

Univerzita Karlova v Praze  
Matematicko-fyzikální fakulta

# BAKALÁŘSKÁ PRÁCE



Miroslav Šedivý

## Ovládání PC pomocí webové kamery

Kabinet software a výuky informatiky

Vedoucí bakalářské práce: RNDr. Josef Pelikán

Studijní program: Informatika, Správa počítačových systémů

2009

Na tomto místě bych chtěl velmi poděkovat svým rodičům za nekončící psychickou podporu, která byla značně potřebná při tvorbě této práce. Dále mému bratruvi Milanovi za nedocenitelnou pomoc při tvorbě fotodokumentace. V neposlední řadě bych rád poděkoval RNDr. Josefu Pelikánovi, vedoucímu této práce, za velikou trpělivost a ochotu.

Prohlašuji, že jsem svou bakalářskou práci napsal samostatně a výhradně s použitím citovaných pramenů. Souhlasím se zapůjčováním práce a jejím zveřejňováním.

V Praze dne 6. 8. 2009

Miroslav Šedivý

# Obsah

<b>1. Úvod .....</b>	<b>6</b>
<b>2. Analýza problému .....</b>	<b>8</b>
2.1 Slovník použitých pojmů a zkratek .....	8
2.2 Výchozí situace.....	9
2.3 Rozbor problému.....	10
2.4 Předpoklady .....	12
<b>3. Návrh řešení .....</b>	<b>13</b>
3.1 Kalibrace .....	13
3.2 Návrh řešení.....	18
<b>4. Uživatelská rukavice .....</b>	<b>19</b>
4.1 Výběr vhodných LED diod.....	19
4.2 Konstrukce .....	22
<b>5. Návrh uživatelských gest a jejich využití.....</b>	<b>25</b>
5.1 Koncepce uživatelských gest.....	25
5.2 Změna polohy počítačového kurzoru .....	26
5.3 Určování pozice kurzoru z pozice LED diod .....	27
5.4 Dočasné vypnutí pohybu kurzoru.....	28
5.5 Stisknutí levého tlačítka .....	29
5.6 Stisknutí pravého tlačítka .....	29
<b>6. Programátorská dokumentace.....</b>	<b>30</b>
6.1 Podpůrné knihovny .....	30
6.2 Architektura softwarové části.....	32
<b>7. Uživatelská dokumentace.....</b>	<b>38</b>
7.1 Systémové požadavky.....	38
7.2 Instalace .....	38

7.3	Spuštění.....	38
7.4	Fyzické umístění kamery .....	39
7.5	Manipulace s aplikací .....	39
7.5.1	Zvolení kamery.....	41
7.5.2	Nastavení připojené kamery .....	42
7.5.3	Kalibrace detekčního algoritmu.....	43
7.5.4	Dodatečné nastavení programu TriLED.....	47
7.5.5	Indikátory aktuálního stavu programu.....	49
7.5.6	Ovládání kurzoru myši.....	50
<b>8.</b>	<b>Závěr.....</b>	<b>51</b>
8.1	Naplnění cílů práce.....	51
8.2	Diskuze a možnosti budoucího rozšíření .....	51
<b>9.</b>	<b>Literatura.....</b>	<b>53</b>

## Přílohy

<b>Příloha A.</b>	<b>Obsah příloženého DVD .....</b>	<b>54</b>
-------------------	------------------------------------	-----------

Název práce: Ovládání PC pomocí webové kamery

Autor: Miroslav Šedivý

Katedra (ústav): Kabinet software a výuky informatiky

Vedoucí bakalářské práce: RNDr. Josef Pelikán

e-mail vedoucího: Josef.Pelikan@mff.cuni.cz

Abstrakt: Cílem práce je navrhnout a implementovat interaktivní ovládání PC pomocí běžné webové kamery snímající ruku uživatele s využitím speciálního zařízení - světelné rukavice osazené barevnými LED diodami. Práce zkoumá možnosti rychlého zpracování obrazu z kamery a interaktivního rozpoznávání základních gest. Konečným cílem je pak ovládání počítače pomocí uvedené rukavice, která zcela nahradí základní funkčnost myši.

Klíčová slova: počítačové vidění, segmentace, webová kamera, zpracování obrazu v reálném čase, zařízení uživatelského vstupu

Title: Web-camera based interactive user input

Author: Miroslav Šedivý

Department: Department of software and computer science education

Supervisor: RNDr. Josef Pelikán

Supervisor's e-mail address: Josef.Pelikan@mff.cuni.cz

Abstract: The goal of this thesis is to design and implement an interactive user input device for a PC, consisting of a web-camera scanning the gestures of the user's hand. The user will be wearing a special glove fixed with colored LED diodes to aid the image analysis process. Thesis concentrates on possible approaches to real-time image processing and interactive recognition of gestures. The ultimate goal is to substitute the basic functionality of the "computer mouse".

Keywords: computer vision, segmentation, web-camera, real-time image analysis, user input device

# Kapitola 1

## Úvod

V dnešní době je neodmyslitelnou součástí každého osobního počítače myš. Toto jednoduché zařízení prošlo úspěšně zkouškou času a prokázalo svou použitelnost a funkčnost. Ovšem jednou z jeho méně vnímaných vlastností je, že dnes všechna běžná zařízení zastupující funkcionalitu myši (samotné počítačové myši, touchpady, trackbally, ...) poutají uživatele na určité místo a omezují ho tedy ve volnosti pohybu. Na první pohled se toto nemusí jevit jako nevýhoda, ale pro jisté aplikace tato vlastnost funguje jako omezení. Nezanedbatelná je i skutečnost, že používání těchto zařízení vyžaduje od uživatele jemu nepřirozené pohyby a činnosti, které v dlouhodobém horizontu mohou mít i nepříznivé důsledky na jeho zdraví.

Lze proto očekávat nástup bezkontaktních interaktivních způsobů ovládání. Do této kategorie patří mimo jiné ovládání lidskými gesty (obličej, končetiny či rovnou celého těla). Tento způsob je lidem velmi přirozený, protože představuje jeden z hlavních způsobů běžné mezilidské komunikace. Jedním z příkladů může být systém *virtuální reality*, kdy má jeho uživatel nasazené speciální brýle, do kterých je svět virtuální reality promítán. Rozumným předpokladem je, že uživatel bude chtít mít možnost se přirozeně pohybovat v rámci světa virtuální reality. Lze také očekávat, že s umělým světem bude interagovat přirozeně (tedy tak jak by činil ve světě skutečném), ale stále zde bude potřeba komunikovat se systémem generujícím virtuální realitu. Jako východisko se nabízí právě prostředky přirozené lidské komunikace, kam spadají právě neverbální gesta.

Pro využití možnosti komunikace prostřednictvím neverbálních gest je ovšem nutné zajistit jejich rozpoznání osobním počítačem. Jde o poměrně složitou úlohu z oblasti rozpoznávání obrazů, která je navíc umocněna tím, že vše musí probíhat v reálném čase. Pro zjednodušení situace lze použít některá zařízení, která podstatně neomezí uživatele a přitom napomáhají zjednodušení příslušných algoritmů. V této práci je prozkoumána možnost využití „rukavice“ s upevněnými LED diodami různých barev, které emitují dobře rozpoznatelné

světlo. Tyto diody jsou snímány webovou kamerou a výsledný obraz je dále zpracováván. Na základě analyzovaného sledu obrazů jsou pak generovány zprávy pro operační systém, které mají za následek příslušné akce kurzoru (pohyb, stisk tlačítka apod.).

Cílem práce proto je:

- **Nastudovat a zvolit nejvhodnější metody, algoritmy a datové struktury pro analýzu snímků webové kamery splňující požadavky na rychlost, efektivitu a co nejmenší nároky na výpočetní prostředky.**
- **Navrhnout vhodné provedení pomocného zařízení - uživatelské rukavice.**
- **Vypracovat co nejvhodnější podobu uživatelských gest a navrhnout postup jejich analýzy obslužným programem.** Důraz je kladen především na přirozenost pohybů pro uživatele a jejich co možná nejjednodušší počítačové zpracování.
- **Na základě získaných poznatků vytvořit funkční celkové řešení.** Toto řešení se bude skládat z uživatelské rukavice a softwaru, jež obstará analýzu a interpretaci uživatelských gest.

Práce samotná je rozdělena do několika kapitol:

- **Kapitola 2 se věnuje analýze problému,** zde jsou shrnuty dosavadní poznatky a výsledky včetně poznatků získaných v přípravné fázi;
- **Kapitola 3 se věnuje návrhu řešení -** je zde navržen postup považovaný za optimální;
- **Kapitola 4 se zabývá návrhem a implementací uživatelské rukavice;**
- **V Kapitole 5 jsou popsána uživatelská gesta;**
- **Kapitola 6 je věnována programátorské dokumentaci implementovaného řešení softwarové části;**
- **Kapitola 7 obsahuje uživatelskou dokumentaci;**
- **Kapitola 8 se věnuje závěrečnému shrnutí výsledků, zkoumá naplnění vytyčených cílů a rozebírá možnosti budoucího rozšíření;**
- **Kapitola 9 uzavírá hlavní část práce, uvádí výčet citované literatury a jiných použitých zdrojů informací;**
- **V Příloze A je uveden obsah příloženého datového nosiče, na kterém se nacházejí zdrojové kódy softwarové části rozhraní TriLED.**

# Kapitola 2

## Analýza problému

Jak je zmíněno v úvodu, problematika rozpoznávání gest na snímcích z kamery, obzvláště v reálném čase, není jednoduchá záležitost a je poměrně rozsáhlá. Analýze problému a hledání uspokojivého řešení je věnována tato kapitola.

### 2.1 Slovník použitých pojmů a zkratek

#### Obrazová segmentace

Obrazová segmentace je termín z oblasti počítačového vidění označující proces, kdy se rozdělí obraz na oblasti obrazových dat mající nějaký vztah k vyobrazeným objektům. Zároveň je jedním z nejdůležitějších kroků na cestě k analýze obrazových dat. [1]

#### GUI

Graphic User Interface - Grafické uživatelské rozhraní

#### Uživatelské rozhraní TriLED

Pracovní název pro navržené uživatelské rozhraní. Skládá se ze softwarové části (obslužný program s GUI) a hardwarové části (uživatelská rukavice s LED diodami).

#### Počítačový kurzor

Grafický indikátor generovaný operačním systémem a zobrazovaný na počítačovém monitoru, či jiném zobrazovacím zařízení, který reaguje



definovaným způsobem na informace poskytnuté vybraným vstupním zařízením.

### **Kurzor myši**

Počítačový kurzor patřící počítačové myši.

### **Obrazový bod**

V anglosaské literatuře označován jako *pixel*, je nejmenší jednotka digitálního obrazu. V případě barevných obrazů nese informaci o barvě, v případě černobílých obrazů pak informaci o úrovni jasu.

### **Obrazová data**

Množina obrazových bodů.

## **2.2 Výchozí situace**

Výchozí stav se dá shrnout následovně: máme standardní osobní počítač, ke kterému je připojená webová kamera. Cílový stav je, že uživatel bude vykonávat gesta, která budou snímána a překládána do povelů pro kurzor myši v rámci *operačního systému*<sup>1</sup> počítače. Tuto službu bude obstarávat software, který bude pomocí připojené webové kamery snímat uživatelskou ruku vybavenou speciální rukavicí.

V podstatě jde o analýzu kontinuálního proudu snímků zachycujících uživatelská gesta, přičemž je potřeba interpretovat potřebným způsobem obrazová data těchto snímků a jejich změny. Zde je nutno uvést, že uvedený postup spadá obecně mezi netriviální problémy z oblasti počítačového vidění a zároveň je poměrně náročný na výpočetní kapacitu. Pokud má být analýza prováděna v reálném čase s minimálním zpožděním, je nezbytné, aby výpočetní nároky procesu interpretace obrazových dat byly co možná nejmenší<sup>2</sup>, nicméně musí být garantována dostatečná přesnost rozpoznání gest. Pokud mají být tyto do jisté míry protichůdné nároky uspokojeny, je nutno nalézt určitý kompromis, který sice na jedné straně nepatrně uživatele omezí, na straně druhé však výrazně napomůže zjednodušení procesu rozpoznávání a zrychlení algoritmů. Optimálním řešením je použití speciální uživatelské rukavice, jež má na koncích

---

<sup>1</sup> Nadále v textu označován zkratkou OS.

<sup>2</sup> A to i z toho důvodu, že uživatel bude používat rozhraní TriLED pouze jako prostředek na ovládání počítače, a tedy bude chtít mít k dispozici co nejvíce systémových prostředků pro svou vlastní práci.

palce, ukazováku a prostředníku různobarevně svítivé LED diody. Tímto se poměrně komplexní problém *obrazové segmentace* celé dlaně s prsty a interpretace její pózy převede na efektivněji řešitelný problém segmentace jasných, od pozadí zřetelně odlišitelných, LED diod.

Principem analýzy uživatelových gest tedy bude detekce diod na snímcích webové kamery a následná interpretace změn jejich poloh mezi jednotlivými navazujícími snímky. Je nutné opět zdůraznit, že rychlost zpracování a vyhodnocení jednoho snímku z kamery je klíčová. Zaručuje totiž, že kurzor myši v počítači se nebude opožďovat za gesty konanými před webovou kamerou. Dále je neméně důležitým prvkem počet snímků za vteřinu (označován dále jako FPS<sup>3</sup>), jež dokáže webová kamera vyprodukovat. Dostatečně velký počet snímků za vteřinu zaručuje plynulost pohybu kurzoru myši a přesnější analýzu dalších gest.

Neopomenutelnou částí úkolu je navrhnout přiměřená uživatelská gesta, která budou co nejvíce jednoduchá na algoritmické vyhodnocení, dostatečně nezaměnitelná a zároveň maximálně přirozená pro samotného uživatele.

## 2.3 Rozbor problému

Při bližším pohledu na řešení problému se jeví jako optimální rozložit celý proces detekce a zpracování gest na oddělené dílčí procesy. Jde o následující kroky (fáze), které jsou respektovány následně i v SW řešení ve formě modulů:

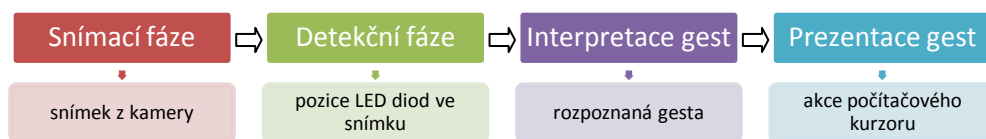
- **Snímací fáze** (neboli získání snímků z kamery)
- **Detekční fáze** (lokalizace LED diod v rámci snímku)
- **Interpretace gest** (tedy změn pozic LED diod mezi snímky)
- **Prezentace gest** (ve formě akcí kurzoru myši)
- **GUI** (jeho návrh a vytvoření)

Mezi těmito dílčími procesy dále definujeme jasně stanovené vztahy, které v programu zohledníme příslušně navrženými rozhraními mezi jednotlivými moduly.

Tyto moduly tvoří určitou „výrobní linku“ na instrukce pro počítačový kurzor. Graficky je proces znázorněn na obrázku 2-A.

---

<sup>3</sup> Z anglické fráze Frames Per Second, jejímž překladem je právě počet snímků za vteřinu.



**Obrázek 2-A** *Schéma procesu zpracování gest. Pod názvem modulu je uveden produkt příslušného modulu, který je předán dalšímu článku v pořadí.*

Z důvodů snadného praktického nasazení nebylo přistoupeno k vlastnímu vývoji knihoven, které by řešily problematiku jednotlivých modulů na nejnižší úrovni. Jednak jsou k dispozici veřejně dostupné precizně provedené programové knihovny třetích stran, pokrývající oblasti uvedených podproblémů, jednak by řešení pomocí vlastního kódu na nejnižší úrovni mohlo časem vést ke ztrátě kompatibility. Navíc zvolená cesta umožňuje neztrácet čas návrhem a implementací sekcí programu, které nesouvisí přímo s úkolem řešeným v této práci a věnovat více úsilí vlastní problematice.

Zavedená modularita řešení a jasně vymezená rozhraní mezi moduly také umožňují provádět zásahy do jednotlivých modulů, aniž by byl ovlivněn zbytek kódu. Tato vlastnost umožňuje například pohodlnější výměnu konkrétních pomocných knihoven pro jednotlivé úseky<sup>4</sup>. Další výhoda by se projevila při uzpůsobování programu na jiný operační systém, při kterém by jen stačilo provést změny v modulech využívající nepřenositelný kód nebo knihovny. Tyto změny by se pak nepropagovaly za hranice modulu.

Následuje rozbor jednotlivých modulů.

### **Snímací fáze**

V této fázi příslušný modul získává v pravidelných intervalech snímky z webové kamery. K tomu je využita knihovna poskytující univerzální a unifikovaný programátorský přístup k webovým kamerám a jejich funkcím. Získané snímky ve vhodném formátu jsou dále předávány modulu zajišťujícímu detekční fázi.

<sup>4</sup> Což může být výhodné v případě, že se vyskytne v daném odvětví nějaká nová knihovna, poskytující lepší funkčnost.

## Detekční fáze

Jde o fázi, v níž se nachází těžiště celého zkoumání. V této části se v dodaném snímku vyhledají obrazové informace náležící jednotlivým LED diodám. Pro každou diodu se poté určí jejich *referenční bod*, který bude nadále využíván jako reprezentace pozice diody na snímku. Takto získané informace se předají dále. V této fázi bude využita vhodná knihovna základních funkcí pro počítačové vidění.

## Interpretace gest

Modul realizující tuto fázi na vstupu obdrží referenční body jednotlivých LED diod, resp. informaci o jejich nepřítomnosti na aktuálním snímku. Interně si tato data bude v průběhu ukládat a na základě jejich změn v průběhu času bude vyhodnocovat, jaká gesta uživatel provádí.

## Prezentace gest

Zde se budou analyzovaná gesta delegovat operačnímu systému v podobě příkazů pro počítačový kurzor.

## GUI

Modul stojící poněkud stranou od předešlých, ale zároveň neméně důležitý. Zastřešuje zbylé moduly a umožňuje jejich propojení. Zároveň prezentuje uživateli grafické ovládací prvky, jimiž může monitorovat a (dle potřeby) nastavit chování jednotlivých modulů.

## 2.4 Předpoklady

Je zřejmé, že výhody plynoucí z použití osvětlené uživatelské rukavice budou přítomny pouze v případě, že budou platit tyto podmínky:

- Na snímcích pořízených webovou kamerou musí být LED diody nezanedbatelně jasnější nežli zbytek snímané scény, dále označovaný jako *pozadí*.
- Na snímané scéně se nevyskytuje žádný jiný objekt, mající podobné jasové a zároveň barevné vlastnosti jako některá z LED diod uživatelské rukavice.

Tyto předpoklady lze považovat za přijatelné, v převážné části prostředí, v nichž bude zařízení využíváno, jsou snadno splnitelné. Splnění těchto předpokladů značnou měrou ulehčuje detekční fázi.

## Kapitola 3

### Návrh řešení

Ze studia dostupné literatury vyplynul poznatek, že každý konkrétnější problém z oblasti počítačového vidění vyžaduje individuální přístup a aplikaci specifických metod, které jsou jemu uzpůsobeny. Obecně známé jsou pouze šablony metodik řešení určitých skupin problémů.

Sledovaným způsobem segmentace obrazových dat se již několik prací zabývalo a jsou tedy známy různě efektivní metody pro různé konkrétní okolnosti jejich aplikace. Při studiu literatury byly prostudovány vybrané zdroje [1][2][3][4] a byla vyhodnocena vhodnost jednotlivých probíraných postupů pro účely této práce. Dále uvedený způsob řešení vychází zejména z práce [2].

Při zpracování návrhu vlastního řešení byly nastudované poznatky podpořeny vlastními experimenty. Již od začátku se jevila jako optimální metoda filtrující jednotlivé obrazové body na základě jasových a barevných charakteristik jednotlivých diod. Nutí sice k použití jisté počáteční „kalibrační fáze“, nicméně samotná segmentace obrazu za podmínek, stanovených v kapitole 2.4, probíhá velmi efektivně. Časová složitost na zpracování jednoho snímku odpovídá  $O(n)$ , kde  $n$  je počet obrazových bodů, tedy je lineární – tato skutečnost by měla garantovat velmi rychlé zpracování v reálném čase.

Dále jsou popsány principy kalibrace a detekce diod, které tvoří základ praktické implementace.

#### 3.1 Kalibrace

V procesu detekce diod je využíváno informací digitálního snímku, kdy je jako základ použito vyjádření jednotlivých bodů snímku ve formě uspořádaných

trojic  $(R, G, B)$ . Základní myšlenka spočívá v přiřazení určitých hodnot těchto trojic jednotlivým použitým diodám LED a na základě sledování vybraných charakteristik určit pozici jednotlivých diod.

Z předpokladů v kapitole 2.4 vyplývá, že si při segmentaci obrazových dat patřících LED diodám lze provést první selekci dat podle úrovně jasu jednotlivých *obrazových bodů*. Obrazové body s jasovou složkou pod určitou hranicí musí patřit k pozadí a není třeba je dále zkoumat. Naopak u bodů s jasovou složkou nad touto hranicí je vysoká pravděpodobnost, že patří k LED diodám<sup>5</sup>. Tato skutečnost je dále zohledněna a aplikace uvedeného prahu hraje významnou roli v procesech detekce, jak bylo i experimentálně ověřeno; z experimentů zároveň vyplynula optimální hodnota prahu.

Dále popsany postup představuje jistou modifikaci postupu uvedeného v odborném článku [2]. Pro účely snadnějšího popisu budeme využívat prostor barev RGB jako prostor uspořádaných trojic  $(R, G, B)$ , kde jednotlivé složky mohou nabývat hodnot 0 až  $RGBmax$ , kde  $RGBmax$  je dáno barevným rozlišením použité kamery (v rámci experimentů se hodnota  $RGBmax$  rovnala 255).

Pokud by dále bylo využíváno pouze uvedeného prostoru barev, pro jednotlivé barevné body by existovalo velké množství možností trojic  $(R, G, B)$ , což by mohlo dále vést k neúměrné výpočetní náročnosti. Z tohoto a jiných důvodů, které budou uvedeny dále, byla použita určitá normalizace barevné scény, která spočívá v následujícím:

- interval  $0..RGBmax$  se rozdělí na  $M$  stejně velkých intervalů (jako optimální se v průběhu experimentů ukázala hodnota  $M = 32$ )
- prostor trojic  $(R, G, B)$  se tak rozdělí na krychle, tyto krychle nazveme *kyblíky*

Tímto se prostor barev rozdělil do  $M^3$  kyblíků, tento nově vzniklý prostor umožňuje efektivnější práci s informacemi o scéně. Souřadnice v tomto prostoru budeme označovat pro odlišení  $(r, g, b)$ . Libovolnému obrazovému bodu  $X$  určitého snímku s barevnými složkami  $(R, G, B)$  je tedy možné přiřadit souřadnice  $(r, g, b)$  kyblíku, do něž spadá jeho barevné složky.

Vzhledem k tomu, že v průběhu snímání gest každý snímek představuje scénu, na níž se nalézá nějaké pozadí a tři rozsvícené diody, je prvním úkolem v rámci detekce odlišit diody od pozadí. V zásadě lze použít postupy jako nastavení prahu jasu (tento postup využívá skutečnosti, že jas diod převyšuje jas pozadí)

---

<sup>5</sup> Jedná se o způsob obrazové segmentace zvaný thresholding a zde je velmi účinný vzhledem k zmíněným předpokladům na snímanou scénu.

nebo vyhledávání určitých barev na snímku. Jako optimální se ukázal postup kombinující oba základní postupy.

Aby bylo možné odlišit diody od pozadí u zařízení, které ještě nebylo ozkoušeno, je třeba provést tzv. kalibraci. Ta spočívá ve využití tzv. *barevného profilu* (označíme ho CP), což je matice o rozměrech  $M \times M \times M$ , která reprezentuje prostor trojic  $(R, G, B)$  rozdělený na kyblíky. Její prvky jsou struktury obsahující určité informace o kyblíku, konkrétně:

- údaj o tom, zda daný kyblík patří diodě nebo pozadí
- údaj o tom, jestli kyblík patřící diodě patří jedné vybrané diodě (a jaké) nebo je společný pro více diod (hodnoty 0, 1, 2 pro jednotlivé diody, hodnota 3 pro kyblík společný více diodám)
- údaj charakterizující pravděpodobnost, že kyblík skutečně patří diodě (tzv. *relevance*)

Uvedený barevný profil se v průběhu kalibrace vytváří následujícím způsobem.

Nejprve se vytvoří  $K$  snímků pozadí (tedy scény obsahující pouze pozadí, případně uživatelskou rukavici se zhasnutými diodami) a pro každý obrazový bod daného snímku se určí, do kterého kyblíku patří. Z těchto hodnot se sestaví binární histogram kyblíků daného snímku (každý kyblík se započítá maximálně jednou, pokud hodnota jasu nepřevyší práh, potom se kyblík nezapočítává) tedy funkce  $N_{BG}^i(r, g, b)$  nabývající hodnot 0 nebo 1 ( $i$  je pořadové číslo snímku). Tento postup se provede pro každý snímek a následně se sestaví výsledný histogram charakterizující, na kolika snímcích se daný kyblík vyskytl, tedy zavedeme funkci

$$N_{BG}(r, g, b) = \sum_{i=1}^K N_{BG}^i(r, g, b)$$

V další fázi již použijeme snímky scény, kdy jsou již diody rozsvícené a obdobně zavedeme a spočteme hodnoty funkce  $N_{LED}(r, g, b)$  charakterizující na kolika snímcích se daný kyblík vyskytl.

Nakonec pro trojice  $(r, g, b)$  takové, že  $N_{LED}(r, g, b) \neq 0$ , vypočteme veličinu

$$V_{CP}(r, g, b) = \frac{N_{LED}(r, g, b) - N_{BG}(r, g, b)}{N_{LED}(r, g, b)}$$

V případě, že  $N_{LED}(r, g, b) = 0$ , můžeme položit  $V_{CP}(r, g, b) = 0$ .

Tato veličina v podstatě charakterizuje kyblíky po stránce toho, zda by se mohlo jednat o kyblík patřící LED diodě či nikoliv a to proto, že pokud je tato hodnota rovna 1, potom se jedná o kyblík, který se objevil pouze na snímcích

s rozsvícenými diodami, a tedy nutně musí patřit diodě. Totéž platí i v případě, kdy je hodnota blízká 1, jen s menší pravděpodobností. Naopak pokud je hodnota blízká nule, nulová nebo záporná, bude se jednat o kyblík patřící pozadí.

Uvedenou hodnotu můžeme normovat na hodnotu přímo charakterizující pravděpodobnost toho, že se jedná o kyblík spojený s diodou následujícím zavedením funkce

$$P_{LED}(r, g, b) = V_{CP}(r, g, b)$$

pokud je  $V_{CP}(r, g, b)$  nezáporná, v ostatních případech pokládáme

$$P_{LED}(r, g, b) = 0$$

Pro jednodušší vyjádření pak tuto hodnotu normujeme na celá čísla v rozsahu 0 až 100 a tuto veličinu nazveme *relevance* a označíme  $Q_{LED}(r, g, b)$ . Hodnotu  $Q_{LED}(r, g, b)$  zapíšeme do barevného profilu.

Aby bylo možné odlišit lépe diody od pozadí v okamžiku, kdy se hledají kontury jednotlivých diod, zavádí se pojem *dolního a horního prahu*.

Pro každý bod daného snímku s výjimkou okrajových bodů spočítáme kumulativní hodnotu relevance pro tento bod a jeho nejbližší sousedy, tedy

$$Q(X) = \sum_Y Q_{LED}(i, j, k)$$

kde sčítáme přes souřadnice  $(i, j, k)$  prostoru kyblíků přiřazené hodnotám barevných složek nejbližších sousedů  $Y$  bodu  $X$  včetně bodu samotného. Tento výpočet provedeme pro všechny body všech snímků pozadí a určíme maximum uvedených hodnot, které označíme  $Q_{min}$ . Obdobně postupujeme pro všechny snímky s rozsvícenými LED diodami a výslednou hodnotu označíme  $Q_{max}$ .

Z uvedeného postupu vyplývá, že by mělo platit, že

$$Q_{min} < Q_{max}$$

Pokud je naopak  $Q_{min} \geq Q_{max}$ , znamená to, že nejsou splněny podmínky kapitoly 2.4 (např. protože je pozadí přesvícené). Rozdíl  $Q_{max} - Q_{min}$  je možné chápat jako indikátor kvality scény, čím je hodnota vyšší, tím je větší odstup jasu diod od jasu pozadí a tedy i určení polohy diod je přesnější. Hodnoty  $Q_{min}$  a  $Q_{max}$  se využijí dále.

Nyní se do barevného profilu pro každý kyblík zapíše informace o tom, zda kyblík patří nebo nepatří diodě (podle toho, zda pro příslušnou hodnotu



relevance platí  $Q_{LED}(r, g, b) > 0$ ). Posledním krokem přípravné fáze je určení pro každý kyblík, který přísluší nějaké diodě, které konkrétní diodě patří.

Pro zjištění, ke které diodě kyblík patří, se provede proces *detekce objektu*. Tento proces využívá skutečnosti, že je znám výřez snímku, v němž se jednoznačně nachází obraz určité rozsvícené diody. Pro toto místo se pak aplikuje následující algoritmus:

- každému bodu výřezu se přiřadí hodnoty souřadnic  $(r, g, b)$  kyblíku, který odpovídá barvě obrazového bodu
- pro body  $X$ , které neleží na hranici výřezu, se spočte hodnota  $Q(X)$  rovná součtu relevancí daného bodu a jeho nejbližších sousedů (relevance se bere z kyblíku, jehož souřadnice  $(r, g, b)$  byly danému bodu přiřazeny)
- pokud hodnota  $Q(X)$  leží mezi hodnotami  $Q_{min}$  a  $Q_{max}$ , přiřadí se bodu  $X$  hodnota  $T(x) = 1$ , v opačném případě se přiřadí hodnota  $T(x) = 0$
- výsledkem je bitová mapa výřezu snímku, kde body s hodnotami  $T(x) = 1$  označují kandidáty na obrazové body diody
- na vzniklou mapu se aplikuje postup vyhledání souvislé komponenty (viz např. [1] nebo [5]), nalezená komponenta se ohraničí nejmenším možným obdélníkem.

Uvedený postup se aplikuje postupně pro všechny tři diody a následně se zjišťuje, které odpovídající kyblíky z barevného profilu odpovídají bodům v jednotlivých obdélnících. Pokud pro určitý kyblík platí, že se pouze v jednom z obdélníků našly body s hodnotou souřadnic  $(r, g, b)$  odpovídající tomuto kyblíku, přiřadí se tomuto kyblíku hodnota odpovídající dané diodě, ke které obdélník patří (0, 1 nebo 2). Pokud se pro daný kyblík najdou body v různých obdélnících, pak nelze určit spolehlivě diodu a podle toho se přiřadí hodnota příslušnosti rovna 3.

Tímto způsobem je barevný profil zcela definován a zároveň je ukončena zcela fáze kalibrace.

Fáze detekce se uplatňuje v okamžiku, kdy již je požadováno určení pozic jednotlivých diod. Pro tuto fázi se použije obdobný postup, jako pro kalibraci nyní již s využitím definovaného barevného profilu.

Pro každý snímek se provede následující:

- každému bodu snímku se přiřadí hodnoty souřadnic  $(r, g, b)$  kyblíku, který odpovídá barvě obrazového bodu

- pro body  $X$ , které neleží na hranici výřezu, se spočte hodnota  $Q(X)$  rovná součtu relevancí daného bodu a jeho nejbližších susedů (relevance se bere z kyblíku, jehož souřadnice  $(r, g, b)$  byly danému bodu přiřazeny)
- pokud hodnota  $Q(X)$  leží mezi hodnotami  $Q_{min}$  a  $Q_{max}$ , přiřadí se bodu  $X$  hodnota  $T(x) = 1$ , v opačném případě se přiřadí hodnota  $T(x) = 0$
- výsledkem je bitová mapa snímku, kde body s hodnotami  $T(x) = 1$  označují kandidáty na obrazové body jednotlivých diod
- na vzniklou mapu se aplikuje postup vyhledání souvislých komponent, nalezené komponenty se ohraničí nejmenším možným obdélníkem – pokud jsou splněny podmínky týkající se prostředí, měly by být nalezeny celkem tři souvislé komponenty, každá odpovídající jedné diodě
- pro každý obdélník se určí, jaké diodě odpovídá tak, že se nastaví hodnoty  $N_R$ ,  $N_G$  a  $N_B$  na nulovou hodnotu a pro všechny body  $X$  s hodnotou  $T(x) = 1$  určí odpovídající kyblík a započte se skóre jednotlivých diod, tedy se zvýší hodnota  $N_R$ ,  $N_G$  nebo  $N_B$  o 1, podle toho, které diodě je kyblík přiřazen. Nakonec se určí „vítěz“ – dioda, jejíž index  $D$  má maximální hodnotu  $N_D$
- na závěr se spočítá těžiště těch bodů, které patří „vítězné“ diodě spolu s body patřícími více diodám, jako váhu jednotlivých bodů bereme odpovídající hodnotu relevance. Takto získaný bod je referenčním bodem dané diody, celkem se tedy získají tři referenční body odpovídající jednotlivým diodám.

Souřadnice uvedených referenčních bodů slouží jako vstup do další fáze, fáze rozpoznání gest. Principy rozpoznání gest jsou popsány v kapitole 5.

## 3.2 Návrh řešení

Návrh řešení využívá poznatků uvedených v kapitole 3.1. Postup je následující:

Pro zjištění barevného profilu je aplikován postup kalibrace uvedený v kapitole 3.1. Tento postup lze opakovat několikrát, s cílem nastavit optimální parametry prostředí.

Následně je již spuštěna detekce využívající postupy ze zmíněné kapitoly. Pomocí detekce jsou zjištěny referenční body jednotlivých diod a je provedena analýza jejich vzájemné pozice, kdy se využívá pozice jednotlivých bodů vůči sobě, případně vůči jejich společnému těžišti. Získané informace jsou převedeny na jednotlivé situace způsobem popsaným v kapitole 5 a následně jsou prostředky operačního systému prezentovány jako akce kurzoru.

# Kapitola 4

## Uživatelská rukavice

### 4.1 Výběr vhodných LED diod

Volba vhodných LED diod zásadně ovlivňuje možnosti zpracování uživatelských gest, neboť je potřeba zachovat maximální viditelnost světelné plochy v průběhu jejich vykonávání. Vzhledem k tomu, že je potřeba detekovat správně polohu diody i při třírozměrných gestech jako je např. náklon prstu s diodou, kdy dioda sama nemusí být v ideální poloze vůči objektivu kamery, je nezbytné zajistit dostatečnou svítivost i v takových případech. V průběhu přípravné fáze projektu bylo proto testováno několik různých dostupných velikostí a provedení LED diod.

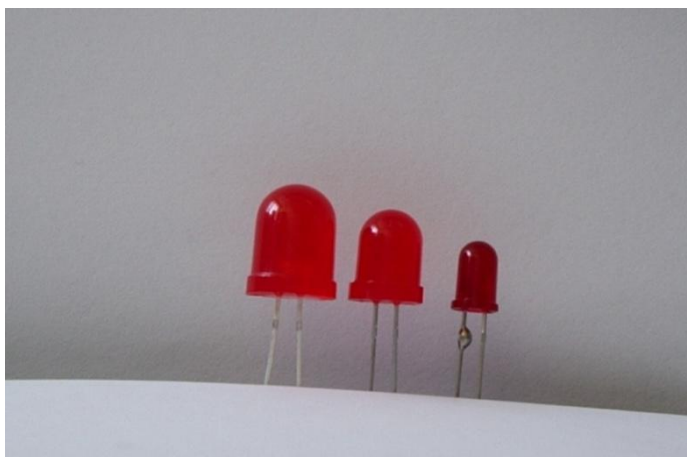
Co se velikosti týče, byly zkoušeny LED diody *klasického provedení*<sup>6</sup> o průměrech 5mm, 8mm a 10mm. V průběhu experimentu bylo patrné, že důležitou vlastností je *vyzařovací úhel*<sup>7</sup>, který by měl být co možná největší, aby si LED diody i při náklonu vůči kameře zachovaly co možná nejvíce správných světelných charakteristik<sup>8</sup>.

---

<sup>6</sup> Pojmem klasické provedení se myslí LED dioda válcovitého tvaru těla s vyzařovacím koncem zaobleným do polokoule, která na sobě nemá žádné další montážní prvky, viz obrázek 4-A.

<sup>7</sup> Vyzařovací úhel značí rozptyl světelných paprsků od vyzařovací osy LED diody. Neboli, jak moc je vidět emitované světlo při pohledu ze strany.

<sup>8</sup> Tím se rozumí, že obrazové informace na snímcích kamery patřící LED diodě mají stejné nebo velmi blízké vlastnosti (tedy barevnou informaci, jas, apod.)



**Obrázek 4-A** Zástupci klasického provedení LED diod o velikostech průměru 10mm, 8mm a 5mm (zleva doprava).

Při testování také byli podrobeni zkoumání zástupci jak *čirého*, tak *difúzního*<sup>9</sup> provedení. Zde se zdá jako výhodnější použití difúzní alternativy. Obecně u testovaných zástupců čirého provedení bylo emitované světlo až příliš silné (což se při snímání kamerou projevilo značným přesvícením a tedy i ztrátou barevné informace) a při pohledu ze strany bylo naopak poměrně nevýrazné. U difúzních verzí bylo světlo vyzařované v definovaném úhlu rovnoměrnější a samotný vyzařovací úhel byl většinou větší. Také při pohledu ze strany je LED dioda s difúzním tělem výrazněji lépe snímatelná, protože jí emitované světlo se rovnoměrným způsobem šíří celým jejím tělem.

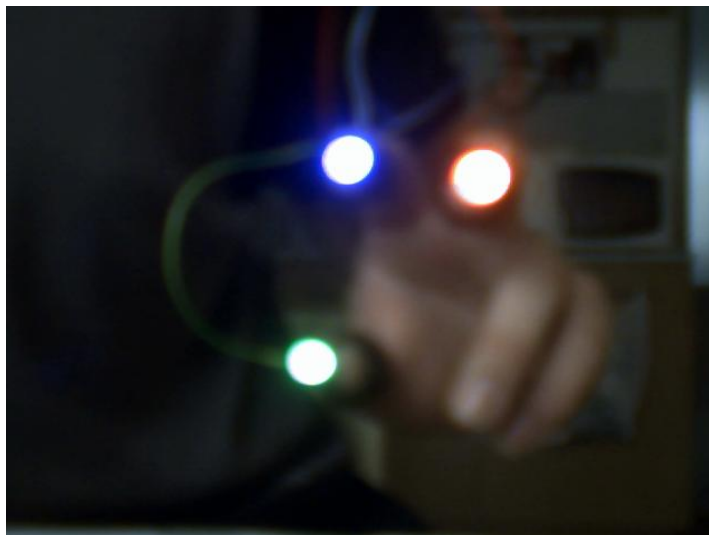


**Obrázek 4-B** Zástupci difúzního a čirého provedení těla LED diody (vedeny zleva doprava) o průměru 5mm.

---

<sup>9</sup> Pojem čiré a difúzní provedení se vztahuje ke konstrukci těla diody. Při difúzním provedení je tělo zakalené, při čirém je průhledné. Dopad na světelné vlastnosti je podobný jako u čirých a matných neúsporných žárovek. Viz obrázek 4-B.

Jak již bylo uvedeno, je potřebné volit diody s přiměřenou svítivostí. Příliš nízká by způsobila, že diody by mohly být na snímcích kamery příliš nevýrazné a splývat tak s pozadím. Zároveň nadměrně velká svítivost způsobuje přesvícení a LED dioda tak na snímcích ztrácí veškerou barevnou informaci. Viz obrázek 4-C.



**Obrázek 4-C** Na obrázku je vidět obrazová reprezentace LED diod, které mají svítivost na hraně přijatelné meze.

Jako nejvhodnější kandidáti byly shledány LED diody *klasického difúzního provedení*, s lehce zvýšenou svítivostí<sup>10</sup>, s *vyzařovacím úhlem* 50° a průměrem 10mm. Při experimentech bylo patrné, že takto velká LED dioda produkuje oproti menším více obrazových dat a uživatel tak může být dále od kamery. Zároveň se ukázalo, že si jejich snímaná obrazová reprezentace dostatečně zachovává charakteristické vlastnosti i při rozumných náklonech osy vyzařování od normály roviny kamery, které doprovázejí vykonávání uživatelských gest. Zvolená svítivost se ukázala jako nejlepší, protože si při ní diody uchovávají jasový odstup od pozadí snímané scény a zároveň si ponechávají dostatečné množství barevné informace sloužící k jejich vzájemnému odlišení.

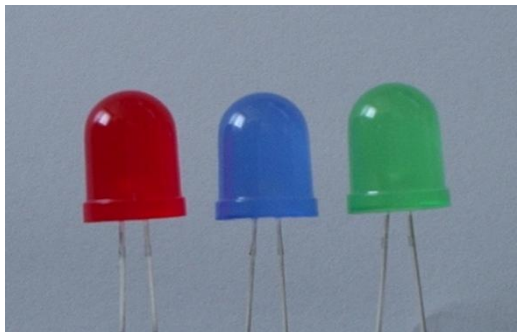
Barva LED diod byla volena na základě potřeby umět je co nejrychleji a nepřesněji od sebe na scéně odlišit. Přírozenou volbou se proto stala trojice *červená, zelená a modrá*<sup>11</sup>. Díky rozložení jednotlivých vlnových délek této trojice v barevném spektru je zaručena vysoká pravděpodobnost, že i po

---

<sup>10</sup> V jednotkách luminiscence 600-1500mcd.

<sup>11</sup> Vlnové délky světla vyzařované konkrétními vybranými LED diodami uvedené v jejich dokumentaci jsou 470nm (modrá), 525nm (zelená) a 630nm (červená).

zpracování webovou kamerou budou mít na snímcích LED diody vzájemně odlišné barevné informace. Tato vlastnost bude zachována i při různých světelných podmínkách, kdy se může barevnost lehce odlišovat<sup>12</sup>.



**Obrázek 4-D** Vybrané tři LED diody použité dále v projektu.

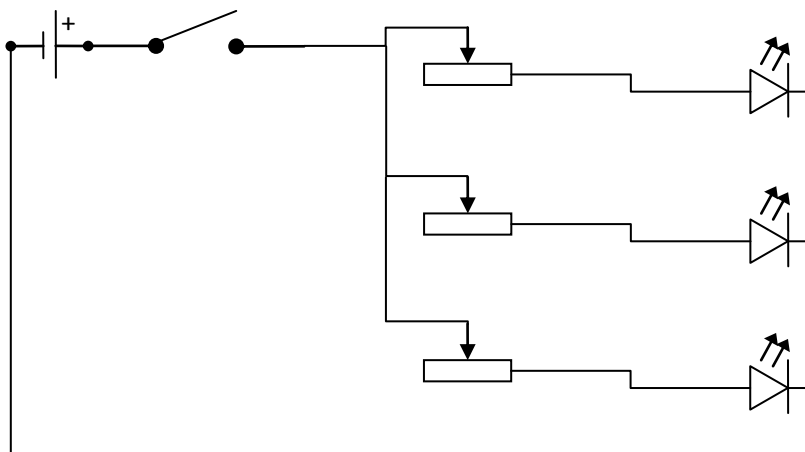
## 4.2 Konstrukce

Uživatelská rukavice se skládá z upínatelného popruhu, který se upevňuje kolem uživatelova zápěstí. Tento popruh slouží jako nosná konstrukce, na kterou se přidávají zbylé komponenty. Konkrétně jde o držák pro 3 tužkové baterie (typové označení velikosti AA, napětí jednoho článku 1,5V), vypínač a tištěný spoj, do kterého jsou zasazeny potenciometry. Vše je propojeno vodiči podle schématu zapojení uvedeného na obrázku 4-E. Pro zajištění maximální možnosti přizpůsobení konkrétním podmínkám byl místo předřadného odporu u každé diody použit potenciometr<sup>13</sup>, aby se dala jejich svítivost dle potřeby doladit.

---

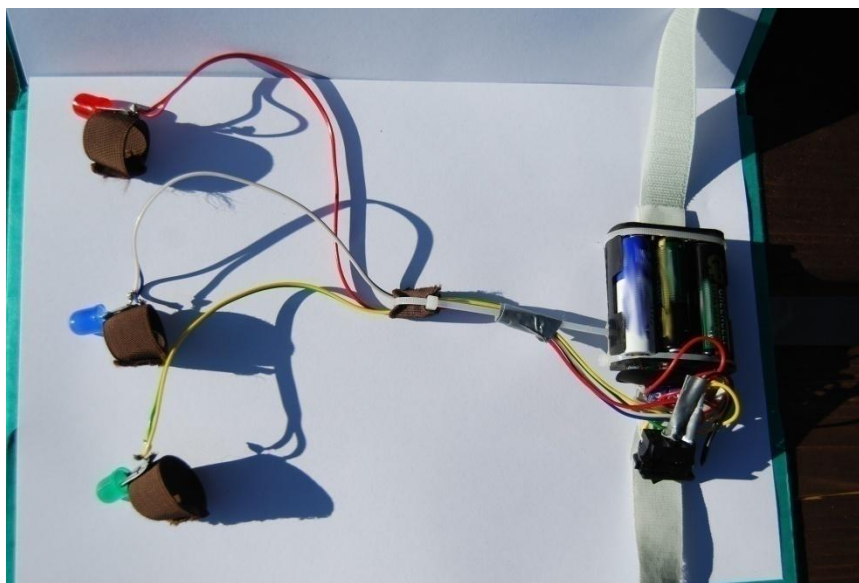
<sup>12</sup> Tato odlišnost vzniká při procesu digitalizace snímků webovou kamerou. Záleží na hardwarovém a softwarovém vybavení konkrétní kamery, jak dobře dokáže vyhodnotit světelné podmínky.

<sup>13</sup> Konkrétně s rozsahem 0 až 1k $\Omega$ .

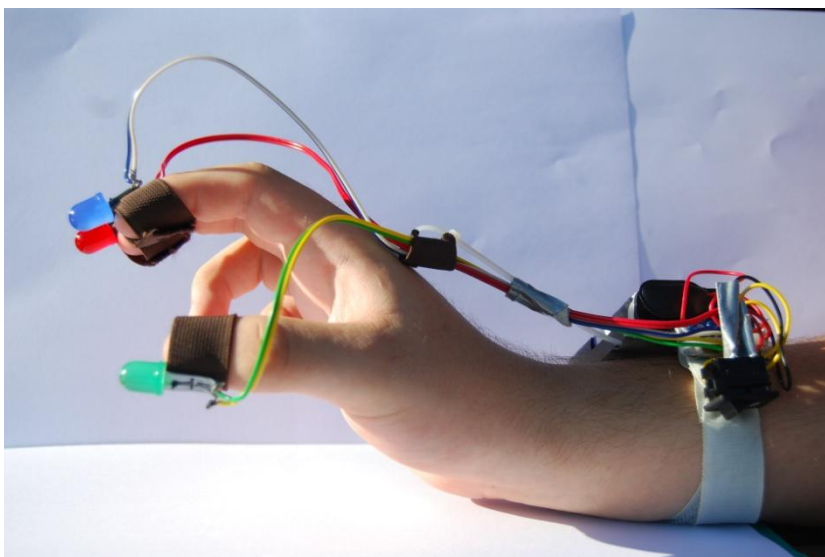


**Obrázek 4-E** Schéma zapojení elektrických součástek uživatelské rukavice.

Na koncích palce, ukazováku a prostředníku se nacházejí speciální „náprstky“ tvořené z elastického pásu obínajícího konec prstu. Na „vrchní“ části (tj. v oblasti nehtu) těchto náprstků je připevněna LED dioda příslušné barvy. Diody jsou spojeny vodiči se zbylými prvky konstrukce uživatelské „rukavice“. Podoba použitého konstrukčního řešení je zachycena na obrázcích 4-F a 4-G.



**Obrázek 4-F** Použité konstrukční řešení uživatelské „rukavice“.



**Obrázek 4-G** *Použité konstrukční řešení uživatelské „rukavice“ nasazené na uživatelovu ruku.*

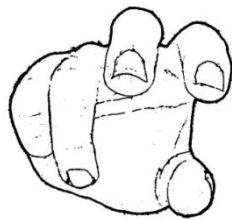


# Kapitola 5

## Návrh uživatelských gest a jejich využití

### 5.1 Koncepce uživatelských gest

Již na začátku přípravné fáze byl vybrán obecný koncept držení prstů, které jednak respektuje požadavek přirozenosti držení ruky, jednak zajišťuje dostatečné možnosti tvorby gest. Základní pozice je zobrazena na obrázku 5-A.



**Obrázek 5-A** *Schéma základního držení ruky.*

Zobrazené držení ruky představuje zároveň *gesto klidového stavu* - tedy stavu, kdy není vyhodnocováno žádné dodatečné gesto, kromě případné *změny polohy počítačového kurzoru*.

Z tohoto klidového stavu vycházejí zbylá gesta, jmenovitě *změna polohy počítačového kurzoru*, *akce stisknutí levého tlačítka*, *akce stisknutí pravého tlačítka* a dodatečně pak *gesto zvednutí myši z podložky*, jehož potřeba je objasněna v kapitole 5.2

Na tomto místě je také vhodné upozornit, že se předpokládá mít možnost při provádění změny polohy zároveň provádět akci stisknutí jednoho ze dvou definovaných tlačítek.

## 5.2 Změna polohy počítačového kurzoru

Základní schopnost jakéhokoliv zařízení zastupujícího funkcionalitu počítačové myši je měnit polohu počítačového kurzoru myši na základě akcí uživatele. U standardní počítačové myši toho docílíme tak, že myši posouváme po podložce, zabudované mechanické nebo optické čidlo na spodní straně myši registruje změnu rychlosti a směru jejího pohybu. Tyto údaje pak vhodným způsobem převádí na změny polohy počítačového kurzoru.

Výše uvedená metoda by se dala nazvat *relativní určování polohy*, jelikož zařízení neví „kde se nachází“, ale pouze interpretuje pohyb určitým směrem ze stávající pozice.

Dalším způsobem je *absolutní určování polohy*; zde zařízení přesně nastavuje polohu počítačového kurzoru, protože z akcí uživatele je odvoditelné, jaké konkrétní souřadnice měl na mysli. Do této kategorie lze zařadit například různé dotykové displeje, kde si uživatel doslova „ukáže“, kde chce provést odpovídající akci.

V raných fázích projektu byla zvažována implementace absolutního určování polohy s tím, že by se poloha na obrazovce odvozovala z umístění obrazu LED diody umístěné na ukazováku. V tomto scénáři by se v rámci snímku z kamery vyhradil obdélníkový prostor, v kterém by se mohla tato dioda pohybovat, přičemž by byla její poloha (v rámci tohoto obdélníku) přepočítávána do souřadnic aktuálního počítačového displeje. Vyhrazený obdélníkový prostor je tvořen oblastí pozic diody ukazováku, ve kterých se nachází při splnění podmínky, že na snímku jsou přítomny i zbylé diody - tato podmínka plyne z toho, že potřebujeme při přepočítávání pozice polohovací diody mít na scéně k dispozici zbylé diody, abychom mohli detekovat zbylá gesta.

Ovšem během předběžných experimentů bylo zjištěno, že tento přístup má nemalý počet nevýhod. Jednak by bylo nutné ošetřit situaci, kdy má uživatel více aktivních displejů, propojených do jednoho celku, který nemusí nutně tvořit obdélník, ale složitější útvar. Dále je zřejmé, že i kdyby bylo použito *subpixelové určování pozic diod*<sup>14</sup> na snímcích, díky přepočtu polohy do systému souřadnic uživatelova displeje by významně klesala rozlišovací schopnost u displejů větších rozlišení. Toto by mělo za následek snížení schopnosti nastavit pozici počítačového kurzoru přesně na vybranou pozici. V neposlední řadě by bylo nutno kalibrovat rozměry vyhrazené oblasti pro onu diodu, protože používat

---

<sup>14</sup> Určení pozice LED diod je s vyšší přesností, která je vyjádřena zlomky rozměrů obrazového bodu a ne pouze jejími celými násobky

univerzálně předdefinovanou by mohlo být v některých případech neefektivní<sup>15</sup>, což by vedlo k rozšíření kalibrační fáze a přeneseně i zvýšení uživatelské nepohodlí.

Z testů tedy vyplynulo, že výhodnější bude vydat se cestou relativního určování polohy. Tato metoda postrádá veškeré výše zmíněné nevýhody, ale nese s sebou následující specifika (podobná jsou i u klasické počítačové myši):

### **Dočasné vypnutí snímání**

Na rozdíl od absolutního určování polohy, je zde potřeba dodefinovat jedno gesto navíc, označme ho *zvednutí myši z podložky*. Toto gesto u normální počítačové myši použijeme většinou, když s ní „dojedeme“ na kraj podložky a potřebujeme získat prostor na její další pohyb stejným směrem. Zvednutím myši z podložky znemožníme jejímu čidlu snímat pohyb, dokud ji opět nepoložíme. Podobně v tomto případě se lehce stane, že se dostaneme na kraj zorného pole kamery. A ze stejných důvodů jako u myši, potřebujeme řící systém, že následující pohyb má ignorovat, abychom si přesunem ruky mohli vytvořit dost místa pro další pohyby.

### **Akcelerace počítačového kurzoru**

U relativního určování polohy je potřeba nastavit správný poměr mezi změnou pozice na snímku kamery a promítnuté změny pozice na displeji počítače. Pokud bude tento vztah nesprávný, může se lehce stát, že se bude uživatel každou chvíli dostávat mimo zorné pole kamery, nebo naopak nebude mít dostatečnou přesnost (např. při snaze kliknout do cílové oblasti menších rozměrů). Proto je obzvláště vhodné zavést tzv. *akceleraci pohybu kurzoru myši*. Takto bude daný poměr mezi změnami proměnný a to tak, že pokud uživatel bude konat pomalé změny pozice, bude se počítačový kurzor posouvat po malých kouscích. Naopak pokud bude činit rychlé změny pozice, bude se počítačový kurzor posouvat po větších dílcích. Opět se jedná o funkci známou i u běžné počítačové myši.

## **5.3 Určování pozice kurzoru z pozice LED diod**

Ve finální fázi byl spolu s přechodem k relativnímu určování pozice pozměněn způsob odvození pozice kurzoru ze snímaných dat. Konkrétněji tuto úlohu převzala dioda umístěna na palci. Pohyb jejího *referenčního bodu* je tedy

---

<sup>15</sup> Vzdálenosti uživateli ruky od kamery, případně její rozměry se mohou mezi jednotlivými uživateli lišit. Proto může být předdefinovaná oblast zbytečně malá, či příliš velká.

sledován a změny patřičně promítány do změn počítačového kurzoru, přičemž v okamžiku, kdy se diody dostanou mimo záběr kamery, se pohyb kurzoru zastaví.

Akcelerace kurzoru se řeší tak, že se vezme v potaz vektor pohybu referenčního bodu diody palce  $\vec{v}_{rb}$ , který se získá rozdílem pozic příslušného bodu na snímku předchozím  $P_{n-1}$  a nyníjším  $P_n$ , a sice

$$\vec{v}_{rb} = (v_{rb_1}, v_{rb_2}) = (P_n - P_{n-1})$$

Z něj vytvoříme vektor pohybu počítačového kurzoru  $\vec{v}_{pk} = (v_{pk_1}, v_{pk_2})$  následujícím způsobem

$$\vec{v}_{pk} = M_a \times \vec{v}_{rb}$$

kde  $M_a$  označuje koeficient akcelerace, který se získá z následujícího vztahu

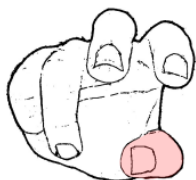
$$M_a = K_a \times |\vec{v}_{rb}|^2 + K_r \times |\vec{v}_{rb}|$$

Jedná se o kvadratickou funkci s nulovým absolutním členem, koeficientem kvadratického členu  $K_a$  (značícím míru akcelerace pohybu kurzoru) a koeficientem lineárního členu  $K_r$  (značícího rychlost pohybu kurzoru myši). Koeficienty  $K_a$  a  $K_r$  budou standardně nastaveny na nejvýhodnější, empiricky naměřené, hodnoty, přičemž je bude moci uživatel upravit.

## 5.4 Dočasné vypnutí pohybu kurzoru

Dočasné vypnutí pohybu kurzoru následuje bezprostředně po detekci určitého gesta palce. Toto gesto spočívá v pohybu palce směrem do dlaně, kdy se dostává dioda vlevo (vztaženo k obrázku 5-B) od společného těžiště. V okamžiku, kdy nastane situace, že dioda palce protne svislici procházející těžištěm, se pohyb kurzoru vypne do doby, než se dioda palce vrátí. Nepředpokládá se slučitelnost s akcemi pro stisknutí levého či pravého tlačítka.

Obrázek 5-B ukazuje příslušné gesto.



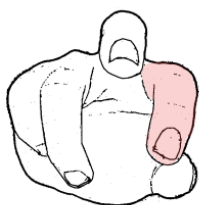
**Obrázek 5-B** Schéma držení ruky pro indikaci zvednutí myši z podložky.

## 5.5 Stisknutí levého tlačítka

Gesta, která jsou vyhodnocena jako stisk tlačítka, vychází z imitace přirozeného pohybu prstu při poklepání např. na stůl a to pro levé tlačítko poklepání ukazovákem, pro pravé pak prostředníkem.

Stisknutí levého tlačítka je tedy indikováno pohybem ukazováku směrem dolů, kdy je vyhodnocena poloha diody ukazováku vůči těžišti trojúhelníku tvořeného diodami. V okamžiku, kdy se dioda ukazováku nachází pod těžištěm, je toto bráno jako stisknutí levého tlačítka myši. Poté, co se zpětným pohybem ukazováku dostane příslušná dioda nad těžiště, je toto vyhodnoceno jako uvolnění tlačítka.

Obrázek 5-C ukazuje příslušné gesto.

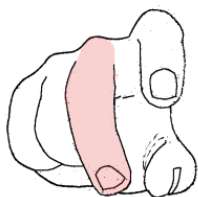


**Obrázek 5-C** Schéma držení ruky pro indikaci stisku levého tlačítka.

## 5.6 Stisknutí pravého tlačítka

Stisknutí pravého tlačítka je realizováno obdobně, jako stisknutí levého tlačítka - je indikováno pohybem prostředníku směrem dolů, kdy je vyhodnocena poloha diody prostředníku vůči těžišti trojúhelníku tvořeného diodami. V okamžiku, kdy se dioda prostředníku nachází pod těžištěm, je toto bráno jako stisknutí pravého tlačítka myši. Poté, co se zpětným pohybem prostředníku dostane příslušná dioda nad těžiště, je toto vyhodnoceno jako uvolnění tlačítka.

Obrázek 5-D ukazuje příslušné gesto.



**Obrázek 5-D** Schéma držení ruky pro indikaci stisku pravého tlačítka.

# Kapitola 6

## Programátorská dokumentace

Jako cílová platforma softwarové části rozhraní TriLED byl zvolen operační systém **Microsoft Windows XP**. V průběhu vývoje byl brán zřetel na eventuální budoucí převod na jiný operační systém. Tento aspekt byl zohledněn při volbě programátorských knihoven, kdy byly použity přednostně ty, u nichž existují verze pro více druhů nejpoužívanějších operačních systémů pro osobní počítače. Programovací jazyk je **C++**. Pro vývoj softwarové části bylo využito vývojové prostředí **Microsoft Visual Studio 2008**. Pro překlad byl využit integrovaný překladač. Kód je tedy přizpůsoben tomuto prostředí a využívá/předpokládá jeho přítomnost; zároveň jsou na přiloženém CD přítomny veškeré soubory projektu potřebné pro jeho editaci a překlad v použitém vývojovém prostředí.

### 6.1 Podpůrné knihovny

Některé konkrétní podproblémy ze SW části projektu již byly nejméně jednou řešeny jinými autory. Některá z takovýchto známějších řešení jsou k dispozici ve formě volně šiřitelných programových knihoven. Proto se o funkčnost částí projektu, přímo nevyžadující originální autorovu implementaci, starají knihovny třetích stran. Při hledání vhodných kandidátů byl kladen důraz primárně na nabízenou funkčnost, na maximálně flexibilní licenci (pod níž jsou knihovny dostupné) a v neposlední řadě míra přenosnosti mezi různými platformami.

Oblasti implementace projektu, u kterých se ukázalo být výhodné využít funkcí cizích knihoven, jsou následující:

- **Získávání snímků z webové kamery**
- **Pokročilé funkce počítačového vidění** (u kterých není jejich implementační návrh předmětem zkoumání této práce)
- **Generování a obsluha prvků GUI aplikace**

Následuje seznam použitých knihoven třetích stran.

### **videoInput<sup>16</sup>**

Knihovna zapouzdřující komunikaci s webovou kamerou. Poskytuje jednoduché programátorské rozhraní, které zároveň rozsahem svých funkcí plně vyhovuje pro účely rozhraní TriLED. Obecně poskytuje velmi rychlý přístup k snímaným obrázkům z kamery. Zároveň umožňuje pohodlným způsobem nastavit parametry snímání (např. jas, kontrast, apod.). [6]

### **OpenCV<sup>17</sup>**

Knihovna OpenCV obsahuje velmi rychlé implementace běžně používaných algoritmů a datových struktur v oblasti počítačového vidění. Přínos je znásoben optimalizacemi pro instrukční sady víceméně všech novějších procesorů od firmy **Intel**, pokud je samozřejmě procesor v cílovém počítači přítomen.

Z návrhu plyne, že jádrové datové struktury budou víceméně vlastní, stejně tak algoritmy pro jejich zpracování. Nicméně při potřebě uplatnit složitější a již zavedené postupy je použití této knihovny velmi výhodné. [5]

### **wxWidgets<sup>18</sup>**

Knihovna poskytující nástroje pro vykreslování *grafického uživatelského rozhraní*. Mezi její přednosti patří přenosnost mezi vícero operačními systémy. Zároveň je její používání velmi intuitivní. Celý proces návrhu GUI je ulehčen dostupností tzv. *designérů*<sup>19</sup> pro návrh vzhledu rozhraní pro wxWidgets. Při tvorbě softwarové části byl využit OpenSource designer **wxFormBuilder**<sup>20</sup>. Takto byla podstatným způsobem urychlena tvorba části

---

<sup>16</sup> Viz internetové stránky knihovny na adrese <http://muonics.net/school/spring05/videoInput/>

<sup>17</sup> Viz internetové stránky knihovny na adrese <http://opencv.willowgarage.com/wiki/>

<sup>18</sup> Viz internetové stránky knihovny na adrese <http://wxwidgets.org/>

<sup>19</sup> Vývojová prostředí pro návrh vzhledu GUI.

<sup>20</sup> Viz internetové stránky programu na adrese <http://wxformbuilder.org/>

kódu starajícího se o rozmístění GUI komponent, který by musel být jinak tvořen ručně. [7]

## 6.2 Architektura softwarové části

Celý projekt je (v souladu s návrhem) rozdělen do logických modulů s uspořádáním vyobrazeném na schématu 6-A, přičemž schéma interakce spodních čtyř modulů je zachycena na obrázku 2-A.



**Schéma 6-A** Schéma rozvržení implementace do přírodních částí.

Konkrétní implementační detaily lze vyčíst z okomentovaných zdrojových kódů (které jsou k nalezení na příloženém CD), případně z automaticky vygenerované referenční příručky (generované nástrojem **Doxygen**<sup>21</sup> a k nalezení tamtéž). Zde jsou probrány hlavní myšlenky, stěžejní použité datové struktury a funkce.

Následuje rámcové seznámení s použitou implementací.

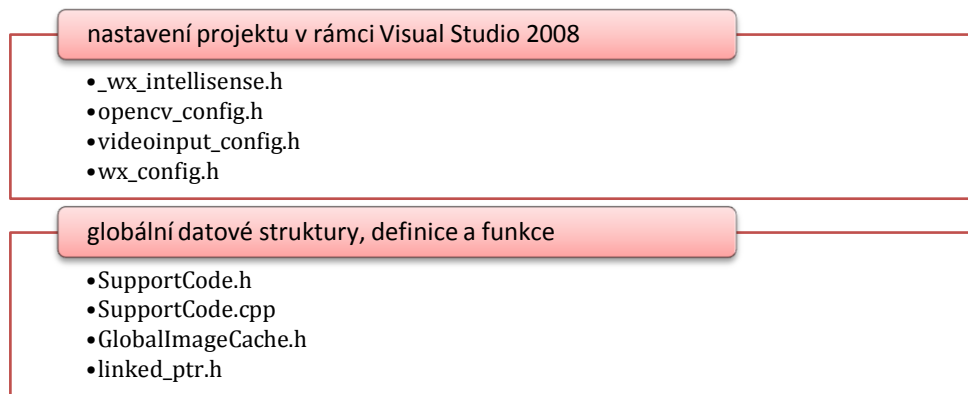
### Podpůrný kód

Do této sekce patří poměrně široká škála souborů, které konfigurují vlastnosti projektového souboru vývojového prostředí **Visual Studio 2008**, definují podpůrné datové struktury, pomocné funkce a globální definice. Organizace tohoto segmentu je vystihnuta na schématu 6-B.

---

<sup>21</sup> Viz internetové stránky programu na adrese <http://www.doxygen.org/>





**Schéma 6-B** Rozložení zdrojových souborů v rámci sekce „Pomocný kód“.

Soubor `_wx_intellisense.h` obsahuje kód, který je způsobem, jak přimět systém automatického dokončování **Intellisense** vývojového prostředí **Visual Studio 2008** k indexování jmen tříd a funkcí z knihovny `wxWidgets`. Zbylé soubory, končící na `_config.h`, slouží k přilinkování příslušných souborů `*.lib` příslušných knihoven.

Zmiňované podpůrné datové struktury, pomocné funkce a globální definice jsou vedeny v souborech `SupportCode.cpp` a `SupportCode.h`.

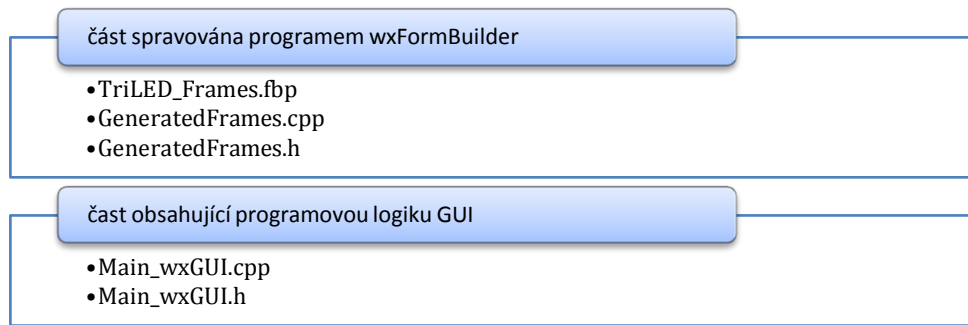
Soubor `GlobalImageCache.h` definuje a inicializuje objekty pojímající grafické popisky explicitně načítané z příložených souborů.

Soubor `linked_ptr.h` obsahuje implementaci tzv. *chytrého ukazatele*<sup>22</sup>, což je objekt, který navenek poskytuje operace běžné pro ukazatel v jazyce C++, ale interně sleduje, jak je s ním nakládáno. Výsledkem je, že se programátor nemusí starat o uvolnění paměti dynamicky přidělené v rámci objektu, na který se tento chytrý ukazatel odkazuje. Pokud chytrý ukazatel zjistí, že již není potřeba, zavolá příslušný destruktork a všechnu paměť uvolní (přehledný úvod do problematiky lze nalézt v [8]). Takto je v zásadě suplována funkce tzv. *garbage collectoru*, známého například z jazyku **Java**, či **C#**.

## Grafické uživatelské rozhraní

Rozložení zdrojových souborů modulu GUI je pro přehlednost vyobrazeno na schématu 6-C.

<sup>22</sup> Konkrétně jde o implementaci od autora jménem Yonat Sharon, který ji nabízí zdarma ke stažení/použití na svých stránkách na adrese [http://ootips.org/yonat/4dev/linked\\_ptr.h](http://ootips.org/yonat/4dev/linked_ptr.h)



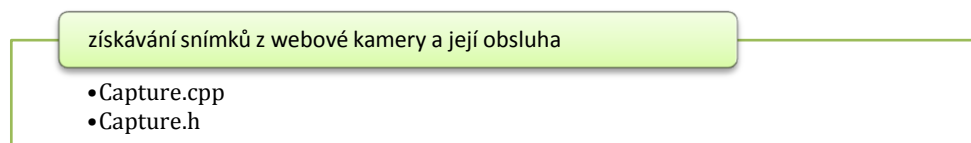
**Schéma 6-C** Rozložení zdrojových souborů v rámci modulu GUI.

Soubory **GeneratedFrames.cpp** a **GeneratedFrames.h** jsou generovány programem **wxFormBuilder** z informací uložených v příloženém souboru **TriLED\_Frames.fbp**. Tento soubor obsahuje informace o rozložení prvků GUI a je upravovatelný právě zmíněným programem. Generované soubory pak obsahují hierarchickou objektovou reprezentaci použitých GUI komponent, které jsou dále využívány.

Soubory *části obsahující programovou logiku GUI* pojmají funkčnost nutnou pro propojení jednotlivých GUI komponent s danými akcemi. Dále je zde obsažena významná třída **TLED\_InnerEngine**, jejíž metody jsou volány v rámci akcí GUI. Z této třídy jsou volány potřebné funkce modulů *snímání*, *detekce*, *interpretace* a *presentace*. Obecně tato třída zodpovídá za zmiňované zastřešení uvedených modulů tím, že v daném pořadí volá jejich stěžejní metody a předává mezi nimi jejich meziprodukty.

## Modul Snímání

Následuje schéma 6-D, kde je vyobrazen seznam zdrojových souborů.

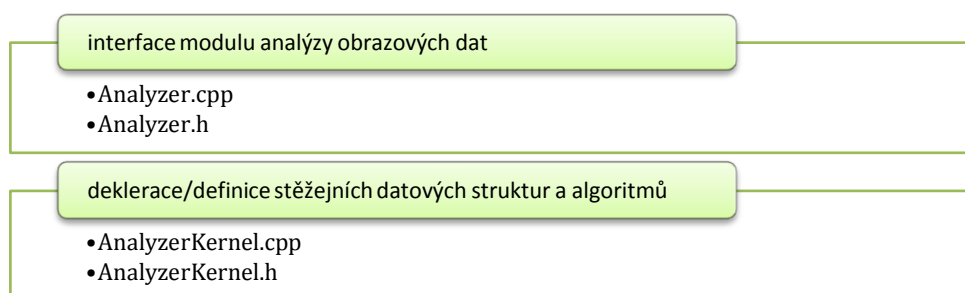


**Schéma 6-D** Rozložení zdrojových souborů v rámci modulu snímání.

V tomto modulu se za použití funkcí knihovny `videoInput` získávají v pravidelných intervalech snímky z připojené webové kamery. Ty se umístí do rámcové struktury `TLED_CapturedFrame`; reprezentovaná chytrým ukazatelem `TLED_CapturedFrame_SmartPointer` je připojena na jeden konec oboustranné fronty implementované třídou `TLED_CapturedFrameQueue`; z druhého konce této fronty pak odebírá rámcové struktury třída `TLED_InnerEngine` (z modulu GUI) a předává je modulu detekce. Interface daného modulu poskytuje hlavní třída `TLED_CaptureModule`, která obsahuje zmíněný kontejner na získané snímky. Dále nabízí funkce pro nastavení vlastností připojené kamery, zahájení a zastavení získávání snímků a stěžejní metodu `getNextFrame`, která předá nejnovější získaný snímek ve formě chytrého ukazatele na rámcovou strukturu `TLED_CapturedFrame`.

## Modul Detekce

Doprovodné schéma 6-E znázorňuje seznam zdrojových souborů.



**Schéma 6-E** Rozložení zdrojových souborů v rámci modulu detekce.

Tento modul se drží poznatků uvedených v kapitole 3 a uvádí je do praxe. Hlavní třídou tohoto modulu je třída `TLED_AnalyzerModule`, která nabízí interface potřebný pro provedení jednotlivých kroků kalibrační fáze a při jejich volání postupně stěradá jednotlivá potřebná kalibrační data. Z těchto dat si graduálně buduje (v souladu s postupem v kapitole 3) *barevný profil* vyjádřený třídou `TLED_ColorProfile` (s *kyblíky* vyjádřenými strukturou `TLED_ColorProfile_Bucket`). K uložení snímků při kalibraci pozadí a scény s rozsvícenými LED diodami je využita kontejnerová třída `TLED_TrainingImageBuffer`. Ústřední metodou hlavní třídy je metoda `parseFrame`. Tato metoda podrobí snímek z kamery (v podobě chytrého ukazatele na rámcovou strukturu `TLED_CapturedFrame`) obdrženy

z modulu snímání analýze. Za použití funkcí knihovny **OpenCV**, převede obrazová data do formátu **IplImage** (struktura na uchovávání obrazových dat v rámci knihovny **OpenCV**) a podrobí je detekčnímu procesu (popsanému v kapitole 3). V tomto detekčním procesu je s výhodou využita další funkce knihovny **OpenCV**, a sice funkce **cvFindContours**, která slouží k nalezení komponent spojitosti v rámci detekčního procesu. Výsledkem detekčního procesu metody **parseFrame** je struktura **TLED\_LEDInfoSet**, obsahující referenční body jednotlivých diod a jiné informace, které potřebuje ke své práci modul interpretace; jemuž je struktura připravena na předání.

## Modul Interpretace

Následuje schéma 6-F, kde je vyobrazen seznam zdrojových souborů.



**Schéma 6-F** Rozložení zdrojových souborů v rámci modulu interpretace.

Tento modul je vyjádřen třídou **TLED\_InterpretationModule**, která poskytuje prostředky na analýzu uživatelských gest na základě dodaných referenčních bodů z modulu detekce, především pak nabízí stěžejní metodu **addParsedLEDState**. Tato metoda obdrží strukturu **TLED\_LEDInfoSet**, ta je podrobena zkoumání vztahů referenčních bodů jednotlivých diod a společného těžiště těchto bodů. Vyhodnotí splnění podmínek pro uznání jednotlivých gest (definovaných v kapitole 5) a na základě vyhodnocených informací je vytvořena struktura **TLED\_PresentationInfo**, obsahující informace o relativní změně pozice počítačového kurzoru, stavu jednotlivých tlačítek (zmáčknuto nebo nezmáčknuto) a zdali uživatel neprovedl *gesto zvednutí myši z podložky*. Tato struktura je pak připravena na další zpracování modulem prezentace.

## Modul Prezentace

Schéma 6-G uvádí soupis zdrojových souborů modulu.

prezentace intepretovaných gest v podobě akcí počítačového kurzoru

- Presentation.cpp
- Presentation.h

**Schéma 6-G** *Rozložení zdrojových souborů v rámci modulu prezentace.*

Stěžejní třídou tohoto modulu je třída **TLED\_PresentationModule**, která svou funkčnost shrnuje do metody **presentData**. Tato metoda dostane jako parametr strukturu **TLED\_PresentationInfo**, která byla připravena modulem interpretace. Na základě obsažených informací vyvolá patřičné události počítačového kurzoru myši v rámci OS Windows[9]. K vyvolání těchto událostí využívá systémového volání **mouse\_event**.

# Kapitola 7

## Uživatelská dokumentace

V této kapitole je uveden návod na zacházení s aplikací TriLED určený pro uživatele.

### 7.1 Systémové požadavky

Program vyžaduje ke svému spuštění počítač s nainstalovaným operačním systémem **Microsoft Windows XP** a s připojenou **webovou kamerou**, pro kterou byly nainstalovány ovladače dodané jejím výrobcem. Takt procesoru počítače by měl být alespoň **1GHz**; vícejádrový procesor je přínosem. V průběhu kalibrační fáze potřebuje mít program k dispozici pro bezproblémový chod až **170MB** operační paměti.

### 7.2 Instalace

Instalace v pravém smyslu není potřeba. Stačí rozbalit přiložený soubor **TriLED.zip** do cílového adresáře (lze použít jakýkoliv dekomprimační nástroj, podporující daný formát komprese). Je pouze nutné se ujistit, že byly rozbaleny veškeré soubory z archivu tak, že byla zachována jejich vnitřní adresářová struktura.

### 7.3 Spuštění

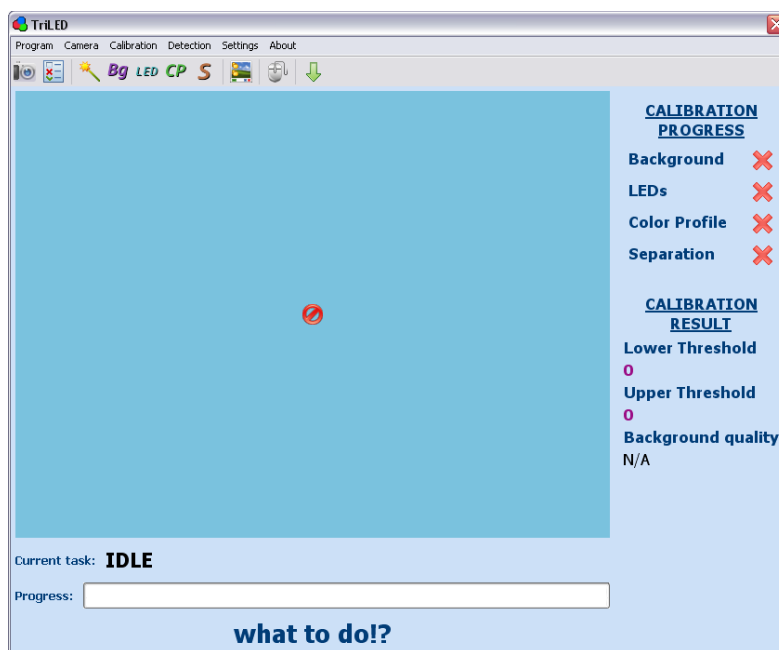
V adresáři, do něž byl rozbalen přiložený archiv **TriLED.zip** spustíme soubor programu **TriLED.exe**. Pro spuštění nejsou vyžadovány žádné dodatečné úkony.

## 7.4 Fyzické umístění kamery

Kameru je vhodné umístit zhruba 0,5m od oblasti, ve které se bude pohybovat uživatelská ruka a to tak, že uživatel může rukou pohybovat v celém zorném poli kamery. Zároveň je potřeba se ujistit, že se v cestě výhledu kamery nenachází nějaká překážka.

## 7.5 Manipulace s aplikací

Po spuštění aplikace se zobrazí hlavní okno GUI (viz obrázek 7-A), zároveň je v *notifikační oblasti hlavního panelu* systému **Windows** zobrazena ikona (viz obrázek 7-B), u které lze vyvolat kontextová nabídka s akcemi GUI aplikace **TriLED**. Hlavní okno lze převést do tzv. *režimu práce na pozadí*, kdy není toto okno viditelné, ale aplikace **TriLED** pracuje nerušeně dál (přepnout do tohoto režimu lze volbou hlavního menu *Program* → *Minimize to system tray*); hlavní okno lze opět vrátit do původního stavu zvolením příslušné akce z kontextové nabídky ikony v notifikační oblasti.

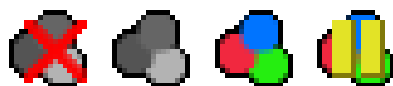


**Obrázek 7-A** Hlavní programové okno aplikace *TriLED*, zobrazené uživateli po spuštění.



**Obrázek 7-B** *Ikona programu TriLED zobrazená v notificační části hlavního panelu systému Windows.*

Tato ikona zároveň signalizuje momentální stav, v němž se aplikace nachází. Může nabývat čtyř podob v závislosti na stavu; tyto jsou znázorněny na obrázku 7-C. Zleva doprava znamenají „potřeba kalibrace“, „nakalibrováno, kurzor myši není ovládán“, „kurzor myši je ovládán“ a „snímání změny pohybu je dočasně vypnuto“ (v důsledku nepřítomnosti diod na scéně).

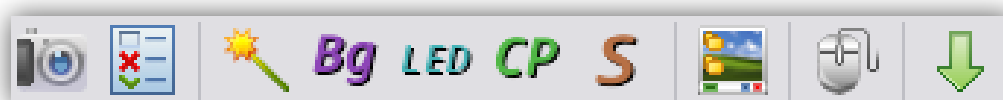


**Obrázek 7-C** *Možné podoby ikony programu TriLED zobrazené v notificační části hlavního panelu systému Windows.*

Typická práce s programem se skládá z ustáleného sledu akcí, které zvolí a nastaví webovou kameru a vedou ke kalibrování interního detekčního algoritmu a následného „ovládání kurzoru myši“. Poté již ovládá uživatel kurzor myši gesty konanými v zorném poli kamery rukou se světelnou rukavicí.



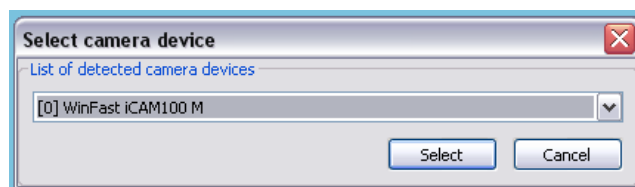
K usnadnění používání aplikace je přítomna tzv. *lišta nástrojů*, která obsahuje tlačítka napojená na nejčastěji používané akce (viz obrázek 7-D). Jednotlivá tlačítka znamenají (zleva doprava): „vybrat kameru“, „nastavit připojenou kameru“, „zahájit poloautomatický proces kalibrace“, „zahájit 1. fázi kalibrace“, „zahájit 2. fázi kalibrace“, „zahájit 3. fázi kalibrace“, „zahájit 4. fázi kalibrace“, „zobrazit náhled na detekční proces“, „ovládnout kurzor myši“ a „přepnout do režimu práce na pozadí“.



**Obrázek 7-D** *Nástrojová lišta osazená tlačítky (s grafickými popisky) nejpoužívanějších akcí.*

### 7.5.1 Zvolení kamery

Kliknutím na příslušné tlačítko na liště nástrojů, případně zvolením patřičné akce v hlavním menu aplikace *Camera* → *Select device*, se zobrazí dialog se seznamem připojených zařízení (viz obrázek 7-E). V tomto seznamu uživatel zvolí kameru, kterou chce využívat pro snímání gest; volbu potvrdí stisknutím tlačítka *Select* nebo zruší stisknutím tlačítka *Cancel*.

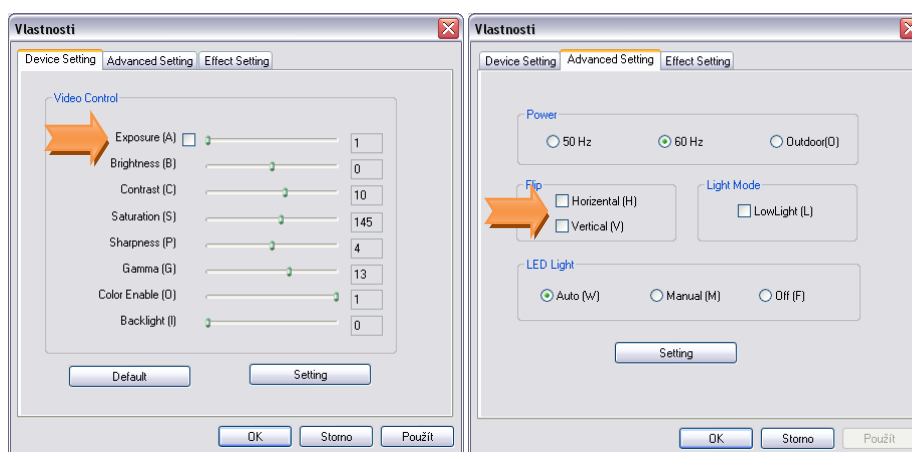


**Obrázek 7-E** *Dialog se seznamem rozpoznaných připojených kamer.*

Kamera bude odmítnuta, pokud nepodporuje rozlišení 640x480 obr. bodů. Není se nutné toho obávat, toto rozlišení dnes zvládají i všechny běžně dostupné webové kamery, řadí se k těm „levnějším“.

## 7.5.2 Nastavení připojené kamery

Kliknutím na příslušné tlačítko na liště nástrojů, případně zvolením patřičné akce v hlavním menu aplikace *Camera* → *Configure current device*, se zobrazí dialog s nastaveními kamery. Tento dialog se může lišit kamera od kamery, různý může taktéž být seznam parametrů, které lze měnit, pravděpodobně však bude vypadat přibližně jako na obrázku 7-F.



**Obrázek 7-F** Dialog nastavení parametrů připojené kamery. Vlevo je zvýrazněno nastavení expozice, upravo je zvýraznění nastavení převrácení obrazu.

Pro nejvhodnější podmínky snímání gest, je vhodné nastavit především tzv. *čas expozice* (anglicky **exposition time**) a tzv. *míru zesílení* (anglicky **gain**) na co možná nejmenší hodnoty. Dále, pokud má kamera funkci automatického dynamického nastavování některé z těchto veličin, je záhodné tuto funkci vypnout. Takto se zajistí, že světelné diody budou dosti vynikat oproti pozadí, které bude díky nízkému času expozice velmi tmavé.

Je nutné také nastavit funkci *horizontálního převrácení obrazu* a ujistit se, že funkce *vertikálního převrácení obrazu* je vypnutá. Jde o naprosto standardní

nastavitelné parametry každé běžné webkamery a jejich nastavení je potřebné pro správnou práci programu. Viz obrázek 7-F.

### 7.5.3 Kalibrace detekčního algoritmu

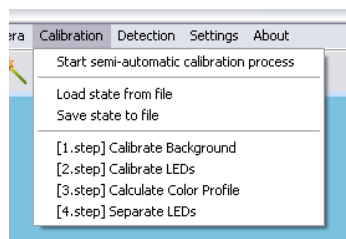
Kalibraci lze provést „poloautomaticky“ nebo manuálně „po částech“. Kalibrační proces se skládá ze 4 fází:

- 1.fáze** Analýza scény, která obsahuje pouze pozadí (resp. neobsahuje rozsvícené diody)
- 2.fáze** Analýza scény, na které se vyskytují všechny tři rozsvícené LED diody
- 3.fáze** Výpočet tzv. *barevného profilu*
- 4.fáze** Identifikace barevných charakteristik jednotlivých diod za uživatelské asistence

Způsob kalibrace „po částech“ je varianta, kdy si uživatel odděleně provede jednotlivé fáze kalibrace; v případě potřeby i opakovaně, pro dosažení nejlepších výsledků. Je nutné brát v úvahu, že provedení kalibrační fáze 3 vyžaduje předešlé provedení 1. a 2. fáze. Dále provedení kalibrační fáze 4 vyžaduje hotové provedení 3. fáze. Vyvolání jednotlivých fází kalibrace se provede stisknutím relevantního tlačítka na nástrojové liště aplikace, nebo zvolením položky hlavního menu aplikace

*Calibration* → [X.step](...)

kde *X* označuje číslo fáze a (...) značí popisek příslušné fáze (viz obrázek 7-G).



**Obrázek 7-G** Část hlavního menu věnována kalibraci.

V případě „poloautomatického“ způsobu kalibrace jsou v návazném sledu provedeny jednotlivé fáze a to vždy jednou, dodatečně pak může uživatel provést manuálně fáze kalibrace, s jejichž výsledky není spokojen. Tento způsob se doporučuje pro prvotní kalibraci. Proces poloautomatické kalibrace se spustí stisknutím relevantního tlačítka na nástrojové liště aplikace, nebo zvolením položky hlavního menu aplikace

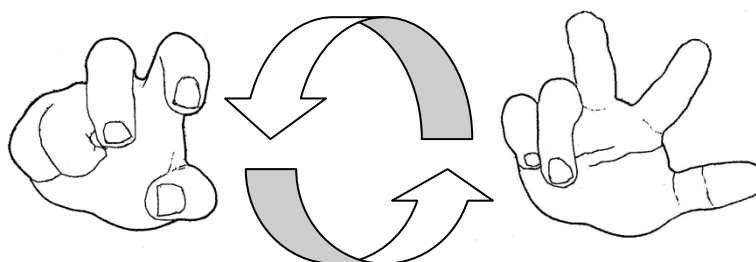
#### *Calibration → Start semi-automatic calibration process*

Výsledky kalibračního procesu lze uložit do souboru volbou v hlavním menu *Calibration → Save state to file*. Takto uložené výsledky lze opět nahrát z takto uloženého souboru pomocí volby hlavního menu *Calibration → Load state from file*. Je dobré poznamenat, že výsledky kalibračního procesu se vztahují vždy k fyzické pozici kamery, konkrétním světelným podmínkám snímané scény, či vlastnostem použitých LED diod. Nelze tedy předpokládat, že bude předešle uložený stav kalibrace stejně efektivní za všech podmínek.

Následují pokyny a poznámky k jednotlivým fázím kalibrace. Každá takováto fáze kalibrace (kromě fáze 3, která proběhne bez nutnosti účasti uživatele) vyčká před svým započítáním na signál uživatele ve formě stisku klávesy. Po stisknutí klávesy následuje analýza určitého počtu snímků. Průběh je uživateli sdělován pomocí indikátoru průběhu kalibrace.

#### **1.fáze – „Kalibrace pozadí“**

V této fázi se snímá a analyzuje *pozadí*, tedy se nesmí na scéně objevit rozsvícené LED diody. Doporučuje se během analýzy snímků držet ruku s vypnutými diodami v zorném poli. Předejde se tak případným následným problémům při detekci. Dále je vhodné rukou s LED diodami pohybovat, případně provádět gestikulace vyobrazené na obrázku 7-H.



**Obrázek 7-H** *Doporučené gestikulace v 1. a 2. fázi kalibračního procesu.*

Po dokončení této fáze se zobrazí úspěšnost kalibrace v podání procentuálního zastoupení obrazových bodů, jež neprošly prahovou segmentací (100 % je nejlepší výsledek, 0 % nejhorší). V případě nutnosti zvýšte práh jasové segmentace (viz kapitola 7.5.4) anebo fázi opakujte.

## **2.fáze – „Kalibrace LED diod“**

V této fázi se snímá a analyzuje scéna, na které se musí vyskytovat rozsvícené LED diody. Během analýzy snímků tedy držte ruku se zapnutými diodami v zorném poli. Pro maximálně kvalitní výsledek kalibrace, rukou s LED diodami pohybujte v rámci zorného pole a provádějte gestikulaci vyobrazené na obrázku 7-H.

## **3.fáze – „Výpočet barevného profilu“**

Tato fáze probíhá bez účasti uživatele. Během této fáze se spočítá tzv. *barevný profil*. Pro uživatele jsou podstatné pouze získané hodnoty tzv. *dolního* a *horního prahu* (v programu se vyskytují jako *Lower threshold* a *Upper threshold*).

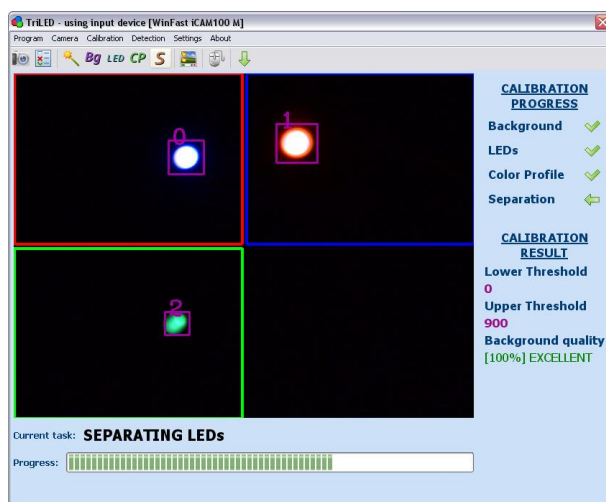
Obě hodnoty jsou v celočíselném rozmezí 0 až 900. Hodnota dolního prahu by měla být co nejmenší (nejlépe nulová). Hodnota horního prahu by měla být co největší (nejlépe rovná 900). V každém případě musí pro obě hodnoty platit vztah

$$\textit{Lower threshold} \ll \textit{Upper threshold}$$

V opačném případě kalibrační proces selhal a diody nelze spolehlivě detekovat, což může být v důsledku málo výrazných LED diod, přsvíceného pozadí nebo případně špatně nastaveného prahu jasové segmentace. Zkuste tedy upravit parametry kamery (viz kapitola 7.5.2), pozměnit dodatečná nastavení programu TriLED (viz kapitola 7.5.4) či upravit světelné podmínky na scéně a kalibrační fáze 1-3 opakovat.

#### 4.fáze – „Separace LED diod“

V této fázi budete požádáni o držení rozsvícených LED diod ve vyobrazených regionech zorného pole. Pokud možno lehce diody natáčejte do stran; přitom ovšem neopouštějte vyhrazené regiony. Viz obrázek 7-I.



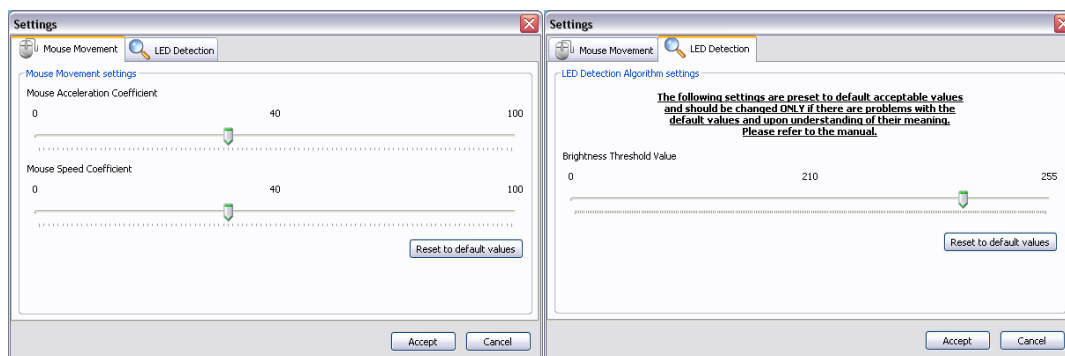
Obrázek 7-I Probíhající 4. fáze kalibrace.

Po úspěšném dokončení 4. fáze je již vše připraveno pro detekci diod „naživo“.

## 7.5.4 Dodatečné nastavení programu TriLED

Dodatečná nastavení programu TriLED se nacházejí na konfiguračním dialogu (viz obrázek 7-J), který lze zobrazit volbou hlavního menu

*Settings* → *Program settings*.



**Obrázek 7-J** Dialog dodatečných nastavení programu TriLED.

Dialog dodatečných nastavení nabízí možnost změnit hodnoty *akcelerace pohybu kurzoru myši* (*Mouse Acceleration Coefficient*), *rychlosti pohybu kurzoru myši* (*Mouse Speed Coefficient*) a *prahu jasové segmentace* (*Brightness Threshold Value*). Každou hodnotu lze vrátit na tzv. *defaultní* (implicitně přednastavenou) hodnotu stisknutím příslušného tlačítka *Reset to default values*.

### Akcelerace pohybu kurzoru myši

Akcelerace umožňuje pomalými pohyby ruky vykonávat drobné změny polohy počítačového kurzoru a rychlými větší.

Rozsah hodnot je 0 až 100. Defaultní hodnota je 40. Čím je hodnota vyšší, tím víc je rychlejší pohyb ruky vyhodnocen jako větší ušlý úsek počítačového kurzoru a naopak.

Zmenšete, pokud se vám zdá, že kurzor při pohybu skáče po velkých úsecích. Zvětšete, pokud se vám zdá, že s kurzorem nejste schopni na jeden „zátah“ ujít potřebnou vzdálenost.

## **Rychlost pohybu kurzoru myši**

Rychlost pohybu kurzoru myši značí, jak moc se kurzor pohne při plynulém pohybu ruky.

Rozsah hodnot je 0 až 100. Defaultní hodnota je 40. Čím je hodnota vyšší, tím se počítačový kurzor pohybuje rychleji a naopak.

Zmenšete, pokud se vám zdá, že je pohyb kurzoru příliš rychlý a naopak.

## **Práh jasové segmentace**

Práh jasové segmentace určuje hladinu jasu obrazových bodů, do které (včetně) budou v kalibrační a detekční fázi obrazové body snímků z kamery ignorovány (tedy automaticky přiřazeny pozadí).

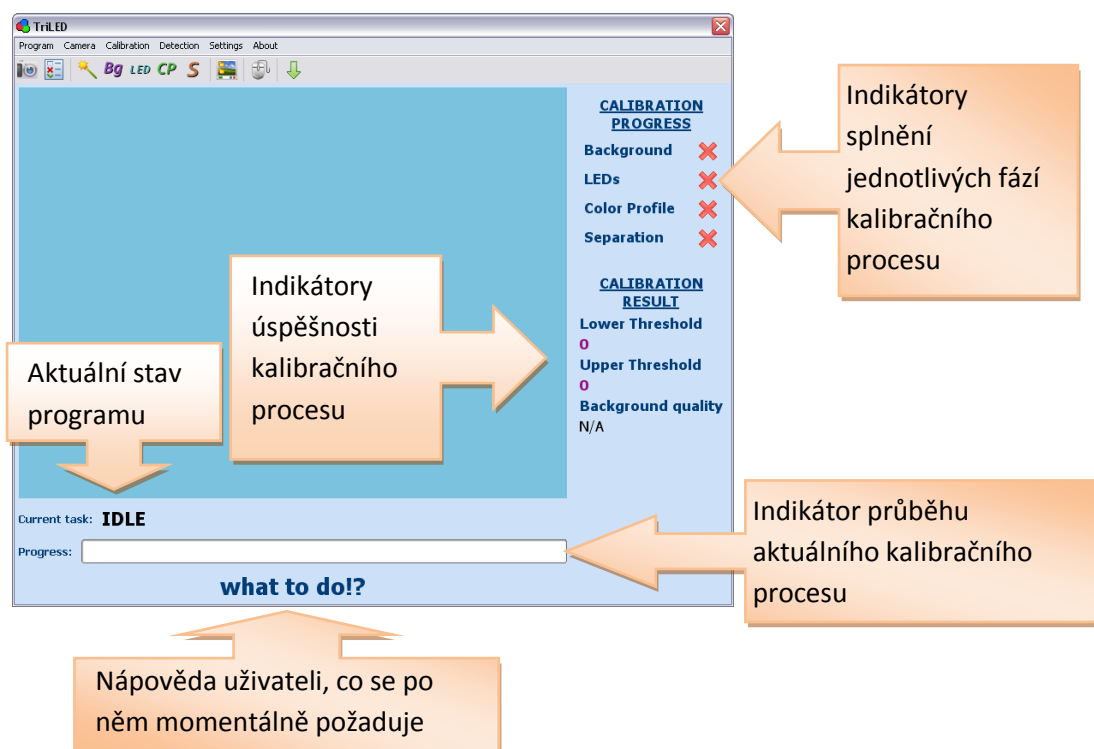
Rozsah hodnot je 0 až 255, defaultní hodnota je 210. Čím je hodnota vyšší, tím větší je šance, že budou obrazové body pozadí správně ignorovány. Příliš velká hodnota naopak může způsobit ignorování obrazových bodů patřících LED diodám.

Doporučuje se ponechat defaultní hodnotu a měnit ji, až na základě explicitní potřeby.



## 7.5.5 Indikátory aktuálního stavu programu

Ke sledování stavu programu, stavu procesu kalibrace a jeho úspěšnosti slouží několik indikátorů, které jsou ilustrovány na obrázku 7-K.



Obrázek 7-K Popsané indikátory stavu programu TriLED.

## 7.5.6 Ovládání kurzoru myši

Po úspěšně provedené kalibraci lze zapnout interpretaci uživatelských gest, neboli „ovládnout kurzor myši“. Před samotným ovládnutím kurzoru, lze zobrazit tzv. *režim náhledu na detekci diod* (volba hlavního menu *Detection* → *Preview LED detection*). Zde jsou uživateli zobrazovány snímky z kamery, na kterých jsou vyznačeny nalezené referenční body LED diod. Pokud je vše v pořádku, lze přistoupit k samotnému ovládání kurzoru myši volbou hlavního menu *Detection* → *Take control of Mouse Cursor*. Nyní již uživatel ovládá počítačový kurzor myši svými gesty (popsanými v kapitole 5). Doporučuje se následně přepnout aplikaci do režimu práce na pozadí (popsaném v úvodu kapitoly 7.5). Jak režim náhledu na detekci diod, tak režim kontroly kurzoru myši, lze vypnout opětovným vyvoláním stejné volby. Jde o tzv. volbu stavovou.

# Kapitola 8

## Závěr

V této kapitole je shrnut stav naplnění vytyčených cílů. Dále se zde pojednává o možnostech budoucího rozšíření.

### 8.1 Naplnění cílů práce

Cílem práce bylo prozkoumat možnosti ovládní kurzoru myši pomocí gest lidské ruky, analyzovat možné postupy, jak gesta ruky přenést do povelů pro akce kurzoru myši, navrhnout a ověřit algoritmy, které analyzují snímky s gesty a nakonec získané poznatky využít při návrhu zařízení a obslužného programu pro detekci a interpretaci gest.

Výsledkem je návrh algoritmu, který ve spojení s HW zařízením (webová kamera a rukavice s LED diodami) dokáže přenést uživatelova gesta na akce kurzoru (pohyb, stisk levého a pravého tlačítka, tažení objektu myši apod.). Podařilo se rovněž vyřešit některé extrémní situace jako výstup rukavice mimo záběr kamery.

Předložené řešení využívá rychlé algoritmy, které zajišťují použitelnost navrženého řešení i na PC s relativně nízkým výkonem při použití běžně dostupných webových kamer.

### 8.2 Diskuze a možnosti budoucího rozšíření

Možnosti dalšího rozvoje či zdokonalení jsou poměrně široké. Lze například zvýšit robustnost použitých kalibračních, detekčních a adaptačních algoritmů tak, aby bylo možné používat zařízení i při relativně špatných podmínkách

scény. Lze také uvažovat o implementaci navrženého software přímo do ovladače kamery.

Je možné rovněž zkoumat možnost použití IR diod a odpovídajícího filtru kamery tak, aby bylo přijímáno pouze infračervené spektrum.

Další možností je použití speciálních reflexních barev místo diod, které by byly nanесeny na jednotlivé prsty či speciální rukavici.

Možnosti uplatnění podobných způsobů ovládání jsou poměrně široké a bezpochyby představují jeden ze směrů, kterými by se ovládání jakýchkoliv elektronických zařízení mohlo ubírat.

# Kapitola 9

## Literatura

- [1] **Sonka M., Hlavac V. a Boyle R.** *Image processing, Analysis and Machine Vision.*: Chapman & Hall, 1994, stránky 112-191.
- [2] *Object Detection Using Colour.*  
**Duffy N., Crowley J. a Lacey G.** : ICPR'00, 2000.
- [3] *Using Histograms to Detect and Track Objects in Color Video.*  
**Duric Z. a Mason M.** : AIPR'01, 2001.
- [4] *A Realtime Object Tracking System using a Color Camera.*  
**Paul G. V., Beach G. J. a Cohen Ch. J.** : AIPR'01, 2001.
- [5] OpenCV Reference Manual. *Open Computer Vision Library.* [Online] Intel.  
<http://opencv.willowgarage.com/wiki>.
- [6] **Watson T.** Komentáře v rámci zdrojových kódů knihovny. *videoInput Homepage.*  
[Online] <http://muonics.net/school/spring05/videoInput/>.
- [7] wxWidgets Reference Manual. *wxWidgets library.* [Online]  
<http://docs.wxwidgets.org/stable>.
- [8] **Sharon Y.** Overview of smart pointers. *Yonat Sharon's Homepage.* [Online]  
<http://ootips.org/yonat/4dev/smart-pointers.html>.
- [9] Mouse Input. MSDN Library. [Online] Microsoft.  
[http://msdn.microsoft.com/en-us/library/ms645533\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/ms645533(VS.85).aspx).

# Příloha A

## Obsah přiloženého DVD

Obsah přiloženého DVD je organizován do následující adresářové struktury:

### `\bin`

Obsahuje archiv se zkompilevaným programem a vším co potřebuje ke svému běhu.

### `\src`

Zde se nacházejí zdrojové kódy aplikace TriLED a soubory potřebné pro práci s nimi ve vývojovém prostředí Visual Studio 2008. V souboru README.TXT je popsán postup kompilace.

### `\doc`

Zde je umístěna referenční programová dokumentace vygenerovaná programem Doxygen.

### `\thesis`

Na tomto místě se nachází text této bakalářské práce ve formátu PDF.