, počet rozšiřujících slotů PCIe a počet konektorů pro pevné disky.

#### **Procesor**

Výpočetní výkon je soustředěn v procesoru a grafické kartě. Jelikož náročnost jednotlivých úkolů je vysoká, musí mít i procesor dostatečný výkon. Pro naše účely byl vybrán procesor Intel Core i7-3930K, který je tím nejlepším, co nabízí segment desktopových procesorů. Procesor Intel Core i7-3930K je určen pro stolní počítače, je 64bitový, nabízí možnost úpravy výkonu přetaktováním, se šesti jádry s podporou technologie Hyper-Threading a tedy dvanácti jádry virtuálními a stejným počtem najednou vykonávaných instrukcí. Každé z fyzických jader je taktováno na 3.2 GHz. Tento procesor podporuje technologii Intel Turbo Boost, která mu umožní v případě potřeby zvednout pracovní frekvenci na 3.8 GHz. Procesor je typu Sandy Bridge série E pro patičku Socket 2011. Procesor je vyráběn 32 nm technologií a je vybaven 4kanálovým paměťovým řadičem pro paměti typu DDR3, 12 MB vyrovnávací paměti třetí úrovně (L3 Cache) a 6x 256 kB vyrovnávací paměti druhé úrovně (L2 Cache). Procesor podporuje řadu různých rozšíření a technologií zaměřených především na multimédia (instrukční sady SSE, SSE2, SSE3, SSE4.1, SSE4.2 včetně AES), technologie Enhanced Speedstep, která je zodpovědná za vypínání jednotlivých částí procesoru a tím i za efektivnější provoz a chlazení. Maximální tepelný výkon udávaný výrobcem činí 130 W, což je údaj důležitý pro správné dimenzování zdroje a především chladiče.

#### **Chladič**

Procesor je potřeba chladit buď aktivně, nebo pasivně. Pasivním chladičem rozumíme chladič, který neobsahuje žádnou aktivní část (ventilátor), naopak aktivní tuto část obsahují. Pro vybraný procesor byl zvolen kvalitní chladič CoolerMaster Hyper 412 Slim se dvěma ventilátory, který je dostatečně dimenzován na tepelný výkon vybraného procesoru. Chladič je standardně osazen dvěma ventilátory o průměru 120 mm a výšce 15 mm s PWM regulací, které se otáčejí rychlostí od 500 do 1600 otáček za minutu.

#### **Paměť**

Ze základních úvah a propočtů je zřejmé, že systém bude muset poskytovat velké množství operační paměti, proto také byla vybrána základní deska, která to umožňuje (až 128 GB paměti). Pro procesory Core i7 je typické využití pamětí typu DDR3 s rychlostí oficiálně od 1066 do 1333 MHz. Z velkého množství dostupných variant byla zvolena paměť Kingston HyperX Red XMP, moduly o kapacitě 8 GB v počtu 8 kusů. Moduly Kingston KHX16C10B1RK2/16 jsou dvojicí paměťových modulů typu DDR3 určených pro pracovní frekvence až do 1600 MHz (odpovídá PC12800) a časování CL10 na své pracovní frekvenci při napětí 1,5 V. Paměťové moduly podporují funkci Intel XMP (*Extreme Memory Profiles*), kdy paměťové moduly obsahují

několik přednastavených časovacích profilů pro usnadnění provozu v různých situacích. Paměťové moduly jsou dodávány s hliníkovým chladičem, který odvádí teplo od paměťových čipů.

### **Síťová karta**

Dle specifikace požadavků je nutné dodat do systému další ethernetová rozhraní. Běžně jsou na základních deskách dostupné 1-2 konektory RJ-45 pro připojení k síti ethernet kabelem UTP. Je tedy nutné systém doplnit o další síťovou kartu. Vhodná síťová karta musí co nejméně zatěžovat procesor a musí mít nízkou latenci. Takové podmínky splňují například serverové síťové karty od firmy Intel. Zvolena byla karta Intel PRO/1000 PT Quadro Port Server Adapter, která využívá sběrnice PCI Express k zajištění vysokorychlostního přenosu dat přes gigabitové ethernetové spoje. Dedikovaná propustnost je díky zapojení do portu PCI Express 4x zajištěna a plný výkon lze garantovat na každém ze čtyř portů bez sdílení sběrnice.

### **Úložiště dat**

Pro účely ukládání dat je potřeba vybrat diskové jednotky. Díky vývoji v oblasti SSD (*Solid State Drive*) se ukazuje jako velmi dobrá možnost použít tento typ disku pro ukládání dat. Pro dlouhodobé ukládání dat je vhodné vytvořit buď diskové pole, nebo pořídit pomalejší velkokapacitní disk, jehož pořizovací cena je dnes vcelku nízká. SSD je mnohonásobně rychlejší při čtení i zápisu než klasické HDD a neobsahuje pohyblivé části, čímž se snižuje možnost poškození. Nevýhodou je vyšší pořizovací cena, celkově nižší kapacita a zatím nejasná životnost dat.

### **Grafická karta**

Pro zvýšení výpočetního výkonu u dobře paralelizovaných úloh využijeme moderních grafických karet. Toto řešení se jeví jako schůdnější varianta než použití například doplňující desky s DSP, proto se zaměříme i na grafické akcelerátory. Důležitým kritériem je také způsob programování těchto karet a způsob využití jejich výkonu. Karty nVidia lze programovat prostřednictvím CUDA, což je prostředí navržené právě pro tyto karty, které nám poskytuje programátorské rozhraní pro tvorbu aplikací využívajících masivně paralelního výkonu grafických karet. Z hlediska programátorského se jako příjemnější jeví programování v prostředí CUDA, neboť je jednodušší a provozní režie je nižší než u konkurenčních řešení, což je zřejmě dáno i tím, že ostatní řešení jsou univerzální, kdežto CUDA je cílená na konkrétní hardware od jediné firmy. Konkrétním vybraným modelem byla karta GIGABYTE GTX 670 Ultra Durable OC 2GB, která splňovala veškeré naše požadavky.

### **Napájecí zdroj**

Napájecí zdroj musí být schopen zásobovat celý hlavní výpočetní systém energií. Pro běžné výpočetní systémy se typicky využívají zdroje typu ATX, které transformují střídavý proud na stejnosměrný proud s tím, že poskytují několik různých úrovní napětí (typicky  $\pm 12$  V,  $\pm 5$  V a  $\pm 3,3$  V) s různou vydatností (dle kvality zdroje až v desítkách ampér především na větvi +12 V). Pro náš systém vychází odhad spotřeby na minimum 400 W a optimum ca 450 W. Počítáme-li s dalšími disky na archivaci videa a dalšími dvěma grafickými kartami, dosáhne optimální hodnota výkonu zdroje hodnoty 800 W. Abychom předešli havárii zdroje a zajistili dostatečnou rezervu výkonu, přikloníme se ke zdroji schopnému dodat výkon až 1200 W, konkrétně pak byl jako nejvhodnější vybrán zdroj Fortron Aurum Pro 1200 W, 80+ Gold (tedy účinný zdroj s dostatečnou rezervou), který je navíc modulární, řeší tedy problém nadbytečné kabeláže v prostoru výpočetního systému.

### **Interakce s uživatelem**

Důležitou součástí systému je interakce s uživatelem. Pro komunikaci s uživatelem slouží jako hlavní prostředek vstupní dotykové a zobrazovací zařízení v podobě dotykového displeje, který zároveň slouží

jako vstup i výstup. Další formou interakce s uživatelem je klávesnice a ovladač s joysticky a tlačítka – joypad.

Výběr řešení s dotykovým ovládáním vychází z předchozích zkušeností ovládáním software pro sledování – přináší komfort uživateli a patří dnes mezi dostupná řešení. Díky rozvoji technologií, ale i poptávky a nabídky v tomto segmentu (i díky masivnímu rozšíření tabletů) je možné vybírat z různých variant řešení. Abychom zpříjemnili uživatelskou zkušenost se systémem, rozhodli jsme se pro dotekový monitor s dostatečným rozlišením a úhlopříčkou. Našimi požadavky byly rozlišení alespoň 1920 × 1080 bodů a úhlopříčka větší než 20". Vybrán byl monitor firmy Iiyama T2250MTS.

Alternativním způsobem ovládání, a pro určité situace i nezbytným, je vstup z klávesnice, který umožňuje specifikovat detailně některé informace (například popis vzniklé situace, popis cíle, apod.). Klávesnice je běžná USB nebo PS/2, zde není potřeba vybírat nějaké specializované zařízení.

Joypad je dalším způsobem ovládání, které je zaměřeno na přímé řízení systému. Joypady se stávají velice populárním v mnoha oblastech a i ve vojenské sféře se více a více uplatňují tyto dříve spíše herní prvky. Cílem je poskytnout uživateli systému možnost pohodlného ovládání, neboť uživatelský komfort je také důležitým kritériem kvality celého řešení. Joypad je zařízení, které se skládá typicky ze dvou joysticků, křížového tlačítkového ovladače a několika tlačítek (typicky 8-12) a lze je hrubě rozdělit na běžné a specializované. Mezi běžné patří dostupné herní ovladače, které pro naše účely plně dostačují, ale lze vybrat i specializované, které mají robustní konstrukci, řádově vyšší pořizovací cenu a velmi často jsou ve formě zařízení, která je nutné zabudovat (tedy jakési polotovary).

### 3.4 Shrnutí

Výsledným řešením pro kamerový systém, k němuž jsme se přiklonili, je tedy centralizovaný výpočetní systém bez dalších vložených podsystémů. Celý kamerový systém se pak skládá z navržené a vyrobené nosné platformy, manipulátoru, kamerové hlavy, hlavního výpočetního systému, další drobné elektroniky a vstupně výstupních periférií.

Výpočetní hardware se opírá o základní desku MSI s novější patičí LGA2011 pro procesory Intel. Operační paměť má kapacitu 64 GB, což umožní ukládat naprostou většinu dat do paměti nebo využít paměťový disk. Pro akceleraci a paralelní výpočty byla zvolena výkonná grafická karta s 1344 procesory s čipem nVidia GTX 670, procesor byl zvolen z řady Intel Core i7 39xx, jako základní deska byla zvolena MSI X79A-GD65. Jako úložiště poslouží běžný SSD disk Corsair Force GS 360 GB s kapacitou 360 GB. Cena výsledné sestavy v oblasti kolem 50 000,- Kč bez DPH je velmi příznivá pro takto výkonný systém.

Elektronický systém kamerového systému se skládá z několika prvků – elektroniky kamerové hlavy v podobě kamery samotné a zařízení pro kontrolu stavu zoomu a ostření, hlavního výpočetního systému, který je srdcem celého zařízení a slouží ke komunikaci s uživatelem a k provádění výpočtů, a vstupně-výstupních ovládacích prvků.

Manipulátor a kamerová hlava jsou komerčními produkty, které jsme upravili opět pro potřeby kamerového systému tak, aby splňovaly naše požadavky. Zvolen byl manipulátor od firmy EVPÚ Defence MST-01, který v podstatě téměř beze zbytku splnil naše očekávání a kamerová hlava, kterou jsme upravili pro naše potřeby integrací elektroniky pro řízení zoomu a ostření a čtení zpětnovazebních informací. Kamerová hlava je odolná konstrukce a zahrnuje v sobě kameru Allied Vision Prosilica GX1660, objektiv Fujinon C22x17B-Y41 a zařízení pro čtení stavu zoomu a ostření objektivu v podobě desky STM32F4DISCOVERY. Hlavní výpočetní systém je složen z komerčně dostupných prvků, jejichž výběr je založen na teoretických výpočtech, které stanovují výpočetní nároky hardware pro kamerový systém. Jako vstupní zařízení slouží běžná klávesnice, ovládací zařízení joypad a doteková plocha zobrazovacího zařízení.

Výstupem (i dotykovým vstupem) je pak LCD panel Ilyama T2250MTS. Na obrázku 3.14 je snímek celého zařízení.



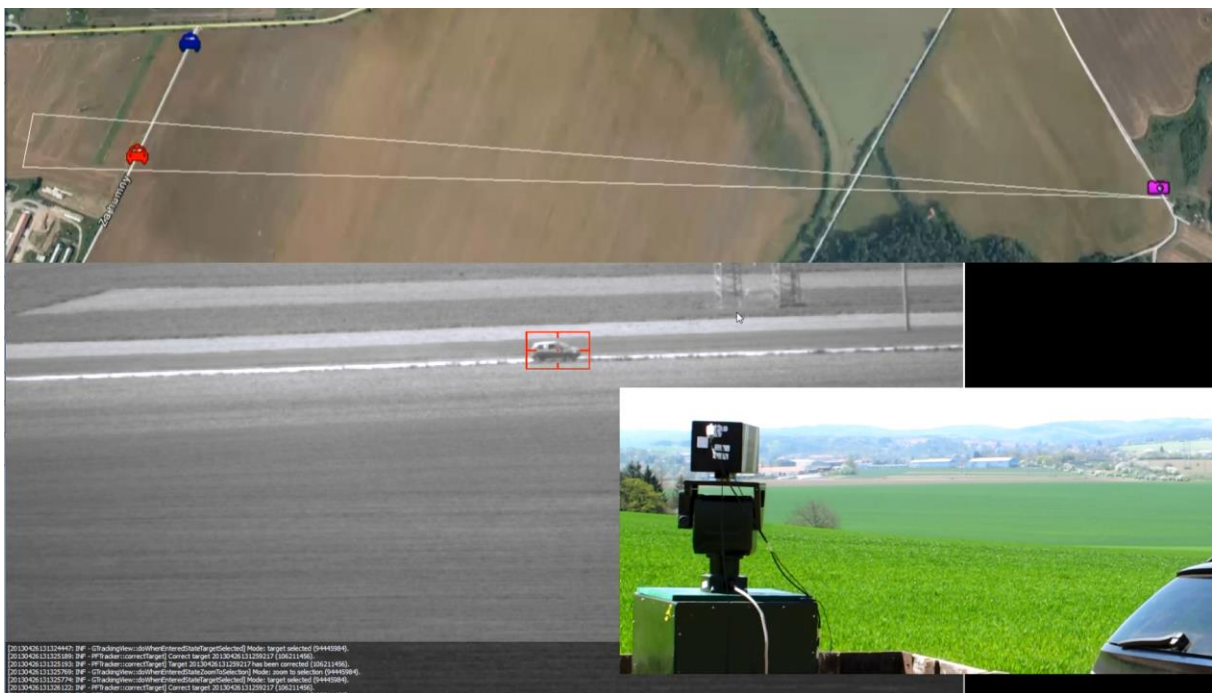
Obrázek 3.2: Snímek funkčního vzoru kamerového systému.



## 4 Závěr

Tato práce shrnuje poznatky, které mají vztah k problematice digitální stabilizace obrazu a sledování objektu především pro účely bezpečnostních aplikací. Vybrané metody byly implementovány a porovnány jejich vlastnosti především s přihlédnutím na požadavky aplikací v reálném světě. Porovnání metod bylo založeno především na subjektivním hodnocení kvality výstupu, rychlosti a potřeb pro zvolenou úlohu. Zvážení výhod a nevýhod nás vedlo k tomu, že mezi mnoha dostupnými řešeními jsme pro účely sledování objektu zvolili algoritmus založený na sledování částic, konkrétně pak varianta Bayesovský bootstrap filtr s tím, že dále byly provedeny optimalizace algoritmu tak, aby splňoval požadavky, které jsou stanoveny v úvodní kapitole.

Původní algoritmus byl do detailu rozebrán a implementován. Na základě znalostí z oblasti paralelizace výpočtů na grafických akcelerátorech, byly provedeny úpravy vedoucí ke značnému zrychlení algoritmu. Kromě toho jsme také navrhli mnohá vylepšení, která měla zamezit nežádoucímu chování algoritmu v kritických situacích. Silnými stránkami algoritmu je především odolnost vůči částečnému zakrytí (ale i úplnému zmizení) objektu, odolnost vůči pohybu kamery během sledování a schopnost práce s monochromatickým (ale i barevným) obrazem. Naproti tomu slabými stránkami jsou například problém ztráty sledovaného objektu v případě míjení dvou objektů, sledování nevýrazného objektu, v případě tvarové změny objektu nebo chybně označeného objektu.



Obrázek 4.1: Záběr z prezentačního videa.

Další částí práce je výroba hardware systému pro automatické sledování objektů. Byl navržen systém skládající se z nosné platformy, manipulátoru, kamerové hlavy a drobné elektroniky. Elektronický systém kamerového systému se skládá z elektroniky kamerové hlavy v podobě kamery samotné a zařízení pro kontrolu stavu zoomu a ostření, hlavního výpočetního systému, který je srdcem celého zařízení a slouží ke komunikaci s uživatelem a k provádění výpočtů, a vstupně-výstupních ovládacích prvků.

Nosná platforma je řešena jednoduše, nicméně jde o výstup dostačující a plní svou funkci – tedy nese manipulátor, kamerovou hlavu a je v ní integrován celý elektronický subsystém včetně hlavního výpočetního systému. Byla navržena ve spolupráci s firmou Creinos, která taktéž zajistila výrobu platformy.

Manipulátor a kamerová hlava jsou komerčními produkty, které jsme upravili opět pro potřeby kamerového systému tak, aby splňovaly naše požadavky. Zvolen byl manipulátor od firmy EVPÚ Defence MST-01, který splnil naše očekávání a kamerová hlava, kterou jsme upravili pro naše potřeby integrací elektroniky pro řízení zoomu a ostření a čtení zpětnovazebních informací. Kamerová hlava je odolné konstrukce a zahrnuje v sobě kameru Allied Vision Prosilica GX1660, objektiv Fujinon C22x17B-Y41 a zařízení pro čtení stavu zoomu a ostření objektivu v podobě desky STM32F4DISCOVERY. Hlavní výpočetní systém je složen z komerčně dostupných prvků, jejichž výběr je založen na teoretických výpočtech, které stanovují výpočetní nároky hardware pro kamerový systém. Jako vstupní zařízení slouží běžná klávesnice, ovládací zařízení joypad a doteková plocha zobrazovacího zařízení. Výstupem (i dotykovým vstupem) je pak LCD panel.

Takto vytvořený funkční vzor kamerového kompletu naplňuje naše představy o pasivním, kamerovém sledovacím systému. Je zřejmé, že je mnoho míst, která lze na stávajícím návrhu vylepšit, a předpokládáme, že v budoucnu budeme toto zařízení dále rozvíjet a využívat. Na obrázku 4. 1 je snímek z prezentačního videa, kde jsou vidět záběry z hlavní kamery, přehledové kamery snímající celou scénu (je vidět manipulátor s kamerovou hlavou na vozíku) a mapa se zakreslením situace. Celý komplet umožňuje, kromě sledování objektu, také přibližně určit jeho polohu, jak je také vidět na obrázku.

Experimenty ukazují, že se algoritmus chová dle našich požadavků ve většině situací, před něž byl postaven. Pracuje dobře na monochromatických záběrech, při pohybu velmi dobře stabilizuje snímky z kamery a ke „ztrátě“ sledovaného objektu dochází v předem očekávaných situacích. Na práci lze navázat především v oblasti zlepšení chování algoritmu v situacích, které byly identifikovány jako problematické.



## 5 Literatura

- [1] Automated Precision Inc. [online, cit. 2013]. Prezentace firmy a jejich produktů. Dostupné z <<http://www.apisensor.com/>>.
- [2] iMAR Navigation & Control [online, cit. 2013]. Prezentace firmy a jejich produktů. Dostupné z <<http://www.imar.de/>>.
- [3] IMAGO Video Trackers [online, cit. 2013]. Prezentace firmy a jejich produktů. Dostupné z <<http://www.imagotrackers.com/>>.
- [4] Electro-Optical Imaging, Inc. [online, cit. 2013]. Prezentace firmy a jejich produktů. Dostupné z <<http://www.eoimaging.com/>>.
- [5] Moog Inc. [online, cit. 2013]. Prezentace firmy a jejich produktů. Dostupné z <<http://www.moog.com/>>.
- [6] Yilmaz, A., Javed, O., Shah, M. Object Tracking: A Survey. *ACM Computing Surveys*, svazek 38, č. 4, článek 13, 12/2006.
- [7] Olšák, P. Úvod do algebry, zejména lineární. Praha: Skriptum FEL ČVUT, 2007.
- [8] Jan, J. Číslíková filtrace, analýza a restaurace signálů. 1. vydání. Brno: VUT v Brně, 1997. ISBN: 80-214-0816-2.
- [9] Jalal, A.S. et al. The State-of-the-Art in Visual Object Tracking [online, cit. 2012]. Publikováno 09-2012. *Informatica*, svazek 36, číslo 3. ISSN: 1854-3871. Dostupné z <[http://www.informatica.si/PDF/Informatica\\_2012\\_3.pdf](http://www.informatica.si/PDF/Informatica_2012_3.pdf)>.
- [10] Beneda, M. Homografie a epipolární geometrie [online, cit. 2012], publikováno 31-10-2010. In: *Trilobot*. Zlín: Univerzita Tomáše Bati ve Zlíně, číslo 2, 2010. ISSN: 1804-1795. Dostupné z <<http://trilobot.fai.utb.cz/homografie-a-epipolarni-geometrie>>.
- [11] Dubrofsky, E. Homography Estimation. Diplomová práce. Vancouver: Univerzita Britské Kolumbie, 2009.
- [12] Brooks, A. C. Real-Time Digital Image Stabilization, EE 420 Image Processing Computer Project Final Paper, EED Northwestern University, USA, March 2003, p. 10.
- [13] Ko, S. J., Lee, S. H., Jeon, S. W. Fast Digital Image Stabilizer Based on Gray-Coded Bit-Plane Matching, IEEE, USA, 1999, pp. 90-91, ISBN 0-7803-5123-1.
- [14] Vella, F., Castorina, A., Mancuso, M., Messina, G. Robust Digital Image Stabilization Algorithm Using Block Motion Vectors, IEEE, USA, 2002, pp. 234-235, ISBN 0-7803-7300-6.
- [15] Chang, J. Y., Hu, W. F., Cheng, M. H., Chang, B. S. Digital image translational and rotational motion stabilization using optical flow technique. In: *IEEE Transactions on Consumer Electronics*, svazek 48, číslo 1, s. 108-115, únor 2002. ISSN 0098-3063. doi: 10.1109/TCE.2002.1010098.
- [16] Kupka, K. Statistické řízení jakosti: interaktivní analýza a interpretace dat pro řízení jakosti a ekonomiku. Pardubice : TriloByte, 1997. ISBN: 80-238-1818-X.
- [17] Bay, H., Tuytelaars, T. a Gool, L. SURF: Speeded Up Robust Features. In: *Computer Vision – ECCV 2006*. Berlín: LNCS Springer, s. 404-417, 2006. doi 10.1007/11744023\_32. ISBN 978-3-540-33832-1.
- [18] Brown, M. a Lowe, D. Invariant Features from Interest Point Groups. In: *British Machine Vision Conference 2002*. Cardiff, Wales. 09-2002.
- [19] Shi, J., Tomasi, C. Good Features to Track. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, s. 593-600, 06-1994. ISBN: 0-8186-5825-8.

- [20] DeGroot, M.H., Schervish, M.J. *Probability and Statistics*. Pearson, 4. edice, 1. 1. 2011, s. 912. ISBN: 978-0321709707.
- [21] Hartigan, J.A. *Clustering Algorithms*. New York: John Wiley & Sons, Inc., 1975, s. 351. ISBN: 978-0-471-35645-5.
- [22] Stauffer, C. „Adaptive background mixture models for real-time tracking“. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Fort Collins, CO, USA, svazek 2, 1999. ISBN: 0-7695-0149-4.
- [23] Piccardi, M. „Background subtraction techniques: a review“. In: *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*. Sydney, Austrálie, svazek 4, s. 3099-3104, 2004. ISBN: 0-7803-8566-7.
- [24] Chen, Z.: *Bayesian Filtering: From Kalman Filters to Particle Filters, and Beyond*. First Edition, O'Reilly Media, Sebastopol, CA, 2008, s. 415-453, ISBN: 978-0-596-51613-0.
- [25] Rubinstein, R.Y., Dirk, P. K., *Simulation and the Monte Carlo Method*, Wiley & Sons, New York, 2008, ISBN: 978-0-470-17794-5.
- [26] Gordon, N., J., Salmond, D., J., Smith, A., F., M., *Novel approach to nonlinear/non-Gaussian Bayesian state estimation*, Radar and Signal Processing, IEE Proceedings F In Radar and Signal Processing, 1993
- [27] Shi, J., Tomasi, C., *Good Features to Track*, In: Computer Vision and Pattern Recognition, 1994, ISBN:0-8186-5825-8
- [28] Lucas, B. D., Kanade, T. „An iterative image registration technique with an application to stereo vision“. In: *Proceedings of the 7th International Joint Conference on Artificial Intelligence (IJCAI '81)*, s. 674-679.
- [29] Bouguet, J., Y., *Pyramidal Implementation of the Lucas Kanade Feature Tracker*, Intel Corporation, Microprocessor Research Labs, 2000.
- [30] Craciunescu, T. et al. Application of optical flow method for imaging diagnostic in JET. In: *Journal of Nuclear Materials*. Elsevier, svazek 400, vydání 3, 2010, s. 205-212. ISSN: 0022-3115.
- [31] Nummiaro, K., Koller-Meier, E., Gool, V., L., *An Adaptive Color-Based Particle Filter*, 2002
- [32] Isard, M., Blake, A., *Condensation – Conditional Density Propagation for Visual Tracking*, International Journal of Computer Vision, 1997
- [33] Isard, M., Blake, A., *Contour Tracking by Stochastic Propagation of Conditional Density*, European Conference on Computer Vision, 1996
- [34] Heap, T., Hogg, D. *Wormholes in Shape Spaces: Tracking through Discontinuous Changes in Shape*, International Conference on Computer Vision, 1998
- [35] Luo, X., Huang, Y. *Visual Tracking With Singular Value Particle Filter*, International Workshop on Machine Learning for Signal Processing, 2010
- [36] Pérez, P., Hue, C., Vermaak, J., Gangnet, M., *Color-Based Probabilistic Tracking*, ECCV 2002
- [37] Khalid, S., M., Ilyas, U., M., Sarfaraz, S., M., Ajaz, A., M. *Bhattacharyya Coefficient in Correlation of Gray-Scale Objects*, Journal Of Multimedia, 2006
- [38] Nguyen, H., T., Worring, M., Boomgaard, R., *Occlusion Robust Adaptive Template Tracking*, 2001
- [39] Matthews, I., Ishikawa, T., Baker, S. *The Template Update Problem*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 2003
- [40] Jurie, F., Dhome, M. *Real Time Robust Template Matching*, BMVC, 2002

- [41] Mei, X., Zhou, K, S., Porikli, F., *Probabilistic Visual Tracking Via Robust Template Matching and Incremental Subspace Update*, 2008
- [42] Barrera, P., Canas, J., M., Matellan, V. *Visual object tracking in 3D with color based particle filter*, 2005.
- [43] Sobral, A.C. *BGSLibrary: A OpenCV C++ Background Subtraction Library* [online, cit. 2012]. Software, 2012, dostupné z <<http://code.google.com/p/bgslibrary/>>
- [44] Y. Benezeth, Y., Jodoin, P.M., Emile, B., Laurent, H., Rosenberger, C. „Review and Evaluation of Commonly-Implemented Background Subtraction Algorithms“. In: *Proceedings of the 19th International Conference on Pattern Recognition (ICPR 2008)*, 8-11. 12. 2008, Tampa, Florida, USA. IEEE 2008, 2008, ISBN: 978-1-4244-2175-6.
- [45] Collins, R.T., Zhou, X., Teh, S.K. „An Open Source Tracking Testbed and Evaluation Web Site“. In: *IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS 2005)*, leden, 2005.
- [46] Collins, R.T. et al. *VIVID Tracking Evaluation Web Site* [online, cit. 2012]. Software Dostupné z <<http://vision.cse.psu.edu/data/vividEval/software.html>>
- [47] Corradi, A., Bellavista, P., Giannelli, C.: *Mobility and Handover Prediction mechanism: a performance comparison exploiting several filters* [online, cit. 2012]. Publikováno 04-05-2010. Dostupné z <<http://lia.deis.unibo.it/Research/SOMA/MobilityPrediction/filters.shtml>>
- [48] Arulampalam, S.; Maskell, S.; Gordon, N.; et al. „A Tutorial on Particle Filters for On-line Non-linear/Non-Gaussian Bayesian Tracking“. In: *IEEE Transactions on Signal Processing*, ročník 50, č. 2, 2002: s. 174–188, ISSN 1053-587 X.
- [49] Bernardin, K.; Stiefelhagen, R.: Evaluating multiple object tracking performance: the CLEAR MOT metrics. *J. Image Video Process.*, 2008: s. 1–10, ISSN 1687-5176.
- [50] Bradski, G.; Kaehler, A.: *Learning OpenCV*. O’Reilly Media, 2008, ISBN 978-0-596-51613-0.
- [51] Isard, M.; Blake, A.: CONDENSATION – Conditional Density Propagation for Visual Tracking. *International Journal of Computer Vision*, ročník 29, č. 1, 1998: s. 5 –28, ISSN 0920-5691.
- [52] van der Merwe, R., de Freitas, N., Doucet, A., a další: „The Unscented Particle Filter“. In: *Advances in Neural Information Processing Systems 13*, 2001.
- [53] Trucco, E., Plakas, K.: „Video Tracking: A Concise Survey“. In: *IEEE Journal of Oceanic Engineering*, ročník 31, č. 2, 2006: s. 520–529, ISSN 0364-9059.
- [54] Welch, G., Bishop, G.: *An Introduction to the Kalman Filter*. Technická zpráva, Chapel Hill, NC, USA, 1995.
- [55] Fischler, M. A., Bolles, R. C. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. In: *Comm. of the ACM* 24 (6): s. 381–395, červen 1981. doi:10.1145/358669.358692.
- [56] Digia: *oficiální stránky knihovny Qt* [online, cit. 2012]. Dostupné z <<http://qt.digia.com>>.
- [57] *Welcome - OpenCV Wiki*. 2010-04-06 [cit. 2011-10-11]. Dostupné z <<http://opencv.willowgarage.com/wiki>>.
- [58] *opencvx - Project Hosting on Google Code*. [cit. 2011-10-11]. Dostupné z <<http://code.google.com/p/opencvx>>.
- [59] Munshi, A.: *The OpenCL Specification* [online]. Publikováno 6. 1. 2011 [cit. 2012]. Technická specifikace, verze 1.1, revize 44, Khronos OpenCL Working Group. Dostupné z <<http://www.khronos.org/registry/cl/>>.

- [60] nVidia Corporation: *CUDA C Programming Guide* [online]. Publikováno 10-2012 [cit. 2012], nVidia. Dostupné z <[http://docs.nvidia.com/cuda/pdf/CUDA\\_C\\_Programming\\_Guide.pdf](http://docs.nvidia.com/cuda/pdf/CUDA_C_Programming_Guide.pdf)>.
- [61] Triolet, D.: *Nvidia Fermi: the GPU Computing revolution* [online]. Publikováno 09-10-2009 [cit. 2012]. Dostupné z <<http://www.behardware.com/articles/772-6/nvidia-fermi-the-gpu-computing-revolution.html>>.
- [62] Joseph, M.: *Real-Time Systems Specification, Verification and Analysis*. London: Prentice Hall, 1996. ISBN 0-13-455297-0.
- [63] Šumbera, P.: *Porovnávací test operačních systémů reálného času* [online, cit. 2010]. Publikováno 2000. In: *Automa*. Dostupné z <[http://www.odbornecasopisy.cz/index.php?id\\_document=27836](http://www.odbornecasopisy.cz/index.php?id_document=27836)>.
- [64] Gallaughier, J.: *Information Systems: A Manager's Guide to Harnessing Technology, v. 1.3*. Flat World Knowledge, 2011, s. 352. ISBN 978-1-4533-3272-6.
- [65] Silberschatz, A., Galvin, P. B.: *Operating System Concepts*. Hoboken, NJ: John Wiley & Sons Inc., 2008, s. 992. ISBN 978-0470128725.
- [66] Doepfner, T. W.: *Operating Systems In Depth: Design and Programming*. John Wiley & Sons, 2010, s. 444, ISBN 978-0471687238.
- [67] *Wireless Gigabit Alliance* [online, cit. 2012]. Dostupné z <<http://wirelessgigabitalliance.org/>>.
- [68] Wikipedie: IEEE802.11ac [online, cit. 2012]. Dostupné z <[http://en.wikipedia.org/wiki/IEEE\\_802.11ac](http://en.wikipedia.org/wiki/IEEE_802.11ac)>.
- [69] Cisco: 802.11ac: *The Fifth Generation of Wi-Fi*. Technická zpráva. 2012, s. 25.
- [70] Wikipedie: *Quadrature Amplitude Modulation* [online, cit. 2012]. Dostupné z <[http://en.wikipedia.org/wiki/Quadrature\\_amplitude\\_modulation](http://en.wikipedia.org/wiki/Quadrature_amplitude_modulation)>.
- [71] Broadcom: Broadcom Launches First Gigabit Speed 802.11ac Chips - Opens 2012 CES with 5th Generation (5G) Wi-Fi Breakthrough [online, cit. 2012]. Dostupné z <<http://www.broadcom.com/press/release.php?id=s637241>>
- [72] Románek, D.: *Co je CANopen a jak na něj* [online]. Vystaveno 20. 3. 2006 [cit. 6. 3. 2011]. Dostupné z <<http://hw.cz/Produkty/Obecne-produkty/ART1563-Co-je-CANopen-a-jak-na-naj.html>>.
- [73] ST Microelectronics: *STM32F4DISCOVERY: Discovery kit for STM32 F4 series - with STM32F407 MCU Active* [online]. Technická dokumentace. Dostupné z <<http://www.st.com/internet/evalboard/product/252419.jsp>>.
- [74] Microchip: *MCP2551. Microchip* [online]. [cit. 2012-11-27]. Technická dokumentace. Dostupné z: <<http://www.microchip.com/wwwproducts/Devices.aspx?dDocName=en010405>>
- [75] *Sparkfun level shifter* [online]. Dostupné z <<http://www.sparkfun.com/datasheets/BreakoutBoards/Level-Converter-v10.pdf>>
- [76] *FreeRTOS* [online]. Oficiální internetová prezentace operačního systému. Dostupné z <<http://www.freertos.org/>>
- [77] MCUHobby.cz: *Představujeme FreeRTOS* [online]. Publikováno 30. 1. 2010 [cit. 2012]. Dostupné z <<http://www.mcuohobby.cz/2010/01/predstavujeme-freertos/>>